Reingeniería para sistemas legados desarrollados por TuXSTONe Technologies implementando un framework PHP mediante el rediseño del software de contratación web.

Diana Carolina Tarapues Chirivi
Abril 2018.

Universidad Cooperativa de Colombia.
Facultad de Ingeniería.
Ingeniería de Sistemas.
# Tabla de Contenidos

| Capítulo 1 Descripción general de la empresa | ......................................................... 1 |
| Capítulo 2 Requerimiento de la organización | ............................................................. 3 |
| Capítulo 3 Plan de acción | ................................................................. 4 |
| Cronograma de actividades | ................................................................. 7 |
| Capítulo 4 Actividades realizadas | ................................................................. 10 |
| Etapa 1. Elección de un framework php | ............................................................. 10 |
| Búsqueda y comparación de frameworks | ............................................................. 10 |
| Elección de framework para php | ............................................................. 12 |
| Estudio de framework laravel | ................................................................. 13 |
| Documentación del framework | ................................................................. 13 |
| Etapa 2. Estudio del framework bootstrap | .......................................................... 14 |
| Etapa 3. Aplicación metodología de reingeniería | ......................................................... 14 |
| Análisis de inventarios | ................................................................. 14 |
| Reestructuración de documentos | ............................................................. 16 |
| Ingeniería inversa | ................................................................. 17 |
| Ingeniería hacia adelante | ................................................................. 29 |
| Prototipos | ......................................................................... 45 |
| Prototipo interfaz | ......................................................................... 45 |
| Prototipo funciones principales | ......................................................................... 63 |
| Prototipo final | ......................................................................... 71 |
| Etapa 4. Desarrollo de la guía de reingeniería | ......................................................... 84 |
| Etapa 5. Socialización de la guía y del desarrollo de la aplicación al interior de la empresa | ................................................................. 85 |
| Logros y lecciones aprendidas | ................................................................. 86 |
| Limitaciones | ................................................................. 87 |
| Conclusiones | ................................................................. 88 |
| Recomendaciones | ................................................................. 89 |
| Bibliografía | ................................................................. 90 |
| Anexos | ......................................................................... 92 |
Listado de tablas

Tabla 1. Cronograma de actividades................................................................. 7
Tabla 2. Inventario de aplicaciones desarrolladas ............................................. 14
Tabla 3. Análisis de deficiencias aplicaciones desarrolladas ............................ 15
Tabla 4. Diccionario de datos de la aplicación original...................................... 27
Tabla 5. Usuarios del sistema .......................................................................... 32
Tabla 6. Requerimiento radicar solicitud .......................................................... 32
Tabla 7. Requerimiento imprimir solicitud .......................................................... 33
Tabla 8. Requerimiento guardar información solicitud ....................................... 34
Tabla 9. Requerimiento consultar estado de la solicitud .................................... 34
Tabla 10. Requerimiento inicio de sesión ........................................................... 35
Tabla 11. Requerimiento cambio de contraseña ............................................... 36
Tabla 12. Requerimiento crear usuario .............................................................. 37
Tabla 13. Requerimiento listar usuarios ............................................................ 37
Tabla 14. Requerimiento editar usuario ............................................................. 38
Tabla 15. Requerimiento eliminar usuario ......................................................... 38
Tabla 16. Requerimiento filtrar solicitudes ......................................................... 39
Tabla 17. Requerimiento guardar listado solicitudes ......................................... 40
Tabla 18. Requerimiento ver solicitud ............................................................... 40
Tabla 19. Requerimiento pasar siguiente estado una solicitud ......................... 41
Tabla 20. Nuevo diccionario de datos ............................................................... 43
Lista de figuras

Figura 1. Logo de la empresa........................................... 1
Figura 2. Ubicación de la oficina principal, tomada de www.google.com.co .......................... 2
Figura 4. Proceso de ingeniería inversa .................................................. 17
Figura 5. Página registro solicitud aplicación original ........................................... 19
Figura 6. Página solicitud registrada aplicación original ........................................... 19
Figura 7. Página consulta de estado aplicación original ........................................... 20
Figura 8. Estado de la solicitud ................................................................. 20
Figura 9. Página inicio de sesión ................................................................. 21
Figura 10. Diseño preliminar de entidades ................................................... 22
Figura 11. Funcionamiento global del sistema ................................................ 23
Figura 12. Funcionamiento radicación de una solicitud ........................................... 24
Figura 13. Funcionamiento consulta de una solicitud ........................................... 25
Figura 14. Diseño de la base de datos de la aplicación original ...................... 26
Figura 15. Diagrama entidad relación de la aplicación original ...................... 28
Figura 16. Arquitectura de la aplicación ......................................................... 30
Figura 17. Nuevo modelo de datos ..................................................................... 42
Figura 18. Nuevo modelo entidad relación ......................................................... 43
Figura 19. Borrador formulario radicación de una solicitud ...................... 46
Figura 20. Borrador pantalla confirmación de radicación ...................................... 47
Figura 21. Borrador documento en PDF que confirma la radicación de una solicitud .... 48
Figura 22. Borrador formulario consulta y datos solicitud .............................. 48
Figura 23. Borrador formulario inicio de sesión ................................................. 49
Figura 24. Borrador panel de administración ....................................................... 49
Figura 25. Borrador formulario creación de usuario ............................................. 50
Figura 26. Borrador listado de usuarios ............................................................... 51
Figura 27. Borrador formulario para la búsqueda de solicitudes ...................... 51
Figura 28. Borrador listado de solicitudes ........................................................... 52
Figura 29. Borrador detalle solicitud .................................................................... 53
Figura 30. Formulario cambio de estado ............................................................ 54
Figura 31. Formulario radicación de una solicitud ............................................. 55
Figura 32. Página respuesta a la radicación ....................................................... 55
Figura 33. Documento PDF con la información de la solicitud .................... 56
Figura 34. Página en la que el cliente consulta una solicitud ......................... 57
Figura 35. Página para inicio de sesión ............................................................... 57
Figura 36. Panel de administración del sistema ............................................... 58
Figura 37. Página formulario creación de usuario ............................................. 58
Figura 38. Lista de usuario .............................................................................. 59
Figura 39. Búsqueda y lista de solicitudes .......................................................... 59
Figura 40. Información de una solicitud ............................................................. 60
Figura 41. Modificación interfaz consulta de una solicitud ............................ 61
Figura 42. Diagrama arquitectura prototipo interfaz ......................................... 62
Figura 43. Diagrama de validación formulario registro de solicitud .................. 66
Figura 44. Mensajes validación formulario registro solicitud. ........................................66
Figura 45. Diagrama registro solicitud ........................................................................... 67
Figura 46. Diagrama respuesta solicitud radicada ........................................................... 68
Figura 47. Diagrama búsqueda solicitud ........................................................................... 69
Figura 48. Diagrama arquitectura prototipo funciones principales .............................. 71
Figura 49. Diagrama inicio de sesión .............................................................................. 74
Figura 50. Diagrama acceso por tipo de usuario .............................................................. 74
Figura 51. Diagrama crear usuario ................................................................................... 76
Figura 52. Diagrama lista usuarios .................................................................................. 76
Figura 53. Diagrama formulario búsqueda de solicitudes ............................................. 77
Figura 54. Archivo Excel generado por la aplicación ...................................................... 78
Figura 55. Diagrama solicitud archivo Excel ................................................................. 79
Figura 56. Diagrama consulta solicitud en módulo administración ............................ 80
Figura 57. Primera ventana asignación nuevo estado .................................................... 81
Figura 58. Segunda ventana (formulario) asignación nuevo estado ............................ 81
Figura 59. Diagrama asignación nuevo estado ............................................................... 82
Figura 60. Filtro de búsqueda modificado ..................................................................... 83
Figura 61. Lista de solicitudes modificado ..................................................................... 84
Capítulo 1

Descripción general de la empresa

TuXSTONe Technologies

Fundada en Agosto de 2001, es una empresa de consultoría especializada en Ingeniería, Tecnologías de Información (IT), Comunicaciones, Seguridad Informática para proveer soluciones que se fundamentan en la integración de sistemas y plataformas libres y comerciales en entornos corporativos, privados y públicos.

Figura 1. Logo de la empresa

Su misión es “Proveemos soluciones integrales e innovadoras en Tecnologías de Información y Comunicaciones, anticipando y superando las necesidades de nuestros clientes con los más altos niveles de calidad, confiabilidad y seguridad, y permaneciendo en constante desarrollo tecnológico apoyados en nuestro talento humano, la generación de valor a través del conocimiento y nuestra continua evolución personal y profesional” (TuXSTONe Ltd., 2001-2018).

Y su visión es “Para el año 2020 seremos el núcleo del clúster empresarial líder en innovación y desarrollo tecnológico regional con la confianza y fidelidad de nuestros colaboradores, aliados y clientes y siendo un significativo aliado en la realización de proyectos de vida para cada uno de ellos” (TuXSTONe Ltd., 2001-2018).
Ubicación

TuXSTONe Technologies está ubicada en Villavicencio, capital del departamento del Meta. Su oficina principal se encuentra en el Centro Comercial Villacentro, sobre la avenida 40 con calle 15, oficina 404 (Figura 2)

*Figura 2. Ubicación de la oficina principal, tomada de www.google.com.co*
Capítulo 2
Requerimiento de la organización

La empresa TuXSTONe Technologies ha desarrollado diversas aplicaciones web para sus clientes, las cuáles se desarrollaron ya hace varios años y aún son utilizadas. Dada la naturaleza de la web, el internet, y las herramientas en general para el desarrollo de estas tecnologías, hace que la desactualización del software sea muy rápida. Esto implica que la organización realice un proceso de reingeniería de software a sus desarrollos como son sistemas de Contratación, PQR, sistema de gestión académica, software de inventario y aprovisionamiento SBA ETB.

Para realizar este proceso de reingeniería se quiere hacer primero en uno de sus desarrollos y de esta experiencia replicar a los sistemas con características y necesidades similares. Para este fin se realizará un proceso de reingeniería de software sobre el módulo de PQR para el Hospital Departamental de Granada.

Este proceso de reingeniería busca que se cumplan los siguientes objetivos:

**Objetivo general**
Ejecutar un proceso de reingeniería orientado al rediseño de las aplicaciones desarrolladas por TuXSTONe, Technologies mediante el uso de un framework PHP

**Objetivos específicos**
- Evaluar alternativas de framework
- Implementar una metodología de reingeniería para el software de contratación utilizando el framework seleccionado
- Documentar y socializar el proceso de desarrollo y reingeniería usando el framework al interior de la empresa.
Capítulo 3
Plan de acción

Descripción

Con el fin de lograr los objetivos planteados se dividió el desarrollo del proyecto en 5 etapas:

1) Elección de un framework para lenguaje PHP.
2) Aprendizaje del framework bootstrap.
3) Aplicación metodología de reingeniería.
4) Desarrollo de la guía para reingeniería.
5) Socialización de la guía y del desarrollo de la aplicación al interior de la empresa.

Al finalizar las etapas propuestas la empresa TuXSTONe Technologies obtuvo el software de Contratación rediseñado y cuenta con una guía para la reingeniería del software desarrollado por la empresa.

1) **Elección de un framework para lenguaje PHP.**

En esta etapa se buscó información sobre los frameworks para PHP más utilizados para trabajar con este lenguaje. Se analizaron las características principales de estos, con el fin de identificar el que sea apropiado y se adapte mejor a los requerimientos del software a rediseñar.

Una vez elegido el framework se estudió y documentó a detalle para que al momento de implementarlo se aprovechen mejor sus características.

2) **Aprendizaje del framework bootstrap.**

Puesto que ya se ha definido que las aplicaciones deben responder al diseño adaptativo, el framework bootstrap permite de la forma más eficiente cumplir con este objetivo sin hacer más compleja y extensa la tarea de codificación.

