



Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación
Centro de Investigación en Sistemas de Información



DESARROLLO DE UNA APLICACIÓN WEB DE GESTIÓN DE PAQUETES Y ENVÍOS INTERNACIONALES QUE GARANTICE LA DISTRIBUCIÓN A NIVEL NACIONAL.

Trabajo Especial de Grado
presentado ante la Ilustre
Universidad Central de Venezuela
por el Bachiller:

Jeisson Arturo Ferreira Pereira
C.I. 18.003.702

para optar al título de Licenciado en Computación

Tutor: Dr. Franklin Sandoval

miércoles, 6 de noviembre de 2019



Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación
Centro de Investigación en Sistemas de Información



DESARROLLO DE UNA APLICACIÓN WEB DE GESTIÓN DE PAQUETES Y ENVÍOS INTERNACIONALES QUE GARANTICE LA DISTRIBUCIÓN A NIVEL NACIONAL

Autor:

Jeisson Ferreira
V.18.003.702

Tutor:

Franklin Sandoval

Fecha: octubre de 2019.

RESUMEN

Actualmente la tecnología se encuentra en una era donde el crecimiento va acelerado y exponencial, donde existen infinidad de temas y áreas desarrolladas con el mismo o similar fin; para tener una mejora evolutiva en una sociedad o país, como la visión de administrar mejor el consumo de tiempo y energía, es necesario hacer la integración de nuevas tecnologías que ayuden a la automatización y eficiencia en la realización de las tareas diarias. En este trabajo de grado se toma como propósito el desarrollo de una solución orientada a mejorar la manera en que se implementa el seguimiento de los paquetes de Lear, una empresa de envío internacional. El objetivo general fue desarrollar una solución óptima de gestión que permita la visualización del estado de los paquetes comprados por los usuarios, haciendo un registro del paquete y por medio de este, realizar un seguimiento desde el almacén de Estados Unidos, hasta la sede en la que se registró el retiro de la paquetería. Así mismo, esta solución utilizara la web para así poder llegar a todo el territorio nacional. Durante el desarrollo se recogió la información de desarrollo del proyecto, en el cual se aplicaron fundamentos y herramientas aptas para el desarrollo adecuado de una solución que integrara una aplicación web para que los clientes tengan un mejor manejo de su paquete.

Palabras Claves: Aplicación web, envío, paquetería, tecnologías.



ACTA

Quienes suscriben, Miembros del Jurado designado por el Consejo de la Escuela de Computación para examinar el Trabajo Especial de Grado, presentado por el Bachiller Jeisson Arturo Ferreira Pereira C.I.: 18.003.702, con el título **“DESARROLLO DE UNA APLICACIÓN WEB DE GESTIÓN DE PAQUETES Y ENVÍOS INTERNACIONALES QUE GARANTICE LA DISTRIBUCIÓN A NIVEL NACIONAL.”**, a los fines de cumplir con el requisito legal para optar al título de Licenciado en Computación, dejan constancia de lo siguiente:

Leído el trabajo por cada uno de los Miembros del Jurado, se fijó el día 22 de octubre de 2019 a las 10 am, para que su autor lo defendiera en forma pública, en el aula PBIII, mediante una exposición oral de su contenido, y luego de la cual respondió satisfactoriamente a las preguntas que le fueron formuladas por el Jurado, todo ello conforme a lo dispuesto en la Ley de Universidades y demás normativas vigentes de la Universidad Central de Venezuela. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobarlo con la nota de 18 puntos.

En fe de lo cual se levanta la presente acta, en Caracas el 22 de octubre de 2019 a las 10 am, dejándose también constancia de que actuó como coordinador del Jurado el Profesor Tutor Franklin Sandoval.

Prof. Franklin Sandoval
Tutor

Profa. Keyla Rivas

Jurado

Prof. Franky Uzcátegui

Jurado

DEDICATORIAS

El presente trabajo está dedicado a mi familia por haber sido mi apoyo a lo largo de toda mi carrera universitaria y a lo largo de mi vida.

Primeramente, quiero agradecer a mi familia, entre ellos, a mis difuntos abuelos Antonio Pereira, María Emilia Da Costa y Adosinda Quintairos, que siempre desearon verme bajo las Nubes de Calder y que no pudieron, debido a enfermedad, a mi único abuelo vivo, Álvaro Ferreira, que aunque este fuera del país, siempre está pendiente de mi graduación y aun cuando no pueda estar en ella, sé que se alegra al saber que logre la meta que siempre les prometí.

A mi Madre que siempre estuvo allí, en las buenas y en las malas y me cuidó con mucho amor, a mi papa que, si bien no siempre estuvo a mi lado, sé que me quiere y que sin ellos no habría cabida para esta meta y gracias a ellos hoy tengo la oportunidad de alguna forma retribuirles todo lo que han hecho por mí.

A mi hermana que, aunque nos la pasamos peleando, siempre puedo confiar en ella, así como ella en mí, en donde este.

Y a todas las personas especiales que me acompañaron en esta etapa, aportando a mi formación tanto profesional y como ser humano.

AGRADECIMIENTOS

Para terminar este Trabajo Especial de Grado, muchas personas sirvieron de apoyo y ayuda para que se realizara con total éxito, por tal motivo quiero agradecerles el hecho de haberme ayudado, así sea en un mínimo instante o durante toda la carrera.

Gracias a mis amigos por acompañarme durante esta aventura, por estar ahí en los momentos más difíciles, cuando eran más necesarios. El impulso que me dieron fue vital para culminar la carrera exitosamente.

Agradezco a los profesores de la Facultad de Ciencias de la Universidad Central de Venezuela y a la Universidad, por brindar el conocimiento, las metodologías y, sobre todo, crear una forma de pensar distinta a muchas otras universidades que me brinda una resolución de problemas esencial para ser un profesional exitoso en el área de la computación e impulsarme a ser cada día mejor con la enseñanza brindada.

Gracias a mi tutor, Franklin Sandoval, por no solo ser profesor y amigo durante la carrera y proyecto de grado, sino también excelente persona, profesional y futuro colega. Gracias por estar en todo momento y por darme la fuerza y la motivación de saber que siempre hay luz al final del túnel. De verdad gracias.

Y no menos importante, gracias a Lear Express® por darme la oportunidad de trabajar para ellos y poder comenzar con esta aplicación

Índice General

RESUMEN	II
ACTA	¡ERROR! MARCADOR NO DEFINIDO.
DEDICATORIAS.....	IV
AGRADECIMIENTOS	V
ÍNDICE	VI
TABLA DE ILUSTRACIONES	IX
ÍNDICE DE TABLAS.....	X
INTRODUCCIÓN	1
CAPÍTULO 1 PLANTEAMIENTO DEL PROBLEMA.....	3
1.1. SITUACIÓN ACTUAL Y PLANTEAMIENTO DEL PROBLEMA.....	3
1.2. PROPUESTA DE VALOR.....	4
1.3. SOLUCIÓN PROPUESTA	5
1.4. ALCANCE	5
1.4.1. <i>Usuarios</i>	5
1.4.2. <i>Arquitectura</i>	6
1.5. TECNOLOGÍAS PARA EL DESARROLLO DE LA SOLUCIÓN PROPUESTA	7
1.6. OBJETIVO GENERAL	4
1.7. OBJETIVOS ESPECÍFICOS	4
CAPÍTULO 2 MARCO CONCEPTUAL	8
2.1. ANTECEDENTES ASOCIADOS	8
2.2. ORGANIZACIONES	8
2.3. GESTIÓN	8
2.4. SISTEMA DE GESTIÓN	9
2.5. PRODUCTO.....	9
2.6. PAQUETE	9
2.7. DISTRIBUCIÓN.....	10
2.8. ENVÍO.....	10
2.9. CURRIER	10

2.10. SISTEMAS DE INFORMACIÓN	10
2.11. SISTEMA GESTOR DE BASE DE DATOS	11
2.12. APLICACIONES WEB	11
2.13. PORTAL WEB.	11
2.13.1. <i>Funcionalidades De Un Portal Web</i>	11
2.13.2. <i>Objetivo De Un Portal Web</i>	12
2.13.3. <i>Características De Los Portales Web</i>	12
2.13.4. <i>Clasificación De Los Portales Web</i>	12
2.14. ARQUITECTURA CLIENTE - SERVIDOR	14
2.15. MARCO DE TRABAJO (FRAMEWORK).....	15
2.16. PATRÓN DE MODELO-VISTA-CONTROLADOR (MVC)	15
2.17. TECNOLOGÍAS PARA EL DESARROLLO DE LA SOLUCIÓN PROPUESTA.....	17
2.17.1. <i>Front End</i>	17
2.17.2. <i>Back End</i>	18
CAPÍTULO 3 MARCO METODOLÓGICO	20
3.1. METODOLOGÍA PARA EL DESARROLLO DE LA APLICACIÓN	20
3.2. TEORÍA DE SCRUM	21
3.2.1. <i>Transparencia:</i>	21
3.2.2. <i>Inspección:</i>	21
3.2.3. <i>Adaptación</i>	21
3.3. EL EQUIPO SCRUM	22
3.3.1. <i>Equipo de desarrollo (Development Team)</i>	22
3.3.2. <i>Facilitador (Scrum Master)</i>	22
3.3.3. <i>Dueño del producto (Product Owner)</i>	22
3.4. ARTEFACTOS.....	23
3.5. EVENTOS	23
3.6. RAZONES QUE JUSTIFICAN LA SELECCIÓN DEL MARCO DE TRABAJO SCRUM.....	24
CAPÍTULO 4 MARCO APLICATIVO.....	26
4.1. REQUERIMIENTOS FUNCIONALES	26
4.2. REQUERIMIENTOS NO FUNCIONALES	26

4.3. USUARIOS POTENCIALES	26
4.4. PERFILES DE USUARIO	27
4.4.1. <i>Administrador</i>	27
4.4.2. <i>Usuario</i>	27
4.5. APLICACIÓN DE LA METODOLOGÍA SCRUM	27
4.5.1. <i>Lista de objetivos (Product Backlog)</i>	27
4.5.2. <i>Lista de tareas de la iteración (Sprint Backlog)</i>	28
4.6. TECNOLOGÍAS Y HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO	30
4.6.1. <i>Análisis del modelo de datos</i>	31
4.7. PRINCIPALES INTERFACES DISEÑADAS.....	35
4.8. PRUEBAS REALIZAS Y RESULTADOS	43
4.8.1. <i>Pruebas realizadas en el servidor</i>	43
4.8.2. <i>Pruebas realizadas en el sistema administrativo</i>	44
4.8.3. <i>Pruebas de aceptación</i>	44
4.9. RESULTADOS ASOCIADOS A LOS REQUERIMIENTOS FUNCIONALES	48
4.10. RESULTADOS ASOCIADOS A LOS REQUERIMIENTOS NO FUNCIONALES.....	49
CONCLUSIONES Y RECOMENDACIONES	50
CONCLUSIONES.....	50
RECOMENDACIONES	51
REFERENCIAS BIBLIOGRÁFICAS Y DÍGITALES	52

Tabla de Ilustraciones

Ilustración 1. Arquitectura General Propuesta.....	6
Ilustración 2. Portal Horizontal.....	13
Ilustración 3. Portal Vertical.....	13
Ilustración 4. Portal Diagonal.....	14
Ilustración 5. Arquitectura Cliente-Servidor.....	15
Ilustración 6. Comunicación entre componentes MVC.....	16
Ilustración 7. Ciclo de Vida de Scrum.....	25
Ilustración 8. Modelo de la base de datos.....	33
Ilustración 9. Modelo de la base de datos (Parte 2).....	34
Ilustración 10. Registro en el sistema.....	35
Ilustración 11. Inicio del sistema administrativo.....	36
Ilustración 12. Inicio del sistema.....	37
Ilustración 13. Inicio del sistema (usuario registrado).....	38
Ilustración 14. Lista de Trackings prealertados.....	38
Ilustración 15. Consulta de envíos.....	39
Ilustración 16. Lista de autorizados.....	39
Ilustración 17. Registro de Autorizados.....	39
Ilustración 18. Lista de autorizados.....	40
Ilustración 19. Inicio del sistema (usuario administrador).....	40
Ilustración 20. Valijas.....	41
Ilustración 21. Paquetes.....	41
Ilustración 22. Gráfico de guías.....	41
Ilustración 23. Permisos.....	42
Ilustración 24. Roles.....	42
Ilustración 25. Usuarios.....	43
Ilustración 26. Resultado de prueba de servidor.....	43
Ilustración 27. Resultado de prueba de funcionalidad.....	44
Ilustración 28. Instrucciones suministradas para las pruebas de aceptación.....	45
Ilustración 29. Pruebas de aceptación (1).....	45
Ilustración 30. Pruebas de aceptación (2).....	46
Ilustración 31. Pruebas de aceptación (3).....	46
Ilustración 32. Pruebas de aceptación (4).....	47
Ilustración 33. Pruebas de aceptación (5).....	47
Ilustración 34. Pruebas de aceptación (6).....	48
Ilustración 35. Pruebas de aceptación (7).....	48

Índice de Tablas

Tabla 1. Perfiles de Usuario.....	27
Tabla 2. Lista de Sprint realizados en el desarrollo del sistema	28
Tabla 3. Tablas involucradas en la estructura de la base de datos	32

INTRODUCCIÓN

En la actualidad, pequeñas y grandes empresas invierten recursos monetarios y talento humano para la automatización de procesos dentro de las organizaciones, aplicando diversas estrategias de gestión como; la automatización de los procesos de negocios, con la finalidad de adquirir reportes automáticos, para una mejor toma de decisiones

Esto, sin importar si se trata de una empresa de alto nombre o una empresa que está iniciando, el definir las actividades por roles y áreas para automatizar los procesos, permite hacer más eficiente la operación logrando reducir costos.

Por este motivo, esta corriente de automatización de procesos surge ante la necesidad de centralizar la información y la obtención del control sobre los procesos de negocio, esto para apoyar en la correcta toma de decisiones, aumentar la productividad y mejorar continuamente la efectividad de los mismos.

Con el uso de aplicaciones web se busca transmitir las tareas recurrentes realizadas por una persona a una unidad de cómputo. Estas pondrán a disposición de los usuarios diversas herramientas que serán almacenadas y ejecutadas en un servidor web y serán accedidas mediante el uso del navegador.

La adopción de este tipo de tecnología trae consigo enormes beneficios como: portabilidad, escalabilidad, disponibilidad, control, usabilidad y seguridad de todos los procesos de negocio involucrados.

Sin embargo, la tecnología ha dejado de ser inalcanzable o en su defecto, un privilegio en el mundo, su uso se ha convertido en un elemento fundamental en el ámbito personal y empresarial. Es por ello, que las organizaciones deben ser rápidas y eficientes con todos sus recursos, teniendo a disposición tecnología con sistemas innovadores y adaptables a las necesidades.

Las organizaciones aceleran sus procesos manteniendo competitividad en el mercado, reduciendo el trabajo manual y aumentando la productividad de sus colaboradores apoyándose en los sistemas de información (y aplicaciones web) para lograr sus objetivos de negocio. Provocando la comunicación entre empleados y ejecutivos haciendo posible contar con otros métodos de notificación como correo electrónico.

Para la elaboración de este proyecto se plantea el desarrollo de una aplicación que permita la gestión, seguimiento de información de los paquetes enviados desde Estados Unidos, mediante Laravel como tecnología de desarrollo, PostgreSQL como el Sistema Manejador de Base de Datos y Apache como servidor.

Con base en la problemática planteada, se presenta la propuesta de Trabajo Especial de Grado, a partir de la cual se propone desarrollar e implementar una aplicación para la gestión de dicho proceso de forma automatizada y centralizada.

En el primer capítulo, se hablará de la situación actual, el planteamiento del problema, estableciendo la propuesta de valor, estableciendo la solución propuesta, con sus respectivos alcances, así mismo los tipos de usuarios que usaran dicha aplicación, estableciendo la arquitectura, mencionando las tecnologías para el desarrollo y para finalizar los objetivos, tanto el general como los específicos.

En el segundo capítulo, se detallará el marco conceptual, que se define el antecedente del proyecto en Lear Express®, así como los conceptos necesarios para el entendimiento del proyecto, ya sean teóricos como tecnológicos, incluyendo las tecnologías para el desarrollo.

En el siguiente capítulo, se podrá estudiar el marco metodológico, la metodología ágil usada, involucrando cada uno de los aspectos teóricos.

Y para culminar, se observará el marco aplicativo, en donde se atenderá los requerimientos funcionales y no funcionales, como se aplicará la metodología usada, analizando el modelo de datos, mostrando las principales interfaces y pruebas realizadas.

CAPÍTULO 1 PLANTEAMIENTO DEL PROBLEMA

1.1. Situación Actual y Planteamiento del Problema

Las organizaciones son estructuras sociales de índole gubernamental, económica, social o político. Abarcan departamentos con niveles y funciones específicas, cuentan con recursos materiales y humanos, a fin de alcanzar eficiencia dentro de los planes y objetivos propuestos. Tienen éxito cuando las personas se comunican y trabajan de manera coordinada para lograr su misión. Establecen normas para el cumplimiento de sus propósitos en el cual la planificación, la dirección y el control hacen parte de la gestión de la organización.

Algunas empresas consideran que el avance tecnológico facilita la comunicación persona a persona y la inversión en recursos tecnológicos permite replantear e integrar los procesos, reducir costos e implementar una cultura sólida. Para (Gibson, 1997) señala la tecnología, las opciones estratégicas, la estructura organizacional, los procesos y la cultura, como determinantes en el éxito organizacional.

Por su parte (Rodríguez, 1994), asume la tecnología como un conjunto de saberes inherentes al diseño y concepción de los instrumentos (artefactos, sistemas, procesos y ambientes), creados por el hombre a través de la historia para satisfacer sus necesidades, requerimientos personales y colectivos. Considera que tiene un significado general desde diferentes aspectos culturales (objetivos, valores y códigos éticos, creencia en el progreso, conciencia y creatividad); organizacional (actividad económica y técnica, actividad profesional, usuarios y consumidores) y técnico (conocimiento, destreza, herramientas, maquinas).

