

GESTOR DE FORMULARIOS Y MATRICULAS

**FERNANDO SANTA LÓPEZ
DAVID PULGARIN VALENCIA**

**UNIVERSIDAD TECNOLÓGICA DE PEREIRA FACULTAD DE INGENIERÍAS
ELÉCTRICA, ELECTRÓNICA, FÍSICA Y CIENCIAS DE LA COMPUTACIÓN
PROGRAMA INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
PEREIRA - RISARALDA
2016**

GESTOR DE FORMULARIOS Y MATRICULAS

**FERNANDO SANTA LÓPEZ
DAVID PULGARIN VALENCIA**

PLATAFORMA WEB ECONTINUA ASEUTP

**Directora: Paula Andrea Villa Sánchez
Docente investigadora Universidad Tecnológica de Pereira**

**UNIVERSIDAD TECNOLÓGICA DE PEREIRA FACULTAD DE INGENIERÍAS
ELÉCTRICA, ELECTRÓNICA, FÍSICA Y CIENCIAS DE LA COMPUTACIÓN
PROGRAMA INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
PEREIRA – RISARALDA**

2016

NOTA DE ACEPTACIÓN:

FIRMA DEL DIRECTOR DEL PROYECTO:

TABLA DE CONTENIDO

INTRODUCCIÓN.....	3
1. DIAGNÓSTICO DE LOS PROCESOS DE LA ASEUTP	4
2. ANÁLISIS DE HERRAMIENTAS EXISTENTES PARA GESTORES DE FORMULARIOS	7
2.1. ANÁLISIS DE LAS HERRAMIENTAS:	9
3. SELECCIÓN DE FRAMEWORK Y LENGUAJES DE PROGRAMACIÓN.....	9
3.1. LENGUAJES DE PROGRAMACIÓN BACK-END.....	10
3.1.1. PHP.....	10
3.1.2. PYTHON.....	11
3.1.3. RUBY.....	12
3.1.4. JAVA.....	13
3.2. LENGUAJES DE PROGRAMACIÓN FRONT-END.....	14
3.2.1. HTML.....	14
3.2.2. JAVASCRIPT	15
3.2.3. CSS.....	16
3.3. PATRONES DE DISEÑO	18
3.3.1. Modelo - Vista - Controlador (MVC)	18
3.3.2. Modelo - Vista - Template (MVT).....	19
3.4. FRAMEWORKS.....	20
3.4.1. ZEND(PHP)	20
3.4.2. RAILS(RUBY).....	21
3.4.3. OPENXAVA(JAVA).....	21
3.5. FRAMEWORK Y LENGUAJES A USAR PARA EL DESARROLLO.	22
4. DESARROLLO HERRAMIENTA DE GESTIÓN DE FORMULARIOS.....	23
4.1. ALCANCE.....	23
4.2. LIMITACIONES.....	23
4.3. RIESGOS	24
4.3.1. Identificación de los riesgos	24
4.3.2. Clasificación de los riesgos	25
4.3.3. Análisis de riesgos.....	26
4.3.4. Plan de mitigación	26

4.3.5.	Plan de contingencia.....	27
4.4.	DOCUMENTO DE REQUERIMIENTOS.....	28
4.5.	VISTAS 4+1.....	28
4.5.1.	Diagrama de Casos de Uso.....	28
4.5.2.	Diagramas de Secuencia.....	29
4.5.3.	Diagrama de Componentes.....	45
4.5.4.	Diagrama de Despliegue.....	46
4.6.	PLAN DE PRUEBAS	47
4.6.1.	Identificador plan de pruebas	47
4.6.2.	Introducción.....	47
4.6.3.	Elementos	47
4.7.	EJECUCIÓN Y REPORTE DE PRUEBAS	53
5.	EVALUACIÓN DE MEJORA	53
6.	CONCLUSIONES.....	57
	REFERENCIAS	58
	LISTA DE ANEXOS.....	59

INTRODUCCIÓN

La Asociación Nacional de Egresados de la Universidad Tecnológica de Pereira (ASEUTP) es una entidad sin ánimo de lucro, de derecho privado y beneficio común, la cual busca fomentar el desarrollo integral de la comunidad, la empresa, la universidad y los egresados. Para cumplir con sus objetivos la ASEUTP a lo largo de muchos años ha prestado servicios tales como, una bolsa de empleo, educación continuada y beneficios a sus egresados brindando espacios para la integración de estos, descuentos especiales, actividades académicas e información virtual enviada a los correos electrónicos de sus miembros. Todo esto con la finalidad de estrechar entre sus miembros vínculos de fraternidad al provenir de la misma alma máter y siempre buscando el desarrollo y la exaltación del buen nombre de la Universidad Tecnológica de Pereira [1]

El proyecto de un gestor de formularios para el proceso de educación continua que trabaja la Asociación Nacional de Egresados de la Universidad Tecnológica de Pereira (ASEUTP) se realizó buscando en un principio la automatización de los formularios necesarios para realizar la inscripción a los cursos bajo la categoría de educación continua pero terminó siendo una plataforma mucho más completa que brinda funcionalidades tanto para el usuario administrador de dicha plataforma, como para los profesores y demás usuarios que vayan a hacer uso de la misma.

En el presente documento se podrá encontrar una pequeña investigación donde se determinó porque era necesario realizar esta plataforma, se rescataron características de gestores de formularios con mucho tiempo y experiencia en el mercado, y se determinó luego de analizar diferentes lenguajes de programación y herramientas de desarrollo la más óptima para la ejecución de este proyecto. Además, también se podrá encontrar información de ingeniería del producto desarrollado.

1. DIAGNÓSTICO DE LOS PROCESOS DE LA ASEUTP

El diagnóstico realizado a todos los procesos de la Asociación Nacional de Egresados de la Universidad Tecnológica de Pereira dejó como resultado una serie de oportunidades en temas de automatización debido a que en muchos de los procesos se encontró que aún se maneja demasiada documentación de manera manual y archivos de Excel. Los aspectos más relevantes que se obtuvieron de este diagnóstico y que se dejan plasmados en este documento con la finalidad que más adelante puedan ser abarcados por la ASEUTP para su mejora son: definir e implementar la gestión y el monitoreo de proyectos, un sistema de información de asociados y la instalación de un software para realizar la gestión financiera. Como la finalidad del proyecto se encuentra en el proceso de educación continua, para este el diagnóstico fue algo más detallado y se describe a continuación.

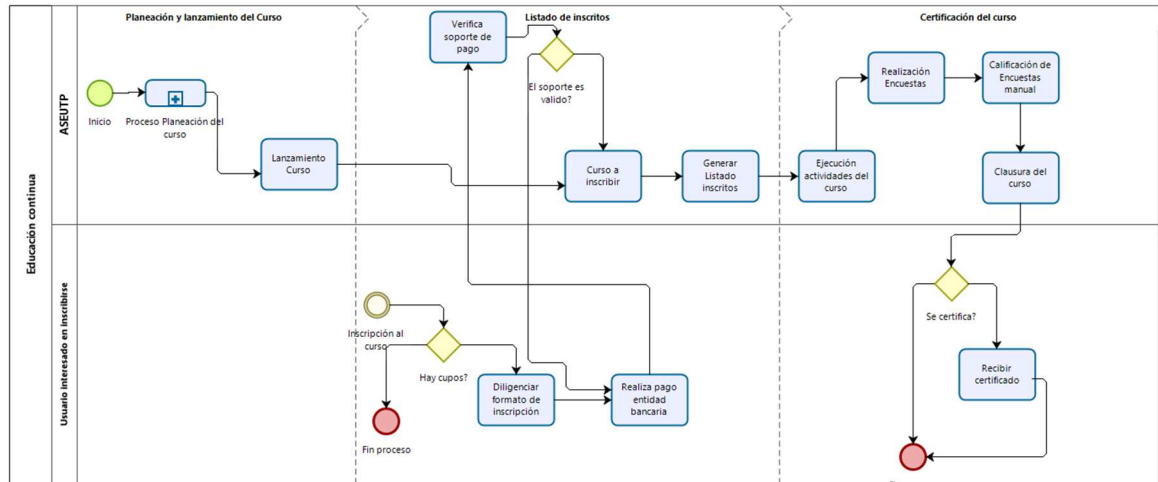
Para el proceso de educación continua, se utilizó la técnica de entrevistas, las cuales se realizaron a los funcionarios de la ASEUTP con el fin de entender el funcionamiento del proceso Educación continua, además a esto, la entrevista realizada tenía como propósito el permitir obtener algunos valores en cuanto a tiempos empleados para realizar las inscripciones y cada etapa que conlleva el crear un curso de educación continuada.

Como resultado final de la entrevista realizada además de obtener los tiempos que se buscaban, se pudo realizar la deducción del siguiente diagrama que muestra más claramente el proceso para realizar una inscripción en un curso de educación continua.

Nombre: *Diagrama de flujo del proceso educación continua*

Descripción:

El siguiente diagrama de flujo muestra el procedimiento llevado a cabo para la inscripción de un curso, teniendo en cuenta desde la planeación del curso hasta su clausura.



Para llevar a cabo el diagnóstico del proceso de la ASEUTP, se utilizó la técnica de entrevistas, las cuales se realizaron a los funcionarios de la ASEUTP con el fin de entender el funcionamiento del proceso *Educación continuada*, además a esto, dichas entrevistas se anexan a este documento como soporte.

La ASEUTP dentro de su proceso de educación continuada maneja diversos cursos que permiten a egresados de la Universidad Tecnológica de Pereira y otras personas profesionales continuar con su preparación algunos de ellos son: diplomados, seminarios y otros tipos de cursos que se habilitan cada cierto tiempo.

La manera como los interesados en estos cursos acceden a ellos es mediante un formulario de inscripción, el cual se puede encontrar en la página de la ASEUTP, descargar e imprimir y diligenciar manualmente o en físico al acercarse a la ASEUTP. Además de este formulario, se hace necesario que cada interesado del curso realice el pago en una entidad bancaria y posterior a esto, lleve el soporte del pago a la ASEUTP con el fin de asegurar su cupo para el curso de interés, realizando filas en entidades bancarias lo que hace el proceso de inscripción sea lento.

Al no tener sistematizado ninguna de las etapas que se tienen para realizar un curso nuevo, esto desde el momento de inscripción hasta el momento de certificar a las personas asistentes a los cursos, hace que la persona encargada de los procesos de educación continuada se tome una gran tiempo en recibir todos los formatos en físico de inscripción, verificar datos, agregarlos a un

listado de asistentes a los cursos, y finalmente esperar que los inscritos entreguen el soporte de pago en entidades bancarias lo que hace aún más lento este proceso.

Teniendo en cuenta lo anterior también se puede observar que una persona que participa un número N de veces en diferentes cursos tiene que llenar la misma cantidad N de veces el formato de inscripción generando esto un gasto de tiempo en repetir información y generando mayor documentación repetida con pequeños cambios, la cual teniendo en cuenta que toda esta información está en físico, ocupa un mayor espacio en archivadores lo que implica que al momento de realizar búsqueda sobre esta información hace que sean más dispendiosas y poco productivas.

La siguiente tabla contiene información acerca de los criterios de mejora que se tendrán para verificar la mejora del procedimiento.

PRESENTACIÓN DE CRITERIOS DE MEJORA		
#	ITEM	PROCEDIMIENTO EDUCACIÓN CONTINUA
1	$\frac{\# \text{ Número de veces a diligenciar formato de inscripción}}{\# \text{ Número de cursos}}$	$\frac{N}{N} = 1$
2	$\frac{\# \text{ Número de preguntas a revisar por encuesta}}{\# \text{ Número de estudiantes}}$	$\frac{N}{N} = 1$

Para garantizar la mejora del procedimiento los valores nuevos que se obtengan deberán ser menores a los actuales, ya que de esta manera se ahorrara tiempo para el empleado de la ASEUTP y al interesado a inscribir al curso.

El tiempo promedio que tarda un usuario para inscribirse a un curso asumiendo que la persona descargó su formulario de internet y lo diligencio en su casa:

5 minutos llenando información de los formularios de inscripción.

1 hora 20 minutos realizando el pago incluyendo desplazamiento al banco y a la asociación.

20 minutos llevando soporte de pago a la asociación

Asumiendo que la persona diligencio el formulario en la ASEUTP y pago en el FAVI:

5 minutos llenando información de los formularios de inscripción.

20 minutos realizando el pago

5 minutos llevando el soporte a la ASEUTP.

El tiempo que gasta un funcionario organizando información de los cursos (Listas de asistencia, Verificación de pagos):

Generando lista de asistencia: 1 hora

Verificando soporte de pagos: 2 horas

2. ANÁLISIS DE HERRAMIENTAS EXISTENTES PARA GESTORES DE FORMULARIOS

En la actualidad la necesidad que presentan tanto empresas como personas, de gestionar formularios web que años atrás se manejaban en formatos físicos, es demasiado común, por lo que para ello en internet se pueden encontrar distintos tipos de aplicaciones y herramientas que facilitan la gestión de formularios personalizados y de acuerdo a las necesidades de cada usuario, teniendo en cuenta que al usar estas herramientas el usuario debe adaptar sus necesidades a las limitaciones que ellas tengan.

Algunas de las herramientas existentes para la gestión de formularios desde la web y sus características principales son:

Google Forms:

Este gestor de formularios desarrollado por los ingenieros de la compañía internacional Google, es en la actualidad una de las herramientas más utilizadas debido a su facilidad de manejo para todas las personas, incluyendo aquellas que no saben sobre conceptos técnicos de informática.

Entre sus características principales se encuentran:

- Fácil creación de formularios web manejando distintos tipos de respuesta según las necesidades de cada usuario.
- Acceso gratuito a creación de N formularios.
- Posibilidad de configurar entornos gráficos agradables a la vista del usuario sin mayor complejidad.
- Recopilación automática de la respuesta de cada usuario en una hoja de cálculo, con gráficos y datos de las respuestas en tiempo real.

- Facilidad de responder las encuestas a través de distintas plataformas y en cualquier lugar del mundo, desde que se tenga conexión a internet.
- Permite la creación de encuestas de manera grupal, es decir, varias personas podrán trabajar desde distintos dispositivos en la creación de la misma encuesta. [2]

Formbakery:

Esta herramienta online permite la creación de formularios web de manera muy sencilla ya que dispone una serie de plantillas las cuales se usan con tan solo arrastrarlas al espacio del formulario y adaptarlas o modificarlas según la necesidad, no es necesario escribir ni una sola línea de código para llegar a crear un formulario de aspecto profesional, como características principales Formbakery presenta:

- Facilidad de añadir o eliminar opciones a los formularios según sea necesario.
- Generación automática de código para insertar los formularios en sitios web, redes sociales o demás proyectos al momento de terminar la construcción del formulario.
- Posibilidad

Survey Monkey

Posiblemente esta herramienta es una de la más usadas y populares en el mercado de los gestores de formularios, con más de 20 millones de clientes a nivel mundial. Sus diferentes funcionalidades que permiten la creación de formularios para conocer desde la satisfacción de los clientes de una empresa, proyectos simples o para la contratación de empleados, hacen que esta potente herramienta esté posicionada fuertemente en el mercado.

Sus principales características son:

- Posee en su interior una biblioteca de preguntas de encuestas validadas por expertos que se pueden usar libremente.
- Permite distribuir las encuestas por páginas web y redes sociales de manera fácil.
- Recolección de las respuestas de los formularios en tiempo real.

- En su versión gratuita tiene algunas limitaciones en cuanto a número de preguntas por encuesta y respuestas, tampoco permite descargar las respuestas en ningún formato, ni ningún tipo de exportación, si se desea tener esta opción debe contratarse alguno de los planes pagos que ofrece esta plataforma.
- En los planes pagos las características que ofrece esta herramienta son mucho más amplias y potentes.

2.1. ANÁLISIS DE LAS HERRAMIENTAS:

Estas herramientas permiten la creación de formularios dinámicos, algunos más agradables que otros como es el caso de google forms que permite crear formularios con entornos gráficos más agradables y sin tener un límite de formularios por usuario, lo que es muy beneficioso en caso de encuestas académicas o que no requieran gran análisis, puesto que los diferentes gestores que se encuentran disponibles en internet y en caso particular de este análisis las herramientas Survey monkey y Formbakery para explotar mayor capacidad de análisis y funcionalidades requieren ser pago, nuevamente google forms es mucho mejor en estos casos ya que aunque también tenga su opción de contratar un plan con muchas más funcionalidades, la opción gratuita es muy amplia y resulta muy útil en diferentes escenarios.

Con las opciones que maneja google forms y survey monkey serian herramientas ideales para la gestión de los formularios en el caso que no se necesitará tener un manejo de perfiles y cuentas de usuario, es decir, para la recolección de información son buenas herramientas en sus versiones gratuitas y algunas de sus características de diseño serán tenidas en cuenta para el desarrollo a realizar. Pero no basta solo con tener una buena fuente de recolección de información para las necesidades actuales de la ASEUTP, por lo que es indispensable después de analizar las herramientas existentes, realizar un desarrollo a la medida que permite optimizar considerablemente los esfuerzos en cuanto a tiempo, trabajo y espacio de almacenamiento físico tanto por parte de la ASEUTP, como de los interesados en participar de los cursos de educación continua.

3. SELECCIÓN DE FRAMEWORK Y LENGUAJES DE PROGRAMACIÓN

A continuación, se describen algunos de los lenguajes de programación y frameworks más usados

en la actualidad de tal manera que posteriormente se pueda hacer una selección de uno de ellos, teniendo diferentes opciones con diferentes características y así elegir el que más se adapte o facilite la realización del proyecto.