3) **Aplicación metodología de reingeniería.**

Se utilizó el modelo de reingeniería de software (Pressman, 2010, p.661) que lleva el siguiente esquema:
Pressman (2010) propone que “El paradigma de reingeniería que se muestra en la figura 3 es un modelo cíclico. Esto significa que cada una de las actividades presentadas como parte del paradigma puede revisarse. Para algún ciclo particular, el proceso puede terminar después de cualquiera de estas actividades” (p. 662).

**Análisis de inventarios:** Es un inventario de todas las aplicaciones de la organización, para el objetivo de este proyecto fueron las aplicaciones desarrolladas por la organización. Puede ser en un formato de hojas de cálculo que ofrezca una descripción detallada (tamaño, edad, soporte, importancia empresarial, etc.) de cada aplicación activa. Con este se identificaron los posibles candidatos para reingeniería.

**Reestructuración de documentos:** En esta actividad se determinó si era conveniente o no, documentar el software legado. De todas formas se debe tratar de documentar a lo mínimo requerido, ya que es una tarea que consume demasiados recursos.

**Ingeniería inversa:** Aquí se estudió el software con el fin de recuperar el diseño con el que fue desarrollado. Se obtuvo información de la arquitectura y de los procedimientos que empleaba.

**Reestructuración del código:** Aunque el software pueda estar diseñado con una buena arquitectura, su comportamiento interno puede tener fallas o no cumplir con los estándares actuales. En esta tarea se analizó el código fuente y se mejoró.
Reestructuración de datos: En esta actividad se hizo un análisis de los datos (obtener los objetos, información del flujo de datos y entender su estructura). Se pasó después al rediseño de datos, que puede ser en su forma más simple buscando un estándar en la forma de nombrar los datos por ejemplo o cambiando la forma física de almacenarlos, por ejemplo cambiar de un formato de archivo a otro o cambiando de una base de datos a otra.

En general la reestructuración de código y datos no modifica la arquitectura global del programa, si esto llega a necesitarse ya se debería hablar de ingeniería hacia adelante.

**Ingeniería hacia adelante:** En la mayoría de los casos se hace uso del sistema actual agregándole mejoras y nuevas funcionalidades. En otros casos se debe hacer casi todo el desarrollo desde cero. Aquí se empleó toda la información obtenida en las etapas anteriores para implementar una metodología de desarrollo.

Aplicación metodología de desarrollo (Modelo por prototipos): Es un modelo de desarrollo cíclico evolutivo que llevó las siguientes etapas:

- **Plan rápido:** recolección y definición de requerimientos.
- **Modelado – diseño rápido:** Modelo de datos, arquitectura del sistema, diseño de la interfaz.
- **Construcción del prototipo.** En esta tarea se utilizaron los frameworks estudiados, bootstrap para el desarrollo de interfaz adaptativa y el framework para lenguaje php para la codificación del lado del servidor.
- **Despliegue, entrega y retroalimentación.** Evaluación por parte del cliente.
- **Comunicación.** Refinamiento del prototipo.

4) **Desarrollo de la guía para reingeniería**

Se documentó en forma de guía para la reingeniería del software, todo el proceso de rediseño y reconstrucción aplicado para el software de contratación, con el objetivo que se utilice como marco de trabajo para posteriores tareas de actualización del software que ha desarrollado la empresa. Aquí también encontrarán la documentación del framework php elegido para el proyecto.
5) Socialización de la guía y del desarrollo de la aplicación al interior de la empresa.

Se explicó al interior de la empresa la metodología elegida y su implementación. La apropiación del framework para php es fundamental, ya que posiblemente lo utilicen para sus desarrollos futuros. De igual manera con la empresa hubo una comunicación constante ya que ellos conocen el software desarrollado y gracias a su amplia experiencia guiaron el desarrollo de este proyecto.

Cronograma de actividades

*Tabla 1. Cronograma de actividades.*

<table>
<thead>
<tr>
<th>ACTIVIDADES</th>
<th>Tiempo de ejecución en semanas</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>ETAPA 1: Elección de un framework PHP</td>
<td></td>
</tr>
<tr>
<td>Búsqueda y comparación de frameworks para PHP</td>
<td></td>
</tr>
<tr>
<td>Elección de framework para PHP</td>
<td></td>
</tr>
<tr>
<td>Estudio de framework para PHP</td>
<td></td>
</tr>
<tr>
<td>Documentación del framework</td>
<td></td>
</tr>
<tr>
<td>ETAPA 2: Aprendizaje del framework bootstrap</td>
<td></td>
</tr>
<tr>
<td>Estudio y practica del framework</td>
<td></td>
</tr>
<tr>
<td>ETAPA 3: Aplicación metodología reingeniería</td>
<td></td>
</tr>
<tr>
<td>Análisis de inventarios</td>
<td></td>
</tr>
<tr>
<td>Reestructuración de documentos</td>
<td></td>
</tr>
<tr>
<td>-------------------------------</td>
<td>--</td>
</tr>
<tr>
<td>Ingeniería inversa</td>
<td></td>
</tr>
<tr>
<td>Reestructuración del código</td>
<td></td>
</tr>
<tr>
<td>Reestructuración de datos</td>
<td></td>
</tr>
<tr>
<td><strong>Ingeniería hacia adelante (Modelo Prototipos)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Prototipo Interfaz</strong></td>
<td></td>
</tr>
<tr>
<td>Plan rápido</td>
<td></td>
</tr>
<tr>
<td>Modelado</td>
<td></td>
</tr>
<tr>
<td>Construcción del prototipo</td>
<td></td>
</tr>
<tr>
<td>Despliegue, entrega,</td>
<td></td>
</tr>
<tr>
<td>retroalimentación</td>
<td></td>
</tr>
<tr>
<td>Comunicación</td>
<td></td>
</tr>
<tr>
<td><strong>Prototipo Funciones principales</strong></td>
<td></td>
</tr>
<tr>
<td>Plan rápido</td>
<td></td>
</tr>
<tr>
<td>Modelado</td>
<td></td>
</tr>
<tr>
<td>Construcción del prototipo</td>
<td></td>
</tr>
<tr>
<td>Despliegue, entrega,</td>
<td></td>
</tr>
<tr>
<td>retroalimentación</td>
<td></td>
</tr>
<tr>
<td>Comunicación</td>
<td></td>
</tr>
<tr>
<td><strong>Prototipo Final</strong></td>
<td></td>
</tr>
<tr>
<td>Plan rápido</td>
<td></td>
</tr>
<tr>
<td>Modelado</td>
<td></td>
</tr>
<tr>
<td>Construcción del prototipo</td>
<td></td>
</tr>
<tr>
<td>Despliegue, entrega,</td>
<td></td>
</tr>
<tr>
<td>retroalimentación</td>
<td></td>
</tr>
<tr>
<td>Comunicación</td>
<td></td>
</tr>
</tbody>
</table>

ETAPA 4: Desarrollo de la guía de reingeniería
<table>
<thead>
<tr>
<th>Documentación de la guía para reingeniería de software</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Integración documentación del framework para PHP</td>
<td></td>
</tr>
<tr>
<td><strong>ETAPA 5: Socialización de la guía y del desarrollo de la aplicación al interior de la empresa.</strong></td>
<td></td>
</tr>
<tr>
<td>Socialización de metodología de reingeniería</td>
<td></td>
</tr>
<tr>
<td>Socialización del uso del framework para PHP</td>
<td></td>
</tr>
</tbody>
</table>
Capítulo 4

Actividades realizadas

Según el cronograma de actividades se enumeran en el orden programado las primeras actividades realizadas:

Etapa 1. Elección de un framework php

Búsqueda y comparación de frameworks.

Para el lenguaje php existe gran variedad de frameworks ya que es uno de los lenguajes más requeridos para el desarrollo de un sitio web. Se seleccionaron los tres frameworks más reconocidos para realizar la comparación:

**CodeIgniter**

Es uno de los frameworks más utilizados, se encuentra bajo una licencia open source, es muy liviano y tiene una buena documentación.

**Ventajas**

* En su nueva versión CodeIgniter 4 soportará el uso de namespaces PSR-4 y requerirá de php 7.
* Es compatible con versiones antiguas de php. En este momento es compatible desde la versión 4 de php.
* Tiene una comunidad bastante active.

**Desventajas**

* No tiene un manejador de dependencias para php.
* No tiene una interfaz de comandos para interactuar con el framework.
* Para actualizar el framework se debe descargar la nueva versión y modificar la carpeta del sistema.
* Al ser compatible con versiones antiguas de php hace que no sea un framework realmente orientado a objetos.
* No implementa los patrones de organización de código PSR-2.
* Su ORM no es su fuerte, lo que parece tener es un constructor de consultas.
**Symfony**

Es un robusto framework que trabaja muy bien para complejas y grandes aplicaciones empresariales.

**Ventajas**

* Está orientado a la programación por objetos (POO)
* Maneja inyección de dependencias.
* Swiftmailer Librería para el envío de correos electrónicos.
* Doctrine es su ORM, muy completo en donde se accede a las tablas de la base de datos por medio de objetos, tanto para la inserción o la consulta de datos.
  * Si se utiliza apropiadamente es más rápido que otros frameworks.
  * Compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft
* Se puede ejecutar en plataformas Unix como en Windows.

**Desventajas**

* Su curva de aprendizaje es compleja. Para personas que estén iniciando su estudio puede ser muy difícil.
  * La configuración de la base de datos relacional puede ser engorroso.

**Laravel**

Es un framework muy completo, cumple las normas de estandarización de código y cuenta con una comunidad muy grande, activa y en crecimiento.

**Ventajas**

* Tiene una excelente documentación.
* Trabaja de la mano de Composer, manejador de dependencias para php.
* Tiene una interfaz de comandos (artisan) para la creación de código de forma automática
  * Para tener las nuevas versiones del framework se actualiza composer y este renueva todas las dependencias del framework como las de terceros que se esté utilizando el proyecto.
* Tiene un ORM (Eloquent) muy robusto y en constante actualización. Eloquent trabaja bajo el patrón Active Record en el que se interactúa con las tablas de la base de datos como si se trataran de clases y con las filas como si fueran objetos.
* Se encuentran muchos paquetes disponibles para laravel y mucho material para aprender a utilizar el framework.
* La documentación es uno de sus fuertes, que a pesar de estar en inglés es clara y precisa.
* Sistema de plantillas (Blade) que hace el código más limpio y más rápido en ejecución.

**Desventajas**
* Algunos métodos para el enrutamiento inverso pueden ser complejos.
* Está pensado para las nuevas versiones de php y solo soporta desde PHP 5.4

**Elección de framework para php.**

Al consultar sobre los frameworks para php más recomendables en este momento, se encontró que CodelIgniter, Symfony y Laravel son la tendencia para desarrollo web con este lenguaje.

Los tres tienen características a favor y en contra. Pero entre ellos el que más presenta desventajas es CodeIgniter, por ser un framework que aunque lleva más tiempo en el mercado no presenta las características de un framework que cumpla con la estandarización de código y no cuenta con herramientas potentes como un ORM, un manejador de plantillas o un inyector de dependencias que si tienen Laravel y Symfony. Es de fácil instalación, rápido aprendizaje y arquitectura sencilla, pero para tratar de desarrollar aplicaciones que requieran más que tres capas como son MVC, se queda muy atrás.

Laravel y Symfony tienen características muy similares y buenos comentarios en foros y cursos en la web. Los desarrolladores de php recomiendan mucho estos frameworks para desarrollar aplicaciones tanto pequeñas como de gran escala. La diferencia marcada
entre estos dos es su curva de aprendizaje, siendo la de symfony un reto entender y aprenderlo al principio mientras que laravel es mucho más sencillo y rápido.

Teniendo en cuenta estas características y la necesidad del proyecto se elige Laravel como framework del lenguaje php para el desarrollo de este proyecto. Laravel es el más adecuado por las herramientas que posee, la hermosa documentación, su activa y en crecimiento comunidad en la web, su rápido aprendizaje y por que cumple con el requerimiento de este proyecto al permitir la actualización de aplicaciones y su posterior mantenimiento.