Las aplicaciones Web se codifican en lenguajes soportados por los navegadores web y confía la ejecución al navegador accediendo a un servidor web a través de la red (internet o intranet) sin necesidad de descargarla en su ordenador ni ejecutar archivos. Además, no ocupa espacio en el disco duro y no tiene que esperar que el servidor notifique si existe o no algún tipo de actualización, porque la realiza de manera automática. Es importante señalar que una institución tiene éxito en su gestión cuando se apoya en un personal técnico, financiero y tecnológico en busca de mejora.

Referente a los envíos de paquetes, están en boga en la actualidad, ya que es más fácil y a veces más económico el comprar cosas en páginas extranjeras, que, en Venezuela, ya sea para su uso o venta, se recomienda emplear tecnología de vanguardia que involucre aplicaciones web que permitan tener una organización y planificación adecuada para la distribución.

Actualmente en Lear Express ® se tiene una aplicación desarrollada por una empresa de consultoría externa en ASP classic, sin supervisión directa, lo que ocasionó un manejo erróneo de la base de datos, siendo dificultoso manejar grandes cantidades de paquetes, teniendo en cuenta que debido a esto la

aplicación se volvió inestable y con gran cantidad de tiempos de carga, además de que las tablas de la base de datos no estaban normalizadas, trayendo como consecuencia:

“Inexactitud de los sistemas de bases de datos. Ralentización de los procesos. Ineficiencia en las operaciones.” (PowerData, 2016)

Además de que era muy complicado querer agregar un nuevo módulo, por todas las incongruencias que tenía esta aplicación, sin que esta usara un patrón de diseño de programación, cayendo en uno de los más conocidos anti patrones, el código espagueti, un programa, aplicación o sistema que contiene muy poca estructura de software, comprometiéndola hasta el punto de que carece de claridad, incluso para el desarrollador original, (Source Making, s.f.) debido a que no es modular, haciendo que un cambio, modifique varios llamados de estos.

Luego de haber tenido un mal inicio, se decidió que el personal de la empresa se haría cargo de la aplicación aunque esta no estaba 100% completa y que no estaba calificado para desarrollar una aplicación de tal envergadura, debido a esto, se tomaron la estructuras de base de datos y empezaron a programar en ASP Classic, lenguaje que no poseían tantos conocimientos, se iban agregando cosas sin una buena dirección del equipo de desarrollo por lo que no mantenían buena comunicación con los demás miembros de equipo, creando muchas funciones sin prestar atención a las que ya estaban desarrolladas o si estas afectaban en otros módulos desarrollados, sin reutilizar el código existente, lo que generaba tiempos de carga altos. Trayendo como consecuencia que, al querer hacer una modificación en un módulo, la mayoría de las veces, había que hacer dichos campos en varios lugares del código, porque una misma funcionalidad está repetida en varios sitios.

1.2. Objetivo General

Desarrollar una aplicación web de gestión de paquetes y envíos internacionales que garantice la distribución a nivel nacional.

1.3. Objetivos específicos

- Definir los requerimientos funcionales y no funcionales necesarios en el desarrollo de una aplicación web para la gestión de paquetes que garantice la distribución a nivel nacional.
- Generar los modelos y diagramas que se ajusten al proceso de gestión de paquetes y envíos.
- Desarrollar las interfaces y módulos de la propuesta de la aplicación web.
- Realizar un conjunto de pruebas funcionales.

1.4. Propuesta de Valor

El valor que aporta la propuesta a los clientes, es el mantenerlos informados del estado del paquete, de esta forma, sabrán en qué lugar se encuentra dicha carga, dando más seguridad a los nuevos usuarios o a los recurrentes, esto dentro de un mejor diseño y más intuitivo al usar dicho sistema. Transmitiendo con eso, confianza.

1.5. Solución Propuesta

La automatización juega un papel importante en el crecimiento de la empresa ya que esta incentiva la necesidad de estar actualizándose constantemente, es por eso que, se decidió cambiar el lenguaje en el que se estaba programando, debido a que, el anterior era muy limitante, por la forma en la que estaba hecho y porque las personas que lo hicieron eran externas a la empresa, por lo que era complicado hacer cambios, no solo por la falta de comunicación entre la empresa y el outsourcing, sino también el modo de elaboración del sistema, existían muchos anti patrones en el sistema.

Es por estos motivos, que se decidió elaborar un sistema en PHP, con Laravel, en el que se tenga más control internamente, asimismo poder actualizarlo constantemente, además, agregarle funcionalidades y agregarle más vistosidad al sistema, para que sea más intuitivo y con un diseño más agradable

1.6. Alcance

El alcance del sistema propuesto en este trabajo especial de grado se basa en crear un sistema de gestión, asegurando su funcionalidad y operatividad, dividido en dos subsistemas, en el primero el usuario podrá crear prealerta de paquetes, realizar seguimiento, crear autorizaciones; y en el sistema administrativo, se podrá registrar los paquetes, las valijas, actualizar el estatus de los paquetes y revisar las autorizaciones. La aplicación se diseñará de la forma más generalizada posible, de manera que pueda ser utilizada para gestionar cualquier envío. Sin embargo, la solución se limitará al diseño y despliegue de las necesidades y requerimientos de la empresa Lear Express como caso de prueba para evaluar la efectividad y éxito de la aplicación web.

La aplicación web a desarrollar:

- Permitirá a los clientes registrarse e iniciar sesión en la aplicación.
- Permitirá realizar seguimiento de paquetes.
- Permitirá visualizar un perfil resumen con las principales características de la persona.
- Permitirá a las personas actualizar la información de su hoja de perfil.
- Permitirá a los clientes registrar solicitudes de envíos.
- Permitirá al gerente visualizar la cantidad total de envíos activos.
- Permitirá a los despachadores crear guías de despacho en formato PDF.
- Proveerá a los involucrados recibir notificaciones vía correo electrónico acerca de las operaciones que realicen en la aplicación.

1.6.1. Usuarios

Existen dos tipos de usuarios:

- **Usuario Registrado:** persona que realizara solicitudes de envíos, puede registrar y consultar los estatus de guías y actualizar su hoja de perfil.
- **Usuario Administrador:** posee los permisos necesarios para gestionar a usuarios y analistas.

1.6.2. Arquitectura

La arquitectura de la propuesta de solución a desarrollar en la aplicación será como se muestra en la ilustración 1.

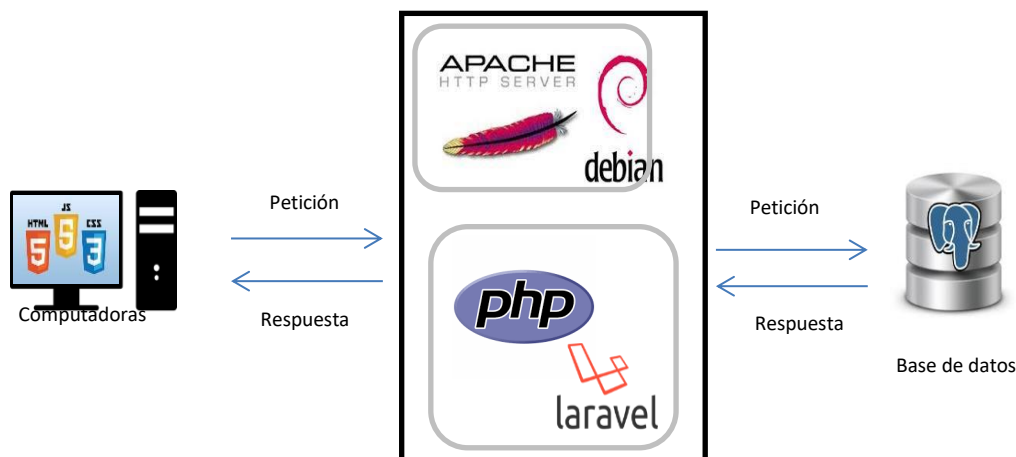


Ilustración 1. Arquitectura General Propuesta.

Fuente: Elaborada por el autor

La solución estará conformada por:

- Modulo cliente administrador.
- Modulo cliente usuario.

Requerimientos Funcionales

- Registrarse e iniciar sesión en la aplicación.
- Registrar envíos.
- Registrar autorizaciones.
- Descargar guía de pago.
- Enviar notificaciones vía correo electrónico a los usuarios.
- Generar reporte de número total de envíos realizados, así como el porcentaje de aumento o disminución con respecto al mes anterior.
- Garantía de que los datos están siendo tratados de forma privada, de manera de que no puedan ser obtenido por agentes externos

Requerimientos No Funcionales

- La aplicación debe tener el mismo comportamiento en diversos navegadores y en diversos dispositivos (computador, tableta, televisor, teléfono inteligente).
- El sistema debe ser capaz de manejar excepciones, para ofrecer respuestas oportunas al usuario en caso de que ocurran fallas.
- Interfaces que sean de fácil usabilidad y accesibilidad, que no requieran que el usuario posea conocimientos avanzados en computación.
- El tiempo de respuesta de la aplicación debe ser apropiado.

La arquitectura propuesta será de tipo cliente-servidor. El cliente iniciará la comunicación realizando una petición y el servidor se encargará de enviar la respuesta. En caso de un nuevo registro de usuario, solicitud, cliente o producto en la aplicación, el sistema accederá a la base de datos para almacenar la información, en caso de involucrar alguna información referente a usuario, solicitud, cliente, producto, guía de despacho, el sistema accederá a la base de datos donde se encuentra almacenada esa información.

1.7. Tecnologías para el desarrollo de la solución propuesta

Para la implementación de la aplicación propuesta se consideran las siguientes tecnologías:

- Sistema Operativo: Ubuntu 18.10.
- Servidor Web: Apache 2.
- Lenguaje de programación del lado del Cliente: HTML5, CSS3 y JavaScript 5.
- Sistema Manejador de Bases de Datos: PostgreSQL.
- Lenguaje de programación del lado del Servidor: PHP 7.
- Framework del lado del Servidor: Laravel 5 con Eloquent.
- Manejador de Versiones: Git (Bitbucket).

Aplicación Web:

- PHP como lenguaje de programación para la implementación de la aplicación Web por la sencillez, robustez, flexibilidad y por la vasta experiencia que posee las personas que se encargaran del mantenimiento de la misma
- Laravel como framework de PHP basadas en web que simplifica el desarrollo de interfaces de usuario, manejo de eventos, definición de esquemas de navegación y accesibilidad.
- En la parte visual, se realizará con HTML ayudado con los estilos de CSS y Bootstrap. Haciéndolo muy ligero.
- PostgreSQL como base de datos relacional que por su característica principal Hot Standby. Característica que permite que los usuarios puedan conectarse al servidor y ejecutar búsquedas en la base de datos mientras se encuentra en modo de recuperación, haciendo que las búsquedas regresen con resultados.

CAPÍTULO 2 MARCO CONCEPTUAL

En este capítulo se explicarán antecedentes, teorías, conceptos, procesos y plataforma tecnológica relacionados con el problema a resolver y con la solución dirigida a la gestión de envío de paquetes para garantizar la distribución a nivel nacional. Se plantea una reseña a las plataformas que brindan una solución.

2.1. Antecedentes asociados.

El primer proyecto corresponde al desarrollado en Lear Express (2014) por una empresa outsourcing, en el que se realizó la aplicación web, que hasta ahora está vigente, este proyecto se desarrolló en ASP Classic usando lo que en aquel momento era un patrón, código espagueti, pero en la actualidad es un anti patrón, por lo que es complicado realizar ajustes o cambios en dicha aplicación. (Lear Express, s.f.)

2.2. Organizaciones

Una organización es un sistema constituido con el fin de alcanzar un objetivo en común. Puede contar con subsistemas internos, que tengan asignados tareas específicas. Sus miembros se ordenan jerárquicamente, estableciendo reglas, apuntado un objetivo conocido; crean poder, lo tienen y lo transmiten, generando una cultura propia. Cuentan con determinados recursos (económicos, humanos, tecnológicos y naturales) para poder alcanzar los objetivos planteados de manera satisfactoria.

Según (Abrahamsson, 1993) las organizaciones son estructuras configuradas según un plan diseñado por una persona, grupo, o clase con el deliberado y expreso propósito de conseguir ciertos objetivos. Es empleada como recurso por distintos agentes como medio para realizar esfuerzos racionales y planificados. Su existencia es la realización de un trabajo, o de un proceso de producción.

2.3. Gestión

Es una diligencia, entendida como trámite necesario para conseguir algo o resolver un asunto, habitualmente de carácter administrativo o que conlleva documentación. También un conjunto de acciones u operaciones relacionadas con la administración y dirección de una organización. Se utiliza para hablar de proyectos o en general de cualquier tipo de actividad que requiera procesos de planificación, desarrollo, implementación y control. La gestión a menudo es incluida como un factor de producción junto con maquinaria, materiales y dinero. De acuerdo al gurú de la gestión, Peter Drucker (1909-2005), las tareas básicas de la gestión incluyen tanto mercadeo como innovación (Sáez, s.f.).

2.4. Sistema de Gestión

Un sistema de gestión está definido como el “sistema para establecer la política y los objetivos y para lograr esos objetivos” (ISO 9001 calidad, 2019). Un sistema de gestión es una estructura o modelo de administración eficaz y eficiente que busca mejorar el funcionamiento de una organización. Incluye un proceso de ideación, planeación, implementación y control. Ofrecen pautas, estrategias y técnicas para optimizar los procesos y los recursos de una entidad. Se usan en organizaciones de carácter empresarial y abordan diferentes ámbitos como la gestión de la calidad y la rentabilidad. La implantación de este sistema de gestión permite introducir mecanismos orientados a la renovación y adaptación a la realidad de una organización y al entorno en que se desarrolla su actividad. Se basan en cuatro pilares:

- **Planificar:** es la fijación y programación de las metas u objetivos de la empresa a nivel estratégico, así como el establecimiento de los recursos humanos y técnicos necesarios para llevar a cabo la misión prevista de la forma más eficaz posible.
- **Organizar:** se trata de la asignación y coordinación de las diferentes tareas, definiendo quién, cómo y cuándo se van a ejecutar.
- **Dirigir:** relacionado con la figura de los líderes o ejecutivos, esta función consiste en motivar, impulsar y cohesionar a los distintos miembros de la organización para que desarrollen las funciones asignadas.
- **Controlar:** radica en la supervisión del trabajo, analizando las fortalezas y debilidades de la compañía, con el objetivo de subsanar los defectos que puedan surgir durante el proceso de gestión empresarial.

2.5. Producto

Es el resultado de una serie de procedimientos para armar, construir, fabricar y elaborar algún objeto con una función útil para las personas. En el marketing se establece que los productos son los elementos fabricados con el destino de satisfacer al cliente es fundamental en las empresas manufactureras, las cuales producen grandes cantidades de productos con el fin de cubrir una demanda por parte del consumidor. La distribución de los productos debe causar un impacto en la sociedad favorable para la realización de una cadena productiva con el fin de mantener al cliente satisfecho con lo que se le está ofreciendo.

Según (Santesmases), un producto “es cualquier bien material, servicio o idea que posea un valor para el consumidor o usuario y sea susceptible de satisfacer una necesidad”.

2.6. Paquete

“Un paquete es envoltorio que contiene algo o un conjunto de cosas, también llamado bulto.” (Busca palabra, s.f.) Este conjunto de objetos se ata o se envuelve formando un bloque para de esta forma ser transportados.

2.7. Distribución

La distribución es el conjunto de acciones realizadas por las empresas para colocar el producto elaborado por un fabricante hasta que es comprado por el cliente. A estas empresas se les conoce como canales de distribución. Según (Isabel Milio Balanza, 2000), los canales de distribución son todas aquellas empresas a través de las cuales se facilita la información y acceso del comprador al producto. Por su parte, (Lamb Charles, 2002), indica que la distribución constituye uno de los pilares básicos de la función de marketing en la economía, estableciendo que la distribución es “la organización del intercambio voluntario y competitivo que asegure un ajuste eficiente entre la oferta y demanda de productos y servicios”.

Esto implica según Santesmases (Santesmases Mestre, 2012), un conjunto de actividades tales como: información, promoción y presentación del producto en el punto de venta, a fin de estimular su adquisición. Para (Escribano, p.160); es la ruta por la que circulan los productos desde su creación en el origen hasta su consumo o uso en el destino final. Está formado por el conjunto de personas u organizaciones que facilitan la circulación del producto hasta llegar a manos del consumidor (Mendoza, 2017).

2.8. Envío

Es el movimiento físico del producto de un punto a otro, como el movimiento de mercancías desde el almacén hasta el cliente. El proceso de envío sigue la fabricación y el embalaje de los productos y será controlado por una empresa de envío o logística.

2.9. Currier

Un Courier, es un ente que envía documentos o paquetes de un tamaño y/o peso limitado a un determinado costo, que puede ser distribuido a nivel nacional e internacional. (Courier, 2011), también llamada empresa logística.

2.10. Sistemas de Información

Es un conjunto de elementos como: personas, datos, actividades o técnicas de trabajo y recursos materiales en general, para la administración de datos e información. Para (Laudon, 2014); es un conjunto de componentes interrelacionados que recolectan (o recuperan), procesan, almacenan y distribuyen información para apoyar los procesos de toma de decisiones y control en una organización. Para usar los sistemas de información con efectividad se requiere de tres dimensiones: organizaciones, administración y tecnología. La dimensión organizacional involucra la jerarquía de la organización, las especialidades funcionales y los procesos de negocios. La dimensión de administración incluye aspectos como liderazgo, estrategia y comportamiento gerencial. La dimensión de tecnología consiste en hardware y software de computador, tecnología de almacenamiento de datos y tecnología de redes y telecomunicaciones.

2.11. Sistema Gestor de Base de datos

Es una aplicación informática que permite a los usuarios definir, crear, mantener y consultar sus bases de datos. Según (Castaño y Piattini, 1993) Un conjunto coordinado de programas, procedimientos, lenguajes, etc., que suministra, tanto a los usuarios no informáticos como a los analistas, programadores, o al administrador, los medios necesarios para describir, recuperar y manipular los datos almacenados en la base, manteniendo su seguridad.

