

**ANÁLISIS Y DISEÑO DE APLICACIÓN PARA LA GESTIÓN ECONÓMICA Y DE
SERVICIOS DE LA EMPRESA VITAL PLUS**



DANIEL DARÍO PERTUZ OROZCO

**FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS
BARRANQUILLA**

2018

CONTENIDO

Tablas	6
Imágenes.....	7
Anexos.....	8
Introducción	9
1. Planteamiento del problema.....	11
2. Formulación del problema	12
3. Objetivos	12
3.1. Objetivo general	12
3.2. Objetivos específicos.....	12
4. Justificación	13
5. Marco De Referencia	15
5.1. Marco Teórico	15
5.1.1. La Programación Orientada A Objetos	15
5.1.2. Adaptabilidad.....	15
5.1.3. Reusabilidad.	15
5.1.4. Mantenibilidad.....	15
5.1.5. Abstracción.....	16
5.1.6. Encapsulamiento.....	16
5.1.7. Modularidad.....	17
5.1.8. Principio de ocultación.	17
5.1.9. Herencia.....	17
5.1.10. Atributos (o variables miembros, variables de clase).....	17
5.1.11. Operaciones (o métodos, funciones miembros).	17
5.1.12. Identidad.....	18
5.1.13. Comportamiento.....	18
5.1.14. Estado.....	18
5.1.15. Ciclo De Vida Del Software	19
5.1.16. Metodología SCRUM.....	20
5.1.16.7. <i>Backlog</i>	21
5.1.17. Ciclo de vida del proceso unificado.	21

5.1.18. Modelo en V.	21
5.1.19. Ingeniería inversa.	22
5.1.20. Reingeniería.	22
5.1.21. Base De Datos	23
5.1.22. Características De Las Bases De Datos	23
5.1.23. Arquitectura De Las Bases De Datos	24
5.1.24. Elementos De Las Bases De Datos.....	25
5.1.25. MySQL	27
5.1.26. Características De MySQL.....	27
5.1.27. Arquitectura De MySQL	27
5.1.28. Arquitectura Cliente/Servidor.....	28
5.1.29. Entornos De Desarrollo Integrados IDE.....	28
5.1.30. Características De NetBeans	29
5.1.31. Lenguaje De Programación – Java	30
5.1.32. Patrones De Arquitectura Por Capas	31
5.1.33. Persistencia	32
5.1.34. ORM	32
5.1.35. Hibernate	33
5.1.36. Frameworks	34
5.1.37. JasperReports.....	35
5.2. Marco Conceptual	35
5.2.1. Programación Orientada A Objetos.....	36
5.2.2. Implementación De Lenguaje De Programación Java	36
5.2.3. Uso De Gestor De Base De Datos	36
5.2.4. Diccionario De Datos	37
5.2.5. Método De Pago	37
5.2.6. Tipo De Abono	38
5.2.7. Tipo De Tratamiento	38
5.2.8. Tipo De Usuario	38
5.2.9. Datos Personales De Los Pacientes.....	39
5.3. Marco Contextual.....	39

5.3.1. Reseña Histórica De <i>Vital Plus</i>	39
5.3.2. Misión.....	39
5.3.3. Visión.....	40
5.3.4. Organigrama	40
5.4. Antecedentes	40
5.4.1. Antecedentes nacionales.....	40
5.4.2. Antecedentes internacionales	42
6. Marco Metodológico.....	44
6.1. Metodología De La Investigación	44
6.1.1. Paradigma de investigación	44
6.1.2. Tipo de investigación.....	44
6.1.3. Metodología De Desarrollo	44
6.1.4. Fuentes Y Técnicas De Recolección De Información.....	45
6.1.5. Universo O Población.....	45
6.1.6. Muestra	45
6.1.7. Delimitación Y Alcance	45
6.1.8. Técnicas de análisis de datos	46
6.1.9. Técnicas de procesamiento de datos.....	46
6.1.10. Esquema Temático	46
7. Cronograma De Actividades.....	47
8. Recursos Necesarios	47
9. Resultados Esperados.....	47
10. Capítulo I	47
10.1. Análisis.....	47
10.1.1. Programación orientada a objetos.....	49
10.1.2. Implementación de lenguaje de programación Java.....	49
10.1.3. Uso de gestor de base de datos.	50
10.1.4. Rendimiento.....	50
10.1.5. Confiabilidad.	50
10.1.6. Disponibilidad.	51
10.1.7. Seguridad.	51

10.1.8. Mantenibilidad.....	51
10.1.9. Entorno De Administrador.....	52
10.1.10. Entorno De Agente.....	53
10.2. Procesos.....	55
10.3. Diagrama De Procesos.....	57
10.4. Actores.....	57
10.4.1. Administradores Y Agentes.....	57
10.4.2. Pacientes.....	58
10.4.3. Profesionales.....	58
10.4.4. Laboratorios.....	58
10.4.5. Tratamientos.....	58
10.5. Requerimientos Funcionales.....	58
10.5.1. Comportamiento.....	59
10.5.2. Restricciones.....	59
10.5.3. Calidad esperada.....	59
10.6. Diagrama General.....	60
10.7. Lista De Requerimientos Funcionales (Específicos).....	60
10.7.1. Datos.....	60
10.7.2. Operaciones y flujos.....	61
10.7.3. Reportes.....	62
10.7.4. Regulaciones.....	62
10.8. Diagramas De Secuencia.....	63
10.8.1. Rol Agente.....	63
10.8.2. Rol Administrador.....	64
10.9. Diagrama De Actividades.....	65
11. Capítulo II.....	66
11.1. Diseño De Base De Datos.....	66
11.2. Diccionario De Datos.....	67
11.2.1. Paciente.....	67
11.2.2. Profesional.....	67
11.2.3. Laboratorio.....	68

11.2.4. Procedimiento (tratamiento)	68
11.2.5. Usuarios	68
11.2.6. Abonos.....	69
11.2.7. Factura	69
11.2.8. Ingreso	70
11.2.9. Salida	70
11.2.10. Estado	70
11.3. Diagramas De Diseño.....	71
11.4. Diagrama De Clases	72
11.5. Diseño De Interfaz Gráfica	72
12. Capítulo III.....	73
12.1. Prototipo.....	73
12.2. Codificación Y Pruebas.....	76
12.3. Estándares De Codificación	78
12.4. Diagrama De Despliegue	80
12.5. Diagrama De Componentes	81
13. Análisis De Resultados	82
13.1. Identificación De Requerimientos.....	82
13.2. Definición De Paradigma De Programación.....	83
13.3. Diseño De Medio De Almacenamiento	83
13.4. Desarrollo De Prototipo	83
14. Recomendaciones	84
15. Conclusiones	85
16. Bibliografía	87

Tablas

Tabla 1. Cronograma de actividades.....	47
Tabla 2. Recursos necesarios	47
Tabla 3. Resultados esperados	47

Imágenes

Imagen 1. Modelo en V	22
Imagen 2. Los tres niveles de arquitectura.....	25
Imagen 3. Persistencia y ORM	33
Imagen 4. Organigrama de empresa Vital Plus	40
Imagen 5. Diagrama de procesos	57
Imagen 6. Diagrama General	60
Imagen 7. Diagrama de secuencia - Agente.....	63
Imagen 8. Diagrama de secuencia - Administrador.....	64
Imagen 9. Diagrama de actividades	65
Imagen 10. Diseño de base de datos	66
Imagen 11. Diagrama Entidad-Relación.....	71
Imagen 12. Diagrama de clases	72
Imagen 13. Diagrama físico.....	73
Imagen 14. Prototipo de pantalla para inicio de sesión	73
Imagen 15. Prototipo de pantalla principal de espacio para agente	73
Imagen 16. Prototipo de ventana de abonos	74
Imagen 17. Prototipo de pantalla principal de espacio para administrador	74
Imagen 18. Prototipo de ventana para registro de usuarios	75
Imagen 19. Prototipo de ventana para registro de tratamientos.....	75
Imagen 20. Prototipo de ventana para facturación.....	76
Imagen 21. Diagrama de despliegue.....	80
Imagen 22. Diagrama de componentes.....	81

Anexos

Anexo A	89
Anexo B	100
Anexo C	119
Anexo D	173

Introducción

En esta tesis de grado se trata el análisis y el diseño de una aplicación que se desea construir en la empresa *Vital Plus* para el seguimiento de la gestión y administración económica de las labores cotidianas de esta organización especializada en tratamientos y cuidados de la salud oral.

El producto final de esta tesis es una aplicación de los conocimientos adquiridos durante el proceso académico en ingeniería de software. A lo largo de esta tesis se confirma la premisa de que una aplicación de cualquier índole requiere de un análisis previo para cumplir con las expectativas de un cliente y no sólo desarrollar códigos de programación sin fundamentos.

La importancia de la aplicación radica en poder mostrar un entorno sistemático que logre suplir la necesidad que tiene la empresa *Vital Plus* en optimizar el proceso de facturación de los procedimientos odontológicos que presta, mediante la organización y ejecución de acciones sobre la información relacionada a sus pacientes suscritos que garanticen una mejor prestación en servicios de salud oral.

Se han observado algunas aplicaciones desarrolladas para el sector de la salud dental, como lo son el caso de *ODONTOWEB* y *GESCLIDENT*, las cuales fueron herramientas de mucha utilidad para realizar la gestión económica de una clínica dental y que sirvieron de guía para el desarrollo de este proyecto.

Para el desarrollo de esta tesis de grado se fija como meta principal del proyecto realizar un análisis exhaustivo de la actividad principal de la empresa, los procesos que realiza y los reglamentos a los que se amolda para prestar sus servicios, para luego diseñar un sistema que permita llevar un control de los procesos que se realizan durante los tratamientos odontológicos

que solicitan los pacientes para mejorar su salud oral y con esto cumplir con las expectativas de satisfacción tanto en la empresa para tener una mejor organización, como en los pacientes para recibir una mejor atención y asesoría.

El alcance de este proyecto comprende el control de la información relacionada con los pacientes suscritos y los tratamientos odontológicos que se realizan en la sede de la empresa *Vital Plus* ubicada al norte de la ciudad de Barranquilla, Atlántico (Calle 96A No. 49C – 50). Los pacientes suscritos son personas naturales que pueden acceder directamente a las instalaciones de la empresa o también pueden ser remitidas de otras entidades prestadoras de salud que solicitan los servicios de la empresa para tratar a sus pacientes de una manera más específica y profunda.

Para la correcta planificación y gestión de esta tesis, se realizaron reuniones presenciales con los empleados de la empresa *Vital Plus* en las instalaciones de la sede mencionada anteriormente. A partir de estas reuniones, se establecieron varios aspectos esenciales para la construcción de la aplicación, como las directrices que debe seguir la aplicación que se quiere desarrollar para cumplir con las exigencias estipuladas desde el departamento administrativo de la empresa y así lograr la satisfacción de sus usuarios suscritos al momento de requerir servicios de facturación, los roles de las personas que tendrán incidencia en el desarrollo de la aplicación (programadores, probadores de software, ingenieros de redes y de bases de datos, etc.), el estudio de los recursos de hardware y software que se hallan disponibles en la empresa para que el montaje de la aplicación en la empresa fuera lo más práctico posible y la presentación de un prototipo de la aplicación en funcionamiento teniendo todos los requerimientos cumplidos y los recursos disponibles para ejecución de la misma.

1. Planteamiento del problema

Desde sus inicios, la empresa *Vital Plus* se ha desempeñado de manera desacorde a los avances que se presentan en el manejo de información en la actualidad. Se hace uso de papel y lápiz, en lugar del manejo de la información de forma digital y computarizada. Esto hace engorrosa y complicada la atención de la empresa a sus pacientes, dando lugar a la imprecisión y pérdida de datos importantes.

Detalles como los son el número de documento de identidad, la edad, el profesional a cargo, tratamiento que reciben y los abonos que consignan los pacientes son información sensible que la empresa no ha podido administrar de manera correcta y hacen que se presenten inconsistencias para la empresa e inconvenientes para los pacientes al momento de realizar facturaciones de cualquier proceso odontológico, generando molestias e inconformidades en la prestación del servicio de salud oral.

No se poseen herramientas que permitan generar archivos ofimáticos de forma automática con la información que los pacientes suscritos a la empresa necesitan para tener más conocimiento de los movimientos económicos que han sucedido durante los procedimientos odontológicos solicitados a la empresa.

Es imperiosa la necesidad de que la aplicación se pueda desarrollar bajo un entorno que sea lo menos propenso a caídas o retrasos, es decir que sea robusta en operatividad, con el fin de que la asistencia de la empresa sea diligente y oportuna ante sus pacientes.

2. Formulación del problema

¿Cómo analizar y diseñar una aplicación de escritorio que permita llevar un control de los tratamientos odontológicos e historias clínicas de los pacientes de la empresa *Vital Plus* para prestar un óptimo servicio de salud oral?

3. Objetivos

3.1. Objetivo general

Analizar y diseñar una aplicación de escritorio que permita llevar un control de los tratamientos odontológicos y las historias clínicas de los pacientes de la empresa *Vital Plus*, con el fin de optimizar su prestación de servicios en salud oral.

3.2. Objetivos específicos

- Identificar los requerimientos de desarrollo de la aplicación para satisfacer las necesidades sistemáticas de la empresa.
- Definir un paradigma de programación que facilite la construcción de la aplicación que realizará el control de las operaciones de la empresa.
- Diseñar un medio de almacenamiento que permita gestionar de manera óptima las transacciones derivadas de los servicios y operaciones de *Vital Plus*.
- Desarrollar un prototipo de la aplicación de acuerdo con las especificaciones de análisis y diseño, que contemple los requerimientos para dar solución a la problemática de *Vital Plus*.

4. Justificación

La empresa *Vital Plus* posee un personal de sistemas que realiza operaciones de mantenimiento e innovación en la empresa para contribuir en la sistematización de los servicios de salud oral que ofrece a sus pacientes.

Como parte de dicha contribución, este proyecto de investigación realiza un análisis exhaustivo de todas las acciones que se efectúan internamente en la empresa para prestar el mejor servicio de salud oral posible a sus pacientes suscritos.

Se discute durante las reuniones de análisis y diseño de la aplicación con el personal del departamento de sistemas acerca de la escogencia del tipo de aplicación ideal para la actividad de la empresa. Existen las aplicaciones web y las de escritorio. La primera de ellas requiere de conexiones externas para su funcionamiento y es muy dependiente de la capacidad de dichas conexiones para mantener la comunicación de la aplicación con el servidor. Además, se debe vigilar que la aplicación sea compatible con los diferentes navegadores web que existen en el mercado. La segunda opción consiste en aplicaciones más robustas que pueden brindar soluciones mucho más inmediatas y pueden llegar a ser compatibles con cualquier tipo de sistema operativo, dependiendo del lenguaje de programación que se utilice.

Se decide por realizar el diseño de una aplicación de escritorio conectada a un servidor de base de datos desarrollada en un lenguaje multiplataforma que le permita funcionar en cualquier sistema operativo, debido a que ofrece un manejo de información más asequible, un tiempo de respuesta muy rápido y su ejecución no requiere de una comunicación con el exterior y así se evita que la información llegue a ser manipulada por algún ente ajeno a la empresa.

Luego de definir el tipo de aplicación que se va a desarrollar, se procede a determinar el tipo de arquitectura o paradigma de programación que se utilizará para que la aplicación pueda tener un buen mantenimiento y robustez, por lo que se escoge la metodología de programación por capas. Esta arquitectura presenta muchas ventajas para el programador en su interés por desarrollar una aplicación que pueda separar la interfaz gráfica de la lógica del negocio y los datos y así obtener una mayor reutilización del código, puesto que lo que antes se hacía revisando todo el código sólo se realizará en la capa en la que sea necesario su ajuste.

Se busca que dicha aplicación cumpla a cabalidad con los requerimientos de la empresa en seguridad, confiabilidad, comodidad y estabilidad, para que sus operaciones logren la satisfacción de sus pacientes durante el tiempo que estén suscritos a la empresa y así garantizar un mejor posicionamiento de la entidad ante la sociedad en la que se desenvuelve.

A través de este proyecto aplicado se espera que su diseño le permita a la empresa *Vital Plus* una optimización en los servicios de registro de los movimientos contables curriculares u ordinarios que realiza con sus pacientes suscritos, así como también de cualquier evento o novedad que presenten, ayudando a su crecimiento industrial en la prestación de servicios de salud oral.

También se deja por sentado que este proyecto aplicado puede servir como guía para futuras investigaciones que tengan relación con la administración de operaciones que realizan las empresas pertenecientes a la industria de servicios de salud oral.

5. Marco De Referencia

5.1. Marco Teórico

5.1.1. La Programación Orientada A Objetos

Para la correcta implementación de una aplicación especializada en la satisfacción de la necesidad que tiene una empresa para brindar las mejores soluciones en su área comercial, se determina el empleo del modelo de programación orientada a objetos (POO) como alternativa ideal para ello.

Durán, Gutiérrez y Pimentel (2007) permiten inferir que la POO tiene características ideales para explicar la complejidad de un sistema, como lo son:

5.1.2. Adaptabilidad.

La facilidad que ofrece para transportar de un sistema a otro. Esto significa que una aplicación se puede adaptar y transportar a cualquier sistema.

5.1.3. Reusabilidad.

Se reducen costos y reutilizan componentes de software cuya confiabilidad es previamente comprobada.

5.1.4. Mantenibilidad.

Las aplicaciones permanecen funcionando activamente a pesar del cambio de personas que la construyen, siempre y cuando su codificación sea fácil de comprender y mantener.

Para conseguir estos objetivos, es de vital importancia la aplicación de criterios de diseño claros y definidos que permitan comprender la complejidad de las aplicaciones. Por medio de la

POO se logra identificar cuáles son los objetos integrantes del sistema, las clases en que se pueden agrupar y las relaciones existentes entre los objetos.

El objeto es el elemento básico de trabajo en la programación orientada a objetos. Un objeto representa los objetos reales o aun no siendo tan real se puede tener una idea de que es un elemento existente con ciertas propiedades y que están regidos por entidades generales que afectan su comportamiento.

Dichas entidades generales son llamadas clases. Son las plantillas que definen un conjunto de objetos. La creación de un objeto a partir de una clase se denomina instanciación. Es estricta la ley de la POO de que a partir de la creación de una clase es que se puede crear una instancia (objeto) de la clase.

Además de esto, también se deduce que la programación orientada a objetos se basa también en los siguientes conceptos para su andamiaje:

5.1.5. Abstracción.

Es la propiedad que establece las características esenciales de un objeto y en donde se capturan sus comportamientos. La abstracción facilita el proceso de análisis y diseño de objetos, debido a que permite modelar la realidad o el problema a solucionar mediante un conjunto de clases correctamente construido.

5.1.6. Encapsulamiento.

Es el proceso de reunir los elementos que pueden pertenecer a una misma entidad. Se caracteriza por localizar las variables de un objeto en el núcleo del objeto y esconder el núcleo del objeto ante otros objetos en el programa a través de los métodos.

5.1.7. Modularidad.

Es la subdivisión de una aplicación en partes más pequeñas (módulos), cada una trabajando y siendo compilada de manera independiente pero que tienen conexiones con otros módulos.

5.1.8. Principio de ocultación.

Cada objeto está aislado del exterior, es un módulo natural, y cada tipo de objeto expone una "interfaz" a otros objetos que especifica cómo pueden interactuar con los objetos de la clase. El aislamiento protege a las propiedades de un objeto contra su modificación por quien no tenga derecho a acceder a ellas; solamente los propios métodos internos del objeto pueden acceder a su estado.

5.1.9. Herencia.

Es el conjunto de propiedades que comparten dos clases entre sí, aunque sean distintas y cumplen un proceso de reutilización de operaciones que se necesitan para la solución de uno o más problemas.

Las clases poseen dos propiedades fundamentales que direccionan a los objetos:

5.1.10. Atributos (o variables miembros, variables de clase).

Describen el estado interno de cada objeto.

5.1.11. Operaciones (o métodos, funciones miembros).

Describen lo que se puede hacer con el objeto, los servicios que proporciona.

Los objetos son la parte ejecutable de la programación orientada a objetos y se manejan a través de variables que pertenecen a la misma clase. Poseen características fundamentales que

permiten conocerlos mediante la observación, identificación y el estudio posterior de su comportamiento; estas características son:

5.1.12. Identidad.

Es la propiedad que permite distinguir a un objeto de otro. En Java, la creación de un objeto se realiza mediante la declaración de una referencia de la clase, usando el operador *new* invocando al constructor correspondiente a su clase y conectando el objeto con la referencia.

5.1.13. Comportamiento.

Es la descripción de su funcionalidad y determina las operaciones que este puede realizar o a las que puede responder ante mensajes enviados por otros objetos.

5.1.14. Estado.

Se refiere a los atributos y valores que presenta el objeto en un instante de tiempo dado. El estado de un objeto es afectado directamente por su comportamiento.

Los métodos (comportamiento) y atributos (estado) son propiedades primitivas de los objetos. Estas propiedades implican que una clase requiere de métodos para poder tratar los atributos con los que cuenta.

Los atributos son las características que tiene la clase y los métodos son algoritmos asociados a los objetos que se ejecutan tras la recepción de un mensaje y pueden producir cambios en las propiedades del objeto o la generación de un evento que produzca un nuevo mensaje para otro objeto del sistema.

5.1.15. Ciclo De Vida Del Software

La Universidad de Sevilla en su documento de investigación (2013) define el ciclo de vida del software como el marco de referencia que contiene los procesos, las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un software, abarcando la vida del sistema desde la definición hasta la finalización de su uso (ISO 12207).

Se determina el enfoque de desarrollo que se emplea en el software, indicando los procesos, actividades y tareas que se van a realizar y en qué orden, y los productos que se van a generar, también los que se van a entregar al cliente y en qué orden se van a entregar.

Se destacan varios tipos de ciclo de vida de un software:

5.1.15.1. Ciclo de vida clásico (en cascada).

En esta arquitectura se asume que se conocen todos los requisitos de desarrollo por parte del cliente. Cada fase comienza cuando termina la anterior, aunque se tarda mucho en disponer del software. Es el más fácil de planificar.

5.1.15.2. Ciclo de vida evolutivo.

Este ciclo de vida de software se caracteriza por trabajar utilizando el conjunto de procesos *requisitos-desarrollo-evaluación*, en donde el resultado de la evaluación permite evolucionar hacia la siguiente versión. Obtener todos los requisitos al comienzo es prácticamente imposible, las necesidades de clientes y usuarios evolucionan durante el desarrollo.

5.1.15.3. Ciclo de vida incremental.

Se repiten varios ciclos de vida en cascada para aplicarlos en desarrollos de gran tamaño. Al final de cada ciclo se entrega una versión parcial del software, incrementada con cierta

funcionalidad nueva respecto a las anteriores. Los usuarios disponen antes del software, aunque no sea completo y pueden sugerir mejoras.

5.1.15.4. *Ciclo de vida en los métodos ágiles.*

Son ciclos de vida evolutivos con iteraciones de corta duración (de 2 semanas a 2 meses) para favorecer la comunicación con clientes y usuarios, en donde cada iteración incorpora nuevas peticiones de clientes y usuarios.

Para la correcta construcción de un proyecto de software es importante establecer un buen equipo de trabajo que se adapte al entorno para evitar la menor cantidad de errores posible. Luego de definir el equipo de trabajo, se procede a mantener una interacción constante con el cliente, de tal manera que la colaboración entre ambos será la que marque la marcha del éxito y asegure su éxito.

5.1.16. Metodología SCRUM.

Es la metodología ágil más usada actualmente. Se basa en iteraciones de 30 días:

5.1.16.5. Sprints. Producen código potencialmente entregable. No se admiten cambios ni de requisitos ni de miembros del equipo de desarrollo.

5.1.16.6. Agile meeting. Reuniones cortas y frecuentes donde cada miembro del equipo expone lo que ha realizado desde la última reunión, los problemas que ha tenido y propone lo que se va a desarrollar en la próxima reunión.

5.1.16.7. Backlog. Lista priorizada de tareas. Se organizan las actividades que se realizarán estructuralmente en cada iteración de construcción del producto.

5.1.17. Ciclo de vida del proceso unificado.

Proceso iterativo e incremental propuesto por los creadores de UML. Se caracteriza por tener 6 fases: inicio, elaboración, construcción, transición, producción y retirada. En cada fase del desarrollo se producen una o más iteraciones y se obtiene una versión evaluable del software.

5.1.18. Modelo en V.

Es una arquitectura que asocia un tipo de pruebas a cada producto de cada fase según su nivel de abstracción.

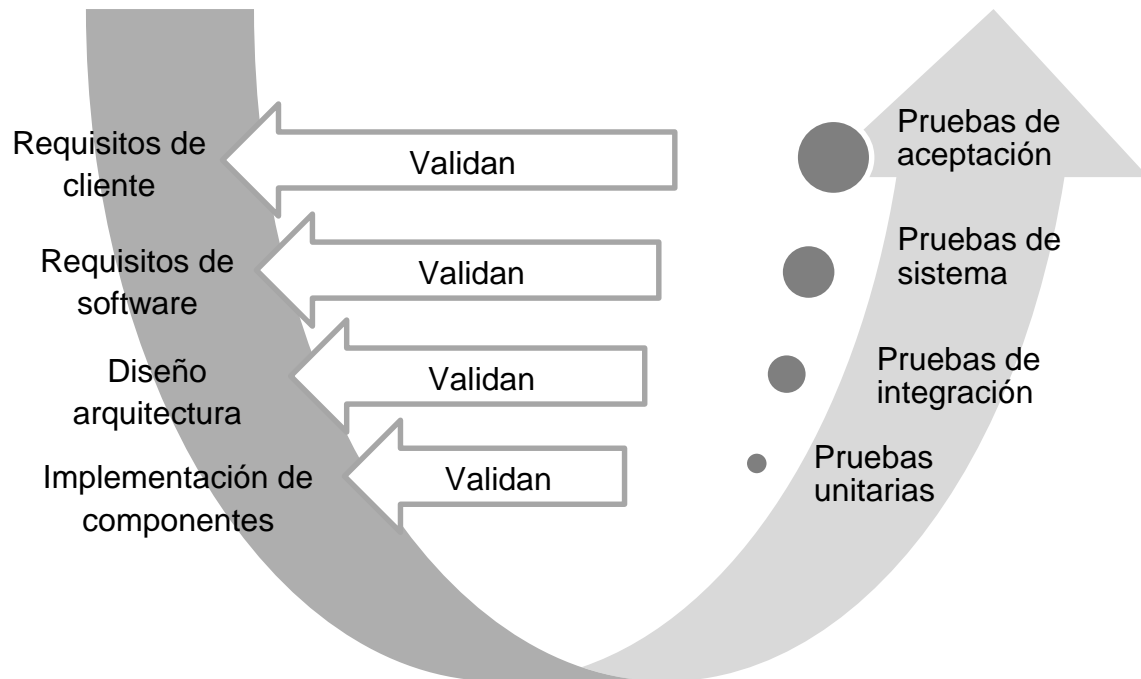


Imagen 1. Modelo en V. Fuente: IISSI (2013). Ciclo de Vida del Software. Recuperado de <http://www.lsi.us.es/docencia/get.php?id=7550>.

5.1.19. Ingeniería inversa.

Consiste en analizar el resultado de una fase del desarrollo de software para obtener el resultado de la anterior, normalmente analizar el código para obtener el diseño.

5.1.20. Reingeniería.

La reingeniería utiliza la información obtenida por la ingeniería inversa para aplicar cualquier tipo de mantenimiento. El mantenimiento preventivo del efecto 2000 ha sido el mayor esfuerzo de ingeniería inversa y reingeniería en la historia del software hasta la fecha. (El ciclo de vida del software, 2013)

5.1.21. Base De Datos

Date y López Gamino (2001) refieren que una base de datos se define como un conjunto de datos persistentes que es utilizado por los sistemas de aplicación de alguna empresa dada. Una base de datos representa algunos aspectos del mundo real, aquellos que le interesan al usuario y que almacena datos con un propósito específico. El término “persistente” se refiere al tipo de datos de la base de datos que difiere de otros datos más efímeros, como los datos de entrada, los datos de salida, las instrucciones de control, las colas de trabajo, los bloques de control de software, los resultados intermedios y de manera más general, cualquier dato que sea de naturaleza transitoria.

Las empresas son identificadas como cualquier organización independiente de tipo comercial, técnico, científico u otro, que puede estar compuesta por un solo individuo (con una pequeña base de datos personal), toda una corporación o un gran consorcio similar (con una gran base de datos compartida). Con la palabra “datos” se hace referencia a hechos conocidos que pueden registrarse, como ser números telefónicos, direcciones, nombres, etc.

5.1.22. Características De Las Bases De Datos

Arias (2014) afirma que las bases de datos se construyen a partir de los siguientes modelos:

5.1.22.1. Flat File (Archivo plano). Archivos planos que almacenan información.

5.1.22.2. Relacional. Organiza los datos en tablas y establece relaciones entre las tablas.

5.1.22.3. Orientado a objetos. Ordena los datos en clases y objetos.

5.1.22.4. Jerárquico. Modela los datos en forma de árbol.

5.1.22.5. Red. Se crea una estructura en forma de red, a partir de las referencias establecidas entre los datos.

5.1.22.6. Dimensional. Facilita la consulta de datos por medio de su integración en múltiples dimensiones.

5.1.23. Arquitectura De Las Bases De Datos

Date y López Gamino (2001) dictan que las arquitecturas de las bases de datos se basan en el modelo propuesto por el Grupo de Estudio en Sistemas de Administración de Bases de Datos de ANSI/SPARC. La arquitectura ANSI/SPARC se divide en tres niveles:

5.1.23.7. El nivel interno (también conocido como el nivel físico). Es el que está más cerca del almacenamiento físico; es decir, es el que tiene que ver con la forma en que los datos están almacenados físicamente.

5.1.23.8. El nivel externo (también conocido como el nivel lógico de usuario). Es el más próximo a los usuarios; es decir, el que tiene que ver con la forma en que los usuarios individuales ven los datos.

5.1.23.9. El nivel conceptual (también conocido como el nivel lógico de la comunidad, o en ocasiones sólo como el nivel lógico, sin calificar). Es un nivel de indirección entre los otros dos.

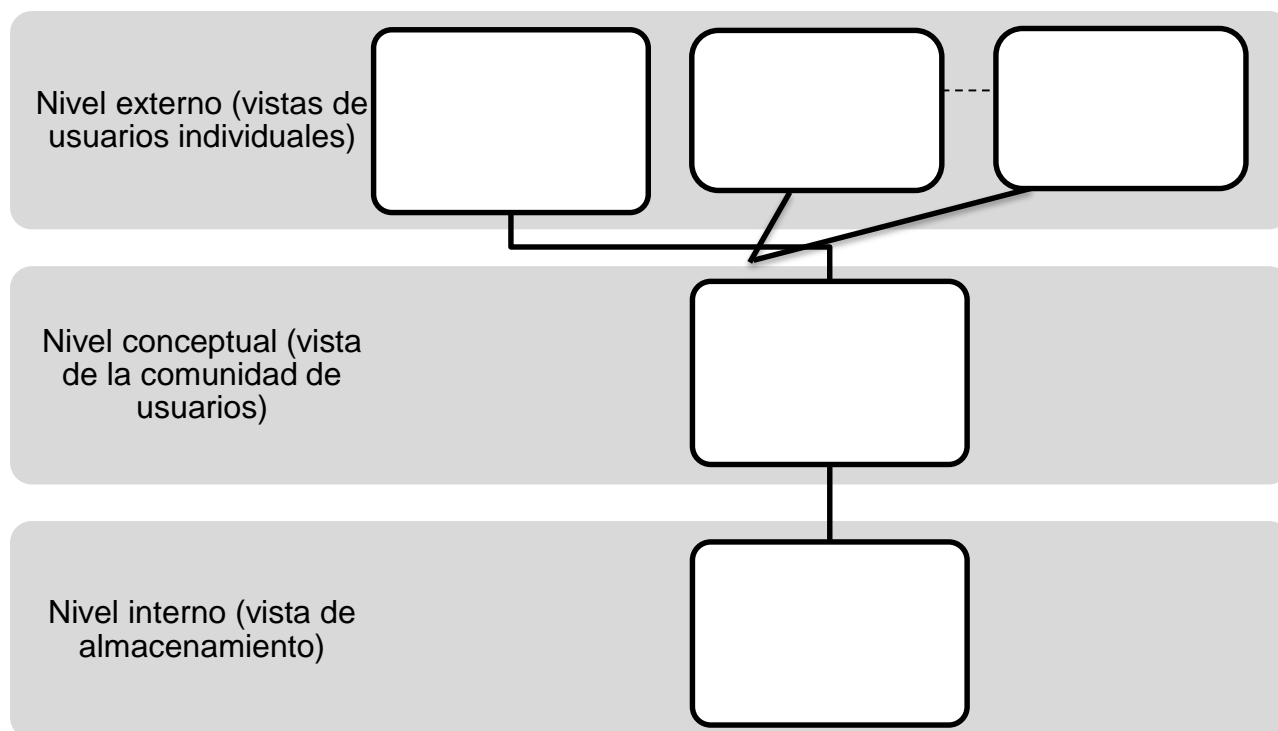


Imagen 2. Los tres niveles de arquitectura. Fuente: Bases de datos estructuradas. Recuperado de <https://es.slideshare.net/EduardoAlejandroGomez/bases-datos-estructuradas>

5.1.24. Elementos De Las Bases De Datos

Date y López Gamino (2001) explican que los elementos esenciales de las bases de datos son:

5.1.24.1. Datos. Son símbolos que describen condiciones, hechos, situaciones o valores. Los datos se caracterizan por no contener ninguna información. Un dato puede significar un número, una letra, un signo ortográfico o cualquier símbolo que represente una cantidad, una medida, una palabra o una descripción.

5.1.24.2. Métodos de acceso. Funciones que se encuentran en los entornos principales de computación que permite el acceso a los datos en dispositivos externos.

5.1.24.3. DBMS (sistemas de gestores de base de datos). Son sistemas computarizados que permiten almacenar y posteriormente acceder a las bases de datos de forma rápida y estructurada. Es el medio por el cual los usuarios tienen la posibilidad de realizar las operaciones básicas para manipular los datos.

5.1.24.4. Usuario. Se clasifican en tres tipos:

5.1.24.4.1. Programadores de aplicaciones.

Responsables de escribir los programas de aplicación de base de datos en algún lenguaje de programación como COBOL, PL/1, C++, Java o algún lenguaje de alto nivel.

5.1.24.4.2. Usuarios finales.

Son los que interactúan con el sistema desde estaciones de trabajo o terminales en línea. Un usuario final puede acceder a la base de datos a través de las aplicaciones en línea, o bien puede usar una interfaz proporcionada como parte integral del software del sistema de base de datos.

5.1.24.4.3. Administrador de base de datos.

Es el encargado de la toma las decisiones de estrategia y política con respecto a los datos de la empresa y proporciona el apoyo técnico necesario para implementar dichas decisiones.

5.1.25. MySQL

Es una base de datos relacional que utiliza el lenguaje SQL. Es un sistema de base de datos de código abierto, lanzado en 1995, que más tarde fue adquirido por Sun Microsystems en 2008 (luego Oracle compró Sun en 2009). (Benítez, 2017)

5.1.26. Características De MySQL

De acuerdo a lo expuesto por Fossati (2014), MySQL se caracteriza por tener:

- Soporte Multiusuario: múltiples usuarios tienen acceso a una o más bases de datos.
- Sistema de privilegios de usuarios potente y flexible.
- Esquemas de autenticación basados en usuario-máquina.
- Escalabilidad: las bases de datos pueden llegar a tener hasta 50 millones de registros.
- Portabilidad en múltiples sistemas operativos y procesadores.
- Fácil ajuste a diferentes estándares SQL.

5.1.27. Arquitectura De MySQL

La arquitectura de MySQL tiene como característica más notable el separar el motor de almacenamiento (que se encarga de los detalles de entrada-salida y representación de la información en memoria secundaria) del resto de los componentes de la arquitectura. Es decir, el diseño del gestor está preparado para que se pueda cambiar el gestor de almacenamiento. Esto

permite incluso crear nuevos motores de almacenamiento especializados para ciertas tareas o tipos de aplicaciones. (Cnx.org, 2018)

5.1.28. Arquitectura Cliente/Servidor

Ramos Martín y Ramos Martín (2007) afirman que en esta arquitectura se requiere que las aplicaciones se instalen en cada puesto de trabajo para que puedan acceder a la central de acceso de información. Sus componentes consisten en:

5.1.28.5. Servidor. Es el propio sistema gestor de base de datos. Se trata de un gran ordenador que actúa como depósito de los datos y permite llevar a cabo todas las funciones de un Sistema Gestor de Base de Datos (SGBD).

5.1.28.6. Cliente. Son los usuarios o aplicaciones de la arquitectura ubicados en estaciones de trabajo o computadores personales que solicitan servicios al servidor.

