INDICE

1- Análisis de requerimientos, investigación previa y diseño de la arquitectura de software.

* Preparación del proyecto para dar inicio al desarrollo.(herramientas y área de trabajo necesario)
* Investigación de tecnologías con la cual se va a desarrollar el sistema.
* Explicación de las especificaciones pedidas con el jefe inmediato para ver si fueron comprendidas.
* Investigación de la arquitectura de software a utilizar.

2-Modelado del sistema web.

* Se analizará las diferentes tipos de requisitos y ver que especificaciones son más importantes para el giro dentro del desarrollo.
* Se investigarán los diseños que existen en la actualidad.
* Se realizará el eventual diseño del sistema, para ver el modelado del producto final.

3-Construccion del sistema web.

* Se empieza la codificación del sistema con los lenguajes de programación seleccionados (HTML, PHP, JAVASCRIPT, MYSQL, o los elegidos por el encargado).
* Pruebas funcionales al sistema para verificar que se cumplan los requisitos.

4-Despliegue del sistema web.

* Se preparará el entorno en el cual será lanzado el sistema. Se probará para ver el funcionamiento del sistema en diferentes entornos.

5-Correcciones Finales.

* Se corregirán los errores que salgan en el transcurso final, así como errores mínimos que contenga el software.
* Posibles nuevas funcionalidades.

6-Documentación.

* Documentación del sistema web.

Índice de imágenes

**1. Introducción**

En el presente documento se redacta la descripción del proyecto “**Desarrollo de sistema de software para la integración de proyectos en subdirección gestión de fondos**”.

El presente proyecto será detallado por su servidor como persona que cursa estudios en un centro docente en el **Instituto Tecnológico de Culiacán** para elaborar un sistema web que ayudará a la empresa a realizar sus funciones administrativas referentes a la subdirección de gestión de fondos de forma más sencilla.

La iniciativa de crear la fundación Markoptic Medical Technology con un centro de investigación científico, desarrollo tecnológico e innovación, surge por la necesidad de desarrollar tecnología para controlar un padecimiento. Fundacion Markoptic con el tiempo se dio cuenta de la cantidad tan grande de personas que padecían diferentes tipos de complicaciones por lo que hace tiempo se ha dedicado a hacer un labor social, donde no solo se realizaran microválvulas, si no también prótesis físicas entre otros avances tecnológicos en beneficio al ser humano, el cual ayudaran a tener una mejor calidad de vida

Este sistema web facilitara el trabajo del departamento de gestión de fondos que lleva un registro de distintos proyectos de la fundación y del centro de investigación, la cual se hace de forma arcaica, solicitando los proyectos a diferentes instituciones y capturándolos de manera convencional, quedando archivados los proyectos en un mismo ordenador y solo siendo accesible por una sola persona. Este sistema permitirá poder capturar proyectos en diferentes puntos, acelerando la captura de estos mismos, y poder visualizarlos desde diferentes ordenadores para que ya no estén concentrados en un mismo espacio. También se tiene pensado que con el tiempo, con el mismo sistema, en lugar de solicitar los proyectos , sean las instituciones quienes capturen sus propios proyectos. Además el sistema estará diseñado para ser escalable.

Cualquier persona que este registrada en el sistema podrá interactuar con el sistema, pero las opciones que tendrá disponibles en el sistema dependerá de su perfil, por lo cual se agregarán perfiles a los usuarios, por ejemplo, si es administrador además de las opciones normales tendrá otras que los usuarios normales no tendrían, como ver y editar los datos de los demás usuarios.

Para la creación del sistema se utilizara una arquitectura de tres capas, es decir, un estilo de programación por el medio del cual se pretende separar la capa de presentación, la capa del dominio o negocio y la capa de persistencia de datos. A continuación se incluye una pequeña explicación de cada capa.

* Capa de presentación: Es la que se encarga de que el sistema interactúe con el usuario y viceversa, muestra el sistema al usuario, le presenta la información y obtiene la información del usuario en un mínimo de proceso. En el mundo de la computación es conocida como la Interfaz  gráfica
* Capa del dominio: Lógica del sistema, en la cual se encuentran implementados todos los requerimientos definidos por el cliente.  Aquí es donde se establecen todas las reglas que deben cumplirse.
* Capa de datos: Esta capa es la encargada de almacenar los datos del sistema y de los usuarios. Su función es almacenar y devolver datos a la capa de negocio. Se encuentra implementado el soporte de datos (Base de Datos) para el sistema.

Para crear un sistema web completo se debe tener en cuenta lo siguiente:

* El front-end (que es la parte en la que interactúa el usuario).
* El back-end (que es la parte en que resuelve las peticiones del usuario).

En la parte del front-end se utilizarán las siguientes tecnologías:

* HTML5.
* CSS3.
* JavaScript.
* Jquery.
* Bootstrap.

En la parte del back-end se utilizaran las siguientes tecnologías:

* PHP
* MYSQL
* Laravel

**2. Justificación**

Fundación Markoptic, al igual que cualquier otra organización, cuenta con diferentes departamentos, uno de ellos es el área de subdirección gestión de fondos que tiene como objetivo participar en diferentes convocatorias del país e internacionales, para ello es necesario documentar una serie de requisitos físicos y electrónicos. Es por eso que se requiere un software que ayude a las personas encargadas de integrar el proyecto para las convocatorias de los diferentes fondos a los cuales gestiona.

El desarrollo de este proyecto comprende la implementación de un sistema que logre hacer más práctica la comunicación entre el jefe inmediato de Subdirección de Fondos y el encargado de las actividades de integración de los requerimientos de cada convocatoria. El software diseñado deberá ser una herramienta para hacer más productivo el participar en convocatorias.

Llevar el control manual de las diferentes actividades puede ser difícil. Con este sistema se trata de evitar lo siguiente:

1. El congestionamiento de documentos.
2. La difícil organización de documentos.
3. Extravió de algún documento.
4. Lentitud a la hora de capturar algún proyecto.
5. Falta de comunicación a la hora de realizar alguna captura.
6. Falta de organización y control.
7. Trabajo tedioso.
8. Entre otras acciones que entorpezcan las actividades.

Por estas razones se requiere la creación de un sistema que automatice el registro y administración de los proyectos para las convocatorias de los diferentes fondos a los cuales se gestiona y con esto facilitar el trabajo del administrador del área de subdirección gestión de fondos.

Por lo tanto, el software contara con ciertas características que brindaran soluciones adecuadas. Con esto el sistema constara de las siguientes características:

1. Un software amigable.
2. Un software fácil de usar.
3. Disponibilidad.
4. Compatible con los diferentes navegadores.
5. Seguro.
6. Actual.
7. Eficiente.
8. Tiempos de respuestas favorables.
9. Fácil de mantener.
10. Capaz de responder de manera adecuada a los cambios.

**3. Objetivos**

**3.1 Objetivo General:**

Desarrollar un sistema de software para el área de gestión de fondos basado en la web para la fundación Markoptic el cual se especializará en el registro, manejo y gestión de los proyectos en los cuales participe la institución, siendo esto el principal objetivo del sistema.

El sistema debe de ser desarrollado usando tecnologías HTML5, CSS, JavaScript y PHP , así como con frameworks necesarios , con una arquitectura que se adapte a los cambios que en un futuro puedan presentarse, tomando la metodología más acorde.

**3.2 Objetivos Específicos:**

* Definir la información requerida en los proyecto.
* Creación de la base de datos donde se almacenarán los proyectos.
* Creación de un sistema de autentificación.
* Creación del sistema de registro de los proyectos
* Creación de un CRUD para el administrador en el Content Managent System.
* Creación de un modulo de búsqueda para el administrador donde pueda visualizar los proyectos registrados en el sistema.
* El sistema debe poder ser accesible a través de internet.
* Crear un módulo de retroalimentación para futuras ideas.
* Crear un módulo para ayuda en línea.
* Diseñar el sistema tomando en cuenta la posibilidad de añadir otras características en un futuro.

**4. Problemas a resolver**

Hoy en día es muy común ver como las empresas deciden adentrarse en la tecnología, adquiriendo así sistemas para sus propias exigencias, pero se debe de tener en cuenta que las necesidades de cada organización son distintas y que es muy común que las soluciones de software que ya existen en el mercado no se ajusten a las necesidades específicas de las empresas, y por lo tanto, se vean en el apuro de crear una solución de software que cumpla con sus requerimientos.

Este es el principal problema que se busca resolver en este proyecto, un sistema software que se ajuste precisamente a las necesidades que ocupe la empresa, que como se suscita anteriormente, es el registro y manejo de proyectos de las convocatorias que se hayan encontrado a nivel nacional e internacional, para así, una vez gestionados en el sistema, en este mismo se pueda la asignación del proyecto de acuerdo a la pertinencia del perfil para empezar a trabajar en el proyecto , o incluso, que los propios investigadores revisen los proyectos y ellos mismos tomen algún proyecto de su interés.

Existen otros problemas secundarios que se buscan resolver con el sistema. Que a continuación se mencionan:

* La captura simultánea de proyectos en la web, aumentando así, la rapidez de la captura de estos mismos.
* Multiplicidad de usuarios, varios usuarios conectados al sistema realizando diferente labores.
* Un manual que auxilien a los usuarios, a observar la manera correcta de llenado.
* Indicaciones que asistan a los usuarios.
* Un apartado de retroalimentación para sugerencias o mejoras al sistema.
* Visión al futuro. Que con el tiempo sean las mismas instituciones las que realicen el registro de sus proyectos.

Un tema muy delicado es la forma de poder comunicarte (vía web) con el encargado del sistema por si llegan a surgir dudas, por lo cual se pensó implementar un chat para crear un puente entre los usuarios y el encargado del sistema.

**5.- Procedimiento y descripción de las actividades realizadas.**

Las actividades realizadas durante el periodo de residencias profesionales van desde el comienzo del ciclo de vida del software iniciando con el análisis de los requisitos, hasta la implementación del mismo. Estas actividades pueden clasificarse en comunicación, análisis, modelado, construcción e implementación.

A continuación se plasmaran los términos usados en la elaboración de este proyecto de residencias que fueron necesarios saber para llevar a cabo este propósito, así como las bases teóricas sobre las que se sustenta el mismo. Se describen lenguajes, patrones, estilos arquitectónicos, etc.

**5.1 - Preparación del proyecto para dar inicio al desarrollo**

En fechas recientes se han vuelto indispensable el uso de las nuevas tecnologías ya que gracias a ellas podemos sistematizar ciertas operaciones que realiza al empresa, existiendo en el mercado numerosas herramientas, con lo cual se pueden desarrollar estas operaciones.

Con este tipo de herramientas se pueden realizar infinidad de cosas relacionadas con la informática, además de mejorar el rendimiento de trabajo. Pero antes de entrar a la informática, se deben comprender los conceptos con los cuales se trabajarán y entender el entorno en cual se desarrollara el sistema. Por consiguiente se especifican los conceptos primordiales a entender:

* **Gestión de fondos**

La gestión de fondos es un elemento vital a la hora de operar en los mercados financieros, sobre todo en épocas de volatilidad. Se trata de una técnica defensiva que te mantiene con fondos para que puedas seguir operando y contribuye a la obtención de ganancias. En resumen, te indica si tienes o no dinero suficiente para seguir operando.

Una buena gestión de fondos consiste en entender que utilizar una parte pequeña de tu capital es la forma más inteligente de empezar.

* **Gestión de riesgos**

El famoso escritor renacentista italiano Maquiavelo es conocido por la frase “el fin justifica los medios”. La gestión de riesgos consiste en manejar bien los medios para conseguir los fines. Los traders principiantes se centran en el aumento de capital y no en su conservación. A veces es mejor asumir una pérdida para evitar pérdidas mucho mayores y más catastróficas de los fondos que tanto te ha costado ganar. Debes estar preparado para perder batallas si quieres ganar la guerra.

* **Proyecto**

El término proyecto proviene del latín proiectus y cuenta con diversas significaciones. Podría definirse a un proyecto como el conjunto de las actividades que desarrolla una persona o una entidad para alcanzar un determinado objetivo. Estas actividades se encuentran interrelacionadas y se desarrollan de manera coordinada.

Lo habitual es que el objetivo perseguido por el proyecto deba ser cumplido en un cierto periodo temporal definido con anterioridad y respetando un presupuesto: de lo contrario, se dirá que el proyecto ha fracasado.

* **Productividad**

Los sistemas enfocados a evaluar la productividad de los empleados, llevan una cuantificación del tiempo y las actividades que generan resultados positivos dentro de la organización. Se miden las actividades de cada uno de los empleados. De esta manera se cuantifica el costo “real” (horas hombre) que le toma a cada uno de ellos realizar sus labores, asi como las herramientas que utilizan para alcanzar objetivos.

**5.2 - Investigación de tecnologías para desarrollar el sistema.**

Se requiere que la aplicación sea web y sea moderna, por lo tanto se utilizaran tecnologías basadas en el desarrollo web y que sean tecnologías novedosas. El sistema debe funcionar de forma similar sin importar desde que dispositivo se visualice el contenido.

Para cumplir con estos objetivos se hará uso de las tecnologías para desarrollo web que se listan a continuación.

**5.2.1 – HTML 5**

HTML5 es un lenguaje markup (de hecho, las siglas de HTML significan Hyper Text Markup Language) usado para estructurar y presentar el contenido para la web. Es uno de los aspectos fundamentales para el funcionamiento de los sitios, pero no es el primero. Es de hecho la quinta revisión del estándar que fue creado en 1990. A fines del año pasado, la W3C la recomendó para transformarse en el estándar a ser usado en el desarrollo de proyectos venideros. Por así decirlo, qué es HTML5 está relacionado también con la entrada en decadencia del viejo estándar HTML 4, que se combinaba con otros lenguajes para producir los sitios que podemos ver hoy en día. Con HTML5, tenemos otras posibilidades para explotar usando menos recursos. Con HTML5, también entra en desuso el formato XHTML, dado que ya no sería necesaria su implementación.

Se trata de un sistema para formatear el layout de nuestras páginas, así como hacer algunos ajustes a su aspecto. Con HTML5, los navegadores como Firefox, Chrome, Explorer, Safari y más pueden saber cómo mostrar una determinada página web, saber dónde están los elementos, dónde poner las imágenes, dónde ubicar el texto.

La diferencia principal de sus antecesores es el nivel de sofisticación del código que podremos construir usando HTML5.

HTML5 establece una serie de nuevos elementos y atributos que reflejan el uso típico de los sitios web modernos. Algunos de ellos son técnicamente similares a las etiquetas <div> y <span> , pero tienen un significado semántico, como por ejemplo <nav> (bloque de navegación del sitio) o <footer>(bloque de pie del sitio). Otros elementos proporcionan nuevas funcionalidades a través de una interfaz estandarizada, como los elementos <audio> y <video>. Hay otras etiquetas inauguradas por HTML5, que se pueden contemplar diversas referencias bibliográficas o virtuales.

**5.2.2 – CCS**

Las hojas de estilo en cascada (CSS por sus siglas en ingles) hacen referencia a un lenguaje de hojas de estilos usado para describir la presentación semántica (el aspecto y formato) de un documento escrito en lenguaje de marcas. Su aplicación más común es dar estilo a las páginas webs escritas en lenguaje de HTML y XHTML (García, 2009).

El organismo encargado de la estandarización al respecto es el llamado W3C que definió la primera versión CSS1 en 1996. Posteriormente se han desarrollado las revisiones 2, 2.1 y 3 que es la más actual. El lenguaje CSS seguirá evolucionando, pero hoy día puede decirse que ha sido un éxito al simplificar el mantenimiento de páginas web. Antes, los contenidos y los estilos de presentación estaban mezclados. Con CSS, quedan separados y se hace más fácil el diseño y mantenimiento de páginas web.

CSS, es una tecnología que nos permite crear páginas web de una manera más creativa, gracias a esto tenemos más control sobre el aspecto final de la pagina, pudiendo hacer muchas cosas que no se podía hacer utilizando solamente HTML, como incluir márgenes, tipos de letra, fondos, colores, efecto, animaciones, etc.

**5.2.3 – JavaScript**

JavaScript es un lenguaje de programación utilizado para crear pequeños programitas encargados de realizar acciones dentro del ámbito de una página web. Con JavaScript podemos crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones JavaScript y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso, y tal vez el único, con que cuenta este lenguaje es el propio navegador. (Gandy,2005).

Entre las acciones típicas que se pueden realizar en JavaScript tenemos dos vertientes. Por un lado los efectos especiales sobre páginas web, para crear contenidos dinámicos y elementos de la página que tengan movimiento, cambien de color o cualquier otro dinamismo. Por el otro, javaScript nos permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que podemos crear páginas interactivas con programas como calculadoras, agendas, o tablas de cálculo.

Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos en JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios.

A pesar de su nombre, JavaScript no guarda ninguna relación directa con el lenguaje de programación Java. JavaScript (abreviado comúnmente "JS") es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. JavaScript no es más que la implementación que realizó la empresa Netscape del estándar ECMAScript.

**5.2.4 – Jquery**

jQuery fue publicado por primera vez en Enero del 2006 en “BarCamp NYC” por John Resign. Soporte para AJAX fue agregado un mes después, y el modelo de licenciamiento open source del MIT fue adoptado en Mayo de ese mismo año.

jQuery es una librería JavaScript open-source, que funciona en múltiples navegadores, y que es compatible con CSS3. Su objetivo principal es hacer la programación “scripting” mucho más fácil y rápida del lado del cliente. Con jQuery se pueden producir páginas dinámicas así como animaciones parecidas a Flash en relativamente corto tiempo.

 jQuery no es un lenguaje, sino una serie de funciones y métodos de Javascript. Por tanto, Javascript es el lenguaje y jQuery es una librería que podemos usar opcionalmente si queremos facilitar nuestra vida cuando programamos en Javascript.

Con la llegada de jQuery la principal ventaja es que ya no necesitamos preocuparnos sobre si el navegador del usuario es Explorer, Chrome, Firefox, etc.

Definitivamente es una de las mejores librerías en el mundo JavaScript. Y claramente, al final del día, jQuery es como cualquier otra herramienta, que será tan buena como tan bueno sea el programador.

**5.2.5 – PHP**

PHP fue creado en 1994 por Rasmus Lerdorf para tener un conteo de cuantas personas visitaban su página. Fue creciendo hasta llegar a ser utilizado en situaciones más profesionales.

PHP es un lenguaje interpretado embebido en HTML lo que significa que se puede escribir código PHP adentro de etiquetas HTML. Por ser un lenguaje interpretado significa que no genera un código objeto, sino que se va interpretando en tiempo de ejecución en instrucciones legibles para la computadora.

Otra característica importante es que es un lenguaje del lado del servidor. Para ejecutarlo es necesario contar con un servidor web y hacerle peticiones al mismo. Existen diferentes servidores que pueden trabajar con PHP, pero el servidor por excelencia de PHP es el servidor Apache. Sin embargo también es posible ejecutarlo en otros servidores tales como IIS de Microsoft.

Que el procesamiento se haga del lado del servidor reduce la carga de la máquina cliente y además proporciona mecanismos más seguros para realizar operaciones delicadas.

Con todas sus características, PHP se convierte en un lenguaje de programación web que permite crear sitios dinámicos y realizar procesos complejos de las reglas de negocio. Su sintaxis es muy simple pero a la vez es un lenguaje poderoso e ideal.

**5.2.6 – Twitter Bootstrap**

Bootstrap fue creado por Mark Otto y Jacbod Thornton para mejorar las herramientas internas en Twitter. Antes se utilizaban muchas librerías diferentes y esto hacía el mantenimiento bastante complicado. En agosto del 2011 Twitter liberó Bootstrap como código abierto (bajo la licencia MIT) y desde febrero del 2012 se convirtió en el proyecto con más forks y favoritos de Github.

Bootstrap es un framework HTML, CSS y JavaScript que podemos utilizar como base para crear nuestros sitios o aplicaciones web.

Twitter Bootstrap es un framework para diseño de sitios y aplicaciones web de forma responsiva. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como, extensiones de JavaScript opcionales.

El framework proporciona clases Css y código JavaScript para definir el layout de la página, crear componentes que respondan a eventos y estilizar los elementos HTML más habituales.

Los principios en los que se basa el framework son:

* **Responsive Design:** Consiste en que la pagina trata de “hacer lo correcto” al ser visualizada independientemente del dispositivo y tamaño de pantalla.
* **Cross Browser**: Trata de ser compatible con la mayoría de los navegadores

**5.2.7 – Laravel – Framework PHP**

Laravel es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP 5. Su filosofía es desarrollar código PHP de forma elegante y simple, evitando el “código spaghetti”. Fue creado en 2011 por Taylor Otwell y tiene una gran influencia de frameworks como Ruby on Rails, Sinatra y ASP.NET MVC.