2.12. Aplicaciones Web

La aplicación web es un software que se codifica en un lenguaje soportado por los navegadores web. Se ejecuta parcialmente en un servidor remoto, al que se accede a través de internet o intranet mediante un navegador web. La interfaz de usuario es accesible desde un cliente web, normalmente un navegador web. Según (Luján Mora, 2002) las aplicaciones web son herramientas donde los usuarios pueden acceder a un servidor web a través de la red mediante un navegador determinado.

Para (Niño, 2010) las aplicaciones web son aplicaciones a las que se accede mediante un navegador y están alojadas en servidores dentro de una intranet o en internet. Las de escritorio sustituyen a los programas que se instalan en los ordenadores y que realizan tareas como organizar agendas, calendarios, leer correo electrónico, etcétera. Son útiles en la empresa y ayudan a solucionar los problemas y dependencias que presentan los sistemas, ya que pueden utilizarse independientemente del sistema operativo o dispositivo que se use para acceder a las mismas. Además, numerosas soluciones de software libre lo utilizan para su desarrollo.

2.13. Portal Web.

Un Portal web es un sitio web cuya característica fundamental es la de servir de Puerta de entrada para ofrecer al usuario, de forma fácil e integrada, el acceso a una serie de recursos y de servicios relacionados a un mismo tema. (Andrade Fatima P, 2011).

2.13.1. Funcionalidades De Un Portal Web

Un portal web debe proveer funciones como lo son la seguridad, publicación y herramientas de colaboración. Cuenta con una serie de funcionalidades que permiten a los usuarios, personalizar y organizar su propia vista del portal, manejar perfiles y compartir documentos.

Se puede decir que un portal ofrece servicios para la navegación en el Internet, logrando incrementar la intensidad de tráfico en el mismo.

Un portal debe tener (necesariamente) un tema y este tema debe estar referido a algún tema de interés de un grupo importante, pero especializado de internautas. Dado que un portal presenta la información de una forma fácil e integrada, todo portal debe tener una forma simple de acceder a toda (no a una parte) la información referida al tema del mismo. Toda esta información no necesariamente está contenida dentro

del mismo portal, porque el portal, normalmente, se encarga de centralizar enlaces en una forma fácil y organizada que facilite la navegación dentro de un tema específico en la red.

2.13.2. Objetivo De Un Portal Web

El objetivo principal de un portal web es ayudar a los usuarios a encontrar lo que necesitan sin salir del mismo fidelizándoles e incentivándoles a utilizarlo de forma continuada.

Los 3 pilares fundamentales de un **Portal** para atraer la atención del usuario y son:

1. **Información** (Buscadores, directorios, noticias, catálogos y servicios).
2. **Participación** (E-mail, foros, chat, redes sociales).
3. **Comodidad** (brindar la mayor cantidad de información en un solo espacio y tenerlo todo a mano).

De esa forma el Portal web gana tráfico constante y se autofinancia con publicidad.

2.13.3. Características De Los Portales Web

- Ofrecen servicios gratuitos y de utilidad al internauta.
- Poseen recursos informativos y organizados.
- Interacción personalizada con los servicios que ofrece el sitio web.
- Integración de herramientas de colaboración como grupos de trabajos, comunidades, foros y chats.
- Cuentan con un motor de búsqueda, bien sea de recursos propios o ajenos.

2.13.4. Clasificación De Los Portales Web

Según la autora (González Jessenia M, 2010), explica la clasificación de portales web de la siguiente manera:

Portales Horizontales

Es un portal con contenidos de carácter general orientado a todo tipo de usuario. Incluye servicios como: Comunidades Virtuales, Chat, correos, Foros, Noticias, Buscadores e información.

Son portales para usuarios no muy experimentados en el mundo de internet a los que se les brinda una gran gama de información y servicios sin obligarles a navegar por varios sitios para obtenerlos.

Estos sitios han prácticamente desaparecido en detrimento de los portales especializados. Como, por ejemplo: Terra, AOL, AltaVista, UOL, Lycos, Yahoo y MSN.



Ilustración 2. Portal Horizontal

Fuente: <https://espanol.yahoo.com/>

Portales Verticales

Es un sitio web que provee de información y servicios a un sector en particular, con contenidos concretos y centrados en un tema como puede ser un portal de música, empleo, inmobiliario, un portal de finanzas, arte o deportes.

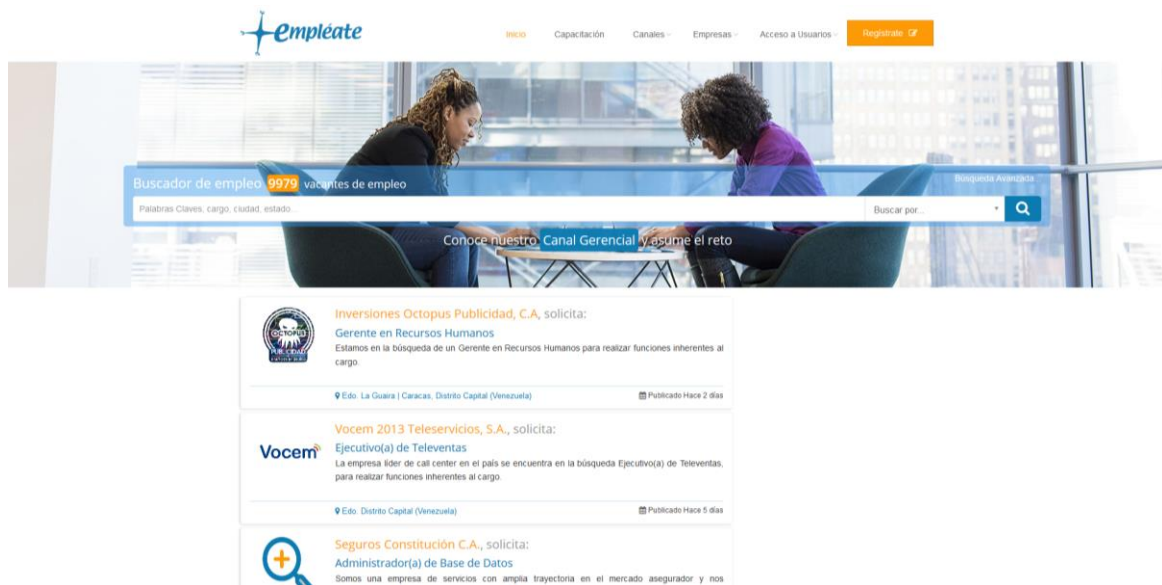


Ilustración 3. Portal Vertical

Fuente: <https://www.empleate.com/venezuela/>

Portales Diagonales

Se trata de una mezcla entre el portal horizontal y el vertical. Suelen ser redes sociales como Facebook, LinkedIn, Flickr, Youtube, entre otras. Complementadas con contenidos y utilidades dirigidas a un público en concreto.

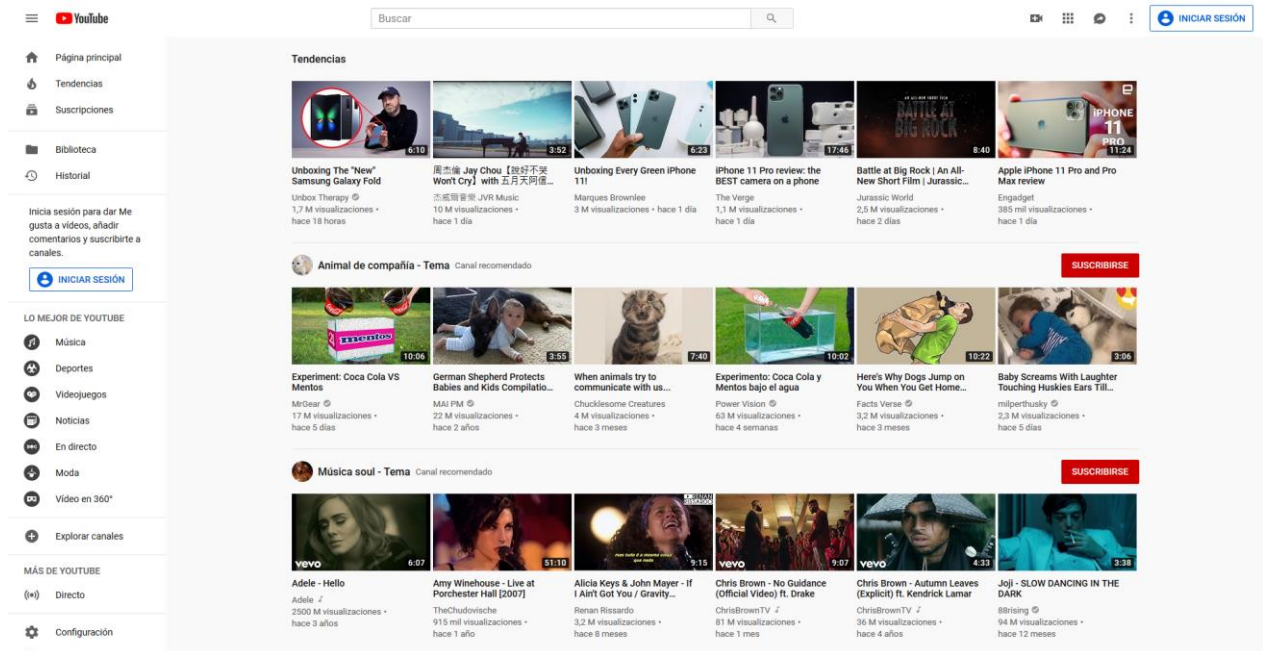


Ilustración 4. Portal Diagonal

Fuente: <https://www.youtube.com/>

2.14. Arquitectura Cliente - Servidor

Para Shakirat (2014), se puede definir como una arquitectura de software compuesta por el cliente y el servidor, por lo que los clientes siempre envían solicitudes mientras el servidor responde a las solicitudes enviadas. El servidor cliente proporciona una comunicación entre procesos porque implica el intercambio de datos tanto del cliente como del servidor, por lo que cada uno de ellos realiza diferentes funciones.

Esta arquitectura consiste en que uno o varios clientes realizan peticiones a un programa (el servidor) que entrega las respuestas a las solicitudes realizadas. Los clientes y los servidores pueden estar conectados en una red local, una red amplia o la red mundial.

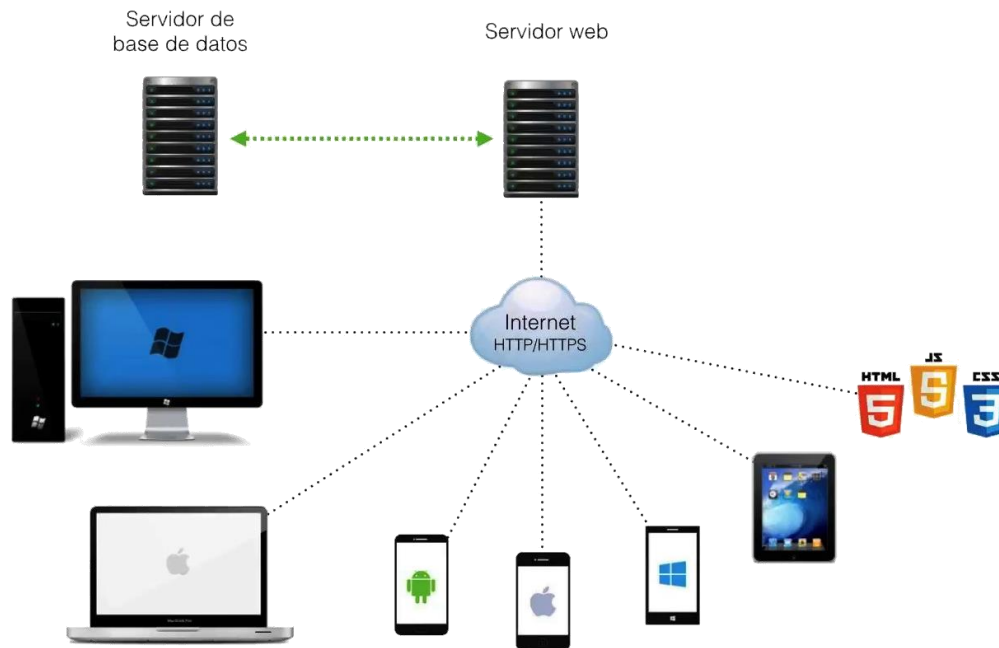


Ilustración 5. Arquitectura Cliente-Servidor

Fuente: <https://plataforma.keepcoding.io/courses/>

2.15. Marco de trabajo (Framework)

Hace referencia a una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. Se puede considerar como una aplicación genérica incompleta y configurable a la que se le puede añadir las últimas piezas para construir una aplicación concreta. Según (Gamma 1995), el framework determina la arquitectura general. Este es un buen enfoque, ya que el framework se encarga de definir la estructura general, sus peticiones en clases y objetos. Todos estos parámetros son definidos por el framework, evitando que el usuario tenga que definirlos y se puedan enfocar en cosas específicas de su aplicación. Además, captura las decisiones de diseño que son comunes a su dominio de aplicación. Es importante mencionar que las aplicaciones que se desarrollan a partir de un framework, está ligada al mismo, por eso las aplicaciones deben evolucionar y crecer al mismo tiempo que crece el framework, debido que un cambio en alguna interfaz del mismo significara un cambio de la aplicación, dependiendo de qué tan drástico sea el cambio.

2.16. Patrón de Modelo-Vista-Controlador (MVC)

Según (Harrop, 2005), "El propósito de este patrón es simplificar la implementación de aplicaciones de acuerdo a las peticiones de los usuarios y los datos a desplegar".

Para Gómez (2015), es un patrón de diseño de software en torno a la interconexión de los tres tipos de componentes principales en un lenguaje de programación como PHP, a menudo con un fuerte enfoque en

la programación orientada a objetos. Estos tres tipos de componentes son llamados: modelos, vistas y controladores. La Ilustración 6 que se presenta a continuación muestra las relaciones entre los tres componentes.

El modelo, es donde se guarda toda la lógica del negocio de una aplicación, la lógica del negocio puede ser cualquier cosa específica acerca de cómo una aplicación almacena los datos, o utiliza servicios de terceros con el fin de cumplir con sus necesidades. Si la aplicación debe acceder a la información en una base de datos, el código deberá estar guardado en el modelo.

La vista, es donde se encontrarán todos los elementos de la interfaz de usuario de una aplicación. Cualquier cosa que el usuario pueda ver, es guardada en la vista, y algunas veces lo que ve el usuario actualmente es la combinación de varias vistas en la misma petición.

El controlador, es el componente encargado de conectar el modelo con la vista. Los controladores aíslan la lógica del negocio de un modelo de los elementos de la interfaz de usuario de una vista y maneja la forma en la que la aplicación responde a la interacción del usuario en la vista. Los controladores son el primer punto de entrada en estos componentes, ya que la primera solicitud se pasa a un controlador, que luego instancia a los modelos y vistas requeridas para cumplir con una petición a la aplicación.

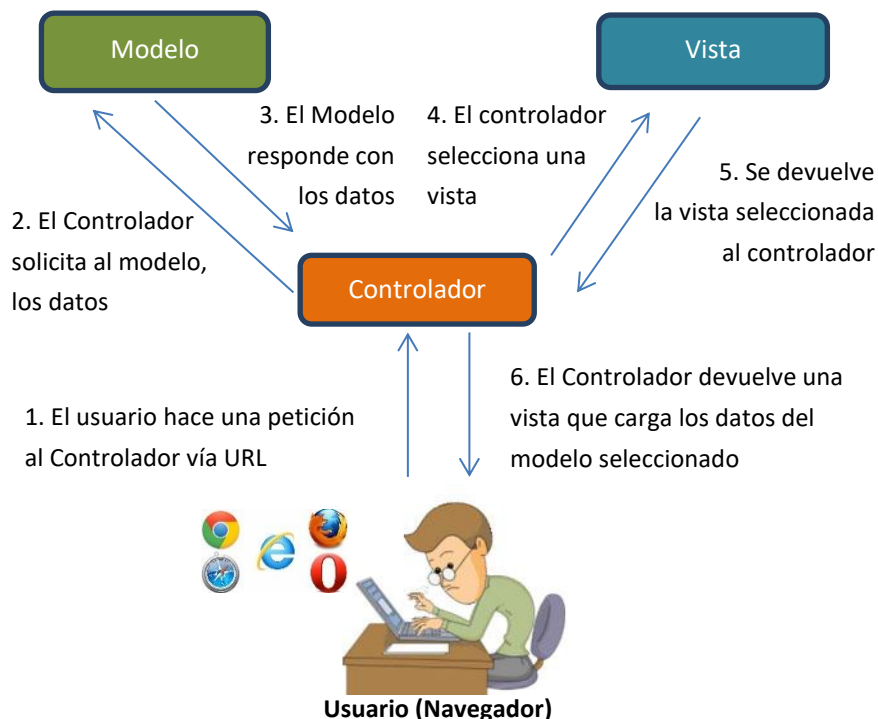


Ilustración 6. Comunicación entre componentes MVC

Fuente: Elaborada por el autor

2.17. Tecnologías para el desarrollo de la solución propuesta

Las tecnologías de desarrollo web han evolucionado mucho en la última década, existen diversidad de aplicaciones, frameworks, librerías, arquitecturas y sistemas que reciben cambios o mejoran con el tiempo. Estas tecnologías actualmente se han dividido en dos, Front End y Back End, que se explicaran de mejor manera a continuación.

2.17.1. Front End

El Front End es considerado el componente de desarrollo de software que interactúa directamente con el usuario en la capa de presentación y es el encargado de presentar los datos al usuario, así como, el diseño que ve el usuario, las imágenes, la estructura y su contenido. Este es desarrollado generalmente con los siguientes lenguajes:

2.17.1.1. HTML

Según (W3C), es el lenguaje para describir la estructura de las páginas web que provee los medios para:

- Publicar documentos en línea con encabezados, texto, tablas, listas, fotos, etc.
- Recuperar información en línea a través de enlaces de hipertexto, con el clic de un botón.
- Diseñar formularios para realizar transacciones con servicios remotos, para usar en la búsqueda de información, hacer reservaciones, ordenar productos, etc.
- Incluir hojas de cálculo, videoclips, clips de sonido y otras aplicaciones directamente en los documentos.