3.1. LENGUAJES DE PROGRAMACIÓN BACK-END

Los lenguajes de programación back-end son aquellos que implementan sus desarrollos del lado del servidor, es decir, allí se programan funcionalidades tales como conexiones a las bases de datos, plantillas del lado del servidor, entre otras. Básicamente en estos lenguajes se desarrollan funcionalidades para recibir la información, procesarla y enviarla a las vistas del usuario ya sea en código HTML o también se podrían enviar los datos de manera pura (formatos tales como JSON, RSS, entre otros) para luego ser procesados por JavaScript y finalmente ser entregados en las vistas del usuario.

3.1.1. PHP

Debido a la gran capacidad de dinamismo e interactividad que exige actualmente el desarrollo web en sus aplicaciones y a que lenguajes tales como HTML imponen limitaciones al momento de generar aplicaciones web interactivas y personalizadas, PHP es una alternativa la cual proporciona a los desarrolladores formas de solucionar estas limitaciones anteriormente dichas.

Como tal PHP es un lenguaje de programación back-end el cual se ejecuta del lado del servidor, lo que quiere decir que antes de que la información o páginas sean vistas por el usuario, en el servidor las páginas pueden realizar una serie de diferentes acciones para las cuales están programadas (solicitudes a bases de datos, conexiones de red, entre otras) para posteriormente ser enviadas al usuario con la información que estos esperaban.

Como tal PHP fue uno de los primeros lenguajes de programación que permitía integrarse a HTML y así generar dinamismo en las páginas web sin necesidad de realizar llamados a archivos externos.

Ventajas:

- Una de sus grandes y principales ventajas es que PHP es un lenguaje multiplataforma que posee una capacidad de conexión con la mayoría de motores de bases de datos usados en la actualidad, principalmente MySQL y PostgreSQL.
- Posee una serie de módulos llamados extensiones los cuales permiten potenciar las capacidades del PHP.
- Su fuerte son la generación de aplicaciones web dinámicas permitiendo acceso a bases de datos, lo que hace que PHP sea una de las primeras opciones a tener en cuenta al momento de generar desarrollos web.
- Otra ventaja no menos importante es que PHP es software libre.

Desventajas:

- Al ser un lenguaje que se ejecuta del lado del servidor, es muy factible que colapse cuando el número de solicitudes de páginas aumente considerablemente.
- Se requiere de un servidor que soporte PHP para poder ver o testear las páginas o implementaciones que se vayan haciendo.
- Al ser software libre no tiene una compañía que de soporte de él, aunque exista una comunidad grande que podría dar soporte.

3.1.2. PYTHON

Desarrollado a principios de los años noventa por Guido van Rossum Python es un lenguaje de programación multiparadigma, lo que quiere decir que puede usarse para programación orientada a objetos, imperativo o funcional, de licencia de código abierto, multiplataforma e interpretado lo que ahorra tiempo ya que no requiere ser compilado.

Al tener una sintaxis fácil de aprender y ser demasiado legible Python es uno de los lenguajes de programación y desarrollo más usados actualmente incluso empresas como YAHOO y GOOGLE lo usan para sus desarrollos.

Una de sus características principales y factor que lo ayudó a posicionarse como un lenguaje potente es el gran soporte que Python maneja y la facilidad de integrarse con otros lenguajes y herramientas y el hecho de que tiene integradas varias librerías estándar que facilitan el trabajo al desarrollador.

Ventajas:

- Posee una gran cantidad de funciones y librerías para diferentes usos lo que permite un desarrollo más rápido y reducen el trabajo del desarrollador ya que las bibliotecas hacen gran parte de este.
- Es un lenguaje multiplataforma que a la vez soporta diferentes bases de datos lo que lo hace excelente a la hora de generar desarrollos web ya que ofrece variedad a los desarrolladores.
- Es un lenguaje open source, con una curva de aprendizaje corta.

Desventajas:

- Al ser un lenguaje interpretado, los desarrollos son más lentos que en lenguajes compilados cuando se trata de códigos de gran tamaño.

3.1.3. RUBY

Este lenguaje de programación de muy alto nivel y poco conocido fue desarrollado en 1993 por el japonés Yukihiro Matsumoto y en su sintaxis se refleja mucha similitud con otros lenguajes tales como PHP o Python.

Como tal Ruby en sus inicios manejaba solo documentación en japonés motivo que generó su poca acogida fuera de Asia, pero en la actualidad es posible encontrar numerosos tutoriales y documentación en castellano e inglés.

Ruby al igual que Python es un lenguaje multiparadigma e interpretado, el cual es considerado un lenguaje flexible debido a que permite a los usuarios alterarlo libremente [3]. En otras palabras, Ruby quita ataduras a los desarrolladores sobre lo que ya está y permite que estos agreguen, modifiquen o quiten funcionalidades de acuerdo a las necesidades que se tengan en el momento.

Otra de sus características importantes es que Ruby permite la integración fácilmente con bases de datos y otras plataformas, lo que en últimas favorece al desarrollador.

Ventajas:

- Los costos en la actualidad son un factor fundamental al momento de realizar un desarrollo, un punto a favor de Ruby es que permite implementaciones a costos bajos en comparación a otros lenguajes.
- Tiene facilidad de implementación y es soportado por casi todas las plataformas web.
- Es un lenguaje multiplataforma, que integra librerías (gemas) para facilitar el trabajo al desarrollador, incluso para el manejo de base de datos integra comandos.

Desventajas:

- En la actualidad de Ruby está siendo cada vez más conocido, pero aún no es un lenguaje de uso extendido como algunos otros por lo que su soporte no es tan amplio como lo podría ser el de PHP u otros lenguajes también con licencias abiertas, pero con comunidades mucho más grandes.
- Procesamiento lento al ser un lenguaje interpretado.

3.1.4. JAVA

Java es un lenguaje de programación back-end, orientado a objetos y de plataforma independiente, desarrollado por la compañía Sun Microsystems y la cual desde sus inicios fue concebida con la finalidad de desarrollar páginas web.

La sintaxis de Java deriva de lenguajes de programación tales como C y C++. Java busca principalmente que las aplicaciones sean escritas una sola vez y de allí pueda ser compilado en cualquier otro dispositivo, esto significa que el código se compila una vez en alguna plataforma y de allí en adelante cualquier dispositivo que vaya a hacer uso de este código no requiere volver a compilarlo, lo que produce ganancia en tiempos de ejecución.

Actualmente Java es uno de los lenguajes de programación que más popularidad tienen en el desarrollo de aplicaciones cliente-servidor. Esto podría deberse a la gran variedad de aplicativos que se pueden generar usando este lenguaje, un ejemplo de ello son los *applets*¹.

Ventajas:

- Cualquier navegador que sea compatible con Java ejecuta cualquier aplicación hecha en

¹ Los applets son aplicaciones especiales que se ejecutan dentro de un navegador al cargar una página web HTML.

este lenguaje de programación.

- Permite agregar a las aplicaciones web cualquier tipo de elemento de multimedia y con ellos generar interactividad en las aplicaciones sin generar costos adicionales en paquetes multimedia.
- El mercado para aplicaciones Java es abundante, de igual manera la documentación y el soporte a ellas.
- Es un lenguaje Simple (comparado con C), robusto y seguro.

Desventaja:

- La velocidad es algo que aún no se maximiza, debido a que es un lenguaje interpretado aún no logra la velocidad de un ejecutable.
- Algunas de las herramientas que se puedan llegar a necesitar a la hora de lograr un desarrollo complejo y en óptimas condiciones podrían requerir un pago costoso.

3.2. LENGUAJES DE PROGRAMACIÓN FRONT-END

Los lenguajes de programación front-end son aquellos con los cuales se realizan diseños o implementaciones que están del lado del cliente, normalmente estas implementaciones son realizadas por desarrolladores front-end pero hay algunos diseñadores los cuales también manejan de excelente forma estos lenguajes, debido a que en el front-end se centra básicamente en el diseño y la estructura de lo que el usuario va a visualizar.

3.2.1. HTML

*HTML*² es un lenguaje estático el cual actualmente se encuentra en su quinta versión HTML5, también se puede definir como un estándar a cargo de la W3C³ la cual es la organización que se dedica a estandarizar la mayoría de tecnologías orientadas a la web. HTML se encarga de definir la estructura básica de las páginas web mediante un código el cual está basado en etiquetas.

Básicamente es un lenguaje interpretado por los navegadores web que en sí dentro de sus etiquetas solo contiene texto y para agregar archivos multimedia lo hace utilizando un sistema de

² sigla en inglés de HyperText Markup Language (lenguaje de marcas de hipertexto).

³ worl wide web consortium (Consortio WWW)

referencias.

Debido a los diferentes cambios que presenta HTML agregando y suprimiendo funcionalidades a través de sus diferentes versiones en las que se busca siempre mejorar y hacer más eficiente la creación de páginas web se tiene que para que los navegadores web ejecuten correctamente las páginas creadas en las últimas versiones de HTML estos también deberán estar actualizados a sus últimas versiones.

Ventajas:

- HTML es un lenguaje fácil de aprender debido a su sencillez y a que es muy intuitivo.
- Es posible encontrar un gran número de aplicaciones que generan código HTML automáticamente al realizar diseños visuales.
- Es el lenguaje más reconocido y utilizado a nivel mundial.
- Su código funciona en todos los navegadores sin necesidad de herramientas externas.
- La documentación, tutoriales y libros acerca de HTML se encuentran por montones.

Desventajas:

- Al ser un lenguaje estático limita las posibilidades de generar aplicaciones web dinámicas e interactivas.
- No permite manejo de bases de datos dentro de sus funcionalidades.
- El aspecto de sus páginas web o aplicaciones varían notoriamente dependiendo el dispositivo donde se visualice y el navegador.

3.2.2. JAVASCRIPT

Al igual que el HTML, JavaScript es un lenguaje interpretado, lo que quiere decir que son los navegadores los encargados de procesar e interpretar sus implementaciones. También tiene similitudes con lenguajes tales como Java Y C, pero a diferencia de estos dos, JavaScript no es un lenguaje orientado a objetos, sino basado en prototipos.

Es poco probable encontrar una aplicación web implementada netamente en JavaScript, regularmente las aplicaciones de JavaScript se encuentran en el cuerpo de páginas web creadas usando HTML y CSS para agregar dinamismo e interactividad a ellas, por ejemplo, para acceder a imágenes, chequear formularios en tiempo real, hacer validaciones de todo tipo, entre otras.

En la actualidad se crearon unos motores para optimizar el tiempo de interpretación de JavaScript, la clave de estos es transformar código JavaScript en código máquina para que los tiempos de ejecución sean similares a los de aplicaciones de escritorio [4].

Los sectores de código JavaScript en medio de código HTML, CSS u otro son denominados como scripts.

Ventajas:

- Es un lenguaje de programación sencillo, liviano y que se recomienda siempre al momento de crear páginas o aplicaciones web.
- La facilidad de integrarlo con diferentes lenguajes y crear implementaciones robustas y seguras es un punto a favor de JavaScript.
- El hecho de que sea compatible con la mayoría de navegadores web hacen que sea actualmente uno de los lenguajes de desarrollo más usados.
- Ayuda a facilitar desarrollos interactivos, con una gran cantidad de efectos visuales y utiliza poca memoria, permitiendo además un fácil manejo de datos.

Desventajas:

- Una de las principales desventajas que presenta JavaScript es el hecho de que los usuarios puedan desactivar JavaScript en sus navegadores, lo que generaría fallos en las aplicaciones.
- Otro de lo que podría ser un problema para JavaScript es que sus recursos propios no son tan extensos, es decir, tiene capacidades limitadas lo que prácticamente obliga a utilizarlo en conjunto con otros lenguajes de programación.
- Antes de ejecutar un script el navegador para poder interpretarlo debe descargarlo en su totalidad, lo que podría ser perjudicial en el caso que el tamaño y los datos que contenga sean muchos ya que podría volver lenta la carga de la aplicación o página.

3.2.3. CSS

CSS⁴ es el lenguaje que se utiliza en las aplicaciones web para describir la presentación de documentos HTML o XML[5], en otras palabras mediante CSS se describe cómo se deben

⁴ Cascading Style Sheets (hojas de estilo en cascada)

interpretar los atributos de estos documentos. Su uso está ligado a establecer el diseño de documentos HTML, XML o cualquier otro de este tipo. La corporación encargada de mantener CSS es la W3C.

CSS se centra en establecer parámetros de diseño para páginas e interfaces de usuario, en cuanto a separación del contenido y la forma de presentar este, mediante algunas características como lo son los layouts o capas, la fuente de la letra, su tamaño, los colores, fondos, entre otros. aunque no solo en el ámbito visual gracias a las hojas de estilo auditivas. [6]

La unión entre HTML, JavaScript y CSS permiten la creación de sitios web con interfaces atractivas visualmente, lo que hace que esta tecnología sea muy usada actualmente.

Ventajas:

- Se pueden realizar modificaciones a los elementos HTML sin necesidad de modificar este archivo, por lo que se ahorra esfuerzo y tiempo de edición, teniendo en cuenta que con una sola hoja CSS le podríamos atributos a todos los archivos HTML.
- Se evita tener que recurrir a prácticas complejas para llegar a soluciones requeridas en cuanto a estilo, debido a la gran variedad de efectos que permite CSS.
- ¡Mediante el comando `IMPORTANT!` los usuarios pueden modificar los estilos impuestos por el autor y hacer que sus propios estilos sean cargados por el navegador de manera local.
- CSS brinda la posibilidad de tener hojas de estilo para diferentes navegadores de manera independiente.

Desventajas:

- Es probable que algunas propiedades CSS sean bloqueadas o pasadas por alto por los navegadores y se generen errores al momento de cargar la página.
- Otro de las pocas desventajas que presenta CSS es que al no tener un soporte por parte de los navegadores puede que una hoja de estilo sea reconocida bien por uno y con errores en otro, lo que generaría que los usuarios vean información con formatos que el desarrollador no quiera.

3.3. PATRONES DE DISEÑO

Básicamente los patrones de diseño son la base para solucionar problemas comunes en el desarrollo de software y también en el diseño de interfaces. Pero no todas las soluciones a problemas de diseño o desarrollo son patrones, para que una solución pueda ser considerada un patrón de diseño debe cumplir con ciertas características como por ejemplo que esta solución pueda resolver problemas semejantes en otras situaciones y que ya tenga un historial de problemas resueltos, es decir, compruebe su efectividad. [7]

Se puede decir entonces que un patrón de diseño brinda una solución a un problema de software ya probada y documentada en otros proyectos similares; Todo patrón de diseño consta de unos elementos los cuales son: Nombre, problema al que enfrenta el patrón, solución implementada para resolver el problema y efectos de usar dicho patrón.

Existen tres tipos diferentes de clasificar los patrones de diseño los cuales son:

Creacionales: Se encargan de definir cómo iniciar y configurar los objetos.

Estructurales: Se encargan de solucionar problemas de agregación de clases y objetos, formando estructuras más grandes.

De comportamiento: Se encargan de ofrecer soluciones en cuanto a la comunicación entre clases y objetos. [8]

Teniendo en cuenta las anteriores definiciones, a continuación, se dará un ejemplo de dos patrones de diseño creacionales.

3.3.1. Modelo - Vista - Controlador (MVC)

Este patrón de diseño realiza una división en tres capas o componentes distintos los cuales son el modelo, la vista y el controlador, es decir, este patrón hace una división de la forma en que se va a representar la información y la forma en que se va a presentar la interacción del usuario con el sistema.

Se basa en ideas tales como reutilizar código y separar conceptos de tal manera que se pueda dividir el proyecto en secciones más pequeñas que faciliten la tarea del desarrollador y que en el

momento de realizar mantenimiento sea posible hacerlo de forma más rápida. [9]

En el modelo se maneja todo lo relacionado con la información por la cual el sistema opera, es decir, este se encarga de realizar consultas, actualizaciones, establecer privilegios de acceso a información y demás. Su relación con los otros componentes básicamente es la de enviar información a las vistas y recibir solicitudes por parte del controlador.

Las vistas se centran básicamente en mostrar al usuario de una forma estética la información o lógica del negocio. Su relación o comunicación con el modelo es que esta presenta la información que allí se tiene y mediante sus funcionalidades hacen solicitudes al controlador.

El controlador finalmente es el encargado de responder a los eventos solicitados desde la vista y realizar en caso que sea necesario peticiones al modelo, aunque también este podría dar respuesta a la vista sin necesidad de hacer comunicación con el modelo.

Ventajas:

- Al dividir el proyecto, se hace más escalable y permite ser trabajado de manera más cómoda para los desarrolladores.
- Muchos frameworks permiten abstracción de datos al usar MVC y esto permite realizar consultas a bases de datos de manera más sencilla.
- Los frameworks que utilizan MVC optimizan los recursos del servidor.

Desventajas:

- El MVC es un patrón de diseño orientado objetos por lo que utilizarlo con lenguajes que no incluyan este paradigma resulta complejo y costoso.

3.3.2. Modelo - Vista - Template (MVT)

Este patrón de diseño es muy semejante en teoría al MVC, se diferencia del anterior en lo siguiente:

Para el MVT las vistas hacen referencia a la lógica del negocio, es decir, esta capa es el puente entre Modelo y Template, es algo así como el controlador en el MVC.

El término diferente y que varía un poco a comparación del MVC es el Template, este componente lo que hace es tener un conjunto de planillas que se diseñan estandarizadas, a estos Templates se les encarga mostrar la información al usuario y capturar las interacciones de ellos con el sistema.

Y finalmente el componente Modelo no varía del patrón MVC al patrón MVT, sigue siendo una capa de acceso a la base de datos, que maneja permisos, solicitudes, búsquedas, y demás.