**Estudio de framework laravel.**

Se hace un estudio inicial del framework por medio de su documentación. Para que se entiendan las bases del framework se hace un curso básico con el cual se desarrolla una pequeña aplicación en donde se maneja solo unas capas de su arquitectura pero que aplican los componentes más importantes como son la configuración básica de una aplicación, su ORM que es Eloquent, Blade su administrador de plantillas, la interfaz de comandos artisan, su administrador de dependencias via composer insertando depencias de terceros, migraciones a la base de datos, creación de helpers, enrutamiento, controladores, validators y middleware.

**Documentación del framework.**

Se escribe un documento con los conceptos básicos del framework, sus componentes, la arquitectura de una solicitud, instalación, configuración de la aplicación y el manejo básico del MVC. No se detalla las funcionalidades y opciones de laravel, ya que esto se encuentra en la documentación en su página web https://laravel.com. En este mismo documento se ha escrito todo un primer ejercicio práctico, basándose en la aplicación solicitada que es de PQR.
Etapa 2. Estudio del framework bootstrap

Se estudia las funcionalidades de este framework por medio de la página [https://www.w3schools.com/bootstrap/default.asp](https://www.w3schools.com/bootstrap/default.asp), se hacen algunos ejercicios prácticos y se empieza a crear una plantilla simple para la aplicación.

Etapa 3. Aplicación metodología de reingeniería

**Análisis de inventarios.**

Como se pudo ver en el plan de acción el análisis de inventarios es un recuento (para los objetivos de este proyecto) de las aplicaciones desarrolladas por la empresa. Se crean dos formatos uno como inventarios de las aplicaciones activas y otro de las deficiencias que presentan estas aplicaciones. Por medio de estos se podrá determinar los posibles candidatos a un proceso de reingeniería.

**Tabla 2. Inventario de aplicaciones desarrolladas.**

<table>
<thead>
<tr>
<th>Aplicación</th>
<th>Tipo</th>
<th>Tamaño</th>
<th>Longevidad</th>
<th>Importancia empresarial</th>
</tr>
</thead>
<tbody>
<tr>
<td>SBA / Sistema aprovisionamiento</td>
<td>Web</td>
<td>500Mb</td>
<td>5 años</td>
<td>Muy Alta</td>
</tr>
<tr>
<td>Modulo PQR</td>
<td>Web</td>
<td>2M</td>
<td>10 años</td>
<td>Media</td>
</tr>
<tr>
<td>Modulo Contratacion</td>
<td>Web</td>
<td>2M</td>
<td>10 años</td>
<td>Media</td>
</tr>
<tr>
<td>Campus</td>
<td>Web</td>
<td>100M</td>
<td>8 años</td>
<td>Media</td>
</tr>
</tbody>
</table>
Tabla 3. Análisis de deficiencias aplicaciones desarrolladas.

<table>
<thead>
<tr>
<th>Responsable</th>
<th>Fecha</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Nombre del funcionario encargado)</td>
<td>año mes día</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Aplicación</th>
<th>D1</th>
<th>D2</th>
<th>D3</th>
<th>D4</th>
<th>D5</th>
<th>D6</th>
<th>D7</th>
</tr>
</thead>
<tbody>
<tr>
<td>SBA / Sistema aprovisionamiento</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modulo PQR</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Modulo Contratacion</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Campus</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

A continuación se listan y describen los tipos de deficiencias utilizadas en el análisis de la tabla 3:

- Portabilidad (D1)  Ejecución en diferentes sistemas operativos.
- Seguridad (D2)  Inserción de publicidad ajena al negocio.
- Código fuente (D3)  Perdida de código fuente.
- Lenguaje desactualizado (D4) Desarrollo en un lenguaje y con herramientas desactualizadas, utilizando técnicas y estilos de programación distintos a los actuales.
- Código (D5)  El código existente presenta varios síntomas indeseables que afectan su legibilidad y por ende su mantenimiento: modularización del código baja; presencia de segmentos de código repetido; falta estándares de codificación; designación de nombres de variables y módulos no significativos; etc.
- Documentación técnica (D6)  Inexistencia o poca documentación técnica.
- Documentación operativa (D7)  Inexistencia o poca documentación operativa del sistema.
En la tabla 2 se puede observar que son aplicaciones con una longevidad considerable y por ser aplicaciones web al tiempo de hoy demandan características que no poseen y que se ven reflejadas en las deficiencias que presentan:

* No son aplicaciones que se adapten a diferentes dispositivos y por tal a diferentes sistemas operativos.
* Presentan problemas de seguridad al permitir la inserción de publicidad ajena al negocio.
* Código fuente desorganizado sin una arquitectura definida.
* Poca o inexistente documentación.

Estas características demuestran que estas aplicaciones son candidatas a un proceso de reingeniería, por esto se debe realizar una guía de trabajo general y específico que pueda ser implementado en cada sistema.

De estas aplicaciones se elige la aplicación PQR (Peticiones, quejas y reclamos) del Hospital Departamental de Granada, como primer ejercicio y posterior guía de reingeniería para las aplicaciones desarrolladas por la empresa Tuxstone Technologies.

**Reestructuración de documentos.**

En esta actividad se debe determinar si es conveniente o no documentar la aplicación. De todas formas se debe tratar de documentar a lo mínimo requerido, ya que es una tarea que consume demasiados recursos. Las opciones pueden ser:

1. La creación de la documentación consume demasiado tiempo. La aplicación no sufrirá cambios significativos, funciona sin inconvenientes, así que se puede elegir no invertir recursos en documentación.

2. La documentación debe actualizarse, pero la organización tiene recursos limitados. Se puede documentar las secciones del sistema que sufran cambios, se intenta documentar solo cuando sea necesario.

3. El sistema tiene importancia empresarial y debe volver a documentarse por completo. Incluso si se debe tomar esta opción, lo más conveniente es documentar lo mínimo, sólo lo que sea necesario.
De estas opciones las que más convienen a la empresa es documentar sólo cuando sea necesario, intentado no centrar recursos en esta parte. Pero si la aplicación sufre cambios importantes, de estos se hará documentación técnica y operativa.

**Ingeniería inversa.**

Aquí se estudia el software con el fin de recuperar el diseño con el que fue desarrollado. Se obtiene información de la arquitectura, de los procedimientos que emplea y de la interfaz de usuario. Siguiendo la metodología de reingeniería de Roger Pressman, en este apartado seguiremos su modelo de proceso de ingeniería inversa (Figura 4).

![Figura 4. Proceso de ingeniería inversa](#)

---

Reestructuración del código

En la reestructuración de código se hace a un lado la arquitectura y se revisan los módulos individuales para detectar código que no se pueda entender, el uso inadecuado de variables o técnicas de codificación inadecuadas que dificultan las pruebas y los posteriores mantenimientos. Se codifica de nuevo los módulos que lo requieran y se verifica que no se introdujeron cambios al sistema.

Para el caso de este proyecto no aplica la reestructuración de código, a pesar de que se utilice el mismo lenguaje. El lenguaje ha cambiado ya que al momento de la implementación de la aplicación era un lenguaje estructurado y ahora es un lenguaje orientado a objetos. Igualmente el framework laravel es orientado a objetos y sumado a la sintaxis propia de este y a los nuevos requerimientos de la aplicación, el código y su estructura cambian radicalmente.

 Esto nos lleva a desarrollar la aplicación desde ceros, sería tiempo perdido hacer mejoras a un código que no se puede reutilizar.

Extracción de abstracciones

En la ingeniería inversa la extracción de abstracciones es fundamental ya que obtenemos información del diseño. Se obtienen representaciones del diseño enfocado a los procedimientos (abstracción de nivel bajo), información de la estructura del programa y de los datos (abstracción de nivel medio), modelo de datos, de objetos y/o flujo de control (abstracción de nivel medio alto) y modelos de entidad relación (abstracción de nivel alto).

Por medio de esta información se facilita la comprensión del programa.

Abstracción de interfaz

En esta primera iteración se interactúa con el módulo de radicación y consulta de una solicitud, los formularios y reportes generados por el sistema.

A continuación se muestran imágenes de la interfaz del módulo de radicación y consulta de una solicitud:
**Figura 5.** Página registro solicitud aplicación original

**Figura 6.** Página solicitud registrada aplicación original
Figura 7. Página consulta de estado aplicación original

Figura 8. Estado de la solicitud
Figura 9. Página inicio de sesión

Requerimientos funcionales

Del análisis de la interfaz se logra una representación de los requerimientos funcionales los cuáles se especifican a continuación:

Actores

- Cliente: es la persona que radica una solicitud. Esta persona no tiene una cuenta de usuario en el sistema.
- Administrador: es la persona que puede acceder a las solicitudes radicadas y cambiarles su correspondiente estado. Tiene una cuenta de usuario en el sistema.

Radicación

- La radicación de una solicitud se hace por medio de un formulario, que solicita los datos correspondientes de la solicitud y de la persona quien la radica.
- Se le informa al cliente que la radicación fue exitosa y se le da a conocer el código y clave generado para la consulta de estado de la solicitud.

Consulta de solicitud

- Por medio del código y clave se consulta el estado de una solicitud.
- Se imprime en pantalla los datos más importantes de la solicitud y los estados por los que ha pasado.
Acceso al sistema
- Para acceder al sistema el administrador accede por la página inicio de sesión en donde digite sus datos de usuario.

Manejo de solicitudes
- El administrador genera un listado de las solicitudes registradas y puede acceder a cada una de ellas para poder actualizar su estado.
- El administrador puede generar un archivo en Excel que lleva las solicitudes que resultaron en el listado.

Diseño preliminar de entidades
De la interfaz analizada y los requerimientos funcionales observados, se hace un diseño de posibles entidades para el desarrollo del sistema. En esta primera iteración se interactúa con el módulo de radicación y consulta de una solicitud, los formularios y reportes generados por el sistema.

![Figura 10. Diseño preliminar de entidades.](image-url)
Abstracción de procedimientos

Aquí se trata de entender y extraer las abstracciones de procedimientos con el código fuente. Esto se puede hacer en varios niveles, el principal es el del funcionamiento global del sistema como vemos en la figura 11.

Figura 11. Funcionamiento global del sistema.

A Este sistema cuenta con dos módulos el de radicación y consulta de solicitudes y el módulo de administración del sistema en donde se hace control de usuarios y se administra los estados de una solicitud.

Se puede ver que es un sistema que no utiliza una arquitectura definida, maneja una programación procedimental, en donde la presentación, lógica y conexión a la base de datos se hace desde un mismo archivo en cada módulo, como se ve en las figuras 12 y 13.
Figura 12. Funcionamiento radicación de una solicitud.
Figura 13. Funcionamiento consulta de una solicitud.
Abstracción de datos

Se trata de abstraer la definición de objetos globales. Se hace a nivel interno definiendo las clases de objetos y a nivel de estructura de la base de datos.

La estructura de datos internos no aplica a este desarrollo puesto que la programación no ha sido orientada a objetos y no se definen en el código clases o modelos de los datos administrados en la base de datos.

La estructura de la base de datos no maneja relaciones foráneas (figura 14), estos controles se hicieron a nivel de código en la aplicación.

Figura 14. Diseño de la base de datos de la aplicación original.