Laravel tiene como objetivo ser un framework que permite el uso de sintaxis elegante y expresiva para crear código de forma sencilla y permitiendo multitud de funcionalidades. Intenta aprovechar lo mejor de otros framework y aprovechar las características de las ultimas versiones de PHP (Dayle Rees,2013).

Laravel posee Rutas, Modelos, Plantillas, Vistas y Controladores, además de un motor propio para el manejo de plantillas que se denomina Blade.

Características Básicas:

* Sistema de ruteo, soporta RESTful
* Blade, Motor de plantillas
* Peticiones Fluent
* Eloquent ORM
* Basado en Composer
* Soporte para el caché
* Soporte para MVC
* Usa componentes de Symfony

**Sistemas de Gestión de Base de Datos**

Los sistemas de gestión de base de datos (en ingles database managent system, abreviado DBMS) son un tipo de software muy específico, dedicado a servir de interfaz entre las bases de datos, el usuario y las aplicaciones que la utilizan. El propósito general de los sistemas de gestión de bases de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante para una organización.

**5.2.8 Mysql**

Es un sistema de gestión de bases de datos relacional, fue creada por la empresa sueca MySQL AB, la cual tiene el copyright del código fuente del servidor SQL, asi como también de la marca.

MySQL es un software de código abierto , licenciado bajo la GPL de la GNU, aunque MySQL AB distribuye una versión comercial, en lo único que se diferencia de la versión libre, es en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de otra manera, se vulneraria la licencia GPL.

MySQL surgió alrededor de la década del 90, Michael Windenis comenzó a usar mSQL para conectar tablas usando propias rutinas de bajo nivel (ISAM). Tras unas primeras pruebas, llego a la conclusión de que mSQL no era lo bastante flexible ni rápido para lo que necesitaba, por lo que tuvo que desarrollar nuevas funciones. Esto resulto en una interfaz SQL a su base de datos, totalmente compatible a mSQL.

Las principales características de MySQL son:

* El principal objetivo de MySQL es velocidad y robustez.
* Soporta gran cantidad de tipos de datos para las columnas.
* Gran portabilidad entre sistemas, puede trabajar en distintas plataformas y sistemas operativo
* Cada base de datos cuenta con 3 archivos: Uno de estructura, uno de datos y uno de índice y soporta hasta 32 índices por tabla.
* Aprovecha la potencia de sistemas multiproceso, gracias a su implementación multihilo.
* Flexible sistema de contraseñas (passwords) y gestión de usuarios, con un muy buen nivel de seguridad de datos.
* El servidor soporta mensajes de error en distintas lenguas.

**5.3 - Análisis de Requisitos**

El análisis de requisitos es un proceso de descubrimiento, refinamiento, modelado y especificación. Se refinan en detalle los requisitos del sistema y el papel asignado al software.

Los requisitos son capacidades y condiciones con las cuales debe ser conforme el sistema - y mas ampliamente, el proyecto [JBR99]-. El primer reto del trabajo de los requisitos es encontrar, comunicar y recordar (que normalmente significa registrar) lo que se necesita realmente , de manera que tenga un significado claro para el cliente y los miembro del equipo de desarrollo.

Tanto el desarrollador como el cliente tienen un papel activo – un conjunto de actividades que son denominadas análisis – El cliente intenta replantear un sistema confuso, a nivel de descripción de datos, funciones y comportamiento, en detalles concretos. El desarrollador actúa como interrogador, como consultor, como persona que resuelve problemas y como negociador.

El análisis y la especificación de requisitos pueden parecer una tarea relativamente sencilla, pero las apariencias engañan. El contenido de comunicación es muy denso. Abundan las ocasiones para malas interpretaciones o falta de información. Es muy probable que haya ambigüedad. El dilema al que se enfrenta el ingeniero de software puede entenderse muy bien repitiendo la famosa frase de un cliente anónimo: “Sé que cree que entendió lo que piensa que dije, pero no estoy seguro de que se dé cuenta de que lo que escuchó no es lo que yo quise decir”.

El análisis de requerimientos permite al ingeniero de sistemas especificar las características operacionales del software (función, datos y rendimientos), indica la interfaz del software con otros elementos del sistema y establece las restricciones que debe cumplir el software.

**5.3.1 – Técnicas de Investigación**

El ingeniero en sistemas debe poseer habilidades particulares para facilitar la comunicación con el cliente y ganar su confianza.

El método de investigación empleado en este trabajo corresponde a un estudio de caso basado en diversas técnicas de investigación que son: la entrevista, las observaciones y la revisión de registros.

Como primer paso se llevo a cabo un estudio de caso. Robert K. Yin establece en su libro ***Case Study Research*** un caso de uso como: “la estrategia preferida cuando como o porque son preguntas dadas, cuando el investigador tiene un pequeño control sobre los eventos”.

De acuerdo con Babbie (1999) la entrevista se da a través de encuentros personales, siendo este tipo uno de los más prácticos y aceptados por la gente. Por lo tanto, una entrevista es una relación, mediante la cual uno de los integrantes trata de obtener datos.

Para el estudio del caso de este proyecto se tomo en cuenta para su complemento la forma descriptiva, de la cual se obtuvieron los datos necesarios y suficientes para dicho entendimiento del problema.

Las técnicas de investigación empleadas en dicho estudio fueron:

1. Como primar paso se realizo una entrevista a la jefa inmediata. Dicha entrevista se llevo a cabo con la finalidad de obtener información del software que se estaba solicitando así como la operatividad de este mismo
2. Posteriormente se llevo a cabo una observación de cómo se realizaban las actividades solicitadas para tener un mejor entendimiento de estas mismas.
3. Por último, la última tarea realizada fue la revisión de registros que me permitió conocer ciertos procedimientos y técnicas necesarias para poder empezar a realizar los requisitos y ver como seria su funcionalidad.

El instrumento de medición que se utilizo en este proyecto fue la entrevista de tipo libre. Este tipo de entrevista permita llevar en la mente lo que se desea saber, sin que se tenga que seguir un orden riguroso.

**5.3.2 – Casos de usos**

Un caso de uso es un conjunto de escenarios que tienen una meta de usuario en común (Martin Fowler). Un escenario es una secuencia de acciones e interacciones (pasos) entre los usuarios (actores) y el sistema. Un actor representa el rol jugado por una persona o cosa que actúa con el sistema.

Los casos de uso no son artefactos orientados a objetos - son simplemente historias escritas-.Sin embargo, son una herramienta muy popular en análisis de requisitos y son una parte importante del Proceso Unificado.

De acuerdo a la información recabada con las técnicas de investigación mencionadas anteriormente, se obtuvieron los siguientes casos de uso:

¿Quiénes Interactúan con el Sistema?

|  |  |
| --- | --- |
| **Nombre:** | Investigador |
| **Descripción:**  Representa al usuario de nivel bajo. Generalmente este tipo de usuario puede revisar los proyectos, puede enviar mensajes al administrador para sugerencias o retroalimentación y puede platicar vía online con el administrador del sistema, para aclarar cualquier duda. Además puede actualizar su información, así como cambiar su contraseña. | |

|  |  |
| --- | --- |
| **Nombre:** | Usuario |
| **Descripción:**  Descripción: Representa al usuario intermedio. Generalmente este tipo de usuario serán los que ayuden al administrador con algunas tareas que se realizarán en el sistema, como la captura de proyectos o la modificación de estos mismos. Además tiene los beneficios que tiene el usuario investigador. | |

|  |  |
| --- | --- |
| **Nombre:** | Administrador |
| **Descripción:**  Representa al súper usuario. Puede realizar lo mencionado anteriormente por el investigador y el usuario normal. Además puede acceder al modulo de CMS (manejador de contenido del sistema), donde puede agregar nuevos usuarios para poder entrar al sistema, modificar la información de estos mismos y eliminar a usuarios. También se le atribuye la opción de poder eliminar un proyecto y consultar los proyectos de una manera distinta que los investigadores. | |

Que debe de hacer el sistema?

CU01 – Login (Escenario).

El usuario introduce su correo electrónico y su contraseña. El sistema verifica la validez del correo electrónico y de la contraseña en la base de datos y le permite al usuario el acceso al sistema. El sistema muestra una página de bienvenida y esta la opción para ir al dashboard.

|  |  |
| --- | --- |
| **Nombre:** | Acceder al sistema |
| **Autor:** | Daniel Paredes |
| **Fecha:** | 16/02/2015 |
| **Descripción:**  El usuario introduce su correo electrónico y su contraseña. El sistema verifica la validez del correo electrónico y de la contraseña en la base de datos y le permite al usuario el acceso al sistema. | |
| **Actores:**  Investigador, Usuario y Administrador. | |
| **Pre-condiciones:**  El usuario debe de estar dado de alta en el sistema. | |
| **Flujo Normal:**  1.-El usuario accede al sistema en la url necesaria.  2.-El sistema solicita las credenciales.  3.-El usuario ingresa sus datos y confirma.  4.-El sistema comprueba la validez de los datos del usuario y le da la bienvenida. | |
| **Flujo Alternativo 1:**  4.-El sistema comprueba la validez de los datos, si los datos son incorrectos, se avisa al usuario de ello permitiéndolos que los corrija. | |
| **Flujo Alternativo 2:**  3.-El usuario no recuerda su password por lo que puede solicitar el caso de uso **Recuperar Contraseña**  **Excepciones.**  FA2. – El correo electrónico no está registrado en el sistema. El sistema notifica el error. | |
| **Post-condiciones:**  El usuario accede al sistema y puede realizar alguna tarea en específica. | |

CU02- Recuperar Contraseña (Escenario).

El usuario accede a la sección Recuperar Contraseña, teniendo que seguir unos pasos para poder recuperar su contraseña. El usuario introduce su correo electrónico. El sistema verifica la validez del correo electrónico y permite al usuario iniciar los pasos para la recuperación de su contraseña.

|  |  |
| --- | --- |
| **Nombre:** | Recuperar contraseña |
| **Autor:** | Daniel Paredes |
| **Fecha:** | 16/02/2015 |
| **Descripción:**  El usuario accede a la sección Recuperar Contraseña, teniendo que seguir unos pasos para poder recuperar su contraseña. El usuario introduce su correo electrónico. El sistema verifica la validez del correo electrónico y permite al usuario iniciar los pasos para la recuperación de su contraseña. | |
| **Actores:**  Investigador, usuario y Administrador | |
| **Precondiciones:**  Que el correo electrónico exista en el sistema. | |
| **Flujo Normal:**  1.-El usuario accede al apartado recuperar contraseña.  2.-El sistema muestra un formulario solicitando el correo electrónico.  3.-El usuario ingresa el correo electrónico correspondiente.  4.-El sistema comprueba la validez correo electrónico en la base de datos y envía un correo al correo electrónico dado.  5.-El usuario ingresa a su correo electrónico y revisa el email enviado a su correo por parte del sistema, verifica la información proporciona y entra al link proporcionado en el email.  6.-El sistema muestra un formulario donde puede cambiar su contraseña y debe repetir la contraseña dada.  7.-El usuario ingresa sus datos y confirma  8.-El sistema comprueba la validez de los datos y lo redirige al login. | |
| **Flujo Alternativo:**  4.-El sistema comprueba la validez del correo electrónico, si el correo no existe en la base de datos, se avisa al actor de ello permitiéndolo que lo corrija.  5.-Si el usuario tarda más de 60 minutos en verificar el correo y entrar al link proporcionado, este expira.  8.-El sistema comprueba que la contraseña ingresada sea la misma que el campo repetir contraseña. | |
| **Post-condiciones:**  La contraseña fue almacenada en la base de datos y puede proseguir con el CU01. | |

CU03 - Consultar información (Escenario).

El usuario puede acceder a su información desde cualquier parte del sistema. Para esto debe acceder a la sección Mi perfil y el seleccionar la opción Ver Información. La página de consulta se carga en el browser y sale la información correspondiente a la petición.

|  |  |
| --- | --- |
| **Nombre:** | Consultar información |
| **Autor:** | Daniel Paredes |
| **Fecha:** | 16/02/2015 |
| **Descripción:**  Permite al actor poder acceder a su información desde cualquier parte del sistema. La ventana de consulta se carga en el browser y sale la información correspondiente a la petición, para así poder ser consultada. | |
| **Actores:**  Investigador, Usuario y Administrador | |
| **Precondiciones:**  Estar logueado en el sistema. | |
| **Flujo Normal:**  1-El usuario escoge el apartado ‘Mi Perfil’.  2-El sistema despliega un menú con varias opciones.  3-El usuario selecciona ‘Ver Información’.  4-El sistema carga en el browser la ventana de consulta solicitada. | |
| **Flujo Alternativo 1:**  4.-El actor cancela la operación y el sistema aborta el proceso. | |
| **Flujo Alternativo 2:**  4b- El usuario invoca el caso de uso **Modificar Información. (CU04)** | |
| **Post-condiciones:**  Permite al actor saber que tiene toda su información de manera correcta. | |

CU04- Modificar información (Escenario).

El usuario puede modificar su información desde cualquier parte del sistema. Para esto debe de entrar a Mi Perfil y seleccionar la opción Editar información, teniendo como resultado una ventana de modificación de sus datos personales y en esa misma ventana poder escoger cambiar su contraseña por una contraseña más personalizada, abriendo la solicitud solicitada en otra página.

|  |  |
| --- | --- |
| **Nombre:** | Modificar Información |
| **Autor:** | Daniel Paredes |
| **Fecha:** | 16/02/2015 |
| **Descripción:**  Permite al actor poder modificar su información desde cualquier parte del sistema. La ventana de modificación se carga en el browser y sale la información correspondiente a la petición, para así poder ser modificada. | |
| **Actores:**  Investigador, Usuario y Administrador | |
| **Precondiciones:**  Estar logueado en el sistema, y antes se debió de ver ejecutado el caso de uso **Consultar Información (CU03).** | |
| **Flujo Normal:**  1-El usuario selecciona editar información.  2-El sistema hace una consulta a la base de datos y regresa la información solicitada.  3-El usuario modifica los datos que requiera necesarios.  4-El sistema comprueba la validez de los datos proporcionados en los diferentes campos y los guarda en el sistema. | |
| **Flujo Alternativo:**  4-El sistema comprueba la validez de los datos proporcionados, si los datos son incorrectos, se avisa al actor de ellos permitiéndolos que los corrija. | |
| **Post-condiciones:**  Permite al actor tener toda su información actualizada. | |

CU05- Agregar Proyecto (Escenario).

Permite a cualquier actor agregar un proyecto nuevo al sistema. Para asi quede almacenada en el sistema.

|  |  |
| --- | --- |
| **Nombre:** | Agregar Proyecto |
| **Autor:** | Daniel Paredes |
| **Fecha:** | 16/02/2015 |
| **Descripción:**  Permite a cualquier actor agregar un proyecto nuevo al sistema. Para asi quede almacenada en el sistema. | |
| **Actores:**  Usuario y Administrador | |
| **Precondiciones:**  Estar logueado en el Sistema. | |
| **Flujo Normal:**  1-El usuario, en la sección de proyectos, selecciona Agregar proyecto.  2-El sistema regresa una página que se carga en el browser, con un formulario.  3-El usuario ingresa los datos correspondientes en el formulario, una vez que el formulario esta completo, confirma en el botón de enviar.  4.- El sistema comprueba la validez de los datos proporcionados en los diferentes campos y permite seguir con el proceso de agregar un proyecto.  5- Se repiten los pasos 2, 3 y 4 hasta llegar al final del proceso.  6-El sistema muestra un mensaje de que el proceso de agregar un proyecto ha sido finalizado y lo regresa a la sección de proyectos. | |
| **Flujo Alternativo:**  4.- El sistema comprueba la validez de los datos proporcionados por el usuario, si los datos son incorrectos, se avisa al actor de ellos permitiéndolos que los corrija.  5.-No se puede saltar ningún paso, por lo tanto si intenta saltarse un formulario. Este no lo permitirá. | |
| **Post-condiciones:**  Se agrega un proyecto nuevo al sistema. | |

CU06- Consultar Proyectos (Escenario).

Permitir que el usuario o administrador pueda ver de una manera rápida cuantos proyectos se encuentran en el sistema, así como el status actual del proyecto y alguna otra información de estos mismos.

|  |  |
| --- | --- |
| **Nombre:** | Consultar Proyectos |
| **Autor:** | Daniel Paredes |
| **Fecha:** | 16/02/2015 |
| **Descripción:**  Permitir que el usuario o administrador pueda ver de una manera rápida cuantos proyectos se encuentran en el sistema, así como el status actual del proyecto y alguna otra información de estos mismos. | |
| **Actores:**  Investigador, Usuario y Administrador | |
| **Precondiciones:**  Estar logueado en el sistema y tener acceso. | |
| **Flujo Normal:**  1.-El usuario, en la sección de proyectos, selecciona Ver proyecto.  2.-El sistema carga una ventana en el browser, con una tabla donde se muestran los diversos proyectos existentes con los datos correspondientes del proyecto.  3.-El usuario decide ordenar los proyectos por folio.  4.-El sistema muestra los proyectos ordenados en la tabla. | |
| **Flujo Alternativo:**  5.-El usuario puede invocar el caso de uso **Modificar Proyecto (CU07)** o el caso de uso **Buscar Proyecto (CU08).** | |
| **Post-condiciones:**  Permite al usuario o administrador ver los proyectos existentes en el sistema, y ser capaz de efectuar el caso de uso Modificar Proyecto o Buscar Proyecto. | |

CU07-Modificar Proyecto (Escenario).

Permite al actor editar la información de un proyecto existente en el sistema.

|  |  |
| --- | --- |
| **Nombre:** | Modificar Proyecto |
| **Autor:** | Daniel Paredes |
| **Fecha:** | 16/02/2015 |
| **Descripción:**  Permite al actor editar la información de un proyecto existente en el sistema. | |
| **Actores:**  Usuario y Administrador | |
| **Precondiciones:**  Estar logueado en el sistema , tener acceso y antes se debió de ver ejecutado el caso de uso **Consultar Proyectos(CU06).** | |
| **Flujo Normal:**  1.-El usuario ingresa a la sección de proyectos , en el apartado de ver proyectos  2.-El sistema carga una ventana en el browser, con una tabla donde se muestran los diversos proyectos existentes con los datos correspondientes del proyecto.  3.- El usuario selecciona el proyecto a editar.  4.-El sistema muestra un formulario con los respectiva información del proyecto seleccionado.  5.-El Usuario actualiza el formulario con los datos correspondientes, una vez que el formulario esta completo, el usuario da click en el botón Guardar  6.-El sistema comprueba la validez de los datos y se guarda la nueva información. | |
| **Flujo Alternativo:**  1.-Debe ser un actor con perfil necesario para tener acceso de lo contrario no se le permitirá entrar a esta sección.  3.-El usuario cancela la operación y vuelve a la ventana anterior.  5.- El sistema comprueba la validez de los datos proporcionados por el usuario, si los datos son incorrecto, se avisa al actor de ellos permitiéndolos que los corrija. | |
| **Post-condiciones:**  El proyecto ha sido modificado correctamente. | |

CU08-Buscar Proyecto (Escenario).

Permite a cualquier actor poder buscar un proyecto en el sistema de una manera fácil y rápida.

|  |  |
| --- | --- |
| **Nombre:** | Buscar Proyecto |
| **Autor:** | Daniel Paredes |
| **Fecha:** | 16/02/2015 |
| **Descripción:**  Permite a cualquier actor poder buscar un proyecto en el sistema de una manera fácil y rápida. | |
| **Actores:**  Investigador, Usuario y Administrador | |
| **Precondiciones:**  Estar logueado en el sistema , tener acceso, que exista un proyecto en el sistema y que antes se debió de ver ejecutado el caso de uso **Consultar Proyectos(CU06).** | |
| **Flujo Normal:**  1.-El usuario, en la sección de proyectos, selecciona Buscar proyecto.  2.-El sistema muestra una página en el browser, con una caja de búsqueda y todos los proyectos existentes en el sistema  3.- El usuario, en la caja de búsqueda, puede buscar algun proyectos.  4.-El sistema regresa el proyecto a buscar.  5.-El usuario puede ser capaz de ver ese proyecto y todo lo que conlleva este mismo  6.-Se puede repetir el paso 3,4 y 5. | |
| **Flujo Alternativo:**  2.-En caso de no haber proyectos en el sistema, no se muestra ninguno.  4.-En caso de no existe el proyecto en la base de datos, no se regresa ningún proyecto.  5.-Si el usuario decide no ver el proyecto, no se abre el pdf. | |
| **Post-condiciones:**  El usuario puede observar con lujo de detalle todas las características del proyecto. Puede leerlo online o descargarlo. | |

CU09-Eliminar Proyecto (Escenario).