3.4. FRAMEWORKS

Este concepto de frameworks es muy utilizado en el desarrollo web, aunque su ámbito es general, es decir, existen frameworks para todo tipo de desarrollo (juegos, aplicaciones de escritorio, aplicaciones móviles y demás).

Este término se refiere a una estructura de software predeterminada la cual está compuesta por un grupo de componentes personalizables e intercambiables los cuales sirven como plantilla y facilitan la creación de aplicaciones. Se puede hacer referencia a un framework como un programa genérico incompleto el cual se puede configurar de tal manera que se logren resultados específicos. [10]

Existen diferentes tipos de frameworks y por lo general cada lenguaje de programación tiene uno o varios de estos. a continuación, se darán a conocer unos de los más populares y sus características.

3.4.1. ZEND(PHP)

Zend es un framework de código abierto el cual está diseñado para facilitar la creación de aplicaciones web y basado en el uso de servicios de PHP, su implementación utiliza en un 100% el paradigma orientado a objetos y a su vez un bajo acoplamiento entre componentes para así facilitar el uso de cada uno de estos componentes por separado.

Entre sus características principales, zend presenta el uso del patrón MVC por defecto, también una abstracción de base de datos y un manejo de formularios con todas las características de estos

por defecto de tal manera que los desarrolladores puedan hacer un uso de estos de manera sencilla.

Con más de 89 millones de instalaciones Zend Framework es uno de los frameworks más usados en la actualidad para el desarrollo de aplicaciones web. [11]

3.4.2. RAILS(RUBY)

Rails o también conocido como RoR (Ruby on Rails) es un framework de desarrollo web escrito en el lenguaje de programación Ruby el cual fue creado en el año 2003 por David Heinemeier Hansson y que desde entonces ha sido extendido por más de 2000 colaboradores.

Se caracteriza por ser un entorno de desarrollo de código abierto optimizado de tal manera que los desarrolladores no necesitan repetir código y la configuración de todo se vuelve mucho más sencilla.

De tal manera que se puede decir que rails es un conjunto de librerías, automatismos y convenciones que está enfocada en la solución de los problemas comunes que se presentan siempre a la hora de realizar una aplicación web y así permite que los desarrolladores se centren en los aspectos relevantes de sus proyectos, es decir, lo que no es general a todos los proyectos y requiere un poco más de trabajo.

Las gemas que son códigos adicionales que permiten agregar funcionalidades para optimizar el desarrollo, es otra característica que potencia la fuerza de este framework.[12]

3.4.3. OPENJAVA(JAVA)

Openjava es un framework basado en estándares java para el desarrollo de aplicaciones empresariales de forma rápida que presenta como características principales su alta productividad, es decir, no requiere que los desarrolladores escriban interfaces de usuario o conectividad en bases de datos, estas se proveen automáticamente al escribir la lógica del negocio y la estructura de los datos.

Otra de sus características principales es que genera aplicaciones con muchas funcionalidades y si se desea, se puede generar una interfaz móvil a partir del mismo código fuente.

También presenta características como el manejo de una licencia LGPL que permite desarrollar aplicaciones comerciales sin costo alguno, es decir, código abierto. A su favor también tiene el manejo de diferentes idiomas en sus etiquetas y mensajes.

Finalmente se puede decir que la integración de herramientas de terceros en sus desarrollos potencia de gran manera su rendimiento y facilidad de trabajo para desarrolladores. [13]

3.5. FRAMEWORK Y LENGUAJES A USAR PARA EL DESARROLLO.

Basados en lo anterior, se seleccionó Ruby como lenguaje de programación back-end, ya que es un lenguaje que en la actualidad está cogiendo fuerza y presenta características importantes para el desarrollo que se requiere, la facilidad de manejo, los bajos costos y sus implementaciones interactivas fueron los puntos por los cuales se seleccionó este lenguaje. Y el hecho de que es un lenguaje del que se espera en los próximos años se comience a trabajar de manera mucho más importante en el mercado mundial hizo fijar la atención en él, queriendo estar siempre preparados y conociendo las herramientas que se encuentran a la vanguardia o que aspiran posicionarse allí.

También se tiene en cuenta que para el front-end del desarrollo se va a realizar un híbrido entre los tres tipos de lenguajes que se describieron anteriormente con el fin de crear un diseño interactivo, agradable a la vista y funcional, se toma esta elección teniendo en cuenta la compatibilidad que presta tanto el lenguaje back-end como los lenguajes de front-end y aprovechando esta unión se puede obtener mejores resultados del desarrollo.

Como framework se escogió Rails, ya que facilita mucho la creación de los módulos requeridos para este proyecto y al ser un proyecto corto las características que ofrece rails como la reutilización del código, plantillas estándares para solución de problemas comunes, e interacción y facilidad de configuración entre módulos es más rápida.

4. DESARROLLO HERRAMIENTA DE GESTIÓN DE FORMULARIOS

Para realizar el desarrollo de esta herramienta es necesario definir diferentes aspectos para garantizar la funcionalidad de esta. Dentro de estos aspectos se encuentran la definición del alcance de la herramienta, con el objetivo de aclarar hasta qué punto se llevará a cabo el desarrollo de funcionalidades de esta. También se debe tener muy claro que limitaciones y riesgos se tiene y contar con un plan de contingencia para estos riesgos, ya que uno de ellos puede materializarse en cualquier momento. Además, es muy importante definir cada uno de los requerimientos en conjunto con los interesados en usar la herramienta para conocer de manera detallada cada una de las funcionalidades que tendrá. Una vez definidos y ajustados los requerimientos, se deben realizar los diseños que servirán como base para el desarrollo de la herramienta y finalmente se deben realizar pruebas al desarrollo, para garantizar la funcionalidad correcta de este y verificar el cumplimiento de cada uno de los requerimientos.

4.1. ALCANCE

La aplicación *econtinua* de una manera eficiente, permitirá a los usuarios administradores, docentes y asistentes a los cursos (egresados, estudiantes universitarios, agremiados), la creación y gestión de sus propias cuentas de una manera sencilla e intuitiva. Además, el usuario administrador tendrá la posibilidad de realizar una gestión de los cursos que la asociación de egresados pretenda ofrecer, administrar la inscripción a dichos cursos y recepción de evidencias de pago por medio de la aplicación.

La aplicación web desarrollada proporcionará a los docentes un listado de alumnos con su respectiva información en cada curso que estén dictando. Los usuarios sólo necesitarán realizar el registro en la aplicación una sola vez y desde allí podrán inscribirse en los cursos que estén activos cuando lo deseen.

4.2. LIMITACIONES

- Las limitantes que se pueden presentar durante la ejecución del proyecto y que tiene gran incidencia en el éxito y cumplimiento de este son:
- La disponibilidad horaria de los miembros del equipo, debido a la carga académica y laboral que se tiene durante el mismo tiempo de ejecución del proyecto.

- El poco tiempo de los funcionarios de la ASEUTP

4.3. RIESGOS

4.3.1. Identificación de los riesgos

- Abandono temporal o completo de un miembro del equipo.
- Problemas de comunicación entre los miembros del equipo
- Incoherencias o información incompleta proporcionada por la ASEUTP para ser insertada en el software desarrollado
- Cambios en los requerimientos, que no estaban previstos inicialmente
- Problemas de motivación en el personal
- Accidentes laborales
- Problemas de orden público
- Poca aceptación de la plataforma

La siguiente tabla define la probabilidad de riesgo en 5 calificaciones

Calificación	Descriptor	Descripción
5	Inminente	Se sabe que el suceso sucederá en la mayoría de las circunstancias
4	Probable	El suceso normalmente ocurre
3	Normal	El suceso puede ocurrir en algunas de las circunstancias
2	Improbable	El suceso puede ocurrir pero no suele suceder.
1	Casi imposible	El suceso solo podría ocurrir en circunstancias excepcionales

La siguiente tabla define las escalas de medición de impacto de riesgo en 3 calificaciones:

Calificación	Descriptor	Tiempo
--------------	------------	--------

3	Mayor	30% incremento en el tiempo.
2	Moderado	10% incremento en el tiempo.
1	Insignificante	3% incremento en el tiempo.

4.3.2. Clasificación de los riesgos

#	Riesgo	Probabilidad de ocurrencia	Impacto en el proyecto	Probabilidad por impacto
1	Abandono temporal o completo de un miembro del equipo	3	3	9
2	Problemas de comunicación entre los miembros del equipo	4	3	12
3	Incoherencias o información incompleta proporcionada por la ASEUTP para ser insertada en el software desarrollado	1	1	1
4	Cambios en los requerimientos, que no estaban previstos inicialmente	2	2	4

5	Problemas de motivación en el personal	3	3	9
6	Accidentes Laborales	1	2	2
7	Problemas de orden público	2	1	2
8	Poca aceptación de la plataforma	3	2	6

4.3.3. Análisis de riesgos

Impacto	Insignificante	Moderado	Mayor
Probabilidad	1	2	3
Inminente 5			
Probable 4			2
Posible 3		8	1, 5
Improbable 2	7	4	
Casi Imposible 1	3	6	

Aclaración: Los números que se encuentran en cada casilla corresponden al número asignado a cada riesgo en la tabla anterior nombrada “RIESGOS”.

4.3.4. Plan de mitigación

- Problemas de comunicación entre los miembros del equipo: Hacer reuniones en las cuales

se dé un espacio para dialogar sobre las opiniones del equipo en cuanto a el desarrollo del proyecto, para así poder estar seguros de que se está teniendo en cuenta todas las opiniones y diferentes puntos de vista que se pueden presentar.

- Abandono temporal o permanente de un miembro del equipo: Mantener un buen ambiente en el equipo de trabajo, estar al tanto de su estado de salud, su situación emocional y económica hasta donde sea posible, de tal manera que se pueda ayudar siempre a tiempo en caso de que esto sea necesario.
- Problemas de motivación en el personal: Incentivos y apoyos entre todos los miembros del equipo.
- Poca aceptación de la aplicación: Crear una aplicación amigable e intuitiva y corroborar que esto si se está realizando de esta manera teniendo en cuenta la opinión de los usuarios finales.

4.3.5. Plan de contingencia

- Problemas de comunicación entre los miembros del equipo: Hacer una mediación en donde ambos miembros del equipo y la directora del proyecto cuenten sus inconformidades y a partir de allí buscar la mejor solución.
- Abandono temporal de un miembro del equipo: Conciliación con un acuerdo para la distribución de cargas.
- Abandono completo de un miembro del equipo: Dialogar con la ASEUTP para replantear el alcance del proyecto, puesto que está diseñado para 2 miembros.
- Problemas de motivación en el personal: Hacer una integración de libre esparcimiento para motivar a los integrantes del equipo.
- Problemas de orden público: Realizar reuniones fuera del área de trabajo o realizar encuentros de forma virtual.

- Poca aceptación de la aplicación: Dialogar con la ASEUTP para realizar pequeñas modificaciones a la aplicación para que tenga una mayor acogida, sin que la arquitectura de la aplicación se vea afectada puesto que esto traería otro tipo de problemas.

4.4. DOCUMENTO DE REQUERIMIENTOS

Ver anexo 7. Documento de requerimientos.

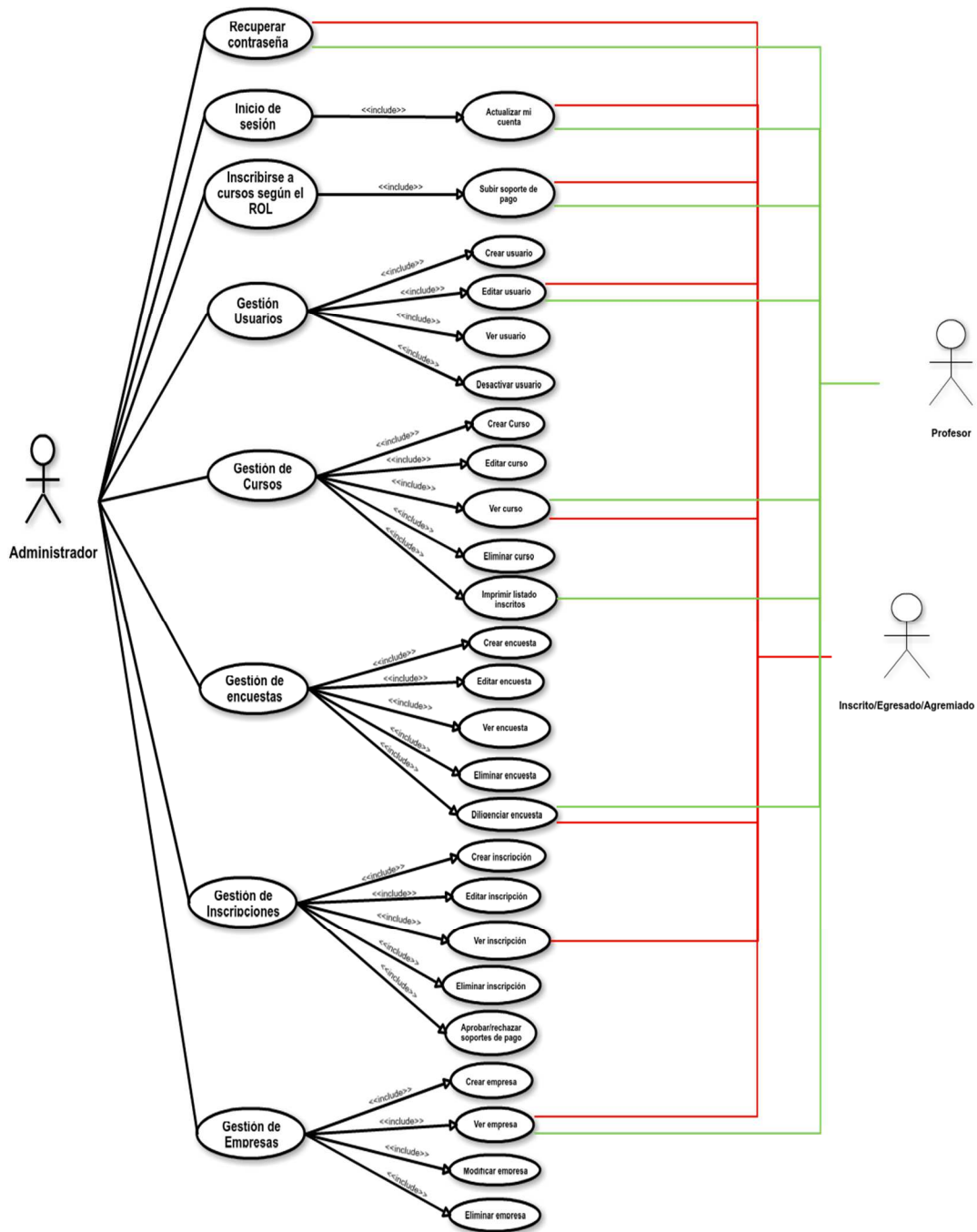
4.5. VISTAS 4+1

4.5.1. Diagrama de Casos de Uso

Nombre: *Figura 1: Diagrama de casos de uso*

Descripción:

El siguiente diagrama muestra la funcionalidad a la cual podrá acceder cada uno de los usuarios según el role asignado, teniendo que cuenta que el requisito principal para para acceder a estas funcionalidades es estar identificado en el sistema mediante el procedimiento de login.



4.5.2. Diagramas de Secuencia

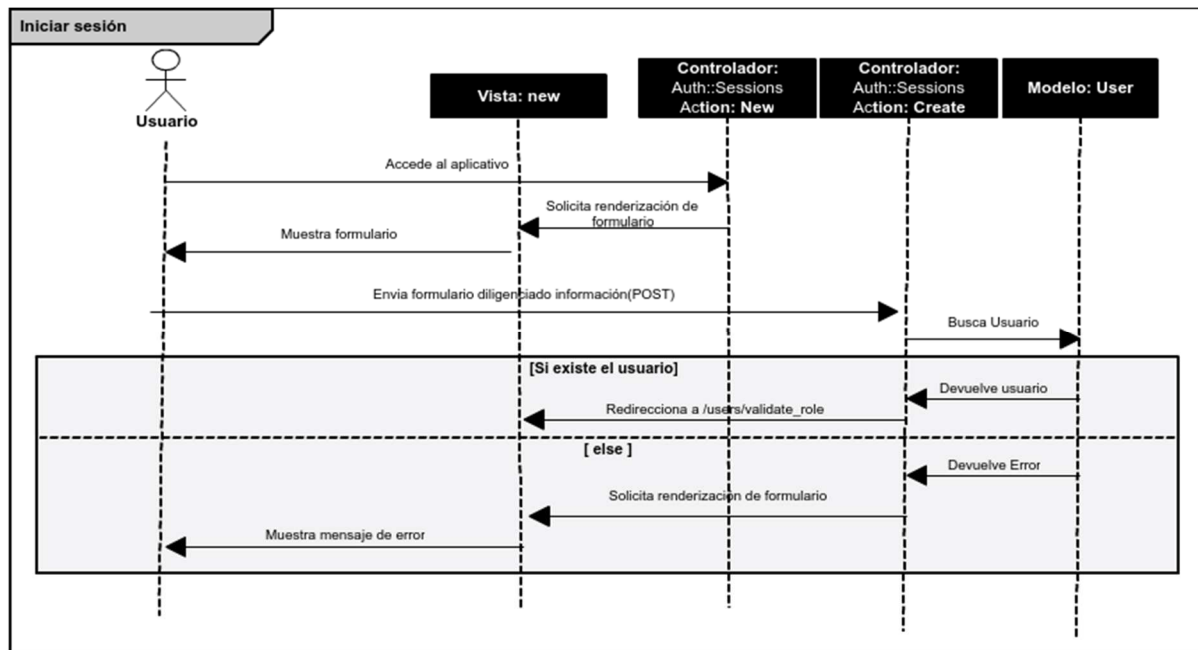
Para ver con una mayor resolución los siguientes diagramas podrá revisar los anexos 3.x.