5.1.29. Entornos De Desarrollo Integrados IDE

Un entorno de desarrollo integrado es una aplicación virtual que sirve para la construcción de aplicaciones a partir de componentes. Poseen estas características generales:

- Una o más “paletas para mostrar los componentes disponibles en forma de iconos.
- Un “lienzo” o “contenedor” en el cual se colocan los componentes y se interconectan entre sí.
- Editores específicos para configurar y especializar los componentes.
- Ojeadores (browsers) para localizar componentes de acuerdo a ciertos criterios de búsqueda.

- Directorios de componentes.
- Acceso a editores, intérpretes, compiladores y depuradores para desarrollar nuevos componentes.
- Acceso a herramientas de control y gestión de proyectos, esenciales para grandes proyectos de software.

Ejemplos de IDE son Visual Studio de Microsoft, NetBeans de Apache Software Foundation, VisualAge de IBM o VisualCafe de Symantec, complementados con lenguajes de configuración como VBScript y JavaScript. (Ramos Salavert y Lozano Pérez, 2000)

5.1.30. Características De NetBeans

NetBeans es el IDE oficial para Java. De acuerdo a la página de Netbeans.org (2018), con sus editores, analizadores de código y conversores, puede actualizar sus aplicaciones de forma rápida y sin problemas para usar nuevas construcciones de lenguaje Java 8, como lambdas, operaciones funcionales y referencias de métodos.

Los analizadores y convertidores por lotes se proporcionan para buscar a través de múltiples aplicaciones al mismo tiempo, haciendo coincidir los patrones para la conversión a nuevas construcciones de lenguaje Java.

Con su editor Java en constante mejora, muchas funciones completas y una amplia gama de herramientas, plantillas y muestras, *NetBeans* IDE establece el estándar para el desarrollo con tecnologías de vanguardia listas para usar.

El editor de NetBeans introduce líneas, coincide con las palabras y los corchetes, y resalta el código fuente sintácticamente y semánticamente. Permite refaccionar fácilmente el código, con una

gama de herramientas útiles y potentes, mientras que también proporciona plantillas de códigos, sugerencias de codificación y generadores de códigos. El editor admite varios idiomas, desde Java, C / C ++, XML y HTML, hasta PHP, Groovy, Javadoc, JavaScript y JSP. Como el editor es extensible, puede conectar el soporte para muchos otros idiomas.

Con *NetBeans* se diseñan GUIs para aplicaciones Java SE, HTML5, Java EE, PHP, C / C ++ y Java ME de forma rápida y sin problemas mediante el uso de editores y herramientas de arrastrar y colocar en el IDE.

5.1.31. Lenguaje De Programación – Java

El lenguaje de programación Java es el más apropiado para explicar y aplicar la orientación a objetos. Muchas aplicaciones y sitios web no funcionan a menos que Java se encuentre instalado y cada día se crean más. Sus principales ventajas son su habilidad de permanencia que garantiza el funcionamiento de las aplicaciones a largo plazo y su fácil legibilidad.

Java se lanzó en 1991 como un lenguaje cuyos programas podían correr en cualquier plataforma y cumplir con la premisa de “escribir una vez, correrlo en cualquier parte”. Para lograr esto, Java corre sobre una máquina virtual, la cual funciona aislando al programa que corre sobre ella de los distintos hardware y sistemas operativos. De esta forma, la máquina donde corre el programa desarrollado siempre es la misma.

En la actualidad, Java forma parte de uno de los sistemas operativos para celular más famosos: Android. Este sistema operativo fue creado por los desarrolladores de Google; está basado en Java, por lo cual todas las aplicaciones para los celulares con Android están hechas

utilizando este lenguaje. El futuro de Java es prometedor, ya que está respaldado por gigantes de la informática y por una enorme comunidad de programadores. (Vivona, 2011)

5.1.32. Patrones De Arquitectura Por Capas

La programación en capas es una arquitectura que se basa en distribuir jerárquicamente los roles y las responsabilidades de un sistema para proporcionar una división efectiva de los problemas a resolver.

Los roles indican el tipo y la forma de la interacción con otras capas y las responsabilidades la funcionalidad que implementan. (Torre Llorente et al., 2010)

5.1.32.1. Capa de presentación. Es la capa encargada de la presentación visual de la aplicación. La capa de presentación mantiene una comunicación con la capa de negocio, transmitiéndole las solicitudes del usuario, ya sea de almacenaje, edición, o de recuperación de la información para la consulta respectiva y recibiendo información para ser mostrada. (Torre Llorente et al., 2010)

5.1.32.2. Capa de lógica de negocio. Aquí es donde se encuentran las funciones clases y funciones o procedimientos que serán invocados a través de la interfaz gráfica. Esta capa intermedia contendría los objetos que se corresponden con las entidades de la aplicación. Esta capa intermedia es la que conlleva a la aplicación a la capacidad de mantenimiento y de reutilización. Contendrá objetos definidos por clases reutilizables que se pueden utilizar una y otra vez en otras aplicaciones. (Torre Llorente et al., 2010)

5.1.32.3. Capa de datos. Se encuentran las clases y funciones que se conectan a la base de datos y es donde se realizan transacciones con SQL para leer, insertar, modificar o eliminar información en la base de datos. Es aquí donde se implementa las conexiones al servidor y la base de datos propiamente dicha, se invoca a los procedimientos almacenados los cuales reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio. (Pressman et al., 2002)

5.1.33. Persistencia

Este concepto se refiere al almacenamiento permanente de los datos. Su importancia radica en la finalidad de la mayoría de las aplicaciones de mostrar datos a los usuarios para que éstos puedan editarlos, lo cual implica transportar datos entre un origen y un destino; por ejemplo, los objetos obtenidos del modelo relacional de una base de datos y la interfaz gráfica de la aplicación que consume esos datos. Cuando la aplicación finalice, los datos deben persistir incluyendo los cambios que sobre ellos se hayan realizado. (Ceballos Sierra, 2015)

5.1.34. ORM

Ceballos Sierra (2015) define a un *ORM (Object Relational Mapping, mapeo objeto-relacional)* como la correspondencia bidireccional entre el modelo relacional y el modelo de objetos con el que trabajará una aplicación. Una forma sencilla de realizar un mapeo objeto-relacional para la gestión de datos relacionales en aplicaciones Java es JPA (*Java Persistent API*).

JPA es una especificación de Java para el acceso, la persistencia y la gestión de datos entre objetos y clases Java y una base de datos relacional. Se trata de una especificación (un

conjunto de interfaces), no de un producto; es decir, JPA no puede realizar la persistencia o cualquier otra cosa, sino que requiere de una implementación.

El principal objetivo de un ORM es delegar en utilidades externas la tarea de crear una correspondencia entre objetos y tablas. Una de las herramientas para desarrollar esta labor es *Hibernate*.

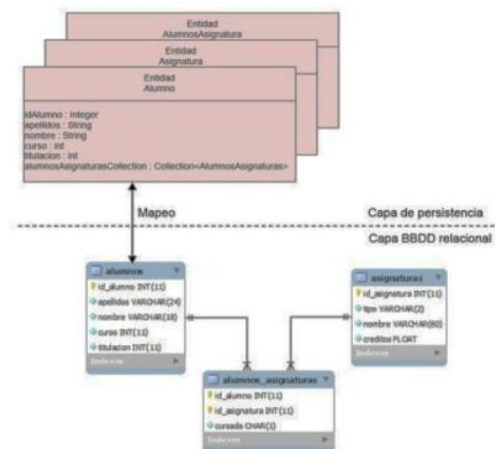


Imagen 3. Persistencia y ORM. Fuente: Ceballos Sierra, F. (2015). *JAVA. Interfaces gráficas y aplicaciones para Internet. 4ª ed.* Madrid, España: Grupo Editorial RA-MA.

5.1.35. Hibernate

Hibernate permite a los desarrolladores escribir con mayor facilidad aplicaciones cuyos datos sobreviven al proceso de solicitud. Como marco de ORM, *Hibernate* se preocupa por la persistencia de los datos tal como se aplica a las bases de datos relacionales (a través de JDBC).

Hibernate desarrolla clases persistentes siguiendo expresiones idiomáticas orientadas a objetos naturales que incluyen herencia, polimorfismo, asociación, composición y el marco de colecciones de Java. No requiere interfaces o clases base para clases persistentes y permite que cualquier clase o estructura de datos sea persistente. Es conocido por su excelente estabilidad y calidad, probado por la aceptación y el uso de decenas de miles de desarrolladores de Java.

Hibernate ofrece constantemente un rendimiento superior al código JDBC directo, tanto en términos de productividad del desarrollador como de rendimiento en tiempo de ejecución. Fue diseñado para funcionar en un clúster de servidores de aplicaciones y ofrecer una arquitectura altamente escalable. Se puede utilizar para manejar una Intranet que sirve a cientos de usuarios o para aplicaciones de misión crítica que sirven a cientos de miles. (hibernate.org/orm/, 2018)

5.1.36. Frameworks

Es el conjunto de clases cooperativas que construyen un diseño reutilizable para un tipo específico de software. Un Framework proporciona la arquitectura partiendo el diseño en clases abstractas y definiendo sus responsabilidades y colaboraciones. Un desarrollador realiza una aplicación haciendo subclases y componiendo instancias a partir de las clases definidas por el Framework.

Un framework es una aplicación semi-completa, de control invertido, así un Framework difiere de una librería de clases:

- En una librería de clases, el control del flujo se encuentra en el código de la aplicación que realiza llamadas a los métodos de la librería de clases.
- En un framework, el control del flujo está en código del framework que realiza llamadas al código de la aplicación (control invertido)

Los frameworks son la aplicación, de forma que manipulan el flujo de control e invocan el código del usuario. Este código puede derivar partes del Framework. Es muy poca la cantidad de nuevo código a implementar y permite su reutilización, no se necesita de mucho tiempo para desarrollarlo y su costo de mantenimiento es bajo. (Galindo Haro, J. and Camps Riba, J., 2018)

5.1.37. JasperReports

JasperReports es una biblioteca íntegramente desarrollada en Java y que puede ser usada en una variedad de aplicaciones de este lenguaje, incluyendo J2EE o aplicaciones web, para generar contenido dinámico. Esta biblioteca genera reportes que pueden ser utilizados para mostrar información en pantalla, enviarlos a colas de impresión o a archivos en formatos PDF, HTML, RTF, XLS, ODT, CSV o XML.

Al igual que la mayoría de aplicativos de reporte, *JasperReports* utiliza plantillas de reporte estructuradas en múltiples secciones, como el título, resumen, detalles, cabeceras y pie de páginas. Cada sección tiene un diseño de forma libre en el que puede colocar varios tipos de elementos, incluidas imágenes, campos de texto estáticos y dinámicos, líneas y rectángulos. El motor de creación de informes utiliza esta plantilla para organizar los datos en un archivo XML (JRXML) o crearlos mediante programación utilizando la API de la biblioteca.

Estos datos pueden provenir de varias fuentes de datos, incluidas bases de datos relacionales, colecciones o matrices de objetos Java o datos XML. (Danciu y Chirita, 2007)

Este proyecto aplicado posee un manual de instalación de esta biblioteca como parte de la aplicación desarrollada (Ver anexo A).

5.2. Marco Conceptual

El desarrollo del software de este proyecto de investigación se realiza con base en tres pilares fundamentales:

5.2.1. Programación Orientada A Objetos

Paradigma de la programación general que se caracteriza por su explicación de la realidad del negocio que rodea con mayor claridad que otros paradigmas. Su metodología se basa en modelar los problemas y caracteres del entorno a través de objetos.

Por medio de los objetos se definen los atributos y acciones que hacen parte de su identidad, comportamiento y estado. A partir de la creación de los objetos se procede a definir la interacción que puede existir entre ellos y la solución que desarrollan en su entorno.

5.2.2. Implementación De Lenguaje De Programación Java

Java es el lenguaje de programación que mejor explica la connotación que tienen los objetos en un sistema. Los objetos son prioridad del desarrollo de software en Java debido a que no se puede hacer nada sin usar al menos un objeto, a diferencia del lenguaje C++, que desarrolla acciones sin crear objetos.

Para desarrollar en Java se emplea software especializado *NetBeans*. Es un entorno de desarrollo compilado para interpretar diversas plataformas de programación y proporciona herramientas esenciales para una creación de software mucho más dinámica y robusta.

5.2.3. Uso De Gestor De Base De Datos

La elección de un gestor de base de datos es una pieza importante para el desarrollo de un software que maneje datos que incidan en la mecánica del negocio. Existen diferentes gestores de base de datos, pero para el caso de este proyecto de investigación se busca la opción más económica que realice esta tarea. Es por ello que se decide emplear MySQL mediante el servidor XAMPP. Esta herramienta se caracteriza por su acceso gratuito y código abierto, por lo que su uso es ideal para comprender la complejidad que puede presentar una base de datos relacional.

Con estas tres características se espera que la satisfacción de los empleados y los pacientes suscritos a la empresa *Vital Plus* sea un hecho y que contribuya a la evolución de la empresa.

5.2.4. Diccionario De Datos

Uno de los elementos más importantes de un gestor de base de datos es el diccionario de datos. Camuña Rodríguez (2015) define el diccionario de datos como el lugar donde se almacena la información correspondiente a todos los datos que forman la base de datos y sus objetos.

Por medio del diccionario de datos se establece y describe la estructura física y lógica de la base de datos, la definición y espacio ocupado por sus objetos, los valores por defecto de las columnas de las tablas, las restricciones de integridad de los datos y los roles y privilegios de los usuarios.

Los diccionarios de datos resultan de gran utilidad para conocer y gestionar de mejor manera los lineamientos fundamentales que poseen las bases de datos de gran tamaño y que se encuentran en constante cambio.

Durante las operaciones transaccionales de la empresa *Vital Plus*, se pueden determinar las siguientes variables:

5.2.5. Método De Pago

Los pacientes pueden realizar el pago de abonos de sus tratamientos mediante tres formas: efectivo, tarjeta de crédito y tarjeta débito. La primera de ellas registra el pago sin ningún descuento aplicable, en cambio, las dos formas restantes aplican una variación en el monto final del abono. El pago realizado con tarjeta de crédito recibe un descuento del 5.5%,

mientras que a la transacción realizada con tarjeta débito se le aplica un descuento del 4.6%. A través de estos descuentos que se realizan o no en los abonos, es que se determina el saldo pendiente de cada paciente en el total del costo total del tratamiento en el que está suscrito. Cada paciente obtiene la información del descuento aplicado y del saldo que tiene pendiente con la empresa.

5.2.6. Tipo De Abono

La aplicación distingue qué clase de abono se va a registrar en base de datos. En tratamientos ordinarios se registran cuotas iniciales en primera instancia para que luego se puedan registrar abonos regulares. La aplicación valida si hay una cuota inicial consignada en el tratamiento para poder registrar los subsiguientes abonos regulares.

5.2.7. Tipo De Tratamiento

En la empresa se dan dos tipos de conceptos dentro del registro de los abonos: tratamientos y gastos extra. Los tratamientos requieren de una cuota inicial registrada para poder empezar su contabilidad y seguimiento. En cambio, los gastos extra no necesitan de una cuota inicial para ser registrados en base de datos.

5.2.8. Tipo De Usuario

Existen dos tipos de usuario: administrador y agente. El primer registro de usuario lo realiza el administrador principal de la aplicación. Luego, la empresa determina los roles de los usuarios que trabajarán en la aplicación para que el administrador principal pueda registrarlos y darles los privilegios acordes a sus roles.

5.2.9. Datos Personales De Los Pacientes

Los usuarios agentes realizan el registro, modificación y consulta de los datos personales de los pacientes suscritos a la empresa. Datos como nombre, dirección y teléfono son consultados en base de datos y siempre dispuestos a ser modificados cuando el paciente lo requiera.

Estos conceptos mencionados tienen gran incidencia en las labores cotidianas de la empresa y son de gran importancia para que la satisfacción de los pacientes suscritos se garantice.

5.3. Marco Contextual

5.3.1. Reseña Histórica De *Vital Plus*

La empresa *Vital Plus* se ha dedicado a prestar servicios de salud oral a pacientes que acuden directamente a sus instalaciones y a pacientes que son remitidos de otras entidades para una atención más especial y dedicada, como parte de su actividad como clínica odontológica en la ciudad de Barranquilla. Su fundación data del año 2007 por parte de jóvenes inversionistas de origen turco que se reunieron en esta ciudad con la idea de postular una alternativa de solución innovadora en tratamientos odontológicos a la industria de la ciudad.

5.3.2. Misión

Ofrecer integridad de servicios de salud oral que contribuya a mejorar la calidad de vida de nuestros pacientes, poniendo a su disposición profesionales capacitados y tratamientos asequibles para que puedan tener sonrisas sanas y bonitas sin traumas.

5.3.3. Visión

Alcanzar una posición de prestigio y reconocimiento en la ciudad de Barranquilla como ente especializado en la prestación de servicios de salud oral capaz de satisfacer las necesidades de nuestros pacientes y presentando una organización sólida y confiable.

5.3.4. Organigrama

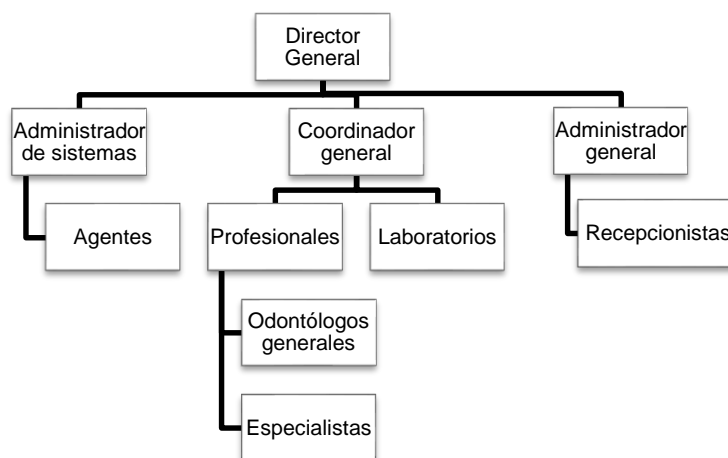


Imagen 4. Organigrama de empresa Vital Plus

5.4. Antecedentes

5.4.1. Antecedentes nacionales

5.4.1.1. WebPicture.

WebPicture (Guzmán, 2014) es una herramienta de arquitectura empresarial construida para generar editores gráficos ad-hoc en la ciudad de Bogotá, Colombia. El objetivo de WebPicture es ofrecer a los arquitectos un conjunto de herramientas que les permitan crear editores ad-hoc fácil y eficientemente utilizando como base herramientas estándar de modelamiento.

WebPicture permite a los usuarios crear editores a la medida utilizando como base un meta modelo creado en EMF y la representación gráfica de sus elementos descrita en *Picture*.

5.4.1.2. Diseño e implementación de un prototipo para el manejo de los datos estadísticos del uso de los recursos electrónicos de la biblioteca de la Universidad de Los Andes.

Eraso Peña (2014) describe al Sistema de Información de los Recursos Electrónicos de Bibliotecas (SIREB) como un proyecto que se desarrolla en conjunto con el Sistema de Bibliotecas de la Universidad de Los Andes (SBUA) y la Dirección de Servicios de Información y Tecnología (DSIT) en Bogotá, Colombia, con el objetivo de encontrar una solución al manejo de los datos estadísticos de los recursos electrónicos que el sistema de bibliotecas ofrece. Con el proyecto se busca manejar de manera óptima la obtención, procesado y almacenamiento de los datos estadísticos de tal forma que puedan ofrecer una serie de reportes informativos, en un tiempo óptimo que apoyen el proceso de toma de decisiones.

5.4.1.3. Generación de un campus activo a través de una solución basada en movilidad tipo pervasive.

En el proyecto de Melani De La Hoz (2014) realizado en la ciudad de Bogotá, Colombia, se desarrolla una aplicación móvil que, a través del uso de sensores GPS y *beacons*, es capaz de ubicar al usuario en el campus y proveerle información acerca del lugar en que se encuentra. La solución igualmente puede identificar lugares que son frecuentemente visitados por el usuario y el horario en que frecuenta estos sitios para enviarle información acerca de éstos de manera anticipada. Se presentan eventos que ocurren en el campus y principalmente se muestran aquellos que tratan temas de interés para el usuario.

5.4.2. Antecedentes internacionales

5.4.2.1. GESCLIDENT.

Benayas Álamos, Marco Bahón y Torrijos Santos (2008) en la ciudad de Madrid, España, realizaron el “desarrollo de una aplicación para la gestión de una clínica dental con accesibilidad a través de la Web utilizando tecnologías en auge como MYSQL o PHP 5 que permiten tener un completo control de Bases De Datos desde cualquier punto.

Los elementos básicos de información y organización tradicionales tales como agendas o grandes almacenes con fichas de pacientes van a quedar obsoletos gracias a este sistema de gestión. Otro objetivo es proporcionar al paciente un acercamiento a toda su información a través de la red teniendo acceso a todo su historial, así como a los presupuestos y pagos realizados.

Esta herramienta es flexible, abierta y ampliable de modo que se mejora la gestión, el acceso, la interactividad y la utilidad de la gran cantidad de información de la que se dispone en un negocio de estas características ya que está adaptada para cualquier tipo de usuario (con conocimientos avanzados de informática o sin ellos) además de mejorar la experiencia del paciente.”

5.4.2.2. ODONTOWEB.

Duarte (2017) presentó un proyecto monográfico en Managua, Nicaragua, que tenía como meta “facilitar las gestiones primordiales de las clínicas odontológicas, entre ellas: gestión de pacientes, citas, registro de odontogramas, proformas y pagos por servicios, entre otros, por medio de una aplicación web. ODONTOWEB es una aplicación web orientada al sector de medicina dental nicaragüense, desarrollado en la herramienta Java Enterprise Edition de Eclipse, la cual incluye extensiones para la creación de aplicaciones web de manera sencilla.

También se utilizó HTML5 y CSS3 para la capa de vista del lado del cliente, utilizando la normativa ISO 9241 como estándar de diseño Web. Posee recargado asíncrono de objetos gracias a la implementación de AJAX utilizando JavaScript y JQuery. Este sistema fue ideado con la visión de ser integral, de manera que con solo registrarse y realizar algunas configuraciones básicas pueda ser utilizado por los odontólogos, desde sus clínicas o consultorios.

Surgió como alternativa a varias herramientas que son ofertadas a precios y condiciones de uso que las hacen prohibitivas al entorno local. La oportunidad de contar con información actualizada, gráfica – por medio del odontograma – y de acceso inmediato, es muy importante, tanto para los odontólogos como para los pacientes, dado que facilita la revisión del historial odontológico y el progreso en los tratamientos y trabajos dentales realizados.”

5.4.2.3. Análisis, diseño e implementación de un portal web para la gestión y administración hospitalaria de la clínica dental Barrera.

Palacios Villafuerte (2013) desarrolló en Sangolquí, Ecuador, una aplicación que podía “automatizar y mejorar los procesos relacionados con la administración de pacientes de la Clínica Dental Barrera.

Este proyecto se realizó utilizando la metodología ASD – Desarrollo Adaptivo de Software, que se orienta hacia la entrega de componentes útiles y funcionales para el usuario final en períodos cortos de tiempo, en lugar de documentación excesiva o procesos innecesarios, haciendo que el usuario interactúe de manera proactiva y constante durante la construcción del software, a más de ello propone que el equipo de desarrollo esté dispuesto a dar solución a los cambios solicitados por el usuario, y a la entrega de un producto final a la medida de las necesidades de la clínica.

El software fue desarrollado en un patrón de diseño Modelo Vista Controlador, lenguaje de programación JAVA, los frameworks ICEFACES, HIBERNATE, SPRING, y MySQL como motor de base de datos, obteniendo así una aplicación robusta, distribuida y de alto rendimiento.”

6. Marco Metodológico

6.1. Metodología De La Investigación

6.1.1. Paradigma de investigación

Este proyecto aplicado está enmarcado en el paradigma cuantitativo, puesto que se realizan grupos de discusión con el personal de sistemas de la empresa *Vital Plus* para proceder al análisis e interpretación de datos.

6.1.2. Tipo de investigación

Para el desarrollo de este proyecto, se realiza una investigación-acción participativa para mejorar la situación contable de la empresa, asegurando también la participación de los propios empleados de la empresa. Con esto se logra que la empresa pase de ser un objeto de estudio a ser sujeto protagonista de la investigación y se lleve a cabo un proceso de interacción permanente entre el personal técnico investigador y la empresa, lo cual permite obtener un mayor involucramiento y colaboración de los empleados para actuar progresivamente en la prestación de servicios de salud oral.

6.1.3. Metodología De Desarrollo

6.1.3.1. Procedimiento

La consecución de objetivos específicos para lograr el objetivo general de este proyecto aplicado plantea el desarrollo de un prototipo. Esta metodología implica la elaboración de

modelos experimentales que permitan ver la funcionalidad básica del sistema, con el fin de que el personal administrativo de la empresa *Vital Plus* pueda realizar una evaluación anticipada del producto e interactuar con el personal de sistemas para saber si se cumplen con los requerimientos propuestos. Esta metodología proporciona un alto grado de participación del personal administrativo de la empresa para lograr su satisfacción en soluciones de software.

6.1.4. Fuentes Y Técnicas De Recolección De Información

6.1.4.2. Fuentes primarias

6.1.4.2.1. Libros certificados

6.1.4.2.2. Artículos

6.1.4.3. Fuentes secundarias

6.1.4.3.1. Revistas

6.1.5. Universo O Población

Personal administrativo y operativo de la empresa *Vital Plus*, pacientes suscritos a la empresa.

6.1.6. Muestra

Registros de facturación de los pacientes en los tratamientos en que se encuentran suscritos.

6.1.7. Delimitación Y Alcance

El alcance de este proyecto aplicado está determinado para la empresa *Vital Plus* en su sede en la ciudad de Barranquilla.

6.1.8. Técnicas de análisis de datos

En la presente investigación se utilizarán técnicas de análisis de datos cualitativos, los cuales permitirán comprender los atributos que deben ser registrados para que la participación de los actores y la ejecución de procesos del sistema sean efectivas. También se realizan análisis cuantitativos, debido a que se necesita controlar la contabilidad de las facturas generadas por conceptos de abonos de pacientes suscritos a la empresa.

6.1.9. Técnicas de procesamiento de datos

Se trabaja en tiempo real, requiriendo datos iniciales de entrada que conllevan a realizar operaciones computacionales que se registran en base de datos para su posterior manipulación.

6.1.10. Esquema Temático

- Levantamiento de información.
- Análisis de actores y procesos de la empresa *Vital Plus*.
- Organigrama de la empresa *Vital Plus*.
- Hardware disponible en la empresa *Vital Plus*.
- Paradigma de programación orientada a objetos.
- Lenguajes de programación orientada a objetos.
- Desarrollo de aplicaciones Java.
- Uso de entornos de desarrollo de software.
- Desarrollo de aplicaciones de escritorio.
- Montaje de servidor de base de datos.

7. Cronograma De Actividades

ACTIVIDAD	Inicio	Fin	Duración	Enero				Febrero				Marzo				
				Semana 1	Semana 2	Semana 3	Semana 4	Semana 1	Semana 2	Semana 3	Semana 4	Semana 1	Semana 2	Semana 3	Semana 4	
Análisis de requerimientos	8/01/2018	14/01/2018	1 semana		■											
Desarrollo de la aplicación	15/01/2018	25/02/2018	6 semanas		■	■	■	■	■	■						
Pruebas	26/02/2018	4/03/2018	1 semana								■					
Montaje en la empresa	5/03/2018	11/03/2018	1 semana									■				
Inducción de funcionamiento	12/03/2018	18/03/2018	1 semana										■			

Tabla 1. Cronograma de actividades

8. Recursos Necesarios

RECURSO	DESCRIPCIÓN	PRESUPUESTO
Equipo Humano	Un programador	\$0
Equipos y Software	Un computador, software <i>NetBeans</i>	\$1'637.000
Viajes y Salidas de Campo	Reuniones con departamento de sistemas en la empresa (Transporte)	\$40.000
Materiales y suministros	Servidor de base de datos	\$620.000
Bibliografía		\$0
TOTAL		\$2'297.000

Tabla 2. Recursos necesarios

9. Resultados Esperados

RESULTADO/PRODUCTO ESPERADO	INDICADOR	BENEFICIARIO
Aplicación desarrollada en Java	Cualitativo	Empresa

Tabla 3. Resultados esperados

10. Capítulo I

10.1. Análisis

La empresa *Vital Plus* necesita un sistema que le ayude a llevar un control detallado de sus operaciones, lograr una mejor visualización y entendimiento por parte de sus trabajadores y prestar un mejor servicio a sus pacientes suscritos.

Dicha necesidad hace que éste proyecto aplicado elabore un software de aplicación exclusiva para la empresa, ya que este tipo de software resuelve la necesidad específica del negocio. Estas aplicaciones se caracterizan por su procesamiento de datos comerciales o técnicos de tal manera que las operaciones de negocios o la toma de decisiones administrativas o técnicas son más acertadas. Este tipo software permite controlar las funciones que realiza la empresa en tiempo real.

La empresa necesita que la aplicación se use dentro de la empresa, por lo cual se analiza el lenguaje de programación que se utilizará para desarrollar la aplicación. Siendo que lenguajes como PHP o HTML5 sólo trabajan en el ámbito web y tienen ciertas vulnerabilidades que la empresa desea evitar para que la información que maneja no sea manipulada por entes externos, estos lenguajes son descartados.

Es por ello que se decide desarrollar en Java debido a que es un lenguaje de programación multiplataforma, es decir, que una aplicación desarrollada en Java puede funcionar en cualquier tipo de computador o sistema operativo. Además, permite crear aplicaciones de escritorio distribuidas (se pueden ejecutar dentro de la red propia de la empresa) y operar con mucha estabilidad en caso de realizar análisis de datos de gran magnitud, aspectos fundamentales que necesita la empresa para optimizar sus labores diarias ante una posible alta demanda de procesamiento.

El desarrollo del software de este proyecto de investigación se realiza con base en tres pilares fundamentales:

10.1.1. Programación orientada a objetos.

Paradigma de la programación general que se caracteriza por su explicación de la realidad del negocio que rodea con mayor claridad que otros paradigmas. Su metodología se basa en modelar los problemas y caracteres del entorno a través de objetos.

Por medio de los objetos se definen los atributos y acciones que hacen parte de su identidad, comportamiento y estado. A partir de la creación de los objetos se procede a definir la interacción que puede existir entre ellos y la solución que desarrollan en su entorno.

10.1.2. Implementación de lenguaje de programación Java.

Java es el lenguaje de programación que mejor explica la connotación que tienen los objetos en un sistema. Los objetos son prioridad del desarrollo de software en Java debido a que no se puede hacer nada sin usar al menos un objeto, a diferencia del lenguaje C++, que desarrolla acciones sin crear objetos.

Para desarrollar en Java se emplea software especializado NetBeans. Es un entorno de desarrollo compilado para interpretar diversas plataformas de programación y proporciona herramientas esenciales para una creación de software mucho más dinámica y robusta.

Java ofrece alternativas de seguridad más efectivas en comparación a los lenguajes de programación web, Creus Solé (2005) afirma que Java proporciona numerosas comprobaciones durante la compilación y ejecución de las aplicaciones para hacerlas robustas y así evitar infiltraciones a los códigos por parte de terceros o virus que pueden afectar el buen andamiaje de las labores de la empresa. Se implementa el software especializado llamado NetBeans para desarrollar, debido a su orientación a objetos y su compatibilidad con múltiples lenguajes y entornos de desarrollo programado y de almacenamiento.

10.1.3. Uso de gestor de base de datos.

La elección de un gestor de base de datos es una pieza importante para el desarrollo de un software que maneje datos que incidan en la mecánica del negocio. Existen diferentes gestores de base de datos, pero para el caso de este proyecto de investigación se busca la opción más económica que realice esta tarea. Es por ello que se decide emplear MySQL mediante el paquete XAMPP. Esta herramienta se caracteriza por su acceso gratuito y código abierto, por lo que su uso es ideal para comprender la complejidad que puede presentar una base de datos relacional.

Haciendo uso de estas tres principales herramientas, se logra que los empleados (administradores) puedan exportar la información relacionada a los pacientes y todos los procesos que se realizan en la empresa a archivos ofimáticos, lo cual confirma su propiedad de desarrollo ideal para entornos empresariales especializados debido a su robustez y capacidad de computación científica.

Los siguientes son requerimientos no funcionales que debe cumplir el diseño de la aplicación:

10.1.4. Rendimiento.

Una vez se establece la comunicación con la base de datos (de manera automática al iniciar), la aplicación no presenta demoras en su ejecución. Cada vez que un botón es presionado, se realiza una acción que llama a otra ventana de acuerdo con lo requerido por el usuario.

10.1.5. Confiabilidad.

La aplicación despliega la contabilidad de los abonos registrados de los pacientes de forma detallada según las solicitudes de visualización para la empresa y permite realizar oportunamente las operaciones básicas (inserción, modificación, eliminación y consulta) sobre la

información almacenada en base de datos de los distintos actores y procedimientos que hacen parte de la empresa.

10.1.6. Disponibilidad.

Siempre que se mantenga activo el servidor de base de datos, la aplicación va a responder y ejecutarse en cualquier momento que se solicite durante las jornadas con la misma velocidad.

10.1.7. Seguridad.

Información sensible, como contraseñas, se mantiene codificada bajo el algoritmo de cifrado SHA-1, el cual garantiza la seguridad de los usuarios que utilizan la aplicación en caso de sufrir algún tipo de ataque informático externo a la empresa. La aplicación arranca a trabajar solamente si la validación de credenciales del administrador o jefe de sistemas para acceder a la base de datos es correcta. Ningún usuario diferente al administrador tiene acceso a la base de datos. La aplicación sólo se instalará en las dependencias en donde la empresa la necesite y no se encontrará disponible en ningún espacio web, por lo cual no habrá lugar a una manipulación de la información que se maneje en la empresa por parte de cualquier ente ajeno.

10.1.8. Mantenibilidad.

En caso de presentar fallos o solicitar alguna actualización, dentro y fuera de la empresa se encuentra personal de apoyo capacitado en sistemas de información con conocimientos en Java para realizar su análisis de errores y desarrollar su actualización, respectivamente.

Se realizan reuniones con el personal de sistemas de la empresa *Vital Plus* con el fin de realizar el análisis de requerimientos, diagnosticar las necesidades que tiene la empresa en sus recursos sistemáticos, determinar los actores y la estructura básica del sistema y las pruebas,

ejecución y montaje de la aplicación en todas las dependencias o zonas laborales en donde la necesite la empresa.

La información relacionada a los especialistas que laboran en la empresa, los tratamientos que ofrece, los usuarios que estarán operando la aplicación, las posibles variantes de transacción (abonos para tratamiento o cargo extra) y operaciones aritméticas que se realizan dependiendo del método de pago utilizado por los pacientes (tarjeta de crédito, tarjeta débito o efectivo) son suministrados por el personal de sistemas durante las reuniones efectuadas en la empresa.

A través de dichas reuniones se determina que la empresa dispone de dos posibles actores operadores de la aplicación: administrador y agente. Cada uno con sus privilegios y labores específicas acordes a su rol dentro de la empresa. Dependiendo del actor, la aplicación posee entornos relacionados con las tareas que el actor debe cumplir en la empresa. Los entornos deben estar diseñados de la siguiente manera:

10.1.9. Entorno De Administrador.

10.1.9.1. Registro, modificación, eliminación y consulta de usuarios agentes.

Espacio para la manipulación de información correspondiente a los agentes que manejarán la aplicación.

10.1.9.2. Registro, modificación, eliminación y consulta de profesionales.

Espacio para la manipulación de información correspondiente a los profesionales que laboran en la empresa.

10.1.9.3. Registro, modificación, eliminación y consulta de laboratorios.

Espacio para la manipulación de información correspondiente a los laboratorios relacionados con la empresa.

10.1.9.4. Registro, modificación, eliminación y consulta de tratamientos.

Espacio para la manipulación de información correspondiente a los tratamientos que ofrece a sus pacientes, mostrándose como solución alternativa para la salud oral.

10.1.9.5. Registro, modificación, eliminación y consulta en facturación.

Espacio para la manipulación de información correspondiente a las facturas generadas durante los tratamientos o cargos extra que pagan los pacientes a la empresa por la prestación de servicios.

10.1.9.6. Consulta de actividad de agentes.

Se buscan los agentes que hicieron uso o ingresaron y cerraron sesión de la aplicación durante una fecha determinada.

10.1.10. Entorno De Agente.**10.1.10.7. Registro, modificación, eliminación y consulta de pacientes.**

Espacio para la manipulación de información correspondiente a los pacientes afiliados a la empresa (datos personales, tratamiento que requiere, especialista encargado).

