

M^a del Carmen Bona Gracia

Creación de BIOIN ANESTHESOM.
Aplicación en los hospitales Miguel
Servet y Clínico Lozano Blesa de
Zaragoza

Departamento

Medicina, Psiquiatría y Dermatología

Director/es

RUBIO ARANDA, ENCARNACIÓN
RUBIO CALVO, EMILIO

<http://zaguan.unizar.es/collection/Tesis>



Reconocimiento – NoComercial – SinObraDerivada (by-nc-nd): No se permite un uso comercial de la obra original ni la generación de obras derivadas.

© Universidad de Zaragoza
Servicio de Publicaciones

ISSN 2254-7606



Universidad
Zaragoza

Tesis Doctoral

CREACIÓN DE BIOIN ANESTHSOM. APLICACIÓN
EN LOS HOSPITALES MIGUEL SERVET Y CLÍNICO
LOZANO BLESA DE ZARAGOZA

Autor

M^a del Carmen Bona Gracia

Director/es

RUBIO ARANDA, ENCARNACIÓN
RUBIO CALVO, EMILIO

UNIVERSIDAD DE ZARAGOZA
Medicina, Psiquiatría y Dermatología

2019

Creación de BIOIN ANESTHSOM.

**Aplicación en los hospitales
Miguel Servet y Clínico Lozano
Blesa de Zaragoza.**

María del Carmen Bona Gracia

**Dirigida por:
Emilio Rubio Calvo
Encarnación Rubio Aranda**

Dedicatoria

A mis padres y a Esther por haberme forjado como la persona que soy en la actualidad, todos mis logros os los debo a vosotros por ponérmelo tan fácil y motivarme con constancia para alcanzar mis sueños.

A mi familia y amigos por apoyarme y quererme tal y como soy.

A Borja, porque espero que esta creación sólo sea la primera de muchas en común.

Vamos a hacer con el futuro un canto a la esperanza y poder encontrar tiempos cubiertos con las manos los rostros y los labios que sueñan libertad. Somos como esos viejos árboles.

José Antonio Labordeta.

Elogiar la palabra es como elogiarnos a nosotros mismos, porque la palabra es la materia básica para entender lo humano

Fernando Lázaro Carreter.

Agradecimientos

En primer lugar, quiero agradecer a mis directores de tesis Encarnación Rubio Aranda y en especial a Emilio Rubio Calvo por su incansable colaboración y empeño para sacar este trabajo adelante y sobre todo por sus palabras de ánimo cuando pensaba en abandonar.

A Rafael García-Foncillas, indispensable colaborador, ya que con su trabajo se asentaron los cimientos del mío.

A Pablo Torner Pelay porque sin sus conocimientos informáticos esta monografía nunca hubiese salido a la luz.

A Rubén Casans Francés, excelente compañero que siempre me ha tendido su mano sin esperar nada a cambio.

A Borja Rubio García porque gracias a ti comenzó todo.

Prólogo

Desde el comienzo de mi residencia en Anestesiología y Reanimación en el Hospital Clínico Universitario Lozano Blesa de Zaragoza, fui consciente de la importancia de trabajar fomentando la seguridad del paciente, aspecto esencial de la calidad asistencial, intentando en todo momento disminuir al máximo el daño causado por el propio acto médico.

Además de los grandes avances de la Medicina en general y de los avances en la Anestesiología en particular, como nuevos fármacos, nuevas técnicas de anestesia locorregional, dispositivos para la monitorización exhaustiva de los pacientes, etc. siempre he tenido interés en la evaluación de esas técnicas y en conocer la eficacia real obtenida, así como en los mecanismos de control de los que disponemos para optimizar la atención al paciente.

Por este motivo durante los años que duró mi residencia procuré participar en todas las actividades relacionadas con la seguridad del paciente y la calidad asistencial. Participé en el Proyecto de evaluación de la implantación en el Hospital Clínico Universitario Lozano Blesa del “Listado de Verificación de Seguridad Quirúrgica” correspondiente al año 2013 y en los Proyectos de Mejora de la Calidad en SALUD “Introducción en Historia Clínica de la Hoja Informativa de vía aérea difícil” en 2014, “Protocolo de analgesia postoperatoria para la cesárea, basado en el bloqueo guiado por ecografía de los nervios iliohipogástrico e ilioinguinal” en 2015, “Infusión de HES 6% 130/0,4 para la prevención de hipotensión intraoperatoria en cesárea electiva” en 2015, “Consentimientos informados para la realización de técnicas de anestesia locorregional” en 2016 y en el Curso de Seguridad del Paciente en Anestesiología 6ª Edición de SENSAR en 2016.

Conforme fui profundizando más en mi especialidad, comencé a ser consciente de que a pesar de ser una especialidad en la que se trabaja con mucha seguridad y ocurren pocos eventos adversos, no existe obligación ni cultura de declaración de los incidentes, por lo que saber en estos momentos

realmente cómo estamos en el tema de la seguridad, es realmente complicado.

A lo largo de estos años de trayectoria profesional hice un rotatorio externo en el Hospital Universitario Miguel Servet donde tuve la suerte de conocer al Dr. Borja Rubio, igualmente interesado en este tema, y autor de la investigación “Búsqueda de un indicador dinámico de análisis de las complicaciones en Anestesiología y Reanimación: ANESTHSOM” que sintetizando se trataba de aplicar un indicador numérico para analizar y cuantificar los incidentes críticos que ocurrían en mi especialidad. Gracias a él, tuve la oportunidad de relacionarme con la Cátedra de Bioestadística.

Una de las cosas que me ofreció la Cátedra de Bioestadística es integrarme en la investigación para darle una aplicabilidad informática, pues hasta ahora, la aplicación de ANESTHSOM en el análisis hospitalario se realizaba a través de Excel, lo que constituía una labor muy pesada.

En dicha cátedra se estaba implementando, para la planificación y control de la accidentalidad laboral, distintos programas informáticos que se denominaron genéricamente BIOIN OMEGA. Me ofrecieron que me integrara en su equipo informático para poder desarrollar lo que denominaremos BIOIN ANESTHSOM.

ÍNDICE

CAPITULO I INTRODUCCIÓN.....	1
1. Anestesiología. Incidencias y su medición.....	3
1.1 Historia de la anestesiología general.....	4
1.2 Historia de la anestesiología locorregional.....	9
1.3 Incidentes en Anestesiología.....	11
1.4 Seguridad del paciente.....	14
1.4.1. Desde sus comienzos.....	14
1.4.2. En la actualidad.....	16
1.4.3. SENSAR.....	17
1.5 Estadística y su relación con la Anestesiología.....	19
2. ACSOM y su Programa informático BIOIN.....	22
2.1.- ACSOM. Introducción.....	22
2.2.- Concepto de ACSOM.....	23
2.3.- Aplicación informática de ACSOM: Programa BIOIN.....	24
2.3.1. Gestión telemática.....	24
2.3.2. Ejemplo de funcionamiento.....	26
2.4.- BIOIN y su lenguaje de programación.....	30
3. ANESTHSOM.....	38
3.1. Adaptación del método ACSOM a Anestesiología: ANESTHSOM.....	38
3.2. Análisis de la matriz FR x C.....	42
4. JUSTIFICACIÓN Y OBJETIVO	
Justificación.....	49
Objetivo Principal.....	49
Objetivos específicos.....	50
CAPÍTULO II MATERIAL Y METODOS.....	51
1. MATERIAL.....	53
2. MÉTODOS	54
2.1 Metodología ANESTHOM	54
Análisis estadístico del cuerpo de la tabla.....	54
Análisis estadístico de los marginales de la tabla.....	57

3. BIOIN ANESTHSOM.....	58
Diagramas de flujo.....	59
3.1 Programa.....	61
Carpeta BIOIN ANESTHSOM.....	61
4. CONTROL DE CAMBIOS.....	63
5. APLICACIÓN DE BIOIN ANESTHSOM.....	68

CAPITULO III RESULTADOS Y DISCUSIÓN.....77

1. ANÁLISIS DESCRIPTIVO.....	79
1.1 Residuales estandarizados de Habertman (RA).....	84
1.2 Distancias de Minkowski.....	87
2. RESULTADOS DE BIOIN ANESTHSOM.....	93
2.1 BLOQUE GENERAL.....	94
2.2 TRAUMATOLOGÍA.....	102
2.3 MATERNO-INFANTIL.....	110
2.4 HOSPITAL UNIVERSITARIO MIGUEL SERVET	118
2.5 HOSPITAL CLÍNICO UNIVERSITARIO LOZANO BLESA	123

Tablas resumen del Análisis de Resultados129 y 130

CONCLUSIONES.....131

BIBLIOGRAFIA.....135

ANEXO.....Ver CD adjunto

CAPÍTULO I

INTRODUCCIÓN

1. ANESTESIOLOGÍA. INCIDENCIAS Y SU MEDICIÓN.

Según la Real Academia Española de la lengua, la acepción anestesia viene del latín *anaesthesia*, y este a su vez del griego *ἀναισθησία anaisthēsía* que quiere decir “insensibilidad”. Se define como “pérdida temporal de las sensaciones de tacto y dolor producida por un medicamento”, “acción y efecto de anestesiar” y “sustancia anestésica” (RAE, 2016).

Según el Ministerio de Educación y Ciencia, en su guía de formación de especialistas de 1996, es la pérdida reversible del nivel de consciencia, sensibilidad táctil y dolorosa y relajación neuromuscular con abolición de reflejos, mediado por fármacos y habitualmente con un fin quirúrgico.

Se define a la especialidad de Anestesiología y Reanimación como una especialidad médica cuyos principales objetivos son el estudio, docencia, investigación y aplicación clínica de métodos y técnicas para insensibilizar al paciente al dolor y protegerlo de la agresión quirúrgica, manteniendo sus funciones vitales. Manejar a los pacientes cuyas funciones vitales se encuentren gravemente comprometidas con medidas terapéuticas apropiadas hasta que supere la situación de riesgo vital. También es la principal especialidad que se encarga del tratamiento del dolor agudo y crónico (Ministerio de Educación y Ciencia, 1996).

1.1 Historia de la anestesiología general

Desde el principio de los tiempos el hombre ha convivido con el dolor y la enfermedad y ha puesto todo su empeño en mitigarlo o hacerlo desaparecer. En las culturas primitivas el control del dolor era cometido de las autoridades religiosas o chamanes. Era considerado normalmente como un castigo o penitencia por los pecados cometidos.

La palabra dolor procede del término griego “*ποινή*” (*poine*) que significa multa. Aristóteles creía que el dolor era una emoción que emanaba del corazón, pero Galeno observó en animales que el cerebro era parte indispensable para sentir el dolor, e incluso propuso que esta sensación dolorosa era debida la existencia de un tejido nervioso (Larson, 2010).

En 1865 se descubrió al sur de Francia una tumba que databa de la edad de piedra con un cráneo trepanado. Con este hallazgo se demostró que hace unos 8.000 años ya se realizaban intervenciones quirúrgicas a las que algunos pacientes sobrevivían. Esto nos lleva a pensar que en aquella época podían mitigar el dolor probablemente con hierbas, al no encontrar evidencia científica que lo corrobore, no podemos decir que el comienzo de la terapia del dolor tenga aquí su inicio (Carretón, 2016).

Los asirios, más de 3.000 años a.C. realizaban intervenciones quirúrgicas gracias a la inconsciencia que conseguían por la hipoxia cerebral al comprimir las arterias carótidas a nivel cervical (Patiño, 1997).

Se han encontrado pergaminos egipcios (2.500-3.000 a.C.) en los que se observan pictogramas que representan una operación de una mano mediante compresión de los nervios de la zona antecubital. En otros pergaminos se puede ver a un paciente comprimiéndose a sí mismo el plexo braquial mientras se le estaba interviniendo el antebrazo (Barash, 2006).

Hipócrates y Galeno fueron pioneros en la lucha contra el dolor, utilizaban “esponjas soporíferas” impregnadas de mandrágora, beleño y opio, que colocaban sobre la nariz del paciente y al inhalar conseguían cierta hipnosis bajo la que realizar la intervención. Estas prácticas dejaron de utilizarse a mediados del siglo XV por la influencia religiosa que tenía la Inquisición. Se perseguía a todo aquel que las

practicaba ya que eran acusados de ejercer la magia negra, la brujería o de rendir culto al diablo. (Martina, 2016).

Algunas técnicas poco ortodoxas heredadas de la antigüedad, se utilizaban en los siglos XVII y XVIII para mitigar el dolor durante las intervenciones quirúrgicas. Una de ellas era la anestesia por hipoxia cerebral por compresión carotídea, otra técnica consistía en la contusión cerebral golpeando la cabeza del paciente con un trozo de madera hasta hacerlo caer inconsciente. En este mismo periodo, John Hunter, un famoso cirujano inglés, realizaba amputaciones de extremidades previa sección de los nervios de dicha extremidad (Patiño, 1997).

En esta misma época, Marco Aurelio Severino desarrolló “*refrigeration anesthesia*”, colocaba nieve sobre la zona a operar, lo que producía cierta analgesia durante un breve periodo de tiempo. Esta técnica no llegó a popularizarse por su escasa eficacia, pero Severino ha pasado a la historia por ser el primero en realizar traqueostomías para asegurar la vía aérea de pacientes durante una epidemia de difteria (Barash, 2006).

Josef Priestley, reverendo inglés, descubrió en 1771 el oxígeno y su papel en la fotosíntesis y en la respiración de los animales. Junto a Joseph Black descubrió el óxido nitroso y fue considerado un gas peligroso capaz de contagiar y diseminar enfermedades (American Chemical Society, 2016).

Años más tarde, Humphry Davy (1778–1829) inhaló el óxido nitroso y descubrió que le aliviaba el dolor que le provocaba la erupción de una muela del juicio, y recomendó su uso como gas anestésico para la realización de cirugías menores. Fue ignorado por la comunidad científica del momento y abandonó esta línea de investigación. Fue también el descubridor de varios iones: potasio, sodio, calcio, bario y magnesio (Science History Institute, 2016).

Un Gran descubrimiento fue el descubrimiento de la morfina a partir del opio que realizó F.W. Serturmer en 1806. Al tratar el opio con amoniaco se crean unos cristales blancos, que al purificarlos con ácido sulfúrico y alcohol producían somnolencia en animales, y lo llamó “*morfium*” en honor al dios del sueño, Morfeo. Este fue el primer opioide aislado que todavía se sigue utilizando en la actualidad por ser relativamente seguro y de sencillo manejo. Serturmer fue el primero en utilizar los

efectos de las plantas medicinales y manipularlas para convertirlas en medicamentos como alcaloides, atropina, escopolamina... base de la farmacología actual (Martina, 2016).

En 1842, el doctor Crawford W. Long, médico rural en Jefferson, Georgia, aplicó éter inhalado a un paciente y le extirpó con éxito dos tumores que presentaba en la nuca. No publicó inmediatamente su hazaña y el Dr. Morton se le adelantó, por lo tanto, es a los médicos Horace Wells y William Morton a quienes se les considera los precursores de la anestesia moderna (Martina, 2016)

Horace Wells, odontólogo, y su colega y amigo, el doctor William Morton, se trasladaron juntos a Boston para trabajar buscando una técnica que redujese el dolor de las intervenciones dentales que realizaban, pero desafortunadamente no llegaron a un entendimiento y decidieron continuar sus investigaciones por separado. Horace Wells volvió a su ciudad natal donde presencié un espectáculo público, en el cual un químico llamado Colton demostraba los efectos hilarantes de la inhalación de óxido nitroso. Wells, tras inhalar el óxido nitroso, se deja extraer una pieza dental por el doctor Riggs. Wells utilizaba con éxito el óxido nitroso en sus pacientes. El 15 de enero de 1845 es invitado por el cirujano y profesor Warren a hacer una demostración pública de su técnica en el Hospital General de Massachusetts de Boston. La demostración fue un fracaso, el paciente tras haber sido anestesiado con óxido nitroso, comenzó a gritar desesperadamente al extraerle la muela por lo que Wells fue considerado un farsante. Este fracaso lo llevó a la perdición, se volvió adicto al alcohol y al éter y se acabó suicidando en la cárcel de Nueva York en 1848 (Martina, 2016).

A partir de entonces, Morton comienza a utilizar el éter en animales, en sus amigos y en sí mismo. Puso a prueba este método con un paciente con dolor de muelas, le aplicó éter inhalado y le extrajo la muela sin dolor. Morton fue también el precursor del aparato de anestesia que manejamos en la actualidad. Diseñó uno formado por una esfera de vidrio con fieltro en su interior y dos orificios: uno superior por el que se introducía el éter y uno inferior del que salía una boquilla por la que se aspiraba el gas producido. Pidió a la facultad de medicina de Harvard realizar una demostración de su técnica anestésica. El 16 de octubre de 1846, realizó la extirpación de un voluminoso tumor submandibular de un paciente anestesiado con

éter y esta vez sí, la demostración fue todo un éxito. El químico Jackson alegó que fue él quien aconsejó a Morton trabajar con ese gas, así que ambos investigadores tuvieron el reconocimiento merecido. Se considera a Jackson el descubridor de la anestesia con éter y al doctor Morton el promotor y quien desarrolló la técnica (Martina, 2016).

James Young Simpson, médico ginecólogo en Edimburgo, utilizó otro compuesto, cloroformo, que fue sintetizado por Justus Liebig y por el francés Soubeyran en 1831 casi a la vez. Comenzó en 1847 a realizar anestесias con cloroformo a 50 pacientes para luego publicar un informe donde explicaba el éxito conseguido. Su uso en la práctica médica de la época fue relativamente breve porque presentaba una gran toxicidad cardiaca y hepática. A pesar de conocer los efectos secundarios, la reina Victoria solicitó el uso de cloroformo en abril de 1853 para el nacimiento de su octavo hijo, el príncipe Leopoldo. Sus médicos fueron James Clark y su anestesista John Snow. Debido a la alta morbimortalidad que se veía tras el uso de cloroformo, se creó una comisión en 1864 para investigar estos efectos secundarios. Los resultados devastadores, lo que hizo que la comunidad médica de la época volviera a utilizar el éter (Martina, 2016).

Durante las primeras décadas del siglo XX, los médicos de la época trabajaban para encontrar nuevos gases anestésicos. El cloruro de etilo se sintetizó y se vio que tenía propiedades tanto de anestésico local como anestésico general si se inhalaba, pero era altamente inflamable por lo que no superó al éter. Otros gases como ciclopropano, tricloroetileno, decloroacetileno... fueron sintetizados en ésta época, pero su uso como agentes anestésicos fue breve porque ninguno de ellos superaba en seguridad ni en calidad anestésica al éter ni al nitroso e iban siendo abandonados (Larson, 2010)

En los años treinta se comenzaron a desarrollar los gases fluorados que se utilizaban y comercializaban en un principio como refrigerantes. John C. Krantz de la Universidad de Meryland utilizó el gas etil-viniléter y lo modificó hasta convertirlo en trifluoroetil-viniléter o flurexeno que se convirtió en el primer anestésico hidrocarburo halogenado. Se introdujo en la práctica clínica en 1954 pero se retiró del mercado por su toxicidad e inflamabilidad. En 1956 se sintetizó el halotano que tenía ventajas sobre su predecesor como un olor más agradable, más potencia y no inflamable lo

que hizo que pronto tuviera una gran aceptación mundial. El 1958 se publicó un caso de hepatitis fulminante en una mujer de 39 años que 11 días antes había sido intervenida de colecistectomía bajo anestesia general con halotano. En 1963 nueve casos más de hepatitis fulminante tras su uso fueron descritos y se descubrió que producía un metabolito tóxico capaz de causar necrosis hepática. Otros gases tales como metoxifluorano y enflorano se utilizaron en el pasado y fueron retirados por sus efectos secundarios o por haber sido superados por otros gases. Isoflorano, desflorano y sevoflurano son los gases que están siendo utilizados hoy en día. El futuro parece venir de la mano del gas Xenón, un gas inerte, que se lleva años investigando, pero todavía es muy cara su producción y se necesitan dosis elevadas anestesia (Larson, 2010).

Al mismo tiempo que la anestesia inhalatoria se desarrollaba, durante los siglos XVIII y XIX se fueron sintetizando fármacos intravenosos con el mismo fin, mitigar el dolor y producir hipnosis. Estos fármacos intravenosos mejoraban a la anestesia inhalatoria con una más rápida inducción y además evitaba al paciente la mala experiencia de la inhalación de los gases. Pierre-Cyprien Oré realizó la primera anestesia intravenosa con hidrato de cloral, a pesar de que parecía ser mejor que la anestesia con cloroformo, no fue una medida bien adoptada por sus colegas de la época (Barash, 2006).

Durante la Primera guerra mundial se popularizó una mezcla de morfina y escopolamina, principalmente para anestesia obstétrica. En 1903, Emil Fischer sintetizó el primer barbitúrico, hexobarbital, pero no se introdujo en la práctica clínica hasta treinta años después. Otros fármacos hipnóticos intravenosos como tiopental, etomidato, ketamina, benzodiacepinas y propofol se fueron desarrollando durante la segunda parte del siglo XX y son los fármacos que se utilizan en nuestros días (Larson, 2010).

1.2 Historia de la anestesiología locorregional.

El primer documento escrito que se conoce sobre el uso de la planta de coca como anestésico local, es del Jesuita español Bernabé Cobo, quien en 1653 al masticar la hoja de coca se dio cuenta de que le aliviaba un dolor de muelas que sufría. Pero no fue hasta 1856 cuando el alemán Albert Niemann aisló el alcaloide a partir de las hojas secas de la planta de coca, y le puso el nombre de cocaína al principio activo. Vasili von Anrep fue el primero en reparar en las propiedades como anestésico local de la cocaína, y sugirió empezar a utilizarlo como analgésico en intervenciones quirúrgicas (Patiño, 1997).

La idea de utilizarlo como anestésico local fue de William Halsted y Alfred Hall, quienes vieron que al inyectar 15 miligramos de cocaína al 4% en el antebrazo, se producía un bloqueo sensorial por interrupción de la transmisión nerviosa, y se producía analgesia desde el punto de punción hacia abajo. A partir de esto se fueron desarrollando los distintos tipos de bloqueos neuroaxiales como el bloqueo de plexo braquial, bloqueo del nervio ciático, de nervios dentales... Para intentar evitar los efectos indeseables de la sobredosificación de morfina que se producía con estas técnicas, se fueron sintetizando otros compuestos como la procaína, por Braun en 1905, menos tóxica que la cocaína (Larson, 2010).

En 1908 August Bier creó un nuevo método de producir anestesia regional, el conocido bloqueo de Bier o anestesia regional intravenosa, que consistía en la exanguinación de la extremidad mediante un "Esmarch" o compresivo, después, la colocación de 2 torniquetes que eviten la vascularización del miembro, y se inyecta procaína intravenosa. El bloqueo se instaura casi inmediatamente y persiste hasta que se suelta el torniquete. Hoy en día está prácticamente en desuso debido a los efectos indeseables del paso del anestésico local en grandes cantidades a nivel sistémico (Larson, 2010).

El éxito de la anestesia regional ha favorecido la aparición de nuevos fármacos anestésicos locales como lidocaína, mepivacaína, bupivacaína y levobupivacaína que son más seguros. También han aparecido otras herramientas que nos ayudan a que las punciones sean más seguras como el uso de neuroestimuladores y lo más reciente, el uso de ecografía para la localización de

nervios periféricos.

En 1921, Fidel Pagés, un cirujano militar oscense, descubrió la técnica epidural con procaína. Escribió un artículo donde exponía 43 casos de anestesia epidural a nivel lumbar o torácico con buenos resultados, pero no tuvo demasiado impacto ya que la revista en la que lo publicó carecía de prestigio. Pagés murió de accidente de tráfico poco tiempo después y ninguno de sus alumnos había aprendido la técnica así que su descubrimiento quedó en el olvido y en 1931 Achille Mario Dogliotti volvió a describir esta técnica y trabajó para desarrollar una técnica sencilla (Larson, 2010).

En España hasta bien entrados los años 70 no se comenzó a popularizar la anestesia intradural y la anestesia hasta los 90 (Larson, 2010).

1.3 Incidentes en Anestesiología

Según el Sistema Español de Notificación en Seguridad en Anestesia y Reanimación, SENSAR, en 2018, “un incidente relacionado con la seguridad del paciente es cualquier circunstancia que suponga o pueda suponer un riesgo innecesario para el paciente durante la atención sanitaria y puede ser comunicado por médicos, enfermeras o resto de personal sanitario dependiendo del centro hospitalario”.

Todo acto anestésico-quirúrgico tiene asociado un riesgo. Nuestro trabajo como anestesiólogos consiste en reconocer aquellas situaciones que producen un riesgo y tratar de eliminarlo o al menos reducirlo con los medios de que dispongamos (Rodríguez Zazo, 1990).

Según la Real Academia Española de la lengua (RAE), la palabra “riesgo” viene del italiano “*rischio*” y éste del árabe clásico “*rizq*”, que quiere decir *lo que depara la providencia*. Se define también como: contingencia o proximidad de un daño (RAE, 2016).

Cada año se realizan en el mundo más de 200 millones de cirugías mayores, de la cuales entre un 0,4-0,8% tienen secuelas muy graves o resultado de muerte. Una revisión sistemática realizada por De Vries en 2008 donde se analizaron 70.000 historias clínicas de pacientes ingresados y se observaron eventos adversos en el 9,2% de los ingresos de los cuales el 43,5% pudieron haberse evitado (Gómez-Arnau, Bartolomé, García Del Valle y González, 2011).

Hasta los años 80 la muerte relacionada directamente con la anestesia era relativamente alta, 1 de cada 2.500-5.000 anestесias. En los últimos años una serie de artículos publicados han demostrado que se ha reducido de forma sustancial. La incidencia actual de mortalidad relacionada con la anestesia es de 1 de cada 100.000, de las que el 15,3% eran debidas a sobredosificación o equivocación en la administración de fármacos, el 13,9% debido a arritmias severas y también fueron causas de muerte el infarto de miocardio, la hipovolemia, la hemorragia y problemas relacionados con la vía aérea (Kawashima et al., 2003).

En otro estudio francés, Lienhart et al. en 2006 publicó a partir del registro de defunciones que, de 537.459 muertes, 4.200 se habían codificado como muerte relacionada con anestesia, según el código de la Clasificación Internacional de Enfermedad (ICD-9), se observó un aumento claro en la mortalidad en los pacientes que habían sido clasificados como ASA IV por encima de los menos graves ASA I. Se realizó un análisis causa-raíz que demostró que además de las causas de mortalidad relacionadas con la enfermedad del paciente, también estuvieron implicados factores de la organización y factores de equipo o del individuo.

En Holanda, Arbous et al. en 2001 publicaron que era la patología cardiovascular la primera causa de muerte, seguida de factores asociados a problemas de medicación y vía aérea. Coincide con Lienhart et al. (2006) en que el fallo humano es un factor primordial, encontrando asociación causal hasta en el 75 % de los casos.

Aunque estos artículos tienen diferencias metodológicas importantes que nos dificultan comparar adecuadamente sus resultados, podemos decir que todos ellos demuestran que ha habido una importante reducción de la mortalidad asociada a la anestesia en las últimas décadas y esto es debido con casi total seguridad al importante avance que esta especialidad ha tenido tanto en monitorización y vigilancia de los pacientes como en los tratamientos farmacológicos y las propias técnicas anestésicas que tanto han evolucionado. Estos estudios coinciden en que las principales causas de muerte son la cardiovascular y la respiratoria, aunque también son muy significativas las causas relacionadas con catéteres venosos centrales y errores de medicación. Sin olvidar que el error humano está presente en un porcentaje muy alto de los casos en todos los estudios. Otro aspecto muy importante en el que coinciden es que una inadecuada evaluación preoperatoria está presente en el 33% de estas muertes.

Con todos estos resultados, se debe trabajar, y de hecho ya se está haciendo, para minimizar todos estos incidentes, principalmente aquellos relacionados con el error humano. Se han establecido una serie de protocolos y directrices a seguir por todos los Servicios de Anestesiología en Europa, que surgieron de la celebración de la reunión que el 13 de junio de 2010 tuvo lugar en Helsinki con los principales líderes de las sociedades de anestesia europeas y se creó la Declaración de Helsinki

para garantizar la seguridad de los pacientes durante su proceso asistencial. En esta declaración se recogen una serie de medidas de seguridad que han demostrado reducir o minimizar los eventos adversos como realizar una adecuada valoración preoperatoria, el etiquetado de jeringas de fármacos, el control de equipos de anestesia, la realización de una hoja de Verificación de Seguridad Quirúrgica o “*Check List*” (Declaración de Helsinki, 2010).

La Declaración de Helsinki nos enseña la importancia que tiene para la Anestesiología la seguridad del paciente y que, afortunadamente, los profesionales de este campo estemos cada vez más concienciados en trabajar por su implantación. El registro y estudio de los incidentes críticos ha demostrado ser de vital relevancia para el aprendizaje y la adopción de medidas y creación de protocolos para evitar que se repitan (Arnal et al., 2011).

1.4.- Seguridad del paciente

1.4.1.- Desde sus comienzos

Los términos “seguridad del paciente” y “calidad asistencial” han tomado gran importancia en las últimas décadas. Lejos de lo que parece, no son conceptos nuevos, sino que las consecuencias de los actos médicos, ya en la antigüedad, preocupaban tanto a médicos como a pacientes.

La primera referencia de la que tenemos constancia es en el Código de Hammurabi, escrito hacia el año 1700 a.C. Es el primer conjunto de leyes de la historia, donde Hammurabi plasma todas las normas que dios le ha transmitido para fomentar el bien y el orden de su pueblo. Formado por 282 leyes de las cuales 11 son referentes a la Medicina, algunas de ellas muestran la preocupación por la seguridad del paciente de la época. *“Si un médico ha tratado una herida grave al esclavo de un plebeyo con el punzón de bronce y lo ha matado, devolverá esclavo por esclavo. Si ha abierto la nube con la lanceta de bronce y ha destruido el ojo, pagará en plata la mitad del precio del esclavo. Si un médico ha tratado a un hombre libre de una herida grave con la lanceta de bronce y ha hecho morir al hombre, o si ha abierto la nube del hombre con la lanceta de bronce y destruye el ojo del hombre, se le cortarán las manos.”* (Forero, 2003).

A Hipócrates (460 a.C.), considerado el padre de la Medicina, se le atribuye la conocida frase *“Primum non nocere”* que aparece, aunque no textualmente en el *Corpus Hippocraticum* y significa “lo primero es no hacer daño” (Larson, 2010).

Pierre Charles Alexandre Louis (1787-1872), pionero de lo que hoy en día conocemos como “medicina basada en la evidencia”, fue el principal defensor de lo que llamó *“méthode numérique”* que consistía en cuantificar la medicina, de esta forma sería posible conocer la importancia de los signos y síntomas y conocer la evolución, duración y gravedad de las enfermedades y conocer su incidencia en nuestro medio. Con esta cuantificación se podía también llegar a conocer la eficiencia de los tratamientos. Fue el primer médico en aplicar la estadística a su actividad clínica diaria (Moreno, 2013).

El cirujano Ernest Codman (1869-1940), puede ser considerado el primer médico en publicar un error médico que él mismo cometió, durante una intervención

quirúrgica, admitió haber ligado el conducto hepático común, lo que produjo la muerte del paciente. Realizó además un seguimiento de todos los pacientes que intervenía y registró el resultado final de sus cuidados. Entre los años 1911 y 1916 registró alrededor de 120 errores y los organizó en errores causados por falta de conocimientos o inexperiencia, error en la indicación quirúrgica y por fallo de recursos bien personales o bien materiales. Publicaba sus errores en un informe anual, que hacía llegar a otros hospitales norteamericanos instándoles a hacer lo mismo. Decía: *“Soy excéntrico por decir en público que los hospitales, si desean estar seguros de su mejora, deben conocer sus resultados. Deben analizarlos para encontrar sus puntos fuertes y débiles. Deben comparar sus resultados con los de otros hospitales. Estas opiniones no serán de “excéntricos” en unos años”* (Aranaz, 2011c).

En los años 50 del siglo XX se publican los primeros estudios sobre los riesgos de la hospitalización y los errores médicos. Es precisamente la especialidad de Anestesiología una de las primeras en alertar sobre la importancia de evaluar los métodos empleados para prevenir las muertes relacionadas con la anestesia (Aranaz, 2011c).

Pero no fue hasta noviembre de 1999, cuando el “Institute of Medicine” publicó *“To err is human: building a safer health system”*, un artículo que alarmó a la sociedad diciendo que los errores médicos evitables provocaban más muertes que el cáncer de mama, el SIDA y los accidentes de tráfico juntos. A partir de ese momento no solo los profesionales de la salud comenzaron a estar preocupados por el concepto de “seguridad del paciente” sino el mundo entero (Institute of Medicine, 1999).

A raíz de todos estos trabajos, poco a poco, con el paso de los años, se ha ido consiguiendo la puesta en marcha de medidas de seguridad con las que se ha conseguido disminuir drásticamente la muerte relacionada con la asistencia médica en general y con la anestesia en particular en los últimos 25 años (SENSAR, 2018).

1.4.2 En la actualidad:

La seguridad del paciente, piedra angular de la calidad asistencial, engloba las actuaciones dirigidas a reducir o eliminar los eventos adversos de todo acto médico. Se considera un evento adverso relacionado con la asistencia a cualquier suceso que tiene lugar debido al proceso asistencial y que tiene consecuencias negativas para el paciente. La consecuencia de este evento adverso puede ser fallecimiento, lesión, incapacidad, aumento de la estancia hospitalaria o incremento de consumo de recursos. Principalmente la seguridad del paciente está basada en la prevención de los eventos adversos que son derivados de la relación asistencial y no de las complicaciones o consecuencias del desarrollo natural de la enfermedad del paciente. Sin embargo, las características de la mayor parte de los pacientes que atendemos en la actualidad como el envejecimiento, las condiciones socioeconómicas, el soporte familiar y el nivel educativo, son una amenaza para la seguridad, porque aumentan la vulnerabilidad del paciente (Aranaz, 2011c).

Para conseguir una atención médica mejor y más segura es necesario un aprendizaje continuo, poniendo en práctica las medidas que han demostrado que disminuyen la incidencia de eventos adversos y errores, desarrollar mecanismos que nos permitan detectar fácilmente estos errores y ser capaces, cuando aparezcan, de minimizar sus consecuencias en la medida de lo posible (Aranaz, 2011c).

A nivel mundial hay una serie de instituciones y agencias internacionales encargadas de regular todas estas medidas encaminadas a reducir eventos adversos como el Programa de Seguridad del Paciente de la OMS, la *Joint Commission on Accreditation of Healthcare Organizations* y la *Agency for Health Care Research and Quality* (AHRQ) en los Estados Unidos. *The Australian Patient Safety Foundation*, la *Haute Autorité en Santé de Francia*, la *National Patient Safety* (NPSA) del Reino Unido (Aranaz, 2011c).

En España, cada vez son más los grupos de investigación interesados en este tema. La Agencia de Calidad del Sistema Nacional de Salud recomienda en su Taller de Expertos celebrado en febrero de 2005 para la Estrategia en Seguridad del Paciente, que hay que incentivar las actividades de mejora de la calidad fomentando la participación de los profesionales (Rubio García, García y Díaz, 2014; Bona Gracia, Rubio García y García-Foncillas, 2017; Bona Gracia, Torner Pelay y García-

Foncillas, Rubio García, 2018a). Es un problema que se ha abordado desde las instituciones, pero es un problema que preocupa y debe solucionarse mediante el personal clínico (Fernández García, Martínez Terrer y Rubio, 2002).

El Ministerio de Sanidad, Servicios Sociales e Igualdad (MSSSI) ha desarrollado varios estudios para determinar la magnitud y los factores determinantes del riesgo asociado a la asistencia sanitaria en nuestro país. Entre ellos destaca el estudio ENEAS (Estudio Nacional de Eventos Adversos relacionados con la Hospitalización), se trata de un estudio retrospectivo con 5.624 pacientes ingresados en 24 hospitales públicos en España, que estuvieron ingresados al menos 24 horas entre las fechas 4 y 10 de julio de 2005. La incidencia de pacientes con eventos adversos reconocidos fue del 9,3%. De estos, el 37,4% estaban relacionados con la medicación, el 25,3% con infecciones nosocomiales y el 25% con problemas técnicos en la realización de algún procedimiento. El 45% fueron leves, 38,9% moderados y el 16% graves. En total, el 42,8% de los eventos adversos se consideraron evitables (ENEAS, 2005). Estos resultados son de una gran importancia porque se han obtenido de una muestra representativa y se han obtenido a partir de una metodología reproducible que se está aplicando a nivel nacional (Aranaz et al., 2008, 2009, 2011, 2011 a y b).

Uno de los principales problemas en nuestro país, y en la mayor parte de países europeos, en la actualidad es que no existe obligación de declaración de los incidentes, por lo que realizar estudios sobre eventos adversos recogidos en bases de datos es complicado (Rees et al., 2017). Un estudio reciente, demostró que simplificar, facilitar e informatizar la recogida de incidentes, produce motivación con el consiguiente incremento de casos reportados (Williams, Muffly, Mendoza, Wixson, Leong y Claire, 2017).

1.4.3. SENSAR

SENSAR es el Sistema Español de Notificación en Seguridad en Anestesia y Reanimación. Se trata de una asociación científica sin ánimo de lucro cuyo objetivo es la promoción de la cultura de la Seguridad del Paciente quirúrgico, el paciente crítico y el tratamiento del dolor, a través del aprendizaje de los errores. Es una asociación íntimamente ligada la Sociedad Española de Anestesia, Reanimación y

Tratamiento del Dolor (SEDAR), la Sociedad Española del Dolor (SED) y la Sociedad Europea de Anestesiología (ESA).

La idea de crear un sistema de notificación de eventos adversos en el campo de la anestesiología nació en 1999 en el propio Servicio de Anestesiología y Reanimación del Hospital Universitario Fundación Alcorcón en Madrid, su idea se fue expandiendo hasta hacerse de ámbito nacional en 2009.

SENSAR se encarga de recoger los incidentes relacionados con la seguridad del paciente que tienen lugar en los hospitales adscritos a la organización. Esta recogida es anónima, confidencial, voluntaria y no punitiva. Los incidentes recogidos se analizan por un grupo de expertos para la búsqueda de los fallos que pueden haber causado ese incidente e intentar que no vuelvan a suceder. Con los resultados obtenidos de los análisis se crean alertas y protocolos de actuación con los que intentan disminuir la incidencia de eventos críticos (Gómez-Arnau Díaz-Cañabate, Bartolomé Ruibal, Santa-Ursula Tolosa, González Arévalo y García del Valle Manzano, 2006).

En la actualidad SENSAR es un sistema de comunicación, análisis y gestión de incidentes relacionados con la seguridad en anestesia al que están adscritos 71 hospitales y en el que se analizan más de 650 reportes al año (Arnal et al., 2011).

1.5.- Estadística y su relación con la Anestesiología

El uso de la Estadística en las Ciencias de Salud, y en especial en Medicina, se ha ido incrementando en estos últimos años, hasta el punto de convertirse en algo básico para la investigación y para la formación de los médicos hoy en día. Ya en la Universidad se tiene contacto con ella, siendo una de las asignaturas principales en los primeros años de carrera. Tal es su importancia que se le conoce con un nombre específico, Bioestadística. Según la RAE, bioestadística es “*la ciencia que aplica el análisis estadístico a los problemas y objetos de estudio de la biología*” (RAE, 2016).

Este aumento de la importancia, se debe principalmente al auge de la informatización en Medicina, siendo actualmente habitual el acceso a programas de fácil manejo que nos proporcionan análisis estadísticos muy complejos y con menos errores que los que se realizaban de forma manual. En Anestesiología, como en el resto de especialidades, los investigadores tienen la necesidad de justificar científicamente los resultados de sus estudios, es decir, de describir, relacionar e inferir en los problemas sobre los que se trabaja.

Estadística significa “ciencia del Estado”, y proviene de término alemán *Statistik*, debido a que originariamente se utilizaba solo con fines estatales. Los gobiernos por razones de organización, necesitaban conocer las características de su población para el reparto de tierras, pago de impuestos, reclutamiento de soldados etc. así que los primeros trabajos fueron en el área de la demografía (Instituto Nacional de Estadística, 2016).

No se conoce con certeza el origen de la aplicación de la Estadística para Medicina, pero Pierre Charles Alexandre Louis (1787-1782) como ya vimos, fue el principal creador del “*methode numérique*” que consistía en cuantificar todo lo que ocurría en medicina con el fin de poder objetivar sus resultados. En otros muchos documentos aparece como primer autor Dunn, quien en 1929 publicó un artículo donde desarrolla los principales conceptos de la estadística (Dunn, 1929).

Si nos referimos a la estadística aplicada a la medicina de la forma en que la conocemos hoy en día, es Sir Austin Hill su auténtico creador, en 1937 publicó en *The Lancet*, una serie de artículos donde fundamenta y desarrolla tales conceptos.

La evolución de esta ciencia se desarrolla muy lentamente, y no es hasta la década de los 60 cuando se generaliza su uso (Schor y Karten, 1996).

En los últimos años se ha estudiado también la adecuación de las técnicas estadísticas en estudios Originales en revistas con impacto de Anestesiología, como hicieron Ortiz Sagristá en 2004 y Rubio García en 2005, que publicaron trabajos que analizan las técnicas estadísticas y los programas informáticos con los que han trabajado para detectar errores estadísticos e informáticos y también han estudiado el grado de accesibilidad análisis-dependiente y artículo-dependiente del lector (Rubio García, Rodríguez, Martínez y Rubio Calvo, 2010).

En los últimos tiempos tanto el número de técnicas utilizadas como su complejidad ha crecido de forma exponencial. Al principio, en los trabajos en los que aparecía algo de estadística, era únicamente estadística descriptiva, en cambio hoy en día la estadística es mucho más compleja y es habitual hablar de diseño, tamaño muestral, pruebas no paramétricas, regresión logística, potencia, etc.

Merece resaltar la técnica del Metaanálisis, es una metodología para la revisión sistemática y cuantitativa de la literatura conocida sobre un tema, mucho más potente que la información extraída de una determinada muestra, aunque la gran densidad de información debe ser interpretada con precaución y es una técnica consolidada y muy aplicada en la Medicina (Marín Martínez, Sánchez Meca y López López, 2009).

Otra técnica estadística, utilizada recientemente en Medicina es el Método Delphi, en concreto en la especialidad de Anestesiología ha sido utilizada por Rubio García en 2012. El objetivo del método Delphi consiste en la obtención de un consenso a partir de la discusión entre expertos en un tema. Estos expertos deben ser especialistas reconocidos en el tema a estudio y además deben presentar diversidad de opiniones para evitar la aparición de sesgos (Astigarraga, 2000). Se lleva a cabo mediante la elaboración de un primer cuestionario que debe ser contestado por los expertos, los juicios u opiniones de este primer grupo de expertos se toman como base para ser valorado por otro grupo de expertos y se debe alcanzar un consenso de grupo. Finalmente, el responsable del estudio elaborará sus conclusiones mediante el análisis de los datos obtenidos. (Scott, 2001).

Utilizando este método Delphi se creó el cuestionario que generó ANESTHSOM (Rubio García, 2012), y que en este trabajo se aplica.

La aparición de errores es un problema que sigue preocupando a pesar de la informatización de estos trabajos, ya que tener un potente programa informático para la realización de la estadística no significa que se haya utilizado correctamente por parte de los investigadores. Además, los propios programas informáticos en sí mismos pueden contener errores o equivocaciones, bien por fallos de programación, bien por problemas de la documentación que incluyen o bien por errores en la introducción de datos. (Marrugat, Sanz, Porta Serra y Sancho, 1989).

Según Ortiz (2004) en su tesis doctoral, en Anestesiología hay pocos artículos que dediquen esfuerzo a analizar la metodología estadística y los errores en su aplicación (Avram, Shanks, Dykes, Ronai y Stiers, 1985) así como tampoco hay muchos artículos en la especialidad que analicen la calidad y utilidad de los estudios que se han publicado (Greene, 1992). Y más recientemente los publicados por Rubio García en 2005 y 2010.

A pesar de todos estos posibles errores que pueden aparecer, las ventajas de la informatización de estos trabajos, es tan grande que supera con creces los inconvenientes. Nos facilita la realización de cálculos automatizados en estadística multivariante y la realización de cálculos en grandes bases de datos que antiguamente eran prácticamente imposibles de realizar manualmente. Nos permite también controlar los posibles errores en la introducción de los datos.

2. ACSOM Y SU PROGRAMA INFORMÁTICO BIOIN.

2.1.- ACSOM. Introducción.

ACSOM es un modelo matemático creado para el análisis, planificación y control de los accidentes laborales (Conte y Rubio, 2006). Nace de la necesidad que tenían algunos inspectores de trabajo para tratar la accidentalidad laboral desde un punto de vista estadístico ya que clásicamente se realizaba una valoración subjetiva de lo ocurrido generando en ocasiones cierta insatisfacción (Conte, Rubio, Domínguez y García, 2004 a; Conte, Domínguez, García y Rubio, 2004 b). Es un nuevo enfoque para el análisis de los accidentes de trabajo basado en técnicas estadísticas multivariantes, engloba la evaluación de riesgos y su análisis para prevenirlos, así como también realiza un análisis tradicional del accidente (Conte, 2004; Conte y Rubio, 2005a; Conte, Marcos, García Felipe y Rubio, 2007; Conte, García Felipe y Rubio García, 2007 a). Es una metodología potente porque puede aplicarse a empresas de todo tipo y tamaño. (García Felipe, Conte, Rubio y Pérez Prados, 2009; Rubio García, 2012).

El método ACSOM facilita el contraste estadístico, entre una empresa que se quiera analizar, con un “patrón” o entidad superior (García Felipe et al., 2009). Finalmente, con los datos obtenidos, se crea un plan de actuación sobre aquellos riesgos que han demostrado causar más accidentes que son los que el método considera puntos débiles.

Este método ha sido validado al aplicarse a diferentes empresas de diversas actividades y así se ha plasmado en diversas publicaciones (Conte, Rubio y García, 2005; Conte, Rubio, García y Domínguez, 2006 a; Conte, Rubio, García, Domínguez, Bascuas, 2006 b; García Felipe et al, 2009; Gimeno, 2012).

2.2.- Concepto de ACSOM

En ACSOM, la forma de analizar el accidente laboral es considerándolo como un par, en el que un hecho o acción que denominamos RIESGO, provoca una consecuencia que denominamos LESIÓN.

Para su validación, se decidió analizar el registro español de accidentes laborales entre los años 1990 a 2000 (García-Foncillas, 2015), de la siguiente manera:

1.- El accidente se describe a través de un par (riesgo-lesión). Se definieron 19 riesgos y 19 lesiones a partir de las generadas por la Organización Internacional del Trabajo (OIT) y se creó la matriz de frecuencias (absolutas y relativas), en cada año, de los de riesgos y lesiones registrados.

2.- Se tomó como referente de España el año medio, de los 11 años analizados.

3.- Se aplicó una batería de test estadísticos multivariantes que se registró como OMEGA®, que realizaba en primer lugar un análisis factorial (AF), seguido de un análisis de correspondencias (ACS), con lo que se obtenía la relación de máxima afinidad entre riesgos y lesiones creando grupos de tipos de accidente. Se mejoró aplicando después un análisis factorial (AF), análisis de segmentación, un análisis de correspondencias y completado con un análisis clúster jerárquico. Esto les permitió estudiar los riesgos y las lesiones, tanto por separado como por influencias y relaciones de unos con otros (Conte y Rubio, 2005; Conte, Cano, García, Molina y Rubio Calvo, 2008; Conte, Rubio, García y Cano, 2011a).

ACSOM es una potente herramienta matemática, es científica y objetiva, con la que es posible realizar una Previsión, Predicción, Seguimiento y Control del Accidente Laboral. ACSOM tiene capacidad suficiente como para convertirse en una herramienta básica de trabajo para todos los profesionales de la Previsión de Riesgos Laborales (PRL).

2.3.- Aplicación informática de ACSOM: Programa BIOIN.

La metodología de ACSOM se aplica a través de un programa informático, BIOIN ACSOM (Rubio, Torner, Conte y García, 2008; Rubio, Torner, García, Conte, 2008 a; Torner, Conte, García y Rubio, 2008; Torner, García, Conte y Rubio, 2008 a y b; Rubio Calvo et al., 2015) que facilita todo lo anterior realizándolo de forma automática, y nos aporta información sobre los puntos de actuación prioritaria.

BIOIN simplifica en dos gráficas las poligonales de la entidad que se quiere analizar comparada con un patrón elegido. Crea una gráfica en la que detalla los riesgos (causas) y otra para las lesiones (consecuencias). Genera también otras dos gráficas donde representa el análisis estadístico (una de causas y otra de consecuencias). Una tercera gráfica aporta información ordenando las causas y consecuencias por fases de actuación prioritaria.

2.3.1. Gestión telemática.

El programa BIOIN permite e-gestión, es decir, permite su uso a través de internet (Rubio, Conte y García, 2007; Rubio Calvo et al., 2012; García-Foncillas, García Felipe, Bona Gracia y Rubio García, 2016).

Con los datos de accidentes (Riesgos-Lesiones) recogidos de la empresa a estudio, se crea la matriz RxL, con la que se podrá analizar la estructura de los Riesgos, las Lesiones y su interrelación. Es decir, hay un análisis de Riesgos o de Lesiones y otro de Riesgos y Lesiones. Según los resultados obtenidos, se generarán los informes oportunos (García Felipe, Conte y Rubio García, 2007) (figura 1.1).

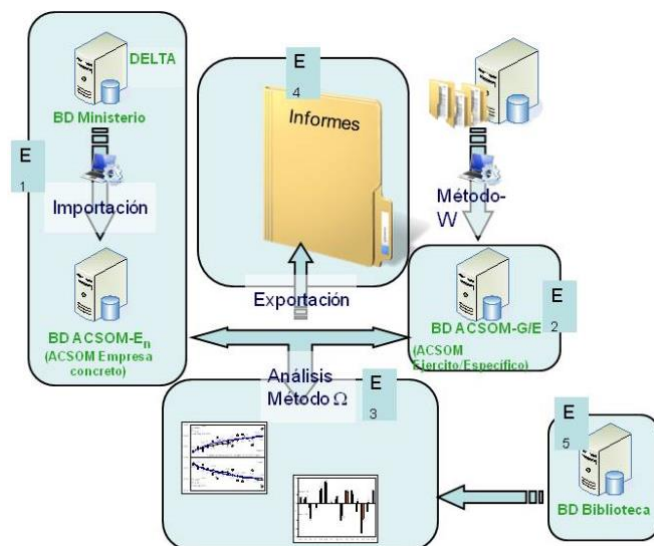


Figura 1.1 Flujo de información para el funcionamiento de ACSOM. Fuente: García Felipe et al., (2007).

BIOIN permite acceder y trabajar con la base de datos ACSOM para importar, analizar y exportar datos desde cualquier dispositivo siempre y cuando esté conectado a internet (Rubio et al, 2007; García Felipe et al, 2007) (figura 1.2).



Figura 1.2 Escenario de trabajo de ACSOM con BIOIN. Fuente: García Felipe et al., (2007).

Para trabajar con BIOIN, en primer lugar, debe seleccionarse el patrón que en este caso es España, pero podría tratarse de cualquier otro con el que se desee comparar como una rama de actividad, una Comunidad Autónoma, otro país, etc. Este patrón ha de englobar siempre el ítem que queremos analizar y debe ser una entidad superior.

En segundo lugar, se selecciona el ítem que se quiere analizar que, por ejemplo, puede ser Aragón.

2.3.2. Ejemplo de funcionamiento.

Un ejemplo para ver el funcionamiento y validación de BIOIN, fue el análisis Riesgos y Lesiones de una empresa de industria bioquímica frente a otras empresas de su misma rama de actividad elegidas como patrón. Los resultados que se obtuvieron mediante el análisis de BIOIN ACSOM se pueden ver en las figuras 1.3 y 1.5 mediante poligonales, en las figuras 1.4 y 1.6 mediante valoraciones estadísticas, desviaciones y translocaciones, y en las figuras 1.7 y 1.8 se pueden ver las fases de planificación de actuación prioritaria o *Planning Report* (García-Foncillas, 2015).

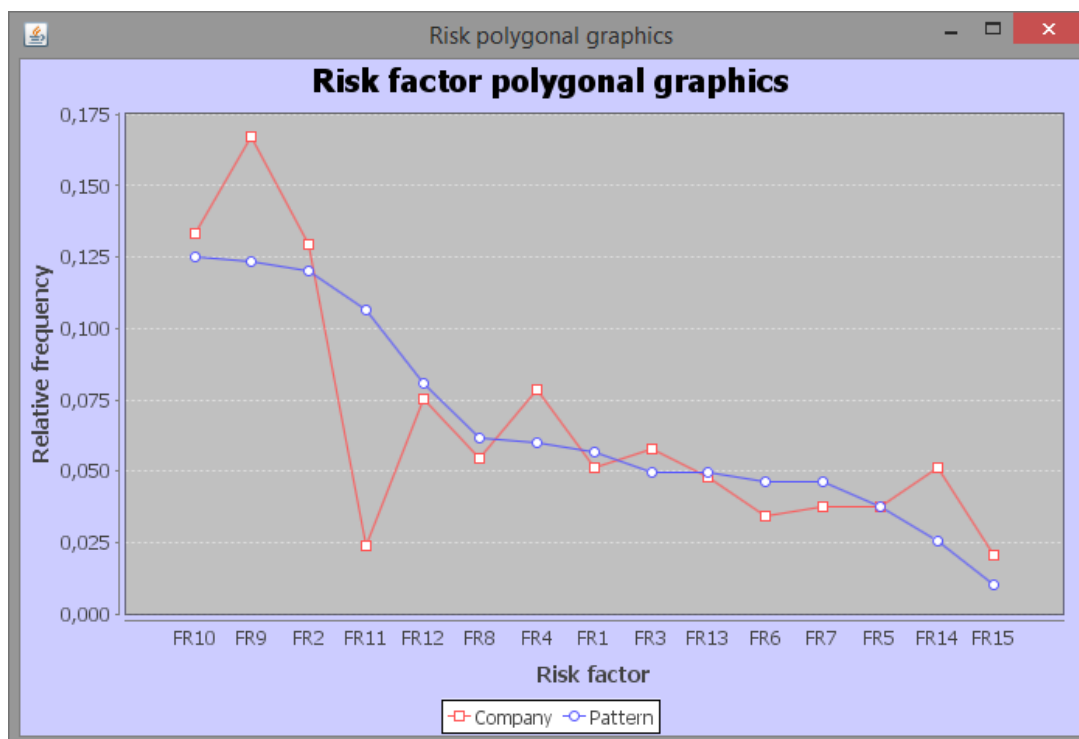


Figura 1.3 Poligonal de Riesgos Planta Bioquímica vs Rama de Actividad. Programa BIOIN. Fuente: García Felipe et al., (2009)

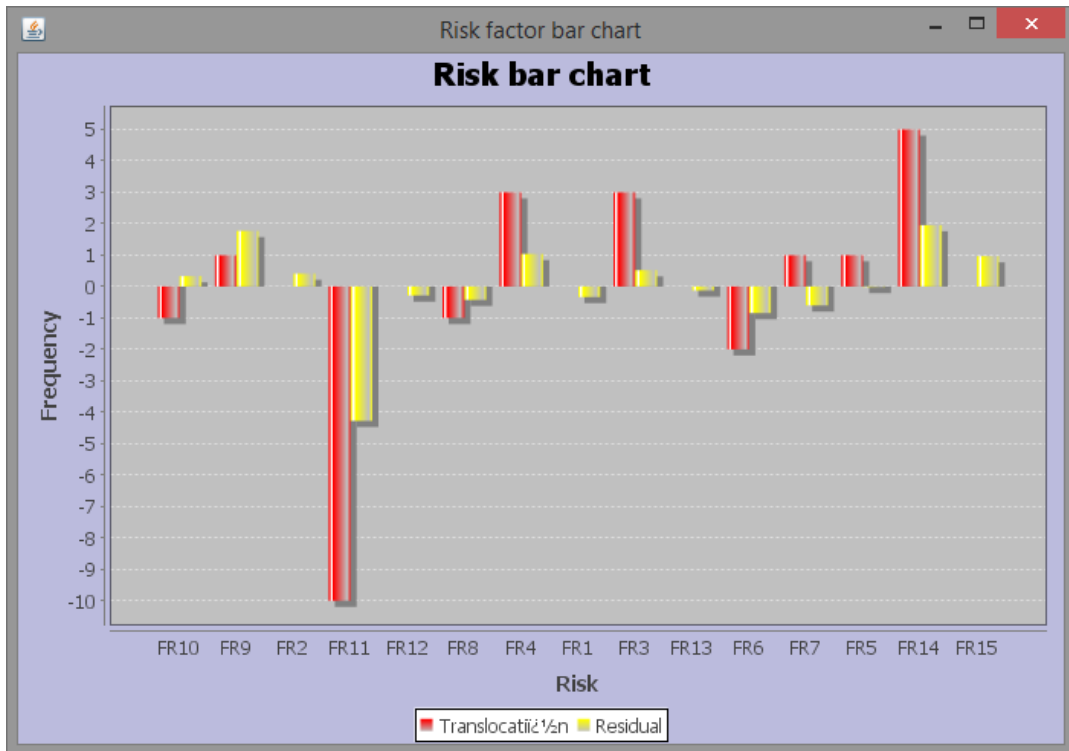


Figura 1.4 Valoraciones estadísticas de los Riesgos, a través de las desviaciones y translocaciones. Planta Bioquímica vs Rama de Actividad. Programa BIOIN. Fuente: García Felipe et al., (2009).

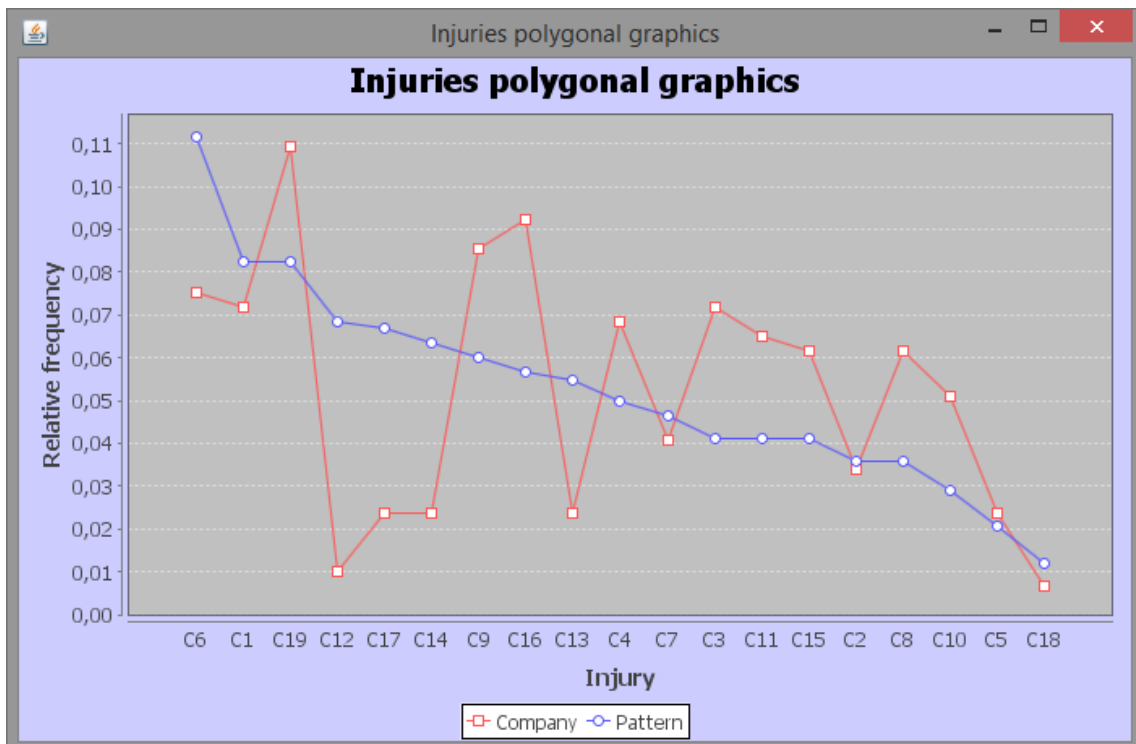


Figura 1.5 Poligonal de Lesiones Planta Bioquímica vs Rama de Actividad. Programa BIOIN. Fuente: García Felipe et al., (2009).

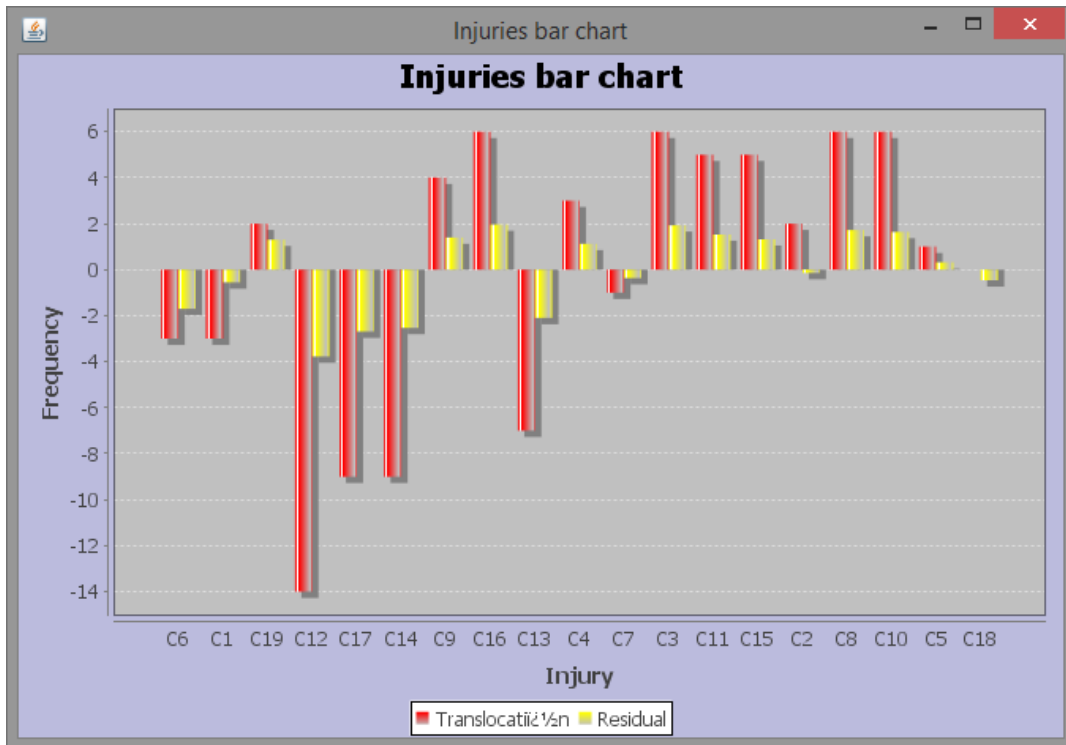


Figura 1.6 Valoraciones estadísticas de las Lesiones, a través de las desviaciones y translocaciones. Planta Bioquímica vs Rama de Actividad. Programa BIOIN. Fuente: García Felipe et al., (2009).

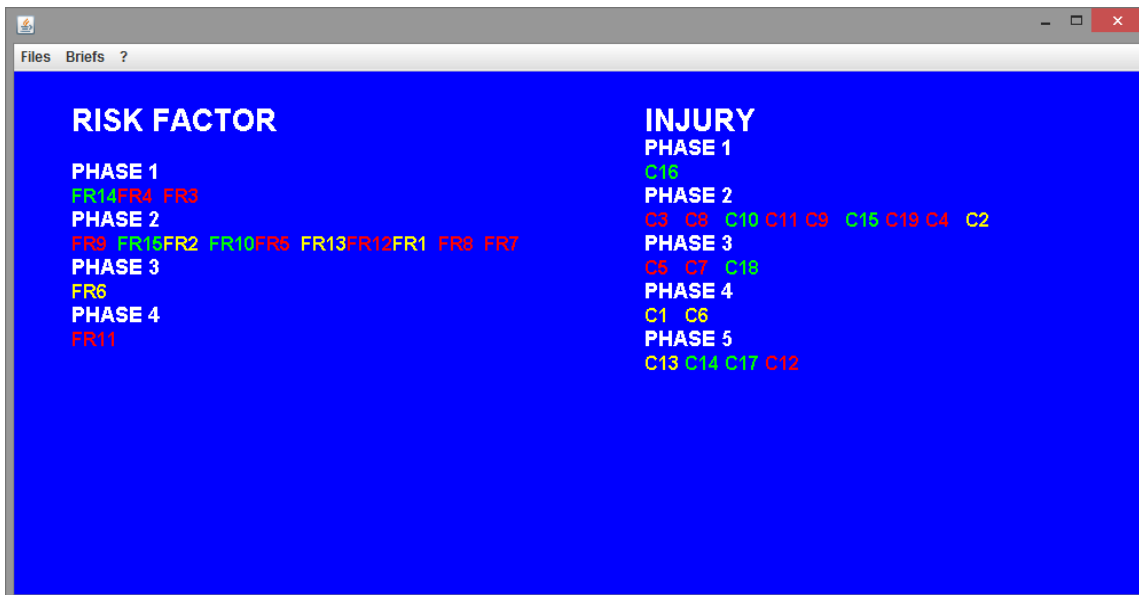


Figura 1.7 Fases para la planificación de los Riesgos y de las lesiones. Programa BIOIN. Planta Bioquímica vs Rama de Actividad. Salida BIOIN. Fuente: García-Foncillas et al., (2016).

Si se pulsa sobre la pestaña “Briefs” de la barra superior, aparecen las tres fases primeras de Riesgos y Lesiones. Si se activan dichas fases, aparece una breve descripción de las lesiones/riesgos y su aparición (García-Foncillas et al., 2016) (Figura 1.8).



Figura 1.8 Fases para la planificación de los Riesgos y de las Lesiones. Programa BIOIN. Con Briefs activado. Planta Bioquímica vs Rama de Actividad. Fuente: García-Foncillas et al., (2016).

La poligonal y la gráfica de resultados nos ofrecen tener una visión global de los Riesgos y Lesiones que se deben atender prioritariamente.

En la primera fase están incluidos aquellos Riesgos o Lesiones que están peor en la entidad que queremos analizar que en el patrón. Los puntos más débiles son los primeros de la primera fase, que son los de actuación prioritaria y son a los que deben dedicarse mayores esfuerzos y recursos para minimizarlos. En el ejemplo de esta empresa, C16 es el prioritario (figura 1.7).

En cambio, aquellos que aparecen en última fase, en nuestro ejemplo C13, C14, C17 y C12, son puntos fuertes en esta empresa con respecto al patrón.

2.4 BIOIN y su lenguaje de programación.

El desarrollo de BIOIN ANESTHSOM se ha trabajado con lenguaje de programación Java, por su adaptabilidad para ser interpretado desde cualquier dispositivo y por su seguridad, lo que aporta una gran tranquilidad al usuario. Para su utilización es necesario instalar una máquina virtual Java, en la actualidad existen versiones para todos los entornos, ordenador, Smartphone, tablet...

Los archivos *.class* pueden ejecutarse desde cualquier dispositivo que admita código Java. La necesidad de memoria RAM es mínima, no se precisa más recursos que los que permitan ejecutarse el sistema operativo de cada máquina. Además, si se utiliza la gestión telemática, a través de internet, no se precisa espacio en el disco duro porque no se almacena la información. Gracias a esta cualidad es posible utilizar este programa desde dispositivos con espacio en el disco duro limitado, como puede ser un teléfono móvil o una tablet. Pero, en caso de que fuera necesario almacenar sus datos en un disco duro, este programa no tiene grandes requerimientos de espacio.

Por todo eso se ha creído conveniente explicar brevemente en esta introducción el origen del lenguaje Java, desde una perspectiva comprensible para cualquier auditorio.

El ordenador fue inventado para facilitar el trabajo intelectual. Su funcionamiento actualmente, es similar a una cadena (García-Foncillas et al., 2016) (figura 1.9).

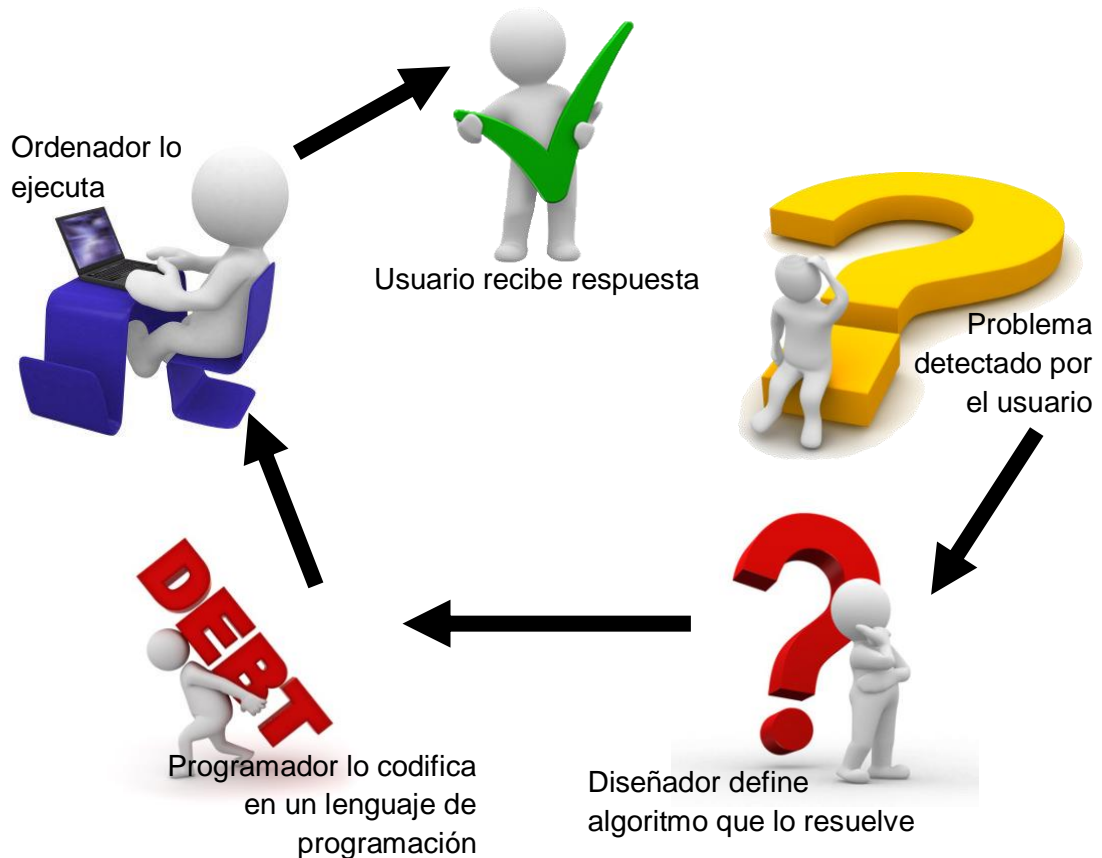


Figura 1.9 Esquema de trabajo, interactiva, en el planteamiento y solución de problemas, desde la informática. Fuente: García-Foncillas (2015).

Un programa informático es una secuencia de órdenes escritas para el ordenador con el objetivo de que resuelva un problema o realice una acción.

Los lenguajes de programación son el medio de comunicación entre el hombre y la máquina, son una forma de expresión del conocimiento.

Al principio, los programas eran muy lentos, ya que se hacían en binario. Rápidamente, a principios de la década de los 50, se evolucionó creando el código de ensamblaje (ASSEMBLY), que mediante abreviaturas mnemotécnicas se representan las funciones básicas, como ADD (sumar), STORE (copiar)... Al principio, la traducción del código de ensamblaje al código máquina se realizaba de forma manual, pero pronto se inventó un programa traductor ensamblador conocido como ASSEMBLER (Trigo Aranda, 2004).

Con los años y gracias al desarrollo de la informática estos lenguajes de programación primitivos fueron sustituidos por otros de más alto nivel y más sencillo manejo, puesto que están estructurados de forma que son más intuitivos y se asemejan más al pensamiento humano. Algunos ejemplos son BASIC, PASCAL, C, etc. En la actualidad existen cientos de lenguajes de programación, pero solo unos pocos son ampliamente utilizados. A continuación, se expone de forma breve el lenguaje de programación más relacionado con JAVA, que es el que hemos utilizado en BIOIN.

El Lenguaje C, fue creado por K. Thompson y D. Ritchie, famosos informáticos entre otras cosas por ser los precursores del sistema operativo UNIX, en 1969, mientras trabajaban para los Laboratorios Bell, en New Jersey.

Thompson desarrolló en 1970, un lenguaje experimental, que denominó "B". En 1972 su compañero Ritchie, basándose en este lenguaje creó uno nuevo al que denominó C (UNIX en C). Este lenguaje C fue uno de los más utilizados debido a su gran portabilidad y a la gran cantidad de prestaciones que tiene. Posteriormente, en 1984, se creó una evolución de este lenguaje al que se llamó C++, que aumentaba la potencialidad de su precursor. C++ está principalmente enfocado a la programación orientada a objetos (Trigo Aranda, 2004).

El lenguaje Java, conocido previamente como OAK, es ampliamente utilizado hoy en día en Internet. Se creó en 1990 por J. Gosling, de Sun Microsystems, a partir de los lenguajes C y C++.

La intención con la que Sun desarrolló este lenguaje era la de crear un interfaz atractivo e intuitivo para electrónica de consumo como calculadoras, televisión interactiva... En cambio, en estos dispositivos no tuvo aceptación y durante sus primeros años este lenguaje permaneció aparcado. B. Joy, cofundador de Sun, lo consideró un lenguaje muy útil para desarrollarlo a través de Internet y lo adaptó para tal uso. En 1995, ya denominado como JAVA, se presentó al público.

Internet ha hecho que Java se corone líder de los lenguajes de programación, y a su vez, Java ha tenido un importante efecto sobre Internet, porque además de hacer más sencilla la programación Web, permitió innovar con un nuevo tipo de programación para la red llamado *applet* que revolucionó el mundo "*on line*". Con el

desarrollo de Java se resolvieron algunos de los problemas más difíciles que presentaba hasta ese momento Internet, la portabilidad y la seguridad.

