**Implementación de aplicación web SETUL para realizar pruebas institucionales en la facultad de derecho de la Universidad Libre seccional Pereira**

Alejandro Carvajal Muñoz

Cristian Camilo Agudelo Rivera

Marzo 2017.

Universidad Libre Seccional Pereira.

Facultad de Ingenierías.

**Tabla de Contenidos**

[Descripción de la situación 1](#_Toc26280911)

[Estado del arte 1](#_Toc26280912)

[Descripción del problema 2](#_Toc26280913)

[Justificación 2](#_Toc26280914)

[Objetivos 3](#_Toc26280915)

[Objetivo general 3](#_Toc26280916)

[Objetivos específicos 3](#_Toc26280917)

[Delimitación del proyecto 4](#_Toc26280918)

[Marco referencial 5](#_Toc26280919)

[Marco teórico 5](#_Toc26280920)

[Ingeniería de software 5](#_Toc26280921)

[Elicitación de requerimientos 7](#_Toc26280922)

[Programación orientada a objetos 8](#_Toc26280923)

[MVC 10](#_Toc26280924)

[Laravel 11](#_Toc26280925)

[Marco conceptual 12](#_Toc26280926)

[Marco metodológico 14](#_Toc26280927)

[Tipo de investigación 14](#_Toc26280928)

[Método de investigación 14](#_Toc26280929)

[Marco legal y normativo 15](#_Toc26280930)

[Plan de Desarrollo de Software 16](#_Toc26280931)

[Alcance 16](#_Toc26280932)

[Vista General del Proyecto 16](#_Toc26280933)

[Propósito, Alcance y Objetivos 16](#_Toc26280934)

[Suposiciones y Restricciones 19](#_Toc26280935)

[Entregables del proyecto 19](#_Toc26280936)

[Evolución del Plan de Desarrollo del Software 22](#_Toc26280937)

[Organización del Proyecto 22](#_Toc26280938)

[Participantes en el Proyecto 22](#_Toc26280939)

[Interfaces Externas 23](#_Toc26280940)

[Roles y Responsabilidades 23](#_Toc26280941)

[Gestión del Proceso 24](#_Toc26280942)

[Plan de las Fases 24](#_Toc26280943)

[Calendario del Proyecto 25](#_Toc26280944)

[Seguimiento y Control del Proyecto 28](#_Toc26280945)

[Gestión de Requisitos 28](#_Toc26280946)

[Control de Plazos 28](#_Toc26280947)

[Control de Calidad 28](#_Toc26280948)

[Gestión de Riesgos 28](#_Toc26280949)

[Gestión de Configuración 28](#_Toc26280950)

[Modelado del Negocio 29](#_Toc26280951)

[Preparatorios derechos 29](#_Toc26280952)

[Modelo de casos de uso del negocio 30](#_Toc26280953)

[Modelo de dominio 31](#_Toc26280954)

[Modelo de Objetos 31](#_Toc26280955)

[Visión 33](#_Toc26280956)

[Alcance 33](#_Toc26280957)

[Oportunidad de Negocio 33](#_Toc26280958)

[Sentencia que define el problema 34](#_Toc26280959)

[Sentencia que define la posición del Producto 35](#_Toc26280960)

[Resumen de Stakeholders 36](#_Toc26280961)

[Resumen de Usuarios 36](#_Toc26280962)

[Entorno de usuario 37](#_Toc26280963)

[Perfil de los Stakeholders 37](#_Toc26280964)

[Perfiles de Usuario 38](#_Toc26280965)

[Jefe de Sistemas 38](#_Toc26280966)

[Estudiante 39](#_Toc26280967)

[Entidad Externa 39](#_Toc26280968)

[Perspectiva del producto 39](#_Toc26280969)

[Resumen de características 40](#_Toc26280970)

[Descripción Global del Producto 41](#_Toc26280971)

[Especificaciones de Casos de Uso 41](#_Toc26280972)

[Especificaciones Adicionales 42](#_Toc26280973)

[Prototipos de Interfaces de Usuario 42](#_Toc26280974)

[Interfaces de administrador 42](#_Toc26280975)

[Asignaturas 42](#_Toc26280976)

[Configuración 44](#_Toc26280977)

[Login 46](#_Toc26280978)

[Administración 47](#_Toc26280979)

[Participantes 48](#_Toc26280980)

[Preguntas 51](#_Toc26280981)

[Tests 55](#_Toc26280982)

[Usuarios 57](#_Toc26280983)

[Modelo de Análisis y Diseño 59](#_Toc26280984)

[Modelo de Datos 61](#_Toc26280985)

[Modelo de Implementación 63](#_Toc26280986)

[Modelo de Despliegue 63](#_Toc26280987)

[Casos de Prueba 64](#_Toc26280988)

[Manual de Instalación 64](#_Toc26280989)

[Material de Apoyo al Usuario Final 64](#_Toc26280990)

[Producto 64](#_Toc26280991)

[Bibliografía 66](#_Toc26280992)

**Lista de tablas**

[Tabla 1 Matriz metodológica. 16](#_Toc31387570)

[Tabla 2: Puesto y responsabilidad 27](#_Toc31387571)

[Tabla 3: Fase con su cantidad de iteraciones y duración 27](#_Toc31387572)

[Tabla 4: Fase descripción 28](#_Toc31387573)

[Tabla 5: Tareas 28](#_Toc31387574)

[Tabla 6 Stackeholders 39](#_Toc31387575)

[Tabla 7 Resumen de usuarios 39](#_Toc31387576)

[Tabla 8 Resumen de características 43](#_Toc31387577)

**Lista de figuras**

[Ilustración 1 Modelo subsistemas involucrados 30](#_Toc26280870)

[Ilustración 2 Modelo de casos de uso del negocio 30](#_Toc26280871)

[Ilustración 3 Modelo de dominio 31](#_Toc26280872)

[Ilustración 4 Modelo de objetos de preparación event 31](#_Toc26280873)

[Ilustración 5 Modelo de objetos de creación de preguntas1 32](#_Toc26280874)

[Ilustración 6 Modelo de objetos de creación de preguntas 2 32](#_Toc26280875)

[Ilustración 7 Modelo de objetos configuración sistema 32](#_Toc26280876)

[Ilustración 8 Listar asignaturas 43](#_Toc26280877)

[Ilustración 9 Creación de asignatura 43](#_Toc26280878)

[Ilustración 10 Edición de asignatura 44](#_Toc26280879)

[Ilustración 11 Listar configuración 45](#_Toc26280880)

[Ilustración 12 Creación de configuración 46](#_Toc26280881)

[Ilustración 13 Edición de configuración 46](#_Toc26280882)

[Ilustración 14 Login 47](#_Toc26280883)

[Ilustración 15 Inicio de administración 48](#_Toc26280884)

[Ilustración 16 Listar participantes 49](#_Toc26280885)

[Ilustración 17 Creación de participante 50](#_Toc26280886)

[Ilustración 18 Edición de participante 50](#_Toc26280887)

[Ilustración 19 Importar participantes 51](#_Toc26280888)

[Ilustración 20 Listar preguntas 52](#_Toc26280889)

[Ilustración 21 Creción de pregunta 53](#_Toc26280890)

[Ilustración 22 Edición de pregunta 54](#_Toc26280891)

[Ilustración 23 Importar preguntas 55](#_Toc26280892)

[Ilustración 24 Listar tests 56](#_Toc26280893)

[Ilustración 25 Creación de test 56](#_Toc26280894)

[Ilustración 26 Edicón de test 57](#_Toc26280895)

[Ilustración 27 Lista de usuarios 58](#_Toc26280896)

[Ilustración 28 Creación de usuario 59](#_Toc26280897)

[Ilustración 29 Edición de usuario 59](#_Toc26280898)

[Ilustración 30 Diagrama de clases Controller 60](#_Toc26280899)

[Ilustración 31 Diagrama de clases Model 61](#_Toc26280900)

[Ilustración 32 Modelo Entidad Relación 62](#_Toc26280901)

[Ilustración 33 Modelo de Implementación 63](#_Toc26280902)

[Ilustración 34 Modelo de despliegue 64](#_Toc26280903)

# Descripción de la situación

## Estado del arte

Trivial Pursuit fue creado en 1979 por Chris Haney y Scott Abbott, quienes trabajaban en el periódico The Gazette. Luego de inventar el juego iniciaron la búsqueda de capital que les ayudara a impulsar el mismo. Como no tenían certeza de cómo lograrlo, tomaron la decisión de asistir como periodistas a la feria internacional del juguete en Montreal donde lograron recaudar por medio de inversores la cantidad necesaria para dar visibilidad a su nuevo producto.

En 1981 Trivial Pursuit salió al mercado y pese a que con la inversión inicial su visibilidad se incrementó, su salida fue por solo 15 dólares, cuando su producción costaba 75 dólares. Posteriormente y gracias al impacto positivo que genero el “voz a voz” en 1984 Se alcanzó la venta de 20 millones de unidades por valor de 500 millones de dólares. Desde entonces los derechos del juego han pasado por diferentes dueños y hoy en día es Hasbro el poseedor de los mismos. (deafinder, 2005)

Trivial Pursuit es un juego de preguntas y respuestas, el cual abarca diferentes temáticas como ciencia, historia, geografía, arte y deportes. Es considerado un éxito mundial con ventas en 26 países y en varios idiomas.

## Descripción del problema

En la Universidad Libre seccional Pereira, se encuentra la necesidad de emplear herramientas tecnológicas que permitan la evaluación de los estudiantes de una forma más interactiva.

Actualmente se maneja el software Moodle, el cual es un LMS que permite realizar la entrega de trabajos, realizar evaluaciones. Pero no posee opciones para interactuar en clase, para realizar equipos de trabajo y resolver las preguntas de forma interactiva.

# Justificación

En la era digital de la actualidad, debido a la constante evolución tecnológica, al desarrollo de software y las diferentes formas de comunicación, se hace necesario el uso de tecnologías de información en la educación y aprendizaje, por lo tanto, es necesario romper esquemas y temores con respecto al uso de las TIC.

Teniendo presente que la apropiación de las TIC en los procesos de aprendizaje y enseñanza, permiten nuevos escenarios de innovación educativa y adquisición de nuevas competencias. Los beneficios de adaptarse a las TIC es que se contribuye a una mejor utilización de la tecnología y recursos (Iriarte, 2013).

La población estudiantil de la Universidad Libre Seccional Pereira se ve en la necesidad de tener herramientas tecnológicas que permitan innovar en la enseñanza y retención de conocimiento a través de las diversas asignaturas que cursan los estudiantes. Por lo tanto, se convierte en una oportunidad el desarrollo de la aplicación SETUL, el cual brinda preguntas, respuestas y ayudas sobre temas específicos a la población estudiantil.

La aplicación SETUL a su vez, es una herramienta para los docentes que permite diagnosticar e identificar la situación actual de un aula de clase y la evolución en sí misma, como también confrontar en un histórico las temáticas que han sido más impactantes o que han generado más dificultad en la adquisición de conocimiento por partes de sus asistentes. Lo anterior gracias al contenido dinámico que permite la utilización del aplicativo. Como valor agregado presenta las condiciones para fomentar el intercambio de conocimiento, puesto que las opciones de ayuda permiten que los compañeros realicen un proceso de retroalimentación del conocimiento.

# Objetivos

## Objetivo general

Implementar aplicación web SETUL para la realización de pruebas institucionales en la Universidad Libre seccional Pereira.

## Objetivos específicos

* Realizar ingeniería de requerimientos con metodología SCRUM
* Diseñar aplicación web SETUL
* Analizar infraestructura necesaria para el desarrollo de la aplicación SETUL
* Desarrollar aplicación web SETUL
* Realizar pruebas de software con metodología AGILE

# Delimitación del proyecto

El proyecto será ejecutado durante 20 semanas, el tiempo se dividirá en diferentes etapas compuestas de una serie de actividades en las cuales se establece como etapa inicial la investigación para la adquisición y fortalecimiento de las bases del conocimiento. Posteriormente se realizarán las etapas de análisis, diseño, desarrollo e implementación del aplicativo web SETUL para seguir con los lineamientos del ciclo de vida del software y finalmente hacer entrega del sistema el cual estará disponible en la Universidad libre seccional Pereira.

# Marco referencial

## Marco teórico

Ingeniería de software

La ingeniería de software es un área de las ciencias de la computación que tiene como objetivo definir métodos y técnicas para desarrollar software con calidad. En la Ingeniería del Software existen diversos modelos para desarrollar software según la necesidad y características del proyecto a realizar, pero tienen en común que al finalizar el producto de software debe ser eficiente y de confianza (EcuRed, Ingeniería de software, 2013).

La Ingeniería de software toma importancia debido a la alta tasa de cambios en los requerimientos iniciales y el entorno en que se desarrolla, las necesidades que la ingeniería de software busca mitigar son (tutorialspoint.com, s.f.):

* Escalabilidad: La tendencia de un software es evolucionar y crecer al pasar el tiempo, por tal razón su diseño y construcción debe soportar cambios, a su vez estos cambios deben poderse implementar sin complejidad.
* Costos: Debe realizar seguimiento continuo a los costos, evitando los reprocesos en el desarrollo y el reproceso en los cambios de requerimientos.
* Naturaleza dinámica: Se debe asumir desde un comienzo que el software a desarrollar es cambiante y aumentará sus funcionalidades con el tiempo.
* Gestión de calidad: Crear software de calidad, confiabilidad.

La ingeniería de Software posee 7 etapas, las cuales deben cumplirse para cualquier tipo de desarrollo de software, son:

* Análisis: Es el proceso que se realiza para darle claridad al problema que se pretende solucionar, se definen los componentes que tendrá el software.
* Diseño: Es el proceso donde se crean los modelos, características y componentes que se determinaron en la etapa de análisis.
* Desarrollo: Es el proceso donde se realiza la construcción del software a partir del diseño realizado.
* Pruebas: Consiste en revisar el desarrollo con respecto al análisis y diseño, debe coincidir en características, validaciones y en datos.
* Implementación: Es el proceso en el que se realiza la instalación y puesta en marcha de la aplicación realizada.
* Mantenimiento: Es el proceso en el cual se realizan mejoras y/o corrección de errores a la aplicación desarrollada.
* Final de vida (EOL): Es la etapa donde el software deja de funcionar, se deja de comercializar y dar soporte.

Elicitación de requerimientos

La elicitación de requerimientos es la actividad fundamental en el desarrollo de software, debido a que ayuda a entender con claridad las necesidades del cliente y es el momento en el que se identifica el objetivo del software y que parámetros debe cumplir al momento de la entrega final del software (EcuRed, Metodología para la elicitación de requisitos, 2012).

La elicitación de requerimientos debe estar muy bien documentada, debido a que en base a la claridad en la que se definan los requisitos se inicia con el análisis y diseño de los modelos UML para el desarrollo y comprensión del software.

Existen diferentes técnicas para realizar la elicitación de requerimientos las cuales se usan de acuerdo con la necesidad del proyecto que se realice, algunas de estas técnicas son:

* Entrevistas

Esta técnica es la más usada, debido a que es la forma de comunicación natural entre las personas, las entrevistas deben tener objetivos y tareas claras como definir el dominio del problema, identificar el entorno en el que se desarrollará e identificar las necesidades y problemáticas que debe solucionar el software (EcuRed, Metodología para la elicitación de requisitos, 2012).

* JAD (Join Application Development)

Esta técnica es una variación de la entrevista, se pretende guiar a los clientes y hacerlos participantes del desarrollo. El objetivo de esta técnica es usar ayudas visuales para definir los objetivos y requisitos que debe cumplir el sistema, identificar el alcance del sistema y realizar documentación de todo lo que se trate en las reuniones (EcuRed, Metodología para la elicitación de requisitos, 2012).

* Brainstorm

El objetivo del brainstorm es recopilar información sobre los requisitos del sistema, esta técnica es fundamental cuando los requerimientos del sistema no están claros o son difusos, debido a que el objetivo es tener diferentes puntos de vista de los problemas y tener una amplia formulación de soluciones (EcuRed, Metodología para la elicitación de requisitos, 2012).