De esta información también se genera un diccionario de datos (tabla 4)
**Tabla 4. Diccionario de datos de la aplicación original.**

<table>
<thead>
<tr>
<th>Tabla</th>
<th>Llave</th>
<th>Columna</th>
<th>Tipo</th>
<th>Nulo</th>
<th>Descripción</th>
<th>Predeterminado</th>
</tr>
</thead>
<tbody>
<tr>
<td>estados</td>
<td>PK</td>
<td>estado</td>
<td>int(11)</td>
<td>No</td>
<td>Código único de un estado</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>descripcion</td>
<td>varchar(80)</td>
<td>No</td>
<td>Nombre del estado</td>
<td></td>
</tr>
<tr>
<td>sesiones</td>
<td>PK</td>
<td>session</td>
<td>int(11)</td>
<td>No</td>
<td>Código único de una sesión</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>usuario</td>
<td>char(100)</td>
<td>No</td>
<td>Nombre usuario</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>hash</td>
<td>char(100)</td>
<td>No</td>
<td>Clave de acceso (encriptada)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ip</td>
<td>char(30)</td>
<td>No</td>
<td>ip</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>conexion</td>
<td>timestamp</td>
<td>No</td>
<td>Fecha y hora de inicio de sesión (CURRENT_TIMESTAMP)</td>
<td></td>
</tr>
<tr>
<td>solicitud</td>
<td>PK</td>
<td>solicitud</td>
<td>int(11)</td>
<td>No</td>
<td>Código único radicado PQR</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>fecha</td>
<td>date</td>
<td>No</td>
<td>Fecha de radicación de la solicitud</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>identificacion</td>
<td>varchar(20)</td>
<td>No</td>
<td>Número documento identificación persona quien radica la solicitud</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>nombre</td>
<td>varchar(200)</td>
<td>No</td>
<td>Nombre persona quien radica la solicitud</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>correo</td>
<td>varchar(200)</td>
<td>No</td>
<td>Dirección de correo electrónico persona quien radica la solicitud</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>texto</td>
<td>text</td>
<td>No</td>
<td>Argumento de la solicitud</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FK</td>
<td>tipo_solicitud</td>
<td>int(11)</td>
<td>No</td>
<td>Tipo de solicitud</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>archivo</td>
<td>varchar(200)</td>
<td>No</td>
<td>Nombre del archivo adjunto como argumento de la solicitud</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FK</td>
<td>estado</td>
<td>int(11)</td>
<td>No</td>
<td>Estado actual de la solicitud</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>fecha_revision</td>
<td>date</td>
<td>No</td>
<td>Fecha de revisión de la solicitud</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>fecha_respuesta</td>
<td>date</td>
<td>No</td>
<td>Fecha en que se contesta la solicitud radicada</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>clave</td>
<td>varchar(30)</td>
<td>No</td>
<td>Clave única de acceso a la solicitud</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>telefono</td>
<td>varchar(30)</td>
<td>No</td>
<td>Número de teléfono persona quien radica la solicitud</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>respuesta</td>
<td>text</td>
<td>No</td>
<td>Respuesta que se da a la solicitud</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>archivo_respuesta</td>
<td>varchar(200)</td>
<td>No</td>
<td>Archivo de respuesta a la solicitud</td>
<td></td>
</tr>
<tr>
<td>tipos_solicitud</td>
<td>PK</td>
<td>tipo</td>
<td>int(11)</td>
<td>No</td>
<td>Código único de un tipo de solicitud</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>descripcion</td>
<td>varchar(80)</td>
<td>No</td>
<td>Nombre del tipo de solicitud</td>
<td></td>
</tr>
<tr>
<td>usuarios</td>
<td></td>
<td>id</td>
<td>int(11)</td>
<td>No</td>
<td>Código de identificación de un usuario</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>nombre</td>
<td>char(140)</td>
<td>No</td>
<td>Nombre del tipo de usuario</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>password</td>
<td>char(50)</td>
<td>No</td>
<td>Contraseña de un usuario</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>usuario</td>
<td>char(100)</td>
<td>No</td>
<td>Nombre corto de un usuario</td>
<td></td>
</tr>
</tbody>
</table>
Se realiza un diagrama de entidad relación para un mejor entendimiento del diseño de la aplicación original (Figura 15)

Figura 15. Diagrama entidad relación de la aplicación original.

Refinamiento y simplificación

Con el desarrollo de los dos primeros pasos de la ingeniería inversa (reestructuración de código y extracción de abstracciones) se obtiene una versión mejorada del código original y una especificación inicial de la estructura de la aplicación. Con la reestructuración de datos se modifica el código fuente y el modelo de datos, con la intención de que se sean susceptibles a cambios en el futuro. Pero la reestructuración que exige este proyecto modifica la arquitectura global de sistema, haciendo que sea necesario pasar inmediatamente a la ingeniería hacia adelante. “En general la reestructuración de código y datos no modifica la arquitectura global del programa, si esto llega a necesitarse ya se debería hablar de ingeniería hacia adelante” (Pressman, R, 2010).
**Ingeniería hacia adelante.**

En el refinamiento y simplificación de la fase de ingeniería inversa se determinó que los cambios a la aplicación afectan la arquitectura global del sistema y por tal, se pasa a la ingeniería hacia adelante. A continuación se detalla el nuevo diseño del sistema.

** Diseño del sistema **

Descripción técnica

El sistema será una aplicación web desarrollada utilizando el framework **laravel**, la cual será adaptable a diferentes dispositivos de hardware que tengan acceso a internet, de fácil y entendible manejo para diferentes tipos de personas que la utilicen.

**Lenguajes de programación**

Para el desarrollo de la interfaz se utilizará los lenguajes de programación para la web **HTML5** y **CSS3**. Que trabajarán de la mano con **javascript** y **jquery**. Igualmente por medio del framework **bootstrap** se conseguirá que la aplicación web sea adaptativa.

Del lado del servidor estará el framework laravel versión 5.2 y el lenguaje PHP en su versión 5.5.

**Base de datos**

Como manejador de la base de datos se utilizará **MYSQL** versión 5.6

**Arquitectura del sistema**

Laravel es un framework que implica o induce al manejo de una arquitectura organizada y definida para sus aplicaciones. Aunque se hace referencia de que este framework maneja una arquitectura **MVC**, en realidad sus creadores no lo definen así, “MVC no es una arquitectura” (Styde.net, 2017).

Basándonos en el framework laravel y utilizando lo más básico de su arquitectura, nuestro proyecto actuaría como vemos en la figura 16
Figura 16. Arquitectura de la aplicación.

Sin tener en cuenta el funcionamiento que debe llevar a cabo el framework, esta aplicación sería de arquitectura de seis capas, definidas en orden de funcionamiento así:

1. Capa enrutamiento: que se encuentra en el archivo app/Http/routes.php y será el encargado de direccionar las diferentes peticiones de un actor del sistema. Las peticiones son enrutadas a los correspondientes controladores.

2. Capa controladores: Ubicados en el directorio app/Http/Controllers Los controladores manejan la lógica de la aplicación. Son la conexión entre los filtros, validaciones de formularios y conexión a la base de datos para generar o llamar la vista solicitada. Cada recurso que maneje el sistema tendrá su propio controlador y habrá un controlador para las solicitudes de navegación general de la aplicación.

El constructor de cada controlador llama a los middleware o filtros establecidos para la aplicación, si pasa, se ejecuta el método solicitado. Un método puede o no llamar a un request para la validación de formularios, si pasa, se ejecuta el método dentro del método, que podrá conectarse a la base de datos y así enviar la vista solicitada.
3. Capa middleware: o capa de filtro puede o no ser ejecutada, dependerá del controlador y/o de los métodos a los que aplique dentro de este. Para este desarrollo habrán dos filtros:

- Middleware autenticación: que determina si se ha iniciado sesión por un usuario registrado en el sistema, si no es así, no se permite el acceso a la solicitud realizada y se envía al usuario a la vista de inicio de sesión.
- Middleware administración: determina si el usuario logeado es un usuario de tipo administrador. Si no es un usuario administrador no se permite el acceso a la solicitud realizada y se envía al usuario al panel principal de administración.

Si la solicitud pasa los filtros se llama al método solicitado dentro del controlador. Los middlewares están ubicados en el directorio `app/Http/Middleware`.

4. Capa request: o validación de formularios. Ubicados en el directorio `app/Http/Requests`. Si el método dentro del controlador recibe como parámetro un request se procede a validar los datos que lleva el request. Se puede hacer la validación que se requiera para cada tipo de dato entrante y se genera un array con mensajes de error en caso de que aplique para algún dato. Después de evaluar todas las entradas, si se generan errores se retorna al formulario que envió los datos. Si los datos pasan la validación se retorna al método que hizo el llamado en el controlador y se ejecuta.

5. Capa modelo: Los modelos de datos estarán ubicados en el directorio `app`. Los modelos serán las clases que representan a las tablas de la base de datos. El framework laravel contiene un ORM (object relational model) que administra estas tablas por medio del modelo. El controlador llamará a los modelos y con el resultado puede llamar a las vistas. Los modelos y sus métodos pueden ser llamados desde cualquier lugar de la aplicación.

6. Capa vista: las vistas están contenidas en el directorio `resources/views`. Son las diferentes interfaces con las que interactúa el usuario del sistema.
Módulos

Esta aplicación contará con tres módulos:

- Módulo cliente: Acceso a las opciones de radicación y consulta de solicitudes.
- Módulo asistente: Acceso a las opciones de solicitudes registradas en el sistema, listados, consultas, cambios de estados y generación de archivos.
- Módulo administrador: Acceso al módulo asistente y la administración de usuarios.

Requerimientos funcionales

Usuarios del sistema

Tabla 5. Usuarios del sistema

<table>
<thead>
<tr>
<th>Nombre</th>
<th>Tipo</th>
<th>Descripción</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administrador</td>
<td>admin</td>
<td>Administrador del sistema puede acceder a todo el sistema, la creación de usuarios y control de solicitudes</td>
</tr>
<tr>
<td>Asistente</td>
<td>asistente</td>
<td>Administra su cuenta de usuario y control de solicitudes.</td>
</tr>
<tr>
<td>Cliente</td>
<td>n/a</td>
<td>Persona natural o jurídica que radica una solicitud. No tendrá una cuenta de usuario en el sistema.</td>
</tr>
</tbody>
</table>

Definición de requerimientos

Tabla 6. Requerimiento radicar solicitud

<table>
<thead>
<tr>
<th>Historia de Usuario</th>
<th>CODIGO</th>
<th>US-01</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enunciado de la historia</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ROL</th>
<th>Funcionalidad</th>
<th>Resultado</th>
</tr>
</thead>
<tbody>
<tr>
<td>Como cliente</td>
<td>necesito radicar una solicitud PQR</td>
<td>con la finalidad de que se genere el código y la clave que hace referencia a mi solicitud</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Prioridad en negocio</th>
<th>Riesgo en desarrollo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alta</td>
<td>Baja</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Puntos estimados</th>
<th>Iteración asignada</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>2</td>
</tr>
</tbody>
</table>

DESCRIPCION

Se genera el formulario para la radicación de una solicitud, se valida los datos introducidos y se le da al cliente el código de solicitud y un código de verificación para que haga seguimiento a la misma. Los campos del formulario son: número documento, nombre, teléfono, email, tipo de solicitud, descripción de la solicitud, archivo adjunto, captcha. Todos son obligatorios, excepto el archivo adjunto.
### CRITERIOS DE ACEPTACION

<table>
<thead>
<tr>
<th>No.</th>
<th>Título</th>
<th>Contexto</th>
<th>Evento</th>
<th>Comportamiento esperado</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Campos obligatorios</td>
<td>En caso que el cliente no digita los campos identificación, nombre, email, teléfono y/o código de seguridad</td>
<td>Cuando pulsa el botón &quot;Registrar solicitud&quot;</td>
<td>Se muestra símbolo de aviso y se indican los campos requeridos.</td>
</tr>
<tr>
<td>2</td>
<td>Código de seguridad</td>
<td>En caso que el código que digita el cliente no coincida con el código de la imagen</td>
<td>Cuando pulsa el botón &quot;Registrar solicitud&quot;</td>
<td>Se muestra mensaje &quot;El código de seguridad no coincide&quot; y no se permite la radicación de la solicitud.</td>
</tr>
<tr>
<td>2</td>
<td>Solicitud radicada con éxito</td>
<td>N/A</td>
<td>Cuando pulsa el botón &quot;Registrar solicitud&quot;</td>
<td>El sistema muestra al usuario el código de solicitud y la clave de verificación para realizarle un seguimiento.</td>
</tr>
</tbody>
</table>

### Tabla 7. Requerimiento imprimir solicitud

<table>
<thead>
<tr>
<th>Historia de Usuario</th>
<th>CODIGO</th>
<th>US-02</th>
<th>Enunciado de la historia</th>
<th>Prioridad en negocio</th>
<th>Riesgo en desarrollo</th>
<th>Iteración asignada</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rol</td>
<td>Funcionalidad</td>
<td>Resultado</td>
<td>Prioridad en negocio</td>
<td>Riesgo en desarrollo</td>
<td>Iteración asignada</td>
<td></td>
</tr>
<tr>
<td>Como cliente</td>
<td>puedo imprimir la solicitud radicada</td>
<td>con la finalidad de tener en físico la información de la solicitud radicada</td>
<td>Baja</td>
<td>Baja</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

**DESCRIPCION**

Cuando se radica una solicitud se le lleva al cliente a una página con el código y clave de la solicitud radicada. En esta página se da la opción de imprimir la solicitud en donde se mostrará: información básica de la empresa de servicios, código de radicado, clave de radicado, fecha, persona quien radica, celular, email y tipo de solicitud.