Permite al administrador eliminar un proyecto.

|  |  |
| --- | --- |
| **Nombre:** | Eliminar Proyecto |
| **Autor:** | Daniel Paredes |
| **Fecha:** | 16/02/2015 |
| **Descripción:**  Permite al administrador eliminar un proyecto. | |
| **Actores:**  Administrador | |
| **Precondiciones:**  Estar logueado en el sistema y tener acceso apartado eliminar proyecto | |
| **Flujo Normal:**  1.-El usuario, dentro de la sección proyectos, escoge eliminar proyecto.  2.-El sistema carga una ventana en el browser y carga todos los proyectos disponibles.  3.-El usuario, en la caja de búsqueda, escribe el nombre del proyecto o folio a buscar.  4.-El sistema regresa el proyecto escogido. | |
| **Flujo Alternativo:**  1-Debe ser un actor con perfil de administrador de lo contrario no se le permitirá entrar a esta sección.  2-el actor cancela la operación y vuelve a la ventana anterior. | |
| **Post-condiciones:**  Se elimina el proyecto del sistema. | |

CU10- Chat en Línea (Escenario).

Tener la facilidad de iniciar un chat en vivo con el administrador del sistema para poder aclarar dudas o preguntar cosas en específico respecto a un tema.

|  |  |
| --- | --- |
| **Nombre:** | Chat en Línea |
| **Autor:** | Daniel Paredes |
| **Fecha:** | 16/02/2015 |
| **Descripción:**  Tener la facilidad de iniciar un chat en vivo con el administrador del sistema para poder aclarar dudas o preguntar cosas en específico respecto a un tema. | |
| **Actores:**  Investigador, Usuario y Administrador | |
| **Precondiciones:**  Estar logueado en el sistema | |
| **Flujo Normal:**  1-El usuario elige en el dashboard la opción ayuda en línea  2.-El sistema regresa una página que se carga en el browser , con el status del chat y un apartado donde se ingresa el nombre del interesado para iniciar el chat  3-El usuario ingresa su nombre y confirma iniciar chat.  4.-El sistema comprueba la validez del nombre del usuario y lo redirige a otra página donde muestra el chat.  5-El usuario escribe su duda o pregunta.  6-El administrador responde la duda o pregunta.  7-Se repiten los pasos 5 y 6, hasta que se termine la conversación.  8-El usuario cierra la página del chat y continua con la navegación normal dentro del sistema | |
| **Flujo Alternativo 1 :**  3.-El sistema comprueba la validez del nombre proporcionado, si el nombre no cumple con los requisitos, se avisa al actor de ello permitiéndole que corrija los errores.  6.-El administrador no se encuentra disponible en ese momento. | |
| **Flujo Alternativo 2:**  2.-El status del chat es offline , por lo tanto el usuario no puede confirmar para iniciar chat. | |
| **Post-condiciones:**  El actor pudo aclarar todas sus dudas y puede continuar con el proceso que estaba realizando. | |

CU11- Correos de Retroalimentación (Escenario).

Cualquier actor logueado en el sistema puede expresar sus ideas u opiniones para mejorar el sistema o reportar algún problema en este mismo para que el desarrollador trabaje sobre esos problemas.

|  |  |
| --- | --- |
| **Nombre:** | Correos de Retroalimentación |
| **Autor:** | Daniel Paredes |
| **Fecha:** | 16/02/2015 |
| **Descripción:**  Cualquier actor logueado en el sistema puede expresar sus ideas u opiniones para mejorar el sistema o reportar algún problema en este mismo para que el desarrollador trabaje sobre esos problemas. | |
| **Actores:**  Investigador, Usuario y Administrador | |
| **Precondiciones:**  Estar logueado en el sistema. | |
| **Flujo Normal:**  1- El usuario ingresa a la sección de sugerencias.  2.-El sistema carga una página con su respectivo formulario.  3.-El usuario ingresa los datos correspondientes en el formulario, , una vez que el formulario esta completo, confirma en el botón de enviar.  4.-El sistema comprueba la validez de los datos del usuario y se manda el correo. | |
| **Flujo Alternativo:**  4-El sistema comprueba la validez de los datos proporcionados por el usuario, si los datos son incorrecto, se avisa al actor de ellos permitiéndolos que los corrija. | |
| **Post-condiciones:**  Se envía un correo al desarrollador de manera satisfactoria. | |

CU12- Consultar Usuarios (Escenario).  
Permitir al administrador que pueda ver de una manera rápida cuantos usuarios se encuentran dados de alta en el sistema, así como su información personal.

|  |  |
| --- | --- |
| **Nombre:** | Consultar Usuarios |
| **Autor:** | Daniel Paredes |
| **Fecha:** | 16/02/2015 |
| **Descripción:**  Permitir al administrador que pueda ver de una manera rápida cuantos usuarios se encuentran dados de alta en el sistema, así como su información personal. | |
| **Actores:**  Administrador | |
| **Precondiciones:**  Estar logueado en el sistema y tener acceso. | |
| **Flujo Normal:**  1.-El usuario ingresa a la sección de content managent system  2.-El sistema carga una ventana en el browser, con una tabla donde se muestran los diversos usuarios existentes con los datos correspondientes del usuario. | |
| **Flujo Alternativo 1:**  2.-El sistema no devuelve ningún usuario. | |
| **Flujo Alternativo 2:**  3a.- El administrador puede invocar el caso de uso **Modificar Usuario (CU14).**  3b.- El administrador puede Invocar el caso de uso **Eliminar Usuario (CU15).**  3c.- El administrador puede invocar el caso de uso **Buscar Usuario (CU16).** | |
| **Post-condiciones:**  Permite al usuario ver a los usuarios actuales, y ser capaz de efectuar los casos de usos Modificar Usuario (CU14) y Eliminar Usuario (CU15). | |

CU13-Agregar Usuario (Escenario).

Pedirle toda la información a la persona para darlo de alta en el sistema y así la persona pueda empezar a usar los servicios de este mismo.

|  |  |
| --- | --- |
| **Nombre:** | Agregar Usuario |
| **Autor:** | Daniel Paredes |
| **Fecha:** | 16/02/2015 |
| **Descripción:**  Pedirle toda la información a la persona para darlo de alta en el sistema y así la persona pueda empezar a usar los servicios de este mismo. | |
| **Actores:**  Administrador | |
| **Precondiciones:**  Estar logueado en el sistema y tener acceso al content managent system. | |
| **Flujo Normal:**  1.-El usuario ingresa a la sección de content managent system y selecciona la opción “Agregar Usuario”, para ingresar los datos deseados.  2.-El sistema carga una ventana en el browser, y aparece un formulario con varios campos correspondientes según la vista seleccionada.  3.-El usuario llena el formulario con los datos, una vez que el formulario esta completo, el usuario da click en el botón Crear Usuario  4.-El sistema comprueba la validez de los datos y se crea el nuevo usuario. | |
| **Flujo Alternativo:**  1.-Debe ser un actor perfil administrador de lo contrario no se le permitirá entrar a esta sección.  2.- El usuario cancela la operación y vuelve a la ventana anterior.  4a.- El sistema comprueba la validez de los datos proporcionados por el usuario, si los datos son incorrectos, se avisa al actor de ellos permitiéndolos que los corrija.  4b.-Se verifica que el usuario que se intenta agregar no esté ya en la base de datos, con el fin de evitar duplicidad de usuarios. | |
| **Post-condiciones:**  El usuario ya está dado de alta y puede acceder al sistema. | |

CU14-Modificar Usuario (Escenario).

Permite al administrador editar la información de cualquier usuario dado de alta en el sistema.

|  |  |
| --- | --- |
| **Nombre:** | Modificar Usuario |
| **Autor:** | Daniel Paredes |
| **Fecha:** | 16/02/2015 |
| **Descripción:**  Permite al administrador editar la información de cualquier usuario dado de alta en el sistema. | |
| **Actores:**  Administrador | |
| **Precondiciones:**  Estar logueado en el sistema, tener acceso al content managent system y antes se debió de ver ejecutado el caso de uso **Consultar Usuarios (CU12).** | |
| **Flujo Normal:**  1.-El usuario ingresa a la sección de content managent system y selecciona el usuario a editar.  2.-El sistema carga una ventana en el browser, y aparece un formulario con varios campos correspondientes a la información de ese usuario seleccionado.  3.-El usuario actualiza el formulario con los datos correspondientes, una vez que el formulario esta completo, el usuario da click en el botón Guardar  4.-El sistema comprueba la validez de los datos y se guarda la nueva información. | |
| **Flujo Alternativo:**  1.-Debe ser un actor perfil administrador de lo contrario no se le permitirá entrar a esta sección.  2.-El usuario cancela la operación y vuelve a la ventana anterior.  4a.- El sistema comprueba la validez de los datos proporcionados por el usuario, si los datos son incorrecto, se avisa al actor de ellos permitiéndolos que los corrija.  4b.-En caso de que se modifique el correo electrónico, se verifica que no exista en la base de datos | |
| **Post-condiciones:**  El usuario ha sido modificado. | |

Con esto evitando la duplicidad de usuarios y que los usuarios registrados, puedan recuperar su cuenta)

CU15-Eliminar Usuario (Escenario).

Permite al administrador eliminar un usuario en caso de que este haya sido de baja o por otras razones propias de la empresa.

|  |  |
| --- | --- |
| **Nombre:** | Eliminar Usuario |
| **Autor:** | Daniel Paredes |
| **Fecha:** | 16/02/2015 |
| **Descripción:**  Permite al administrador eliminar un usuario en caso de que este haya sido de baja o por otras razones propias de la empresa. | |
| **Actores:**  Administrador | |
| **Precondiciones:**  Estar logueado en el sistema, tener acceso al content managent system y antes se debió de ver ejecutado el caso de uso **Consultar Usuarios (CU12).** | |
| **Flujo Normal:**  1.-El usuario ingresa a la sección de content managent system y selecciona el usuario a eliminar.  2.-El sistema carga una ventana en el browser, y aparece un aviso respecto a la eliminación.  3.-El usuario selecciona eliminar.  4.-El sistema elimina al usuario y regresa a la ventana con los usuarios actualizados. | |
| **Flujo Alternativo:**  1-Debe ser un actor perfil administrador de lo contrario no se le permitirá entrar a esta sección.  2-el actor cancela la operación y vuelve a la ventana anterior. | |
| **Post-condiciones:**  Se elimina al usuario del sistema y este ya no puede acceder al sistema. | |

CU16-Buscar Usuario (Escenario).

Permite al administrador tener la manera de buscar a los usuarios mediante una caja de búsqueda.

|  |  |
| --- | --- |
| **Nombre:** | Buscar Usuario |
| **Autor:** | Daniel Paredes |
| **Fecha:** | 16/02/2015 |
| **Descripción:**  Permite al administrador tener la manera de buscar a los usuarios mediante una caja de búsqueda. | |
| **Actores:**  Administrador | |
| **Pre-condiciones:**  Estar logueado en el sistema, tener acceso al content managent system y antes se debió de ver ejecutado el caso de uso **Consultar Usuarios (CU12).** | |
| **Flujo Normal:**  1.-El usuario ingresa a la sección de content managent system.  2.-El sistema carga una ventana en el browser y carga todos los usuarios disponibles.  3.-El usuario, en la caja de búsqueda, escribe el nombre del usuario a buscar.  4.-El sistema regresa el usuario escogido. | |
| **Flujo Alternativo:**  1-Debe ser un actor perfil administrador de lo contrario no se le permitirá entrar a esta sección.  4.-El sistema no regresa nada debido a que no existe dicho usuario. | |
| **Post-condiciones:**  Permite al administrador manipular a dicho usuario, siendo capaz de efectuar el caso de uso Modificar Usuario o Eliminar Usuario. | |

**5.3.3 – Requisitos de calidad**

1. El sistema debe de asegurar la integridad de la información.
2. El sistema debe estar en línea las 24 horas.
3. El sistema debe ser fácil de mantener.
4. El sistema debe ser fácil de extender y mejorar.
5. El sistema debe de contar con una interfaz amigable
6. El texto del sistema debe ser de capaz de verse a una distancia considerable.
7. El tiempo de respuesta deberá ser corto
8. No usar colores de fondo que no contraste con el texto.

**5.4 – Arquitectura de Software**

“La Arquitectura de Software es, a grandes rasgos, una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes según se la percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar el objetivo del sistema. La vista arquitectónica es una vista abstracta, aportando el más alto nivel de compresión y la supresión o diferimiento del detalle inherente a la mayor parte de las abstracciones" (Paul Clements, 1996).

Este punto es uno de los más importantes al construir una aplicación, pues de ello depende la mantenibilidad y flexibilidad al crecimiento.

La arquitectura elegida para el sistema es una mezcla entre la arquitectura de Cliente – Servidor y la arquitectura en Capas.

**5.4.1 – Cliente – Servidor**

La arquitectura cliente-servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores , y los demandantes, llamados clientes. Un ciente realiza peticiones a otro programa, el servidor, quien le da respuesta.

En esta arquitectura la capacidad de proceso esta repartida entre los clientes y los servidores, aunque son mas importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema, esta es una parte muy importante dentro de nuestro sistema, ya que será el servidor donde se mantendrá a la información referente a los proyectos, manteniéndola disponible para que los “clientes” puedan tener acceso a ella.

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola maquina ni es necesariamente un solo programa (figura x).

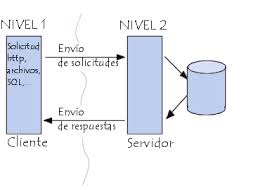


Figura x.

**5.4.2 - Arquitectura en Capas**

Garlan y Shaw definen el estilo en capas como una organización jerarquica tal que cada capa proporciona servicios a la capa inmediatamente superior y recibe servicios que le brinda la inferior.

En términos más simples, se dividen las funcionalidades y responsabilidades del sistema en distintas capas y cada una se encarga de atender una responsabilidad específica comunicándose solo con su capa adyacente, pero sin conocer mas alla de las capas superior e inferior.

El estilo de capas es uno de los mas comunes y existen una gran cantidad de variantes de esta arquitectura, pero la mas común es la que separa el sistema en una capa de presentación, una capa de reglas de negocio y una capa de servicios de datos, se le conoce como patrón Modelo-Vista-Controlador.

En el patrón de Modelo-Vista-Controlador(MVC) se separa el modelado del dominio, la presentación y acciones basadas en datos ingresados por el usuario en tres clases diferentes (Carlos Reynoso,2004).

* **Modelo**: En él se encapsulan las reglas de negocio, el comportamiento de los datos y la manera en el que las entidades se encargan de suplir los requerimientos de la aplicación.
* **Vista:** Es la interfaz en la que el usuario final puede ver representados los datos de la aplicación.
* **Controlador:** Es el intermediario entre el usuario y la aplicación, y entre el model y la vista. Es el controlador el encargado de servirle la vista al usuario y de comunicar la vista con el modelo

Esta arquitectura se esta volviendo muy popular en los desarrollos web. Actualmente muchos de los frameworks de programación web están basados en esta arquitectura (Figura y).

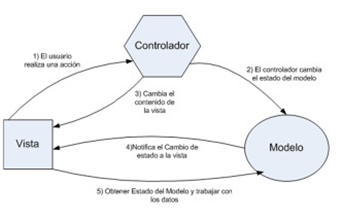


Figura x. Arquitectura en Capas

**5.5 – Investigación de diseños actuales**

“Si los productos se diseñan para encajar mejor en las tendencias naturales del comportamiento humano, entonces la gente estará más satisfecha, más completa y será mas productiva. “ (susan Weinschenk)

Se pueden considerar los siguientes (alguno o todos) como aspectos de diseño de una aplicación web.

* **Diseño de la interfaz.**

Describe la estructura y organización de la interface de usuario. Incluye una representación de diseño de pantallas. Debe de ofrecer al usuario final una experiencia satisfactoria y gratificante.

* **Diseño de estética (también llamada diseño grafico).**

Describe el “look and feeling” del sistema. Incluye diseño geométrico, tamaño del texto, uso de graficas y decisiones de estética.

Sin él una Aplicación web puede ser funcional pero no atractiva.

* **Diseño de contenido.**

Define el diseño para el contenido que es representado como parte del sistema.

* **Diseño de navegación.**

Representa el flujo de navegación. Es el acceso al contenido y las funciones del sistema.

* **Diseño arquitectónico.**

Identifica la estructura global para el sistema.

Forma en que los objetos de contenido se estructuran para su presentación y navegación y así se pueda gestionar la interacción del usuario.

* **Diseño de componentes**

Desarrolla la lógica de procesamiento detallado necesario para implementar los componentes funcionales que implementan una función del sistema completo.

De los cuales algunos aspectos de diseño se utilizarán en el proyecto de residencias y se explicarán más adelante.

**6 – Resultados, planos, gráficos, prototipos, maquetas y programas.**

**6.1 Diagrama de la base de datos del sistema.**

A continuación se muestra el diagrama de la estructura de la base de datos (figura x) y se hace una descripción de las relaciones y el funcionamiento de cada uno de los componentes que la conforman.

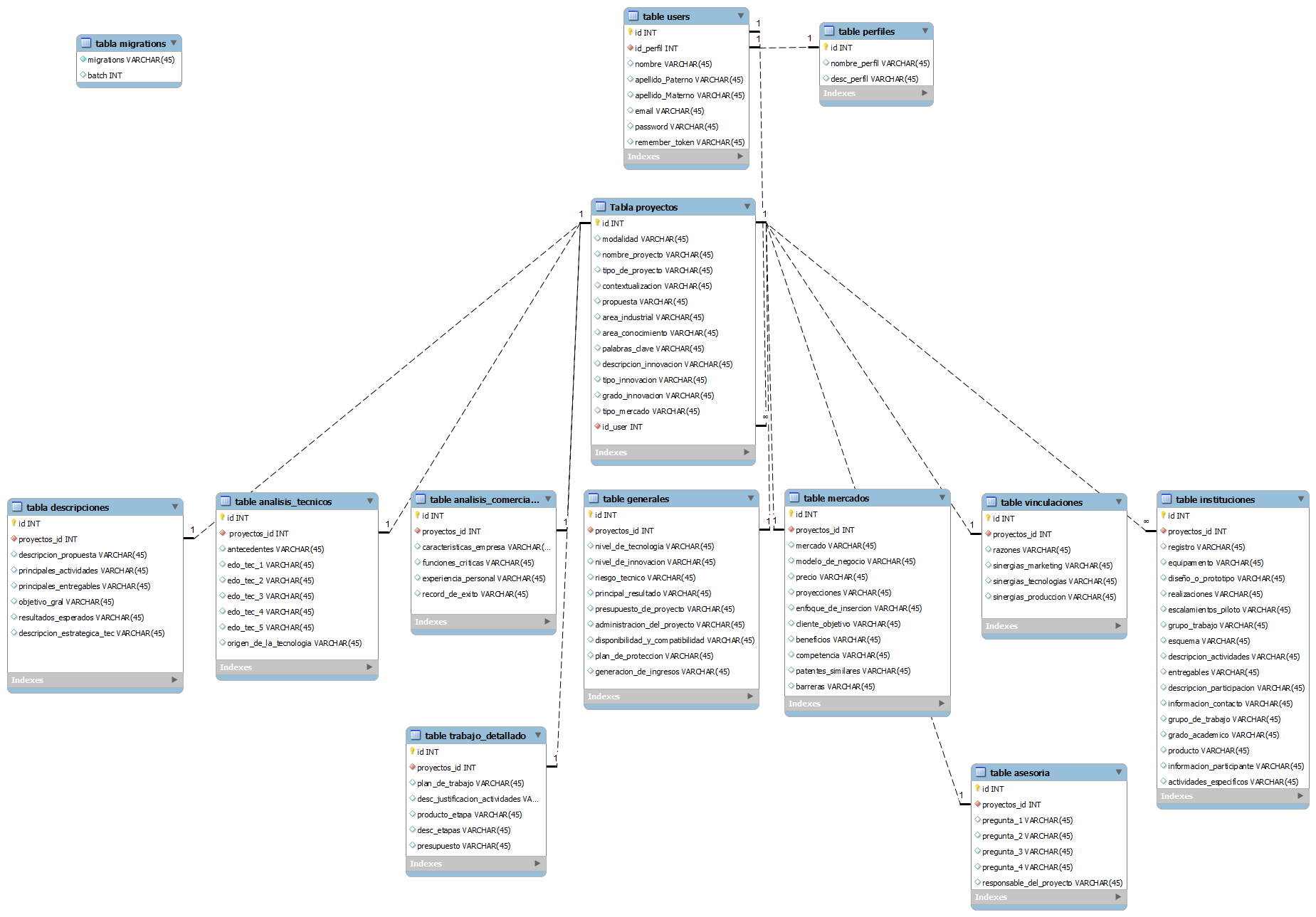


Figura x.