10.1.10.8. Registro, modificación, eliminación y consulta de abonos.

Espacio para la manipulación de información correspondiente a los abonos que realizan los pacientes dependiendo del método de pago y el tipo de concepto de pago.

10.1.10.9. Registro, modificación, y consulta de tratamientos auxiliares.

Espacio para la manipulación de información correspondiente a cargos extra que se pueden presentar como novedad dentro del lapso del tratamiento en el que está suscrito el paciente y que se incluyen en la historia clínica del paciente. También se puede ingresar uno o varios tratamientos adicionales a los que el paciente se suscribe y que, por ende, harán parte de su historia clínica con la empresa.

10.1.10.10. Consulta de balance de abonos por criterio

Espacio dedicado a la consulta en base de datos de los abonos registrados por un criterio determinado por el agente y que se muestran como mensaje en un cuadro de diálogo:

10.1.10.10.1. Por profesional.

Se puede consultar el balance total en dinero recogido por el profesional y en una fecha determinada por el agente.

10.1.10.10.2. Por especialidad.

Se consulta el balance total en dinero y en una fecha o un mes determinado por el agente bajo el concepto de la especialidad escogida para consulta.

10.1.10.10.3. Por día.

Se muestra la cantidad de dinero registrada durante el día en curso o en una fecha determinada por el agente.

10.2. Procesos

La aplicación realiza operaciones matemáticas de descuento acorde a cómo se desarrollen las transacciones de abono por parte de los pacientes en el entorno de registro de abonos del espacio de agentes y que se almacenan en la base de datos para su manipulación. Si el registro del abono se hace usando una tarjeta de crédito, el paciente obtiene un 5,5% de descuento en el abono del tratamiento. Si el paciente paga su abono con una tarjeta débito, entonces recibe un 4,6% de descuento en el abono. Si el pago se realiza en efectivo, se registra el valor del abono sin aplicar ningún descuento.

Además, la aplicación debe disponer de opciones para exportar información detallada a archivos planos en distintos formatos, ya sea en PDF (Adobe Reader), XLS (Microsoft Excel), HTML (Internet Explorer) o DOC (Microsoft Word), con el fin de que el administrador tenga, desde su entorno, la oportunidad de tener un mejor manejo de las transacciones u operaciones que se han realizado en la empresa.

Para el almacenamiento de los datos que se utilizan en la empresa, se emplea MySQL. Este es un sistema de administración de base de datos, utilizado por el lenguaje SQL como interfaz. Las principales características de este motor son:

- Software libre
- Utiliza pocos recursos de hardware
- Compatible con múltiples lenguajes de programación (Delphi, Java, C/C++, Visual Basic, Python, Perl, PHP, ASP y Ruby)
- Excelente estabilidad
- Interfaces gráficas de fácil utilización

- Portable

MySQL representa menos costos para cualquier entidad que desee usarla, pues no necesita el pago de ninguna licencia, a diferencia de otros sistemas de gestión de base de datos igual de potentes como DB2.

Para utilizar una base de datos en MySQL, se debe realizar la instalación de un servidor que pueda servir de intermediario de cualquier aplicación que se desee ejecutar. Por esto, se utiliza XAMPP.

XAMPP es un servidor web de plataforma, software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script PHP y Perl. El nombre es en realidad un acrónimo: **X** (para cualquiera de los diferentes sistemas operativos), **A**pache, **M**ariaDB/MySQL, **P**HP, **P**erl. Es de fácil descarga e instalación y ofrece garantías para proteger las partes más importantes de una base de datos empleada por una aplicación o página web.

10.3. Diagrama De Procesos

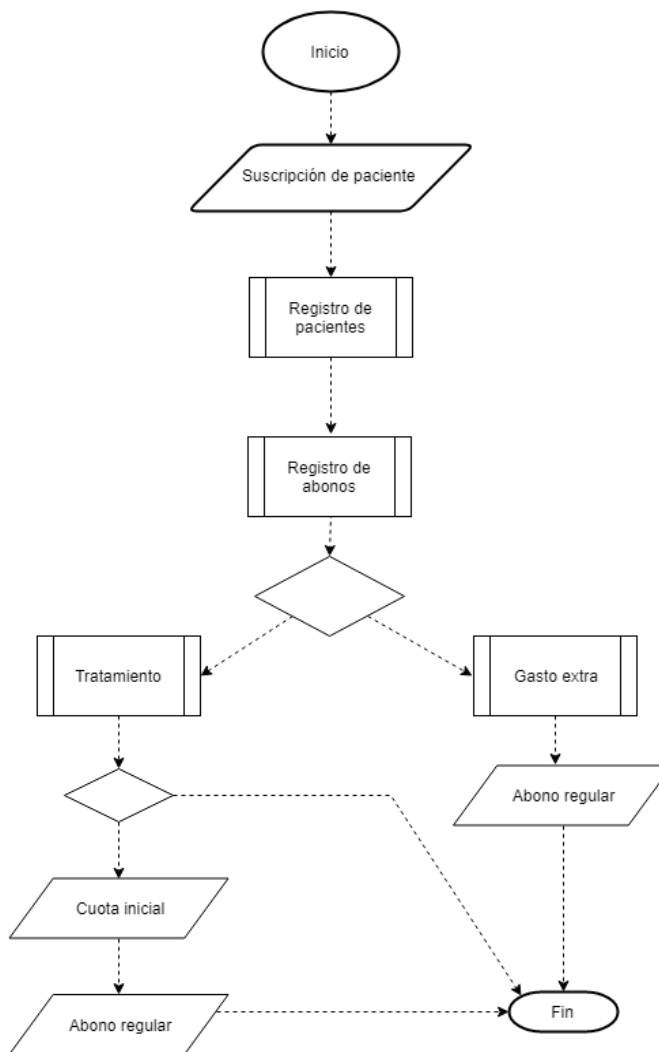


Imagen 5. Diagrama de procesos

10.4. Actores

Los actores que hacen parte del engranaje de las actividades de la empresa son los siguientes:

10.4.1. Administradores Y Agentes

Éstos roles son asignados entre los empleados de la empresa por parte del personal administrativo y de sistemas. Los agentes serán los recepcionistas y registradores de los abonos y

datos que les ingresen los pacientes. Por otro lado, los administradores son pertenecientes al personal autorizado por la empresa para hacerse cargo de los agentes. Controlan el registro de los agentes en base de datos para que puedan acceder a la aplicación.

10.4.2. Pacientes

Los pacientes pueden suscribirse a uno o más tratamientos ofrecidos por la empresa y pueden presentar gastos extracurriculares durante el tiempo que llevan en sus tratamientos. Cada paciente es guiado por un profesional y puede ser remitido por otro profesional para una mejor atención o no.

10.4.3. Profesionales

Los profesionales se encargan de atender a uno o más pacientes suscritos a la empresa y cada profesional está sujeto a un laboratorio que administra sus utensilios de trabajo.

10.4.4. Laboratorios

La información de los laboratorios sólo puede ser ingresada por los administradores. En cambio, los tratamientos son registrados por los agentes y administradores. El registro de los laboratorios concierne a su NIT y nombre.

10.4.5. Tratamientos

Los tratamientos poseen atributos como su código, nombre, costo, descripción y el paciente que está suscrito a él.

10.5. Requerimientos Funcionales

Como software especializado, esta aplicación cumple con los requerimientos propuestos por la empresa (cliente) de la siguiente manera:

10.5.1. Comportamiento.

La aplicación muestra espacios dedicados a cada tipo de transacción que se realice durante su funcionamiento; en cuanto a inserción, actualización, eliminación y consulta de datos consecuentes a los actores (administradores, usuarios agentes, pacientes, profesionales) y al seguimiento de los tratamientos odontológicos ofrecidos por la empresa.

10.5.2. Restricciones.

Cada empleado de la empresa tiene su entorno adaptado al rol que tenga asignado en el sistema. Cada rol posee su propio acceso a informaciones y procedimientos de forma independiente, conforme a las exigencias establecidas por la empresa para prestar sus servicios de la manera más adecuada posible.

10.5.3. Calidad esperada.

Al ingresar las credenciales de usuario, la estabilidad y operatividad de la aplicación es iniciada con la conexión que se realiza con el servidor de base de datos, logrando así que siempre esté disponible para procesar todas las tareas solicitadas por el paciente sin inconvenientes.

10.6. Diagrama General

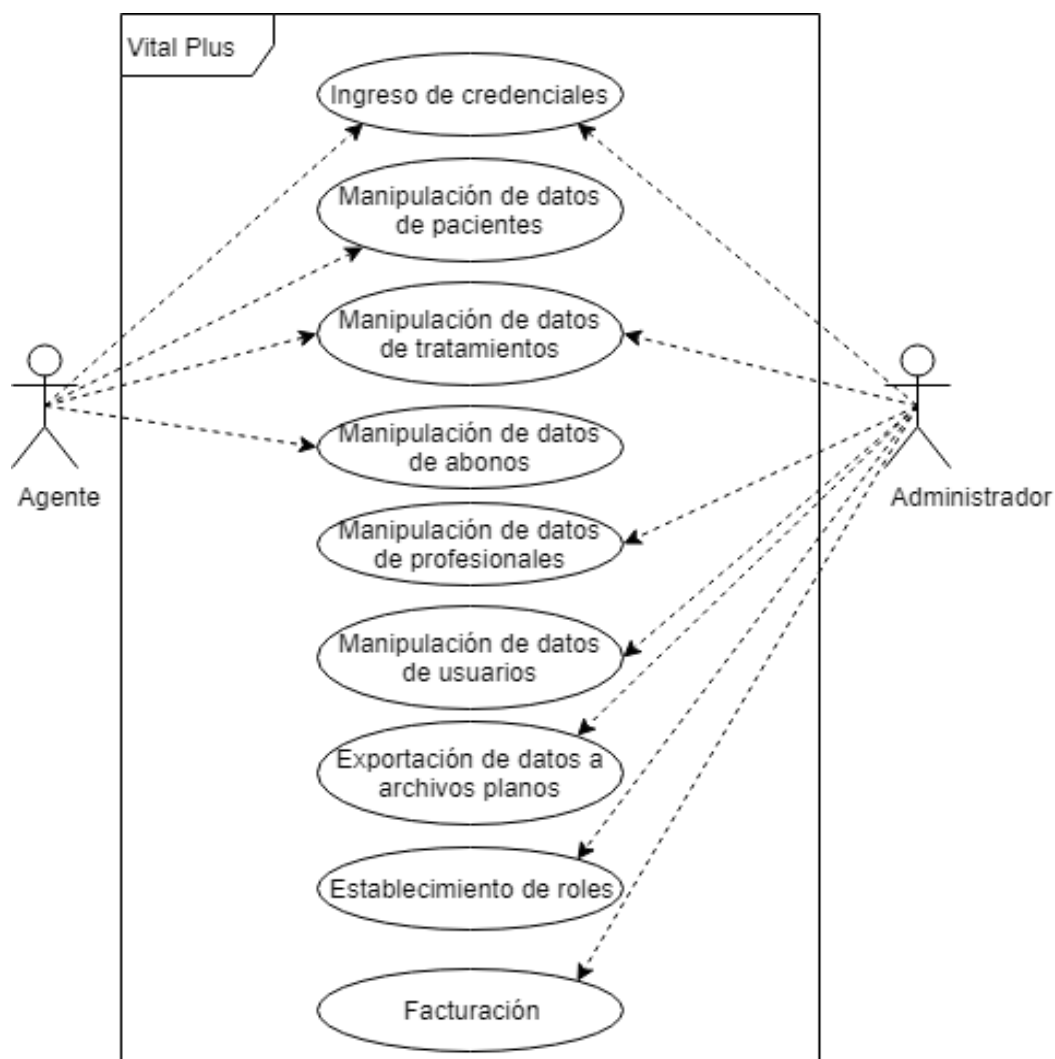


Imagen 6. Diagrama General

10.7. Lista De Requerimientos Funcionales (Específicos)

10.7.1. Datos.

Para acceder a la aplicación, cada empleado debe ingresar sus credenciales (nombre de usuario y contraseña) que serán consultados en base de datos para validación.

Si el usuario es un agente, éste tiene la posibilidad de ingresar información personal relacionada con los pacientes para ser insertada a base de datos. Luego, estos datos son manipulados al momento de ingresar la información correspondiente al registro de abonos de los pacientes a los tratamientos o gastos extra que se presenten durante su estadía en el servicio odontológico.

Por otra parte, el usuario tipo administrador puede ingresar datos consecuentes a usuarios que manejarán la aplicación, a laboratorios aliados a la empresa, profesionales de medicina oral activos y a tratamientos que se realizan a los pacientes suscritos. Los datos consecuentes a los pacientes sólo son consultados y exportados.

10.7.2. Operaciones y flujos.

Los agentes pueden acceder a la aplicación a partir de las credenciales otorgadas por los administradores. Una vez los usuarios agentes ingresan sus credenciales, pueden añadir los datos personales de los pacientes a la base de datos.

Éstos pueden consultar la información de los pacientes (datos personales y tratamiento suscrito) al tiempo que registran sus abonos a tratamientos odontológicos, cumpliendo con los requerimientos de ingreso de información de los abonos en la base de datos desde la misma ventana.

Adicionalmente, pueden consultar el monto económico almacenado por concepto de abonos durante el día en curso o un día en específico. También pueden realizar el mismo tipo de consulta usando la información de los profesionales encargados y tratamientos registrados como parámetros.

Con los abonos registrados por parte de los agentes, los administradores pueden realizar consultas en base de datos para generar y exportar la facturación de dichos abonos a un reporte virtual que puede ser guardado en cualquier formato ofimático a partir de diversos parámetros de consulta.

Además, los administradores también pueden tener acceso al inicio y fin de sesión trabajada por los agentes en una fecha determinada y exportar esa información a reportes virtuales.

10.7.3. Reportes.

La aplicación posee un complemento que le permite exportar la información almacenada en base de datos a un entorno de reportes detallados que se utilizan para entregar facturas a los pacientes cuando se requiera y el registro de sesiones de trabajo de agentes. Dicho entorno complementario de Java es llamado *iReport*, el cual suministra facilidades de navegación y almacenamiento que ayudan a una mejor manipulación de los datos que consulte el usuario para prestar un buen servicio a los pacientes y tener un óptimo control de las actividades laborales de los agentes y los movimientos económicos de los pacientes durante el periodo de sus tratamientos.

10.7.4. Regulaciones.

La organización cuenta con un servidor de base de datos que se conecta con los puntos de ingreso y consulta de datos (computadores de agentes) y también carga la base de datos que manejará las transacciones de la empresa.

Los complementos de Java son instalados en los computadores de los agentes para que la aplicación pueda arrancar; en caso de no poseer los complementos, la aplicación es capaz de alertar sobre su ausencia y, a su vez, realizar su instalación correspondiente.

10.8. Diagramas De Secuencia

10.8.1. Rol Agente

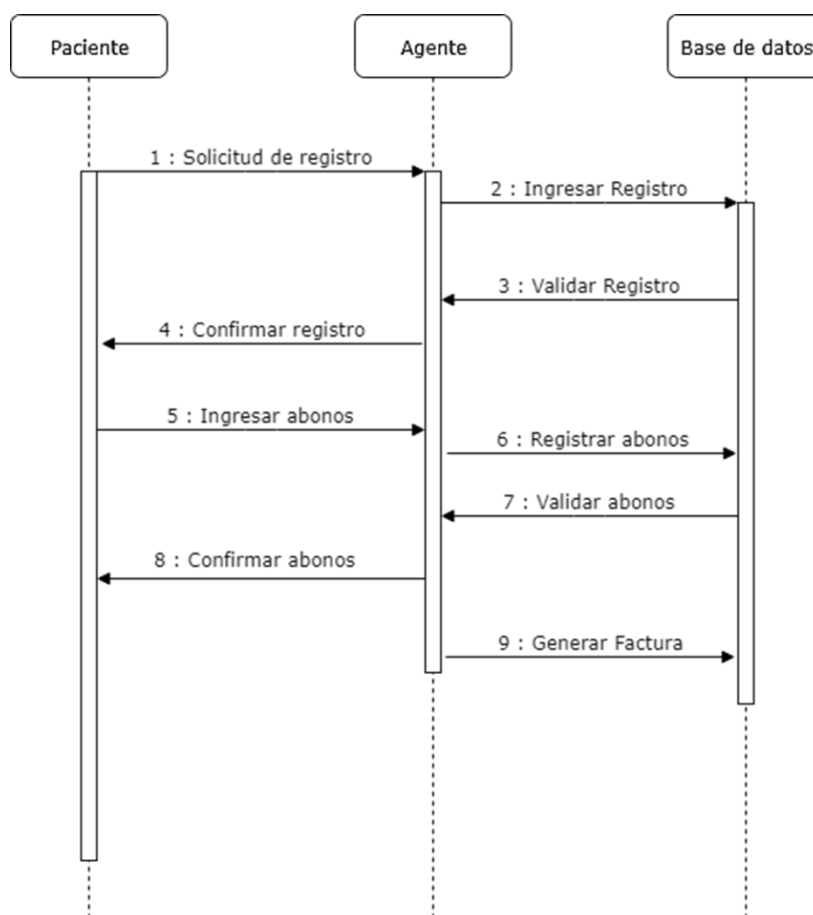


Imagen 7. Diagrama de secuencia - Agente

10.8.2. Rol Administrador

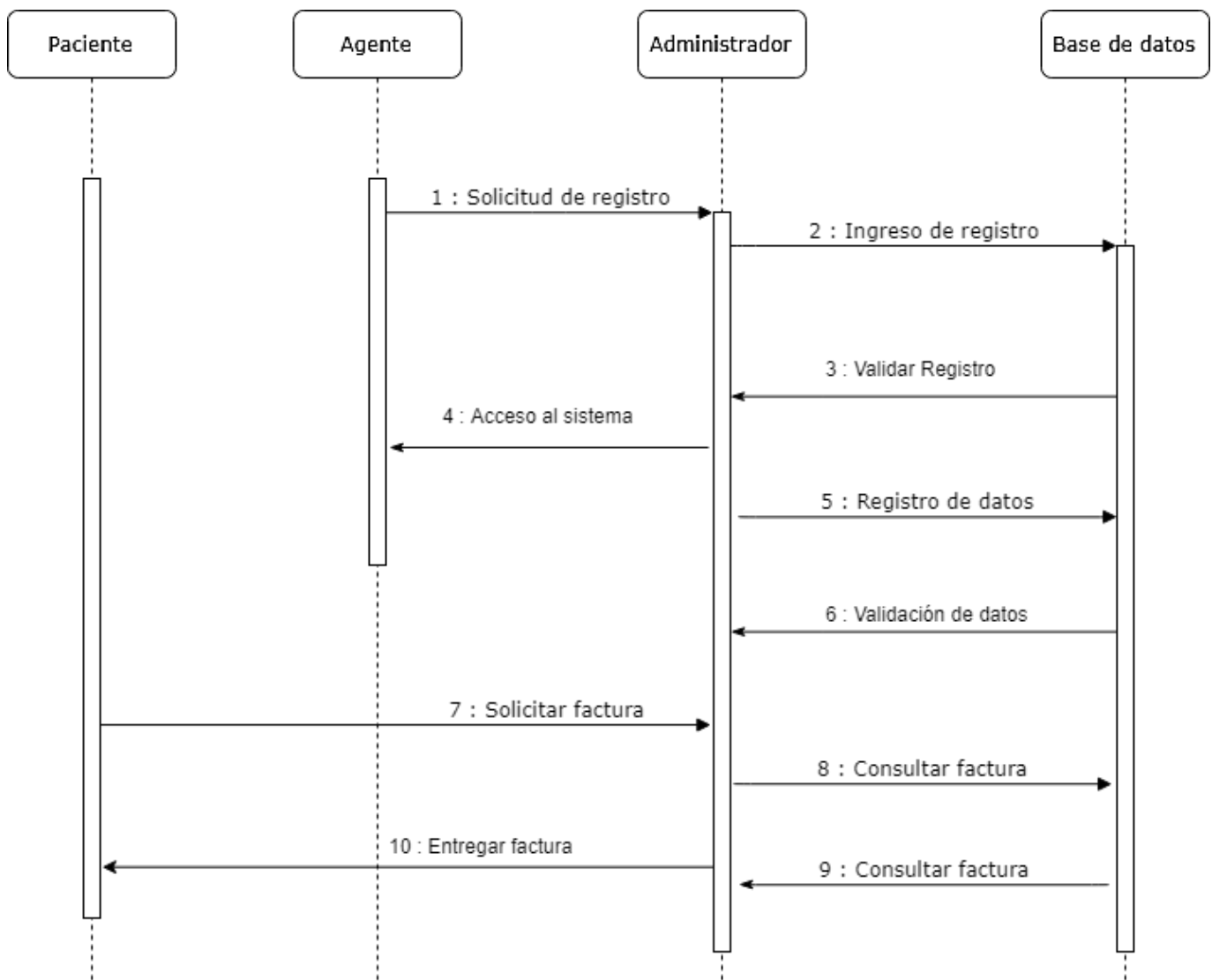


Imagen 8. Diagrama de secuencia - Administrador

10.9. Diagrama De Actividades

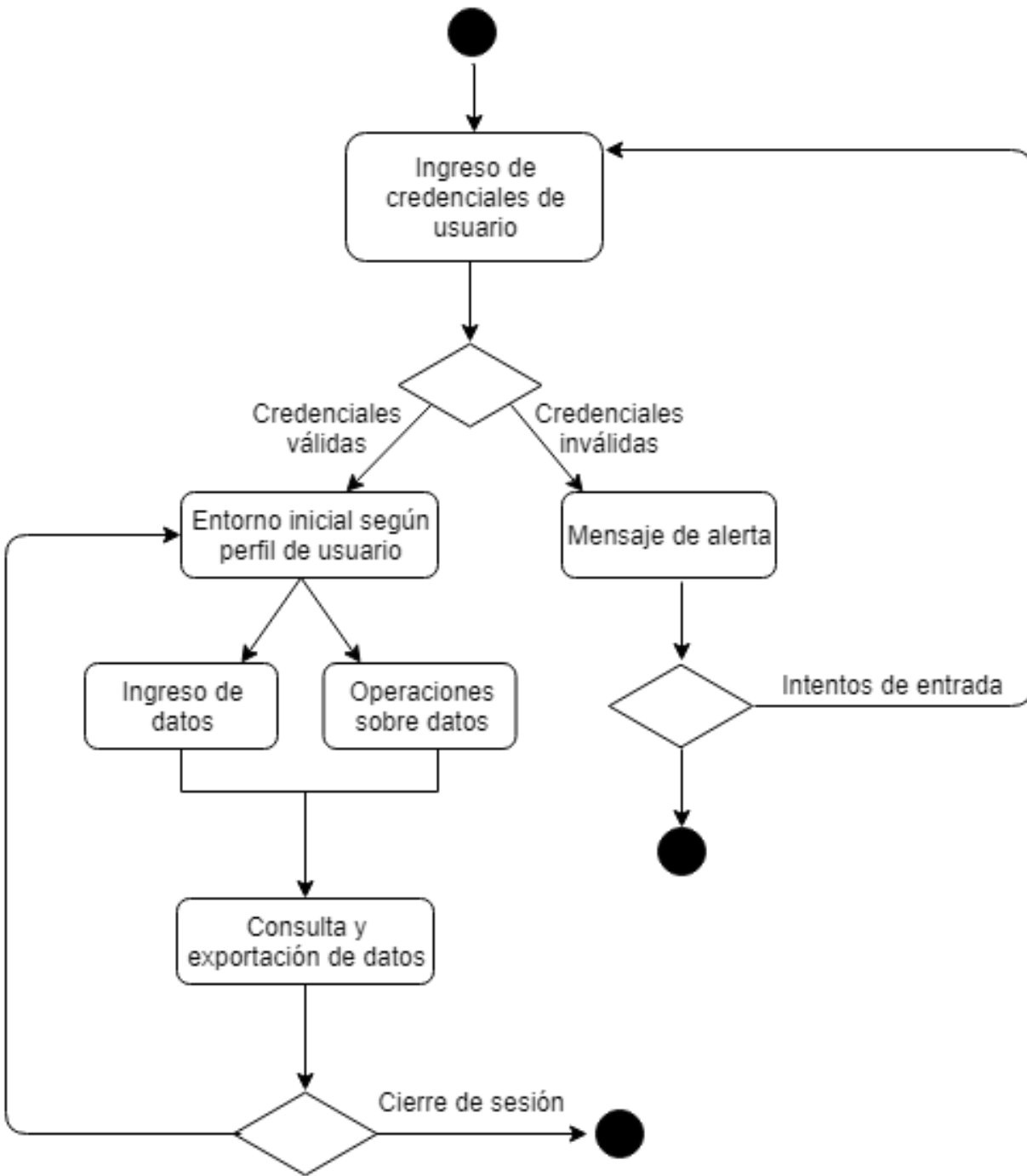


Imagen 9. Diagrama de actividades

11. Capítulo II

11.1. Diseño De Base De Datos

El montaje de la base de datos debe cumplir con el andamiaje que siguen las labores cotidianas de la empresa *Vital Plus* para que se pueda manipular de manera ideal los datos correspondientes a los actores y procesos que se presentan allí. Cada uno de ellos debe estar registrado con la totalidad de sus atributos en los campos de la base de datos.

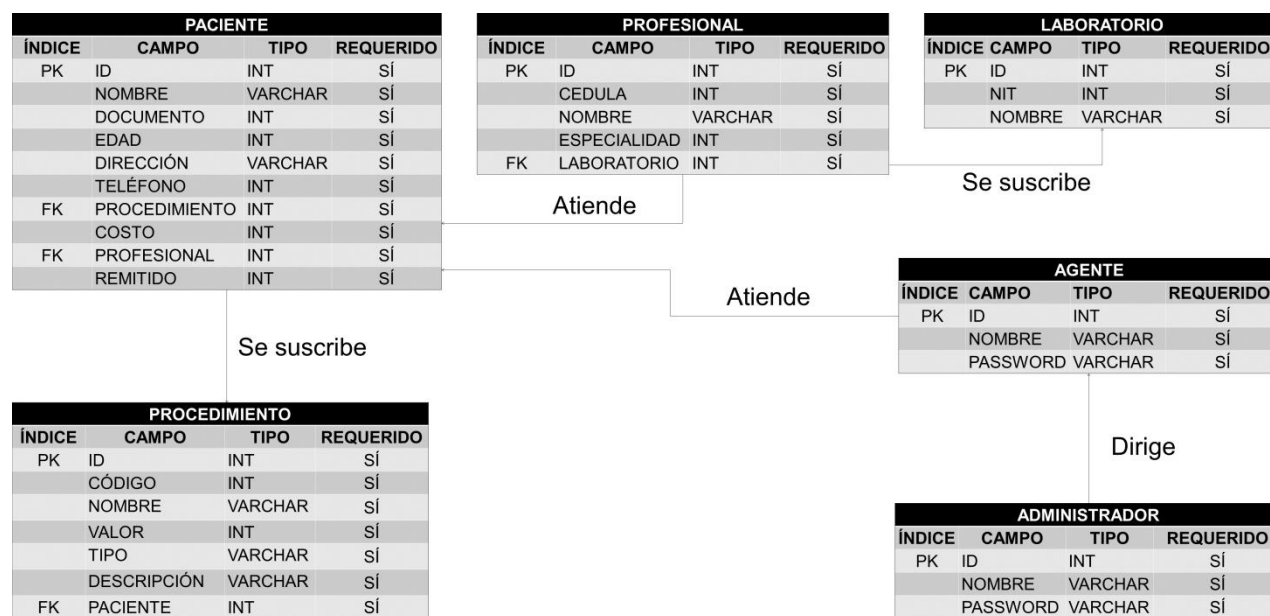


Imagen 10. Diseño de base de datos

11.2. Diccionario De Datos

11.2.1. Paciente: Información concerniente a los pacientes suscritos a la empresa.

Campo	Tamaño	Tipo de dato	Descripción
ID	10	Numérico	Identificación interna del paciente en base de datos
Nombre	30	Caracteres	Nombre del paciente
Documento	30	Numérico	Número de documento de identidad del paciente
Teléfono	30	Numérico	Teléfono del paciente
Dirección	50	Caracteres	Dirección del paciente
Edad	3	Numérico	Edad del paciente
Tratamiento	30	Numérico	Tratamiento al que suscribe el paciente desde su ingreso
Costo	20	Numérico	Costo del tratamiento al que se suscribe el paciente
Profesional	30	Numérico	Número de documento de identidad del profesional encargado del paciente
Remitido	30	Numérico	Número de documento de identidad del profesional que remite al paciente
Estado	1	Numérico	Código que identifica el estado del paciente en el sistema.

11.2.2. Profesional: Información relacionada a los profesionales de la salud de la empresa.

Campo	Tamaño	Tipo de dato	Descripción
ID	10	Numérico	Identificación interna del profesional en base de datos
Nombre	30	Caracteres	Nombre del profesional
Cedula	10	Numérico	Número de documento de identidad del profesional
Especialidad	30	Caracteres	Especialidad del profesional
Laboratorio	30	Numérico	Laboratorio al que el profesional se encuentra suscrito
Estado	1	Numérico	Código que identifica el estado del profesional en el sistema.

11.2.3. Laboratorio: Información de los laboratorios aliados a la empresa.

Campo	Tamaño	Tipo de dato	Descripción
ID	10	Numérico	Identificación interna del laboratorio en base de datos
NIT	30	Numérico	Código NIT del laboratorio
Nombre	30	Caracteres	Nombre del laboratorio
Estado	1	Numérico	Código que identifica el estado del laboratorio en el sistema.

11.2.4. Procedimiento (tratamiento): Información de los procedimientos odontológicos ofrecidos por la empresa a sus pacientes.

Campo	Tamaño	Tipo de dato	Descripción
ID	10	Numérico	Identificación interna del tratamiento en base de datos
Código	30	Numérico	Código del tratamiento
Nombre	30	Caracteres	Nombre del tratamiento
Valor	20	Numérico	Valor o costo del tratamiento
Tipo	20	Caracteres	Tipo de procedimiento (Tratamiento o Gasto Extra)
Descripción	1500	Caracteres	Descripción del procedimiento
Paciente	30	Numérico	Identificación interna del paciente en base de datos
Estado	1	Numérico	Código que identifica el estado del tratamiento en el sistema.

11.2.5. Usuarios: Información asociada a los usuarios que ejecutan la aplicación.

Campo	Tamaño	Tipo de dato	Descripción
ID	10	Numérico	Identificación interna del usuario en base de datos
Nombre	30	Caracteres	Nombre del usuario
Password	300	Caracteres	Contraseña del usuario
Tipo usuario	30	Caracteres	Rol del usuario
Estado	1	Numérico	Código que identifica el estado del usuario en el sistema.

11.2.6. Abonos: Información concerniente a los abonos que realizan los pacientes en la empresa.

Campo	Tamaño	Tipo de dato	Descripción
Código	30	Numérico	Identificación interna del abono en base de datos
Valor	20	Numérico	Valor del abono consignado por el paciente
Tipo	20	Caracteres	Tipo de abono consignado
Método	30	Caracteres	Método de pago utilizado para el pago del abono (Efectivo, Tarjeta de crédito o Tarjeta débito)
Fecha		Fecha	Fecha y hora del registro del abono
Procedimiento	30	Numérico	Identificación interna del tratamiento en base de datos
Cédula de paciente	30	Numérico	Identificación interna del paciente en base de datos
Código Profesional	30	Numérico	Identificación interna del profesional en base de datos
Estado	1	Numérico	Código que identifica el estado del abono en el sistema.

11.2.7. Factura: Información relacionada con las facturas que se generan por los pagos de los abonos realizados por los pacientes.

Campo	Tamaño	Tipo de dato	Descripción
Cédula de paciente	30	Numérico	Identificación interna del paciente en base de datos
Cédula de profesional	30	Numérico	Identificación interna del profesional en base de datos
Medio de pago	30	Caracteres	Método de pago utilizado para el pago del abono (Efectivo, Tarjeta de crédito o Tarjeta débito)
Código de procedimiento	30	Caracteres	Identificación interna del tratamiento en base de datos
Código	30	Caracteres	Código de la factura (Comienza con las letras “VP”)
Fecha		Fecha	Fecha y hora del registro del abono
Valor	30	Numérico	Valor del abono consignado por el paciente
Consecutivo	30	Numérico	Identificación interna de la factura en base de datos
Estado	1	Numérico	Código que identifica el estado de la factura en el sistema.