4.5.2.1. Diagramas de secuencia para manejo de sesión

Nombre: *Figura 2.1: Diagrama secuencia para Inicio de sesión*

Descripción:

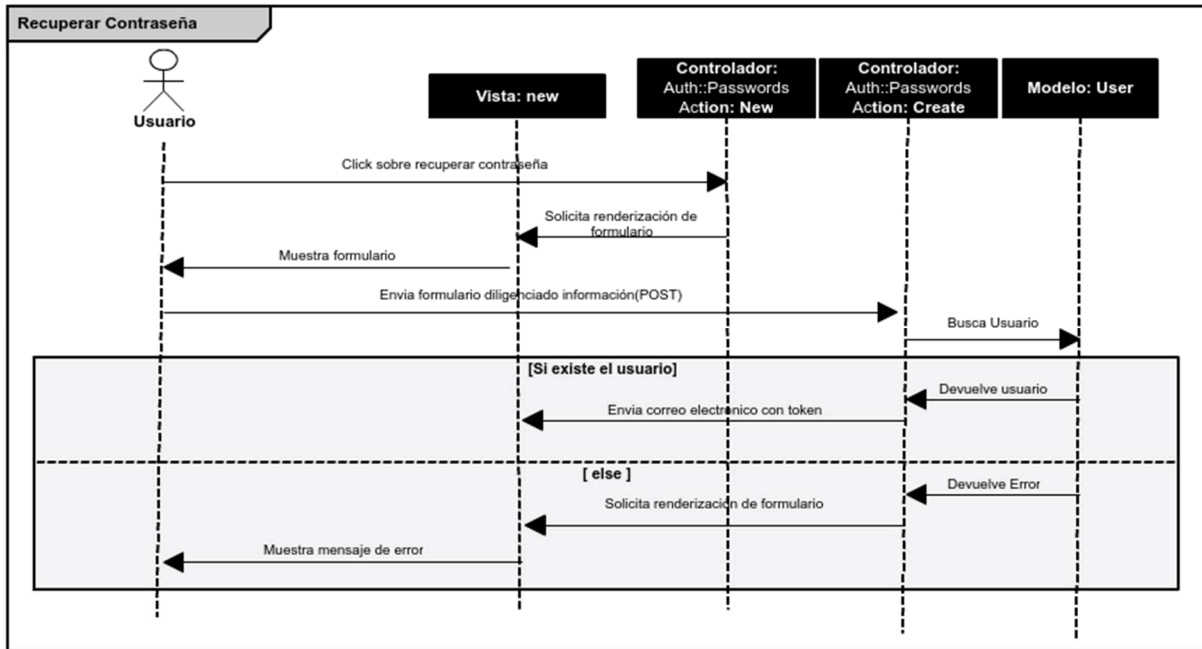
A continuación, se presenta la secuencia lógica para que un usuario realice el inicio de sesión, mostrando cada una de las interacciones entre los diferentes actores y componentes que harán parte de la herramienta.



Nombre: *Figura 2.2: Diagrama secuencia para recuperar contraseña*

Descripción:

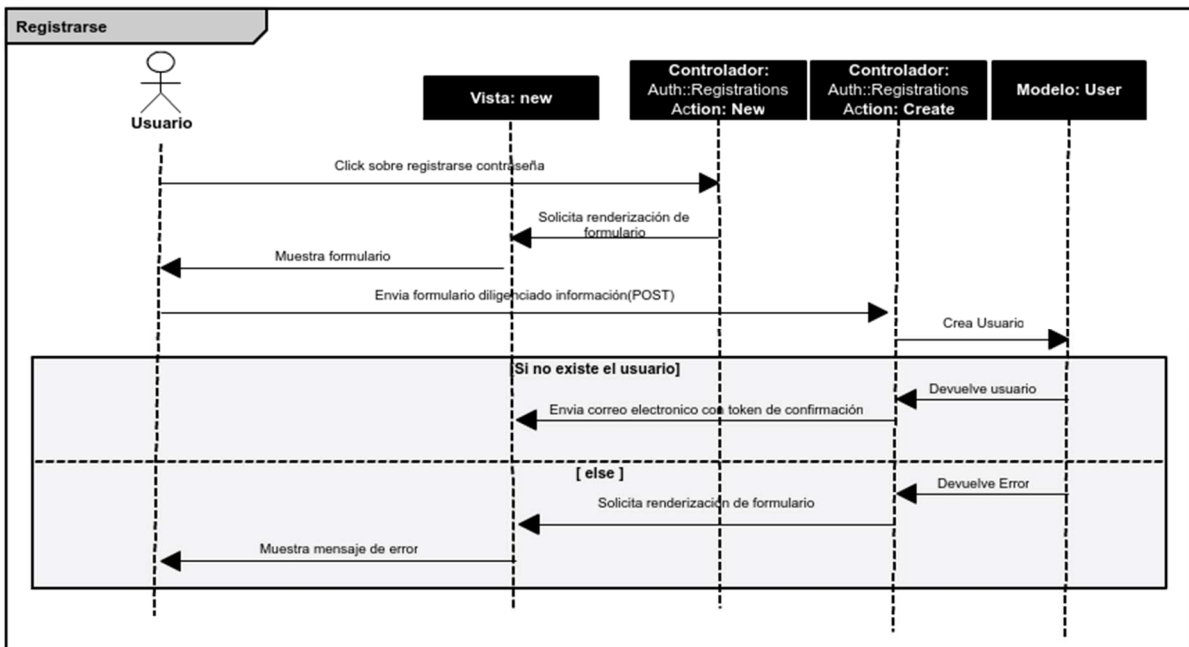
En el siguiente diagrama, se presenta la secuencia lógica para que un usuario recupere su contraseña en caso que la haya olvidado, mostrando cada una de las interacciones entre los diferentes actores y componentes que harán parte de la herramienta.



Nombre: *Figura 2.3: Diagrama secuencia para registrarse*

Descripción:

Continuando con los diagramas de secuencia, se presenta la secuencia lógica para realizar el registro de un nuevo usuario, mostrando cada una de las interacciones entre los diferentes actores y componentes que harán parte de la herramienta.

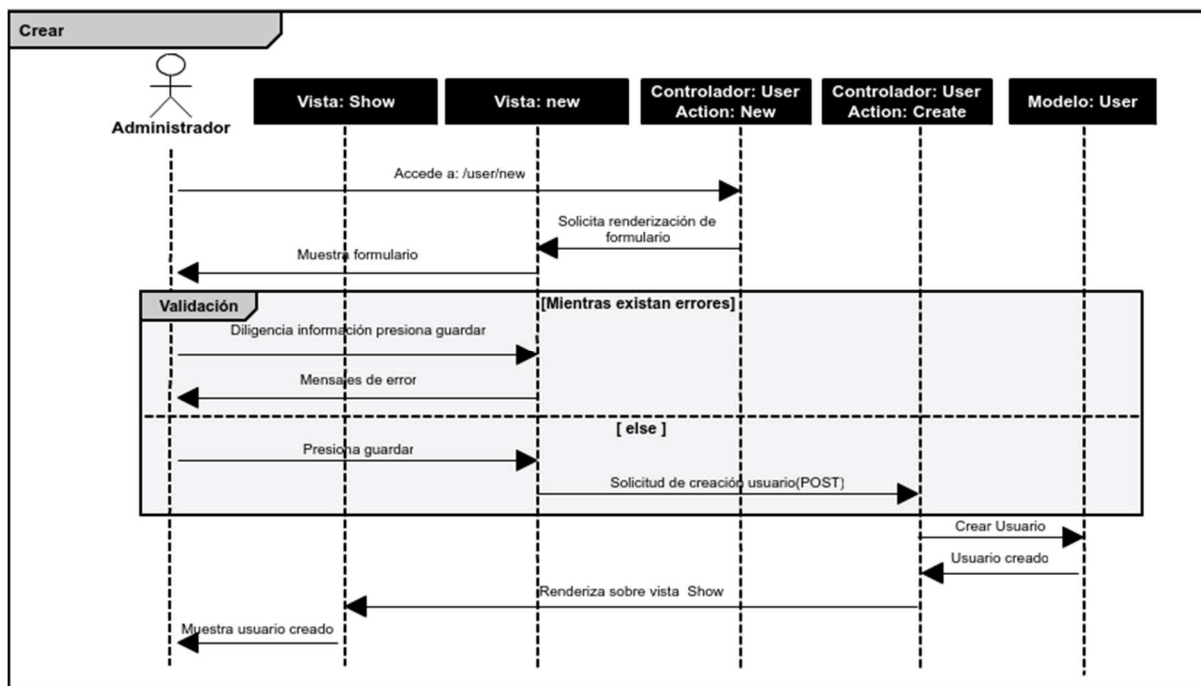


4.5.2.2. Diagramas de gestión de usuarios

Nombre: *Figura 3.1: Diagrama secuencia para crear usuario*

Descripción:

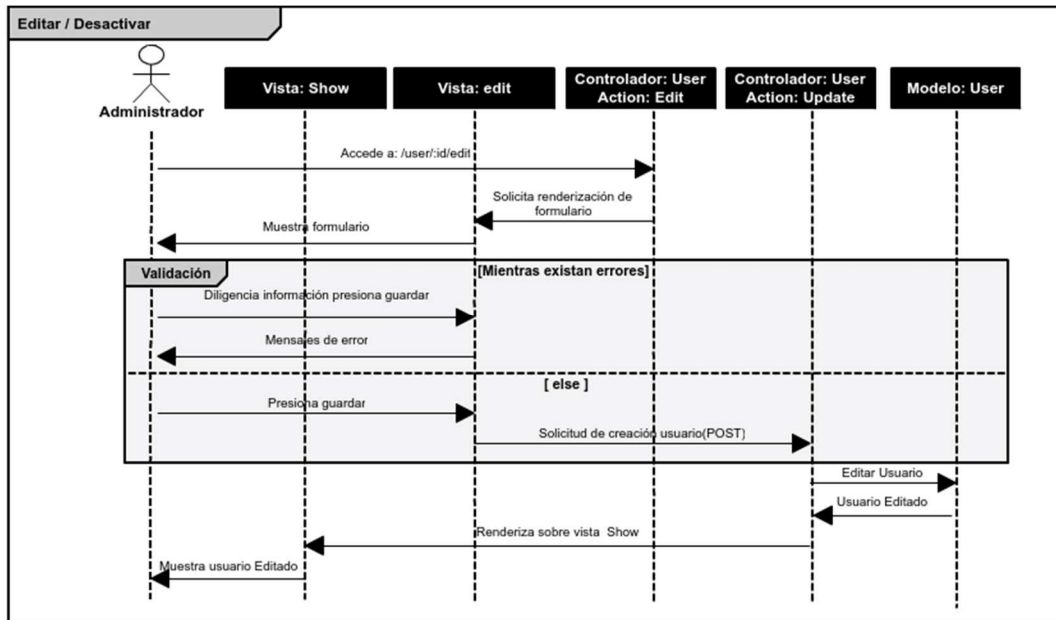
En el siguiente diagrama, se presenta la secuencia lógica para que un usuario administrador realice el registro de otros usuarios, mostrando cada una de las interacciones entre los diferentes actores y componentes que harán parte de la herramienta.



Nombre: *Figura 3.2: Diagrama secuencia para editar/desactivar usuario*

Descripción:

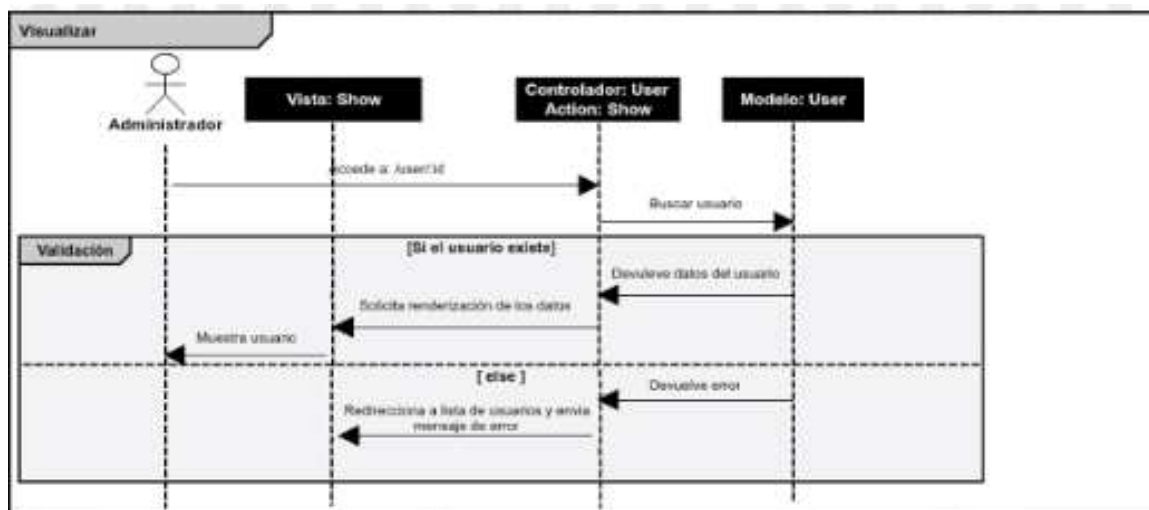
Continuando con los diagramas de secuencia, se presenta la secuencia lógica para que un usuario administrador realice la actualización de un usuario, mostrando cada una de las interacciones entre los diferentes actores y componentes que harán parte de la herramienta.



Nombre: *Figura 3.3: Diagrama secuencia para visualizar usuario*

Descripción:

Continuando con los diagramas de secuencia, se presenta la secuencia lógica para que un usuario administrador visualice la información de un usuario, mostrando cada una de las interacciones entre los diferentes actores y componentes que harán parte de la herramienta.

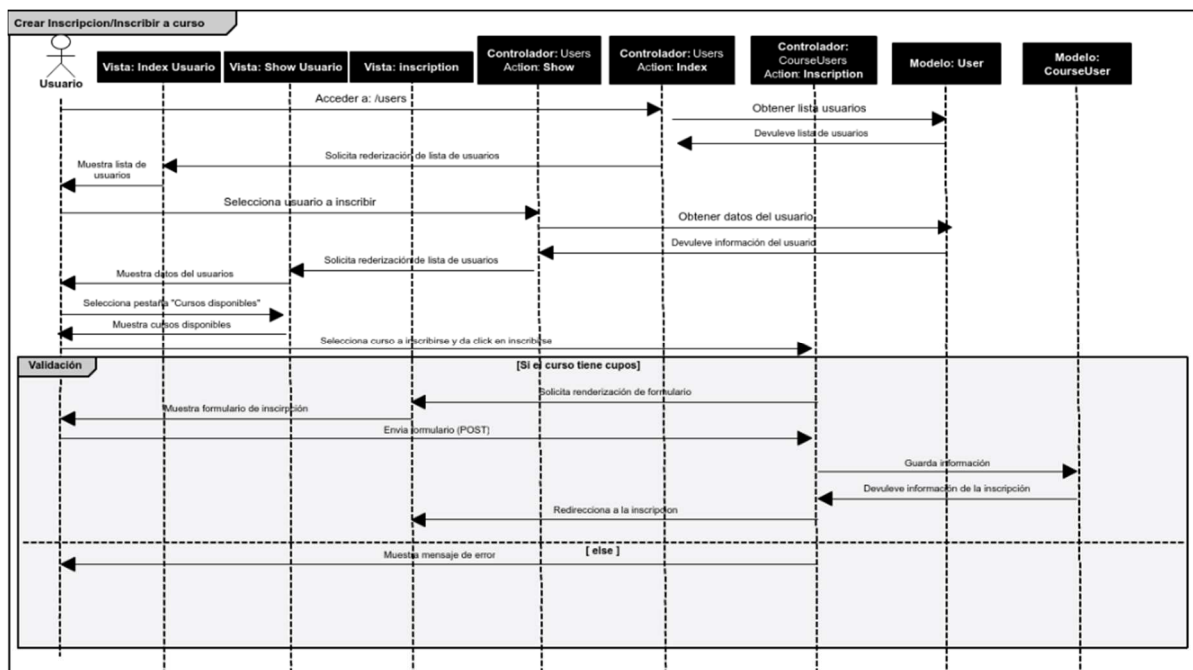


4.5.2.3. Diagramas de gestión de inscripciones

Nombre: *Figura 4.1: Diagrama secuencia para crear inscripción*

Descripción:

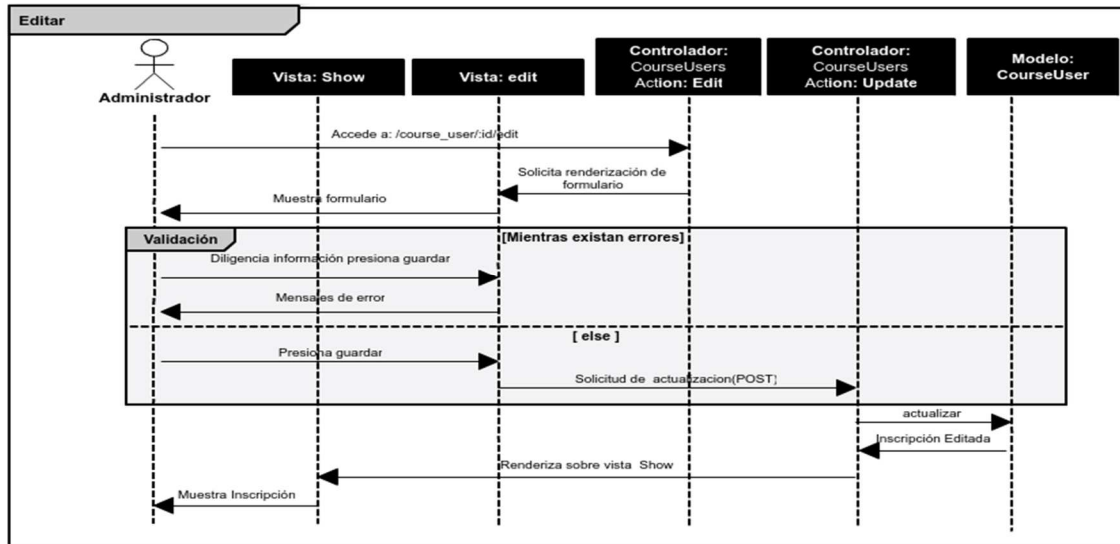
En el siguiente diagrama, se presenta la secuencia lógica para que un usuario administrador realice la inscripción de un usuario a un curso, mostrando cada una de las interacciones entre los diferentes actores y componentes que harán parte de la herramienta.