Un *applet* es un programa especial en Java diseñado para transmitirse a través de Internet y es ejecutado automáticamente por un navegador compatible con Java. Un *applet* es descargado si el usuario hace clic sobre un vínculo que lo contiene. Se descarga automáticamente y es ejecutado en el navegador. Los *applets* son pequeños programas que se utilizan normalmente para desplegar datos proporcionados por el servidor, gestionar entradas del usuario, o proveer funciones simples. En resumen, el *applet* permite a algunas funcionalidades ser movidas del servidor al cliente.

El nacimiento de los *applets* cambió la programación para Internet como hasta entonces se conocía, porque aumentó enormemente la cantidad de objetos que pueden ser movidos libremente en el ciberespacio. Existen dos tipos de objetos que se transmiten entre servidores y clientes, información pasiva y los programas activos. Un ejemplo de información pasiva es leer un correo electrónico, también lo es la descarga de un programa, porque el código del programa es sólo información pasiva hasta que se lo ejecuta. Los *applets* en cambio son dinámicos ya que ellos mismos se ejecutan. Estos programas dinámicos en internet son muy utilizados, pero tienen defectos respecto a seguridad, porque un programa que se instala y se ejecuta solo en el ordenador de un usuario debe ser vigilado para evitar que tenga contenido malicioso y pueda ocasionar daños.

Al descargar un programa en nuestro ordenador lo sometemos a un riesgo, porque el código que hemos descargado puede contener virus, Troyano o algún otro código malicioso que pueden causar daños porque se le da acceso a los recursos del sistema.

Para garantizar la seguridad de los clientes que trabajaban con estas *applets* se evitó que pudieran hacer ese tipo de acciones. Por eso Java limita a los *applets* a ser ejecutados en un ambiente controlado sin que tengan acceso a los recursos completos del sistema.

Esta posibilidad que nos ofrece JAVA de descargar *applets* con la seguridad de que no van a causar daño, es considerada por muchos la característica más

importante e innovadora de Java (Trigo Aranda, 2004).

Otra característica cuya importancia ha ido incrementándose en los últimos años, es la portabilidad, debido a la existencia de muchos y diferentes tipos de dispositivos y sistemas operativos. Un programa de Java que vaya a ser ejecutado sobre cualquier dispositivo conectado a la red, debe tener la posibilidad de habilitar al programa para que se pueda ejecutar en diferentes sistemas. No es práctico ni rentable necesitar versiones diferentes del *applet* para diferentes sistemas, es óptimo que el mismo código funcione, que sea un código ejecutable portable.

Otro concepto importante que se debe conocer, es el *bytecode*. Es un conjunto de instrucciones altamente optimizado, que está diseñado para ejecutarse a través una máquina virtual denominada Java Virtual Machine (JVM, por sus siglas en inglés). Esta máquina virtual inicialmente fue creada como un intérprete de *bytecode*. Esto puede llamarnos la atención puesto que muchos lenguajes de programación en la actualidad están creados para ser compilados en código ejecutable con el objetivo de lograr el mayor rendimiento. Sin embargo, el hecho de que un programa en Java sea ejecutado por la JVM permite resolver los problemas asociados con los programas basados en Web.

Traducir un programa Java en *bytecode* hace que su ejecución sea mucho más simple, y esto es debido a que para cada plataforma sólo es necesario implementar el intérprete de Java. Una vez que el sistema de ejecución existe para una plataforma determinada, cualquier programa de Java puede ejecutarse ya que, aunque los detalles de la JVM sean distintos en las distintas plataformas, todas ellas interpretan el mismo Java *bytecode*. Además, la ejecución del *bytecode* a través de la JVM es la manera sencilla de crear código realmente portable.

Que Java sea interpretado también mejora su seguridad, además de ciertas restricciones que existen en JAVA. La ejecución de cada programa de Java está bajo el control de la JVM, ésta es capaz de contener el programa e impedir efectos no deseados en el resto del sistema

Aunque Java fue diseñado como un lenguaje interpretado, esto no impide la compilación del *bytecode* en código nativo para aumentar su rendimiento. Debido a esta razón, Sun empezó a distribuir su tecnología *HotSpot* poco tiempo después del

primer lanzamiento de Java. *HotSpot* proporciona un compilador de *bytecode* a código nativo denominado *Just-in-Time* (JIT por sus siglas en inglés).

Cuando un compilador JIT es parte de la JVM, partes de *bytecode* son compiladas en código ejecutable en tiempo real sobre un esquema de pieza por pieza. Una consideración importante es que no es práctico compilar un programa de Java en código ejecutable en su totalidad, porque Java realiza varias revisiones durante el tiempo de ejecución que no podrían ser realizadas. Un compilador JIT compila código al tiempo que lo va necesitando durante la ejecución.

Se va a explicar a continuación el concepto de *servlet*. Un *servlet* es un pequeño programa que se ejecuta en el servidor. Así como los *applets* extienden dinámicamente la funcionalidad del navegador Web, los *servlets* extienden la del servidor Web. Tras la aparición de los *servlets*, Java se convirtió en un lenguaje de programación útil tanto para usuarios como para servidores.

Los *servlets* sirven para enviar al usuario contenido creado y generado dinámicamente. A pesar de que hoy en día ya existen varios mecanismos para generar contenido dinámico en la Web, como Common Gateway Interface (GCI), los *servlets* tienen la ventaja de tener un mejor rendimiento. Además, los *servlets* son muy portables porque como todos los programas de Java son compilados a *bytecode* y ejecutados por una máquina virtual, lo que nos asegura que el mismo *servlet* pueda emplearse en distintos servidores. La única exigencia es que el servidor debe tener una JVM y un contenedor de *servlets*.

A pesar de lo innovador de la primera versión de Java, pronto evolucionó a la versión Java 2, segunda generación, que Sun rebautizó como J2SE (Java 2 Platform Standard Edition). Esta nueva versión presentaba mejores características como los componentes *Swing* y la estructura de colecciones. Mejoraba la máquina virtual y ofrecía nuevas herramientas de programación.

Esta segunda versión no supuso el fin de su evolución, la primera gran actualización fue J2SE 1.3, que añadía funcionalidad y “estrechaba” el entorno de desarrollo. La siguiente versión J2SE 1.4 tenía actualizaciones, mejoras y adiciones importantes, añadió una nueva palabra clave “*assert*”, (que significa excepciones encadenadas) y añadía también un subsistema basado en canales para E/S. A

pesar del gran cambio que supuso, la versión 1.4 consiguió hacerse compatible con casi todas las versiones previas.

La siguiente versión fue J2SE 5, y sus mejoras fueron diferentes a la mayoría de las mejoras previas, principalmente expandió el alcance, poder y rango de acción del lenguaje. Las herramientas de desarrollo se denominaron JDK 5 (en inglés siglas de Java Development Kit).

Otra versión más evolucionada de Java se llama Java SE 6. Su creador decidió nuevamente cambiar el nombre de Java, y lo denominó Java Platform, Standard Edition 6. Esta versión que se construyó sobre la base de J2SE 5 tenía ciertas mejoras, aumenta la cantidad de bibliotecas en el API del lenguaje y presenta mejoras en el tiempo de ejecución.

En la actualidad, Julio de 2018, en cualquier equipo actualizado que se precie de usar recursos existentes en Internet tendrá una versión de Java SE 8.

Tras todo lo anterior, comprendemos que gracias a esta tecnología se podrá trabajar, a través de internet, en un entorno informático mucho más seguro. Los juegos en línea, chats, ver las predicciones meteorológicas de tu ciudad y escuchar tu cantante favorito sin temor a los archivos dañinos.

Pero para poder ejecutar sin problemas cualquier programa, necesitamos la consola de Java para el dispositivo en cuestión, diversamente denominada como Java Runtime Environment, Java Runtime, Runtime Environment, Runtime, JRE, máquina virtual de Java, máquina virtual, Java VM, JVM, VM, *plugin* de Java, complemento de Java o descarga de Java. Esta consola de Java viene con la instalación del JRE, tiene información acerca de la versión del entorno de ejecución, el directorio de inicio del usuario y los mensajes de error que aparezcan durante la ejecución de un *applet* o una aplicación. Esta consola nos puede mostrar los mensajes de error que aparecen durante la ejecución de una aplicación, lo que es de gran ayuda. Por defecto la consola de Java no se muestra, por lo que habrá que habilitarla a propósito.

El software de Java Web Start nos posibilita descargar y ejecutar aplicaciones Java desde Internet. Este software también nos permite activar las aplicaciones con un simple golpe de ratón, se encarga de que se esté ejecutando la última versión de

la aplicación (si tenemos activada la opción de descargas automáticas de la actualización), y nos ahorra tediosos procedimientos de instalación o actualización de los mismos.

Java Web Start se incluye en Java Runtime Environment (JRE) a partir de la versión Java 5.0, al instalar Java, Java Web Start se instalará automáticamente. El software de Java Web Start se inicia automáticamente, cuando una aplicación Java que con tecnología Java Web Start se descarga por primera vez. El programa de Java Web Start almacena en caché toda la aplicación de forma local en el ordenador y gracias a esto, las siguientes veces que se quiere iniciar, se hace prácticamente de forma inmediata porque todos los recursos necesarios ya están disponibles localmente. Cada vez que se inicia la aplicación, el componente de programa de Java Web Start comprueba en la página web de la aplicación si existe una nueva versión y en caso afirmativo, la descarga e inicia automáticamente, siempre que se tenga activa la opción de actualización automática (JAVA, 2017).

La forma de ejecutar un programa depende del medio desde el cual se quiera realizar la acción. Desde un explorador es necesario marcar con el ratón sobre un enlace de una página web. Desde el icono del escritorio. Si se emplea frecuentemente una aplicación se puede crear un acceso directo en el escritorio o en el menú Inicio. Java Web Start pregunta si se desea crear accesos directos o una entrada en el menú Inicio. Si la respuesta es 'Sí', a partir de ese momento, la aplicación se puede iniciar sin un explorador desde un visor de caché de la aplicación Java (Trigo Aranda, 2004).

3. ANESTHSOM

3.1. Adaptación del método ACSOM a Anestesiología: ANESTHSOM.

ANESTHSOM es una adaptación de ACSOM para anestesiología. Ambos modelos se crearon en base a los siguientes principios:

1.- Según los esquemas de contagio (base de la estadística actuarial en la que se basan las compañías aseguradoras), la probabilidad de sufrir un segundo accidente entre quienes ya han tenido uno, es mayor a la de tener un accidente entre quienes no han tenido ninguno.

2.- Según la Segunda Ley de los Grandes Números, los sucesos muy raros ocurren pocas veces. Su interpretación complementaria es: *si un suceso ocurre a menudo “no es raro”*.

Según estos dos principios, abandonamos la tan extendida idea de que cualquier riesgo puede ocasionar cualquier lesión, y además ya no se clasifica la probabilidad de riesgo según el criterio de inspector en baja, media o alta, sino que se clasifica cuantitativamente según su frecuencia de aparición en los datos analizados. Consecuentemente, la mejor forma de predecir los accidentes de una empresa o de un hospital es con el análisis de los accidentes en dicho lugar.

En primer lugar, para adaptar el método ACSOM a Anestesiología y Reanimación, hubo que crear una Hoja de Recogida de Datos (figura 1.10), para ello se enumeraron, evitando duplicidades y pérdida de información, los Factores de Riesgo y Complicaciones más frecuentes en esta especialidad. Se aplicó metodología Delphi simplificada o también conocida como mini-Delphi (Astigarraga, 2000) para la creación de esta Hoja de Recogida de Datos y posteriormente la creación de la matriz FRxC que nos proporcionará ANESTHSOM (Rubio García et al., 2014).

Se obtuvo una matriz con 15 Factores de Riesgo (FR) y 19 Complicaciones (C), que son las que aparecen en la figura 1.10 (Rubio García et al., 2014; Bona Gracia et al., 2017 y Bona Gracia et al., 2018):

FR FACTORES DE RIESGO		C COMPLICACIONES	
FR DEL PACIENTE		C. en la intubación	
FR1	Alteraciones antropométricas	C1	Rotura diente/s
FR2	Alteraciones cardio-circulatorias	C2	Edema de glotis
FR3	Alteraciones respiratorias	C. Cardio-circulatorias	
FR4	Alteraciones metabólicas/obesidad	C3	Arritmias
FR5	Alteraciones renales	C4	IAM/Angor
FR6	Alteraciones neurológicas	C5	TEP
FR7	Alteraciones hematológicas	C6	Shock
FACTORES HUMANOS		C. respiratorias	
FR8	Inexperiencia	C7	Laringo/Broncoespasmo
FR9	Técnica incorrecta	C8	Atelectasia
FR10	SITUACION URGENTE <input type="checkbox"/>	C9	Aspiración-Neumonía
PROBLEMAS EN LA PUNCION		C10	C. Metabólicas <input type="checkbox"/>
FR11	Punción dural en epidural	C11	C. Renales <input type="checkbox"/>
FR12	Punción carótida/humeral	C12	C. Neurológicas
FR13	PROBLEMAS EN INTUBACION PREVIOS <input type="checkbox"/>	C13	Cefalea
FR14	FARMACOS (especificar cuál)	C14	Les. Nervio/Bloq. N. prolongado
			Coma
FR15	OTROS	C. Generales	
		C15	Temblores/Agitación
		C16	Hiperalgnesia postoperatoria
		C. en punciones	
		C17	Neumotórax
		C18	Infección
		C19	Hematoma

Figura 1.10 Hoja de Recogida de Datos para ANESTHSOM. Fuente: Rubio García (2012).

Factores de Riesgo.

1. FR del paciente:

- a. FR 1. Alteraciones antropométricas.
- b. FR 2 Alteraciones cardio-circulatorias.
- c. FR 3. Alteraciones respiratorias.
- d. FR 4. Alteraciones metabólicas.
- e. FR 5. Alteraciones renales.
- f. FR 6. Alteraciones neurológicas.
- g. FR 7. Alteraciones hematológicas.

2. FR humanos:

- a. FR 8. Inexperiencia.
- b. FR 9. Técnica incorrecta.

3. FR 10. Situación urgente.

4. FR propios de técnicas invasivas:

- a. FR 11. Punción dural en epidural.
- b. FR 12. Punción arterial no deseada.

5. FR 13. Problemas previos en la intubación.

6. FR 14. Fármacos.

7. FR 15. Otras no clasificadas anteriormente.

Complicaciones:

1. C. En la intubación.
 - a. C1. Rotura de diente/s.
 - b. C2. Edema de glotis.
2. C. Cardio-circulatorias.
 - a. C3. Arritmias.
 - b. C4. Infarto agudo de miocardio (IAM)/Ángor.
 - c. C5. Tromboembolismo pulmonar (TEP).
 - d. C6. Shock.
3. C. Respiratorias.
 - a. C7. Laringo/Broncoespasmo.
 - b. C8. Atelectasia.
 - c. C9. Aspiración/Neumonía.
4. C 10. C. Metabólicas.
5. C 11. C. Renales.
6. C 12. Neurológicas.
 - a. C12. Cefalea.
 - b. C13. Lesión Nerviosa/Bloqueo Nervioso Prolongado.
 - c. C14. Coma.
7. C. Generales.
 - a. C15. Temblores/Agitación.
 - b. C16. Hiperalgia postoperatoria.
8. C. en Punciones.
 - a. C17. Neumotórax.
 - b. C18. Infección.
 - c. C19. Hematoma.

3.2. Análisis de la matriz FR x C.

Los datos se recogen en forma de tabla de contingencia (tabla 1.1), la metodología desarrollada es capaz de analizar sus distintos componentes. A grandes rasgos, el análisis de los datos del cuerpo de la tabla nos informa de las relaciones entre Factores de Riesgo y Complicaciones y también de las relaciones entre Factores de Riesgo entre si y Complicaciones entre sí. En cambio, el análisis de la columna marginal nos informa de las relaciones y/o diferencias entre los Factores de Riesgo o Complicaciones del hospital que estamos analizando y el patrón, por lo que es el análisis que nos ayuda a controlar y planificar estrategias de prevención.

A continuación, se va a realizar un pequeño resumen de la metodología de ANESTHOM, que será vista con más detalle en el capítulo II de Material y Métodos

C. FR.	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	Total
FR1																				
FR2																				
FR3																				
FR4																				
FR5																				
FR6																				
FR7																				
FR8																				
FR9																				
FR10																				
FR11																				
FR12																				
FR13																				
FR14																				
FR15																				
Total																				

Tabla 1.1: Matriz resumen (tabla de contingencia) de Factores de Riesgo y Complicaciones en ANESTHOM. Fuente: Rubio García et al. (2014).

En el cuerpo de la tabla se recogen los valores calculados tras la aplicación de distintas técnicas estadísticas multivariantes mediante paquete estadístico SPSS 15, SPSS19 y SPSS23 y se dividen en varias fases:

En la primera fase, en primer lugar, se realiza un análisis exploratorio previo a través de los residuales estandarizados de Habertman (García Felipe et al., 2004; Rubio García, 2004 y 2012). De la lectura de estos residuales se obtiene una idea global de la situación de las relaciones entre los Factores de Riesgo y las Complicaciones, por tanto, puede considerarse como un *screening* previo, analítico, porque solo da información en dos dimensiones.

En segundo lugar, se seleccionan los residuales mayores de 1,96, lo que quiere decir que existe relación con una significación del 95%. De estos residuales seleccionados, consideramos todas las relaciones entre FR_i y C_j, desde el punto de vista de cada Factor de Riesgo (mayor diferencia en la fila) desde el punto de vista de la Complicación (mayor diferencia en columna) y en conjunto.

En tercer lugar se realiza en esta primera fase otro análisis exploratorio, un análisis de escalamiento multidimensional o MDS (Conte et al., 2011a) que nos permite representar en un gráfico la relación entre las categorías de factores de riesgo FR_i entre si y entre las de complicaciones C_j entre sí (intrarrelaciones), pero no entre FR y C (interrelaciones). Nos ayuda a obtener evaluaciones comparativas entre objetos cuando las bases específicas no se conocen o no están definidas.

En una segunda fase se aplica el análisis de correspondencias, luego el de conglomerados. En paralelo al análisis de conglomerados, utilizando también las puntuaciones factoriales, se aplica un Análisis Factorial de componente principales (AF) (Rubio García, 2012).

En una tercera fase se obtiene una medida de distancia, Distancia de Minkowski, entre cada Factor de Riesgo y cada Complicación, que complementa la información obtenida en los métodos anteriores, pues a esta nueva matriz se le aplica de nuevo un Análisis Factorial de componentes principales y escalamiento multidimensional, con todo esto se obtiene el grafico del ANESTHOM (Rubio García et al, 2014).

El análisis de los marginales de la tabla el que se utilizó en BIOIN ANESTHSOM, sirve fundamentalmente para contrastar cada unidad que se quiera comparar (sección hospitalaria, hospital, provincia, región, etc.) con el patrón elegido. Los dos estadísticos que se utilizaron fueron los Contrastes y Translocaciones. Al valorarlos en conjunto aparecen las fases de actuación. Los Factores de Riesgo o Complicaciones que aparecen en primera fase, son los que se deben tratar con prioridad (Rubio García, 2012). En cambio, los que aparecen en la última fase, son los que tienen mejores resultados en la unidad analizada que en el patrón (Bona Gracia et al., 2018). Del análisis de estos datos se obtiene información para la planificación y el control de los incidentes más frecuentes del hospital analizado (Rubio García et al., 2014 y Bona Gracia et al., 2017).

Como ejemplo, y para ilustrar el método, vamos a ver a continuación el análisis realizado por Rubio García et al. (2014) de los Factores de Riesgo del bloque General del Hospital Universitario Miguel Servet con el patrón, que en este caso es el Hospital Universitario Miguel Servet en todas sus secciones: General; Materno-infantil y Traumatología.

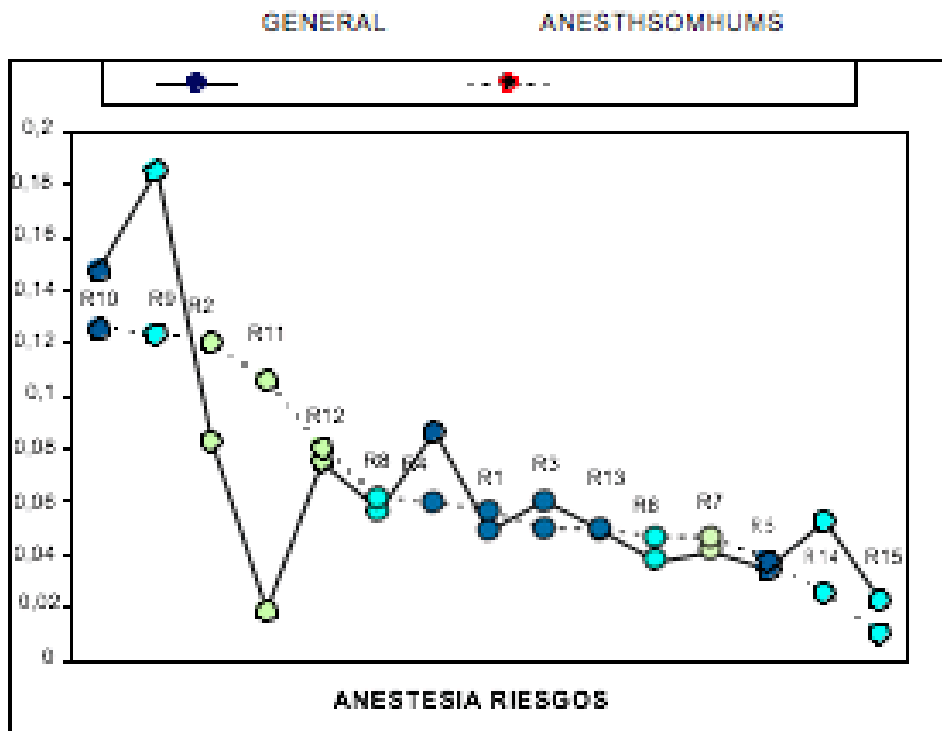


Figura 1.11 Poligonal de riesgos del Bloque General vs HUMS. Fuente: Rubio García et al. (2014).

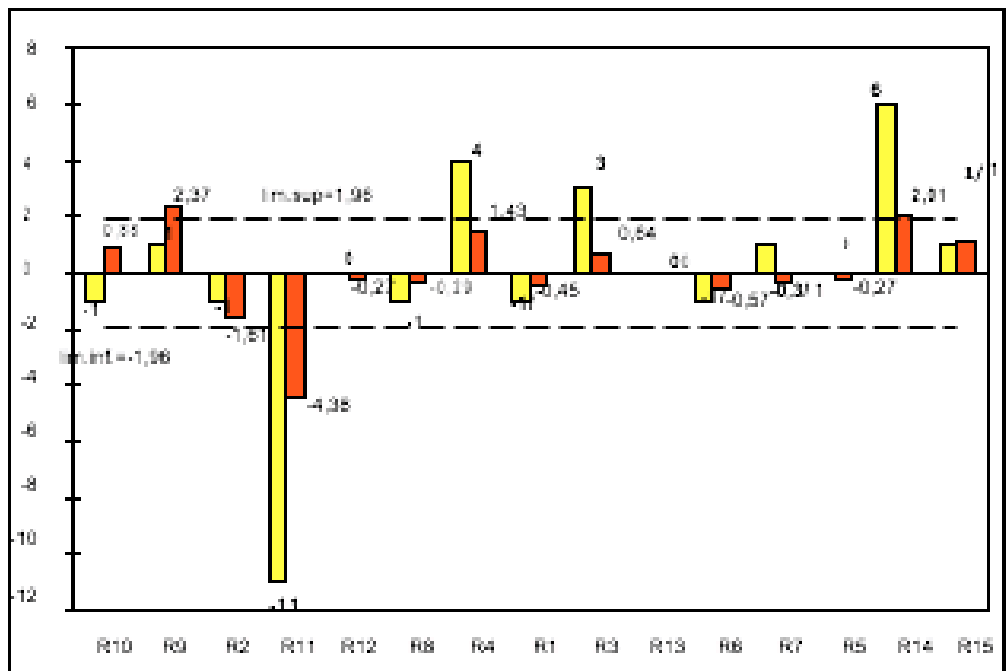


Figura 1.12 Translocaciones y Desviaciones de los riesgos del Bloque general vs HUMS. Fuente: Rubio García et al. (2014).

El análisis consiste en la comparación del polígono de frecuencias de los Factores de Riesgo del Bloque General con el patrón elegido. Con la gráfica se obtiene la situación relativa de cada uno. Para objetivar y priorizar los puntos más débiles y vulnerables es necesario el uso de métodos analíticos.

Se expone a continuación, un resumen del procedimiento que se ha seguido en este caso concreto. Se llevan a cabo tres pasos. En el primero, se tiene la planificación Tipo I o “secuencia bruta” (figura 1.13), en la que se presentan las posiciones que ocupan los diferentes Factores de Riesgo de la sección Bloque General junto con el marginal del patrón (Hospital Miguel Servet) (Rubio García et al., 2014).

En segundo lugar, se obtiene la planificación Tipo II o “secuencia semidepurada” (figuras 1.14 y 1.15), donde se consideran las Translocaciones o Desviaciones por separado.

Por último, para obtener de forma ordenada o priorizada los Factores de Riesgo, se consideran ambas técnicas a la vez, las Translocaciones y Desviaciones conjuntamente, así se consiguen dos posibles ordenaciones, según si se considera en primer lugar Translocaciones o Desviaciones. Con todo esto conseguimos establecer las fases de actuación prioritaria. La primera fase será la de actuación prioritaria, es decir, en la que hay que invertir mayores esfuerzos y recursos porque debe ser mejorada. En cambio, en la última fase están los Factores de Riesgo que están en la mejor situación en esa sección. Con este procedimiento se resuelve la Planificación Tipo III (figura 1.16). Son “secuencias depuradas” ya que incluyen las dos formas de optimizar.

SECUENCIA DE PLANIFICACIONES. FR DEL BLOQUE GENERAL

Posición GENERAL														
R ₉	R ₁₀	R ₄	R ₂	R ₁₂	R ₃	R ₈	R ₁₄	R ₁	R ₁₃	R ₇	R ₆	R ₅	R ₁₅	R ₁₁
Posición ANESTHSOMHUMS														
R ₁₀	R ₉	R ₂	R ₁₁	R ₁₂	R ₈	R ₄	R ₁	R ₃	R ₁₃	R ₆	R ₇	R ₅	R ₁₄	R ₁₅

Figura 1.13 Planificación Tipo I (secuencia bruta). Fuente: Rubio García et al. (2014).

Translocación GENERAL / ANESTHSOMHUMS														
Avanzada					Igual					Retrocedida				
Fase1					Fase2					Fase3				
R ₁₄	R ₄	R ₃	R ₉	R ₇	R ₁₅	R ₁₂	R ₁₃	R ₅	R ₁₀	R ₂	R ₈	R ₁	R ₆	R ₁₁

Figura 1.14 Planificación de Tipo-II (secuencia semidepurada, criterio translocación) Fuente: Rubio García et al. (2014).

Desviación GENERAL / ANESTHSOMHUMS														
Exceso		Igual											Defecto	
Fase 1		Fase 2											Fase 3	
R ₉	R ₁₄	R ₄	R ₁₅	R ₁₀	R ₃	R ₁₃	R ₁₂	R ₅	R ₈	R ₇	R ₁	R ₆	R ₂	R ₁₁

Figura 1.15 Planificación de Tipo-II (secuencia semidepurada, criterio desviaciones estadísticas) Fuente: Rubio García et al. (2014).

Desviación-Translocación GENERAL / ANESTHSOMHUMS														
Exceso		Igual											Defecto	
Fase1		Fase2				Fase3				Fase4				Fase5
R ₉	R ₁₄													
		R ₄	R ₁₅	R ₃	R ₇									
						R ₁₃	R ₁₂	R ₅						
									R ₁₀	R ₈	R ₁	R ₆	R ₂	
														R ₁₁

Figura 1.16 Planificación Tipo-III secuencia depurada, conjunta, priorizando desviaciones Fuente: Rubio García et al. (2014).

Translocación –Desviación Sección 1/ ANESTHSOM HUMS														
Exceso		Igual					Defecto							
Fase1		Fase2				Fase3			Fase4				Fase5	
R ₁₄	R ₉													
		R ₄	R ₃	R ₁₅	R ₇									
						R ₁₃	R ₁₂	R ₅						
									R ₁₀	R ₈	R ₁	R ₆	R ₂	
														R ₁₁

Figura 1.17 Planificación Tipo-III secuencia depurada, conjunta, priorizando translocaciones Fuente: Rubio García et al. (2014).

Resultado final. ANESTHSOM.

Al igual que de ACSOM se obtiene una gráfica, de ANESTHSOM se obtiene otra en la que se ven agrupadas por colores los Factores de Riesgo y las Complicaciones.

También se obtienen las relaciones principales entre Factores de Riesgo y Complicaciones (Rubio García et al., 2014).

En ANESTHSOM que se presenta a continuación, corresponde al Hospital Universitario Miguel Servet de Zaragoza, que son los datos con los que se trabajó y creó inicialmente ANESTHSOM (Rubio García et al., 2014) (figura 1.18).

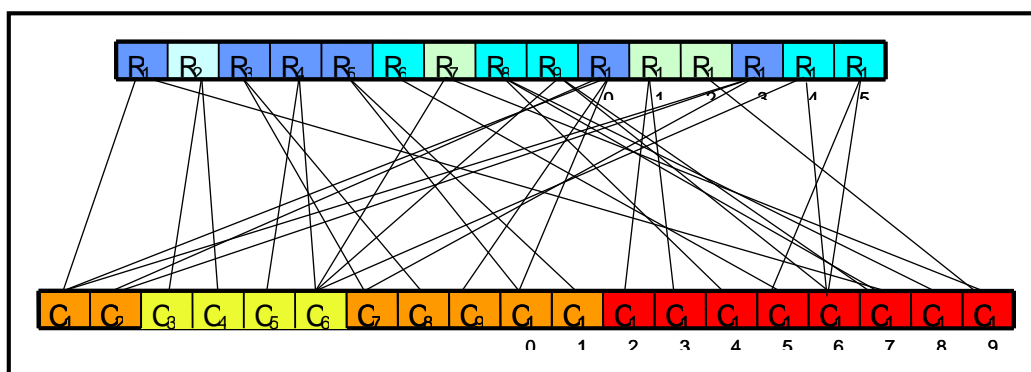


Figura 1.18 ANESTHSOM HUMS. (Rubio García et al., 2014).

4. JUSTIFICACIÓN Y OBJETIVO.

Justificación:

La seguridad del paciente, aunque no es concepto nuevo, ha cobrado una gran importancia en los últimos tiempos, principalmente desde que en 1999 se publicara, como ya hemos visto, *“To err is human: building a safer health system”*. Este artículo fue el punto de inflexión a partir del cual los sanitarios y el público en general, comenzamos a ser conscientes de las grandes consecuencias que tienen los actos médicos, que pueden ir desde el aumento de la estancia hospitalaria hasta la muerte.

Para poder implementar medidas de mejora y evitar o minimizar estos eventos adversos, es necesario objetivar y cuantificar dónde está el problema. La especialidad de Anestesiología, muy probablemente debido a la rápida consecuencia que tienen nuestras actuaciones sobre el paciente, ha encabezado este nuevo enfoque de la seguridad y la calidad asistencial. Rubio García (2012), fue el creador de ANESTHSOM, modelo matemático creado para el análisis y control de los incidentes que ocurren en anestesiología. Para facilitar y agilizar el tratamiento de los datos de incidentes ocurridos en anestesiología, que hasta este momento se realizaba de forma manual y era una tarea ardua y laboriosa, se va a aplicar el programa informático BIOIN ANESTHSOM, como ya hizo García-Foncillas (2016) con la problemática de la accidentalidad laboral.

Objetivo principal:

El objetivo de este estudio, es la creación del nuevo BIOIN ANESTHSOM, herramienta informática para el análisis de incidentes en anestesiología, mediante el método ANESTHSOM, con los datos y características correspondientes recogidos mediante la revisión de Historias Clínicas de los hospitales Miguel Servet y Clínico Lozano Blesa de Zaragoza, para que en un futuro pueda utilizarse como herramienta de mejora de la calidad asistencial en Anestesiología.

Objetivos específicos:

Comprobar si ha habido cambios al comparar los perfiles de los resultados obtenidos en el trabajo previo ANESTHSOM, con los datos actuales analizados con BIOIN ANESTHSOM.

Evaluar con el nuevo patrón, los perfiles de Factores de Riesgo y Complicaciones de cada sección del Hospital Universitario Miguel Servet y de cada hospital estudiado.

Describir que Factores de Riesgo y Complicaciones de cada sección del Hospital Universitario Miguel Servet y cada hospital analizado, necesitan actuación prioritaria por encontrarse en peor situación en las entidades a estudio que en el patrón.

Comprobar que BIOIN ANESTSHSOM facilita el tratamiento estadístico de los datos, haciéndolo más rápido y sencillo comparado con la metodología previa ANESTSHOM.

CAPÍTULO II

MATERIAL

Y

MÉTODOS

1. MATERIAL

Para la validación de nuestro programa BIOIN ANESTHSOM, se recogieron durante los años 2015 y 2016, los datos de los incidentes registrados mediante la revisión de 813 historias clínicas de pacientes quirúrgicos de forma retrospectiva, de las cuales 225 casos fueron en el Hospital Clínico Lozano Blesa y 588 del Hospital Universitario Miguel Servet, de las que 250 fueron del bloque General, 160 casos de la sección Materno-infantil y 190 de Traumatología.

El patrón es la suma de las matrices de los hospitales, Miguel Servet, (Traumatología, General y Materno-infantil) y del Lozano Blesa, 813 casos entre ambos hospitales.

Se utilizarán matrices bidimensionales como se hizo en el sistema ANESTHSOM, donde figurarán los marginales totalizados para los Factores de Riesgo y las Complicaciones que BIOIN ANESTHOM manejará, es decir, 15 Factores de Riesgo y 19 Complicaciones, y constituye el documento fuente que, a posteriori se incluirá para su tratamiento y análisis. Su distribución se verá en la tabla 3.1 y 3.2 del capítulo de Resultados y Discusión.

2. MÉTODO

Antes de continuar, es importante aclarar que esta monografía, al ser un trabajo fundamentalmente metodológico, algunas ideas se solapan. Conceptos ya vistos en la Introducción pueden haberse visto repetidos en Material y Métodos, al igual que algunos conceptos explicados en este capítulo se repetirán o nos referiremos a ellos en el capítulo siguiente de Resultados y Discusión.

2.1 Metodología ANESTHSOM

2.1.1 Análisis estadístico del cuerpo de la tabla

Como ya vimos en la introducción, en el cuerpo de la tabla de la matriz de Factores de Riesgo y Complicaciones (FRXC), se recogen los valores calculados con distintas técnicas estadísticas multivariantes mediante el paquete estadístico SPSS 15, SPSS 19 y SPSS 23, y se dividen en varias fases:

Primera fase: Se realizó un análisis exploratorio previo a través de los residuales estandarizados de Habertman (García Felipe, Álvarez Zarate, Alcalde Lapidra, Rubio García y Bascuas Hernández, 2004; Rubio García, 2004).

A continuación, sobre esta tabla, FRxC, se realizó un triple análisis:

Primero se seleccionan los residuales mayores de 1,96, es decir, aquellos que indican relación con una significación del 95%. A partir de estos residuales, se consideran todas las relaciones entre FR_i y C_j , desde el punto de vista de cada Factor de Riesgo (mayor diferencia en la fila) desde el punto de vista de cada Complicación (mayor diferencia en columna) y en conjunto.

En segundo lugar, se realizó un Escalamiento Multidimensional para determinar cuántos grupos se formaban en cada ítem estudiado (Factores de Riesgo y Complicaciones, que daban lugar a los diferentes colores), resultado que se plasma en la figura 1.18 (vista en capítulo anterior) Rubio García et al. (2014).

Por último, con todo esto lo que se quiere es representar en un espacio geométrico, de pocas dimensiones, la proximidad existente entre un conjunto de

variables a partir de las similitudes entre ellos, en este caso entre FR_i y entre C_j utilizando para medir la bondad del modelo, el Índice de esfuerzo o Stress de Kruskal y el coeficiente de correlación al cuadrado o RSQ.

Este método nos aporta información del número de dimensiones necesarias. El resultado es para dos dimensiones ya que el de tres dimensiones no nos aporta más información de las relaciones en este primer análisis, aunque si se aportan las medidas de bondad del modelo para las distintas dimensiones.

Esta información obtenida se analiza, desarrolla y complementa mediante el análisis de otros estadísticos multivariantes, que permitan nuevos enfoques, principalmente con otras dimensiones no valoradas hasta ahora y mediante la cuantificación de datos cualitativos originales que refuerzan y complementan con nueva información a los anteriores.

Segunda fase: En primer lugar, se aplica el análisis de correspondencias, (Conte, Rubio, García y Cano, 2011). Es un método con el que se consigue la reducción de dimensiones, se parte de 15 en Factores de Riesgo y 19 en Complicaciones. Al aplicarlo se consigue tener el número de dimensiones necesarias para modelizar los FR o C a partir del espacio de k-dimensión, se tratará de encontrar un subespacio n-dimensional con n del menor tamaño posible pero que no se pierda información acerca de la similitud entre las categorías.

Mediante el análisis de la tabla de contingencia, se interpretan las coincidencias entre las categorías de una variable respecto a las categorías de la otra y las relaciones entre las categorías de ambas variables.

Se realiza análisis para dos, tres y cuatro dimensiones y se decide utilizar el modelo de tres dimensiones y una vez definidas estas tres dimensiones con las puntuaciones factoriales de cada una de ellas se realiza un análisis de conglomerados.

Antes de llevar a cabo el análisis de conglomerados, se hacen dos análisis de correspondencias sobre las tablas obtenidas de colapsamientos de FR o C que aparecen separados del grueso en la nube de puntos, son las fases de aclaramiento.

Conte et al. en 2011 lo denominó interacciones de aproximaciones sucesivas, se realizan en dos fases, los FR y C que se colapsen se representan en el gráfico mediante un único punto y se analizan sus relaciones.

Se realiza un análisis de conglomerados jerárquicos con las puntuaciones factoriales obtenidas previamente utilizando el método de Ward y medida de distancia euclídea al cuadrado. Con el objetivo de establecer grupos diferenciadores de FR y de C que complementen la información obtenida con los métodos previamente aplicados.

Al mismo tiempo que se realiza el análisis de conglomerados, con las mismas puntuaciones factoriales, se aplica un Análisis Factorial de componentes principales (AF). Con este análisis se ratifican los resultados del Análisis de Correspondencias, para la reducción de dimensiones y agrupamientos obtenidos mediante análisis de conglomerados se utiliza KMO y prueba de Barlett como test estadísticos previos para conocer la conveniencia de realizar el Análisis Factorial.

Se analiza también la comunalidad para ver que variables se explican mejor con ese modelo. El número de factores se decide según el porcentaje de varianza explicada, orientándonos por el gráfico de sedimentación y los autovalores. Se utilizará la matriz de componentes rotados por el procedimiento *Varimax* para una mejor solución. En el primer intento de análisis de resultados, no pudieron valorarse, ya que tanto KMO como Barlett mostraron la no idoneidad de AF.

Tercera fase: Se utilizan las puntuaciones del Análisis Factorial de Correspondencias y se obtiene una medida de distancia, distancia de Minkowski, entre cada FR y cada C, que completa la información obtenida con los análisis previos. Se ha ensayado con potencia 2, 3 y 4 y finalmente se ha decidido utilizar la potencia 3 puesto que la potencia 2 resultaba pobre en representatividad de los datos y la potencia 4 no mejoraba en nada significativamente a la información aportada por la potencia 3, todo esto se ha realizado como en todos los métodos previos, el principio de parsimonia (Hair, Anderson, Tatham y Black, 1999; Real, 2001).

Cuarta fase: A las distancias de Minkowski, se les aplica nuevamente el Análisis Factorial de Componentes Principales para mejorar los resultados del análisis factorial anterior y se obtienen nuevas agrupaciones en FR y C.

Para terminar, se aplicó otra vez a las distancias de Minkowski, el Escalamiento Multidimensional para ver si aparecía alguna información adicional puesto que esta vez el análisis sería un escalamiento multidimensional métrico.

Con lo obtenido tras todos los análisis previos se estableció una afinidad entre FR y C entre si y las asociaciones entre Factores de Riesgo y Complicaciones. La representación gráfica que resumía toda esta información se denomina ANESTHSOM.

Con este análisis previo se obtuvieron los Factores de Riesgo y Complicaciones mayores y sus relaciones. En un segundo tiempo ANESTHSOM estudió las causas (FR) y las consecuencias (C) del problema concreto que se estaba analizando, las interrelaciones (FR con C y viceversa) y las intrarrelaciones (FR con FR y C con C) es decir, los grupos más afines y las relaciones más fuertes entre ellos (Bona Gracia et al, 2017).

2.1.2 Análisis estadístico de los marginales de la tabla.

Es el análisis que se utilizó en BIOIN ANESTHSOM. Se realiza para contrastar la entidad que se quiera analizar con el patrón elegido. Los dos test estadísticos que se aplicaron fueron Contrastes y Translocaciones que, al valorarlos en conjunto, aparecen las fases de actuación o *Planning Report*. El objetivo de este análisis es la PLANIFICACION y el CONTROL, es decir, ver que FR y que C están peor en el hospital analizado que en el patrón (aparecen en primera fase) y con esos resultados tomar las medidas preventivas necesarias para mejorarlo. Se realizaron los dos análisis en paralelo, uno para los FR y otro para las C.

3. BIOIN ANESTHSOM.

De la misma forma que BIOIN ACSOM es el programa informático sobre ACSOM (Rubio Calvo et al., 2015; García-Foncillas et al., 2016), BIOIN ANESTHSOM es el programa informático que opera con ANESTHSOM (Bona Gracia et al., 2018 a), tiene una serie de características que le permiten trabajar en red a través de internet y está registrado.

A continuación, se presentan los diagramas de flujo creados para la Entrada de Datos y para la Salida de Resultados¹ (García-Foncillas, 2015; García-Foncillas et al., 2016; Bona et al., 2017) (figuras 2.1 y 2.2).

¹ NOTA. En los diagramas siguientes:

- Los símbolos que corresponden a elipses son Inicio y Final.
- Los correspondientes a romboides, Interacción con usuario.
- Los correspondientes a rectángulos, Acción
- Los correspondientes a rombos, Condicionalidad
- Los círculos son para poder poner Si o No

Diagramas de flujo.

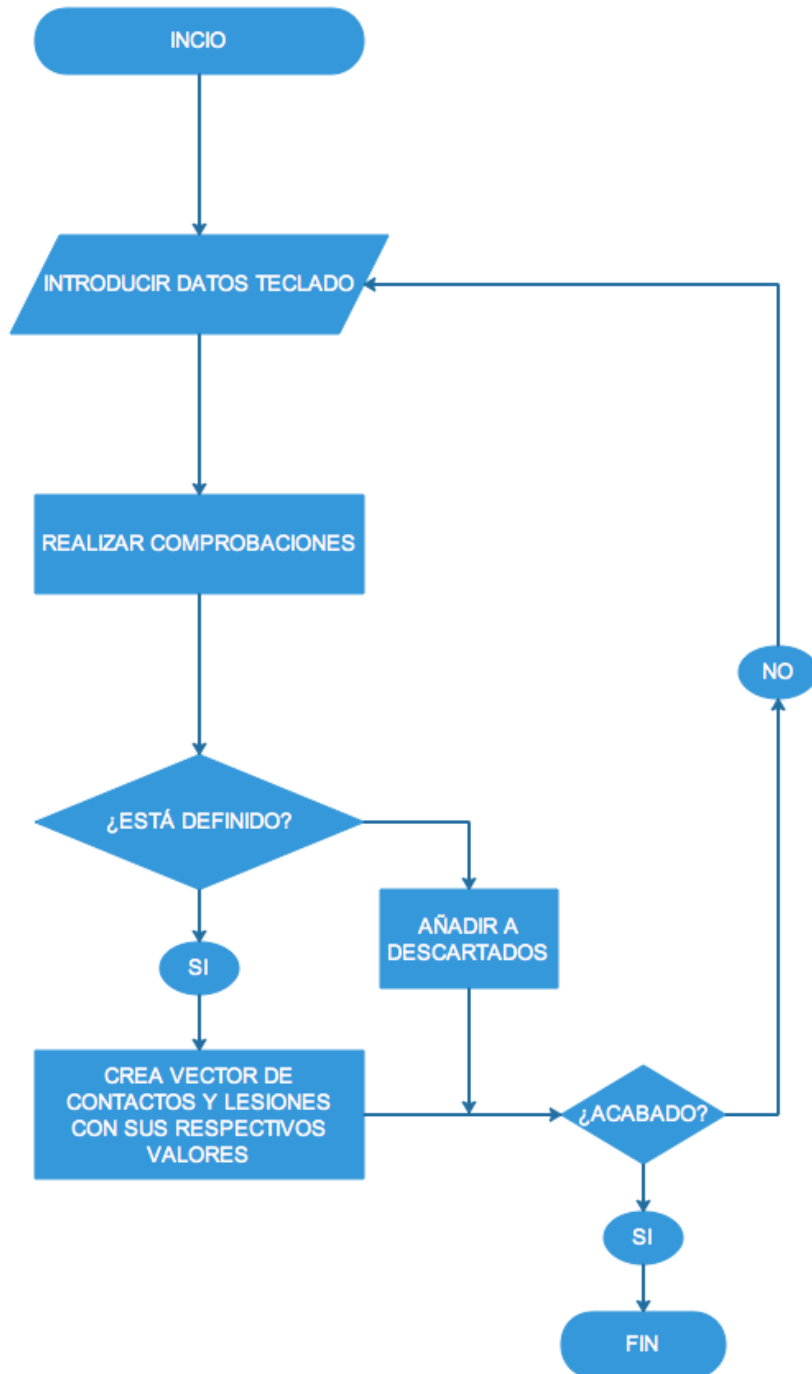


Figura 2.1 Diagrama de flujo de entrada de datos del programa informático BIOIN ANESTHSOM. Fuente: García-Foncillas (2016) y Bona Gracia et al. (2017).

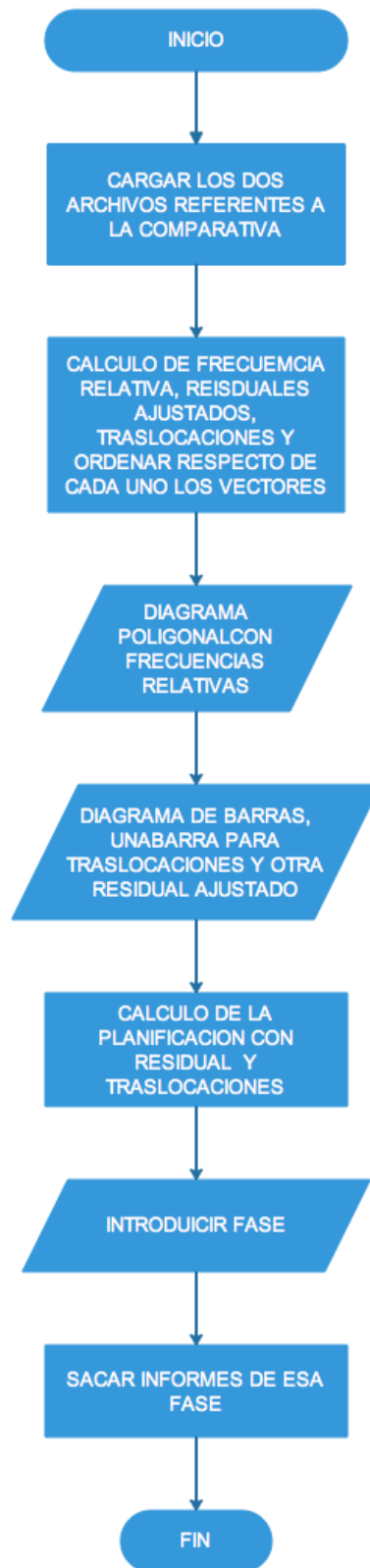


Figura 2.2 Diagrama de flujo, por pasos, de la salida de resultados del programa informático BIOIN ANESTHSOM. Fuente: García-Foncillas (2016) y Bona Gracia et al. (2017).

3.1 Programa

El programa BIOIN Omega (precursor de BIOIN DELTA), viene desarrollado en Rubio Calvo et al. (2012), el programa BIOIN DELTA en Rubio Calvo et al. (2015); García-Foncillas (2015) y García-Foncillas et al. (2016) y el programa BIOIN ANESTHSOM en Bona Gracia et al. (2018 a).

Su desarrollo se detalla en el Anexo. (CD adjunto).

Su esquema es:

Carpeta BIOIN ANESTHSOM

CONTROLADOR

Controlador.class

Controlador_1.class

MODELO

Alm1.class

Alm2.class

Alm3.class

Command.class

Contenedor.class

DiagramaBarras.class

DiagramaBarras 1.class

DibujarBarras.class

DibujarPoligonal.class

Informe.class

Informe 1.class

LISTA

ListaAlm1.class

ListaAlm2.class

PlantillaMetodo.class

Poligonal.class

Poligonal.class 1

VISTA

AcercaDe.class

AcercaDe2.class

Central.class

Central 1.class

Guardar.class

Guardar 1.class

Lienzo.class

MargenConstrains.class

Operaciones.class

OperacionesDelta.class

OperacionesInformeCompleto.class

OperacionesOIT.class

Planificacion.class

OrgBIOINomegaANESTHSOM

4. CONTROL DE CAMBIOS.

En el esquema del modo operativo que se presenta a continuación, se observa que pueden intervenir diversos usuarios que tengan relación con la aplicación. El diagrama de uso y su control se representa en las figuras siguientes (García-Foncillas, 2015; Bona Gracia et al., 2017)

Aparecen en escena diversos "actores" que intervienen en los distintos momentos y que lógicamente tienen sus correspondientes controles. En la figura 2.3, se representan los distintos actores, desde el Usuario pasando por el Inspector y el Administrador, hasta la Administración.

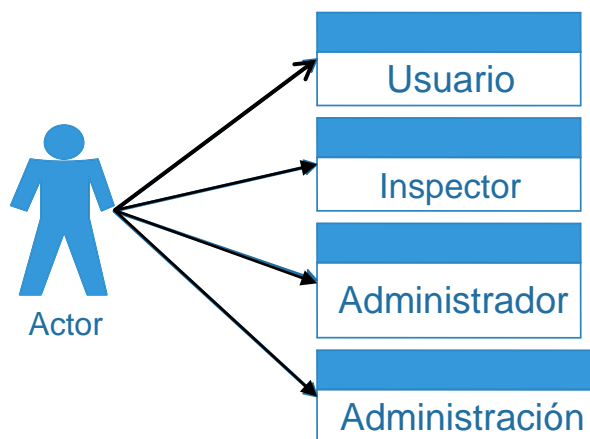


Figura 2.3 Esquema de los distintos actores de BIOIN ANESTHSOM. Fuente: Bona Gracia et al. (2017)

Se llama usuario a la persona del hospital encargada de su manejo, que quiere saber cuál es la situación real de su hospital respecto a los Factores de Riesgo y Complicaciones que presenta y los quiere comparar con otra entidad superior, con el objetivo de poder tomar las medidas preventivas necesarias para disminuir la incidentalidad. Desde el punto de vista del funcionamiento del programa, (figura 2.4) el usuario es quien se encarga de introducirse en el sistema para analizar su hospital y, además, puede cambiar su contraseña. Estas dos acciones tienen que ser validadas por el administrador (García-Foncillas et al., 2016).

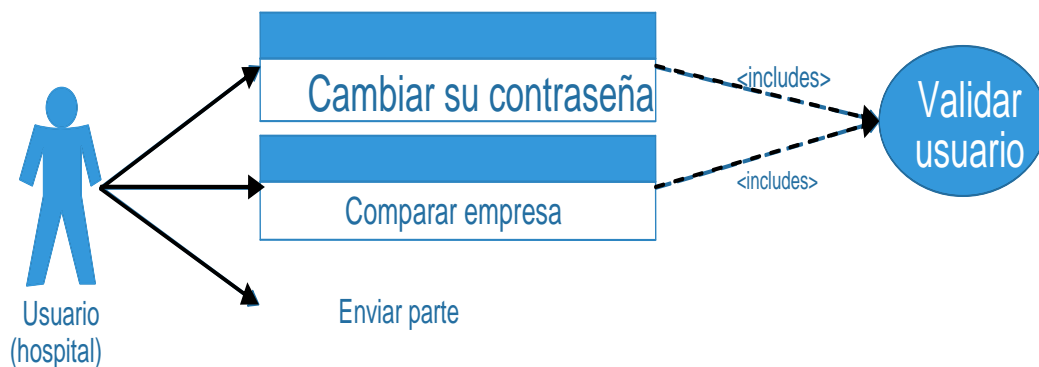


Figura 2.4 Esquema del usuario. Fuente: Bona Gracia et al. (2017)).

El inspector, es un controlador externo, que podrá conocer en aquel hospital o empresa que inspeccione, cuáles son los puntos débiles sobre los que tendrá que actuar, comparando el objeto a estudio con el patrón que considere, lo que incrementa la efectividad de la inspección y del control. En caso de que un ocurra incidente mortal o de extrema gravedad se le debe avisar obligatoriamente (figura 2.5).

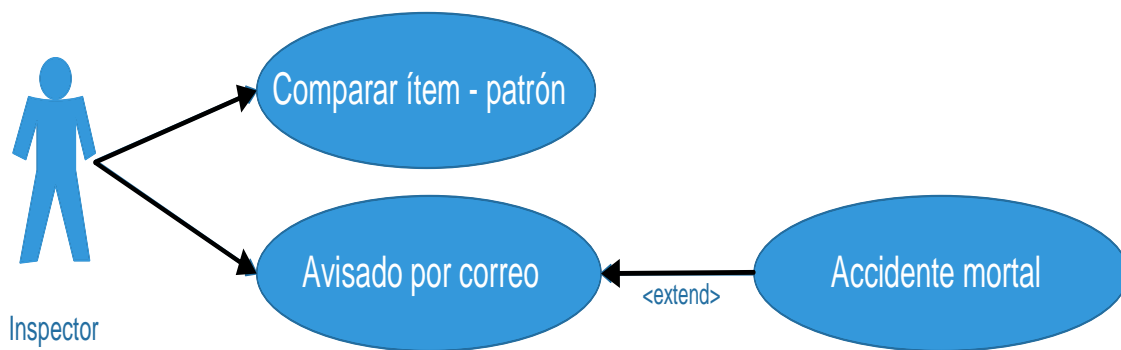


Figura 2.5 Esquema del inspector de la accidentalidad laboral. Inspector de trabajo o persona encargada de dicha labor. Fuente: Bona Gracia et al. (2017).

El Administrador o Administrativo, miembro del servicio de informática del hospital (servicio que ya existe en los hospitales) es quien se encarga de realizar las labores de mantenimiento de la aplicación, puede “añadir un incidente”, “añadir datos” y “cambiar contraseña”. Es una figura nueva que se deberá crear cuando se implante, pues actualmente el programa no está en uso. Lo que el administrativo no puede hacer es “borrar incidentes” o “borrar usuarios”, esto es competencia de la Administración (figura 2.6).

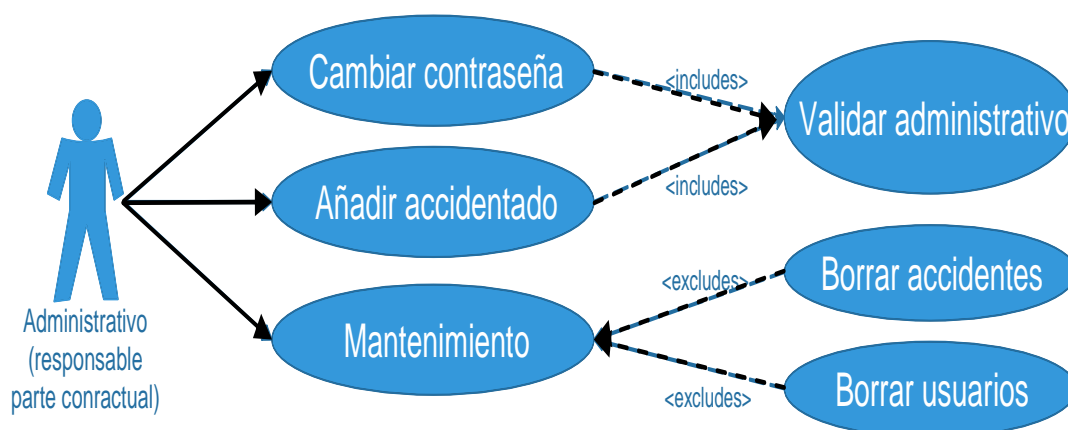


Figura 2.6 Esquema del administrador de la empresa, o de la unidad que lo utilice.

Fuente: Bona Gracia et al. (2017).

En último lugar, el epígrafe “Administración” se refiere a los equipos centrales de la administración del hospital, es el único de los cuatro actores previamente mencionados que puede analizar y controlar todas las actividades que se lleven a cabo. Es la encargada de su salvaguarda (figura 2.7)

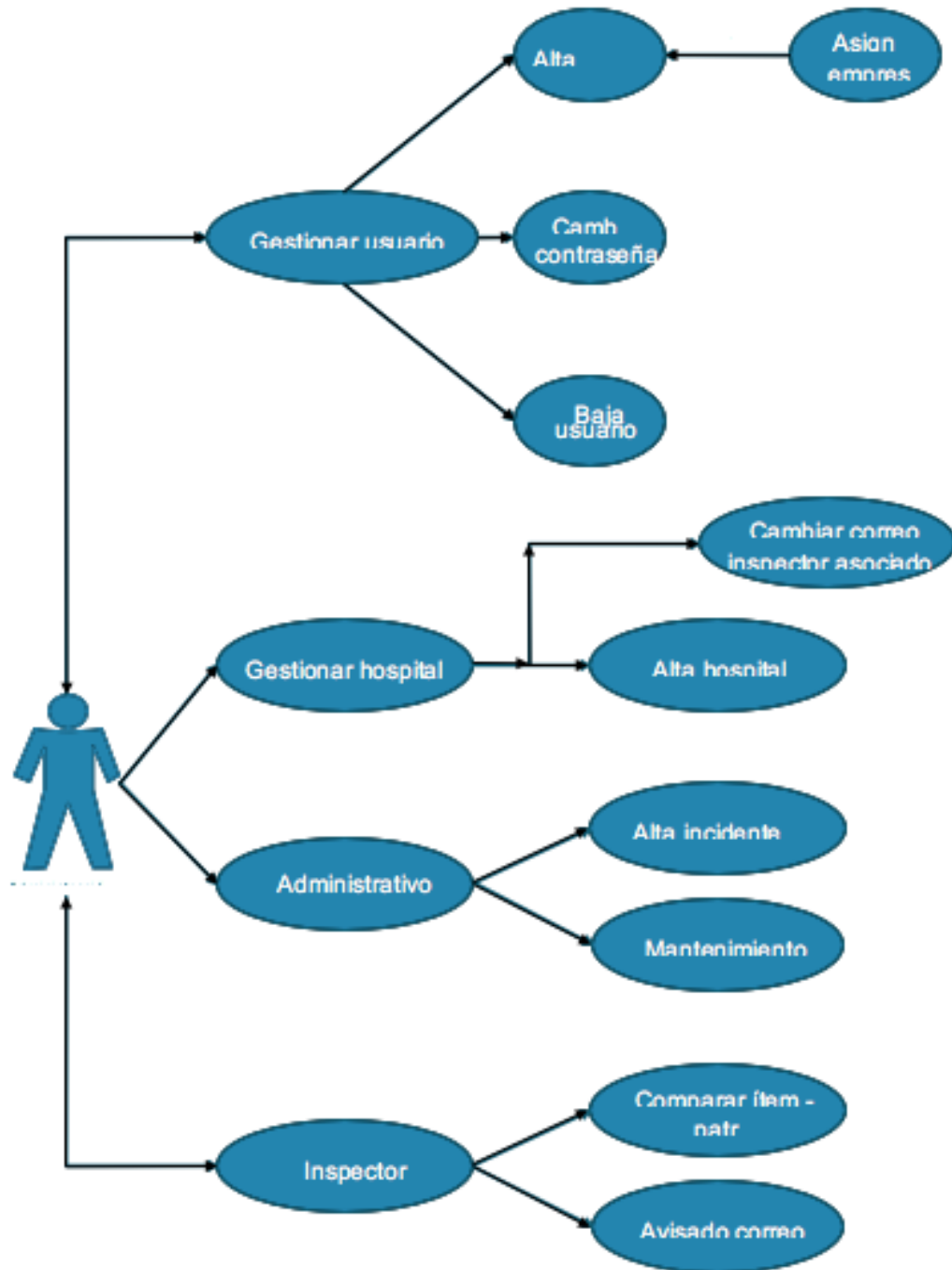


Figura 2.7 Esquema de la administración (servicios centrales) del hospital. Fuente: Bona Gracia et al. (2017).

5. APLICACIÓN BIOIN ANESTHSOM

La aplicación BIOIN ANESTHSOM presenta la siguiente cadena de funcionamiento:

- Selección del programa (figuras 2.8 y 2.9).
- Entrada de datos
- Obtención de los resultados

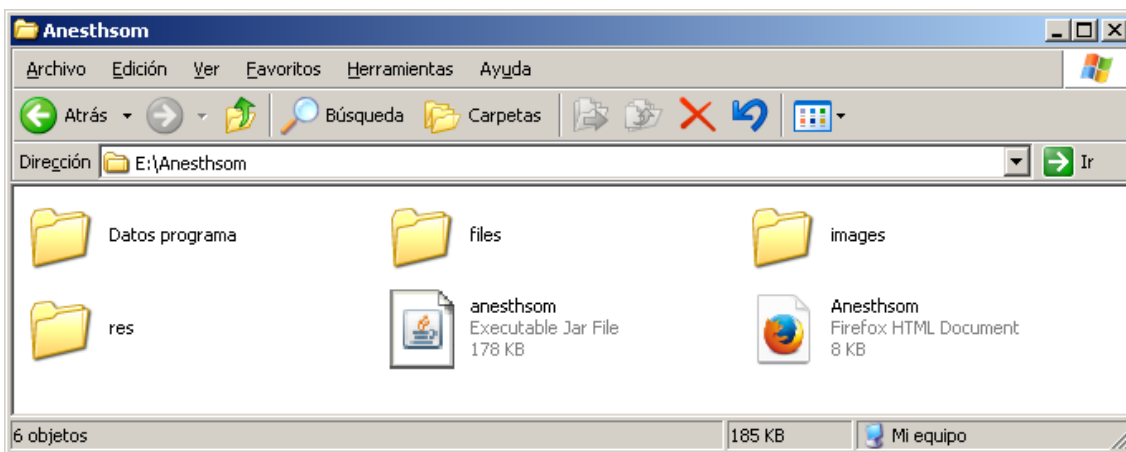


Figura 2.8 Carpeta de BIOIN ANESTHSOM. Fuente: Bona Gracia et al. (2017).

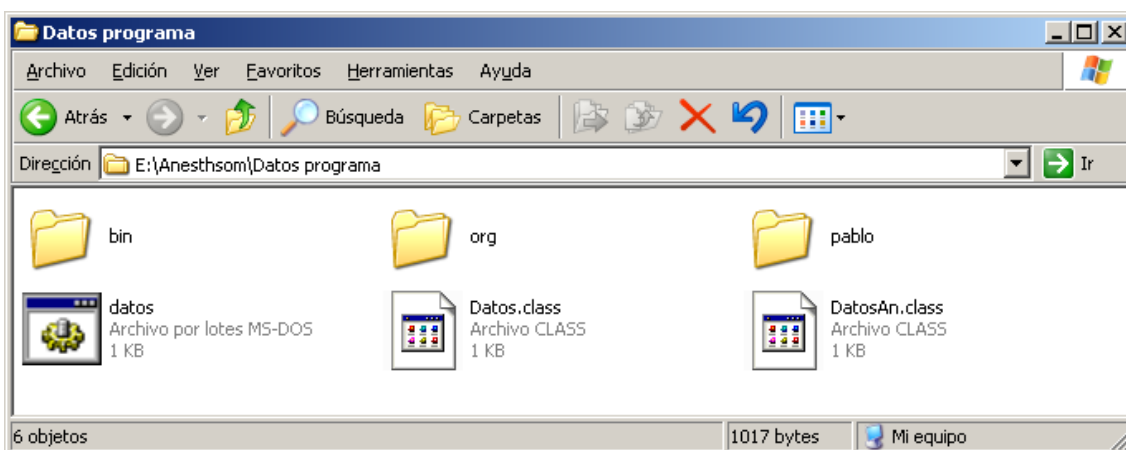


Figura 2.9 Carpeta de datos del programa. Fuente: Bona Gracia et al. (2017).

En la figura 2.10 se muestra la hoja que representa a la matriz de datos.

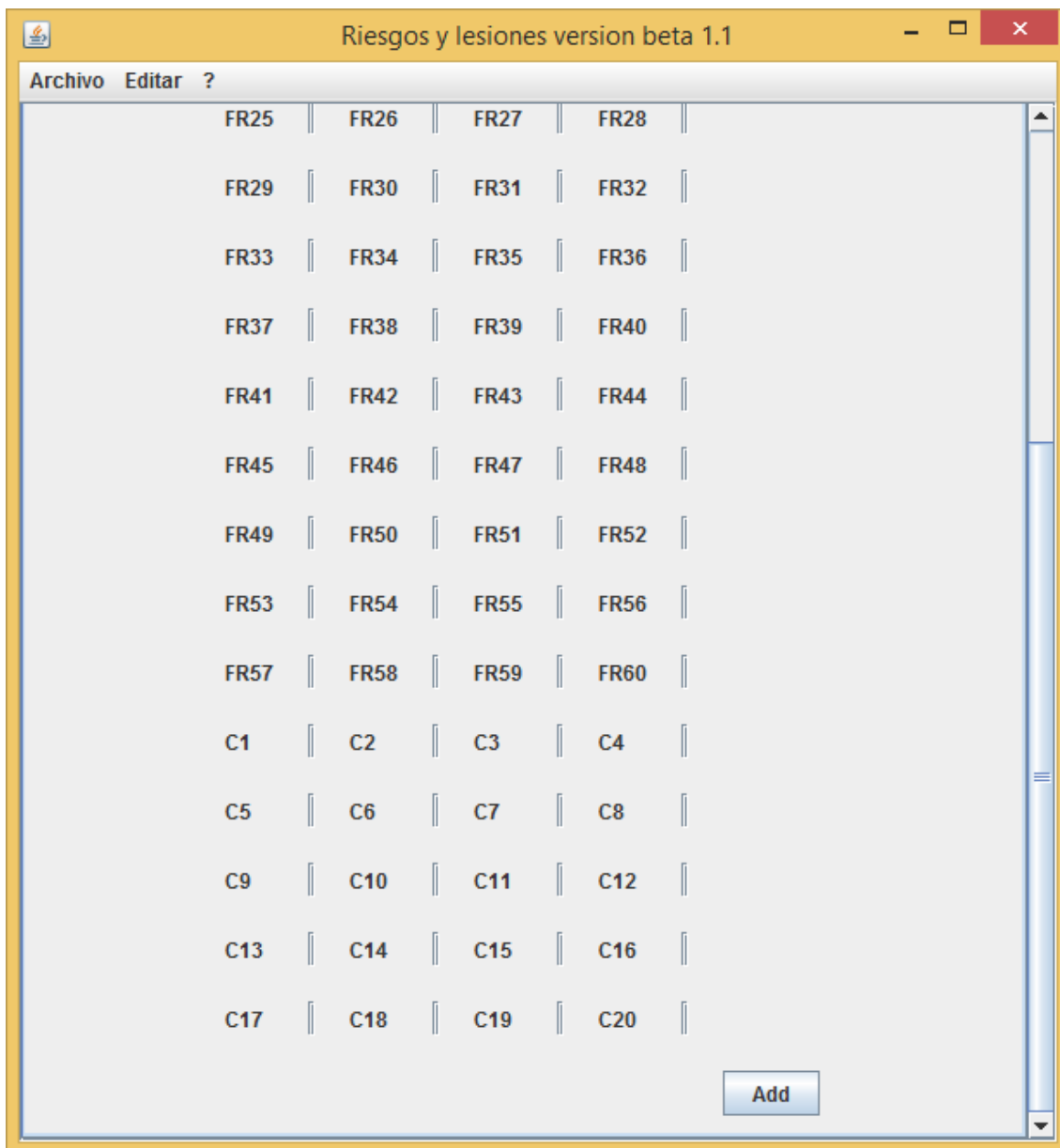


Figura 2.10 Ficha de Factores de Riesgo y Complicaciones, para introducir los datos que se quieran analizar en BIOIN ANESTHSOM. Fuente: Bona Gracia et al. (2017).

En este punto se debió tomar una decisión respecto al número de ítems de entrada de Factores de Riesgo y del número de ítems de entrada de Complicaciones. El primer ACSOM que se realizó tenía 18 causas y 18 consecuencias. El segundo 19 y 19. ANESTHOM se creó con 15 entradas para Factores de Riesgo y 19 entradas para Complicaciones.

Con el objetivo final de poder adaptar este programa a cualquier base de datos potente, como podría ser la de SENSAR, se llegó a la conclusión de que 15 entradas en FR era un número muy reducido, mientras que 19 entrada en Complicaciones se adaptaba bastante bien. Así que, finalmente, se optó por crear una hoja de entrada de datos de 60 por 20, de esta forma BIOIN ANESTHSOM puede ser útil, sin cambios, para SENSAR.

No obstante, se debe recordar que cuanto más reducido sea el número de ítems, sin perder información, más útil será.

Ya que BIOIN ANESTHOM permite trabajar con cualquier matriz de datos, siempre y cuando su número de filas o de columnas sea inferior a 60x20. Es decir, una matriz 50x18 BIOIN ANESTHSOM la puede procesar, en cambio, una matriz 56x21 no (Rubio Calvo et al., 2015; Bona Gracia et al., 2018 a)

Al pulsar sobre la pestaña “*AddAnesth*” aparece una pantalla de introducción de datos.

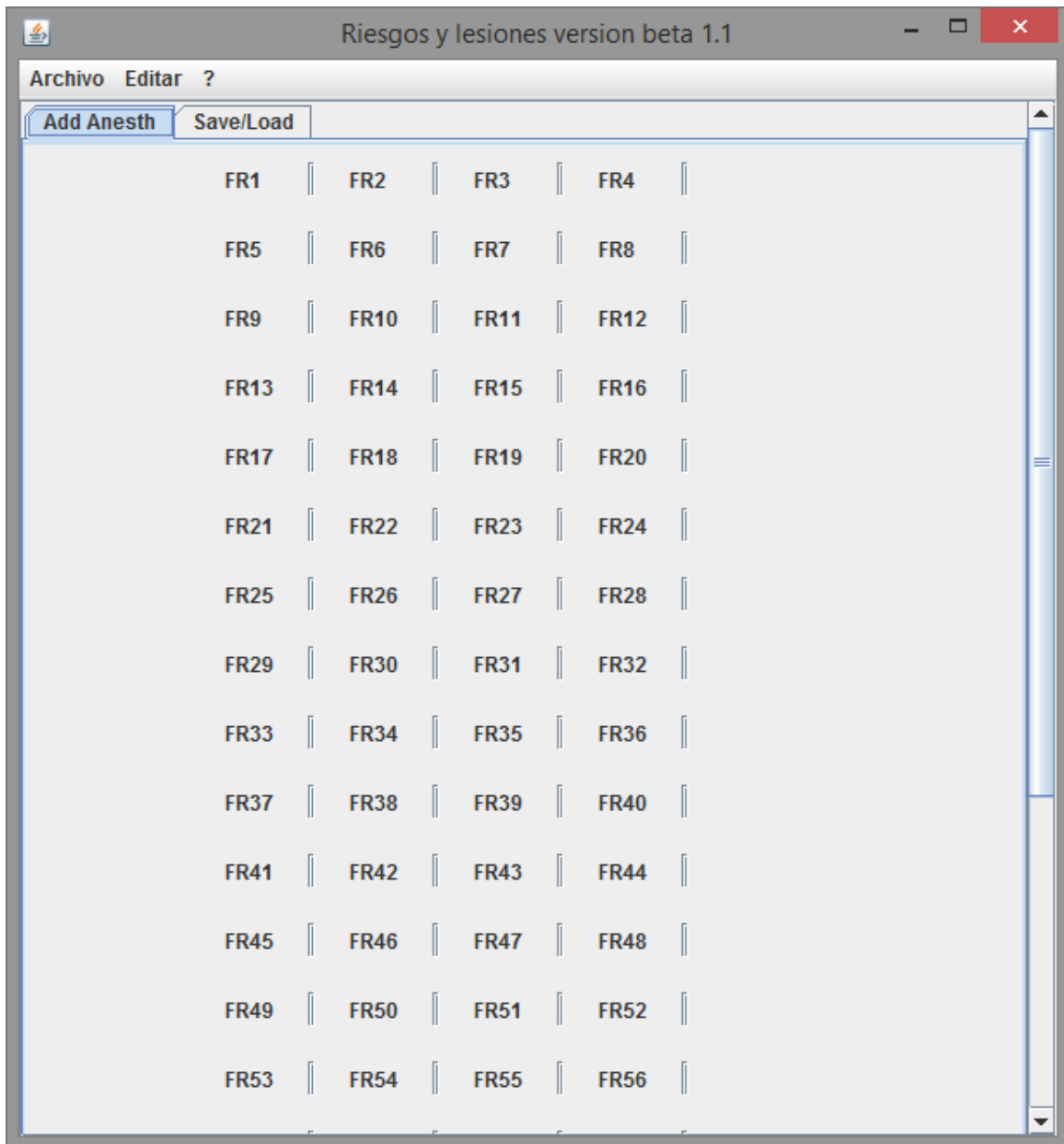


Figura 2.11 Ficha parcial de Factores de Riesgo, para introducir los datos que se quieran analizar. Fuente: Bona Gracia et al. (2017).