Programación orientada a objetos

Es un paradigma del desarrollo de software, el cual usa objetos e interacciones con el fin de diseñar aplicaciones (EcuRed, Programación orientada a objetos, 2015). La programación orientada a objetos intenta simular el mundo real a través del uso de características y funciones, este paradigma utiliza 4 herramientas fundamentales que lo caracterizan con respecto a otros paradigmas de desarrollo de software los cuales son:

* Herencia:

Este paradigma permite relacionar las clases a nivel jerárquico con el fin de que los objetos hereden comportamiento y propiedades en común, facilitando la mantenibilidad y escalabilidad del código ya que permite la centralización de procesos generales (EcuRed, Programación orientada a objetos, 2015).

* Abstracción:

Esta propiedad, permite asignarle comportamientos y acciones a un objeto, mediante esto se modelan las clases con el fin de que se asemeje con el mundo real. Esta propiedad es clave en el proceso de análisis y diseño de clases (EcuRed, Programación orientada a objetos, 2015).

* Encapsulamiento:

Consiste en organizar los datos y métodos con el objetivo de evitar acceso por medios no especificados, esto brinda seguridad e integridad a los datos de los objetos, permitiendo al final una mayor integración a los procesos modulares (Lara, 2007).

* Polimorfismo:

Consiste en tener comportamientos diferentes asociados a objetos distintos, pero con el mismo nombre, es decir, en una clase pueden existir métodos con el mismo nombre, pero recibe parámetros diferentes y realizan operaciones diferentes (EcuRed, Programación orientada a objetos, 2015).

MVC

Es un patrón de diseño de software el cual propone separar responsabilidades del sistema, sus siglas significan Modelo, Vista y Controlador, estas 3 capas son los pilares de este diseño de software. (Alvarez, 2014)

Este patrón de diseño es utilizado en su mayoría por las aplicaciones que tienen interfaz gráfica de usuario, al dividir las responsabilidades, permite que el software pueda llegar a ser muy robusto, escalable, tener un mantenimiento de una forma más ágil , efectiva y facilita su documentación.

El objetivo del MVC a nivel de desarrollo es separar la lógica de la aplicación del entorno gráfico, a pesar de que ambos son realizados a nivel de código, es importante separarlos con el fin de tener mayor escalabilidad, robustez y reutilización en el código que se genera.

Modelos:

Es la capa encargada de trabajar con los datos, posee diferentes formas de consultar, modificar y agregar información a dichos datos. Usualmente estos datos son obtenidos a través de una base de datos (Alvarez, 2014).

Vistas

Esta capa tiene la tarea de brindar visualización al usuario, toda interacción de entrada y salida del sistema se realiza a través del entorno gráfico, estas vistas requieren de los modelos para poder mostrarle respuestas al usuario (Alvarez, 2014).

Controladores

Es la capa encargada de realizar los procedimientos y acciones que se solicitan en el software, a través de las vistas. Un controlador interactúa obteniendo resultados y solicita cambios en el modelo para enviar una respuesta a las vistas (Alvarez, 2014).

Laravel

“[Laravel](http://laravel.com/) es uno de los frameworks de código abierto más fáciles de asimilar para PHP. Es simple, muy potente y tiene una interfaz elegante y divertida de usar” (García, 2015), este framework se caracteriza por ser fácil de aprender, ser robusto y tener una comunidad activa a nivel mundial. Está basado en el framework Symfony, del cual usa sus dependencias, pero se enfoca en que el código sea sencillo.

Las características principales de laravel son:

* Blade: Es un motor de plantillas con el objetivo de crear vista, permite extender, crear secciones, acceder a variables e interactuar con código PHP.
* Eloquent: Es un ORM con el objetivo de facilitar las interacciones con la base de datos, ayuda a optimizar tiempo cuando se realiza software que servirá para múltiples motores de bases de datos y ayuda evitando la inyección SQL.
* Routing: Es un sistema con el fin de gestionar y organizar las rutas del proyecto, con esto se cubre una capaz extra de seguridad debido a que se obliga a definir a que controlador o vista debe dirigirse cuando se accede a alguna ruta.
* Middleware: Son controladores que se ejecutan en las peticiones del servidor, con el fin de validar información antes de que sea procesada.

Hypertext Preprocessor (PHP)

Es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. PHP está enfocado principalmente a la programación de scripts del lado del servidor, por lo que se puede hacer cualquier cosa que pueda hacer otro programa CGI, como recopilar datos de formularios, generar páginas con contenidos dinámicos, o enviar y recibir cookies.(PHP, s.f.).

SCRUM

Es una metodología para realizar procesos de forma rápida, consiste en realizar entregas parciales y funcionales. Es una buena práctica para trabajar colaborativamente y en equipo (Francia, 2017).

**Object relational mapping (ORM)**

Son herramientas que permiten convertir tablas de la base de datos en entidades con el fin de facilitar el trabajo en las conexiones a bases de datos y brindar una capa de seguridad contra ataques informáticos (Educació, 2018).

## Marco conceptual

**Elicitación de requerimientos**

Se utiliza la técnica de la entrevista para conocer el estado actual de la necesidad y los resultados esperados que son necesarios para la optimización del proceso. De esta técnica se obtienen los requerimientos posteriormente clasificados en funcionales y no funcionales para la ejecución del proyecto.

**Hypertext Preprocessor (PHP)**

Se utiliza PHP como lenguaje de programación debido a que su entorno de ejecución permite que el proyecto cumpla con su alcance de poder ser accedido desde diferentes ubicaciones, personas y dispositivos. Gracias a su integración con HTML, CSS y librerías también permite que el entorno gráfico sea muy agradable lo cual colabora para el diseño intuitivo de cada una de las pantallas que lo componen.

**Laravel**

Se realiza la implementación en el proyecto debido a los beneficios frente a temas de seguridad, agilidad de procesamiento, comunidad, escalamiento, reutilización y estilización de código que permiten que las actividades a realizar se disminuyan en tiempo además de que se garantiza se utilizan buenas prácticas de programación.

**MVC**

Junto con la implementación de laravel este patrón de diseño nos permite realizar la división de la programación por capas. Pensando en la escalabilidad o cambios que puede llegar a tener el proyecto el patrón MVC nos da el beneficio de alterar elementos visuales sin afectar la lógica de ejecución como también permite cambiar lógica y no correr con el inconveniente de ir a modificar la parte visual.

**SCRUM**

Se aplican técnicas de esta metodología para llevar el control de las reuniones y requerimientos que se establecían de cada una de estas. Los tiempos de esta metodología ágil permite la entrega de resultados a corto plazo al igual que la reducción en el riesgo de realizar entregas que se desalinean del requisito original.

**Object relational mapping (ORM)**

Con la implementación del framework laravel se utilizó su ORM Eloquent para la generación de entidades que permiten la interacción con la base de datos de manera más limpia. Adicional pensando en la escalabilidad e integración que puede tener el software el uso de esta tecnología permite la instalación de este en diferentes motores mitigando el inconveniente de compatibilidad.

# Marco metodológico

## Tipo de investigación

Debido a la necesidad de encontrar variables sociales que influyen en el desarrollo de la aplicación y la necesidad examinar los datos de manera científica, se determina que la investigación es clasificada como exploratoria y descriptiva, por lo tanto, es una investigación mixta.

La investigación exploratoria ayuda a tener un primer acercamiento del problema, por lo tanto, es necesario indagar sobre el uso de herramientas tecnológicas en las clases que se dictan en la Universidad Libre seccional Pereira, luego de tener un panorama de las TIC’s se emplea la investigación descriptiva, la cual ayuda a plantear los hechos relevantes y situaciones concretas que debe solucionar el aplicativo SETUL.

## Método de investigación

La presente investigación se desarrolla bajo del ciclo de vida del software ya que al estar dirigido principalmente a este tipo de proyectos permite establecer de manera ordenada los pasos para lograr un adecuado levantamiento de necesidades, puntos de interacción con los afectados, diseños arquitectónicos del software, integraciones parciales para percepciones iniciales, pruebas con el entorno y mejoras continuas. Al final de la ejecución del método se obtendrán resultados muy exactos debido a su detallada ejecución en cada una de sus fases principales (Análisis, Diseño, Ejecución, Mejoras).

Tabla 1 Matriz metodológica.

|  |  |
| --- | --- |
| Objetivos | Actividades |
| Realizar ingeniería de requerimientos con metodología SCRUM | 1. Generar cronograma de actividades.  2. Realizar entrevista para conocer el estado actual del procedimiento que se lleva a cabo.  3. Realizar 2 entrevistas a los 2 actores principales que intervienen en la ejecución del proyecto.  4. Realizar documentación de análisis de la información obtenida de las entrevistas.  5. Generar bocetos de las interfaces gráficas.  6. definición de requisitos funcionales y no funcionales.  7. Realizar etapa de análisis de la ingeniería de software. |
| Diseñar aplicación web SETUL | 1. Establecer casos de uso de acuerdo con las necesidades del proyecto.  2. Especificar herramientas de trabajo (Lenguaje de programación, framework, IDE de desarrollo)  3. Se establecen los recursos multimedia a utilizar.  4. Establecer la interface gráfica de la aplicación. |
| Analizar infraestructura necesaria para el desarrollo de la aplicación SETUL | 1. Definición de entorno de ejecución de la aplicación.  2. Definición de librerías externas para el uso en la aplicación.  3. Generación de diagramas de componentes y de implementación.  4. definición de requisitos mínimos para el funcionamiento de la aplicación. |
| Desarrollar aplicación web SETUL | 1. Codificación de la aplicación  2. Documentación del código fuente de la aplicación.  3. Creación de prototipos |
| Realizar pruebas de software con metodología AGILE | 1.Realizar pruebas de los prototipos y entregable final.  2. Realizar y probar casos de prueba.  3. Realizar pruebas de funcionamiento en un ambiente de uso real. |

La anterior tabla permite alcanzar el objetivo general Implementar aplicación web SETUL para la realización de pruebas institucionales en la Universidad Libre seccional Pereira.

# Marco legal y normativo

Ley 1273 de 2009 – “Se modifica el Código Penal, se crea un nuevo bien jurídico tutelado - denominado “de la protección de la información y de los datos"- y se preservan integralmente los sistemas que utilicen las tecnologías de la información y las comunicaciones, entre otras disposiciones” (Colombia, Ley 1273 del 2009, 2009).

Ley N° 1403 de 2010 (19 de julio) – “Por la cual se adiciona la Ley N° 23 de 1982 sobre derechos de autor, se establece una remuneración por comunicación pública a los artistas, intérpretes o ejecutantes de obras y grabaciones Audiovisuales o Ley Fanny Mikey” (Colombia, Ley 1403 del 2010, 2010).

Decreto 1360 de 1989 “Por el cual se reglamenta la inscripción de soporte lógico (software) en el Registro Nacional del Derecho de Autor” (Colombia, Decreto 1360 de 1989, 1989).

Ley 1581 de 2012 – “Por el cual se dictan disposiciones generales para la protección de datos personales” (Colombia, Ley 1581 de 2012, 2012).

Decisión andina “351 de 1993 régimen común sobre derecho de autor y derechos conexos” (DNDA, 1993).

# Plan de Desarrollo de Software

## Alcance

El actual plan de desarrollo del software describe de manera global el propósito y tareas para el correcto desarrollo para la preparación y evaluación del sistema. Dentro del documento se indica las características que debe cumplir el sistema los cuales generan las bases para establecer las iteraciones que presentara el proyecto. En el análisis para la primera iteración se proyecta para la toma de requerimientos que se le realizara a los directivos y docentes de la seccional de derecho, para posteriormente llevar a cabo el desarrollo de los modelos basados en UML e iniciar con el diseño que llevara el software aplicando el patrón de arquitectura modelo vista controlador de lo cual se genera un primer prototipo y pantallas guía de los cuales se espera surgen nuevos requerimientos y ajustes los cuales se documentaran para posteriormente dar inicio a la siguiente iteración.

La segunda iteración al igual que la anterior será documentada lo cual generará la versión 2 del actual documento. De igualmente se espera se presenten cambios sobre el modelamiento debido a la proyección que maneja el personal al cual está destinado el software, pero se establece que es lo adecuado para el desarrollo del proyecto

## Vista General del Proyecto

## Propósito, Alcance y Objetivos

Implementación de aplicación web SETUL para realizar pruebas institucionales en la facultad de derecho de la Universidad Libre seccional Pereira

La universidad Libre seccional Pereira realiza anualmente los denominados preparatorios para el área de derecho los cuales consisten en una serie de preguntas que se les realizan a los estudiantes de esta facultad para incentivarlos y prepararlos para sus exámenes finales. Teniendo en cuenta el avance en tecnología, requisitos que establecen los nuevos modelos de educación, la necesidad de generar una dinámica mas llamativa e integrar en este proceso otras universidades de la región. La universidad libre opta por actualizar su metodología por medio de un software tipo trivia el cual se encargara de llevar el listado de participantes a las pruebas, permitirá la gestión de las preguntas y respuestas que se presentan durante el evento, permitirá la generación de pruebas de manera grupal e individual por medio de la asignación de test los cuales contienen una serie de preguntas aleatorias definidas por el sistemas basado en una característica de las preguntas denominada como “peso”, permitirá la creación de reportes y cartas que se utilizan para el control y posible premiación de los asistentes. Al final de esta implementación lo que se busca además de lo anteriormente mencionado es disminuir los tiempos de organización del evento y agilizar el desarrollo del mismo debido a que actualmente el proceso de asignación de preguntas se realiza manualmente en una herramienta ofimática que dificulta que el evento se desarrolle con fluidez

El proyecto presenta en su propuesta las tareas y operaciones que permitirán el desarrollo basado en el ciclo de vida de la ingeniería de software lo cual permitirá un mayor control y estandarización al momento de revisar y ejecutar los objetivos los cuales se pueden clasificar en:

1. Administración de usuarios.
   1. Crear.
   2. Editar.
   3. Consultar
   4. Eliminar.
2. Administración de asignaturas.
   1. Crear.
   2. Editar.
   3. Consultar
   4. Eliminar.
3. Administración de participantes.
   1. Crear.
   2. Editar.
   3. Consultar
   4. Eliminar.
   5. Cargar masivamente por medio de una hoja de cálculo.
4. Administración de preguntas y respuestas.
   1. Crear.
   2. Editar.
   3. Consultar
   4. Eliminar.
   5. Cargar masivamente por medio de una hoja de cálculo.
5. Crear test (listado de preguntas)
   1. Crear.
   2. Editar.
   3. Consultar
   4. Eliminar.
6. Administrar configuraciones
   1. Crear.
   2. Editar.
   3. Consultar
   4. Eliminar.
7. Asignar preguntas aleatoriamente a los test
8. Administración de perfiles para el control del sistema.
9. Administrar el fondo de las pantallas en el modo “juego”
10. Administrar temporizador para llevar control del tiempo entre preguntas en el modo “juego”

## Suposiciones y Restricciones

Las suposiciones y restricciones se obtienen a través de entrevistas con los solicitantes del sistema, las cuales son:

1. Debe funcionar en entorno web
2. Debe ajustarse para funcionar en equipos móviles (Responsive).
3. Solo debe permitir un administrador total del sistema.
4. La generación de las preguntas para los test debe garantizar que no se presenten preguntas repetidas en el mismo.
5. Las preguntas que se presentan en los test no se deben repetir como mínimo en los 2 siguientes test de la misma categoría.
6. Las pantallas del modo juego deben generar sonidos que permitan un ambiente dinámico para los participantes y asistentes presentes en el momento de los eventos.
7. Se deben poder cargar los escudos o logos de las instituciones participantes de los eventos.
8. El sistema debe garantizar que no se puede acceder a sus pantallas de administración sin estar previamente logueado.

## Entregables del proyecto

De acuerdo con la filosofía de RUP (y de todo proceso iterativo e incremental), a pesar de que los artefactos se modifican a lo largo del tiempo hasta el proceso de desarrollo se encuentre terminado, solo se entregarán los artefactos finales, la cual equivale a la versión definitiva y estable del sistema, por lo tanto, se entrega:

1. Plan de Desarrollo del Software

Es el documento actual.