Con HTML, los programadores describen la estructura de las páginas utilizando el marcado. Los elementos del lenguaje etiquetan partes del contenido como párrafos, listas, tablas entre otros.

2.17.1.2. CSS

Según (W3C), es el lenguaje para describir la presentación de las páginas web, incluye los colores, el diseño y las fuentes. Permite adaptar la presentación a diferentes tipos de dispositivos, como pantallas grandes, pantallas pequeñas o impresoras. CSS es independiente de HTML y se puede utilizar con cualquier lenguaje de marcado basado en XML. La separación de HTML de CSS hace que sea más fácil mantener los sitios, compartir hojas de estilo en las páginas y adaptar las páginas a diferentes entornos.

2.17.1.3. Javascript

Según (Mozilla, 2018), es un lenguaje ligero e interpretado, orientado a objetos con funciones de primera clase, más conocido como el lenguaje de script para páginas web.

Es un lenguaje dinámico, utilizado principalmente en la Web para mejorar las páginas HTML y se encuentra comúnmente incrustado en el código HTML. No requiere ser compilado porque es un lenguaje interpretado. JavaScript hace que las páginas web sean interactivas y dinámicas lo que permite a las páginas reaccionar

a eventos, exhibir efectos especiales, aceptar texto variable, validar datos, crear cookies, detectar el navegador de un usuario.

2.17.1.4. Bootstrap

Como se pudo encontrar en (getbootstrap.com), es “un conjunto de herramientas de código abierto para desarrollar con HTML, CSS y Javascript”. También se define como una biblioteca de código abierto utilizada para diseñar las aplicaciones web que provee la facilidad de crear páginas adaptables y móviles en el ámbito web.

2.17.2. Back End

Dentro del desarrollo de software, el Back End forma parte de un subsistema que permite crear abstracción al momento de consultar y modificar datos relacionados al sistema que lo contiene. Es fundamental a la hora de desarrollar el Back End de cualquier software, conocer los diferentes tipos de lenguajes de programación que existen, para seleccionar el que mejor se adapte a los requerimientos del sistema en cuestión. A continuación, se mencionan dos de los lenguajes de programación más utilizados al momento de construir el Back End de un software:

2.17.2.1. PHP

Según (Mozilla, 2018), “es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML”. Este lenguaje de programación es ejecutado en el servidor generando HTML y enviándolo al cliente. Básicamente, tiene la habilidad de recopilar datos de formularios, generar contenidos dinámicos, o enviar y recibir cookies.

2.17.2.2. Laravel Framework

Según Quora (2016), Laravel es un framework web PHP gratuito y de código abierto, creado por Taylor Otwell en 2011 inspirándose en Ruby on Rails y Symfony, de los cuales ha adoptado sus principales ventajas, destinado al desarrollo de aplicaciones web siguiendo el modelo MVC, el cual posee una sintaxis simple, expresiva y elegante

Para (EcuRed) es un framework que permite el uso de una sintaxis elegante y expresiva para crear código de forma sencilla y permitiendo multitud de funcionalidades. Intenta aprovechar lo mejor de otros frameworks y aprovechar las características de las últimas versiones de PHP.

2.17.2.3. PostgreSQL

Según (Momjian, 2001), es un avanzado sistema de gestión de bases de datos relacionales basado en código abierto. Esto quiere decir que el código fuente del programa está disponible a cualquier persona libre de cargos directos, permitiendo a cualquiera colaborar con el desarrollo del proyecto o modificar el sistema para ajustarlo a sus necesidades.

2.17.2.4. Eloquent ORM

Como se pudo encontrar en (Laravel.com), es un ORM (mapeo de objeto relacional) que proporciona una implementación de funciones para acceso a datos de manera simple y armónica para trabajar en una base de datos. Cada tabla de base de datos tiene un “Modelo” correspondiente que se utiliza para interactuar con esa tabla.

Se utiliza para manejar de una forma fácil y sencilla los procesos correspondientes al manejo de bases de datos en el proyecto, gracias a las funciones que provee se pueden realizar consultas complejas de base de datos sin escribir una sola línea de código SQL.

CAPÍTULO 3 MARCO METODOLÓGICO

Una metodología permite definir límites, un orden y un contenido. Construir un software complejo requiere de un gran esfuerzo, tecnología, dinero y recursos. Personas que interactúan entre sí, con diferentes grados de conocimiento, con diferentes roles, con diferentes intereses. Una metodología propone un esquema de trabajo que permite entender cuál es nuestro rol dentro del proyecto, acercando una cierta sensación de control y de seguridad. Sin un proceso determinado no se sabe cómo comenzar y cuándo terminar.

El término metodología se define como el grupo de mecanismos o procedimientos racionales, empleados para el logro de un objetivo, o serie de objetivos que dirige una investigación científica. Este término se encuentra vinculado directamente con la ciencia, sin embargo, la metodología puede presentarse en otras áreas.

Existen una gran variedad de metodologías para el desarrollo de un software por lo que se debe saber seleccionar la metodología a usar, el objetivo de aplicar una metodología es mejorar la eficiencia por medio de la gestión sistemática de los procesos de negocio, los cuales se deben modelar, integrar, monitorizar y optimizar de manera continua.

Cabe destacar que una metodología puede ser aplicada no solo para una investigación científica, sino también en diversos ámbitos como una exposición doctrinal o tareas que requieran habilidades, conocimientos o cuidados específicos. (Pérez Porto & Gardey, 2008). Además de lo anterior, el marco metodológico se podría definir como la organización previa que permite una mejor planificación en tiempo, resultados y pruebas en las que se basa el proyecto.

3.1. Metodología para el desarrollo de la aplicación

En la actualidad existen diversas metodologías para el desarrollo de software, teniendo como mayor auge e índice de éxito las metodologías o marcos de trabajo ágiles. Las metodologías ágiles son una serie de técnicas para la gestión de proyectos que cumplen con los siguientes cuatro valores:

- Los individuos y su interacción, por encima de los procesos y las herramientas.
- El software que funciona, frente a la documentación exhaustiva.
- La colaboración con el cliente, por encima de la negociación contractual.
- La respuesta al cambio, por encima del seguimiento de un plan (Raya, 2016).

Según el portal web de OBS (Business School) de la Universidad de Barcelona, entre los marcos de trabajo y metodologías ágiles más utilizados en la actualidad se tienen: Extreme Programming (XP), Scrum, Kanban y Agil Inception (OBS Business School, 2018). Para este proyecto se ha decidido aplicar el marco de trabajo ágil Scrum. Scrum es un marco de trabajo para el desarrollo de software y el mantenimiento de productos complejos donde se pueden emplear varias técnicas y procesos. Este marco de trabajo es ligero,

fácil de entender y difícil de llegar a dominar. Muestra la eficacia relativa de las prácticas de gestión de producto y las prácticas de desarrollo, de modo que se pueda ir mejorando en el proceso de desarrollo. Las bases fundamentales de este marco de trabajo son: los Equipos Scrum, los roles, los eventos, los artefactos y las reglas asociadas. Cada uno de estos componentes sirve a un propósito específico y es esencial para el éxito de Scrum y para su uso (Schwaber & Sutherland, 2013).

3.2. Teoría de Scrum

Scrum se basa en la teoría de control de procesos empírica o empirismo (Manager, 2015). El empirismo asegura que el conocimiento procede de la experiencia y de tomar decisiones basándose en lo que se conoce (Significados, 2018). Scrum emplea un enfoque iterativo e incremental para optimizar la predictibilidad y el control del riesgo. A continuación, se describen los tres pilares que soportan toda la implementación del control de procesos empírico:

3.2.1. Transparencia:

Los aspectos significativos del proceso deben ser visibles para aquellos que son responsables del resultado. La transparencia requiere que dichos aspectos sean definidos por un estándar común, de tal modo que se comparta un entendimiento frecuente de lo que se está viendo. Por ejemplo, aquellos que desempeñan el trabajo y aquellos que aceptan el producto de dicho trabajo deben compartir una definición habitual de “Terminado” (Rosales & Adonis, 2016).

- Definición de “Terminado” (Definition of “Done”): todos los miembros del Equipo deben tener un entendimiento compartido de lo que significa que un elemento de la Pila de producto se describa como “Terminado”, para asegurar la transparencia. El propósito de cada Sprint es entregar incrementos de funcionalidad que se puedan poner en producción, y que se ajusten a la definición de “Terminado” actual del Equipo Scrum. Cada Incremento se integra con todos los Incrementos anteriores asegurando que todos los Incrementos funcionan en conjunto. A medida que los Equipos Scrum maduran, se espera que su Definición de “Terminado” se amplíe para incluir criterios más rigurosos que garanticen una mayor calidad.

3.2.2. Inspección

Los usuarios de Scrum deben inspeccionar frecuentemente los artefactos de Scrum y el progreso hacia un objetivo, para detectar variaciones. Su inspección no debe ser tan frecuente como para que interfiera en el trabajo. Las inspecciones son más beneficiosas cuando se realizan de forma diligente por inspectores expertos, en el mismo lugar de trabajo.

3.2.3. Adaptación

Si un inspector determina que uno o más aspectos de un proceso se desvían de límites aceptables, y que el producto resultante no será aceptable, el proceso o el material que está siendo procesado deben ser

ajustados. Dicho ajuste debe realizarse cuanto antes para minimizar desviaciones mayores. Scrum tiene cuatro eventos formales, contenidos dentro del Sprint, para la inspección y adaptación los cuales son:

- Reunión de planificación del sprint (Sprint Planning Meeting).
- Scrum diario (Daily Scrum).
- Revisión del sprint (Sprint Review).
- Retrospectiva del sprint (Sprint Retrospective). (Rosales & Adonis, 2016).

3.3. El Equipo Scrum

El Equipo Scrum está conformado por un Dueño del producto (Product Owner), el Equipo de desarrollo (Development Team) y un Facilitador (Scrum Master). Los Equipos Scrum son autoorganizados y multifuncionales (Roche & Julio, 2018). Los equipos autoorganizados eligen la mejor forma de llevar a cabo su trabajo y no son dirigidos por personas externas al equipo. Los equipos multifuncionales tienen todas las competencias necesarias para llevar a cabo el trabajo sin depender de otras personas que no son parte del equipo. El modelo de equipo en Scrum está diseñado para optimizar la flexibilidad, la creatividad y la productividad. Los Equipos Scrum entregan productos de forma iterativa e incremental, maximizando las oportunidades de obtener retroalimentación. Las entregas incrementales de producto “Terminado” aseguran que siempre estará disponible una versión potencialmente útil y funcional del producto. A continuación, se describe cada uno de los roles pertenecientes al Equipo Scrum:

3.3.1. Equipo de desarrollo (Development Team)

Está conformado por los profesionales que realizan el trabajo de entregar un Incremento de producto “Terminado”, que potencialmente se pueda poner en producción, al final de cada iteración. Solo deben haber cambio en los miembros del equipo entre las iteraciones.

3.3.2. Facilitador (Scrum Master)

Es un líder que está al servicio del Equipo Scrum, responsable de asegurar que Scrum sea entendido y adaptado, ayudando a todos en cada una de las interacciones con el fin maximizar el valor creado por el Equipo Scrum. Además, se caracteriza por las siguientes características:

- 50% desarrollador, no sólo perfil gerencial.
- Escucha el progreso y remueve los impedimentos.
- Es el moderador de los eventos.

3.3.3. Dueño del producto (Product Owner)

Es el responsable de maximizar el valor del producto y del trabajo del Equipo de Desarrollo. El cómo se lleva a cabo esto puede variar entre distintas organizaciones, Equipos Scrum e individuos. El Dueño del producto es una única persona, no un comité y es responsable de gestionar la Pila del producto (Product Backlog). La gestión de la Pila del producto incluye:

- Definir los elementos de la Pila del producto.
- Priorizar los elementos en la Pila del producto para alcanzar los objetivos y misiones de la mejor manera posible.
- Revisar junto con otros involucrados el producto al final de cada sprint.
- Asegurar que la Pila del producto es visible, transparente y clara para todos, y que muestra aquello en lo que el equipo trabajará a continuación.
- Asegurar que el Equipo de Desarrollo entiende los elementos de la Pila del producto al nivel necesario. Para que el Dueño de producto pueda hacer bien su trabajo, toda la organización debe respetar sus decisiones. Las decisiones del Dueño de producto se reflejan en el contenido y en la priorización de la Pila del producto. (Roche & Julio, 2018).

3.4. Artefactos

Los artefactos definidos por Scrum están diseñados para maximizar la transparencia de la información clave, que es necesaria para asegurar que todos tengan el mismo entendimiento del artefacto.

- Pila del producto (Product Backlog): es una lista ordenada de todo lo que podría ser necesario para construir el producto, y es la única fuente de requisitos para cualquier cambio a realizarse en el producto. Es priorizada por el Dueño del producto, puede ser re-priorizada en la planificación de cada sprint. La Pila del producto enumera todas las características, funcionalidades, requisitos, mejoras y correcciones que constituyen cambios a ser hechos sobre el producto para entregas futuras. Los elementos de la Pila del producto tienen como atributos la descripción, la ordenación, la estimación y el valor.
- Pila de pendientes del sprint (Sprint Backlog): es el conjunto de elementos de la Pila de producto seleccionados para el Sprint, más un plan para entregar el Incremento de producto y conseguir el objetivo del Sprint. La Lista de Pendientes del sprint es una predicción hecha por el Equipo de Desarrollo acerca de qué funcionalidad formará parte del próximo Incremento y del trabajo necesario para entregar esa funcionalidad en un Incremento “Terminado”.
- Incremento: El Incremento es la suma de todos los elementos de la Pila del producto completados durante un Sprint y el valor de los incrementos de todos los Sprints anteriores. Al final de un Sprint, el nuevo Incremento debe cumplir la Definición de “Terminado” del Equipo Scrum.

3.5. Eventos

Según (Schwaber & Sutherland, 2013) los eventos de la metodología Scrum son los siguientes:

- Reunión de planificación del sprint (Sprint Planning Meeting): El trabajo a realizar durante el Sprint se planifica en la Reunión de planificación del sprint. Este plan se crea mediante el trabajo colaborativo del Equipo Scrum completo y tiene un máximo de duración de ocho horas para un Sprint de un mes. El Scrum Master se asegura de que el evento se lleve a cabo y que los asistentes entiendan su propósito.

- Scrum diario (Daily Scrum): es una reunión con un bloque de tiempo de 15 minutos para que el Equipo de Desarrollo sincronice sus actividades y cree un plan para las siguientes 24 horas. Esto se lleva a cabo inspeccionando el trabajo avanzado desde el último Scrum Diario y haciendo una proyección acerca del trabajo que podría completarse antes del siguiente. El Scrum Diario se realiza a la misma hora y en el mismo lugar todos los días para reducir la complejidad. Durante la reunión, cada miembro del Equipo de Desarrollo explica: ¿Qué hice ayer?, ¿Qué voy a hacer hoy?, ¿Tengo impedimentos?
- Revisión del sprint (Sprint Review): se basa en una reunión informal, no una reunión de seguimiento, para inspeccionar el Incremento y adaptar la Pila del producto si fuese necesario. Durante la Revisión del sprint, el Equipo Scrum y los interesados colaboran acerca de lo que se hizo durante el Sprint. Basándose en esto, y en cualquier cambio a la Pila del producto durante el Sprint, los asistentes colaboran para determinar las siguientes cosas que podrían hacerse para optimizar el valor. El resultado de la Revisión del sprint es una Pila del producto revisada, que define los elementos de la Pila del producto posibles para el siguiente Sprint.
- Retrospectiva del sprint (Sprint Retrospective): es una oportunidad para el Equipo Scrum de inspeccionarse a sí mismo y crear un plan de mejoras que sean abordadas durante el siguiente Sprint. La Retrospectiva del sprint tiene lugar después de la Revisión del sprint y antes de la siguiente Reunión de planificación del sprint. El Scrum Master alienta al equipo para que mejore su proceso de desarrollo y sus prácticas para hacerlos más efectivos y amenos para el siguiente Sprint. Para el final de este evento, el Equipo Scrum debería haber identificado mejoras que implementará en el próximo Sprint.

3.6. Razones que justifican la selección del marco de trabajo Scrum

Entre las principales ventajas y beneficios que se tienen al aplicar el marco de trabajo Scrum en el desarrollo de una aplicación, se tienen los siguientes:

- Permite involucrar más al cliente con el desarrollo del producto, lo que genera mayor satisfacción.
- Permite mayor integración entre todas las personas involucradas en el proceso de desarrollo del proyecto.
- Busca la mejora continua durante todo el proceso de desarrollo en todos los niveles y aspectos del proyecto.
- Está abierto al cambio, acepta los cambios sin ningún problema buscando siempre la forma de maximizar beneficios y/o minimizar impactos negativos.
- Permite entregar un producto de valor y útil para el cliente luego de cada iteración.
- Permite tener una buena organización a nivel interno y externo.
- Permite establecer tiempos fijos de entrega, lo cual impide que se atrase cualquier entregable al cliente, siempre y cuando las condiciones para el desarrollo estén disponibles al cien por ciento.

- Se aprovecha mejor el tiempo, dado que se tiene una lista con todas las actividades pendientes por hacer durante cada sprint, y cualquier integrante del equipo puede ir tomando dichas actividades siguiendo el orden previamente establecido para las actividades.
- Permite reconocer los errores y fortalezas de los integrantes del equipo, con el fin de buscar la mejora constante a medida que avanza el proyecto (Raya, 2016).