11.2.8. Ingreso: Registro de inicios de sesión de los usuarios en la aplicación.

Campo	Tamaño	Tipo de dato	Descripción
ID	10	Numérico	Identificación interna del usuario en base de datos
Nombre	30	Caracteres	Nombre del usuario
Tipo	300	Caracteres	Rol del usuario
Ingreso	30	Caracteres	Fecha y hora de inicio de sesión

11.2.9. Salida: Registro de cierres de sesión de los usuarios en la aplicación.

Campo	Tamaño	Tipo de dato	Descripción
ID	10	Numérico	Identificación interna del usuario en base de datos
Nombre	30	Caracteres	Nombre del usuario
Tipo	300	Caracteres	Rol del usuario
Ingreso	30	Caracteres	Fecha y hora de cierre de sesión

11.2.10. Estado: Condición en la que se encuentra un actor o proceso del sistema. Tiene dos posibles valores: 1 para “Activo” y 0 para “Inactivo”

Campo	Tamaño	Tipo de dato	Descripción
ID	10	Numérico	Identificación del estado
Nombre	30	Caracteres	Nombre del estado

11.3. Diagramas De Diseño

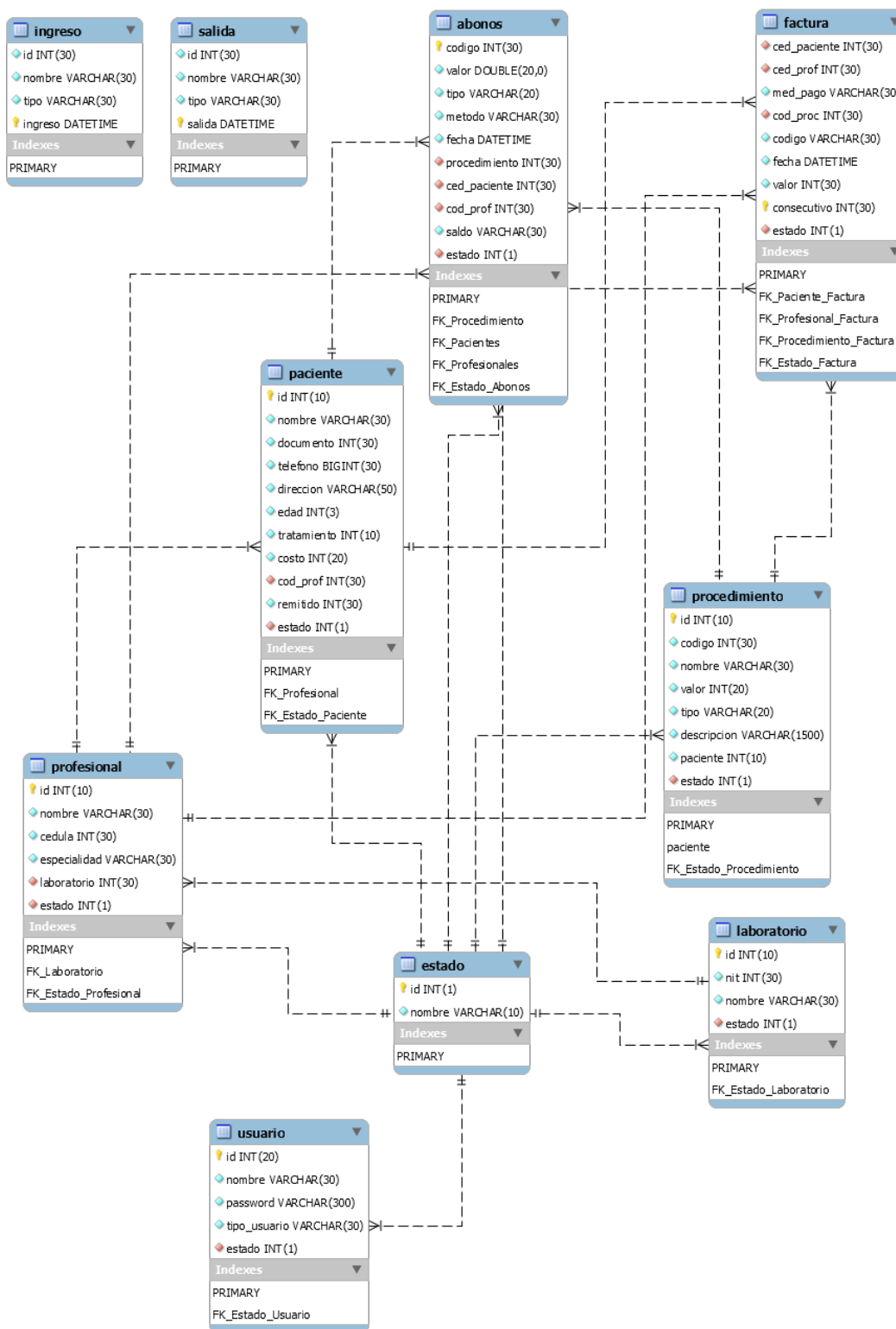


Imagen 11. Diagrama Entidad-Relación

11.4. Diagrama De Clases

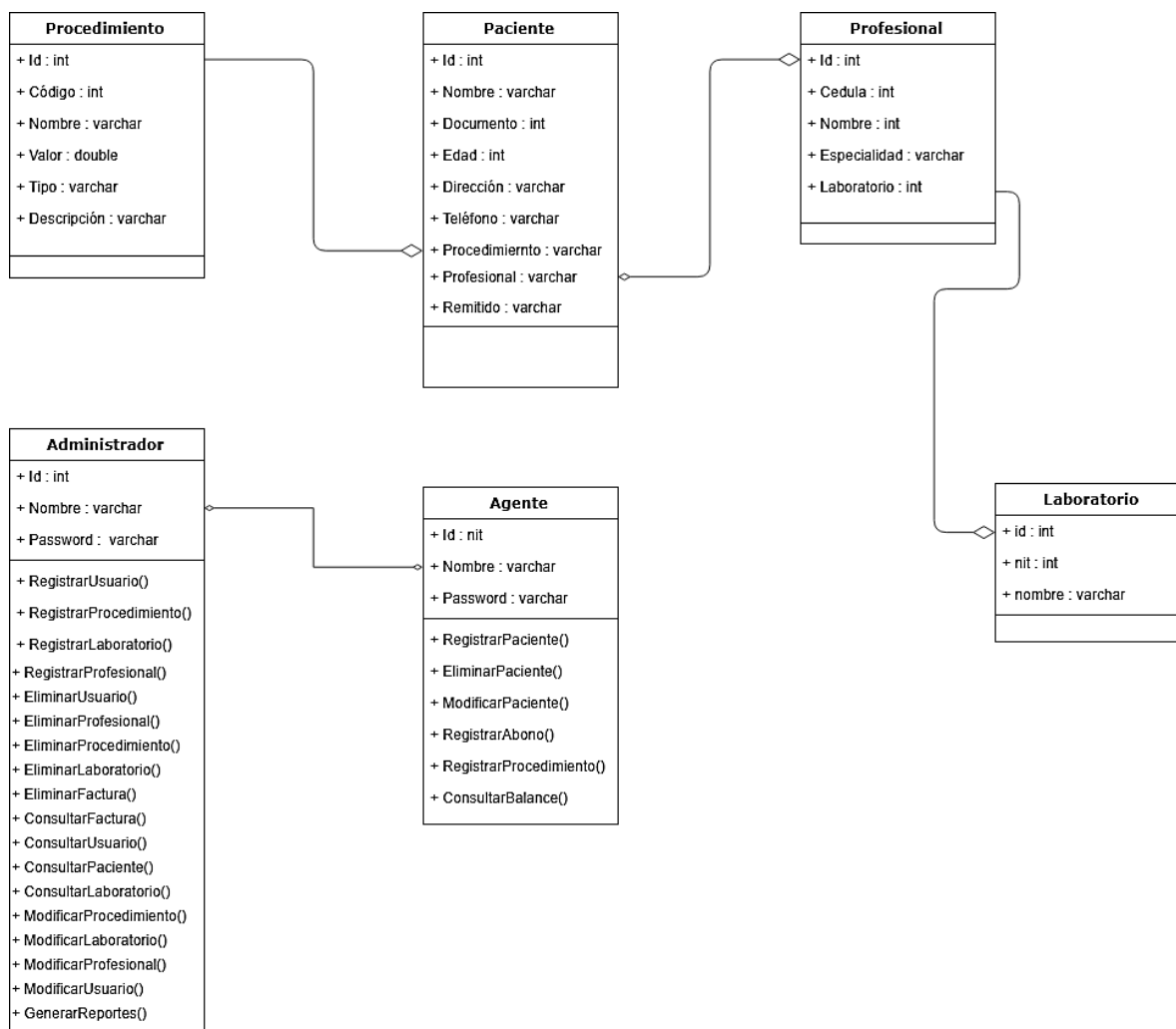


Imagen 12. Diagrama de clases

11.5. Diseño De Interfaz Gráfica

Para tener una guía de cómo podrían diseñarse las ventanas de la aplicación en donde se ejecutan dichas labores, se utilizó *Balsamiq* como herramienta de diseño. Esta es una herramienta disponible en su propia página web, la cual tiene la posibilidad de operar como aplicativo web, de escritorio o como complemento manejado desde Google Drive. Se caracteriza

por su enmarcación que ayuda a trabajar más rápido y con más inteligencia. Reproduce la experiencia de dibujar en una pizarra, pero usando una computadora. (balsamiq.com, 2018)

12. Capítulo III

12.1. Prototipo

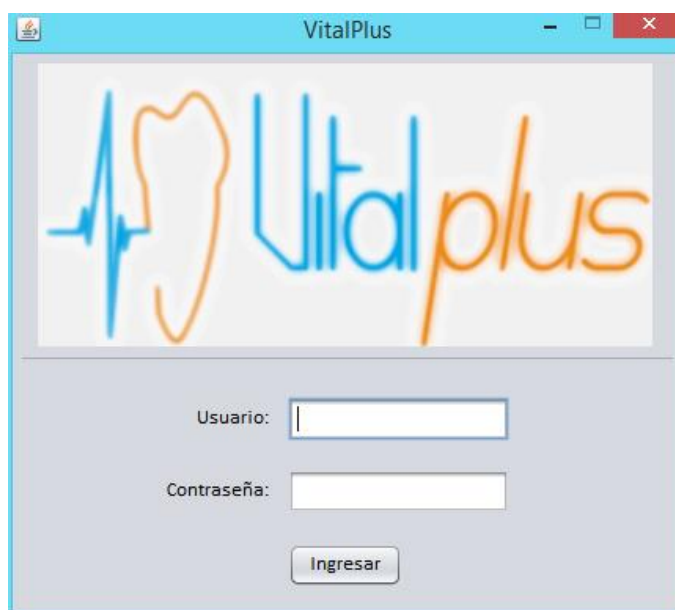


Imagen 14. Prototipo de pantalla para inicio de sesión

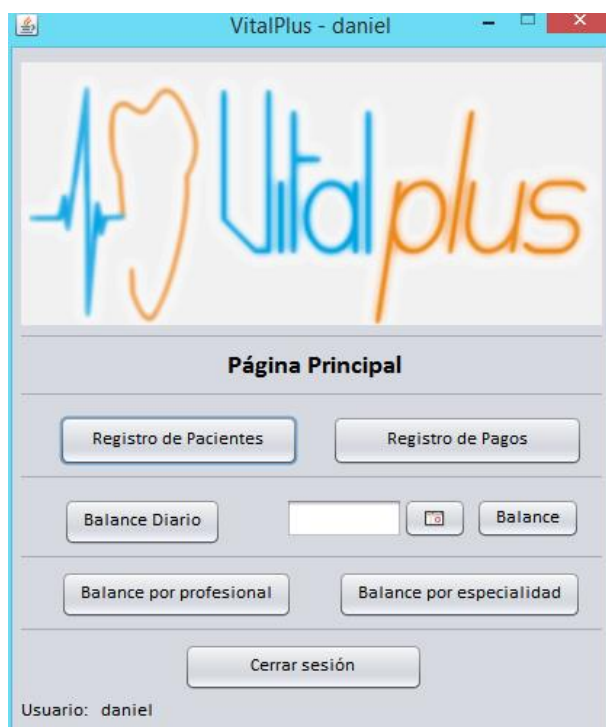


Imagen 15. Prototipo de pantalla principal de espacio para agente

Abonos

Documento: Nombre: Edad: Tratamiento:

Código de Observ.: Nombre: Valor de obs.: Tipo: Descripción: Descuento (%):

Valor de abono: \$ Tratamiento: Profesional: Medio de pago:

Valor Pendiente: Valor con descuento (Medio de pago):

Abonos registrados

Nombre	Tratamiento	Profesional	Fecha	Factura	Abono	Pendiente

Usuario: daniel

Imagen 16. Prototipo de ventana de abonos

Admin. Usuarios **Admin. Profesionales** **Admin. Pacientes** **Admin. Laboratorios** **Admin. Tratamientos**

Facturación...

Registros...

Cerrar sesión

Usuario: admin

Imagen 17. Prototipo de pantalla principal de espacio para administrador

Usuario: Contraseña:

ID:

Tipo:

Lista de usuarios

Digite código:

ID	Nombre de Usuario	Tipo de Usuario
87005	Carlos	Administrador
87004	Fabio	Agente
87005	345	Agente
87006	887	Agente
87007	123	Agente
1045879022	Lucas	Agente

Usuario: admin

Imagen 18. Prototipo de ventana para registro de usuarios

Nombre: Código: Costo:

Documento de paciente: Tipo:

Descripción:

Lista de Tratamientos

Digite código del tratamiento:

Código	Nombre	Costo	Descripción
110	General	500000	Servicio de procedimie...
155	Nuevo Tratamiento O	300000	Nuevo tratamiento O
156	Nuevo 4	1500000	Nuevo 4
188	DDDD	2000	DDDD
225	Ortodoncia	3000000	Procedimiento para la ...
228	Ortodoncia	2300000	Ortodoncia al 20% desd...

Usuario: admin

Imagen 19. Prototipo de ventana para registro de tratamientos



Imagen 20. Prototipo de ventana para facturación

Más adelante, se encuentra el compilado de diseño de pantallas con el que se inició el proceso de creación, diseño y compilación de cada una de las ventanas que aparecen durante funcionamiento de la aplicación (ver anexo C) utilizando productos de terceros y el póster de este proyecto que lo explica de forma resumida (ver anexo D).

12.2. Codificación Y Pruebas

Archivo Conexión.java (Conexión a base de datos)

```
package odontologia;
```

```
import java.sql.*;
```

```
/**
```

```
*
```

```
* @author Daniel
```

```
*/
```

```
public class Conexion {
```

```
Connection conexion = null;

Statement sentencia;

ResultSet datos;

PreparedStatement prepararSentencias;

/**
 *
 * @return
 * @throws Exception
 * @throws ClassNotFoundException
 */
public Connection conexion() throws Exception, ClassNotFoundException
{
    try {
        Class.forName("com.mysql.jdbc.Driver");
        String url = "jdbc:mysql://localhost/vital_plus";
        //conexion = DriverManager.getConnection(url, "administrador", "josmifube");
        conexion = DriverManager.getConnection(url, "root", "");
        sentencia = conexion.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
        ResultSet.CONCUR_READ_ONLY);
    } catch (ClassCastException | SQLException ex) {
        throw ex;
    }
    return conexion;
}
}
```

Los demás archivos de código fuente de la aplicación se anexan a este documento para su correspondiente revisión y el manual instructivo de la aplicación, incluyendo la explicación de su proceso de compilación (ver anexo B).

12.3. Estándares De Codificación

Para la elaboración de esta aplicación, se siguió el estándar de programación que Java permite desarrollar para preponderar la interacción del usuario con el sistema: la programación basada en eventos.

Este estándar se basa en construir un diseño de interfaces gráficas de usuarios (GUI: *Graphics User Interface*) que toma en cuenta toda la variedad de posibles interacciones del usuario con el sistema, a diferencia de los programas de consolas, que siguen un flujo secuencial en el que típicamente se tienen 3 ciclos:

Entrada → Procesamiento → Salida

En lugar de un único flujo de entrada de datos por consola, las GUIs permiten muchas más acciones del usuario. Por ejemplo, es posible presionar botones gráficos, escribir texto en un campo de texto, o mover alguna barra de desplazamiento.

Una forma de manejar todo tipo de posibles interacciones de usuarios es el uso de interrupciones. De esta manera la CPU no pierde tiempo “mirando” los posibles eventos de usuarios, sino simplemente responde al evento y reanuda su procesamiento normal (otras tareas).

Comúnmente, lenguajes de programación no dan acceso directo a eventos asincrónicos. Lenguajes como Java permiten definir y manejar interrupciones o eventos por software. La API

de Java permite a los programadores crear clases de objetos, llamados *listeners*, que responden a interrupciones causadas por la GUI.

La API de Java tiene interfaces que deben ser implementadas por las clases que manejan los eventos posibles. Los métodos de la interfaz (“equivalen a las rutinas de servicio de interrupción”) son llamados cuando un evento específico ocurre.

Los *listeners* son instancias de las clases definidas por el usuario y que implementan los métodos definidos por la Interfaz (estos métodos equivalen a “las rutinas de atención de Interrupción”). Las componentes gráficas y los datos del evento son objetos de las clases de Java.

12.4. Diagrama De Despliegue

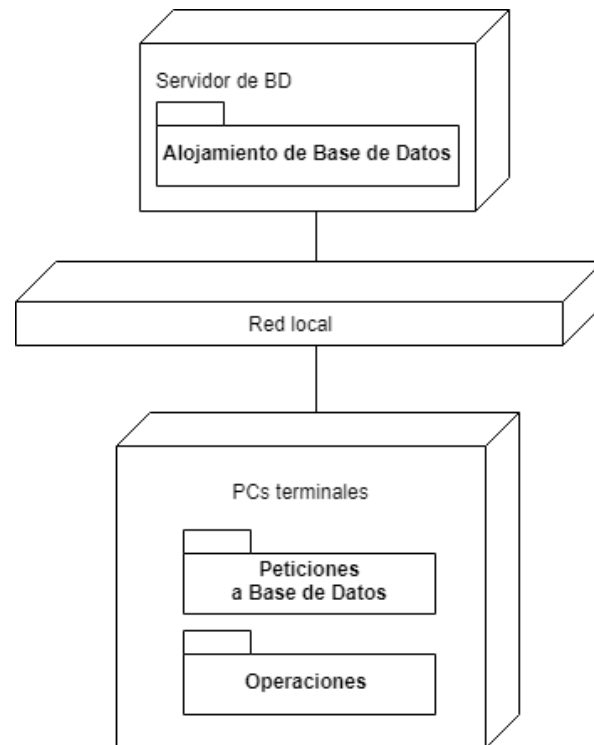


Imagen 21. Diagrama de despliegue

12.5. Diagrama De Componentes

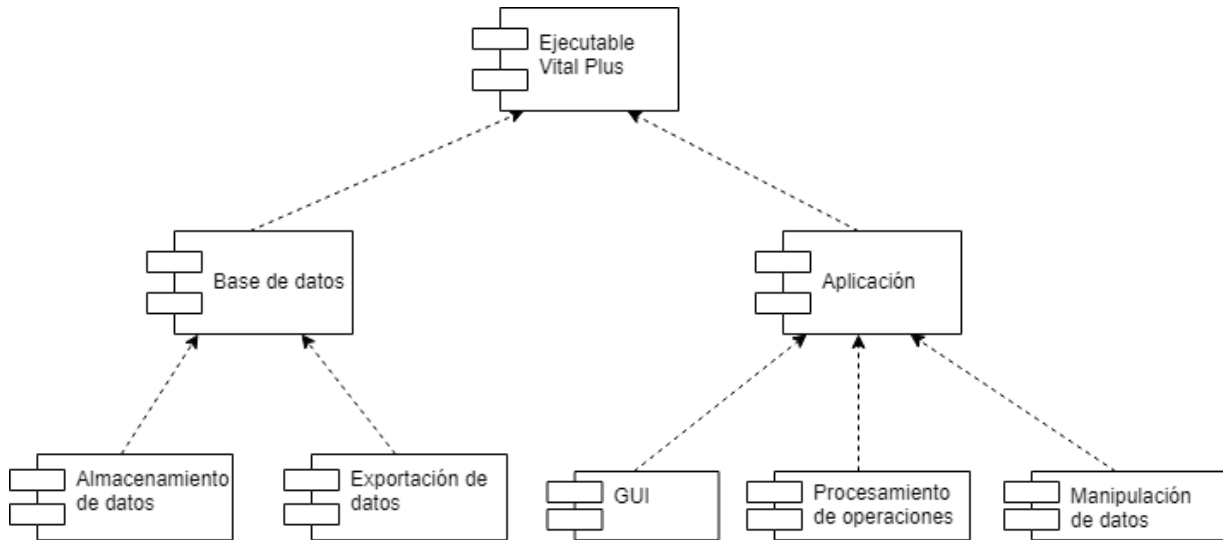


Imagen 22. Diagrama de componentes

13. Análisis De Resultados

13.1. Identificación De Requerimientos

La identificación de los requerimientos de la aplicación permitió tener una visión general de los procesos que se necesitaban desarrollar dependiendo del tipo de usuario. Los administradores pueden exportar cualquier información almacenada en la base de datos a un reporte que puede ser manejado también en cualquier formato de visualización. Se gestiona de manera correcta la información de los usuarios, permitiéndoles actualizar datos sensibles (como contraseñas). La información correspondiente a los tratamientos o procedimientos, los profesionales y los laboratorios cumplen con los requisitos de manipulación de datos (inserción, modificación, eliminación y consulta), mientras que la información de los pacientes suscritos a la empresa es exportada para que el administrador pueda tener más detalles de ellos. El entorno de facturación permite hacer lectura y exportación de sus correspondientes datos, así como también la posibilidad de eliminar facturas en caso de que el administrador lo crea conveniente.

Para los usuarios agentes la página principal está compuesta de dos espacios: el primero comprende el registro de información personal de los pacientes y el registro de sus abonos durante sus tratamientos, cada uno de ellos cuenta con la posibilidad de realizar acciones de registro, borrado y consulta de datos de los pacientes y de registro y consulta de datos correspondiente a los abonos. El segundo se compone de entornos de consulta de datos para proporcionar información esencial de forma inmediata a los administradores, en lo que compete al balance económico que puede haber en un periodo de tiempo o por criterio (por profesional o por especialidad de profesional).

13.2. Definición De Paradigma De Programación.

La utilización de la programación por capas permitió que se construyera la aplicación de una manera muy eficaz, puesto que las operaciones sobre los datos almacenados fueron reutilizadas dependiendo de las necesidades que tenía cada entorno de usuario. Cada nivel de trabajo fue desarrollado de manera independiente y, a su vez, fue aislado del otro para que el mantenimiento del código fuera más diligente y lograr que la aplicación sea robusta en su compilación.

13.3. Diseño De Medio De Almacenamiento

La elección del sistema de gestión de base de datos permitió mantener la premisa de relacionar los actores y procesos que hacen parte de la empresa, con el fin de que las actividades que se llevan a cabo durante los tratamientos odontológicos puedan ser plasmadas en la base de datos que utiliza la aplicación. Se lograron relacionar a los actores (tablas) entre sí mediante sus llaves primarias, siguiendo las teorías de las bases de datos relacionales que el motor MySQL permite establecer.

13.4. Desarrollo De Prototipo

Se presentó un prototipo de la aplicación de escritorio que puede suplir las necesidades de la empresa *Vital Plus* en su departamento de facturación. Se exponen las utilidades e instructivos que posee la aplicación a los usuarios que la utilizarán, con la expectativa de que ellos se sientan más cómodos para atender a los pacientes y, a su vez, la experiencia de recepción de servicios de facturación y salud oral sea mucho más satisfactoria en los pacientes suscritos.

14. Recomendaciones

Para la correcta ejecución de la aplicación desarrollada para la empresa *Vital Plus*, se realizan las siguientes recomendaciones:

- Mantener activo el servidor de base de datos durante las horas de operaciones de la empresa, con el fin de que la aplicación pueda ejecutarse de la manera óptima como fue desarrollada a través de la conexión que establece desde su inicio.
- Asegurar la vigencia de versiones de Java actualizada, puesto que la aplicación requiere de este complemento básico para ser compilada y así trabajar en cualquier computador personal.
- Se recomienda realizar reuniones inductivas programadas para orientar a los trabajadores acerca de las funcionalidades que posee la aplicación desarrollada para la optimización de procesos internos de facturación de la empresa.
- Sostener la comunicación entre el personal de sistemas y los ingenieros desarrolladores externos para que se brinde soporte de manera oportuna en las posibles correcciones o actualizaciones que la empresa necesite en el futuro.

15. Conclusiones

Con esta aplicación desarrollada en lenguaje Java se pudo aplicar los conocimientos adquiridos durante la carrera de Ingeniería de Sistemas en la Universidad Nacional Abierta y a Distancia (UNAD).

La identificación de cada uno de los requerimientos que la empresa *Vital Plus* planteaba conllevó a un completo entendimiento de su estructura organizacional y operativa para que el desarrollo de la aplicación solicitada contribuyera a que la prestación de servicios de la empresa a sus pacientes fuese más óptimo y satisfactorio.

El empleo de la programación por capas como paradigma de programación facilitó el desarrollo de la aplicación, puesto que este estilo permite tener un mantenimiento más inmediato en caso de que se requiera aplicar algún cambio, gracias a su metodología por niveles. Además, esta arquitectura ahorra tiempo en la generación de código, haciendo que la aplicación sea mucho más escalable y robusta.

La elección de XAMPP fue de gran importancia para que el desarrollo y diseño del prototipo de la aplicación fuera más acorde a lo requerido, ya que la conexión que realiza la aplicación por medio de los servicios de MySQL y Apache para ponerse en funcionamiento tiene un grado alto de garantía y logra que el sistema sea muy poco propenso a pérdida de datos esenciales que se necesitan durante las transacciones que se realizan en la empresa.

La inclusión de las librerías *iReport* y *JasperReports* como complementos de la aplicación permitió observar las ventajas que ofrecen para construir reportes de información almacenada en base de datos y exportarla a archivos ofimáticos en varios formatos (PDF,

HTML, XLS, DOC o CSV) para que el usuario pueda realizar cualquier tipo de operaciones sobre los datos contenidos allí de una manera más sencilla y rápida.

El diseño del prototipo de la aplicación conduce a que la empresa pueda tener una noción de lo que se desarrolló en este proyecto para poder suplir las necesidades que tiene en su departamento de facturación, todo con la meta de mejorar la experiencia en la contabilidad de los servicios de salud oral que puedan solicitar sus pacientes.

16. Bibliografía

- Arias, A. (2014). *Bases de datos con MySQL*. Lexington, KY: IT Campus Academy.
- Benayas Álamos, A., Marco Bahón, A., & Torrijos Santos, P. (2008). *GESCLIDENT*. Obtenido de Archivo Institucional E-Prints Complutense - E-Prints Complutense: http://eprints.ucm.es/9854/1/Gesclident_2008.pdf
- Benítez, M. (2017). En *Curso de introducción a la administración de base de datos* (pág. 23). IT Campus Academy.
- Camuña Rodríguez, J. (2015). *Lenguajes de definición y modificación de datos SQL (UF1472)*. Madrid, España: IC Editorial.
- Ceballos Sierra, F. (2018). *JAVA. Interfaces gráficas y aplicaciones para Internet*. Madrid, España: Grupo Editorial RA-MA.
- Creus Solé, A. (2005). *Fiabilidad y seguridad: su aplicación en procesos industriales* (2 ed.). Barcelona, España: Marcombo.
- Danciu, T., & Chirita, L. (2007). *The Definitive Guide to JasperReports*. Berkeley, CA, Estados Unidos: Apress.
- Date, C., Ruiz Faudón, S., & López Gamino, F. (2001). *Introducción a los sistemas de bases de datos* (7 ed.). México D. F.: Pearson Educación.
- Documentación web de MDN. (2005). *Detalles del modelo de objetos*. Obtenido de https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Details_of_the_Object_Model.
- Duarte, B. (2017). *ODONTOWEB*. Obtenido de SISTEMA WEB DE GESTIÓN DE PACIENTES ODONTOLÓGICOS: <http://ribuni.uni.edu.ni/1494/1/80742.pdf>
- Durán, F., & Gutiérrez, F. (2007). *Programación orientada a objetos con Java* (1 ed.). Madrid, España: Thomson-Paraninfo.
- Eraso Peña, S. (2014). *Diseño e implementación de un prototipo para el manejo de los datos estadísticos del uso de los recursos electrónicos de la biblioteca*. Proyecto de grado, Universidad de Los Andes, Departamento de Ingeniería de Sistemas y Computación, Bogotá. Obtenido de <https://sistemas.uniandes.edu.co/es/proyectos-grado>
- Fossati, M. (2014). *Todo sobre MySQL: Libro ideal para ingresar en el mundo de la base de datos MySQL* (1 ed.). Natsys.
- Galindo Haro, J., & Camps Riba, J. (2018). *Diseño e implementación de un marco de trabajo (framework) de presentación para aplicaciones JEE*. Obtenido de <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/876/1/00765tfc.pdf>.
- González, A. (2018). *Programación basada en eventos "Event Based Programming": Conceptos*. (Universidad Técnica Federico Santa María, Ed.) Valparaíso, Chile.

- Grupo de Ingeniería del Software y Bases de Datos. (2013). *El ciclo de vida del software*. Sevilla, España: Universidad de Sevilla. Obtenido de <http://www.lsi.us.es/docencia/get.php?id=7550>
- Guzmán, A. (2014). *WebPicture*. Universidad de Los Andes, Cundinamarca. Bogotá: Departamento de Ingeniería de Sistemas y Computación. Obtenido de <https://sistemas.uniandes.edu.co/es/proyectos-grado>
- La Red Martínez, O., David, L., Acosta, J. C., Mata, L. E., Bachmann, N. G., & Vallejos. (2012). *Aprendizaje combinado, aprendizaje electrónico centrado en el alumno y nuevas tecnologías*.
- Melani De La Hoz, A. (2014). *Generación de un campus activo a través de una solución basada en movilidad tipo pervasiva*. Proyecto de grado, Universidad de Los Andes, Departamento de Ingeniería de Sistemas y Computación, Bogotá. Obtenido de <https://sistemas.uniandes.edu.co/es/proyectos-grado>
- Netbeans.org. (2018). *Welcome to NetBeans*. Obtenido de <https://netbeans.org>
- Palacios Villafuerte, D. (2013). *ANÁLISIS DISEÑO E IMPLEMENTACIÓN DE UN PORTAL WEB PARA LA GESTIÓN Y ADMINISTRACIÓN HOSPITALARIA DE LA CLÍNICA DENTAL BARRERA*. Obtenido de REPOSITORIO DE ESCUELA POLITÉCNICA DEL EJÉRCITO: <https://repositorio.espe.edu.ec/bitstream/21000/6815/1/T-ESPE-047272.pdf>
- Pressman, R. (2002). *Ingeniería del software* (7 ed.). México D. F.: McGraw-Hill.
- Ramos Martín, A., & Ramos Martín, M. (2007). *Operaciones con bases de datos ofimáticas y corporativas*. Madrid, España: Paraninfo.
- Ramos Salavert, I., & Lozano Pérez, M. (2000). En *Ingeniería del software y bases de datos* (pág. 78). Cuenca, España: Ediciones de la Universidad de Castilla-La Mancha.
- Torre Llorente, C., Zorrilla Castro, U., Calvarro Nelson, J., Ramos Barros, M., Manteiga Pacios, C., Cortés Hierro, F., & García Mesa, I. (2010). *Guía de arquitectura N-Capas orientada al dominio con .NET 4.0*. Vigo, España: Krasis Press.
- Vivona, I. (2011). En *Java, Domine el lenguaje líder en aplicaciones cliente-servidor* (1 ed., págs. 30-32). Buenos Aires, Argentina: Fox Andina.

MANUAL DE INSTALACIÓN DE IREPORT Y JASPERREPORTS EN NETBEANS



REALIZADO POR: DANIEL DARÍO PERTUZ OROZCO

BARRANQUILLA, 2018

Contenido

Introducción	91
Instalación De iReport	92
Instalación De JasperReports	96

Imágenes

Imagen 1. Escoger opción plugins desde menú de Herramientas.....	92
Imagen 2. Ventana de plugins.....	93
Imagen 3. Escogencia de archivos de plugin <i>iReport</i>	93
Imagen 4. Botón para añadir fuente de datos.....	94
Imagen 5. Añadir fuente de datos	94
Imagen 6. Estableciendo tipo de conexión de base de datos	95
Imagen 7. Ingresando credenciales de base de datos	95
Imagen 8. Escogiendo opción de librerías en menú de Herramientas	96
Imagen 9. Añadir nueva librería	97
Imagen 10. Establecer nombre de librería	97
Imagen 11. Añadir archivos de librería <i>JasperReports</i>	98
Imagen 12. Escogencia de archivos de librería <i>JasperReports</i>	98
Imagen 13. Estableciendo posiciones de archivos de librería <i>JasperReports</i>	99
Imagen 14. Confirmación de librería <i>JasperReports</i> añadida	99

Introducción

Por medio de este manual se describe la secuencia de pasos que se deben seguir para aprovechar las utilidades que ofrecen las librerías *iReport* y *JasperReport* para generar informes especiales e informativos.

Su programación en Java permite que se puedan diseñar reportes dinámicos y ayuda a crear documentos de tipo páginas, preparados para ser impresos en diversos formatos y en una forma simple y flexible.

El entorno de desarrollo integrado utilizado para incluir estas librerías de reportes es *NetBeans*, ideal para la construcción moderna de programas en Java, por su simplicidad y dinamismo para diseñar interfaces gráficas y establecer operaciones acordes a las necesidades de un software.

A través de este manual se podrá observar el empleo de un asistente de informes que permite realizar un bosquejo de cómo se visualizaría el reporte cuando el usuario lo requiera en tiempo de ejecución de la aplicación Java.

El asistente de informes de *iReport* puede generar informes a partir de una plantilla, y viene con algunas plantillas de informes básicos (o puede crear las suyas propias). El asistente de informes se inicializa a partir de una consulta SQL. Después de seleccionar columnas del resultado de la consulta, se puede personalizar el informe y ejecutarlo utilizando datos en tiempo real.

Una vez que se crea un informe, el diseño se almacena en un archivo XML de *JasperReports*. Se puede ejecutar el informe en *iReport* utilizando un origen de datos JDBC y guardar el resultado en un archivo PDF, HTML, XLS, DOC o CSV. También se puede integrar el informe en las aplicaciones utilizando la API de *JasperReports*.

La combinación de *iReport* y *JasperReport* resulta ser una herramienta poderosa que mejora la experiencia del desarrollador para exportar información almacenada en bases de datos a archivos ofimáticos que facilitan la visualización y el análisis a cualquier tipo de usuario final que no posea conocimientos en sistemas.

Instalación De iReport

Para poder visualizar las utilidades de esta librería en una aplicación desarrollada en NetBeans, se necesita la siguiente lista de archivos complementarios:

- `${distro.zipfilename.extensions}.nbm`
- `iReport-5.5.0.nbm`
- `jasperreports-components-plugin-5.5.0.nbm`
- `jasperserver-plugin-5.5.0.nbm`

Para instalar los archivos mencionados, se debe ir a la barra de menús y seleccionar “Tools” (Herramientas) y luego hacer clic en la opción “Plugins”.

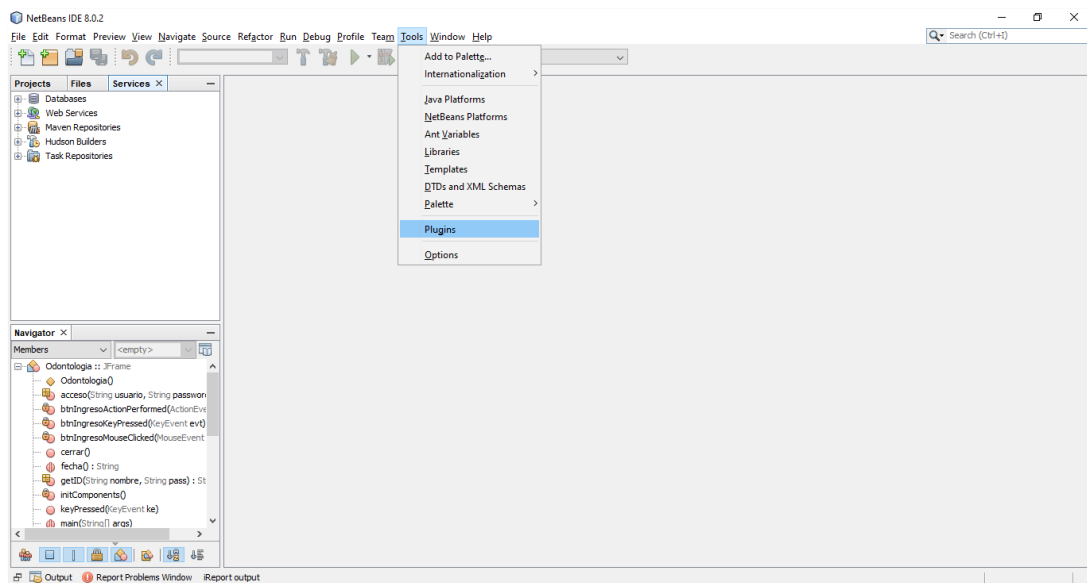


Imagen 23. Escoger opción plugins desde menú de Herramientas

Luego de esto, aparecerá una ventana alternativa que mostrará los plugins o complementos que se encuentran previamente instalados y funcionando en *NetBeans* para cualquier necesidad de programación. Allí se debe seleccionar la pestaña “Downloaded” (Instalados) para luego hacer clic en el botón “Add plugins” y buscar en el explorador de Windows la ubicación de los archivos que se necesitan para tener la librería de *iReport* disponible, seleccionarlos y hacer clic en el botón “Install” (Instalar) para así seguir la continuación de los pasos de instalación desde *NetBeans*. Una vez confirmada la instalación de esta librería, *NetBeans* debe indicar que se reiniciará automáticamente para completar el proceso.

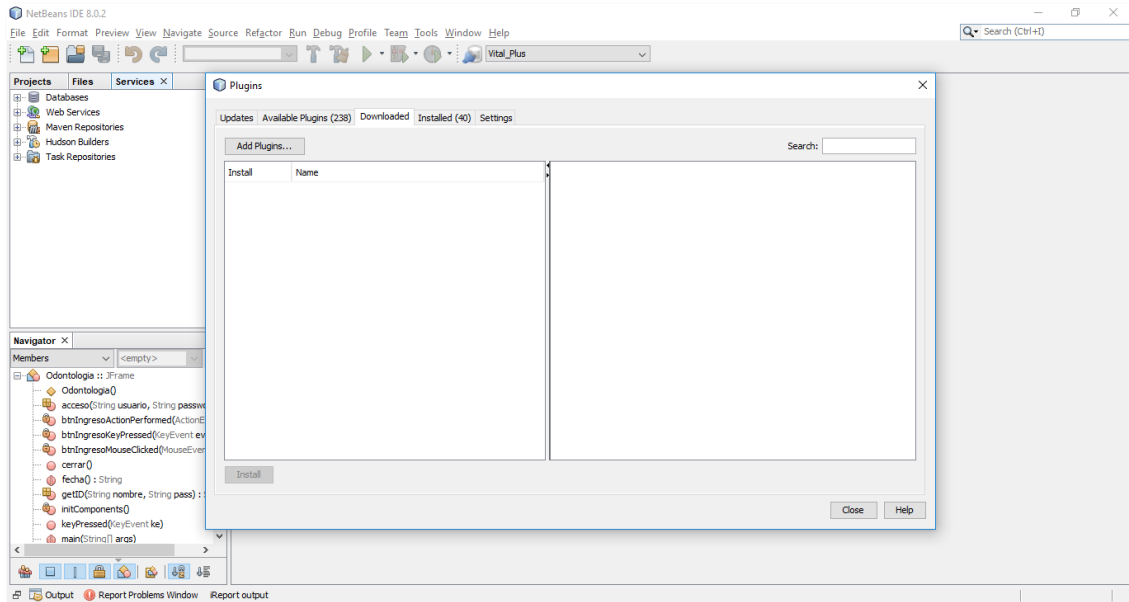


Imagen 24. Ventana de plugins

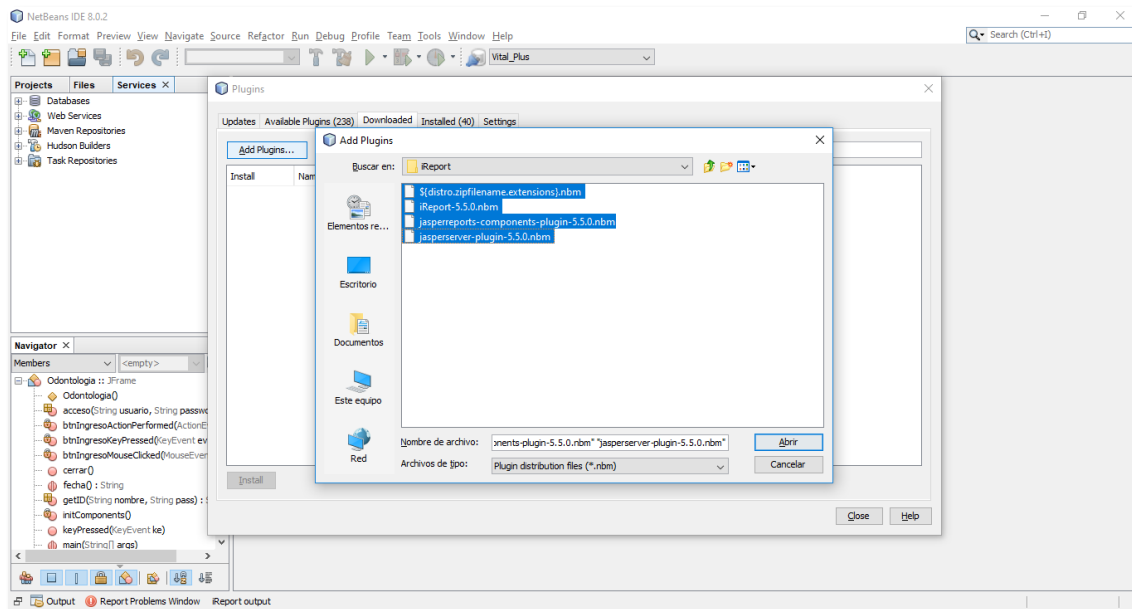


Imagen 25. Escogencia de archivos de plugin iReport

Después de reiniciarse, se puede proceder a establecer la conexión a la base de datos que usará la aplicación que se va a desarrollar en *NetBeans*, a través de un botón que activa la librería *iReport* ubicado en la barra de controles rápidos en la parte superior.