1. Modelo de Casos de Uso del Negocio

Modela las funciones del negocio tomando en cuenta la visión de los actores externos. Este modelo se representa con diagrama de casos de uso

1. Modelo de Objetos del Negocio

Modela cada caso de uso del negocio, se establecen actores internos y flujos de trabajo asociados al uso del negocio, para este se realizan diagramas de secuencia, colaboración, clases y de actividad en los cuales se evidencie el flujo de trabajo (Orea, 2009).

1. Glosario (MARCO CONCEPTUAL)

Define los términos relevantes usados en el presente proyecto

1. Modelo de Casos de Uso

Representa los actores y sus funciones en el sistema, se realiza a través de diagrama de casos de uso.

1. Visión

Se especifican las características y necesidades del producto, son requisitos del sistema

1. Especificaciones de Casos de Uso

Se realiza descripción detallada de los casos que lo requieran, este documento debe incluir: precondiciones, post-condiciones, flujo de eventos, requisitos funcionales y no-funcionales asociados. Cuando los eventos tienen un flujo complejo, se debe realizar diagrama de actividad.

1. Especificaciones Adicionales

Se registran los requerimientos y/o requisitos que no se han incluido en los casos de uso y se definen como requerimientos no funcionales.

1. Prototipos de Interfaces de Usuario

Son los prototipos de interfaces que tendrá el sistema, deben coincidir con los requisitos presentados por los usuarios.

1. Modelo de Análisis y Diseño

Este modelo es donde se traduce los requerimientos del sistema a un modelo abstracto para el desarrollo de la aplicación.

1. Modelo de Datos

El sistema es será soportado por un modelo relacional de datos, se realiza a través de diagramas de clases.

1. Modelo de Implementación

Este modelo es el conjunto de componentes necesarios para realizar el despliegue de la aplicación.

1. Modelo de Despliegue

El modelo muestra la manera en que se realizará el despliegue de los componentes (Orea, 2009).

1. Casos de Prueba

Es un documento que indica las entradas, condiciones de ejecución y los resultados esperados.

1. Manual de Instalación

El documento tiene instrucciones para realizar la instalación del producto.

1. Material de Apoyo al Usuario Final

Es un conjunto de documentos que incluyen las guías de usuario, guías de operación y guías de mantenimiento.

1. Producto

Se debe almacenar el producto en un CD con el manual de instalación con el fin de facilitar la instalación.

## Evolución del Plan de Desarrollo del Software

Se realizarán revisiones quincenales y se realizarán los cambios antes del comienzo de una nueva iteración, basados en El Plan de Desarrollo del Software.

## Organización del Proyecto

## Participantes en el Proyecto

Por parte de la universidad libre los encargados de proporcionar los requisitos y realizar la validación del funcionamiento del sistema serán:

**Solicitante**: Las personas definidas bajo este perfil serán los docentes del área de derecho encargados de dirigir o apoyar la preparación del evento. Adicionalmente también incluirá a la persona con cargo jefe de área derecho laboral con la cual se tendrá la mayor parte de la captura de requerimientos.

El personal asignado para el desarrollo del proyecto para las fases de análisis, diseño, desarrollo, pruebas e implementación durante las dos iteraciones que se tienen planeadas se conformaran por los siguientes actores:

**Líder de Proyecto:** Persona con la capacidad de documentar las solicitudes y requerimientos planteados por los solicitantes de manera que sean lo suficientemente claros y viables para el desarrollo del proyecto.

**Analista de Sistemas:** Persona con conocimiento en el diagramado de casos de uso y con la facilidad de interpretar los requerimientos iniciales y establecer unas bases sobre las que se construirá el sistema.

**Desarrolladores:** Persona con experiencia en desarrollo de software sobre el cual se realizará el proyecto.

## Interfaces Externas

Los participantes del proyecto serán definidos por la Universidad Libre. Se debe definir quienes proporcionarán requerimientos del sistema, quienes serán los encargados de evaluar y probar el funcionamiento de los prototipos entregados.

El equipo de desarrollo interactuará activamente con los docentes de derecho y la jefatura para especificación y validación de los artefactos generados.

## Roles y Responsabilidades

Según los roles que se desempeñan en RUP, estas son las responsabilidades principales de los roles en el equipo desarrollo durante las fases del proyecto.

Tabla 2: Puesto y responsabilidad

|  |  |
| --- | --- |
| Puesto | Responsabilidad |
| Líder de Proyecto | Asigna y gestiona los recursos y prioridades, canal entre el cliente y usuarios, enfoca el equipo de trabajo con los objetivos del proyecto, gestiona los riegos, planificación y realiza controles del proyecto de acuerdo al cronograma de trabajo |
| Analista del Sistema | Recibe, detalla y valida requerimientos, interactúa con los usuarios a través de entrevistas. Realiza los modelos de análisis y diseño. Elabora las pruebas funcionales |
| Programador | Encargado de la construcción de prototipos, creación de pruebas funcionales, creación de los modelos de datos. |

## Gestión del Proceso

## Plan de las Fases

El desarrollo del sistema se realiza en base a fases con 2 iteraciones en cada una de ellas. La tabla a continuación se evidencia la fase, el número de iteraciones y duración de las mismas.

Tabla 3: Fase con su cantidad de iteraciones y duración

|  |  |  |
| --- | --- | --- |
| Fase | Nro. Iteraciones | Duración |
| Fase de Inicio | 2 | 4 semanas |
| Fase de elaboración | 2 | 3 semanas |
| Fase de construcción | 2 | 7 semanas |
| Fase de transición | 1 | 1 semana |

Los hitos que marcan el final de cada fase se describen en la siguiente tabla.

Tabla 4: Fase descripción

|  |  |
| --- | --- |
| Descripción | Hito |
| Fase de inicio | En esta fase desarrolla los requisitos del producto desde la visión del usuario final, será establecido en el artefacto Visión. Los principales casos de uso serán identificados. La aceptación del artefacto Visión y el Plan de Desarrollo marcan el final de esta fase. (Orea, 2009) |
| Fase de Elaboración | En esta fase se analizan los requerimientos y se desarrollan prototipos funcionales, con el fin de que el usuario final apruebe los requerimientos y se realicen ajustes a los mismos. (Orea, 2009) |
| Fase de Construcción | Durante la fase de construcción se realiza una versión beta, basado en los requerimientos y validaciones. A esta versión beta se le aplican todas las pruebas y por último se válida con el usuario final. (Orea, 2009) |
| Fase de Transición | En esta fase se prepara la versión final para la implantación. Se entrega todos los manuales técnicos y de usuario. (Orea, 2009) |

## Calendario del Proyecto

Se establece el siguiente calendario donde se marca la descripción de la tarea, la duración, jerarquía y fecha en la cual se realiza.

Tabla 5: Tareas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID TAREA** | **DESCRIPCIÓN** | **DURACIÓN** | **PREDECESOR** | **FECHA** |
| 1 | 1.1; Definición de problemas y oportunidades | 1 | 0 | 1-oct |
| 2 | 1.1.1; Entrevista al cliente | 2 | 1 | 3-oct |
| 3 | 1.1.2; Revisar el proceso actual del cliente | 3 | 2 | 6-oct |
| 4 | 1.1.3; Documentar problemas y oportunidades | 5 | 3 | 11-oct |
| 5 | 1.2; Establecer equipo de trabajo | 5 | 4 | 16-oct |
| 6 | 1.3; Requerimientos del sistema | 1 | 5 | 17-oct |
| 7 | 1.3.1; Definir requerimientos funcionales | 3 | 6 | 20-oct |
| 8 | 1.3.2; Definir requerimientos no funcionales | 3 | 7 | 23-oct |
| 9 | 1.4; Definición de la arquitectura física del sistema | 3 | 8 | 26-oct |
| 10 | 2.1; Identificar datos de los requerimientos | 2 | 8,9 | 28-oct |
| 11 | 2.1.1; Identificar entradas de datos | 2 | 10 | 30-oct |
| 12 | 2.1.2; Identificar salidas de datos | 2 | 11 | 1-nov |
| 13 | 2.2; Diseño de prototipo para el usuario | 5 | 12 | 6-nov |
| 14 | 2.2.1; Diseñar pantallas interactivas | 10 | 13 | 16-nov |
| 15 | 2.2.2; Diseñar reportes | 2 | 14 | 18-nov |
| 16 | 2.3; Diseño de base de datos | 5 | 15 | 23-nov |
| 17 | 2.3.1; Definición lógica de datos y sus relaciones | 3 | 16 | 26-nov |
| 18 | 2.3.2; Diseñar estructura de datos | 3 | 17 | 29-nov |
| 19 | 2.3.3; Verificación diseño de base de datos | 1 | 18 | 30-nov |
| 20 | 2.4; Diseño de interfaces | 2 | 19 | 2-dic |
| 21 | 2.4.1; Diseño de la interfaz de administración | 5 | 20 | 7-dic |
| 22 | 2.4.2; Diseño de la interfaz de usuario | 5 | 21 | 12-dic |
| 23 | 2.5; Diseño de pruebas | 9 | 22 | 21-dic |
| 24 | 2.5.1; Identificar requerimientos a probar | 5 | 23 | 26-dic |
| 25 | 2.5.2; Crear checklist de pruebas | 8 | 24 | 3-ene |
| 26 | 2.6; Validar diseño lógico | 2 | 25 | 5-ene |
| 27 | 2.6.1; Validar relación de datos | 2 | 26 | 7-ene |
| 28 | 2.6.2; Validar relación de datos con los requerimientos | 2 | 27 | 9-ene |
| 29 | 3.1; Desarrollo de módulos lógicos | 20 | 28 | 29-ene |
| 30 | 3.2; Validación de los módulos lógicos | 3 | 29 | 1-feb |
| 31 | 3.3; Correcciones en los módulos lógicos | 5 | 30 | 6-feb |
| 32 | 3.4; Desarrollar módulos de pruebas unitarias | 8 | 31 | 14-feb |
| 33 | 3.5; Ejecución de pruebas unitarias | 1 | 32 | 15-feb |
| 34 | 3.6; Corrección de errores post pruebas unitarias | 5 | 33 | 20-feb |
| 35 | 4.1; Realizar pruebas funcionales | 3 | 34 | 23-feb |
| 36 | 4.2; Realizar pruebas del checklist | 3 | 35 | 26-feb |
| 37 | 5.1; Crear manual de usuario | 5 | 36 | 3-mar |
| 38 | 5.2; Crear manual de instalación | 5 | 37 | 8-mar |

## Seguimiento y Control del Proyecto

## Gestión de Requisitos

Los requerimientos del sistema son especificados en el documento Visión (Orea, 2009), cada requerimiento debe estar definido bajo importancia, estado con el fin de realizar seguimiento.

## Control de Plazos

El líder de proyecto será el encargado de realizar el seguimiento y evaluación al cronograma del proyecto, esto se realizará de forma semanal

## Control de Calidad

En el momento que de detecten defectos se deberán formalizar con una solicitud de cambio y se les realizará seguimiento para garantizar la conformidad a la solución.

## Gestión de Riesgos

Desde la fase de Inicio se realizará un listado de riesgos del proyecto y las acciones que se deben realizar para mitigarlos, esto se deberá revisar en cada iteración.

## Gestión de Configuración

Se realizará para llevar un registro de los artefactos generados y sus versiones. Se debe gestionar las solicitudes de cambio y modificaciones que se produzcan.

# Modelado del Negocio

A continuación, se define el modelo de casos de uso del negocio, objetos del negocio, modelo de datos, modelo de análisis y diseño. También se muestran los diagramas de componentes y diagramas de despliegue del proyecto.

## Preparatorios derechos

La universidad libre solicitante del proyecto de software tiene definido unos roles que se verán beneficiados con la realización de la aplicación. Las unidades involucradas se representan con los perfiles de Jefatura, Docente, Área de sistemas, Estudiante, Universidad externa. En donde la forma de interactuar se ve orientada a lo siguiente:

La jefatura solicita a los docentes de la universidad libre una serie de preguntas para realizar los eventos, los docentes se encargan de diseñar las preguntas con sus respectivas respuestas, asignara un peso y las dividirá por asignatura, En caso de que el evento sea a nivel general las universidades externas asumirán el Rol como docente. Cuando se tiene la información lista para el evento el área de sistemas se encargará de realizar la preparación del aplicativo y el equipo que se utilizara con los estudiantes en los auditorios donde se presentaran las pruebas.

El diagrama que representa los diferentes subsistemas en los que se ha dividido los involucrados en el sistema a nivel de abstracción es el siguiente:

Ilustración 1 Modelo subsistemas involucrados



# Modelo de casos de uso del negocio

Ilustración 2 Modelo de casos de uso del negocio



# Modelo de dominio

Ilustración 3 Modelo de dominio



# Modelo de Objetos

Ilustración 4 Modelo de objetos de preparación event



Ilustración 5 Modelo de objetos de creación de preguntas1

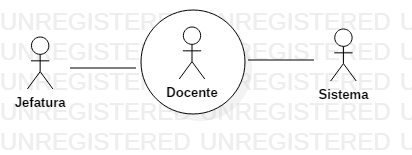


Ilustración 6 Modelo de objetos de creación de preguntas 2



Ilustración 7 Modelo de objetos configuración sistema



# Visión

## Alcance

El sistema de preparación y evaluación para los estudiantes de derecho de la universidad libre será desarrollado por los integrantes y desarrolladores del proyecto Cristian Camilo Agudelo y Alejandro Carvajal.

El sistema permitirá a la parte administrativa de la seccional Pereira de derecho, jefatura y docentes preparar de manera más rápida y efectiva las evaluaciones y preparatorios que se establecen para los estudiantes de derecho. Además de tener información histórica y en tiempo real de los participantes que han realizado las pruebas.

## Oportunidad de Negocio

El software permitirá a la Universidad libre sistematizar el proceso de evaluación y preparación de los estudiantes de derecho. Lo cual permitirá generar eventos de manera más constante y con mayor facilidad debido al banco de preguntas que se va generando en el sistema e igualmente permitirá el diseño de pruebas variables debido a la capacidad que tendrá el sistema de generar aleatoriamente las pruebas a las cuales harán frente los estudiantes. Una de las ventajas que más se verán reflejadas con la aplicación del sistema es que los docentes no tendrán que preocuparse por el cambio de preguntas, los empates y el tiempo de duración del evento ya que el sistema realizara esta labor por ellos.

## Sentencia que define el problema

* El problema: Generar de manera más constante, rápida y dinámica las pruebas para la evaluación y preparación de los estudiantes de derecho de la universidad libre y las demás entidades invitadas aplicando las nuevas tecnologías.
* Afecta a:
  + Universidad Libre
  + Jefatura
  + Docentes de derecho
  + Área de sistemas
  + Estudiantes
  + Entidades invitadas a los eventos
* El impacto asociado es: Permitir a los docentes generar las preguntas para los eventos de manera más sencilla, controlar dificultad y tiempos de respuesta gracias al apoyo del software que después de tener la información generara de manera aleatoria las preguntas a las cuales se enfrentaran los estudiantes.
* Solución adecuada: Agilizar el diseño de pruebas para la evaluación de los estudiantes por medio de un sistema que se encargue de generar las preguntas basado en el banco de preguntas que se va alimentando por los diferentes docentes

## Sentencia que define la posición del Producto

* Para:
  + Universidad Libre
  + Jefatura
  + Docentes de derecho
  + Área de sistemas
  + Estudiantes
  + Entidades invitadas a los eventos
* Quienes: Son los involucrados y beneficiarios del desarrollo del sistema
* Nombre del producto: Es una herramienta software. (SETUL)
* Que: Almacena la información de las preguntas, respuestas, docentes y participantes. Genera de manera automática aleatoria los test que deben realizar los participantes.
* No como: El sistema actual. La plantilla en powerpoint
* Nuestro producto: Permite la administración de la información como los docentes, asignaturas, participantes, test, cantidad de preguntas y dificultad de los “juegos” que llevara a cabo cada participante.