En el diagrama de la base de datos del sistema se pueden observar las distintas tablas que componen la estructura de la base de datos (Para un mejor visualización, se anexa la imagen dentro del disco para que se pueda ver con detalle).

Cada de las tablas tienen su función, pero las tablas relevantes son: table user y table proyectos.

* **Table user**: En esta tabla se almacena la información básica de los usuarios, y aquí mismo se permite verificar el inicio de sesión, tiene relación con la tabla perfiles para así poder identificar el perfil que tiene cada usuario y así se pueda validar los accesos dentro del sistema. Uno de los pilares del sistema, porque esta tabla decide quien tiene acceso al sistema y quien.
* **Table proyectos**: En esta tabla se almacena la información básica del proyecto para poder identificar cada proyecto, además es la columna principal a la hora de formar un proyecto pues tiene varias relaciones con distintas tablas dentro de la base de datos y que juntas nos permiten observar un proyecto completo

**6.2 – Diseño de interfaz**

A continuación se muestran una serie de imágenes donde se muestran los diseños que se planearon para la interfaz del sistema de gestión de proyectos, se busca que mediante un buen diseño de la interfaz la aplicación sea fácil de entender y de usar.

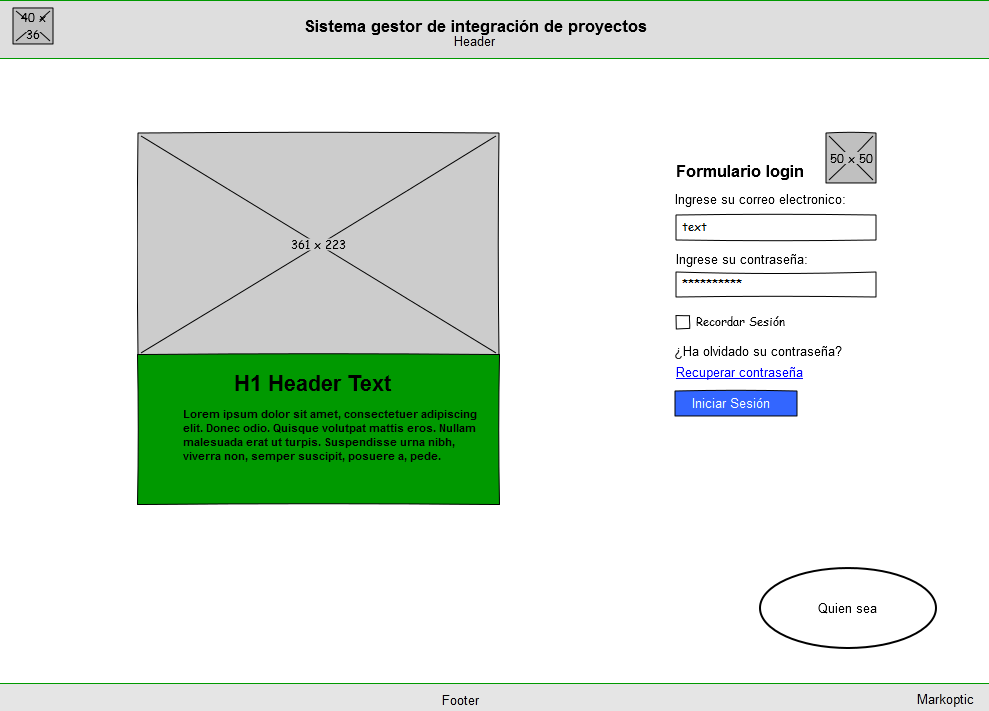


Figura x. Login

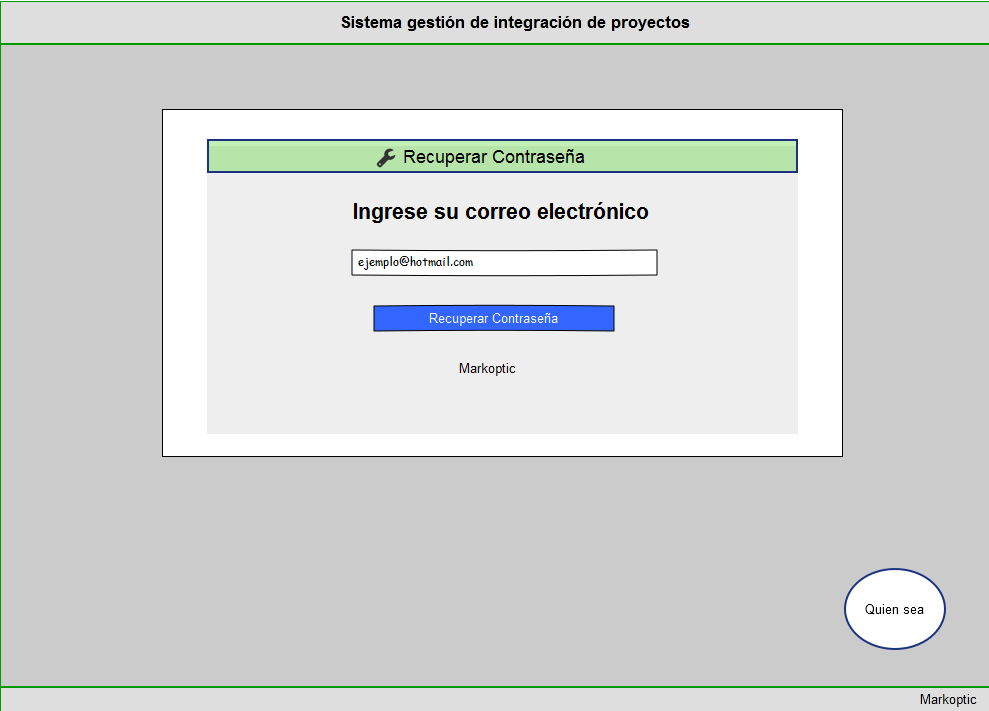


Figura x. Recuperar Contraseña



Figura x. Bienvenida

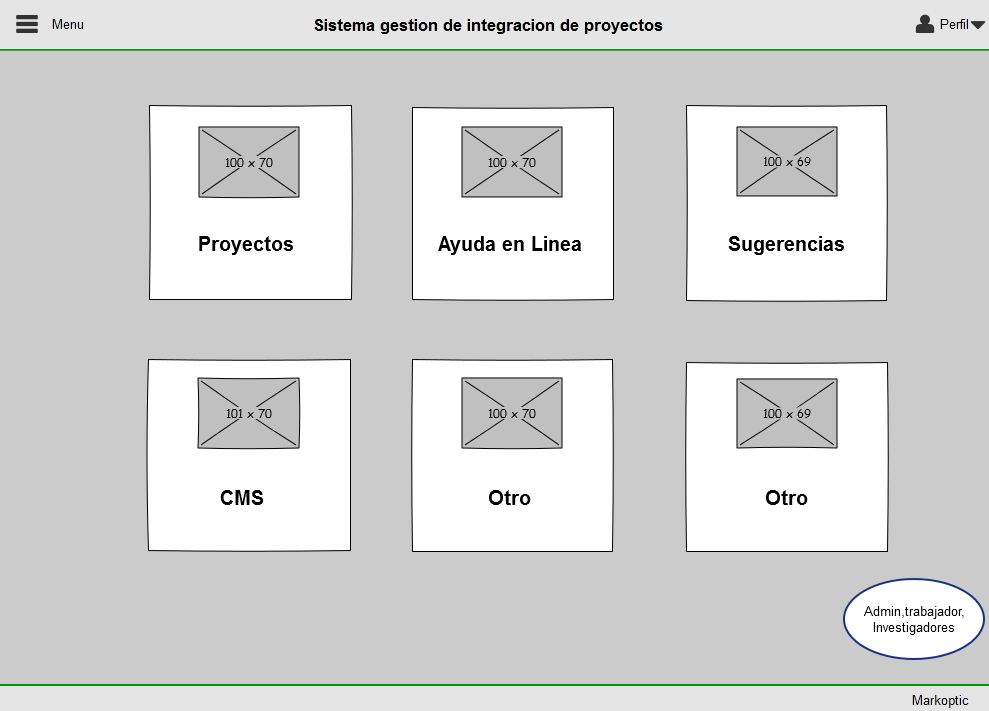


Figura x. Dashboard

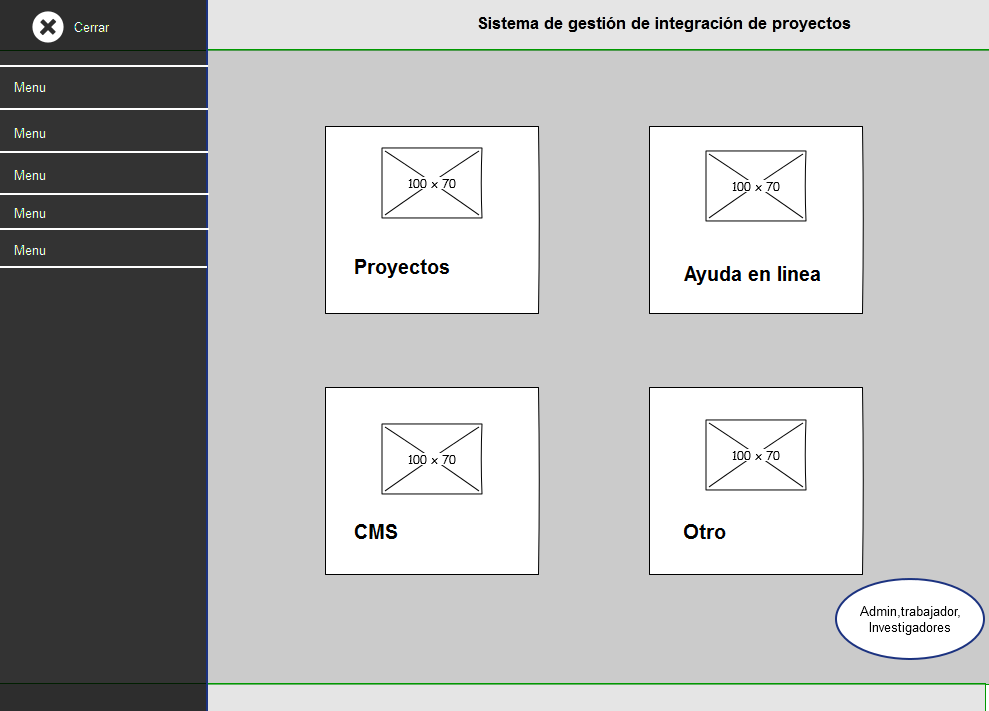


Figura x. Menú Abierto

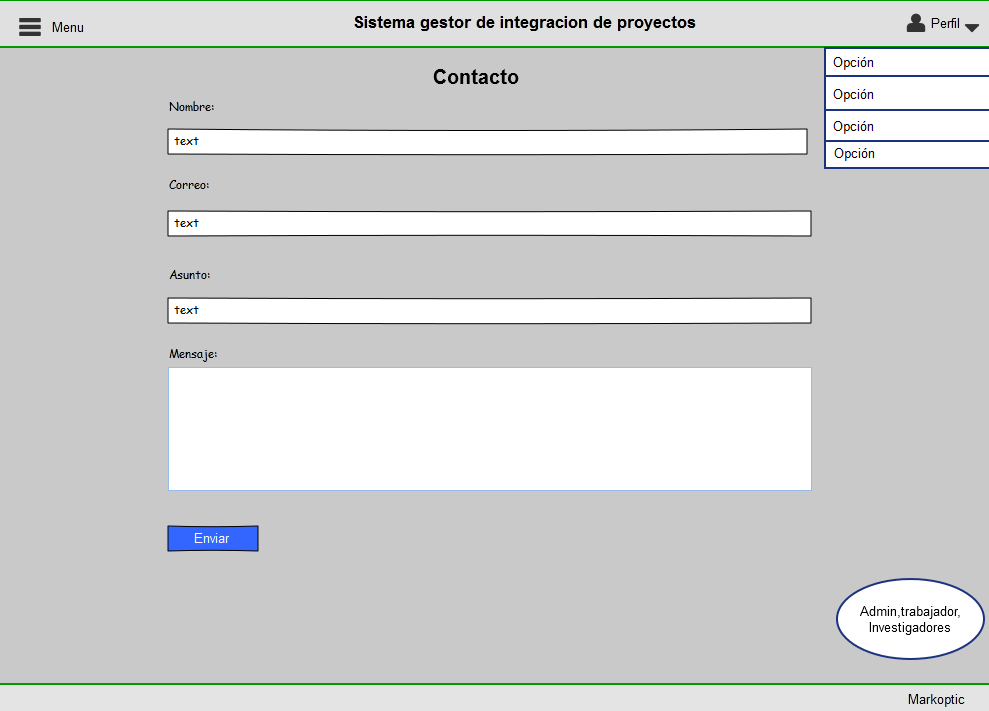
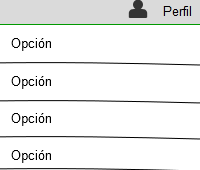