Una vez que se han introducido todos los datos, hay que incorporarlos, para ello se debe pulsar en primer lugar sobre “Save/Load”, y después sobre el botón “Save” (figura 2.12).

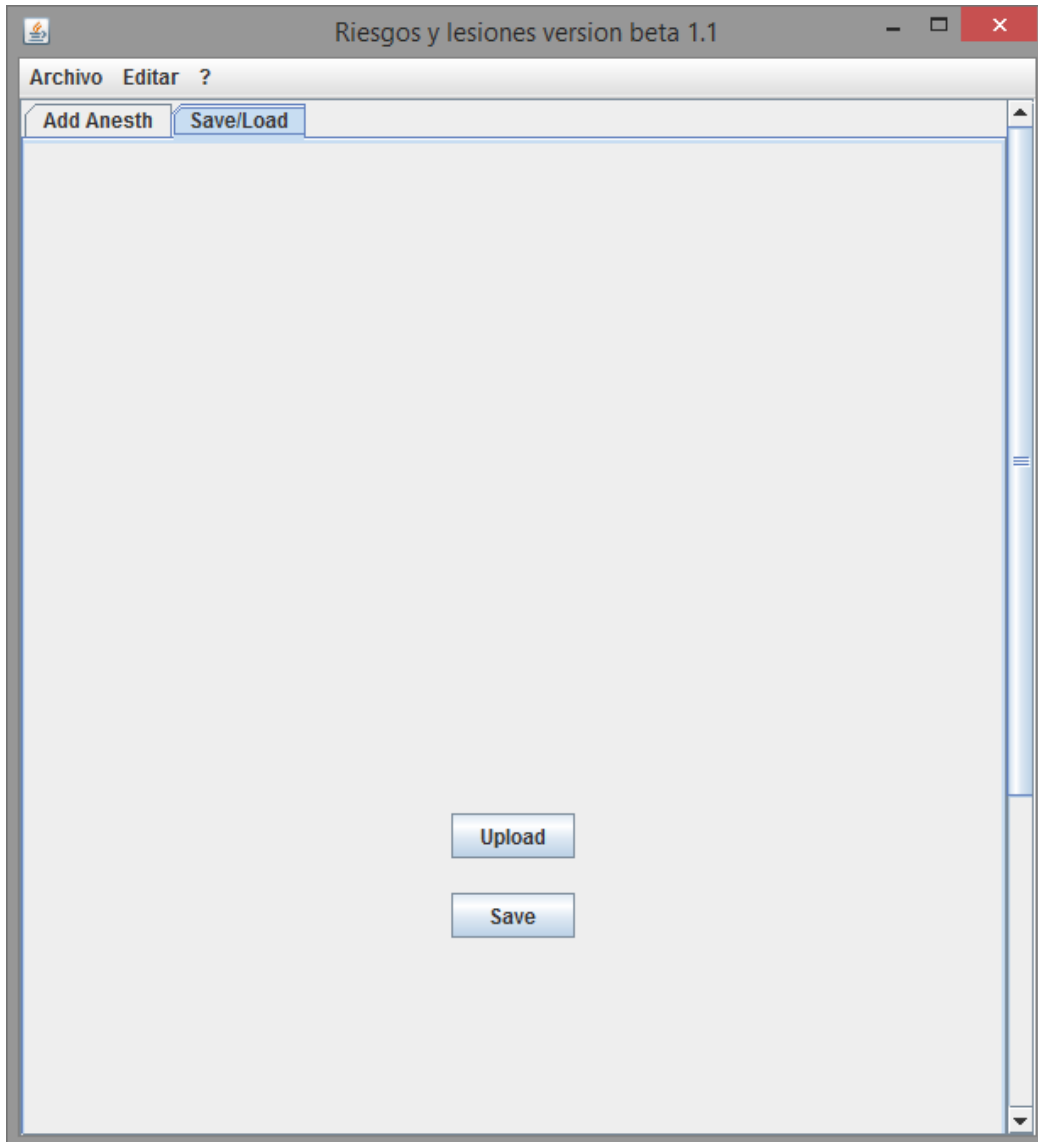


Figura 2.12 Carpeta de BIOIN ANESTHSOM. Fuente: Bona Gracia et al. (2017).

Una vez introducidos y guardados los datos, para activar el programa BIOIN ANESTHSOM, se debe hacer “click” sobre “*Anesthsom.html*” dentro de la carpeta Anesthsom (figura 2.13).

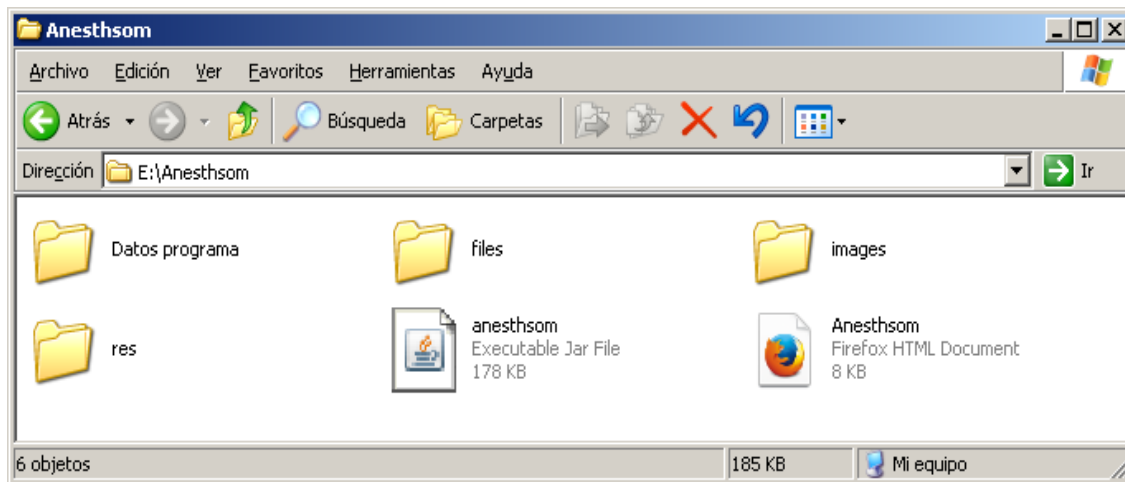


Figura 2.13 Pantalla directorio E:/Anesthsom. Fuente: Bona Gracia et al. (2017).

Y aparece la pantalla inicial de BIOIN ANESTHSOM (figura 2.14) y en la pestaña “Save” es en la que se realiza la selección del patrón, pulsando la tecla “UploadAnesthsom” (en este caso el patrón será la suma de los dos hospitales, Lozano Blesa y Miguel Servet).

Dentro de esta misma pestaña, también se selecciona el ente a ser analizado, pulsando la tecla “Upload hospital”. En este caso puede ser la sección General, Materno-Infantil, Traumatología, Hospital Universitario Miguel Servet u Hospital Clínico Universitario Lozano Blesa.



Figura 2.14 Pantalla inicial de Demo Programa BIOIN ANESTHSOM. Fuente: Bona Gracia et al. (2017).

Seguidamente, pulsando sobre la pestaña “*Operations*” se obtendrán las poligonales, sus diagramas de barras (*Bar chart*) y los “*Planning Reports*” o fases de actuación prioritaria si activamos cada una de las opciones. (figura 2.15)



Figura 2.15 Pantalla de Operaciones de Demo Programa BIOIN ANESTHSOM. Fuente: Bona Gracia et al. (2017).

Si se activa la tecla “*Polygonal graphics*” aparecen dos tablas poligonales, una para las Complicaciones y otra para los Factores de Riesgo.

Si se activa la tecla “*Bar chart*” salen dos tablas de diagrama de barras (translocaciones representadas en amarillo y desviaciones en rojo) una para las Complicaciones y otra para los Factores de Riesgo.

Por último, si se activa la tecla “*Planning Reports*” se crea la tabla de las fases de actuación prioritaria para Complicaciones y Factores de Riesgo.

CAPÍTULO III

RESULTADOS

Y

DISCUSIÓN

1. ANÁLISIS DESCRIPTIVO.

Desde el punto de vista descriptivo en las tablas 3.1 y 3.2 pueden observarse los Factores de Riesgo y las Complicaciones más frecuentes de cada una de las secciones y de los dos hospitales estudiados.

MATERIAL. DATOS DE LOS FACTORES DE RIESGO						
General (236)	Trauma. (190)	Mat-Infantil (162)	HUMS (588)	HCULB (225)	TOTAL Patrón (813)	
FR1	13	11	13	37	18	55
FR2	19	8	25	52	26	78
FR3	14	2	9	25	25	50
FR4	19	12	4	35	28	63
FR5	9	6	4	19	11	30
FR6	9	12	10	31	9	40
FR7	10	12	10	32	9	41
FR8	14	22	9	45	10	55
FR9	44	27	4	75	11	86
FR10	34	12	33	79	12	91
FR11	4	32	33	69	4	73
FR12	18	32	4	54	3	57
FR13	12	2	4	18	16	34
FR14	12	0	0	12	28	40
FR15	5	0	0	5	15	20
FR del Hospital Universitario Miguel Servet (588) y Hospital Clínico Universitario Lozano Blesa (225). Total Patrón (813)						

Tabla 3.1: Datos de los Factores de Riesgo de los hospitales analizados y sus secciones.

MATERIAL DATOS DE LAS COMPLICACIONES						
	General (236)	Trauma. (190)	Mat-infantil (162)	HUMS (588)	HCULB (225)	TOTAL Patrón (813)
C1	19	9	24	52	4	56
C2	9	4	12	25	22	47
C3	17	0	3	20	18	38
C4	9	0	4	13	6	19
C5	7	6	3	16	3	19
C6	19	17	35	71	23	94
C7	10	11	7	28	21	49
C8	14	0	4	18	15	33
C9	20	0	10	30	9	39
C10	14	0	0	14	26	40
C11	13	2	5	20	10	30
C12	1	8	32	41	7	48
C13	4	26	6	36	8	44
C14	5	38	0	43	3	46
C15	17	9	0	26	14	40
C16	23	5	3	31	29	60
C17	6	38	4	48	2	50
C18	2	3	6	11	0	11
C19	27	14	4	45	5	50
. Complicaciones del Hospital Universitario Miguel Servet, 588, Hospital Clínico Universitario Lozano Blesa 225. Total Patrón (813)						

Tabla 3.2: Datos de las Complicaciones de los hospitales analizados y sus secciones.

En el bloque General (tabla 3.1) se observa que los tres Factores de Riesgo más frecuentes son FR9 (Técnica incorrecta) y FR10 (Situación urgente) y en tercer lugar a bastante distancia FR2 (Alteraciones cardio-circulatorias) y FR4 (Alteraciones metabólicas) empatadas. En la tabla 3.2 de Consecuencias, el orden es, C19 (Hematoma), C16 (Hiperalgnesia postoperatoria) y C9 (Aspiración/Neumonía).

En la sección de Traumatología (tabla 3.1), en primer lugar, como Factores de Riesgo más frecuentes están empatados FR11 (Punción dural en epidural) y FR12 (Punción arterial no deseada), y en tercer lugar FR9 (Técnica incorrecta). Si nos referimos a las Consecuencias (tabla 3.2), el orden es, C14 (Coma) y C17 (Neumotórax) con la misma frecuencia de aparición, y C13 (Lesión Nerviosa/Bloqueo Nervioso Prolongado) en tercer lugar.

Los Factores de Riesgo más frecuentes en la sección Materno-Infantil (tabla 3.1), en primer y segundo lugar, empatados, están FR10 (Situación urgente) y FR11 (Punción dural en epidural) y en tercer lugar FR2 (Alteraciones cardio-circulatorias). En el caso de las Consecuencias (tabla 3.2), el orden es C6 (Shock), C12 (Cefalea) y C1 (Rotura de dientes).

Por último, en el conjunto del Hospital Universitario Miguel Servet, se puede observar que los tres Factores de Riesgo (tabla 3.1) que tienen lugar con mayor frecuencia son FR10 (Situación urgente), FR9 (Técnica incorrecta) y FR11 (Punción dural en epidural), que además son con diferencia los más frecuentes. Cuando hablamos de Consecuencias (tabla 3.2) el orden de aparición es C6 (Shock), C1 (Rotura de dientes) y C17 (Neumotórax).

En el Hospital Clínico Universitario Lozano Blesa, como Factores de Riesgo (tabla 3.1) con mayor frecuencia de aparición, están empatados FR4 (Alteraciones metabólicas) y FR14 (Fármacos) y en tercer lugar FR 2 (Alteraciones cardiocirculatorias). Cuando nos referimos a las Consecuencias (tabla 3.2), el orden es en primer lugar C16 (Hiperalgnesia postoperatoria), en segundo lugar, C10 (Complicaciones metabólicas) y tercer puesto C6 (Shock).

Destacando lo distinto de los perfiles entre las tres secciones y los dos hospitales vemos que los primeros Factores de Riesgo de cada unidad son: en

el bloque General FR9 (Técnica incorrecta), en Materno-Infantil FR10 (Situación urgente) y FR11 (Punción dural en epidural), en Traumatología FR11 (Punción dural en epidural), en HUMS FR10 (Situación urgente) y en HCULB FR4 (Alteraciones metabólicas). Podemos observar que, pese a la disparidad existente entre las secciones u hospitales estudiados, el FR 10 (Situación urgente) aparece en dos ocasiones, en la sección Materno-infantil y en el HUMS, y es que según Maaløe (2006), los incidentes son más dados a ser considerados críticos en pacientes clasificados como más graves (ASA III o superior) y en cirugías urgentes.

En el caso de Complicaciones predominan en el bloque General C19 (Hematoma), en Materno-Infantil C6 (Shock), en Traumatología C14 (Coma) y C17 (Neumotórax), en HUMS C6 (Shock) y en HCULB C16 (Hiperalgia postoperatoria). Mediante este análisis descriptivo no vemos coincidencias entre las diferentes secciones u hospitales analizados.

Obsérvense los perfiles tan diferentes que presentan, en Complicaciones, las tres secciones del Hospital Miguel Servet de Zaragoza. Sin embargo, las diferencias frente a ambos patrones son muy pequeñas, salvo en la sección General.

Los perfiles de ambos hospitales Clínico Lozano Blesa y Miguel Servet son también dispares, esto puede ser debido a la distinta estructuración que presentan, ya que el Hospital Universitario Miguel Servet se divide en secciones diferentes entre sí, y el Hospital Clínico se organiza como un hospital general.

Se han obtenido también, los residuales estandarizados de Habertman (RA), (tabla 3.3) de la matriz formada por los 15 Factores de Riesgo y 19 Complicaciones, junto con el total absoluto de cada marginal se presentan tanto las frecuencias de cada par (FR-C) como los residuales ajustados de Habertman. Señalamos el residual en amarillo para poder hacer una primera interpretación de las relaciones existentes entre los diferentes Factores de Riesgo y Complicaciones, aquellos valores de residuales que son mayores que 1,96.

Factores de Riesgo	Complicaciones																			Total	
	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19		
FR1	RA	4,7	,8	1,5	-1,4	-9	-4	-1,3	-1,1	-1,5	-1,0	-1,2	-2	,9	-,1	-1,2	,1	2,7	-,7	-1,8	55
FR2	RA	-2,7	-1,7	7,8	13,2	1,4	2,9	-2,0	-4	-1,7	-1,5	,1	-2,4	-2,1	-,8	-1,8	-2,2	-1,4	-1,0	-2,7	78
FR3	RA	,4	-1,1	-1,1	-1,3	-8	-2,0	5,1	4,0	1,8	3,6	3,6	-1,5	-1,3	-1,4	-1,1	-1,4	-,7	-,8	-1,7	50
FR4	RA	-1,8	1,8	3,1	1,0	2,8	-5	-1,3	-1,2	1,4	,0	1,4	-1,7	-1,5	-,2	-1,3	-1,5	3,9	-,7	-1,8	63
FR5	RA	,9	2,6	-1,0	-1,1	-,7	,4	-1,1	-,9	-1,2	3,0	6,7	-1,3	-1,2	-1,2	-1,0	-1,2	1,3	-,5	-1,4	30
FR6	RA	-,2	2,1	-1,1	-1,2	-,8	,0	-1,2	-1,0	1,1	-,9	-1,1	-1,4	-1,3	5,1	6,8	-1,3	-1,4	-,8	-1,6	40
FR7	RA	-,9	-1,0	-1,1	-1,2	-,8	6,9	-1,2	-1,0	-1,3	2,6	-1,1	-1,4	-1,3	-1,4	-1,1	-1,3	-,6	1,2	3,4	41
FR8	RA	3,2	-1,2	-1,3	-1,4	2,7	-2,2	-1,4	-1,2	-1,6	-1,1	-1,3	,4	,0	,5	3,0	,7	1,8	-,7	,6	55
FR9	RA	-1,3	-1,8	-1,9	-2,1	1,3	,0	2,2	-,4	-2,3	,7	1,3	-2,5	-1,1	2,3	,7	5,4	,1	-1,0	,5	86
FR10	RA	,5	,9	-1,9	-2,1	-1,3	,3	2,2	1,6	6,6	-,1	-1,9	-2,5	-2,2	-2,4	-,6	3,7	-1,4	1,3	-,5	91
FR11	RA	-2,5	-1,6	-1,7	-1,9	-1,2	-3,0	-1,2	-,2	-2,1	-1,4	-1,7	14,7	7,4	,6	-1,7	-2,0	-2,2	1,5	-,5	73
FR12	RA	-2,1	-1,4	-1,5	-1,6	-1,0	-2,5	-1,6	,3	-1,8	-1,2	-1,5	-,1	4,3	1,3	-1,5	-1,8	1,7	,6	9,5	57
FR13	RA	6,0	5,1	-1,1	-1,3	-,8	-,7	3,3	2,0	,2	-1,0	-1,1	-1,5	-1,3	-1,4	-1,1	1,4	-1,5	-,8	-1,7	34
FR14	RA	-1,2	-,8	,5	-,9	-,6	2,8	-,9	-,8	1,2	,9	,5	-1,1	-,9	-1,0	3,1	1,3	-1,1	-,4	-1,2	40
FR15	RA	-,7	-,5	-,5	-,6	-,4	-,9	-,5	-,5	2,8	-,4	-,5	-,7	-,6	-,6	3,6	1,2	-,7	3,5	-,7	20
Total		48	21	24	29	12	65	27	21	35	17	24	40	32	37	24	33	39	7	48	813

Tabla 3.3: Residuales estandarizados de Habertman (RA). Estudio analítico de las relaciones entre FR y C.

Al analizar la tabla 3.3, encontramos las siguientes relaciones entre los Factores de Riesgo y las Complicaciones registradas en el Patrón.

Nota 1. Resaltamos que tienen que ser residuales superior a 1,96. En el caso que haya varios, solo indicaremos los tres primeros.

Nota 2. Si hay varios con residual superior a 1,96 pero si hay diferencias muy acusadas entre los valores de los residuales, para resaltar la relación, solo marcaremos los superiores.

Residuales estandarizados de Habertman (RA).

FR1 (Alteraciones antropométricas) con C1 (Rotura de diente/s) y C17 (Neumotórax)

FR2 (Alteraciones cardio-circulatorias) con C3 (Arritmias), C4 (Infarto agudo de miocardio) y C6 (Shock).

FR3 (Alteraciones respiratorias) con C7, (Laringo/Broncoespasmo) y C8, (Atelectasia).

FR4 (Alteraciones metabólicas) con C17 (Neumotórax), C3 (Arritmias) y C5 (Tromboembolismo pulmonar)

FR5 (Alteraciones renales) con C11 (Complicaciones Renales).

FR6 (Alteraciones neurológicas) con C15 (Temblores/Agitación), C14 (Coma) y C2 (Edema de glotis).

FR7 (Alteraciones hematológicas) con C6 (Shock) y C9 (Aspiración/Neumonía) y C10 (Metabólicas).

FR8 (Inexperiencia) con C1 (Rotura de diente/s), C15 (Temblores/Agitación) y C5 (Tromboembolismo pulmonar)

FR9 (Técnica incorrecta) con C16 (Hiperalgiesia postoperatoria).

FR10 (Situación urgente) con C9 (Aspiración/Neumonía)

FR11 (Punción dural en epidural) con C12 (Cefalea) y C13 (Lesión Nerviosa/Bloqueo Nervioso Prolongado).

FR12 (Punción arterial no deseada) con C19 (Hematoma) y con C13 (Lesión Nerviosa/Bloqueo Nervioso Prolongado)

FR13 (Problemas previos en la intubación), con C1 (Rotura de diente/s) y C2 (Edema de glotis).

FR14 (Fármacos) con C15 (Temblores/Agitación) y C6 (Shock).

FR15 (Otras no clasificadas anteriormente) con C15 (Temblores/Agitación), C18 (Infección) y C9 (Aspiración/Neumonía).

C1 (Rotura de diente/s) con FR13 (Problemas previos en la intubación), FR1 (Alteraciones antropométricas) y FR8 (Inexperiencia).

C2 (Edema de glotis) con FR13 (Problemas previos en la intubación), FR5 (Alteraciones renales) y FR6 (Alteraciones neurológicas).

C3 (Arritmias) con FR2 (Alteraciones cardio-circulatorias) y FR4 (Alteraciones metabólicas).

C4 (Infarto agudo de miocardio (IAM)/Ángor) con FR2 (Alteraciones cardio circulatorias).

C5 (Tromboembolismo pulmonar) con FR4 (Alteraciones metabólicas) y FR8 (Inexperiencia).

C6 (Shock) con FR7 (Alteraciones hematológicas) y FR14 (Fármacos).

C7 (Laringo/Broncoespasmo) con FR3 (Alteraciones respiratorias) y FR13 (Problemas previos en la intubación).

C8 (Atelectasia) con FR3 (Alteraciones respiratorias) y FR13 (Problemas previos en la intubación).

C9 (Aspiración/Neumonía) con FR10 (Situación urgente) y FR 15 (Otras no clasificadas anteriormente).

C10 (Metabólicas) con FR3 (Alteraciones respiratorias), FR5 (Alteraciones renales) y FR7 (Alteraciones hematológicas).

C11 (Complicaciones Renales) con FR5 (Alteraciones renales) y FR 3 (Alteraciones respiratorias).

C12 (Cefalea) con FR11 (Punción dural en epidural).

C13 (Lesión Nerviosa/Bloqueo Nervioso Prolongado), con FR11 (Punción dural en epidural) y FR 12 (Punción arterial no deseada).

C14 (Coma) con FR6 (Alteraciones neurológicas).

C15 (Temblores/Agitación) con FR6 (Alteraciones neurológicas).

C16 (Hiperalgnesia postoperatoria) con FR9 (Técnica incorrecta).

C17 (Neumotórax) con FR4 (Alteraciones metabólicas) y FR1 (Alteraciones antropométricas).

C18 (Infección) con FR15 (Otras no clasificadas anteriormente).

C19 (Hematoma) con FR12 (Punción arterial no deseada) y con FR7 (Alteraciones hematológicas).

Resaltamos los casos en que la relación era máxima en ambas direcciones, es decir, cuando existe coincidencia.

FR2 (Alteraciones cardio-circulatorias) con C4 (Infarto agudo de miocardio).

FR3 (Alteraciones respiratorias) con C7 (Laringo/Broncoespasmo).

FR4 (Alteraciones metabólicas) con C17 (Neumotórax).

FR5 (Alteraciones renales) con C11 (Renales).

FR6 (Alteraciones neurológicas) con C15 (Temblores/Agitación).

FR7 (Alteraciones hematológicas) con C6 (Shock).

FR9 (Técnica incorrecta) con C16 (Hiperalgnesia postoperatoria).

FR10 (Situación urgente) con C9 (Aspiración/Neumonía).

FR11 (Punción dural en epidural) con C12 (Cefalea).

FR12 (Punción arterial no deseada) con C19 (Hematoma).

FR13 (Problemas previos en la intubación) con C1 (Rotura de diente/s).

Análisis de las Distancias de Minkowski.

A continuación, en la tabla 3.4 presentamos los resultados correspondientes a las distancias Minkowski potencia 2, 3, 4 según utilizemos para las mismas las puntuaciones factoriales con dos, tres o cuatro dimensiones.

	P	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19
FR1	3	0,745	0,850	2,113	2,821	0,832	0,722	1,001	0,425	0,907	0,779	0,731	2,357	1,747	0,298	0,574	0,723	0,170	0,722	0,567
	4	0,722	0,841	1,916	2,579	0,886	0,949	1,119	0,831	0,843	1,385	1,252	2,241	1,573	1,062	2,191	0,723	0,773	0,890	1,989
	3	0,768	0,776	1,839	2,490	0,809	0,883	1,038	0,781	0,794	1,200	1,200	2,188	1,646	1,018	2,087	0,894	0,732	0,840	1,892
	4	2,635	2,600	0,165	0,652	1,335	1,626	2,885	2,402	2,533	2,590	2,148	3,438	2,946	2,113	3,000	2,612	1,932	2,473	2,658
FR2	4	2,584	2,570	0,150	0,551	1,285	1,588	2,837	2,321	2,616	2,510	2,072	3,405	2,893	1,982	2,785	2,567	1,851	2,328	2,397
	3	0,283	0,164	2,710	3,420	1,473	1,189	0,195	0,573	0,124	0,283	0,744	3,181	2,955	1,229	0,344	0,288	0,985	1,401	1,433
	4	1,178	1,343	2,749	3,371	2,063	1,856	0,938	1,507	0,995	0,796	0,796	2,985	2,442	2,348	3,458	1,900	1,656	1,953	2,741
	3	1,327	1,233	1,319	2,034	0,191	0,231	1,434	1,119	1,135	0,672	2,926	2,369	2,265	1,073	1,280	1,297	0,725	1,571	1,482
FR3	3	1,291	1,309	1,265	1,935	0,509	0,582	1,575	1,151	1,299	1,410	1,109	2,929	2,387	2,385	1,280	0,740	1,500	1,981	1,482
	4	1,248	1,253	1,249	1,909	0,478	0,572	1,511	1,047	1,273	1,329	1,036	2,903	2,357	1,158	2,255	1,226	0,670	1,452	1,824
	3	0,493	0,306	2,244	2,903	1,029	0,712	0,628	0,513	0,343	0,215	0,276	3,091	2,477	1,040	0,493	0,467	0,577	1,368	1,356
	4	1,188	1,353	2,229	2,857	1,508	1,322	0,696	0,931	1,544	0,493	0,263	2,992	2,373	2,066	3,308	1,591	1,140	1,743	2,284
FR4	3	0,298	0,398	2,456	3,173	1,168	0,914	0,461	0,138	0,380	0,304	0,609	2,776	2,158	0,782	0,145	0,183	0,581	1,008	1,022
	4	1,923	1,882	2,653	3,215	1,812	2,203	2,685	2,793	1,989	3,151	3,063	2,924	2,821	1,267	0,342	1,620	2,356	1,865	3,048
	3	0,773	0,782	1,929	2,644	0,632	0,488	1,018	0,437	0,848	0,793	0,509	2,571	1,989	0,528	0,710	0,748	0,102	0,982	0,921
	4	2,014	2,225	2,184	2,690	1,482	1,002	2,150	1,983	1,846	1,386	1,829	3,020	2,084	1,844	3,199	1,503	0,934	1,437	0,934
FR5	3	1,907	2,109	1,981	2,500	1,411	0,935	2,102	1,948	1,811	1,352	1,821	2,783	1,972	1,824	3,190	1,434	0,887	1,388	0,925
	4	0,709	0,905	2,428	3,131	1,151	1,049	0,944	0,428	0,956	0,895	0,981	2,208	1,591	0,325	0,642	0,698	0,501	0,458	0,458
	3	1,177	1,399	2,212	2,858	1,035	0,971	1,474	1,251	1,021	1,331	2,243	2,243	1,472	0,347	0,710	0,622	0,437	0,456	1,043
	4	1,143	1,341	2,093	2,732	0,980	1,063	1,545	1,457	0,941	1,822	1,764	2,086	1,453	0,338	1,523	0,596	1,023	0,520	1,730
FR6	3	0,349	0,468	2,359	3,076	1,054	0,836	0,504	0,524	0,416	0,500	2,657	2,040	2,040	0,651	0,278	0,325	0,446	0,907	0,907
	4	1,170	1,369	2,733	2,984	1,096	0,785	1,455	1,287	0,847	0,730	1,214	2,618	1,894	0,585	0,534	0,745	0,453	0,876	1,151
	3	1,183	1,351	2,163	2,820	0,995	0,718	1,435	1,304	0,868	1,131	1,283	2,481	1,830	0,587	2,223	0,469	0,469	0,871	1,221
	4	0,116	0,158	2,631	3,345	1,388	1,081	0,222	0,399	0,181	0,215	0,692	3,007	2,390	1,056	0,170	0,114	0,834	1,228	1,298
FR7	3	0,528	0,734	2,526	3,197	1,366	1,097	0,813	0,714	0,222	0,195	0,750	2,757	2,784	1,095	1,122	0,586	0,861	1,151	1,798
	4	0,547	0,735	2,503	3,163	1,353	1,060	0,893	0,812	0,436	1,296	1,134	2,625	2,073	1,058	2,098	0,591	0,854	1,095	1,748
	3	2,744	2,969	3,337	3,835	2,513	2,752	2,847	2,483	3,009	2,919	2,928	0,154	0,483	1,907	2,682	2,738	2,321	1,626	1,598
	4	2,483	2,704	3,239	3,605	2,687	2,788	2,646	2,277	2,750	2,693	2,794	0,233	0,313	2,062	2,717	2,583	2,311	1,688	2,437
FR8	3	2,377	2,609	3,216	3,535	2,623	2,728	2,523	2,214	2,646	2,660	2,793	0,230	0,308	1,941	2,690	2,424	2,238	1,535	2,385
	4	1,528	1,736	2,630	3,295	1,584	1,634	1,743	1,269	1,786	1,695	1,734	1,377	0,760	0,712	1,484	1,520	1,138	0,428	0,375
	3	2,182	2,400	2,681	3,051	1,696	1,704	2,451	2,207	2,028	1,918	2,322	2,412	1,367	0,855	1,321	1,461	1,323	0,955	0,353
	4	2,096	2,329	2,422	2,845	1,742	1,680	2,322	2,138	2,037	1,791	2,195	2,400	1,362	1,756	3,127	1,581	1,267	1,277	0,331
FR9	3	0,637	0,469	2,778	3,430	1,989	1,376	0,362	0,667	0,988	1,110	0,916	2,897	2,430	2,061	2,292	1,755	1,752	1,863	2,844
	4	0,626	0,476	2,729	3,381	1,887	1,848	0,349	0,724	1,002	1,369	1,128	2,793	2,255	1,994	2,633	1,754	1,687	1,753	2,794
	3	1,269	1,434	2,066	2,716	0,887	0,949	1,551	1,387	0,921	0,792	1,244	3,015	2,323	0,926	0,614	0,326	0,541	1,342	1,434
	4	1,295	1,424	2,026	2,676	0,872	0,940	1,653	1,581	0,892	1,892	1,823	2,886	2,299	0,911	1,438	0,532	1,106	1,346	1,898
FR10	3	0,185	0,353	2,810	3,525	1,532	1,260	0,141	0,486	0,357	0,419	0,890	2,981	3,349	1,097	0,225	0,194	0,948	1,193	1,234
	4	1,373	1,562	2,717	3,376	1,500	1,195	1,634	1,523	1,021	0,917	1,467	2,874	2,130	1,029	0,334	0,274	0,966	1,069	1,209
	3	1,978	1,931	2,787	3,386	1,912	2,263	2,632	2,674	1,574	3,193	3,059	2,741	2,516	1,364	0,319	1,678	2,416	1,855	3,074
	4	0,299	0,274	2,620	3,531	1,574	1,275	0,103	0,616	0,240	0,370	0,856	3,182	2,568	1,271	0,370	0,315	1,068	1,402	1,445
FR11	3	0,626	0,476	2,729	3,430	1,989	1,376	0,362	0,667	0,988	1,110	0,916	2,897	2,430	2,061	2,292	1,755	1,752	1,863	2,844
	4	0,626	0,476	2,729	3,430	1,989	1,376	0,362	0,667	0,988	1,110	0,916	2,897	2,430	2,061	2,292	1,755	1,752	1,863	2,844
	3	1,269	1,434	2,066	2,716	0,887	0,949	1,551	1,387	0,921	0,792	1,244	3,015	2,323	0,926	0,614	0,326	0,541	1,342	1,434
	4	1,295	1,424	2,026	2,676	0,872	0,940	1,653	1,581	0,892	1,892	1,823	2,886	2,299	0,911	1,438	0,532	1,106	1,346	1,898
FR12	3	0,185	0,353	2,810	3,525	1,532	1,260	0,141	0,486	0,357	0,419	0,890	2,981	3,349	1,097	0,225	0,194	0,948	1,193	1,234
	4	1,373	1,562	2,717	3,376	1,500	1,195	1,634	1,523	1,021	0,917	1,467	2,874	2,130	1,029	0,334	0,274	0,966	1,069	1,209
	3	1,978	1,931	2,787	3,386	1,912	2,263	2,632	2,674	1,574	3,193	3,059	2,741	2,516	1,364	0,319	1,678	2,416	1,855	3,074
	4	0,299	0,274	2,620	3,531	1,574	1,275	0,103	0,616	0,240	0,370	0,856	3,182	2,568	1,271	0,370	0,315	1,068	1,402	1,445

Tabla 3.4: Distancia de Minkowski. Potencias 2,3 y 4. Con potencia 3 distancias más cortas de las relaciones entre FR y C.

Si analizamos Minkowski con potencia 3 se señala de la tabla las distancias más cortas que corresponderán a las relaciones entre Factores de Riesgo y Complicaciones. Se consideraron las distancias inferiores a 1, y se indicaron un máximo de tres (tabla 3.4).

FR1 (Alteraciones antropométricas) con C1 (Rotura de diente/s) y C17 (Neumotórax).

FR2 (Alteraciones cardio-circulatorias) con C3 (Arritmias) y C4 (Infarto agudo de miocardio (IAM)/Ángor).

FR3 (Alteraciones respiratorias) relacionado con C7 (Laringo/Broncoespasmo) y C8 (Atelectasia).

FR4 (Alteraciones metabólicas) con C5 (Tromboembolismo pulmonar), C6 (Shock) y C17 (Neumotórax).

FR5 (Alteraciones renales) con C10 (Metabólicas) y C11 (Renales).

FR6 (Alteraciones neurológicas) con C15 (Temblores/Agitación).

FR7 (Alteraciones hematológicas) con C17 (Neumotórax) y C19 (Hematoma).

FR8 (Inexperiencia) con C17 (Neumotórax) y C18 (Infección).

FR9 (Técnica incorrecta) con C16 (Hiperalgnesia postoperatoria), C17 (Neumotórax) y C6 (Shock).

FR10 (Situación urgente) con C9 (Aspiración/Neumonía) y C10 (Metabólicas).

FR11 (Punción dural en epidural) con C12 (Cefalea) y C13 (Lesión Nerviosa/Bloqueo Nervioso Prolongado).

FR12 (Punción arterial no deseada) con C19 (Hematoma) y C14 (Coma).

FR13 (Problemas previos en la intubación) con C2 (Edema de glotis), C7 (Laringo/ Broncoespasmo) y C1 (Rotura de diente/s).

FR14 (Fármacos) con C6 (Shock), C16 (Hiperalgnesia postoperatoria) y C17 (Neumotórax).

FR15 (Otras no clasificadas anteriormente) con C15 (Temblores/Agitación) y C16 (Hiperalgnesia postoperatoria).

C1 (Rotura de diente/s) con FR10 (Situación urgente) y FR1 (Alteraciones antropométricas).

C2 (Edema de glotis) con FR13 (Problemas previos en la intubación) y FR10 (Situación urgente).

C3 (Arritmias) con FR2 (Alteraciones cardio-circulatorias).

C4 (Infarto agudo de miocardio (IAM)/Ángor) con FR2 (Alteraciones cardio-circulatorias).

C5 (Tromboembolismo pulmonar) con FR4 (Alteraciones metabólicas) y FR14 (Fármacos).

C6 (Shock) con FR4 (Alteraciones metabólicas) y FR14 (Fármacos).

C7 (Laringo/Broncoespasmo) con FR13 (Problemas previos en la intubación) y FR3 (Alteraciones respiratorias).

C8 (Atelectasia) con FR5 (Alteraciones renales) y FR3 (Alteraciones respiratorias).

C9 (Aspiración/Neumonía) con FR10 (Situación urgente) y FR1 (Alteraciones antropométricas).

C10 (Metabólicas) con FR10 (Situación urgente) y FR5 (Alteraciones renales).

C11 (Renales) con FR5 (Alteraciones renales) y FR10 (Situación urgente).

C12 (Cefalea) con FR11 (Punción dural en epidural).

C13 (Lesión Nerviosa/Bloqueo Nervioso Prolongado) con FR11 (Punción dural en epidural).

C14 (Coma) con FR8 (Inexperiencia) y FR9 (Técnica incorrecta).

C15 (Temblores/Agitación) con FR6 (Alteraciones neurológicas), FR15 (Otras no clasificadas anteriormente).

C16 (Hiperalgia postoperatoria) con FR9 (Técnica incorrecta) y FR15 (Otras no clasificadas anteriormente).

C17 (Neumotórax) con FR8 (Inexperiencia), FR9 (Técnica incorrecta) y FR14 (Fármacos).

C18 (Infección) con FR8 (Inexperiencia) y FR9 (Técnica incorrecta).

C19 (Hematoma) con FR12 (Punción arterial no deseada) y FR7 (Alteraciones hematológicas).

FR2 (Alteraciones cardio-circulatorias) con C3 (Arritmias).

FR4 (Alteraciones metabólicas) con C5 (Tromboembolismo pulmonar).

FR5 (Alteraciones renales) con C11 (Renales).

FR8 (Inexperiencia) con C14 (Coma).

FR10 (Situación urgente) con C10 (Metabólicas).

FR11 (Punción dural en epidural) con C12 (Cefalea).

FR12 (Punción arterial no deseada) con C19 (Hematoma).

FR13 (Problemas previos en intubación) con C7 (Laringo/Broncoespasmo).

FR15 (Otras no clasificadas anteriormente) con C16 (Hiperalgia postoperatoria).

Análisis comparativo de Residuales de Habertman y Distancias de Minkowski

Los resultados son distintos, lo que puede comprobarse al comparar los análisis conjuntos, para el análisis descriptivo se consideran complementarios ya que su utilidad es ver la relación existente entre Factores de Riesgo y

Complicaciones. En esta monografía no son fundamentales, pero hemos creído oportuno actualizar la clasificación de Rubio García et al. (2014).

En el análisis de residuales de Habertman FR1 (Alteraciones antropométricas), FR8 (Inexperiencia), FR14 (Fármacos) y FR15 (Otras no clasificadas anteriormente) no tienen respuesta única.

En el análisis de distancias de Minkowski los que no tienen respuesta única son FR1 (Alteraciones antropométricas), FR3 (Alteraciones respiratorias), FR6 (Alteraciones neurológicas), FR7 (Alteraciones hematológicas), FR9 (Técnica incorrecta), FR12 (Punción arterial no deseada) y FR14 (Fármacos).

Las únicas parejas comunes en ambos métodos son FR5 (Alteraciones renales) con C11 (Renales) y FR11 (Punción dural en epidural) con C12 (Cefalea).

2. RESULTADOS DE BIOIN ANESTHSOM.

Se va a realizar, a partir de las gráficas obtenidas con BIOIN ANESTHSOM, un análisis del contraste de cada hospital o sección estudiado frente al patrón, que es la suma de los dos hospitales. Se han obtenido también las gráficas de cada una de las secciones General, Traumatología y Materno-infantil comparadas con el Hospital Miguel Servet, que se analizarán únicamente si existe algún contraste muy significativo. Se compararán, además, las fases de contraste o *Planning Report*.

2.1 BLOQUE GENERAL

RESULTADOS DEL BLOQUE GENERAL frente al PATRÓN

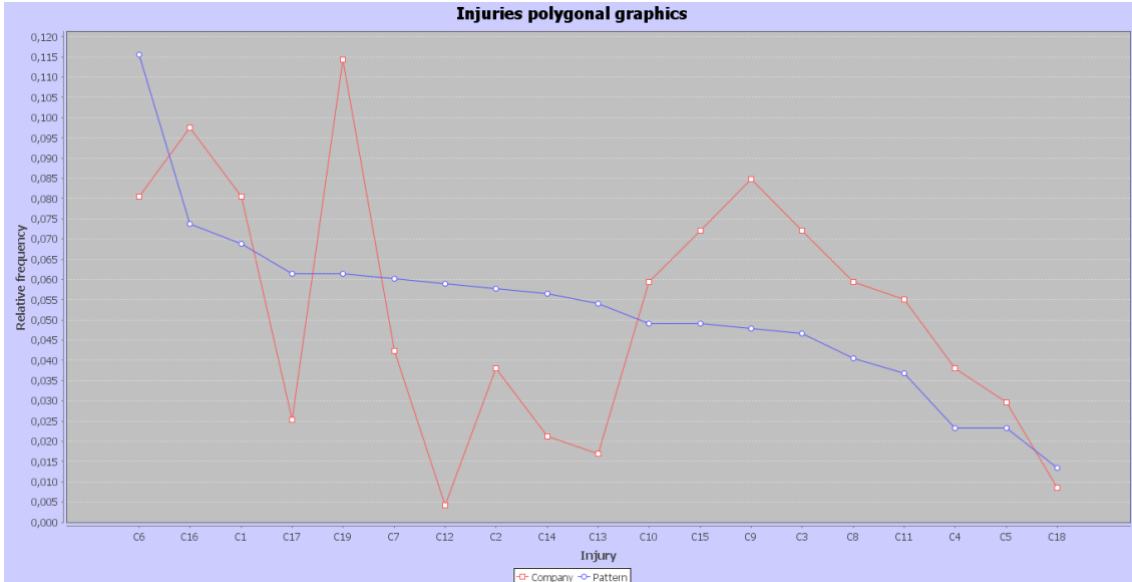


Figura 3.1 Poligonales de Complicaciones del bloque General vs Patrón

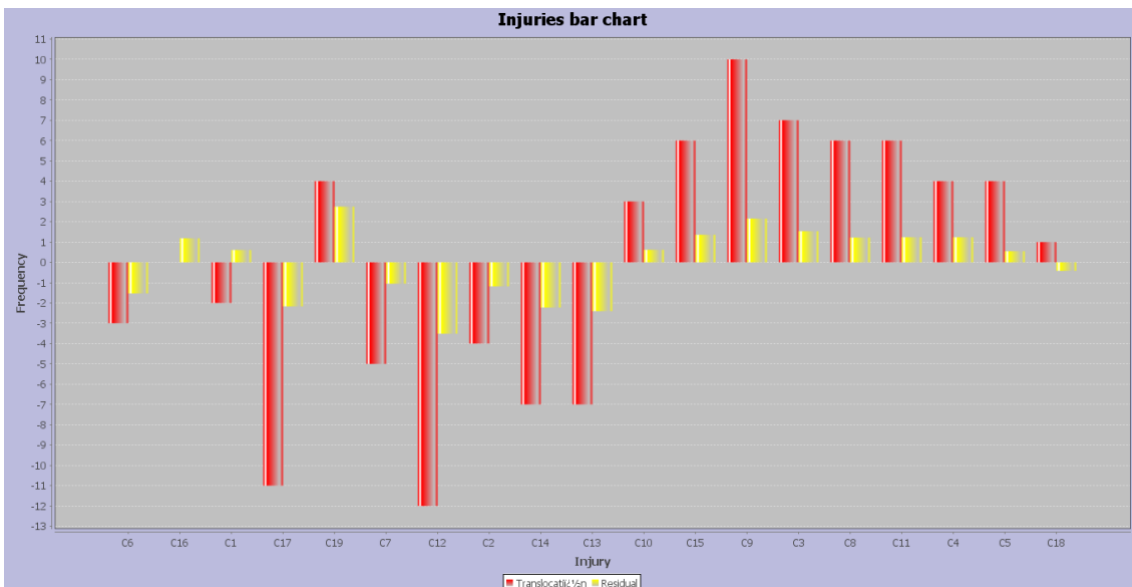


Figura 3.2 Diagramas de barras de Complicaciones del bloque General vs Patrón

En “Complicaciones”, podemos observar que las poligonales son muy distintas (patrón y general). En negativo y con diferencias importantes con respecto al patrón, destacan C19 (Hematoma) y C9 (Neumonía) y en tercer lugar C3 (Arritmias). En sentido contrario, las menos frecuentes y con diferencias acusadas con el patrón son C12 (Cefalea), C13 (Lesión nerviosa) y C17 (Neumotórax) (Figura 3.1).

Estas diferencias son las que se representan en el diagrama de barras, el contraste estadístico pone de manifiesto matices que a simple vista no se notan. Los puntos débiles de esta sección General no mantienen el mismo orden que en las poligonales, en las translocaciones (figura 3.2) el primero es C9 (Neumonía), y después C3 (Arritmias), seguido de varias con el mismo valor de contraste. En residuales o contrastes destaca C19 (Hematoma).

Es razonable la aparición de C9 (neumonía) en este bloque puesto que la infección respiratoria es una de las complicaciones más frecuentes en el postoperatorio, sobre todo si se ha sometido al paciente a una anestesia general. También es muy lógico que C12 (Cefalea), C13 (Lesión Nerviosa) y C17 (Neumotórax) sean poco frecuentes ya que son complicaciones directamente relacionadas con la anestesia locorregional.

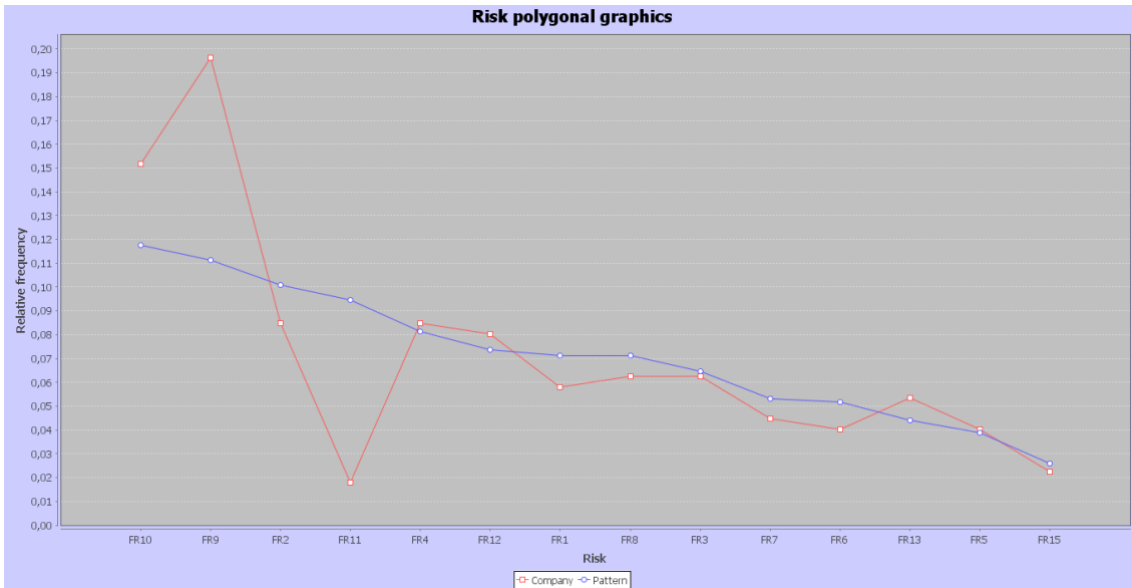


Figura 3.3 Poligonales de Factores de Riesgo del bloque General vs Patrón.

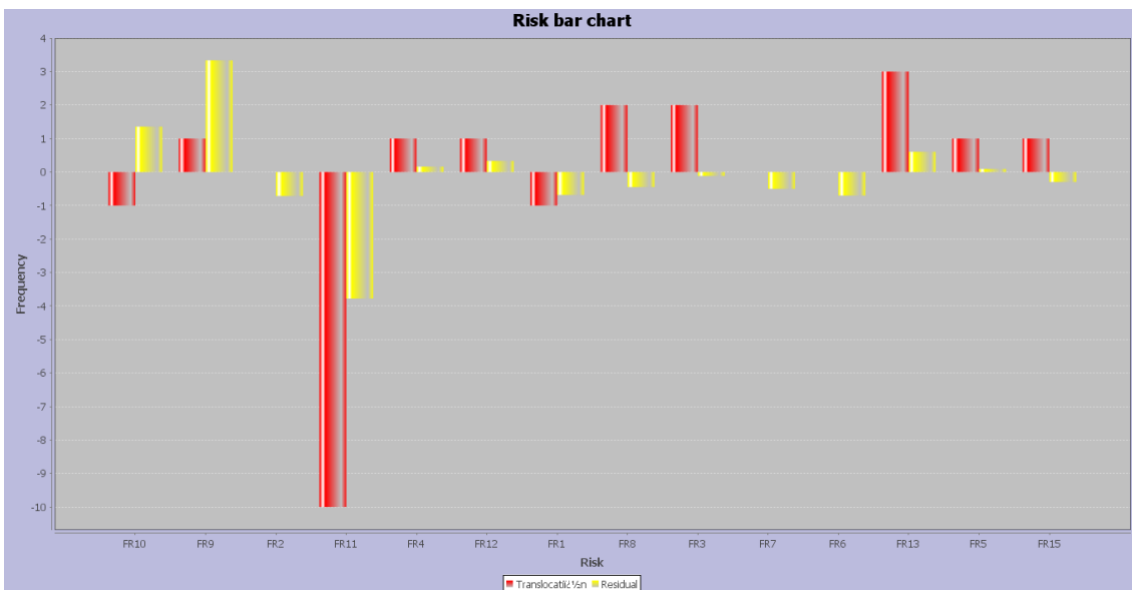


Figura 3.4 Diagramas de barras de Factores de Riesgo del bloque General vs Patrón.

En el caso de los Factores de Riesgo, la poligonal del bloque General es bastante similar con la del patrón, salvo FR9 (Técnica incorrecta) que destaca negativamente presentado una alta frecuencia de aparición. Por otro lado, F11 (Punción dural en epidural) destaca en positivo. El resto de Factores de Riesgo son muy similares (Figura 3.3).

En el diagrama de barras, destaca en positivo FR11 (Punción dural en epidural), tanto en contraste como en translocaciones. En negativo FR9 (Técnica incorrecta) en contraste y F13 (Problemas previos a la intubación) en translocaciones (Figura 3.4).

Que FR11 destaque en positivo entra dentro de lo lógico puesto que en este bloque las técnicas de anestesia locoregional epidural son excepcionales, se utilizan únicamente en determinados pacientes como técnica de analgesia postoperatoria y no tienen la incidencia que pueden presentar otras secciones.

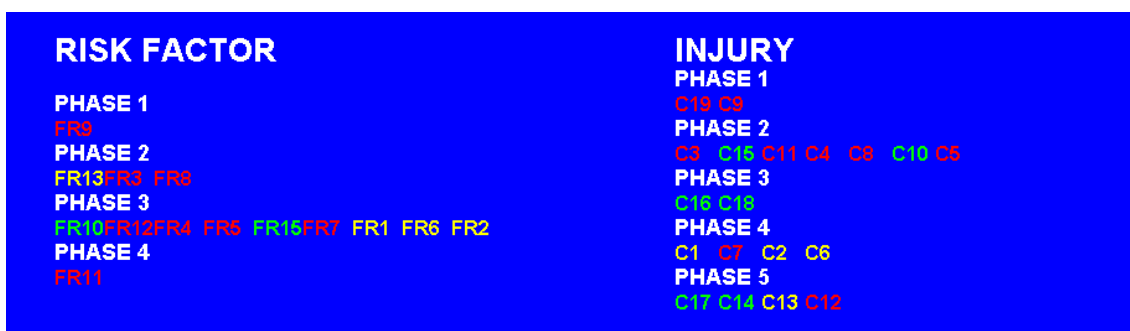


Figura 3.5. FASES del contraste o *Planning Report* del bloque General vs Patrón.

El programa BIOIN ANESTHSOM las valora y presenta en el “*Planning Report*”. Para ordenarlas por fases se consideran translocaciones y contrastes. Las que aparecen en primera fase, es decir, que son de actuación prioritaria son C19 (Hematoma) y C9 (Neumonía). En segunda fase, la primera, es C3 (Arritmias). Para los Factores de Riesgo, en primera fase destaca solo FR9 (Técnica incorrecta). En segunda fase, las dos primeras son FR13 (Problemas previos a la intubación) y FR3 (Alteraciones respiratorias) (Figura 3.5).

Como Complicación y Factor de Riesgo más frecuentes (figuras 3.1 y 3.2) aparecen C19 (Hematoma) y FR9 (Técnica Incorrecta) que, en este caso concreto de este bloque General, ambas coinciden con la de actuación prioritaria de su “*Planning Report*”.

RESULTADOS DEL BLOQUE GENERAL frente al HUMS

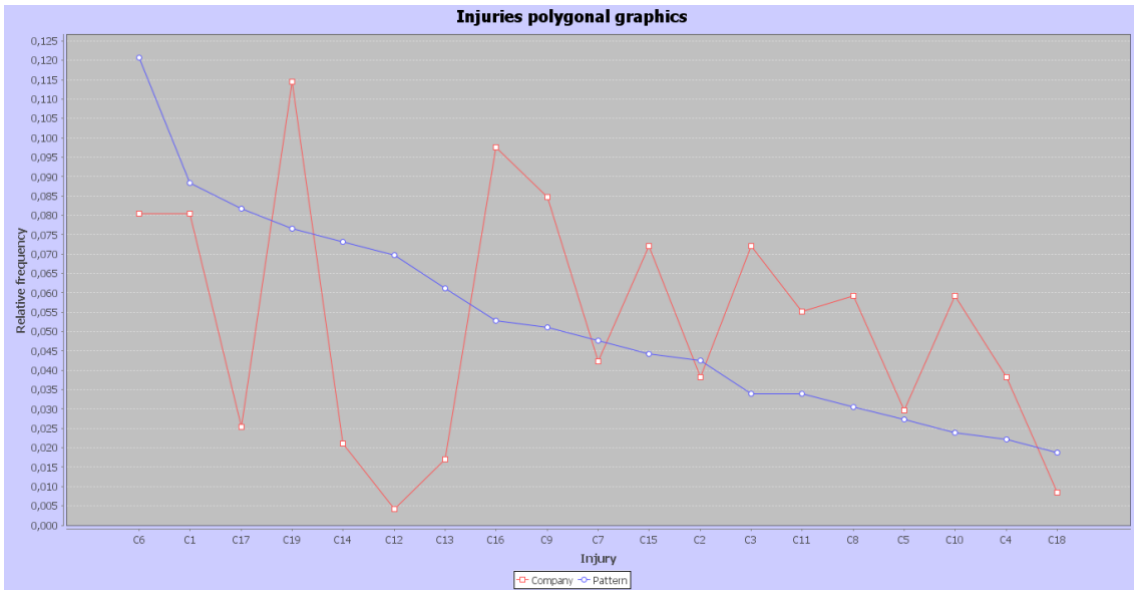


Figura 3.6 Poligonales de Complicaciones del bloque General vs HUMS

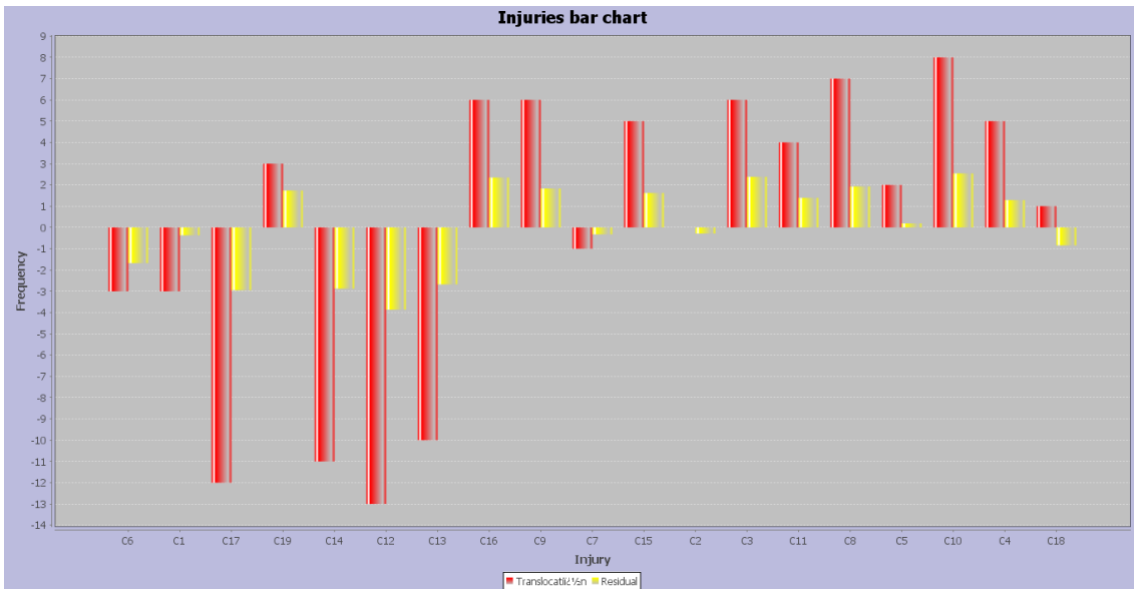


Figura 3.7 Diagramas de barras de Complicaciones del bloque General vs HUMS

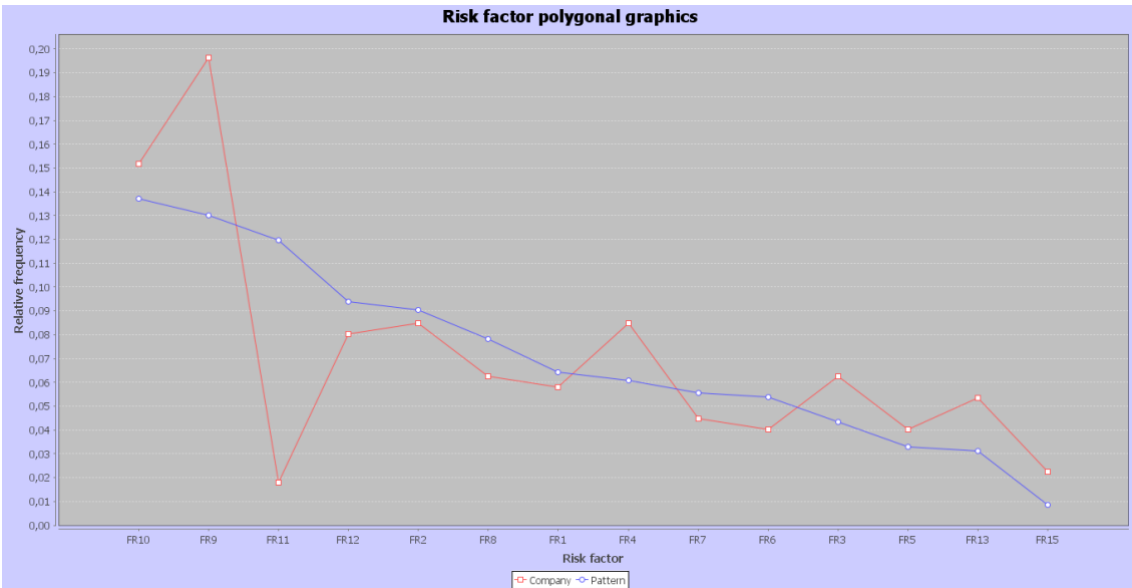


Figura 3.8 Poligonales de Factores de Riesgo del bloque General vs HUMS

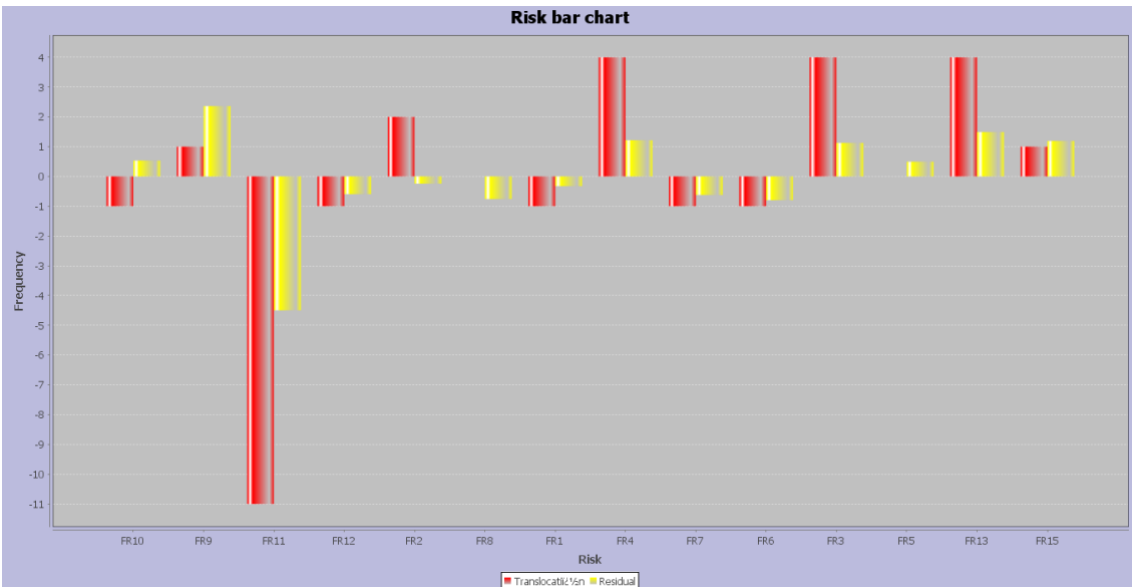


Figura 3.9 Diagramas de barras de Factores de Riesgo del bloque General vs HUMS



Figura 3.10 FASES del contraste o *Planning Report* del bloque General vs HUMS

Cuando se compara el bloque General con el HUMS los resultados son muy distintos a los que vemos cuando comparamos frente al patrón, en primera fase aparecen C10 (Metabólicas), C3 (Arritmias) y C16 (Hiperalgnesia postoperatoria) (Figuras 3.6, 3.7 y 3.10).

Al comparar la sección General con el HUMS los dos primeros Factores de Riesgo coinciden con los que aparecen si se compara con el patrón, FR9 (Técnica incorrecta) en primera fase, y FR13 (Problemas previos a la intubación) el primero de la segunda fase, el tercero en cambio es FR4 (Alteraciones metabólicas) (Figuras 3.8, 3.9 y 3.10).

Si comparamos nuestros resultados con los resultados de ANESTHSOM (Rubio García, 2012), FR9 (Técnica incorrecta), FR10 (Situación Urgente) y FR4 (Alteraciones metabólicas), coinciden en ser las de mayor frecuencia de aparición. Estos resultados son comprensibles debido a que en este bloque General es frecuente la realización de cirugías de urgencia, por lo tanto, es lógico que FR10 (Situación Urgente) destaque aquí.

En el caso de las Complicaciones, aunque los resultados son muy similares existe alguna diferencia. Con nuestros nuevos datos, vemos que la de mayor frecuencia que es C19 (Hematoma) y la tercera en frecuencia C9 (Aspiración/neumonía) coinciden en ambas poligonales, en cambio en segundo lugar, en nuestro trabajo está C16 (Hiperalgnesia postoperatoria), mientras que en ANESTHSOM está C6 (Shock). Estos resultados también son congruentes puesto que la mayor parte de estas complicaciones son producidas en pacientes sometidos a anestesia general, que es la técnica anestésica que más se utiliza en este bloque debido a la idiosincrasia de las especialidades quirúrgicas que en ella trabajan.

2.2 BLOQUE DE TRAUMATOLOGÍA

TRAUMATOLOGÍA frente al PATRÓN

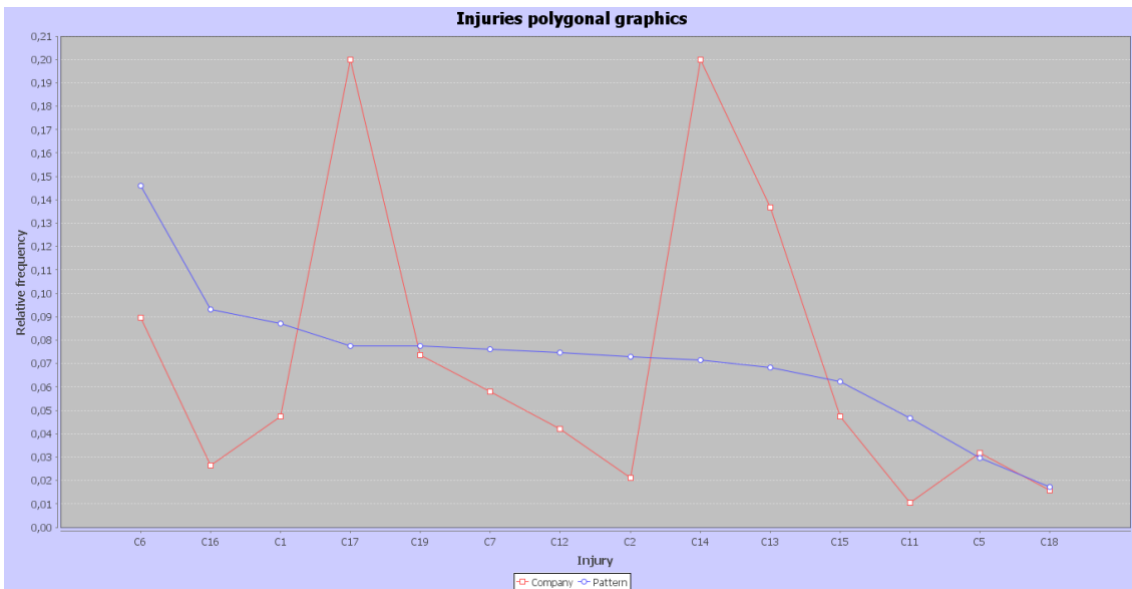


Figura 3.11 Poligonales de Complicaciones de Traumatología vs Patrón.

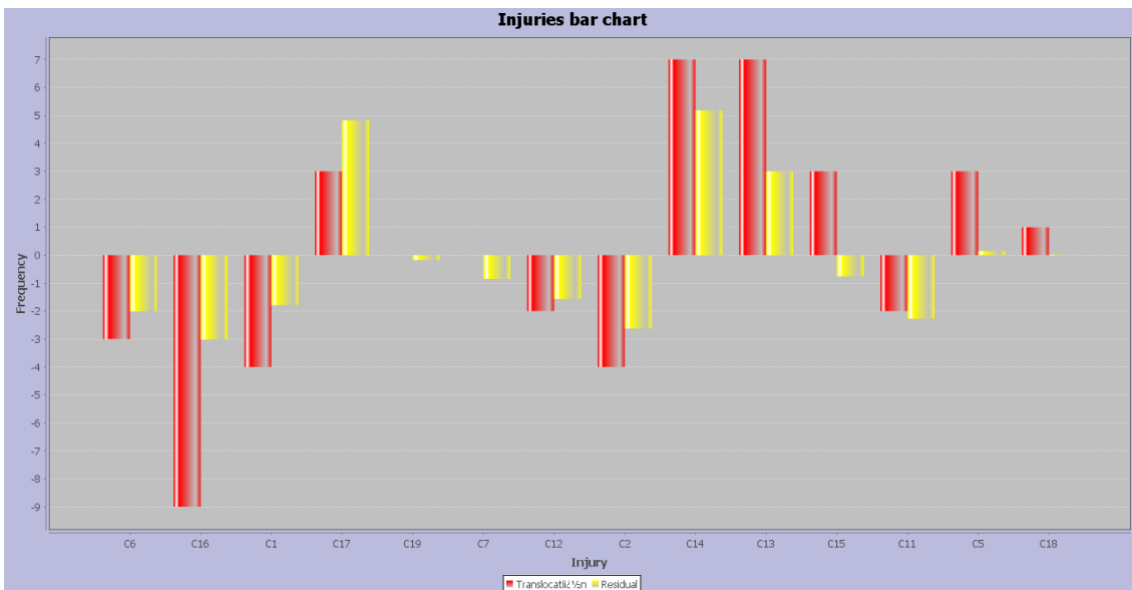


Figura 3.12 Diagramas de barras de Complicaciones de Traumatología vs Patrón.

En Complicaciones se observa una poligonal irregular de Traumatología respecto al Patrón. Destaca la gran frecuencia de aparición que tienen C17 (Neumotórax) y C14 (Coma) y su escasa frecuencia en el patrón. La tercera de mayor aparición es C13 (Lesión nerviosa), que tampoco tiene ninguna similitud con la frecuencia en el patrón. Respecto a las de menor frecuencia, C11 (Complicaciones renales) es la de menor frecuencia seguida por C2 (Edema de glotis) y C16 (Hiperalgnesia postoperatoria). Las tres tienen también poco que ver con el patrón (Figura 3.11)

Estas diferencias son las que se manifiestan en el diagrama de barras. Muy claras en C13 (Lesión nerviosa) y C14 (Coma) tanto en contraste como en translocaciones. En el caso de C17 (Neumotórax) destaca en contraste o residuales, como segunda de mayor variación (Figura 3.12).

Estos resultados tienen lógica puesto que las Complicaciones con más frecuencia de aparición C17 (Neumotórax) y C13 (Lesión nerviosa), son complicaciones directas e bloqueos locorregionales muy usados en esta especialidad. C14 (Coma) que puede parecer chocante, cobra sentido si recordamos que en este bloque se engloba también la especialidad de Neurocirugía.

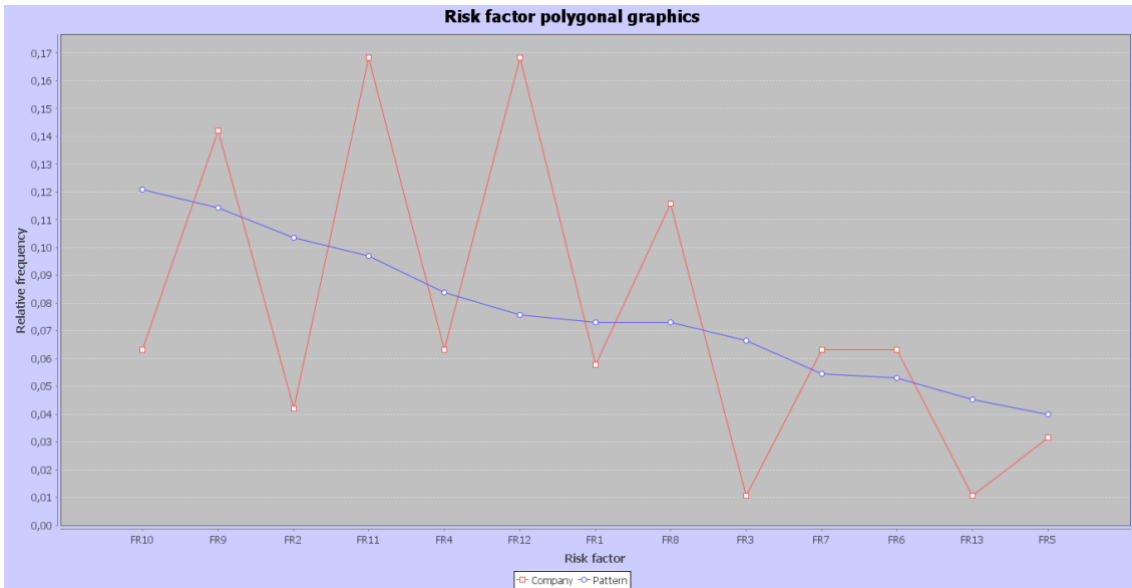


Figura 3.13 Poligonales de Factores de Riesgo de Traumatología vs Patrón.

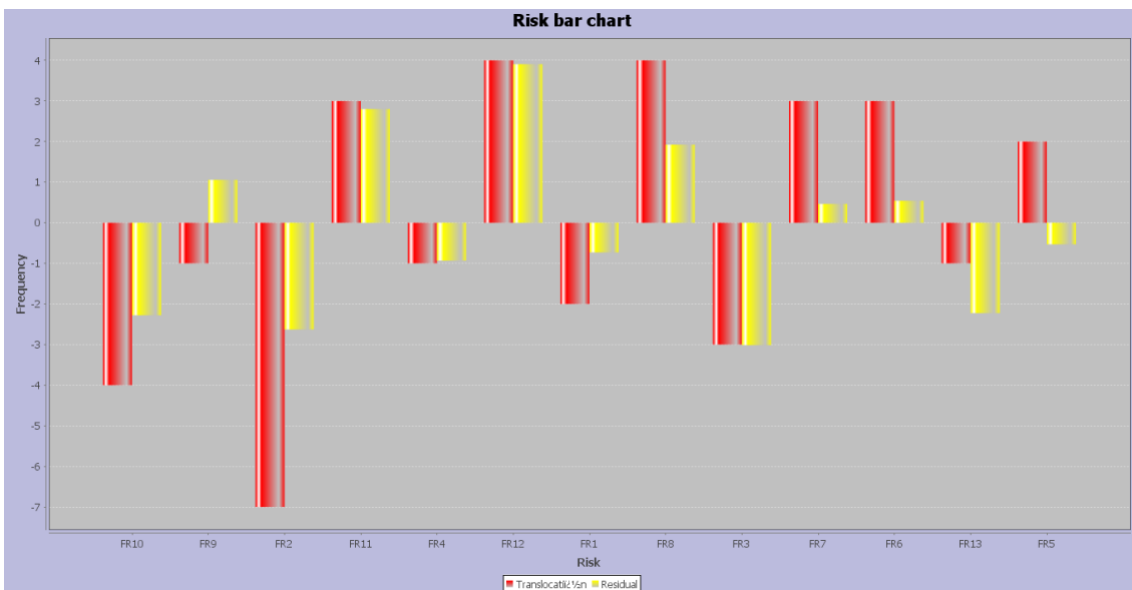


Figura 3.14 Diagramas de barras de Factores de Riesgo de Traumatología vs Patrón.

En el caso de Factores de Riesgo en la poligonal destaca FR11 (Punción dural en epidural), FR12 (Punción arterial no deseada) y FR9 (Técnica incorrecta), con diferencias acusadas frente al patrón los dos primeros y con una frecuencia menos dispar en FR9 (Técnica incorrecta) y en FR8 (Inexperiencia) (Figura 3.13).