## Resumen de Stakeholders

Tabla 6 Stackeholders

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Descripción** | **Responsabilidades** |
| Amparo Rengifo Santibañez | Jefe de Área Derecho Laboral | Representa a todos los usuarios posibles del sistema.  Seguimiento del desarrollo del proyecto.  Aprueba requisitos y funcionalidades administrativas del software |
| Francisco | Docente derecho | Aprueba los requisitos y funcionalidades del control del modo juego del software |

## Resumen de Usuarios

Tabla 7 Resumen de usuarios

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre** | **Descripción** | | **Stakeholder** |
| **ACT1 Jefe eJefeJefe del área de derecho** | | Responsable de generar los eventos y establecer la logística para la ejecución de los mismos | Administración | |
| **Docente** | | Encargado de diseñar las preguntas y crear los test para los participantes a los eventos. También se encarga de manejar el software al momento de realizar las competencias controlando los tiempos y haciendo de presentador. | Docencia | |
| **ACT3 Jefe de Jefe de sistemas** | | Encargado de preparar el equipo sobre el cual se utilizará el software. También se encarga de alistar el auditorio o espacio donde se desarrollan los eventos verificando que los instrumentos técnicos y tecnológicos funcionen correctamente. | Sistemas | |
| **EstudianteACT4 Es** | | Persona a la cual se le aplicaran los test diseñados por el sistema para evaluar su conocimiento y dar un espacio de preparación para sus futuras pruebas. | STK3 EstuEstudiante | |
| **ACT5 Entidad Externa** | | Invitado a las pruebas el cual también puede cumplir como rol de docente aportando información(preguntas) para el desarrollo del evento | Externo | |

## Entorno de usuario

Los usuarios pueden hacer uso del sistema de manera web accediendo al dominio donde se encuentre alojado el sistema o de manera iniciando los servicios de APACHE y de MYSQL. El aplicativo se ejecutará sobre un navegador web del cual se recomienda Google Chrome.

## Perfil de los Stakeholders

Jefe del área de derecho

|  |  |
| --- | --- |
| **Representante** | Amparo Rengifo |
| **Descripción** | Jefe del área de derecho |
| **Tipo** | Abogada y docente |
| **Responsabilidades** | Responsable de generar los eventos y establecer la logística para la ejecución de los mismos |
| **Criterio de Éxito** | A definir por el cliente |
| **Grado de participación** | Revisión de requerimientos, estructura del sistema |
| **Comentarios** | Ninguno |

## Perfiles de Usuario

Docente

|  |  |
| --- | --- |
| **Representante** | STK1 DocenteDocente |
| **Tipo** | Docente. |
| **Responsabilidades** | Encargado de diseñar las preguntas y crear los test para los participantes a los eventos. También se encarga de manejar el software al momento de realizar las competencias controlando los tiempos y haciendo de presentador. |
| **Criterio de Éxito** | A definir por el cliente |
| **Grado de participación** | Alto |
| **Comentarios** | Ninguno |

### Jefe de Sistemas

|  |  |
| --- | --- |
| **Representante** | STK2 Jefe Ingeniero de sistemas |
| **Descripción** | Jefe de sistemas. |
| **Tipo** | Técnico |
| **Responsabilidades** | Encargado de preparar el equipo sobre el cual se utilizará el software. También se encarga de alistar el auditorio o espacio donde se desarrollan los eventos verificando que los instrumentos técnicos y tecnológicos funcionen correctamente.. |
| **Criterio de Éxito** | A definir por el cliente |
| **Grado de participación** | Bajo |
| **Comentarios** | Ninguno. |

### Estudiante

|  |  |
| --- | --- |
| **Representante** | STK2 EstudiantEstudiante |
| **Descripción** | Persona a la cual se le aplicaran los test diseñados por el sistema para evaluar su conocimiento y dar un espacio de preparación para sus futuras pruebas |
| **Tipo** | Asistente. |
| **Responsabilidades** | N/A. |
| **Criterio de Éxito** | A definir por el cliente |
| **Grado de participación** | Bajo |
| **Comentarios** | Ninguno. |

### Entidad Externa

|  |  |
| --- | --- |
| **Representante** | STK3 Entidad Entidad Externa |
| **Descripción** | Invitado al desarrollo de las pruebas |
| **Tipo** | Usuario invitado. |
| **Responsabilidades** | Cumplir como rol de docente aportando información(preguntas) para el desarrollo del evento no siempre se presenta |
| **Criterio de Éxito** | A definir por el cliente |
| **Grado de participación** | Medio |
| **Comentarios** | Ninguno. |

## Perspectiva del producto

El producto a desarrollar es un sistema el cual permite realizar de manera más recurrente y ágil los eventos en los cuales se prepara y evalúa a los estudiantes del área de derecho garantizando que las pruebas realizadas por un estudiante no serán igual a la anterior.

## Resumen de características

A continuación, se mostrará un listado con los beneficios que obtendrá el cliente a partir del producto:

Tabla 8 Resumen de características

|  |  |
| --- | --- |
| **Beneficio del cliente** | **Características que lo apoyan** |
| Obtener una herramienta que permita realizar de manera controlada las dinámicas de evaluación | Aplicación web desde la cual se administran las dinámicas de los eventos. |
| Presentar dinamismo en la generación de pruebas para cada estudiante | Sistema genera las preguntas que contienen los test de manera aleatoria. |
| Agilizar el proceso de registro de preguntas y respuestas. | Sistema permite cargar de manera manual y masiva por medio de un archivo xlsx la carga de información |
| Agilizar la inscripción de los participantes para las pruebas. | Posibilidad de registrar participantes de manera indivual y grupal. |
| Dinamizar las competencias con la generación de preguntas previamente establecidas. | La aplicación permitirá la administración de asignaturas y test previa a los eventos para no tener que hacer cambios durante el transcurso del evento |
| Guiar/preparar a los estudiantes para los exámenes que presentan al final de la carrera. | El sistema por ser una herramienta tecnológica y diseñada para ser dinámica y agradable permite más fácil la asimilación de las preguntas |
| Permitir un control automático sobre los listados de estudiantes que realizan las pruebas. | El sistema permite la generación de reportes sobre los participantes con capacidad de filtrar por fechas |
| Generar histórico de los puntajes obtenidos en las pruebas. | El sistema permite generar listados históricos de los participantes y sus resultados. |

## Descripción Global del Producto

Los denominados preparatorios para el área de derecho los cuales consisten en una serie de preguntas que se les realizan a los estudiantes de esta facultad para incentivarlos y prepararlos para sus exámenes finales. Teniendo en cuenta el avance en tecnología, requisitos que establecen los nuevos modelos de educación, la necesidad de generar una dinámica más llamativa e integrar en este proceso otras universidades de la región. La universidad libre opta por actualizar su metodología por medio de un software tipo trivia el cual se encargara de llevar el listado de participantes a las pruebas, permitirá la gestión de las preguntas y respuestas que se presentan durante el evento, permitirá la generación de pruebas de manera grupal e individual por medio de la asignación de test los cuales contienen una serie de preguntas aleatorias definidas por el sistemas basado en una característica de las preguntas denominada como “peso”, permitirá la creación de reportes y cartas que se utilizan para el control y posible premiación de los asistentes. Al final de esta implementación lo que se busca además de lo anteriormente mencionado es disminuir los tiempos de organización del evento y agilizar el desarrollo del mismo debido a que actualmente el proceso de asignación de preguntas se realiza manualmente en una herramienta ofimática que dificulta que el evento se desarrolle con fluidez

# Especificaciones de Casos de Uso

Se realiza descripción detallada de los casos que lo requieran, este documento debe incluir: precondiciones, post-condiciones, flujo de eventos, requisitos funcionales y no-funcionales asociados. Cuando los eventos tienen un flujo complejo, se debe realizar diagrama de actividad.

# Especificaciones Adicionales

Se registran los requerimientos y/o requisitos que no se han incluido en los casos de uso y se definen como requerimientos no funcionales.

# Prototipos de Interfaces de Usuario

## Interfaces de administrador

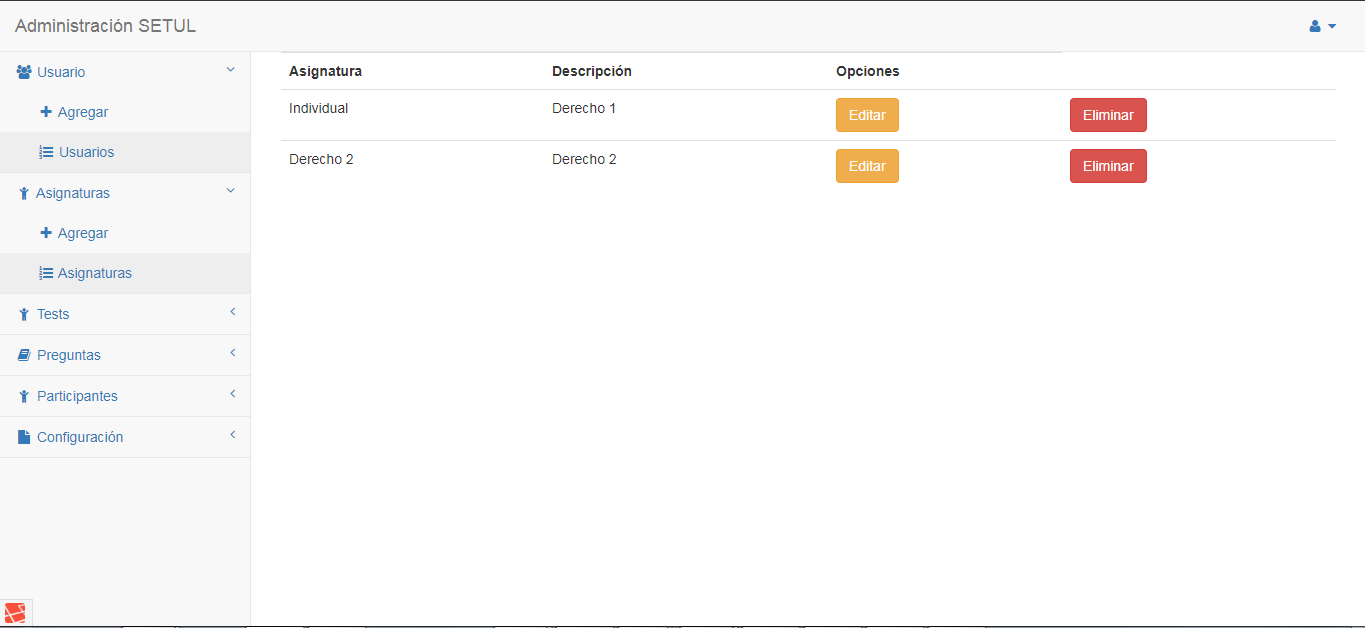
Son el conjunto de interfaces que permite configurar todas las opciones y datos que permiten la ejecución de SETUL, estas interfaces son:

Asignaturas

Son las interfaces encargadas de administrar las asignaturas que se evalúan en SETUL:

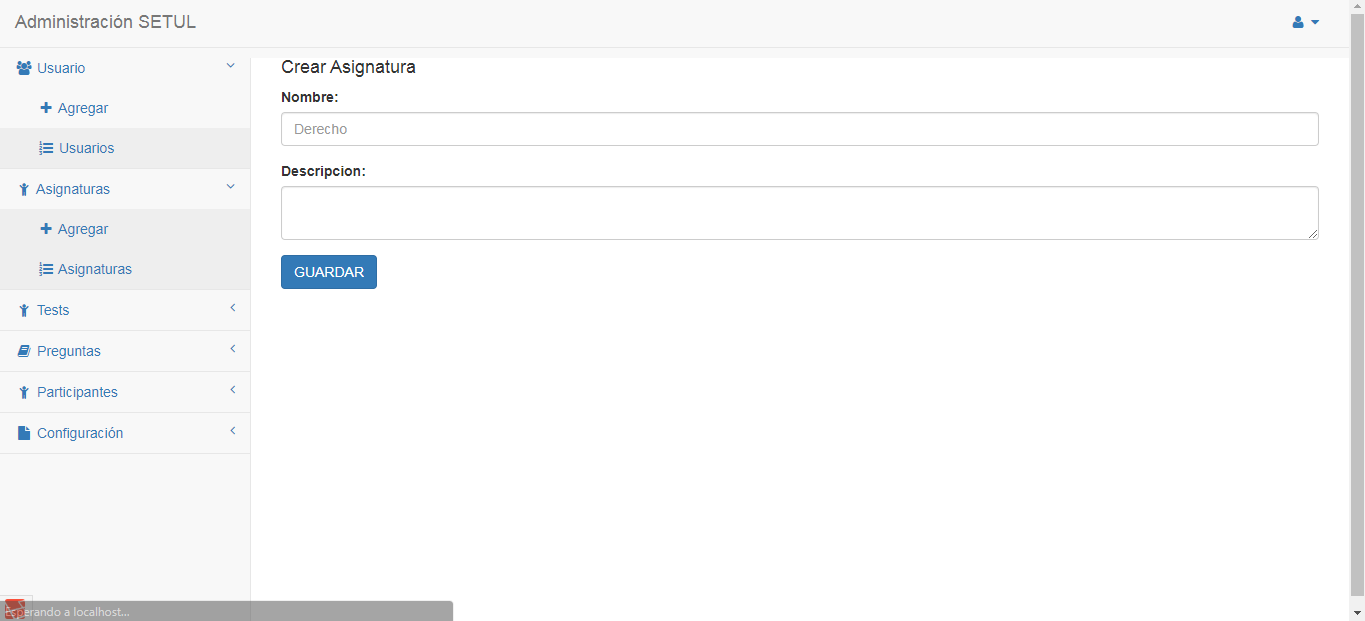
* Listar: Se encarga de mostrar todas las asignaturas con una descripción y permite las acciones de editar y eliminar

Ilustración 8 Listar asignaturas



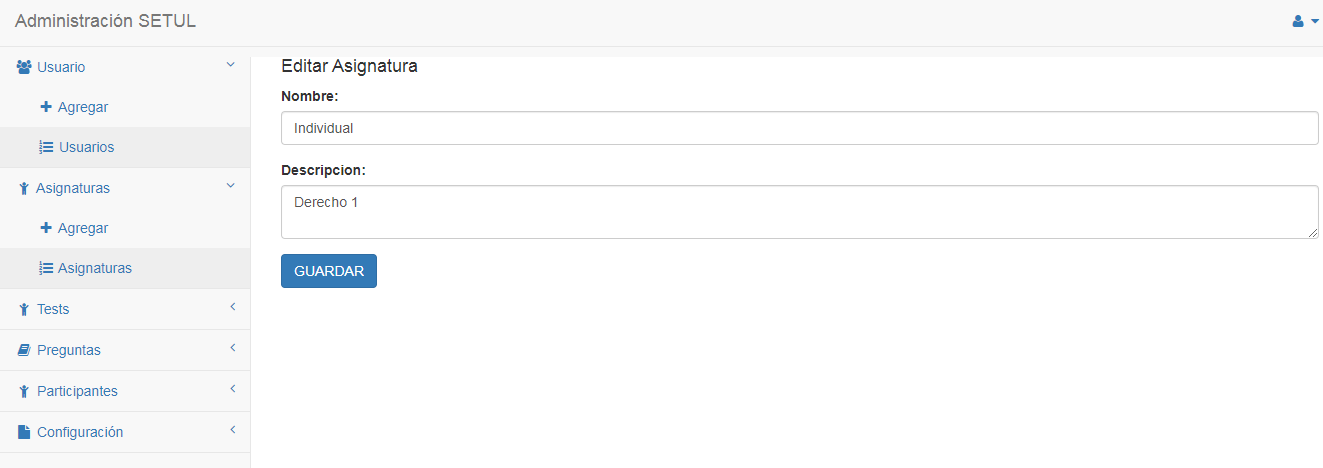
* Crear: Se encarga de mostrar un formulario con los campos necesarios para la creación de la asignatura