<table>
<thead>
<tr>
<th>No.</th>
<th>Título</th>
<th>Contexto</th>
<th>Evento</th>
<th>Comportamiento esperado</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Comprobante solicitud radicada</td>
<td>N/A</td>
<td>Cuando pulsa el botón &quot;Imprimir&quot;</td>
<td>Se muestra en el navegador un archivo pdf con la información de la solicitud</td>
</tr>
</tbody>
</table>
Tabla 8. Requerimiento guardar información solicitud

<table>
<thead>
<tr>
<th>Historia de Usuario</th>
<th>CODIGO</th>
<th>US-03</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enunciado de la historia</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ROL</td>
<td>Funcionalidad</td>
<td>Resultado</td>
</tr>
<tr>
<td>Como cliente</td>
<td>puedo guardar la solicitud radicada</td>
<td>con la finalidad de tener en un archivo digital la información de la solicitud radicada</td>
</tr>
<tr>
<td>Prioridad en negocio</td>
<td>Riesgo en desarrollo</td>
<td>Baja</td>
</tr>
<tr>
<td>Puntos estimados</td>
<td>Iteración asignada</td>
<td>1</td>
</tr>
</tbody>
</table>

DESCRIPCION

Cuando se radica una solicitud se lleva al cliente a una página con el código y clave de la solicitud radicada. En esta página se da la opción de guardar la solicitud en donde se mostrará: información básica de la empresa de servicios, código de radicado, clave de radicado, fecha, persona quien radica, celular, email y tipo de solicitud.

CRITERIOS DE ACEPTACION

<table>
<thead>
<tr>
<th>No.</th>
<th>Título</th>
<th>Contexto</th>
<th>Evento</th>
<th>Comportamiento esperado</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Comprobante solicitud radicada</td>
<td>N/A</td>
<td>Cuando pulsa el botón &quot;Guardar&quot;</td>
<td>Se genera un archivo pdf con la información de la solicitud y se pide al usuario el destino de almacenamiento. El nombre del archivo será el código de radicado.</td>
</tr>
</tbody>
</table>

Tabla 9. Requerimiento consultar estado de la solicitud

<table>
<thead>
<tr>
<th>Historia de Usuario</th>
<th>CODIGO</th>
<th>US-04</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enunciado de la historia</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ROL</td>
<td>Funcionalidad</td>
<td>Resultado</td>
</tr>
<tr>
<td>Como cliente</td>
<td>necesito consultar el estado de una solicitud radicada</td>
<td>con la finalidad de hacerle un seguimiento</td>
</tr>
<tr>
<td>Prioridad en negocio</td>
<td>Riesgo en desarrollo</td>
<td>Media</td>
</tr>
<tr>
<td>Puntos estimados</td>
<td>Iteración asignada</td>
<td>3</td>
</tr>
</tbody>
</table>

DESCRIPCION

Se muestra el formulario de consulta de estado de una solicitud en donde se pide el número de esta y el código de verificación para mostrarle al usuario la información de su solicitud.

CRITERIOS DE ACEPTACION

<table>
<thead>
<tr>
<th>No.</th>
<th>Título</th>
<th>Contexto</th>
<th>Evento</th>
<th>Comportamiento esperado</th>
</tr>
</thead>
</table>
35

<table>
<thead>
<tr>
<th></th>
<th>Campos obligatorios</th>
<th>En caso que el cliente no digite alguno de los campos</th>
<th>Cuando pulsa el botón &quot;Consultar&quot;</th>
<th>Se muestra símbolo de aviso y se indican los campos requeridos.</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Campos inválidos</td>
<td>En caso que el código de solicitud y/o verificación sean inválidos</td>
<td>Cuando pulsa el botón &quot;Consultar&quot;</td>
<td>Se muestra mensaje &quot;El número de solicitud o la clave no son válidos!&quot; y se dá la opción de volver al formulario de radicación de solicitud</td>
</tr>
<tr>
<td>3</td>
<td>Solicitud encontrada</td>
<td>N/A</td>
<td>Cuando pulsa el botón &quot;Consultar&quot;</td>
<td>Se muestran los datos y estado de la solicitud</td>
</tr>
</tbody>
</table>

**Tabla 10. Requerimiento inicio de sesión**

<table>
<thead>
<tr>
<th>Historia de Usuario</th>
<th>CODIGO</th>
<th>US-05</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Enunciado de la historia</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>ROL</th>
<th>Funcionalidad</th>
<th>Resultado</th>
</tr>
</thead>
<tbody>
<tr>
<td>Como administrador o asistente</td>
<td>necesito iniciar sesión</td>
<td>para ingresar al panel de administración del sistema</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Prioridad en negocio</th>
<th>Riesgo en desarrollo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Media</td>
<td>Media</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Puntos estimados</th>
<th>Iteración asignada</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

**DESCRIPTICION**

Se muestra el formulario de inicio de sesión solicitando usuario y contraseña

**CRITERIOS DE ACEPTACION**

<table>
<thead>
<tr>
<th>No.</th>
<th>Título</th>
<th>Contexto</th>
<th>Evento</th>
<th>Comportamiento esperado</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Campos obligatorios</td>
<td>En caso que el administrador no digite alguno de los campos</td>
<td>Cuando pulsa el botón &quot;Acceso&quot;</td>
<td>Se muestra símbolo de aviso y se indican los campos requeridos.</td>
</tr>
<tr>
<td>2</td>
<td>Campos inválidos</td>
<td>En caso que el nombre de usuario y/o contraseña sean inválidos</td>
<td>Cuando pulsa el botón &quot;Acceso&quot;</td>
<td>Se muestra mensaje &quot;Para poder hacer uso de las funciones de administracion debe ingresar credenciales de acceso:&quot; y se redirige al usuario al formulario de inicio de sesión</td>
</tr>
<tr>
<td>3</td>
<td>Usuario registrado</td>
<td>N/A</td>
<td>Cuando pulsa el botón &quot;Acceso&quot;</td>
<td>Se redirige al administrador al panel de administración</td>
</tr>
</tbody>
</table>
Tabla 11. Requerimiento cambio de contraseña

<table>
<thead>
<tr>
<th>Historia de Usuario</th>
<th>CODIGO</th>
<th>US-06</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enunciado de la historia</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ROL</th>
<th>Funcionalidad</th>
<th>Resultado</th>
</tr>
</thead>
<tbody>
<tr>
<td>Como administrador o asistente</td>
<td>necesito cambiar mi contraseña</td>
<td>para asegurar mi cuenta de usuario</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Prioridad en negocio</th>
<th>Riesgo en desarrollo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baja</td>
<td>Baja</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Puntos estimados</th>
<th>Iteración asignada</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

**DESCRIPCIÓN**
Se muestra el formulario de nueva contraseña solicitando contraseña actual, nueva contraseña y confirmación de nueva contraseña.

**CRITERIOS DE ACEPTACIÓN**

<table>
<thead>
<tr>
<th>No.</th>
<th>Título</th>
<th>Contexto</th>
<th>Evento</th>
<th>Comportamiento esperado</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Campos obligatorios</td>
<td>En caso que el administrador o asistente no digite alguno de los campos</td>
<td>Cuando pulsa el botón &quot;Actualizar&quot;</td>
<td>Se muestra símbolo de aviso y se indican los campos requeridos.</td>
</tr>
<tr>
<td>2</td>
<td>Campos inválidos</td>
<td>En caso que la contraseña actual sea inválida</td>
<td>Cuando pulsa el botón &quot;Actualizar&quot;</td>
<td>Se redirige al usuario al formulario de cambio de contraseña y se muestra mensaje &quot;¡El password antiguo está errado!&quot;</td>
</tr>
<tr>
<td>3</td>
<td>Campos inválidos</td>
<td>En caso que la contraseña nueva sea muy corta</td>
<td>Cuando pulsa el botón &quot;Actualizar&quot;</td>
<td>Se redirige al usuario al formulario de cambio de contraseña y Se muestra mensaje &quot;¡El password nuevo es demasiado corto!&quot;</td>
</tr>
<tr>
<td>4</td>
<td>Campos inválidos</td>
<td>En caso que la contraseña nueva y la contraseña de confirmación no sean iguales</td>
<td>Cuando pulsa el botón &quot;Actualizar&quot;</td>
<td>Se redirige al usuario al formulario de cambio de contraseña y Se muestra mensaje &quot;¡Los passwords no coinciden!&quot;</td>
</tr>
<tr>
<td>5</td>
<td>Contraseña actualizada</td>
<td>N/A</td>
<td>Cuando pulsa el botón &quot;Actualizar&quot;</td>
<td>Se redirige al usuario al formulario de cambio de contraseña y Se muestra mensaje &quot;¡El password se ha cambiado correctamente!&quot;</td>
</tr>
</tbody>
</table>
### Tabla 12. Requerimiento crear usuario

<table>
<thead>
<tr>
<th>Historia de Usuario</th>
<th>CODIGO</th>
<th>US-07</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enunciado de la historia</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ROL</strong></td>
<td><strong>Funcionalidad</strong></td>
<td><strong>Resultado</strong></td>
</tr>
<tr>
<td>Como administrador</td>
<td>necesito crear un usuario</td>
<td>para que tenga acceso al sistema</td>
</tr>
<tr>
<td><strong>Prioridad en negocio</strong></td>
<td><strong>Riesgo en desarrollo</strong></td>
<td></td>
</tr>
<tr>
<td>Media</td>
<td>Media</td>
<td></td>
</tr>
<tr>
<td><strong>Puntos estimados</strong></td>
<td><strong>Iteración asignada</strong></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

**DESCRIPCION**
Se muestra el formulario de nuevo usuario solicitando nombre, email, contraseña, confirmación de contraseña y tipo de usuario.