Nombre: *Figura 4.2: Diagrama secuencia para editar inscripción*

Descripción:

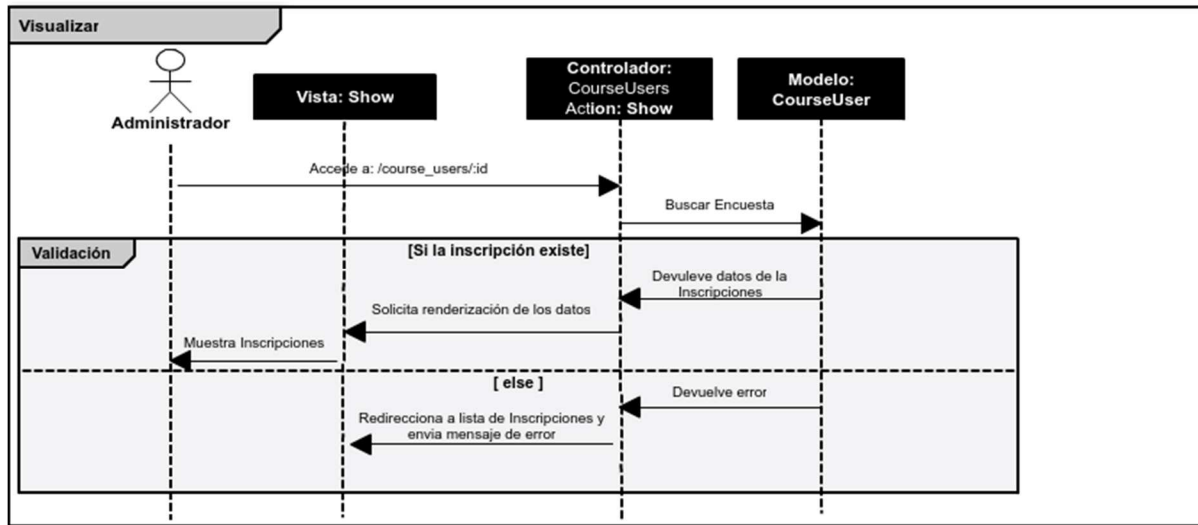
Continuando con los diagramas de secuencia, se presenta la secuencia lógica para que un usuario administrador edite una inscripción de un usuario a un curso, mostrando cada una de las interacciones entre los diferentes actores y componentes que harán parte de la herramienta.



Nombre: *Figura 4.3: Diagrama secuencia para visualizar inscripción*

Descripción:

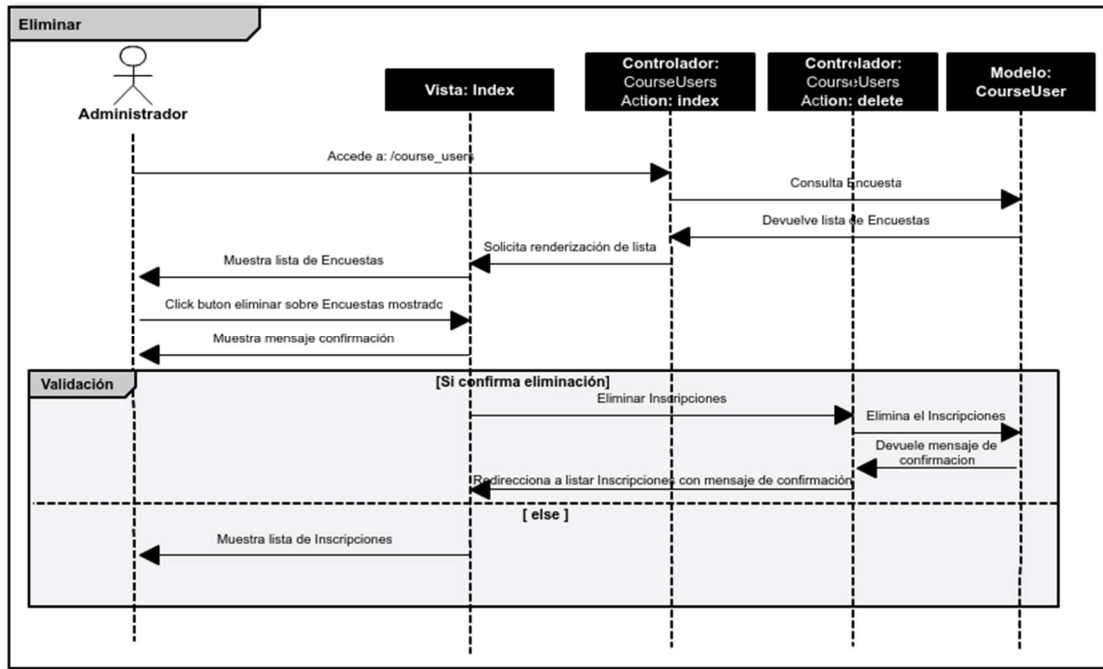
En el siguiente diagrama, se presenta la secuencia lógica para que un usuario administrador visualice una inscripción un curso, mostrando cada una de las interacciones entre los diferentes actores y componentes que harán parte de la herramienta. En esta misma interfaz de visualización se podrán subir los soportes de pago de la inscripción.



Nombre: *Figura 4.4: Diagrama secuencia para eliminar inscripción*

Descripción:

Continuando con los diagramas de secuencia, se presenta la secuencia lógica para que un usuario administrador elimine una inscripción de un usuario a un curso, mostrando cada una de las interacciones entre los diferentes actores y componentes que harán parte de la herramienta.

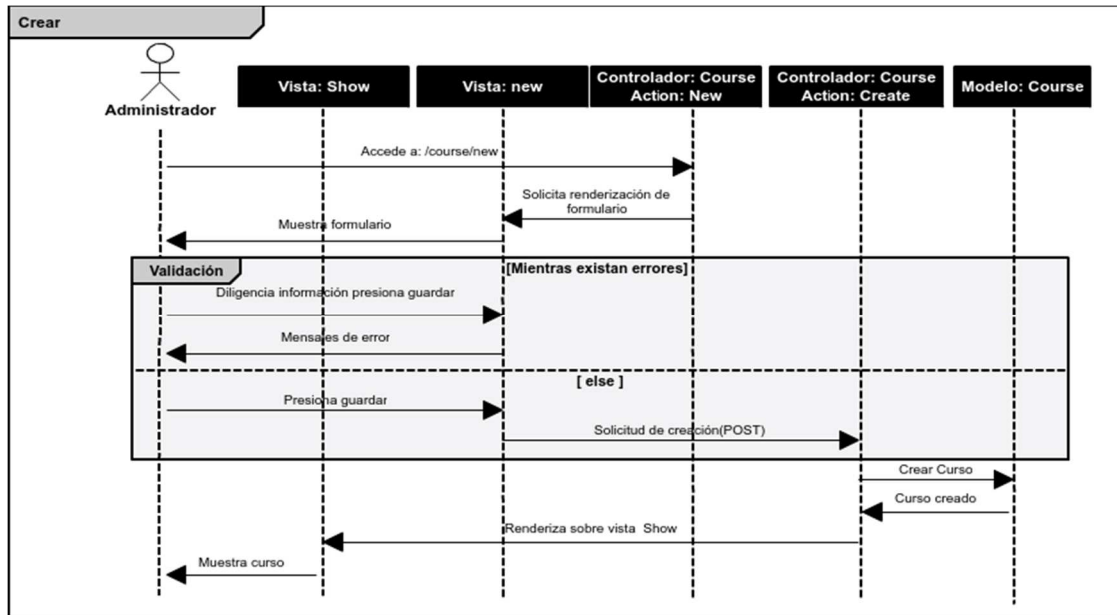


4.5.2.4. Diagramas de gestión de cursos

Nombre: *Figura 5.1: Diagrama secuencia para crear curso*

Descripción:

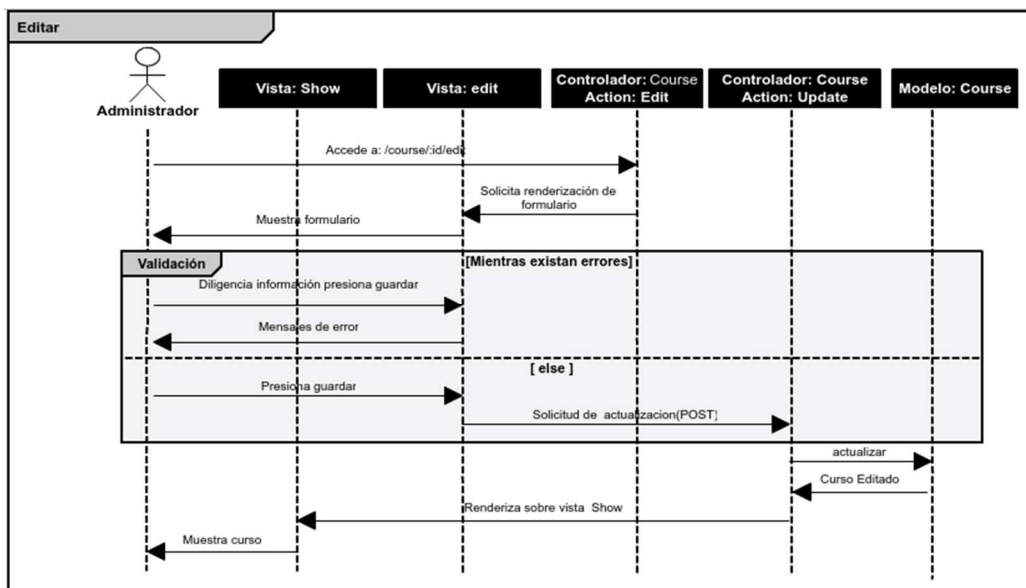
En el siguiente diagrama, se presenta la secuencia lógica para que un usuario administrador cree un nuevo curso, mostrando cada una de las interacciones entre los diferentes actores y componentes que harán parte de la herramienta. En esta misma interfaz de visualización se podrán subir los soportes de pago de la inscripción.



Nombre: *Figura 5.2: Diagrama secuencia para editar curso*

Descripción:

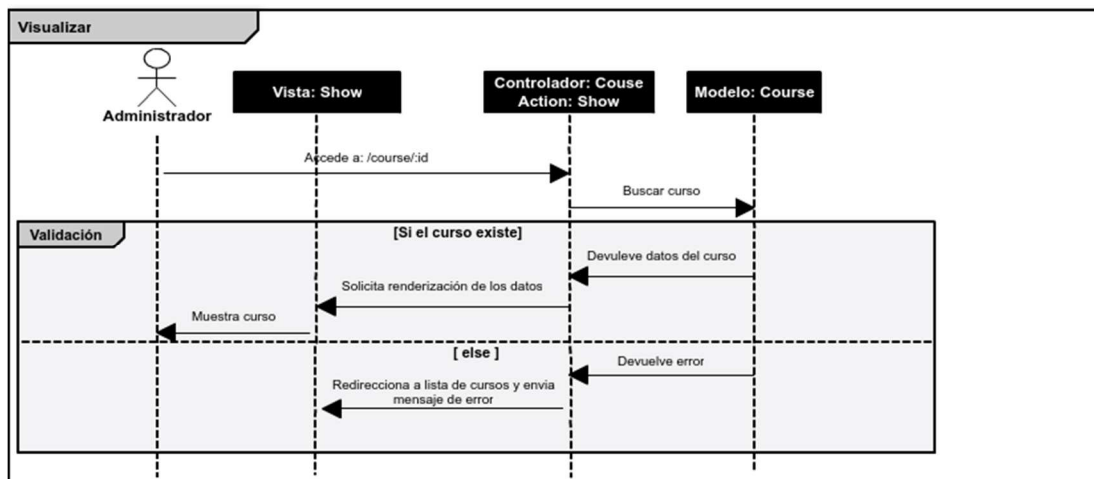
En el siguiente diagrama, se presenta la secuencia lógica para que un usuario administrador edite un curso existente, mostrando cada una de las interacciones entre los diferentes actores y componentes que harán parte de la herramienta. En esta misma interfaz de visualización se podrán subir los soportes de pago de la inscripción.



Nombre: *Figura 5.3: Diagrama secuencia para visualizar curso*

Descripción:

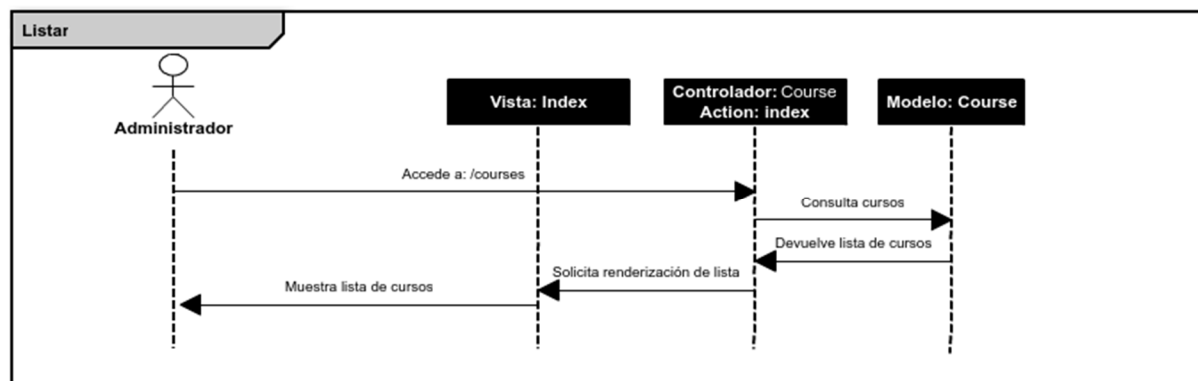
En el siguiente diagrama, se presenta la secuencia lógica para que un usuario administrador visualice un curso existente, mostrando cada una de las interacciones entre los diferentes actores y componentes que harán parte de la herramienta. En esta misma interfaz de visualización se podrán subir los soportes de pago de la inscripción.



Nombre: *Figura 5.4: Diagrama secuencia para listar curso*

Descripción:

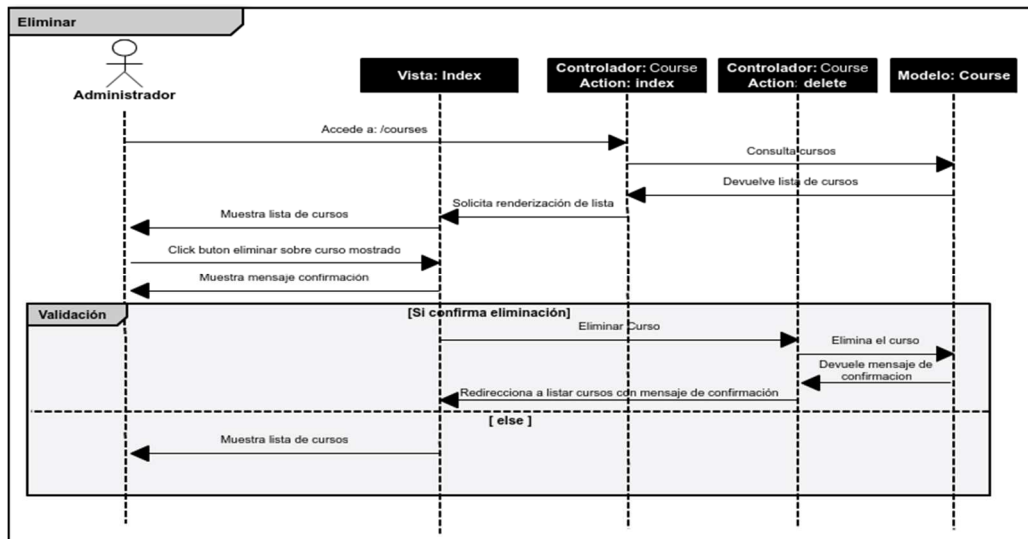
En el siguiente diagrama, se presenta la secuencia lógica para que un usuario administrador listar los cursos existentes en el sistema, mostrando cada una de las interacciones entre los diferentes actores y componentes que harán parte de la herramienta. En esta misma interfaz de visualización se podrán subir los soportes de pago de la inscripción.



Nombre: *Figura 5.5: Diagrama secuencia para eliminar curso*

Descripción:

A continuación, se presenta la secuencia lógica para que un usuario administrador eliminar un curso existente del sistema, mostrando cada una de las interacciones entre los diferentes actores y componentes que harán parte de la herramienta. En esta misma interfaz de visualización se podrán subir los soportes de pago de la inscripción.