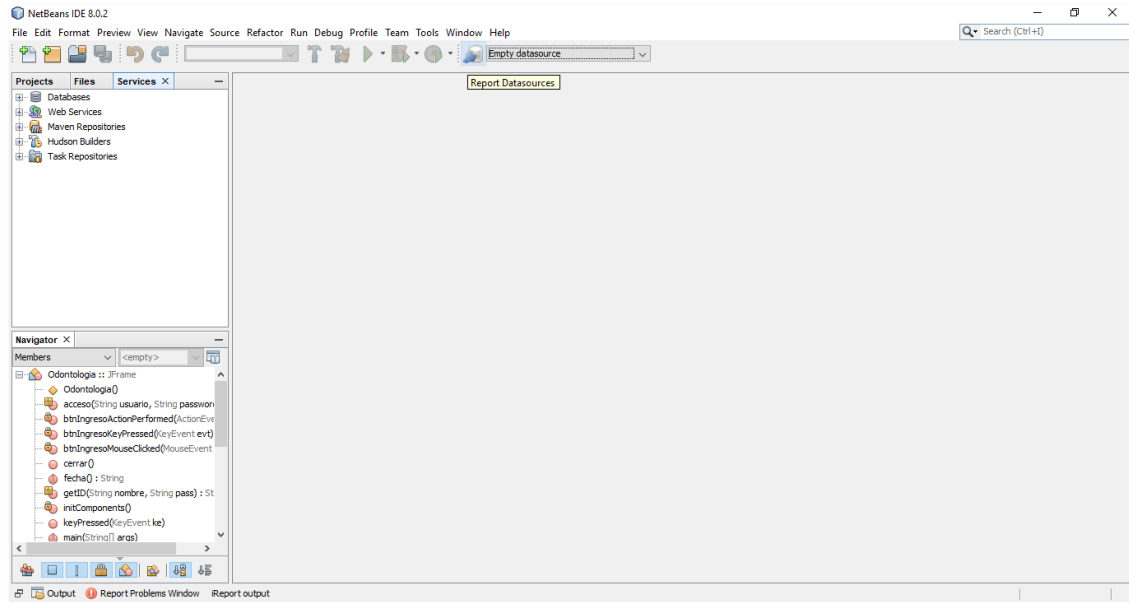


Imagen 26. Botón para añadir fuente de datos

En la ventana que aparece, se debe hacer clic en el botón “New” (Nueva) para ingresar el tipo de base de datos que se va a utilizar, el nombre de la base de datos y sus credenciales de acceso. Por último, se debe hacer clic en el botón de testeo de la parte inferior de la ventana para verificar que la conexión a la base de datos está garantizada.

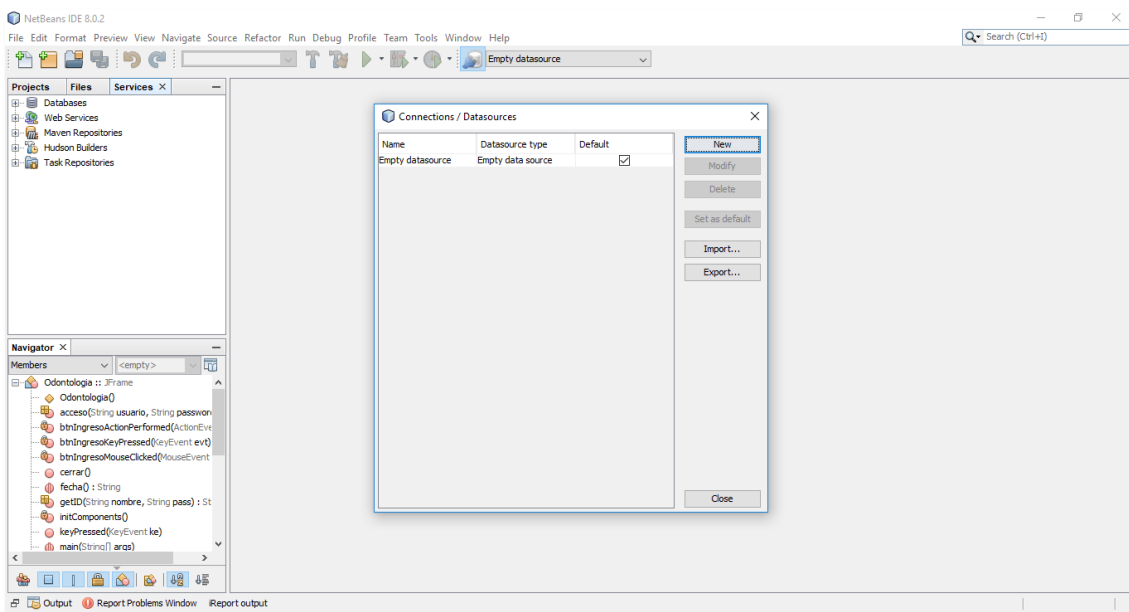


Imagen 27. Añadir fuente de datos

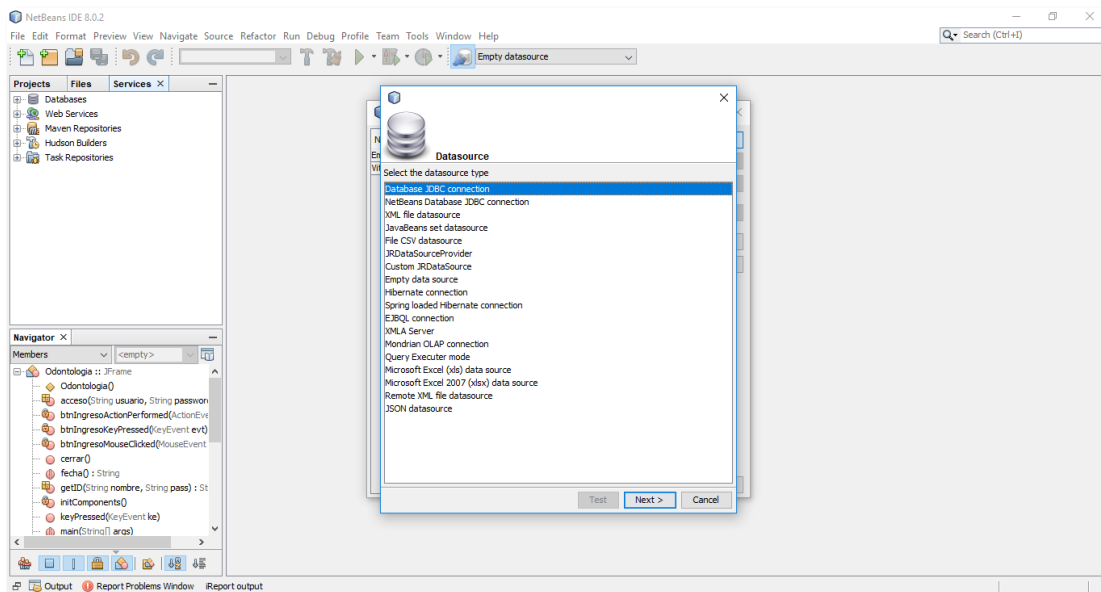


Imagen 29. Estableciendo tipo de conexión de base de datos

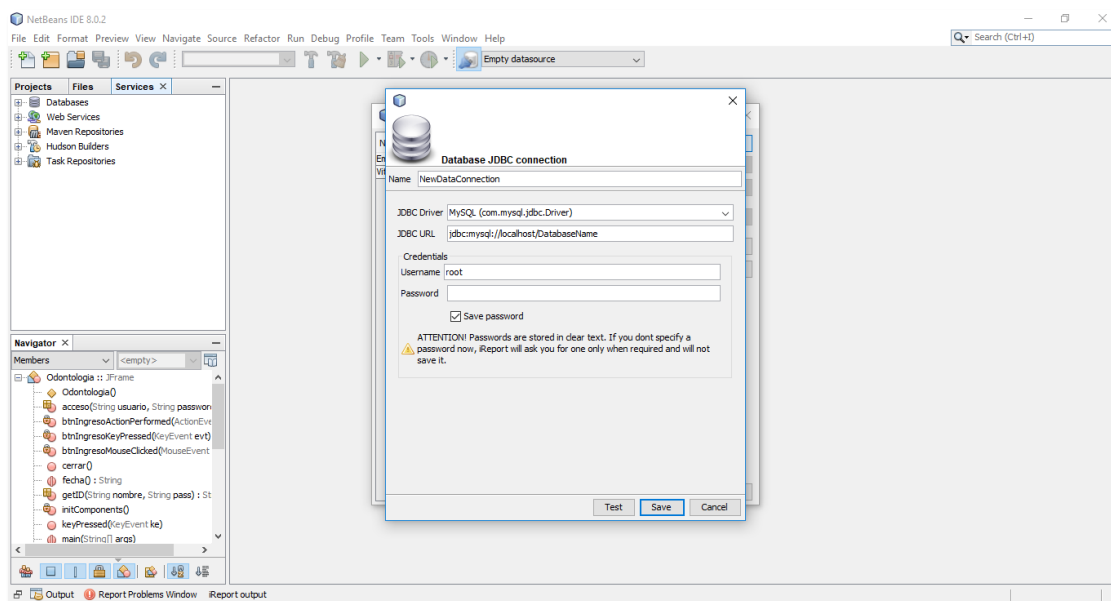


Imagen 28. Ingresando credenciales de base de datos

Una vez se confirma la conexión de *NetBeans* a la base de datos escogida, se completa la instalación de *iReport* en el entorno de desarrollo y la librería se encuentra lista para trabajar.

Instalación De JasperReports

Luego de la instalación de *iReport*, se procede a instalar la librería de *JasperReports* integrando los siguientes archivos ejecutables en Java (extensión .jar) en su orden:

- jasperreports-javaflow-4.7.1.jar
- jasperreports-4.7.1.jar
- commons-beanutils-1.8.2.jar
- commons-collections-3.2.1.jar
- commons-digester-2.1.jar
- commons-javaflow-20060411.jar
- commons-logging-1.1.1.jar
- groovy-all-1.7.5.jar
- iText-2.1.7.jar
- png-encoder-1.5.jar
- poi-3.7.jar

Para la inclusión de estos archivos, se debe seleccionar el menú “Tools” (Herramientas), el mismo que se utilizó para instalar *iReport*, pero se debe seleccionar la opción de “Libraries” (Librerías).

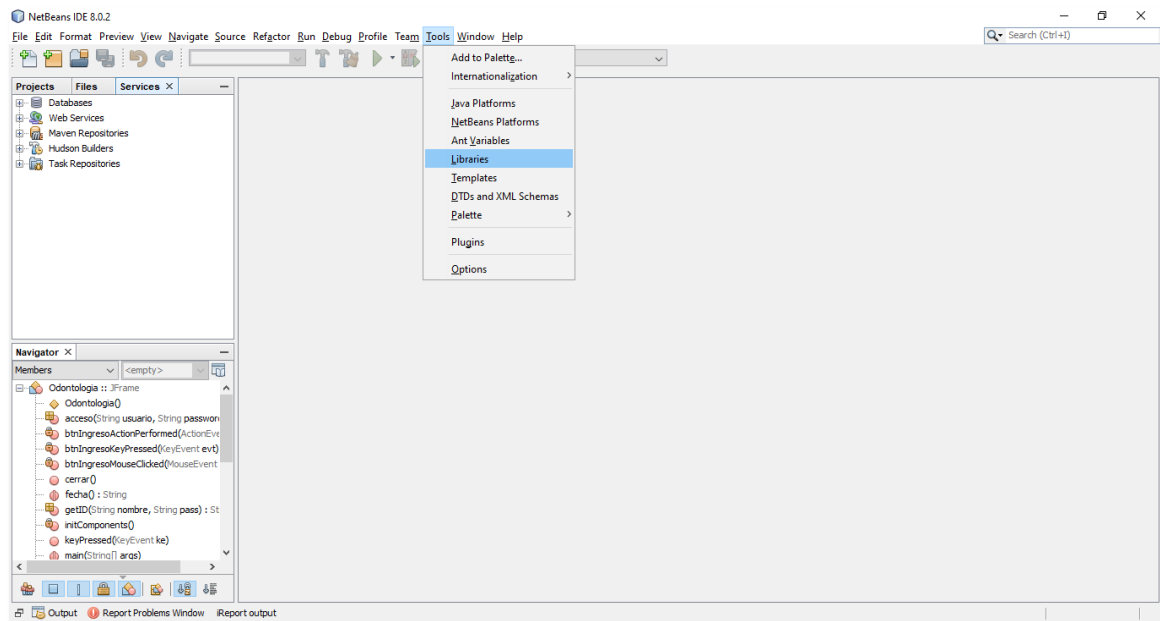


Imagen 30. Escogiendo opción de librerías en menú de Herramientas

En la consecuente ventana se observarán las librerías con las que dispone el entorno de desarrollo y para agregar la librería de *JasperReports*, se debe hacer clic en el botón “New Library...” (Nueva librería) localizado en la parte inferior izquierda de la ventana emergente y desde allí establecer su nombre.

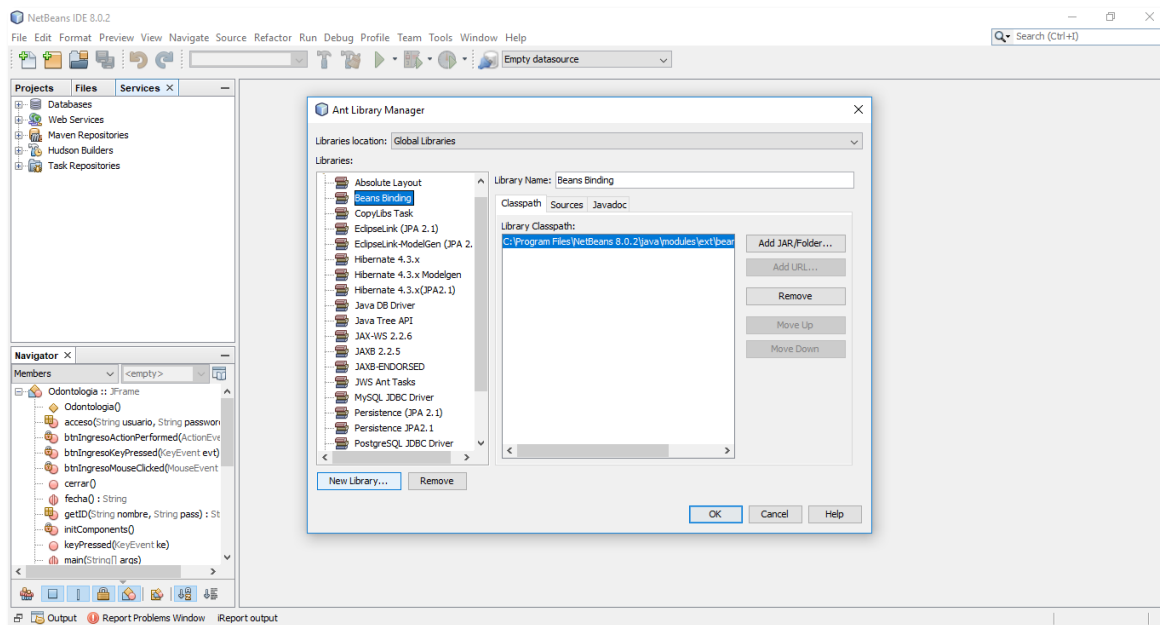


Imagen 32. Añadir nueva librería

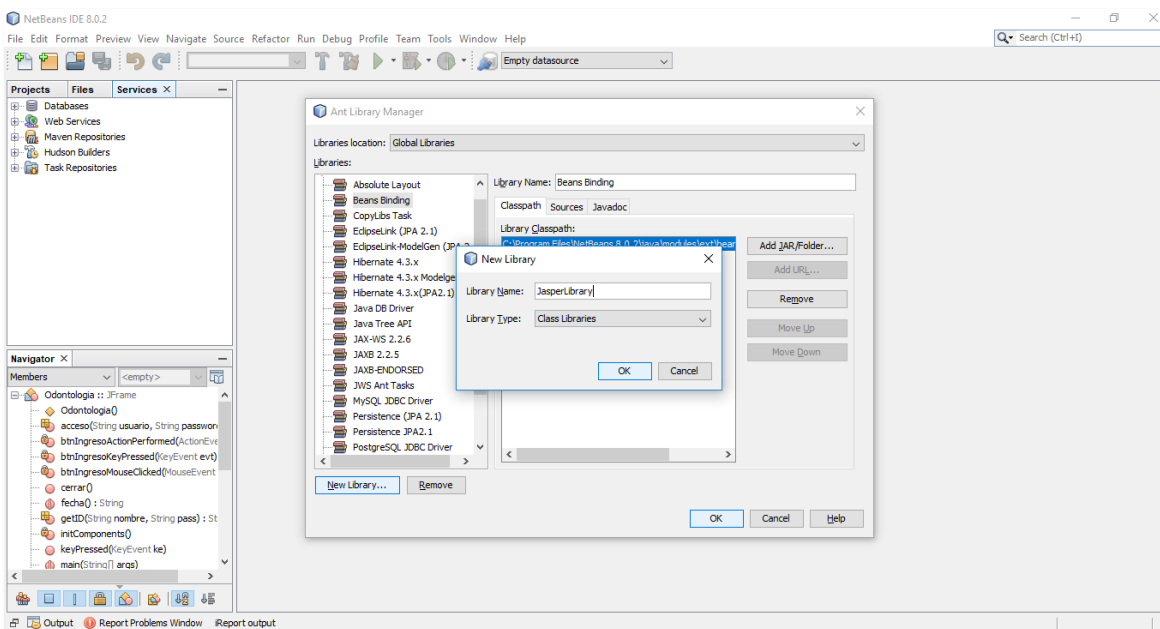


Imagen 31. Establecer nombre de librería

Luego de esto, se debe hacer clic al botón “Add JAR/Folder” para ubicar los archivos que se necesitan para que la librería funcione en *NetBeans*, siguiendo estrictamente el orden mencionado.

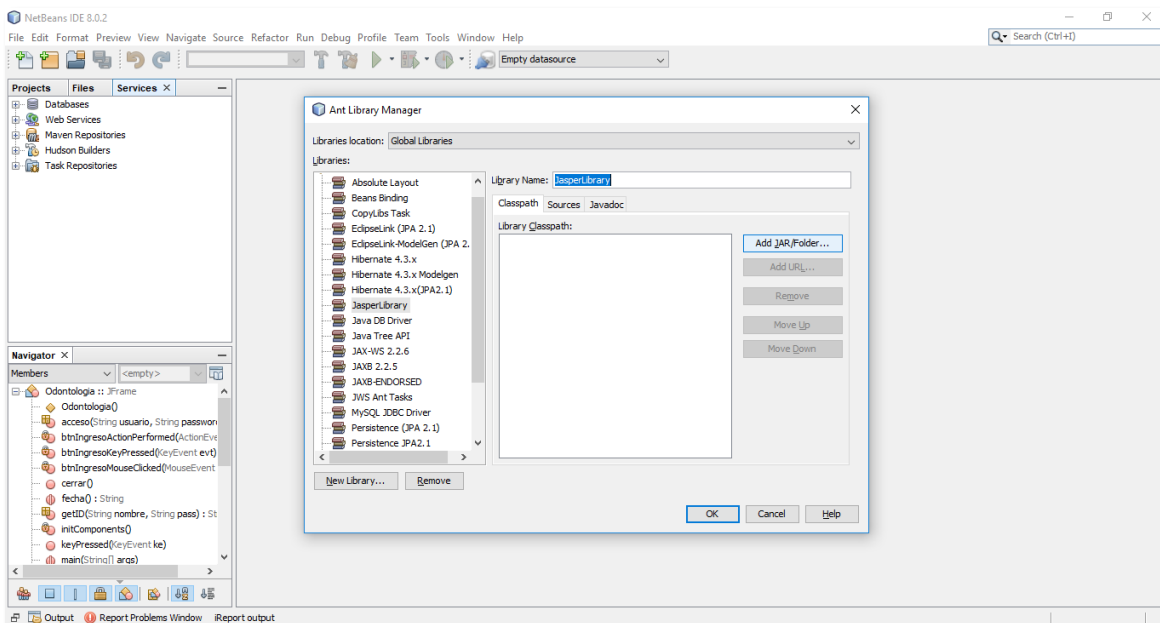


Imagen 33. Añadir archivos de librería JasperReports

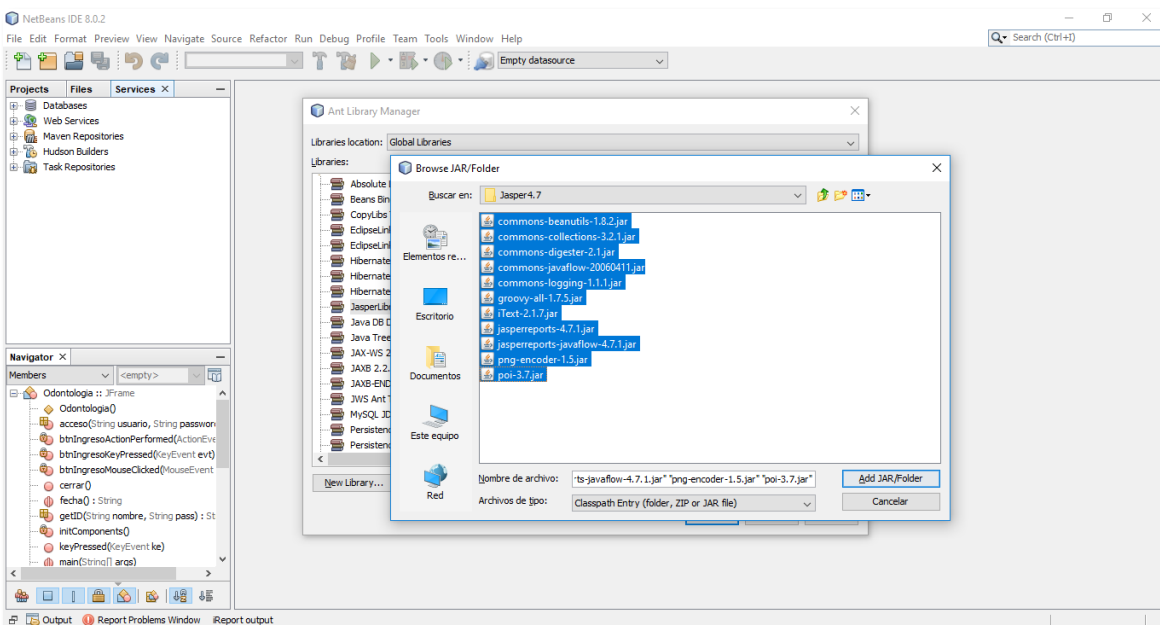


Imagen 34. Escogencia de archivos de librería JasperReports

En caso de que al incluir los archivos ejecutables no se muestren en dicho orden, la ventana habilita los botones de navegación para subir (Move Up) y bajar (Move Down) los archivos de posición. En este caso, los archivos *jasperreports-javaflow-4.7.1.jar* y *jasperreports-4.7.1.jar* deben estar en primera y segunda posición en la lista, respectivamente, para que la librería funcione correctamente. El resto de archivos pueden permanecer en el orden en que se encuentren.

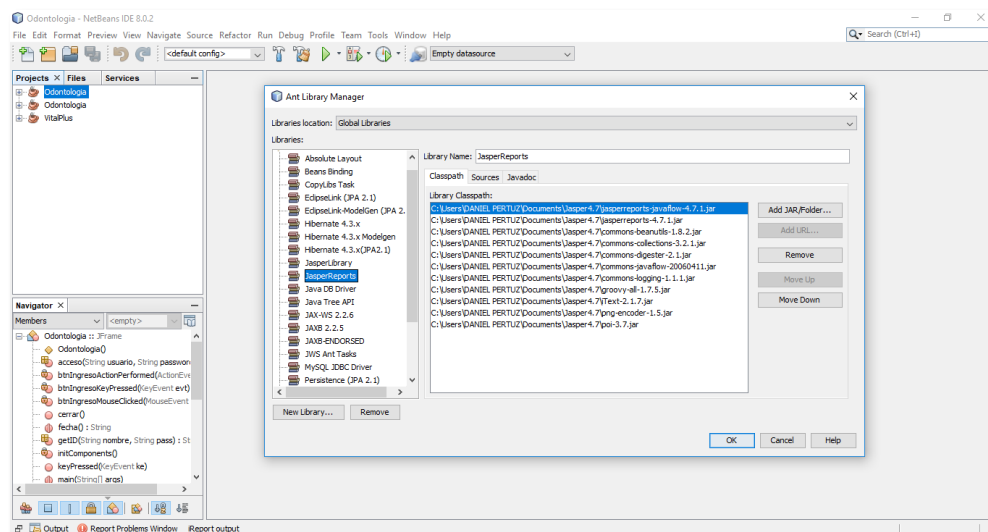


Imagen 35. Estableciendo posiciones de archivos de librería JasperReports

Se debe hacer clic en el botón “OK” para culminar el proceso de instalación de la librería *JasperReports*. Para confirmarlo, cualquier proyecto que se escoja, debe disponer de la opción para empezar a diseñar un reporte.

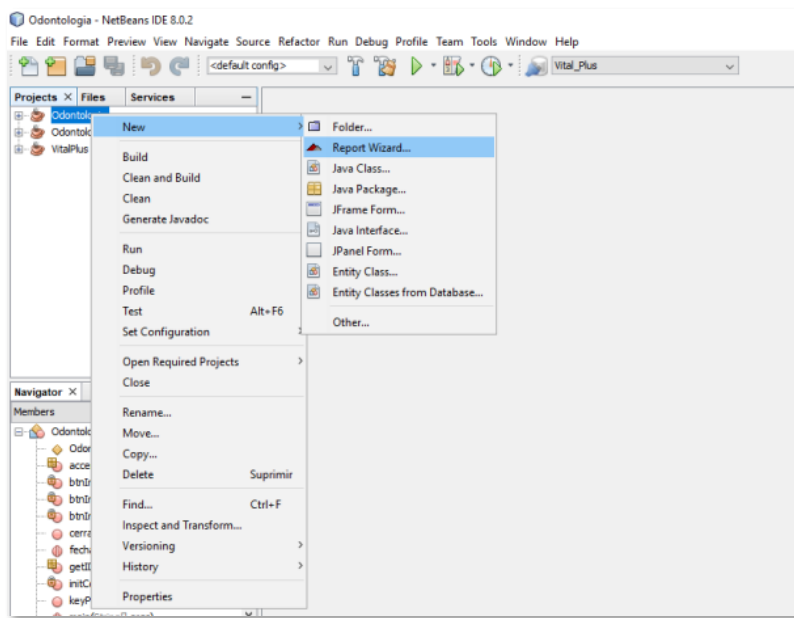


Imagen 36. Confirmación de librería JasperReports añadida

MANUAL DE INSTRUCCIONES DE APLICACIÓN JAVA PARA VITAL PLUS



REALIZADO POR: DANIEL DARÍO PERTUZ OROZCO

BARRANQUILLA, 2018

Contenido

Imágenes	101
Introducción	102
Prerrequisitos	103
Proceso de compilación	103
Acceso A La Aplicación	109
Entorno De Administrador.....	110
Administración De Usuarios.....	110
Administración De Profesionales	111
Administración De Pacientes	111
Administración De Laboratorios.....	112
Administración De Tratamientos.....	112
Facturación.....	113
Reportes	113
Registros	114
Entorno De Agente	114
Registro De Pacientes	115
Registro De Abonos.....	116
Conversión De Archivo <i>.jar</i> A <i>.exe</i>	116

Imágenes

Imagen 1. Pantalla de inicio de sesión	109
Imagen 2. Página principal de entorno para administrador	110
Imagen 3. Administración de usuarios.....	110
Imagen 4. Administración de profesionales.....	111
Imagen 5. Administración de pacientes	111
Imagen 6. Administración de laboratorios.....	112
Imagen 7. Administración de tratamientos	112
Imagen 8. Facturación.....	113
Imagen 9. Reportes	113
Imagen 10. Registros	114
Imagen 11. Página principal de agente	114
Imagen 12. Registro de pacientes	115

Imagen 13. Registro de abonos.....	116
Imagen 14. Información básica de archivo ejecutable (.exe).....	117
Imagen 15. Versión mínima de JRE para archivo ejecutable	117
Imagen 16. Compilación de archivo ejecutable	118

Introducción

Este manual de instrucciones representa una guía de entendimiento de los procedimientos y reglamentaciones que influyen en el andamiaje de la aplicación desarrollada en el lenguaje de programación Java para la empresa Vital Plus.

El propósito de esta aplicación se enfatiza en dar solución a la problemática que presenta la empresa Vital Plus en su proceso de contabilidad y facturación, la cual impide que pueda realizar una prestación óptima y sistemática a sus pacientes suscritos en servicios de salud oral y que conlleve a mantener o mejorar una posición favorable de la empresa dentro del área industrial en que se desenvuelve.

Esta aplicación de escritorio realiza sus operaciones teniendo en cuenta las condiciones de suscripción de los pacientes, las operaciones contables que se deben ejecutar durante pagos, los roles de los usuarios operarios de la empresa y sus privilegios y las necesidades que tiene la empresa para la administración y visualización de la información que maneja durante sus jornadas laborales.

A través de este manual se observarán ilustraciones didácticas que servirán de guía para una mejor explicación de los entornos de trabajo que posee la aplicación. Estos entornos están dedicados a los roles de usuario agente y usuario administrador, cada uno con sus privilegios y actividades especiales para brindar un mejor servicio a los pacientes suscritos.

La aplicación cumple con requisitos de seguridad y digitación utilizando algoritmos de encriptación y restricción de caracteres que definen su alta confiabilidad y capacidad en la prevención de posibles errores que se pueden presentar durante las jornadas laborales de la empresa.

También se observarán las ventajas que ofrece la aplicación para exportar información almacenada en base de datos a archivos ofimáticos de fácil legibilidad a través del empleo de las librerías especializadas iReport y JasperReports, las cuales ofrecen interfaces gráficas agradables al usuario final y diversas alternativas de manipulación de datos.

Se recomienda conservar este manual instructivo para alcanzar el mejor entendimiento posible del funcionamiento de esta aplicación y futuras consultas.

Prerrequisitos

- JVM (Java Virtual Machine).
- Servidor de base de datos.

Proceso de compilación

Esta aplicación fue desarrollada mediante la herramienta *NetBeans*, debido a su compatibilidad con aplicaciones Java y permite crear interfaces gráficas de forma rápida, además de ser extensible por medio de plugins. Esta herramienta intuitiva se encuentra disponible en la página web <http://www.netbeans.org> para ser descargada. También se necesita la plataforma de desarrollo JDK, la cual viene incluida en el paquete de instalación de *NetBeans*.

La instalación de ambos programas resulta sencilla. Sólo se designan los lugares de instalación en el ordenador y avanzar en las ventanas típicas para aceptar los términos de licenciamiento.

Luego de realizar este proceso se necesitan dos componentes fundamentales para el desarrollo de aplicaciones que requieren de conexión y operación sobre bases de datos y de manejo de fechas. La primera de ellas es la API de Java diseñada para el acceso a bases de datos llamada JDBC (*Java Database Connectivity*). Esta API es compatible con los sistemas gestores de bases de datos comerciales y la comunicación entre ellas es establecida mediante controladores JDBC que conceden el acceso a los datos a las aplicaciones Java. Para entender y aplicar esta comunicación entre controladores, se recomienda el sistema MySQL o PostgreSQL, por ser más ligeros y asequibles que otros más sofisticados como Oracle o Informix.

El controlador JDBC de Java para acceso a bases de datos MySQL se distribuye mediante un archivo ejecutable JAR que además ofrece el paquete `java.sql` y puede ser adquirido mediante la página web <http://www.mysql.com> y aparece con el nombre `sqljdbc4-3.0.jar`. Una vez descargado este archivo, se procede a incluirlo en el directorio de librerías que implementará *NetBeans*. Simplemente con acceder al menú de librerías de *NetBeans* se puede llevar a cabo la inclusión de esta librería y a partir de allí se puede empezar a digitar los distintos comandos que realizan el acceso a bases de datos.

Los comandos más útiles de esta librería para trabajar en bases de datos se destacan:

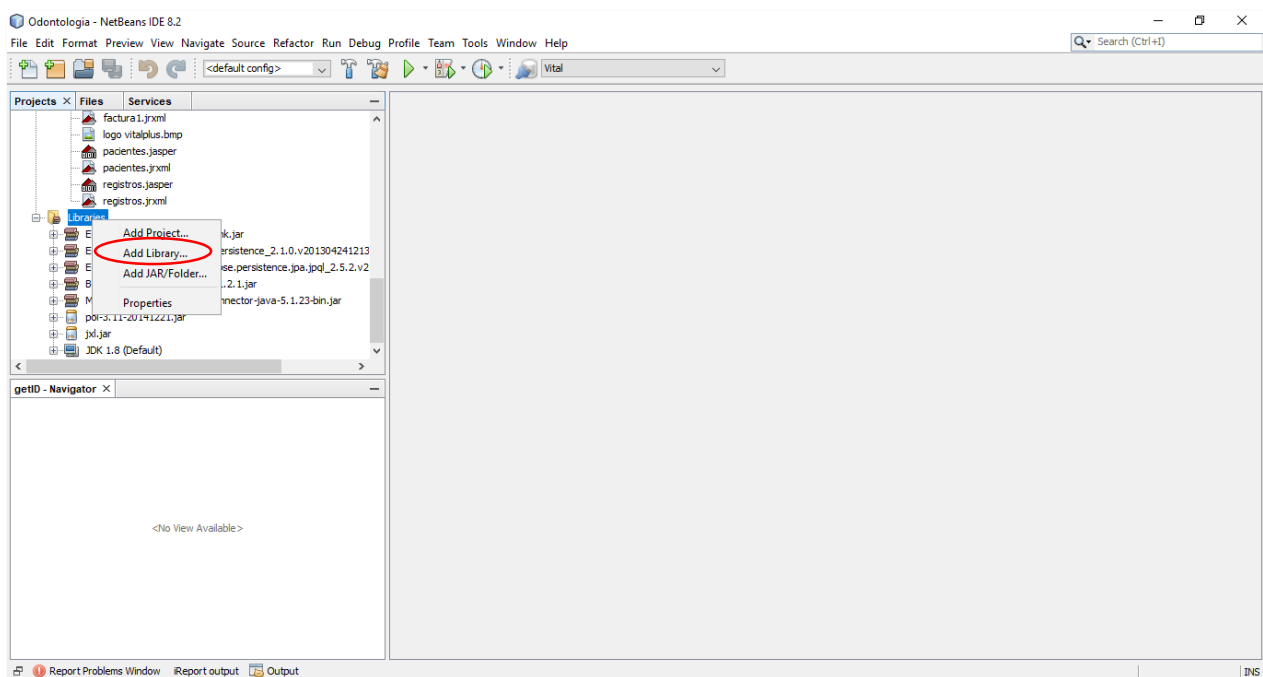
- **DriverManager:** comando para cargar la interacción entre la aplicación y los dispositivos.
- **Connection:** comando para establecer conexiones con las bases de datos.
- **Statement:** comando para ejecutar sentencias SQL y enviarlas a las bases de datos.
- **PreparedStatement:** se establece la ruta de ejecución en el servidor de base de datos para que pueda ser ejecutada varias veces.
- **ResultSet:** comando para almacenar el resultado de una consulta a bases de datos.

Con esta librería la interacción entre una aplicación Java y una base de datos está garantizada y no presenta retrasos en respuestas de ejecución.