#### CRITERIOS DE ACEPTACION

<table>
<thead>
<tr>
<th>No.</th>
<th>Título</th>
<th>Contexto</th>
<th>Evento</th>
<th>Comportamiento esperado</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Campos obligatorios</td>
<td>En caso que el administrador no digite alguno de los campos</td>
<td>Cuando pulsa el botón &quot;Crear&quot;</td>
<td>Se muestra símbolo de aviso y se indican los campos requeridos.</td>
</tr>
<tr>
<td>2</td>
<td>Campos inválidos</td>
<td>En caso que la contraseña actual sea muy corta</td>
<td>Cuando pulsa el botón &quot;Crear&quot;</td>
<td>Se redirige al administrador al formulario de nuevo usuario y se muestra mensaje &quot;¡La contraseña es muy corta, al menos 8 caracteres!&quot;</td>
</tr>
<tr>
<td>3</td>
<td>Campos inválidos</td>
<td>En caso que la contraseña y la contraseña de confirmación no sean iguales</td>
<td>Cuando pulsa el botón &quot;Crear&quot;</td>
<td>Se redirige al usuario al formulario de nuevo usuario y Se muestra mensaje &quot;¡Las contraseñas no coinciden!&quot;</td>
</tr>
<tr>
<td>4</td>
<td>Usuario creado</td>
<td>N/A</td>
<td>Cuando pulsa el botón &quot;Crear&quot;</td>
<td>Se redirige al usuario al formulario de nuevo usuario y Se muestra mensaje &quot;¡El usuario ha sido creado exitosamente!&quot;</td>
</tr>
</tbody>
</table>

### Tabla 13. Requerimiento listar usuarios

<table>
<thead>
<tr>
<th>Historia de Usuario</th>
<th>CODIGO</th>
<th>US-08</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enunciado de la historia</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ROL</strong></td>
<td><strong>Funcionalidad</strong></td>
<td><strong>Resultado</strong></td>
</tr>
<tr>
<td>Como administrador</td>
<td>necesito una lista de los usuarios del sistema</td>
<td>para tener acceso a cada uno de ellos</td>
</tr>
<tr>
<td><strong>Prioridad en negocio</strong></td>
<td><strong>Riesgo en desarrollo</strong></td>
<td></td>
</tr>
<tr>
<td>Media</td>
<td>Media</td>
<td></td>
</tr>
<tr>
<td><strong>Puntos estimados</strong></td>
<td><strong>Iteración asignada</strong></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

**DESCRIPCION**
Se muestra un listado de los usuarios del sistema en forma de tabla.

#### CRITERIOS DE ACEPTACION

<table>
<thead>
<tr>
<th>No.</th>
<th>Título</th>
<th>Contexto</th>
<th>Evento</th>
<th>Comportamiento esperado</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Lista usuarios</td>
<td>N/A</td>
<td>Cuando accede al módulo de administración - Usuario - usuarios</td>
<td>Se muestra el listado de usuarios registrados en el sistema</td>
</tr>
</tbody>
</table>
Tabla 14. Requerimiento editar usuario

<table>
<thead>
<tr>
<th>Historia de Usuario</th>
<th>CODIGO</th>
<th>US-09</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enunciado de la historia</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ROL</strong></td>
<td>Funcionalidad</td>
<td>Resultado</td>
</tr>
<tr>
<td>Como administrador</td>
<td>necesito editar un usuario</td>
<td>para que su información pueda ser actualizada</td>
</tr>
<tr>
<td><strong>Prioridad en negocio</strong></td>
<td><strong>Riesgo en desarrollo</strong></td>
<td></td>
</tr>
<tr>
<td>Media</td>
<td>Media</td>
<td></td>
</tr>
<tr>
<td><strong>Puntos estimados</strong></td>
<td><strong>Iteración asignada</strong></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

**DESCRIPCION**
Se muestra el formulario de edición de usuario solicitando nombre, email, contraseña, confirmación de contraseña y tipo de usuario

<table>
<thead>
<tr>
<th>CRITERIOS DE ACEPTACION</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>No.</strong></td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
</tbody>
</table>

Tabla 15. Requerimiento eliminar usuario

<table>
<thead>
<tr>
<th>Historia de Usuario</th>
<th>CODIGO</th>
<th>US-10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enunciado de la historia</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ROL</strong></td>
<td>Funcionalidad</td>
<td>Resultado</td>
</tr>
<tr>
<td>Como administrador</td>
<td>necesito eliminar un usuario</td>
<td>para que no pueda acceder al sistema</td>
</tr>
<tr>
<td><strong>Prioridad en negocio</strong></td>
<td><strong>Riesgo en desarrollo</strong></td>
<td></td>
</tr>
<tr>
<td>Media</td>
<td>Media</td>
<td></td>
</tr>
<tr>
<td><strong>Puntos estimados</strong></td>
<td><strong>Iteración asignada</strong></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

**DESCRIPCION**
Se muestra el listado de usuarios y la opción para eliminar usuario

<table>
<thead>
<tr>
<th>CRITERIOS DE ACEPTACION</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>No.</strong></td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
</tbody>
</table>

**Tabla 16. Requerimiento filtrar solicitudes**

<table>
<thead>
<tr>
<th>Historia de Usuario</th>
<th>Enunciado de la historia</th>
<th>CODIGO</th>
<th>US-11</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROL</td>
<td>Funcionalidad</td>
<td>Resultado</td>
<td></td>
</tr>
</tbody>
</table>

- **Como asistente**
  - necesito filtrar las solicitudes
  - Con la finalidad de obtener el listado de las solicitudes que cumplen ciertos requisitos

<table>
<thead>
<tr>
<th>Prioridad en negocio</th>
<th>Riesgo en desarrollo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alta</td>
<td>Alta</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Puntos estimados</th>
<th>Iteración asignada</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>3</td>
</tr>
</tbody>
</table>

**DESCRIPCION**

Se muestra el formulario de búsqueda con las opciones de búsqueda por: código, rango de fecha, documento de identificación, tipo de solicitud y/o estado.

**CRITERIOS DE ACEPTACION**

<table>
<thead>
<tr>
<th>No.</th>
<th>Título</th>
<th>Contexto</th>
<th>Evento</th>
<th>Comportamiento esperado</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Búsqueda con un solo resultado</td>
<td>En caso que solo una solicitud coincida con los criterios de búsqueda</td>
<td>Cuando se despliegue la solicitud</td>
<td>Se muestra en forma de tabla la información principal de la solicitud</td>
</tr>
<tr>
<td>2</td>
<td>Búsqueda sin ningún resultado</td>
<td>En caso que ninguna solicitud coincida con los criterios de búsqueda</td>
<td>Cuando se despliegue la solicitud</td>
<td>Se muestra el mensaje “Ninguna solicitud coincide con los criterios de búsqueda”</td>
</tr>
<tr>
<td>3</td>
<td>Ordenamiento de las solicitudes</td>
<td>En caso que el asistente no elija ningún criterio de búsqueda</td>
<td>Cuando se despliegue el listado de solicitudes</td>
<td>Se muestran todas las solicitudes registradas desde la más reciente a la primera creada.</td>
</tr>
<tr>
<td>4</td>
<td>Ordenamiento de las solicitudes</td>
<td>En caso que el asistente elija uno o más criterios de búsqueda</td>
<td>Cuando se despliegue el listado de solicitudes</td>
<td>Se muestran los solicitudes que coinciden con los criterios de búsqueda desde la más reciente a la primera creada.</td>
</tr>
</tbody>
</table>
**Tabla 17. Requerimiento guardar listado solicitudes**

<table>
<thead>
<tr>
<th>Historia de Usuario</th>
<th>CODIGO</th>
<th>US-12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enunciado de la historia</td>
<td>Funcionalidad</td>
<td>Resultado</td>
</tr>
<tr>
<td>ROL</td>
<td>Como asistente</td>
<td>puedo guardar los listados de solicitudes</td>
</tr>
<tr>
<td>Prioridad en negocio</td>
<td>Riesgo en desarrollo</td>
<td></td>
</tr>
<tr>
<td>Media</td>
<td>Media</td>
<td></td>
</tr>
<tr>
<td>Puntos estimados</td>
<td>Iteración asignada</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

**DESCRIPCIÓN**

Al generar un listado según la historia de usuario US-11 se da la opción de generar el archivo en formato hoja de cálculo.

**CRITERIOS DE ACEPTACIÓN**

<table>
<thead>
<tr>
<th>No.</th>
<th>Título</th>
<th>Contexto</th>
<th>Evento</th>
<th>Comportamiento esperado</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Archivo en formato hoja de cálculo</td>
<td>N/A</td>
<td>Cuando pulsa el botón “Guardar”</td>
<td>Se genera un archivo excel con el listado de solicitudes y se pide al usuario el destino de almacenamiento.</td>
</tr>
</tbody>
</table>

**Tabla 18. Requerimiento ver solicitud**

<table>
<thead>
<tr>
<th>Historia de Usuario</th>
<th>CODIGO</th>
<th>US-13</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enunciado de la historia</td>
<td>Funcionalidad</td>
<td>Resultado</td>
</tr>
<tr>
<td>ROL</td>
<td>Como asistente</td>
<td>necesito seleccionar una solicitud</td>
</tr>
<tr>
<td>Prioridad en negocio</td>
<td>Riesgo en desarrollo</td>
<td></td>
</tr>
<tr>
<td>Media</td>
<td>Media</td>
<td></td>
</tr>
<tr>
<td>Puntos estimados</td>
<td>Iteración asignada</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

**DESCRIPCIÓN**

Al generar un listado según la historia de usuario US-11 se da la opción de ver la información de cada solicitud.

**CRITERIOS DE ACEPTACIÓN**

<table>
<thead>
<tr>
<th>No.</th>
<th>Título</th>
<th>Contexto</th>
<th>Evento</th>
<th>Comportamiento esperado</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Solicitud seleccionada</td>
<td>N/A</td>
<td>Cuando pulsa el código de una solicitud</td>
<td>Se muestran los datos y estado de la solicitud</td>
</tr>
</tbody>
</table>
Tabla 19. Requerimiento pasar siguiente estado una solicitud

<table>
<thead>
<tr>
<th>Historia de Usuario</th>
<th>CODIGO</th>
<th>US-14</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enunciado de la historia</td>
<td>Funcionalidad</td>
<td>Resultado</td>
</tr>
<tr>
<td>Como asistente</td>
<td>necesito pasar al siguiente estado una solicitud</td>
<td>con la finalidad de actualizar el proceso de la solicitud</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ROL</th>
<th>Prioridad en negocio</th>
<th>Riesgo en desarrollo</th>
<th>Puntos estimados</th>
<th>Iteración asignada</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alta</td>
<td>Alta</td>
<td></td>
<td>5</td>
<td>3</td>
</tr>
</tbody>
</table>

DESCRIPTOR

En el listado de solicitudes se muestra la información principal de la solicitud y se hace clic sobre el botón cambio de estado

<table>
<thead>
<tr>
<th>CRITERIOS DE ACEPTACION</th>
</tr>
</thead>
<tbody>
<tr>
<td>No.</td>
</tr>
<tr>
<td>-----</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
</tbody>
</table>

Modelo de datos

Para este proyecto el nuevo diseño de la base de datos tiene algunos cambios que son importantes (Figura 17):

- Se elimina la tabla sesiones, no es necesaria ya que el framework administra y controla las sesiones y se agregan las tablas migrations y password_resets que utiliza el framework para el control de migraciones y el manejo de contraseñas respectivamente.

- en la tabla usuarios se agrega el campo tipo para definir los roles de usuarios. A saber en la aplicación se manejarán dos tipos de usuarios. El usuario administrador quien administra la aplicación y el usuario asistente quien administrará las solicitudes radicadas.

- Se crea la tabla historias en la cual se llevará el registro de cada uno de los estados por los que pase una solicitud. Con esta tabla logramos que la aplicación sea
dinámica ya que estaba limitada por la tabla solicitudes a tener solo tres estados que se registraban en esta tabla sin la posibilidad de agregar archivos adjuntos y describir la situación de la solicitud al cambiar de estado.

- Se agrega a la tabla **estados** la columna **orden** para definir la jerarquía de los estados y así dinámicamente la aplicación controlar el proceso de respuesta a una solicitud PQR.
- Se implementan relaciones foráneas entre tablas para que el gestor de base de datos (en este caso MySql) controle la integridad referencial de la información almacenada.

*Figura 17. Nuevo modelo de datos*
Modelo entidad relación

De este modelo se genera un nuevo modelo entidad relación (figura 18).

Figura 18. Nuevo modelo entidad relación

Diccionario de datos

Cambiamos el modelo de datos y aquí definimos los nuevos atributos de la base de datos (tabla 20).