Las de menor frecuencia respecto al patrón son FR2 (Alteraciones cardio-circulatorias), FR3 (Alteraciones respiratorias) y FR13 (Problemas previos a la intubación) y FR10 (Situación urgente) (Figura 3.13).

En el diagrama de barras, las diferencias negativas más acusadas están en FR12 (punción arterial no deseada), puesto que destacan tanto en contrastes como en translocaciones y también FR8 (inexperiencia). En positivo FR2 (alteraciones cardio-circulatorias) y FR10 (Situación urgente) (Figura 3.14).

Estos Factores de Riesgo predominantes concuerdan con lo esperado en este bloque, principalmente FR12 (Punción arterial no deseada) puesto que es algo que puede ocurrir cuando se realiza un bloqueo locorreional periférico, que es una de las técnicas anestésicas más utilizadas en este bloque de Traumatología. En la actualidad la tendencia es realizar todas estas técnicas locorreionales guiadas por ecografía con visión directa de las estructuras, y con ello evitar o disminuir la incidencia de punciones arteriales. Una medida de mejora a adoptar podría ser, en este bloque, dotar de ecógrafos suficientes al servicio de Anestesiología, así como formar a los anestesiólogos para su uso adecuado.

RISK FACTOR	INJURY
PHASE 1	PHASE 1
FR12FR11	C14 C17 C13
PHASE 2	PHASE 2
FR8 FR6 FR7 FR5	C5 C15
PHASE 3	PHASE 3
FR9 FR4 FR0	C18 C19 C7
PHASE 4	PHASE 4
FR1	C12 C1
PHASE 5	PHASE 5
FR13	C6 C11 C2 C16
PHASE 6	
FR10FR2 FR3	

Figura 3.15 FASES del contraste o *Planning Report* de Traumatología vs patrón

El programa BIOIN ANESTHSOM las valora y presenta en el “*Planning Report*” como prioritarias C14 (Coma), C17 (Neumotórax) y C13 (Lesión nerviosa). Las tres son de primera fase. Para los Factores de Riesgo, en primera fase aparecen dos, FR12 (punción arterial no deseada) y FR11 (punción dural en epidural). El tercer Factor de Riesgo es el primero de la segunda fase y es FR8 (Inexperiencia) (Figura 3.15).

Según lo visto en la tabla 3.2, la Complicación más frecuente es C14 (Coma) y empatada, con la misma frecuencia, C17 (Neumotórax), que coinciden siendo la primera y segunda en “*Planning Report*”. El Factor de Riesgo más frecuente es FR 12 (Punción arterial no deseada) empatado con FR11 (Punción dural en epidural) que están, por ese orden, en primer y segundo lugar en “*Planning Report*”.

TRAUMATOLOGIA frente a HUMS

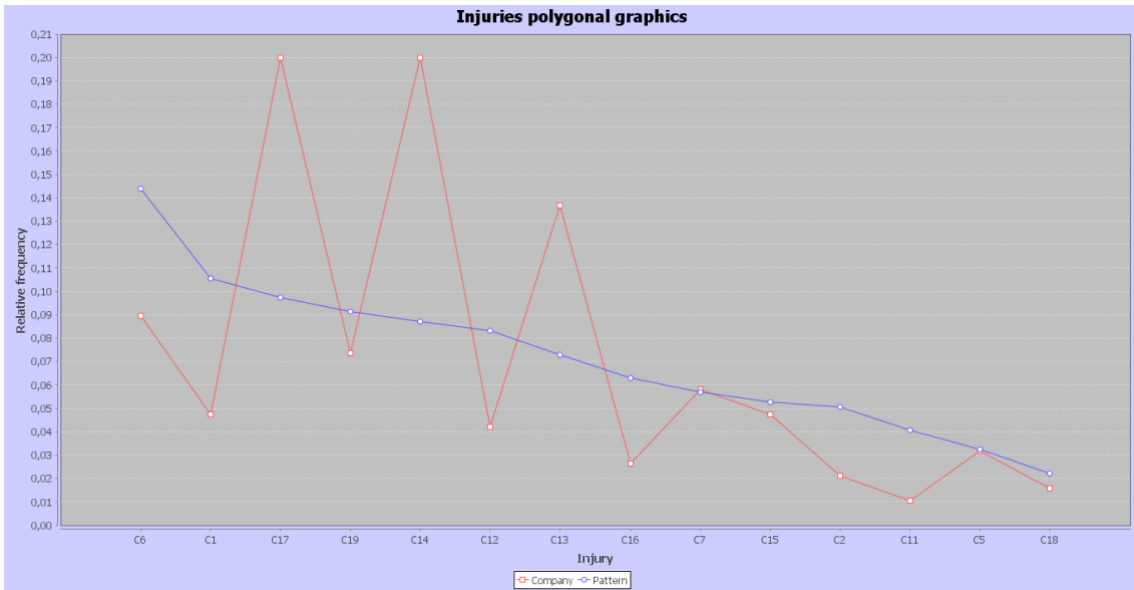


Figura 3.16 Poligonales de Complicaciones de Traumatología vs HUMS

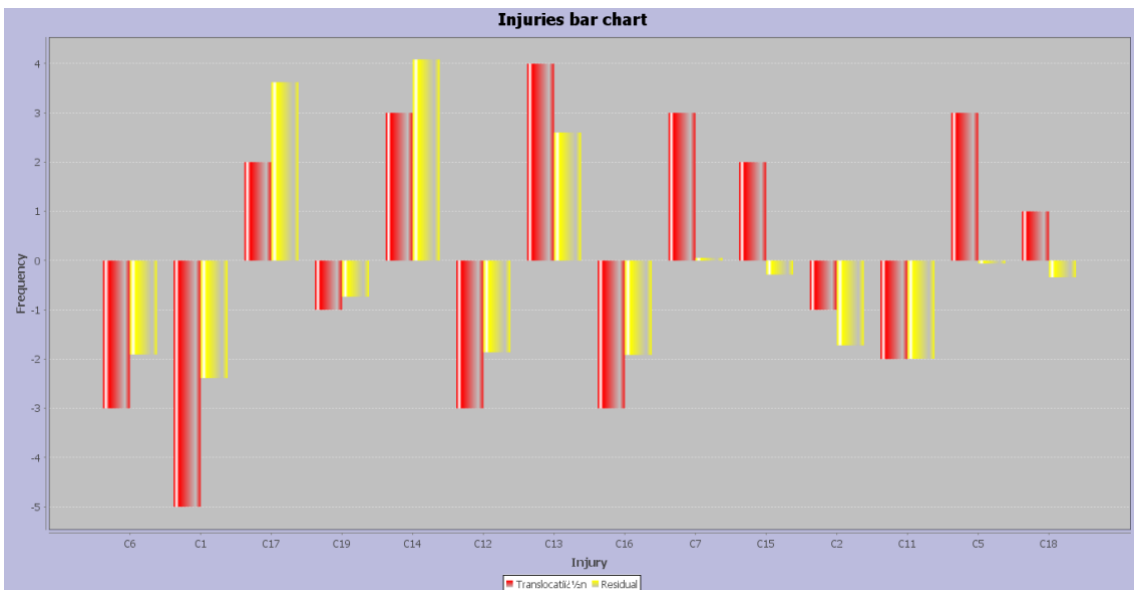


Figura 3.17 Diagramas de barras de Complicaciones de Traumatología vs HUMS

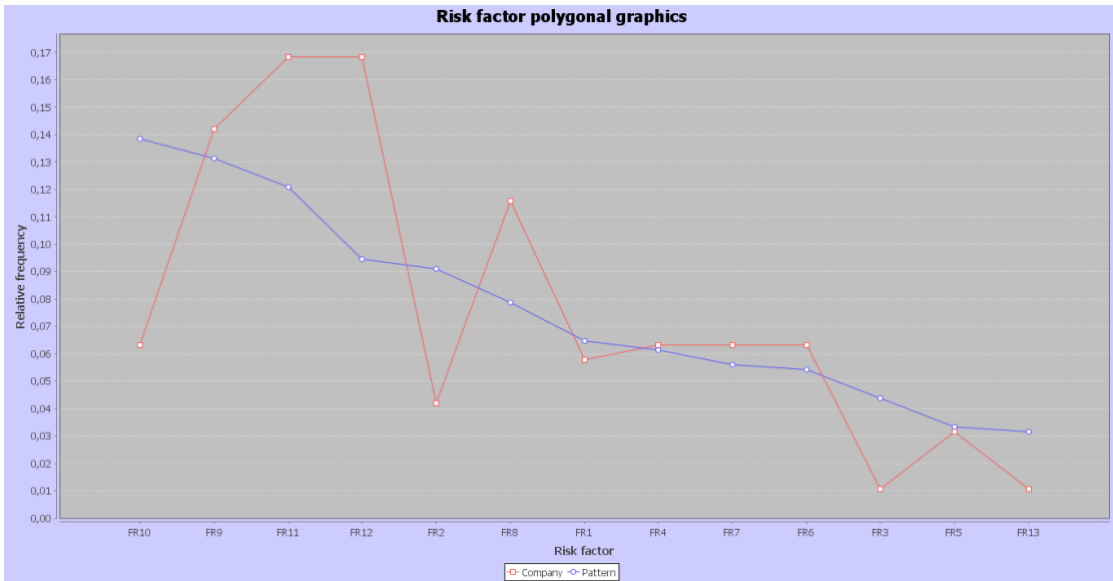


Figura 3.18 Poligonales de Factores de Riesgo de Traumatología vs HUMS

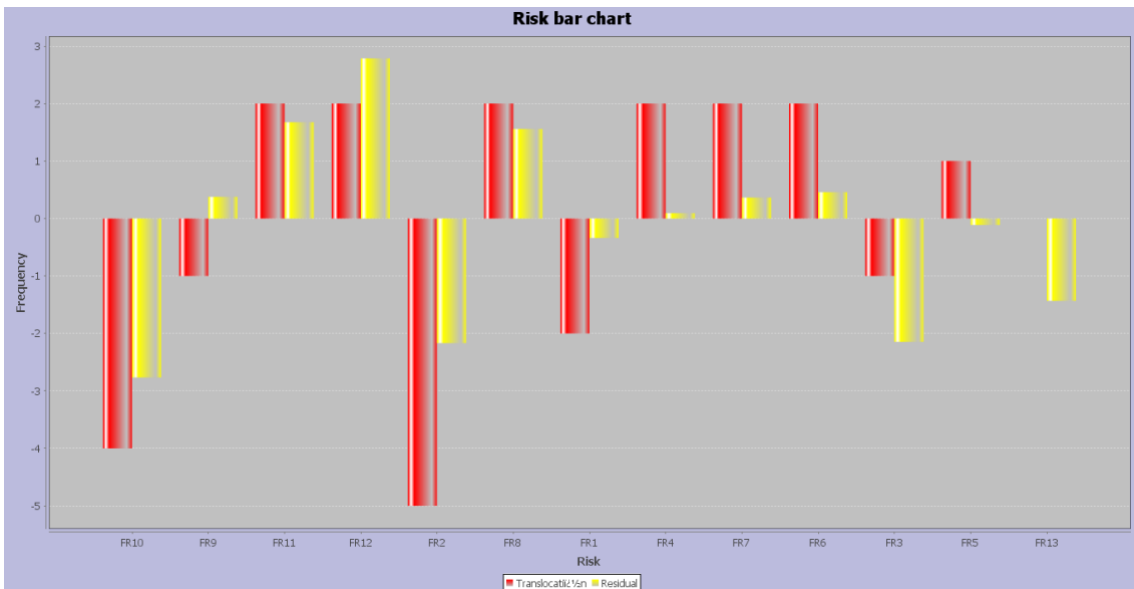


Figura 3.19. Diagramas de barras de Factores de Riesgo de Trauma vs HUMS.

RISK FACTOR	INJURY
PHASE 1	PHASE 1
FR12	C14 C17 C13
PHASE 2	PHASE 2
FR11FR8 FR6 FR7 FR4	C7 C5 C15
PHASE 3	PHASE 3
FR9 FR5 FR13FR0	C18 C19 C2
PHASE 4	PHASE 4
FR1	C12 C6 C16
PHASE 5	PHASE 5
FR3	C11 C1
PHASE 6	
FR2 FR10	

Figura 3.20 FASES del contraste o *Planning Report* de Traumatología vs HUMS

La coincidencia es total al ver las fases en el contraste con el HUMS tanto en Complicaciones como en Factores de Riesgo. Cambian un poco en fase dos, pero las diferencias son muy pequeñas.

Cuando se compara con el HUMS, en primera fase solo aparece FR12 (punción arterial no deseada) y los dos primeros de la segunda fase son FR11 (Punción dural en epidural) y FR8 (Inexperiencia). Coinciden salvo por las fases, pues FR11 (Punción dural en epidural), aparece ahora en segunda fase (Figura 3.20).

Si además comparamos con los resultados del análisis de ANESTHSOM hecho por Rubio García (2012), vemos que los tres Factores de Riesgo coinciden, FR11 (Punción dural en epidural), FR 12 (Punción carótida/humeral) y FR8 (Inexperiencia). Lo mismo ocurre con las Complicaciones, coinciden las 3 en ambos análisis, son: C17 (Neumotórax), C14 (Coma) y C13 (Lesiones nerviosas/bloqueo neurológico prolongado).

Tanto en Factores de Riesgo como en Complicaciones estos resultados son lógicos porque todas son derivadas de la anestesia locorreional (con diferencia la más utilizada en cirugía traumatológica) además, en este bloque en el Miguel Servet se encuentra también Neurocirugía, por lo tanto, como hemos comentado anteriormente, es comprensible que aparezca C14 (Coma).

2.3 MATERNO-INFANTIL

MATERNO-INFANTIL frente a PATRÓN

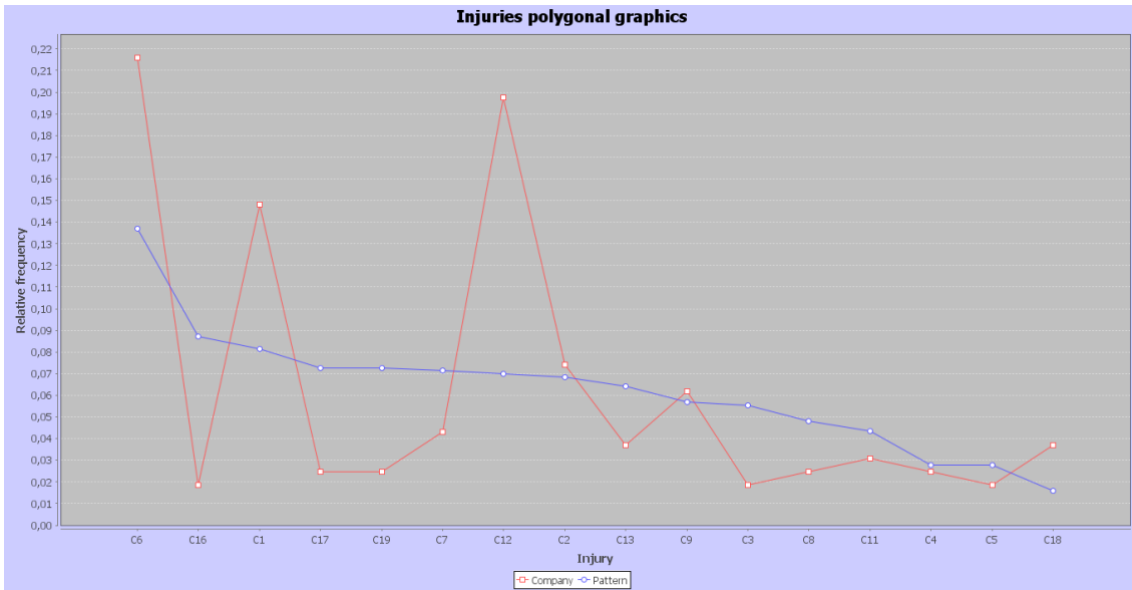


Figura 3.21 Poligonales de Complicaciones del Materno-Infantil vs Patrón.

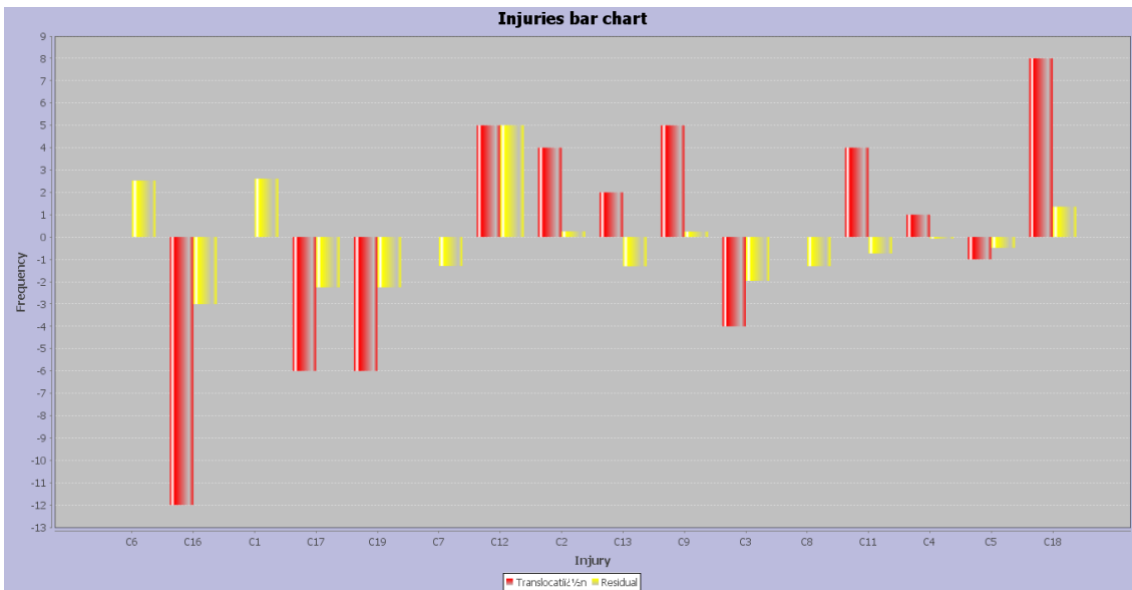


Figura 3.22 Diagramas de barras de Complicaciones del Materno-Infantil vs Patrón.

En la poligonal de “Complicaciones” destaca su irregularidad comparada con el patrón. En negativo, es decir, están peor que en el patrón, C6 (Shock), C12 (Cefalea) y C1 (Rotura de dientes). La mayoría están en positivo, o sea, están mejor que el patrón, de ellos destacamos C16 (Hiperalgia postoperatoria), C17 (Neumotórax), C19 (Hematoma) y C3 (Arritmias) (Figura 3.21).

Es llamativa la alta frecuencia relativa de C6 (Shock) que también es la más alta en el patrón, y de C12 (Cefalea) que en el patrón presenta una frecuencia intermedia, lo que es lógico porque la cefalea postpunción es una de las complicaciones más frecuentes de la analgesia epidural. La tercera de mayor frecuencia es C1 (Rotura de dientes) que coincide siendo también la tercera más frecuente en el patrón, también es lógico porque la gestante, por los cambios morfológicos que ocurren durante este periodo, suele presentar dificultad de intubación orotraqueal, además, por definición se les considera paciente con estómago lleno y la intubación se realiza mediante técnica de secuencia rápida. Destacamos también que C18 (Infección), la de menor frecuencia en el Patrón, en esta sección está por encima de forma clara.

Respecto a las Complicaciones menos frecuentes en la sección Materno-Infantil que en el Patrón, es interesante resaltar C16 (Hiperalgia postoperatoria). También arrojan mejores resultados que en el Patrón C17 (Neumotórax), C19 (Hematoma) y C3 (Arritmias).

Estas diferencias son las que se manifiestan en el diagrama de barras. C18 (Infección), C9 (Neumonía) y C12 (Cefalea) con unas diferencias de translocaciones muy claras, y en contraste destaca C12 (Cefalea) (Figura 3.22).

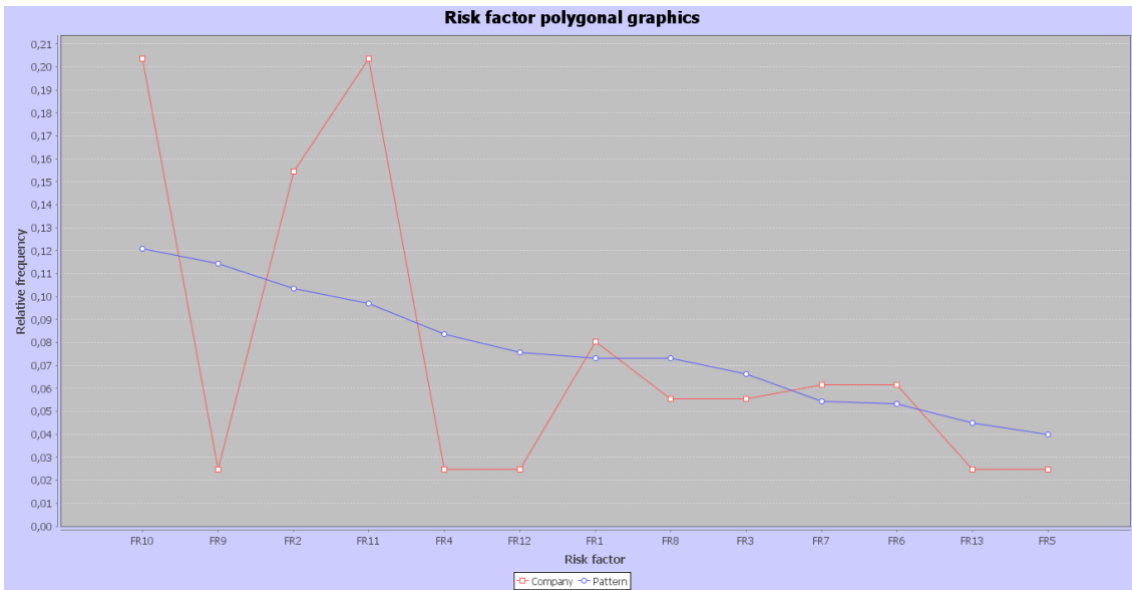


Figura 3.23 Poligonales de Factores de Riesgo del Materno-Infantil vs Patrón.

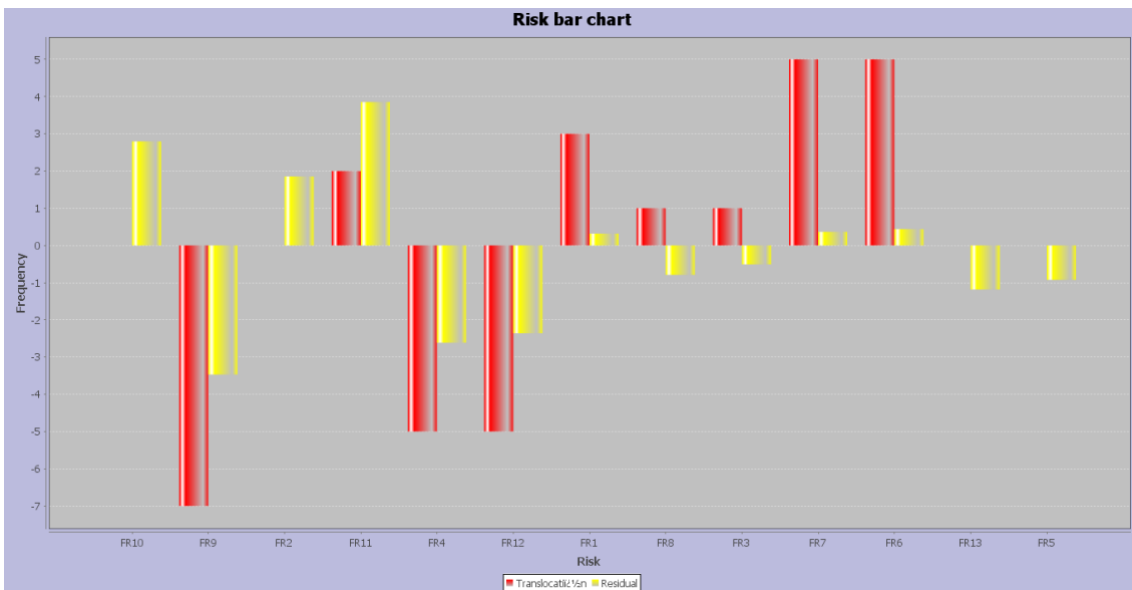


Figura 3.24 Diagramas de barras de Factores de Riesgo del Materno-Infantil vs Patrón.

Si interpretamos la poligonal de Factores de Riesgo, destacan FR10 (Situación urgente) y FR11 (Punción dural en epidural) como las más frecuentes, seguidas de FR2 (Alteraciones cardio-circulatorias). Las diferencias de las tres, respecto con el patrón son reseñables. En el caso concreto de FR10 (Situación urgente) en ambas poligonales es la complicación con mayor frecuencia de aparición.

Las tres menos frecuentes son FR9 (Técnica incorrecta), FR4 (Alteraciones metabólicas) y FR12 (Punción arterial no deseada), la diferencia más acusada con respecto al patrón está en FR9 (Técnica incorrecta) (Figuras 3.23 y 3.24).

En este caso también es lógico la alta frecuencia de estos Factores de Riesgo. En el hospital materno-infantil muchas cirugías, principalmente cesáreas, se hacen de urgencia. Además, la alta incidencia de analgesia epidural en los partos hace que las complicaciones de esta técnica aumenten con respecto a otras secciones. FR2 (Alteraciones cardio-circulatorias) también es lógico porque muchas cirugías de la embarazada cursan con sangrado importante, de hecho, la hemorragia obstétrica es la primera causa de muerte en la embarazada en nuestro medio.

RISK FACTOR	INJURY
PHASE 1	PHASE 1
FR11 FR10	C12 C6
PHASE 2	PHASE 2
FR6 FR7 FR1	C1
PHASE 3	PHASE 3
FR2 FR3 FR8 FR5 FR13	C18 C2 C9 C11 C13
PHASE 4	PHASE 4
FR12 FR4 FR9	C4 C5 C7 C8
	PHASE 5
	C3 C17 C19 C16

Figura 3.25 FASES del contraste o *Planning Report* del Materno-Infantil vs Patrón.

El programa BIOIN ANESTHSOM las valora y presenta en el “*Planning Report*”, en primera fase, aparecen FR11 (Punción dural en epidural) y FR10 (Situación urgente) y en segunda fase la primera es FR6 (Alteraciones neurológicas). En Complicaciones, de primera fase son C12 (Cefalea) y C6 (Shock). En segunda fase aparece una única Consecuencia, C1 (Rotura de dientes) (Figura 3.25).

La Complicación con mayor frecuencia de aparición (tabla 3.2) es C6 (Shock), que es la segunda en “*Planning Report*”, la primera de actuación prioritaria es C12 (Cefalea). Los dos Factores de Riesgo más frecuentes (tabla 3.1) son FR10 (Situación urgente) y FR11 (Punción dural en epidural), en el “*Planning Report*” podemos ver que son los mismos los que necesitan actuación prioritaria, pero cambian el orden.

MATERNO-INFANTIL frente a HUMS

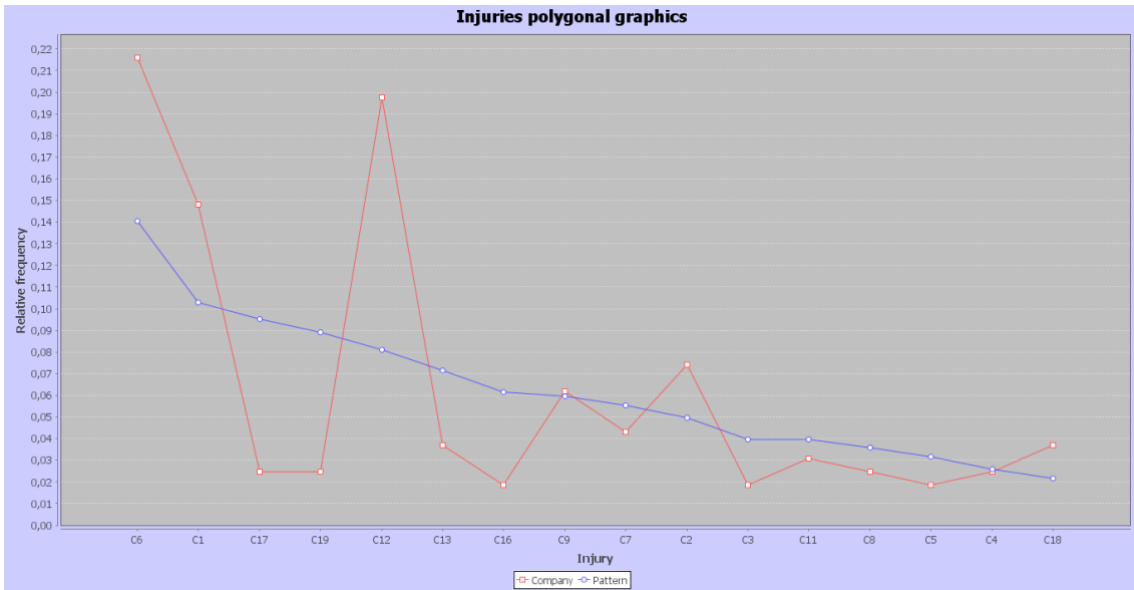


Figura 3.26 Poligonales de Complicaciones del Materno-Infantil vs HUMS



Figura 3.27 Diagramas de barras de Complicaciones del Materno-Infantil vs HUMS

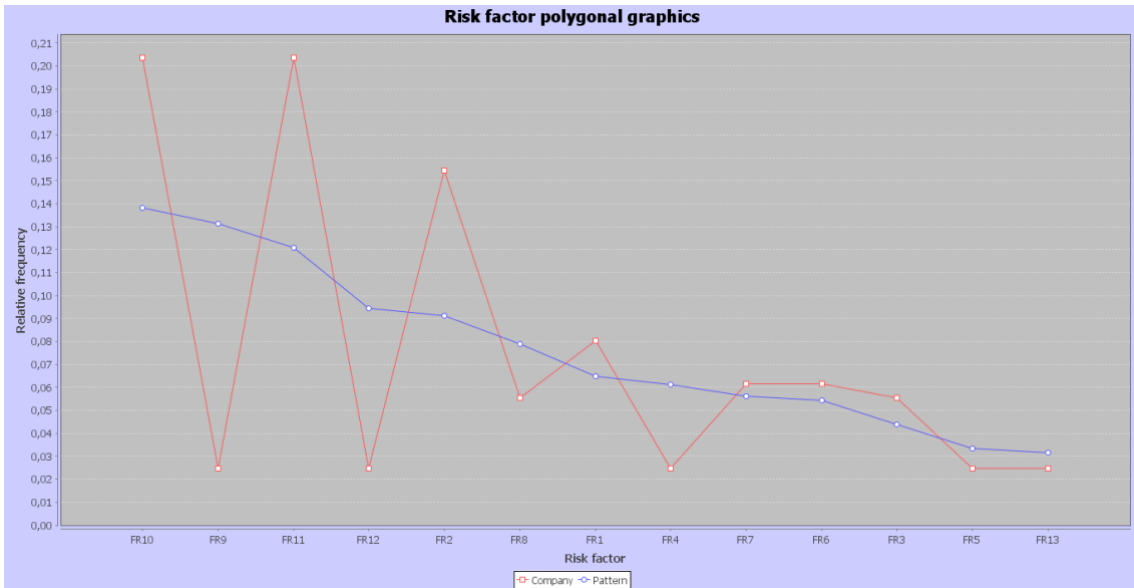


Figura 3.28 Poligonales de Factores de Riesgo del Materno-Infantil vs HUMS

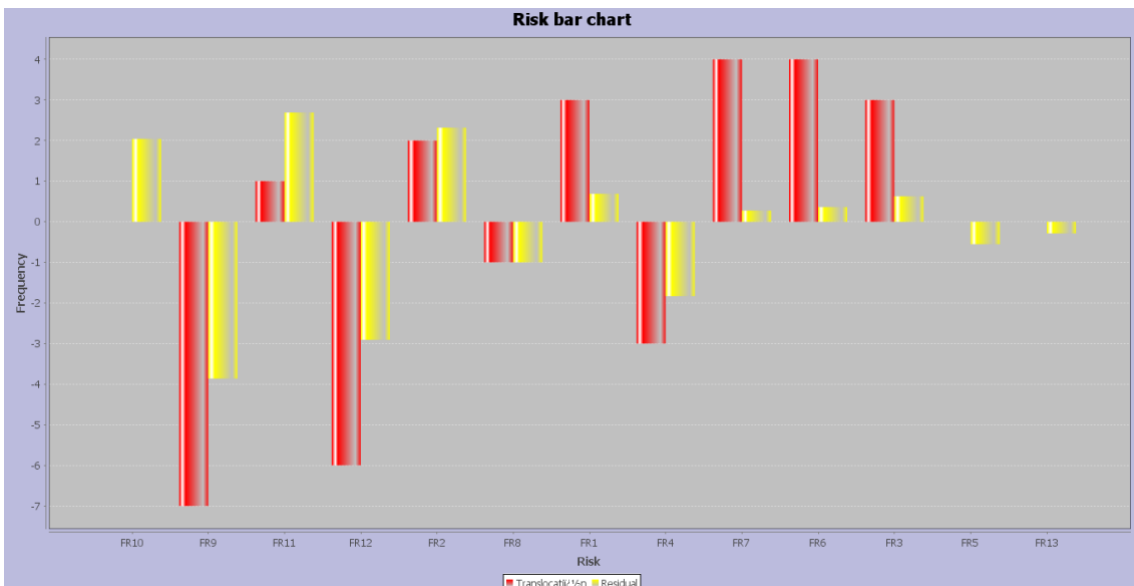


Figura 3.29 Diagramas de barras de Factores de Riesgo del Materno-Infantil vs HUMS

RISK FACTOR	INJURY
PHASE 1	PHASE 1
FR2 FR10	C12 C6
PHASE 2	PHASE 2
FR11	C2 C18 C9 C4 C11 C7
PHASE 3	PHASE 3
FR1 FR3 FR6 FR7	C1 C8 C13
PHASE 4	PHASE 4
FR13 FR5 FR8	C5 C3
PHASE 5	PHASE 5
FR4	C16 C19 C17
PHASE 6	
FR12 FR9	

Figura 3.30 FASES del contraste o *Planning Report* del Materno-Infantil vs HUMS

Al analizar el contraste frente a HUMS, en primera fase aparecen igualmente C12 (Cefalea) y C6 (Shock). En segunda fase aparecen seis consecuencias, pero la primera es C2 (Edema de glotis). C1 (Rotura de dientes) pasa a tercera fase. Al realizar el contraste de esta sección con el HUMS los tres Factores de Riesgo primeros, son FR2 (Alteraciones cardio-circulatorias) y FR10 (Situación urgente) en la primera fase y FR11 (Punción dural en epidural) único que aparece en la segunda. Desaparece FR6 (Alteraciones neurológicas) y entra en su lugar FR2 (Alteraciones cardio-circulatorias) (Figura 3.30).

Si comparamos con los resultados de Rubio García (2012) vemos que en el Hospital Materno-infantil coinciden los tres más frecuentes: FR10 (Situación urgente), FR11 (Punción dural en epidural) y FR2 (alteraciones cardio-circulatorias). Solo hay pequeñas diferencias en los siguientes, en nuestro trabajo FR 1 (alteraciones antropométricas) sería la cuarta frecuencia y en ANESTHSOM se encuentra en quinto puesto. En Complicaciones los resultados también son muy similares, predominan C6 (Shock), C12 (Cefalea) y C1 (rotura de dientes) en ambos. Estos problemas, como ya hemos avanzado previamente, son coherentes teniendo en cuenta que en este hospital donde se atienden un gran número de embarazadas, todas estas circunstancias tienen cabida en este tipo de pacientes e intervenciones que se realizan. Un estudio reciente en un hospital pediátrico francés (Kupersztych-Hagege, Duracher-Gout, Ortego, Carli y Orliaguet, 2017) obtuvo como complicación más frecuente la patología respiratoria con un 28,8%, resultado que no concuerda con nuestro análisis, esto puede ser debido a que nuestra recogida de datos tiene deficiencias y que nuestro principal objetivo es demostrar que este método funciona.

Se han analizado las tres secciones General, Traumatología y Materno-infantil con el Hospital Universitario Miguel Servet y se han comparado los resultados que hemos obtenido con BIOIN ANESTHSOM y nuestros nuevos datos con los resultados obtenidos en ANESTHSOM (Rubio García, 2012) y no

se han encontrado cambios significativos, lo que demuestra la fortaleza del método, es un método robusto.

2.4 HOSPITAL UNIVERSITARIO MIGUEL SERVET

HUMS frente a PATRÓN

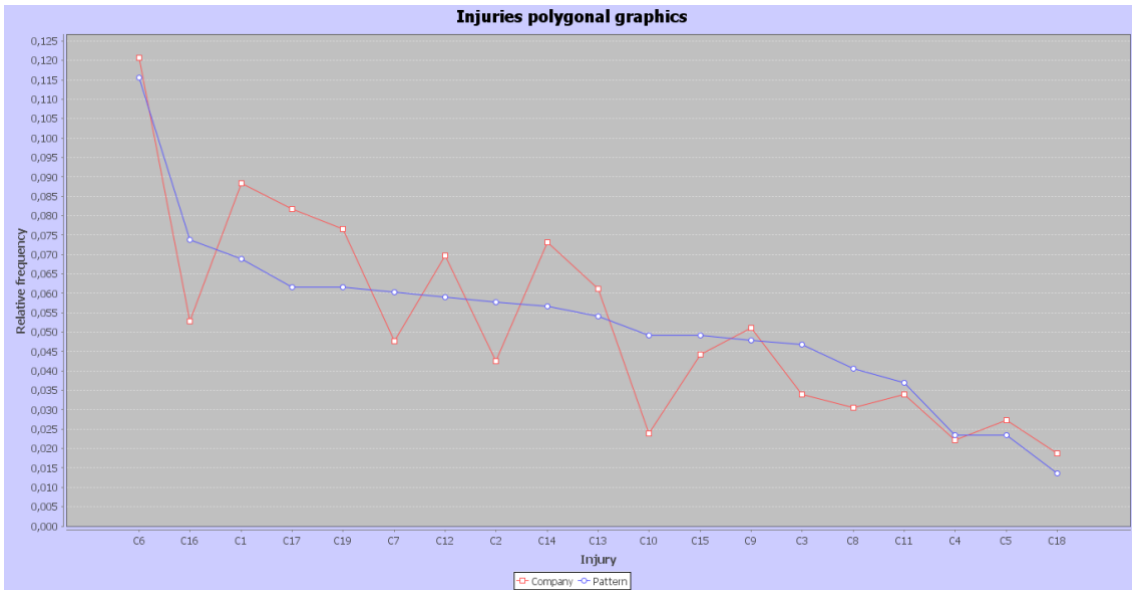


Figura 3.31 Poligonales de Complicaciones del HUMS vs Patrón.

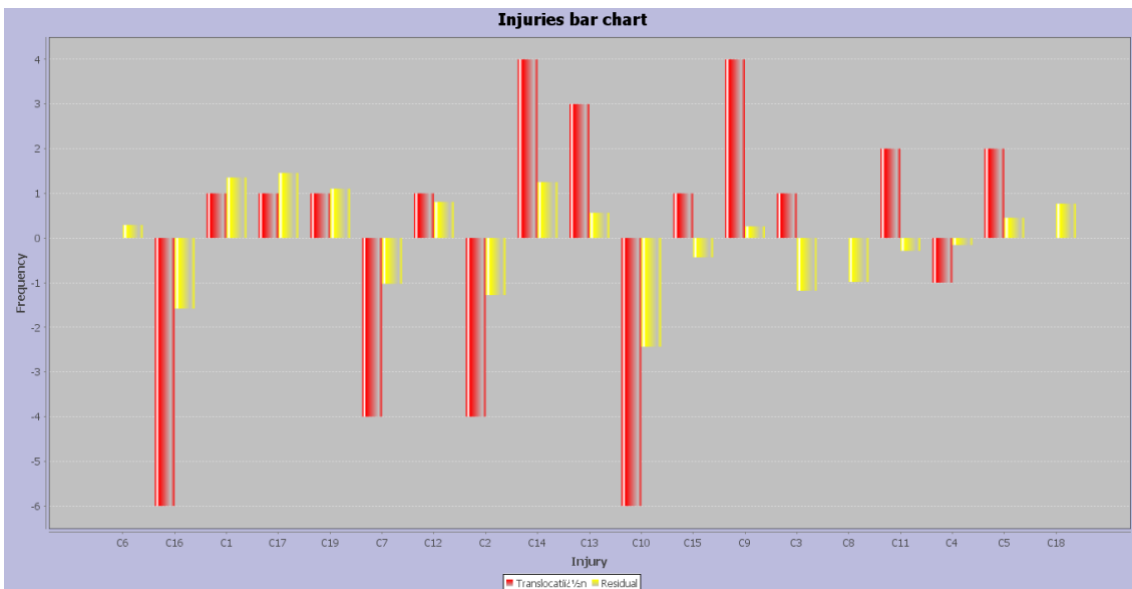


Figura 3.32 Diagramas de barras de Complicaciones del HUMS vs Patrón.

En Complicaciones se observa que la poligonal es la más próxima al patrón. Tiene diferencias, pero muy pequeñas. Destacamos en negativo, C1 (Rotura de diente/s), C14 (Coma) y C17 (Neumotórax). En positivo C10 (Metabólicas), C2 (Edema de glotis) y C7 (Laringo/Broncoespasmo) (Figura 3.31).

En diagramas de barras C14 (Coma), C9 (Aspiración/Neumonía) y C13 (Lesión Nerviosa/Bloqueo Nervioso Prolongado). En el plano positivo destacan C16 (Hiperalgia postoperatoria), C10 (Metabólicas), C7 (Laringo/Broncoespasmo) y C2 (Edema de glotis). En contraste en negativo no destaca ninguna y en positivo solo destaca C10 (Metabólicas). Es decir, C10 (Metabólicas) de manera significativa está mejor en el HUMS que en el patrón tanto en translocación como en contrastes. (Figura 3.32).

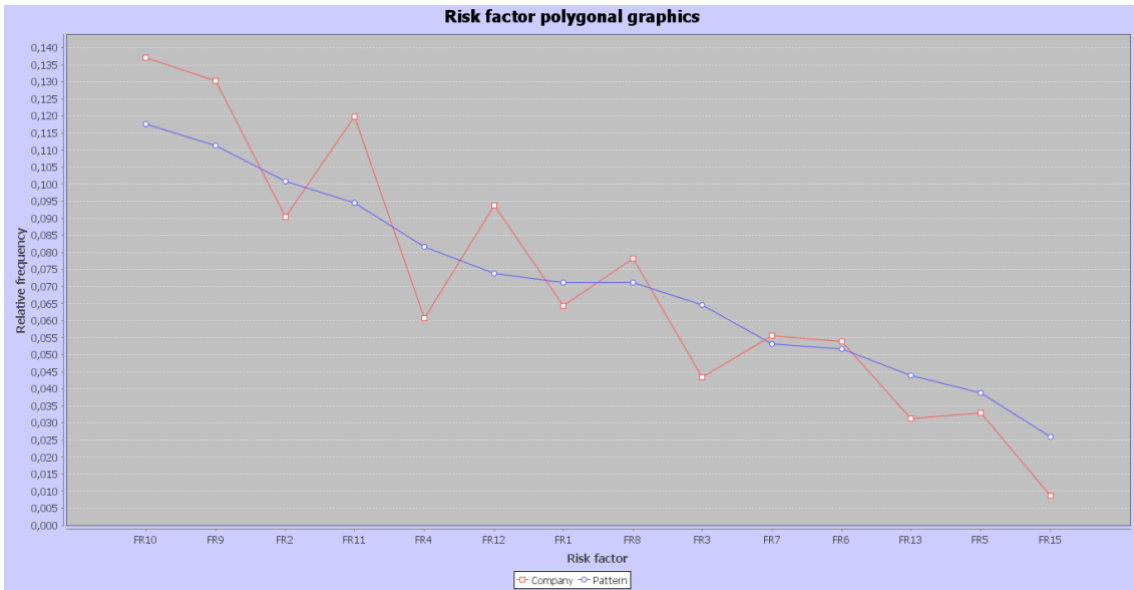


Figura 3.33 Poligonales de Factores de Riesgo del HUMS vs Patrón.

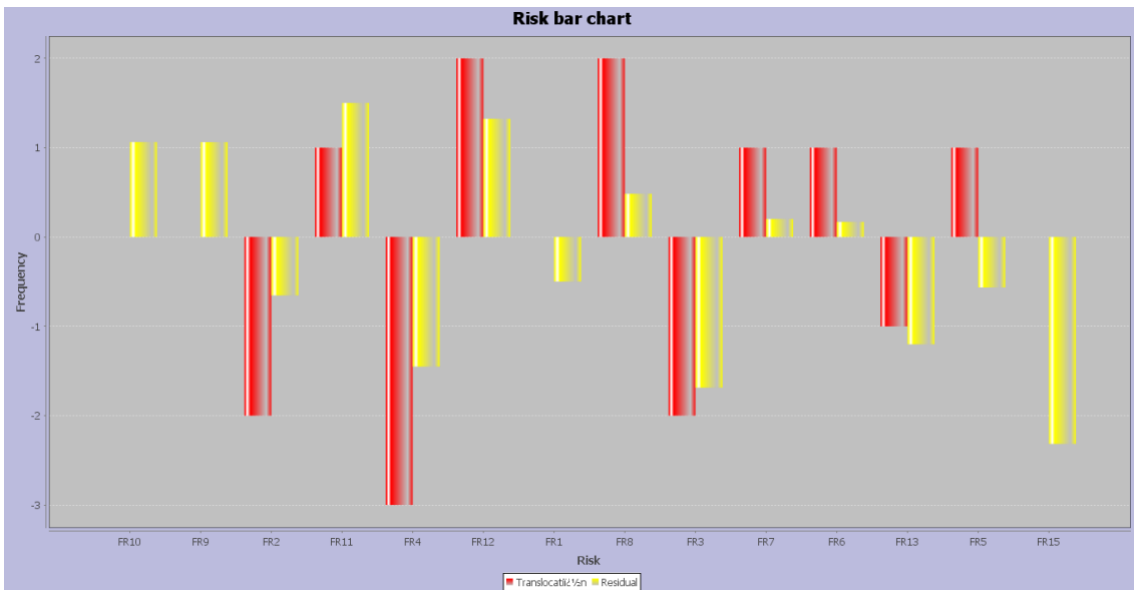


Figura 3.34. Diagramas de barras de Factores de Riesgo del HUMS vs Patrón.

En el caso de Factores de Riesgo la poligonal también es muy parecida al patrón. Las diferencias más claras en negativo son FR11 (Punción dural en epidural), FR12 (Punción arterial no deseada) y FR9 (Técnica incorrecta. Que estén mejor que en el patrón, destacan FR3 (Alteraciones respiratorias), FR4 (Alteraciones metabólicas) y FR13 (Problemas previos en la intubación) (Figura 3.33).

Al analizar los diagramas de barras en negativo solo destacan FR12 (Punción arterial no deseada) y FR8 (Inexperiencia). En positivo FR4 (Alteraciones metabólicas), FR2 (Alteraciones cardio-circulatorias) y FR3 (Alteraciones respiratorias). En contrastes o residuales en negativo FR11 (Punción dural en epidural) y FR12 (Punción arterial no deseada) y en positivo FR3 (Alteraciones respiratorias) y FR4 (Alteraciones metabólicas) (Figura 3.34).

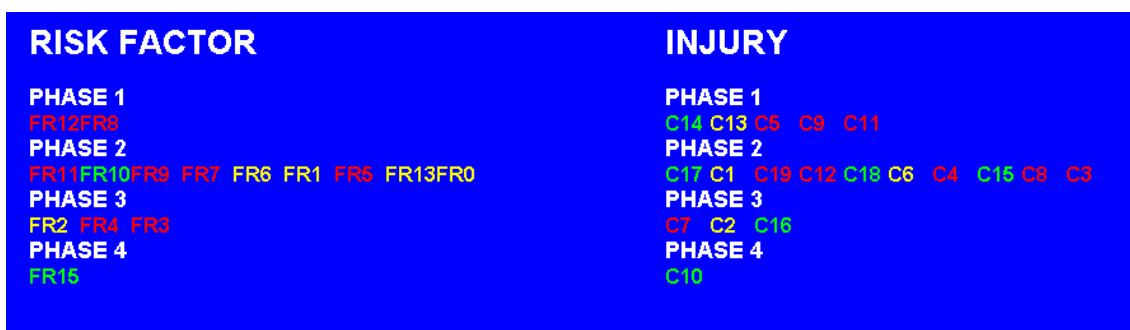


Figura 3.35 FASES del contraste o *Planning Report* del HUMS vs Patrón.

El programa BIOIN ANESTHSOM las valora y presenta en el “*Planning Report*” como prioritarias, todas en fase primera, C14 (Coma), C13 (Lesión Nerviosa/Bloqueo Nervioso Prolongado) y C5 (Tromboembolismo pulmonar). En este caso en fase primera están también C9 (Aspiración/Neumonía) y C11 (Renales), llama la atención que se encuentren cinco Complicaciones en primera fase.

Obsérvese la importancia del análisis matemático-estadístico frente al visual. C1 (Rotura de diente/s) y C17 (Neumotórax) que destacaban, aparecen en los lugares 7º y 6º respectivamente.

En el “*Planning Report*” de los Factores de Riesgo, como prioritarias aparecen FR12 (Punción arterial no deseada) y FR8 (Inexperiencia), coincidiendo con el diagrama de barras de contrastes. La primera de la segunda fase es FR11 (Punción dural en epidural), que aparecía en translocaciones como la primera (Figura 3.35).

En esta sección, la Complicación con mayor frecuencia de aparición (tabla 3.2) es C6 (Shock), que no está recogida entre las tres primeras en necesitar ser mejoradas, la primera en el “*Planning Report*”, es C14 (Coma). El Factor de Riesgo que más frecuentemente aparece (tabla 3.1) es FR10 (Situación urgente), que no se encuentra entre las tres primeras que indica el “*Planning Report*” que necesitan mejorar. El primero en este caso es FR12 (Punción arterial no deseada). Estos resultados ponen de manifiesto la importancia del análisis estadístico que se realiza con BIOIN, ya que un análisis de frecuencias por sí solo, no nos aporta la misma información que si lo comparamos con el patrón.

2.5 HOSPITAL CLINICO UNIVERSITARIO LOZANO BLESA

HCULB frente a PATRÓN

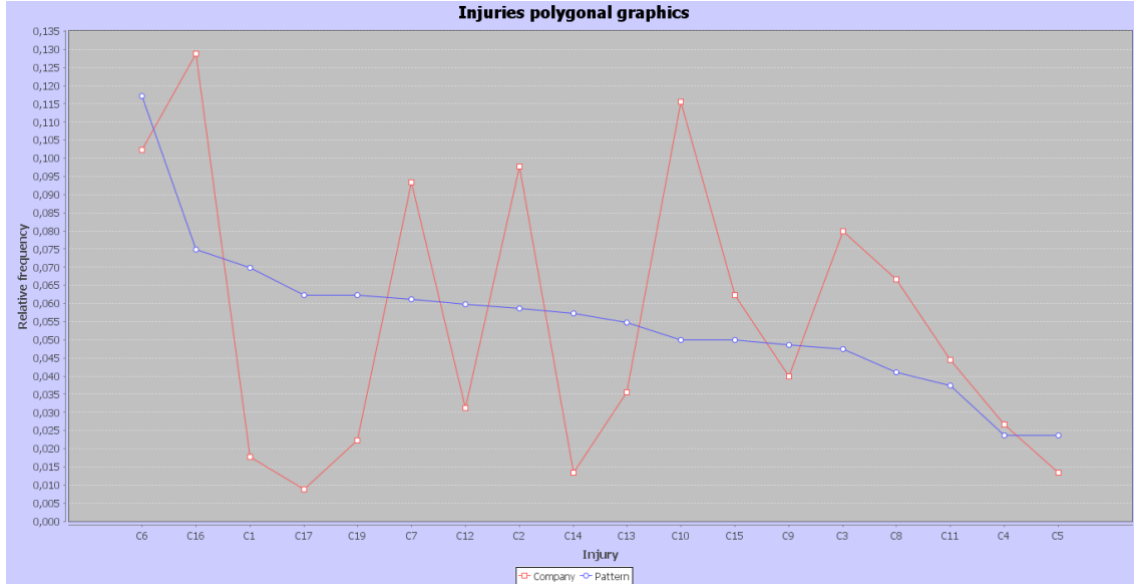


Figura 3.36 Poligonales de Complicaciones del HCULB vs Patrón.

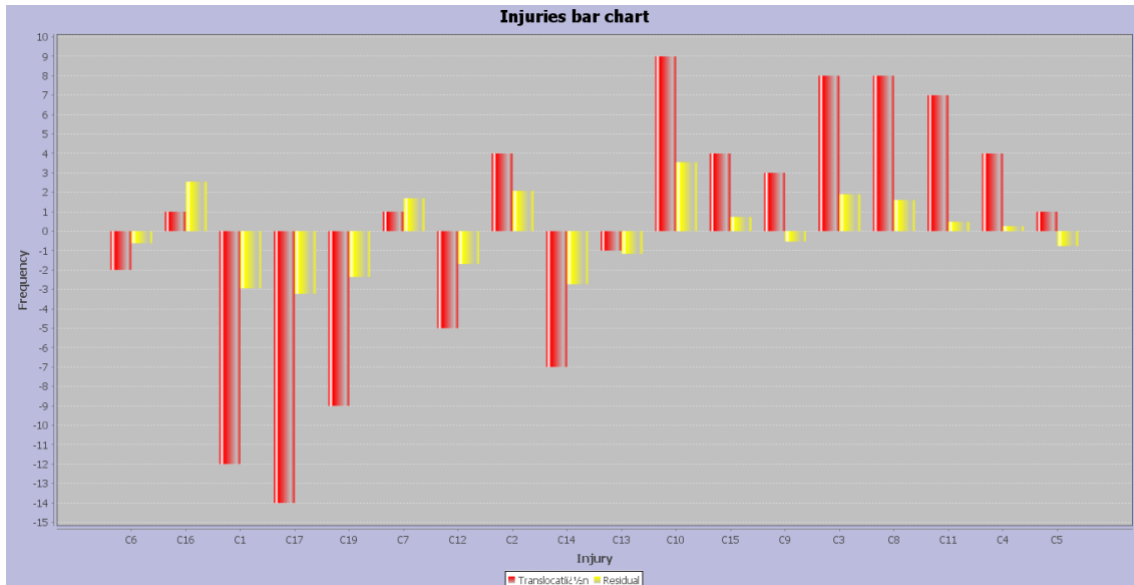


Figura 3.37. Diagramas de barras de Complicaciones del HCULB vs Patrón.

En Complicaciones se observa unas diferencias muy acusadas, tanto en la poligonal como en el diagrama de barras.

En la poligonal destacan en negativo, es decir, están peor en el Hospital Clínico que en el patrón: C16 (Hiperalgnesia postoperatoria), C2 (Edema de glotis), C10 (Metabólicas), C8 (Atelectasia) y C7 (Laringo/Broncoespasmo). Mejor que en el patrón están C14 (Coma), C17 (Neumotórax), C1 (Rotura de diente/s) y C19 (Hematoma).

En el diagrama de barras en negativo destacan C10 (Metabólicas), C3 (Arritmias) y C8 (Atelectasia) y en positivo, es decir, mejor que en el patrón: C1 (Rotura de diente/s) C17 (Neumotórax) y C19 (Hematoma) (Figura 3.36).

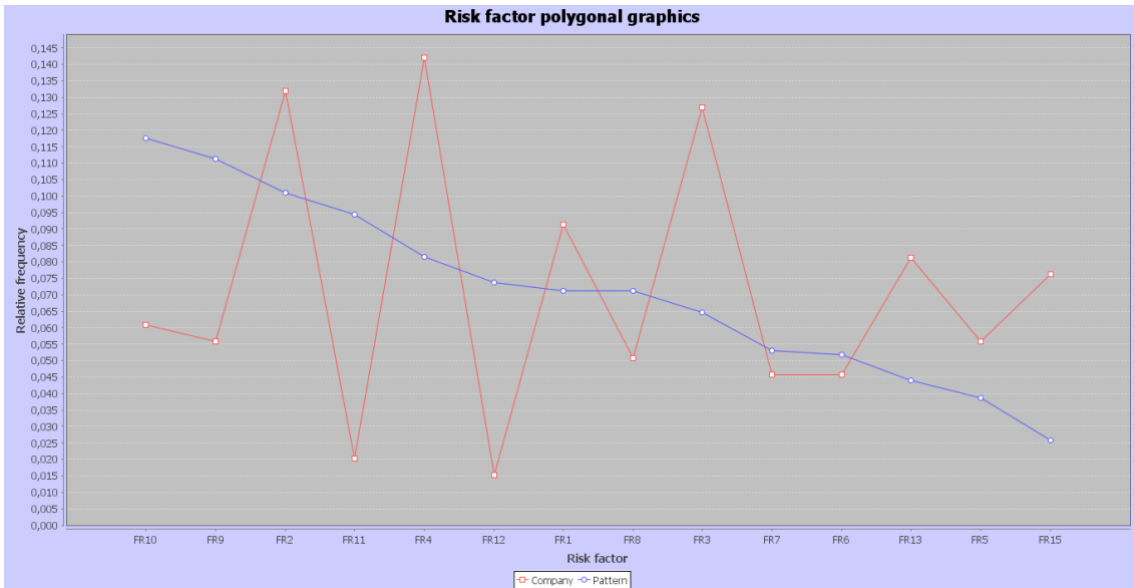


Figura 3.38. Poligonales de Factores de Riesgo del HCULB vs Patrón.

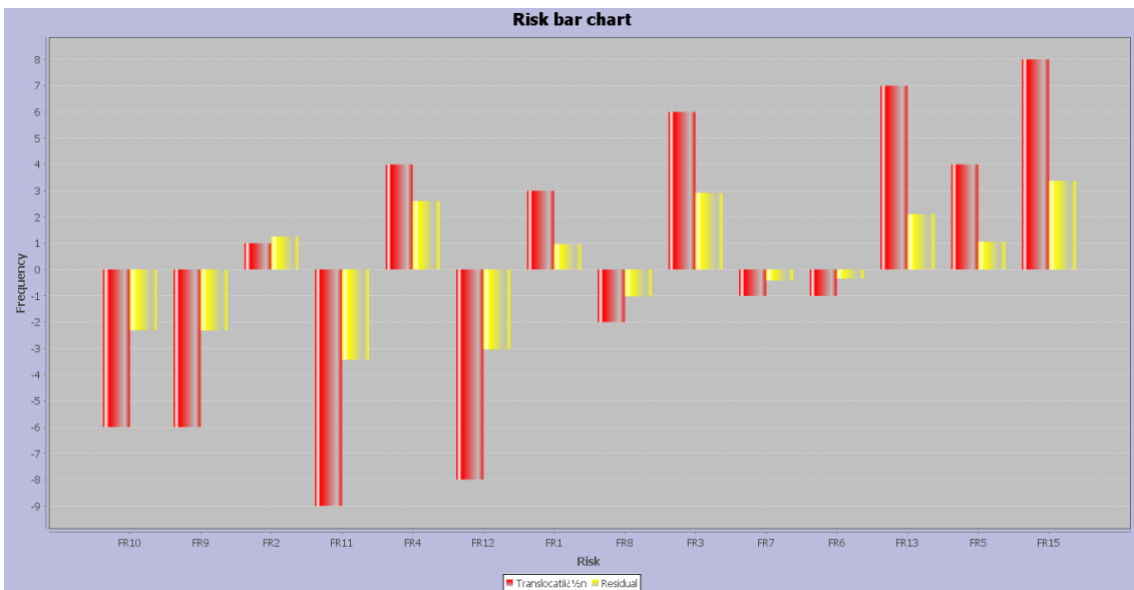


Figura 3.39. Diagramas de barras de Factores de Riesgo del HCULB vs Patrón.

En el caso de Factores de Riesgo hay diferencias entre ambas poligonales, pero menos acusadas que en el caso de Complicaciones,

Las tres diferencias, en negativo, más acusadas las tienen: FR15 (Otras no clasificadas anteriormente), FR4 (Alteraciones metabólicas) y FR3 (Alteraciones respiratorias). Al contrario, mejor que el patrón, están FR12 (Punción arterial no deseada), FR11 (Punción dural en epidural) y FR9 (Técnica incorrecta).

En el diagrama de barras FR15 (Otras no clasificadas anteriormente), FR13 (Problemas previos en la intubación) y FR3 (Alteraciones respiratorias) destacan tanto en translocaciones como en contrastes. Todo lo contrario ocurre con FR12 (Punción arterial no deseada) y con FR11 (Punción dural en epidural) (Figuras 3.38 y 3.39).

RISK FACTOR	INJURY
PHASE 1	PHASE 1
FR15 FR3 FR4 FR13	C10 C2
PHASE 2	PHASE 2
FR5 FR1	C16
PHASE 3	PHASE 3
FR2 FR6 FR7	C3 C8 C15 C11 C4 C9
PHASE 4	PHASE 4
FR8	C7 C5 C13
PHASE 5	PHASE 5
FR10 FR9 FR12 FR11	C6 C12
	PHASE 6
	C19 C14 C1 C17

Figura 3.40. FASES del contraste o *Planning Report* del HCULB vs Patrón.

En “*Planning Report*”, en primera fase aparecen dos que son: C10 (Metabólicas) y C2 (Edema de glotis). En segunda fase solo aparece una, C16 (Hiperalgnesia postoperatoria). Los Factores de Riesgo con necesidad de actuación prioritaria, son cuatro, las tres primeras: FR15 (Otras no clasificadas anteriormente), FR3 (Alteraciones respiratorias) y FR4 (Alteraciones metabólicas) (Figura 3.40).

La de mayor frecuencia de aparición (tabla 3.2) es C16 (Hiperalgnesia postoperatoria) que es la tercera en necesidad de mejora. En cambio, en este hospital, la primera en “*Planning Report*” es C10 (Complicaciones Metabólicas) que es el segundo en frecuencia. Como Factor de Riesgo más frecuente (tabla 3.1) aparece FR4 (Alteraciones metabólicas) que es el tercero en necesidad de mejora según el “*Planning Report*”. Los dos primeros de fase 1 son FR15 (Otras no clasificadas anteriormente) el séptimo en frecuencia y FR3 (Alteraciones respiratorias) en cuarto puesto. De nuevo se pone de manifiesto la importancia del contraste estadístico.

Todos estos resultados son congruentes porque el Hospital Clínico Universitario Lozano Blesa tiene principalmente características de hospital General, es cierto que incluye también incidentes de pacientes procedentes de Traumatología y de Materno-infantil, pero son pocos y no se han eliminado.

El Hospital Universitario Miguel Servet, en cambio, tiene cinco secciones: Traumatología, Materno-infantil, General, Dolor y Reanimación. En ANESTHSOM (Rubio García et al., 2014) se consideraron las cinco, pero en este trabajo y debido a las particularidades de estas dos últimas, se ha considerado que deberán ser estudiadas por separado.

Por todo esto, es comprensible que los resultados de ambos hospitales sean tan dispares, el Hospital Clínico tiene más similitudes con el bloque General porque es más bien un hospital general, ya que a pesar de tener sección de Traumatología y Materno-infantil no independientes, éstas están muy diluidas.

Presentamos a continuación, en las tablas 3.5 y 3.6, un resumen de todo lo comentado anteriormente. Se destaca la más importante en el contraste, "*Planning Report*", y la más importante en frecuencia de aparición.

ANÁLISIS DE LOS PERFILES DE COMPLICACIONES

	Prioridades en planificación	Descriptiva Mayor frecuencia
GENERAL		
vs HUMS	Metabólicas (C10), Arritmias (C3)	Hiperalgiesia post. (C16)
vs Patrón	Hematoma (C19) Neumonía (C9)	Arritmias (C3) Hematoma (C19)
MATERNO INFANTIL		
vs HUMS	Cefalea (C12) Shock (C6)	Rotura dientes (C1) Shock (C6)
vs Patrón	Cefalea (C12) Shock (C6)	Edema de glotis (C2)
TRAUMATOLOGIA		
vs HUMS	Coma (C14) Neumotórax (C17)	Les. Nerv. (C13) Neumotórax (C17)
vs Patrón	Coma (C14) Neumotórax (C17)	Les. Nerv. (C13) Neumotórax (C17)
HOSPITAL CLINICO UNIVERSITARIO LOZANO BLESIA		
vs Patrón	Metabólicas (C10) Edema de glotis (C2)	Hiperalgiesia post. (C16) Hiperalgiesia post. (C16)
HOSPITAL CLINICO UNIVERSITARIO MIGUEL SERVET		
vs Patrón	Coma (C14), Lesión Nerviosa (C13)	Tromboembolismo pulmonar (C5) Shock (C6)

Tabla 3.5. Resumen de Complicaciones.

ANÁLISIS DE LOS PERFILES DE LOS FACTORES DE RIESGO

	Prioridades en planificación		Descriptiva Mayor frecuencia
GENERAL			
vs HUMS	Tec incorrecta (FR9)	Pro prev intub (FR13)	Técnica incorrecta (FR9)
vs Patrón	Tec incorrecta (FR9)	Pro prev intub (FR13)	Alt metab (FR4) Alt resp (FR3)
MATERNO INFANTIL			
vs HUMS	Urgencia (FR10)	Pun du epi (FR11)	Son dos: FR10 y FR11
vs Patrón	Pun du epi (FR11)	Urgencia (FR10)	Alt neur (FR6).
TRAUMATOLOGIA			
vs HUMS	Pun. Ar. no d. (FR12)	Pun. du. epi.(FR11)	Inexperiencia (FR8)
vs Patrón	Pun. Ar. no d. (FR12)	Pun. du. epi.(FR11)	Inexperiencia (FR8) Pun. dural en epi.(FR11)
HOSPITAL CLINICO UNIVERSITARIO LOZANO BLESA			
vs Patrón	Otras no clas. (FR15)	Alt. Respirat. (FR3)	Altera. metabólicas (FR4)
HOSPITAL UNIVERSITARIO MIGUEL SERVET			
vs Patrón	Pun. Ar. no d. (FR12)	Inexperiencia (FR8)	Pun du epi (FR11) Situación urgente (F10)

Tabla 3.6. Resumen de Factores de Riesgo.

CONCLUSIONES

CONCLUSIONES

CONCLUSIONES

1.- No se han encontrado diferencias importantes entre los resultados obtenidos con el programa BIOIN ANESTHSOM y con el trabajo previo ANESTHSOM.

2.- Con este nuevo patrón, los Factores de Riesgo que aparecen más frecuentemente son: Urgencia (FR10), Técnica incorrecta (FR9) y Alteraciones cardio-circulatorias (FR2). Y las Complicaciones más frecuentes son las siguientes: Shock (C6), Hiperalgnesia postoperatoria (C16) y Rotura de diente/s (C1).

3.- En el bloque **General** del HUMS, los Factores de Riesgo que requieren actuación prioritaria son: Técnica incorrecta (FR9), Problemas previos a la intubación (FR13) y Alteraciones respiratorias (FR3). En el caso de las Complicaciones son: Hematoma (C19), Neumonía (C9) y Arritmias (C3).

4.- En la sección **Materno-Infantil**, los Factores de Riesgo que necesitan atenderse preferentemente son: Punción dural en epidural (FR11), Urgencia (FR10) y Alteraciones neurológicas (FR6). Las Complicaciones que necesitan mayor atención son: Cefalea (C12), Shock (C6) y Edema de glotis (C2).

5.- En el bloque de **Traumatología**, los Factores de Riesgo que están en peor situación que en el patrón son: Punción Arterial no deseada (FR12), Punción dural en epidural (FR11) e Inexperiencia (FR8). Cuando los referimos a las Complicaciones estas son: Coma (C14), Neumotórax (C17) y Lesión Nerviosa (C13).

6.- En el **Hospital Universitario Miguel Servet** los Factores de Riesgo que necesitan actuación prioritaria son: Punción Arterial no deseada (FR12), Inexperiencia (FR8) y Punción dural en epidural (FR11). Las Complicaciones que se encuentran en la misma situación son: Coma (C14), Lesión Nerviosa (C13) y Tromboembolismo pulmonar (C5).

7.- En el **Hospital Clínico Universitario Lozano Blesa** los Factores de Riesgo en peor situación con respecto al patrón son: Otras no clasificadas anteriormente (FR15), Alteraciones Respiratorias (FR3) y Alteraciones metabólicas (FR4). Si nos referimos a las Complicaciones que están peor que

CONCLUSIONES

en el patrón encontramos: Metabólicas (C10), Edema de glotis (C2) e Hiperalgnesia postoperatoria (C16).

8.- Hemos comprobado que BIOIN ANESTHOM facilita el manejo estadístico de los datos comparado con ANESTHOM y que podrá mejorar el uso los métodos empleados actualmente para la recogida de incidentes críticos en Anestesiología y permitirá ser utilizada como herramienta de control de calidad.

Entre las **limitaciones** de este trabajo resaltamos que los datos recogidos mediante la revisión de Historias Clínicas, son solo de 2 hospitales de Zaragoza el Hospital Universitario Miguel Servet y el Hospital Clínico Universitario Lozano Blesa. Con esto se quiere explicar que este estudio no es representativo ni siquiera de Aragón, porque para obtener un patrón que represente a toda la Comunidad Autónoma, sería necesario tomar datos de todos los hospitales de la misma, incluidos aquellos de pequeño tamaño.

Estas limitaciones nos llevan a insistir en la necesidad de aplicar BIOIN ANESTHSOM a una base de datos lo más potente posible, preferiblemente de ámbito nacional. Un ejemplo ideal podría ser aplicarlo la base de datos de SENSAR. De esta forma sus resultados serían mucho más representativos de la realidad hospitalaria nacional.

Resaltamos como **fortaleza** de BIOIN ANESTHSOM el hecho de que facilita el manejo estadístico de los datos.

BIBLIOGRAFÍA

BIBLIOGRAFÍA

1. American Chemical Society. International Historic Chemical Landmarks. *Discovery of Oxygen by Joseph Priestley*. Washington: American Chemical Society. Recuperado el [21 feb 2016]. De [\[http://www.acs.org/content/acs/en/education/whatischemistry/landmarks/joseph_priestleyoxygen.html\]](http://www.acs.org/content/acs/en/education/whatischemistry/landmarks/joseph_priestleyoxygen.html)
2. Aranaz, JM., et al. (2008). *ENEAS work group. Incidence of adverse events related to health care in Spain: results of the Spanish National Study of Adverse Events*. J Epidemiol Community Health,62(12), 1022-1029.
3. Aranaz, JM., et al. (2009). ENEAS work group. Impact and preventability of adverse events in Spanish public hospitals: results of the Spanish National Study of Adverse Events (ENEAS). Int J Qual Health Care, 21(6), 408-414
4. Aranaz, JM., et al. (2011). ENEAS Working Group. What makes hospitalized patients more vulnerable and increases their risk of experiencing an adverse event?. Int J Qual Health Care, 23(6), 705-712.
5. Aranaz, J.M., et al. (2011a). Diseño del estudio IBEAS: prevalencia de efectos adversos en hospitales de Latinoamérica. Rev Calid Asist. 26(3), 194-200.
6. Aranaz, J.M., et al. (2011b). Prevalence of adverse events in the hospitals of five Latin American countries: results of the 'Iberoamerican study of adverse events' (IBEAS). BMJ Qual Saf,20(12), 1043-1051.
7. Aranaz, J.M. Aportaciones de la medicina preventiva y salud pública a la seguridad del paciente. An. R. Acad. Med. Comunitat Valenciana. 2011;12:1-18. (2011c)
8. Arbous, M.S., et al. (2001). *Mortality associated with anaesthesia: a qualitative analysis to identify risk factors*. Anaesthesia, 56, 1141-1153
9. Arnal, D., Gómez-Arnau, J.I., Bartolomé, A., Puebla, G., Santa Úrsula, J.A., Marzal, J.A. (2011). *Cómo aprender de los errores: Tres incidentes del Sistema Español de Notificación en Seguridad en Anestesia y Reanimación (SENSAR)*. Rev Esp Anesthesiol Reanim.58(Supl. 3), S22-S27

10. Astigarraga, E. (2000). *El Método Delphi*. Universidad de Deusto. ESTE. San Sebastián.
11. Avram, M.J., Shanks, C.A., Dykes, M.H., Ronai, A.K., Stiers, W.M. (1985). *Statistical methods in anesthesia articles*. *Anaesthesia & Analgesia*, 64, 607-611
12. Barash P. (2006). *Clinical Anaesthesia*. 5th Edition. Lippincott Williams s.l.
13. Bona Gracia, C., Rubio García, B., García-Foncillas, R. (2017). Programa BIOIN ANESTHSOM. Para el control de la accidentalidad en Anestesiología. Saarbrücken, Alemania: Ed Editorial Académica Española.
14. Bona Gracia, C., Rubio García, B., García-Foncillas, R. (2018). Aplicación de modelos asociativos Causa-Consecuencia de accidentalidad laboral en el estudio de la seguridad en anestesia. *Rev Esp Anesthesiol Reanim*, 65, 173-174
15. Bona Gracia, C., Torner Pelay, P., García Foncillas, R., Rubio García, B. (2018 a). BIOIN ANESTHSOM. N° Solicitud: Z-134-18. N° Asiento Registral: 10/2018/260. España.
16. Carretón, A. *Trepanación o la cirugía del cráneo en la prehistoria*. *Arqueoblog*. Recuperado el [18 de Feb de 2016] de [<http://arqueoblog.com/trepanacion/>].
17. Conte, J.C. (2004). *Teoría del accidente en poblaciones laborales. Bases matemáticas*. (Tesis doctoral). Universidad de Zaragoza, Zaragoza.
18. Conte, J.C., Rubio, E., Domínguez, A.I., García, A.I. (2004 a). Análisis de la problemática preventiva para el control de la accidentabilidad laboral. *Rev Arch Fac Med*, 44(2), 113-118
19. Conte, J.C., Domínguez, A.I., García, A.I., Rubio García, B. (2004b). Interpretaciones fenomenológicas clásicas de la accidentabilidad laboral. Planteamiento de una orientación alternativa. *Rev. Arch. Fac. Med*, 44 (2), 119-124.
20. Conte, J.C., Rubio, E., García, A.I. (2005). *Teoría del accidente en poblaciones laborales. Bases matemáticas, aplicadas a la Situación RIESGO-LESION*. N° Solicitud: B-3845-05. N° Asiento Registral: 02/2005/5668. España.