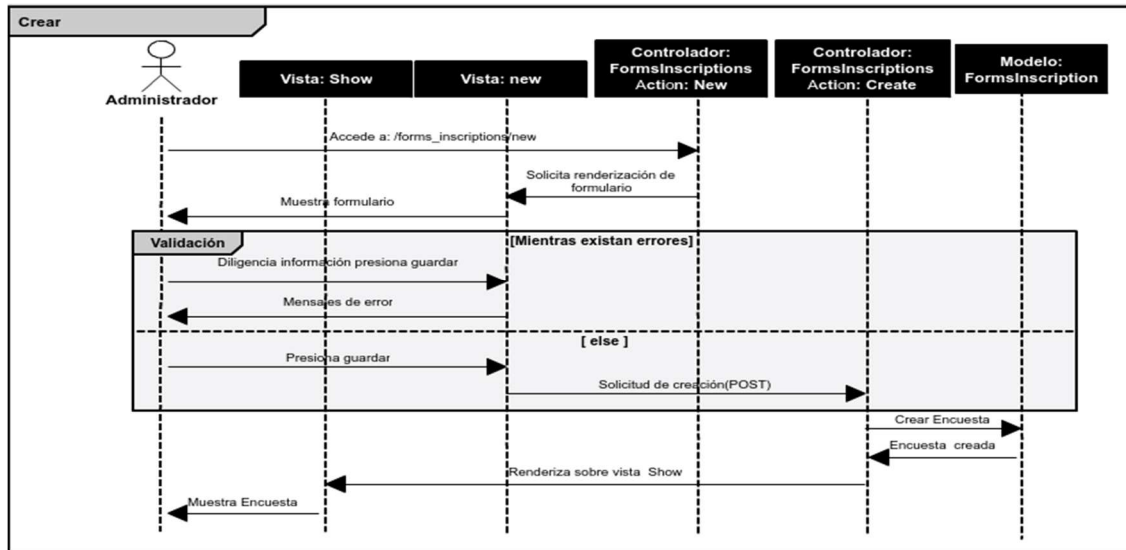


4.5.2.5. Diagramas de gestión de encuestas

Nombre: *Figura 6.1: Diagrama secuencia para crear encuesta*

Descripción:

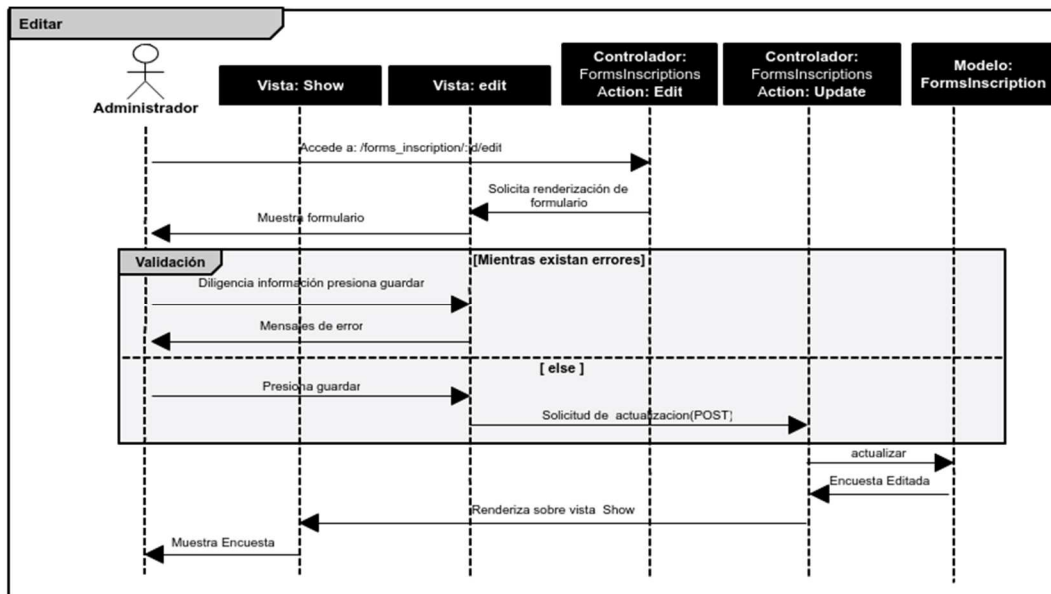
A continuación, se presenta la secuencia lógica para que un usuario administrador cree una encuesta en el sistema, mostrando cada una de las interacciones entre los diferentes actores y componentes que harán parte de la herramienta. En esta misma interfaz de visualización se podrán subir los soportes de pago de la inscripción.



Nombre: *Figura 6.2: Diagrama secuencia para editar encuesta*

Descripción:

A continuación, se presenta la secuencia lógica para que un usuario administrador edite una encuesta en el sistema, mostrando cada una de las interacciones entre los diferentes actores y componentes que harán parte de la herramienta. En esta misma interfaz de visualización se podrán subir los soportes de pago de la inscripción.

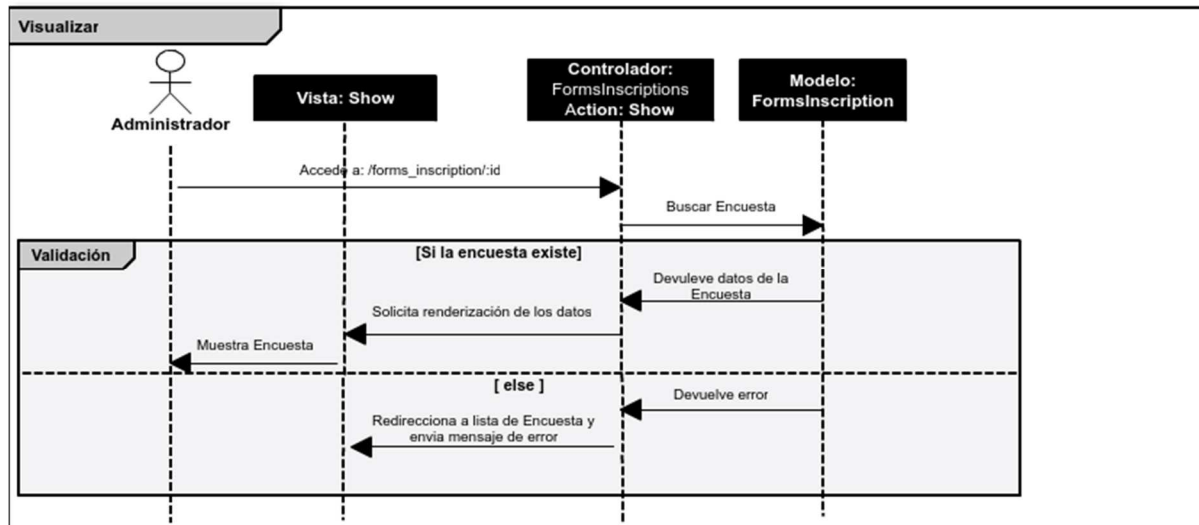


Nombre: *Figura 6.3: Diagrama secuencia para visualizar encuesta*

Descripción:

El en el siguiente diagrama, se muestra la secuencia lógica para que un usuario administrador

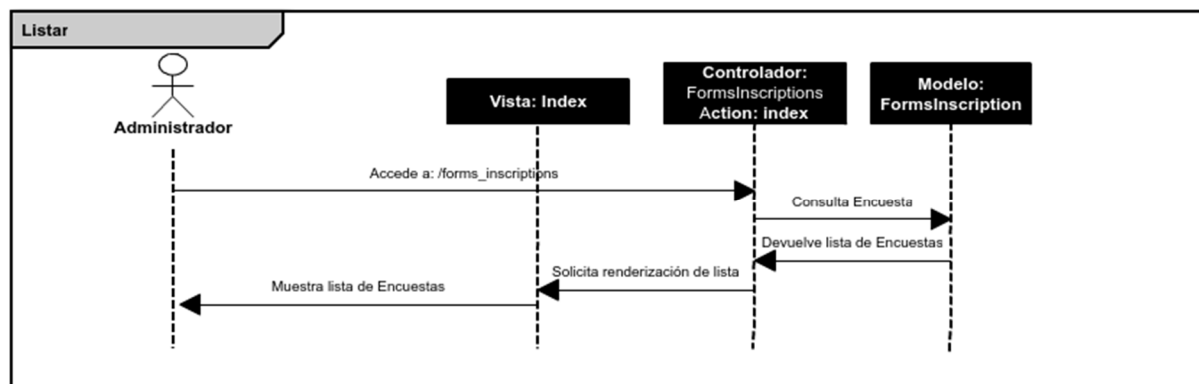
visualice una encuesta en el sistema, mostrando cada una de las interacciones entre los diferentes actores y componentes que harán parte de la herramienta. En esta misma interfaz de visualización se podrán subir los soportes de pago de la inscripción.



Nombre: *Figura 6.4: Diagrama secuencia para listar encuestas*

Descripción:

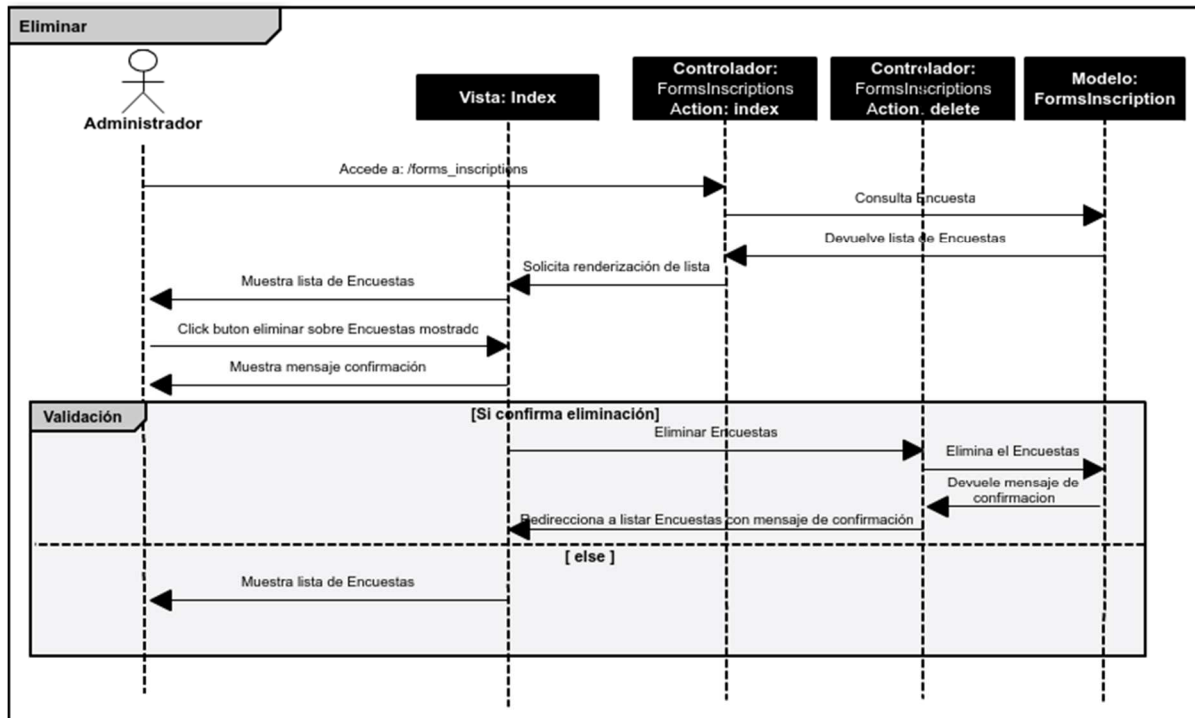
El en el siguiente diagrama, se muestra la secuencia lógica para que un usuario administrador visualice las encuestas existentes en el sistema como una lista, mostrando cada una de las interacciones entre los diferentes actores y componentes que harán parte de la herramienta. En esta misma interfaz de visualización se podrán subir los soportes de pago de la inscripción.



Nombre: *Figura 6.5: Diagrama secuencia para eliminar encuestas*

Descripción:

A continuación, se muestra la secuencia lógica para que un usuario administrador elimine una encuesta existente en el sistema, mostrando cada una de las interacciones entre los diferentes actores y componentes que harán parte de la herramienta. En esta misma interfaz de visualización se podrán subir los soportes de pago de la inscripción.

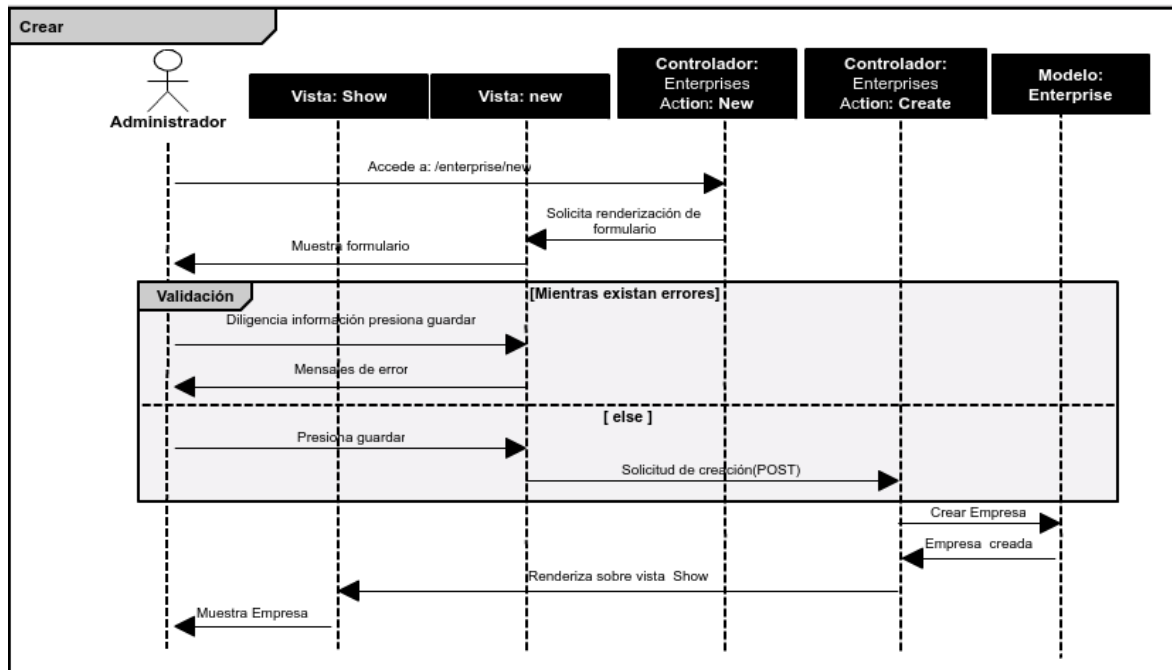


4.5.2.6. Diagramas de gestión de empresas

Nombre: *Figura 7.1: Diagrama secuencia para crear empresa*

Descripción:

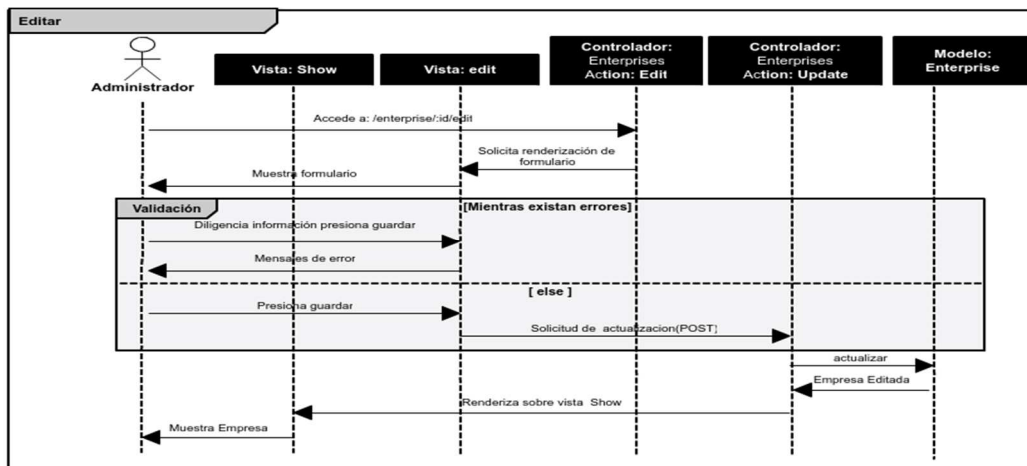
A continuación, se muestra la secuencia lógica para que un usuario administrador crear una empresa nueva en el sistema, mostrando cada una de las interacciones entre los diferentes actores y componentes que harán parte de la herramienta. En esta misma interfaz de visualización se podrán subir los soportes de pago de la inscripción.