Tabla 20. Nuevo diccionario de datos

<table>
<thead>
<tr>
<th>Tabla</th>
<th>Llave</th>
<th>Columna</th>
<th>Tipo</th>
<th>Nulo</th>
<th>Descripción</th>
</tr>
</thead>
<tbody>
<tr>
<td>estados</td>
<td>PK</td>
<td>estado</td>
<td>int(11)</td>
<td>No</td>
<td>Código único de un estado</td>
</tr>
<tr>
<td></td>
<td></td>
<td>descripcion</td>
<td>varchar(80)</td>
<td>No</td>
<td>Nombre del estado</td>
</tr>
<tr>
<td></td>
<td></td>
<td>estado</td>
<td>varchar(8)</td>
<td>No</td>
<td>Determina si está activo</td>
</tr>
<tr>
<td>solicitudes</td>
<td>PK</td>
<td>codigo</td>
<td>int(11)</td>
<td>No</td>
<td>Código único radicado PQR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>clave</td>
<td>varchar(30)</td>
<td>No</td>
<td>Clave única de acceso a la solicitud</td>
</tr>
<tr>
<td>Tabla</td>
<td>Columna</td>
<td>Tipo</td>
<td>Largo</td>
<td>Descripción</td>
<td></td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------</td>
<td>------------</td>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>identificacion</td>
<td>identificacion</td>
<td>varchar(20)</td>
<td>No</td>
<td>Número documento identificación persona quien radica la solicitud</td>
<td></td>
</tr>
<tr>
<td>nombre</td>
<td>nombre</td>
<td>varchar(200)</td>
<td>No</td>
<td>Nombre persona quien radica la solicitud</td>
<td></td>
</tr>
<tr>
<td>telefono</td>
<td>telefono</td>
<td>varchar(30)</td>
<td>No</td>
<td>Teléfono persona quien radica</td>
<td></td>
</tr>
<tr>
<td>correo</td>
<td>correo</td>
<td>varchar(200)</td>
<td>No</td>
<td>Dirección de correo electrónico persona quien radica la solicitud</td>
<td></td>
</tr>
<tr>
<td>FK tipo_solicitud</td>
<td>tipo_solicitud</td>
<td>int(11)</td>
<td>No</td>
<td>Tipo de solicitud</td>
<td></td>
</tr>
<tr>
<td>FK estado</td>
<td>estado</td>
<td>int(11)</td>
<td>No</td>
<td>Estado actual de la solicitud</td>
<td></td>
</tr>
<tr>
<td>created_at</td>
<td>created_at</td>
<td>timestamp</td>
<td>No</td>
<td>Fecha creación del registro</td>
<td></td>
</tr>
<tr>
<td>update_at</td>
<td>update_at</td>
<td>timestamp</td>
<td>No</td>
<td>Fecha última actualización del registro</td>
<td></td>
</tr>
<tr>
<td>tipos_solicitud</td>
<td>tipo</td>
<td>int(11)</td>
<td>No</td>
<td>Código único de un tipo de solicitud</td>
<td></td>
</tr>
<tr>
<td></td>
<td>descripcion</td>
<td>varchar(80)</td>
<td>No</td>
<td>Nombre del tipo de solicitud</td>
<td></td>
</tr>
<tr>
<td></td>
<td>estado</td>
<td>varchar(8)</td>
<td>No</td>
<td>Determina si está activo</td>
<td></td>
</tr>
<tr>
<td>users</td>
<td>id</td>
<td>int(11)</td>
<td>No</td>
<td>Código de identificación de un usuario</td>
<td></td>
</tr>
<tr>
<td></td>
<td>nombre</td>
<td>char(140)</td>
<td>No</td>
<td>Nombre del tipo de usuario</td>
<td></td>
</tr>
<tr>
<td></td>
<td>email</td>
<td>varchar(255)</td>
<td>No</td>
<td>Email de usuario</td>
<td></td>
</tr>
<tr>
<td></td>
<td>password</td>
<td>char(50)</td>
<td>No</td>
<td>Contraseña de un usuario</td>
<td></td>
</tr>
<tr>
<td></td>
<td>tipo</td>
<td>varchar(30)</td>
<td>No</td>
<td>Tipo de usuario</td>
<td></td>
</tr>
<tr>
<td></td>
<td>remember_token</td>
<td>varchar(100)</td>
<td>No</td>
<td>Token</td>
<td></td>
</tr>
<tr>
<td></td>
<td>created_at</td>
<td>timestamp</td>
<td>No</td>
<td>Fecha creación del registro</td>
<td></td>
</tr>
<tr>
<td></td>
<td>update_at</td>
<td>timestamp</td>
<td>No</td>
<td>Fecha última actualización del registro</td>
<td></td>
</tr>
<tr>
<td>historias</td>
<td>id</td>
<td>int(11)</td>
<td>No</td>
<td>Código único de una historia</td>
<td></td>
</tr>
<tr>
<td>FK solicitud</td>
<td>solicitud</td>
<td>int(11)</td>
<td>No</td>
<td>Solicitud referenciada</td>
<td></td>
</tr>
<tr>
<td>FK estado</td>
<td>estado</td>
<td>int(11)</td>
<td>No</td>
<td>estado de la historia</td>
<td></td>
</tr>
<tr>
<td>fecha</td>
<td>fecha</td>
<td>date</td>
<td>No</td>
<td>fecha de registro de la historia</td>
<td></td>
</tr>
<tr>
<td>descripcion</td>
<td>descripcion</td>
<td>text</td>
<td>No</td>
<td>Descripción de la historia</td>
<td></td>
</tr>
<tr>
<td>archivo</td>
<td>archivo</td>
<td>varchar(200)</td>
<td>No</td>
<td>Archivo adjunto a la historia</td>
<td></td>
</tr>
</tbody>
</table>
Prototipos.

Para el desarrollo de este proyecto se utilizaron las prácticas de la metodología por prototipos, un modelo de desarrollo de software cíclico evolutivo. Este proyecto generó 3 prototipos, cada uno con sus correspondientes etapas y que se exponen a continuación:

Prototipo interfaz
Fase recolección y refinamiento de requisitos
En este prototipo se desarrolla la interfaz básica para la radicación y consulta de una solicitud y la interfaz para la administración del sistema.

Requerimientos

- Formulario para la radicación de una solicitud.
- Confirmación radicación de la solicitud.
- Impresión de una solicitud
- Formulario para que el cliente consulte una solicitud
- Datos de una solicitud radicada.
- Formulario inicio sesión
- Panel de administración
- Módulo usuarios
- Módulo solicitudes
Fase diseño rápido

Radicación solicitud

Figura 19. Borrador formulario radicación de una solicitud
Confirmación de radicación

Figura 20. Borrador pantalla confirmación de radicación

Impresión solicitud
Figura 21. Borrador documento en PDF que confirma la radicación de una solicitud

Consulta e información de una solicitud

Figura 22. Borrador formulario consulta y datos solicitud
Inicio de sesión

Figura 23. Borrador formulario inicio de sesión

Administración

Figura 24. Borrador panel de administración
Módulo usuarios

Figura 25. Borrador formulario creación de usuario
Figura 26. Borrador listado de usuarios

Módulo solicitudes

Figura 27. Borrador formulario para la búsqueda de solicitudes
Figura 28. Borrador listado de solicitudes

<table>
<thead>
<tr>
<th>#</th>
<th>Código</th>
<th>Código de Identificador</th>
<th>Nombre</th>
<th>Fecha</th>
<th>Solicitud</th>
<th>Estado</th>
<th>Operación</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>23541</td>
<td>11213689521</td>
<td>Rafael Nadal</td>
<td>10-08-2017</td>
<td>Petición</td>
<td>Atendido</td>
<td>20-08-2017</td>
</tr>
<tr>
<td>2</td>
<td>23236</td>
<td>64369874</td>
<td>Leonard Cobalit</td>
<td>12-08-2017</td>
<td>Queja</td>
<td>Atendida</td>
<td>22-08-2017</td>
</tr>
</tbody>
</table>
Figura 29. Borrador detalle solicitud
Figura 30. Formulario cambio de estado
Construcción del prototipo

Formulario para la radicación de una solicitud

![Imagen de formulario]

**Figura 31.** Formulario radicación de una solicitud

Confirmación de radicación

![Imagen de confirmación]

**Figura 32.** Página respuesta a la radicación
Impresión solicitud

![Impresión solicitud](image)

Figura 33. Documento PDF con la información de la solicitud
Consulta de una solicitud

**Figura 34.** Página en la que el cliente consulta una solicitud

Inicio de sesión

**Figura 35.** Página para inicio de sesión
Panel Administración

Figura 36. Panel de administración del sistema

Módulo usuarios

Figura 37. Página formulario creación de usuario
Figura 38. Lista de usuario

Módulo solicitudes

Figura 39. Búsqueda y lista de solicitudes
Figura 40. Información de una solicitud

Evaluación y refinamiento del prototipo

La interfaz es de apariencia sencilla pero gracias al framework bootstrap podrá visualizarse adecuadamente en los dispositivos de escritorio de uso más común.

Laravel por medio de su manejo de plantillas (Blade) permitió la reutilización de código HTML de forma eficaz, permitiendo un código más limpio y entendible.
Se observa que se muestra diferente información de una solicitud al cliente y al asistente, siendo que debería ser igual. La información para el asistente es más detallada, se decide entonces mostrar al cliente en detalle la información y estados de la solicitud (Figura 41).

**Figura 41.** Modificación interfaz consulta de una solicitud

En términos generales se cumplen los requerimientos del prototipo.

**Producto de ingeniería**

En el prototipo de interfaz, se implementa la navegación que deberá tener la aplicación. Para esto se enrutan las vistas implementando tres de las capas de la aplicación, capa enrutamiento, capa controlador y capa vista.
Para esto se crean los controladores AdminController.php, CaptchaController.php, PqrController.php, UsuarioController.php. Los cuáles son almacenados en el directorio **PQR/app/Http/Controllers**.

Estos controladores son llamados por el enrutador e inicialmente tendrán las funciones que hacen el llamado a las vistas para ser enviadas como **response** al navegador.

![Diagrama arquitectura prototipo interfaz.](image-url)
Prototipo funciones principales

Fase recolección y refinamiento de requisitos
En este prototipo se desarrolla la funcionalidad para que el cliente registre y consulte la solicitud radicada

Requerimientos
- Validación formulario registro de una solicitud.
- Registro de una solicitud.
- Impresión datos de una solicitud
- Búsqueda de una solicitud radicada
- Obtener datos de la solicitud

Fase diseño rápido
Validación formulario registro de una solicitud
Para este formulario son datos obligatorios: identificación, nombre, teléfono, email, tipo de solicitud, descripción y el código captcha.
El campo nombre, no debe llevar números.
El campo email, debe tener el patrón de una dirección de correo electrónico.
Para el campo archivo sólo se permiten archivos con extensión jpeg, png o pdf.
Estas validaciones se realizarán cuando los datos lleguen al servidor, serán evaluados en la capa request. Y devolverá al usuario al formulario con los datos enviados y el mensaje de los datos faltantes o permitirá el acceso a la función del controlador en caso que los datos sean los solicitados.
Registro de una solicitud

En la función de almacenamiento se toman los datos enviados en el request, después de la validación, se conecta con la base de datos y se almacena en la tabla solicitudes y se crea el primer estado de la solicitud en tabla historias. Se crea el código único de la solicitud y una clave única para la solicitud.

Impresión datos de la solicitud

Después de registrar una solicitud se muestra la pantalla de respuesta, con los datos generales de la solicitud y la opción de imprimirlos o guardarlos en un archivo pdf.

Búsqueda de una solicitud

El cliente puede consultar los datos y estado de una solicitud registrada por medio de su código único y clave dada en el momento del registro. Si los datos son correctos se muestra en pantalla los datos generales del solicitante, los datos de la solicitud y el historial de estados.

Construcción del prototipo

Inicialmente se crea la base de datos en el gestor de base de datos MySql. La definición de las tablas se hace haciendo uso del codefirst, es decir, por medio de código dentro de la aplicación creamos el esquema de las tablas y luego las migramos hacia la base de datos. Se crean las migraciones para las tablas tipo_solicitudes, estados, solicitudes, historias. Laravel en el momento de la creación del proyecto crea la tabla users, la cual es recomendable utilizar para el manejo de usuarios. Estas migraciones están almacenadas en el directorio PQR/database/migrations.