21. Conte, J.C., Rubio, E. (2005a). *Estudio de la relación riesgo-lesión mediante análisis factorial*. Real Acad. Ciencias de Zaragoza, 60, 65-85.
22. Conte, J.C., Rubio, E. (2006). ACSOM y ProtoACSOM como indicadores centinelas. N° Solicitud: B-2337-06. N° Asiento Registral: 02/2006/4314. España.
23. Conte, J.C., Rubio, E., García, A.I., Domínguez, A.I. (2006 a). Metodología para la Previsión del Accidente en Poblaciones Laborales. Aplicación a Microempresas. *Medicina y Seguridad en el trabajo*, 52 (203), 27-37.
24. Conte, J.C., Rubio, E., García, A.I., Domínguez, A.I., Bascuas, J. (2006 b). *Metodología para la Previsión del Accidente en Poblaciones Laborales. Aplicación a las pequeñas y grandes empresas*. *Medicina y Seguridad en el trabajo*, 52 (203), 39-49.
25. Conte, J.C., Marcos, G., García Felipe, A.I., Rubio, E. (2007). *Análisis del problema empírico de identificación del riesgo*. *Cuad Bioest Apl Inform*, 17, 12-25.
26. Conte, J.C., García Felipe, A.I., Rubio García, B. (2007 a). *Teoría del Accidente en Poblaciones Laborales*. *Cuad Bioest Apl Inform*, 17, 26-38.
27. Conte, J.C., Cano, F., García, A.I., Molina, A., Rubio Calvo, E. (2008). *Interpretación de las relaciones intragrupalas de Riesgos y de lesiones mediante análisis cluster jerarquizado*. *Revista de Matemática: Teoría y Aplicaciones*, 15(2), 175-186.
28. Conte, J.C., Rubio, E., García, A.I., Cano, F. (2011). *Occupational accidents model based on risk-injury affinity groups*. *Safety Sci*, 49(2), 306-314.
29. Conte, J.C., Rubio, E., García, A.I., Cano, F. (2011 a). *Correspondence model of occupational accidents*. *Anais da Academia Brasileira de Ciências*, 83(3), 1131-1146.
30. De Vries, E.N., Ramrattan, M.A., Smorenburg, S.M., Gouma, D.J., Boermeester, M.A. (2008). *The incidence and nature of in-hospital adverse events: a systematic review*. *Qual Saf Health Care*, 17, 216-223.

31. Declaración de Helsinki. (2010). Rev. Esp. Anestesiol. Reanim, 57, 594-595.
32. Dunn, H.L (1929). *Application of statistical methods in Physiology*. Physiological Reviews, 9, 275-398.
33. ENEAS. Estudio Nacional sobre los Efectos Adversos ligados a la Hospitalización. (2005). Ministerio de sanidad y consumo.
34. Fernández García, J., Martínez Terrer, T., Rubio, B. (2002). ¿Queremos saber lo que hacemos? Propuesta de una metodología para el control de calidad. VI Reunión Nacional. Asociación Española de coloproctología. Sección de Coloproctología de la AEC. Pamplona.
35. Forero, H. (2003). Fundamentos sociológicos de la medicina primitiva. Bogotá, Colombia: Ed. Academia Nacional de Medicina.
36. García Felipe, A., Álvarez Zarate, J.M., Alcalde Lapiedra, V., Rubio García, B., Bascuas Hernandez, J. (2004). *La Monitorización Epidemiológica en Vigilancia de la Salud mediante el Residual de Incidentabilidad Estandarizado (RIS)*. Revista MAPFRE Seguridad, 24(93), 33-42.
37. García Felipe, A.I., Conte, J.C., Rubio Garcia, B. (2007). *Propuesta de un proceso automático para planificación en prevención de riesgos laborales*. Cuad. Bio. Apl. Inf, 17, 39-49
38. García Felipe, A.I., Conte, J.C., Rubio, E., Pérez Prados, A. (2009). *Workaccident. ACSOM a new orientation for automatic risk management*. Ann. Sist. Sanit. Navar, 32(1), 23-34
39. García-Foncillas, R. (2015). *Creación del BIOIN ACSOM adaptado a la Declaración Electrónica de Trabajadores Accidentados*. (Tesis doctoral). Universidad de Zaragoza, Zaragoza.
40. García-Foncillas, R., García Felipe, A.I., Bona Gracia, M.C., Rubio García, B. (2016). *Creación del BIOIN Delt@. Aplicación en Aragón y Navarra*. Saarbrücken, Alemania: Ed Editorial Académica Española.

41. Gimeno, Y. (2012). *Aplicación del modelo ACSOM al sector agrario aragonés* (Tesina final Master Prevención de Riesgos laborales, especialidad seguridad en el trabajo) Universidad Complutense, Madrid.
42. Greene, N.M. (1992). *Anesthesiology Journals. Anaesthesia & Analgesia*, 74, 116-120
43. Gómez-Arnau Díaz-Cañabate, J.I., Bartolomé Ruibal, A., Santa-Ursula Tolosa, J.A., González Arévalo, A. y García del Valle Manzano, S. (2006). *Incident reporting systems and patient safety in anesthesia*. *Rev Esp Anesthesiol Reanim*, 53 (8), 488-499.
44. Gómez-Arnau, J.I., Bartolomé, A., García Del Valle, S., González, A. (2011). *Mortalidad y morbilidad. Riesgo en Anestesia*. *Rev Esp Anesthesiol Reanim*, 58(Supl. 3), S3-S9.
45. Hair, J.F., Anderson, R.E., Tatham, R.L., Black, W.C. (1999). *Análisis multivariante*. Madrid: Prentice Hall Ibérica.
46. Instituto Nacional de estadística. "*Explica. Te ayudamos con la estadística*". Recuperado el [29 de marzo de 2016] de [www.ine.es.]
47. Institute of Medicine. (1999). *To err is human: building a safer health system*. National Academy Press, p. 8.
48. JAVA. Oracle Corporation. Recuperado el [5 de marzo de 2017] de [https://java.com/es/download/faq/java_webstart.xml]
49. Kawashima, Y., Takahashi, S., Suzuki, M., Morita, K., Irita, K., Iwao, Y. et al. (2003). *Anesthesia-related mortality and morbidity over a 5-year period in 2,363,038 patients in Japan*. *Acta Anaesthesiol Scand*, 47, 809-817.
50. Kupersztynch-Hagege, E., Duracher-Gout, C., Ortego, R., Carli, P., Orliaguet, G. (2017). *Critical incidents in a French department of paediatric anaesthesia*. *Anaesth Crit Care Pain Med*, 36(2):103-107
51. Larson, M.D. (2010). *History of Anesthetic Practice*. En: Miller R, Miller's Anaesthesia.(pp. 3-52). Vol 1. 7ª ed. San Francisco, California: Churchill Livingstone.

52. Lienhart, A., Auroy, Y., Pequignot, F., Benhamou, D., Warszawski, J., Bovet, M. et al. (2006). *Survey of anesthesia-related mortality in France*. *Anesthesiology*, 105, 1087-1097.
53. Maaløe, R., la Cour, M., Hansen, A., Hansen, E.G., Hansen, M., Spangsberg, N.L., Landsfeldt, U.S., Odorico, J., Olsen, K.S., Møller, J.T., Pedersen, T. (2006). *Scrutinizing incident reporting in anaesthesia: why is an incident perceived as critical?* *Acta Anaesthesiol Scand*, 50(8). 1005-1013
54. Marín Martínez, F., Sánchez Meca, J., López López, J. (2009). *El metaanálisis en el ámbito de las Ciencias de la Salud: una metodología imprescindible para la eficiente acumulación del conocimiento*. *Fisioterapia*, 31(3), 107–114
55. Martina, J.F. *Historia de la anestesia*. Recuperado el [31 de marzo de 2016.] de [<http://www.csen.com/historia.pdf>]
56. Marrugat, J., Sanz, F., Porta Serra, M., Sancho, J. (1989). *La influencia de la informática en la investigación clínica y epidemiológica*. *Med Clin*, 92(19), 742-748.
57. Ministerio de Educación y Ciencia. (1996). *Guía de formación de especialistas en Anestesiología y Reanimación*. 47-55.
58. Moreno Altamirano, L. (2013). *Epidemiología clínica*. 3ªEd. México: McGrawHill.
59. Ortiz Sagristá, J.C. (2004). *Uso de técnicas de Bioestadística en revistas de Anestesiología*. (Tesis Doctoral). Universidad de Barcelona, Barcelona.
60. Patiño, W. (1997) *Fundamentos de cirugía*. Anestesiología. 2ºEd. Colombia: Corporación para investigaciones biológicas.
61. RAE. *Diccionario de la lengua española*. Recuperado el [28 de marzo de 2016.] de [www.dle.rae.es.]
62. Real, J.E. (2001). *Escalamiento multidimensional*. Madrid: La muralla S.A.
63. Rees, P., Edwards, A., Powell, C., Hibbert, P., Williams, H., Makeham, M., Carter, B., Luff, D., Parry, G., Avery, A., Sheikh, A., Donaldson, L., Carson-Stevens, A. (2017). *Patient Safety Incidents Involving Sick Children in Primary*

- Care in England and Wales: A Mixed Methods Analysis*. PLoS Med,14(1). doi: 10.1371/journal.pmed.1002217.
64. Rodríguez Zazo, A. (1990). *Estudio de la evaluación de Factores de Riesgo en Anestesiología y Reanimación*. (Tesis doctoral). Universidad de Zaragoza, Zaragoza.
65. Rubio, E., Conte, J.C., García, A.I. (2007). *Minería de datos, Salud Laboral y Sistemas Inteligentes*. Cuad. Bioestad. Apl. Inform, 17, 2-11.
66. Rubio, E., Torner, P., Conte, J.C., García, A.I. (2008). *Programa BIOIN Omega I. Sistema traductor de base de datos en código Delta a códigos OIT*. Registro de propiedad intelectual (objeto y clase: programa de ordenador). Nº asiento registral: 10/2008/305. España.
67. Rubio, E., Torner, P., García, A.I., Conte, J.C. (2008 a). *Programa BIOIN Omega II. Análisis de la variabilidad de perfiles para la planificación preventiva de riesgos y lesiones vulnerables. Método ACSOM*. Registro de propiedad intelectual (objeto y clase: programa de ordenador). Nº registro: 10/2008/306. España.
68. Rubio Calvo, E., García Felipe, A.I., Conte, J.C., García Foncillas, R., Torner, P., Rubio García, E. y Gracia García, E. (2012). *ACSOM. Herramienta para el análisis de la accidental laboral y su programa informático BIOIN*. Saarbrücken, Alemania: Editorial Académica Española.
69. Rubio Calvo, E., García, A.I., Gracia, E., García-Foncillas, R., Rubio, B. Y Rubio E.J. (2015). *Creación de BIOIN Delt@*. Registro de propiedad intelectual. Nº asiento registral: 10/2015/228. España.
70. Rubio García, B. (2004). *Búsqueda de un indicador dinámico para la detección y seguimiento de los riesgos ocupacionales: el RIS. Aplicación del RIS en el Sistema de Vigilancia de la Salud de Opel España*. (Tesina de Licenciatura). Universidad de Zaragoza, Zaragoza.
71. Rubio García, B. (2005). *Estudio del uso de técnicas estadísticas en Anestesiología*. (D.E.A.). Universidad de Zaragoza, Zaragoza.





72. Rubio García, B., Rodríguez, A., Martínez, T., Rubio Calvo, E. (2010). *Análisis comparativo de tres revistas de Anestesiología en cuanto al uso de la estadística y accesibilidad de los artículos originales publicados*. Rev Esp Anestesiol y Reanim, 57(5), 281-287
73. Rubio García, B. (2012). *Búsqueda de un indicador dinámico de análisis de las complicaciones en Anestesiología y Reanimación: ANESTHSOM. Validación empírica en el hospital Universitario Miguel Servet de Zaragoza*. (Tesis doctoral). Universidad de Zaragoza, Zaragoza.
74. Rubio García, B., García, A.I., Díaz, L. (2014). *Análisis de complicaciones en anestesiología y reanimación: ANESTHSOM*. Saarbrücken, Alemania: Ed Editorial Académica Española.
75. Schor, S., Karten, I. (1996). *Statistical evaluation of medical journal manuscripts*. JAMA, 195(13), 1123-1128.
76. Science History Institute. *Humphry Davy*. Philadelphia: Science History Institute; Recuperado el [01 de junio de 2016] de [<https://www.sciencehistory.org/historical-profile/humphry-davy>]
77. Scott, G. (2001). *Strategic Planning for High-Tech Product Development*. Analysis & Strategic Management, 13(3), 343-364.
78. Sistema Español de Notificación en seguridad en Anestesia y Reanimación. SENSAR. Recuperado el [01 de julio de 2018] de [<http://www.sensar.org/>]
79. Trigo Aranda, V. (2004). *Historia y evolución de los lenguajes de programación*. Manual Formativo de ACTA, ISSN 1888-605, 34, 85-95
80. Torner, P., Conte, J.C., García, A., Rubio, E. (2008). *Programa BIOIN: traductor de Bases de Datos Delta a códigos OIT*. N° Solicitud: Z-193-08. Fecha: 08/05/2008. España.
81. Torner, P., García, A., Conte, J.C., Rubio, E. (2008 a). *Programa BIOIN I. Sistema II. Análisis de la variabilidad de perfiles para la Planificación preventiva de riesgos y lesiones vulnerables*. Método ACSOM. N° Solicitud: Z-194-08. Registro de propiedad intelectual N° Asiento Registral: 10/2008/487. España.

82. Torner, P., García, A.I., Conte, J.C., Rubio, E. (2008 b). *BIOIN- Ω -III. Informes de planificación del Método ACSOM*. Registro de propiedad intelectual N° Asiento Registral: 10/2008/487. España.
83. Williams, G.D., Muffly, M.K., Mendoza, J.M., Wixson, N., Leong, K., Claire, R.E. (2017). *Reporting of Perioperative Adverse Events by Pediatric Anesthesiologists at a Tertiary Children's Hospital: Targeted Interventions to Increase the Rate of Reporting*. *Anesth Analg*, 125(5):1515-1523.

BIBLIOGRAFÍA

ANEXO

ANEXO: Código fuente del programa informático BIOIN ANESTHSOM. Material entregado en registro de la propiedad, bajo patente intelectual.

	
GOBIERNO DE ARAGON	MINISTERIO DE CULTURA Y DEPORTE
REGISTRO TERRITORIAL DE LA PROPIEDAD INTELECTUAL	
REGISTRO GENERAL DE LA PROPIEDAD INTELECTUAL	
Según lo dispuesto en la Ley de Propiedad Intelectual (Real Decreto Legislativo 1/1996, de 12 de abril), quedan inscritos en este Registro los derechos de propiedad intelectual en la forma que se determina seguidamente:	
NÚMERO DE ASIENTO REGISTRAL 10 / 2018 / 260	
Título: BIOIN ANESTHSOM	
Objeto de propiedad intelectual: Programa de ordenador	
Clase de obra: Programa de ordenador	
PRIMERA INSCRIPCIÓN	
Autor/es y titular/es originarios de derechos	
• Apellidos y nombre: TORNER PELAY, PABLO	
Nacionalidad: España	D.N.I./N.I.F./Pasaporte: 29135754K
• Apellidos y nombre: GARCÍA-FONCILLAS LÓPEZ, RAFAEL	
Nacionalidad: España	D.N.I./N.I.F./Pasaporte: 29096677K
• Apellidos y nombre: RUBIO GARCÍA, BORJA	
Nacionalidad: España	D.N.I./N.I.F./Pasaporte: 25477198Y
• Apellidos y nombre: BONA GRACIA, MARÍA DEL CARMEN	
Nacionalidad: España	D.N.I./N.I.F./Pasaporte: 76923367M
Datos de la solicitud	
Núm. solicitud: Z-134-18	
Fecha de presentación y efectos: 19/04/2018	Hora: 10:36
En ZARAGOZA, a veintiuno de junio de dos mil dieciocho	
El/La titular del Registro	
	
Firmado: Marta Sevilla Casbas	
	
10/2018/260	

Nº Registro Salida: 164

MARÍA DEL CARMEN BONA GRACIA
VÍA UNIVERSITAS 2 – 8º – 1ª
5009.-ZARAGOZA

En relación con su solicitud de inscripción n.º Z-134-18 presentada en el Registro Territorial de la Propiedad Intelectual de Aragón, referente a los derechos de propiedad intelectual de su obra, le notifico, a los efectos oportunos, que la misma ha obtenido calificación jurídica favorable y que dichos derechos han quedado inscritos en dicho Registro Territorial de la Propiedad Intelectual.

Le adjunto la matriz de inscripción correspondiente.

Zaragoza, 20 de Abril de 2018



REGISTRO TERRITORIAL DE LA PROPIEDAD INTELECTUAL DE ARAGÓN
Biblioteca de Aragón. C/ Doctor Cerrada 22, 3º
50005 ZARAGOZA

Carpeta BIOIN ANESTHSOM

CONTROLADOR

Controlador.class

Controlador_1.class

MODELO

Alm1.class

Alm2.class

Alm3.class

Command.class

Contenedor.class

DiagramaBarras.class

DiagramaBarras 1.class

DibujarBarras.class

DibujarPoligonal.class

Informe.class

Informe 1.class

LISTA

ListaAlm1.class

ListaAlm2.class

PlantillaMetodo.class

Poligonal.class

Poligonal.class

1

VISTA

AcercaDe.class

AcercaDe2.class

Central.class

Central 1.class

Guardar.class

Guardar 1.class

Lienzo.class

 MargenConstrains.class

Operaciones.class

OperacionesDelta.class

OperacionesInformeCompleto.class

OperacionesOIT.class

Planificacion.class

OrgBIOINomegaANESTHSOM

BIOIN ANESTHSOM**CONTROLADOR**

```

package org.anesthsom.anesthsom.controlador;

import java.awt.Color;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.util.Observable;
import org.anesthsom.anesthsom.modelo.Contenedor;
import org.anesthsom.anesthsom.modelo.DibujarBarras;
import org.anesthsom.anesthsom.modelo.DibujarPoligonal;
import org.anesthsom.anesthsom.modelo.Informe;
import org.anesthsom.anesthsom.modelo.Informel;
import org.anesthsom.anesthsom.modelo.PlantillaMetodo;
import org.anesthsom.anesthsom.vista.Lienzo;
import org.anesthsom.anesthsom.vista.Planificacion;

import com.lowagie.text.Document;
import com.lowagie.text.DocumentException;
import com.lowagie.text.PageSize;
import com.lowagie.text.pdf.PdfWriter;

public class Controlador extends Observable {

    private static Controlador instance;
    private static Contenedor empresa;
    private static Contenedor general;

    public static Controlador getInstance() {
        if (instance == null) {
            instance = new Controlador();
        }
        return instance;
    }

    public void DPoligonal() {
        PlantillaMetodo pol = new DibujarPoligonal();
        pol.setEmpresa(empresa);
        pol.setGeneral(general);
        pol.execute();
        setChanged();
        notifyObservers();
    }

    public void DBarras() {
        PlantillaMetodo bar = new DibujarBarras();
        bar.setEmpresa(empresa);
        bar.setGeneral(general);
        bar.execute();
        setChanged();
        notifyObservers();
    }
}

```



```

}

public void Planificacion() {
    int r[] = empresa.getP().getRiesgo();
    int l[] = empresa.getP().getLesiones();
    int l1[] = general.getP().getLesiones();
    int r1[] = general.getP().getRiesgo();
    float x[] = new float[60];
    float y[] = new float[60];
    float z[] = new float[20];
    float w[] = new float[20];
    float p[] = new float[60];
    float q[] = new float[60];
    float m[] = new float[20];
    float n[] = new float[20];
    float resr[] = new float[60];
    float resl[] = new float[20];
    int orden[] = new int[60];
    int orden1[] = new int[20];
    r1 = ponerACero(r1, r);
    l1 = ponerACero(l1, l);
    int suma =0;
    for (int i=0;i<r.length;i++){
        suma=suma+r[i];
    }
    int suma1 =0;
    for (int i=0;i<r1.length;i++){
        suma1=suma1+r1[i];
    }
    int suma2 = 0;
    for (int i=0;i<l.length;i++){
        suma2=suma2+l[i];
    }
    int suma3=0;
    for (int i=0;i<l1.length;i++){
        suma3=suma3+l1[i];
    }
    orden = or(r1,60);
    orden = ordenarVectorOrden(orden,60);
    x = ordenarVectorconOrden(orden, r,60);
    y = ordenarVectorconOrden(orden, r1,60);
    x = calcularVectorPosicion(x,60);
    y = calcularVectorPosicion(y,60);
    orden1 = or(l1,20);
    orden1 = ordenarVectorOrden(orden1,20);
    z = ordenarVectorconOrden(orden1, l,20);
    w = ordenarVectorconOrden(orden1, l1,20);
    z = calcularVectorPosicion(z,20);
    w = calcularVectorPosicion(w,20);
    x = comparar(x, y,60);
    z = comparar(z, w,20);
    p = ordenarVectorconOrden(orden, r,60);
    q = ordenarVectorconOrden(orden, r1,60);
    resr = cresidual(p, q, suma, suma1,60);
    m = ordenarVectorconOrden(orden1, l,20);
}

```

ANEXO

```
n = ordenarVectorconOrden(orden1, l1,20);
resl = cresidual(m, n, suma2, suma3,20);
int fase1r[] = Fase1(resr, x, r1, orden);
int fase1l[] = Fase1(resl, z, l1, orden1);
int fase2r[] = Fase2(resr, x, r1, orden);
int fase2l[] = Fase2(resl, z, l1, orden1);
int fase3r[] = Fase3(resr, x, r1, orden);
int fase3l[] = Fase3(resl, z, l1, orden1);
int fase4r[] = Fase4(resr, x, r1, orden);
int fase4l[] = Fase4(resl, z, l1, orden1);
int fase5r[] = Fase5(resr, x, r1, orden);
int fase5l[] = Fase5(resl, z, l1, orden1);
int fase6r[] = Fase6(resr, x, r1, orden);
int fase6l[] = Fase6(resl, z, l1, orden1);
int fase7r[] = Fase7(resr, x, r1, orden);
int fase7l[] = Fase7(resl, z, l1, orden1);
int fase8r[] = Fase8(resr, x, r1, orden);
int fase8l[] = Fase8(resl, z, l1, orden1);
int fase9r[] = Fase9(resr, x, r1, orden);
int fase9l[] = Fase9(resl, z, l1, orden1);
if (fase1r!=null)
fase1r = ordenarr(fase1r, resr, x, orden);
if (fase1l!=null)
fase1l = ordenarr(fase1l, resl, z, orden1);
if (fase2r!=null)
fase2r = ordenarr(fase2r, resr, x, orden);
if (fase2l!=null)
fase2l = ordenarr(fase2l, resl, z, orden1);
System.out.println("Longitud:"+fase3r.length);
if (fase3r!=null)
fase3r = ordenarr(fase3r, resr, x, orden);
if (fase3l!=null)
fase3l = ordenarr(fase3l, resl, z, orden1);
if (fase4r!=null)
fase4r = ordenarr(fase4r, resr, x, orden);
if (fase4l!=null)
fase4l = ordenarr(fase4l, resl, z, orden1);
if (fase5r!=null)
fase5r = ordenarr(fase5r, resr, x, orden);
if (fase5l!=null)
fase5l = ordenarr(fase5l, resl, z, orden1);
if (fase6r!=null)
fase6r = ordenarr(fase6r, resr, x, orden);
if (fase6l!=null)
fase6l = ordenarr(fase6l, resl, z, orden1);
if (fase7r!=null)
fase7r = ordenarr(fase7r, resr, x, orden);
if (fase7l!=null)
fase7l = ordenarr(fase7l, resl, z, orden1);
if (fase8r!=null)
fase8r = ordenarr(fase8r, resr, x, orden);
if (fase8l!=null)
fase8l = ordenarr(fase8l, resl, z, orden1);
if (fase9r!=null)
fase9r = ordenarr(fase9r, resr, x, orden);
if (fase9l!=null)
```

```

fase9l = ordenarr(fase9l, resl, z, ordenl);
Lienzo a = new Lienzo();
a.setBackground(Color.black);
a.setFase1l(fase1l);
a.setFase1r(fase1r);
a.setFase2l(fase2l);
a.setFase2r(fase2r);
a.setFase3l(fase3l);
a.setFase3r(fase3r);
a.setFase4l(fase4l);
a.setFase4r(fase4r);
a.setFase5l(fase5l);
a.setFase5r(fase5r);
a.setFase6l(fase6l);
a.setFase6r(fase6r);
a.setFase7l(fase7l);
a.setFase7r(fase7r);
a.setFase8l(fase8l);
a.setFase8r(fase8r);
a.setFase9l(fase9l);
a.setFase9r(fase9r);
Planificacion d = new Planificacion(fase1r, fase2r,
fase3r, fase4r, fase5r, fase6r, fase7r, fase8r, fase9r, fase1l,
fase2l, fase3l, fase4l, fase5l, fase6l, fase7l, fase8l, fase9l);
d.setSize(1000, 500);
//d.setTitle("Planificacion riesgos desviac");
d.add(a);
d.setVisible(true);
}

private int[] Fase9(float[] resr, float[] x, int[] r, int[]
orden) {
    int d = Cuantos(resr, x, 9, r, orden);
    int a[] = new int[d];
    int b = 0;
    for (int i = 0; i < r.length; i++) {
        if (((resr[i] < -1.96) && (x[i] < -1)) &&
(r[orden[i]] != 0)) {
            a[b] = i + 1;
            b++;
        }
    }
    return a;
}

private int[] Fase8(float[] resr, float[] x, int r[], int[]
orden) {
    int d = Cuantos(resr, x, 8, r, orden);
    int a[] = new int[d];
    int b = 0;
    for (int i = 0; i < r.length; i++) {
        if (((resr[i] < -1.96) && ((x[i] == 0) || (x[i] == -
1) || (x[i] == 1))) && (r[orden[i]] != 0)) {
            a[b] = i + 1;
            b++;
        }
    }
}

```

```

    }
    return a;
}

private int[] Fase7(float[] resr, float[] x, int r[], int[]
orden) {
    int d = Cuantos(resr, x, 7, r, orden);
    int a[] = new int[d];
    int b = 0;
    for (int i = 0; i < r.length; i++) {
        if (((resr[i] < -1.96) && (x[i] > 1)) &&
(r[orden[i]] != 0)) {
            a[b] = i + 1;
            b++;
        }
    }
    return a;
}

private int[] Fase6(float[] resr, float[] x, int[] r, int[]
orden) {
    int d = Cuantos(resr, x, 6, r, orden);
    int a[] = new int[d];
    int b = 0;
    for (int i = 0; i < r.length; i++) {
        if (((resr[i] >= -1.96) && (resr[i] <= 1.96) &&
(x[i] < -1)) && (r[orden[i]] != 0)) {
            a[b] = i + 1;
            b++;
        }
    }
    return a;
}

private int[] Fase5(float[] resr, float[] x, int[] r, int[]
orden) {
    int d = Cuantos(resr, x, 5, r, orden);
    int a[] = new int[d];
    int b = 0;
    for (int i = 0; i < r.length; i++) {
        if (((resr[i] >= -1.96) && (resr[i] <= 1.96) &&
((x[i] == 1) || (x[i] == -1) || (x[i] == 0))) && (r[orden[i]] != 0))
{
            a[b] = i + 1;
            b++;
        }
    }
    return a;
}

private int[] Fase4(float[] resr, float[] x, int[] r, int[]
orden) {
    int d = Cuantos(resr, x, 4, r, orden);
    int a[] = new int[d];
    int b = 0;

```

```

        for (int i = 0; i < r.length; i++) {
            if (((resr[i] >= -1.96) && (resr[i] <= 1.96) &&
(x[i] > 1)) && (r[orden[i]] != 0)) {
                a[b] = i + 1;
                b++;
            }
        }
        return a;
    }

    private int[] Fase3(float[] resr, float[] x, int[] r, int[]
orden) {
        int d = Cuantos(resr, x, 3, r, orden);
        int a[] = new int[d];
        int b = 0;
        for (int i = 0; i < r.length; i++) {
            if (((resr[i] > 1.96) && (x[i] < -1)) &&
(r[orden[i]] != 0)) {
                a[b] = i + 1;
                b++;
            }
        }
        return a;
    }

    private int[] Fase2(float[] resr, float[] x, int[] r, int[]
orden) {
        int d = Cuantos(resr, x, 2, r, orden);
        int a[] = new int[d];
        int b = 0;
        for (int i = 0; i < r.length; i++) {
            if (((resr[i] > 1.96) && ((x[i] == 1) || (x[i] == -
1) || (x[i] == 0)) && (i != 0)) && (r[orden[i]] != 0)) {
                a[b] = i + 1;
                b++;
            }
        }
        return a;
    }

    private int[] Fase1(float[] resr, float[] x, int r[], int[]
orden) {
        int d = Cuantos(resr, x, 1, r, orden);
        int a[] = new int[d];
        int b = 0;
        for (int i = 0; i < r.length; i++) {
            if (((resr[i] > 1.96) && (x[i] > 1)) || ((resr[i] >
1.96) && (x[i] == 0) && (i == 0))) && (r[orden[i]] != 0)) {
                a[b] = i + 1;
                b++;
            }
        }
        return a;
    }
}

```

ANEXO

```

private int[] ordenarr(int[] fase1r, float[] resr, float[]
x, int orden[]) {
    int a[] = new int[fase1r.length];
    float b = 0;
    int c = 0;
    float d = 0;
    if (fase1r.length>0){
    for (int i = 0; i < fase1r.length; i++) {
        b = -1000000000;
        c = -1000000000;
        d = -1000000000;
        for (int j = 0; j < fase1r.length; j++) {
            if(fase1r[j]>0){
                if (resr[fase1r[j] - 1] > b) {
                    int y = fase1r[j];
                    y = y - 1;
                    b = resr[y];
                    //b=resr[fase1r[j]-1];
                    c = j;
                    d = x[fase1r[j] - 1];
                } else if ((resr[fase1r[j] - 1] == b) &&
(x[fase1r[j] - 1] > d)) {
                    b = resr[fase1r[j] - 1];
                    c = j;
                    d = x[fase1r[j] - 1];
                }
            }
        }
        if(c != -1000000000){
            x[fase1r[c] - 1] = -1000000000;
            resr[fase1r[c] - 1] = -1000000000;
            a[i] = orden[fase1r[c] - 1] + 1;
        }
    }
    return a;
}

private int Cuantos(float[] resr, float[] x, int a, int[] r,
int[] orden) {
    int d = 0;
    int tope=r.length;
    for (int i = 0; i < r.length; i++) {
        switch (a) {
            case 1:
                if (((resr[i] > 1.96) && (x[i] > 1)) ||
((resr[i] > 1.96) && (x[i] == 0) && (i == 0))) && (r[orden[i]]
!= 0)) {
                    d++;
                }
                ;
                break;
            case 2:
                if (((resr[i] > 1.96) && ((x[i] == 1)|| (x[i]
== -1)|| (x[i] == 0)) && (i != 0)) && (r[orden[i]] != 0)) {
                    d++;
                }
        }
    }
}

```

```

    }
    ;
    break;
case 3:
    if (((resr[i] > 1.96) && (x[i] < -1)) &&
(r[orden[i]] != 0)) {
        d++;
    }
    ;
    break;
case 4:
    if (((resr[i] >= -1.96) && (resr[i] <= 1.96)
&& (x[i] > 1)) && (r[orden[i]] != 0)) {
        d++;
    }
    ;
    break;
case 5:
    if (((resr[i] >= -1.96) && (resr[i] <= 1.96)
&& ((x[i] == 1)|| (x[i] == -1)|| (x[i] == 0)) && (r[orden[i]] !=
0)) {
        d++;
    }
    ;
    break;
case 6:
    if (((resr[i] >= -1.96) && (resr[i] <= 1.96)
&& (x[i] < -1)) && (r[orden[i]] != 0)) {
        d++;
    }
    ;
    break;
case 7:
    if (((resr[i] < -1.96) && (x[i] > 1)) &&
(r[orden[i]] != 0)) {
        d++;
    }
    ;
    break;
case 8:
    if (((resr[i] < -1.96) && ((x[i] == -
1)|| (x[i] == 1)|| (x[i] == 0)) && (i != tope)) && (r[orden[i]] !=
0)) {
        d++;
    }
    ;
    break;
case 9:
    if (((resr[i] < -1.96) && (x[i] < -1)) ||
((resr[i] < -1.96) && (x[i] == 0) && (i == tope-1))) &&
(r[orden[i]] != 0)) {
        d++;
    }
    ;
    break;
}

```

```

    }
    return d;
}

/**
 * Calcula un vector con el orden referente a otro vector
 * dado.
 *
 * @param r es un vector de reales de tipo float
 * @return devuelve un vector con el el orden segun la
 * posicion que se
 * ocupa.
 */
private static float[] calcularVectorPosicion(float[] r,int
a) {
    float x[] = new float[a];
    for (int i = 0; i < (r.length); i++) {
        int contador = 0;
        for (int j = 0; j < (r.length); j++) {
            if ((r[i] < r[j]) || (r[i] == r[j] && (i > j)))
{
                contador++;
            }
        }
        x[i] = contador;
    }
    return x;
}

/**
 * @param contador
 * @param cont
 * @return
 */
/**
 * En todos los sitios donde aparece un cero en un vector de
 * enteros r[] le
 * introduce un cero en r1[]
 *
 * @param r1 Vector a modificar
 * @param r vector que tiene de referencia
 * @return devuelve el mapa r1 con los ceros que tiene r
 */
private int[] ponerACero(int[] r1, int[] r) {
    for (int i = 0; i < r1.length; i++) {
        if (r[i] == 0) {
            r1[i] = 0;
        }
    }
    return r1;
}

/**
 * Obtener el orden de un vector ordenado de 0 a su
 * longitud máxima, que matriz corresponde a su posicion

```


ANEXO

```

    * @param orden es el vector con las posiciones
correspondiente
    * @return el vector numerico ordenado
    */
private static int[] ordenarVectorOrden(int[] orden,int a) {
    int x[] = new int[a];
    for (int i = 0; i < orden.length; i++) {
        int k = 0;
        while (orden[k] != i) {
            k++;
        }
        x[i] = k;
    }
    return x;
}

/**
 * Ordena un vector r1 respecto de un orden dado orden y lo
devuelve como funcion.
 * @param orden
 * @param r1
 * @return
 */
private static float[] ordenarVectorconOrden(int[] orden,
int[] r1,int a) {
    float v[] = new float[a];

    for (int i = 0; i < r1.length; i++) {
        v[i] = r1[orden[i]];
    }
    return v;
}

/**
 * @param r
 * @return
 */
private static int[] or(int[] r,int a) {
    int x[] = new int[a];
    for (int i = 0; i < (r.length); i++) {
        int contador = 0;
        for (int j = 0; j < (r.length); j++) {
            if (r[i] < r[j]) {
                contador++;
            }
        }
        for (int k = 0; k < i; k++) {
            if (x[k] == contador) {
                contador++;
            }
        }
        x[i] = contador;
    }
    return x;
}
}
```

```

/**
 * @param x
 * @param y
 * @param suma
 * @param suma2
 * @return
 */
private static float[] cresidual(float x[], float y[], float
suma, float suma2,int a) {
    float cres[] = new float[a];
    float ob = 0;
    float s = 0;
    float es = 0;
    float rt = 0;
    float vr = 0;
    float ra = 0;
    for (int i = 0; i < x.length; i++) {
        ob = x[i];
        s = x[i] + y[i];
        es = suma * s / (suma + suma2);
        if (es >= 5) {
            rt = (float) ((ob - es) / (Math.sqrt(es)));
            vr = (1 - (s / (suma + suma2))) * (1 - (suma /
(sumas + suma2)));
            ra = (float) (rt / (Math.sqrt(vr)));
        } else {
            ra = (float) ((Math.sqrt(ob)) + (Math.sqrt(ob +
1)) - (Math.sqrt(4 * es + 1)));
        }
        cres[i] = ra;
    }
    return cres;
}

/**
 * @param y
 * @param z
 * @return
 */
private static float[] comparar(float[] y, float[] z,int a)
{
    float r[] = new float[a];
    for (int i = 0; i < y.length; i++) {
        r[i] = z[i] - y[i];
    }
    return r;
}

public void VisualizarInforme(String a) {
    try {
        Runtime.getRuntime().exec("rundll32
url.dll,FileProtocolHandler " + a);
    } catch (IOException e) {
        // TODO Auto-generated catch block

```

```

        e.printStackTrace();
    }

}

    public void CrearInformes(int[] fase1, String a, int b)
throws FileNotFoundException, DocumentException {
    int r[] = empresa.getP().getRiesgo();
    int r1[] = general.getP().getRiesgo();
    double[] frecp = ovector(r1,60);
    double[] frece = ovector(r,60);
    Document document = new Document(PageSize.A4, 50, 50,
100, 72);
    PdfWriter writer = PdfWriter.getInstance(document, new
FileOutputStream(a));
    document.open();
    writer.setPageEvent(new Informe(fase1, document, frecp,
frece, b));

    document.close();
}

    private double[] ovector(int[] r,int c) {
    double[] v = new double[c];
    int a = 0;
    int b = 0;
    int suma = 0;
    for(int i = 0; i < r.length; i++){
        suma=suma+r[i];
    }

    for (int i = 0; i < v.length; i++) {
        a = (r[i] * 100) / suma;
        b = ((r[i] * 1000) / suma) - (a * 10);
        v[i] = a + (0.1 * b);
    }
    return v;
}

    public void CrearInformel(int[] fase1, String a, int b)
throws FileNotFoundException, DocumentException {
    int r[] = empresa.getP().getLesiones();
    int r1[] = general.getP().getLesiones();
    double[] frecp = ovector(r1,20);
    double[] frece = ovector(r,20);
    Document document = new Document(PageSize.A4, 50, 50,
100, 72);
    PdfWriter writer = PdfWriter.getInstance(document, new
FileOutputStream(a));
    document.open();
    writer.setPageEvent(new Informel(fase1, document, frecp,
frece, b));

    document.close();
}

```

ANEXO

```
public void setGeneral(Contenedor contenedor) {
    general = new Contenedor();
    general = contenedor;
}

public void setEmpresa(Contenedor contenedor) {
    empresa = new Contenedor();
    empresa = contenedor;
}

public Contenedor getEmpresa() {
    return empresa;
}

public Contenedor getGeneral() {
    return general;
}
}
```

CONTROLADOR 1

```

package org.anesthsom.anesthsom.controlador;

import java.util.Date;
import java.util.Observable;
import java.util.Vector;

import javax.swing.JOptionPane;
import org.anesthsom.anesthsom.modelo.Alm1;
import org.anesthsom.anesthsom.modelo.Alm2;
import org.anesthsom.anesthsom.modelo.Alm3;
import org.anesthsom.anesthsom.modelo.Contenedor;

/**
 * @author Pablo Torner
 *
 * Descripción: Es la clase que se encarga de comunicar lo
 * perteneciente en el paquete modelo1 con el vista1.
 * Sirve para añadir diferentes individuos a las listas y
 * modificar la matriz.
 */
public class Controlador_1 extends Observable{

private static Controlador_1 instance;

    /**
     * @return
     *
     * Funcion Controlador: Es el metador para trabajar con el
     controlador sin necesidad de definirlo.
     */
    /**
     */
    public static Controlador_1 getInstance(){
        if (instance==null) instance = new Controlador_1();
        return instance;
    }

private Contenedor contenedor;

    /**
     * @return
     *
     * Sirve para leer todo el contenido de un contenedor.
     */
    public Iterable<Contenedor> leerContenedor(){
        Vector<Contenedor> listaContenedor = new
Vector<Contenedor>();
        listaContenedor.add(contenedor);
    }
}

```

```

    return listaContenedor;
}
/**
 * @param recaida
 * @param fechaIngreso
 * @param sexo
 * @param fechaNacimiento
 * @param codigoNacionalidad
 * @param situacionProfesional
 * @param antMeses
 * @param antDias
 * @paramCodigoCnae
 * @param tipCont
 * @param regimss
 * @param plantillaEmpresa
 * @param codigoEmpresarial
 * @param contrata
 * @param eTT
 * @param org1
 * @param org2
 * @param org3
 * @param org4
 * @param org5
 * @param org6
 * @param accLug
 * @param trafico
 * @param perteneceEmpresa
 * @param pertenceOtEmpresa
 * @param centroSubcontratado
 * @param centroETT
 * @param centroOtros
 * @param provcen
 * @param plantic
 * @param cnaec
 * @param fechaAccidente
 * @param fechaBaja
 * @param diaSemana
 * @param horaDia
 * @param horaTra
 * @param trabHab
 * @param riesgos
 * @param tipLug
 * @param tipTrab
 * @param actiEsp
 * @param agente1
 * @param desviac
 * @param agente2
 * @param forma
 * @param aparato
 * @param mastrab
 * @param dlesion
 * @param glesion
 * @param pclesionada
 * @param ambulancia
 * @param hospital
 * @param inden75

```

ANEXO

```

    * @param glesionR
    * @param fechaAlta
    * @param codDiag
    * @param jorNTrab
    * @param cccot
    *
    * Procedimiento: Añade un individuo como informe
    completo, y en todos los subs(Informe DeltaReducido, Informe
    OIT).
    */
    public void addAccidenatdoAlm1(boolean recaida, Date
    fechaIngreso, boolean sexo, int fechaNacimiento, int
    codigoNacionalidad, String situacionProfesional, int antMeses,
    int antDias, intCodigoCnae, String tipCont, int regimss, int
    plantillaEmpresa, int codigoEmpresarial, boolean contrata,
    boolean eTT, boolean org1, boolean org2, boolean org3, boolean
    org4, boolean org5, boolean org6, int accLug, boolean trafico,
    boolean perteneceEmpresa, boolean pertenceOtEmpresa, boolean
    centroSubcontratado, boolean centroETT, boolean centroOtros,
    int provcen, int plantic, int cnaec, Date fechaAccidente, Date
    fechaBaja, String diaSemana, int horaDia, int horaTra, boolean
    trabHab, boolean riesgos, int tipLug, int tipTrab, int actiEsp,
    int agentel, int desviac, int agente2, int forma, int aparato,
    boolean mastrab, int dlesion, int glesion, String pclesionada,
    boolean ambulancia, boolean hospital, int inden75, int glesionR,
    Date fechaAlta, int codDiag, int jorNTrab, int cccot){
        if (contenedor==null){
            contenedor=new Contenedor();
        }
        Alm1 accidentado = new Alm1(recaida, fechaIngreso,
        sexo, fechaNacimiento, codigoNacionalidad, situacionProfesional,
        antMeses, antDias, CodigoCnae, tipCont, regimss,
        plantillaEmpresa, codigoEmpresarial, contrata, eTT, org1, org2,
        org3, org4, org5, org6, accLug, trafico, perteneceEmpresa,
        pertenceOtEmpresa, centroSubcontratado, centroETT, centroOtros,
        provcen, plantic, cnaec, fechaAccidente, fechaBaja, diaSemana,
        horaDia, horaTra, trabHab, riesgos, tipLug, tipTrab, actiEsp,
        agentel, desviac, agente2, forma, aparato, mastrab, dlesion,
        glesion, pclesionada, ambulancia, hospital, inden75, glesionR,
        fechaAlta, codDiag, jorNTrab, cccot);
        contenedor.getAccidentados().add(accidentado);
        Alm2 accidentado2= new Alm2( recaida,
        CodigoCnae, plantillaEmpresa, codigoEmpresarial, contrata,
        accLug, trafico, perteneceEmpresa, pertenceOtEmpresa, actiEsp,
        agentel, desviac, agente2, forma, aparato, mastrab,
        dlesion, glesion, pclesionada, glesionR, cccot);
        contenedor.getReducida().add(accidentado2);
        Alm3 seco= contenedor.getP();
        seco.Add(desviac, forma, dlesion);
        contenedor.setP(seco);
        setChanged();
        notifyObservers();
    }

    /**
    * @param recaida

```

ANEXO

```

* @paramCodigoCnae
* @paramplantillaEmpresa
* @paramcodigoEmpresarial
* @paramcontrata
* @paramaccLug
* @paramtrafico
* @paramperteneceEmpresa
* @parampertenceOtEmpresa
* @paramactiEsp
* @paramagentel
* @paramdesviac
* @paramagente2
* @paramforma
* @paramaparato
* @parammastrab
* @paramdlesion
* @paramglesion
* @parampclesionada
* @paramglesionR
* @paramcccot
*
* Procedimiento: Añade un individuo como informe
DeltaReducido, y en todos los subs(Informe OIT).
*/
public void addAccidentadoAlm2(boolean recaida,int
CodigoCnae, int plantillaEmpresa, int codigoEmpresarial,boolean
contrata, int accLug, boolean trafico, boolean
perteneceEmpresa, boolean pertenceOtEmpresa, int actiEsp, int
agentel, int desviac, int agente2, int forma, int
aparato,boolean mastrab, int dlesion, int glesion, String
pclesionada,int glesionR,int cccot){
    if (contenedor==null){
        contenedor=new Contenedor();
    }
    Alm2 accidentado2= new Alm2( recaida,
CodigoCnae,plantillaEmpresa,codigoEmpresarial, contrata,
accLug,trafico, perteneceEmpresa,pertenceOtEmpresa, actiEsp,
agentel, desviac,agente2, forma, aparato, mastrab,
dlesion,glesion, pclesionada, glesionR, cccot);
    contenedor.getReducida().add(accidentado2);
    Alm3 seco= contenedor.getP();
    seco.Add(desviac,forma,dlesion);
    contenedor.setP(seco);
    setChanged();
    notifyObservers();
}

/**
* @paramdesviac
* @paramforma
* @paramdlesion
*
* Añade un individuo como informe DeltaReducido, y en
todos los significativos de Delta(desviacion, forma y lesion)
para pasarlo a OIT
*/

```


ANEXO

```
public void addAccidentadoAlm3(int desviac,int forma,int
dlesion){
    if (contenedor==null){
        contenedor=new Contenedor();
    }
    if ((desviac==0)|| (forma==0)|| (dlesion==0)){
        JOptionPane.showMessageDialog(null,"No se ha acordado
al tener uno o más campos vacios");
    }
    Alm3 seco= contenedor.getP();
    seco.Add(desviac, forma, dlesion);
    contenedor.setP(seco);
    setChanged();
    notifyObservers();
}

/**
 * @param r1
 * @param r2
 * @param r3
 * @param r4
 * @param r5
 * @param r6
 * @param r7
 * @param r8
 * @param r9
 * @param r10
 * @param r11
 * @param r12
 * @param r13
 * @param r14
 * @param r15
 * @param r16
 * @param r17
 * @param r18
 * @param r19
 * @param l1
 * @param l2
 * @param l3
 * @param l4
 * @param l5
 * @param l6
 * @param l7
 * @param l8
 * @param l9
 * @param l10
 * @param l11
 * @param l12
 * @param l13
 * @param l14
 * @param l15
 * @param l16
 * @param l17
 * @param l18
 * @param l19
```

ANEXO

```

*
* Agrega un individuo como informe OIT
*/
public void addAccidentadoAlm4(int r1, int r2, int r3,int
r4,int r5, int r6, int r7, int r8,int r9,int r10,int r11, int
r12, int r13, int r14,int r15,int r16,int r17, int r18, int
r19,int r20,int r21, int r22, int r23,int r24,int r25, int r26,
int r27, int r28,int r29,int r30,int r31, int r32, int r33, int
r34,int r35,int r36,int r37, int r38, int r39,int r40,int r41,
int r42, int r43,int r44,int r45, int r46, int r47, int r48,int
r49,int r50,int r51, int r52, int r53, int r54,int r55,int
r56,int r57, int r58, int r59,int r60,int l1, int l2, int l3,int
l4,int l5, int l6, int l7, int l8,int l9,int l10,int l11, int
l12, int l13, int l14,int l15,int l16,int l17, int l18, int
l19,int l20){
    if (contenedor==null){
        contenedor=new Contenedor();
    }
    Alm3 seco=new Alm3();
    seco= contenedor.getP();
    seco.Add2( r1, r2, r3, r4, r5, r6, r7, r8, r9,
r10, r11, r12, r13, r14, r15, r16, r17, r18, r19,r20,r21,
r22, r23, r24, r25, r26, r27, r28, r29, r30, r31, r32,
r33, r34, r35, r36, r37, r38, r39,r40,r41, r42, r43, r44,
r45, r46, r47, r48, r49, r50, r51, r52, r53, r54, r55,
r56, r57, r58, r59,r60, l1, l2, l3, l4, l5, l6, l7, l8,
l9, l10, l11, l12, l13, l14, l15, l16, l17, l18, l19,l20);
    contenedor.setP(seco);
    setChanged();
    notifyObservers();
}
/**
 * @return
 *
 * Procedimiento abstracto para tomar valor el contenedor.
 */
public Contenedor getContenedor() {
    return contenedor;
}

/**
 * @param contenedor
 *
 * Procedimiento abstracto que sirve para cambiar el
contenedor
 */
public void setContenedor(Contenedor contenedor) {
    this.contenedor = contenedor;
}

}

```

MODELO**Alm1**

```

package org.anesthsom.anesthsom.modelo;

import java.io.Serializable;
import java.util.Date;

public class Alm1 implements Serializable {
    private static final long serialVersionUID = 1L;
    private boolean recaida;
    private Date fechaIngreso;
    private boolean sexo;
    private int fechaNacimiento;
    private int codigoNacionalidad;
    private String situacionProfesional;
    private int antMeses;
    private int antDias;
    private intCodigoCnae;
    private String tipCont;
    private int regimss;
    private int plantillaEmpresa;
    private int codigoEmpresarial;
    private boolean contrata;
    private boolean eTT;
    private boolean org1;
    private boolean org2;
    private boolean org3;
    private boolean org4;
    private boolean org5;
    private boolean org6;
    private int accLug;
    private boolean trafico;
    private boolean perteneceEmpresa;
    private boolean pertenceOtEmpresa;
    private boolean centroSubcontratado;
    private boolean centroETT;
    private boolean centroOtros;
    private int provcen;
    private int plantic;
    private int cnaec;
    private Date fechaAccidente;
    private Date fechaBaja;
    private String diaSemana;
    private int horaDia;
    private int horaTra;
    private boolean trabHab;
    private boolean riesgos;
    private int tipLug;
    private int tipTrab;
    private int actiEsp;
    private int agentel;

```

```

private int desviac;
private int agente2;
private int forma;
private int aparato;
private boolean mastrab;
private int dlesion;
private int glesion;
private String pclesionada;
private boolean ambulancia;
private boolean hospital;
private int inden75;
private int glesionR;
private Date fechaAlta;
private int codDiag;
private int jorNTrab;
private int cccot;

/**
 * @param recaida
 * @param fechaIngreso
 * @param sexo
 * @param fechaNacimiento
 * @param codigoNacionalidad
 * @param situacionProfesional
 * @param antMeses
 * @param antDias
 * @param codigoCnae
 * @param tipCont
 * @param regimss
 * @param plantillaEmpresa
 * @param codigoEmpresarial
 * @param contrata
 * @param ett
 * @param org1
 * @param org2
 * @param org3
 * @param org4
 * @param org5
 * @param org6
 * @param accLug
 * @param trafico
 * @param perteneceEmpresa
 * @param pertenceOtEmpresa
 * @param centroSubcontratado
 * @param centroETT
 * @param centroOtros
 * @param provcen
 * @param plantic
 * @param cnaec
 * @param fechaAccidente
 * @param fechaBaja
 * @param diaSemana
 * @param horaDia
 * @param horaTra
 * @param trabHab
 * @param riesgos

```

ANEXO

```

* @param tipLug
* @param tipTrab
* @param actiEsp
* @param agentel
* @param desviac
* @param agente2
* @param forma
* @param aparato
* @param mastrab
* @param dlesion
* @param glesion
* @param pclesionada
* @param ambulancia
* @param hospital
* @param inden75
* @param glesionR
* @param fechaAlta
* @param codDiag
* @param jorNTrab
* @param cccot
*/
public Alm1(boolean recaida, Date fechaIngreso, boolean
sexo,
                int fechaNacimiento, int codigoNacionalidad,
                String situacionProfesional, int antMeses, int
antDias,
                int codigoCnae, String tipCont, int regimss, int
plantillaEmpresa,
                int codigoEmpresarial, boolean contrata,
boolean ett,
                boolean org1, boolean org2, boolean org3,
boolean org4,
                boolean org5, boolean org6, int accLug, boolean
trafico,
                boolean perteneceEmpresa, boolean
pertenceOtEmpresa,
                boolean centroSubcontratado, boolean centroETT,
                boolean centroOtros, int provcen, int plantic,
int cnaec,
                Date fechaAccidente, Date fechaBaja, String
diaSemana, int horaDia,
                int horaTra, boolean trabHab, boolean riesgos,
int tipLug,
                int tipTrab, int actiEsp, int agentel, int
desviac, int agente2,
                int forma, int aparato, boolean mastrab, int
dlesion, int glesion,
                String pclesionada, boolean ambulancia, boolean
hospital,
                int inden75, int glesionR, Date fechaAlta, int
codDiag,
                int jorNTrab, int cccot) {
    super();
    this.recaida = recaida;
    this.fechaIngreso = fechaIngreso;
    this.sexo = sexo;

```

ANEXO

```
this.fechaNacimiento = fechaNacimiento;
this.codigoNacionalidad = codigoNacionalidad;
this.situacionProfesional = situacionProfesional;
this.antMeses = antMeses;
this.antDias = antDias;
CodigoCnae = codigoCnae;
this.tipCont = tipCont;
this.regimss = regimss;
this.plantillaEmpresa = plantillaEmpresa;
this.codigoEmpresarial = codigoEmpresarial;
this.contrata = contrata;
eTT = ett;
this.org1 = org1;
this.org2 = org2;
this.org3 = org3;
this.org4 = org4;
this.org5 = org5;
this.org6 = org6;
this.accLug = accLug;
this.trafico = trafico;
this.perteneceEmpresa = perteneceEmpresa;
this.pertenceOtEmpresa = pertenceOtEmpresa;
this.centroSubcontratado = centroSubcontratado;
this.centroETT = centroETT;
this.centroOtros = centroOtros;
this.provcen = provcen;
this.plantic = plantic;
this.cnaec = cnaec;
this.fechaAccidente = fechaAccidente;
this.fechaBaja = fechaBaja;
this.diaSemana = diaSemana;
this.horaDia = horaDia;
this.horaTra = horaTra;
this.trabHab = trabHab;
this.riesgos = riesgos;
this.tipLug = tipLug;
this.tipTrab = tipTrab;
this.actiEsp = actiEsp;
this.agente1 = agente1;
this.desviac = desviac;
this.agente2 = agente2;
this.forma = forma;
this.aparato = aparato;
this.mastrab = mastrab;
this.dlesion = dlesion;
this.glesion = glesion;
this.pclesionada = pclesionada;
this.ambulancia = ambulancia;
this.hospital = hospital;
this.inden75 = inden75;
this.glesionR = glesionR;
this.fechaAlta = fechaAlta;
this.codDiag = codDiag;
this.jorNTrab = jorNTrab;
this.cccot = cccot;
```

}

```
public boolean isRecaida() {
    return recaida;
}

public void setRecaida(boolean recaida) {
    this.recaida = recaida;
}

public Date getFechaIngreso() {
    return fechaIngreso;
}

public void setFechaIngreso(Date fechaIngreso) {
    this.fechaIngreso = fechaIngreso;
}

public boolean isSexo() {
    return sexo;
}

public void setSexo(boolean sexo) {
    this.sexo = sexo;
}

public int getFechaNacimiento() {
    return fechaNacimiento;
}

public void setFechaNacimiento(int fechaNacimiento) {
    this.fechaNacimiento = fechaNacimiento;
}

public int getCodigoNacionalidad() {
    return codigoNacionalidad;
}

public void setCodigoNacionalidad(int codigoNacionalidad) {
    this.codigoNacionalidad = codigoNacionalidad;
}

public String getSituacionProfesional() {
    return situacionProfesional;
}

public void setSituacionProfesional(String
situacionProfesional) {
    this.situacionProfesional = situacionProfesional;
}

public int getAntMeses() {
    return antMeses;
}

public void setAntMeses(int antMeses) {
    this.antMeses = antMeses;
}
```

```
}

public int getAntDias() {
    return antDias;
}

public void setAntDias(int antDias) {
    this.antDias = antDias;
}

public int getCodigoCnae() {
    return CodigoCnae;
}

public void setCodigoCnae(int codigoCnae) {
    CodigoCnae = codigoCnae;
}

public String getTipCont() {
    return tipCont;
}

public void setTipCont(String tipCont) {
    this.tipCont = tipCont;
}

public int getRegimss() {
    return regimss;
}

public void setRegimss(int regimss) {
    this.regimss = regimss;
}

public int getPlantillaEmpresa() {
    return plantillaEmpresa;
}

public void setPlantillaEmpresa(int plantillaEmpresa) {
    this.plantillaEmpresa = plantillaEmpresa;
}

public int getCodigoEmpresarial() {
    return codigoEmpresarial;
}

public void setCodigoEmpresarial(int codigoEmpresarial) {
    this.codigoEmpresarial = codigoEmpresarial;
}

public boolean isContrata() {
    return contrata;
}

public void setContrata(boolean contrata) {
    this.contrata = contrata;
}
```



```
}

public boolean isETT() {
    return eTT;
}

public void setETT(boolean ett) {
    eTT = ett;
}

public boolean isOrg1() {
    return org1;
}

public void setOrg1(boolean org1) {
    this.org1 = org1;
}

public boolean isOrg2() {
    return org2;
}

public void setOrg2(boolean org2) {
    this.org2 = org2;
}

public boolean isOrg3() {
    return org3;
}

public void setOrg3(boolean org3) {
    this.org3 = org3;
}

public boolean isOrg4() {
    return org4;
}

public void setOrg4(boolean org4) {
    this.org4 = org4;
}

public boolean isOrg5() {
    return org5;
}

public void setOrg5(boolean org5) {
    this.org5 = org5;
}

public boolean isOrg6() {
    return org6;
}

public void setOrg6(boolean org6) {
    this.org6 = org6;
}
```

```

}

public int getAccLug() {
    return accLug;
}

public void setAccLug(int accLug) {
    this.accLug = accLug;
}

public boolean isTrafico() {
    return trafico;
}

public void setTrafico(boolean trafico) {
    this.trafico = trafico;
}

public boolean isPerteneceEmpresa() {
    return perteneceEmpresa;
}

public void setPerteneceEmpresa(boolean perteneceEmpresa) {
    this.perteneceEmpresa = perteneceEmpresa;
}

public boolean isPertenceOtEmpresa() {
    return pertenceOtEmpresa;
}

public void setPertenceOtEmpresa(boolean pertenceOtEmpresa)
{
    this.pertenceOtEmpresa = pertenceOtEmpresa;
}

public boolean isCentroSubcontratado() {
    return centroSubcontratado;
}

public void setCentroSubcontratado(boolean
centroSubcontratado) {
    this.centroSubcontratado = centroSubcontratado;
}

public boolean isCentroETT() {
    return centroETT;
}

public void setCentroETT(boolean centroETT) {
    this.centroETT = centroETT;
}

public boolean isCentroOtros() {
    return centroOtros;
}

```

ANEXO

```
public void setCentroOtros(boolean centroOtros) {
    this.centroOtros = centroOtros;
}

public int getProvcen() {
    return provcen;
}

public void setProvcen(int provcen) {
    this.provcen = provcen;
}

public int getPlantic() {
    return plantic;
}

public void setPlantic(int plantic) {
    this.plantic = plantic;
}

public int getCnaec() {
    return cnaec;
}

public void setCnaec(int cnaec) {
    this.cnaec = cnaec;
}

public Date getFechaAccidente() {
    return fechaAccidente;
}

public void setFechaAccidente(Date fechaAccidente) {
    this.fechaAccidente = fechaAccidente;
}

public Date getFechaBaja() {
    return fechaBaja;
}

public void setFechaBaja(Date fechaBaja) {
    this.fechaBaja = fechaBaja;
}

public String getDiaSemana() {
    return diaSemana;
}

public void setDiaSemana(String diaSemana) {
    this.diaSemana = diaSemana;
}

public int getHoraDia() {
    return horaDia;
}
```

ANEXO

```
public void setHoraDia(int horaDia) {
    this.horaDia = horaDia;
}

public int getHoraTra() {
    return horaTra;
}

public void setHoraTra(int horaTra) {
    this.horaTra = horaTra;
}

public boolean isTrabHab() {
    return trabHab;
}

public void setTrabHab(boolean trabHab) {
    this.trabHab = trabHab;
}

public boolean isRiesgos() {
    return riesgos;
}

public void setRiesgos(boolean riesgos) {
    this.riesgos = riesgos;
}

public int getTipLug() {
    return tipLug;
}

public void setTipLug(int tipLug) {
    this.tipLug = tipLug;
}

public int getTipTrab() {
    return tipTrab;
}

public void setTipTrab(int tipTrab) {
    this.tipTrab = tipTrab;
}

public int getActiEsp() {
    return actiEsp;
}

public void setActiEsp(int actiEsp) {
    this.actiEsp = actiEsp;
}

public int getAgentel() {
    return agentel;
}
```

ANEXO

```
public void setAgente1(int agente1) {
    this.agente1 = agente1;
}

public int getDesviac() {
    return desviac;
}

public void setDesviac(int desviac) {
    this.desviac = desviac;
}

public int getAgente2() {
    return agente2;
}

public void setAgente2(int agente2) {
    this.agente2 = agente2;
}

public int getForma() {
    return forma;
}

public void setForma(int forma) {
    this.forma = forma;
}

public int getAparato() {
    return aparato;
}

public void setAparato(int aparato) {
    this.aparato = aparato;
}

public boolean isMastrab() {
    return mastrab;
}

public void setMastrab(boolean mastrab) {
    this.mastrab = mastrab;
}

public int getDlesion() {
    return dlesion;
}

public void setDlesion(int dlesion) {
    this.dlesion = dlesion;
}

public int getGlesion() {
    return glesion;
}
```

ANEXO

```
public void setGlesion(int glesion) {
    this.glesion = glesion;
}

public String getPclesionada() {
    return pclesionada;
}

public void setPclesionada(String pclesionada) {
    this.pclesionada = pclesionada;
}

public boolean isAmbulancia() {
    return ambulancia;
}

public void setAmbulancia(boolean ambulancia) {
    this.ambulancia = ambulancia;
}

public boolean isHospital() {
    return hospital;
}

public void setHospital(boolean hospital) {
    this.hospital = hospital;
}

public int getInden75() {
    return inden75;
}

public void setInden75(int inden75) {
    this.inden75 = inden75;
}

public int getGlesionR() {
    return glesionR;
}

public void setGlesionR(int glesionR) {
    this.glesionR = glesionR;
}

public Date getFechaAlta() {
    return fechaAlta;
}

public void setFechaAlta(Date fechaAlta) {
    this.fechaAlta = fechaAlta;
}

public int getCodDiag() {
    return codDiag;
}
```

ANEXO

```
public void setCodDiag(int codDiag) {
    this.codDiag = codDiag;
}

public int getJorNTrab() {
    return jorNTrab;
}

public void setJorNTrab(int jorNTrab) {
    this.jorNTrab = jorNTrab;
}

public int getCccot() {
    return cccot;
}

public void setCccot(int cccot) {
    this.cccot = cccot;
}
}
```

Alm2

```

package org.anesthsom.anesthsom.modelo;
import java.io.Serializable;

public class Alm2 implements Serializable {
    private static final long serialVersionUID =
7120693496976057338L;

    private boolean recaida;
    private intCodigoCnae;
    private int plantillaEmpresa;
    private int codigoEmpresarial;
    private boolean contrata;
    private int accLug;
    private boolean trafico;
    private boolean perteneceEmpresa;
    private boolean pertenceOtEmpresa;
    private int actiEsp;
    private int agentel;
    private int desviac;
    private int agente2;
    private int forma;
    private int aparato;
    private boolean mastrab;
    private int dlesion;
    private int glesion;
    private String pclesionada;
    private int glesionR;
    private int cccot;
    /**
     * @param recaida
     * @param codigoCnae
     * @param plantillaEmpresa
     * @param codigoEmpresarial
     * @param contrata
     * @param accLug
     * @param trafico
     * @param perteneceEmpresa
     * @param pertenceOtEmpresa
     * @param actiEsp
     * @param agentel
     * @param desviac
     * @param agente2
     * @param forma
     * @param aparato
     * @param mastrab
     * @param dlesion
     * @param glesion
     * @param pclesionada
     * @param glesionR
     * @param cccot
     */
}

```


ANEXO

```
public Alm2(boolean recaida, int codigoCnae, int
plantillaEmpresa,
            int codigoEmpresarial, boolean contrata,
int accLug,
            boolean trafico, boolean perteneceEmpresa,
            boolean pertenceOtEmpresa, int actiEsp,
int agente1, int desviac,
            int agente2, int forma, int aparato,
boolean mastrab, int dlesion,
            int glesion, String pclesionada, int
glesionR, int cccot) {
    super();
    this.recaida = recaida;
    this.codigoCnae = codigoCnae;
    this.plantillaEmpresa = plantillaEmpresa;
    this.codigoEmpresarial = codigoEmpresarial;
    this.contrata = contrata;
    this.accLug = accLug;
    this.trafico = trafico;
    this.perteneceEmpresa = perteneceEmpresa;
    this.pertenceOtEmpresa = pertenceOtEmpresa;
    this.actiEsp = actiEsp;
    this.agente1 = agente1;
    this.desviac = desviac;
    this.agente2 = agente2;
    this.forma = forma;
    this.aparato = aparato;
    this.mastrab = mastrab;
    this.dlesion = dlesion;
    this.glesion = glesion;
    this.pclesionada = pclesionada;
    this.glesionR = glesionR;
    this.cccot = cccot;
}
/**
 * @return
 */
public boolean isRecaida() {
    return recaida;
}
/**
 * @param recaida
 */
public void setRecaida(boolean recaida) {
    this.recaida = recaida;
}
/**
 * @return
 */
public int getCodigoCnae() {
    return this.codigoCnae;
}
/**
 * @param codigoCnae
 */
public void setCodigoCnae(int codigoCnae) {
```

```

       CodigoCnae = codigoCnae;
    }
    /**
     * @return
     */
    public int getPlantillaEmpresa() {
        return plantillaEmpresa;
    }
    /**
     * @param plantillaEmpresa
     */
    public void setPlantillaEmpresa(int plantillaEmpresa)
{
        this.plantillaEmpresa = plantillaEmpresa;
    }
    /**
     * @return
     */
    public int getCodigoEmpresarial() {
        return codigoEmpresarial;
    }
    /**
     * @param codigoEmpresarial
     */
    public void setCodigoEmpresarial(int
codigoEmpresarial) {
        this.codigoEmpresarial = codigoEmpresarial;
    }
    /**
     * @return
     */
    public boolean isContrata() {
        return contrata;
    }
    /**
     * @param contrata
     */
    public void setContrata(boolean contrata) {
        this.contrata = contrata;
    }
    /**
     * @return
     */
    public int getAccLug() {
        return accLug;
    }
    /**
     * @param accLug
     */
    public void setAccLug(int accLug) {
        this.accLug = accLug;
    }
    /**
     * @return
     */
    public boolean isTrafico() {

```

```

        return trafico;
    }
    /**
     * @param trafico
     */
    public void setTrafico(boolean trafico) {
        this.trafico = trafico;
    }
    /**
     * @return
     */
    public boolean isPerteneceEmpresa() {
        return perteneceEmpresa;
    }
    /**
     * @param perteneceEmpresa
     */
    public void setPerteneceEmpresa(boolean
perteneceEmpresa) {
        this.perteneceEmpresa = perteneceEmpresa;
    }
    /**
     * @return
     */
    public boolean isPertenceOtEmpresa() {
        return pertenceOtEmpresa;
    }
    /**
     * @param pertenceOtEmpresa
     */
    public void setPertenceOtEmpresa(boolean
pertenceOtEmpresa) {
        this.pertenceOtEmpresa = pertenceOtEmpresa;
    }
    /**
     * @return
     */
    public int getActiEsp() {
        return actiEsp;
    }
    /**
     * @param actiEsp
     */
    public void setActiEsp(int actiEsp) {
        this.actiEsp = actiEsp;
    }
    /**
     * @return
     */
    public int getAgentel() {
        return agentel;
    }
    /**
     * @param agentel
     */
    public void setAgentel(int agentel) {

```

```

        this.agente1 = agente1;
    }
    /**
     * @return
     */
    public int getDesviac() {
        return desviac;
    }
    /**
     * @param desviac
     */
    public void setDesviac(int desviac) {
        this.desviac = desviac;
    }
    /**
     * @return
     */
    public int getAgente2() {
        return agente2;
    }
    /**
     * @param agente2
     */
    public void setAgente2(int agente2) {
        this.agente2 = agente2;
    }
    /**
     * @return
     */
    public int getForma() {
        return forma;
    }
    /**
     * @param forma
     */
    public void setForma(int forma) {
        this.forma = forma;
    }
    /**
     * @return
     */
    public int getAparato() {
        return aparato;
    }
    /**
     * @param aparato
     */
    public void setAparato(int aparato) {
        this.aparato = aparato;
    }
    /**
     * @return
     */
    public boolean isMastrab() {
        return mastrab;
    }
}

```

```

/**
 * @param mastrab
 */
public void setMastrab(boolean mastrab) {
    this.mastrab = mastrab;
}
/**
 * @return
 */
public int getDlesion() {
    return dlesion;
}
/**
 * @param dlesion
 */
public void setDlesion(int dlesion) {
    this.dlesion = dlesion;
}
/**
 * @return
 */
public int getGlesion() {
    return glesion;
}
/**
 * @param glesion
 */
public void setGlesion(int glesion) {
    this.glesion = glesion;
}
/**
 * @return
 */
public String getPclesionada() {
    return pclesionada;
}
/**
 * @param pclesionada
 */
public void setPclesionada(String pclesionada) {
    this.pclesionada = pclesionada;
}
/**
 * @return
 */
public int getGlesionR() {
    return glesionR;
}
/**
 * @param glesionR
 */
public void setGlesionR(int glesionR) {
    this.glesionR = glesionR;
}
/**
 * @return

```

```
*/  
  
public int getCccot() {  
    return cccot;  
}  
/**  
 * @param cccot  
 */  
public void setCccot(int cccot) {  
    this.cccot = cccot;  
}  
  
}
```

Alm3

```

package org.anesthsom.anesthsom.modelo;
import java.io.Serializable;

import javax.swing.JOptionPane;

public class Alm3 implements Serializable{
    private static final long serialVersionUID = -
2401236156002873588L;
    private int riesgo[]=new int[60];
    private int lesiones[]= new int[20];
    private int desechado;
    public Alm3(){
        super();
    }
    /**
     * @param desviac
     * @param forma
     * @param a
     */
    public void Add(int desviac, int forma, int a) {

        lesiones[a-1]=lesiones[a-1]+1;
        if ((forma==11) || (forma==12)){
            riesgo[15-1]=riesgo[15-1]+1;
        }else if ((forma==13) || (forma==14)){
            riesgo[14-1]=riesgo[14-1]+1;
        }else if
((forma==15) || (forma==16) || (forma==17) || (forma==21) || (forma==23)
){
            riesgo[16-1]=riesgo[16-1]+1;
        }else
if((forma==22) || (forma==61) || (forma==62) || (forma==63) || (forma==6
4)){
            riesgo[11-1]=riesgo[11-1]+1;
        }else if(forma==31){
            riesgo[1-1]=riesgo[1-1]+1;
        }else if(forma==32){
            riesgo[7-1]=riesgo[7-1]+1;
        }else if(forma==41){
            riesgo[10-1]=riesgo[10-1]+1;
        }else if(forma==42){
            riesgo[3-1]=riesgo[3-1]+1;
        }else if((forma==43) || (forma==44) || (forma==46)){
            riesgo[8-1]=riesgo[8-1]+1;
        }else if(forma==45){
            riesgo[12-1]=riesgo[12-1]+1;
        }else if((forma==51) || (forma==52) || (forma==53)){
            riesgo[9-1]=riesgo[9-1]+1;
        }else if((forma==71)){

```

ANEXO

```

        riesgo[13-1]=riesgo[13-1]+1;
    }else
if((forma==73)|| (forma==81)|| (forma==82)|| (forma==83)|| (forma==8
9)) {
        riesgo[19-1]=riesgo[19-1]+1;
    }else{
        if((desviac==11)|| (desviac==12)) {
            riesgo[15-1]=riesgo[15-1]+1;
        }else if((desviac==13)|| (desviac==14)) {
            riesgo[18-1]=riesgo[18-1]+1;
        }else
if((desviac==21)|| (desviac==34)|| (desviac==35)) {
            riesgo[3-1]=riesgo[3-1]+1;
        }else
if((desviac==22)|| (desviac==23)|| (desviac==24)) {
            riesgo[16-1]=riesgo[16-1]+1;
        }else if((desviac==31)|| (desviac==32)) {
            riesgo[10-1]=riesgo[10-1]+1;
        }else if(desviac==33) {
            riesgo[5-1]=riesgo[5-1]+1;
        }else if(desviac==41) {
            riesgo[8-1]=riesgo[8-1]+1;
        }else if (desviac==42) {
            riesgo[12-1]=riesgo[12-1]+1;
        }else if((desviac==43)|| (desviac==49)) {
            riesgo[9-1]=riesgo[9-1]+1;
        }else if(desviac==44) {
            riesgo[4-1]=riesgo[4-1]+1;
        }else
if((desviac==45)|| (desviac==81)|| (desviac==82)|| (desviac==83)|| (
desviac==84)|| (desviac==85)|| (desviac==89)) {
            riesgo[19-1]=riesgo[19-1]+1;
        }else if(desviac==51) {
            riesgo[1-1]=riesgo[1-1]+1;
        }else if (desviac==52) {
            riesgo[2-1]=riesgo[2-1]+1;
        }else if (desviac==61) {
            riesgo[6-1]=riesgo[6-1]+1;
        }else if ((desviac==62)|| (desviac==64)) {
            riesgo[7-1]=riesgo[7-1]+1;
        }else if(desviac==63) {
            riesgo[11-1]=riesgo[11-1]+1;
        }else
if((desviac==71)|| (desviac==72)|| (desviac==73)|| (desviac==74)|| (
desviac==75)) {
            riesgo[13-1]=riesgo[13-1]+1;
        }else{
            desechado++;
        }
    }
}
int b=0;
for(int i=0;i<riesgo.length;i++){
    if(riesgo[i]!=0) {
        b=i+1;
    }
}