Nombre: *Figura 7.2: Diagrama secuencia para editar empresa*

Descripción:

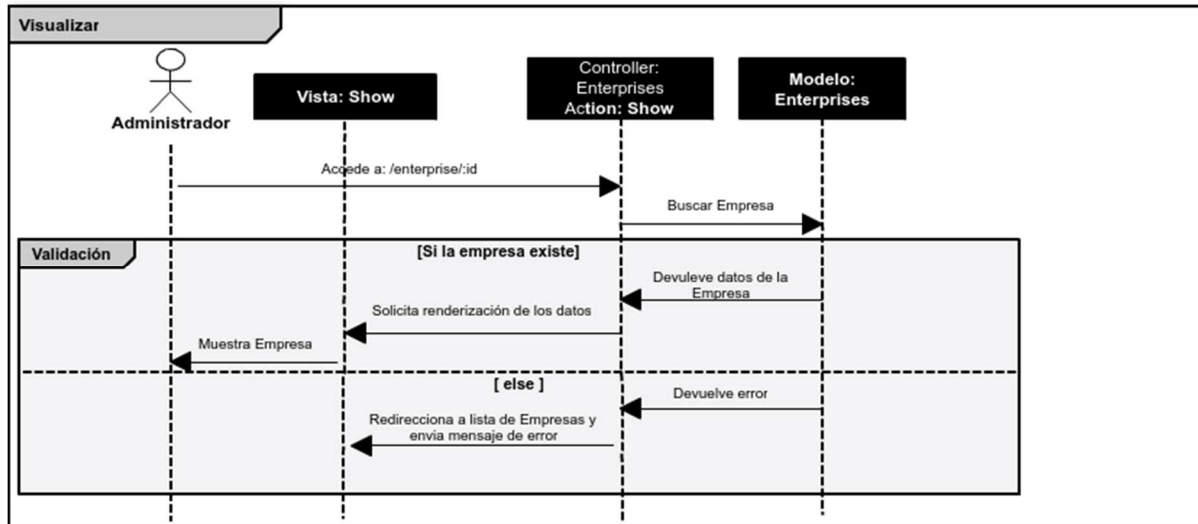
A continuación, se muestra la secuencia lógica para que un usuario administrador editar una empresa en el sistema, mostrando cada una de las interacciones entre los diferentes actores y componentes que harán parte de la herramienta. En esta misma interfaz de visualización se podrán subir los soportes de pago de la inscripción.



Nombre: *Figura 7.3: Diagrama secuencia para visualizar empresa*

Descripción:

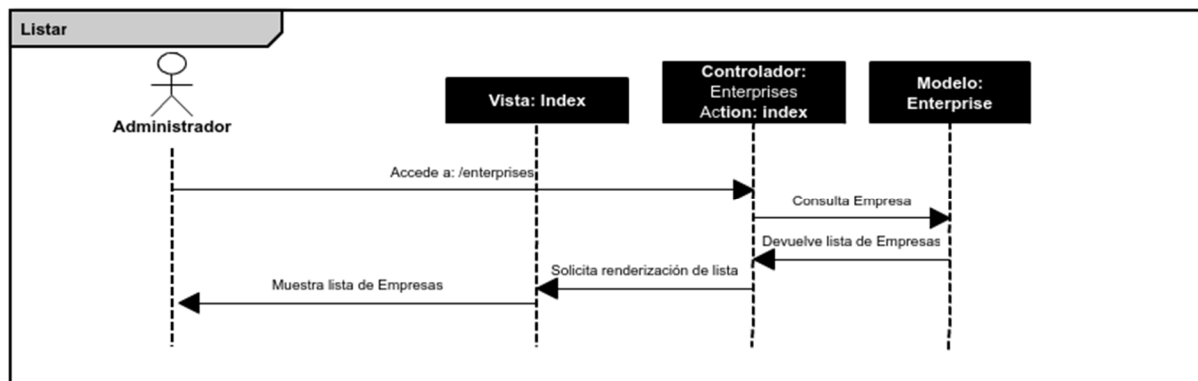
En el siguiente diagrama, se muestra la secuencia lógica para que un usuario administrador visualizar una empresa en el sistema, mostrando cada una de las interacciones entre los diferentes actores y componentes que harán parte de la herramienta. En esta misma interfaz de visualización se podrán subir los soportes de pago de la inscripción.



Nombre: *Figura 7.4: Diagrama secuencia para listar empresas*

Descripción:

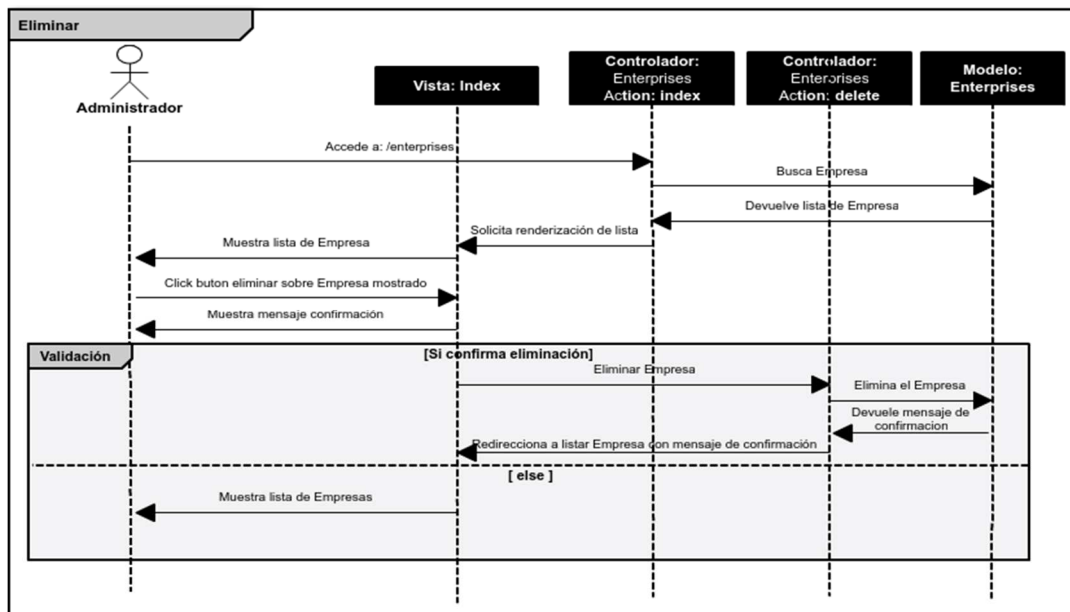
A continuación, se muestra la secuencia lógica para que un usuario administrador listar las empresas en el sistema, mostrando cada una de las interacciones entre los diferentes actores y componentes que harán parte de la herramienta. En esta misma interfaz de visualización se podrán subir los soportes de pago de la inscripción.



Nombre: *Figura 7.5: Diagrama secuencia para eliminar empresa*

Descripción:

A continuación, se muestra la secuencia lógica para que un usuario administrador eliminar una empresa del sistema, mostrando cada una de las interacciones entre los diferentes actores y componentes que harán parte de la herramienta. En esta misma interfaz de visualización se podrán subir los soportes de pago de la inscripción.



4.5.2.7. Diagrama de Clases

Ver Anexos:

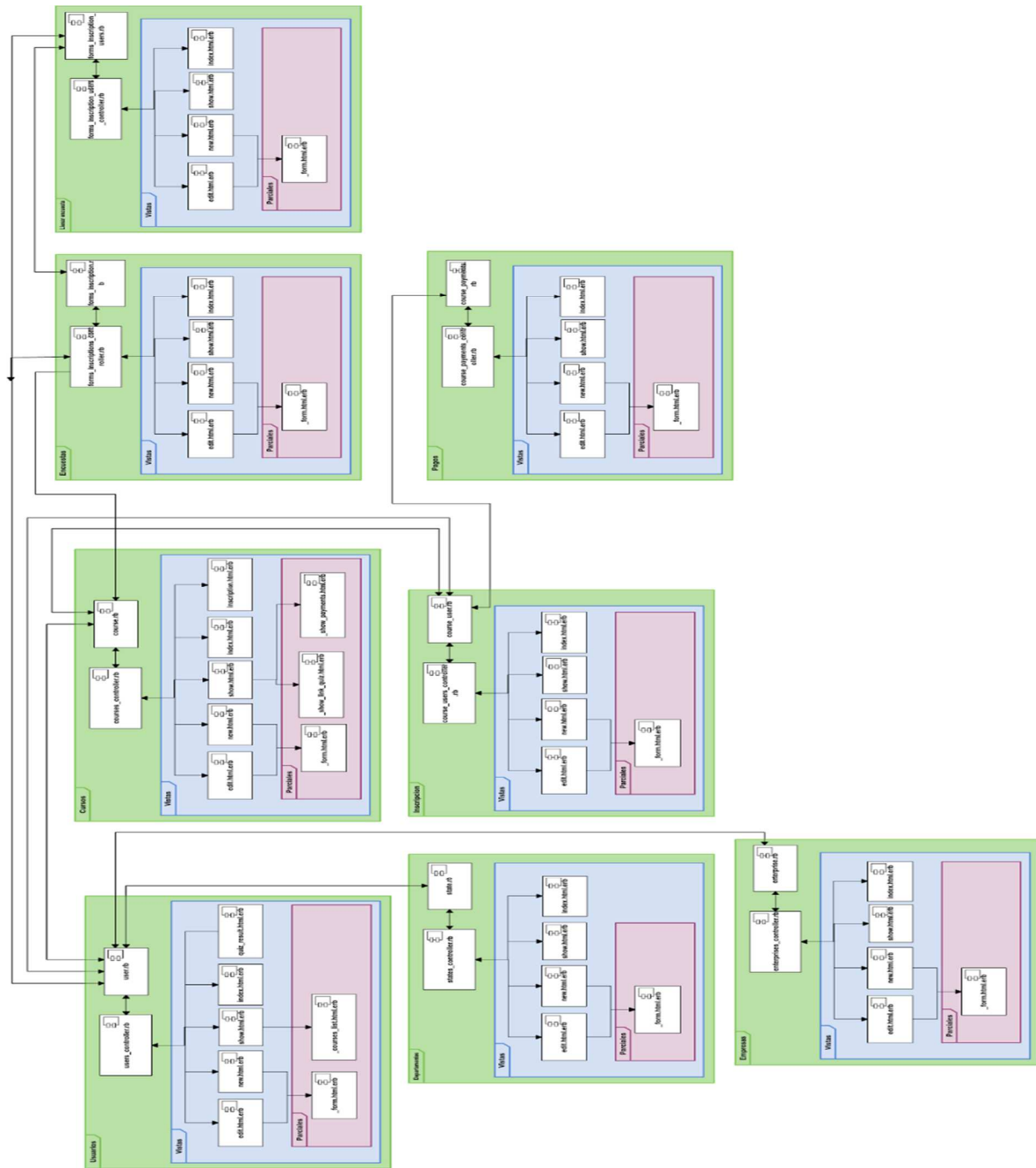
- Anexo 4.1: Diagrama de Clases - Sólo controladores
- Anexo 4.2: Diagrama de Clases - Controladores + Métodos
- Anexo 4.3: Diagrama de Clases - Solo Modelos
- Anexo 4.4: Diagrama de Clases - Modelos + Atributos

4.5.3. Diagrama de Componentes

Nombre: *Figura 8: Diagrama de componentes*

Descripción:

El siguiente diagrama contiene información de los diferentes componentes del sistema y la manera cómo interactúan entre sí.

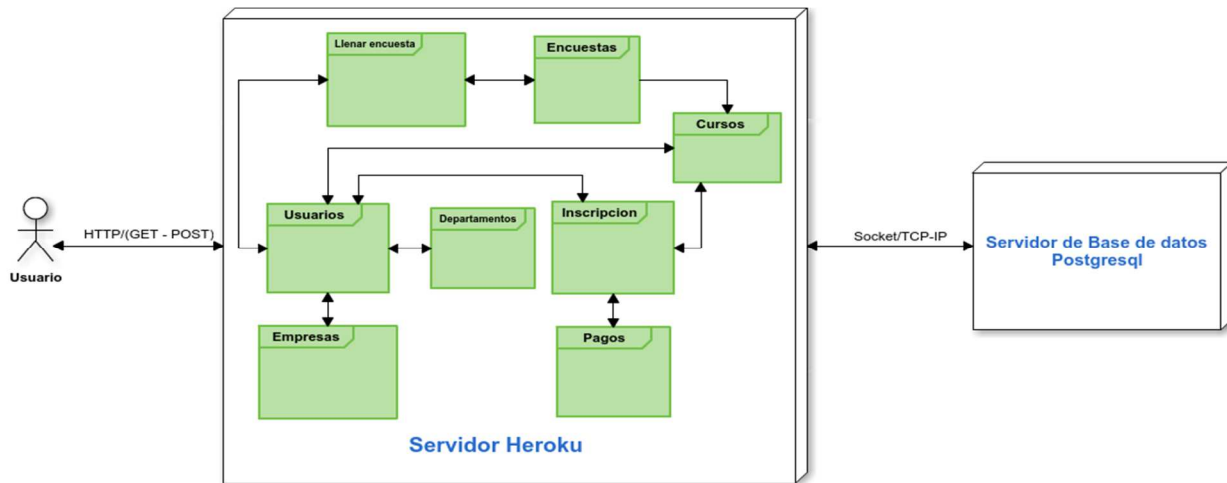


4.5.4. Diagrama de Despliegue

Nombre: *Figura 9: Diagrama de despliegue*

Descripción:

En este diagrama se muestra la manera como interactúa el hardware y la comunicación con el Usuario.



4.6. PLAN DE PRUEBAS

4.6.1. Identificador plan de pruebas

PPGF-01

4.6.2. Introducción

Este documento contiene el plan de pruebas para el proyecto de la ASEUTP - Gestor de Formularios. Este plan describirá las actividades a realizar para las pruebas del proyecto en el cual se incluirán los elementos afectos para el producto. El proyecto tendrá pruebas de cajas negras. El objetivo principal es que los módulos de software y la integración de los mismos cumplan con los requisitos técnicos.

4.6.3. Elementos

4.6.3.1. Software que se van a probar.

4.6.3.1.1. MODULO SESION

Sub. Modulo - Registro de usuario:

Se ingresan todos los datos que son requeridos para realizar el registro, el registro requiere que los campos se diligencien correctamente.

Sub. Modulo – Inicio de sesión:

Se ingresa los campos requeridos para iniciar una sesión, los permisos de usuario dependen del rol de la cuenta.

Sub. Modulo - Cerrar sesión:

Se selecciona la opción de Salir para finalizar la sesión por medio de un botón que retorna a la página de inicio de sesión.

Sub. Modulo - Recuperar contraseña:

En la interfaz de inicio de sesión se brinda la opción de recuperar contraseña por medio del correo electrónico.

4.6.3.1.2. MODULO ADMINISTRADOR

Sub. Modulo – Usuarios:

Permite visualizar la lista de todos los usuarios en el sistema y desde allí el administrador podrá hacer una búsqueda por filtros o visualizar en detalle algún usuario o editarlo. También puede crear nuevos usuarios.

Sub. Modulo – Cursos:

Permite visualizar la lista de todos los cursos en el sistema, crear nuevos cursos, realizar búsquedas por filtro, o hacer una gestión de cada curso (ver, editar o eliminar el curso) y también listar los estudiantes inscritos a el curso, a todos los cursos y el resultado de las encuestas por cada curso.

Sub. Modulo – Encuestas:

Permite visualizar la lista de todas las encuestas en el sistema, hacer una gestión de cada encuesta (ver, crear, editar o eliminar la encuesta), cada encuesta se puede asignar a uno, ninguno o todos los profesores y/o cursos.

Sub. Modulo – Inscripciones Realizadas:

Este sub modulo, lista todos los usuarios con inscripciones a un curso mostrando el estado de la inscripción y permite realizar la visualización de los pagos realizados y desde allí, aprobarlos o rechazarlos, agregar pagos a algún usuario, y además desde allí se puede editar el estado del usuario en el curso (activo, pendiente, certificado, no certificado).

Sub. Modulo – Empresas:

Permite visualizar el listado de todas las empresas en el sistema, realizar búsquedas por filtros y

además permite hacer una gestión (crear, editar, visualizar y eliminar) cada una de ellas.

4.6.3.1.3. MODULO EGRESADO

Sub. Modulo – Información Personal:

Este sub modulo permite visualizar la información personal registrada durante la creación de la cuenta.

Sub. Modulo – Información Empresa:

Permite visualizar la información de la empresa relacionada con la cuenta.

Sub. Modulo – Información de login:

Este sub modulo permite visualizar la información personal registrada durante la creación de la cuenta para el inicio de sesión.

Sub. Modulo – Cursos disponibles:

Permite visualizar la información de los cursos disponibles a la fecha y también a los que esté inscrito. En los cursos disponibles se puede solicitar la inscripción y de allí diligenciar el formulario requerido. En los cursos inscritos se puede visualizar los pagos realizados a la fecha, o agregar nuevos pagos.

4.6.3.1.4. MODULO AGREMIADO

Sub. Modulo – Información Personal:

Este sub modulo permite visualizar la información personal registrada durante la creación de la cuenta.

Sub. Modulo – Información Empresa:

Permite visualizar la información de la empresa relacionada con la cuenta.

Sub. Modulo – Información de login:

Este sub modulo permite visualizar la información personal registrada durante la creación de la cuenta para el inicio de sesión.

Sub. Modulo – Cursos disponibles:

Permite visualizar la información de los cursos disponibles a la fecha y también a los que esté inscrito. En los cursos disponibles se puede solicitar la inscripción y de allí diligenciar el formulario requerido. En los cursos inscritos se puede visualizar los pagos realizados a la fecha, o agregar nuevos pagos.

4.6.3.1.5. MODULO PROFESOR

Sub. Modulo – Información Personal:

Este sub modulo permite visualizar la información personal registrada durante la creación de la cuenta.

Sub. Modulo – Información Empresa:

Permite visualizar la información de la empresa relacionada con la cuenta.

Sub. Modulo – Información de login:

Este sub modulo permite visualizar la información personal registrada durante la creación de la cuenta para el inicio de sesión.

Sub. Modulo – Cursos disponibles:

Permite visualizar la información de los cursos disponibles a la fecha y también a los que esté inscrito. En los cursos disponibles se puede solicitar la inscripción y de allí diligenciar el formulario requerido. En los cursos inscritos se puede visualizar los pagos realizados a la fecha, o agregar nuevos pagos.

Sub. Modulo – Cursos y encuestas:

Este sub modulo permite visualizar las encuestas por curso del usuario profesor, si se desea se podrá abrir cada encuesta y ver en detalles sus resultados.