Figura x. Perfil/Perfil abierto



Figura x. Sugerencias



Figura x. Proyectos board

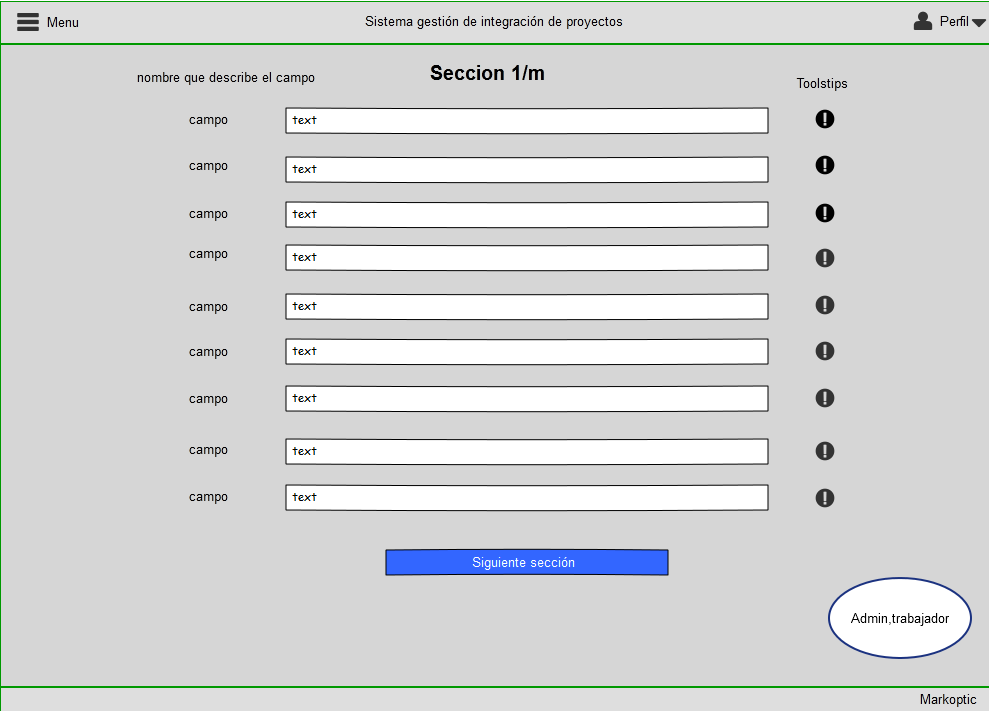


Figura x. Agregar proyecto

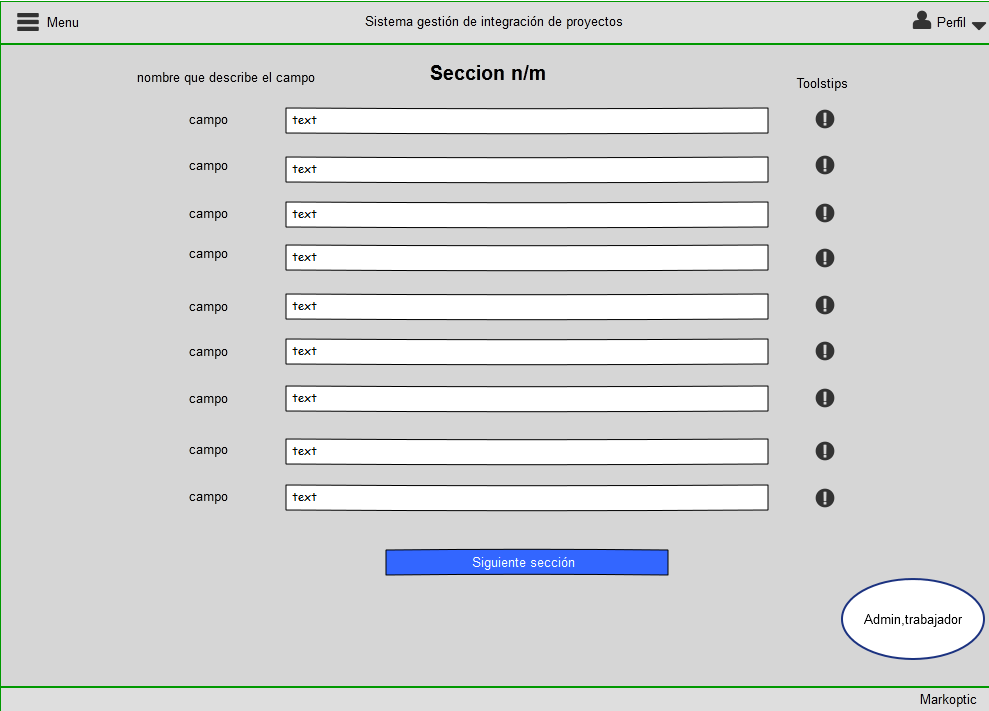


Figura x. Agregar proyecto 2

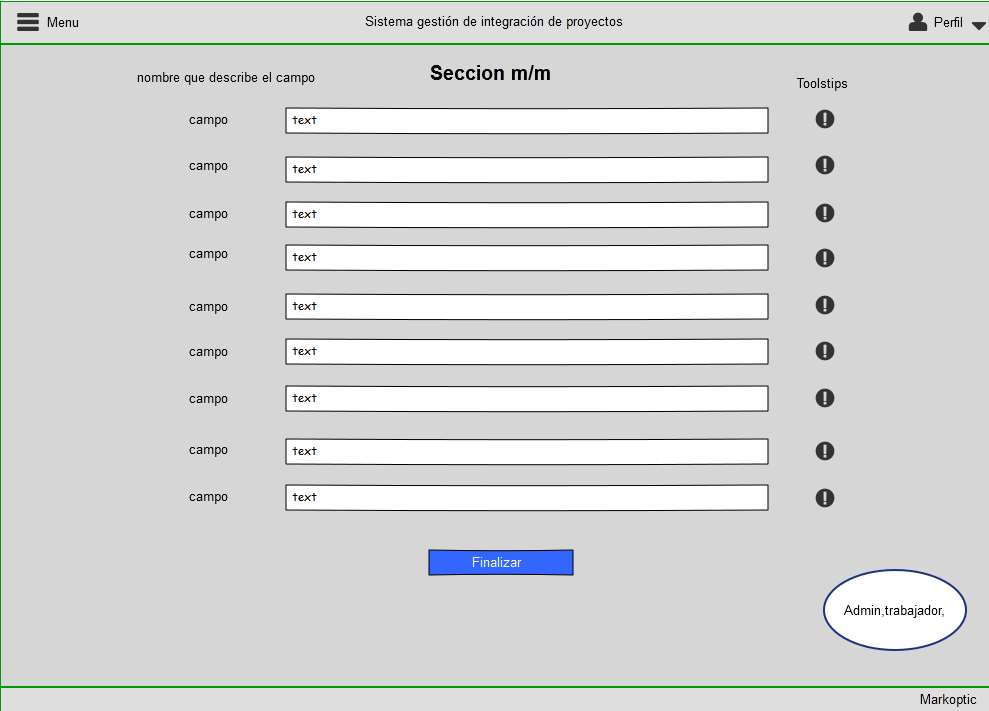


Figura x. Agregar Proyecto Finalizar

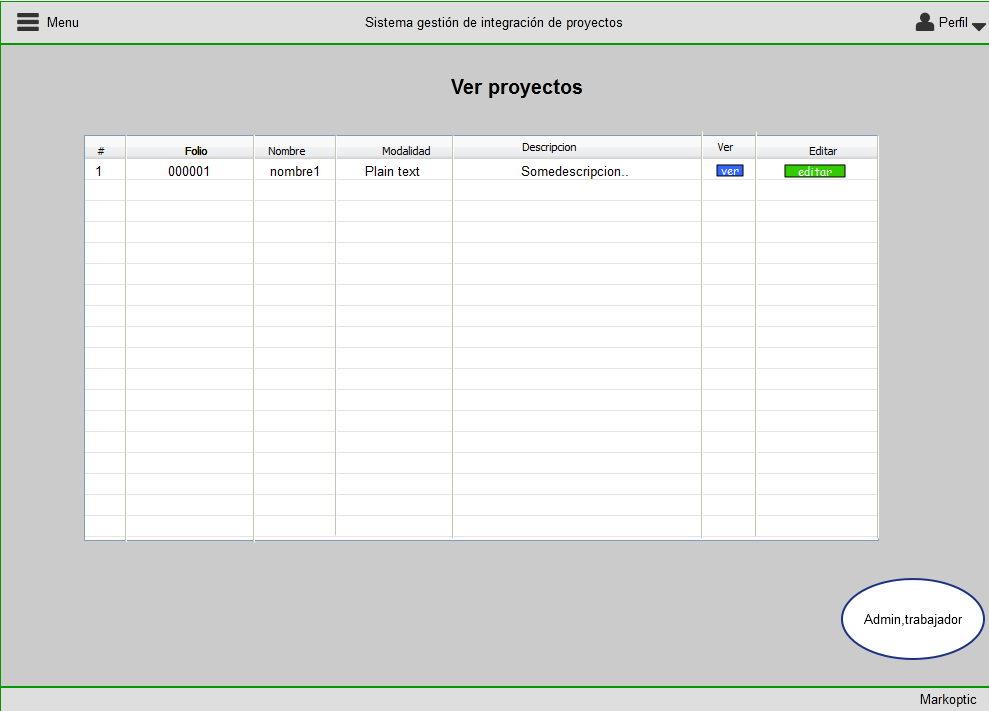


Figura x. Editar Proyectos

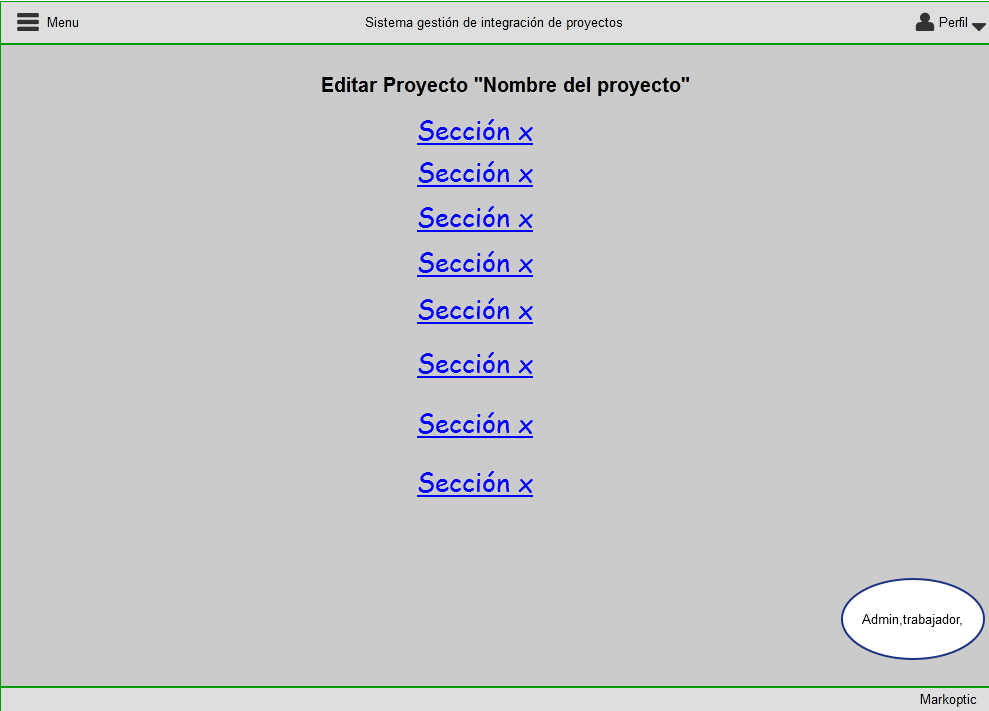


Figura x. Editar proyecto board

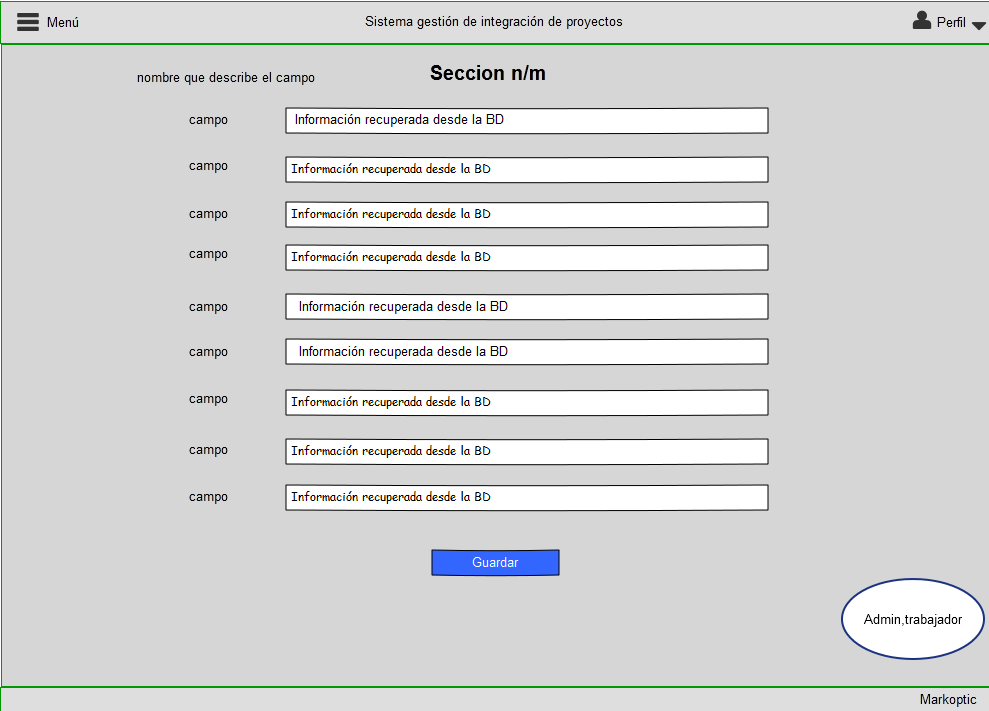


Figura x. Sección a editar

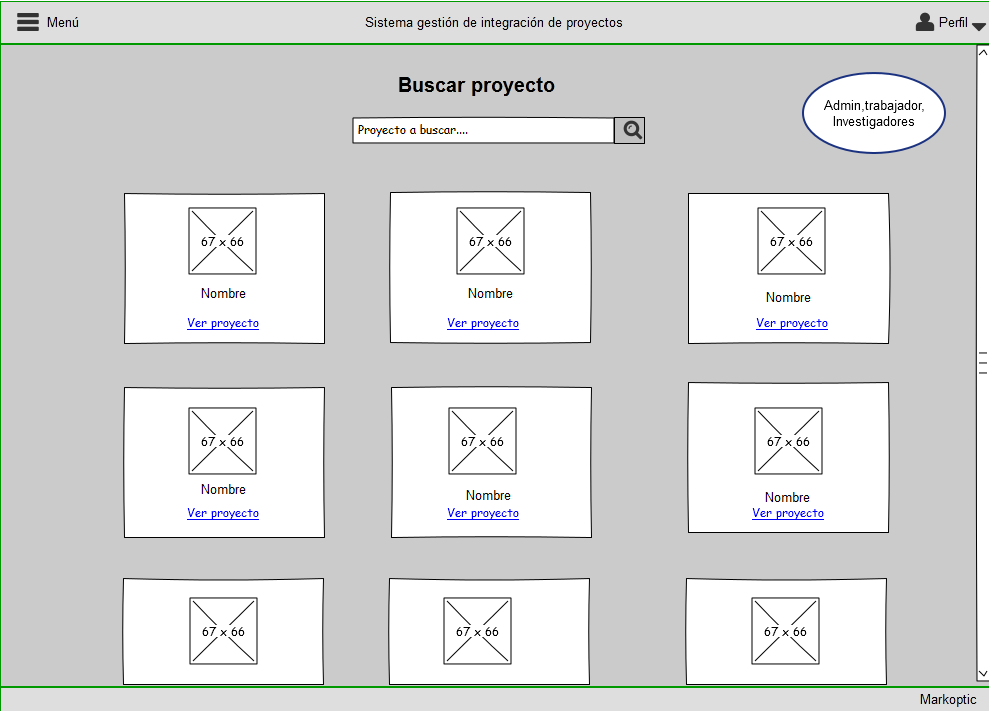


Figura x. Proyecto a buscar

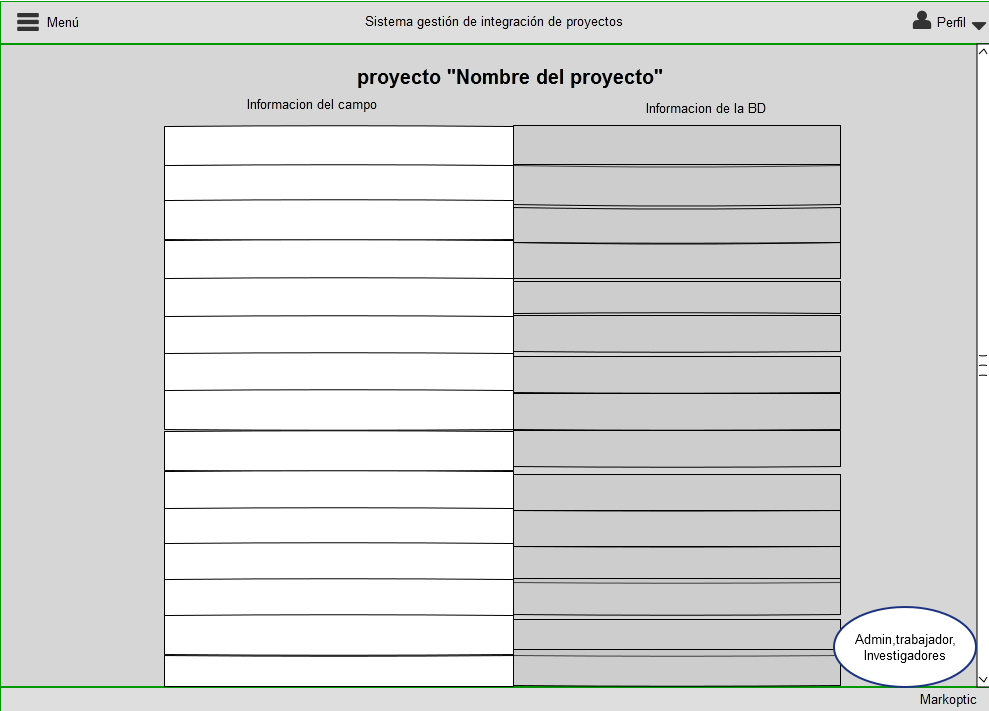


Figura x. Preview del proyecto

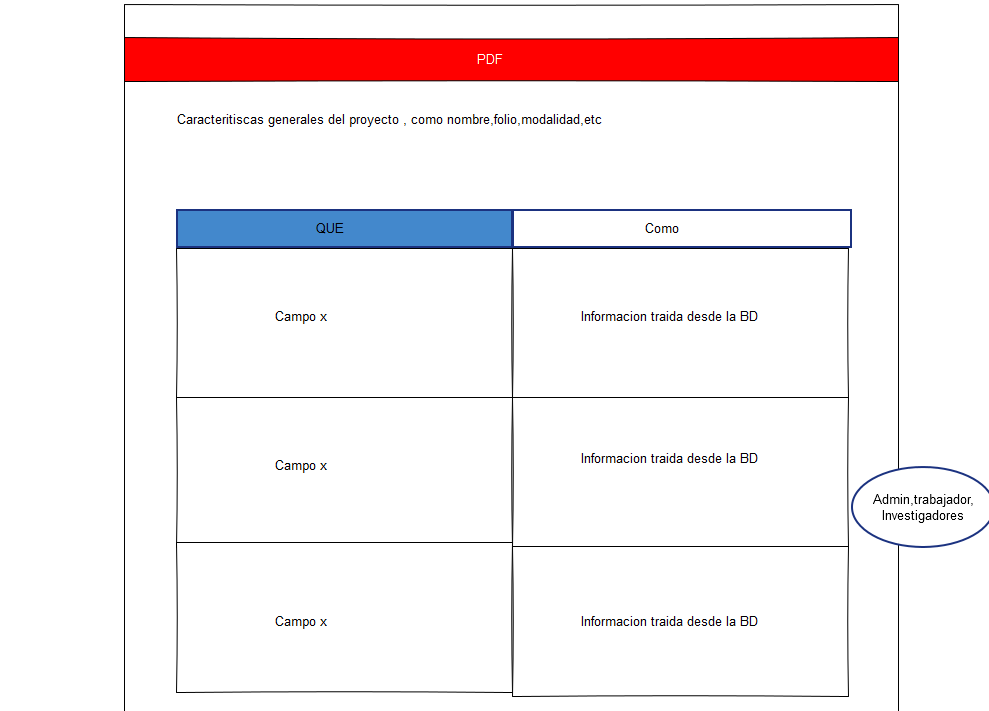


Figura x. Ver proyecto PDF.