```



```

        JOptionPane.showMessageDialog(null, "el riesgo
es: "+ b);
        JOptionPane.showMessageDialog(null, "el numero
de descartados es: "+ desechado);
    }
/**
 * @param r1
 * @param r2
 * @param r3
 * @param r4
 * @param r5
 * @param r6
 * @param r7
 * @param r8
 * @param r9
 * @param r10
 * @param r11
 * @param r12
 * @param r13
 * @param r14
 * @param r15
 * @param r16
 * @param r17
 * @param r18
 * @param r19
 * @param l1
 * @param l2
 * @param l3
 * @param l4
 * @param l5
 * @param l6
 * @param l7
 * @param l8
 * @param l9
 * @param l10
 * @param l11
 * @param l12
 * @param l13
 * @param l14
 * @param l15
 * @param l16
 * @param l17
 * @param l18
 * @param l19
 */
    public void Add2(int r1, int r2, int r3,int r4,int r5, int
r6, int r7, int r8,int r9,int r10,int r11, int r12, int r13, int
r14,int r15,int r16,int r17, int r18, int r19,int l1, int l2,
int l3,int l4,int l5, int l6, int l7, int l8,int l9,int l10,int
l11, int l12, int l13, int l14,int l15,int l16,int l17, int l18,
int l19) {
        riesgo[0]=r1;
        riesgo[1]=r2;
        riesgo[2]=r3;
        riesgo[3]=r4;

```

```

        riesgo[4]=r5;
        riesgo[5]=r6;
        riesgo[6]=r7;
        riesgo[7]=r8;
        riesgo[8]=r9;
        riesgo[9]=r10;
        riesgo[10]=r11;
        riesgo[11]=r12;
        riesgo[12]=r13;
        riesgo[13]=r14;
        riesgo[14]=r15;
        riesgo[15]=r16;
        riesgo[16]=r17;
        riesgo[17]=r18;
        riesgo[18]=r19;
        lesiones[0]=l1;
        lesiones[1]=l2;
        lesiones[2]=l3;
        lesiones[3]=l4;
        lesiones[4]=l5;
        lesiones[5]=l6;
        lesiones[6]=l7;
        lesiones[7]=l8;
        lesiones[8]=l9;
        lesiones[9]=l10;
        lesiones[10]=l11;
        lesiones[11]=l12;
        lesiones[12]=l13;
        lesiones[13]=l14;
        lesiones[14]=l15;
        lesiones[15]=l16;
        lesiones[16]=l17;
        lesiones[17]=l18;
        lesiones[18]=l19;
    }
    /**
     * @return
     */
    public int[] getRiesgo() {
        return riesgo;
    }
    /**
     * @param riesgo
     */
    public void setRiesgo(int[] riesgo) {
        this.riesgo = riesgo;
    }
    /**
     * @return
     */
    public int[] getLesiones() {
        return lesiones;
    }
    /**
     * @param lesiones
     */

```

```

public void setLesiones(int[] lesiones) {
    this.lesiones = lesiones;
}
/**
 * @return
 */
public int getDesechado() {
    return desechado;
}
/**
 * @param desechado
 */
public void setDesechado(int desechado) {
    this.desechado = desechado;
}

public void Add2(int r1, int r2, int r3, int r4, int r5, int
r6, int r7, int r8, int r9, int r10, int r11, int r12, int r13,
int r14, int r15, int r16, int r17, int r18, int r19, int r20,
int r21, int r22, int r23, int r24, int r25, int r26, int r27,
int r28, int r29, int r30, int r31, int r32, int r33, int r34,
int r35, int r36, int r37, int r38, int r39, int r40, int r41,
int r42, int r43, int r44, int r45, int r46, int r47, int r48,
int r49, int r50, int r51, int r52, int r53, int r54, int r55,
int r56, int r57, int r58, int r59, int r60, int l1, int l2, int
l3, int l4, int l5, int l6, int l7, int l8, int l9, int l10, int
l11, int l12, int l13, int l14, int l15, int l16, int l17, int
l18, int l19, int l20) {
    riesgo[0]=r1;
    riesgo[1]=r2;
    riesgo[2]=r3;
    riesgo[3]=r4;
    riesgo[4]=r5;
    riesgo[5]=r6;
    riesgo[6]=r7;
    riesgo[7]=r8;
    riesgo[8]=r9;
    riesgo[9]=r10;
    riesgo[10]=r11;
    riesgo[11]=r12;
    riesgo[12]=r13;
    riesgo[13]=r14;
    riesgo[14]=r15;
    riesgo[15]=r16;
    riesgo[16]=r17;
    riesgo[17]=r18;
    riesgo[18]=r19;
    riesgo[19]=r20;
    riesgo[20]=r21;
    riesgo[21]=r22;
    riesgo[22]=r23;
    riesgo[23]=r24;
    riesgo[24]=r25;
    riesgo[25]=r26;
    riesgo[26]=r27;
    riesgo[27]=r28;
}

```

```
riesgo[28]=r29;
riesgo[29]=r30;
riesgo[30]=r31;
riesgo[31]=r32;
riesgo[32]=r33;
riesgo[33]=r34;
riesgo[34]=r35;
riesgo[35]=r36;
riesgo[36]=r37;
riesgo[37]=r38;
riesgo[38]=r39;
    riesgo[39]=r40;
    riesgo[40]=r41;
riesgo[41]=r42;
riesgo[42]=r43;
riesgo[43]=r44;
riesgo[44]=r45;
riesgo[45]=r46;
riesgo[46]=r47;
riesgo[47]=r48;
riesgo[48]=r49;
riesgo[49]=r50;
riesgo[50]=r51;
riesgo[51]=r52;
riesgo[52]=r53;
riesgo[53]=r54;
riesgo[54]=r55;
riesgo[55]=r56;
riesgo[56]=r57;
riesgo[57]=r58;
riesgo[58]=r59;
    riesgo[59]=r60;
lesiones[0]=l1;
lesiones[1]=l2;
lesiones[2]=l3;
lesiones[3]=l4;
lesiones[4]=l5;
lesiones[5]=l6;
lesiones[6]=l7;
lesiones[7]=l8;
lesiones[8]=l9;
lesiones[9]=l10;
lesiones[10]=l11;
lesiones[11]=l12;
lesiones[12]=l13;
lesiones[13]=l14;
lesiones[14]=l15;
lesiones[15]=l16;
lesiones[16]=l17;
lesiones[17]=l18;
lesiones[18]=l19;
    lesiones[19]=l20;
}
}
```

COMMAND

```
package org.anesthsom.anesthsom.modelo;

/**
 * @author Pablo Torner
 *
 * Interface Command.
 *
 */
public interface Command {
void execute();
}
```

CONTENEDOR

```

package org.anesthsom.anesthsom.modelo;

import java.io.Serializable;

/**
 * @author Pablo Torner Pelay
 *
 *
 * Descripción: esta será la clase que sirva para almacenar
 todas las listas
 * contendrá una lista de datos en formato DELTA(Informe
 Completo y reducido)
 * y un elemento para almacenar OIT. Como almacen implementa
 Serializable para
 * que se pueda guardar
 *
 */
public class Contenedor implements Serializable {

    /**
     *
     */
    private static final long serialVersionUID = 1L;
    @Lista
    private ListaAlm1 accidentados;
    @Lista
    private ListaAlm2 reducida;

    private Alm3 p;

    /**
     * Constructor de la clase crea nuevas listas y un nuevo
 objeto OIT
     */
    public Contenedor() {
        accidentados = new ListaAlm1();
        reducida = new ListaAlm2();
        p = new Alm3();
    }

    public ListaAlm1 getAccidentados() {
        return accidentados;
    }

    public void setAccidentados(ListaAlm1 accidentados) {
        this.accidentados = accidentados;
    }
}

```

```
public ListaAlm2 getReducida() {
    return reducida;
}

public void setReducida(ListaAlm2 reducida) {
    this.reducida = reducida;
}

public Alm3 getP() {
    return p;
}

public void setP(Alm3 rl) {
    p = rl;
}
}
```

DIAGRAMA BARRAS

```

package org.anesthsom.anesthsom.modelo;

import java.awt.Color;
import java.awt.GradientPaint;

import javax.swing.JFrame;

import org.jfree.chart.ChartFactory;
import org.jfree.chart.ChartPanel;
import org.jfree.chart.JFreeChart;
import org.jfree.chart.axis.NumberAxis;
import org.jfree.chart.plot.CategoryPlot;
import org.jfree.chart.plot.PlotOrientation;
import org.jfree.chart.renderer.category.BarRenderer;
import org.jfree.data.category.CategoryDataset;
import org.jfree.data.category.DefaultCategoryDataset;

/**
 * @author Pablo Torner Pelay
 *
 */
public class DiagramaBarras extends JFrame {
    private static final long serialVersionUID = 1L;
    /**
     * @param title
     * @param p
     * @param d
     * @param orden
     * Constructor: Se encarga de crear un grafico de
    barras referente a los riesgos, a partir de:
     * El titulo de la ventana, el residual y la
    diferencia de posicion del patron respecto de la empresa y el
    orden del patron.
     */
    public DiagramaBarras(String title, float p[], float
    d[], int orden[]) {
        super(title);
        CategoryDataset dataset =
    createDataset(p, d, orden);
        JFreeChart chart = createChart(dataset);
        // añadir el chart al panel
        ChartPanel chartPanel = new ChartPanel(chart);
        chartPanel.setPreferredSize(new
    java.awt.Dimension(500, 270));
        setContentPane(chartPanel);
    }
    /**
     * @param p
     * @param d
     * @param orden
     * @return
    */

```


ANEXO

```

    * Funci3n CategoryDatatest: se encarga de pasar o
    incrustar los datos en el panel de la grafica y los ejes.
    */
    private CategoryDataset createDataset(float p[],float
d[],int orden[]) {
        // numero de barras
        String series1 = "Translocati3n";
        String series2 = "Residual";
        // categorias
        String type1 = "FR"+String.valueOf(orden[0]+1);

        String type2 = "FR"+String.valueOf(orden[1]+1);
        String type3 = "FR"+String.valueOf(orden[2]+1);
        String type4 = "FR"+String.valueOf(orden[3]+1);
        String type5 = "FR"+String.valueOf(orden[4]+1);
        String type6 = "FR"+String.valueOf(orden[5]+1);
        String type7 = "FR"+String.valueOf(orden[6]+1);
        String type8 = "FR"+String.valueOf(orden[7]+1);
        String type9 = "FR"+String.valueOf(orden[8]+1);
        String type10 = "FR"+String.valueOf(orden[9]+1);
        String type11 =
"FR"+String.valueOf(orden[10]+1);
        String type12 =
"FR"+String.valueOf(orden[11]+1);
        String type13 =
"FR"+String.valueOf(orden[12]+1);
        String type14 =
"FR"+String.valueOf(orden[13]+1);
        String type15 =
"FR"+String.valueOf(orden[14]+1);
        String type16 =
"FR"+String.valueOf(orden[15]+1);
        String type17 =
"FR"+String.valueOf(orden[16]+1);
        String type18 =
"FR"+String.valueOf(orden[17]+1);
        String type19 =
"FR"+String.valueOf(orden[18]+1);
        String type20 =
"FR"+String.valueOf(orden[19]+1);
        String type21 =
"FR"+String.valueOf(orden[20]+1);
        String type22 =
"FR"+String.valueOf(orden[21]+1);
        String type23 =
"FR"+String.valueOf(orden[22]+1);
        String type24 =
"FR"+String.valueOf(orden[23]+1);
        String type25 =
"FR"+String.valueOf(orden[24]+1);
        String type26 =
"FR"+String.valueOf(orden[25]+1);
        String type27 =
"FR"+String.valueOf(orden[26]+1);
        String type28 =
"FR"+String.valueOf(orden[27]+1);

```

```

        String type29 =
"FR"+String.valueOf(orden[28]+1);
        String type30 =
"FR"+String.valueOf(orden[29]+1);
        String type31 =
"FR"+String.valueOf(orden[30]+1);
        String type32 =
"FR"+String.valueOf(orden[31]+1);
        String type33 =
"FR"+String.valueOf(orden[32]+1);
        String type34 =
"FR"+String.valueOf(orden[33]+1);
        String type35 =
"FR"+String.valueOf(orden[34]+1);
        String type36 =
"FR"+String.valueOf(orden[35]+1);
        String type37 =
"FR"+String.valueOf(orden[36]+1);
        String type38 =
"FR"+String.valueOf(orden[37]+1);
        String type39 =
"FR"+String.valueOf(orden[38]+1);
        String type40 =
"FR"+String.valueOf(orden[39]+1);
        String type41 =
"FR"+String.valueOf(orden[40]+1);
        String type42 =
"FR"+String.valueOf(orden[41]+1);
        String type43 =
"FR"+String.valueOf(orden[42]+1);
        String type44 =
"FR"+String.valueOf(orden[43]+1);
        String type45 =
"FR"+String.valueOf(orden[44]+1);
        String type46 =
"FR"+String.valueOf(orden[45]+1);
        String type47 =
"FR"+String.valueOf(orden[46]+1);
        String type48 =
"FR"+String.valueOf(orden[47]+1);
        String type49 =
"FR"+String.valueOf(orden[48]+1);
        String type50 =
"FR"+String.valueOf(orden[49]+1);
        String type51 =
"FR"+String.valueOf(orden[50]+1);
        String type52 =
"FR"+String.valueOf(orden[51]+1);
        String type53 =
"FR"+String.valueOf(orden[52]+1);
        String type54 =
"FR"+String.valueOf(orden[53]+1);
        String type55 =
"FR"+String.valueOf(orden[54]+1);
        String type56 =
"FR"+String.valueOf(orden[55]+1);

```


ANEXO

```
    if (d[22] != 0) {
    dataset.addValue(p[22], series1, type23);
    if (d[23] != 0) {
    dataset.addValue(p[23], series1, type24);
    if (d[24] != 0) {
    dataset.addValue(p[24], series1, type25);
    if (d[25] != 0) {
    dataset.addValue(p[25], series1, type26);
    if (d[26] != 0) {
    dataset.addValue(p[26], series1, type27);
    if (d[27] != 0) {
    dataset.addValue(p[27], series1, type28);
    if (d[28] != 0) {
    dataset.addValue(p[28], series1, type29);
    if (d[29] != 0) {
    dataset.addValue(p[29], series1, type30);
    if (d[30] != 0) {
    dataset.addValue(p[30], series1, type31);
    if (d[31] != 0) {
    dataset.addValue(p[31], series1, type32);
    if (d[32] != 0) {
    dataset.addValue(p[32], series1, type33);
    if (d[33] != 0) {
    dataset.addValue(p[33], series1, type34);
    if (d[34] != 0) {
    dataset.addValue(p[34], series1, type35);
    if (d[35] != 0) {
    dataset.addValue(p[35], series1, type36);
    if (d[36] != 0) {
    dataset.addValue(p[36], series1, type37);
    if (d[37] != 0) {
    dataset.addValue(p[37], series1, type38);
    if (d[38] != 0) {
    dataset.addValue(p[38], series1, type39);
        if (d[39] != 0) {
    dataset.addValue(p[39], series1, type40);
        if (d[40] != 0) {
            dataset.addValue(p[40], series1,
type41);
    if (d[41] != 0) {
    dataset.addValue(p[41], series1, type42);
    if (d[42] != 0) {
    dataset.addValue(p[42], series1, type43);
    if (d[43] != 0) {
    dataset.addValue(p[43], series1, type44);
    if (d[44] != 0) {
    dataset.addValue(p[44], series1, type45);
    if (d[45] != 0) {
    dataset.addValue(p[45], series1, type46);
    if (d[46] != 0) {
    dataset.addValue(p[46], series1, type47);
    if (d[47] != 0) {
    dataset.addValue(p[47], series1, type48);
    if (d[48] != 0) {
    dataset.addValue(p[48], series1, type49);
    if (d[49] != 0) {
```


ANEXO

```
dataset.addValue(d[15],series2,type16);
if (d[16] != 0) {
dataset.addValue(d[16],series2,type17);
if (d[17] != 0) {
dataset.addValue(d[17],    series2,type18);
if (d[18] != 0) {
dataset.addValue(d[18],series2, type19);
    if (d[19] != 0) {
dataset.addValue(d[19],series2, type20);
    if (d[20] != 0) {
dataset.addValue(d[20], series2, type21);
if (d[21] != 0) {
dataset.addValue(d[21], series2, type22);
if (d[22] != 0) {
dataset.addValue(d[22], series2, type23);
if (d[23] != 0) {
dataset.addValue(d[23], series2, type24);
if (d[24] != 0) {
dataset.addValue(d[24], series2, type25);
if (d[25] != 0) {
dataset.addValue(d[25], series2, type26);
if (d[26] != 0) {
dataset.addValue(d[26], series2, type27);
if (d[27] != 0) {
dataset.addValue(d[27], series2, type28);
if (d[28] != 0) {
dataset.addValue(d[28], series2,type29);
if (d[29] != 0) {
dataset.addValue(d[29], series2,type30);
if (d[30] != 0) {
dataset.addValue(d[30],series2, type31);
if (d[31] != 0) {
dataset.addValue(d[31],series2,type32);
if (d[32] != 0) {
dataset.addValue(d[32],series2,type33);
if (d[33] != 0) {
dataset.addValue(d[33],series2,type34);
if (d[34] != 0) {
dataset.addValue(d[34],series2,type35);
if (d[35] != 0) {
dataset.addValue(d[35],series2,type36);
if (d[36] != 0) {
dataset.addValue(d[36],series2,type37);
if (d[37] != 0) {
dataset.addValue(d[37],    series2,type38);
if (d[38] != 0) {
dataset.addValue(d[38],series2, type39);
    if (d[39] != 0) {
dataset.addValue(d[39],series2, type40);
    if (d[40] != 0) {
dataset.addValue(d[40], series2, type41);
if (d[41] != 0) {
dataset.addValue(d[41], series2, type42);
if (d[42] != 0) {
dataset.addValue(d[42], series2, type43);
if (d[43] != 0) {
```



```

false // URLs?
);
// color de fondo.
chart.setBackgroundPaint(new Color(0xBBBDD));
//categoria del plot
CategoryPlot plot = chart.getCategoryPlot();

NumberAxis rangeAxis = (NumberAxis) plot.getRangeAxis();
rangeAxis.setStandardTickUnits(NumberAxis.createIntegerTick
Units());

BarRenderer renderer = (BarRenderer) plot.getRenderer();

renderer.setItemMargin(0.0);

GradientPaint gp0 = new GradientPaint(
0.0f, 0.0f, Color.red,
0.0f, 0.0f, Color.lightGray
);
GradientPaint gp1 = new GradientPaint(
0.0f, 0.0f, Color.yellow,
0.0f, 0.0f, Color.lightGray
);
renderer.setSeriesPaint(0, gp0);
renderer.setSeriesPaint(1, gp1);
return chart;
}
}

```

DIAGRAMA BARRAS I

```

package org.anesthsom.anesthsom.modelo;

import java.awt.Color;
import java.awt.GradientPaint;

import javax.swing.JFrame;

import org.jfree.chart.ChartFactory;
import org.jfree.chart.ChartPanel;
import org.jfree.chart.JFreeChart;
import org.jfree.chart.axis.NumberAxis;
import org.jfree.chart.plot.CategoryPlot;
import org.jfree.chart.plot.PlotOrientation;
import org.jfree.chart.renderer.category.BarRenderer;
import org.jfree.data.category.CategoryDataset;
import org.jfree.data.category.DefaultCategoryDataset;

/**
 * @author Pablo Torner Pelay
 *
 */
public class DiagramaBarras1 extends JFrame {
    private static final long serialVersionUID = 1L;
    /**
     * @param title
     * @param p
     * @param d
     * @param orden
     *
     * Constructor: Se encarga de crear un grafico de
     barras resferente a las lesiones, a partir de:
     * El titulo de la ventana, el residual y la
     diferencia de posicion del patri;zn respecto de la empresa y el
     orden del patron.
     */
    public DiagramaBarras1(String title,float p[],float
d[],int orden[]) {
        super(title);
        CategoryDataset dataset =
createDataset(p,d,orden);
        JFreeChart chart = createChart(dataset);
        // añ;adir el chart al panel
        ChartPanel chartPanel = new ChartPanel(chart);
        chartPanel.setPreferredSize(new
java.awt.Dimension(500, 270));
        setContentPane(chartPanel);
    }
    /**
     * @param p
     * @param d
     * @param orden

```

ANEXO

```

* @return
*
*Funci3n CategoryDatatest: se encarga de pasar o
incrustar los datos en el panel de la grafica y los ejes.
*/
private CategoryDataset createDataset(float p[],float
d[],int orden[]) {
// numero de barras
String series1 = "Translocati3n";
String series2 = "Residual";
// categorias
String type1 = "C"+String.valueOf(orden[0]+1);
String type2 = "C"+String.valueOf(orden[1]+1);
String type3 = "C"+String.valueOf(orden[2]+1);
String type4 = "C"+String.valueOf(orden[3]+1);
String type5 = "C"+String.valueOf(orden[4]+1);
String type6 = "C"+String.valueOf(orden[5]+1);
String type7 = "C"+String.valueOf(orden[6]+1);
String type8 = "C"+String.valueOf(orden[7]+1);
String type9 = "C"+String.valueOf(orden[8]+1);
String type10 = "C"+String.valueOf(orden[9]+1);
String type11 = "C"+String.valueOf(orden[10]+1);
String type12 = "C"+String.valueOf(orden[11]+1);
String type13 = "C"+String.valueOf(orden[12]+1);
String type14 = "C"+String.valueOf(orden[13]+1);
String type15 = "C"+String.valueOf(orden[14]+1);
String type16 = "C"+String.valueOf(orden[15]+1);
String type17 = "C"+String.valueOf(orden[16]+1);
String type18 = "C"+String.valueOf(orden[17]+1);
String type19 = "C"+String.valueOf(orden[18]+1);
String type20 = "C"+String.valueOf(orden[19]+1);

//Datatest
DefaultCategoryDataset dataset = new
DefaultCategoryDataset();
if (d[0] != 0) {
dataset.addValue(p[0], series1, type1);
if (d[1] != 0) {
dataset.addValue(p[1], series1,
type2);
if (d[2] != 0) {
dataset.addValue(p[2], series1,
type3);
if (d[3] != 0) {
dataset.addValue(p[3],
series1, type4);
if (d[4] != 0) {
dataset.addValue(p[4], series1, type5);
if (d[5] != 0) {
dataset.addValue(p[5], series1, type6);
if (d[6] != 0) {
dataset.addValue(p[6], series1, type7);

```

ANEXO

```

                                                                    if (d[7]
!= 0) {
    dataset.addValue(p[7], series1, type8);
                                                                    if
(d[8] != 0) {
    dataset.addValue(p[8], series1,
        type9);
    if (d[9] != 0) {
    dataset.addValue(p[9], series1,
        type10);
    if (d[10] != 0) {
        dataset.addValue(p[10],
            series1, type11);
    if (d[11] != 0) {
        dataset
            .addValue(
                p[11],
                series1,
                type12);
    if (d[12] != 0) {
        dataset.addValue(
            p[12],
            series1,
            type13);
    if (d[13] != 0) {
        dataset
            .addValue(
                p[13],
                series1,
                type14);

```

ANEXO

```
if (d[14] != 0) {  
    dataset  
        .addValue(  
            p[14],  
            series1,  
            type15);  
    if (d[15] != 0) {  
        dataset  
            .addValue(  
p[15],  
series1,  
type16);  
        if (d[16] != 0) {  
            dataset  
                .addValue(  
p[16],  
series1,  
type17);  
            if (d[17] != 0) {  
                dataset  
                    .addValue(  
p[17],  
series1,
```


ANEXO

```
        if (d[0] != 0) {
            dataset.addValue(d[0], series2, type1);
            if (d[1] != 0) {
                dataset.addValue(d[1], series2,
type2);
                if (d[2] != 0) {
                    dataset.addValue(d[2], series2,
type3);
                    if (d[3] != 0) {
                        dataset.addValue(d[3],
series2, type4);
                        if (d[4] != 0) {
                            dataset.addValue(d[4], series2, type5);
                            if (d[5] != 0) {
                                dataset.addValue(d[5], series2, type6);
                                if (d[6] != 0) {
                                    dataset.addValue(d[6], series2, type7);
                                    if (d[7]
!= 0) {
                                        dataset.addValue(d[7], series2, type8);
                                        if
(d[8] != 0) {
                                            dataset.addValue(d[8], series2,
                                                type9);
                                            if (d[9] != 0) {
                                                dataset.addValue(d[9], series2,
                                                    type10);
                                                if (d[10] != 0) {
                                                    dataset.addValue(d[10],
                                                        series2, type11);
                                                    if (d[11] != 0) {
                                                        dataset
                                                            .addValue(
                                                                d[11],
                                                                series2,
                                                                type12);
                                                        if (d[12] != 0) {
```


ANEXO

```
dataset.addValue(  
    d[12],  
    series2,  
    type13);  
if (d[13] != 0) {  
    dataset  
        .addValue(  
            d[13],  
            series2,  
            type14);  
    if (d[14] != 0) {  
        dataset  
            .addValue(  
                d[14],  
                series2,  
                type15);  
        if (d[15] != 0) {  
            dataset  
                .addValue(  
                    d[15],  
                    series2,  
                    type16);  
            if (d[16] != 0) {  
                dataset  
                    .addValue(  
                        d[16],
```

```
series2,

type17);

if (d[17] != 0) {
    dataset

.addValue(

d[17],

series2,

type18);
if (d[18] != 0)
{
    dataset

.addValue(

    d[18],

    series2,

    type19);
if (d[19] != 0) {

dataset.addValue(d[19],series2,type20);

}

}

}

}
```



```
0.0f, 0.0f, Color.lightGray
);
GradientPaint gp1 = new GradientPaint(
0.0f, 0.0f, Color.yellow,
0.0f, 0.0f, Color.lightGray
);
renderer.setSeriesPaint(0, gp0);
renderer.setSeriesPaint(1, gp1);
return chart;
}
}
```

DIBUJAR BARRAS

```

package org.anesthsom.anesthsom.modelo;

import org.jfree.ui.RefineryUtilities;

/**
 * @author Pablo Torner
 *
 *
 *Clase que deriva de la clase PlantillaMetodo
 */
public class DibujarBarras extends PlantillaMetodo {
    private static Contenedor empresa;
    private static Contenedor general;
    protected void pasol(){
        int r[]=empresa.getP().getRiesgo();
        int l[]=empresa.getP().getLesiones();
        int l1[]=general.getP().getLesiones();
        int r1[]=general.getP().getRiesgo();
        float x[] = new float[60];
        float y[] = new float[60];
        float z[] = new float[20];
        float w[] = new float[20];
        float p[] = new float[60];
        float q[] = new float[60];
        float m[] = new float[20];
        float n[] = new float[20];
        float resr[] = new float[60];
        float resl[] = new float[20];
        int orden[] = new int[60];
        int orden1[] = new int[20];

        r1=PonerCero(r1,r);
        l1=PonerCero(l1,l);
        int suma =0;
        for (int i=0;i<r.length;i++){
            suma=suma+r[i];
        }
        int suma1 =0;
        for (int i=0;i<r1.length;i++){
            suma1=suma1+r1[i];
        }
        int suma2 = 0;
        for (int i=0;i<l.length;i++){
            suma2=suma2+l[i];
        }
        int suma3=0;
        for (int i=0;i<l1.length;i++){
            suma3=suma3+l1[i];
        }
    }
}

```

```

        }
        orden=or(r1,60);
        orden=ord(orden,60);
        x=ordenar(orden,r,60);
        y=ordenar(orden,r1,60);
        x=orde(x,60);
        y=orde(y,60);
        x=comparar(x,y,60);
        p=ordenar(orden,r,60);
        q=ordenar(orden,r1,60);
        resr=cresidual(p,q,suma,suma1,60);

        orden1=or(l1,20);
        orden1=ord(orden1,20);
        z=ordenar(orden1,l,20);
        w=ordenar(orden1,l1,20);
        z=orde(z,20);
        w=orde(w,20);
        z=comparar(z,w,20);
        m=ordenar(orden1,l,20);
        n=ordenar(orden1,l1,20);
        resl=cresidual(m,n,suma2,suma3,20);

        DiagramaBarras demo2 = new DiagramaBarras("Risk
factor bar chart",x,resr,orden);

        demo2.setDefaultCloseOperation(DiagramaBarras.DISPOSE_ON_CL
OSE);
        demo2.pack();
        RefineryUtilities.centerFrameOnScreen(demo2);
        demo2.setVisible(true);
        DiagramaBarras1 demo3 = new DiagramaBarras1("Injuries
bar chart",z,resl,orden1);

        demo3.setDefaultCloseOperation(DiagramaBarras.DISPOSE_ON_CL
OSE);
        demo3.pack();
        RefineryUtilities.centerFrameOnScreen(demo3);
        demo3.setVisible(true);

    }
    /* (non-Javadoc)
     * @see
modelo.PlantillaMetodo#setEmpresa(modelo1.Contenedor)
     */
    public void setEmpresa(Contenedor contenedor) {
        empresa=new Contenedor();
        empresa = contenedor;
    }
    /* (non-Javadoc)
     * @see
modelo.PlantillaMetodo#setGeneral(modelo1.Contenedor)
     */
    @Override
    public void setGeneral(Contenedor contenedor) {

```

```
        general=new Contenedor();
        general = contenedor;

    }
    /* (non-Javadoc)
     * @see modelo.PlantillaMetodo#getEmpresa()
     */
    @Override
    public Contenedor getEmpresa() {
        return empresa;

    }
    /* (non-Javadoc)
     * @see modelo.PlantillaMetodo#getGeneral()
     */
    @Override
    public Contenedor getGeneral() {
        return general;

    }

}
```

DIBUJAR POLIGONAL

```

package org.anesthsom.anesthsom.modelo;

import org.jfree.ui.RefineryUtilities;

/**
 * @author Pablo Torner
 *
 *
 *Clase que deriva de la clase PlantillaMetodo
 */
public class DibujarPoligonal extends PlantillaMetodo {
    private static Contenedor empresa;
    private static Contenedor general;

    /* (non-Javadoc)
     * @see modelo.PlantillaMetodo#pasol()
     */
    protected void pasol(){
        float suma=0;
        float suma1=0;
        float suma2=0;
        float suma3=0;

        int r[]=empresa.getP().getRiesgo();
        int l[]=empresa.getP().getLesiones();
        int l1[]=general.getP().getLesiones();
        int r1[]=general.getP().getRiesgo();
        float riesgo[]=new float[60];
        float riesgo1[]=new float[60];
        float lesion[]=new float[20];
        float lesion1[]=new float[20];
        int orden[]=new int[60];
        int orden1[]=new int[20];
        r1=PonerCero(r1,r);
        l1=PonerCero(l1,l);
        for (int i=0;i<r.length;i++){
            suma=suma+r[i];
        }
        for (int i=0;i<r1.length;i++){
            suma1=suma1+r1[i];
        }
        for (int i=0;i<l.length;i++){
            suma2=suma2+l[i];
        }
        for (int i=0;i<l1.length;i++){
            suma3=suma3+l1[i];
        }
        orden=or(r1,60);
        orden=ord(orden,60);
        riesgo=ordenar(orden,r,60);
    }
}

```



```

riesgol=ordenar(orden,r1,60);
riesgo=ovector(riesgo,suma,60);
riesgol=ovector(riesgol,suma1,60);
Poligonal demo = new Poligonal("Risk polygonal
graphics",riesgo,riesgol,orden);
demo.setDefaultCloseOperation(Poligonal.DISPOSE_ON_CLOSE);
demo.pack();
RefineryUtilities.centerFrameOnScreen(demo);
demo.setVisible(true);

orden1=or(l1,20);
orden1=ord(orden1,20);
lesion=ordenar(orden1,l,20);
lesion1=ordenar(orden1,l1,20);
lesion=ovector(lesion,suma2,20);
lesion1=ovector(lesion1,suma3,20);
Poligonal demol = new Poligonal("Injuries polygonal
graphics",lesion,lesion1,orden1);
demol.setDefaultCloseOperation(Poligonal.DISPOSE_ON_CLOSE);
demol.pack();
RefineryUtilities.centerFrameOnScreen(demol);
demol.setVisible(true);
}
/* (non-Javadoc)
 * @see modelo.PlantillaMetodo#setEmpresa(modelo1.Contenedor)
 */
@Override
public void setEmpresa(Contenedor contenedor) {
    empresa=new Contenedor();
    empresa = contenedor;
}
/* (non-Javadoc)
 * @see modelo.PlantillaMetodo#setGeneral(modelo1.Contenedor)
 */
@Override
public void setGeneral(Contenedor contenedor) {
    general=new Contenedor();
    general = contenedor;
}
/* (non-Javadoc)
 * @see modelo.PlantillaMetodo#getEmpresa()
 */
@Override
public Contenedor getEmpresa() {
    return empresa;
}
/* (non-Javadoc)
 * @see modelo.PlantillaMetodo#getGeneral()
 */
@Override
public Contenedor getGeneral() {
    return general;
}

```

}
}

INFORME

```

package org.anesthsom.anesthsom.modelo;

import com.lowagie.text.Document;
import com.lowagie.text.DocumentException;
import com.lowagie.text.Font;
import com.lowagie.text.Paragraph;
import com.lowagie.text.Phrase;
import com.lowagie.text.pdf.PdfPTable;
import com.lowagie.text.pdf.PdfPageEventHelper;

/**
 * @author Pablo Torner
 *
 * Clase encargada de crear los informes en pdf de la parte
 * referente a riesgos. extiende PdfPageEventHelper
 */
public class Informe extends PdfPageEventHelper{
    public PdfPTable table;
    /**
     * @param fase2
     * @param document
     * @param r
     * @param r1
     *
     * Procedimiento: Crea el pdf a partir de su titulo y los
     * vectores con las frecuencias relativas en porcentaje de las
     * lesiones.
     */
    public Informe(int[]fase2,Document document,double[]r,double
    r1[],int b){

        try{

            document.addTitle("Planning Risk Report");
            document.addAuthor("Metodos Omega Alfa");
            document.add(new Paragraph("Planning Risk Report
"+b));

            document.add(new Paragraph(" "));
            table = new PdfPTable(4);
            table.addCell("");
            table.addCell("National pattern");
            table.addCell("Pattern");
            table.addCell("Company");
            for(int i=0;i<fase2.length;i++){

```

ANEXO

```

        if((r1[fase2[i]-1])!=0){
            table.addCell("FR"+fase2[i]);
            float a=(float)(frecuenciaPN(fase2[i]));

            table.addCell(""+a+" %");

            table.addCell(""+r[fase2[i]-1]+" %");

            table.addCell(""+r1[fase2[i]-1]+" %");
        }
        document.add(table);
        document.add(new Paragraph(" "));
        document.add(new Paragraph(" "));
        document.add(new Paragraph(" "));
        document.add(new Paragraph(" "));

        for(int j=0;j<fase2.length;j++){
            if((r1[fase2[j]-1])!=0){
                textoReferencial(fase2[j],document);
            }
        }

    }catch(Exception e){}

}
/**
 * @param i
 * @param document
 * @throws DocumentException
 *
 * Procedimiento que añade el texto referencia a cada Riesgo.
 */
private void textoReferencial(int i,Document document) throws
DocumentException {
    //negro
    Font fuente0=new Font(Font.COURIER,Font.DEFAULTSIZE);
    fuente0.setColor(0x00, 0x00, 0x00);
    //rojo
    Font fuente1=new Font(Font.COURIER,20,Font.ITALIC);
    fuente1.setColor(0xff, 0x66, 0x00);
    //amarillo
    Font fuente2=new Font(Font.COURIER,20,Font.ITALIC);
    fuente2.setColor(0xff, 0xcc, 0x00);
    Font fuente3=new Font(Font.COURIER,20,Font.ITALIC);
    fuente3.setColor(0x00, 0x66, 0x00);
    switch(i){
        case 1:

            Phrase frase1= new Phrase("FR"+i+": Falls at
different level",fuente1);

            document.add(new Paragraph(frase1));
            document.add(new Paragraph(" "));

```

ANEXO

```
document.add(new Paragraph("Risk included in the
group known as yellow, whose characteristics are: medium-high
frequency, low diversity and high meaningful peer relationships,
a very risk associated with injury and conversely.",fuente0));
document.add(new Paragraph(" "));
break;

case 2:
    Phrase frase2= new Phrase("FR"+i+": Falls to
same level",fuente2);
document.add(new Paragraph(frase2));
document.add(new Paragraph(" "));
document.add(new Paragraph("Risk included in the
group known as yellow, whose characteristics are: medium-high
frequency, low diversity and high meaningful peer relationships,
a very risk associated with injury and conversely.",fuente0));
document.add(new Paragraph(" "));
break;

case 3:
    Phrase frase3= new Phrase("FR"+i+": Object
Falls crash",fuentel);

document.add(new Paragraph(frase3));
document.add(new Paragraph(" "));
document.add(new Paragraph("Risk included in the
group called Red, whose characteristics are: medium-high
frequency and diversity moderately significant. Association are
characteristic of the individual and the
environment.",fuente0));
document.add(new Paragraph(" "));
break;

case 4:
    Phrase frase4= new Phrase("FR"+i+": Fall of
object manipulation",fuentel);

document.add(new Paragraph(frase4));
document.add(new Paragraph(" "));
document.add(new Paragraph("Risk included in the
group called Red, whose characteristics are: medium-high
frequency and diversity moderately significant. Association are
characteristic of the individual and the
environment.",fuente0));
document.add(new Paragraph(" "));
break;

case 5:
    Phrase frase5= new Phrase("FR"+i+": Falling
objects detached",fuentel);

document.add(new Paragraph(frase5));
document.add(new Paragraph(" "));
document.add(new Paragraph("Risk included in the
group called Red, whose characteristics are: medium-high
frequency and diversity moderately significant. Association are
characteristic of the individual and the
environment.",fuente0));
document.add(new Paragraph(" "));
break;
```

ANEXO

```
case 6:
    Phrase frase6= new Phrase("FR"+i+": Stepping
of objects",fuente2);

    document.add(new Paragraph(frase6));
    document.add(new Paragraph(" "));
    document.add(new Paragraph("Risk included in the
group known as yellow, whose characteristics are: medium-high
frequency, low diversity and high meaningful peer relationships,
a very risk associated with injury and conversely.",fuente0));
    document.add(new Paragraph(" "));
    break;

case 7:
    Phrase frase7= new Phrase("FR"+i+": Collisions
with stationary objects",fuente1);

    document.add(new Paragraph(frase7));
    document.add(new Paragraph(" "));
    document.add(new Paragraph("Risk included in the
group called Red, whose characteristics are: medium-high
frequency and diversity moderately significant. Association are
characteristic of the individual and the
environment.",fuente0));
    document.add(new Paragraph(" "));
    break;

case 8:
    Phrase frase8= new Phrase("FR"+i+": Collisions
with moving objects",fuente1);

    document.add(new Paragraph(frase8));
    document.add(new Paragraph(" "));
    document.add(new Paragraph("Risk included in the
group called Red, whose characteristics are: medium-high
frequency and diversity moderately significant. Association are
characteristic of the individual and the environment.
",fuente0));
    document.add(new Paragraph(" "));
    break;

case 9:
    Phrase frase9= new Phrase("FR"+i+": Bruises,
cuts with objects or tools",fuente1);
    document.add(new Paragraph(frase9));
    document.add(new Paragraph(" "));
    document.add(new Paragraph("Risk included in the
group called Red, whose characteristics are: medium-high
frequency and diversity moderately significant. Association are
characteristic of the individual and the
environment.",fuente0));
    document.add(new Paragraph(" "));
    break;

case 10:
    Phrase frase10= new Phrase("FR"+i+":
Projection of fragments or particles",fuente3);
    document.add(new Paragraph(frase10));
    document.add(new Paragraph(" "));
```

ANEXO

```
document.add(new Paragraph("Risk included in the
group called green, whose characteristics are minimal risks
moderately significant frequency and diversity. Are
characteristic of the influence of environment.",fuente0));
document.add(new Paragraph(" "));
break;

case 11:
Phrase frasel1= new Phrase("FR"+i+":
Entrapment by or between objects",fuente1);
document.add(new Paragraph(frasel1));
document.add(new Paragraph(" "));
document.add(new Paragraph("Risk included in the
group called Red, whose characteristics are: medium-high
frequency and diversity moderately significant. Association are
characteristic of the individual and the
environment.",fuente0));
document.add(new Paragraph(" "));
break;

case 12:
Phrase frasel2= new Phrase("FR"+i+": Mobile
machinery accidents",fuente1);
document.add(new Paragraph(frasel2));
document.add(new Paragraph(" "));
document.add(new Paragraph("Risk included in the
group called Red, whose characteristics are: medium-high
frequency and diversity moderately significant. Association are
characteristic of the individual and the
environment.",fuente0));
document.add(new Paragraph(" "));
break;

case 13:
Phrase frasel3= new Phrase("FR"+i+":
Overstrain",fuente2);
document.add(new Paragraph(frasel3));
document.add(new Paragraph(" "));
document.add(new Paragraph("Risk included in the
group known as yellow, whose characteristics are: medium-high
frequency, low diversity and high meaningful peer relationships,
a very risk associated with injury and conversely.",fuente0));
document.add(new Paragraph(" "));
break;

case 14:
Phrase frasel4= new Phrase("FR"+i+":
Exhibition and / or thermal contacts",fuente3);
document.add(new Paragraph(frasel4));
document.add(new Paragraph(" "));
document.add(new Paragraph("RRisk included in
the group called green, whose characteristics are minimal risks
moderately significant frequency and diversity. Are
characteristic of the influence of environment.",fuente0));
document.add(new Paragraph(" "));
break;

case 15:
Phrase frasel5= new Phrase("FR"+i+":
Electrical contacts",fuente3);
document.add(new Paragraph(frasel5));
```

ANEXO

```
        document.add(new Paragraph(" "));
        document.add(new Paragraph("Risk included in the
group called green, whose characteristics are minimal risks
moderately significant frequency and diversity. Are
characteristic of the influence of environment.",fuente0));
        document.add(new Paragraph(" "));
        break;
    case 16:
        Phrase frasel6= new Phrase("FR"+i+":
Interaction with aggressive substances",fuente3);

        document.add(new Paragraph(frasel6));
        document.add(new Paragraph(" "));
        document.add(new Paragraph("Risk included in the
group called green, whose characteristics are minimal risks
moderately significant frequency and diversity. Are
characteristic of the influence of environment.",fuente0));
        document.add(new Paragraph(" "));
        break;
    case 17:
        Phrase frasel7= new Phrase("FR"+i+":  Radiation
exposure",fuente3);
        document.add(new Paragraph(frasel7));
        document.add(new Paragraph(" "));
        document.add(new Paragraph("Risk included in the
group called green, whose characteristics are minimal risks
moderately significant frequency and diversity. Are
characteristic of the influence of environment.",fuente0));
        document.add(new Paragraph(" "));
        break;
    case 18:
        Phrase frasel8= new Phrase("FR"+i+":
Explosions and fires",fuente3);
        document.add(new Paragraph(frasel8));
        document.add(new Paragraph(" "));
        document.add(new Paragraph("Risk included in the
group called green, whose characteristics are minimal risks
moderately significant frequency and diversity. Are
characteristic of the influence of environment.",fuente0));
        document.add(new Paragraph(" "));
        break;
    case 19:
        Phrase frasel9= new Phrase("FR"+i+":  Accidents
caused by living beings",fuente1);
        document.add(new Paragraph(frasel9));
        document.add(new Paragraph(" "));
        document.add(new Paragraph("Risk included in the
group called Red, whose characteristics are: medium-high
frequency and diversity moderately significant. Association are
characteristic of the individual and the
environment.",fuente0));
        document.add(new Paragraph(" "));
        break;
    }
}
```



```

/**
 * @param i
 * @return
 *
 * Funcion doble que devuelve el porcentaje en el patrimonio
nacional de una lesiön dada, del aõ 2000-2005
 */
private double frecuenciaPN(int i) {
    double b=0;
    switch(i){
    case 1: b=8.7;break;
    case 2: b=9.8;break;
    case 3: b=1.4;break;
    case 4: b=6.5;break;
    case 5: b=0.7;break;
    case 6: b=5.8;break;
    case 7: b=4.2;break;
    case 8: b=2.4;break;
    case 9: b=19.2;break;
    case 10: b=6.3;break;
    case 11: b=6.7;break;
    case 12: b=2.7;break;
    case 13: b=22.4;break;
    case 14: b=1.0;break;
    case 15: b=0.3;break;
    case 16: b=1.1;break;
    case 17: b=0.1;break;
    case 18: b=0.2;break;
    case 19: b=0.6;break;
    }
    return b;
}
}

```

INFORME I

```

package org.anesthsom.anesthsom.modelo;

import com.lowagie.text.Document;
import com.lowagie.text.DocumentException;
import com.lowagie.text.Font;
import com.lowagie.text.Paragraph;
import com.lowagie.text.Phrase;
import com.lowagie.text.pdf.PdfPTable;
import com.lowagie.text.pdf.PdfPageEventHelper;

/**
 * @author Pablo Torner
 *
 * Clase encargada de crear los informes en pdf de la parte
 * referente a lesiones. extiende PdfPageEventHelper
 */
public class Informel extends PdfPageEventHelper{
    public PdfPTable table;
    /**
     * @param fase2
     * @param document
     * @param r
     * @param r1
     *
     * Procedimiento: Crea el pdf a partir de su titulo y los
     * vectores con las frecuencias relativas en porcentaje de las
     * lesiones.
     */
    public Informel(int[] fase2, Document document, double[] r, double
    r1[], int b){

        try{

            document.addTitle("Planning injuries report");
            document.addAuthor("Metodos Omega Alfa");
            document.add(new Paragraph("Planning injuries
report"+b));
            document.add(new Paragraph(" "));
            table = new PdfPTable(4);
            table.addCell("");
            table.addCell("Noational pattern");
            table.addCell("Pattern");
            table.addCell("Company");

```

```

        for(int i=0;i<fase2.length;i++){
            if((r1[fase2[i]-1])!=0){
                table.addCell("C"+fase2[i]);
                float a=(float)(frecuenciaPN(fase2[i]));

                table.addCell(""+a+" %");

                table.addCell(""+r[fase2[i]-1]+" %");

                table.addCell(""+r1[fase2[i]-1]+" %");
            }
            document.add(table);
            document.add(new Paragraph(" "));
            document.add(new Paragraph(" "));
            document.add(new Paragraph(" "));
            document.add(new Paragraph(" "));

            for(int j=0;j<fase2.length;j++){
                if((r1[fase2[j]-1])!=0){
                    textoReferencial(fase2[j],document);
                }
            }

        }catch(Exception e){}

    }
    /**
     * @param i
     * @param document
     * @throws DocumentException
     *
     * Procedimiento que añade el texto referencia a cada
     * Lesión.
     */
    private void textoReferencial(int i,Document document) throws
    DocumentException {
        //negro
        Font fuente0=new Font(Font.COURIER,Font.DEFAULTSIZE);
        fuente0.setColor(0x00, 0x00, 0x00);
        //rojo
        Font fuente1=new Font(Font.COURIER,20,Font.ITALIC);
        fuente1.setColor(0xff, 0x66, 0x00);
        //amarillo
        Font fuente2=new Font(Font.COURIER,20,Font.ITALIC);
        fuente2.setColor(0xff, 0xcc, 0x00);
        Font fuente3=new Font(Font.COURIER,20,Font.ITALIC);
        fuente3.setColor(0x00, 0x66, 0x00);
        switch(i){
            case 1:

                Phrase frase1= new Phrase("C"+i+":
                Fractures",fuente1);

```

```

        document.add(new Paragraph(frase1));
        document.add(new Paragraph(" "));
        document.add(new Paragraph("Injury included in
the group called Red, whose characteristics are: medium-high
frequency and diversity moderately significant. Association are
characteristic of the individual and the
environment.",fuente0));
        document.add(new Paragraph(" "));
        break;
    case 2:

        Phrase frase2= new Phrase("C"+i+":
Dislocations",fuente2);
        document.add(new Paragraph(frase2));
        document.add(new Paragraph(" "));
        document.add(new Paragraph("Injury included in
the group known as yellow, whose characteristics are: medium-
high frequency, low diversity and high meaningful peer
relationships, a lesion strongly associated with risk and
conversely.",fuente0));
        document.add(new Paragraph(" "));
        break;
    case 3:

        Phrase frase3= new Phrase("C"+i+":   Strains and
sprains",fuente2);

        document.add(new Paragraph(frase3));
        document.add(new Paragraph(" "));
        document.add(new Paragraph("Injury included in
the group known as yellow, whose characteristics are: medium-
high frequency, low diversity and high meaningful peer
relationships, a lesion strongly associated with risk and
conversely.",fuente0));
        document.add(new Paragraph(" "));
        break;
    case 4:

        Phrase frase4= new Phrase("C"+i+":
Lumbalgias",fuente2);

        document.add(new Paragraph(frase4));
        document.add(new Paragraph(" "));
        document.add(new Paragraph("Injury included in
the group known as yellow, whose characteristics are: medium-
high frequency, low diversity and high meaningful peer
relationships, a lesion strongly associated with risk and
conversely.",fuente0));
        document.add(new Paragraph(" "));
        break;
    case 5:

        Phrase frase5= new Phrase("C"+i+":   Slipped
discs",fuente2);

        document.add(new Paragraph(frase5));
        document.add(new Paragraph(" "));
        document.add(new Paragraph("Injury included in
the group known as yellow, whose characteristics are: medium-

```

```

high frequency, low diversity and high meaningful peer
relationships, a lesion strongly associated with risk and
conversely.",fuente0));
    document.add(new Paragraph(" "));
    break;
    case 6:
        Phrase frase6= new Phrase("C"+i+":  Shocks and
internal injuries",fuente2);

        document.add(new Paragraph(frase6));
        document.add(new Paragraph(" "));
        document.add(new Paragraph("Injury included in
the group known as yellow, whose characteristics are: medium-
high frequency, low diversity and high meaningful peer
relationships, a lesion strongly associated with risk and
conversely.",fuente0));
        document.add(new Paragraph(" "));
        break;
    case 7:
        Phrase frase7= new Phrase("C"+i+":  Amputation
and loss of the eyeball",fuente1);

        document.add(new Paragraph(frase7));
        document.add(new Paragraph(" "));
        document.add(new Paragraph("Injury included in
the group called Red, whose characteristics are: medium-high
frequency and diversity moderately significant. Association are
characteristic of the individual and the
environment.",fuente0));
        document.add(new Paragraph(" "));
        break;
    case 8:
        Phrase frase8= new Phrase("C"+i+":  Opend
wound. Other wounds",fuente1);

        document.add(new Paragraph(frase8));
        document.add(new Paragraph(" "));
        document.add(new Paragraph("Injury included in
the group called Red, whose characteristics are: medium-high
frequency and diversity moderately significant. Association are
characteristic of the individual and the
environment.",fuente0));
        document.add(new Paragraph(" "));
        break;
    case 9:
        Phrase frase9= new Phrase("C"+i+":  Superficial
Injuries",fuente1);
        document.add(new Paragraph(frase9));
        document.add(new Paragraph(" "));
        document.add(new Paragraph("Injury included in
the group called Red, whose characteristics are: medium-high
frequency and diversity moderately significant. Association are
characteristic of the individual and the
environment.",fuente0));
        document.add(new Paragraph(" "));

```

ANEXO

```
        break;
    case 10:
        Phrase frase10= new Phrase("C"+i+": Bruising
or crushing",fuentel);
        document.add(new Paragraph(frase10));
        document.add(new Paragraph(" "));
        document.add(new Paragraph("Injury included in
the group called Red, whose characteristics are: medium-high
frequency and diversity moderately significant. Association are
characteristic of the individual and the
environment.",fuente0));
        document.add(new Paragraph(" "));
        break;
    case 11:
        Phrase frase11= new Phrase("C"+i+": Foreign
bodies in eyes",fuente3);
        document.add(new Paragraph(frase11));
        document.add(new Paragraph(" "));
        document.add(new Paragraph("Injury included in
the group called green, whose characteristics are often minimal
injuries and moderately significant diversity. Are
characteristic of the influence of environment.",fuente0));
        document.add(new Paragraph(" "));
        break;
    case 12:
        Phrase frase12= new Phrase("C"+i+":
Conjunctivitis",fuente3);
        document.add(new Paragraph(frase12));
        document.add(new Paragraph(" "));
        document.add(new Paragraph("Injury included in
the group called green, whose characteristics are often minimal
injuries and moderately significant diversity. Are
characteristic of the influence of environment.",fuente0));
        document.add(new Paragraph(" "));
        break;
    case 13:
        Phrase frase13= new Phrase("C"+i+":
Burns",fuente3);
        document.add(new Paragraph(frase13));
        document.add(new Paragraph(" "));
        document.add(new Paragraph("Injury included in
the group called green, whose characteristics are often minimal
injuries and moderately significant diversity. Are
characteristic of the influence of environment.",fuente0));
        document.add(new Paragraph(" "));
        break;
    case 14:
        Phrase frase14= new Phrase("C"+i+": Poisoning
and intoxications",fuente3);
        document.add(new Paragraph(frase14));
        document.add(new Paragraph(" "));
        document.add(new Paragraph("Injury included in
the group called green, whose characteristics are often minimal
injuries and moderately significant diversity. Are
characteristic of the influence of environment.",fuente0));
        document.add(new Paragraph(" "));
```

```

        break;
    case 15:
        Phrase frasel5= new Phrase("C"+i+":
Environmental effects",fuente3);
        document.add(new Paragraph(frasel5));
        document.add(new Paragraph(" "));
        document.add(new Paragraph("Injury included in
the group called green, whose characteristics are often minimal
injuries and moderately significant diversity. Are
characteristic of the influence of environment.",fuente0));
        document.add(new Paragraph(" "));
        break;
    case 16:
        Phrase frasel6= new Phrase("C"+i+":
Asfixias",fuente3);

        document.add(new Paragraph(frasel6));
        document.add(new Paragraph(" "));
        document.add(new Paragraph("Injury included in
the group called green, whose characteristics are often minimal
injuries and moderately significant diversity. Are
characteristic of the influence of environment.",fuente0));
        document.add(new Paragraph(" "));
        break;
    case 17:
        Phrase frasel7= new Phrase("C"+i+":   Electrical
effects",fuente3);
        document.add(new Paragraph(frasel7));
        document.add(new Paragraph(" "));
        document.add(new Paragraph("Injury included in
the group called green, whose characteristics are often minimal
injuries and moderately significant diversity. Are
characteristic of the influence of environment.",fuente0));
        document.add(new Paragraph(" "));
        break;
    case 18:
        Phrase frasel8= new Phrase("C"+i+":   Radiation
effects",fuente3);
        document.add(new Paragraph(frasel8));
        document.add(new Paragraph(" "));
        document.add(new Paragraph("Injury included in
the group called green, whose characteristics are often minimal
injuries and moderately significant diversity. Are
characteristic of the influence of environment.",fuente0));
        document.add(new Paragraph(" "));
        break;
    case 19:
        Phrase frasel9= new Phrase("C"+i+":   Multiple
Injuries",fuente1);
        document.add(new Paragraph(frasel9));
        document.add(new Paragraph(" "));
        document.add(new Paragraph("Injury included in
the group called Red, whose characteristics are: medium-high
frequency and diversity moderately significant. Association are
characteristic of the individual and the
environment.",fuente0));

```

```

        document.add(new Paragraph(" "));
        break;
    }

}
/**
 * @param i
 * @return
 *
 * Funcion doble que devuelve el porcentaje en el patri;
nacional de una lesii;
nada, del ai;
no 2000-2005
 */
private double frecuenciaPN(int i) {
    double b=0;
    switch(i){
        case 1: b=9.3;break;
        case 2: b=2.2;break;
        case 3: b=27.0;break;
        case 4: b=9.2;break;
        case 5: b=0.1;break;
        case 6: b=1.5;break;
        case 7: b=0.4;break;
        case 8: b=17.4;break;
        case 9: b=4.9;break;
        case 10: b=18.7;break;
        case 11: b=5.2;break;
        case 12: b=0.7;break;
        case 13: b=2.1;break;
        case 14: b=0.1;break;
        case 15: b=0.0;break;
        case 16: b=0.0;break;
        case 17: b=0.1;break;
        case 18: b=0.0;break;
        case 19: b=1.0;break;
    }
    return b;
}
}

```


LISTA

```
package org.anesthsom.anesthsom.modelo;

import java.lang.annotation.ElementType;
import java.lang.annotation.Retention;
import java.lang.annotation.RetentionPolicy;
import java.lang.annotation.Target;

@Target(ElementType.FIELD)
@Retention(RetentionPolicy.RUNTIME)
public @interface Lista {

}
```

LISTA Alm1

```

package org.anesthsom.anesthsom.modelo;

import java.io.Serializable;
import java.util.ArrayList;
import java.util.Iterator;

public class ListaAlm1 implements Iterable<Alm1>, Serializable{
    private static final long serialVersionUID =
5934179848976902521L;
    private ArrayList<Alm1> lista;
    public ListaAlm1() {
        lista = new ArrayList<Alm1>();
    }
    /**
     * @param m
     * @return
     *
     * Descripcion: Se encarga de añadir nuevos Informes
Completos
     */
    public boolean add(Alm1 m) {

        lista.add(m);
        return true;
    }

    /**
     * @param m
     * @return
     *
     * Descripcion: Se encarga de eliminar Informes Completos
     */
    public boolean remove(Alm1 m) {
        return lista.remove(lista.indexOf(m))!=null;
    }

    /**
     * @see java.lang.Iterable#iterator()
     *
     * Descripcion: Sirve para recorrer la lista.
     */
    public Iterator<Alm1> iterator() {
        return lista.iterator();
    }
}

```

LISTA Alm2

```

package org.anesthsom.anesthsom.modelo;
import java.io.Serializable;
import java.util.ArrayList;
import java.util.Iterator;

public class ListaAlm2 implements Iterable<Alm2>, Serializable{
    private static final long serialVersionUID =
5934179848976902521L;
    private ArrayList<Alm2> lista;
    /**
     *
     */
    public ListaAlm2() {
        lista = new ArrayList<Alm2>();
    }
    /**
     * @param m
     * @return
     */
    public boolean add(Alm2 m) {

        lista.add(m);
        return true;
    }

    /**
     * @param m
     * @return
     */
    public boolean remove(Alm2 m) {
        return lista.remove(lista.indexOf(m)) != null;
    }

    /* (non-Javadoc)
     * @see java.lang.Iterable#iterator()
     */
    public Iterator<Alm2> iterator() {
        return lista.iterator();
    }
}

```

PLANTILLA METODO

```

package org.anesthsom.anesthsom.modelo;

/**
 * @author Pablo Torner Pelay
 *
 * Clase PlantillaMetodo: Se trata de una plantilla para crear
 los métodos de dibujar poligonal y dibujar diagrama de barras.
 * Implementa Command.
 */
public abstract class PlantillaMetodo implements Command {
/**
 *
 * Procedimiento execute(): Es una plantilla para crear un
 procedimiento execute.
 */
public void execute(){
    pasol();

}
/**
 * @param orden
 * @return
 *
 * Funcion int: Se encarga de escribir el orden de posiciones de
 un vector.
 */
public static int[] ord(int[] orden,int a) {
    int x[]=new int[a];
    for(int i=0;i<orden.length;i++){
        int k=0;
        while(orden[k]!=i){
            k++;
        }
        x[i]=k;
    }
    return x;
}

/**
 * @param orden
 * @param r1
 * @return
 *
 * Funcion float: Se encarga de ordenar un vector respecto de
 un orden dado por otro vector de enteros.
 */
public static float[] ordenar(int[] orden, int[] r1,int a) {
    float v[]=new float[a];

```

```

        for(int i=0;i<r1.length;i++){
            v[i]=r1[orden[i]];
        }
        return v;
    }

/**
 * @param r
 * @return
 *
 * Funcion int: Calcula el orden de un vector segun su
frecuencia absoluta
 */
public static int[] or(int[] r,int a) {
    int x[]=new int[a];
    for(int i=0;i<(r.length);i++){
        int contador=0;
        for(int j=0;j<(r.length);j++){
            if (r[i]<r[j]){
                contador++;
            }
        }
        for (int k=0;k<i;k++) {
            if (x[k]==contador){
                contador++;
            }
        }
        x[i]=contador;
    }
    return x;
}

/**
 * @param r
 * @param suma
 * @return
 *
 * Funcion float: Sirve para calcular la frecuencia relativa
de un vector de frecuencias absolutas.
 */
public static float[] ovector(float[] r, float suma,int a) {
    float v[]=new float[a];
    for(int i=0;i<v.length;i++){
        v[i]=r[i]/suma;
    }
    return v;
}

public static float[] orde(float[] r,int a) {
    float x[]=new float[a];

    for(int i=0;i<(r.length);i++){
        int contador=0;
        for(int j=0;j<(r.length);j++){
            if (r[i]<r[j]){
                contador++;
            }else if(r[i]==r[j]){
                if (i>j){

```

```

                                contador++;
                                }
                                }
                                }
                                x[i]=contador;
                                }
                                return x;
}
public static float factorial(int contador, int cont) {
    int colo=contador;
    float valor=0;
    float conta=0;
    for(int i=0;i<cont;i++){
        valor=valor+colo;
        colo=colo+1;
    }
    conta=cont;
    valor=valor/conta;

    return valor;
}
/**
 * @param x
 * @param y
 * @param suma
 * @param suma2
 * @return
 *
 * Función: float: Sirve para el calculo del residual de un
vector de dos vectores uno hace referencia a la empresa y otro
al patron.
 */
public static float[] cresidual(float x[],float y[],float
suma,float suma2,int a) {
    float cres[]=new float[a];
    float ob=0;
    float s=0;
    float es=0;
    float rt=0;
    float vr=0;
    float ra=0;
    for(int i=0;i<x.length;i++){
        ob=x[i];
        s=x[i]+y[i];
        es=suma*s/(suma+suma2);
        if(es>=5){
            rt=(float) ((ob-es)/(Math.sqrt(es)));
            vr=(1-(s/(suma+suma2)))*(1-(suma/(suma+suma2))));
            ra=(float) (rt/(Math.sqrt(vr)));
        }else{
            ra=(float) ((Math.sqrt(ob))+(Math.sqrt(ob+1))-
(Math.sqrt(4*es+1)));
        }
        cres[i]=ra;
    }
}

```

```

        return cres;
    }
/**
 * @param y
 * @param z
 * @return
 *
 * Función float: Compara dos vectores de reales.
 */
public static float[] comparar(float[] y, float[] z, int a) {
    float r[] = new float[a];
    for (int i = 0; i < y.length; i++) {
        r[i] = z[i] - y[i];
    }
    return r;
}
/**
 * @param r1
 * @param r
 * @return
 *
 * Función int: Pone a cero los elementos de un vector
respecto de otro.
 */
public int[] PonerCero(int[] r1, int[] r) {
    for (int i = 0; i < r1.length; i++) {
        if (r[i] == 0) {
            r1[i] = 0;
        }
    }
    return r1;
}

/**
 * @param contenedor
 *
 * Procedimiento abstracto que sirve para cambiar el patron
 */
public abstract void setGeneral(Contenedor contenedor);

/**
 * @param contenedor
 *
 * Procedimiento abstracto que sirve para cambiar la empresa
 */
public abstract void setEmpresa(Contenedor contenedor);
/**
 * @return
 *
 * Procedimiento abstracto para tomar el patron de la empresa en
una variable Contenedor
 */
public abstract Contenedor getEmpresa();

/**
 * @return

```

ANEXO

```
*
*
* Procedimiento abstracto para tomar el patron del patrón en
una variable Contenedor
*/
public abstract Contenedor getGeneral();

/**
*
*
* Sera distinto en cada heredero e indicara lo que se va a
ejecutar
*/
protected abstract void pasol();
}
```


POLIGONAL

```

package org.anesthsom.anesthsom.modelo;

import java.awt.Color;
//import java.awt.GradientPaint;

import javax.swing.JFrame;

import org.jfree.chart.ChartFactory;
import org.jfree.chart.ChartPanel;
import org.jfree.chart.JFreeChart;

import org.jfree.chart.plot.CategoryPlot;
import org.jfree.chart.plot.PlotOrientation;
import org.jfree.chart.renderer.category.LineAndShapeRenderer;
import org.jfree.data.category.CategoryDataset;
import org.jfree.data.category.DefaultCategoryDataset;

/**
 * @author Pablo Torner Pelay
 *
 */
public class Poligonal extends JFrame {
    /**
     *
     */
    private static final long serialVersionUID = 1L;

    /**
     * @param title
     * @param p
     * @param d
     * @param orden
     *
     * Constructor: De la clase poligonal pero de la parte
     referenete a los lesiones.
     * se le pasa el titulo de la ventana frecuencia relativa
     de la empresa, frecuencia relativa del patron y el orden.
     * Devuelve una grafica Poligonal realizada gracias a la
     libreria JFreeChart.
     */
    public Poligonal(String title, float p[], float d[], int
orden[]) {
        super(title);
        CategoryDataset dataset = createDataset(p,d,orden);
        JFreeChart chart = createChart(dataset);
        // add the chart to a panel...
        ChartPanel chartPanel = new ChartPanel(chart);
        chartPanel.setPreferredSize(new
java.awt.Dimension(500, 270));
        setContentPane(chartPanel);
    }
}

```

```

}

/**
 * @param p
 * @param d
 * @param orden
 * @return
 *
 *
 * Función CategoryDatatest: se encarga de pasar o
incrustar los datos en el panel de la grafica y los ejes.
 */
private CategoryDataset createDataset(float p[],float
d[],int orden[]) {
    // row keys...
    String series1 = "Company";
    String series2 = "Pattern";
    // column keys...
    String type1 = "C"+String.valueOf(orden[0]+1);
    String type2 = "C"+String.valueOf(orden[1]+1);
    String type3 = "C"+String.valueOf(orden[2]+1);
    String type4 = "C"+String.valueOf(orden[3]+1);
    String type5 = "C"+String.valueOf(orden[4]+1);
    String type6 = "C"+String.valueOf(orden[5]+1);
    String type7 = "C"+String.valueOf(orden[6]+1);
    String type8 = "C"+String.valueOf(orden[7]+1);
    String type9 = "C"+String.valueOf(orden[8]+1);
    String type10 = "C"+String.valueOf(orden[9]+1);
    String type11 = "C"+String.valueOf(orden[10]+1);
    String type12 = "C"+String.valueOf(orden[11]+1);
    String type13 = "C"+String.valueOf(orden[12]+1);
    String type14 = "C"+String.valueOf(orden[13]+1);
    String type15 = "C"+String.valueOf(orden[14]+1);
    String type16 = "C"+String.valueOf(orden[15]+1);
    String type17 = "C"+String.valueOf(orden[16]+1);
    String type18 = "C"+String.valueOf(orden[17]+1);
    String type19 = "C"+String.valueOf(orden[18]+1);
    String type20 = "C"+String.valueOf(orden[19]+1);
    // create the dataset...
    DefaultCategoryDataset dataset = new
DefaultCategoryDataset();
    if (d[0] != 0) {
        dataset.addValue(p[0], series1, type1);
        if (d[1] != 0) {
            dataset.addValue(p[1], series1, type2);
            if (d[2] != 0) {
                dataset.addValue(p[2], series1,
type3);
                if (d[3] != 0) {
                    dataset.addValue(p[3], series1,
type4);
                    if (d[4] != 0) {
                        dataset.addValue(p[4],
series1, type5);
                        if (d[5] != 0) {

```

ANEXO

```
dataset.addValue(p[5], series1, type6);
                                if (d[6] != 0) {

dataset.addValue(p[6], series1, type7);
                                if (d[7] != 0) {

dataset.addValue(p[7], series1, type8);
                                if (d[8]
!= 0) {

dataset.addValue(p[8], series1,
type9);
                                if
(d[9] != 0) {

dataset.addValue(p[9], series1,
type10);

if (d[10] != 0) {

dataset.addValue(p[10],
series1, type11);

if (d[11] != 0) {

dataset

.addValue(

p[11],
series1,
type12);

if (d[12] != 0) {

dataset.addValue(

p[12],
series1,
type13);

if (d[13] != 0) {

dataset

.addValue(

p[13],
```

ANEXO

```

                                series1,
                                type14);
if (d[14] != 0) {
    dataset
                                .addValue(
                                    p[14],
                                    series1,
                                    type15);
    if (d[15] != 0) {
        dataset
                                .addValue(
                                    p[15],
                                    series1,
                                    type16);
        if (d[16] != 0) {
            dataset
                                .addValue(
p[16],
                                series1,
                                type17);
            if (d[17] != 0) {
                dataset
                                .addValue(
p[17],
                                series1,
```


ANEXO

```
        if (d[1] != 0) {
            dataset.addValue(d[1], series2, type2);
            if (d[2] != 0) {
                dataset.addValue(d[2], series2,
type3);
                    if (d[3] != 0) {
                        dataset.addValue(d[3], series2,
type4);
                            if (d[4] != 0) {
                                dataset.addValue(d[4],
series2, type5);
                                    if (d[5] != 0) {
                                        dataset.addValue(d[5], series2, type6);
                                            if (d[6] != 0) {
                                                dataset.addValue(d[6], series2, type7);
                                                    if (d[7] != 0) {
                                                        if (d[8]
!= 0) {
                                                            dataset.addValue(d[8], series2,
type9);
                                                                if
(d[9] != 0) {
                                                                    dataset.addValue(d[9], series2,
type10);
                                                                        if (d[10] != 0) {
                                                                            dataset.addValue(d[10],
series2, type11);
                                                                                if (d[11] != 0) {
                                                                                    dataset
.addValue(
                                                                                        d[11],
                                                                                        series2,
                                                                                        type12);
                                                                                    if (d[12] != 0) {
                                                                                        dataset.addValue(
                                                                                            d[12],
```

ANEXO

```
        series2,
        type13);
if (d[13] != 0) {
    dataset
        .addValue(
            d[13],
            series2,
            type14);
if (d[14] != 0) {
    dataset
        .addValue(
            d[14],
            series2,
            type15);
if (d[15] != 0) {
    dataset
        .addValue(
            d[15],
            series2,
            type16);
if (d[16] != 0) {
    dataset
        .addValue(
d[16],
series2,
type17);
```


}

POLIGONAR

```

package org.anesthsom.anesthsom.modelo;

import java.awt.Color;

import javax.swing.JFrame;

import org.jfree.chart.ChartFactory;
import org.jfree.chart.ChartPanel;
import org.jfree.chart.JFreeChart;

import org.jfree.chart.plot.CategoryPlot;
import org.jfree.chart.plot.PlotOrientation;
import org.jfree.chart.renderer.category.LineAndShapeRenderer;
import org.jfree.data.category.CategoryDataset;
import org.jfree.data.category.DefaultCategoryDataset;

/**
 * @author Pablo Torner Pelay
 *
 */
public class Poligonar extends JFrame {
    /**
     *
     */
    private static final long serialVersionUID = 1L;

    /**
     * @param title
     * @param p
     * @param d
     * @param orden
     *
     * Constructor: De la clase poligonal pero de la
    parte referenete
     * a los riegos. se le pasa el titulo de la
    ventana frecuencia
     * relativa de la empresa, frecuencia relativa
    del patron y el
     * orden. Devuelve una grafica Poligonal
    realizada gracias a la
     * libreria JFreeeeChart.
     */
    public Poligonar(String title, float p[], float d[], int
orden[]) {
        super(title);
        CategoryDataset dataset = createDataset(p, d, orden);
        JFreeChart chart = createChart(dataset);
        // add the chart to a panel...
        ChartPanel chartPanel = new ChartPanel(chart);
        chartPanel.setPreferredSize(new
java.awt.Dimension(500, 270));

```

```

        setContentPane(chartPanel);
    }

    /**
     * @param p
     * @param d
     * @param orden
     * @return Función CategoryDatatest: se encarga de pasar
    o incrustar los
     *      datos en el panel de la grafica y los ejes.
     */
    private CategoryDataset createDataset(float p[], float d[],
    int orden[]) {
        // row keys...
        String series1 = "Company";
        String series2 = "Pattern";
        // column keys...
        String type1 = "FR" + String.valueOf(orden[0] + 1);
        String type2 = "FR" + String.valueOf(orden[1] + 1);
        String type3 = "FR" + String.valueOf(orden[2] + 1);
        String type4 = "FR" + String.valueOf(orden[3] + 1);
        String type5 = "FR" + String.valueOf(orden[4] + 1);
        String type6 = "FR" + String.valueOf(orden[5] + 1);
        String type7 = "FR" + String.valueOf(orden[6] + 1);
        String type8 = "FR" + String.valueOf(orden[7] + 1);
        String type9 = "FR" + String.valueOf(orden[8] + 1);
        String type10 = "FR" + String.valueOf(orden[9] + 1);
        String type11 = "FR" + String.valueOf(orden[10] + 1);
        String type12 = "FR" + String.valueOf(orden[11] + 1);
        String type13 = "FR" + String.valueOf(orden[12] + 1);
        String type14 = "FR" + String.valueOf(orden[13] + 1);
        String type15 = "FR" + String.valueOf(orden[14] + 1);
        String type16 = "FR" + String.valueOf(orden[15] + 1);
        String type17 = "FR" + String.valueOf(orden[16] + 1);
        String type18 = "FR" + String.valueOf(orden[17] + 1);
        String type19 = "FR" + String.valueOf(orden[18] + 1);
        String type20 = "FR" + String.valueOf(orden[19]
+ 1);
        String type21 = "FR" + String.valueOf(orden[20]
+ 1);
        String type22 = "FR" + String.valueOf(orden[21] + 1);
        String type23 = "FR" + String.valueOf(orden[22] + 1);
        String type24 = "FR" + String.valueOf(orden[23] + 1);
        String type25 = "FR" + String.valueOf(orden[24] + 1);
        String type26 = "FR" + String.valueOf(orden[25] + 1);
        String type27 = "FR" + String.valueOf(orden[26] + 1);
        String type28 = "FR" + String.valueOf(orden[27] + 1);
        String type29 = "FR" + String.valueOf(orden[28] + 1);
        String type30 = "FR" + String.valueOf(orden[29] + 1);
        String type31 = "FR" + String.valueOf(orden[30] + 1);
        String type32 = "FR" + String.valueOf(orden[31] + 1);
        String type33 = "FR" + String.valueOf(orden[32] + 1);
        String type34 = "FR" + String.valueOf(orden[33] + 1);
        String type35 = "FR" + String.valueOf(orden[34] + 1);
        String type36 = "FR" + String.valueOf(orden[35] + 1);
        String type37 = "FR" + String.valueOf(orden[36] + 1);
    }