El uso de la metodología ágil de desarrollo Scrum tomó un papel fundamental en la ejecución del proyecto al facilitar la organización de tareas, lo cual permitió ordenar las funcionalidades requeridas a través de bloques de tiempo en los cuales se evidenciaba el avance del Sprint y se destacaban las tareas que quedaban rezagadas en dicho Sprint y se agregaba al Product Backlog. Gracias a los eventos que contiene Scrum, se realizaban revisiones diarias (Daily Scrum) para pautar las actividades a realizar en ese día y cuáles fueron los avances del día anterior. Las ventajas que se evidenciaron al usar esta metodología fueron:

- Optimiza el manejo del tiempo
- Resulta más eficiente desarrollar aplicaciones al tener un seguimiento diario de las actividades realizadas.
- Es abierto a cambios ya que en ocasiones se presentaban inconvenientes con los componentes desarrollados por lo cual debían ser modificados, estos cambios se realizaron sin problema al agregarlos al Product Backlog.

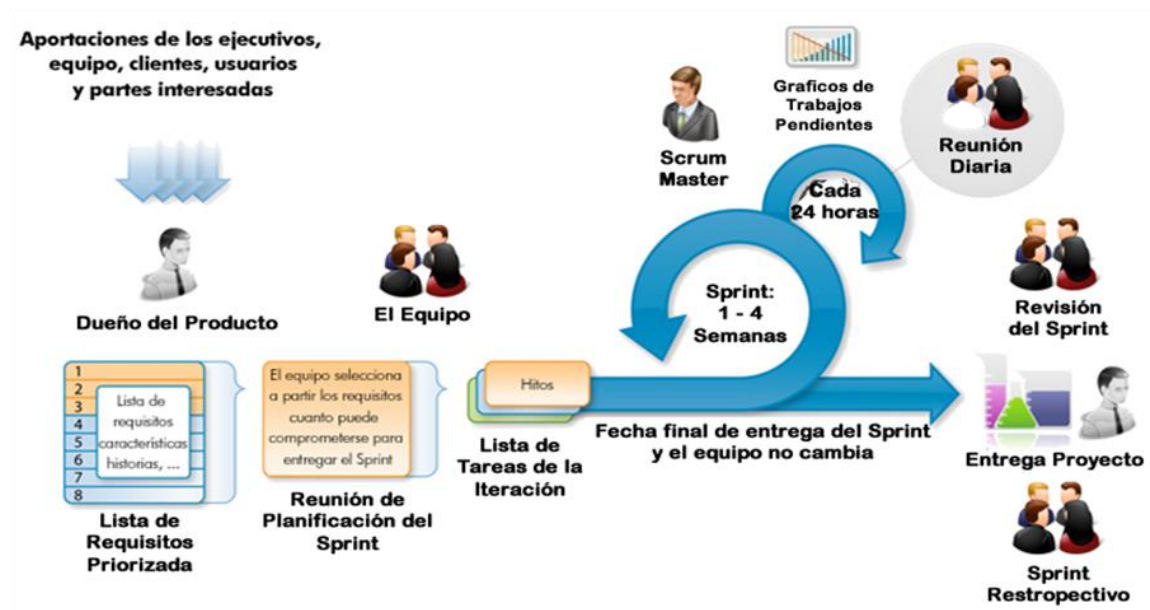


Ilustración 7. Ciclo de Vida de Scrum.

Fuente: Schwaber y Sutherland (2017).

CAPÍTULO 4 MARCO APLICATIVO

En el presente capítulo se definen las funcionalidades del sistema desarrollado para Lear Express ® (empresa de courier que trabaja en el envío de paquetes) para la gestión de los paquetes comprados en estados Unidos y enviados hasta Venezuela. Se describen las tecnologías y herramientas utilizadas, los perfiles de usuario, los Sprints realizados utilizando la metodología de desarrollo ágil SCRUM y las pruebas respectivas realizadas sobre la aplicación.

4.1. Requerimientos funcionales

El sistema presenta una serie de requerimientos funcionales que serán expuestos a continuación:

- Implementar las funcionalidades CRUD para la creación, lectura, actualización de los usuarios.
- Implementar las funcionalidades CRUD para la creación, lectura, actualización de todos los paquetes.
- Implementar la consulta mediante códigos en texto alfanuméricos.
- Implementar las funcionalidades CRUD para la creación y lectura de autorizaciones.
- Implementar jerarquías de roles que incluyan administrador y usuario.
- Permitir el acceso a la aplicación desde dispositivos móviles y de escritorio.
- Permitir la carga y descarga de guías de pago.
- Permitir respaldar los datos del sistema.
- Generar reporte de número total de envíos realizados.
- Enviar notificaciones vía correo electrónico a los usuarios.

4.2. Requerimientos no funcionales

También, se presenta una serie de requerimientos no funcionales que estarán reflejados en el sistema:

- Usabilidad: permitir a los usuarios el fácil acceso y manejo de la aplicación.
- Rendimiento: permitir un tiempo de respuesta bajo y aceptable para las peticiones.
- Disponibilidad: permitir que se tenga acceso a la aplicación en cualquier momento.
- Mantenibilidad: apoyar la facilidad de mantenimiento y cambios en la aplicación.
- Eficiencia: uso racional de los recursos para lograr tiempos de respuesta óptimos.
- Confiabilidad: ampliar la capacidad de la aplicación para cumplir con las funcionalidades anteriormente expuestas.
- Portabilidad: permitir la ejecución de la aplicación en diferentes arquitecturas.

4.3. Usuarios potenciales

La aplicación va dirigida principalmente a usuarios clasificados en dos categorías, en primer lugar, el administrador del sistema y, en segundo lugar, los usuarios externos como usuarios comunes.

4.4. Perfiles de usuario

El sistema de Lear Express posee un esquema de usuarios que facilita el control de las acciones que pueden o no ser realizadas dependiendo de su tipo. A continuación, se evidencian los perfiles que están presentes en el desarrollo.

4.4.1. Administrador

Este tipo usuario tiene la libertad de crear usuarios, instituciones, especímenes, objetos multimedia y también tiene la facultad de editarlos y eliminarlos a voluntad. Cabe destacar que, el administrador puede crear usuarios de cualquier perfil.

4.4.2. Usuario

El usuario Visitante puede consultar toda la información de los especímenes, instituciones y objetos multimedia a través de la aplicación móvil.

Tabla 1. Perfiles de Usuario

Fuente: Creada por el autor, 2019

Funcionalidad	Administrador	Usuario Registrado
Registrarse como usuario	Si	Si
Registrar Prealerta	No	Si
Registrar paquetes	Si	No
Registrar Valijas (envíos)	Si	No
Consultar Envíos	Si	No
Actualizar estatus de paquete	Si	No
Registrar autorización	No	Si

4.5. Aplicación de la metodología SCRUM

El desarrollo del sistema para Lear Express fue basado en la metodología ágil Scrum, apoyándose de sus artefactos como lo son el Product Backlog y Sprint Backlog como también en los eventos asociados Daily Scrum y Sprint Planning Meeting. Esta aplicación de la metodología se verá reflejada a continuación.

4.5.1. Lista de objetivos (Product Backlog)

A continuación, se mostrará una lista de los Sprint de la metodología seleccionada que contribuyeron a el desarrollo del sistema:

Tabla 2. Lista de Sprint realizados en el desarrollo del sistema

Fuente: Creada por el autor, 2019

Sprint	Actividad	Fecha
1	1.1. Creación de proyecto Laravel.	13/05/2019
	1.2. Inicio de repositorios en Git.	24/05/2019
	1.3. Elaboración de la base de datos PostgreSQL.	
	1.4. Creación de Vistas	
2	2.1. Desarrollo de operaciones CRUD en Laravel.	27/05/2019
	2.2. Implementación de los usuarios.	07/06/2019
	2.3. Desarrollo del motor de búsqueda de los paquetes.	
	2.4. Complementar vistas y diseños.	
	2.5. Creación de documentación básica del servidor.	
3	3.1. Ejecución de pruebas unitarias para los servicios de Laravel.	10/06/2019
	3.2. Definición de la estructura de las guías.	21/06/2019
	3.3. Creación de vistas de búsqueda y aplicación de filtros.	
4	4.1. Definición de puntos de interés de usuarios y guías.	25/06/2019
	4.2. Creación del proyecto para el sistema administrador.	04/07/2019
5	5.1. Implementación de operaciones CRUD para crear, modificar y eliminar prealertas, guías, paquetes y valijas.	08/07/2019
		19/07/2019
6	6.1. Culminación del diseño de las vistas en el sistema administrativo.	22/07/2019
		02/08/2019
7	7.1. Ejecución de pruebas unitarias y funcionales para los servicios del sistema administrativo.	05/08/2019
	7.2. Ejecución de pruebas funcionales para los servicios.	16/08/2019
	7.3. Ajustes finales de desarrollo.	

4.5.2. Lista de tareas de la iteración (Sprint Backlog)

Sprint 1: Inicio del proyecto, creación de base de datos, creación de vistas y diseño simples.

En este primer Sprint de la metodología Scrum se creó el proyecto de desarrollo en PHP, además, se creó el repositorio en git del proyecto. Por otro lado, se crearon migraciones para el almacén de datos, iniciándose con la estructura establecida anteriormente y que funcionaría con PostgreSQL ayudado por Laravel. Adicionalmente, se tomaron en cuenta las vistas iniciales en bosquejos simples que en principio no tendrían funcionalidad, estas vistas fueron sugeridas por la empresa.

Sprint 2: Desarrollo de operaciones CRUD, motor de búsqueda de los especímenes, complementar vistas, documentación del API (Application Programming Interface) de Laravel.

Se desarrollaron las operaciones CRUD para todos los registros en el almacén de datos, estas incluyen las funcionalidades de editar, crear y eliminar usuarios, prealertas, paquetes, valijas y autorizados. También, se desarrolló el motor de búsqueda avanzado el cual permitiría buscar los paquetes prealertados.

Se debe definir los tipos de peticiones existentes en la documentación:

- GET: El método GET solicita una representación de un recurso específico. Las peticiones que usan el método GET sólo deben recuperar datos.
- PUT: El modo PUT reemplaza todas las representaciones actuales del recurso de destino con la carga útil de la petición.
- DELETE: El método DELETE borra un recurso en específico.
- POST: El método POST se utiliza para enviar una entidad a un recurso en específico, causando a menudo un cambio en el estado o efectos secundarios en el servidor.

(Puentes, 2018)

Luego se complementaron las vistas con los títulos, campos de texto, sección de usuarios. Para lograr un avance significativo en la aplicación y posteriormente en el sistema administrativo, se realizó una documentación completa de los servicios, contemplando paquetes, autorizados, ejemplo de entrada y ejemplo de salida. Esta documentación también define tipo de datos y formatos de respuesta. Lamentablemente por petición de la empresa, estos no se pueden mostrar.

Cabe destacar que, todos los servicios devuelven una respuesta genérica de la siguiente forma: '{ "status": STRING, "result": STRING o ARRAY }', el campo status contendrá "failed" si ocurrió algún tipo de error y por consiguiente se agregará un nuevo campo a la respuesta de nombre "message" en el que indicará el problema presentado, por el otro lado, si la petición está correcta se enviará en el campo status "success" seguido del campo "result" que depende de la petición se presentará en forma de cadena de caracteres o de un arreglo de datos.

Sprint 3: Ejecución de las pruebas unitarias para los servicios de Laravel Definición de la estructura de las guías (paquetes), creación de vistas de búsqueda y aplicación de filtros.

Al iniciar el Sprint 3 y seguido de la documentación se realizaron pruebas unitarias a cada servicio siguiendo los parámetros establecidos en la misma. En estas pruebas se enviaron datos de prueba para contemplar todos los casos posibles de fallo y éxito. Gracias a estas pruebas de mínima intensidad se lograron captar errores de validación, fallas en la estructura de la base de datos y problemas con las peticiones, que fueron atendidas de inmediato para que la aplicación móvil y el sistema administrativo no tuviera inconvenientes al integrarlas. En el Sprint 7 se realizaron pruebas funcionales con más detenimiento que se verán reflejadas en la sección de Pruebas realizadas y resultados.

Se definió la estructura de las guías, en principio, se estableció utilizar los recursos del servidor para crear los paquetes. Posterior a esto, se creó una funcionalidad en el servidor que permite generar y obtenerlos.

Se crearon las vistas de búsqueda simple de espécimen y de institución, además de la búsqueda avanzada de especímenes por filtros. Además, se realizaron ajustes a las vistas ya creadas anteriormente y el flujo básico de la aplicación.

Sprint 4: Definición de usuarios y guías, creación del proyecto para el sistema administrador.

Se definió la estructura de los puntos de usuarios y guías, que tendrán parámetros como el precio del paquete y la descripción que podrá ser configurado desde el administrador por si se diera el caso de que el usuario registrado no realizará dicha prealerta.

Por último, se definieron los flujos para el sistema administrador usando Laravel, además del uso de las funcionalidades y los permisos de cada usuario.

Sprint 5: Implementación de operaciones CRUD para crear, modificar y eliminar prealertas, guías, paquetes y valijas.

Se definieron las operaciones CRUD en el sistema administrativo como lo son crear, modificar y eliminar prealertas, guías, paquetes y valijas dependiendo de los permisos que tenga dicho usuario autenticado.

Sprint 6: Culminación del diseño de las vistas en el sistema administrativo y móvil.

Se finalizó el diseño de las vistas en el sistema administrativo, se ajustaron las dimensiones para dispositivos móviles y de escritorio, se realizaron ajustes en validación de campos y se agregaron funcionalidades a las vistas para que fuese más intuitivo realizar la catalogación de especímenes.

Sprint 7: Ejecución de pruebas unitarias y funcionales para los servicios del sistema administrativo, ejecución de pruebas funcionales para los servicios, ajustes finales de desarrollo.

Se realizaron las pruebas unitarias del sistema administrativo y de la aplicación móvil, con cada funcionalidad disponible. En el caso del sistema administrativo, se detectaron fallas con respecto a los permisos, que fueron solventados al momento, también, al momento de cargar objetos multimedia ocurría una inconsistencia de datos en el servidor. Además, se realizaron las respectivas pruebas funcionales para cada sección establecida en el sistema administrativo y en la aplicación que se verán reflejadas y explicadas con detenimiento en la sección de Pruebas realizadas y resultados.

4.6. Tecnologías y herramientas utilizadas para el desarrollo

El desarrollo del sistema fue apoyado por las tecnologías: Laravel (PHP) para el servidor. En cuanto a las herramientas utilizadas se tienen en cuenta: HTML para la estructura, CSS junto a Bootstrap para el diseño, JavaScript para la interacción con el cliente, PostgreSQL para el sistema manejador de base de datos y finalmente para el manejador de versiones se utilizó Bitbucket.

Todas las tecnologías mencionadas fueron utilizadas para el desarrollo de la aplicación, el sistema administrativo y los servicios necesarios con el fin de facilitar todo el proceso de codificación y diseño del software.

4.6.1. Análisis del modelo de datos

Con la necesidad de involucrar un sistema para los usuarios al momento de registrar una nueva prealerta se diseñó un modelo de base de datos “híbrido” el cual permite crear, modificar y eliminar la data tanto de forma dinámica como de forma estática. También, es necesario que el sistema manejador de base de datos no sea influyente en este enfoque, la solución fue determinar los datos que serían obligatorios al momento de crear una nueva guía y cuáles no, y así, con el concepto de migraciones que brinda el Laravel, formar la estructura que llevaría la base de datos sin involucrar un sistema manejador específico.

Inicialmente, es necesario tener un manejo de usuarios y permisos para el sistema administrativo, luego, que estos usuarios estuvieran relacionados con los especímenes creados y modificados y estos a su vez conectados con las instituciones establecidas. Además, permitir que agregar nuevos campos fuese intuitivo y opcional sin perder el enfoque híbrido. Por tanto, se presenta un esquema que describe con brevedad las tablas involucradas en la estructura anteriormente mencionada en la estructura de la base de datos) y luego de ello, en la ilustración 8 e ilustración 9 se presenta el modelo de base de datos planteado por el autor, vale acotar que la tabla “*usuarios_casillero*” se difumino los atributos por petición de Lear Express ®.

Tabla 3. Tablas involucradas en la estructura de la base de datos

Fuente: Creada por el autor, 2019

Nombre de la tabla	Descripción
usuarios	Permite almacenar la estructura de autenticación y relacionada directamente con la tabla usuarios_casillero.
usuarios_casillero	Almacena la información básica de los usuarios.
guias	Contiene toda la información del paquete, así como también su estatus.
proforma	Es la que almacena el costo del envío, se trabaja como una factura.
agencias	Tiene la información de las agencias, nombre y tipo de agencia entre algunos atributos
autorizaciones	Almacena la información de las autorizaciones creadas por los usuarios.
permissions	Tiene la información de los permisos de cada usuario.
roles	Contiene los roles que se representa en cada usuario, siendo estos administrativo o usuario registrado.