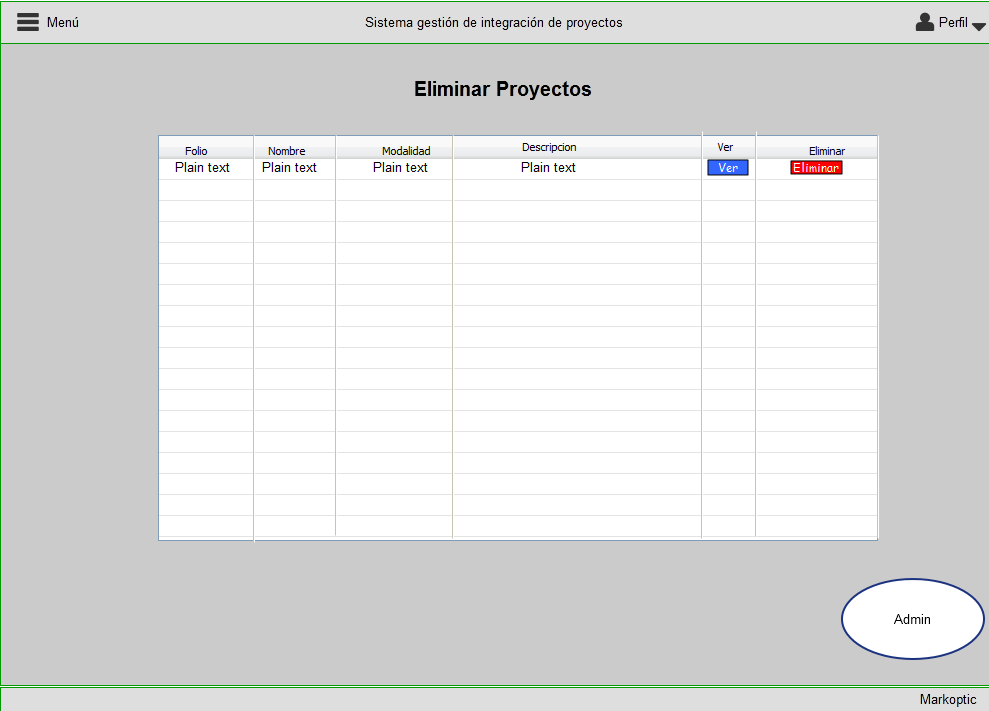


Figura x. Elimina Proyecto

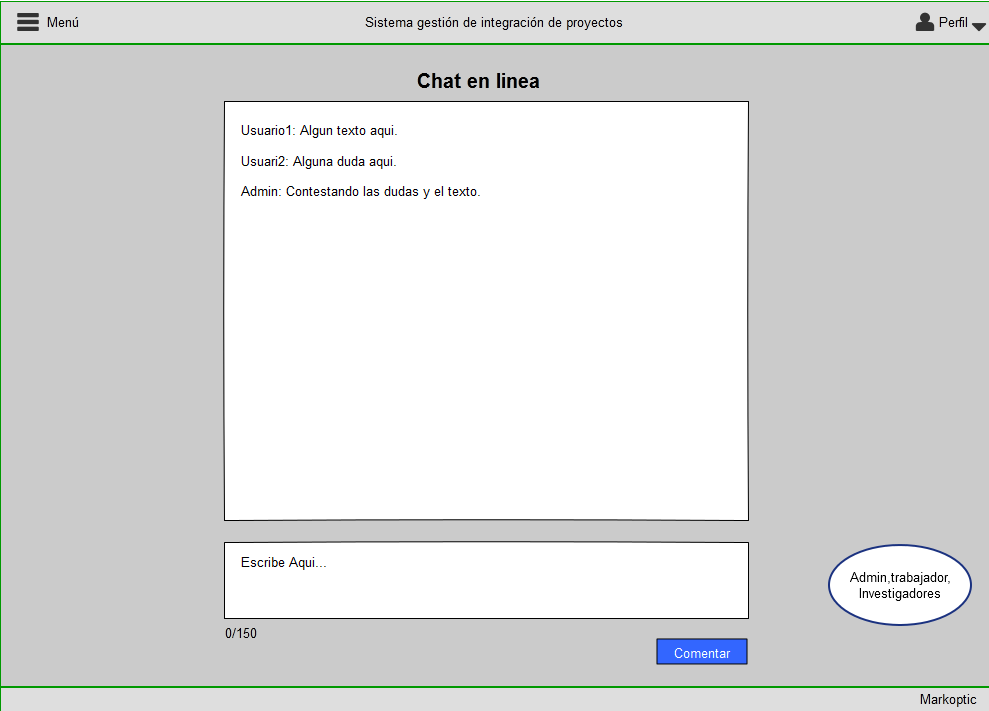


Figura x. Chat en Línea

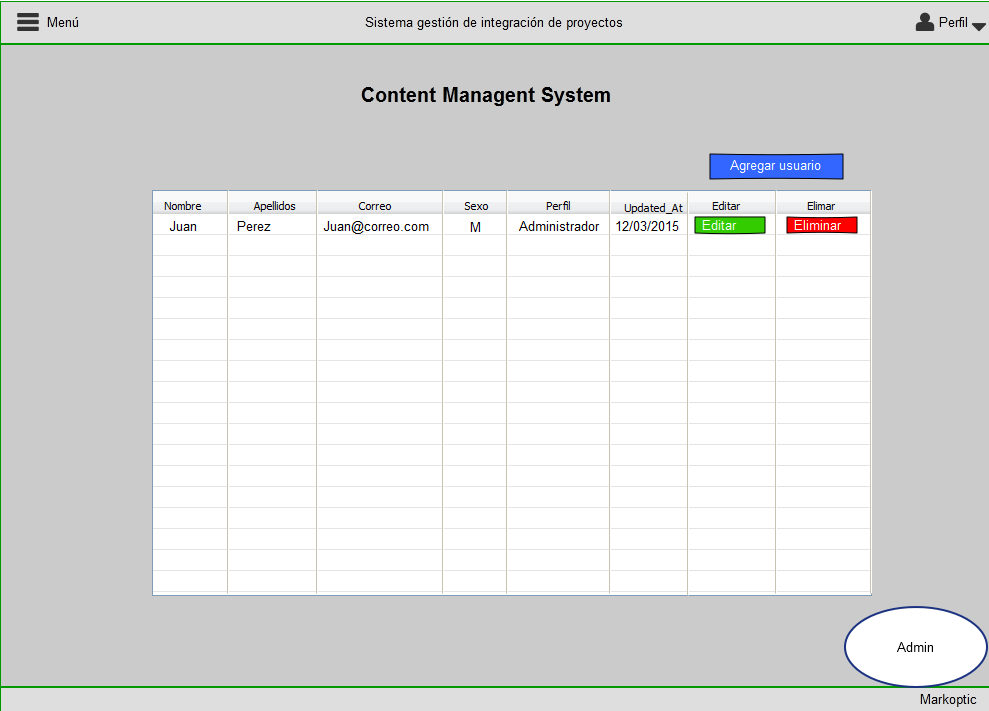


Figura x. CMS

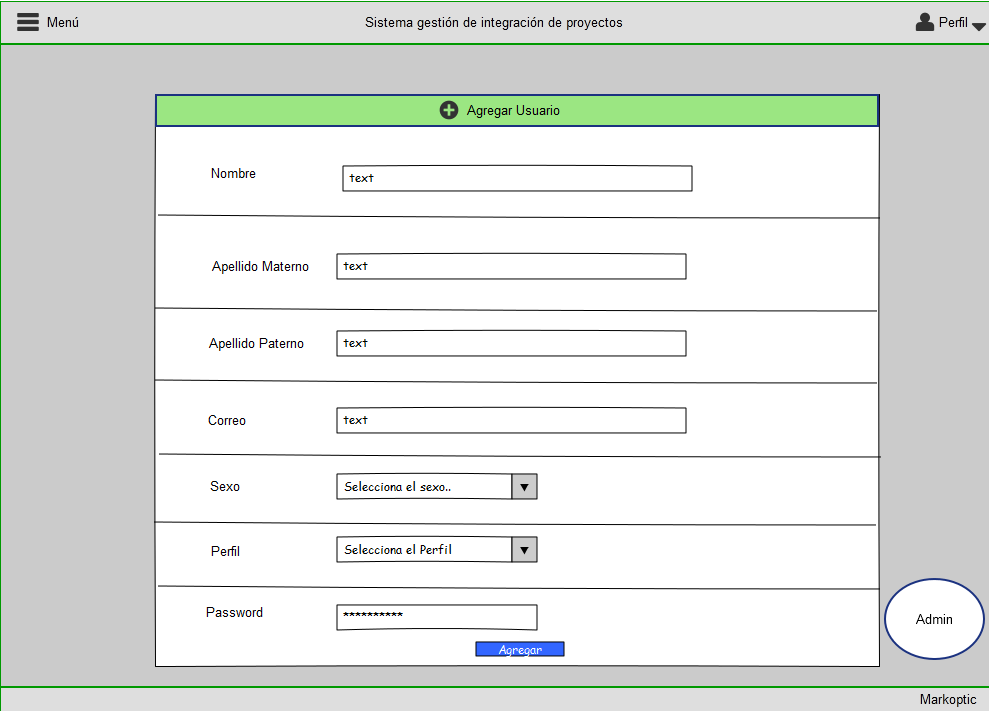


Figura x. Agregar usuario

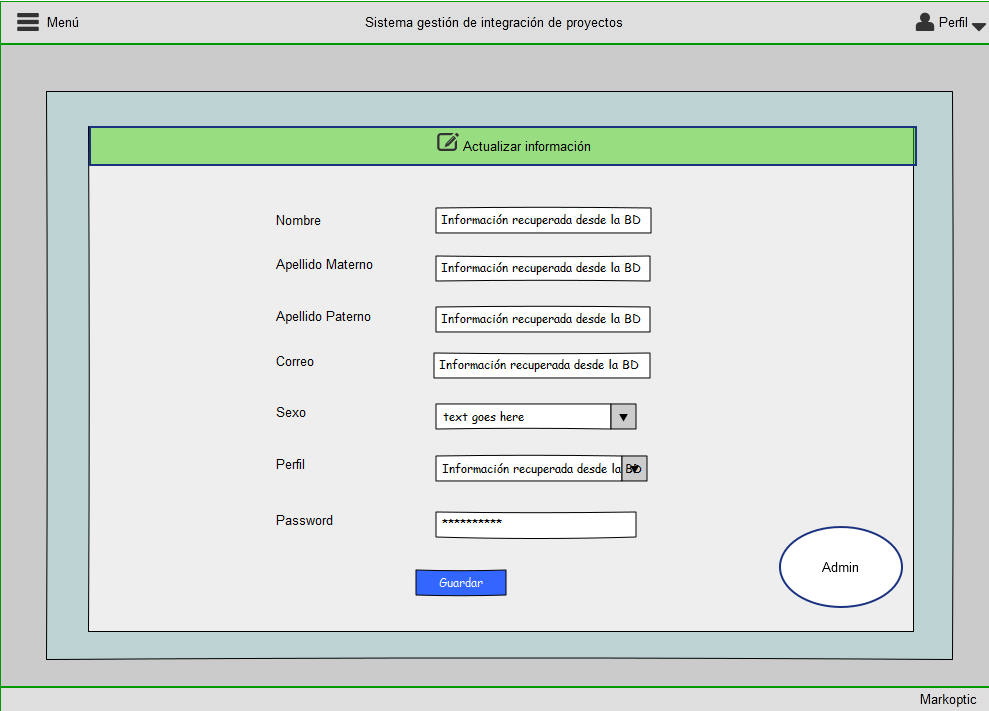


Figura x. Editar usuario

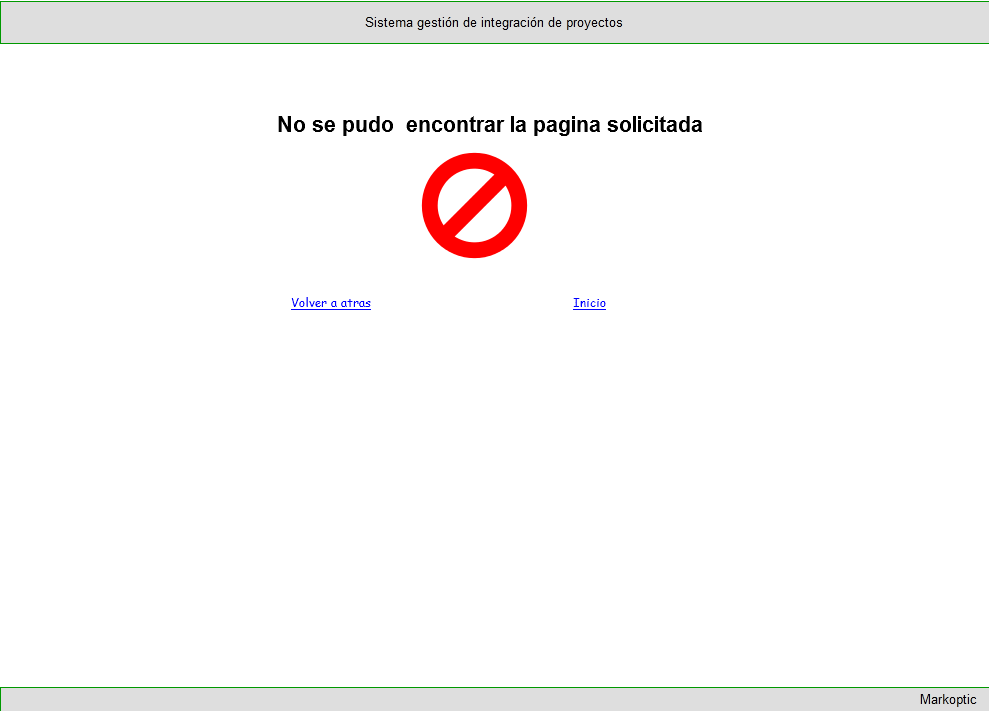


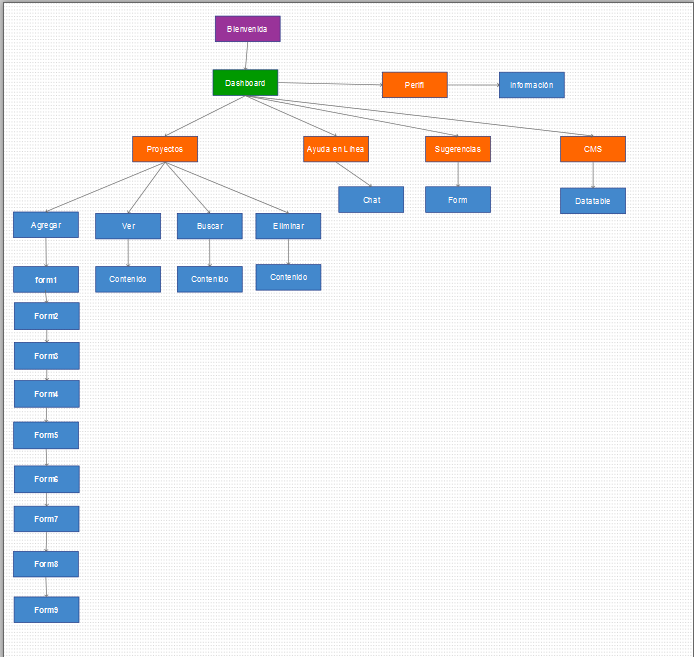
Figura x. Error 404

**6.3 Diseño de navegación**

En un sistema para la web es útil saber cómo están enlazadas las páginas. Ello significa que necesitamos un diagrama conteniendo nodos (nodes) y enlaces (links). Los nodos son unidades de navegación y están conectados por medio de enlaces.

A continuación se presentan 2 diagramas de navegación: uno general y otro más orientado a la ingeniería en sistemas.

**Diagrama general para personas sin conocimiento en ingeniería en sistemas.**

  
Figura x. Diagrama general de navegación.