Ilustración 9 Creación de asignatura



* Editar: Se encarga de mostrar un formulario con los campos diligenciados de la asignatura para su posterior edición

Ilustración 10 Edición de asignatura

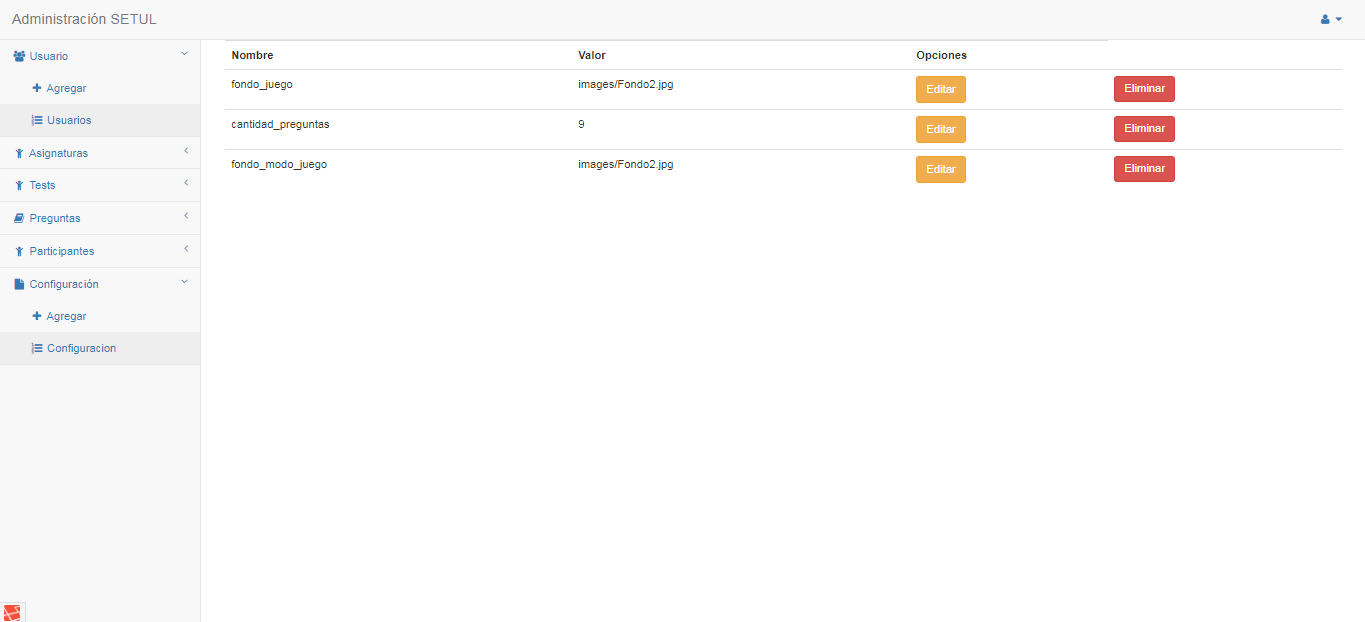


Configuración

Son las interfaces encargadas de administrar las configuraciones personalizables de SETUL:

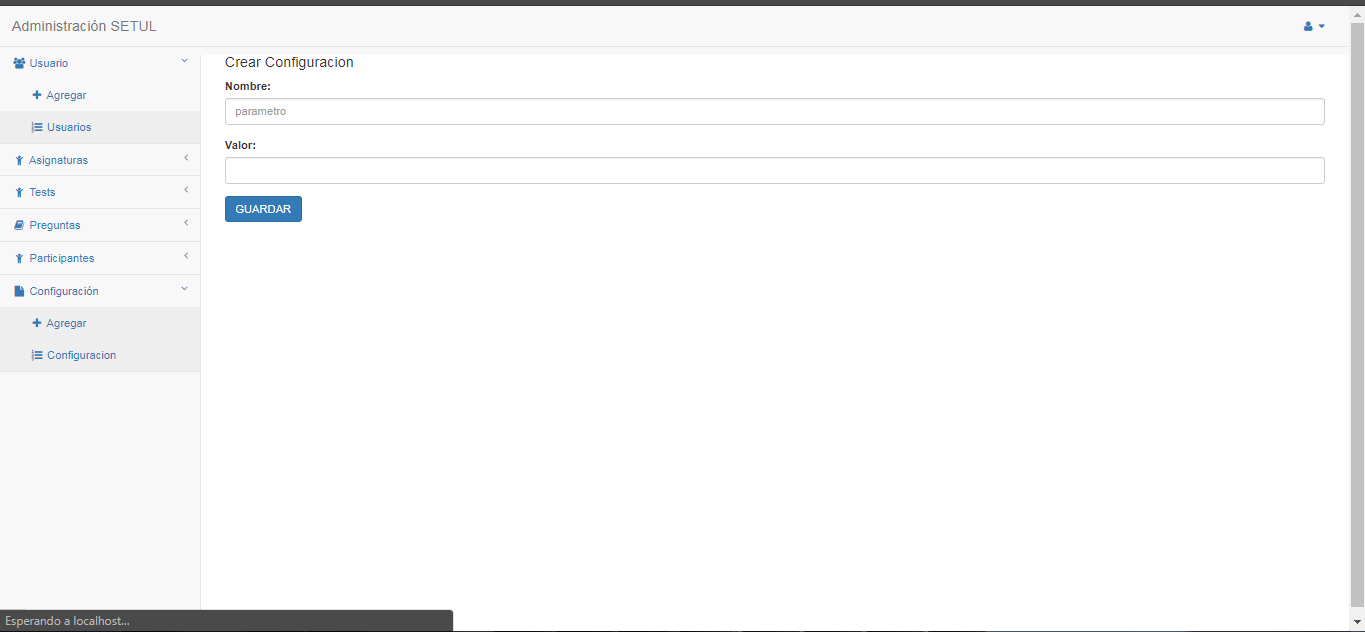
* Listar: Se encarga de mostrar todas las configuraciones con su respectivo nombre, valor y permite las acciones de editar y eliminar.

Ilustración 11 Listar configuración



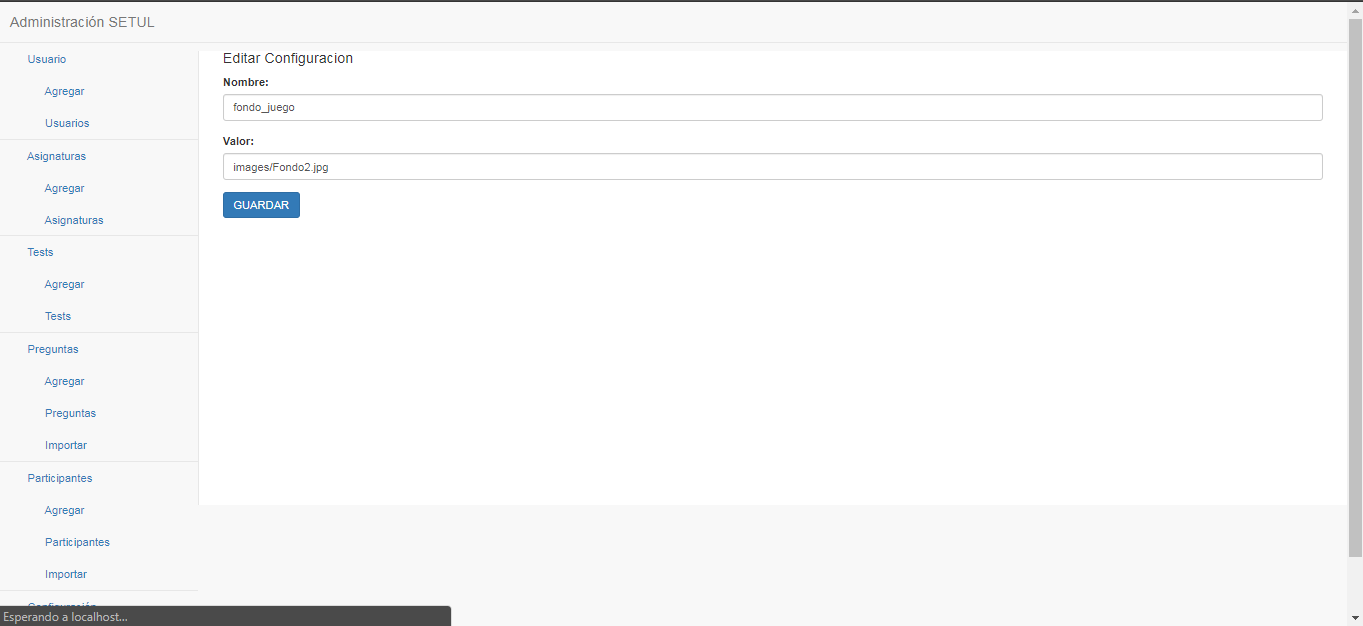
* Crear: Se encarga de mostrar un formulario con los campos necesarios para la creación de una nueva configuración

Ilustración 12 Creación de configuración



* Editar: Se encarga de mostrar un formulario con los campos diligenciados de la configuración para su posterior edición

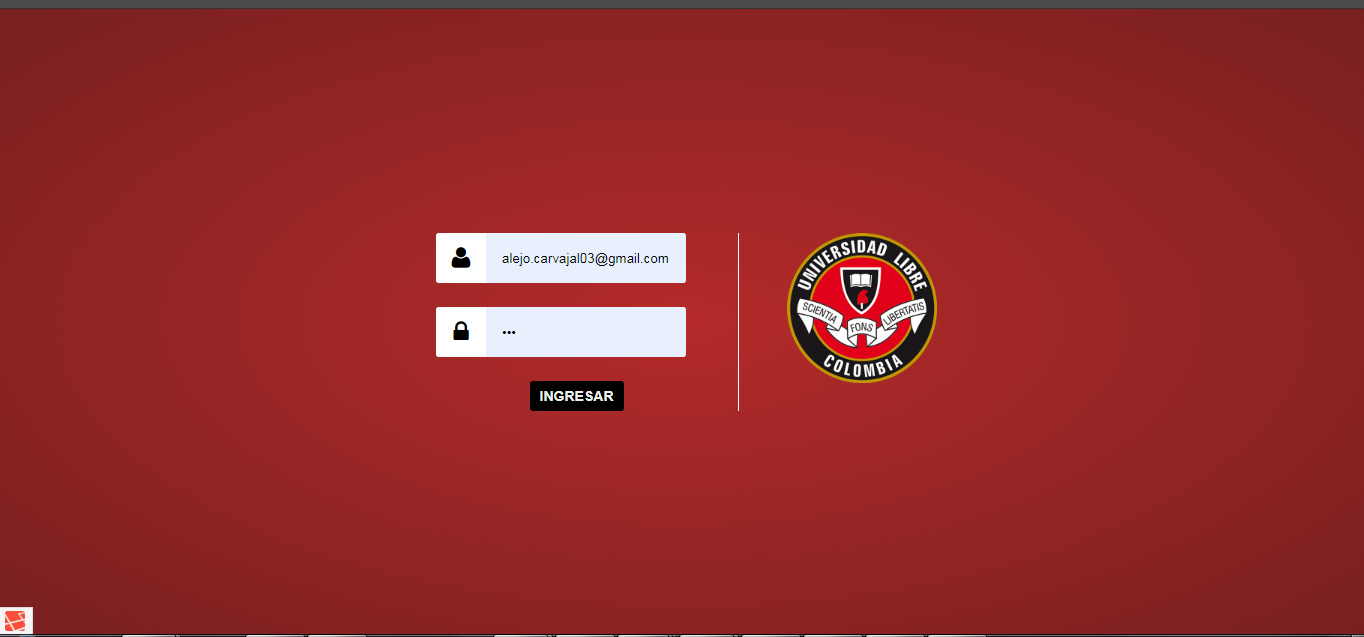
Ilustración 13 Edición de configuración



Login

Es la interfaz encargada de solicitar las credenciales de autenticación para el acceso a la administración de SETUL

Ilustración 14 Login



Administración

Es la interfaz principal de administración de SETUL, inicia mostrando los usuarios actuales, a la izquierda muestra todas las opciones que se pueden administrar y en la parte superior derecha muestra la opción de Logout, la cual permite cerrar la sesión del usuario:

Ilustración 15 Inicio de administración

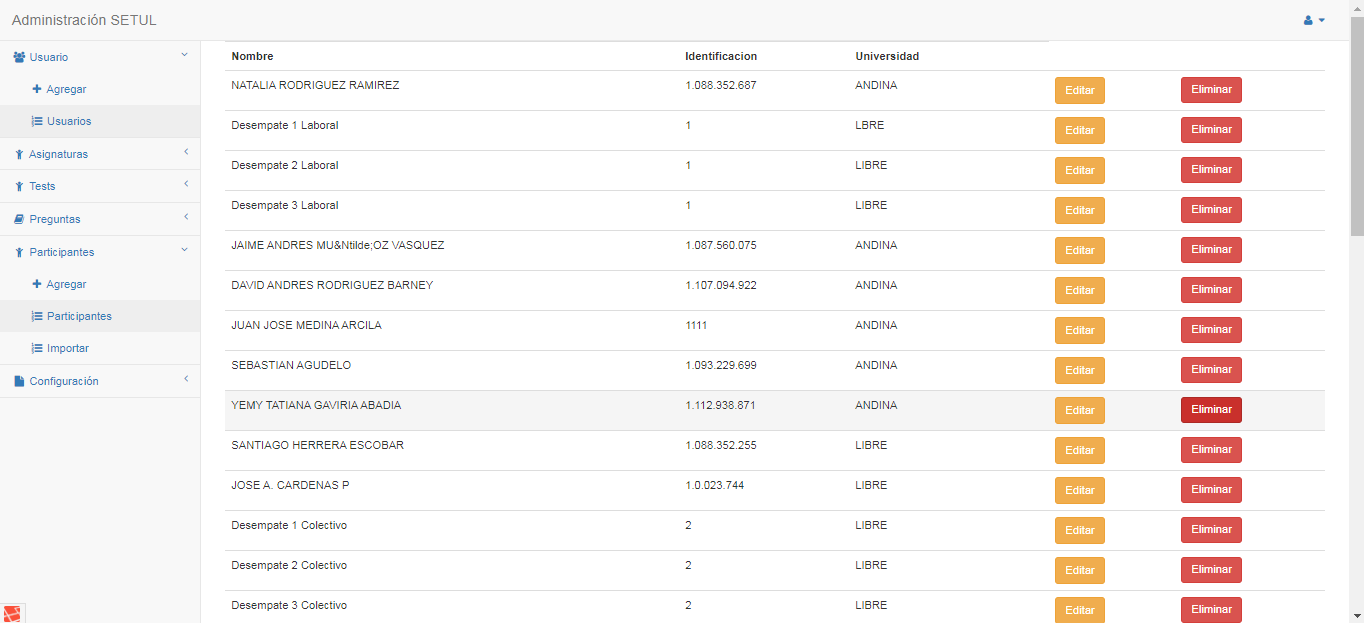


Participantes

Son las interfaces encargadas de administrar los participantes de SETUL:

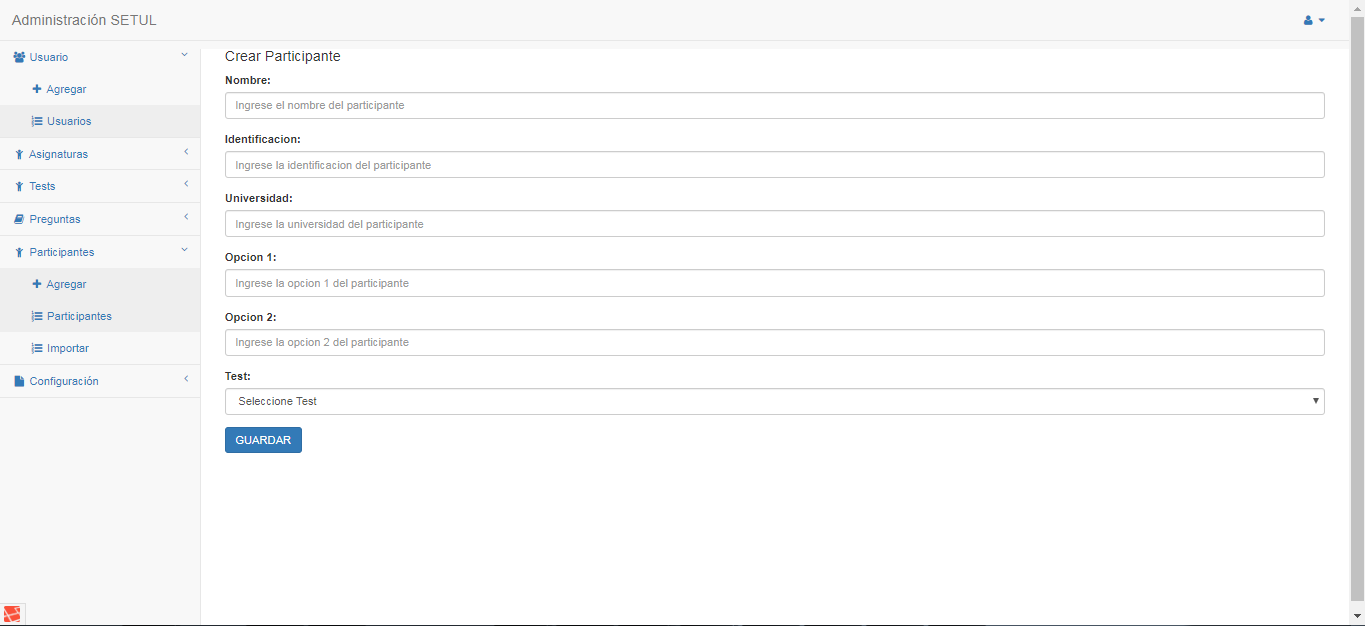
* Listar: Se encarga de mostrar todos los participantes con su respectivo nombre, identificación, universidad y permite las acciones de editar y eliminar.