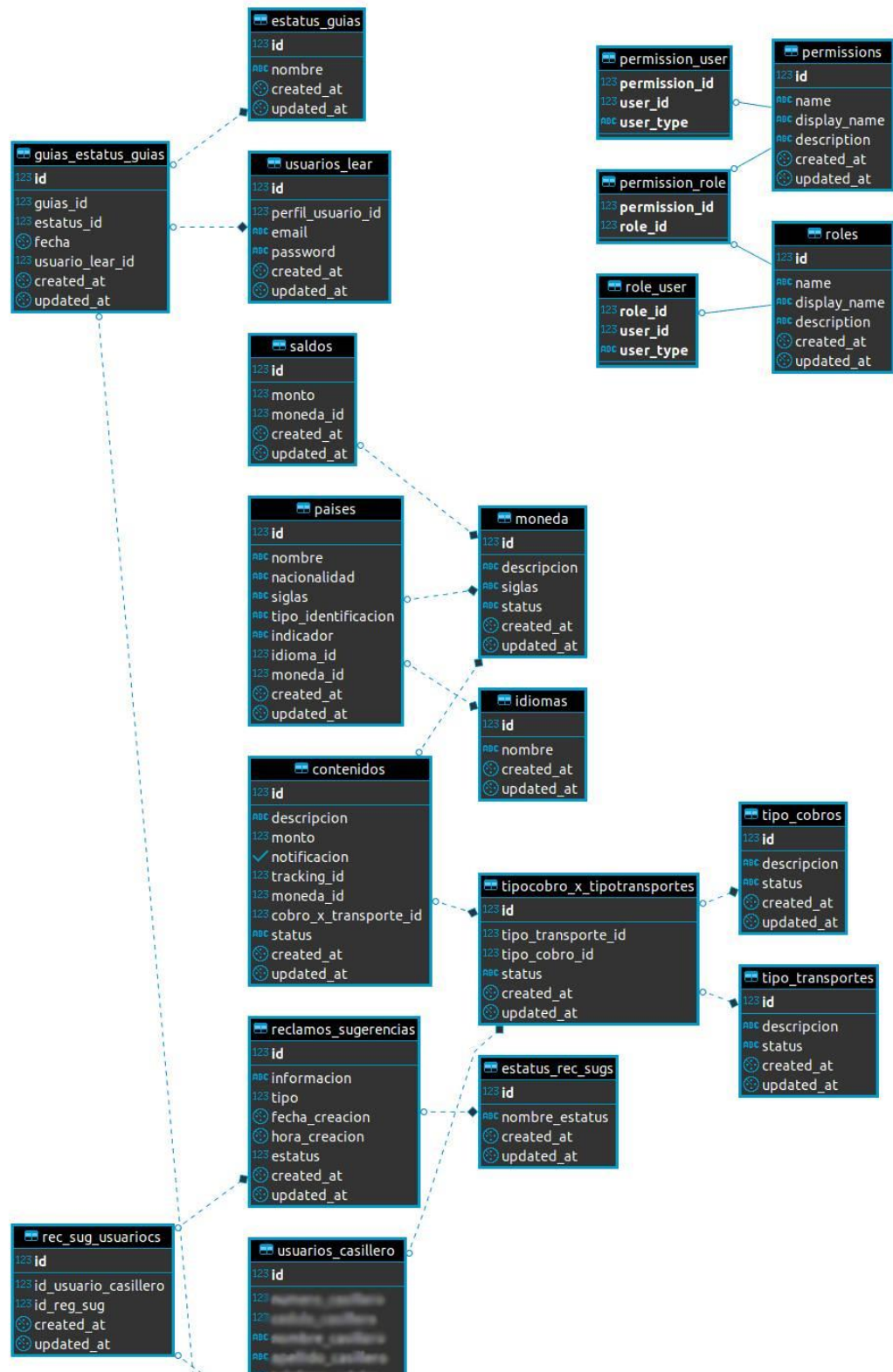


Ilustración 8. Modelo de la base de datos

Fuente: Creada por el autor, 2019



Ilustración 9. Modelo de la base de datos (Parte 2)

Fuente: Creada por el autor, 2019

4.7. Principales interfaces diseñadas

- Registro en el sistema

En el registro del sistema (Ilustración 10) se encuentran una serie de campos orientados a la obtención del correo y contraseña, para de esta forma obtener el usuario, en la que se muestran las características que debe presentar dicha contraseña.

Ilustración 10. Registro en el sistema.

Fuente: Obtenida de la aplicación, 2019

- Registro de casillero

En el registro de casillero (Ilustración 11) se encuentran una serie de campos necesarios para el registro de la persona junto a su casillero virtual, como el nombre, apellido, estado, agencias y forma de pago.

Registro del casillero




Foto de cédula

Browse... No file selected.

Datos del Usuario

Nombre

El nombre es un dato requerido

Apellido

Documento de Identidad

Nacionalidad

Telefono Oficina

Telefono Celular

Telefono Habitacion

Tipo de Contacto

Ciudad

Codigo Postal

Direccion (150 caracteres restantes)

Estado de la Agencia

Agencia

Forma de pago predeterminada

Pago en origen

Aereo

Pago en destino

Marítimo

Ilustración 11. Inicio del sistema administrativo.

Fuente: Obtenida de la aplicación, 2019

- Inicio de sistema

Para iniciar sistema (Ilustración 12), es necesario el usuario y la contraseña, junto al link para pedir cambiar la contraseña (“olvide contraseña”) y la casilla de “Recordarme”.

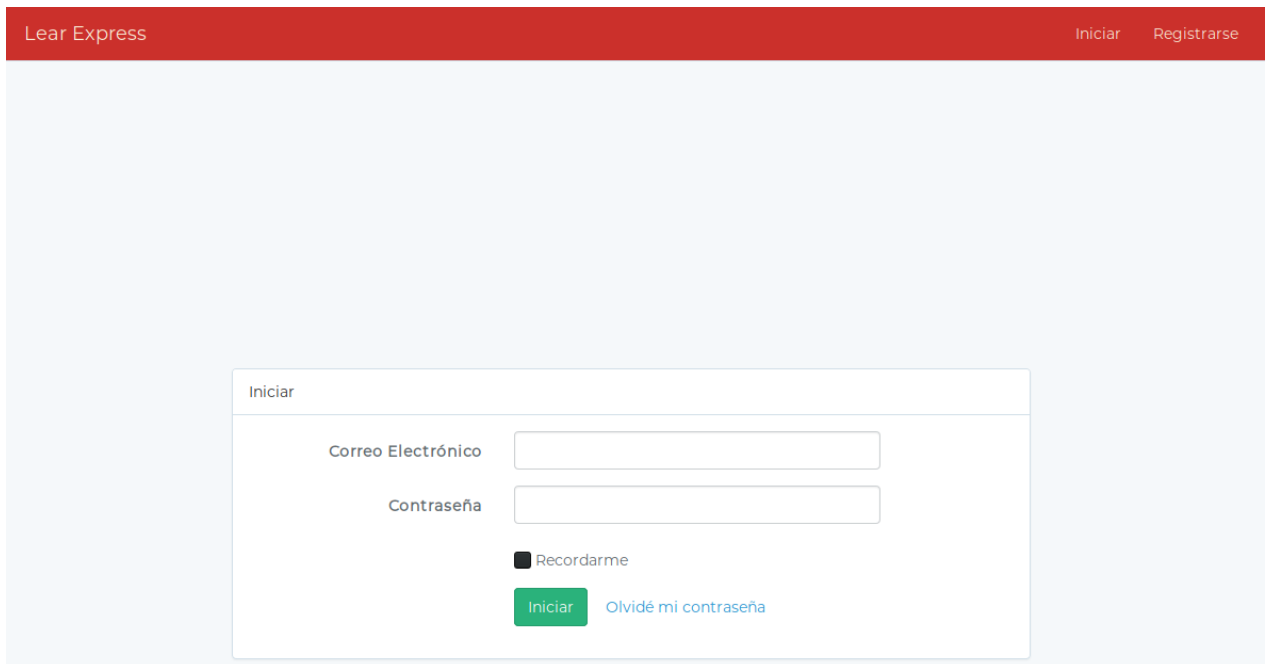


Ilustración 12. Inicio del sistema.

Fuente: Obtenida de la aplicación, 2019

- Inicio de sistema (usuario registrado)

En el inicio del sistema para el usuario registrado (Ilustración 13) se encuentran una serie de botones que reflejan las opciones que el usuario puede realizar, entre ellas esta, prealerta, que es la que se presenta al inicio, el tracking de los paquetes que se realizó la prealerta, la consulta de los envíos, las autorizaciones y los reclamos o sugerencias. Además, se encuentra con un menú lateral en el que los mismos botones, además de la actualización del perfil de usuario y el cambio de contraseña.

En esta página se piden los datos para el registro de los paquetes (la prealerta de los paquetes), el número de Tracking (número que dan las páginas de compra de productos), descripción de productos, el precio, la casilla para mantenerse informado de los estatus de los paquetes y la forma de pago que se registrara dicho paquete.

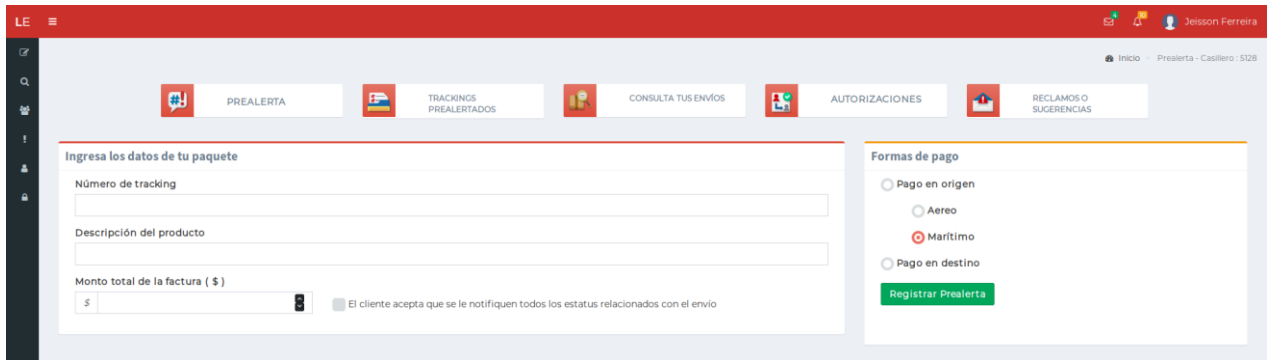


Ilustración 13. Inicio del sistema (usuario registrado).

Fuente: Obtenida de la aplicación, 2019

- Lista de Trackings prealertados

En la lista de trackings prealertados (Ilustración 14) el usuario puede revisar sus trackings, además de tener la oportunidad de agregar nuevos, editar los registrados o eliminarlos, esto solo puede hacerse si aún no se ha registrado un paquete con las características de la prealerta.



Ilustración 14. Lista de Trackings prealertados

Fuente: Obtenida de la aplicación, 2019

- Consulta de envíos

En la consulta de envíos (Ilustración 15) el usuario puede revisar el estatus de los paquetes (envíos).

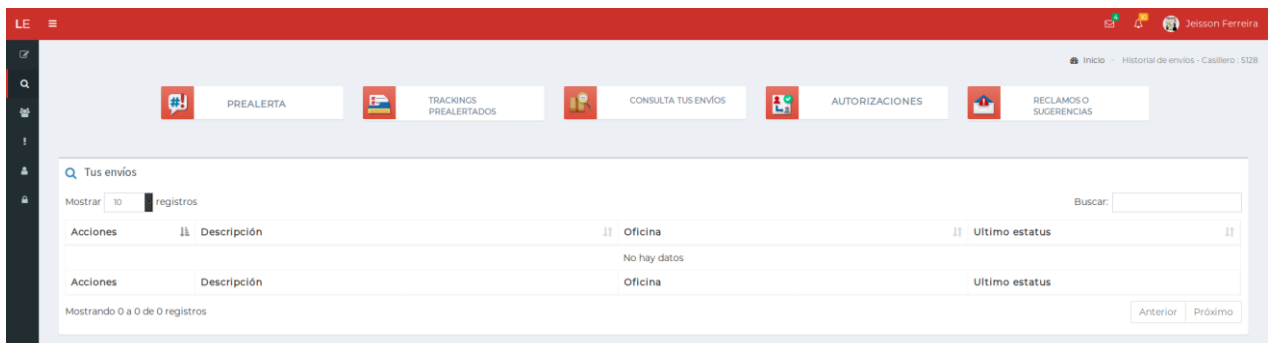


Ilustración 15. Consulta de envíos

Fuente: Obtenida de la aplicación, 2019

- Lista de autorizados

Aquí se presenta la lista de los autorizados (Ilustración 16) pudiendo agregar nuevos, esta lista tiene varias pestañas, siendo la primera llamada autorizados, en donde aparecen todos y las otras dos, pendientes y entregadas, un filtro de la misma, además pudiendo registrar autorizados.

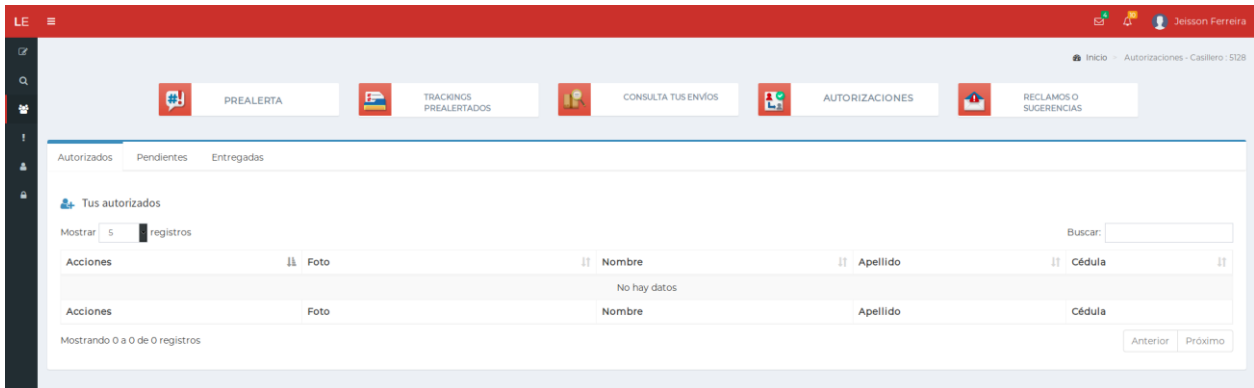


Ilustración 16. Lista de autorizados

Fuente: Obtenida de la aplicación, 2019

- Registro de Autorizados

Al registrar un autorizado (Ilustración 17) se incluirá en el sistema la información de este como el nombre y cedula, agregando también la foto de la cedula, siendo este archivo nombrado de la misma forma que el número de cedula del autorizado, quedando una este autorizado en la lista de como se muestra en la Ilustración 18.

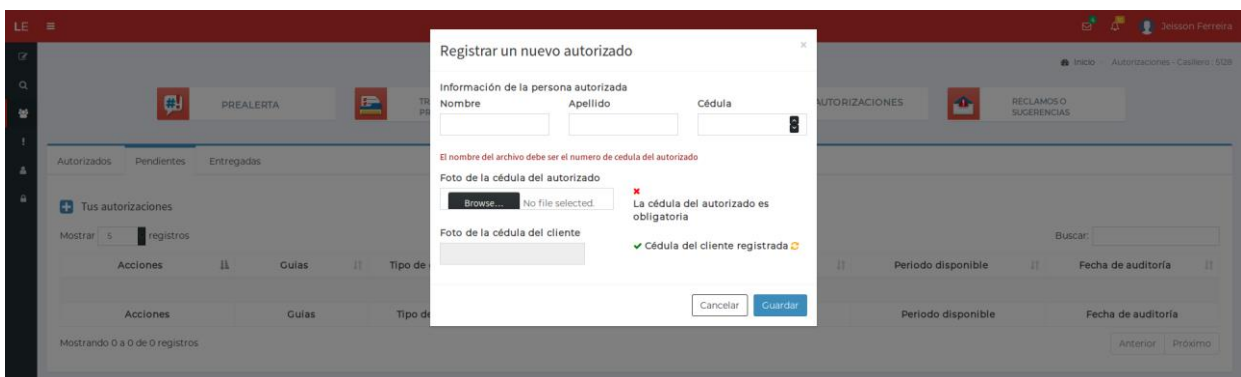


Ilustración 17. Registro de Autorizados

Fuente: Obtenida de la aplicación, 2019

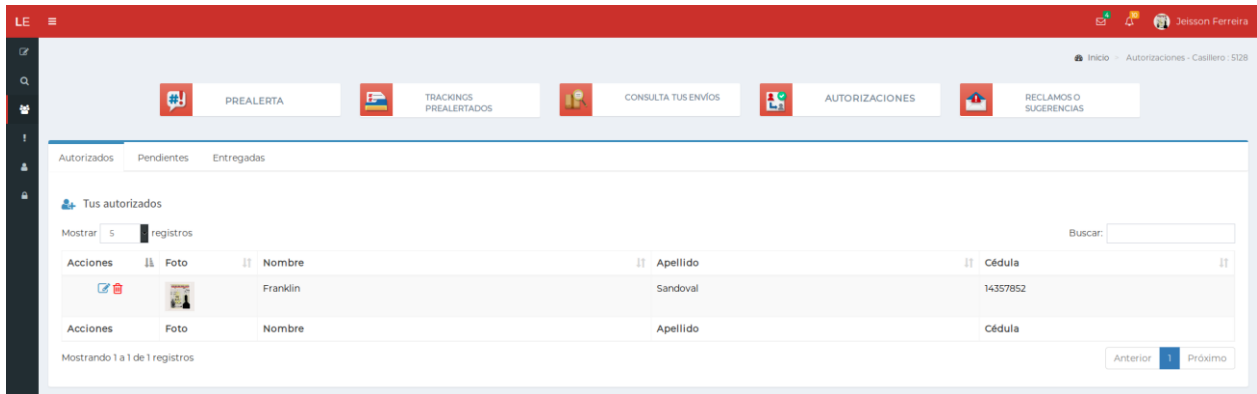


Ilustración 18. Lista de autorizados

Fuente: Obtenida de la aplicación, 2019

- Inicio de sistema (usuario administrador)

En el inicio del sistema para el usuario administrador (Ilustración 19) se encuentran una serie de contadores que reflejan la cantidad de guías facturadas, la cantidad de notas de crédito, pagos por tarjeta y por transferencia. Además, se muestra un resumen del día en el que aparece subtotal y total del mismo.

En esta página también se presenta un menú lateral en que aparece las valijas, los paquetes, un gráfico de las guías, los permisos y roles para los usuarios.