**Diagrama general para personas con conocimiento en ingeniería en sistemas.**

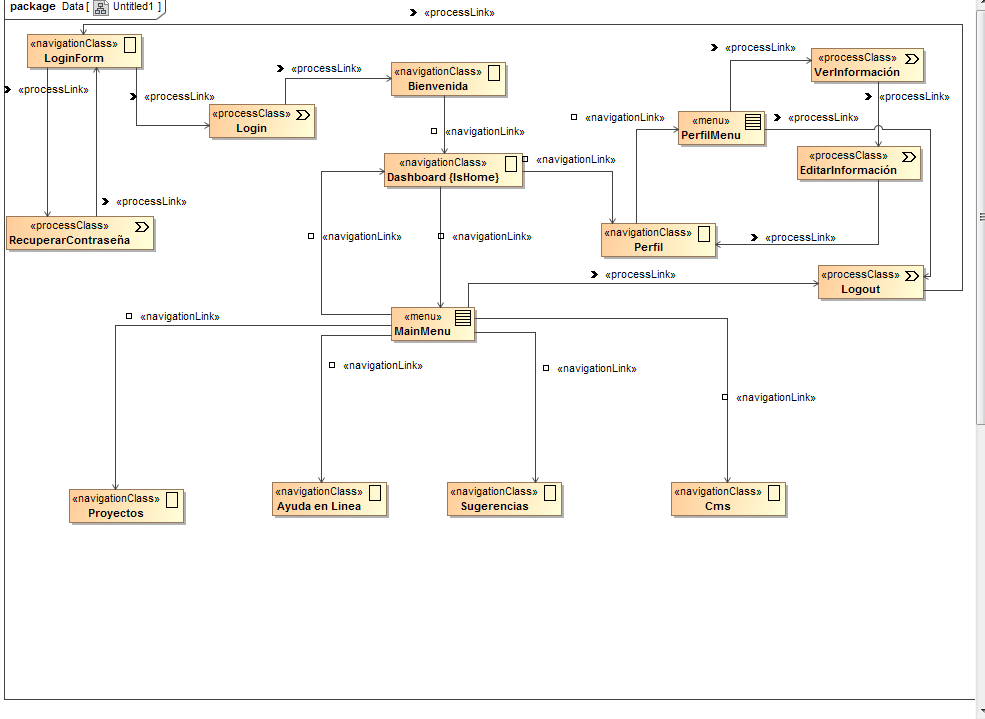


Figura x. Is Home

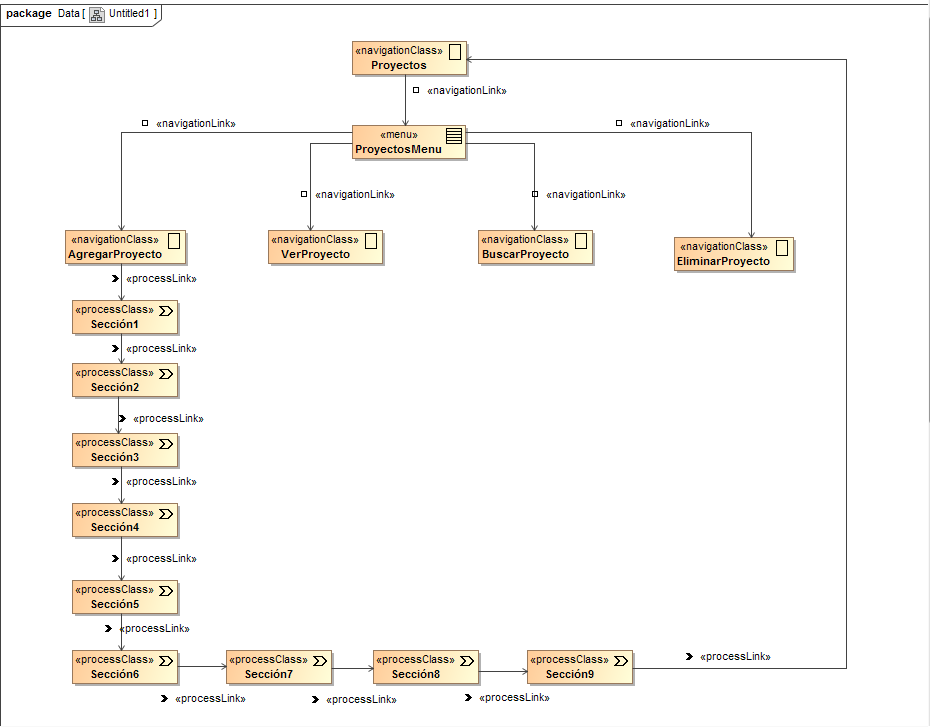


Figura x. Agregar proyecto navegación

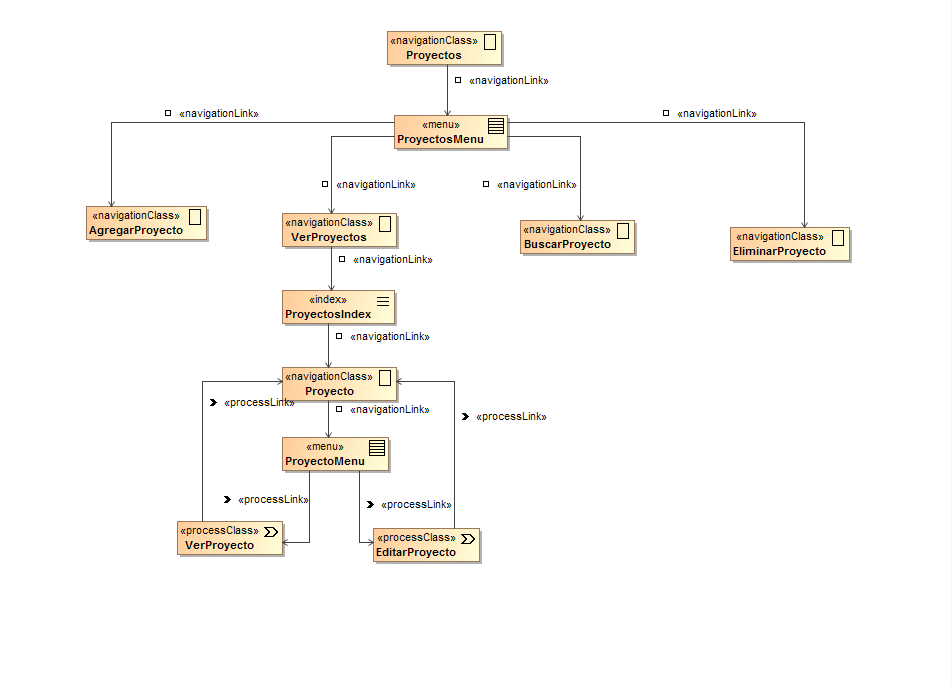


Figura x. Ver proyectos navegación

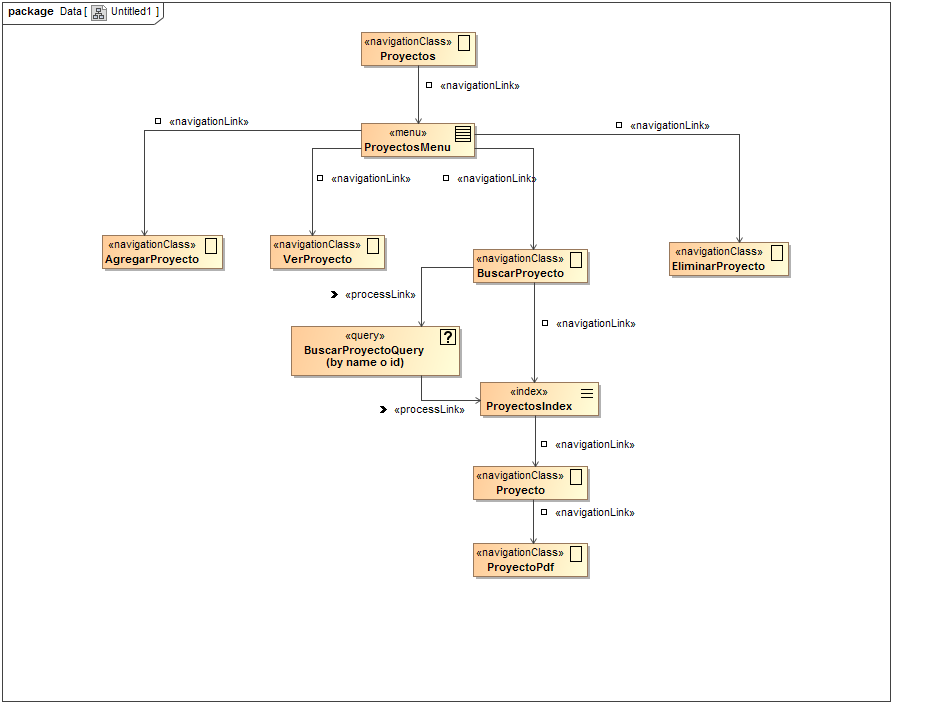


Figura x. Buscar proyecto navegación

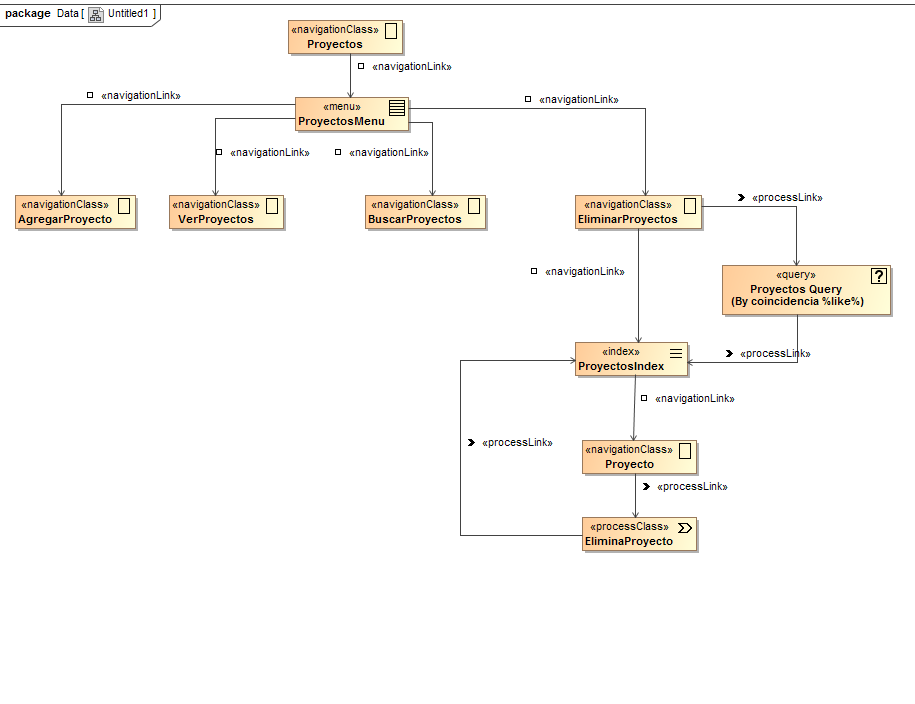


Figura x. Eliminar Proyecto Navegación

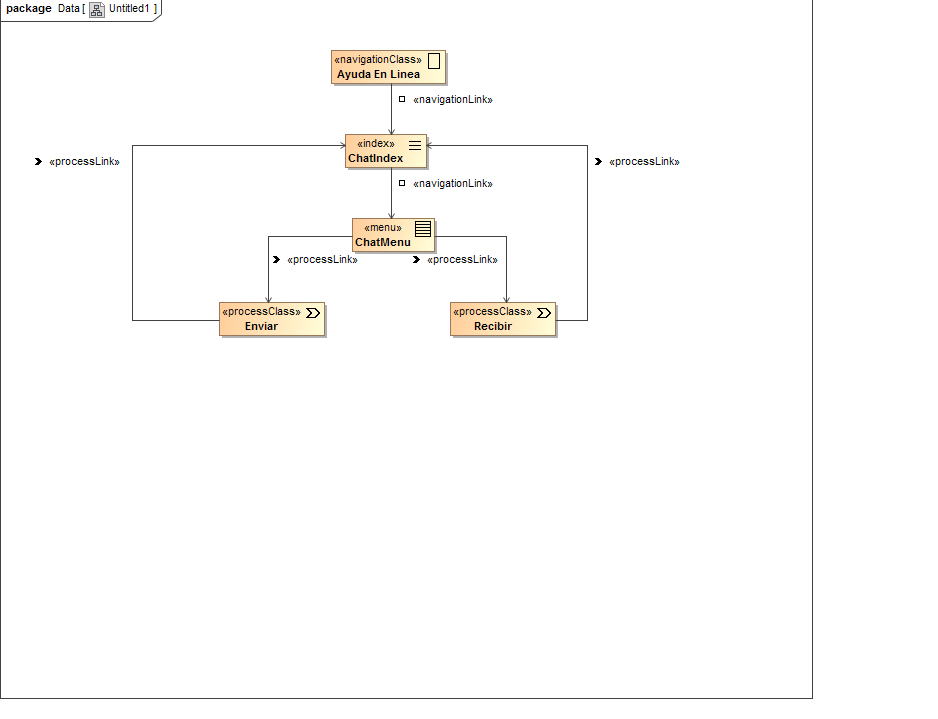


Figura x. Chat en Línea Navegación

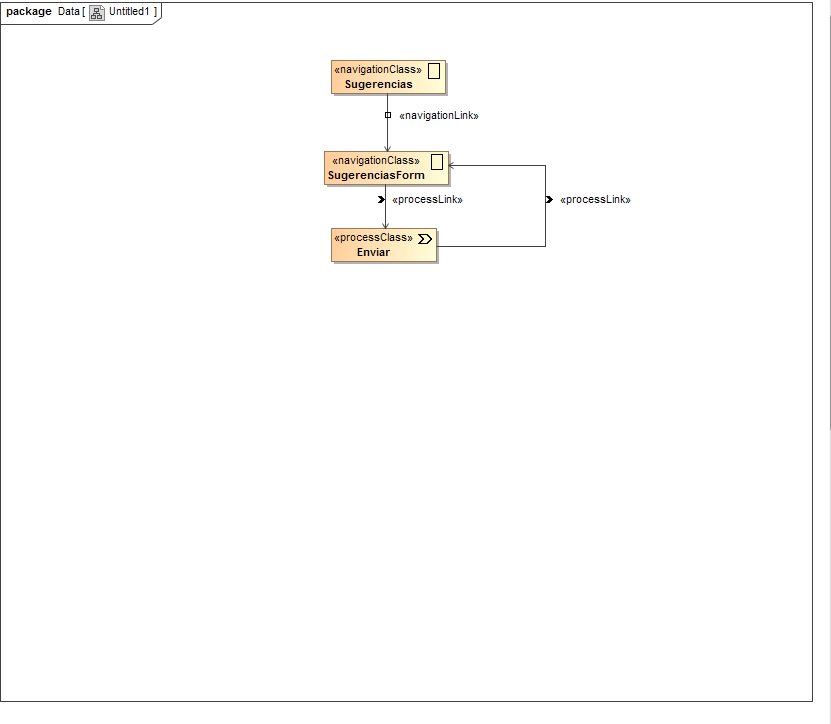


Figura x. Sugerencias Navegación

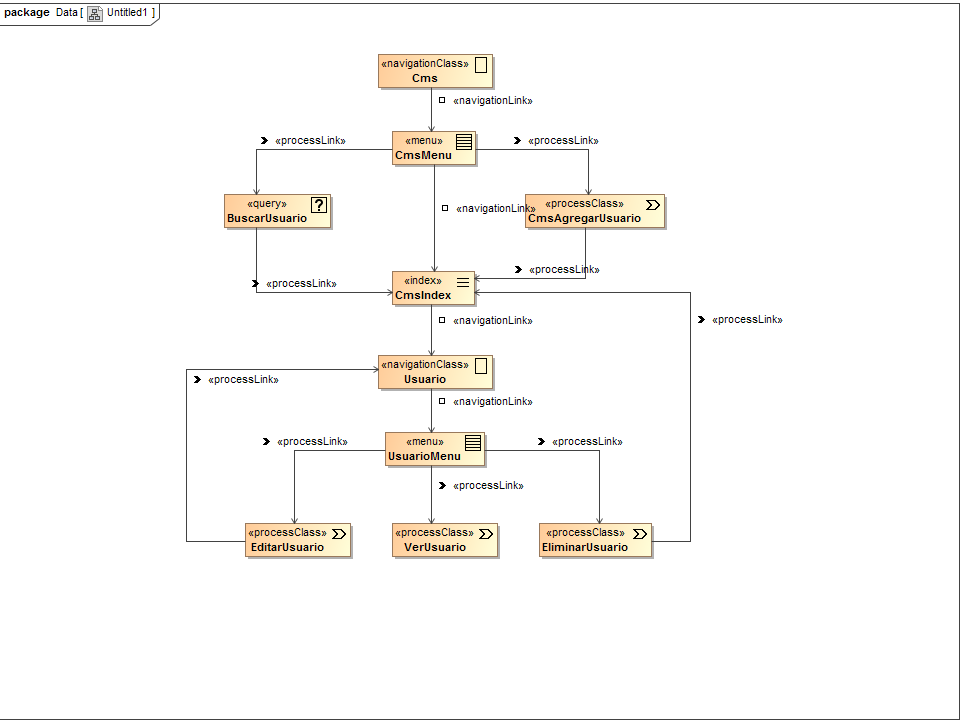


Figura x. CMS navegación

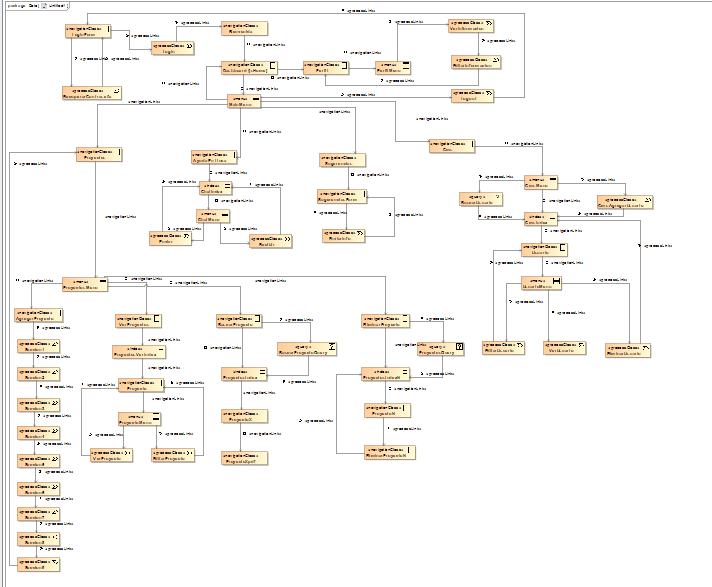


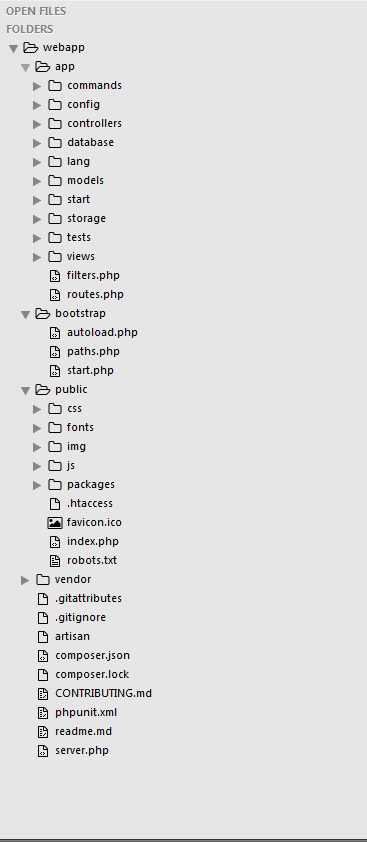
Figura x. Completa

**6.4 Diseño de contenido**

**6.5 Programación de la aplicación**

La programación del sistema se realizo utilizando el lenguaje de desarrollo web PHP, haciendo uso del framework laravel.

La estructura de una aplicación en laravel se puede observar de la siguiente manera:



En el primer nivel se pueden encontrar las carpetas app, public y vendor.

**App:** En esta carpeta se encuentra toda lo relacionado a nuestra aplicación, aquí es donde agregaremos nuestros archivos de programación y haremos las configuraciones correctas para que el entorno sea el adecuado. Más adelante se explicaran con detalle el contenido de las carpetas más importantes de este nivel.

**Public:** En esta carpeta se encuentran todos nuestros archivos externos que vayamos a utilizar en la aplicación, como se pueden apreciar aquí mismo se guardan los archivos css, los archivos javascript, las imágenes y alguno otro archivos que se vaya a utilizar en la aplicación. Necesitan ser colocados en estar carpeta para que puedan ser llamados dentro de la aplicación y así poder hacer usos de ellos, además de que no son archivos que nos puedan comprometer en caso de ser vistos por otras personas.

**Vendor:** En esta carpeta se encuentran los componentes necesarios para que el framework funcione de la manera que lo hace. No es recomendable editar archivos de esta carpeta.

Dentro de la carpeta App existen otras carpetas. De cuales se explicaran las más importantes:

**Config:** Como su nombre lo indica, en esta carpeta se encuentran todos los archivos relacionados a la configuración de nuestra aplicación como lo es: la base de datos, la autentificación, el correo, app.php , entre otros archivos que son de utilidada para tener toda nuestra aplicación en orden.

**Controllers:** Aquí es donde se guardan los controladores.  
  
Los controladores se encarga de la parte conocida como modelo de negocio, en otras palabras, hacen que el programa cumpla las funciones para las que fue creado, procesan la información, realizan tareas, automatizan algún proceso en el sistema, etc. Como se menciono antes , estos se guardan en la carpeta controllers. Otra característica relevante de los controladores es que al crear un nuevo controlador este debe de extender del controlador base que se encuentra por default en el framework y desde ahí entonces ya se crean las clases necesarias para que el controlador realiuce la función deseada.

**Database:** Laravel ofrece la opción de poder crear las tablas de la base de datos y llenar las tablas de información desde la misma aplicación, con el fin de poder exportar las tablas junto con el proyecto, Todo al alcance de un solo comando. Para cada tabla se necesita un archivo y en esta carpeta es donde se guardan los archivos de las tablas , así como los “seeder” o archivos para llenar las tablas con información

**Lang:** Laravel cuenta con sus propios mensajes a la hora de mostrar los errores, los mensajes se cuentan ya predefinidos, colocados en esta carpeta en el idioma inglés. Laravel nos permite seleccionar el idioma en que se pueden mostrar los mensajes de los errores, para esto debemos colocar el idioma en que queremos que salga el mensaje e indicarlo en los archivos de configuración de la app.

**Models:** Aquí es donde se guardan los modelos.

El modelo se encarga de decirle al sistema donde y como debe guardar la información y también como debe accederla, en laravel los modelos de las tablas que componen la base de datos se guardan en la carpeta models, cuando se crea el archivo del modelo de una tabla se le debe especificar que campos contiene esa tabla, en que orden se llenan y si son accesibles o no. Otra característica relevante de los modelos es que al crear un nuevo modelo este debe de extender de Eloquent.

**Storage:** Aquí es donde se almacenan el logs de los errores, muy necesario, a la hora de querer depurar o arreglar un problema.

**Views:** Aquí es donde se guardan las vistas.

Las vistas le indican al sistema de que forma debe mostrar la información que esta almacenada , además permiten que el usuario interactue con el sistema, esta es la capa en la que el usuario toma contato de forma directa. Como se menciono antes, estas se guardan en la carpeta views. En laravel las vistas entan programadas en código HTML y si el usuario desea puede utilizar extensiones del lenguaje de plantillas de Blade del propio framework, con el fin de que todo este mas ordenado y sea mas vistoso.

Como se pueden dar cuenta el framework Laravel está basado en el modelo arquitectónico de capas “MVC” (modelo, vista, controlador), donde cada una de las capas se le asigna un papel dentro del sistema para hacer funcionar a este.