Sub. Modulo – Mis cursos:

Este sub modulo permite visualizar el listado de los cursos asignados a un profesor, realizar búsqueda a través de filtros y permite las opciones de ver los estudiantes en el curso o ver los detalles del curso.

4.6.3.2. Documentos a probar

Para este proyecto no se realizarán pruebas sobre documentos.

4.6.3.3. Características que se van a probar.

1. Tiempos de respuestas
2. Concurrencia de usuarios
3. Disponibilidad
4. Usabilidad
5. Seguridad

4.6.3.4. Características que no se prueba.

- Condiciones de error no detectadas.
- Verificar confiabilidad de los datos.
- Capacidad de adaptación a diferentes resoluciones de pantalla.

4.6.3.5. Enfoque.

Pruebas de caja negra:

Se prueban las funcionalidades de cada módulo con las respectivas entradas de datos y salidas de información.

Se usarán diferentes herramientas como lo son:

- OWASP (PRUEBAS DE SEGURIDAD)
- ASM-CA (TIEMPO DE RESPUESTA)
- LOADIMPACT(CONCURRENCIAS)
- TAWDIS (USABILIDAD)

Para la disponibilidad se tiene en cuenta que al trabajar sobre una cuenta gratuita en el servidor heroku, nos limitamos a su servicio.

Y se probara todo siguiendo el documento de casos de pruebas.

4.6.3.6. Criterios de paso/fallo para cada elemento.

- Si la prueba falla 3 veces en un mismo modulo se considerará fallida.
- Si la prueba no tiene errores en 3 repeticiones consecutivas se considera que paso.

4.6.3.7. Criterios de suspensión y requisitos de reanudación.

- Si está fallando la conexión con la base de datos se suspenden las pruebas.
- Si se presentan fallos consecutivos en respuesta al servidor se suspenden las pruebas que se están realizando.
- Para reanudar las pruebas se necesitará que el desarrollo se corrija pertinentemente.
- Si pasa el 80% de todas las pruebas no críticas, serán aprobadas.

4.6.3.8. Documentos a entregar.

- Plan de pruebas.
- Diseño, ejecución y resultado de pruebas.

4.6.3.9. Actividades de preparación y ejecución de pruebas.

Organización de Equipos

Jefe de equipo

Fernando Santa López – David Pulgarin Valencia

- Preparación de casos de pruebas
- Ejecución de pruebas
- Datos de la prueba
- Preparar informe de resultados de pruebas

4.6.3.10. Necesidades de entorno.

En cuanto a:

SOFTWARE y HARDWARE:

Sistema operativo Linux

HERRAMIENTAS DE PRUEBA:

- OWASP
- SAHI
- ASM-CA
- LOADIMPACT
- TAWDIS

DOCUMENTACION:

Absoluta comodidad, tranquilidad.

4.6.3.11. Responsabilidades en la organización y realización de las pruebas.

Pruebas de software: Fernando Santa López – David Pulgarin Valencia.

4.6.3.12. Necesidades de personal y de formación.

Que sepa la utilización de sistemas operativos (Linux o windows) y lenguaje de programación (sahi).

4.6.3.13. Esquema de tiempos

Fecha	Tareas a realizar	Responsable
12-01-2017	Diseño primer ciclo de pruebas	Fernando Santa López – David Pulgarin Valencia
13-01-2017	Ejecución y reporte primer ciclo de pruebas	
14-01-2017	Diseño segundo ciclo de pruebas	
14-01-2017	Ejecución y reporte segundo ciclo de pruebas	
15-01-2017	Diseño tercer ciclo de pruebas(de ser necesario)	
15-01-2017	Ejecución y reporte tercer ciclo de pruebas(de ser necesario)	
15-01-2017	Ejecución y reporte tercer ciclo de pruebas(de ser necesario)	

4.6.3.14. Aprobaciones.

Los criterios a manejar para aprobar los casos de prueba son:

- Que la respuesta del caso de prueba sea correcta.
- Se responda a un requerimiento del sistema.
- Que se cumpla con los objetivos del caso de la prueba.

Los criterios de rechazo para los casos de prueba son:

- Que no se haya entregado la funcionalidad que se mostraba con el caso de prueba
- Que no se haya realizado una prueba detallada

4.7. EJECUCIÓN Y REPORTE DE PRUEBAS

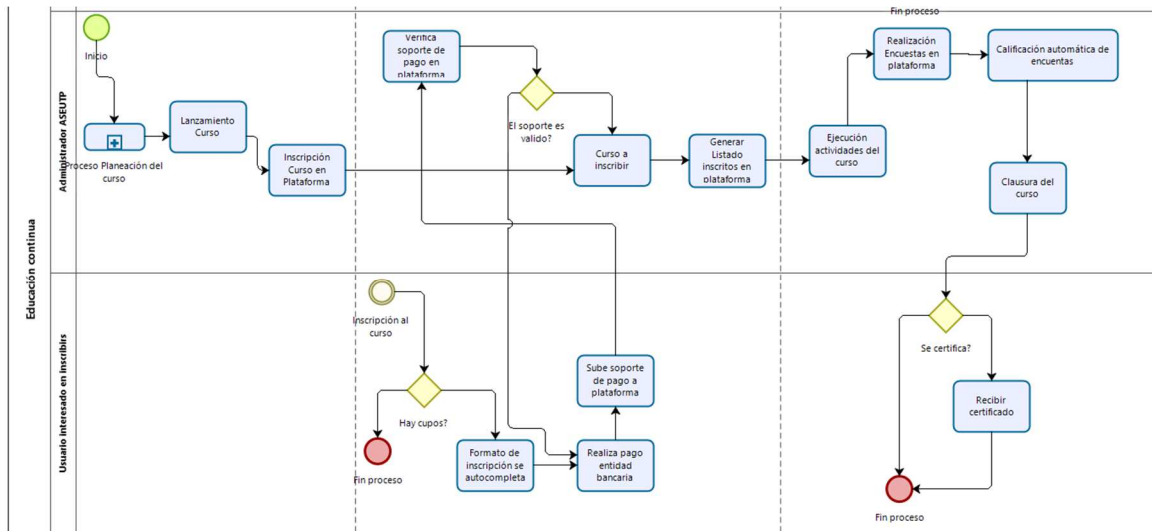
Ver Anexo 1: Diseño-ReportePruebas

5. EVALUACIÓN DE MEJORA

Nombre: Nuevo *Diagrama de flujo del proceso educación continua*

Descripción:

El siguiente diagrama de flujo muestra el procedimiento llevado a cabo para la inscripción de un curso, teniendo en cuenta desde la planeación del curso hasta su clausura.



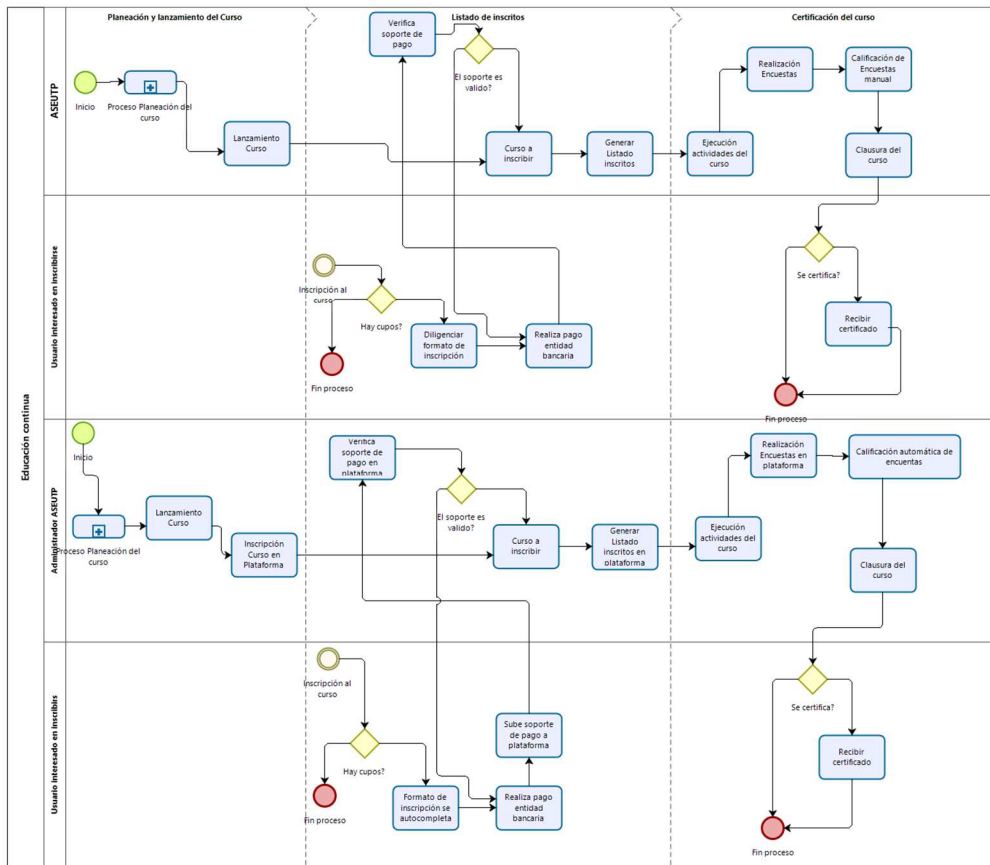
Como resultado de este desarrollo se obtuvo un nuevo procedimiento de educación continua, en el cual algunas de las actividades, como lo son la autogeneración del formulario de inscripción tomando la información ya registrada en el sistema, la autogeneración de listados de cursos, y la autogeneración de resultados de estadísticas de las encuestas realizadas a los participantes del curso se realizan de manera automática por la plataforma.

Estos cambios, permiten tanto a la ASEUTP y a los beneficiarios de los cursos, ahorrar tiempo para consolidar una inscripción y realizar el cierre del curso. Además, este sistema de información permite llevar un registro histórico de los cursos ofrecidos por la Asociación, usuarios que asistieron a estos cursos y los certificados adquiridos por cada uno de los inscriptos.

Nombre: *Diagrama de Flujo antiguo y nuevo del proceso educación continua*

Descripción:

El siguiente diagrama de flujo muestra el procedimiento llevado a cabo para la inscripción de un curso, teniendo en cuenta desde la planeación del curso hasta su clausura.



A continuación, se presenta la comparación entre los criterios establecidos al principio de este documento y los nuevos valores obtenidos de acuerdo al nuevo procedimiento.

PRESENTACIÓN DE MEJORAS			
#	ITEM	PROCEDIMIENTO	PROCEDIMIENTO
		ANTERIOR	NUEVO
1	$\frac{\text{\# Número de veces a diligenciar formato de inscripción}}{\text{\# Número de cursos}}$	$\frac{N}{N} = 1$	$\frac{1}{N} < 1$
2	$\frac{\text{\# Número de preguntas a revisar por encuesta}}{\text{\# Número de estudiantes}}$	$\frac{N}{N} = 1$	$\frac{0}{N} = 0$

Estos valores se calcularon basados en el nuevo procedimiento de educación continua de la ASEUTP, y como se evidencia en la tabla los nuevos valores son menores al valor calculado inicialmente.

Finalmente se realizó una prueba piloto de la plataforma desarrollada y la cual está disponible en el enlace “econtinua.herokuapp.com”, y comparando con los tiempos obtenidos en el diagnóstico

realizado al proceso de educación continua (numeral 1 del presente documento).

El tiempo promedio que tarda un usuario para registrarse a la plataforma o que tarda el administrador en registrar un nuevo usuario es de:

2 minutos llenando información de los formularios de inscripción.

El tiempo promedio que tarda un usuario en inscribirse a un curso es de:

1 minuto terminando de diligenciar algunos campos restantes del formulario de inscripción al curso (la mayoría del formulario se autocompleta con la información del registro inicial).

1 hora realizando el pago incluyendo desplazamiento al banco

2 minutos en el proceso de asociar el pago y la evidencia del mismo desde la plataforma.

Asumiendo que la persona fue registrada en la ASEUTP y pago allí mismo

10 minutos llenando información de los formularios de inscripción, pagando y el funcionario adjuntando la evidencia del pago.

El tiempo que se tomó organizando información de los cursos (Listas de asistencia, Verificación de pagos):

Generando lista de asistencia: generadas automáticamente.

Verificando soporte de pagos: 15 minutos para un curso de 20 usuarios.

Se realiza una prueba en la que se tiene un usuario administrador, se crea un curso nuevo para 20 usuarios y posteriormente se registrar estos usuarios y se inscriben al curso.

El proceso consta de inscribir cada usuario al curso, anexarle un pago y su respectivo soporte de pago, finalmente el administrador verifica usuario por usuario sus soportes de pago y los deja en estado activo en el curso.

Se tiene como resultado que esta prueba tarda alrededor de 40 a 45 minutos, tiempos muy pequeños y que dejan ver la optimización realizada al proceso de educación continua en la ASEUTP.

6. CONCLUSIONES

- Se encontró que se podía realizar la automatización de gran parte del proceso de educación continuada (no solo los formularios) mediante una plataforma. De esta manera todo el proceso que se realiza actualmente de forma manual, tediosa y que requiere gran espacio de almacenamiento de papel, puede reducirse y ser manejado de manera virtual, generando un impacto ambiental positivo, reducción en tiempos y esfuerzos por parte de los funcionarios de la asociación sino también de los interesados en asistir o ser partícipes de este proceso.
- Luego de realizar el análisis de las herramientas se lograron obtener características diferentes de cada una de ellas y de esta manera tener una visión general y más amplia para lograr un desarrollo más completo y que fuera eficiente para las necesidades de la ASEUTP.
- Gracias a la investigación realizada sobre algunas de las herramientas disponibles para realizar la implementación de la plataforma se realizó una elección correcta, la cual fue evidenciada a la hora del desarrollo ya que facilitó e hizo que todo este proceso de desarrollo fuera realizado de manera sencilla y eficiente, ya que Ruby on Rails cuenta con demasiadas gemas que ahorran tiempo y esfuerzo de trabajo.
- Para realizar un desarrollo de software de manera efectiva y eficiente es muy necesario tener una buena base desde el punto de vista de ingeniería (diagramas) ya que estos ayudan a conceptualizar mejor los requerimientos dentro del software y reduce la cantidad de errores que se puedan inyectar. En el desarrollo también se encontró que el seguir una planeación y realizar las tareas de forma secuencial permite trabajar más asertivamente y con resultados de mejor calidad.
- Finalmente al observar la evaluación de los procesos, se puede observar satisfactoriamente que el proceso de automatización realizado en este proyecto en algunas actividades del proceso de educación continua logro una mejoría indiscutible que resulta ser muy provechosa tanto para los funcionarios de la ASEUTP como para todos los usuarios interesados en participar del proceso de educación continua.

REFERENCIAS

- [1] Misión ASEUTP, Disponible en: <http://www.aseutp.com/nosotros/mision>
- [2] Formularios de google, Disponible en: https://www.google.com/intl/es-419_co/forms/about/
- [3] Acerca de Ruby, Disponible en : <https://www.ruby-lang.org/es/about/>
- [4] El gran libro de HTML5, CSS, JavaScript, Pagina 83, Disponible en:
https://books.google.es/books?hl=es&lr=&id=szDMIRzwzUC&oi=fnd&pg=PA1&dq=javascript&ots=0AsM_2uCVd&sig=-yFU_HbCOup4sWrJjTK7r6IVDzU#v=onepage&q=javascript&f=false
- [5] Mozilla Foundation, Disponible en: <https://developer.mozilla.org/es/docs/Web/CSS>
- [6] HTML & CSS – W3C, Disponible en:
<https://www.w3.org/standards/webdesign/htmlcss#whatcss>
- [7] Design Patterns. Elements of Reusable Object-Oriented Software - Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides - Addison Wesley (GoF- Gang of Four)
- [8] Tedeshi Nicolas, ¿Qué es un Patrón de Diseño?, Disponible en:
<https://msdn.microsoft.com/es-es/library/bb972240.aspx>
- [9] Simple Example of MVC (Model View Controller) Design Pattern for Abstraction,
Disponible en: <http://www.codeproject.com/Articles/25057/Simple-Example-of-MVC-Model-View-Controller-Design>
- [10] Javier J. Gutierrez, ¿Que es un Framework web?, Disponible en:
http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf
- [11] Zend Framework, Overview, Disponible en: <https://framework.zend.com/about>
- [12] Rails, Ruby on rails - El desarrollo web que no molesta, Disponible en:
<http://www.rubyonrails.org/es/>
- [13] OpenXava, OpenXava web Framework, Disponible en: <http://openxava.org/es>

LISTA DE ANEXOS

- Anexo 1:** Diseño-ReportePruebas
- Anexo 2:** Diagrama de Casos de uso
- Anexo 3.1:** Diagrama de secuencia – Login
- Anexo 3.2:** Diagrama de secuencia – Usuarios
- Anexo 3.3:** Diagrama de secuencia – Inscripciones
- Anexo 3.4:** Diagrama de secuencia – Cursos
- Anexo 3.5:** Diagrama de secuencia – Encuestas
- Anexo 3.6:** Diagrama de secuencia – Empresas
- Anexo 4.1:** Diagrama de Clases - Solo controladores
- Anexo 4.2:** Diagrama de Clases - Controladores + Métodos
- Anexo 4.3:** Diagrama de Clases - Solo Modelos
- Anexo 4.4:** Diagrama de Clases - Modelos + Atributos
- Anexo 5:** Diagrama de componentes
- Anexo 6:** Diagrama de despliegue
- Anexo 7:** Documento de requerimientos.
- Anexo 8:** Acta de entrega de software final.