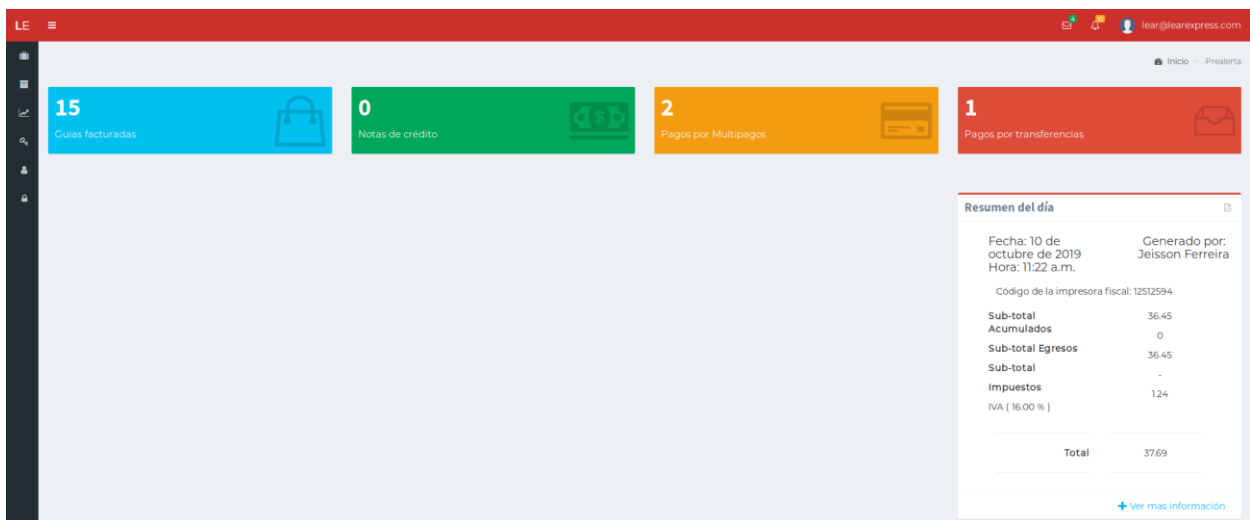


Ilustración 19. Inicio del sistema (usuario administrador).

Fuente: Obtenida de la aplicación, 2019

- Valijas

Aquí se observan las valijas (el conglomerado de paquetes), en el que se puede registrar nuevas valijas.

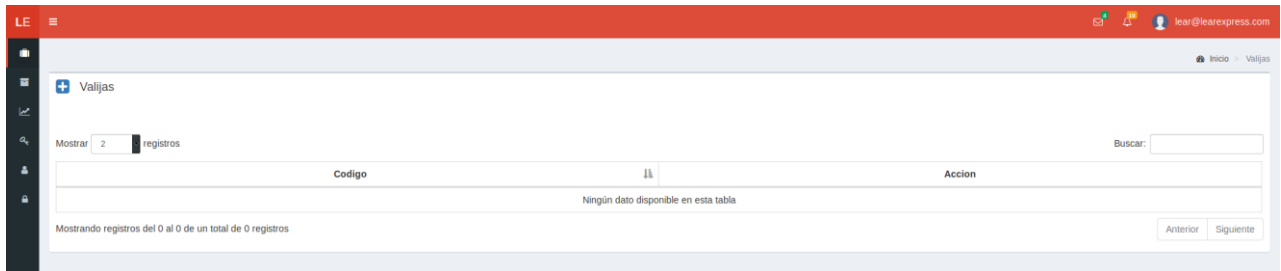


Ilustración 20. Valijas.

Fuente: Obtenida de la aplicación, 2019

- Paquetes

Aquí se observa la lista de los paquetes, en donde además se puede registrar nuevos paquetes.

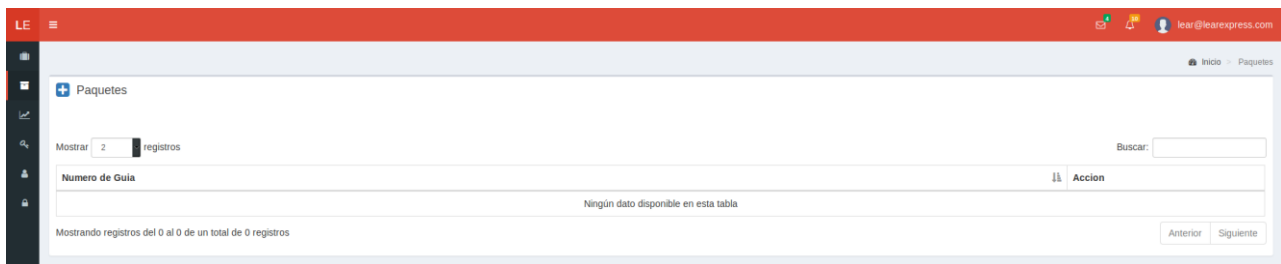


Ilustración 21. Paquetes.

Fuente: Obtenida de la aplicación, 2019

- Gráfico de guías

Aquí se observa el grafico de las guías en la que se observa las guías del mes en curso, además del número total.

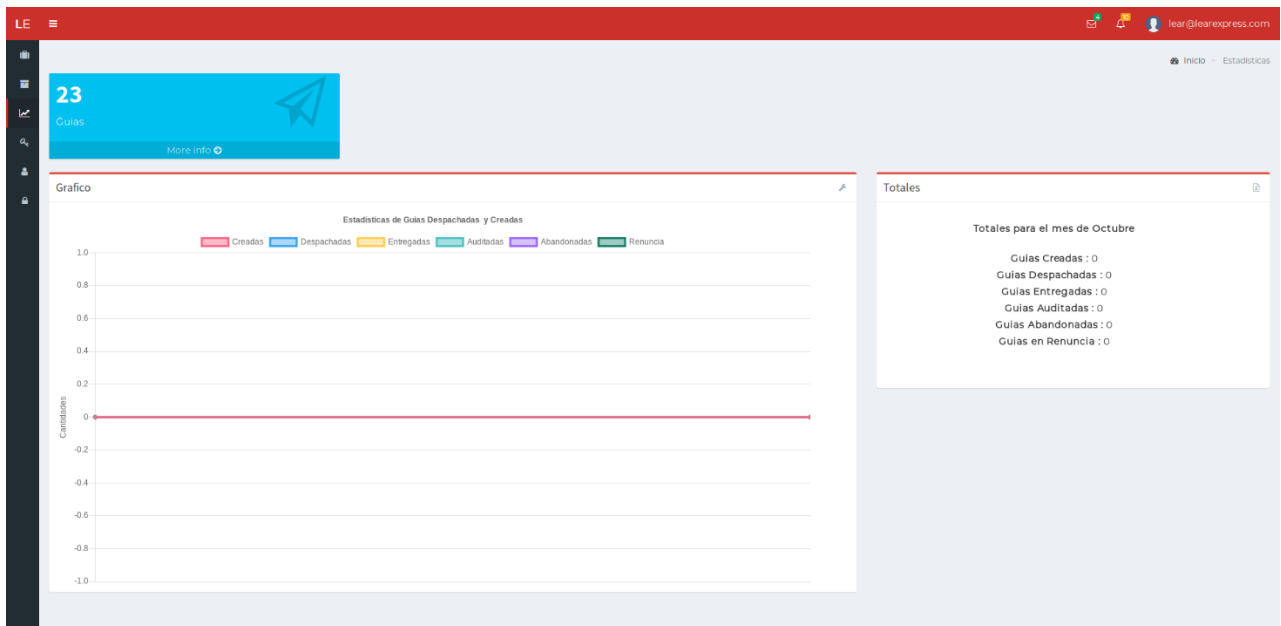


Ilustración 22. Gráfico de guías.

Fuente: Obtenida de la aplicación, 2019

- Permisos

Aquí se observan los permisos que se podrán otorgar a cada usuario.

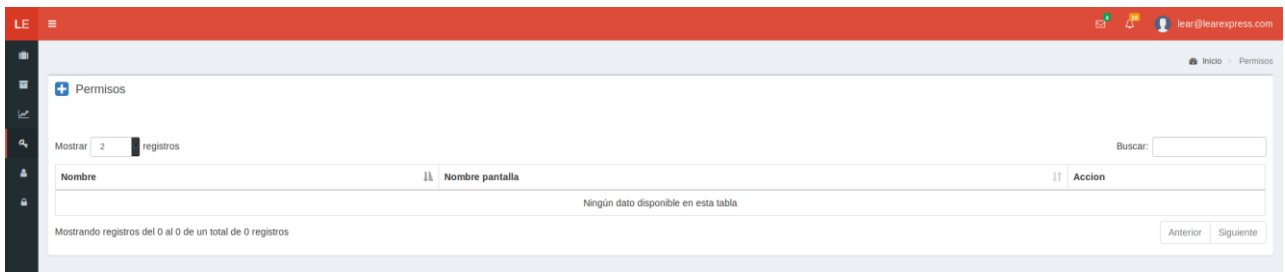


Ilustración 23. Permisos.

Fuente: Obtenida de la aplicación, 2019

- Roles

Aquí se observan los roles que se podrán otorgar a cada usuario.

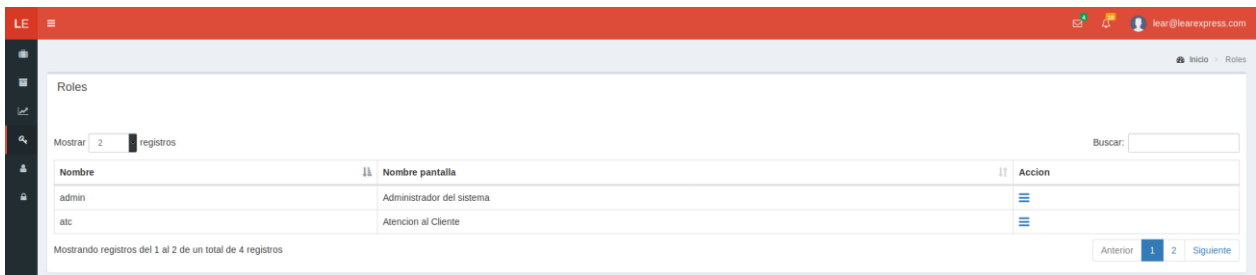


Ilustración 24. Roles.

Fuente: Obtenida de la aplicación, 2019

- Usuarios

Aquí se observa la lista de usuarios, en donde también se podrá editar los mismos.

Nombre	Nombre pantalla	Acción
admin	Administrador del sistema	[Icono de menú]
atc	Atención al Cliente	[Icono de menú]

Ilustración 25. Usuarios.

Fuente: Obtenida de la aplicación, 2019

4.8. Pruebas realizadas y Resultados

Durante el desarrollo del Trabajo Especial de Grado, se llevaron a cabo pruebas unitarias y funcionales tanto del servidor y el sistema administrativo. En el caso de las pruebas unitarias, fueron detalladas en la sección de lista de tareas de la iteración: Sprint Backlog y a continuación se presentan las pruebas funcionales de cada módulo.

4.8.1. Pruebas realizadas en el servidor

Las pruebas funcionales que se realizaron en el servidor estuvieron en función de casos de prueba estructurados de la siguiente manera:

Para las pruebas del lado del servidor, se utilizó la herramienta Postman de Google la cual permite realizar peticiones de cualquier tipo. Está compuesto por diferentes herramientas y utilidades gratuitas que permiten realizar tareas diferentes dentro de los ambientes de servidores.

Para cada funcionalidad que provee el servidor siguiendo el esquema de la documentación previa se crearon peticiones con datos de prueba que buscaban forzar errores en el sistema. Gracias a ello, se hallaron errores de sintaxis de las rutas de algunas funcionalidades, se detectó también una serie de eventos inesperados al momento de obtener los datos de los usuarios, como lo fueron: el último usuario registrado no se reflejaba en el sistema y no se reflejaba la relación entre usuario-institución. En la ilustración 22 se puede observar la distribución de las pruebas.

Servicios

50 respuestas

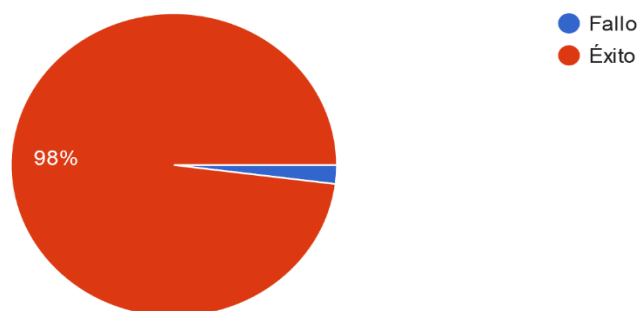


Ilustración 26. Resultado de prueba de servidor. Total 50 servicios, 49 éxito y 1 fallos.

Fuente: Documentos de Google, visualización de respuestas del formulario, 2019

No obstante, este error fue resuelto a medida que se realizaban las pruebas funcionales para cada servicio. También, se diseñaron casos para probar el correcto funcionamiento de las peticiones, esperando obtener una respuesta exitosa del lado del sistema, estas fueron culminadas de manera satisfactoria.

4.8.2. Pruebas realizadas en el sistema administrativo

Las pruebas unitarias y funcionales que se realizaron en el sistema administrativo estuvieron basadas en cada servicio que provee, detallados a continuación:

Las pruebas asociadas al registro de guías en el sistema se basaron en primer lugar en ingresar datos, lo que posteriormente permitió hacer validaciones en el servidor y en el sistema administrativo para evitar esta inconsistencia de datos.

Además, al utilizar la funcionalidad de generar guías el inconveniente de que no eran visualizados correctamente en la pantalla, que posteriormente, se modificaron las dimensiones para solventar esa incidencia. En la ilustración 27 se muestra gráficamente la distribución de las pruebas.

Funcionalidad

20 respuestas

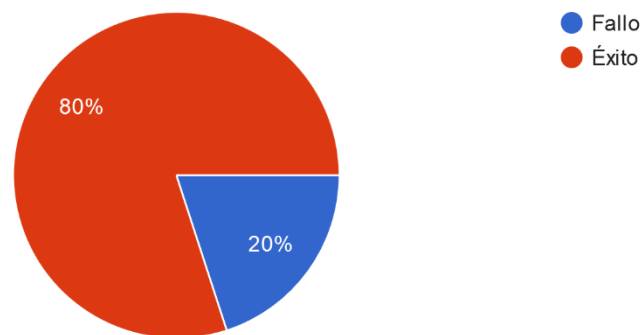


Ilustración 27. Resultado de prueba de funcionalidad.

Total 20 servicios, 16 éxito y 4 fallos.

Fuente: Documentos de Google, visualización de respuestas del formulario, 2019

4.8.3. Pruebas de aceptación

Se realizaron pruebas de aceptación para determinar si la aplicación y el sistema cumplían con las necesidades y los requerimientos del usuario. Estas pruebas fueron basadas en una lista de preguntas e instrucciones que permitían explorar todas las funcionalidades de los productos de software ya mencionados. Cabe destacar que, las pruebas fueron realizadas por el desarrollador, por el tutor del TEG y las personas que usaran dicho sistema (Lear Express ®), siendo las pruebas realizadas a seis personas.

Las pruebas estaban divididas en dos:

- Pruebas de aceptación
- Cuestionario de aceptación
- **Pruebas de aceptación** – Inicio de sesión, en esta sección se les dio a los usuarios credenciales, un ejemplo de estas instrucciones se encuentra en la ilustración 28, para validar el comportamiento permitiéndoles explorar todo su contenido.

¿Se sintió cómodo con el inicio de sesión?

6 respuestas

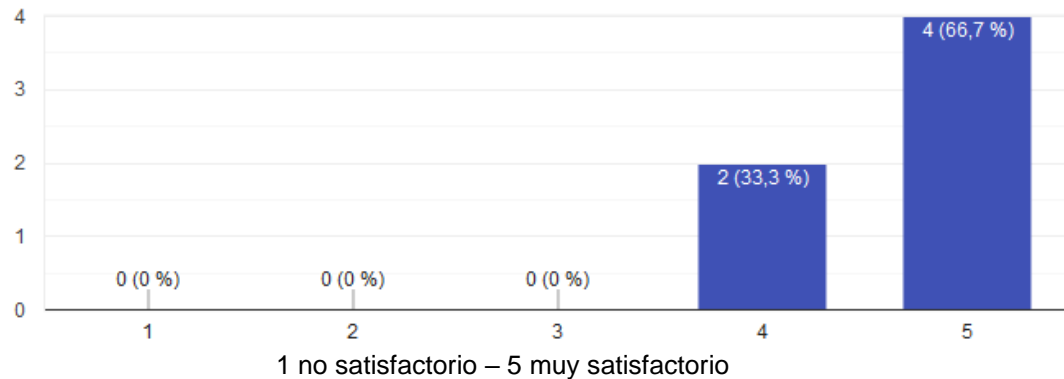


Ilustración 28. Instrucciones suministradas para las pruebas de aceptación.

Fuente: Documentos de Google, visualización de respuestas del formulario, 2019

¿Se sintió cómodo con el diseño de la aplicación?

- **Cuestionario de aceptación**, finalmente en esta sección se presentó una serie de preguntas acerca de la experiencia obtenida con respecto a diseño, usabilidad, satisfacción y detalles adicionales que quisiera agregar el usuario.

En la ilustración 29 se muestran dos gráficos en base a las pruebas de aceptación, obteniendo un 50% en la respuesta muy satisfactorio y un 50% adicional en el valor 4 de aceptación.

¿Se sintió cómodo con el diseño de la aplicación?