```

ANEXO

```

String type38 = "FR" + String.valueOf(orden[37] + 1);
String type39 = "FR" + String.valueOf(orden[38] + 1);
    String type40 = "FR" + String.valueOf(orden[39]
+ 1);
        String type41 = "FR" + String.valueOf(orden[40]
+ 1);
            String type42 = "FR" + String.valueOf(orden[41] + 1);
            String type43 = "FR" + String.valueOf(orden[42] + 1);
            String type44 = "FR" + String.valueOf(orden[43] + 1);
            String type45 = "FR" + String.valueOf(orden[44] + 1);
            String type46 = "FR" + String.valueOf(orden[45] + 1);
            String type47 = "FR" + String.valueOf(orden[46] + 1);
            String type48 = "FR" + String.valueOf(orden[47] + 1);
            String type49 = "FR" + String.valueOf(orden[48] + 1);
            String type50 = "FR" + String.valueOf(orden[49] + 1);
            String type51 = "FR" + String.valueOf(orden[50] + 1);
            String type52 = "FR" + String.valueOf(orden[51] + 1);
            String type53 = "FR" + String.valueOf(orden[52] + 1);
            String type54 = "FR" + String.valueOf(orden[53] + 1);
            String type55 = "FR" + String.valueOf(orden[54] + 1);
            String type56 = "FR" + String.valueOf(orden[55] + 1);
            String type57 = "FR" + String.valueOf(orden[56] + 1);
            String type58 = "FR" + String.valueOf(orden[57] + 1);
            String type59 = "FR" + String.valueOf(orden[58] + 1);
                String type60 = "FR" + String.valueOf(orden[59]
+ 1);
                // create the dataset...
                DefaultCategoryDataset dataset = new
DefaultCategoryDataset();
                if (d[0] != 0) {
                dataset.addValue(p[0], series1, type1);
                if (d[1] != 0) {
                dataset.addValue(p[1], series1, type2);
                if (d[2] != 0) {
                dataset.addValue(p[2], series1, type3);
                if (d[3] != 0) {
                dataset.addValue(p[3], series1, type4);
                if (d[4] != 0) {
                dataset.addValue(p[4], series1, type5);
                if (d[5] != 0) {
                dataset.addValue(p[5], series1, type6);
                if (d[6] != 0) {
                dataset.addValue(p[6], series1, type7);
                if (d[7] != 0) {
                dataset.addValue(p[7], series1, type8);
                if (d[8] != 0) {
                dataset.addValue(p[8], series1, type9);
                if (d[9] != 0) {
                dataset.addValue(p[9], series1, type10);
                if (d[10] != 0) {
                dataset.addValue(p[10], series1, type11);
                if (d[11] != 0) {
                dataset.addValue(p[11], series1, type12);
                if (d[12] != 0) {
                dataset.addValue(p[12], series1, type13);
                if (d[13] != 0) {

```

```

dataset.addValue(p[13],series1,type14);
if (d[14] != 0) {
dataset.addValue(p[14],series1,type15);
if (d[15] != 0) {
dataset.addValue(p[15],series1,type16);
if (d[16] != 0) {
dataset.addValue(p[16],series1,type17);
if (d[17] != 0) {
dataset.addValue(p[17],series1,type18);
if (d[18] != 0) {
dataset.addValue(p[18],series1,type19);
if (d[19] != 0) {
dataset.addValue(p[19],series1,type20);
if (d[20] != 0) {
dataset.addValue(p[20], series1, type21);
if (d[21] != 0) {
dataset.addValue(p[21], series1, type22);
if (d[22] != 0) {
dataset.addValue(p[22], series1, type23);
if (d[23] != 0) {
dataset.addValue(p[23], series1, type24);
if (d[24] != 0) {
dataset.addValue(p[24], series1, type25);
if (d[25] != 0) {
dataset.addValue(p[25], series1, type26);
if (d[26] != 0) {
dataset.addValue(p[26], series1, type27);
if (d[27] != 0) {
dataset.addValue(p[27], series1, type28);
if (d[28] != 0) {
dataset.addValue(p[28], series1,type29);
if (d[29] != 0) {
dataset.addValue(p[29], series1,type30);
if (d[30] != 0) {
dataset.addValue(p[30],series1, type31);
if (d[31] != 0) {
dataset.addValue(p[31],series1,type32);
if (d[32] != 0) {
dataset.addValue(p[32],series1,type33);
if (d[33] != 0) {
dataset.addValue(p[33],series1,type34);
if (d[34] != 0) {
dataset.addValue(p[34],series1,type35);
if (d[35] != 0) {
dataset.addValue(p[35],series1,type36);
if (d[36] != 0) {
dataset.addValue(p[36],series1,type37);
if (d[37] != 0) {
dataset.addValue(p[37],series1,type38);
if (d[38] != 0) {
dataset.addValue(p[38],series1,type39);
if (d[39] != 0) {
dataset.addValue(p[39],series1,type40);
if (d[40] != 0) {
dataset.addValue(p[40], series1, type41);
if (d[41] != 0) {

```



```
dataset.addValue(d[7], series2, type8);
if (d[8] != 0) {
dataset.addValue(d[8], series2, type9);
if (d[9] != 0) {
dataset.addValue(d[9], series2, type10);
if (d[10] != 0) {
dataset.addValue(d[10], series2, type11);
if (d[11] != 0) {
dataset.addValue(d[11], series2, type12);
if (d[12] != 0) {
dataset.addValue(d[12], series2, type13);
if (d[13] != 0) {
dataset.addValue(d[13], series2, type14);
if (d[14] != 0) {
dataset.addValue(d[14], series2, type15);
if (d[15] != 0) {
dataset.addValue(d[15], series2, type16);
if (d[16] != 0) {
dataset.addValue(d[16], series2, type17);
if (d[17] != 0) {
dataset.addValue(d[17], series2, type18);
if (d[18] != 0) {
dataset.addValue(d[18], series2, type19);
if (d[19] != 0) {
dataset.addValue(d[19], series2, type20);
if (d[20] != 0) {
dataset.addValue(d[20], series2, type21);
if (d[21] != 0) {
dataset.addValue(d[21], series2, type22);
if (d[22] != 0) {
dataset.addValue(d[22], series2, type23);
if (d[23] != 0) {
dataset.addValue(d[23], series2, type24);
if (d[24] != 0) {
dataset.addValue(d[24], series2, type25);
if (d[25] != 0) {
dataset.addValue(d[25], series2, type26);
if (d[26] != 0) {
dataset.addValue(d[26], series2, type27);
if (d[27] != 0) {
dataset.addValue(d[27], series2, type28);
if (d[28] != 0) {
dataset.addValue(d[28], series2, type29);
if (d[29] != 0) {
dataset.addValue(d[29], series2, type30);
if (d[30] != 0) {
dataset.addValue(d[30], series2, type31);
if (d[31] != 0) {
dataset.addValue(d[31], series2, type32);
if (d[32] != 0) {
dataset.addValue(d[32], series2, type33);
if (d[33] != 0) {
dataset.addValue(d[33], series2, type34);
if (d[34] != 0) {
dataset.addValue(d[34], series2, type35);
if (d[35] != 0) {
```

ANEXO

```
dataset.addValue(d[35],series2,type36);
if (d[36] != 0) {
dataset.addValue(d[36],series2,type37);
if (d[37] != 0) {
dataset.addValue(d[37],series2,type38);
if (d[38] != 0) {
dataset.addValue(d[38],series2,type39);
if (d[39] != 0) {
dataset.addValue(d[39],series2,type40);
if (d[40] != 0) {
dataset.addValue(d[40], series2, type41);
if (d[41] != 0) {
dataset.addValue(d[41], series2, type42);
if (d[42] != 0) {
dataset.addValue(d[42], series2, type43);
if (d[43] != 0) {
dataset.addValue(d[43], series2, type44);
if (d[44] != 0) {
dataset.addValue(d[44], series2, type45);
if (d[45] != 0) {
dataset.addValue(d[45], series2, type46);
if (d[46] != 0) {
dataset.addValue(d[46], series2, type47);
if (d[47] != 0) {
dataset.addValue(d[47], series2, type48);
if (d[48] != 0) {
dataset.addValue(d[48], series2,type49);
if (d[49] != 0) {
dataset.addValue(d[49], series2,type50);
if (d[50] != 0) {
dataset.addValue(d[50],series2, type51);
if (d[51] != 0) {
dataset.addValue(d[51],series2,type52);
if (d[52] != 0) {
dataset.addValue(d[52],series2,type53);
if (d[53] != 0) {
dataset.addValue(d[53],series2,type54);
if (d[54] != 0) {
dataset.addValue(d[54],series2,type55);
if (d[55] != 0) {
dataset.addValue(d[55],series2,type56);
if (d[56] != 0) {
dataset.addValue(d[56],series2,type57);
if (d[57] != 0) {
dataset.addValue(d[57],series2,type58);
if (d[58] != 0) {
dataset.addValue(d[58],series2,type59);
if (d[59] != 0) {
dataset.addValue(d[59],series2,type60);
}
}
}
}
}
}
}
}
}
```



```

/**
 * @param dataset
 * @return Funcion JFreeChart: A partir de un objeto
CategoryDataset se
 *         encarga de crear el grafico JFreeChart.
 */
private JFreeChart createChart(CategoryDataset dataset) {
    // create the chart...
    JFreeChart chart = ChartFactory.createLineChart(
        "Risk factor polygonal graphics", // chart
        // title
        "Risk factor", // domain axis label
        "Relative frequency", // range axis label
        dataset, // data
        PlotOrientation.VERTICAL, // orientation
        true, // include legend
        true, // tooltips
        false // urls
    );
    chart.setBackgroundPaint(new Color(0xCC, 0xCC,
0xFF));
    CategoryPlot plot = (CategoryPlot) chart.getPlot();
    LineAndShapeRenderer renderer =
(LineAndShapeRenderer) plot
        .getRenderer();
    renderer.setShapesVisible(true);
    renderer.setDrawOutlines(true);
    renderer.setUseFillPaint(true);
    renderer.setFillPaint(Color.white);
    /*
     * CategoryPlot plot = chart.getCategoryPlot();
     *
     * NumberAxis rangeAxis = (NumberAxis)
plot.getRangeAxis();
     *
rangeAxis.setStandardTickUnits(NumberAxis.createIntegerTickUnits
());
     *
     * rangeAxis.setAutoRangeIncludesZero(true);
     * rangeAxis.setUpperMargin(0.20);
rangeAxis.setLabelAngle(Math.PI /
     * 2.0);
     */
    return chart;
}
}

```

VISTA**AcercaDe**

```

package org.anesthsom.anesthsom.vista;

//import java.applet.Applet;
import java.awt.Color;

import java.awt.Font;

import java.awt.Graphics;

import javax.swing.JFrame;

//import javax.swing.JFrame;

public class AcercaDe extends JFrame {

    /**
     *
     */
    private static final long serialVersionUID = -
7119831851606647362L;

    // establecer cadena de la barra de título y dimensiones
de la ventana
    public AcercaDe()
    {
        super( "About..." );

        setSize( 500, 500 );
        setVisible( true );
    }

    // mostrar métrica del tipo de letra
    public void paint( Graphics g )
    {
        super.paint( g ); // llamar al método paint de la
superclase

        g.setFont( new Font( "SansSerif", Font.BOLD, 26 ) );
        g.setColor(Color.blue);
        g.drawString( "Bioin Omega 3: " , 150, 50 );
        g.setColor(Color.blue);
        g.drawString( "Planning Reports" , 100,75 );

        Font font = new Font( "Serif", Font.ITALIC, 22 );
        g.setColor(Color.black);
        g.setFont( font );
    }
}

```

```

        g.drawString( "Produced by: ", 10, 130 );
        g.setFont(new Font( "Serif", Font.ITALIC, 20 ));
        g.drawString( "Empresa Miğtodos Omega Alfa", 50,
155 );
        g.setFont(new Font( "Serif", Font.ITALIC, 16 ));
        g.drawString("Developers and Inventors:",250 , 250);
        g.setFont(new Font( "Serif", Font.ITALIC, 14 ));
        g.drawString("Pablo Torner",350 , 265);
        g.drawString("Emilio Angel Rubio",350 , 280);
        g.drawString("Ana Isabel Garcia",350 , 295);
        g.drawString("Juan Carlos Conte",350 , 310);
        g.setFont( new Font( "SansSerif", Font.BOLD, 16 ) );
        g.drawString( "Contact e-mail: " , 10,400 );
        g.drawString( "metodos.omega.alfa@gmail.com " ,
50,415 );

    }

}

package org.anesthsom.anesthsom.vista;

//import java.applet.Applet;
import java.awt.Color;

import java.awt.Font;

import java.awt.Graphics;

import javax.swing.JFrame;

//import javax.swing.JFrame;

public class AcercaDe2 extends JFrame {

    /**
     *
     */
    private static final long serialVersionUID = -
7119831851606647362L;

    // establecer cadena de la barra de título y dimensiones
de la ventana
    public AcercaDe2()
    {
        //super( "Acerca de..." );

        setSize( 810, 500 );
        setVisible( true );
    }

    // mostrar mítica del tipo de letra

```

ANEXO

```
public void paint( Graphics g )
{
    super.paint( g ); // llamar al método paint de la
superclase

    g.setFont( new Font( "SansSerif", Font.BOLD, 26 ) );
    g.setColor(Color.blue);
    g.drawString( "Bioin Omega 2: " , 150, 50 );
    g.drawString( "Application of analysis and data
representation" , 100,75 );

    Font font = new Font( "Serif", Font.ITALIC, 22 );
    g.setColor(Color.black);
    g.setFont( font );
    g.drawString( "Produced by: " , 10, 130 );
    g.setFont(new Font( "Serif", Font.ITALIC, 20 ));
    g.drawString( "Empresa Métodos Omega Alfa", 50,
155 );

    g.setFont(new Font( "Serif", Font.ITALIC, 16 ));
    g.drawString("Developers and inventors:",250 , 250);
    g.setFont(new Font( "Serif", Font.ITALIC, 14 ));
    g.drawString("Pablo Torner",350 , 265);
    g.drawString("Juan Carlos Conte",350 , 280);
    g.drawString("Ana Isabel Garcia",350 , 295);
    g.drawString("Emilio Angel Rubio",350 , 310);
    g.setFont( new Font( "SansSerif", Font.BOLD, 16 ) );
    g.drawString( "Contact e-mail: " , 10,400 );
    g.drawString( "metodos.omega.alfa@gmail.com " ,
50,415 );

}

}
```

AcercaDe2

```

package org.anesthsom.anesthsom.vista;

//import java.applet.Applet;
import java.awt.Color;

import java.awt.Font;

import java.awt.Graphics;

import javax.swing.JFrame;

//import javax.swing.JFrame;

public class AcercaDe2 extends JFrame {

    /**
     *
     */
    private static final long serialVersionUID = -
7119831851606647362L;

    // establecer cadena de la barra de título y dimensiones
de la ventana
    public AcercaDe2()
    {
        //super( "Acerca de..." );

        setSize( 810, 500 );
        setVisible( true );
    }

    // mostrar métrica del tipo de letra
    public void paint( Graphics g )
    {
        super.paint( g ); // llamar al método paint de la
superclase

        g.setFont( new Font( "SansSerif", Font.BOLD, 26 ) );
        g.setColor(Color.blue);
        g.drawString( "Bioin Omega 2: " , 150, 50 );
        g.drawString( "Application of analysis and data
representation" , 100,75 );

        Font font = new Font( "Serif", Font.ITALIC, 22 );
        g.setColor(Color.black);
        g.setFont( font );
        g.drawString( "Produced by: ", 10, 130 );
        g.setFont(new Font( "Serif", Font.ITALIC, 20 ));

```


ANEXO

```
g.drawString( "Empresa MiMetodos Omega Alfa", 50,
155 );
g.setFont(new Font( "Serif", Font.ITALIC, 16 ));
g.drawString("Developers and inventors:",250 , 250);
g.setFont(new Font( "Serif", Font.ITALIC, 14 ));
g.drawString("Pablo Torner",350 , 265);
g.drawString("Juan Carlos Conte",350 , 280);
g.drawString("Ana Isabel Garcia",350 , 295);
g.drawString("Emilio Angel Rubio",350 , 310);
g.setFont( new Font( "SansSerif", Font.BOLD, 16 ) );
g.drawString( "Contact e-mail: " , 10,400 );
50,415 );
g.drawString( "metodos.omega.alfa@gmail.com " ,

}

}
```

Central

```

package org.anesthsom.anesthsom.vista;
import java.awt.BorderLayout;
import java.awt.event.ActionEvent;
import java.awt.event.KeyEvent;
import javax.swing.JApplet;
import javax.swing.JFrame;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.JPanel;
import javax.swing.JTabbedPane;
import javax.swing.KeyStroke;

/**
 * @author Pablo Torner Pelay
 *
 */
public class Central extends JApplet{

    private static final long serialVersionUID = 1L;

    private JPanel jContentPane = null;

    private JPanel resultados = null;

    private JTabbedPane jTabbedPane = null;

    private Operaciones anadir = null;

    private Guardar guardar = null;
    private JMenuBar barra = new JMenuBar();
    private JMenu archivo=new JMenu("File");

    private JMenu editar=new JMenu("Edit");
    private JMenu ayuda = new JMenu("Help");

    JMenuItem botonAyuda;
    JMenuItem botonAcerca;
    /**
     * This is the default constructor
     */
    public Central() {
        super();
        initialize();
    }

    /**
     * This method initializes this
     *
     * @return void
     */

```

```

private void initialize() {
    this.setSize(500, 200);
    setJMenuBar(barra);
    this.add(barra);
    //A LA BARRA LE AGREGAMOS LOS MENUS.
    barra.add(archivo);
    barra.add(editar);
    barra.add(ayuda);
    ayuda.add(getAyudaItem());
    ayuda.addSeparator();
    ayuda.add(getAcercaItem());
    this.add(getJContentPane());

    //this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    this.setContentPane(getJContentPane());

    //this.setTitle("Riesgos y lesiones version 1.0");
}
private JMenuItem getAcercaItem() {
    if (botonAcerca == null) {
        botonAcerca = new JMenuItem("About...");

        botonAcerca.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_2,
            ActionEvent.ALT_MASK));
        botonAcerca.addActionListener(new
java.awt.event.ActionListener() {
            public void
actionPerformed(java.awt.event.ActionEvent e) {
                AcercaDe2 aplicacion = new AcercaDe2();
                aplicacion.setDefaultCloseOperation(
JFrame.DISPOSE_ON_CLOSE );
            }
        });
    }
    return botonAcerca;
}
private JMenuItem getAyudaItem() {
    if (botonAyuda == null) {
        botonAyuda = new JMenuItem("Help");

        botonAyuda.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_1,
            ActionEvent.ALT_MASK));
        botonAyuda.addActionListener(new
java.awt.event.ActionListener() {
            public void
actionPerformed(java.awt.event.ActionEvent e) {
                System.out.println("Esto es la ayuda");
            }
        });
    }
    return botonAyuda;
}
/**
 * @return
 */

```

```

private JPanel getJContentPane() {
    if (jContentPane == null) {
        jContentPane = new JPanel();
        jContentPane.setLayout(new BorderLayout());
        //jContentPane.add(getResultados(),
BorderLayout.WEST);
        jContentPane.add(getJTabbedPane(),
BorderLayout.CENTER);
    }
    return jContentPane;
}

/**
 * @return
 */
private JTabbedPane getJTabbedPane() {
    if (jTabbedPane == null) {
        jTabbedPane = new JTabbedPane();
        jTabbedPane.addTab("Save", null, getGuardar(),
null);
        jTabbedPane.addTab("Operations", null,
getAnadir1(), null);
    }
    return jTabbedPane;
}

/**
 * @return
 */
private Operaciones getAnadir1() {
    if (anadir == null) {
        anadir = new Operaciones(resultados);
    }
    return anadir;
}

/**
 * @return
 */
private Guardar getGuardar() {
    if (guardar == null) {
        guardar = new Guardar();
    }
    return guardar;
}
}

```

Central I

```

package org.anesthsom.anesthsom.vista;

import java.awt.BorderLayout;
import java.awt.Dimension;

import java.awt.event.ActionEvent;
import java.awt.event.KeyEvent;
import javax.swing.JFrame;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTabbedPane;
import javax.swing.KeyStroke;

/**
 * @author Pablo Torner Pelay
 *
 *
 * Descripción: Es la parte del programa visor de la aplicación
 traductora.
 * Contiene a Guardar y las distintas Operaciones.
 *
 */
public class Central_1 extends JFrame {

    private static final long serialVersionUID = 1L;

    private JPanel jContentPane = null;

    private JPanel resultados = null;

    private JTabbedPane jTabbedPane = null;

    private OperacionesInformeCompleto anadir1 = null;

    private OperacionesDelta anadir2 = null;

    private OperacionesOIT anadir3 = null;

    private Guardar_1 guardar = null;
    private JMenuBar barra = new JMenuBar();
    private JMenu archivo=new JMenu("Archivo");

```

```

private JMenu editar=new JMenu("Editar");
private JMenu ayuda = new JMenu("?");

JMenuItem botonAyuda;
JMenuItem botonAcerca;

/**
 *
 */
public Central_1() {
    super();
    initialize();
}

/**
 *
 */
private void initialize() {
    this.setSize(600, 600);
    setJMenuBar(barra);

    //A LA BARRA LE AGREGAMOS LOS MENUS.
    barra.add(archivo);
    barra.add(editar);
    barra.add(ayuda);
    ayuda.add(getAyudaItem());
    ayuda.addSeparator();
    ayuda.add(getAcercaItem());

    this.setTitle("Riesgos y lesiones version beta 1.1");
    JScrollPane scrollPane = new
JScrollPane(getJContentPane());

scrollPane.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL_S
CROLLBAR_AS_NEEDED);

scrollPane.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROL
LBAR_AS_NEEDED);
    scrollPane.setBounds(0, 0, 600, 600);
    JPanel contentPane = new JPanel(null);
    contentPane.setPreferredSize(new Dimension(600, 600));
    contentPane.add(scrollPane);
    this.setContentPane(contentPane);
    this.pack();
    this.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
    this.setVisible(true);
}

private JMenuItem getAcercaItem() {
    if (botonAcerca == null) {
        botonAcerca = new JMenuItem("Acerca de...");
    }
}

```

ANEXO

```
        botonAcerca.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_2,
        ActionEvent.ALT_MASK));
        botonAcerca.addActionListener(new
java.awt.event.ActionListener() {
        public void
actionPerformed(java.awt.event.ActionEvent e) {
                AcercaDe aplicacion = new AcercaDe();
                aplicacion.setDefaultCloseOperation(
JFrame.DISPOSE_ON_CLOSE );
        }
        });
    }
    return botonAcerca;
}

private JMenuItem getAyudaItem() {
    if (botonAyuda == null) {
        botonAyuda = new JMenuItem("Ayuda genral");
    }
    botonAyuda.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_1,
        ActionEvent.ALT_MASK));
    botonAyuda.addActionListener(new
java.awt.event.ActionListener() {
        public void
actionPerformed(java.awt.event.ActionEvent e) {
                System.out.println("Esto es la ayuda");
        }
    });
}
return botonAyuda;
}

/**
 * @return
 */
private JPanel getJContentPane() {
    if (jContentPane == null) {
        jContentPane = new JPanel();
        jContentPane.setLayout(new BorderLayout());
        jContentPane.add(getJTabbedPane());
    }
    return jContentPane;
}

/**
 * @return
 */
```

```

private JTabbedPane getJTabbedPane() {
    if (jTabbedPane == null) {
        jTabbedPane = new JTabbedPane();
        //jTabbedPane.addTab("Añadir Informe Completo",
null, getanadir1(), null);
        //jTabbedPane.addTab("Añadir Delta", null,
getanadir2(), null);
        jTabbedPane.addTab("Add Anesth", null,
getAnadir3(), null);
        jTabbedPane.addTab("Save/Load", null,
getGuardar(), null);
    }
    return jTabbedPane;
}

/**
 * @return
 */
private OperacionesInformeCompleto getanadir1() {
    if (anadir1 == null) {
        anadir1 = new
OperacionesInformeCompleto(resultados);
    }
    return anadir1;
}

/**
 * @return
 */
private OperacionesOIT getAnadir3() {
    if (anadir3 == null) {
        anadir3 = new OperacionesOIT(resultados);
    }
    return anadir3;
}

/**
 * @return
 */
private OperacionesDelta getanadir2() {
    if (anadir2 == null) {
        anadir2 = new OperacionesDelta(resultados);
    }
    return anadir2;
}

/**
 * @return
 */
private Guardar_1 getGuardar() {
    if (guardar == null) {
        guardar = new Guardar_1();
    }
    return guardar;
}

```



```
}

```

Guardar

```
package org.anesthsom.anesthsom.vista;

import java.awt.GridBagLayout;
import java.io.FileInputStream; //import
import java.io.FileOutputStream;
import java.io.ObjectInputStream; //import
import java.io.ObjectOutputStream;
import java.util.Observable;
import java.util.Observer;

import javax.swing.JButton;
import javax.swing.JFileChooser;
import javax.swing.JPanel;
import javax.swing.JTextArea;

import org.anesthsom.anesthsom.controlador.Controlador;
import org.anesthsom.anesthsom.modelo.Contenedor;

/**
 * @author Pablo Torner Pelay
 *
 */
public class Guardar extends JPanel implements Observer {

    /**
     *
     */
    private static final long serialVersionUID = -
8316795940861636880L;
    private JButton CargarBoton = null;
    private JButton Cargar1Boton = null;
    private JTextArea textoArea;

    /**
     * This is the default constructor
     */
    public Guardar() {
        super();
        initialize();
    }

    /**
     * This method initializes this
     *
     * @return void
     */
    private void initialize() {
        this.setSize(50, 50);
        this.setLayout(new GridBagLayout());
    }
}

```

```

        this.add(getCargarBoton(), new MargenConstrains(2,
0));
        this.add(getCargar1Boton(), new MargenConstrains(2,
1));
    }

    /**
     * @return
     */
    private JButton getCargarBoton() {
        if (CargarBoton == null) {
            CargarBoton = new JButton();
            CargarBoton.setText("Upload Anesthsom");
            CargarBoton.addActionListener(new
java.awt.event.ActionListener() {

                public void
actionPerformed(java.awt.event.ActionEvent e) {
                    JFileChooser fileChooser = new
JFileChooser();

                    int seleccion =
fileChooser.showOpenDialog(textoArea);
                    if (seleccion ==
JFileChooser.APPROVE_OPTION) {
                        String direccion =
fileChooser.getSelectedFile()
                            .getAbsolutePath();
                        try {
                            FileInputStream i = new
FileInputStream(direccion);
                            ObjectInputStream si = new
ObjectInputStream(i);

                            Controlador.getInstance().setGeneral(
                                (Contenedor)
si.readObject());
                        } catch (Exception ex) {
                            ex.printStackTrace();
                        }
                    }
                });
            });
        }
        return CargarBoton;
    }

    /**
     * This method initializes bajaTrabajadorBoton
     *
     * @return javax.swing.JButton
     */

    private JButton getCargar1Boton() {
        if (Cargar1Boton == null) {
            Cargar1Boton = new JButton();
            Cargar1Boton.setText("Upload hospital");

```

```

        Cargar1Boton.addActionListener(new
java.awt.event.ActionListener() {
        /*
        * (non-Javadoc)
        *
        * @see
java.awt.event.ActionListener#actionPerformed(java.awt.event.Act
ionEvent)
        */
        public void
actionPerformed(java.awt.event.ActionEvent e) {
            JFileChooser fileChooser = new
JFileChooser();
            int seleccion =
fileChooser.showOpenDialog(textoArea);
            if (seleccion ==
JFileChooser.APPROVE_OPTION) {
                String direccion2 =
fileChooser.getSelectedFile()
                .getAbsolutePath();
                try {
                    FileInputStream in = new
FileInputStream(direccion2);
                    ObjectInputStream s = new
ObjectInputStream(in);

                    Controlador.getInstance().setEmpresa(
                        (Contenedor)
s.readObject());
                } catch (Exception ex) {
                    ex.printStackTrace();
                }
            }
        });
    }

    return Cargar1Boton;
}

/**
 * This method initializes jTextField
 *
 * @return javax.swing.JTextField
 */

/**
 * @param o
 * @param arg
 */
public void update(Observable o, Object arg) {
}
}

```


Guardar I

```

package org.anesthsom.anesthsom.vista;

import java.awt.GridBagLayout;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.util.Observable;
import java.util.Observer;

import javax.swing.JButton;

import javax.swing.JFileChooser;
import javax.swing.JPanel;
import javax.swing.JTextArea;
import org.anesthsom.anesthsom.controlador.Controlador;
import org.anesthsom.anesthsom.controlador.Controlador_1;
import org.anesthsom.anesthsom.modelo.Contenedor;

/**
 * @author Pablo Torner Pelay
 */
public class Guardar_1 extends JPanel implements Observer{

    private static final long serialVersionUID = 1L;
    private JButton addObraBoton = null;
    private JButton finObraBoton = null;
    private JTextArea textoArea;
    /**
     *
     */
    public Guardar_1() {
        super();
        initialize();
        Controlador.getInstance().addObserver(this);
    }

    /**
     *
     */
    private void initialize() {
        this.setSize(300, 270);
        this.setLayout(new GridBagLayout());
        this.add(getAddObraBoton(), new
MargenConstrains(0,1));
        this.add(getFinObraBoton(), new
MargenConstrains(0,2));

```

```

    }

    /**
     * @return
     */
    private JButton getAddObraBoton() {
        if (addObraBoton == null) {
            addObraBoton = new JButton();
            addObraBoton.setText("Upload");
            addObraBoton.addActionListener(new
java.awt.event.ActionListener() {
                public void
actionPerformed(java.awt.event.ActionEvent e) {
                    JFileChooser fileChooser = new
JFileChooser();

                    int seleccion =
fileChooser.showOpenDialog(textoArea);
                    if (seleccion ==
JFileChooser.APPROVE_OPTION) {
                        String direccion2 =
fileChooser.getSelectedFile().getAbsolutePath();
                        try {
                            FileInputStream in = new
FileInputStream(direccion2);
                            ObjectInputStream s = new
ObjectInputStream(in);

                            Controlador_1.getInstance().setContenedor(
                                                                    (Contenedor)
s.readObject());
                        } catch (Exception ex) {
                            ex.printStackTrace();
                        }
                    }
                }
            });
        }
        return addObraBoton;
    }

    /**
     * @return
     */
    private JButton getFinObraBoton() {
        if (finObraBoton == null) {
            finObraBoton = new JButton();
            finObraBoton.setText("Save");
            finObraBoton.addActionListener(new
java.awt.event.ActionListener() {
                public void
actionPerformed(java.awt.event.ActionEvent e) {
                    JFileChooser jF1= new JFileChooser();
                    String direccion = "";

                    if(jF1.showSaveDialog(null)==JFileChooser.APPROVE_OPTION) {

```

```

                                direccion =
jFl.getSelectedFile().getAbsolutePath();
                                try {
FileOutputStream(out) = new
                                ObjectOutputStream(s) = new
ObjectOutputStream(out);

                                s.writeObject(Controlador_1.getInstance().getContenedor());
                                } catch (Exception ex) {
                                    ex.printStackTrace();
                                }
                                }
                                });
                                }
                                return finObraBoton;
                                }

/* (non-Javadoc)
 * @see java.util.Observer#update(java.util.Observable,
java.lang.Object)
 */
public void update(Observable o, Object arg) {
}

}

```

Lienzo

```

package org.anesthsom.anesthsom.vista;

import java.awt.Color;
import java.awt.Font;
import java.awt.Graphics;

import javax.swing.JPanel;

public class Lienzo extends JPanel {

    /**
     *
     */
    private static final long serialVersionUID = 1L;
    private int[] fase1r;
    private int[] fase2r;
    private int[] fase3r;
    private int[] fase4r;
    private int[] fase5r;
    private int[] fase6r;
    private int[] fase7r;
    private int[] fase8r;
    private int[] fase9r;
    private int[] fase1l;
    private int[] fase2l;
    private int[] fase3l;
    private int[] fase4l;
    private int[] fase5l;
    private int[] fase6l;
    private int[] fase7l;
    private int[] fase8l;
    private int[] fase9l;

    public Lienzo() {
        super();

        setSize(1000,1000);
    }

    public void paint(Graphics g){
        int a=50;
        int d=0;
        //int b=0;
        g.setColor(Color.blue);
        g.drawRect(0, 0, 1000, 1000);
        g.fillRect(0, 0, 1000, 1000);
        g.setColor(Color.white);
        g.setFont(new Font("SansSerif", Font.BOLD, 26));
        g.drawString("RISK FACTOR",50,a);
        a=a+20;

        if (fase1r.length!=0){

```


ANEXO

```
g.setFont(new Font("SansSerif", Font.BOLD, 18));
d++;
g.drawString("PHASE "+d,50,a);
a=a+20;
g.setFont(new Font("SansSerif", Font.BOLD, 16));

for(int i=0;i<fase1r.length;i++){
    int b=i*40;
    color(fase1r[i],1,g);
    g.drawString("FR"+fase1r[i],50+b,a);
}
}
if (fase2r.length!=0){
g.setFont(new Font("SansSerif", Font.BOLD, 18));
g.setColor(Color.white);
d++;
a=a+20;
g.drawString("PHASE "+d,50,a);
a=a+20;
g.setFont(new Font("SansSerif", Font.BOLD, 16));

for(int i=0;i<fase2r.length;i++){
    int b=i*40;
    color(fase2r[i],1,g);
    g.drawString("FR"+fase2r[i],50+b,a);
}
}
if(fase3r.length!=0){
g.setFont(new Font("SansSerif", Font.BOLD, 18));
g.setColor(Color.white);
d++;
a=a+20;
g.drawString("PHASE "+d,50,a);
a=a+20;
g.setFont(new Font("SansSerif", Font.BOLD, 16));

for(int i=0;i<fase3r.length;i++){
    int b=i*40;
    color(fase3r[i],1,g);
    g.drawString("FR"+fase3r[i],50+b,a);
}
}
if(fase4r.length!=0){
g.setFont(new Font("SansSerif", Font.BOLD, 18));
g.setColor(Color.white);
d++;
a=a+20;
g.drawString("PHASE "+d,50,a);
a=a+20;
g.setFont(new Font("SansSerif", Font.BOLD, 16));

for(int i=0;i<fase4r.length;i++){
    int b=i*40;
    color(fase4r[i],1,g);
    g.drawString("FR"+fase4r[i],50+b,a);
}
}
```

ANEXO

```
}
if (fase5r.length!=0){
g.setFont(new Font("SansSerif", Font.BOLD, 18));
g.setColor(Color.white);
d++;
a=a+20;
g.drawString("PHASE "+d,50,a);
a=a+20;
g.setFont(new Font("SansSerif", Font.BOLD, 16));

for(int i=0;i<fase5r.length;i++){
    int b=i*40;
    color(fase5r[i],1,g);
    g.drawString("FR"+fase5r[i],50+b,a);
}
}
if(fase6r.length!=0){
g.setFont(new Font("SansSerif", Font.BOLD, 18));
g.setColor(Color.white);
d++;
a=a+20;
g.drawString("PHASE "+d,50,a);
a=a+20;
g.setFont(new Font("SansSerif", Font.BOLD, 16));

for(int i=0;i<fase6r.length;i++){
    int b=i*40;
    color(fase6r[i],1,g);
    g.drawString("FR"+fase6r[i],50+b,a);
}
}
if(fase7r.length!=0){
g.setFont(new Font("SansSerif", Font.BOLD, 18));
g.setColor(Color.white);
d++;
a=a+20;
g.drawString("PHASE "+d,50,a);
a=a+20;
g.setFont(new Font("SansSerif", Font.BOLD, 16));

for(int i=0;i<fase7r.length;i++){
    int b=i*40;
    color(fase7r[i],1,g);
    g.drawString("FR"+fase7r[i],50+b,a);
}
}
if (fase8r.length!=0){
g.setFont(new Font("SansSerif", Font.BOLD, 18));
g.setColor(Color.white);
d++;
a=a+20;
g.drawString("PHASE "+d,50,a);
g.setFont(new Font("SansSerif", Font.BOLD, 16));
a=a+20;

for(int i=0;i<fase8r.length;i++){
```

ANEXO

```
        int b=i*40;
        color(fase8r[i],1,g);
        g.drawString("FR"+fase8r[i],50+b,a);
    }
}
if(fase9r.length!=0){
g.setFont(new Font("SansSerif", Font.BOLD, 18));
g.setColor(Color.white);
d++;
a=a+20;
g.drawString("PHASE "+d, 50, a);
g.setFont(new Font("SansSerif", Font.BOLD, 16));
a=a+20;

for(int i=0;i<fase9r.length;i++){
    int b=i*40;
    color(fase9r[i],1,g);
    g.drawString("FR"+fase9r[i],50+b,a);
}
}

a=50;
d=0;

g.setColor(Color.white);
g.setFont(new Font("SansSerif", Font.BOLD, 26));
g.drawString("INJURY",550,a);
a=a+20;
if (fase11.length!=0){
g.setFont(new Font("SansSerif", Font.BOLD, 18));
d++;
g.drawString("PHASE "+d,550,a);
a=a+20;
g.setFont(new Font("SansSerif", Font.BOLD, 16));

for(int i=0;i<fase11.length;i++){
    int b=i*35;
    color(fase11[i],1,g);
    g.drawString("C"+fase11[i],550+b,a);
}
}
if (fase21.length!=0){
g.setFont(new Font("SansSerif", Font.BOLD, 18));
g.setColor(Color.white);
d++;
a=a+20;
g.drawString("PHASE "+d,550,a);
a=a+20;
g.setFont(new Font("SansSerif", Font.BOLD, 16));

for(int i=0;i<fase21.length;i++){
    int b=i*35;
    color(fase21[i],1,g);
    g.drawString("C"+fase21[i],550+b,a);
}
}
}
```

ANEXO

```
if(fase3l.length!=0){
g.setFont(new Font("SansSerif", Font.BOLD, 18));
g.setColor(Color.white);
d++;
a=a+20;
g.drawString("PHASE "+d,550,a);
a=a+20;
g.setFont(new Font("SansSerif", Font.BOLD, 16));

for(int i=0;i<fase3l.length;i++){
    int b=i*35;
    color(fase3l[i],1,g);
    g.drawString("C"+fase3l[i],550+b,a);
}
}
if(fase4l.length!=0){
g.setFont(new Font("SansSerif", Font.BOLD, 18));
g.setColor(Color.white);
d++;
a=a+20;
g.drawString("PHASE "+d,550,a);
a=a+20;
g.setFont(new Font("SansSerif", Font.BOLD, 16));

for(int i=0;i<fase4l.length;i++){
    int b=i*35;
    color(fase4l[i],1,g);
    g.drawString("C"+fase4l[i],550+b,a);
}
}
if (fase5l.length!=0){
g.setFont(new Font("SansSerif", Font.BOLD, 18));
g.setColor(Color.white);
d++;
a=a+20;
g.drawString("PHASE "+d,550,a);
a=a+20;
g.setFont(new Font("SansSerif", Font.BOLD, 16));

for(int i=0;i<fase5l.length;i++){
    int b=i*35;
    color(fase5l[i],1,g);
    g.drawString("C"+fase5l[i],550+b,a);
}
}
if(fase6l.length!=0){
g.setFont(new Font("SansSerif", Font.BOLD, 18));
g.setColor(Color.white);
d++;
a=a+20;
g.drawString("PHASE "+d,550,a);
a=a+20;
g.setFont(new Font("SansSerif", Font.BOLD, 16));

for(int i=0;i<fase6l.length;i++){
    int b=i*35;
```

ANEXO

```

        color(fase6l[i],1,g);
        g.drawString("C"+fase6l[i],550+b,a);
    }
}
if(fase7l.length!=0){
g.setFont(new Font("SansSerif", Font.BOLD, 18));
g.setColor(Color.white);
d++;
a=a+20;
g.drawString("PHASE "+d,550,a);
a=a+20;
g.setFont(new Font("SansSerif", Font.BOLD, 16));

for(int i=0;i<fase7l.length;i++){
    int b=i*35;
    color(fase7l[i],1,g);
    g.drawString("C"+fase7l[i],550+b,a);
}
}
if (fase8l.length!=0){
g.setFont(new Font("SansSerif", Font.BOLD, 18));
g.setColor(Color.white);
d++;
a=a+20;
g.drawString("PHASE "+d,550,a);
g.setFont(new Font("SansSerif", Font.BOLD, 16));
a=a+20;

for(int i=0;i<fase8l.length;i++){
    int b=i*35;
    color(fase8l[i],1,g);
    g.drawString("C"+fase8l[i],550+b,a);
}
}
if(fase9l.length!=0){
g.setFont(new Font("SansSerif", Font.BOLD, 18));
g.setColor(Color.white);
d++;
a=a+20;
g.drawString("PHASE "+d, 550, a);
g.setFont(new Font("SansSerif", Font.BOLD, 16));
a=a+20;

for(int i=0;i<fase9l.length;i++){
    int b=i*35;
    color(fase9l[i],1,g);
    g.drawString("C"+fase9l[i],550+b,a);
}
}
}

private void color(int i, int j,Graphics g) {
    switch(j){
    case 1:
        if ((i==1)|| (i==2)|| (i==6)|| (i==13)){
            g.setColor(Color.yellow);

```

```

        }else
if((i==3)|| (i==4)|| (i==5)|| (i==7)|| (i==8)|| (i==9)|| (i==11)|| (i==
12)|| (i==19)) {
            g.setColor(Color.red);
        }else
if((i==10)|| (i==14)|| (i==15)|| (i==16)|| (i==17)|| (i==18)) {
            g.setColor(Color.green);
        }

        break;
    case 2:
        if((i==2)|| (i==3)|| (i==4)|| (i==5)|| (i==6)) {
            g.setColor(Color.yellow);
        }else
if((i==1)|| (i==7)|| (i==8)|| (i==9)|| (i==10)|| (i==19)) {
            g.setColor(Color.red);
        }else
if((i==11)|| (i==12)|| (i==13)|| (i==14)|| (i==15)|| (i==16)|| (i==17)
|| (i==18)) {
            g.setColor(Color.green);
        }

        break;
    }

}

public int[] getFase1r() {
    return fase1r;
}
public void setFase1r(int[] fase1r) {
    this.fase1r = fase1r;
}

public int[] getFase2r() {
    return fase2r;
}
public void setFase2r(int[] fase2r) {
    this.fase2r = fase2r;
}
public int[] getFase3r() {
    return fase3r;
}
public void setFase3r(int[] fase3r) {
    this.fase3r = fase3r;
}

public int[] getFase4r() {
    return fase4r;
}
public void setFase4r(int[] fase4r) {
    this.fase4r = fase4r;
}

public int[] getFase5r() {
    return fase5r;
}

```

```
}  
public void setFase5r(int[] fase5r) {  
    this.fase5r = fase5r;  
}  
  
public int[] getFase6r() {  
    return fase6r;  
}  
public void setFase6r(int[] fase6r) {  
    this.fase6r = fase6r;  
}  
  
public int[] getFase7r() {  
    return fase7r;  
}  
public void setFase7r(int[] fase7r) {  
    this.fase7r = fase7r;  
}  
  
public int[] getFase8r() {  
    return fase8r;  
}  
public void setFase8r(int[] fase8r) {  
    this.fase8r = fase8r;  
}  
  
public int[] getFase9r() {  
    return fase9r;  
}  
public void setFase9r(int[] fase9r) {  
    this.fase9r = fase9r;  
}  
  
public int[] getFase11() {  
    return fase11;  
}  
public void setFase11(int[] fase11) {  
    this.fase11 = fase11;  
}  
  
public int[] getFase21() {  
    return fase21;  
}  
public void setFase21(int[] fase21) {  
    this.fase21 = fase21;  
}  
  
public int[] getFase31() {  
    return fase31;  
}  
public void setFase31(int[] fase31) {  
    this.fase31 = fase31;  
}  
  
public int[] getFase41() {  
    return fase41;  
}
```

```
    }  
    public void setFase4l(int[] fase4l) {  
        this.fase4l = fase4l;  
    }  
  
    public int[] getFase5l() {  
        return fase5l;  
    }  
    public void setFase5l(int[] fase5l) {  
        this.fase5l = fase5l;  
    }  
  
    public int[] getFase6l() {  
        return fase6l;  
    }  
    public void setFase6l(int[] fase6l) {  
        this.fase6l = fase6l;  
    }  
  
    public int[] getFase7l() {  
        return fase7l;  
    }  
    public void setFase7l(int[] fase7l) {  
        this.fase7l = fase7l;  
    }  
  
    public int[] getFase8l() {  
        return fase8l;  
    }  
    public void setFase8l(int[] fase8l) {  
        this.fase8l = fase8l;  
    }  
  
    public int[] getFase9l() {  
        return fase9l;  
    }  
    public void setFase9l(int[] fase9l) {  
        this.fase9l = fase9l;  
    }  
}
```


MargenConstrains

```
package org.anesthsom.anesthsom.vista;

import java.awt.GridBagConstraints;
import java.awt.Insets;

/**
 * @author Pablo Torner Pelay
 * @author Nacho Piñeres Helguera
 */
public class MargenConstrains extends GridBagConstraints {

    private static final long serialVersionUID =
1959374415699848126L;

    public MargenConstrains(int x, int y) {
        gridx = x;
        gridy = y;
        fill = GridBagConstraints.BOTH;
        insets = new Insets(10, 10, 10, 10);
    }
}
```

Operaciones

```

package org.anesthsom.anesthsom.vista;

import java.awt.GridBagLayout;

import javax.swing.JButton;
import javax.swing.JPanel;
import org.anesthsom.anesthsom.controlador.Controlador;

/**
 * @author Pablo Torner Pelay
 *
 */
public class Operaciones extends JPanel {
    /**
     *
     */
    private static final long serialVersionUID = -
6804658819437742936L;
    private JButton altaMaquinaBoton = null;
    private JButton planificacionBoton = null;
    private JButton barrasBoton = null;
    private JPanel resultados = null;
    public Operaciones(JPanel como) {
        super();
        initialize();
        como= resultados;
    }
    /**
     *
     */
    private void initialize() {

        this.setSize(50, 50);
        this.setLayout(new GridBagLayout());

        this.add(getAltaMaquinaBoton(), new
MargenConstrains(0,0));
        this.add(getBarrasBoton(), new MargenConstrains(0,1));
        this.add(getPlanificacionBoton(), new
MargenConstrains(0,2));
    }

    /**
     * @return

```

```

*/
private JButton getAltaMaquinaBoton() {
    if (altaMaquinaBoton == null) {
        altaMaquinaBoton = new JButton();
        altaMaquinaBoton.setText("Polygonal graphics");
        altaMaquinaBoton.addActionListener(new
java.awt.event.ActionListener() {
            public void
actionPerformed(java.awt.event.ActionEvent e) {

                Controlador.getInstance().DPoligonal();
            }
        });
    }
    return altaMaquinaBoton;
}
/**
 * @return
 */
private JButton getBarrasBoton() {
    if (barrasBoton == null) {
        barrasBoton = new JButton();
        barrasBoton.setText("Bar chart");
        barrasBoton.addActionListener(new
java.awt.event.ActionListener() {
            public void
actionPerformed(java.awt.event.ActionEvent e) {

                Controlador.getInstance().DBarras();
            }
        });
    }
    return barrasBoton;
}
private JButton getPlanificacionBoton() {
    if (planificacionBoton == null) {
        planificacionBoton = new JButton();
        planificacionBoton.setText("Planning Reports");
        planificacionBoton.addActionListener(new
java.awt.event.ActionListener() {
            public void
actionPerformed(java.awt.event.ActionEvent e) {

                Controlador.getInstance().Planificacion();
            }
        });
    }
    return planificacionBoton;
}
}
}

```

OperacionesDelta

```

package org.anesthsom.anesthsom.vista;

import java.awt.GridBagLayout;

import javax.swing.JButton;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;
import org.anesthsom.anesthsom.controlador.Controlador_1;

/**
 * @author Pablo Torner Pelay
 * @author Nacho Piñeres Helguera
 */
public class OperacionesDelta extends JPanel {

    private static final long serialVersionUID = 1L;
    private JButton altaMaquinaBoton = null;
    private JPanel resultados = null;
    private JTextField desviacTextField = null;
    private JLabel desviacLabel = null;

    private JTextField formaTextField = null;
    private JLabel formaLabel = null;
    private JTextField dlesionTextField = null;
    private JLabel dlesionLabel = null;
    private JTextField partelesionTextField = null;
    private JLabel partelesionLabel = null;
    public OperacionesDelta(JPanel como) {
        super();
        initialize();
        como= resultados;
    }

    /**
     *
     */
    private void initialize() {

        desviacLabel = new JLabel();
        desviacLabel.setText("Situación");
        formaLabel = new JLabel();
        formaLabel.setText("Forma de contacto");
        dlesionLabel = new JLabel();
        dlesionLabel.setText("Complicación");
        partelesionLabel = new JLabel();

```

```

    partelesionLabel.setText("Parte del cuerpo lesionada");
    this.setSize(500, 500);
    this.setLayout(new GridBagLayout());

    this.add(desviacLabel, new MargenConstrains(0,0));
    this.add(getdesviacTextField(), new
MargenConstrains(1,0));

    this.add(formaLabel, new MargenConstrains(0,1));
    this.add(getformaTextField(), new
MargenConstrains(1,1));
    this.add(dlesionLabel, new MargenConstrains(0,2));
    this.add(getdlesionTextField(), new
MargenConstrains(1,2));
    this.add(partelesionLabel, new MargenConstrains(0,3));
    this.add(getpartelesionTextField(), new
MargenConstrains(1,3));
    this.add(getAltaMaquinaBoton(), new
MargenConstrains(1,4));
    }

    /**
     * @return
     */
    private JButton getAltaMaquinaBoton() {
        if (altaMaquinaBoton == null) {
            altaMaquinaBoton = new JButton();
            altaMaquinaBoton.setText("AltaMaquinaBoton");
            altaMaquinaBoton.addActionListener(new
java.awt.event.ActionListener() {
                public void
actionPerformed(java.awt.event.ActionEvent e) {
                    int
desviac=Integer.parseInt(desviacTextField.getText());
                    int
forma=Integer.parseInt(formaTextField.getText());
                    int
dlesion=Integer.parseInt(dlesionTextField.getText());

Controlador_1.getInstance().addAccidentadoAlm3(desviac, forma,
dlesion);

                }
            });
        }
        return altaMaquinaBoton;
    }

    /**
     * @return
     */
    private JTextField getdesviacTextField() {
        if (desviacTextField == null) {
            desviacTextField = new JTextField();
        }
    }

```

```
        return desviacTextField;
    }
    /**
     * @return
     */
    private JTextField getformaTextField() {
        if (formaTextField == null) {
            formaTextField = new JTextField();
        }
        return formaTextField;
    }

    /**
     * @return
     */
    private JTextField getdlesionTextField() {
        if (dlesionTextField == null) {
            dlesionTextField = new JTextField();
        }
        return dlesionTextField;
    }

    private JTextField getpartelesionTextField() {
        if (partelesionTextField == null) {
            partelesionTextField = new JTextField();
        }
        return partelesionTextField;
    }
}
```

OperacionesInformaCompleto

```

package org.anesthsom.anesthsom.vista;

import java.awt.GridBagLayout;

import javax.swing.JButton;
import javax.swing.JCheckBox;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;
import org.anesthsom.anesthsom.controlador.Controlador_1;

/**
 * @author Pablo Torner Pelay
 * @author Nacho Piñeres Helguera
 */
public class OperacionesInformeCompleto extends JPanel {
    private static final long serialVersionUID = 1L;
    private JButton altaMaquinaBoton = null;
    private JPanel resultados = null;
    private JTextField plantillaEmpresaTextField = null;
    private JLabel plantillaEmpresaLabel = null;
    private JLabel CodigoCnaeLabel = null;
    private JLabel codigoEmpresarialLabel = null;
    private JTextField CodigoCnaeTextField = null;
    private JTextField codigoEmpresarialTextField = null;
    private JTextField accLugTextField = null;
    private JLabel accLugLabel = null;
    private JTextField actiEspTextField = null;
    private JLabel actiEspLabel = null;
    private JTextField agente1TextField = null;
    private JLabel agente1Label = null;
    private JTextField desviacTextField = null;
    private JLabel desviacLabel = null;
    private JTextField agente2TextField = null;
    private JLabel agente2Label = null;
    private JTextField formaTextField = null;
    private JLabel formaLabel = null;
    private JTextField aparatoTextField = null;
    private JLabel aparatoLabel = null;
    private JTextField dlesionTextField = null;
    private JLabel dlesionLabel = null;
    private JTextField glesionTextField = null;
    private JLabel glesionLabel = null;
    private JTextField pclesionadaTextField = null;
    private JLabel pclesionadaLabel = null;
    private JTextField glesionRTextField = null;

```

ANEXO

```
private JLabel glesionRLabel = null;
private JTextField cccotTextField = null;
private JLabel cccotLabel = null;
    private JCheckBox recaidaBox = null;
    private JLabel recaidaLabel=null;
    private JCheckBox contrataBox = null;
    private JLabel contrataLabel=null;
    private JCheckBox traficoBox = null;
    private JLabel traficoLabel=null;
    private JCheckBox perteneceEmpresaBox = null;
    private JLabel perteneceEmpresaLabel=null;
    private JCheckBox perteneceOtEmpresaBox = null;
    private JLabel perteneceOtEmpresaLabel=null;
    private JCheckBox mastrabBox = null;
    private JLabel mastrabLabel=null;
public OperacionesInformeCompleto(JPanel como) {
    super();
    initialize();
    como= resultados;
}

/**
 *
 */
private void initialize() {
    codigoEmpresarialLabel = new JLabel();
    codigoEmpresarialLabel.setText("Codigo Empresarial");
    codigoCnaeLabel = new JLabel();
    codigoCnaeLabel.setText("Codigo CNAE");
    plantillaEmpresaLabel = new JLabel();
    plantillaEmpresaLabel.setText("Plantilla empresa");
    accLugLabel = new JLabel();
    accLugLabel.setText("Lugar del accidente");
    actiEspLabel = new JLabel();
    actiEspLabel.setText("cod. Actividad Especifica");
    agente1Label = new JLabel();
    agente1Label.setText("agente Actividad Especifica");
    desviacLabel = new JLabel();
    desviacLabel.setText("Codigo Desviacion");
    agente2Label = new JLabel();
    agente2Label.setText("agente Desviacion");
    formaLabel = new JLabel();
    formaLabel.setText("Codigo forma");
    aparatoLabel = new JLabel();
    aparatoLabel.setText("Agente forma de contacto");
    dlesionLabel = new JLabel();
    dlesionLabel.setText("Codigo lesion");
    glesionLabel = new JLabel();
    glesionLabel.setText("Grado de lesion");
    pclesionadaLabel = new JLabel();
    pclesionadaLabel.setText("parte cuerpo lesion");
    glesionRLabel = new JLabel();
    glesionRLabel.setText("grado lesion real");
    cccotLabel = new JLabel();
    cccotLabel.setText("codigo cuenta cotizacion");
    recaidaLabel = new JLabel();
```


ANEXO

```
recaidaLabel.setText("Tipo de accidente");
contrataLabel = new JLabel();
contrataLabel.setText("Contrata");
traficoLabel = new JLabel();
traficoLabel.setText("Trafico");
perteneceEmpresaLabel = new JLabel();
perteneceEmpresaLabel.setText("Pertenece a empresa");
perteneceOtEmpresaLabel = new JLabel();
perteneceOtEmpresaLabel.setText("Pertenece otra
empresa");
mastrabLabel = new JLabel();
mastrabLabel.setText("Mas trabajadores afec");
this.setSize(500, 500);
this.setLayout(new GridBagLayout());
this.add(plantillaEmpresaLabel, new
MargenConstrains(0,0));
this.add(CodigoCnaeLabel, new MargenConstrains(0,1));
this.add(getplantillaEmpresaTextField(), new
MargenConstrains(1,0));
this.add(getCodigoCnaeTextField(), new
MargenConstrains(1,1));
this.add(codigoEmpresarialLabel, new
MargenConstrains(0,2));
this.add(getcodigoEmpresarialTextField(), new
MargenConstrains(1,2));
this.add(accLugLabel, new MargenConstrains(0,3));
this.add(getaccLugTextField(), new
MargenConstrains(1,3));
this.add(actiEspLabel, new MargenConstrains(0,4));
this.add(getactiEspTextField(), new
MargenConstrains(1,4));
this.add(agentelLabel, new MargenConstrains(0,5));
this.add(getagentelTextField(), new
MargenConstrains(1,5));
this.add(desviacLabel, new MargenConstrains(0,6));
this.add(getdesviacTextField(), new
MargenConstrains(1,6));
this.add(agente2Label, new MargenConstrains(0,7));
this.add(getagente2TextField(), new
MargenConstrains(1,7));
this.add(formaLabel, new MargenConstrains(0,8));
this.add(getformaTextField(), new
MargenConstrains(1,8));
this.add(aparatoLabel, new MargenConstrains(0,9));
this.add(getaparatoTextField(), new
MargenConstrains(1,9));
this.add(dlesionLabel, new MargenConstrains(0,10));
this.add(getdlesionTextField(), new
MargenConstrains(1,10));
this.add(glesionLabel, new MargenConstrains(0,11));
this.add(getglesionTextField(), new
MargenConstrains(1,11));
this.add(pclesionadaLabel, new MargenConstrains(2,4));
this.add(getpclesionadaTextField(), new
MargenConstrains(3,4));
this.add(glesionRLabel, new MargenConstrains(0,13));
```

```

        this.add(getglesionRTextField(), new
MargenConstrains(1,13));
        this.add(cccotLabel, new MargenConstrains(0,14));
        this.add(getcccotTextField(), new
MargenConstrains(1,14));
        this.add(recaidaLabel, new MargenConstrains(0,15));
        this.add(getrecaidaTextField(), new
MargenConstrains(1,15));
        this.add(contrataLabel, new MargenConstrains(0,16));
        this.add(getcontrataTextField(), new
MargenConstrains(1,16));
        this.add(traficoLabel, new MargenConstrains(2,3));
        this.add(gettraficoTextField(), new
MargenConstrains(3,3));
        this.add(perteneceEmpresaLabel, new
MargenConstrains(2,2));
        this.add(getperteneceEmpresaTextField(), new
MargenConstrains(3,2));
        this.add(perteneceOtEmpresaLabel, new
MargenConstrains(2,1));
        this.add(getperteneceOtEmpresaTextField(), new
MargenConstrains(3,1));
        this.add(mastrabLabel, new MargenConstrains(2,0));
        this.add(mastrabTextField(), new MargenConstrains(3,0));
        this.add(getAltaMaquinaBoton(), new
MargenConstrains(3,10));
    }

    /**
     * @return
     */
    private JButton getAltaMaquinaBoton() {
        if (altaMaquinaBoton == null) {
            altaMaquinaBoton = new JButton();
            altaMaquinaBoton.setText("Alta Maquina");
            altaMaquinaBoton.addActionListener(new
java.awt.event.ActionListener() {
                public void
actionPerformed(java.awt.event.ActionEvent e) {
                    int plantillaEmpresa
=Integer.parseInt(plantillaEmpresaTextField.getText());
                    int codigoCnae =
Integer.parseInt(CodigoCnaeTextField.getText());
                    int codigoEmpresarial =
Integer.parseInt(codigoEmpresarialTextField.getText());
                    int
accLug=Integer.parseInt(accLugTextField.getText());
                    int
actiEsp=Integer.parseInt(actiEspTextField.getText());
                    int
agentel=Integer.parseInt(agentelTextField.getText());
                    int
desviac=Integer.parseInt(desviacTextField.getText());
                    int
agente2=Integer.parseInt(agente2TextField.getText());

```

```

        int
forma=Integer.parseInt(formaTextField.getText());
        int
aparato=Integer.parseInt(aparatoTextField.getText());
        int
dlesion=Integer.parseInt(dlesionTextField.getText());
        int
glesion=Integer.parseInt(glesionTextField.getText());
        String
pclesionada=pclesionadaTextField.getText();
        int
glesionR=Integer.parseInt(glesionRTextField.getText());
        int
cccot=Integer.parseInt(cccotTextField.getText());
        boolean recaida=recaidaBox.isEnabled();
        boolean contrata=contrataBox.isEnabled();
        boolean trafico=traficoBox.isEnabled();
        boolean
perteneceEmpresa=traficoBox.isEnabled();
        boolean
pertenceOtEmpresa=traficoBox.isEnabled();
        boolean mastrab=mastrabBox.isEnabled();

Controlador_1.getInstance().addAccidentadoAlm2(recaida,codigoCna
e, plantillaEmpresa,codigoEmpresarial, contrata,
accLug,trafico, perteneceEmpresa, pertenceOtEmpresa, actiEsp,
agentel, desviac,agente2,forma,aparato,mastrab, dlesion,
glesion, pclesionada, glesionR, cccot);
    }
    });
}
return altaMaquinaBoton;
}

/**
 * @return
 */
private JTextField getplantillaEmpresaTextField() {
    if (plantillaEmpresaTextField == null) {
        plantillaEmpresaTextField = new JTextField();
    }
    return plantillaEmpresaTextField;
}

/**
 * @return
 */
private JTextField getCodigoCnaeTextField() {
    if (CodigoCnaeTextField == null) {
        CodigoCnaeTextField = new JTextField();
    }
    return CodigoCnaeTextField;
}

/**
 * @return

```

```

    */
private JTextField getcodigoEmpresarialTextField() {
    if (codigoEmpresarialTextField == null) {
        codigoEmpresarialTextField = new JTextField();
    }
    return codigoEmpresarialTextField;
}
/**
 * @return
 */
private JTextField getaccLugTextField() {
    if (accLugTextField == null) {
        accLugTextField = new JTextField();
    }
    return accLugTextField;
}
/**
 * @return
 */
private JTextField getactiEspTextField() {
    if (actiEspTextField == null) {
        actiEspTextField = new JTextField();
    }
    return actiEspTextField;
}
/**
 * @return
 */
private JTextField getagente1TextField() {
    if (agente1TextField == null) {
        agente1TextField = new JTextField();
    }
    return agente1TextField;
}
/**
 * @return
 */
private JTextField getdesviacTextField() {
    if (desviacTextField == null) {
        desviacTextField = new JTextField();
    }
    return desviacTextField;
}
/**
 * @return
 */
private JTextField getagente2TextField() {
    if (agente2TextField == null) {
        agente2TextField = new JTextField();
    }
    return agente2TextField;
}
/**
 * @return
 */
private JTextField getformaTextField() {

```

```

    if (formaTextField == null) {
        formaTextField = new JTextField();
    }
    return formaTextField;
}
/**
 * @return
 */
private JTextField getaparatoTextField() {
    if (aparatoTextField == null) {
        aparatoTextField = new JTextField();
    }
    return aparatoTextField;
}
/**
 * @return
 */
private JTextField getdlesionTextField() {
    if (dlesionTextField == null) {
        dlesionTextField = new JTextField();
    }
    return dlesionTextField;
}
/**
 * @return
 */
private JTextField getglesionTextField() {
    if (glesionTextField == null) {
        glesionTextField = new JTextField();
    }
    return glesionTextField;
}
/**
 * @return
 */
private JTextField getpclesionadaTextField() {
    if (pclesionadaTextField == null) {
        pclesionadaTextField = new JTextField();
    }
    return pclesionadaTextField;
}
/**
 * @return
 */
private JTextField getcccotTextField() {
    if (cccotTextField == null) {
        cccotTextField = new JTextField();
    }
    return cccotTextField;
}
/**
 * @return
 */
private JTextField getglesionRTextField() {
    if (glesionRTextField == null) {
        glesionRTextField = new JTextField();
    }
}

```

```

    }
    return glesionRTextField;
}
/**
 * @return
 */
private JCheckBox getrecaidaTextField() {
    if (recaidaBox == null) {
        recaidaBox = new JCheckBox();
    }
    return recaidaBox;
}
/**
 * @return
 */
private JCheckBox getcontrataTextField() {
    if (contrataBox == null) {
        contrataBox = new JCheckBox();
    }
    return contrataBox;
}
/**
 * @return
 */
private JCheckBox gettraficoTextField() {
    if (traficoBox == null) {
        traficoBox = new JCheckBox();
    }
    return traficoBox;
}
/**
 * @return
 */
private JCheckBox getperteneciaEmpresaTextField() {
    if (perteneciaEmpresaBox == null) {
        perteneciaEmpresaBox = new JCheckBox();
    }
    return perteneciaEmpresaBox;
}
/**
 * @return
 */
private JCheckBox getperteneciaOtraEmpresaTextField() {
    if (perteneciaOtraEmpresaBox == null) {
        perteneciaOtraEmpresaBox = new JCheckBox();
    }
    return perteneciaOtraEmpresaBox;
}
/**
 * @return
 */
private JCheckBox mastrabTextField() {
    if (mastrabBox == null) {
        mastrabBox = new JCheckBox();
    }
    return mastrabBox;
}

```

```

    }
}

```

OPERACIONES OIT

```

package org.anesthsom.anesthsom.vista;

import java.awt.GridBagLayout;

import javax.swing.JButton;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTextField;
import org.anesthsom.anesthsom.controlador.Controlador_1;

/**
 * @author Pablo Torner Pelay
 * @author Nacho Pérez Helguera
 */
public class OperacionesOIT extends JPanel {
    private static final long serialVersionUID = 1L;
    private JButton altaMaquinaBoton = null;
    private JPanel resultados = null;
    private JTextField r1 = null;
    private JLabel r1Label = null;
    private JLabel r3Label = null;
    private JLabel r2Label = null;
    private JTextField r2 = null;
    private JTextField r3 = null;
    private JTextField r4 = null;
    private JLabel r4Label = null;
    private JTextField r5 = null;
    private JLabel r5Label = null;
    private JTextField r6 = null;
    private JLabel r6Label = null;
    private JTextField r7 = null;
    private JLabel r7Label = null;
    private JTextField r8 = null;
    private JLabel r8Label = null;
    private JTextField r9 = null;
    private JLabel r9Label = null;
    private JTextField r10 = null;
    private JLabel r10Label = null;
    private JTextField r11 = null;
    private JLabel r11Label = null;
    private JTextField r12 = null;
    private JLabel r12Label = null;
    private JTextField r13 = null;
    private JLabel r13Label = null;
    private JTextField r14 = null;
    private JLabel r14Label = null;
    private JTextField r15 = null;
    private JLabel r15Label = null;
}

```

```
private JTextField r16 = null;
private JLabel r16Label = null;
private JTextField r17 = null;
private JLabel r17Label = null;
private JTextField r18 = null;
private JLabel r18Label = null;
private JTextField r19 = null;
private JLabel r19Label = null;
private JTextField r20 = null;
private JLabel r20Label = null;
private JTextField r21 = null;
private JLabel r21Label = null;
private JLabel r23Label = null;
private JLabel r22Label = null;
private JTextField r22 = null;
private JTextField r23 = null;
private JTextField r24 = null;
private JLabel r24Label = null;
private JTextField r25 = null;
private JLabel r25Label = null;
private JTextField r26 = null;
private JLabel r26Label = null;
private JTextField r27 = null;
private JLabel r27Label = null;
private JTextField r28 = null;
private JLabel r28Label = null;
private JTextField r29 = null;
private JLabel r29Label = null;
private JTextField r30 = null;
private JLabel r30Label = null;
private JTextField r31 = null;
private JLabel r31Label = null;
private JTextField r32 = null;
private JLabel r32Label = null;
private JTextField r33 = null;
private JLabel r33Label = null;
private JTextField r34 = null;
private JLabel r34Label = null;
private JTextField r35 = null;
private JLabel r35Label = null;
private JTextField r36 = null;
private JLabel r36Label = null;
private JTextField r37 = null;
private JLabel r37Label = null;
private JTextField r38 = null;
private JLabel r38Label = null;
private JTextField r39 = null;
private JLabel r39Label = null;
private JTextField r40 = null;
private JLabel r40Label = null;
private JTextField r41 = null;
private JLabel r41Label = null;
private JLabel r43Label = null;
private JLabel r42Label = null;
private JTextField r42 = null;
private JTextField r43 = null;
```



```
private JTextField r44 = null;
private JLabel r44Label = null;
private JTextField r45 = null;
private JLabel r45Label = null;
private JTextField r46 = null;
private JLabel r46Label = null;
private JTextField r47 = null;
private JLabel r47Label = null;
private JTextField r48 = null;
private JLabel r48Label = null;
private JTextField r49 = null;
private JLabel r49Label = null;
private JTextField r50 = null;
private JLabel r50Label = null;
private JTextField r51 = null;
private JLabel r51Label = null;
private JTextField r52 = null;
private JLabel r52Label = null;
private JTextField r53 = null;
private JLabel r53Label = null;
private JTextField r54 = null;
private JLabel r54Label = null;
private JTextField r55 = null;
private JLabel r55Label = null;
private JTextField r56 = null;
private JLabel r56Label = null;
private JTextField r57 = null;
private JLabel r57Label = null;
private JTextField r58 = null;
private JLabel r58Label = null;
private JTextField r59 = null;
private JLabel r59Label = null;
private JTextField r60 = null;
private JLabel r60Label = null;
private JTextField l1 = null;
private JLabel l1Label = null;
private JLabel l3Label = null;
private JLabel l2Label = null;
private JTextField l2 = null;
private JTextField l3 = null;
private JTextField l4 = null;
private JLabel l4Label = null;
private JTextField l5 = null;
private JLabel l5Label = null;
private JTextField l6 = null;
private JLabel l6Label = null;
private JTextField l7 = null;
private JLabel l7Label = null;
private JTextField l8 = null;
private JLabel l8Label = null;
private JTextField l9 = null;
private JLabel l9Label = null;
private JTextField l10 = null;
private JLabel l10Label = null;
private JTextField l11 = null;
private JLabel l11Label = null;
```

```

private JTextField l12 = null;
private JLabel l12Label = null;
private JTextField l13 = null;
private JLabel l13Label = null;
private JTextField l14 = null;
private JLabel l14Label = null;
private JTextField l15 = null;
private JLabel l15Label = null;
private JTextField l16 = null;
private JLabel l16Label = null;
private JTextField l17 = null;
private JLabel l17Label = null;
private JTextField l18 = null;
private JLabel l18Label = null;
private JTextField l19 = null;
private JLabel l19Label = null;
private JTextField l20 = null;
private JLabel l20Label = null;
public OperacionesOIT(JPanel como) {
    super();
    initialize();
}

/**
 *
 */
private void initialize() {
    r1Label = new JLabel();
    r1Label.setText("FR1");
    r2Label = new JLabel();
    r2Label.setText("FR2");
    r3Label = new JLabel();
    r3Label.setText("FR3");
    r4Label = new JLabel();
    r4Label.setText("FR4");
    r5Label = new JLabel();
    r5Label.setText("FR5");
    r6Label = new JLabel();
    r6Label.setText("FR6");
    r7Label = new JLabel();
    r7Label.setText("FR7");
    r8Label = new JLabel();
    r8Label.setText("FR8");
    r9Label = new JLabel();
    r9Label.setText("FR9");
    r10Label = new JLabel();
    r10Label.setText("FR10");
    r11Label = new JLabel();
    r11Label.setText("FR11");
    r12Label = new JLabel();
    r12Label.setText("FR12");
    r13Label = new JLabel();
    r13Label.setText("FR13");
    r14Label = new JLabel();
    r14Label.setText("FR14");
    r15Label = new JLabel();