El segundo componente esencial para esta aplicación es *JCalendar*. Esta librería es compatible con Java y proporciona una interfaz gráfica propia para elegir determinadas fechas. Se divide en varias dependencias que pueden ser empleadas por el usuario final en determinados casos, como lo son *JDateChooser*, *JDayChooser*, *JMonthChooser* y *JYearChooser*, cada una con sus propiedades, iconos y editores, con lo cual pueden ser usados por compiladores de interfaz gráfica sin inconvenientes. Se utiliza el archivo JAR llamado *jcalendar-1.3.3.jar*.

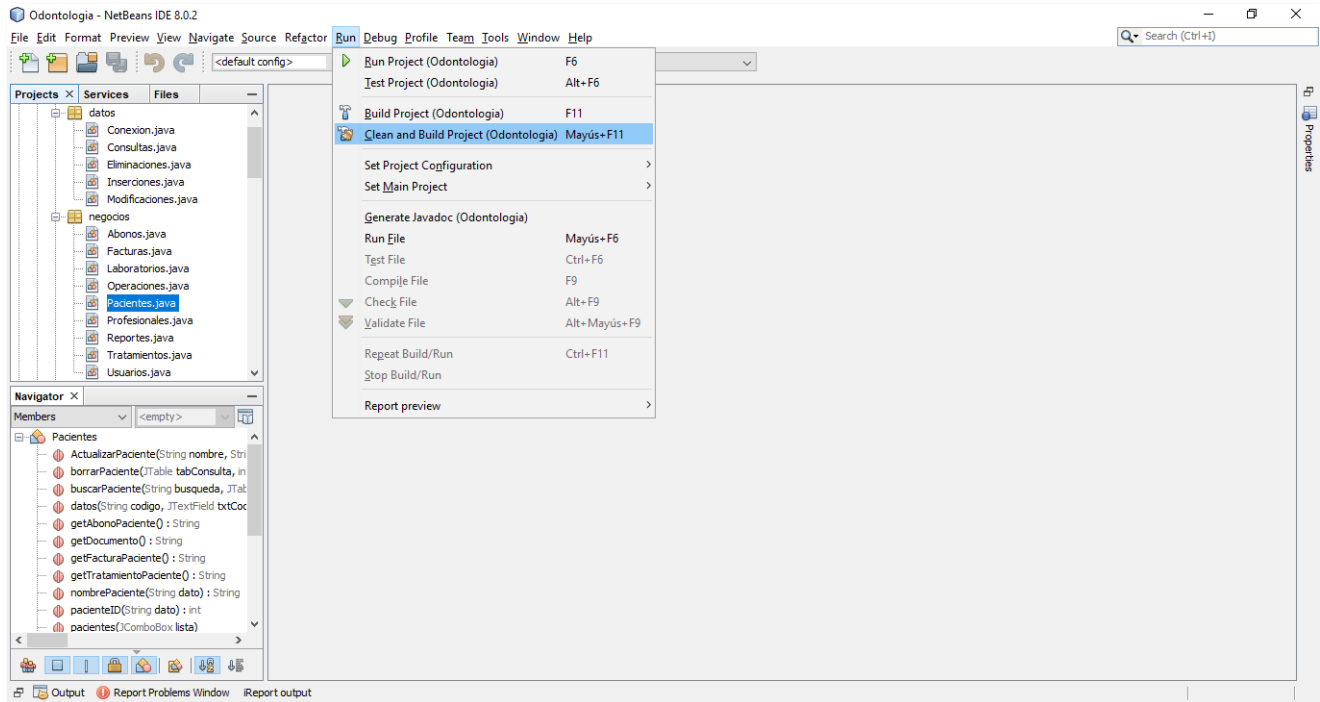
Por ser software libre, este programa puede ser redistribuido y modificado y su instalación sólo requiere de ser incluido en las librerías que pueden ser usadas por *NetBeans*, por lo que resulta ser un proceso sencillo de realizar.

En *NetBeans* ambos complementos son incluidos desde el navegador de carpetas que posee. Sólo se debe hacer clic derecho en el icono carpeta de librerías y luego escoger la opción para agregar archivo JAR. Se abrirá la ventana de explorador de Windows para ubicar los archivos correspondientes a cada librería.

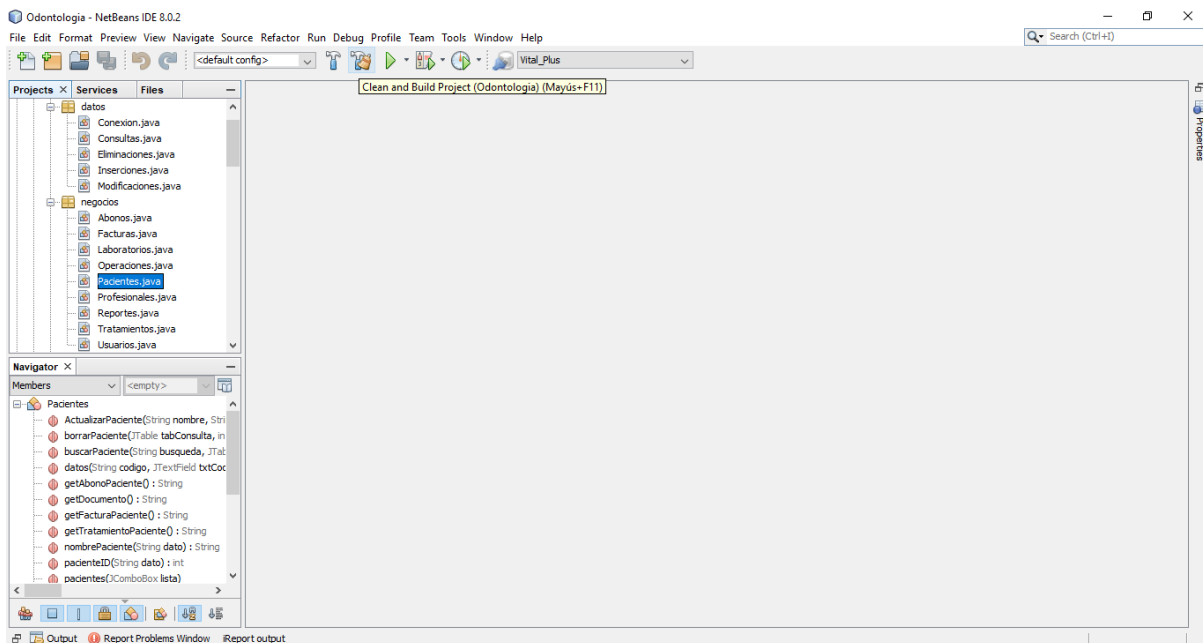


Luego de instalar estos complementos, se procede al desarrollo del código que realiza la construcción de la aplicación tanto en su interfaz gráfica como en sus operaciones lógicas. Una vez se ha concluido esta fase de diseño, se pasa a compilar todos los archivos que componen la aplicación en uno solo con extensión *.jar*. Mediante este archivo se accede a todas las ventanas de la aplicación y las operaciones que se realizan dentro y fuera de ellas.

Para realizar esta compilación desde *NetBeans*, existen dos series de pasos para la misma ejecución. La primera de ellas consiste en dirigirse al menú de “Ejecutar” (“Run”) y luego escoger la opción de “Limpiar y compilar proyecto” (“Clean and Build Project”) para que el entorno de desarrollo empiece a crear el archivo compilado.

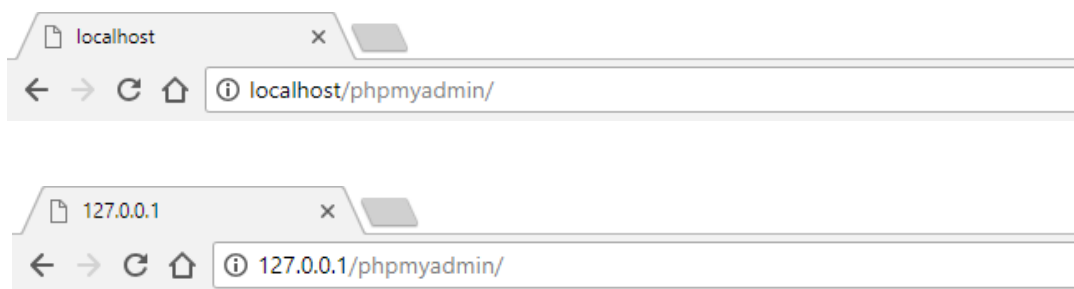
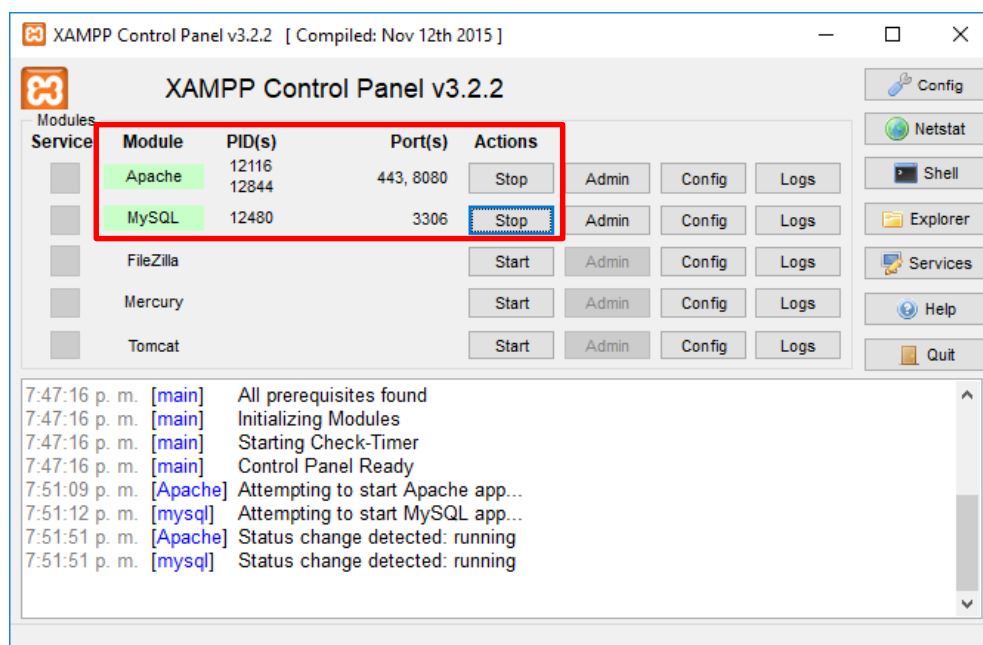


La segunda opción es hacer clic en el botón que realiza este proceso de compilación del archivo de manera más rápida, el cual se encuentra ubicado en la barra de accesos directos de *NetBeans*.



El complemento para el arranque de la aplicación es su conexión a la base de datos que maneja la información relacionada con la empresa. Dicha base de datos es cargada al servicio de MySQL de XAMPP por medio del script que contiene los comandos de creación de las tablas que hacen parte del sistema.

Para cargar este script en XAMPP se deben inicializar los servicios de Apache y MySQL desde el panel de control haciendo clic en los correspondientes botones de “Start” y luego desde el navegador web ingresar la palabra localhost o la dirección IP 127.0.0.1 para acceder al administrador de XAMPP.



Después de acceder al administrador de XAMPP se procede a crear el entorno que almacenará la base de datos del sistema haciendo clic en el botón “Nueva” y luego ingresar el nombre de la base de datos, que en este caso será “vital_plus”. Por último, se hace clic en el botón de “Importar” en la parte superior del administrador, ubicar el archivo que contiene el script de la base de datos y presionar el botón “Continuar”.

phpMyAdmin

Reciente Favoritas

Nueva

information_schema
mysql
new_schema
performance_schema
phpmyadmin
test

Configuraciones generales

Cotejamiento de la conexión al servidor: utf8mb4_unicode_ci

Configuraciones de apariencia

Idioma - Language: Español - Spanish

Tema: pmahomme

Tamaño de fuente: 82%

Más configuraciones

Servidor de base de datos

- Servidor: 127.0.0.1 via TCP/IP
- Tipo de servidor: MariaDB
- Versión del servidor: 10.1.21-MariaDB - mariadb.org binary distribution
- Versión del protocolo: 10
- Usuario: root@localhost
- Conjunto de caracteres del servidor: UTF-8 Unicode (utf8)

Servidor web

- Apache/2.4.25 (Win32) OpenSSL/1.0.2j PHP/7.1.1
- Versión del cliente de base de datos: libmysql - mysqlnd 5.0.12-dev - 20150407 - \$Id: b396954eeb2d1d9ed7902b8bae237b287f21ad9e \$
- extensión PHP: mysqli curl mbstring
- Versión de PHP: 7.1.1

phpMyAdmin

- Acerca de esta versión: 4.6.5.2
- Documentación
- Página oficial de phpMyAdmin
- Contribuir
- Obtener soporte
- Lista de cambios

Consola

phpMyAdmin

Reciente Favoritas

Nueva

information_schema
mysql
new_schema
performance_schema
phpmyadmin
test

Bases de datos

Crear base de datos

vital_plus Cotejamiento Crear

Base de datos	Cotejamiento	Acción
<input type="checkbox"/> information_schema	utf8_general_ci	Seleccionar privilegios
<input type="checkbox"/> mysql	latin1_swedish_ci	Seleccionar privilegios
<input type="checkbox"/> new_schema	latin1_swedish_ci	Seleccionar privilegios
<input type="checkbox"/> performance_schema	utf8_general_ci	Seleccionar privilegios
<input type="checkbox"/> phpmyadmin	utf8_bin	Seleccionar privilegios
<input type="checkbox"/> test	latin1_swedish_ci	Seleccionar privilegios

phpMyAdmin

Reciente Favoritas

Nueva

information_schema
mysql
new_schema
performance_schema
phpmyadmin
test
vital_plus

Importando en la base de datos "vital_plus"

Archivo a importar:

El archivo puede ser comprimido (gzip, bzip2, zip) o descomprimido. Un archivo comprimido tiene que terminar en [formato].[compresión]. Por ejemplo: .sql.zip

Buscar en su ordenador: Seleccionar archivo Ningún archivo seleccionado (Máximo: 2,048KB)

También puede arrastrar un archivo en cualquier página.

Conjunto de caracteres del archivo: utf-8

Importación parcial:

Permitir la interrupción de una importación en caso que el script detecte que se ha acercado al límite de tiempo PHP. (Esto podría ser un buen método para importar archivos grandes; sin embargo, puede dañar las transacciones.)

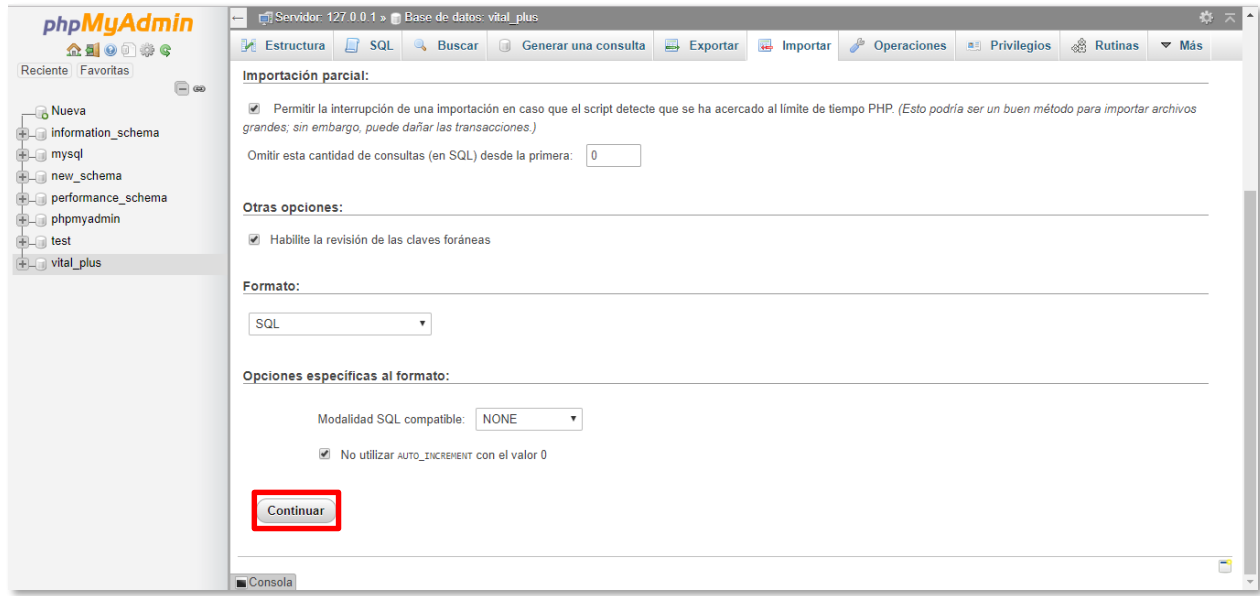
Omitir esta cantidad de consultas (en SQL) desde la primera: 0

Otras opciones:

Habilite la revisión de las claves foráneas

Formato:

Consola



Acceso A La Aplicación

Para el arranque de la aplicación se debe hacer doble clic en el icono del archivo ejecutable localizado en el escritorio del ordenador. La aplicación establece la comunicación con la base de datos para mostrar la página de inicio de sesión de usuarios. Se deben ingresar las credenciales de usuario y hacer clic en el botón “Ingresar” o presionar el botón Enter desde el teclado para acceder al entorno correspondiente a su perfil. El sistema realiza la validación de credenciales para conceder el acceso.

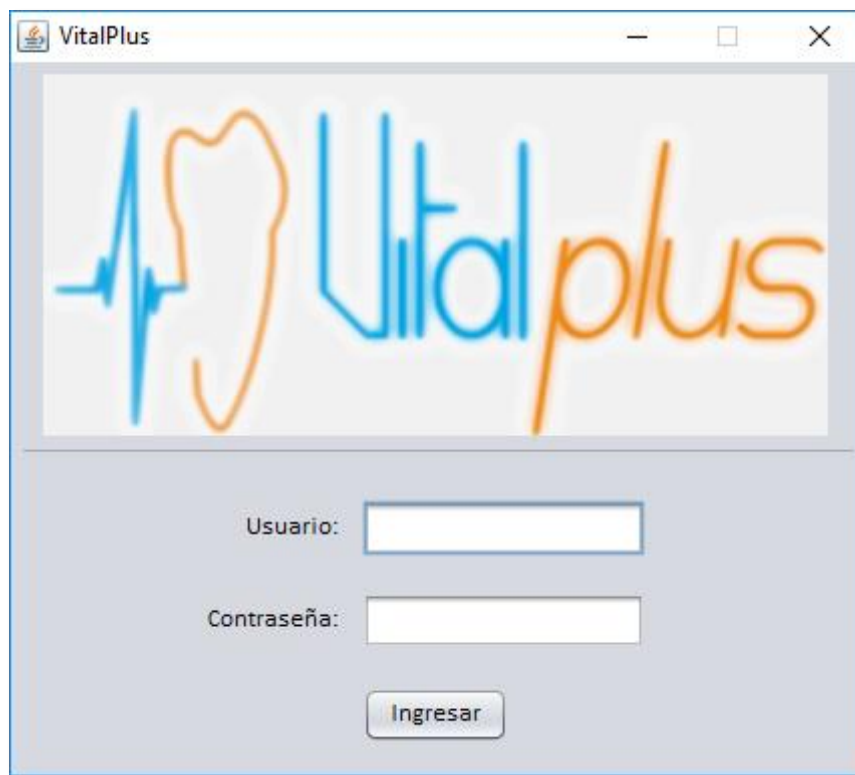


Imagen 37. Pantalla de inicio de sesión

Entorno De Administrador

Este entorno posee botones que dirigen al usuario hacia espacios que permiten la manipulación y exportación de información correspondiente a diferentes actores o procesos que hacen parte de la empresa.



Imagen 38. Página principal de entorno para administrador

Administración De Usuarios

Desde aquí el administrador gestiona la creación, modificación y eliminación de los usuarios agentes u otros administradores que usarán la aplicación dentro de la empresa.

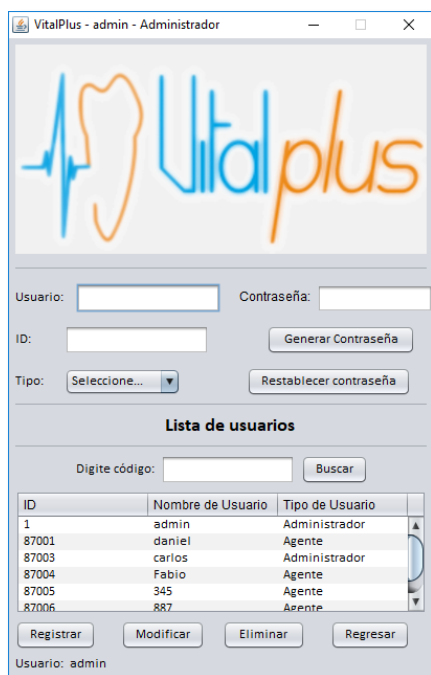


Imagen 39. Administración de usuarios

Administración De Profesionales

Espacio para diligenciar la información correspondiente a los profesionales de la salud oral que trabajan en la empresa.

Código/Cédula	Nombre	Especialidad	Laboratorio
22468779	María Bermúdez	Rehabilitación	lab1
22582332	Jazmín Aycordi	Rehabilitación	lab1
32748799	Claudia Gutiérrez	General	lab1
45480264	Claudia Giamo	Odontopediatria	lab1
52886835	Paola González	Endodoncia	lab1

Imagen 40. Administración de profesionales

Administración De Pacientes

El administrador exporta la lista de pacientes que se encuentran suscritos a la empresa.

Imagen 41. Administración de pacientes

Administración De Laboratorios

La información correspondiente a los laboratorios que se encuentran aliados a la empresa es gestionada en este espacio.

Nit	Nombre
123	lab1
456	lab2

Imagen 42. Administración de laboratorios

Administración De Tratamientos

Se ingresa, consulta, modifica y elimina la información concerniente a los tratamientos que ofrece la empresa a sus pacientes suscritos.

Código	Nombre	Costo	Descripción
110	General	500000	Servicio de procedimie...
225	Ortodoncia	3000000	Procedimiento para la...
5865	Nuevo Proceso	1502000	Nueva descripción
8911	Nuevo	60000	nuevo
8910	kjadndjbquy	1502400	midwuw
23659	Prueba	1245000	Prueba

Imagen 43. Administración de tratamientos

Facturación

En este espacio, el administrador realiza consultas en base de datos para exportar las facturas generadas por los pacientes suscritos a archivos ofimáticos, así como también permite su eliminación.

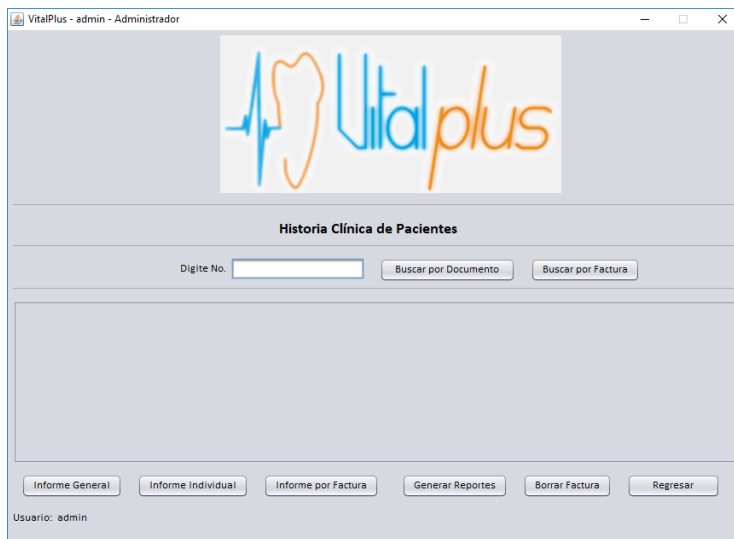


Imagen 44. Facturación

Reportes

Se generan reportes de facturación dentro de periodos de tiempo específicos y por otros criterios de consulta.

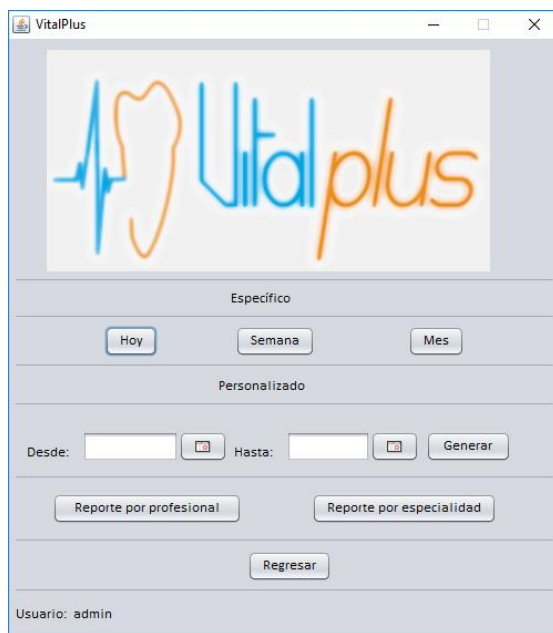


Imagen 45. Reportes

Registros

Por medio de este botón se exporta el registro de actividades que realizaron los usuarios agentes en un día específico.



Imagen 46. Registros

Entorno De Agente

Entorno dedicado a empleados de la empresa asignados para asistir de forma directa a los pacientes en su registro en el sistema y en la gestión de los abonos correspondientes a los tratamientos que solicite. Además, se realizan consultas de abonos registrados en base de datos siguiendo diversos criterios de búsqueda.

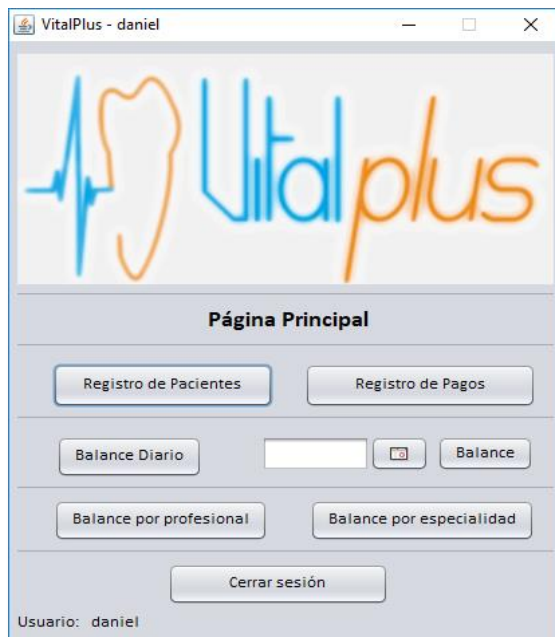


Imagen 47. Página principal de agente

Registro De Pacientes

El agente debe diligenciar el formulario que dispone este entrono para completar el registro del paciente en el sistema. Sumado a esto el agente puede modificar la información del paciente o eliminarlo del sistema.

Registro de Pacientes

Nombre: Documento:

Dirección: Teléfono: Edad:

Código: Tratamiento: Descuento (%):

Descripción: Costo:

Tipo: Remitido por:

Profesionales disponibles

Nombre	Código	Especialidad	Laboratorio
María Bermúdez	22468779	Rehabilitación	lab1
Jazmín Aycordi	22582332	Rehabilitación	lab1
Claudia Gutiérrez	32748799	General	lab1
Claudia Giamo	45480264	Odontopediatría	lab1
Paola González	52886835	Endodoncia	lab1

Guardar Modificar... Limpiar Regresar

Usuario: daniel

Imagen 48. Registro de pacientes

Registro De Abonos

En esta ventana, el agente ingresa los datos necesarios para el registro de los abonos en sistema. Además, tiene la posibilidad de crear nuevos tratamientos o gastos extra relacionados al paciente.

Imagen 49. Registro de abonos

Conversión De Archivo .jar A .exe

Con el fin de que la aplicación tenga un aspecto más familiar con lo que habitualmente se observa en el desarrollo de aplicaciones, se decidió utilizar la herramienta *Launch4j*, el cual se encarga de empaquetar el archivo de extensión *.jar* a un archivo de extensión *.exe* para que pueda controlar las configuraciones usadas en la máquina virtual de Java y así permitir la ejecución de la aplicación en cualquier sistema operativo que se esté usando.

Al abrir este programa se deben ingresar pocos datos que son esenciales para la conversión del archivo y que se encuentran resaltados con un asterisco al costado de cada campo y que se rellenan presionando los botones de carpeta para ubicar los archivos correspondientes. Estos campos son “Archivo de salida” (“Output File”) y “Jar” y otros campos complementarios

como lo son “Icono” (“Icon”) y “Mínima versión de JRE” (“Min JRE versión”), éste último ubicado en la pestaña de JRE del programa y que basta con ingresar el parámetro 1.0.0 en él.

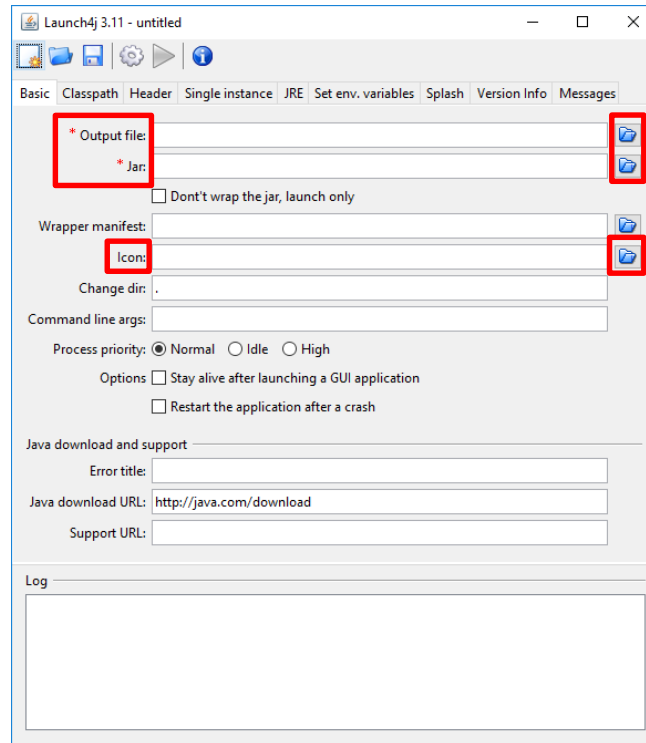


Imagen 50. Información básica de archivo ejecutable (.exe)

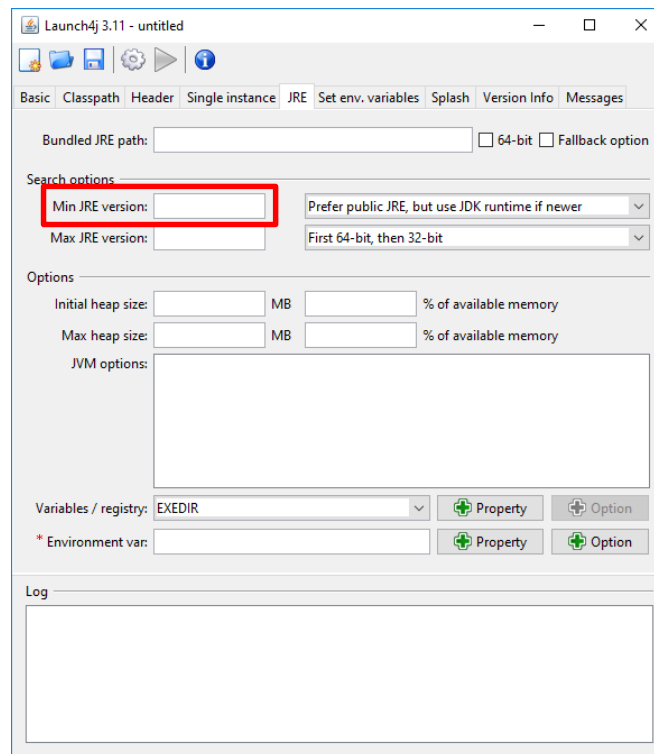


Imagen 51. Versión mínima de JRE para archivo ejecutable

Por último, se hace clic en el botón de compilación en la parte superior de la ventana para luego establecer el archivo de extensión .xml con el que culmina la construcción del archivo ejecutable.