6 respuestas

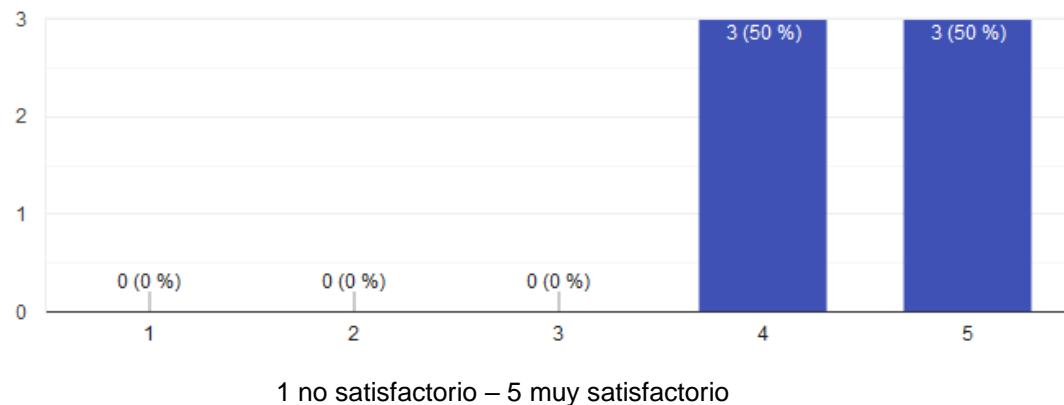


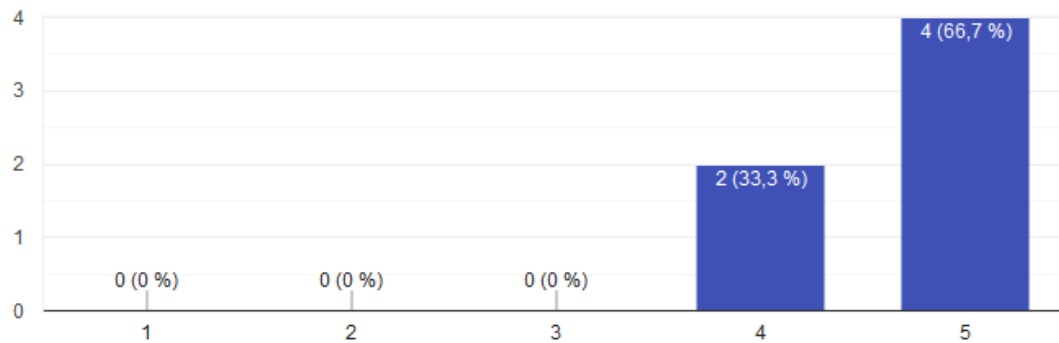
Ilustración 29. Pruebas de aceptación (1)

Fuente: Documentos de Google, visualización de respuestas del formulario, 2019

En la ilustración 30 y 31, se muestran dos gráficos, para el primero se cuestiona sobre las características de la comodidad de la aplicación, obteniendo un 66.6% de muy satisfactorio y un 33.3% de un valor 4 de satisfacción. Para el segundo, se busca conocer si hubo alguna dificultad para crear usuarios, con un 100% de respuesta negativa, por lo que no hubo ningún inconveniente al crear usuarios.

¿Se sintió cómodo con el manejo de la aplicación?

6 respuestas



1 no satisfactorio – 5 muy satisfactorio

Ilustración 30. Pruebas de aceptación (2)

Fuente: Documentos de Google, visualización de respuestas del formulario, 2019

¿Presentó dificultad para crear usuarios?

6 respuestas

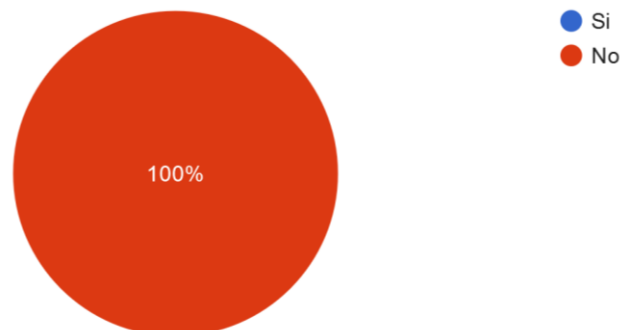


Ilustración 31. Pruebas de aceptación (3)

Fuente: Documentos de Google, visualización de respuestas del formulario, 2019

En la ilustración 32, se presenta la siguiente pregunta de las pruebas de aceptación, en la que se evidencia que los usuarios no se encontraron con problemas al crear una prealerta.

¿Presentó dificultad para crear prealertas?

6 respuestas

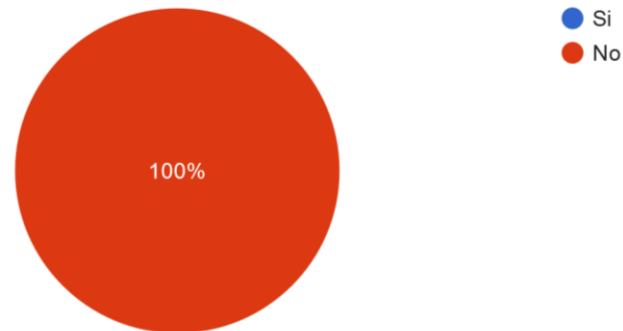


Ilustración 32. Pruebas de aceptación (4)

Fuente: Documentos de Google, visualización de respuestas del formulario, 2019

En la ilustración 33, se obtuvo como resultado que los usuarios tuvieron un tercio de problemas al crear autorizados, ya que les parecía complicado el cambiar el nombre del archivo con el número de cedula del autorizado o no les llegaba el correo para crear dicha autorización, pero el primer inconveniente, se tiene así para que, de esta forma, sea más seguro al crear el autorizado.

¿Presentó dificultad para crear autorizados?

6 respuestas

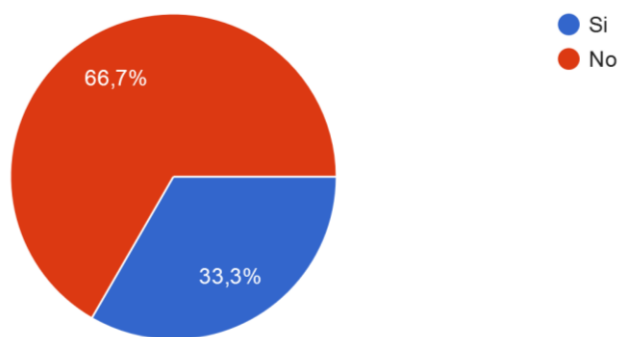


Ilustración 33. Pruebas de aceptación (5)

Fuente: Documentos de Google, visualización de respuestas del formulario, 2019

En la ilustración 34 y 35, se muestran las últimas dos preguntas de aceptación, en la primera se le solicita al usuario que indique si tuvo inconvenientes al crear o modificar el perfil. Para la segunda, se pregunta si tuvo inconveniente al cambiar la contraseña.

¿Presentó dificultad para modificar el perfil?

6 respuestas

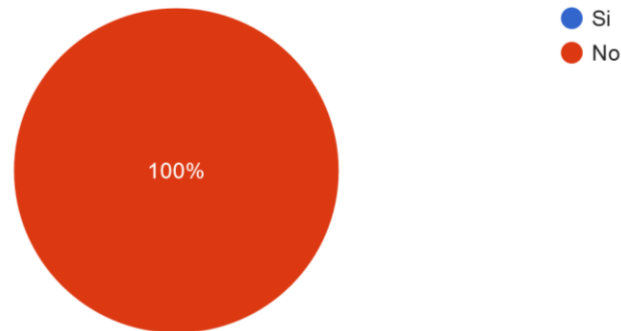


Ilustración 34. Pruebas de aceptación (6)

Fuente: Documentos de Google, visualización de respuestas del formulario, 2019

¿Presentó dificultad para crear cambiar contraseña?

6 respuestas

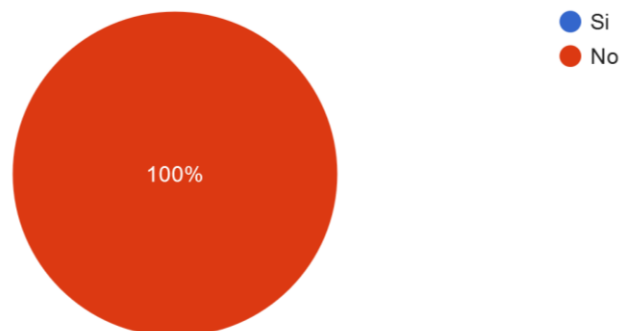


Ilustración 35. Pruebas de aceptación (7)

Fuente: Documentos de Google, visualización de respuestas del formulario, 2019

4.9. Resultados asociados a los requerimientos funcionales

- El requerimiento de implementar las funcionalidades CRUD para la creación, eliminación, actualización y lectura de la información se abarcó exitosamente al proveer servicios de gestión de datos para las guías, paquetes, usuarios, valijas y autorizados.
- Se creó la jerarquía de roles al momento de ingresar al sistema administrativo para restringir las acciones que puede realizar cada usuario.
- Al crear el sistema administrativo con diseño adaptativo se le permite al usuario acceder a los datos a través de dispositivos móviles y de escritorio.
- Al instalar sendmail en el servidor, se puede realizar el envío de notificaciones vía correo electrónico a los usuarios.

- Generar reporte de número total de envíos realizados, así como el porcentaje de aumento o disminución con respecto al mes anterior.
- Al usar Laravel como framework de PHP se garantiza que los datos están siendo tratados de forma privada.

4.10. Resultados asociados a los requerimientos no funcionales

- Usabilidad: se crearon diseños simples e intuitivos para que el usuario pudiera tener un manejo sencillo de la aplicación, realizar consultas de manera intuitiva.
- Rendimiento: el desarrollo se apoyó en tecnologías y herramientas actuales que permitieron crear, modificar, eliminar y consultar los datos de manera simple y óptima con un mínimo procesamiento en todas las peticiones.
- Disponibilidad: se permite que el acceso a la aplicación pueda realizarse en cualquier horario sin inconvenientes ya que todos los servicios están desplegados en un dominio público.
- Mantenibilidad: al desarrollar la aplicación, el sistema administrativo y los servicios de gestión de datos para los recursos, se brinda la posibilidad de realizar cambios en el futuro de manera transparente debido a que la codificación se escribió con un esquema sencillo que permite modificarlo y leerlo fácilmente. Además, se da la libertad de agregar funcionalidades según las necesidades que se tengan.
- Eficiencia: la aplicación se desarrolló con la intención de brindar rapidez y ligereza, esto se cumple gracias a que existe un manejo óptimo de los datos por parte del servidor.
- Confiabilidad: al desarrollar cada servicio y funcionalidad del sistema, se realizaron validaciones en el manejo de datos, peticiones y excepciones para brindar un control de fallos aceptable, los cuales apoyaron las pruebas unitarias y funcionales.
- Portabilidad: gracias a que la aplicación se desarrolló bajo Bootstrap, se permite su ejecución desde distintas arquitecturas (celulares y computadores).

CONCLUSIONES Y RECOMENDACIONES

En esta sección, se presentan las conclusiones del trabajo realizado. Además, se presentan algunas recomendaciones para próximos trabajos.

CONCLUSIONES

El objetivo de este proyecto fue el diseño e implantación de un sistema de gestión, que permitiera el seguimiento de los paquetes, además de tener la posibilidad de crear un autorizado, haciendo uso de un lenguaje de fácil mantenimiento, cumpliendo los objetivos previstos desde el seminario.

A pesar de que no es el lenguaje número uno en el mundo en el desarrollo web, es de muy fácil entendimiento, este permitió desarrollar los funcionamientos necesarios para que la empresa pudiera comenzar con la migración necesaria de un sistema bajo su entendimiento y que no tuviera anti patrones.

Como se pudo observar, se proporcionaron elementos, técnicas y descripción de lenguajes y framework que permitieron desarrollar de manera directa y de forma integrada, tal y como lo preestablece los procesos de metodologías ágiles, en donde no solo llega a el proceso sino a el proyecto como tal.

Para el presente Trabajo Especial de Grado se desarrolló un sistema de seguimiento de paquetería de envíos basado en tres (2) componentes de software, uno de ellos del lado del servidor que contiene la lógica de los datos que, además, permite almacenar la información, el segundo, el sistema administrativo asociado directamente a el servidor y mediante el cual se facilita la gestión de los usuarios, guías y autorizados.

Durante el desarrollo de este Trabajo Especial de Grado fue posible llevar a cabo los objetivos planteados en la fase de investigación por medio de la metodología de desarrollo rápido de aplicaciones. Los objetivos fueron desarrollados, de forma progresiva, a través de la inclusión de nuevos módulos y funcionalidades al sistema en múltiples iteraciones de desarrollo. A su vez, esta metodóloga facilito el proceso de retroalimentación de cada iteración, lo cual conllevó a un proceso más adaptable a la hora de cubrir las necesidades y problemas que surgieron continuamente.

Se aplicó correctamente una metodología ágil, que fue llevada totalmente en la plataforma Bitbucket, en donde la pizarra (como artefacto de la metodología SCRUM) jugo un papel importante, la cual se encargaba de proyectar el estado de las asignaciones o tareas. Además, los problemas prestaban un espacio para abrir discusión y documentar toma de decisiones.

Esta aplicación representa una oportunidad para que los usuarios que compren por internet puedan mantener un seguimiento de sus paquetes, dando una gran oportunidad frente a competidores que solo notifican el estatus del mismo al momento de la llegada al almacén en Venezuela o cuando se realiza una llamada a su sede.

Lo anterior generó como resultado un sistema robusto que dio fe, debido al uso de la versión beta en el semestre I-2019, para que pueda ser usado en un ambiente de producción sin inconveniente alguno.

También debido a las tecnologías usadas podemos garantizar que el sistema se podrá instalar fácilmente en servidores en un futuro inclusive, y la aplicación podrá ser sencillamente modificada por un programador debido a que se usaron lenguajes de programación ampliamente usados y se trabajó siguiendo las directrices establecidas por el framework Laravel.

RECOMENDACIONES

El sistema de gestión cumple con cada requisito y objetivo planteado, de igual manera permite el mejoramiento de proyectos futuros vinculados con este sistema en pro de una mejora e impulso de nuevas tecnologías en Lear Express ®. Es por ello que se presentan algunas recomendaciones para futuros proyectos y desarrollos.

- Agregar la funcionalidad de reclamos e incidencias, siendo esta, muy usada actualmente en manualmente.
- Mantener el enfoque dado por el autor para el desarrollo de futuras funcionalidades.
- Mejorar el panel administrativo para permitir a los usuarios de este rol observar mediante gráficos o tablas información que pueda ser relevante para la toma de decisiones con el objetivo de mejor seguimiento del negocio.

REFERENCIAS BIBLIOGRÁFICAS Y DÍGITALES

- Abrahamsson, B. (1993). *The Logic of Organizations*. Londres: Sage.
- Acosta, A. E. (s.f.). *AgilUs: Construcción ágil de la Usabilidad*. Recuperado el 24 de Julio de 2016, de http://www.ciens.ucv.ve:8080/genasig/sites/interaccion-humano-comp/archivos/234_CLEI_Acosta_Paper.pdf
- Barry, D. K. (2016). *Service Architecture*. Recuperado el 25 de Mayo de 2016, de http://www.service-architecture.com/articles/web-services/service-oriented_architecture_soa_definition.html
- Bootstrap. (s.f.). *Bootstrap. The world's most popular mobile-first and responsive front-end framework*. Recuperado el Mayo de 2016, de <http://getbootstrap.com/>
- Busca palabra. (s.f.). Obtenido de Busca palabra: <https://www.buscapalabra.com/definiciones.html?palabra=paquete>
- Courier. (24 de septiembre de 2011). Obtenido de Courier: <http://courier-pits.blogspot.com/>
- Gibson, J. I. (1997). *Las Organizaciones, comportamientos, estructuras y procesos*. Colombia: Mc Graw Hill.
- Gutiérrez Gomez, I., & Otón Tortosa, S. (2005). *Service Oriente Architecture*. Recuperado el 25 de Mayo de 2016, de <http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-132/paper09.pdf>
- Isabel Milio Balanza, M. C. (2000). *Comercializacion De Productos Y Servicios Turisticos*. Madrid: S.A. Ediciones Paraninfo.
- ISO 9001 calidad. (2019). Obtenido de ISO 9001 calidad: <https://iso9001calidad.com/definicion-de-terminos-586.html>
- Lamb Charles, H. J. (2002). *Marketing*. International Thomson Editores S.A.
- Laudon, K. &. (2014). *Management information systems: Managing the digital firm*. United States: Pearson Education.
- Lear Express. (s.f.). Obtenido de Lear Express: <https://www.learexpress.com>
- Letelier, P., & Penadés, M. C. (s.f.). *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. Recuperado el 25 de Julio de 2016, de https://uvirtual.unet.edu.ve/pluginfile.php/190347/mod_resource/content/0/metodologia_agil_xp_bueno.pdf
- Mendoza, J. L. (2017). Obtenido de http://ri.uaemex.mx/bitstream/handle/20.500.11799/70928/secme-27625_1.pdf;jsessionid=C7B5B5E32900C2156F52C4ED39BAF799?sequence=1

-
- Microsoft. (s.f.). *Visual Studio IDE*. Recuperado el 23 de Julio de 2016, de <https://msdn.microsoft.com/en-us/library/dn762121.aspx>
- Mozilla Developer Network. (s.f.). *Acerca de JavaScript - JavaScript*. Recuperado el Mayo de 2016, de https://developer.mozilla.org/es/docs/Web/JavaScript/Acerca_de_JavaScript
- Mozilla Developer Network. (s.f.). *Introduction - JavaScript*. Recuperado el Mayo de 2016, de https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Introduction#What_is_JavaScript
- Mozilla Developer Network. (s.f.). *JavaScript*. Recuperado el Mayo de 2016, de <https://developer.mozilla.org/es/docs/Web/JavaScript>
- PowerData. (10 de Mayo de 2016). Obtenido de PowerData: <https://blog.powerdata.es/el-valor-de-la-gestion-de-datos/por-que-se-necesita-la-normalizacion-de-base-de-datos>
- Rodríguez, G. Y. (1994). *Ideas preliminares para una propuesta curricular en Educación en Tecnología. Ministerio de Educación Nacional de Colombia*. Santafé de Bogotá.
- Sáez, F. (s.f.). *Facile Things*. Obtenido de Facile Things: <https://facilethings.com/blog/es/peter-drucker-self-management>
- Santesmases Mestre, M. (2012). *Marketing : conceptos y estrategias*. Madrid: Pirámide.
- Santesmases, M. (s.f.). *Términos de Marketing. Diccionario - Base de Datos*. Pirámide.
- Satpathy, T. (2013). *Una guía para el conocimiento de Scrum (Guía SBOK) - 2013 Edición*. Phoenix: SCRUMstudy.
- Source Making. (s.f.). Obtenido de Source Making: <https://sourcemaking.com/antipatterns/spaghetti-code>
- Universidad de Alicante. (s.f.). *Razor. Servicio de Informática ASP.NET MVC 3 Framework*. Recuperado el Julio de 2016, de <http://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/vista/razor.html>
- W3C. (28 de Octubre de 2014). *Introduction - HTML5*. Recuperado el Mayo de 2016, de <https://www.w3.org/TR/html5/introduction.html#introduction>
- W3C. (s.f.). *Guía Breve de CSS*. Recuperado el Mayo de 2016, de <http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo>
- W3C. (s.f.). *Guía Breve de Servicios Web*. Recuperado el 2016, de <http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>