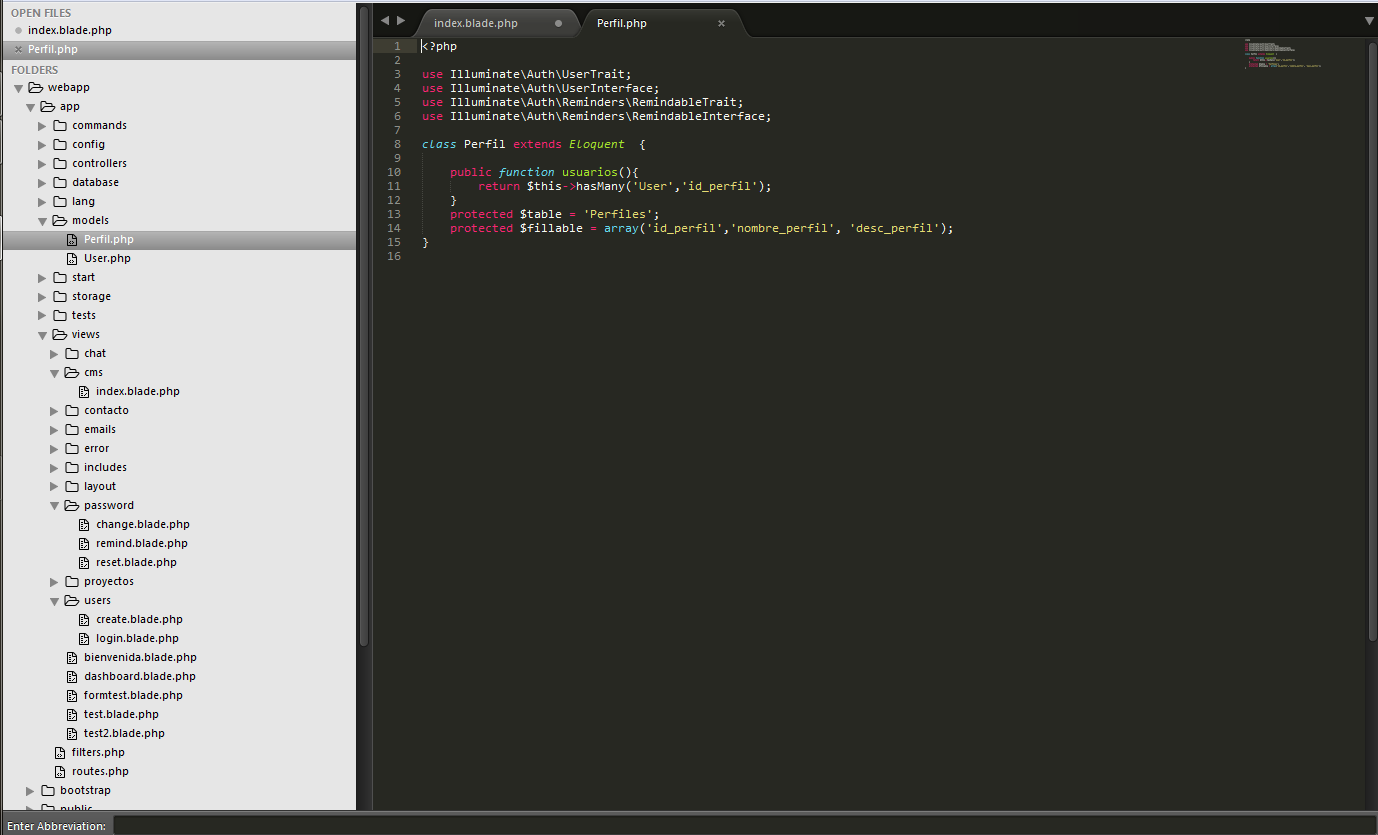


Figura x. Ejemplo de un modelo

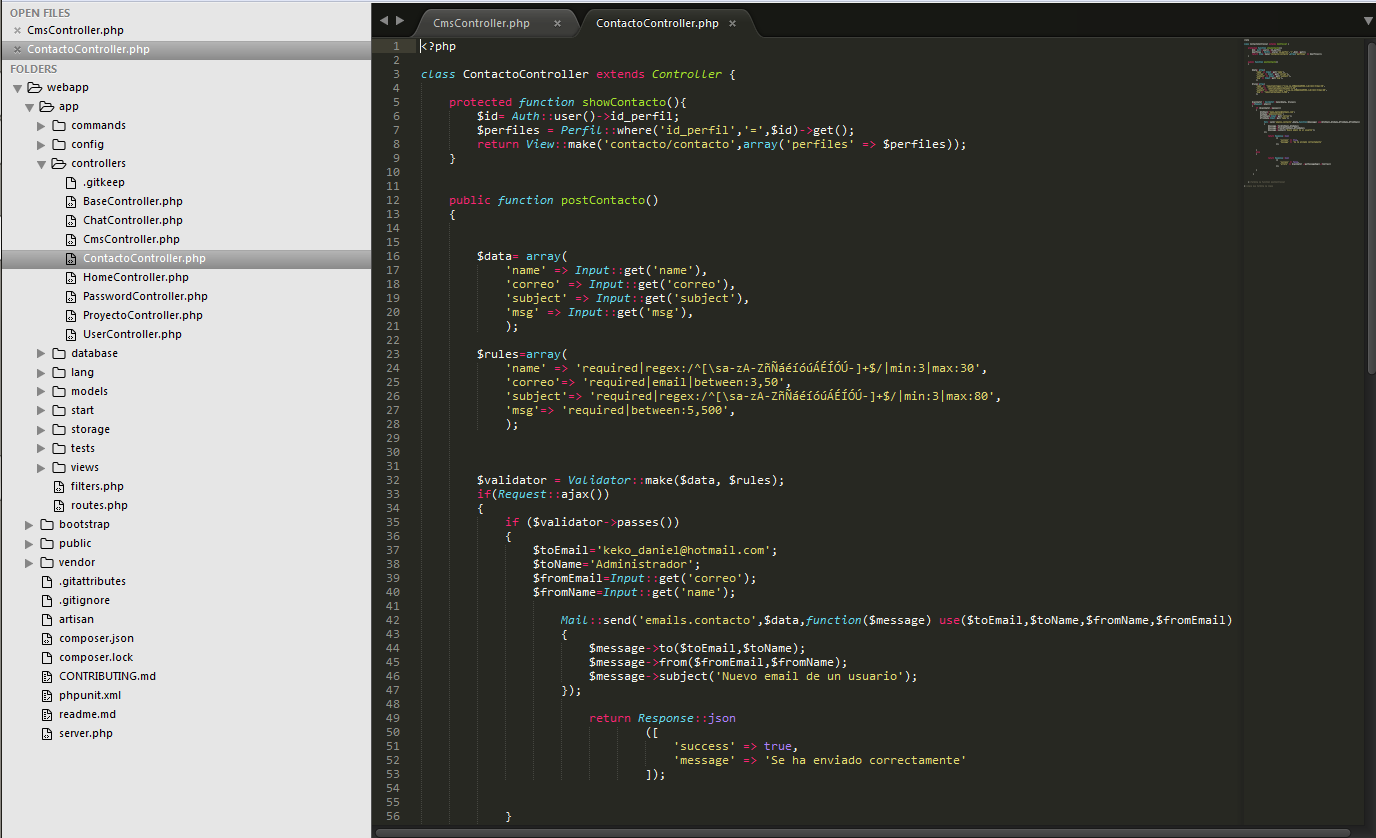


Figura x. Ejemplo de un controller

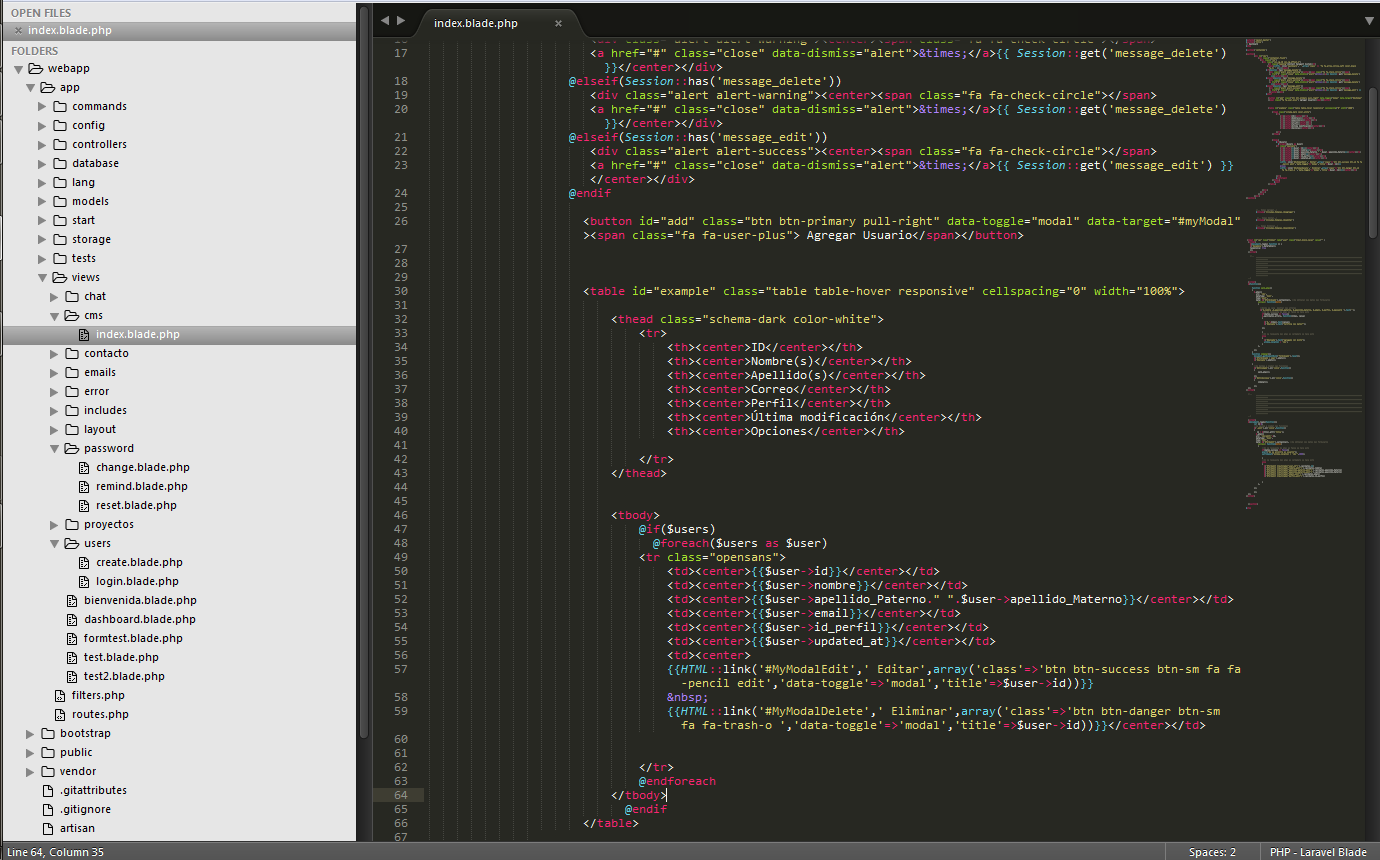


Figura x. Ejemplo de una vista

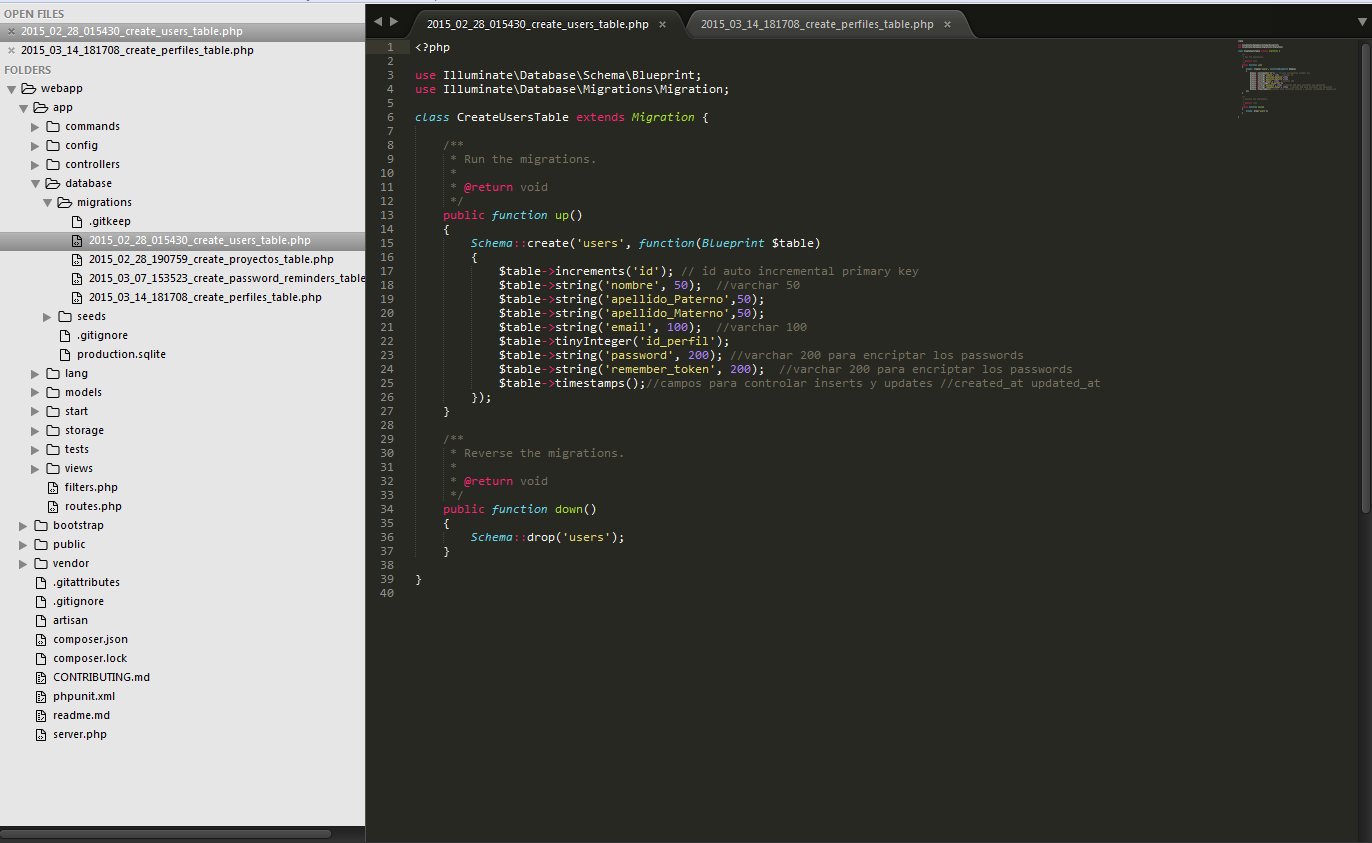


Figura x. Ejemplo de una tabla.

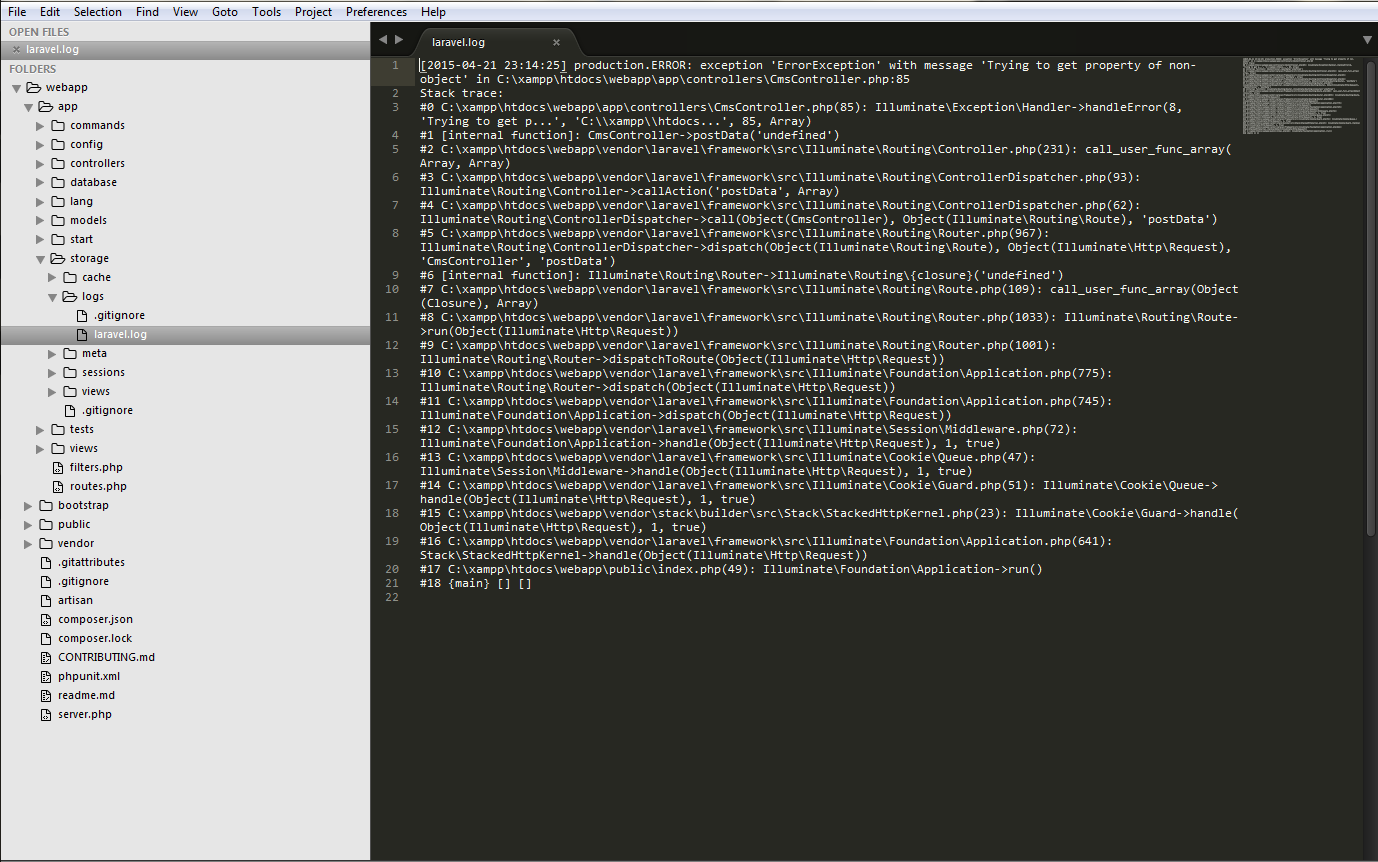


Figura x. Log

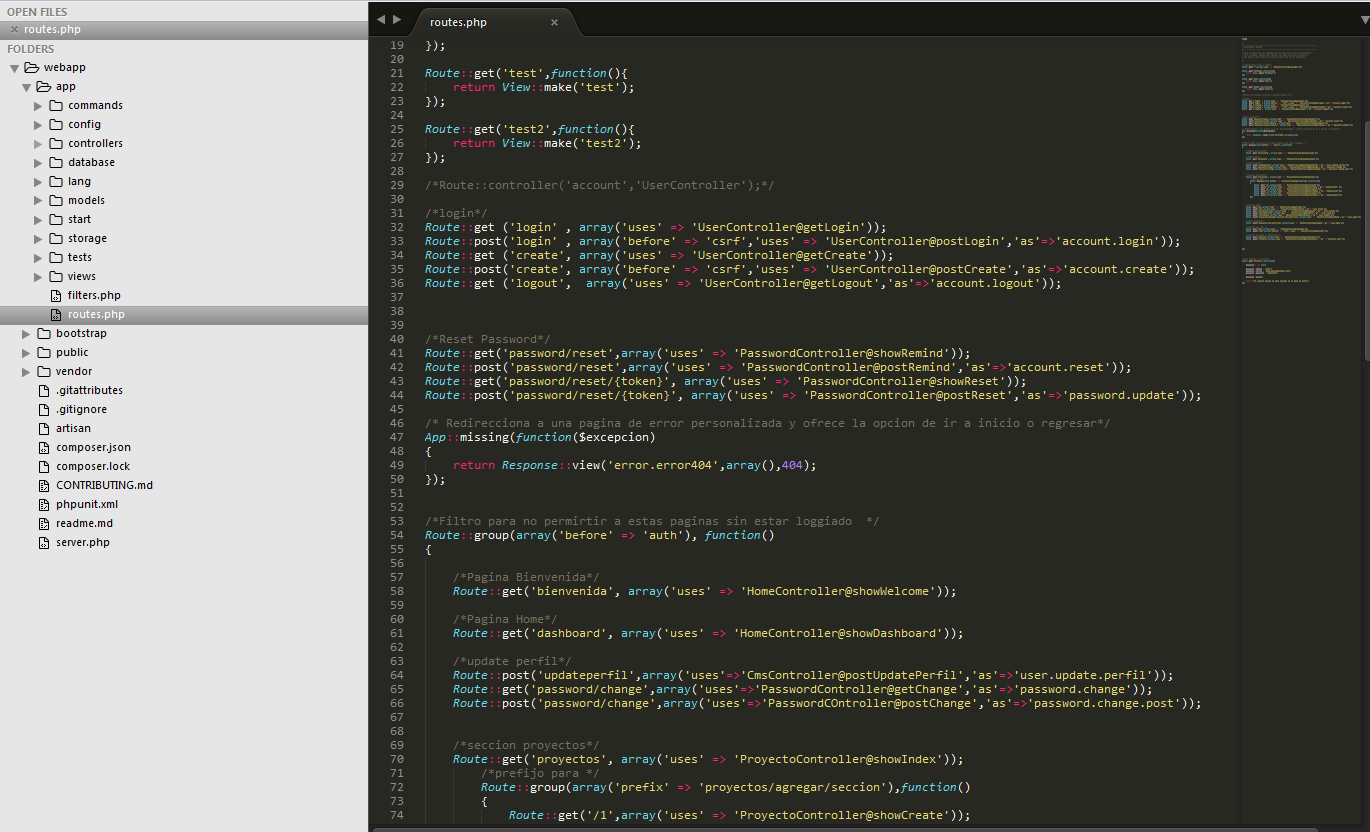


Figura x. Routes

**6.6 Pruebas Funcionales**

Como se sabe hoy en día las pruebas son necesarias en todo desarrollo de software, ya que nos permiten identificar errores o “bugs” en el sistema en las diferentes etapas del desarrollo. Además nos permiten verificar si la aplicación tiene el funcionamiento correcto cuando el usuario falla en algún requisito funcional.

**¿Qué son las pruebas de caja negra?**

Son cuando una aplicación es probada usando su interfaz externa, generalmente la GUI (Katz-Lichstenstein, 2003).

Una prueba de tipo caja negra se lleva a cabo sin tener conocimiento de la estructura/funcionamiento interno del sistema. de ahí su nombre. Quien realiza la prueba solo conoce las entrada apropiadas que deberá recibir la aplicación, así como las correspondientes salidas, sin llegar a saber cómo es que se realiza este proceso (Koudinya,2003).

Las pruebas de caja negra son, ni más ni menos que, pruebas funcionales dedicadas a “mirar” en el exterior de lo que se prueba. Las pruebas de caja negra se limitan a que el tester pruebe con “datos” de entrada y estudie como salen, sin preocuparse de lo que ocurre en el interior.



A continuación se anexan imágenes utilizando las pruebas de caja negra al sistema de gestión de integración de proyectos.

En la siguiente imagen se muestran la secuencia al ingresar los datos de manera incorrecta.

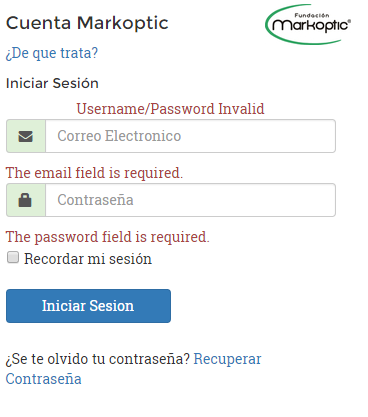


Figura x. Login test

En esta imagen se puede apreciar, como el sistema valido que el correo que se ingreso exista en la base de datos. Antes de eso se tuvo que validar que se ingresar un texto en el input y que se ingresará un formato valido de correo electrónico, pero claro, un usuario normal no conoce el mecanismo de esto.



Figura x. Recuperar Contraseña Test

En esta imagen se aprecia como un actor trata de acceder a una sección a la cual no tiene permiso. En la imagen se muestra cual es el resultado de salida.

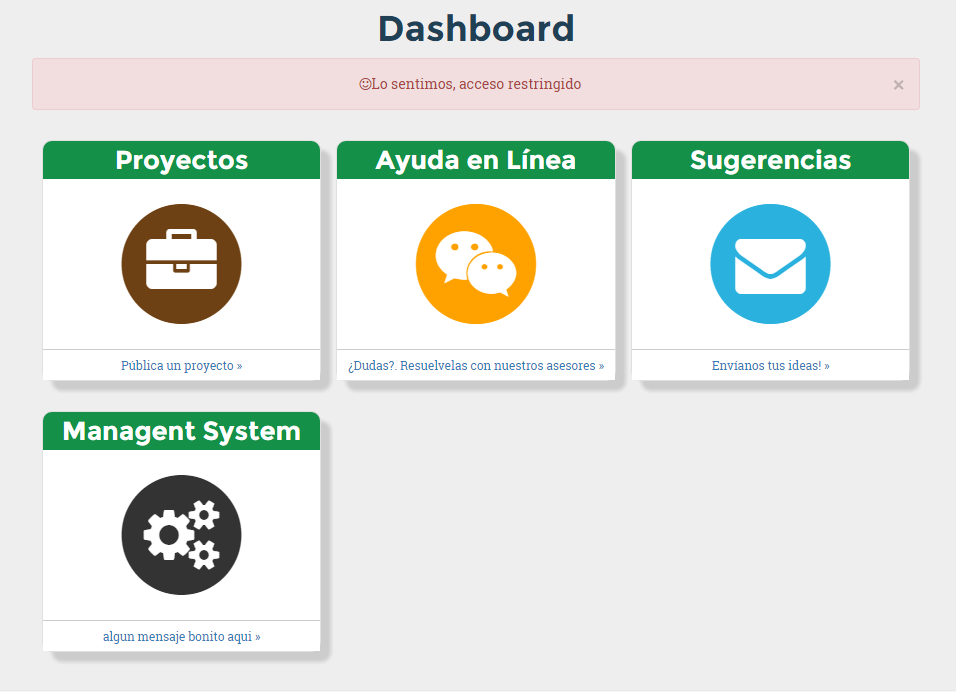


Figura. Secciones test

En esta imagen se aprecia cual es la salida de datos cuando un formulario no se llena completamente.

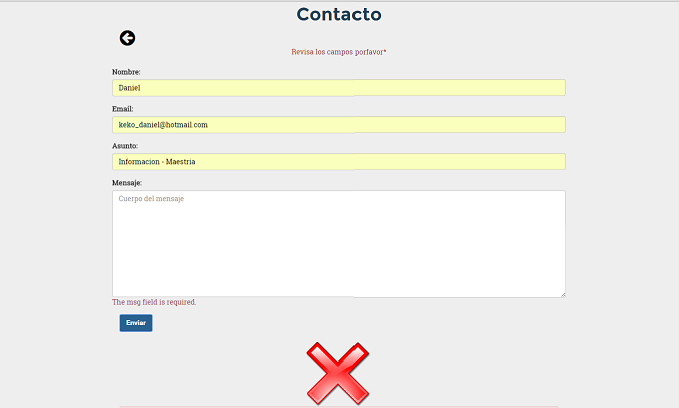


Figura. Form test

En la siguiente imagen se puede apreciar cuando el administrador llena los campos de manera incorrecta.

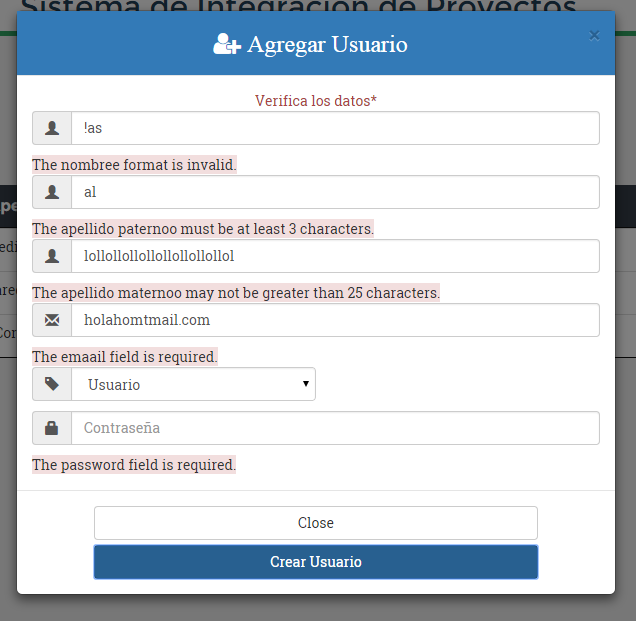


Figura x. Agregar usuario test

**6.7 Funcionamiento del sistema**

**7 Conclusiones y Recomendaciones**

**8- Competencias Desarrolladas y/o aplicadas.**

**9-Referencionas Bibliográficas y virtuales**