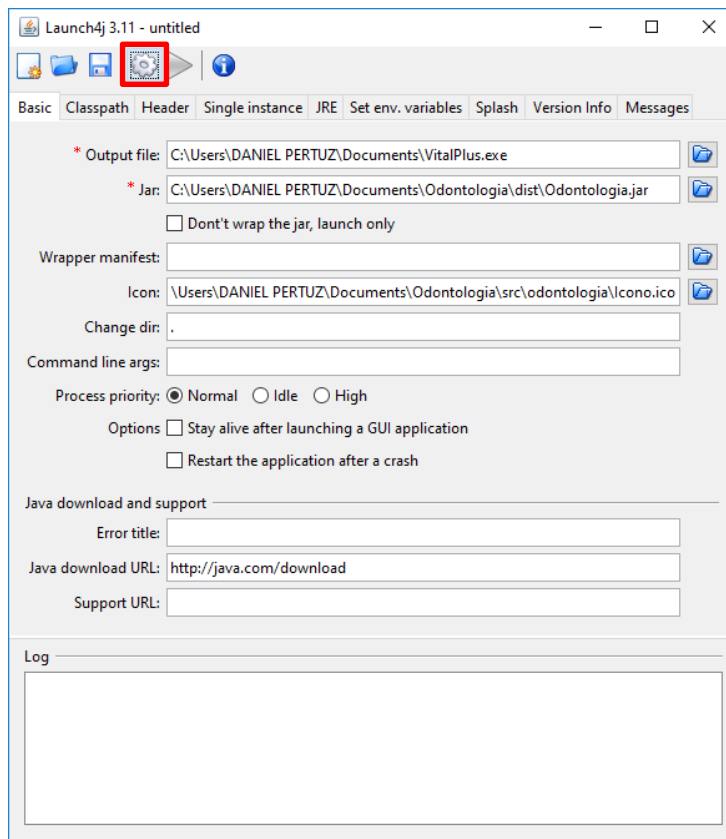


Imagen 52. Compilación de archivo ejecutable

Anexo C

Diseño De Pantallas

Contenido

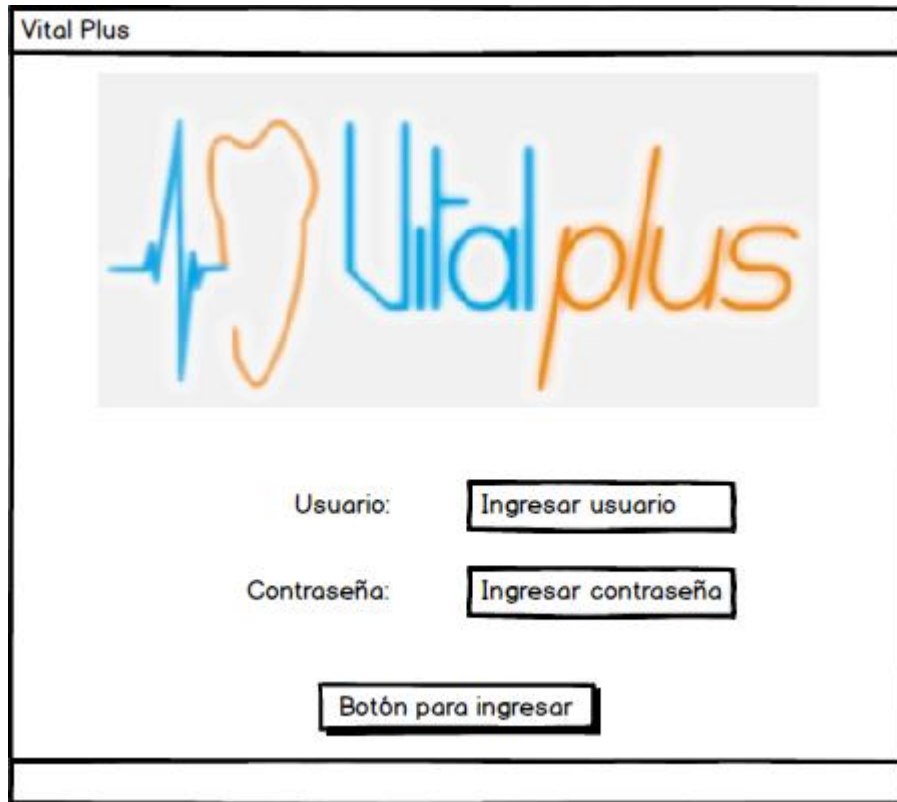
1.	Pantalla Inicial: Inicio De Sesión.....	122
2.	Espacio De Agente.....	123
2.1.	Pantalla Principal.....	123
2.2.	Registro De Pacientes.....	124
2.3.	Registro De Datos.....	125
2.4.	Modificación De Datos.....	126
2.5.	Registro De Abonos.....	127
2.6.	Ingreso De Datos De Abono.....	128
2.6.1.	Ingreso De Nuevos Tratamientos O Gastos Extra.....	130
2.6.2.	Consulta De Balance Económico.....	132
2.6.3.	Cerrar Sesión.....	139
3.	Espacio De Administrador.....	140
3.1.1.	Admin. Usuarios.....	141
3.1.2.	Restablecer Contraseña.....	148
3.1.3.	Admin. Profesionales.....	149
3.1.4.	Admin. Pacientes.....	153
3.1.5.	Admin. Laboratorios.....	154
3.1.6.	Admin. Tratamientos.....	156
3.2.	Informes.....	161
3.2.1.	Informe General.....	161
3.2.2.	Informe Individual.....	162
3.2.3.	Informe Por Factura.....	164
3.2.4.	Borrar Factura.....	165
3.2.5.	Generar Reportes.....	166
3.2.6.	Reporte Por Profesional.....	169
3.2.7.	Reporte Por Especialidad.....	170
3.2.8.	Registros.....	171

Imagen 1. Pantalla de inicio de sesión	122
Imagen 2. Pantalla de inicio de sesión de agentes	123
Imagen 3. Registro de pacientes	124
Imagen 4. Confirmación de registro de pacientes.....	125
Imagen 5. Modificación de información de pacientes	126
Imagen 6. Ingreso de nuevos datos de paciente.....	126
Imagen 7. Confirmación de datos actualizados de pacientes.....	127
Imagen 8. Pantalla de registro de abonos.....	127
Imagen 9. Información de registro escogido.....	128
Imagen 10. Confirmación de registro de abono.....	128
Imagen 11. Abono visualizado en tabla de abonos registrados	129
Imagen 12. Confirmación de abono registrado.....	129
Imagen 13. Abono visualizado en tabla de abonos registrados	130
Imagen 14. Ingreso de datos de nuevo tratamiento o gasto extra	130
Imagen 15. Confirmación de registro de nuevo tratamiento o gasto extra	131
Imagen 16. Confirmación de abono en nuevo tratamiento o gasto extra registrado.....	131
Imagen 17. Visualización de abono extra en tabla de abonos registrados.....	132
Imagen 18. Mensaje de balance del día actual.....	132
Imagen 19. Ingreso de fecha para balance de día específico	133
Imagen 20. Mensaje de Balance de día específico	133
Imagen 21. Selección de profesional a consultar.....	134
Imagen 22. Mensaje de balance de profesional	135
Imagen 23. Mensaje de balance de profesional en fecha específica.....	135
Imagen 24. Pantalla de balance por especialidad.....	137
Imagen 25. Balance de especialidad escogida	137
Imagen 26. Ingreso de criterios de búsqueda de balance de especialidad	138
Imagen 27. Mensaje de balance de especialidad en fecha específica	138
Imagen 28. Mensaje de balance de especialidad en mes específico	139
Imagen 29. Cierre de sesión de usuario agente.....	139
Imagen 30. Inicio de sesión de administrador	140
Imagen 31. Página principal de administrador	141
Imagen 32. Pantalla de administración de usuarios	142
Imagen 33. Ingreso de datos de usuario.....	143
Imagen 34. Confirmación de usuario registrado en base de datos.....	144
Imagen 35. Selección de agentes para eliminación	145
Imagen 36. Búsqueda de usuarios.....	146
Imagen 37. Eliminación de usuario	147
Imagen 38. Confirmación de usuario eliminado.....	148

Imagen 39. Restablecer contraseña de usuario	149
Imagen 40. Ventana de administración de Profesionales	150
Imagen 41. Confirmación de registro de profesional.....	151
Imagen 42. Eliminación de profesional	152
Imagen 43. Confirmación de eliminación de profesional.....	153
Imagen 44. Ventana de administración de pacientes	153
Imagen 45. Reporte de pacientes	154
Imagen 46. Ventana de laboratorios	154
Imagen 47. Confirmación de registro de laboratorio.....	155
Imagen 48. Eliminación de laboratorio.....	155
Imagen 49. Ventana de administración de tratamientos	156
Imagen 50. Ingreso de datos de tratamientos.....	157
Imagen 51. Confirmación de registro de tratamiento	158
Imagen 52. Modificación de tratamiento	159
Imagen 53. Eliminación de tratamientos	160
Imagen 54. Ventana de facturación	161
Imagen 55. Reporte de Informe General.....	162
Imagen 56. Consulta para Informe Individual	163
Imagen 57. Reporte de Informe Individual.....	163
Imagen 58. Consulta de informe por Factura.....	164
Imagen 59. Reporte de Factura	164
Imagen 60. Selección de factura a eliminar	165
Imagen 61. Confirmación de factura eliminada.....	165
Imagen 62. Ventana de Generar Reportes	166
Imagen 63. Reporte Día Actual	167
Imagen 64. Reporte de Semana Actual.....	167
Imagen 65. Reporte por Mes Actual	168
Imagen 66. Ingreso de fechas para reporte personalizado	168
Imagen 67. Reporte personalizado entre fechas ingresadas.....	169
Imagen 68. Ingreso de datos en ventana de Reporte por profesional en rango de fechas	169
Imagen 69. Reporte por profesional.....	170
Imagen 70. Ingreso de datos en ventana de Reporte por especialidad en rango de fechas.....	170
Imagen 71. Reporte por especialidad.....	171
Imagen 72. Ingreso de fecha en ventana de Registros	171
Imagen 73. Reporte de actividades	172

1. Pantalla Inicial: Inicio De Sesión

Espacio para el ingreso de credenciales de usuario. Se puede acceder haciendo clic en el botón “Ingresar” o presionando la tecla *Enter* del teclado.



Vital Plus

Usuario: Ingresar usuario

Contraseña: Ingresar contraseña

Botón para ingresar

Imagen 53. Pantalla de inicio de sesión

2. Espacio De Agente

2.1. Pantalla Principal.

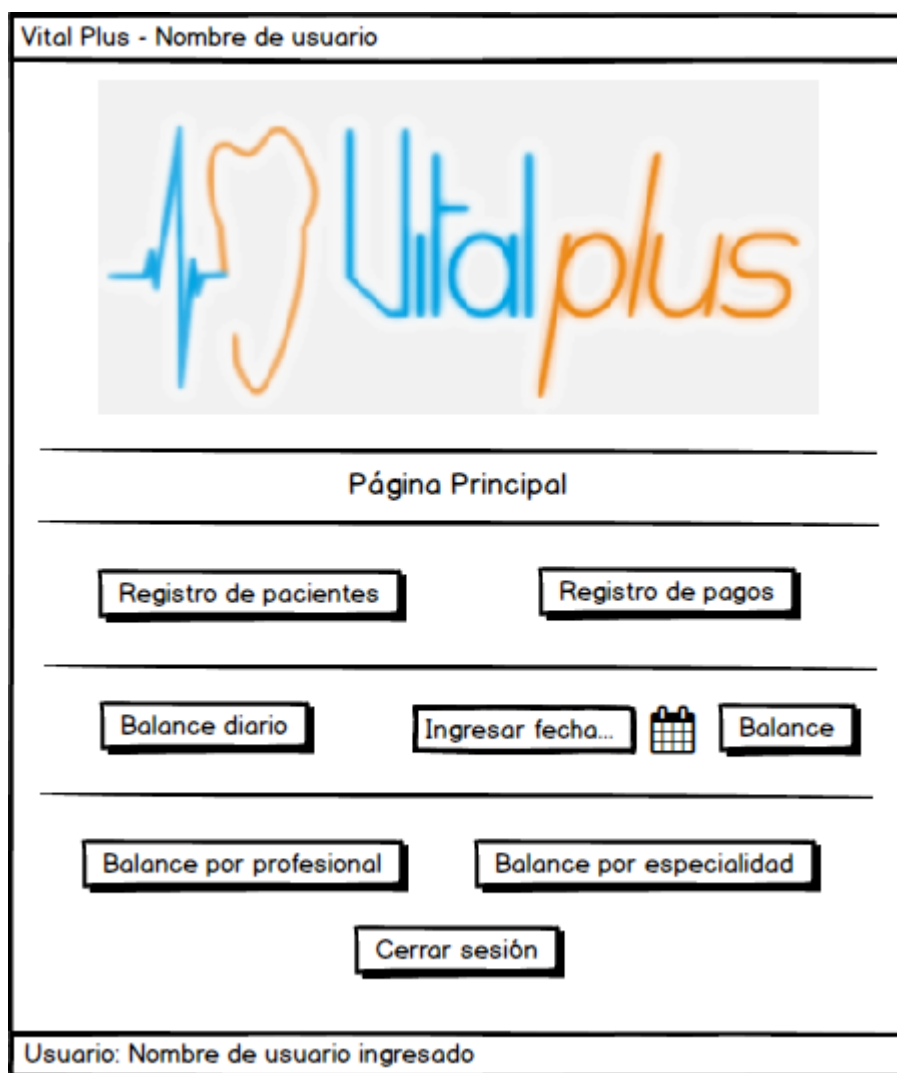



Imagen 54. Pantalla de inicio de sesión de agentes

2.2. Registro De Pacientes.

Vital Plus - Nombre de usuario



Registro de pacientes

Nombre: Documento:

Dirección: Teléfono: Edad:

Código: Nombre: Descuento:

Descripción: Costo:

Usuario: Nombre de usuario ingresado

Imagen 55. Registro de pacientes

2.3. Registro De Datos.

Se ingresan los datos necesarios en las cajas de texto, el sistema convalida y confirma el registro o no de los datos.

Vital Plus - Nombre de usuario

Registro de pacientes

Nombre: Documento:

Dirección: Edad:

Código: Ingresar posible descuento para paciente...

Descripción: costo del tratamiento...

Mensaje

i Paciente registrado exitosamente

Usuario: Nombre de usuario ingresado

Imagen 56. Confirmación de registro de pacientes

2.4. Modificación De Datos.

Se da clic en el botón “Modificar...” desde la ventana anterior y dentro de la tabla del listado de pacientes, se selecciona el paciente al que se quiere modificar su información y luego se presiona el botón “Modificar”.

Vital Plus - Nombre de usuario

Vital plus

Listado de pacientes

Digite No. de documento: Ingresar No. de documento...

Modificar... Borrar Regresar

Usuario: Nombre de usuario ingresado

Imagen 57. Modificación de información de pacientes

Vital Plus - Nombre de usuario

Vital plus

Datos principales de paciente

Documento: Documento registrado de paciente Nombre: Nombre registrado de paciente

Dirección: Dirección registrada de paciente Teléfono: Teléfono registrado de paciente

Edad: Edad registrada de paciente

Guardar Regresar

Usuario: Nombre de usuario ingresado

Imagen 58. Ingreso de nuevos datos de paciente

Vital Plus - Nombre de usuario

Mensaje

Documento: Do do de paciente

Dirección: Direc rado de paciente

Edad: Edad re

Datos Actualizados.

Usuario: Nombre de usuario ingresado

Imagen 59. Confirmación de datos actualizados de pacientes

2.5. Registro De Abonos.

Se ingresa el número de documento que identifica al paciente y se debe dar clic en el botón “Buscar” o presionar *Enter* para cargar la información correspondiente a los abonos registrados por el paciente y algunos datos personales como nombre, edad y tratamiento suscrito.

Vital Plus - Nombre de usuario

Abonos

Documento: Ingresar documento de paciente Nombre: Nombre registrado de paciente Edad: Edad registrada de paciente Tratamiento: Tratamiento registrado de paciente

Código de observ: Ingresar código de nuevo tratamiento o gasto extra. Nombre: Ingresar nombre de nuevo tratamiento o gasto extra. Tipo: Seleccione tipo de observación. Descripción: Ingrese descripción de nuevo tratamiento o gasto extra. Descuento: Ingrese posible descuento de nuevo tratamiento o gasto extra.

Valor: Ingresar valor de abono. Tratamiento: Seleccione tratamientos relacionados a paciente. Info: Profesional: Seleccione entre profesionales disponibles. Método de pago: Seleccione método de pago.

Usuario: Nombre de usuario ingresado

Imagen 60. Pantalla de registro de abonos

2.6. Ingreso De Datos De Abono.

Si el usuario desea conocer información adicional de los tratamientos del paciente, debe seleccionar el tratamiento desde el menú desplegable y luego dar clic en el botón “Info”.

The screenshot shows the 'Abonos' form in the Vital Plus system. A modal window titled 'Info' is open, displaying the following information:

- Descripción:** Descripción del tratamiento elegido
- Costo:** Costo del tratamiento elegido
- Profesional:** Profesional asignado a paciente
- Saldo pendiente:** Saldo pendiente del paciente sobre el tratamiento elegido

The modal window has an 'Aceptar' button at the bottom. The background form contains various input fields and dropdown menus for patient information, treatment selection, and payment method.

Imagen 61. Información de registro escogido

El usuario debe ingresar el monto del abono, seleccionar el tratamiento en el que se va a registrar el abono, el profesional a cargo y el método de pago (efectivo, tarjeta de débito o tarjeta de crédito) y luego hacer clic en el botón “Pagar”.

The screenshot shows the 'Abonos' form in the Vital Plus system. A modal window titled 'Abono registrado' is open, displaying the following information:

- Abono registrado**

The modal window has an 'Aceptar' button at the bottom. The background form contains various input fields and dropdown menus for patient information, treatment selection, and payment method.

Imagen 62. Confirmación de registro de abono

El sistema carga la información de los abonos registrados por parte del paciente en la tabla de abajo, también aparece el valor pendiente del paciente en el tratamiento que abonó y el monto total de la transacción aplicando descuentos por el método de pago o por algún descuento adicional que se pudo ingresar en la caja de texto de “Descuento (%)”.

Vital Plus - Nombre de usuario

Abonos

Document: Ingresar documento de paciente. Nombre: Nombre registrado de paciente. Edad: Edad registrada de paciente. Tratamiento: Tratamiento registrado de paciente

Código de observación: Ingresar código de nuevo tratamiento o gasto extra. Nombre: Ingresar nombre de nuevo tratamiento o gasto extra. Tipo: Seleccione tipo de observación. Descripción: Ingrese descripción de nuevo tratamiento o gasto extra. Descuento: Ingrese posible descuento de nuevo tratamiento o gasto extra.

Valor: Ingresar valor de abono. Tratamiento: Seleccione tratamientos relacionados a paciente. Info: Profesional: Seleccione entre profesionales disponibles. Método de pago: Seleccione método de pago.

Abonos mostrados en tabla

Usuario: Nombre de usuario ingresado

Imagen 63. Abono visualizado en tabla de abonos registrados

A partir de allí, el usuario puede ingresar los siguientes abonos por medio del botón

“Abono”.

Vital Plus - Nombre de usuario

Abonos

Document: Ingresar documento de paciente. Nombre: Nombre registrado de paciente. Edad: Edad registrada de paciente. Tratamiento: Tratamiento registrado de paciente

Código de observación: Ingresar código de nuevo tratamiento o gasto extra. Nombre: Ingresar nombre de nuevo tratamiento o gasto extra. Tipo: Seleccione tipo de observación. Descripción: Ingrese descripción de nuevo tratamiento o gasto extra. Descuento: Ingrese posible descuento de nuevo tratamiento o gasto extra.

Valor: Ingresar valor de abono. Tratamiento: Seleccione tratamientos relacionados a paciente. Info: Profesional: Seleccione entre profesionales disponibles. Método de pago: Seleccione método de pago.


Mensaje: Abono registrado.

Cuota inicial y abonos regulares mostrados en tabla

Usuario: Nombre de usuario ingresado

Imagen 64. Confirmación de abono registrado

Vital Plus - Nombre de usuario




Abonos

Documento: Nombre: Edad: Tratamiento:

Código de observ: Nombre: Tipo: Descripción: Descuento:

Valor: Tratamiento: Info: Profesional: Método de pago:

 Cuota inicial y abonos mostrados en tabla


Usuario: Nombre de usuario ingresado

Imagen 65. Abono visualizado en tabla de abonos registrados

2.6.1. Ingreso De Nuevos Tratamientos O Gastos Extra.

Este espacio de un pequeño formulario para ingresar nuevos tratamientos o gastos extra para el paciente ingresado previamente. El usuario debe ingresar los datos esenciales para el ingreso de dichos conceptos y luego hacer clic en el botón “Registrar”.

Vital Plus - Nombre de usuario




Abonos

Documento: Nombre: Edad: Tratamiento:

Código de observ: Nombre: Tipo: Descripción: Descuento:

Valor: Tratamiento: Info: Profesional: Método de pago:

 Cuota inicial y abonos mostrados en tabla

Usuario: Nombre de usuario ingresado

Imagen 66. Ingreso de datos de nuevo tratamiento o gasto extra

Vital Plus - Nombre de usuario

Abonos

Documento: Ingresar documento de paciente. Nombre: Nombre registro de paciente. Mensaje: Mensaje de paciente. Tratamiento: Tratamiento registrado de paciente.

Código de observación: 4566. Nombre: Cargo extra. Descripción: Cargo extra. Descuento: Registrar

Valor: Ingresar valor de abono. Tratamiento: Seleccione tratamientos relacionados. Profesional: Seleccione entre profesionales disponibles. Método de pago: Seleccione método de pago

Tratamiento 1, Tratamiento 2, ..., Tratamiento n. Profesional 1, Profesional 2, ..., Profesional n.

Registro exitoso

Aceptar

Cuota inicial y abonos mostrados en tabla

Cuota Inicial Abono Regresar

Usuario: Nombre de usuario ingresado

Imagen 67. Confirmación de registro de nuevo tratamiento o gasto extra

Vital Plus - Nombre de usuario

Abonos

Documento: Ingresar documento de paciente. Nombre: Nombre registro de paciente. Mensaje: Mensaje de paciente. Tratamiento: Tratamiento registrado de paciente.

Código de observación: 4566. Nombre: Cargo extra. Descripción: Cargo extra. Descuento: Registrar

Valor: Ingresar valor de abono. Tratamiento: Seleccione tratamientos relacionados. Profesional: Seleccione entre profesionales disponibles. Método de pago: Seleccione método de pago

Tratamiento 1, Tratamiento 2, ..., Tratamiento n. Profesional 1, Profesional 2, ..., Profesional n.

Abono registrado

Aceptar

Cuota inicial y abonos mostrados en tabla

Cuota Inicial Abono Regresar

Usuario: Nombre de usuario ingresado

Imagen 68. Confirmación de abono en nuevo tratamiento o gasto extra registrado

Imagen 69. Visualización de abono extra en tabla de abonos registrados

2.6.2. Consulta De Balance Económico.

2.6.2.1. Consulta Por Día Actual.

Para volver a la página principal, el usuario debe hacer clic en el botón “Regresar”. Una vez allí, el usuario agente puede consultar el monto económico acumulado de abonos en el día que va transcurriendo a modo informativo mediante el botón “Balance Diario”.

Imagen 70. Mensaje de balance del día actual

2.6.2.2. Consulta Por Día Específico.

Desde la misma página, se puede ingresar una fecha, ya sea manualmente o usando el botón de calendario que se encuentra al lado de la caja de texto, para consultar el balance o monto económico de abonos registrados mediante el botón “Balance”.

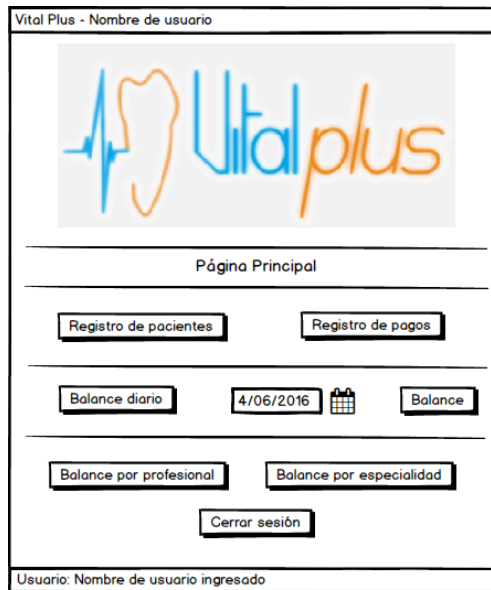


Imagen 71. Ingreso de fecha para balance de día específico

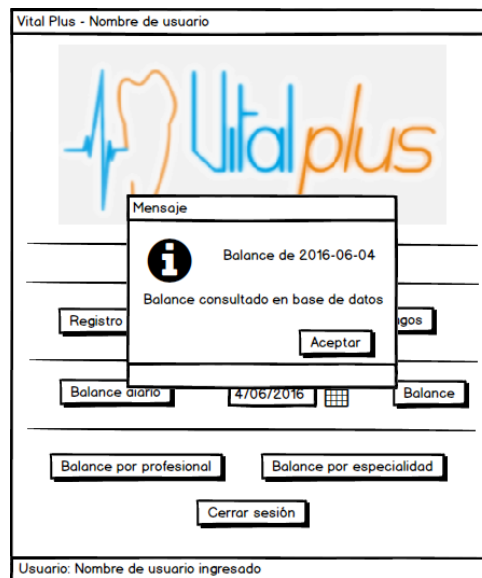
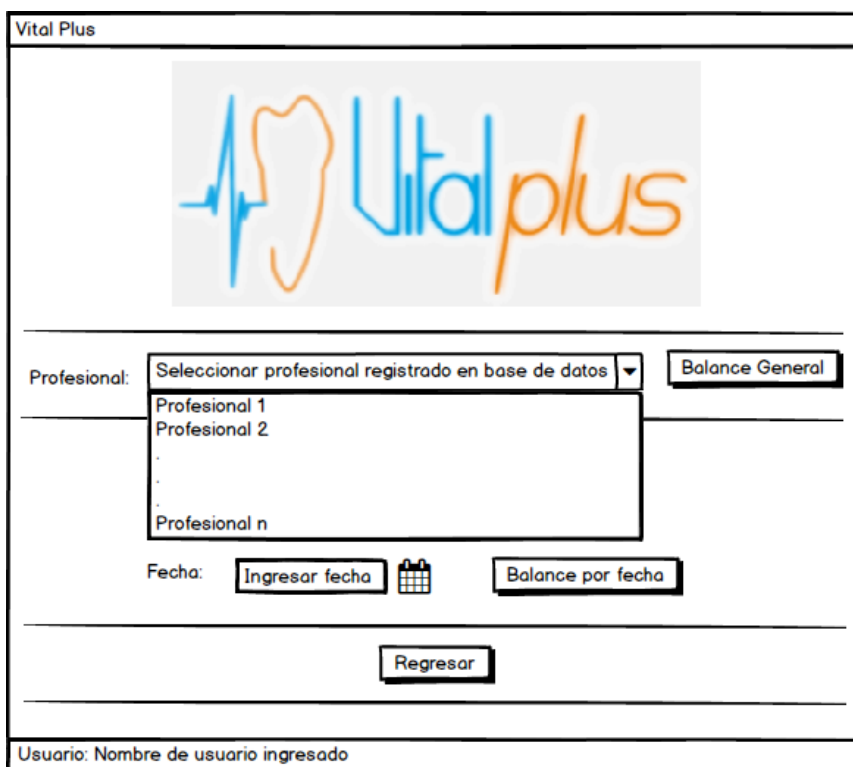


Imagen 72. Mensaje de Balance de día específico

2.6.2.3. Consulta Por Profesional.

Desde el botón “Balance por profesional”, el usuario accede a una nueva ventana para consultar los balances económicos por concepto de abonos de los pacientes que se encuentran a cargo de un profesional específico. El usuario debe elegir entre los profesionales registrados en base de datos desde la lista desplegable y luego hacer clic en el botón “Balance General” y así la aplicación le mostrará la información solicitada.



The screenshot shows a web application window titled "Vital Plus". At the top center is a logo featuring a blue heart rate line, an orange tooth outline, and the text "Vital plus" in blue and orange. Below the logo, there is a form with the following elements:

- A label "Profesional:" followed by a dropdown menu with the text "Seleccionar profesional registrado en base de datos".
- A "Balance General" button to the right of the dropdown.
- A list box containing the items "Profesional 1", "Profesional 2", a vertical ellipsis, and "Profesional n".
- A label "Fecha:" followed by an "Ingresar fecha" button with a calendar icon and a "Balance por fecha" button.
- A "Regresar" button at the bottom center.
- A footer area with the text "Usuario: Nombre de usuario ingresado".

Imagen 73. Selección de profesional a consultar

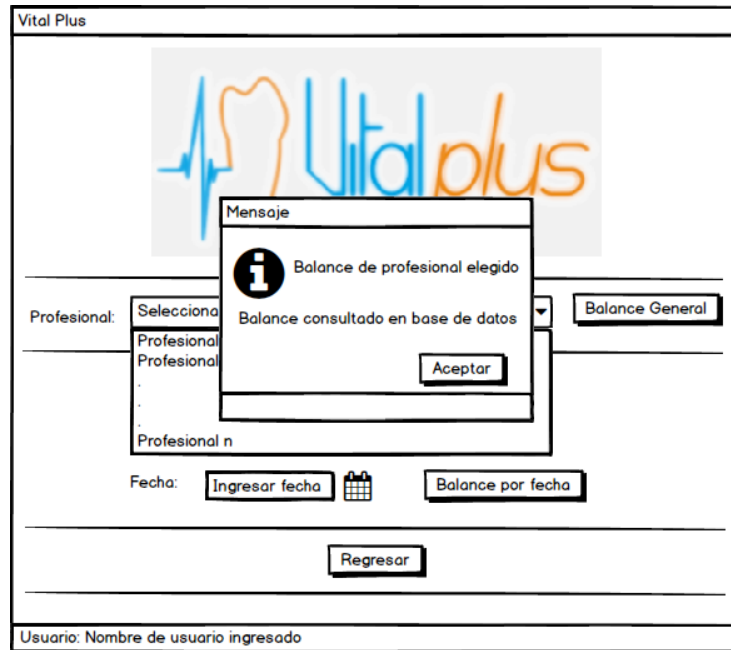


Imagen 74. Mensaje de balance de profesional

También es posible consultar los abonos registrados bajo el nombre de un profesional en una fecha específica. El usuario ingresa la fecha a consultar y luego hace clic en el botón “Balance por fecha” para obtener la información.

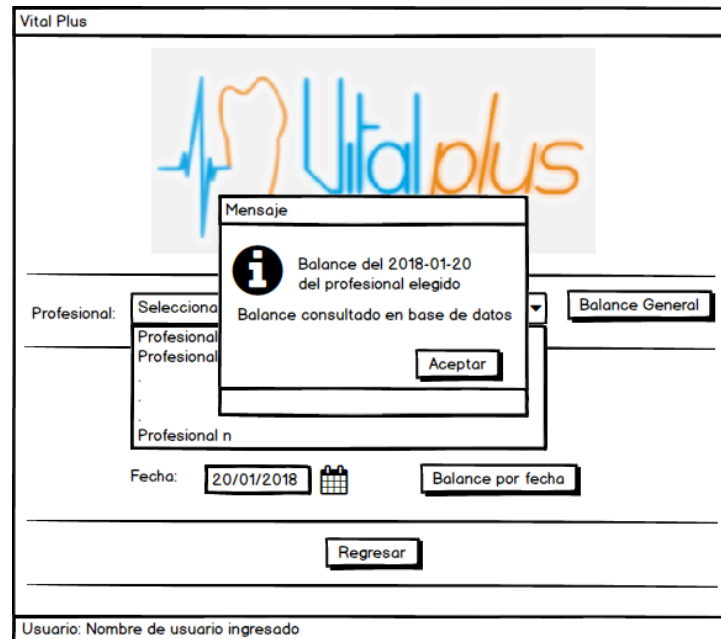


Imagen 75. Mensaje de balance de profesional en fecha específica

2.6.2.4. *Consulta Por Especialidad.*

Para volver a la página principal de agentes, se dispone del botón “Regresar” y así puede hacer clic al botón “Balance por especialidad”. Cuando el agente hace clic en este botón, se despliega una ventana que muestra una lista desplegable con las especialidades de los profesionales inscritos en la empresa. Desde esta ventana se realizan consultas usando varios criterios de búsqueda y haciendo clic en los botones:

2.6.2.4.1. *Balance.*

El usuario escoge la especialidad a consultar y hace clic en este botón para visualizar su balance total.


2.6.2.4.2. *Balance por fecha.*

Con la especialidad escogida, el usuario debe ingresar una fecha específica para consultar.

2.6.2.4.3. *Balance por mes.*

El usuario debe elegir en las listas desplegables el mes y el año de consulta para obtener el balance general de la especialidad durante un mes calendario.

Vital Plus



Especialidad:
 Especialidad 1
 Especialidad 2
 .
 .
 Especialidad n


Fecha:

Mes:

Usuario: Nombre de usuario ingresado

Imagen 76. Pantalla de balance por especialidad

Vital Plus



Mensaje

i Balance de especialidad elegida
 Balance consultado en base de datos

Especialidad:
 Espe
 Espe
 .
 .
 Especialidad n


Fecha:

Mes:

Usuario: Nombre de usuario ingresado

Imagen 77. Balance de especialidad escogida

Vital Plus



Especialidad:

Especialidad 1
Especialidad 2
.
.
Especialidad n


Fecha:

Mes:

Usuario: Nombre de usuario ingresado

Imagen 78. Ingreso de criterios de búsqueda de balance de especialidad

Vital Plus



Mensaje

i Balance en 2018-01-20 de especialidad elegida
Balance consultado en base de datos

Especialidad:

Especialidad 1
Especialidad 2
.
.
Especialidad n

Fecha:

Mes:

Usuario: Nombre de usuario ingresado

Imagen 79. Mensaje de balance de especialidad en fecha específica

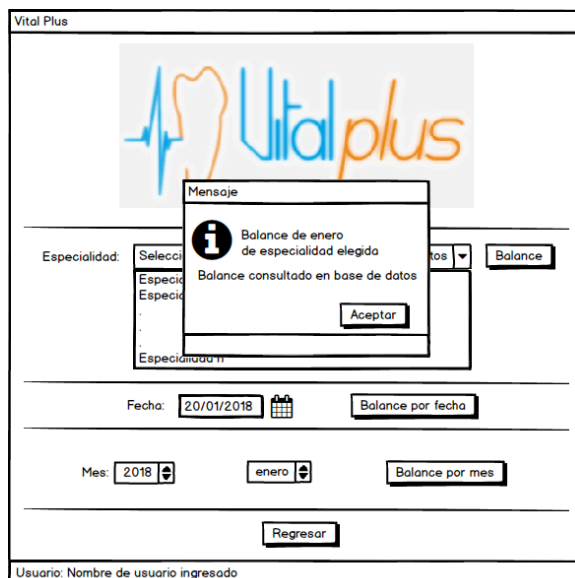


Imagen 80. Mensaje de balance de especialidad en mes específico

2.6.3. Cerrar Sesión

Se dispone del botón “Regresar” para volver a la página principal del espacio de agente. Para terminar o cerrar sesión de usuario, el agente puede hacer clic en el botón “Cerrar sesión” y así registra su final de actividades en la base de datos.

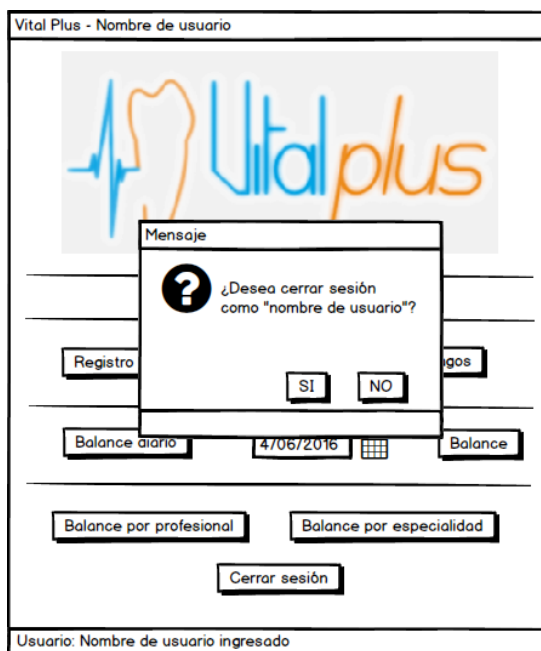
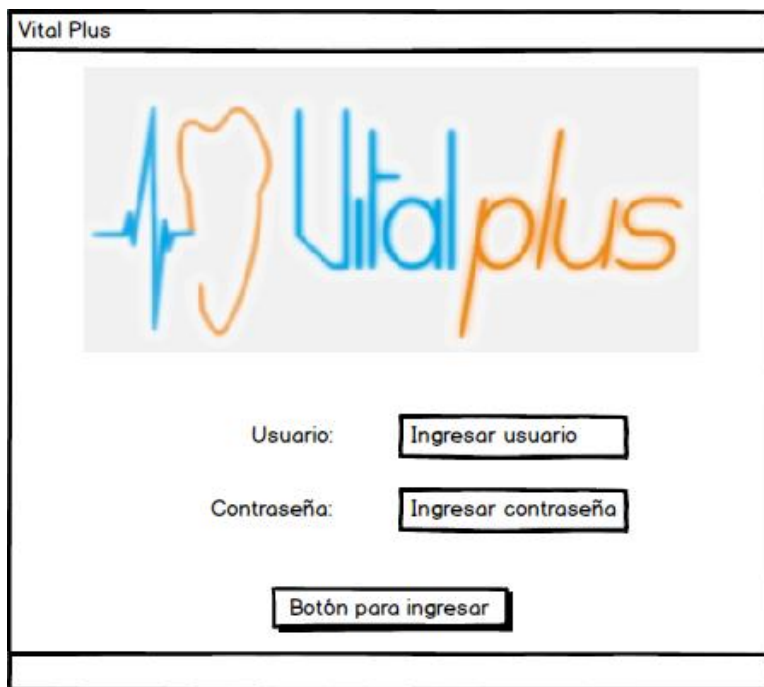


Imagen 81. Cierre de sesión de usuario agente

3. Espacio De Administrador

Desde la pantalla inicial, el administrador ingresa sus credenciales y accede a su página principal.



The image shows a login interface for 'Vital Plus'. At the top left, the text 'Vital Plus' is displayed. Below this is a large graphic featuring a blue ECG line, an orange tooth outline, and the text 'Vital plus' in blue and orange. Underneath the graphic are two input fields: 'Usuario:' followed by a button labeled 'Ingresar usuario', and 'Contraseña:' followed by a button labeled 'Ingresar contraseña'. At the bottom center, there is a button labeled 'Botón para ingresar'.

Imagen 82. Inicio de sesión de administrador

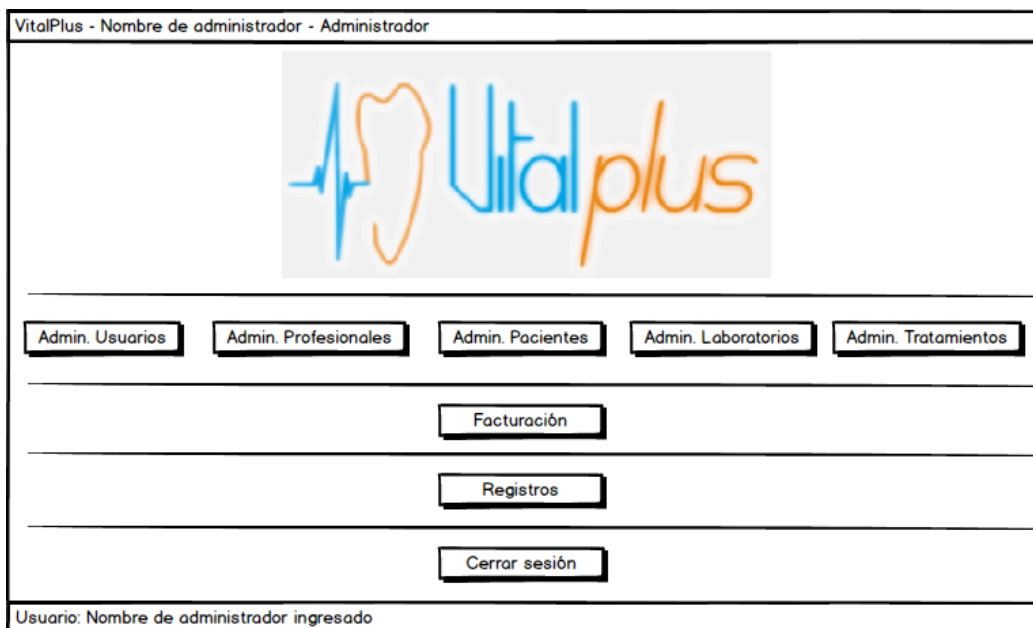



Imagen 83. Página principal de administrador

3.1.1. Admin. Usuarios.

Al hacer clic en este botón, se despliega el menú de opciones relacionadas con los usuarios que usan la aplicación. Desde allí el administrador puede registrar, modificar y eliminar información los usuarios que usan esta aplicación. Para ejecutar el registro de los datos ingresados en la caja de texto se debe hacer clic en el botón “Registrar”. Además, se dispone del botón “Generar Contraseña”, el cual crea contraseñas aleatorias de 4 dígitos que se le asignará al usuario en primera instancia para que pueda acceder a la aplicación.