Ilustración 16 Listar participantes



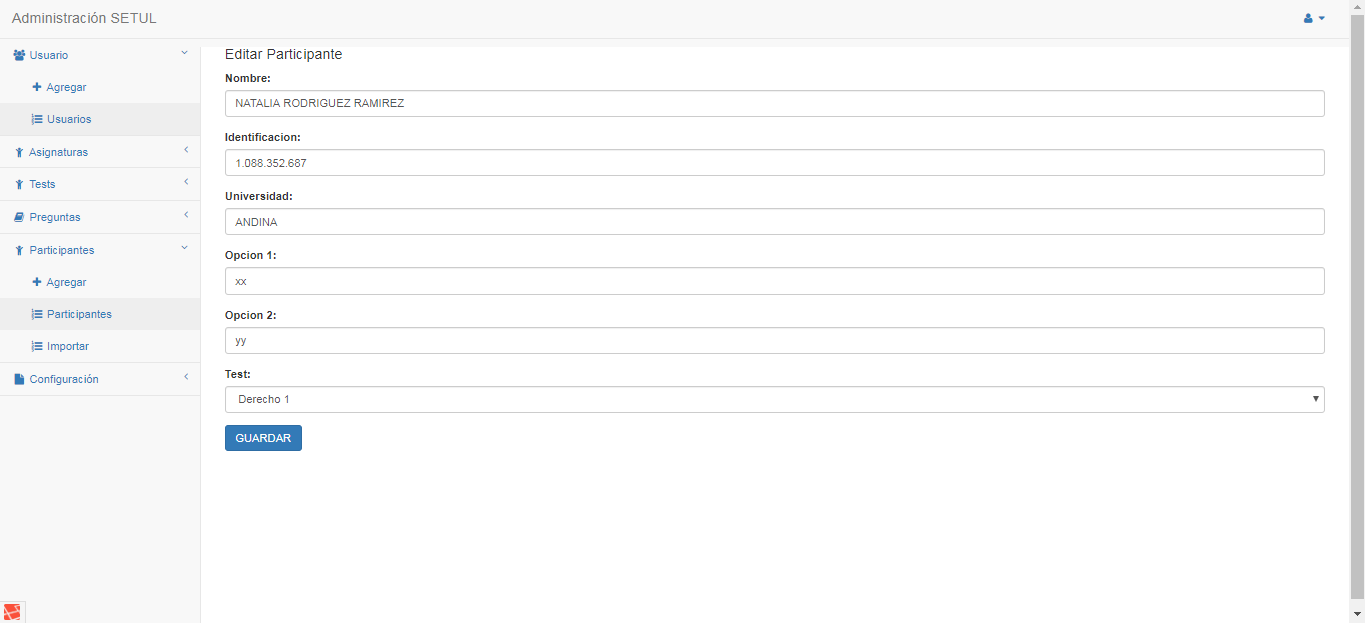
* Crear: Se encarga de mostrar un formulario con los campos necesarios para la creación de un participante

Ilustración 17 Creación de participante



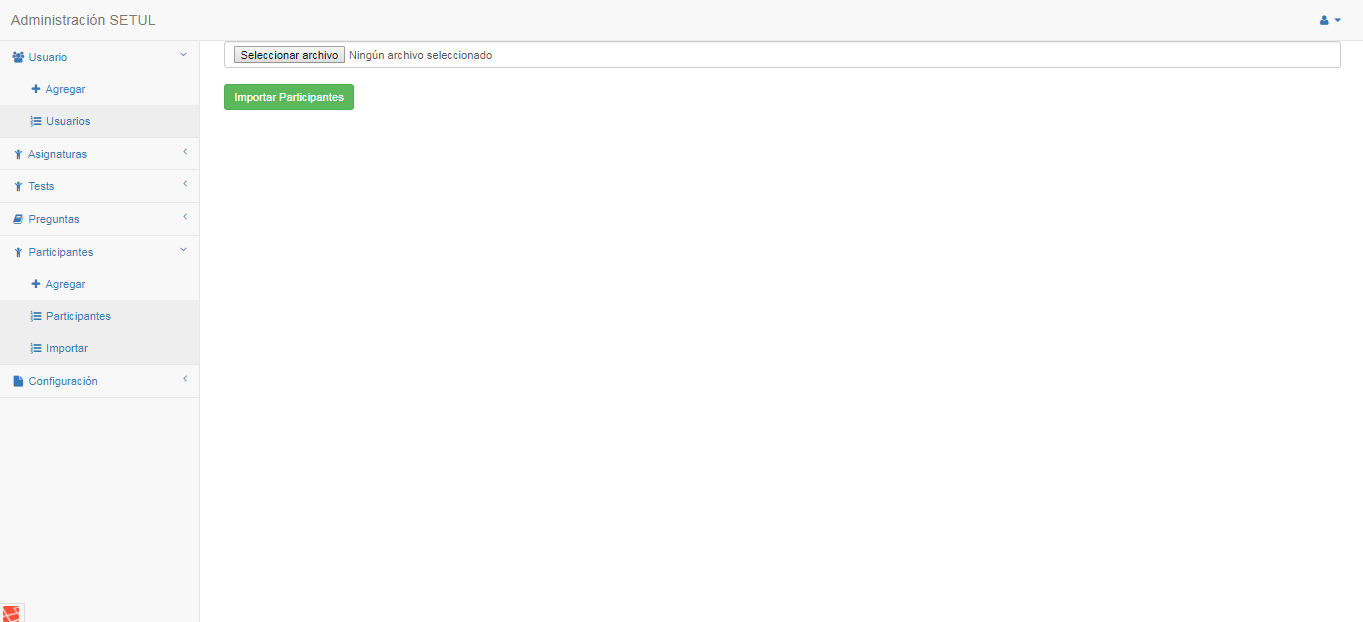
* Editar: Se encarga de mostrar un formulario con los campos diligenciados del participante para su posterior edición

Ilustración 18 Edición de participante



* Importar: Se encarga de mostrar un formulario que permite subir un documento xls/xlsx como adjunto, el cual permite la creación masiva de participantes

Ilustración 19 Importar participantes

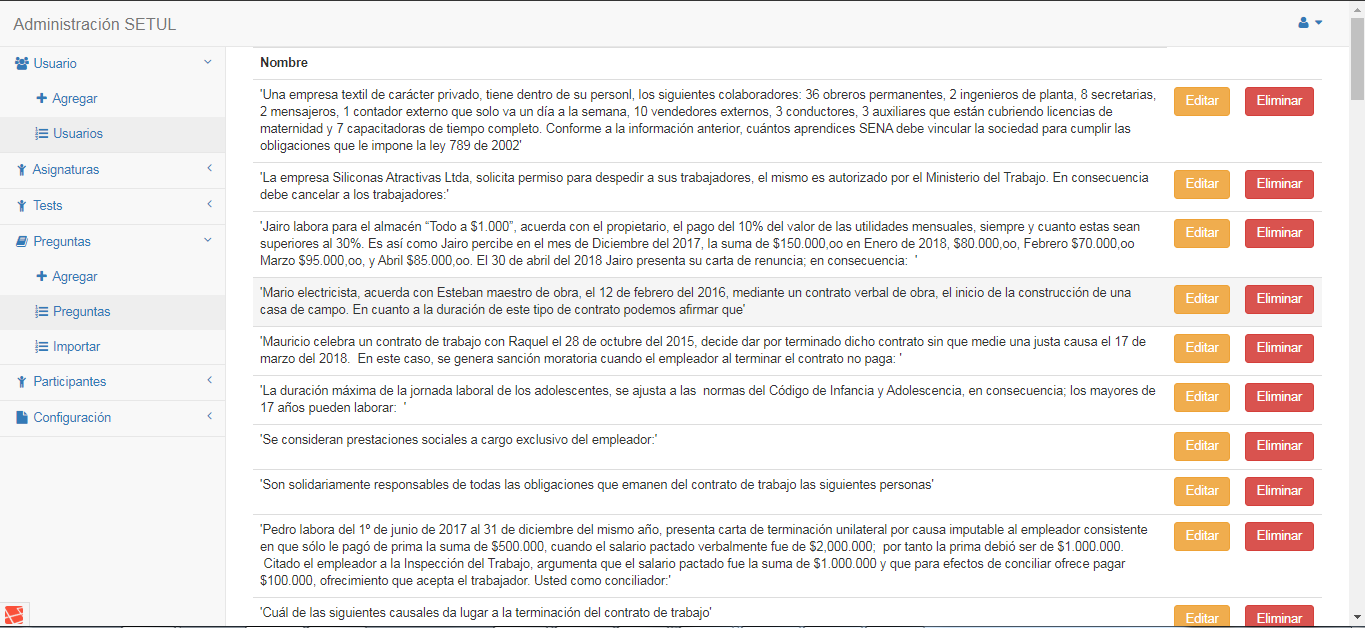


Preguntas

Son las interfaces encargadas de administrar las preguntas de SETUL:

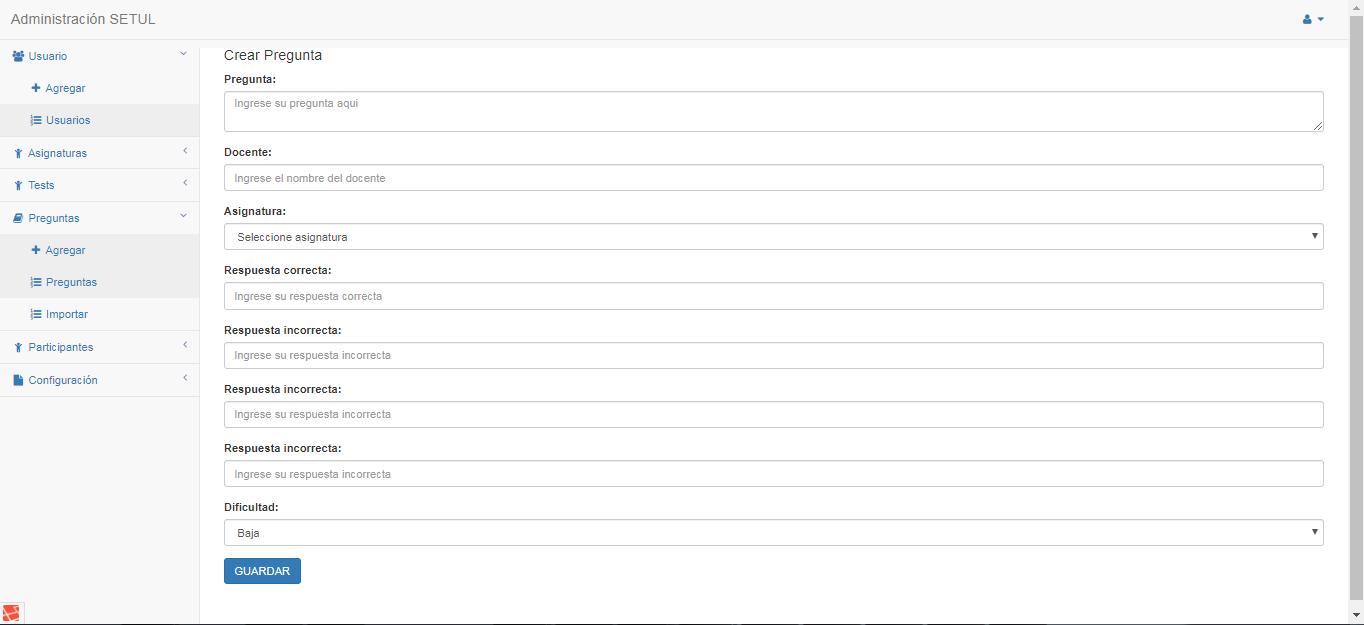
* Listar: Se encarga de mostrar todas las preguntas y permite las acciones de editar y eliminar.