```

```
r15Label.setText("FR15");
r16Label = new JLabel();
r16Label.setText("FR16");
r17Label = new JLabel();
r17Label.setText("FR17");
r18Label = new JLabel();
r18Label.setText("FR18");
r19Label = new JLabel();
r19Label.setText("FR19");
    r20Label = new JLabel();
r20Label.setText("FR20");
    r21Label = new JLabel();
r21Label.setText("FR21");
r22Label = new JLabel();
r22Label.setText("FR22");
r23Label = new JLabel();
r23Label.setText("FR23");
r24Label = new JLabel();
r24Label.setText("FR24");
r25Label = new JLabel();
r25Label.setText("FR25");
r26Label = new JLabel();
r26Label.setText("FR26");
r27Label = new JLabel();
r27Label.setText("FR27");
r28Label = new JLabel();
r28Label.setText("FR28");
r29Label = new JLabel();
r29Label.setText("FR29");
r30Label = new JLabel();
r30Label.setText("FR30");
r31Label = new JLabel();
r31Label.setText("FR31");
r32Label = new JLabel();
r32Label.setText("FR32");
r33Label = new JLabel();
r33Label.setText("FR33");
r34Label = new JLabel();
r34Label.setText("FR34");
r35Label = new JLabel();
r35Label.setText("FR35");
r36Label = new JLabel();
r36Label.setText("FR36");
r37Label = new JLabel();
r37Label.setText("FR37");
r38Label = new JLabel();
r38Label.setText("FR38");
r39Label = new JLabel();
r39Label.setText("FR39");
    r40Label = new JLabel();
r40Label.setText("FR40");
    r41Label = new JLabel();
r41Label.setText("FR41");
r42Label = new JLabel();
r42Label.setText("FR42");
r43Label = new JLabel();
```

```
r43Label.setText("FR43");
r44Label = new JLabel();
r44Label.setText("FR44");
r45Label = new JLabel();
r45Label.setText("FR45");
r46Label = new JLabel();
r46Label.setText("FR46");
r47Label = new JLabel();
r47Label.setText("FR47");
r48Label = new JLabel();
r48Label.setText("FR48");
r49Label = new JLabel();
r49Label.setText("FR49");
r50Label = new JLabel();
r50Label.setText("FR50");
r51Label = new JLabel();
r51Label.setText("FR51");
r52Label = new JLabel();
r52Label.setText("FR52");
r53Label = new JLabel();
r53Label.setText("FR53");
r54Label = new JLabel();
r54Label.setText("FR54");
r55Label = new JLabel();
r55Label.setText("FR55");
r56Label = new JLabel();
r56Label.setText("FR56");
r57Label = new JLabel();
r57Label.setText("FR57");
r58Label = new JLabel();
r58Label.setText("FR58");
r59Label = new JLabel();
r59Label.setText("FR59");
    r60Label = new JLabel();
r60Label.setText("FR60");
l1Label = new JLabel();
l1Label.setText("C1");
l2Label = new JLabel();
l2Label.setText("C2");
l3Label = new JLabel();
l3Label.setText("C3");
l4Label = new JLabel();
l4Label.setText("C4");
l5Label = new JLabel();
l5Label.setText("C5");
l6Label = new JLabel();
l6Label.setText("C6");
l7Label = new JLabel();
l7Label.setText("C7");
l8Label = new JLabel();
l8Label.setText("C8");
l9Label = new JLabel();
l9Label.setText("C9");
l10Label = new JLabel();
l10Label.setText("C10");
l11Label = new JLabel();
```

```

l11Label.setText("C11");
l12Label = new JLabel();
l12Label.setText("C12");
l13Label = new JLabel();
l13Label.setText("C13");
l14Label = new JLabel();
l14Label.setText("C14");
l15Label = new JLabel();
l15Label.setText("C15");
l16Label = new JLabel();
l16Label.setText("C16");
l17Label = new JLabel();
l17Label.setText("C17");
l18Label = new JLabel();
l18Label.setText("C18");
l19Label = new JLabel();
l19Label.setText("C19");
    l20Label = new JLabel();
l20Label.setText("C20");
    this.setSize(700, 700);
    this.setLayout(new GridBagLayout());
    this.add(r1Label, new MargenConstrains(0,0));
    this.add(getr1(), new MargenConstrains(1,0));
    this.add(r2Label, new MargenConstrains(2,0));
    this.add(getr2(), new MargenConstrains(3,0));
    this.add(r3Label, new MargenConstrains(4,0));
    this.add(getr3(), new MargenConstrains(5,0));
    this.add(r4Label, new MargenConstrains(6,0));
    this.add(getr4(), new MargenConstrains(7,0));
    this.add(r5Label, new MargenConstrains(0,1));
    this.add(getr5(), new MargenConstrains(1,1));
    this.add(r6Label, new MargenConstrains(2,1));
    this.add(getr6(), new MargenConstrains(3,1));
    this.add(r7Label, new MargenConstrains(4,1));
    this.add(getr7(), new MargenConstrains(5,1));
    this.add(r8Label, new MargenConstrains(6,1));
    this.add(getr8(), new MargenConstrains(7,1));
    this.add(r9Label, new MargenConstrains(0,2));
    this.add(getr9(), new MargenConstrains(1,2));
    this.add(r10Label, new MargenConstrains(2,2));
    this.add(getr10(), new MargenConstrains(3,2));
    this.add(r11Label, new MargenConstrains(4,2));
    this.add(getr11(), new MargenConstrains(5,2));
    this.add(r12Label, new MargenConstrains(6,2));
    this.add(getr12(), new MargenConstrains(7,2));
    this.add(r13Label, new MargenConstrains(0,3));
    this.add(getr13(), new MargenConstrains(1,3));
    this.add(r14Label, new MargenConstrains(2,3));
    this.add(getr14(), new MargenConstrains(3,3));
    this.add(r15Label, new MargenConstrains(4,3));
    this.add(getr15(), new MargenConstrains(5,3));
    this.add(r16Label, new MargenConstrains(6,3));
    this.add(getr16(), new MargenConstrains(7,3));
    this.add(r17Label, new MargenConstrains(0,4));
    this.add(getr17(), new MargenConstrains(1,4));
    this.add(r18Label, new MargenConstrains(2,4));

```

```

this.add(getr18(), new MargenConstrains(3,5));
this.add(r19Label, new MargenConstrains(4,4));
this.add(getr19(), new MargenConstrains(5,4));
this.add(r20Label, new MargenConstrains(4,4));
this.add(getr20(), new MargenConstrains(5,4));
this.add(r1Label, new MargenConstrains(0,0));
this.add(getr1(), new MargenConstrains(1,0));
this.add(r2Label, new MargenConstrains(2,0));
this.add(getr2(), new MargenConstrains(3,0));
this.add(r3Label, new MargenConstrains(4,0));
this.add(getr3(), new MargenConstrains(5,0));
this.add(r4Label, new MargenConstrains(6,0));
this.add(getr4(), new MargenConstrains(7,0));
this.add(r5Label, new MargenConstrains(0,1));
this.add(getr5(), new MargenConstrains(1,1));
this.add(r6Label, new MargenConstrains(2,1));
this.add(getr6(), new MargenConstrains(3,1));
this.add(r7Label, new MargenConstrains(4,1));
this.add(getr7(), new MargenConstrains(5,1));
this.add(r8Label, new MargenConstrains(6,1));
this.add(getr8(), new MargenConstrains(7,1));
this.add(r9Label, new MargenConstrains(0,2));
this.add(getr9(), new MargenConstrains(1,2));
this.add(r10Label, new MargenConstrains(2,2));
this.add(getr10(), new MargenConstrains(3,2));
this.add(r11Label, new MargenConstrains(4,2));
this.add(getr11(), new MargenConstrains(5,2));
this.add(r12Label, new MargenConstrains(6,2));
this.add(getr12(), new MargenConstrains(7,2));
this.add(r13Label, new MargenConstrains(0,3));
this.add(getr13(), new MargenConstrains(1,3));
this.add(r14Label, new MargenConstrains(2,3));
this.add(getr14(), new MargenConstrains(3,3));
this.add(r15Label, new MargenConstrains(4,3));
this.add(getr15(), new MargenConstrains(5,3));
this.add(r16Label, new MargenConstrains(6,3));
this.add(getr16(), new MargenConstrains(7,3));
this.add(r17Label, new MargenConstrains(0,4));
this.add(getr17(), new MargenConstrains(1,4));
this.add(r18Label, new MargenConstrains(2,4));
this.add(getr18(), new MargenConstrains(3,5));
this.add(r19Label, new MargenConstrains(4,4));
this.add(getr19(), new MargenConstrains(5,4));
this.add(r1Label, new MargenConstrains(0,0));
this.add(getr1(), new MargenConstrains(1,0));
this.add(r2Label, new MargenConstrains(2,0));
this.add(getr2(), new MargenConstrains(3,0));
this.add(r3Label, new MargenConstrains(4,0));
this.add(getr3(), new MargenConstrains(5,0));
this.add(r4Label, new MargenConstrains(6,0));
this.add(getr4(), new MargenConstrains(7,0));
this.add(r5Label, new MargenConstrains(0,1));
this.add(getr5(), new MargenConstrains(1,1));
this.add(r6Label, new MargenConstrains(2,1));
this.add(getr6(), new MargenConstrains(3,1));
this.add(r7Label, new MargenConstrains(4,1));

```

```

this.add(getr7(), new MargenConstrains(5,1));
this.add(r8Label, new MargenConstrains(6,1));
this.add(getr8(), new MargenConstrains(7,1));
this.add(r9Label, new MargenConstrains(0,2));
this.add(getr9(), new MargenConstrains(1,2));
this.add(r10Label, new MargenConstrains(2,2));
this.add(getr10(), new MargenConstrains(3,2));
this.add(r11Label, new MargenConstrains(4,2));
this.add(getr11(), new MargenConstrains(5,2));
this.add(r12Label, new MargenConstrains(6,2));
this.add(getr12(), new MargenConstrains(7,2));
this.add(r13Label, new MargenConstrains(0,3));
this.add(getr13(), new MargenConstrains(1,3));
this.add(r14Label, new MargenConstrains(2,3));
this.add(getr14(), new MargenConstrains(3,3));
this.add(r15Label, new MargenConstrains(4,3));
this.add(getr15(), new MargenConstrains(5,3));
this.add(r16Label, new MargenConstrains(6,3));
this.add(getr16(), new MargenConstrains(7,3));
this.add(r17Label, new MargenConstrains(0,4));
this.add(getr17(), new MargenConstrains(1,4));
this.add(r18Label, new MargenConstrains(2,4));
this.add(getr18(), new MargenConstrains(3,5));
this.add(r19Label, new MargenConstrains(4,4));
this.add(getr19(), new MargenConstrains(5,4));
this.add(r20Label, new MargenConstrains(6,4));
this.add(getr20(), new MargenConstrains(7,4));
this.add(r1Label, new MargenConstrains(0,0));
this.add(getr1(), new MargenConstrains(1,0));
this.add(r2Label, new MargenConstrains(2,0));
this.add(getr2(), new MargenConstrains(3,0));
this.add(r3Label, new MargenConstrains(4,0));
this.add(getr3(), new MargenConstrains(5,0));
this.add(r4Label, new MargenConstrains(6,0));
this.add(getr4(), new MargenConstrains(7,0));
this.add(r5Label, new MargenConstrains(0,1));
this.add(getr5(), new MargenConstrains(1,1));
this.add(r6Label, new MargenConstrains(2,1));
this.add(getr6(), new MargenConstrains(3,1));
this.add(r7Label, new MargenConstrains(4,1));
this.add(getr7(), new MargenConstrains(5,1));
this.add(r8Label, new MargenConstrains(6,1));
this.add(getr8(), new MargenConstrains(7,1));
this.add(r9Label, new MargenConstrains(0,2));
this.add(getr9(), new MargenConstrains(1,2));
this.add(r10Label, new MargenConstrains(2,2));
this.add(getr10(), new MargenConstrains(3,2));
this.add(r11Label, new MargenConstrains(4,2));
this.add(getr11(), new MargenConstrains(5,2));
this.add(r12Label, new MargenConstrains(6,2));
this.add(getr12(), new MargenConstrains(7,2));
this.add(r13Label, new MargenConstrains(0,3));
this.add(getr13(), new MargenConstrains(1,3));
this.add(r14Label, new MargenConstrains(2,3));
this.add(getr14(), new MargenConstrains(3,3));
this.add(r15Label, new MargenConstrains(4,3));

```

```
this.add(getr15(), new MargenConstrains(5,3));
this.add(r16Label, new MargenConstrains(6,3));
this.add(getr16(), new MargenConstrains(7,3));
this.add(r17Label, new MargenConstrains(0,4));
this.add(getr17(), new MargenConstrains(1,4));
this.add(r18Label, new MargenConstrains(2,4));
this.add(getr18(), new MargenConstrains(3,5));
this.add(r19Label, new MargenConstrains(4,4));
this.add(getr19(), new MargenConstrains(5,4));
this.add(r20Label, new MargenConstrains(4,4));
this.add(getr20(), new MargenConstrains(5,4));
this.add(r1Label, new MargenConstrains(0,0));
this.add(getr1(), new MargenConstrains(1,0));
this.add(r2Label, new MargenConstrains(2,0));
this.add(getr2(), new MargenConstrains(3,0));
this.add(r3Label, new MargenConstrains(4,0));
this.add(getr3(), new MargenConstrains(5,0));
this.add(r4Label, new MargenConstrains(6,0));
this.add(getr4(), new MargenConstrains(7,0));
this.add(r5Label, new MargenConstrains(0,1));
this.add(getr5(), new MargenConstrains(1,1));
this.add(r6Label, new MargenConstrains(2,1));
this.add(getr6(), new MargenConstrains(3,1));
this.add(r7Label, new MargenConstrains(4,1));
this.add(getr7(), new MargenConstrains(5,1));
this.add(r8Label, new MargenConstrains(6,1));
this.add(getr8(), new MargenConstrains(7,1));
this.add(r9Label, new MargenConstrains(0,2));
this.add(getr9(), new MargenConstrains(1,2));
this.add(r10Label, new MargenConstrains(2,2));
this.add(getr10(), new MargenConstrains(3,2));
this.add(r11Label, new MargenConstrains(4,2));
this.add(getr11(), new MargenConstrains(5,2));
this.add(r12Label, new MargenConstrains(6,2));
this.add(getr12(), new MargenConstrains(7,2));
this.add(r13Label, new MargenConstrains(0,3));
this.add(getr13(), new MargenConstrains(1,3));
this.add(r14Label, new MargenConstrains(2,3));
this.add(getr14(), new MargenConstrains(3,3));
this.add(r15Label, new MargenConstrains(4,3));
this.add(getr15(), new MargenConstrains(5,3));
this.add(r16Label, new MargenConstrains(6,3));
this.add(getr16(), new MargenConstrains(7,3));
this.add(r17Label, new MargenConstrains(0,4));
this.add(getr17(), new MargenConstrains(1,4));
this.add(r18Label, new MargenConstrains(2,4));
this.add(getr18(), new MargenConstrains(3,5));
this.add(r19Label, new MargenConstrains(4,4));
this.add(getr19(), new MargenConstrains(5,4));
this.add(r1Label, new MargenConstrains(0,0));
this.add(getr1(), new MargenConstrains(1,0));
this.add(r2Label, new MargenConstrains(2,0));
this.add(getr2(), new MargenConstrains(3,0));
this.add(r3Label, new MargenConstrains(4,0));
this.add(getr3(), new MargenConstrains(5,0));
this.add(r4Label, new MargenConstrains(6,0));
```


ANEXO

```
this.add(getr4(), new MargenConstrains(7,0));
this.add(r5Label, new MargenConstrains(0,1));
this.add(getr5(), new MargenConstrains(1,1));
this.add(r6Label, new MargenConstrains(2,1));
this.add(getr6(), new MargenConstrains(3,1));
this.add(r7Label, new MargenConstrains(4,1));
this.add(getr7(), new MargenConstrains(5,1));
this.add(r8Label, new MargenConstrains(6,1));
this.add(getr8(), new MargenConstrains(7,1));
this.add(r9Label, new MargenConstrains(0,2));
this.add(getr9(), new MargenConstrains(1,2));
this.add(r10Label, new MargenConstrains(2,2));
this.add(getr10(), new MargenConstrains(3,2));
this.add(r11Label, new MargenConstrains(4,2));
this.add(getr11(), new MargenConstrains(5,2));
this.add(r12Label, new MargenConstrains(6,2));
this.add(getr12(), new MargenConstrains(7,2));
this.add(r13Label, new MargenConstrains(0,3));
this.add(getr13(), new MargenConstrains(1,3));
this.add(r14Label, new MargenConstrains(2,3));
this.add(getr14(), new MargenConstrains(3,3));
this.add(r15Label, new MargenConstrains(4,3));
this.add(getr15(), new MargenConstrains(5,3));
this.add(r16Label, new MargenConstrains(6,3));
this.add(getr16(), new MargenConstrains(7,3));
this.add(r17Label, new MargenConstrains(0,4));
this.add(getr17(), new MargenConstrains(1,4));
this.add(r18Label, new MargenConstrains(2,4));
this.add(getr18(), new MargenConstrains(3,5));
this.add(r19Label, new MargenConstrains(4,4));
this.add(getr19(), new MargenConstrains(5,4));
this.add(r20Label, new MargenConstrains(6,4));
this.add(getr20(), new MargenConstrains(7,4));
this.add(r1Label, new MargenConstrains(0,0));
this.add(getr1(), new MargenConstrains(1,0));
this.add(r2Label, new MargenConstrains(2,0));
this.add(getr2(), new MargenConstrains(3,0));
this.add(r3Label, new MargenConstrains(4,0));
this.add(getr3(), new MargenConstrains(5,0));
this.add(r4Label, new MargenConstrains(6,0));
this.add(getr4(), new MargenConstrains(7,0));
this.add(r5Label, new MargenConstrains(0,1));
this.add(getr5(), new MargenConstrains(1,1));
this.add(r6Label, new MargenConstrains(2,1));
this.add(getr6(), new MargenConstrains(3,1));
this.add(r7Label, new MargenConstrains(4,1));
this.add(getr7(), new MargenConstrains(5,1));
this.add(r8Label, new MargenConstrains(6,1));
this.add(getr8(), new MargenConstrains(7,1));
this.add(r9Label, new MargenConstrains(0,2));
this.add(getr9(), new MargenConstrains(1,2));
this.add(r10Label, new MargenConstrains(2,2));
this.add(getr10(), new MargenConstrains(3,2));
this.add(r11Label, new MargenConstrains(4,2));
this.add(getr11(), new MargenConstrains(5,2));
this.add(r12Label, new MargenConstrains(6,2));
```

```
this.add(getr12(), new MargenConstrains(7,2));
this.add(r13Label, new MargenConstrains(0,3));
this.add(getr13(), new MargenConstrains(1,3));
this.add(r14Label, new MargenConstrains(2,3));
this.add(getr14(), new MargenConstrains(3,3));
this.add(r15Label, new MargenConstrains(4,3));
this.add(getr15(), new MargenConstrains(5,3));
this.add(r16Label, new MargenConstrains(6,3));
this.add(getr16(), new MargenConstrains(7,3));
this.add(r17Label, new MargenConstrains(0,4));
this.add(getr17(), new MargenConstrains(1,4));
this.add(r18Label, new MargenConstrains(2,4));
this.add(getr18(), new MargenConstrains(3,4));
this.add(r19Label, new MargenConstrains(4,4));
this.add(getr19(), new MargenConstrains(5,4));
this.add(r20Label, new MargenConstrains(6,4));
this.add(getr20(), new MargenConstrains(7,4));
this.add(r21Label, new MargenConstrains(0,5));
this.add(getr21(), new MargenConstrains(1,5));
this.add(r22Label, new MargenConstrains(2,5));
this.add(getr22(), new MargenConstrains(3,5));
this.add(r23Label, new MargenConstrains(4,5));
this.add(getr23(), new MargenConstrains(5,5));
this.add(r24Label, new MargenConstrains(6,5));
this.add(getr24(), new MargenConstrains(7,5));
this.add(r25Label, new MargenConstrains(0,6));
this.add(getr25(), new MargenConstrains(1,6));
this.add(r26Label, new MargenConstrains(2,6));
this.add(getr26(), new MargenConstrains(3,6));
this.add(r27Label, new MargenConstrains(4,6));
this.add(getr27(), new MargenConstrains(5,6));
this.add(r28Label, new MargenConstrains(6,6));
this.add(getr28(), new MargenConstrains(7,6));
this.add(r29Label, new MargenConstrains(0,7));
this.add(getr29(), new MargenConstrains(1,7));
this.add(r30Label, new MargenConstrains(2,7));
this.add(getr30(), new MargenConstrains(3,7));
this.add(r31Label, new MargenConstrains(4,7));
this.add(getr31(), new MargenConstrains(5,7));
this.add(r32Label, new MargenConstrains(6,7));
this.add(getr32(), new MargenConstrains(7,7));
this.add(r33Label, new MargenConstrains(0,8));
this.add(getr33(), new MargenConstrains(1,8));
this.add(r34Label, new MargenConstrains(2,8));
this.add(getr34(), new MargenConstrains(3,8));
this.add(r35Label, new MargenConstrains(4,8));
this.add(getr35(), new MargenConstrains(5,8));
this.add(r36Label, new MargenConstrains(6,8));
this.add(getr36(), new MargenConstrains(7,8));
this.add(r37Label, new MargenConstrains(0,9));
this.add(getr37(), new MargenConstrains(1,9));
this.add(r38Label, new MargenConstrains(2,9));
this.add(getr38(), new MargenConstrains(3,9));
this.add(r39Label, new MargenConstrains(4,9));
this.add(getr39(), new MargenConstrains(5,9));
this.add(r40Label, new MargenConstrains(6,9));
```

```

this.add(getr40(), new MargenConstrains(7,9));
this.add(r41Label, new MargenConstrains(0,10));
this.add(getr41(), new MargenConstrains(1,10));
this.add(r42Label, new MargenConstrains(2,10));
this.add(getr42(), new MargenConstrains(3,10));
this.add(r43Label, new MargenConstrains(4,10));
this.add(getr43(), new MargenConstrains(5,10));
this.add(r44Label, new MargenConstrains(6,10));
this.add(getr44(), new MargenConstrains(7,10));
this.add(r45Label, new MargenConstrains(0,11));
this.add(getr45(), new MargenConstrains(1,11));
this.add(r46Label, new MargenConstrains(2,11));
this.add(getr46(), new MargenConstrains(3,11));
this.add(r47Label, new MargenConstrains(4,11));
this.add(getr47(), new MargenConstrains(5,11));
this.add(r48Label, new MargenConstrains(6,11));
this.add(getr48(), new MargenConstrains(7,11));
this.add(r49Label, new MargenConstrains(0,12));
this.add(getr49(), new MargenConstrains(1,12));
this.add(r50Label, new MargenConstrains(2,12));
this.add(getr50(), new MargenConstrains(3,12));
this.add(r51Label, new MargenConstrains(4,12));
this.add(getr51(), new MargenConstrains(5,12));
this.add(r52Label, new MargenConstrains(6,12));
this.add(getr52(), new MargenConstrains(7,12));
this.add(r53Label, new MargenConstrains(0,13));
this.add(getr53(), new MargenConstrains(1,13));
this.add(r54Label, new MargenConstrains(2,13));
this.add(getr54(), new MargenConstrains(3,13));
this.add(r55Label, new MargenConstrains(4,13));
this.add(getr55(), new MargenConstrains(5,13));
this.add(r56Label, new MargenConstrains(6,13));
this.add(getr56(), new MargenConstrains(7,13));
this.add(r57Label, new MargenConstrains(0,14));
this.add(getr57(), new MargenConstrains(1,14));
this.add(r58Label, new MargenConstrains(2,14));
this.add(getr58(), new MargenConstrains(3,14));
this.add(r59Label, new MargenConstrains(4,14));
this.add(getr59(), new MargenConstrains(5,14));
this.add(r60Label, new MargenConstrains(6,14));
this.add(getr60(), new MargenConstrains(7,14));

this.add(l11Label, new MargenConstrains(0,15));
this.add(getl11(), new MargenConstrains(1,15));
this.add(l12Label, new MargenConstrains(2,15));
this.add(getl12(), new MargenConstrains(3,15));
this.add(l13Label, new MargenConstrains(4,15));
this.add(getl13(), new MargenConstrains(5,15));
this.add(l14Label, new MargenConstrains(6,15));
this.add(getl14(), new MargenConstrains(7,15));
this.add(l15Label, new MargenConstrains(0,16));
this.add(getl15(), new MargenConstrains(1,16));
this.add(l16Label, new MargenConstrains(2,16));
this.add(getl16(), new MargenConstrains(3,16));
this.add(l17Label, new MargenConstrains(4,16));
this.add(getl17(), new MargenConstrains(5,16));

```

```

        this.add(l18Label, new MargenConstrains(6,16));
        this.add(getl18(), new MargenConstrains(7,16));
        this.add(l19Label, new MargenConstrains(0,17));
        this.add(getl19(), new MargenConstrains(1,17));
        this.add(l110Label, new MargenConstrains(2,17));
        this.add(getl110(), new MargenConstrains(3,17));
        this.add(l111Label, new MargenConstrains(4,17));
        this.add(getl111(), new MargenConstrains(5,17));
        this.add(l112Label, new MargenConstrains(6,17));
        this.add(getl112(), new MargenConstrains(7,17));
        this.add(l113Label, new MargenConstrains(0,18));
        this.add(getl113(), new MargenConstrains(1,18));
        this.add(l114Label, new MargenConstrains(2,18));
        this.add(getl114(), new MargenConstrains(3,18));
        this.add(l115Label, new MargenConstrains(4,18));
        this.add(getl115(), new MargenConstrains(5,18));
        this.add(l116Label, new MargenConstrains(6,18));
        this.add(getl116(), new MargenConstrains(7,18));
        this.add(l117Label, new MargenConstrains(0,19));
        this.add(getl117(), new MargenConstrains(1,19));
        this.add(l118Label, new MargenConstrains(2,19));
        this.add(getl118(), new MargenConstrains(3,19));
        this.add(l119Label, new MargenConstrains(4,19));
        this.add(getl119(), new MargenConstrains(5,19));
        this.add(l120Label, new MargenConstrains(6,19));
        this.add(getl120(), new MargenConstrains(7,19));
        this.add(getAltaMaquinaBoton(), new
MargenConstrains(8,20));
    }
    /**
     * @return
     */
    private JButton getAltaMaquinaBoton() {
        if (altaMaquinaBoton == null) {
            altaMaquinaBoton = new JButton();
            altaMaquinaBoton.setText("Add");
            altaMaquinaBoton.addActionListener(new
java.awt.event.ActionListener() {
                public void
actionPerformed(java.awt.event.ActionEvent e) {
                    intril =0;
                    if((r1.getText() !=null) && (!"".equals(r1.getText()))){
                        ril=Integer.parseInt(r1.getText());
                    }
                    int ri2 = 0;
                    if((r2.getText() !=null) && (!"".equals(r2.getText()))){
                        ri2=Integer.parseInt(r2.getText());
                    }
                    int ri3 = 0;
                    if((r3.getText() !=null) && (!"".equals(r3.getText()))){
                        ri3=Integer.parseInt(r3.getText());
                    }
                    int ri4 = 0;

                    if((r4.getText() !=null) && (!"".equals(r4.getText()))){
                        ri4=Integer.parseInt(r4.getText());
                    }
                }
            });
    }

```

```

    }
    int ri5 = 0;
if((r5.getText() != null) && (!"".equals(r5.getText()))){
    ri5=Integer.parseInt(r5.getText());
}
    int ri6 = 0;
if((r6.getText() != null) && (!"".equals(r6.getText()))){
    ri6=Integer.parseInt(r6.getText());
}
    int ri7 = 0;
if((r7.getText() != null) && (!"".equals(r7.getText()))){
    ri7=Integer.parseInt(r7.getText());
}
    int ri8 = 0;
if((r8.getText() != null) && (!"".equals(r8.getText()))){
    ri8=Integer.parseInt(r8.getText());
}
    int ri9 = 0;
if((r9.getText() != null) && (!"".equals(r9.getText()))){
    ri9=Integer.parseInt(r9.getText());
}
    int ri10 = 0;

if((r10.getText() != null) && (!"".equals(r10.getText()))){
    ri10=Integer.parseInt(r10.getText());
}
    int ri11 =0;
if((r11.getText() != null) && (!"".equals(r11.getText()))){
    ri11=Integer.parseInt(r11.getText());
}
    int ri12 = 0;
if((r12.getText() != null) && (!"".equals(r12.getText()))){
    ri12=Integer.parseInt(r12.getText());
}
    int ri13 = 0;
if((r13.getText() != null) && (!"".equals(r13.getText()))){
    ri13=Integer.parseInt(r13.getText());
}
    int ri14 = 0;
if((r14.getText() != null) && (!"".equals(r14.getText()))){
    ri14=Integer.parseInt(r14.getText());
}
    int ri15 = 0;
if((r15.getText() != null) && (!"".equals(r15.getText()))){
    ri15=Integer.parseInt(r15.getText());
}
    int ri16 = 0;
if((r16.getText() != null) && (!"".equals(r16.getText()))){
    ri16=Integer.parseInt(r16.getText());
}
    int ri17 = 0;

if((r17.getText() != null) && (!"".equals(r17.getText()))){
    ri17=Integer.parseInt(r17.getText());
}
    int ri18 = 0;

```

```

if((r18.getText()!=null)&&(!"".equals(r18.getText()))){
    ri18=Integer.parseInt(r18.getText());
}
int ri19 = 0;
if((r19.getText()!=null)&&(!"".equals(r19.getText()))){
    ri19=Integer.parseInt(r19.getText());
}
int ri20 = 0;
if((r20.getText()!=null)&&(!"".equals(r20.getText()))){
    ri20=Integer.parseInt(r20.getText());
}
int ri21 =0;
if((r21.getText()!=null)&&(!"".equals(r21.getText()))){
    ri21=Integer.parseInt(r21.getText());
}
int ri22 = 0;
if((r22.getText()!=null)&&(!"".equals(r22.getText()))){
    ri22=Integer.parseInt(r22.getText());
}
int ri23 = 0;
if((r23.getText()!=null)&&(!"".equals(r23.getText()))){
    ri23=Integer.parseInt(r23.getText());
}
int ri24 = 0;
if((r24.getText()!=null)&&(!"".equals(r24.getText()))){
    ri24=Integer.parseInt(r24.getText());
}
int ri25 = 0;
if((r25.getText()!=null)&&(!"".equals(r25.getText()))){
    ri25=Integer.parseInt(r25.getText());
}
int ri26 = 0;
if((r26.getText()!=null)&&(!"".equals(r26.getText()))){
    ri26=Integer.parseInt(r26.getText());
}
int ri27 = 0;
if((r27.getText()!=null)&&(!"".equals(r27.getText()))){
    ri27=Integer.parseInt(r27.getText());
}
int ri28 = 0;

if((r28.getText()!=null)&&(!"".equals(r28.getText()))){
    ri28=Integer.parseInt(r28.getText());
}
int ri29 = 0;
if((r29.getText()!=null)&&(!"".equals(r29.getText()))){
    ri29=Integer.parseInt(r29.getText());
}
int ri30 = 0;
if((r30.getText()!=null)&&(!"".equals(r30.getText()))){
    ri30=Integer.parseInt(r30.getText());
}
int ri31 =0;
if((r31.getText()!=null)&&(!"".equals(r31.getText()))){
    ri31=Integer.parseInt(r31.getText());
}
}

```

ANEXO

```
int ri32 = 0;

if((r32.getText()!=null)&&(!"".equals(r32.getText()))){
    ri32=Integer.parseInt(r32.getText());
}
int ri33 = 0;

if((r33.getText()!=null)&&(!"".equals(r33.getText()))){
    ri33=Integer.parseInt(r33.getText());
}
int ri34 = 0;

if((r34.getText()!=null)&&(!"".equals(r34.getText()))){
    ri14=Integer.parseInt(r34.getText());
}
int ri35 = 0;

if((r35.getText()!=null)&&(!"".equals(r35.getText()))){
    ri35=Integer.parseInt(r35.getText());
}
int ri36 = 0;

if((r36.getText()!=null)&&(!"".equals(r36.getText()))){
    ri36=Integer.parseInt(r36.getText());
}
int ri37 = 0;

if((r37.getText()!=null)&&(!"".equals(r37.getText()))){
    ri37=Integer.parseInt(r37.getText());
}
int ri38 = 0;

if((r38.getText()!=null)&&(!"".equals(r38.getText()))){
    ri38=Integer.parseInt(r38.getText());
}
int ri39 = 0;

if((r39.getText()!=null)&&(!"".equals(r39.getText()))){
    ri39=Integer.parseInt(r39.getText());
}
int ri40 = 0;

if((r40.getText()!=null)&&(!"".equals(r40.getText()))){
    ri40=Integer.parseInt(r40.getText());
}
int ri41 =0;

if((r41.getText()!=null)&&(!"".equals(r41.getText()))){
    ri41=Integer.parseInt(r41.getText());
}
int ri42 = 0;

if((r42.getText()!=null)&&(!"".equals(r42.getText()))){
    ri42=Integer.parseInt(r42.getText());
}
int ri43 = 0;
```

ANEXO

```
if((r43.getText()!=null)&&(!"".equals(r43.getText()))){
    ri43=Integer.parseInt(r43.getText());
}
int ri44 = 0;

if((r44.getText()!=null)&&(!"".equals(r44.getText()))){
    ri44=Integer.parseInt(r44.getText());
}
int ri45 = 0;

if((r45.getText()!=null)&&(!"".equals(r45.getText()))){
    ri45=Integer.parseInt(r45.getText());
}
int ri46 = 0;

if((r46.getText()!=null)&&(!"".equals(r46.getText()))){
    ri46=Integer.parseInt(r46.getText());
}
int ri47 = 0;

if((r47.getText()!=null)&&(!"".equals(r47.getText()))){
    ri47=Integer.parseInt(r47.getText());
}
int ri48 = 0;

if((r48.getText()!=null)&&(!"".equals(r48.getText()))){
    ri48=Integer.parseInt(r48.getText());
}
int ri49 = 0;

if((r49.getText()!=null)&&(!"".equals(r49.getText()))){
    ri49=Integer.parseInt(r49.getText());
}
int ri50 = 0;

if((r50.getText()!=null)&&(!"".equals(r50.getText()))){
    ri50=Integer.parseInt(r50.getText());
}
int ri51 =0;

if((r51.getText()!=null)&&(!"".equals(r51.getText()))){
    ri51=Integer.parseInt(r51.getText());
}
int ri52 = 0;

if((r52.getText()!=null)&&(!"".equals(r52.getText()))){
    ri52=Integer.parseInt(r52.getText());
}
int ri53 = 0;

if((r53.getText()!=null)&&(!"".equals(r53.getText()))){
    ri53=Integer.parseInt(r53.getText());
}
int ri54 = 0;
```


ANEXO

```
if((r54.getText()!=null)&&(!"".equals(r54.getText()))){
    ri54=Integer.parseInt(r54.getText());
}
int ri55 = 0;

if((r55.getText()!=null)&&(!"".equals(r55.getText()))){
    ri55=Integer.parseInt(r55.getText());
}
int ri56 = 0;

if((r56.getText()!=null)&&(!"".equals(r56.getText()))){
    ri56=Integer.parseInt(r56.getText());
}
int ri57 = 0;

if((r57.getText()!=null)&&(!"".equals(r57.getText()))){
    ri57=Integer.parseInt(r57.getText());
}
int ri58 = 0;

if((r58.getText()!=null)&&(!"".equals(r58.getText()))){
    ri58=Integer.parseInt(r58.getText());
}
int ri59 = 0;

if((r59.getText()!=null)&&(!"".equals(r59.getText()))){
    ri59=Integer.parseInt(r59.getText());
}
int ri60 = 0;

if((r60.getText()!=null)&&(!"".equals(r60.getText()))){
    ri60=Integer.parseInt(r60.getText());
}
int le1 =0;

if((l1.getText()!=null)&&(!"".equals(l1.getText()))){
    le1=Integer.parseInt(l1.getText());
}
int le2 = 0;

if((l2.getText()!=null)&&(!"".equals(l2.getText()))){
    le2=Integer.parseInt(l2.getText());
}
int le3 = 0;

if((l3.getText()!=null)&&(!"".equals(l3.getText()))){
    le3=Integer.parseInt(l3.getText());
}
int le4 = 0;

if((l4.getText()!=null)&&(!"".equals(l4.getText()))){
    le4=Integer.parseInt(l4.getText());
}
int le5 = 0;
```

ANEXO

```
if((l15.getText()!=null)&&(!"".equals(l15.getText()))){
    le5=Integer.parseInt(l15.getText());
}
int le6 = 0;

if((l16.getText()!=null)&&(!"".equals(l16.getText()))){
    le6=Integer.parseInt(l16.getText());
}
int le7 = 0;

if((l17.getText()!=null)&&(!"".equals(l17.getText()))){
    le7=Integer.parseInt(l17.getText());
}
int le8 = 0;

if((l18.getText()!=null)&&(!"".equals(l18.getText()))){
    le8=Integer.parseInt(l18.getText());
}
int le9 = 0;

if((l19.getText()!=null)&&(!"".equals(l19.getText()))){
    le9=Integer.parseInt(l19.getText());
}
int le10 = 0;

if((l110.getText()!=null)&&(!"".equals(l110.getText()))){
    le10=Integer.parseInt(l110.getText());
}
int le11 =0;

if((l111.getText()!=null)&&(!"".equals(l111.getText()))){
    le11=Integer.parseInt(l111.getText());
}
int le12 = 0;

if((l112.getText()!=null)&&(!"".equals(l112.getText()))){
    le12=Integer.parseInt(l112.getText());
}
int le13 = 0;

if((l113.getText()!=null)&&(!"".equals(l113.getText()))){
    le13=Integer.parseInt(l113.getText());
}
int le14 = 0;

if((l114.getText()!=null)&&(!"".equals(l114.getText()))){
    le14=Integer.parseInt(l114.getText());
}
int le15 = 0;

if((l115.getText()!=null)&&(!"".equals(l115.getText()))){
    le15=Integer.parseInt(l115.getText());
}
int le16 = 0;
```

```

if((l16.getText()!=null)&&(!"".equals(l16.getText()))){
    le16=Integer.parseInt(l16.getText());
}
int le17 = 0;

if((l17.getText()!=null)&&(!"".equals(l17.getText()))){
    le17=Integer.parseInt(l17.getText());
}
int le18 = 0;

if((l18.getText()!=null)&&(!"".equals(l18.getText()))){
    le18=Integer.parseInt(l18.getText());
}
int le19 = 0;

if((l19.getText()!=null)&&(!"".equals(l19.getText()))){
    le19=Integer.parseInt(l19.getText());
}
int le20 = 0;

if((l20.getText()!=null)&&(!"".equals(l20.getText()))){
    le20=Integer.parseInt(l20.getText());
}

Controlador_1.getInstance().addAccidentadoAlm4( ri1, ri2, ri3,
ri4, ri5, ri6, ri7, ri8, ri9, ri10, ri11, ri12, ri13,
ri14, ri15, ri16, ri17, ri18, ri19,ri20,ri21, ri22, ri23,
ri24, ri25, ri26, ri27, ri28, ri29, ri30, ri31, ri32, ri33,
ri34, ri35, ri36, ri37, ri38, ri39,ri40,ri41, ri42, ri43,
ri44, ri45, ri46, ri47, ri48, ri49, ri50, ri51, ri52, ri53,
ri54, ri55, ri56, ri57, ri58, ri59,ri60, le1, le2, le3, le4,
le5, le6, le7, le8, le9, le10, le11, le12, le13, le14,
le15, le16, le17, le18, le19, le20);
    }
    });
}
return altaMaquinaBoton;
}

/**
 * @return
 */
private JTextField getl1() {
    if (l1 == null) {
        l1 = new JTextField();
    }
    return l1;
}

/**
 * @return
 */
private JTextField getl2() {
    if (l2 == null) {

```

```
        l2 = new JTextField();
    }
    return l2;
}
/**
 * @return
 */
private JTextField getl3() {
    if (l3 == null) {
        l3 = new JTextField();
    }
    return l3;
}
/**
 * @return
 */
private JTextField getl4() {
    if (l4 == null) {
        l4 = new JTextField();
    }
    return l4;
}
/**
 * @return
 */
private JTextField getl5() {
    if (l5 == null) {
        l5 = new JTextField();
    }
    return l5;
}
/**
 * @return
 */
private JTextField getl6() {
    if (l6 == null) {
        l6 = new JTextField();
    }
    return l6;
}
/**
 * @return
 */
private JTextField getl7() {
    if (l7 == null) {
        l7 = new JTextField();
    }
    return l7;
}
/**
 * @return
 */
private JTextField getl8() {
    if (l8 == null) {
        l8 = new JTextField();
    }
}
```

```

        return l8;
    }
    /**
     * @return
     */
    private JTextField getl9() {
        if (l9 == null) {
            l9 = new JTextField();
        }
        return l9;
    }
    private JTextField getl10() {
        if (l10 == null) {
            l10 = new JTextField();
        }
        return l10;
    }
    /**
     * @return
     */
    private JTextField getl11() {
        if (l11 == null) {
            l11 = new JTextField();
        }
        return l11;
    }
    /**
     * @return
     */
    private JTextField getl12() {
        if (l12 == null) {
            l12 = new JTextField();
        }
        return l12;
    }
    /**
     * @return
     */
    private JTextField getl13() {
        if (l13 == null) {
            l13 = new JTextField();
        }
        return l13;
    }
    /**
     * @return
     */
    private JTextField getl14() {
        if (l14 == null) {
            l14 = new JTextField();
        }
        return l14;
    }
    /**
     * @return
     */

```

```

private JTextField getl15() {
    if (l15 == null) {
        l15 = new JTextField();
    }
    return l15;
}
/**
 * @return
 */
private JTextField getl16() {
    if (l16 == null) {
        l16 = new JTextField();
    }
    return l16;
}
/**
 * @return
 */
private JTextField getl17() {
    if (l17 == null) {
        l17 = new JTextField();
    }
    return l17;
}
/**
 * @return
 */
private JTextField getl18() {
    if (l18 == null) {
        l18 = new JTextField();
    }
    return l18;
}
/**
 * @return
 */
private JTextField getl19() {
    if (l19 == null) {
        l19 = new JTextField();
    }
    return l19;
}
/**
 * @return
 */
private JTextField getl20() {
    if (l20 == null) {
        l20 = new JTextField();
    }
    return l20;
}
/**
 * @return
 */
private JTextField getr1() {
    if (r1 == null) {

```

```

        r1 = new JTextField();
    }
    return r1;
}

/**
 * @return
 */
private JTextField getr2() {
    if (r2 == null) {
        r2 = new JTextField();
    }
    return r2;
}
/**
 * @return
 */
private JTextField getr3() {
    if (r3 == null) {
        r3 = new JTextField();
    }
    return r3;
}
/**
 * @return
 */
private JTextField getr4() {
    if (r4 == null) {
        r4 = new JTextField();
    }
    return r4;
}
/**
 * @return
 */
private JTextField getr5() {
    if (r5 == null) {
        r5 = new JTextField();
    }
    return r5;
}
/**
 * @return
 */
private JTextField getr6() {
    if (r6 == null) {
        r6 = new JTextField();
    }
    return r6;
}
/**
 * @return
 */
private JTextField getr7() {
    if (r7 == null) {
        r7 = new JTextField();
    }
}

```

```

    }
    return r7;
}
/**
 * @return
 */
private JTextField getr8() {
    if (r8 == null) {
        r8 = new JTextField();
    }
    return r8;
}
/**
 * @return
 */
private JTextField getr9() {
    if (r9 == null) {
        r9 = new JTextField();
    }
    return r9;
}
/**
 * @return
 */
private JTextField getr10() {
    if (r10 == null) {
        r10 = new JTextField();
    }
    return r10;
}
/**
 * @return
 */
private JTextField getr11() {
    if (r11 == null) {
        r11 = new JTextField();
    }
    return r11;
}
/**
 * @return
 */
private JTextField getr12() {
    if (r12 == null) {
        r12 = new JTextField();
    }
    return r12;
}
/**
 * @return
 */
private JTextField getr13() {
    if (r13 == null) {
        r13 = new JTextField();
    }
    return r13;
}

```



```

    }
/**
 * @return
 */
private JTextField getr14() {
    if (r14 == null) {
        r14 = new JTextField();
    }
    return r14;
}
/**
 * @return
 */
private JTextField getr15() {
    if (r15 == null) {
        r15 = new JTextField();
    }
    return r15;
}
/**
 * @return
 */
private JTextField getr16() {
    if (r16 == null) {
        r16 = new JTextField();
    }
    return r16;
}
/**
 * @return
 */
private JTextField getr17() {
    if (r17 == null) {
        r17 = new JTextField();
    }
    return r17;
}
/**
 * @return
 */
private JTextField getr18() {
    if (r18 == null) {
        r18 = new JTextField();
    }
    return r18;
}
/**
 * @return
 */
private JTextField getr19() {
    if (r19 == null) {
        r19 = new JTextField();
    }
    return r19;
}
/**

```

```

    * @return
    */
private JTextField getr20() {
    if (r20 == null) {
        r20 = new JTextField();
    }
    return r20;
}
/**
 * @return
 */
private JTextField getr21() {
    if (r21 == null) {
        r21 = new JTextField();
    }
    return r21;
}
/**
 * @return
 */
private JTextField getr22() {
    if (r22 == null) {
        r22 = new JTextField();
    }
    return r22;
}
/**
 * @return
 */
private JTextField getr23() {
    if (r23 == null) {
        r23 = new JTextField();
    }
    return r23;
}
/**
 * @return
 */
private JTextField getr24() {
    if (r24 == null) {
        r24 = new JTextField();
    }
    return r24;
}
/**
 * @return
 */
private JTextField getr25() {
    if (r25 == null) {
        r25 = new JTextField();
    }
    return r25;
}
/**
 * @return

```

```

*/
private JTextField getr26() {
    if (r26 == null) {
        r26 = new JTextField();
    }
    return r26;
}
/**
 * @return
 */
private JTextField getr27() {
    if (r27 == null) {
        r27 = new JTextField();
    }
    return r27;
}
/**
 * @return
 */
private JTextField getr28() {
    if (r28 == null) {
        r28 = new JTextField();
    }
    return r28;
}
/**
 * @return
 */
private JTextField getr29() {
    if (r29 == null) {
        r29 = new JTextField();
    }
    return r29;
}
/**
 * @return
 */
private JTextField getr30() {
    if (r30 == null) {
        r30 = new JTextField();
    }
    return r30;
}
/**
 * @return
 */
private JTextField getr31() {
    if (r31 == null) {
        r31 = new JTextField();
    }
    return r31;
}
/**
 * @return
 */
private JTextField getr32() {

```

```

    if (r32 == null) {
        r32 = new JTextField();
    }
    return r32;
}
/**
 * @return
 */
private JTextField getr33() {
    if (r33 == null) {
        r33 = new JTextField();
    }
    return r33;
}
/**
 * @return
 */
private JTextField getr34() {
    if (r34 == null) {
        r34 = new JTextField();
    }
    return r34;
}
/**
 * @return
 */
private JTextField getr35() {
    if (r35 == null) {
        r35 = new JTextField();
    }
    return r35;
}
/**
 * @return
 */
private JTextField getr36() {
    if (r36 == null) {
        r36 = new JTextField();
    }
    return r36;
}
/**
 * @return
 */
private JTextField getr37() {
    if (r37 == null) {
        r37 = new JTextField();
    }
    return r37;
}
/**
 * @return
 */
private JTextField getr38() {
    if (r38 == null) {
        r38 = new JTextField();
    }
}

```

```

    }
    return r38;
}
/**
 * @return
 */
private JTextField getr39() {
    if (r39 == null) {
        r39 = new JTextField();
    }
    return r39;
}
/**
 * @return
 */
private JTextField getr40() {
    if (r40 == null) {
        r40 = new JTextField();
    }
    return r40;
}
/**
 * @return
 */
private JTextField getr41() {
    if (r41 == null) {
        r41 = new JTextField();
    }
    return r41;
}

/**
 * @return
 */
private JTextField getr42() {
    if (r42 == null) {
        r42 = new JTextField();
    }
    return r42;
}
/**
 * @return
 */
private JTextField getr43() {
    if (r43 == null) {
        r43 = new JTextField();
    }
    return r43;
}
/**
 * @return
 */
private JTextField getr44() {
    if (r44 == null) {
        r44 = new JTextField();
    }
}

```

```

        return r44;
    }
    /**
     * @return
     */
    private JTextField getr45() {
        if (r45 == null) {
            r45 = new JTextField();
        }
        return r45;
    }
    /**
     * @return
     */
    private JTextField getr46() {
        if (r46 == null) {
            r46 = new JTextField();
        }
        return r46;
    }
    /**
     * @return
     */
    private JTextField getr47() {
        if (r47 == null) {
            r47 = new JTextField();
        }
        return r47;
    }
    /**
     * @return
     */
    private JTextField getr48() {
        if (r48 == null) {
            r48 = new JTextField();
        }
        return r48;
    }
    /**
     * @return
     */
    private JTextField getr49() {
        if (r49 == null) {
            r49 = new JTextField();
        }
        return r49;
    }
    /**
     * @return
     */
    private JTextField getr50() {
        if (r50 == null) {
            r50 = new JTextField();
        }
        return r50;
    }
}

```

```
/**
 * @return
 */
private JTextField getr51() {
    if (r51 == null) {
        r51 = new JTextField();
    }
    return r51;
}
/**
 * @return
 */
private JTextField getr52() {
    if (r52 == null) {
        r52 = new JTextField();
    }
    return r52;
}
/**
 * @return
 */
private JTextField getr53() {
    if (r53 == null) {
        r53 = new JTextField();
    }
    return r53;
}
/**
 * @return
 */
private JTextField getr54() {
    if (r54 == null) {
        r54 = new JTextField();
    }
    return r54;
}
/**
 * @return
 */
private JTextField getr55() {
    if (r55 == null) {
        r55 = new JTextField();
    }
    return r55;
}
/**
 * @return
 */
private JTextField getr56() {
    if (r56 == null) {
        r56 = new JTextField();
    }
    return r56;
}
/**
 * @return
```

```
    */
private JTextField getr57() {
    if (r57 == null) {
        r57 = new JTextField();
    }
    return r57;
}
/**
 * @return
 */
private JTextField getr58() {
    if (r58 == null) {
        r58 = new JTextField();
    }
    return r58;
}
/**
 * @return
 */
private JTextField getr59() {
    if (r59 == null) {
        r59 = new JTextField();
    }
    return r59;
}
/**
 * @return
 */
private JTextField getr60() {
    if (r60 == null) {
        r60 = new JTextField();
    }
    return r60;
}
}
```


Planificacion

```

package org.anesthsom.anesthsom.vista;

//import java.applet.Applet;
//import java.awt.Frame;

import java.awt.event.ActionEvent;
import java.awt.event.KeyEvent;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.Observable;
import java.util.Observer;
import java.util.logging.Level;
import java.util.logging.Logger;

import javax.swing.JFrame;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.KeyStroke;

import org.anesthsom.anesthsom.controlador.Controlador;

import com.lowagie.text.DocumentException;

public class Planificacion extends JFrame/*extends Applet*/
implements Observer {
    /**
     *
     */
    private static final long serialVersionUID = 1L;
    private int fase1r[];
    private int fase2r[];
    private int fase3r[];
    private int fase4r[];
    private int fase5r[];
    private int fase6r[];
    private int fase7r[];
    private int fase8r[];
    private int fase9r[];
    private int fase1l[];
    private int fase2l[];

```

```

private int fase3l[];
private int fase4l[];
private int fase5l[];
private int fase6l[];
private int fase7l[];
private int fase8l[];
private int fase9l[];
private JMenuItem fase1rbuton;
private JMenuItem fase2rbuton;
private JMenuItem fase3rbuton;
private JMenuItem fase1lbuton;
private JMenuItem fase2lbuton;
private JMenuItem fase3lbuton;
JMenuItem botonAyuda;
JMenuItem botonAcerca;
private JMenuBar barra = new JMenuBar();
private JMenu archivo=new JMenu("Files");
private JMenu informes=new JMenu("Briefs");
private JMenu ayuda = new JMenu("?");
public Planificacion(int [] fase1r,int []fase2r,int
[]fase3r,int []fase4r,int []fase5r,int []fase6r,int []fase7r,int
[]fase8r,int []fase9r,int []fase1l,int []fase2l,int []fase3l,int
[]fase4l,int []fase5l,int []fase6l,int []fase7l,int []
fase8l,int []fase9l){
    super();
    setJMenuBar(barra);
    this.fase1r=fase1r;
    this.fase2r=fase2r;
    this.fase3r=fase3r;
    this.fase4r=fase4r;
    this.fase5r=fase5r;
    this.fase6r=fase6r;
    this.fase7r=fase7r;
    this.fase8r=fase8r;
    this.fase9r=fase9r;
    this.fase1l=fase1l;
    this.fase2l=fase2l;
    this.fase3l=fase3l;
    this.fase4l=fase4l;
    this.fase5l=fase5l;
    this.fase6l=fase6l;
    this.fase7l=fase7l;
    this.fase8l=fase8l;
    this.fase9l=fase9l;
    //A LA BARRA LE AGREGAMOS LOS MENUS.
    barra.add(archivo);
    barra.add(informes);
    barra.add(ayuda);
    ayuda.add(getAyudaItem());
    ayuda.addSeparator();
    ayuda.add(getAcercaItem());
    int a=1;
    int b=9;
    if (fase1l.length!=0){
        b=1;
        mostrarAyudal(a, fase1l,b);
    }
}

```

ANEXO

```
}
else if (fase2l.length!=0){
    b=2;
    mostrarAyudal(a, fase2l,b);
}
else if (fase3l.length!=0){
    b=3;
    mostrarAyudal(a, fase3l,b);
}
else if (fase4l.length!=0){
    b=4;
    mostrarAyudal(a, fase4l,b);
}
else if (fase5l.length!=0){
    b=5;
    mostrarAyudal(a, fase5l,b);
}
if (fase6l.length!=0){
    b=6;
    mostrarAyudal(a, fase6l,b);
}
else if (fase7l.length!=0){
    b=7;
    mostrarAyudal(a, fase7l,b);
}
else if (fase8l.length!=0){
    b=8;
    mostrarAyudal(a, fase8l,b);
}
else if (fase9l.length!=0){
    b=9;
    mostrarAyudal(a, fase9l,b);
}

informes.addSeparator();
informes.addSeparator();
b=0;
if (fase1r.length!=0){
    b=1;
    mostrarAyudar(a, fase1r,b);
}
else if (fase2r.length!=0){
    b=2;
    mostrarAyudar(a, fase2r,b);
}
}
```

ANEXO

```
else if (fase3r.length!=0){
    b=3;
    mostrarAyudar(a, fase3r, b);
}
else if (fase4r.length!=0){
    b=4;
    mostrarAyudar(a, fase4r, b);
}
else if (fase5r.length!=0){
    b=5;
    mostrarAyudar(a, fase5r, b);
}
if (fase6r.length!=0){
    b=6;
    mostrarAyudar(a, fase6r, b);
}
else if (fase7r.length!=0){
    b=7;
    mostrarAyudar(a, fase7r, b);
}
else if (fase8r.length!=0){
    b=8;
    mostrarAyudar(a, fase8r, b);
}
else if (fase9r.length!=0){
    b=9;
    mostrarAyudar(a, fase9r, b);
}
}
private void mostrarAyudal(int a, int[] fase1, int b) {
    if (a==1){
        informes.add(getFaselesiones(fase1));
    }
    int i=b+1;
    int j=0;
    int[] c= new int[3];
    while ((i<=9) && (j!=3)){
        i++;
        switch(i){
            case 2:
                if(fase2l.length!=0){
                    c[j]=i;
                    j++;
                }break;
            case 3:
                if(fase3l.length!=0){
                    c[j]=i;
                    j++;
                }break;
        }
    }
}
```

```

        }break;
    case 4:
        if(fase4l.length!=0){
            c[j]=i;
            j++;
        }break;
    case 5:
        if(fase5l.length!=0){
            c[j]=i;
            j++;
        }break;
    case 6:
        if(fase6l.length!=0){
            c[j]=i;
            j++;
        }break;
    case 7:
        if(fase7l.length!=0){
            c[j]=i;
            j++;
        }break;
    case 8:
        if(fase8l.length!=0){
            c[j]=i;
            j++;
        }break;
    case 9:
        if(fase9l.length!=0){
            c[j]=i;
            j++;
        }break;
    }
}
informes.add(getFase2lesiones(c));
int k=i;
j=0;
int[] f=new int[6];
while ((k<=9)&&(j!=6)){

    k++;
    switch(i){
    case 2:
        if(fase2l.length!=0){
            f[j]=k;
            j++;
        }break;
    case 3:
        if(fase3l.length!=0){
            f[j]=k;
            j++;
        }break;
    case 4:
        if(fase4l.length!=0){
            f[j]=k;
            j++;
        }break;

```

ANEXO

```

        case 5:
            if(fase5l.length!=0){
                f[j]=k;
                j++;
            }break;
        case 6:
            if(fase6l.length!=0){
                f[j]=k;
                j++;
            }break;
        case 7:
            if(fase7l.length!=0){
                f[j]=k;
                j++;
            }break;
        case 8:
            if(fase8l.length!=0){
                f[j]=k;
                j++;
            }break;
        case 9:
            if(fase9l.length!=0){
                f[j]=k;
                j++;
            }break;
    }
}
informes.add(getFase3lesiones(f));
}
private void mostrarAyudar(int a, int[] fase1, int b) {
    if (a==1){
        informes.add(getFase1riesgo(fase1));
    }
    int i=b+1;
    int j=0;
    int[] c= new int[3];
    while ((i<=9)&&(j!=3)){
        i++;
        switch(i){
            case 2:
                if(fase2r.length!=0){
                    c[j]=i;
                    j++;
                }break;
            case 3:
                if(fase3r.length!=0){
                    c[j]=i;
                    j++;
                }break;
            case 4:
                if(fase4r.length!=0){
                    c[j]=i;
                    j++;
                }break;
            case 5:
                if(fase5r.length!=0){

```

ANEXO

```

        c[j]=i;
        j++;
    }break;
case 6:
    if(fase6r.length!=0){
        c[j]=i;
        j++;
    }break;
case 7:
    if(fase7r.length!=0){
        c[j]=i;
        j++;
    }break;
case 8:
    if(fase8r.length!=0){
        c[j]=i;
        j++;
    }break;
case 9:
    if(fase9r.length!=0){
        c[j]=i;
        j++;
    }break;
    }
}
informes.add(getFase2riesgo(c));
int k=i;
j=0;
int[] f=new int[6];
while ((k<=9) && (j!=6)) {

    k++;
    switch(i) {
    case 2:
        if(fase2r.length!=0) {
            f[j]=k;
            j++;
        }break;
    case 3:
        if(fase3r.length!=0) {
            f[j]=k;
            j++;
        }break;
    case 4:
        if(fase4r.length!=0) {
            f[j]=k;
            j++;
        }break;
    case 5:
        if(fase5r.length!=0) {
            f[j]=k;
            j++;
        }break;
    case 6:
        if(fase6r.length!=0) {
            f[j]=k;

```

ANEXO

```

                j++;
            }break;
        case 7:
            if(fase7r.length!=0){
                f[j]=k;
                j++;
            }break;
        case 8:
            if(fase8r.length!=0){
                f[j]=k;
                j++;
            }break;
        case 9:
            if(fase9r.length!=0){
                f[j]=k;
                j++;
            }break;
        }
    }
    informes.add(getFase3riesgo(f));
}

private JMenuItem getAcercaItem() {
    if (botonAcerca == null) {
        botonAcerca = new JMenuItem("About...");

        botonAcerca.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.
VK_2,
                ActionEvent.ALT_MASK));
        botonAcerca.addActionListener(new
java.awt.event.ActionListener() {
            public void
actionPerformed(java.awt.event.ActionEvent e) {
                AcercaDe aplicacion = new AcercaDe();
                aplicacion.setDefaultCloseOperation(
JFrame.DISPOSE_ON_CLOSE );
            }
        });
    }
    return botonAcerca;
}

private JMenuItem getAyudaItem() {
    if (botonAyuda == null) {
        botonAyuda = new JMenuItem("Help");

        botonAyuda.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.V
K_1,
                ActionEvent.ALT_MASK));
        botonAyuda.addActionListener(new
java.awt.event.ActionListener() {
            public void
actionPerformed(java.awt.event.ActionEvent e) {
                try {

```


ANEXO

```
Runtime.getRuntime().exec("rundll32
url.dll,FileProtocolHandler "+ "Ayuda General.pdf");

} catch (IOException e1) {
// TODO Auto-generated

catch block
e1.printStackTrace();
}

}
});
}
return botonAyuda;
}

private JMenuItem getFase3lesiones(final int fase1[]) {
if (fase3lbuton == null) {
fase3lbuton = new JMenuItem();
fase3lbuton.setText("Phase 3 of injuries");

fase3lbuton.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.
VK_3,KeyEvent.VK_L));
fase3lbuton.addActionListener(new
java.awt.event.ActionListener() {
public void
actionPerformed(java.awt.event.ActionEvent e) {
int[] d;

int v=0;
int s=0;
for (int i=0;i<fase1.length;i++){
switch (fase1[i]){
case 0:break;
case 1: break;
case 2:

s=s+fase2l.length;
break;
case 3:

s=s+fase3l.length;
break;
case 4:

s=s+fase4l.length;
break;
case 5:

s=s+fase5l.length;
break;
case 6:

s=s+fase6l.length;
break;
```

ANEXO

```
        case 7:

            s=s+fase7l.length;
            break;
        case 8:

            s=s+fase8l.length;
            break;
        case 9:

            s=s+fase9l.length;
            break;
        }
    }
    d=new int[s];
    for (int i=0;i<fase1.length;i++){
        switch (fase1[i]){
            case 0:break;
            case 1: break;
            case 2:
                d=anadir(fase2l,v,d);
                v=v+fase2l.length;
                break;
            case 3:
                d=anadir(fase3l,v,d);
                v=v+fase3r.length;
                break;
            case 4:
                d=anadir(fase4l,v,d);
                v=v+fase4l.length;
                break;
            case 5:
                d=anadir(fase5l,v,d);
                v=v+fase5l.length;
                break;
            case 6:
                d=anadir(fase6l,v,d);
                v=v+fase6l.length;
                break;
            case 7:
                d=anadir(fase7l,v,d);
                v=v+fase7l.length;
                break;
            case 8:
                d=anadir(fase8l,v,d);
                v=v+fase8l.length;
                break;
            case 9:
                d=anadir(fase9l,v,d);
                v=v+fase9l.length;
                break;
        }
    }
    try {
```

ANEXO

```
Controlador.getInstance().CrearInforme(d,"Lesiones
3.pdf",3);
        } catch (FileNotFoundException e1) {
            // TODO Auto-generated catch
block
            e1.printStackTrace();
        } catch (DocumentException e1) {

Logger.getLogger(Planificacion.class.getName()).log(Level.SEVERE
, null, e1);
        }

Controlador.getInstance().VisualizarInforme("Lesiones 3.pdf");
        }
    });
}
return fase3lbuton;
}
private JMenuItem getFase2lesiones(final int fase1[]) {
    if (fase2lbuton == null) {
        fase2lbuton = new JMenuItem();
        fase2lbuton.setText("Phase 2 of injuries");

        fase2lbuton.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.
VK_2,KeyEvent.VK_L));
        fase2lbuton.addActionListener(new
java.awt.event.ActionListener() {
            public void
actionPerformed(java.awt.event.ActionEvent e) {

                int[] d;

                int v=0;
                int s=0;
                for (int i=0;i<fase1.length;i++){
                    switch (fase1[i]){
                        case 0:break;
                        case 1: break;
                        case 2:

                            s=s+fase2l.length;
                            break;
                        case 3:

                            s=s+fase3l.length;
                            break;
                        case 4:

                            s=s+fase4l.length;
                            break;
                        case 5:

                            s=s+fase5l.length;
                            break;
                        case 6:
```

ANEXO

```
        s=s+fase6l.length;
        break;
    case 7:

        s=s+fase7l.length;
        break;
    case 8:

        s=s+fase8l.length;
        break;
    case 9:

        s=s+fase9l.length;
        break;
    }
}

d=new int[s];
for (int i=0;i<fase1.length;i++){
switch (fase1[i]){
case 0:break;
case 1: break;
case 2:
d=anadir(fase2l,v,d);
v=v+fase2l.length;
break;
case 3:
d=anadir(fase3l,v,d);
v=v+fase3l.length;
break;
case 4:
d=anadir(fase4l,v,d);
v=v+fase4l.length;
break;
case 5:
d=anadir(fase5l,v,d);
v=v+fase5l.length;
break;
case 6:
d=anadir(fase6l,v,d);
v=v+fase6l.length;
break;
case 7:
d=anadir(fase7l,v,d);
v=v+fase7l.length;
break;
case 8:
d=anadir(fase8l,v,d);
v=v+fase8l.length;
break;
case 9:
d=anadir(fase9l,v,d);
v=v+fase9l.length;
break;
}
}
```

```

        try {

            Controlador.getInstance().CrearInformel(d,"Lesiones
2.pdf",2);
                                } catch (FileNotFoundException e1) {
                                // TODO Auto-generated catch
block
                                e1.printStackTrace();
                                } catch (DocumentException e1) {
                                // TODO Auto-generated catch
block
                                e1.printStackTrace();
                                }

Controlador.getInstance().VisualizarInforme("Lesiones 2.pdf");
        }
    });
    }
    return fase2lbuton;
}
private JMenuItem getFasellesiones(final int fase1[]) {
    if (fase1lbuton == null) {
        fase1lbuton = new JMenuItem();
        fase1lbuton.setText("Phase 1 of injuries");

        fase1lbuton.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.
VK_1,KeyEvent.VK_L));
        fase1lbuton.addActionListener(new
java.awt.event.ActionListener() {
            public void
actionPerformed(java.awt.event.ActionEvent e) {

                try {

                    Controlador.getInstance().CrearInformel(fase1,"Lesion
1.pdf",1);
                                } catch (FileNotFoundException e1) {
                                // TODO Auto-generated catch
block
                                e1.printStackTrace();
                                } catch (DocumentException e1) {
                                // TODO Auto-generated catch
block
                                e1.printStackTrace();
                                }

Controlador.getInstance().VisualizarInforme("Lesion 1.pdf");
        }
    });
    }
    return fase1lbuton;
}

private JMenuItem getFase3riesgo(final int[] fase3) {
    if (fase3rbuton == null) {
        fase3rbuton = new JMenuItem();

```

ANEXO

```
fase3rbuton.setText("Phase 3 of risk");

fase3rbuton.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_3,KeyEvent.VK_R));
fase3rbuton.addActionListener(new
java.awt.event.ActionListener() {
    public void
actionPerformed(java.awt.event.ActionEvent e) {

        int[] d;

        int v=0;
        int s=0;
        for (int i=0;i<fase3.length;i++){
            switch (fase3[i]){
                case 0:break;
                case 1: break;
                case 2:

                    s=s+fase2r.length;
                    break;
                case 3:

                    s=s+fase3r.length;
                    break;
                case 4:

                    s=s+fase4r.length;
                    break;
                case 5:

                    s=s+fase5r.length;
                    break;
                case 6:

                    s=s+fase6r.length;
                    break;
                case 7:

                    s=s+fase7r.length;
                    break;
                case 8:

                    s=s+fase8r.length;
                    break;
                case 9:

                    s=s+fase9r.length;
                    break;
            }
        }
        d=new int[s];
        for (int i=0;i<fase3.length;i++){
            switch (fase3[i]){
                case 0:break;
                case 1: break;
```

ANEXO

```
        case 2:
            d=anadir(fase2r,v,d);
            v=v+fase2r.length;
            break;
        case 3:
            d=anadir(fase3r,v,d);
            v=v+fase3r.length;
            break;
        case 4:
            d=anadir(fase4r,v,d);
            v=v+fase4r.length;
            break;
        case 5:
            d=anadir(fase5r,v,d);
            v=v+fase5r.length;
            break;
        case 6:
            d=anadir(fase6r,v,d);
            v=v+fase6r.length;
            break;
        case 7:
            d=anadir(fase7r,v,d);
            v=v+fase7r.length;
            break;
        case 8:
            d=anadir(fase8r,v,d);
            v=v+fase8r.length;
            break;
        case 9:
            d=anadir(fase9r,v,d);
            v=v+fase9r.length;
            break;
    }
}
try {
    Controlador.getInstance().CrearInformes(d,"Riesgo
3.pdf",3);
    } catch (FileNotFoundException
e1) {
        // TODO Auto-generated
        catch block
            e1.printStackTrace();
    } catch (DocumentException e1) {
        // TODO Auto-generated
        catch block
            e1.printStackTrace();
    }
}
Controlador.getInstance().VisualizarInforme("Riesgo 3.pdf");
    }
    });
}
return fase3rbuton;
}
private JMenuItem getFase2riesgo(final int[] fase2) {
```

ANEXO

```
        if (fase2rbuton == null) {
            fase2rbuton = new JMenuItem();
            fase2rbuton.setText("Phase 2 of risk");

            fase2rbuton.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.
VK_2,KeyEvent.VK_R));
            fase2rbuton.addActionListener(new
java.awt.event.ActionListener() {
                public void
actionPerformed(java.awt.event.ActionEvent e) {
                    int[] d;

                    int v=0;
                    int s=0;
                    for (int i=0;i<fase2.length;i++){
                        switch (fase2[i]){
                            case 0:break;
                            case 1: break;
                            case 2:

                                s=s+fase2r.length;
                                break;
                                case 3:

                                    s=s+fase3r.length;
                                    break;
                                    case 4:

                                        s=s+fase4r.length;
                                        break;
                                        case 5:

                                            s=s+fase5r.length;
                                            break;
                                            case 6:

                                                s=s+fase6r.length;
                                                break;
                                                case 7:

                                                    s=s+fase7r.length;
                                                    break;
                                                    case 8:

                                                        s=s+fase8r.length;
                                                        break;
                                                        case 9:

                                                            s=s+fase9r.length;
                                                            break;
                                                            }
                                                            }
                    d=new int[s];
                    for (int i=0;i<fase2.length;i++){
                        switch (fase2[i]){
                            case 0:break;
```


ANEXO

```
        case 1: break;
        case 2:
            d=anadir(fase2r,v,d);
            v=v+fase2r.length;
            break;
        case 3:
            d=anadir(fase3r,v,d);
            v=v+fase3r.length;
            break;
        case 4:
            d=anadir(fase4r,v,d);
            v=v+fase4r.length;
            break;
        case 5:
            d=anadir(fase5r,v,d);
            v=v+fase5r.length;
            break;
        case 6:
            d=anadir(fase6r,v,d);
            v=v+fase6r.length;
            break;
        case 7:
            d=anadir(fase7r,v,d);
            v=v+fase7r.length;
            break;
        case 8:
            d=anadir(fase8r,v,d);
            v=v+fase8r.length;
            break;
        case 9:
            d=anadir(fase9r,v,d);
            v=v+fase9r.length;
            break;
    }
}
try {

    Controlador.getInstance().CrearInformes(d,"Riesgo
2.pdf",2);
                                } catch (FileNotFoundException e1) {
                                // TODO Auto-generated catch
block
                                e1.printStackTrace();
                                } catch (DocumentException e1) {
                                // TODO Auto-generated catch
block
                                e1.printStackTrace();
                                }

    Controlador.getInstance().VisualizarInforme("Riesgo 2.pdf");
    }

});
}
return fase2rbuton;
```

```

}
private int[] anadir(int[] fase6, int v, int[] d) {

    for(int i=0;i<fase6.length;i++){
        d[v+i]=fase6[i];
    }
    return d;
}
private JMenuItem getFaselriesgo(final int[] fase2) {
    if (faselrbuton == null) {
        faselrbuton = new JMenuItem();
        faselrbuton.setText("Phase 1 of risk");

        faselrbuton.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_2,KeyEvent.VK_R));
        faselrbuton.addActionListener(new
java.awt.event.ActionListener(){
    public void
actionPerformed(java.awt.event.ActionEvent e){

        try {

            Controlador.getInstance().CrearInformes(fase2,"Riesgo
1.pdf",1);
                                } catch (FileNotFoundException e1) {
block
                                    // TODO Auto-generated catch
                                    e1.printStackTrace();
                                } catch (DocumentException e1) {
block
                                    // TODO Auto-generated catch
                                    e1.printStackTrace();
                                }
                                }

Controlador.getInstance().VisualizarInforme("Riesgo 1.pdf");

        }
    });
}
return faselrbuton;
}

public int[] getFaselr() {
    return faselr;
}
public void setFaselr(int[] faselr) {
    this.faselr = faselr;
}
public int[] getFase2r() {
    return fase2r;
}
public void setFase2r(int[] fase2r) {
    this.fase2r = fase2r;
}
public int[] getFase3r() {
    return fase3r;
}

```

```

}
public void setFase3r(int[] fase3r) {
    this.fase3r = fase3r;
}
public int[] getFase4r() {
    return fase4r;
}
public void setFase4r(int[] fase4r) {
    this.fase4r = fase4r;
}
public int[] getFase5r() {
    return fase5r;
}
public void setFase5r(int[] fase5r) {
    this.fase5r = fase5r;
}
public int[] getFase6r() {
    return fase6r;
}
public void setFase6r(int[] fase6r) {
    this.fase6r = fase6r;
}
public int[] getFase7r() {
    return fase7r;
}
public void setFase7r(int[] fase7r) {
    this.fase7r = fase7r;
}
public int[] getFase8r() {
    return fase8r;
}
public void setFase8r(int[] fase8r) {
    this.fase8r = fase8r;
}
public int[] getFase1l() {
    return fase1l;
}
public void setFase1l(int[] fase1l) {
    this.fase1l = fase1l;
}
public int[] getFase2l() {
    return fase2l;
}
public void setFase2l(int[] fase2l) {
    this.fase2l = fase2l;
}
public int[] getFase3l() {
    return fase3l;
}
public void setFase3l(int[] fase3l) {
    this.fase3l = fase3l;
}
public int[] getFase4l() {
    return fase4l;
}
public void setFase4l(int[] fase4l) {

```

```
        this.fase4l = fase4l;
    }
    public int[] getFase5l() {
        return fase5l;
    }
    public void setFase5l(int[] fase5l) {
        this.fase5l = fase5l;
    }
    public int[] getFase6l() {
        return fase6l;
    }
    public void setFase6l(int[] fase6l) {
        this.fase6l = fase6l;
    }
    public int[] getFase7l() {
        return fase7l;
    }
    public void setFase7l(int[] fase7l) {
        this.fase7l = fase7l;
    }
    public int[] getFase8l() {
        return fase8l;
    }
    public void setFase8l(int[] fase8l) {
        this.fase8l = fase8l;
    }
    public int[] getFase9r() {
        return fase9r;
    }
    public void setFase9r(int[] fase9r) {
        this.fase9r = fase9r;
    }
    public int[] getFase9l() {
        return fase9l;
    }
    public void setFase9l(int[] fase9l) {
        this.fase9l = fase9l;
    }
    public void update(Observable o, Object arg) {
        // TODO Auto-generated method stub
    }
}
```

OrgBIOINomegaANESTHSOM

```
package org.anesthsom.anesthsom;

import org.anesthsom.anesthsom.vista.Central_1;

public class OrgBIOINomegaANESTHSOM {
    public static void main(String[] args){
        Central_1 tade=new Central_1();
        tade.setVisible(true);
    }
}
```