VitalPlus - Nombre de administrador - Administrador



Usuario: Contraseña:

ID:

Tipo:

Administrador
Agente

Lista de usuarios


Digite código:

Tabla de la lista de usuarios registrados en base de datos

Usuario: Nombre de administrador ingresado

Imagen 84. Pantalla de administración de usuarios

VitalPlus - Nombre de administrador - Administrador



Usuario: Contraseña:

ID:

Tipo: ▼

Lista de usuarios

Digite código:





Tabla de la lista de usuarios registrados en base de datos

Usuario: Nombre de administrador ingresado

Imagen 85. Ingreso de datos de usuario

VitalPlus - Nombre de administrador - Administrador




Usuario: Contraseña:

ID: Contraseña

Tipo: Contraseña

Mensaje

 Nuevo usuario registrado.

Lista de usuarios

Digite código:

Tabla de la lista de usuarios registrados en base de datos


Usuario: Nombre de administrador ingresado

Imagen 86. Confirmación de usuario registrado en base de datos

Para la eliminación de usuarios, esta ventana permite realizar una búsqueda por medio del código del usuario y seleccionándolo desde la tabla que visualiza los usuarios registrados en base de datos se puede proceder a hacer clic en el botón “Eliminar” para que la aplicación complete la acción. Para el caso del botón “Modificar”, solamente se debe seleccionar el usuario desde la

tabla y las cajas de texto cargarán los datos ID, usuario, y tipo para que la contraseña se pueda actualizar bajo la autorización del usuario administrador.

VitalPlus - Nombre de administrador - Administrador



Usuario: Contraseña:

ID:

Tipo:

Lista de usuarios


Digite código:

Tabla de la lista de usuarios registrados en base de datos.
Datos de campos de texto son cargados a partir del usuario seleccionado en la tabla.

Usuario: Nombre de administrador ingresado

Imagen 87. Selección de agentes para eliminación

VitalPlus - Nombre de administrador - Administrador



Usuario: Contraseña:

ID:

Tipo:

Administrador
Agente

Lista de usuarios

Digite código:





Tabla de la lista de usuarios registrados en base de datos.
Se mostrarán los usuarios que cumplan con el criterio de búsqueda ingresado.

Usuario: Nombre de administrador ingresado

Imagen 88. Búsqueda de usuarios

VitalPlus - Nombre de administrador - Administrador



Usuario: Contraseña:

ID: Contraseña

Tipo:

Adminis
Agente

Eliminar usuario

¿Desea eliminar este usuario?

SI NO

Digite código:

Tabla de la lista de usuarios registrados en base de datos.

Se mostrarán los usuarios que cumplan con el criterio de búsqueda ingresado.


Se selecciona el usuario a eliminar.

Registrar Modificar... Eliminar Regresar

Usuario: Nombre de administrador ingresado

Imagen 89. Eliminación de usuario

VitalPlus - Nombre de administrador - Administrador



Usuario: Contraseña:

ID: Mensaje

Tipo:

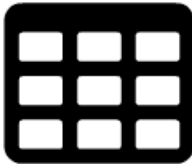
Administrador
Agente

i Usuario eliminado

Aceptar

Lista de usuarios

Digite código:

 Tabla de la lista de usuarios registrados en base de datos.

Se mostrarán los usuarios que cumplan con el criterio de búsqueda ingresado.

Se selecciona el usuario a eliminar.


Usuario: Nombre de administrador ingresado

Imagen 90. Confirmación de usuario eliminado

3.1.2. Restablecer Contraseña.

Por medio del botón “Restablecer contraseña”, se despliega la ventana dedicada a la modificación de la contraseña de los usuarios desde la vista del administrador y que debe ingresar el usuario que desea realizar dicha operación. La aplicación realiza las validaciones correspondientes a lo que se vaya ingresando en los campos de texto.

VitalPlus - Nombre de administrador - Administrador



Restablecer contraseña

Usuario:

Ingrese contraseña:

Verifique contraseña:

Usuario: Nombre de administrador ingresado

Imagen 91. Restablecer contraseña de usuario


3.1.3. Admin. Profesionales.

El botón “Regresar” permite volver a la pantalla principal del administrador para poder acceder al espacio de manipulación de información relacionada a los profesionales de la empresa mediante este botón.

3.1.3.1. Registro De Profesionales.

Para registrar un nuevo profesional, se deben ingresar sus datos en las cajas de texto y seleccionar el laboratorio relacionado desde la lista desplegable y luego hacer clic en el botón “Guardar”.

VitalPlus - Nombre de administrador - Administrador



Nombre: Cédula/Código:

Especialidad: Laboratorio: ▾

- Laboratorio 1
- Laboratorio 2
- Laboratorio n

Lista de profesionales

Digite criterio:

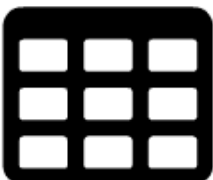



Tabla de la lista de profesionales registrados en base de datos.

Usuario: Nombre de administrador ingresado

Imagen 92. Ventana de administración de Profesionales

VitalPlus - Nombre de administrador - Administrador



Nombre: Mensaje

Especialidad: Profesional registrado exitosamente

Lista de profesionales

Digite criterio:

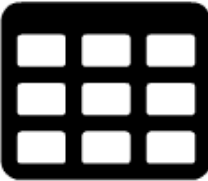


Tabla de la lista de profesionales registrados en base de datos.

Usuario: Nombre de administrador ingresado

Imagen 93. Confirmación de registro de profesional


3.1.3.2. Eliminación De Profesionales.

Para eliminar un profesional de la base de datos, se debe seleccionar al profesional desde la tabla de datos desplegada al final de la ventana y luego hacer clic en el botón “Borrar”.

También se puede buscar al profesional a eliminar por código o nombre desde el buscador

disponible en esta ventana. La aplicación confirma la búsqueda y eliminación del profesional, respectivamente.

VitalPlus - Nombre de administrador - Administrador



Nombre: Mensaje

Especialidad:

¿Desea borrar a este profesional?

Lista de profesionales

Digite criterio:




Tabla de la lista de profesionales registrados en base de datos.

Usuario: Nombre de administrador ingresado

Imagen 94. Eliminación de profesional

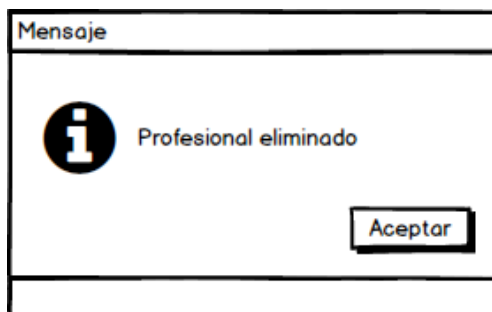


Imagen 95. Confirmación de eliminación de profesional

3.1.4. Admin. Pacientes.

En este espacio sólo se exporta la información almacenada en base de datos relacionada con los pacientes suscritos a la empresa mediante el botón “Exportar”.

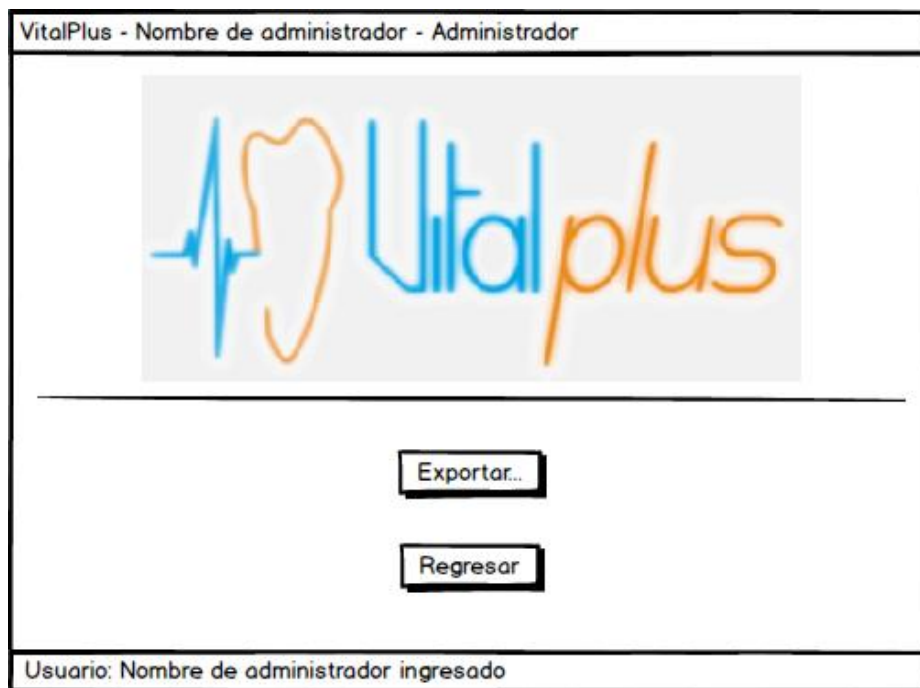


Imagen 96. Ventana de administración de pacientes



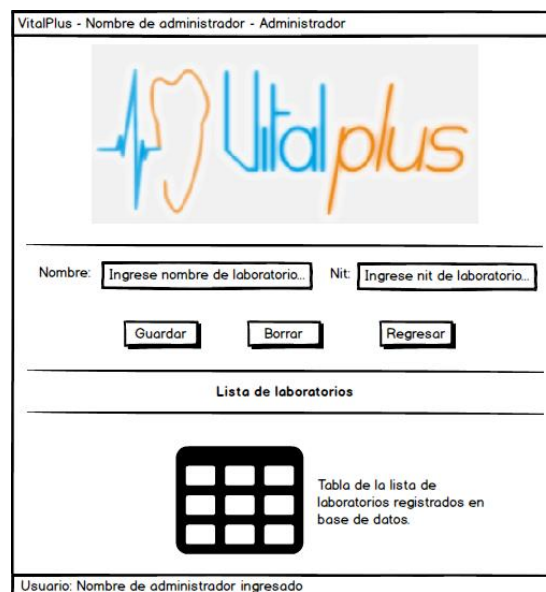
Nombre	Telefono	Dirección	Edad	Tratamiento	Costo	Profesional	Remitido
112							
Sandra Urbe	1234567	qrfrqrg5fhthzdz	24	Ortodoncia	2700000	Francisco Ponce	Ninguno
123							
Daniel Pertuz	3057232166	Calle 1 No. 2 - 3	23	General	3000000	Gloria Salazar Arboleda	Francisco Ponce
334							
Oscar Martinez	982734878	sdcscdeer	28	Ortodoncia	500000	María Bermúdez	Ninguno
567							
María López	3794791	Cra. 4 No. 5 - 6	25	Endodoncia	1500000	Paola González	Ninguno
678							
Fabio Torres	3778899	Calle 1 No. 2 - 3	25	Endodoncia	1500000	Gloria Salazar Arboleda	María Bermúdez

Página 1 de 2


Imagen 97. Reporte de pacientes

3.1.5. Admin. Laboratorios.

Desde este espacio, se puede ingresar, consultar y eliminar los datos pertinentes a los laboratorios aliados a la empresa mediante los botones “Guardar” y “Borrar” seleccionando el laboratorio de la tabla de datos almacenados.




VitalPlus - Nombre de administrador - Administrador



Nombre: Nit:

Lista de laboratorios

 Tabla de la lista de laboratorios registrados en base de datos.

Usuario: Nombre de administrador ingresado

Imagen 98. Ventana de laboratorios

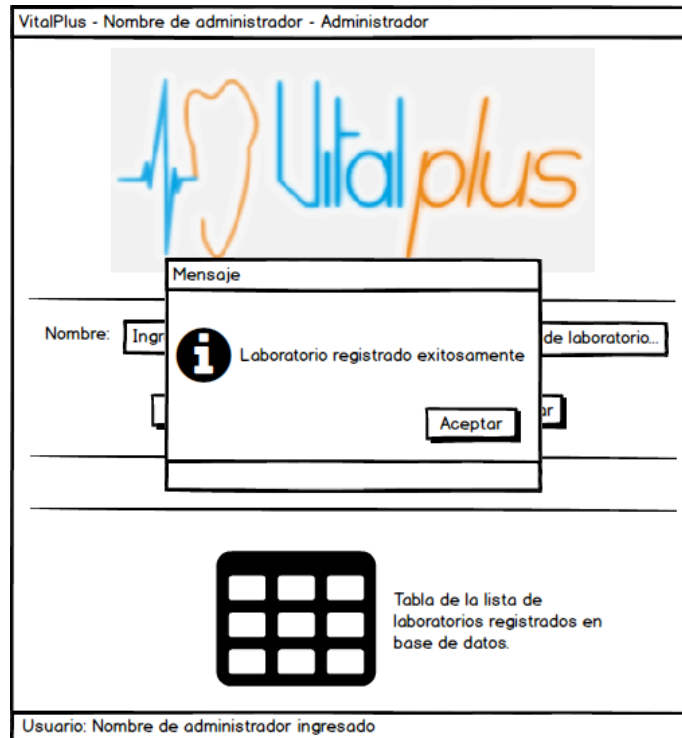


Imagen 99. Confirmación de registro de laboratorio

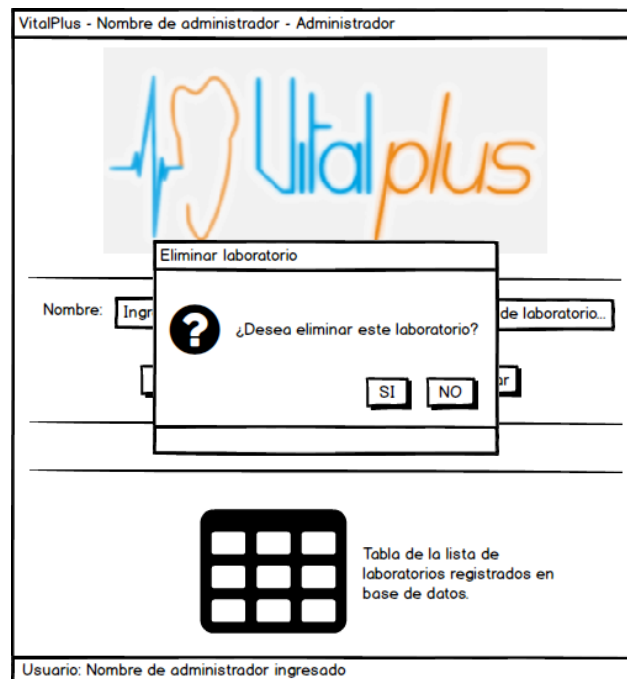



Imagen 100. Eliminación de laboratorio

3.1.6. Admin. Tratamientos.

Desde la ventana de laboratorios haciendo clic en el botón “Regresar”, se devuelve a la pantalla principal y así accionar el botón “Admin. Tratamientos” para llegar a la ventana correspondiente a la manipulación de los datos de los tratamientos que ofrece la empresa a sus pacientes.

VitalPlus - Nombre de administrador - Administrador



Nombre: Código: Costo:

Documento de paciente: Tipo:

Descripción:

Lista de tratamientos




Tabla de la lista de tratamientos registrados en base de datos.


Usuario: Nombre de administrador ingresado

Imagen 101. Ventana de administración de tratamientos

3.1.6.1. Registro De Tratamientos.

Desde esta ventana se puede ingresar la información correspondiente a los tratamientos utilizando las cajas de texto y la lista desplegable. Para registrar los datos ingresados, se debe hacer clic en el botón “Guardar”.

VitalPlus - Nombre de administrador - Administrador



Nombre: Código: Costo:

Documento de paciente: Tipo:

Descripción:

Lista de tratamientos

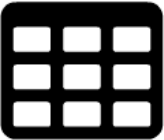



Tabla de la lista de tratamientos registrados en base de datos.

Usuario: Nombre de administrador ingresado

Imagen 102. Ingreso de datos de tratamientos


VitalPlus - Nombre de administrador - Administrador



Nombre: C

Documento de paciente:

Mensaje

 Tratamiento registrado exitosamente

Descripción:

Odontopediatría específica

Lista de tratamientos




Tabla de la lista de tratamientos registrados en base de datos.


Usuario: Nombre de administrador ingresado

Imagen 103. Confirmación de registro de tratamiento

3.1.6.2. Modificación De Tratamientos.

Para modificar los tratamientos registrados, se debe seleccionar el tratamiento desde la tabla ubicada en la parte inferior de la ventana y la aplicación cargará los datos en las cajas de texto y en la lista desplegable para realizar las respectivas modificaciones de los datos, además de mostrar más detalles de los tratamientos.

VitalPlus - Nombre de administrador - Administrador




Nombre: Código: Costo:

Documento de paciente: Tipo:

Descripción:

Lista de tratamientos

 Tabla de la lista de tratamientos registrados en base de datos.
Los datos son cargados a las cajas de texto a partir de la selección del tratamiento en la tabla


Usuario: Nombre de administrador ingresado

Imagen 104. Modificación de tratamiento

3.1.6.3. Eliminación De Tratamientos.

A partir de la selección del tratamiento desde la tabla de datos de la ventana, se puede hacer clic en el botón “Borrar” para realizar la eliminación del tratamiento seleccionado de la base de datos. Mediante el buscador que posee la ventana también se puede buscar cualquier tratamiento por su código o nombre.

VitalPlus - Nombre de administrador - Administrador



Nombre:

Documento de paciente:

Descripción:

Paga 3 rejillas (Cargado de base de datos)

Eliminar tratamiento

Costo:

¿Desea eliminar este tratamiento?

Lista de tratamientos

Tabla de la lista de tratamientos registrados en base de datos.

Los datos son cargados a las cajas de texto a partir de la selección del tratamiento en la tabla


Usuario: Nombre de administrador ingresado

Imagen 105. Eliminación de tratamientos

3.1.6.4. Facturación.

En este espacio se muestran opciones para visualizar facturaciones (Historias Clínicas) desde distintos criterios de búsqueda.

VitalPlus - Nombre de administrador - Administrador



Historia Clínica de Pacientes

Digite No.




Tabla de la lista de facturas registradas en base de datos.

Usuario: Nombre de administrador ingresado

Imagen 106. Ventana de facturación

3.2. Informes

3.2.1. Informe General.

Al hacer clic en este botón, la aplicación exporta todas las facturas registradas en la base de datos a un reporte *iReport* especializado sin necesidad de ingresar cualquier criterio de búsqueda.

VITAL PLUS
FACTURACIÓN

Valor	Tipo	Método de pago	Fecha	Procedimiento	Documento	Profesional	Saldo
VP0000001							
30000.0	Cuota Inicial	Efectivo	4/03/16 11:11 AM	Ortodoncia	123	Gloria Salazar Arboleda	2970000
VP0000002							
35000.0	Regular	Efectivo	3/06/16 11:11 AM	Ortodoncia	123	Gloria Salazar Arboleda	2935000
VP0000003							
35000.0	Regular	Efectivo	4/06/16 11:11 AM	Ortodoncia	123	Gloria Salazar Arboleda	2900000
VP0000004							
35000.0	Regular	Efectivo	4/06/16 11:12 AM	Ortodoncia	123	Gloria Salazar Arboleda	2865000
VP0000005							
39978.0	Cuota Inicial	Tarjeta de Crédito	4/06/16 11:14 AM	Endodoncia	567	Paola González	1460022
VP0000006							
39978.0	Regular	Tarjeta de Crédito	7/06/16 11:14 AM	Endodoncia	567	Paola González	1420044

domingo 21 enero 2018 Page 1 of 8


Pagina 1 de 8

Imagen 107. Reporte de Informe General

3.2.2. Informe Individual.


Para este botón se debe ingresar el documento del paciente en la caja de texto correspondiente y luego hacer clic en el botón “Buscar por Documento”. La aplicación carga los datos en la tabla ubicada en la parte inferior de la ventana y al hacer clic en “Informe Individual” se exportarán las facturas registradas a nombre del paciente ingresado.

VitalPlus - Nombre de administrador - Administrador



Historia Clínica de Pacientes

Digite No.

 Tabla de la lista de facturas registradas en base de datos.
Facturas relacionadas con el número de documento ingresado.

Usuario: Nombre de administrador ingresado

Imagen 108. Consulta para Informe Individual

Informe de Silvia Acosta

VITAL PLUS
FACTURACIÓN

Valor	Tipo	Método de pago	Fecha	Procedimiento	Documento	Profesional	Saldo
VP0000059							
35000.0	Cuota Inicial	Efectivo	20/01/18 11:50 AM	Endodoncia	22434241	Paola González	1265000
VP0000060							
28000.0	Regular	Efectivo	20/01/18 12:18 PM	Endodoncia	22434241	Paola González	1237000
VP0000061							
30000.0	Regular	Efectivo	20/01/18 12:44 PM	Cargo extra	22434241	Paola González	0

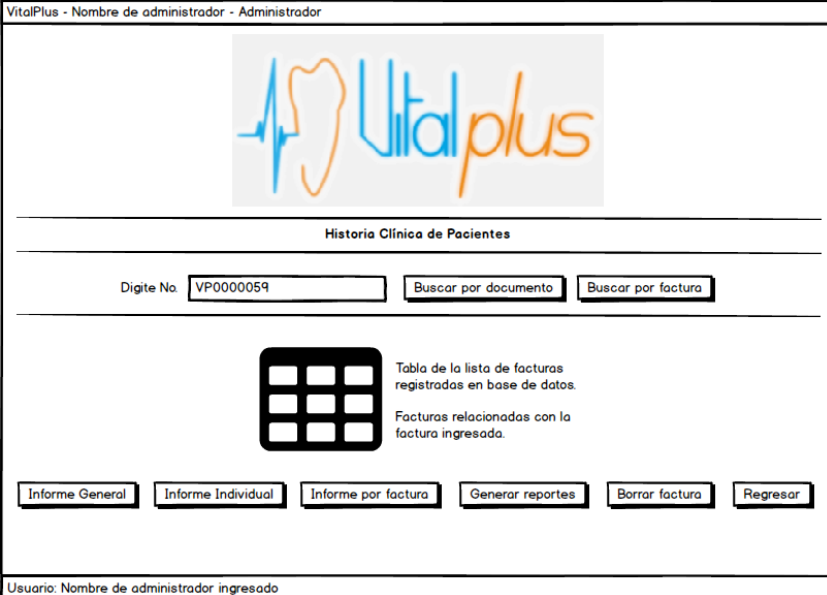
domingo 21 enero 2018 Page 1 of 1

Página 1 de 1

Imagen 109. Reporte de Informe Individual

3.2.3. Informe Por Factura.

Desde la caja de texto el usuario administrador debe ingresar el código de factura a consultar y hacer clic en el botón “Buscar por Factura” para mostrar la información de la factura. Para exportar la factura, el usuario debe hacer clic en la fila de la tabla en donde se encuentra la factura y finalmente hacer clic en el botón “Informe por Factura”.



VitalPlus - Nombre de administrador - Administrador

Historia Clínica de Pacientes

Digite No:

Tabla de la lista de facturas registradas en base de datos.
Facturas relacionadas con la factura ingresada.

Usuario: Nombre de administrador ingresado

Imagen 110. Consulta de informe por Factura



Informe de Factura VP0000059

VITAL PLUS
FACTURACIÓN

Valor	Tipo	Método de pago	Fecha	Procedimiento	Documento	Profesional	Saldo
VP0000059							
35000.0	Cuota Inicial	Efectivo	20/01/18 11:50 AM	Endodoncia	22434241	Paola González	1265000

Página 1 de 1

Imagen 111. Reporte de Factura

3.2.4. Borrar Factura.

Para efectuar las acciones de este botón, se debe seleccionar la fila antes de hacer clic. La aplicación confirmará el cumplimiento de las acciones.

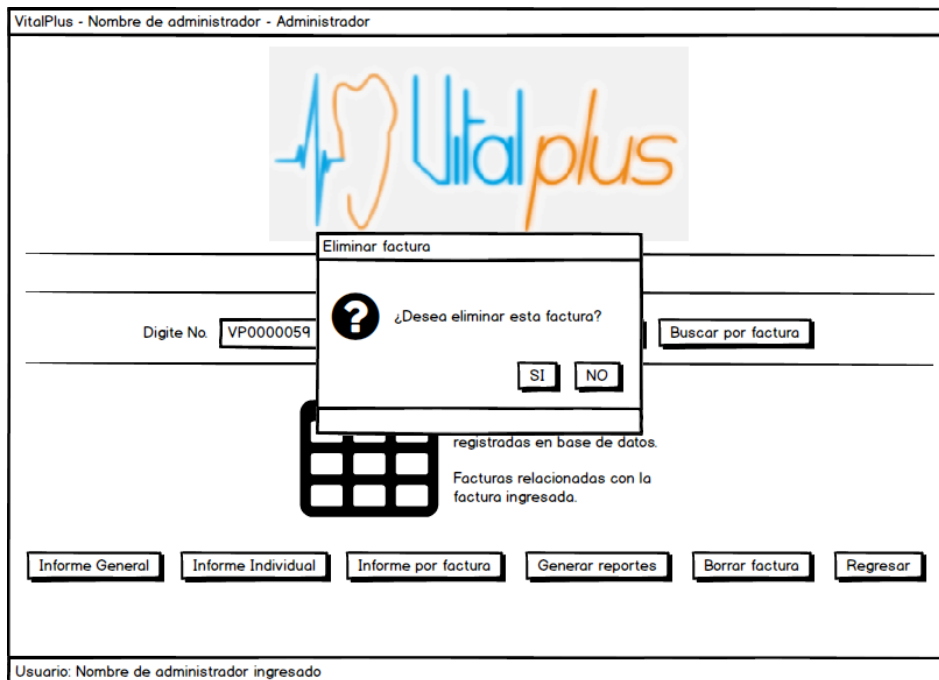


Imagen 112. Selección de factura a eliminar

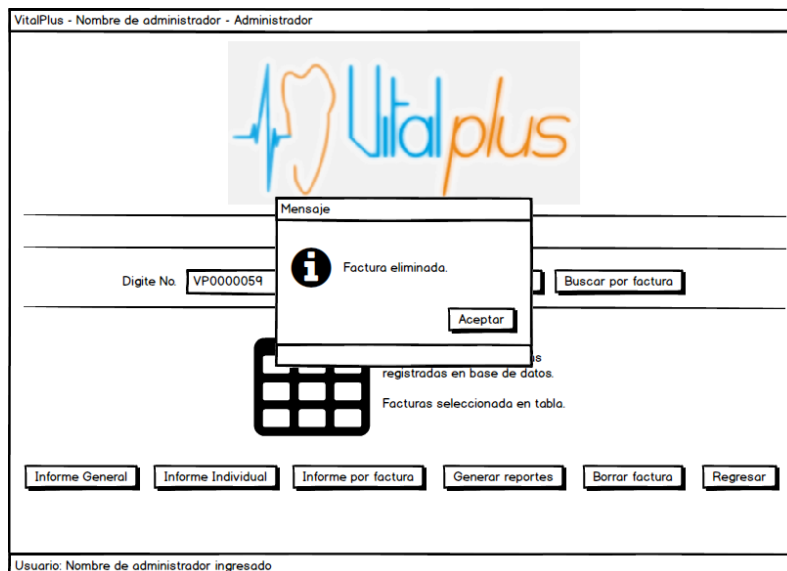



Imagen 113. Confirmación de factura eliminada

3.2.5. Generar Reportes.

Al hacer clic en el botón “Generar Reportes”, se abre una nueva ventana para exportar los reportes bajo diferentes criterios de búsqueda.



Vital Plus - Nombre de usuario



Específico

Hoy Semana Mes

Personalizado

Desde:  Hasta: 

Reporte por profesional Reporte por especialidad

Regresar

Usuario: Nombre de usuario ingresado

Imagen 114. Ventana de Generar Reportes

Con los botones de la sección “Específico”, se consultan los abonos realizados durante el día, la semana y el mes actual.

Reporte Diario

VITAL PLUS
FACTURACIÓN

Valor	Tipo	Metodo	Fecha	Procedimiento	Documento	Profesional	Saldo
VP0000062							
28000.0	Regular	Efectivo	21/01/18 05:19 PM	Endodoncia	22434241	Paola González	1209000

domingo 21 enero 2018 Page 1 of 1

Página 1 de 1

Imagen 115. Reporte Día Actual

Reporte Semanal

VITAL PLUS
FACTURACIÓN

Valor	Tipo	Metodo	Fecha	Procedimiento	Documento	Profesional	Saldo
VP0000062							
28000.0	Regular	Efectivo	21/01/18 05:23 PM	Endodoncia	22434241	Paola González	1209000

domingo 21 enero 2018 Page 1 of 1

Página 1 de 1

Imagen 116. Reporte de Semana Actual

Reporte mensual

VITAL PLUS

FACTURACIÓN

Valor	Tipo	Metodo	Fecha	Procedimiento	Documento	Profesional	Saldo
VP0000047							
45000.0	Cuota Inicial	Efectivo	6/01/18 03:48 PM	Ortodoncia	1234	Francisco Ponce	2255000
VP0000048							
4500.0	Regular	Efectivo	6/01/18 04:14 PM	Yema	1234	Francisco Ponce	0
VP0000049							
4500.0	Regular	Efectivo	9/01/18 08:39 PM	Yema	1234	Francisco Ponce	0
VP0000050							
3000.0	Regular	Efectivo	9/01/18 10:57 PM	Ortodoncia	1234	Francisco Ponce	2252000
VP0000051							
30000.0	Regular	Efectivo	9/01/18 10:58 PM	Ortodoncia	1234	Francisco Ponce	2222000
VP0000052							
5000.0	Regular	Efectivo	10/01/18 10:19 PM	Extra 5	1234	Francisco Ponce	0


domingo 21 enero 2018 Page 1 of 3

Pagina 1 de 3

Imagen 117. Reporte por Mes Actual

En la sección “Personalizado” se consultan las facturas dentro del rango de fechas ingresados en los respectivos campos, ya sea manualmente o con los botones de calendario, y haciendo clic en el botón “Generar”.

Vital Plus - Nombre de usuario



Específico

Personalizado

Desde: Hasta:

Usuario: Nombre de usuario ingresado

Imagen 118. Ingreso de fechas para reporte personalizado

Reporte entre 2016-02-01 y 2016-07-01

VITAL PLUS

FACTURACIÓN

Valor	Tipo	Metodo	Fecha	Procedimiento	Documento	Profesional	Saldo
VP0000001							
30000.0	Cuota Inicial	Efectivo	4/03/16 11:11 AM	Ortodoncia	123	Gloria Salazar Arboleda	2970000
VP0000002							
35000.0	Regular	Efectivo	3/06/16 11:11 AM	Ortodoncia	123	Gloria Salazar Arboleda	2935000
VP0000003							
35000.0	Regular	Efectivo	4/06/16 11:11 AM	Ortodoncia	123	Gloria Salazar Arboleda	2900000
VP0000004							
35000.0	Regular	Efectivo	4/06/16 11:12 AM	Ortodoncia	123	Gloria Salazar Arboleda	2865000
VP0000005							
39978.0	Cuota Inicial	Tarjeta de Crédito	4/06/16 11:14 AM	Endodoncia	567	Paola González	1460022

domingo 21 enero 2018 Page 1 of 3


Pagina 1 de 3

Imagen 119. Reporte personalizado entre fechas ingresadas

3.2.6. Reporte Por Profesional.



Haciendo clic en este botón se accede al menú de informes por profesional dentro de un rango de fechas.

VitalPlus



Profesional:

Profesional 1
 Profesional 2
 Profesional n

Desde:  Hasta: 

Usuario: Nombre de administrador ingresado

Imagen 120. Ingreso de datos en ventana de Reporte por profesional en rango de fechas

Valor	Tipo	Metodo	Fecha	Procedimiento	Documento	Profesional	Saldo
VP0000005							
39978.0	Cuota Inicial	Tarjeta de Crédito	4/06/16 11:14 AM	Endodoncia	567	Paola González	1460022
VP0000006							
39978.0	Regular	Tarjeta de Crédito	7/06/16 11:14 AM	Endodoncia	567	Paola González	1420044


domingo 21 enero 2018 Page 1 of 1
Pagina 1 de 1

Imagen 121. Reporte por profesional



3.2.7. Reporte Por Especialidad.

En esta opción se consultan las facturas de una especialidad dentro de un rango de fechas determinadas y eligiendo a la especialidad en la lista desplegable.

VitalPlus



Especialidad:

Desde:  Hasta: 

Usuario: Nombre de administrador ingresado

Imagen 122. Ingreso de datos en ventana de Reporte por especialidad en rango de fechas

Reporte de Ortodoncia

VITAL PLUS

FACTURACIÓN

Valor	Tipo	Metodo	Fecha	Procedimiento	Documento	Profesional	Saldo
VP0000002							
35000.0	Regular	Efectivo	3/06/16 11:11 AM	Ortodoncia	123	Gloria Salazar Arboleda	2935000
VP0000003							
35000.0	Regular	Efectivo	4/06/16 11:11 AM	Ortodoncia	123	Gloria Salazar Arboleda	2900000
VP0000004							
35000.0	Regular	Efectivo	4/06/16 11:12 AM	Ortodoncia	123	Gloria Salazar Arboleda	2865000
VP0000007							
30000.0	Regular	Efectivo	6/06/16 09:04 PM	Ortodoncia	123	Francisco Ponce	2835000
VP0000010							
5996.0	Cuota Inicial	Tarjeta de Crédito	13/06/16 07:09 PM	Ortodoncia	987	Gloria Salazar Arboleda	2994004

domingo 21 enero 2018 Page 1 of 3


Pagina 1 de 3

Imagen 123. Reporte por especialidad

3.2.8. Registros.

Haciendo clic en este botón, aparece la pantalla para exportar el registro de inicio y cierre de sesión de los usuarios agentes en una fecha determinada y haciendo clic en el botón “Generar”.

VitalPlus



Desde:

Usuario: Nombre de administrador ingresado

Imagen 124. Ingreso de fecha en ventana de Registros

The image shows a software window titled "Reporte de actividades 2017-09-19". The window contains a report with the following content:

VITAL PLUS

REGISTRO DE ACTIVIDADES

ID	Nombre	Ingreso	Salida
87001	daniel	19/09/17 12:55 PM	19/09/17 01:50 PM
87001	daniel	19/09/17 02:46 PM	19/09/17 05:30 PM
87004	Fabio	19/09/17 02:41 PM	19/09/17 06:38 PM

domingo 21 enero 2018 Page 1 of 1

Pagina 1 de 1

Imagen 125. Reporte de actividades

Anexo D



ECBTI – Ingeniería de Sistemas

ANÁLISIS Y DISEÑO DE APLICACIÓN PARA LA GESTIÓN ECONÓMICA Y DE SERVICIOS DE LA EMPRESA VITAL PLUS

DANIEL DARÍO PERTUZ OROZCO

RESUMEN

La necesidad que tiene la empresa *Vital Plus*, ubicada en la ciudad de Barranquilla, consiste en lograr un control detallado de las transacciones, registros y novedades que pueden ocurrir durante los tratamientos odontológicos que ofrece a sus pacientes. Para ello, se desarrolla una aplicación de escritorio personalizada que pueda realizar todas las tareas que se ejecutan en la empresa.

OBJETIVO

Analizar y diseñar una aplicación de escritorio que permita llevar un control de los tratamientos odontológicos y las historias clínicas de los pacientes de la empresa *Vital Plus*, con el fin optimizar su prestación de servicios en salud oral.

DISEÑO

Se evaluaron los diferentes diagramas que hacen parte del desarrollo de un software para cumplir con los requerimientos de la empresa. Además, se estableció el diseño de la base de datos que utiliza el sistema para trabajar.

Diagrama de casos de uso



Diagrama de base de datos

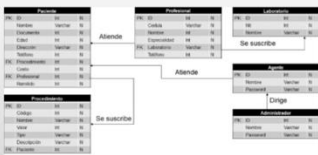


Diagrama de actividades



RESULTADO

Se presenta un prototipo de lo desarrollado en Java al personal administrativo y de sistemas de la empresa *Vital Plus* para proceder a su implementación.

Pantalla para inicio de sesión



Pantalla principal para administrador



Pantalla principal para agente



Junto a este prototipo, se presentan manuales de instalación de la aplicación y de las librerías generadoras de reportes *JasperReports* e *iReport*.



CONCLUSIONES

La identificación de cada uno de los requerimientos que la empresa *Vital Plus* planteaba conllevó a un completo entendimiento de su estructura interna para que el desarrollo de esta aplicación contribuyera a que la prestación de servicios de la empresa a sus pacientes fuese más óptimo y satisfactorio.

El diseño del prototipo de la aplicación conduce a que la empresa pueda tener una noción de lo que se desarrolló en este proyecto para poder suplir las necesidades que tiene en facturación, todo con la meta de mejorar la experiencia de sus pacientes en servicios de salud oral.