Ilustración 20 Listar preguntas



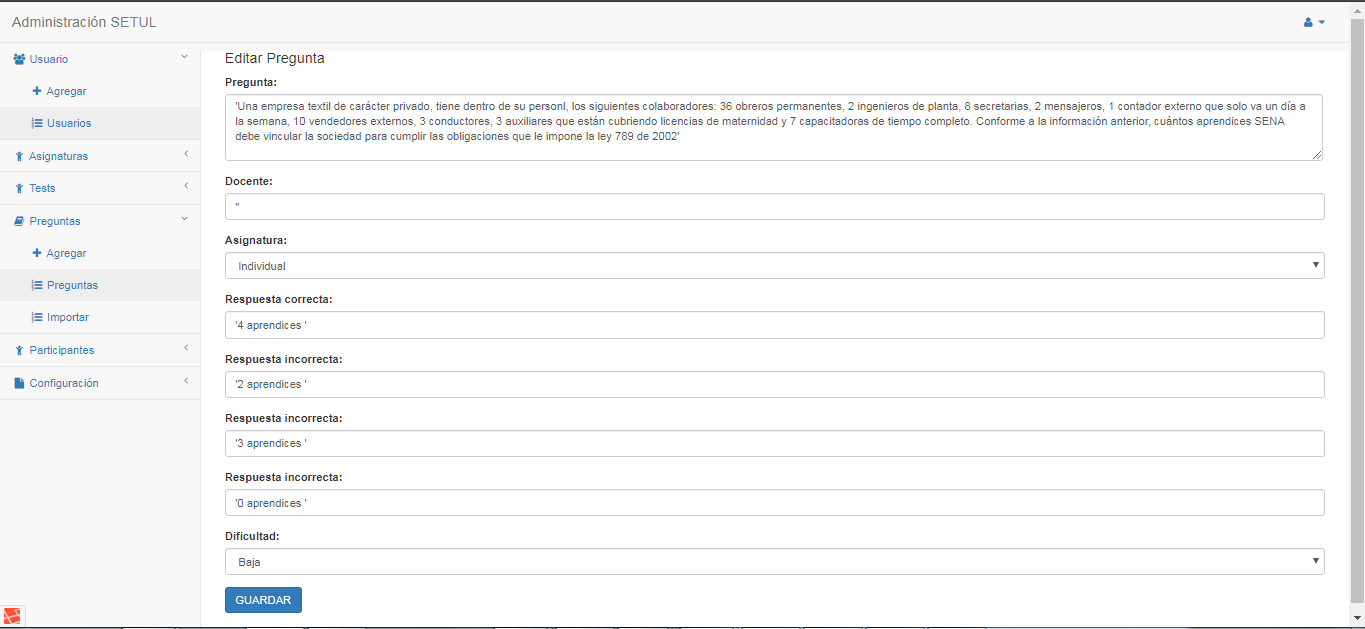
* Crear: Se encarga de mostrar un formulario con los campos necesarios para la creación de una nueva pregunta con sus respectivas respuestas

Ilustración 21 Creción de pregunta



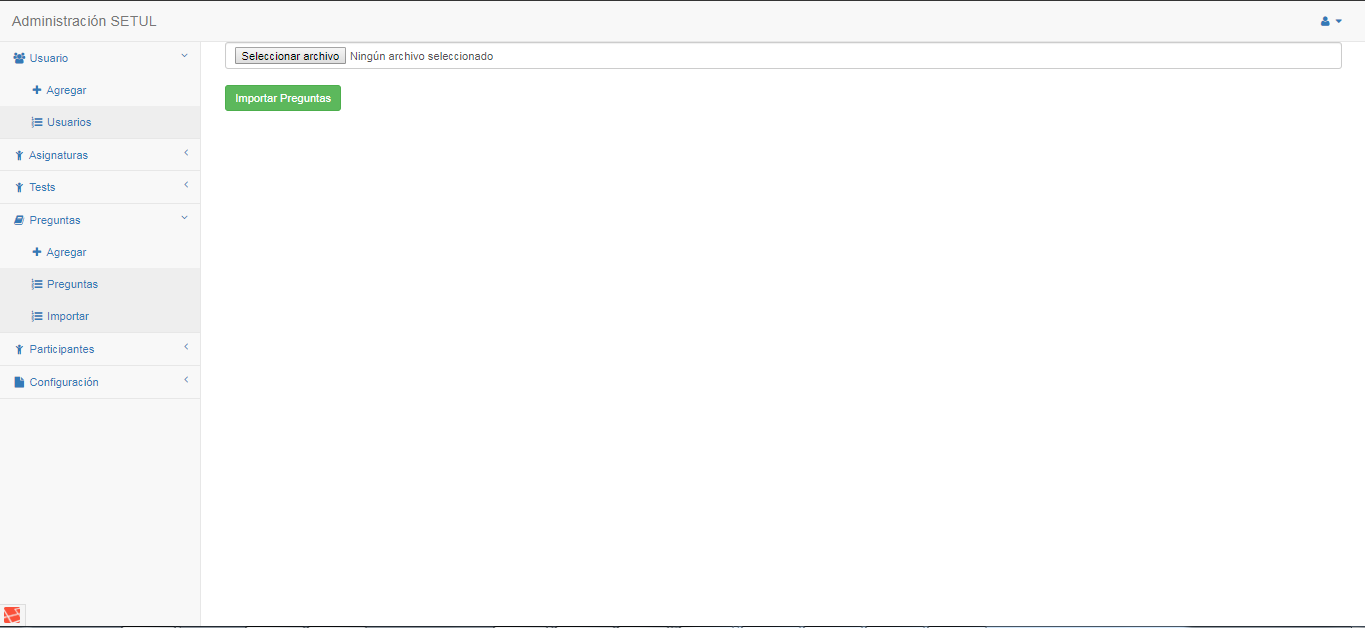
* Editar: Se encarga de mostrar un formulario con los campos diligenciados de la pregunta para su posterior edición

Ilustración 22 Edición de pregunta



* Importar: Se encarga de mostrar un formulario que permite subir un documento xls/xlsx como adjunto, el cual permite la creación masiva de preguntas

Ilustración 23 Importar preguntas

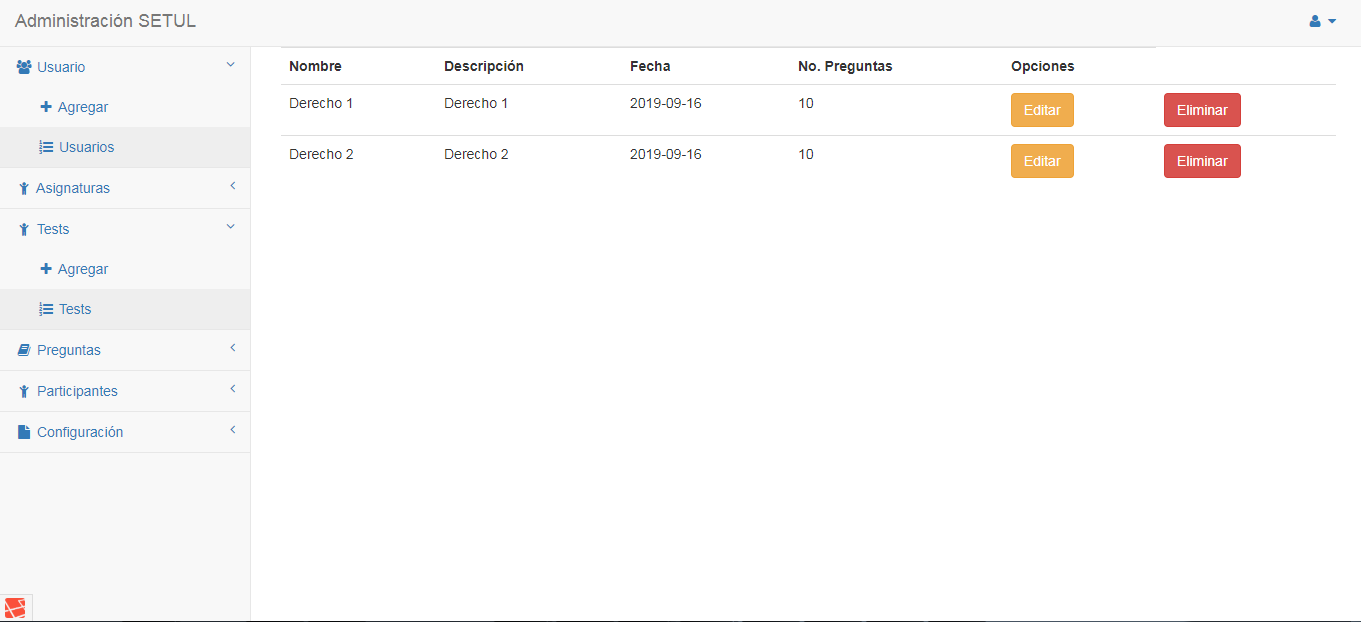


Tests

Son las interfaces encargadas de administrar los tests de SETUL:

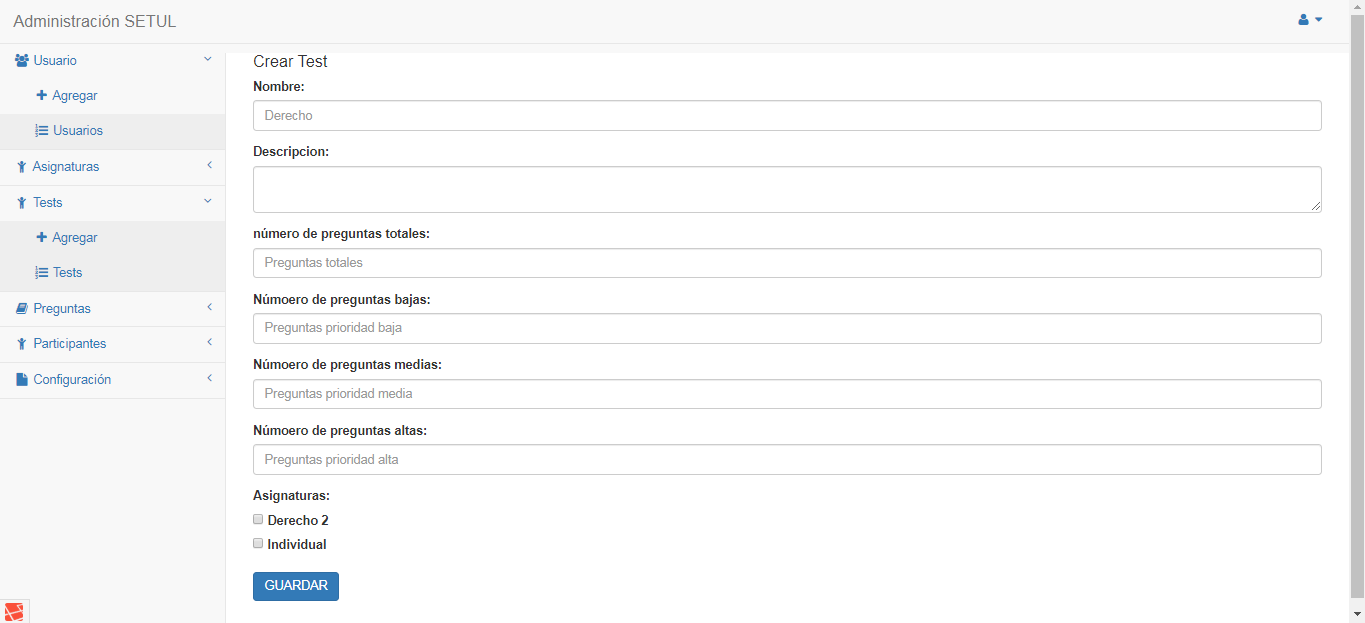
* Listar: Se encarga de mostrar todos los tests con su respectivo nombre, descripción, fecha, número de preguntas y permite las acciones de editar y eliminar.

Ilustración 24 Listar tests



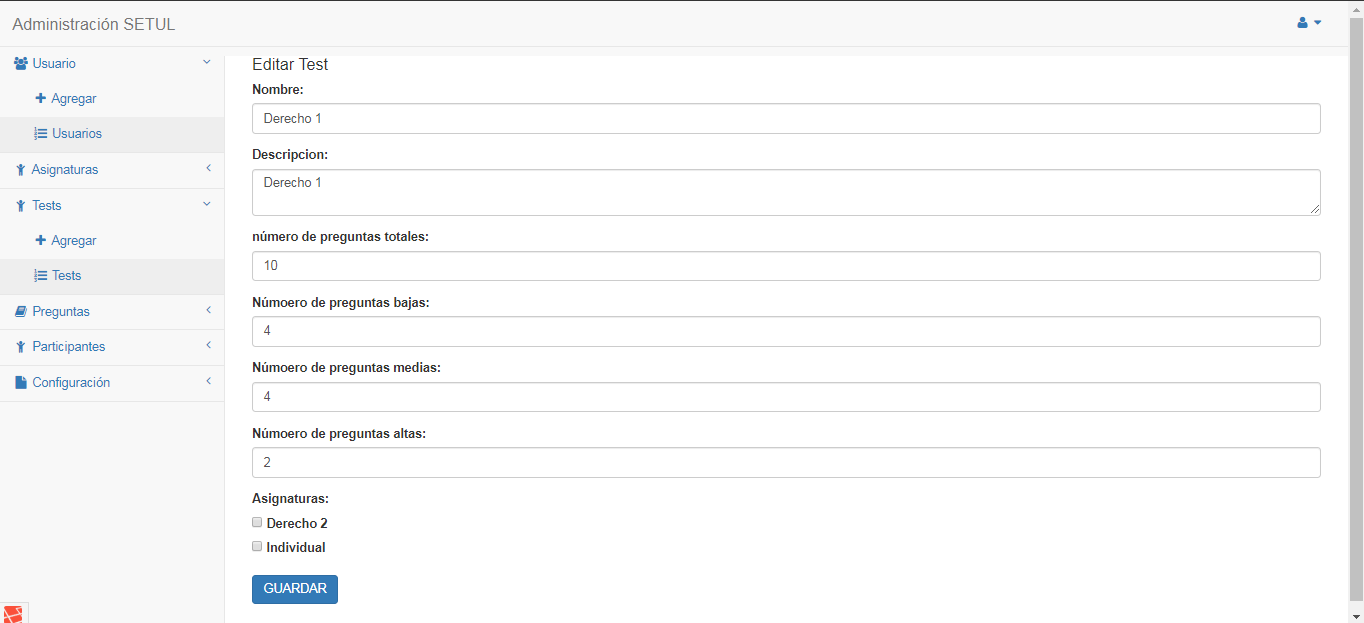
* Crear: Se encarga de mostrar un formulario con los campos necesarios para la creación de un nuevo test

Ilustración 25 Creación de test



* Editar: Se encarga de mostrar un formulario con los campos diligenciados del test para su posterior edición

Ilustración 26 Edicón de test



Usuarios

Son las interfaces encargadas de administrar los usuarios que pueden acceder a la administración de SETUL:

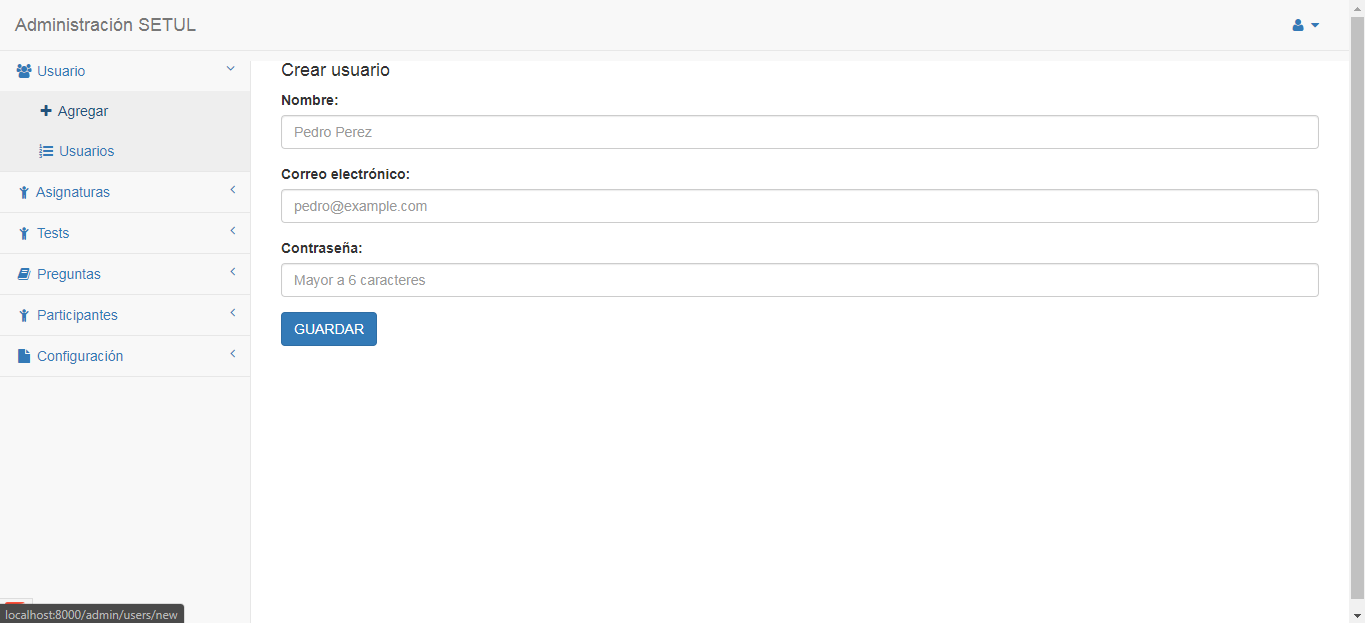
* Listar: Se encarga de mostrar todos los usuarios con su respectivo nombre, email y permite las acciones de editar y eliminar.

Ilustración 27 Lista de usuarios



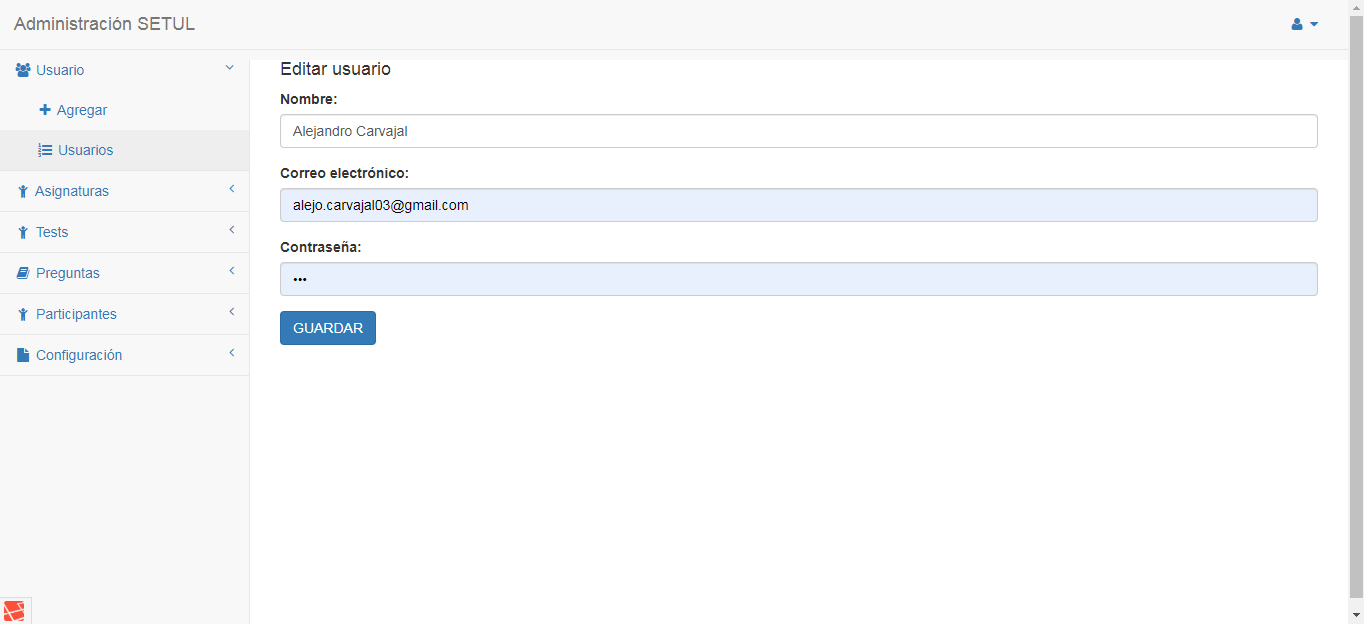
* Crear: Se encarga de mostrar un formulario con los campos necesarios para la creación de un nuevo usuario

Ilustración 28 Creación de usuario



* Editar: Se encarga de mostrar un formulario con los campos diligenciados del usuario para su posterior edición

Ilustración 29 Edición de usuario



# Modelo de Análisis y Diseño

Ilustración 30 Diagrama de clases Controller

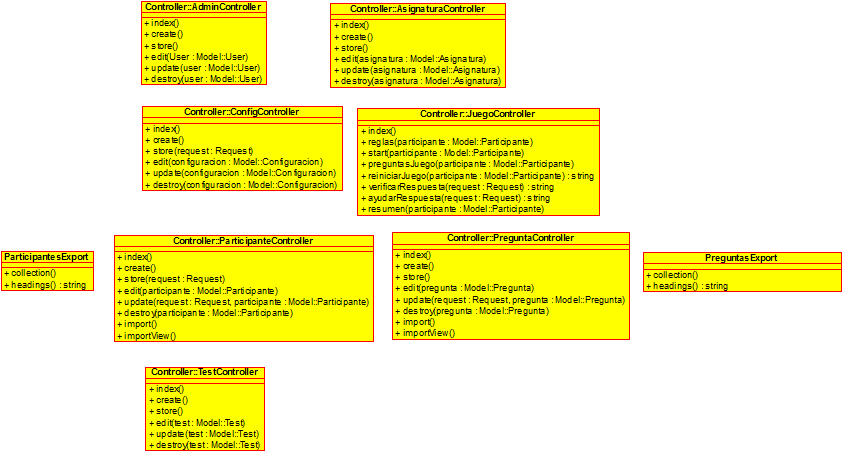
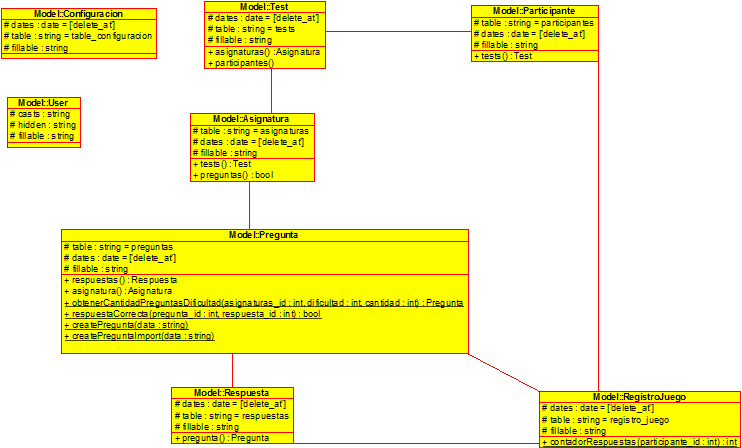
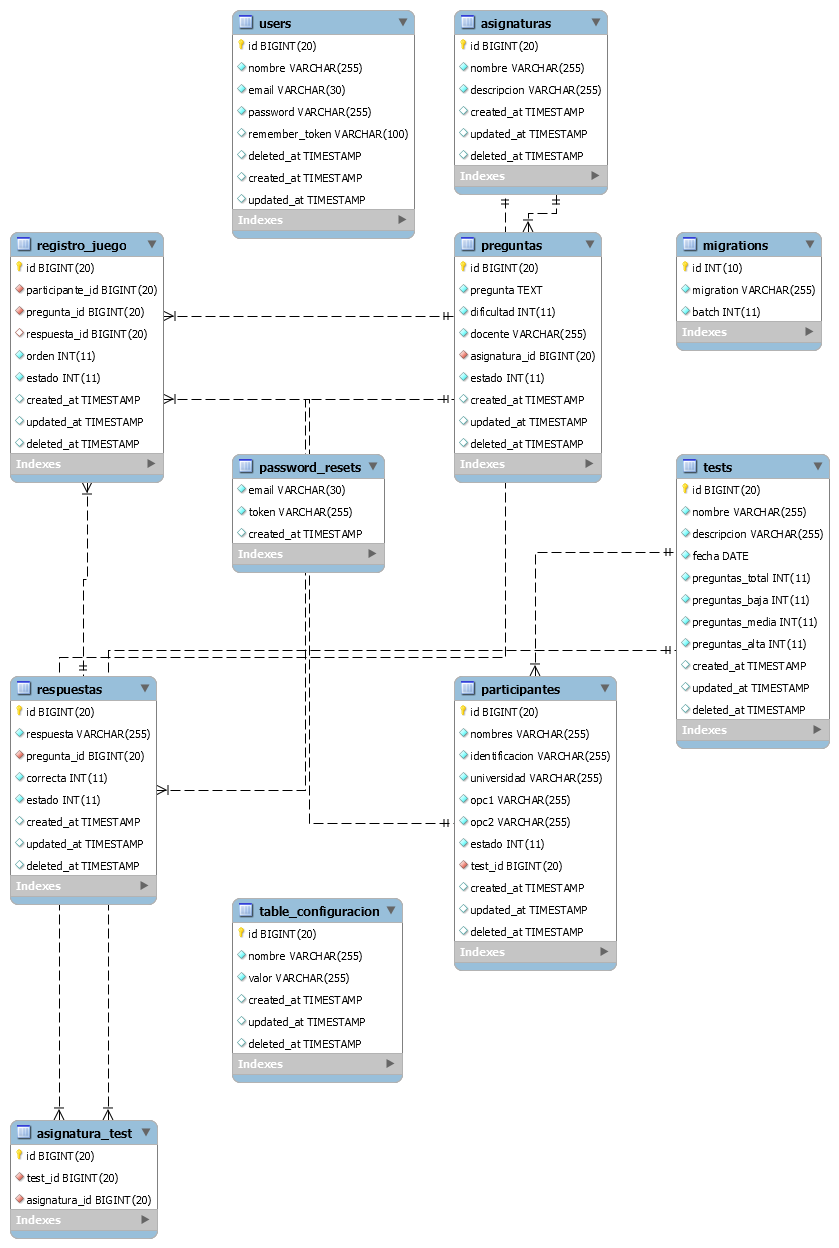


Ilustración 31 Diagrama de clases Model



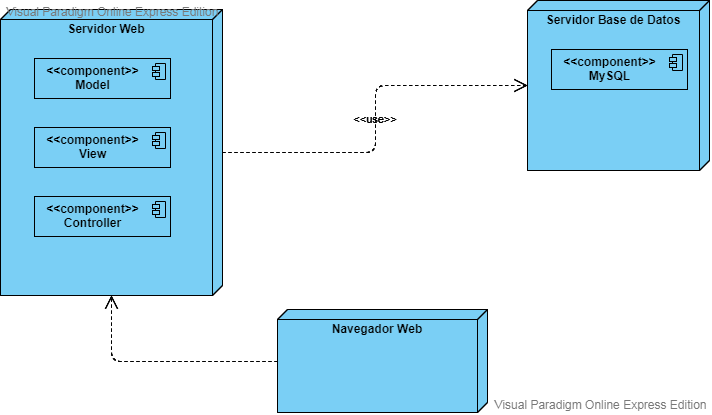
# Modelo de Datos

Ilustración 32 Modelo Entidad Relación



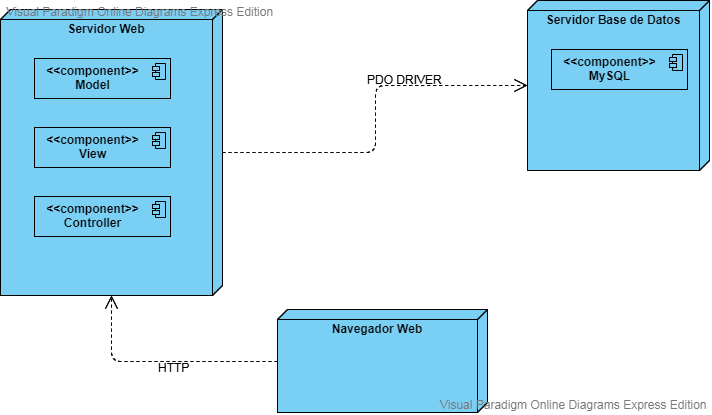
# Modelo de Implementación

Ilustración 33 Modelo de Implementación



# Modelo de Despliegue

Ilustración 34 Modelo de despliegue



# Casos de Prueba

Es un documento que indica las entradas, condiciones de ejecución y los resultados esperados.

# Manual de Instalación

Falta todo

# Material de Apoyo al Usuario Final

Falta la parte del juego

# Producto

Se debe almacenar el producto en un CD con el manual de instalación con el fin de facilitar la instalación.

# Bibliografía

Alvarez, M. A. (02 de 01 de 2014). *Qué es MVC*. Obtenido de https://desarrolloweb.com/articulos/que-es-mvc.html

Colombia, C. d. (23 de 06 de 1989). *Decreto 1360 de 1989*. Obtenido de Alcaldia de Bogotá D.C: https://www.alcaldiabogota.gov.co/sisjur/normas/Norma1.jsp?i=10575

Colombia, C. d. (15 de 01 de 2009). *Ley 1273 del 2009*. Obtenido de Secretaría del senado: http://www.secretariasenado.gov.co/senado/basedoc/ley\_1273\_2009.html

Colombia, C. d. (19 de 07 de 2010). *Ley 1403 del 2010*. Obtenido de Secretaría del senado: http://www.secretariasenado.gov.co/senado/basedoc/ley\_1403\_2010.html

Colombia, C. d. (18 de 10 de 2012). *Ley 1581 de 2012*. Obtenido de Secretaría del Senado: http://www.secretariasenado.gov.co/senado/basedoc/ley\_1581\_2012.html

deafinder. (05 de 2005). *TRIVIAL PURSUIT*. Obtenido de ideafinder: http://www.ideafinder.com/history/inventions/trivialpursuit.htm

Delgado, L. (2017). *El modelo pedagógico constructivista*. Obtenido de gestionparticipativa: http://www.gestionparticipativa.coop/portal/index.php?option=com\_content&view=article&id=255:el-modelo-pedagogico-constructivista&catid=38:travel-tips&Itemid=489

DNDA. (1993). *Decisión Andina*. Obtenido de Dirección Nacional de Derechos de Autor: http://derechodeautor.gov.co/decision-andina

EAN, U. (s.f.). *Universidad EAN*. Obtenido de https://universidadean.edu.co/preguntas-frecuentes/modelo-pedagogico

EcuRed. (26 de 10 de 2012). *Metodología para la elicitación de requisitos*. Obtenido de EcuRed: https://www.ecured.cu/Metodolog%C3%ADa\_para\_la\_elicitaci%C3%B3n\_de\_requisitos

EcuRed. (14 de 08 de 2013). *Ingeniería de software*. Obtenido de https://www.ecured.cu/Ingenier%C3%ADa\_de\_software

EcuRed. (30 de 01 de 2015). *Programación orientada a objetos*. Obtenido de EcuRed: https://www.ecured.cu/Programaci%C3%B3n\_Orientada\_a\_Objetos

Educació, F. J. (18 de 05 de 2018). *Qué es un ORM y por qué utilizarlo en el desarrollo de aplicaciones*. Obtenido de https://fp.uoc.fje.edu/blog/que-es-un-orm-y-por-que-utilizarlo-en-el-desarrollo-de-aplicaciones/

Francia, J. (25 de 09 de 2017). *Que es SCRUM*. Obtenido de SCRUM: https://www.scrum.org/resources/blog/que-es-scrum

García, J. M. (11 de 12 de 2015). *¿Qué es Laravel?* . Obtenido de https://www.arsys.es/blog/programacion/que-es-laravel/

Lara, D. (03 de 08 de 2007). *Encapsulamiento en programación orientada a objetos*. Obtenido de https://laraveles.com/series/poo/encapsulamiento-programacion-orientada-objetos/

Magisterio. (14 de 12 de 2017). Obtenido de ¿Qué es un modelo pedagógico?: https://www.magisterio.com.co/articulo/que-es-un-modelo-pedagogico

Neosoft. (08 de 01 de 2018). *Que es una aplicación web*. Obtenido de neosoft: https://www.neosoft.es/blog/que-es-una-aplicacion-web/

Orea, S. V. (06 de 2009). *Plan de desarrollo de software*. Obtenido de Universidad Tecnológica de Izúcar de Matamoros: http://www.utim.edu.mx/~svalero/docs/pds1.pdf

Ortiz, A. E. (5 de 04 de 2018). *Qué es un framework en informática o programación*. Obtenido de https://blog.hostdime.com.co/que-es-un-framework-informatica-programacion/

PHP. (s.f.). *¿Qué es PHP?* Obtenido de https://php.net/manual/es/intro-whatis.php

pmoinformatica.com. (18 de 03 de 2015). *Que es el Agile Testing y cuáles son sus principios y estrategias*. Obtenido de pmoinformatica: http://www.pmoinformatica.com/2015/03/que-es-el-agile-testing.html

Puerta, A. R. (06 de 2018). *Modelo Pedagógico Romántico: Historia, Ventajas, Desventajas y Características*. Obtenido de Lifeder: https://www.lifeder.com/modelo-pedagogico-romantico

Puerta, A. R. (06 de 2018). *Modelo Pedagógico Tradicional: Origen y Características Principales*. Obtenido de Lifeder: https://www.lifeder.com/modelo-pedagogico-tradicional/

Salazar, D. (06 de 06 de 2013). *modelo pedagogico desarrollista*. Obtenido de modelodesarrollistapedago: http://modelodesarrollistapedago.blogspot.com/

tutorialspoint.com. (s.f.). *SOFTWARE - VISIÓN GENERAL*. Obtenido de https://www.tutorialspoint.com/es/software\_engineering/software\_engineering\_overview.htm