A continuación se crean los modelos que se conectarán a la base de datos por medio del ORM de laravel Eloquent. Los modelos son Estado.php, Historia.php, Solicitud.php, y TipoSolicitud.php. Almacenados en el directorio PQR/app.

Después se crea la función que carga los datos necesarios para el registro de una solicitud y las funciones que los almacenan en la base de datos. Estas en el PqrController.php
Validación formulario registro de una solicitud

Para la validación del envío de datos por formularios se utiliza la capa request en donde se crea un “validador” por cada formulario de la aplicación. Para mejor identificación de estos se nombran con el postifijo Request. Para este formulario se crea PqrRequest.php los cuáles están ubicados en el directorio PQR/app/Http/Request. Se definen las reglas de validación así:

```php
/**
 * Get the validation rules that apply to the request.
 *
 * @return array
 */

public function rules()
{
    return [
        "identificacion" => 'required',
        "nombre"         => 'required|string',
        "telefono"       => 'required',
        "correo"         => 'required|email',
        "tipo_solicitud" => 'required',
        "descripcion"    => 'required',
        "archivo"      => 'mimetypes:image/jpeg,image/png,application/pdf',
        "filtro"         => 'required|captcha'
    ];
}
```
La solicitud http de registro se evalúa según el siguiente diagrama:

**Figura 43.** Diagrama de validación formulario registro de solicitud.

**Figura 44.** Mensajes validación formulario registro solicitud.
Registro de una solicitud

Si la solicitud http pasa correctamente la validación de datos (capa request), se conecta a la base de datos por medio de los modelos para insertar los registros en las tablas solicitudes e historia, y se redirige al cliente a la ventana de respuesta registro solicitud exitosa. La ventana de respuesta se le informa código único de la solicitud y clave de acceso.

*Figura 45. Diagrama registro solicitud*
Impresión datos de solicitud registrada

Se muestra en pantalla mensaje de solicitud radicada y el código único de la solicitud y su clave correspondiente.

Como se puede ver en la figura 33, se le permite al cliente imprimir y/o guardar los datos de la solicitud. Para la impresión de datos se genera un archivo en formato pdf que puede ser visualizado en el navegador o que puede ser descargado al disco local del cliente. Para esta funcionalidad se utiliza un Service Provider desarrollo por terceros. La integración de services providers en laravel es muy sencilla, es una de las características que a título personal, más aprecié.

Figura 46. Diagrama respuesta solicitud radicada
Esto se hace descargando vía composer y especificando el paquete a descargar ó agregando el nombre y versión del paquete en el archivo composer.json y después actualizar las dependencias haciendo un composer update.

Se utiliza el Service Provider dompdf y como se puede ver en la figura 34, se genera el archivo pdf.

Búsqueda de una solicitud radicada
En esta pantalla (figura 34) se consulta la solicitud y en ella misma se muestra si el código y la clave no son válidos o por el contrario si son válidos se muestra toda la información de la solicitud. Para lograr esta forma de presentación se hace uso de Ajax para poder hacer cambios sobre la página sin necesidad de recargarla, mejorando la velocidad de navegación y la experiencia del cliente.

![Diagrama búsqueda solicitud](image)

Figura 47. Diagrama búsqueda solicitud
Evaluación y refinamiento del prototipo
Las características del framework han permitido la modularidad del código fuente, haciendo mucho más organizado, entendible e independiente. El ORM Eloquent es una herramienta potente, ya que ha permitido el acceso a la base de datos por medio de los modelos sin necesidad de insertar código SQL directamente en la aplicación. Su sintaxis y estructura es totalmente orientada a objetos logrando que se apliquen los últimos estándares de desarrollo de software.
Se crean las funcionalidades necesarias para que un cliente registre y consulte sus solicitudes PQR registradas, con una interfaz sencilla pero que esconde una reutilización de recursos considerable.
El diseño no cambia al propuesto en el análisis y el primer prototipo.
En términos generales se cumplen los requerimientos del prototipo.

Producto de ingeniería
En este prototipo se implementan cinco capas de la arquitectura propuesta (figura 48). Todas ellas ya definidas en el framework pero que cada programador o grupo de trabajo decide si utiliza o no.
Se crea la base de datos, se crean las tablas y sus modelos respectivos. Se enrutan todas las solicitudes y se crea un Controlador para los recursos pqr.
En este prototipo se implementa el uso de Service Provider de terceros y la creación de helpers que son funciones que podemos llamar desde cualquier parte del código.
Se implementan las llamadas asíncronas por medio de Ajax para mejorar el rendimiento de la aplicación y la experiencia de usabilidad.
Prototipo final

Fase recolección y refinamiento de requisitos
En este prototipo se desarrolla el modulo administrativo, en donde se podrán controlar usuarios y las solicitudes registradas en el sistema.

Requerimientos
- Inicio de sesión y control por tipo de usuario logeado.
- CRUD usuarios.
- Filtro de búsqueda para solicitudes
- Descarga listado solicitudes en archivo Excel
- Consultar datos solicitud
- Cambiar estado de una solicitud
Fase diseño rápido
Inicio de sesión y control por tipo de usuario logeado.
Al módulo administrativo podrán acceder dos tipos de **usuarios**, usuario tipo **asistente** y usuario tipo **admin**. El asistente podrá administrar las solicitudes y el admin podrá administrar solicitudes y podrá administrar usuarios.
La interfaz de este módulo administrativo es similar a la del cliente, pero contará con un menú lateral con las dos opciones (figura 36) a la cual se accede después de iniciar sesión (figura 35).
También existe la opción para que el usuario registrado cambie su contraseña.

**CRUD usuarios**
En este submódulo del módulo administrativo se podrá crear un nuevo usuario y listar los existentes en el sistema, editar y/o eliminarlos. (Figuras 37 y 38)

**Filtro de búsqueda para solicitudes**
Se tendrá la opción de filtrar la búsqueda de solicitudes por código, documento solicitante, rango de fecha, tipo de solicitud y/o estado de la solicitud (figura 39).

**Descarga listado solicitudes en archivo Excel**
Al generar el listado de solicitudes por medio del formulario de filtro, se crea la opción de guardar en el disco local del usuario, la lista generada en formato Excel.

**Consultar datos solicitud**
A partir de la lista generada se podrá acceder a toda la información de cada solicitud. La cuál es la misma información que se le retorna al cliente al buscar una solicitud.
Cambiar estado de una solicitud

El administrador o asistente podrá asignar el nuevo estado a la solicitud. Un estado no podrá ser eliminado. Al nuevo estado se le asignará una fecha, observaciones y la opción de subir un archivo.

Construcción del prototipo

Inicio de sesión y control por tipo de usuario logeado.

Para controlar el acceso al módulo de administración y tener en sesión los datos del usuario autenticado hacemos uso de los middleware o filtros que proporciona laravel. Para la autenticación el middleware Authenticate nos proporciona el acceso a los datos del usuario logeado y más importante permitirá el acceso a aquellas rutas o controladores para los que sea llamado, es decir, si se intenta acceder a un recurso que lleva este filtro se debe haber autenticado un usuario, en caso contrario redirige la petición a una respuesta Unauthorized.

Los middleware los encontramos en el directorio PQR/app/Http/Middleware.

Así mismo creamos un middleware para filtrar las peticiones Request por tipo de usuario. Así se le permite a cada tipo de usuario acceder a los recursos que le competen únicamente.

Con esto hacemos uso a la sexta capa definida en la arquitectura, la capa middleware o capa filtro que se puede encontrar al definir una ruta o en la definición de un controlador (figuras 49 y 50).
Figura 49. Diagrama inicio de sesión

Figura 50. Diagrama acceso por tipo de usuario
CRUD usuarios

Para este submódulo se crea un controlador de recursos para administrar este tipo de recurso, llamado `UserController.php`. A este sólo podrá acceder un usuario de tipo administrador. Las vistas (figuras 38 y 39), darán acceso a la creación de un usuario y/o listar usuarios, editarlos y eliminarlos.

Igualmente se crea `UsuarioRequest` para la validación de los datos enviados en la creación edición de un usuario. Todos los datos son obligatorios, aquí las reglas de validación:

```php
public function rules()
{
    return [
        "tipo"      => 'required',
        "nombre"    => 'required',
        "email"     => 'required|email|unique:users',
        "password"  => 'required|min:6|confirmed'
    ];
}
```
Figura 51. Diagrama crear usuario

Figura 52. Diagrama lista usuarios
Filtro de búsqueda para solicitudes

Para el submódulo de administración de solicitudes, este formulario de búsqueda será la forma de acceder a una o más solicitudes (figura 39). Para esta función se crea un request `FiltroRequest.php` que valida los datos de búsqueda, las reglas son:

```php
public function rules()
{
    return [
        'codigo' => 'numeric',
        'documento' => 'numeric',
        'fDesde' => 'date',
        'fHasta' => 'date|after_or_equal:fDesde'
    ];
}
```

Figura 53. Diagrama formulario búsqueda de solicitudes
Descarga listado solicitudes en archivo Excel
Cuando se genera la lista con el filtro de búsqueda se puede descargar la misma en formato Excel. Para crear esta funcionalidad se hace uso de un Service Provider desarrollado por terceros llamado maatwebsite/Excel y de la misma forma que dompdf lo incluimos en el proyecto. Con este service provider podemos generar un archivo tipo Excel, con los datos que necesitemos y configurar la forma de presentación de los mismos en la hoja de cálculo (figura 54). De igual manera, le permite al usuario descargar el archivo a su disco local.

Figura 54. Archivo Excel generado por la aplicación
Consultar datos de una solicitud

Al generar el listado de solicitudes por medio del filtro, se permite ver cada solicitud (figura 39), con todos los datos del solicitante, datos de la solicitud e historial de estados. Esta vista es la misma que se le retorna al cliente cuando hace una búsqueda y consulta de su solicitud (figura 40), pero dentro del módulo administrativo. Para esto se reutilizó código por medio de plantillas Blade.
Cambiar estado de una solicitud
Las solicitudes pasan por un estado inicial, unos estados intermedios y uno final. Inicialmente la aplicación maneja tres estados (**Radicado, revisado y atendido**), pero el diseño modularizado y dinámico de la aplicación permitirá agregar nuevos estados en la **tabla estados** (figura 18) sin afectar el funcionamiento de la aplicación.
Al generar el listado de solicitudes por medio del filtro, se permite asignar un nuevo estado a una solicitud (figura 39), si este enlace no aparece es porque la solicitud llegó a su estado final (atendido).
Para asignar un nuevo estado se debe verificar el actual, por medio del historial de la solicitud y haciendo uso de un formulario (figuras 57 y 58) se asigna el siguiente estado.
Figura 57. Primera ventana asignación nuevo estado

Figura 58. Segunda ventana (formulario) asignación nuevo estado

Se crea entonces el Request **EstadoRequest** con el que se valida los datos enviados para la asignación del nuevo estado.
Evaluación y refinamiento del prototipo
El framework laravel ha permitido la integración de funcionalidades desarrolladas por terceros de una forma sencilla, organizada y modularizada, dándole esto un gran potencial al desarrollo de aplicaciones tanto pequeñas como a gran escala.
Los middleware del framework o filtros por los que pasa una solicitud son una ventaja inmensa para el desarrollo ágil y seguro de las aplicaciones web, tanto los que vienen por defecto con el proyecto como los que se desarrollen para los requerimientos de cada aplicación.
El administrador de plantillas Blade ha permitido código limpio y claro en cada archivo que lleva el código html, gracias a la inserción de plantillas, permitiendo la reutilización de código y archivos menos extensos.

Se crean las funcionalidades necesarias para la administración de solicitudes y control de usuarios en la aplicación.

Se modifica funcionamiento del filtro de búsqueda de solicitudes y la lista resultante. Los datos ya no se envían por medio de Ajax y se muestran los resultados en la misma pantalla. Se considera de mejor manejo que la se muestre tanto el formulario como la lista en pantallas diferentes. (Figura 60 y 61).

Figura 60. Filtro de búsqueda modificado
Producto de ingeniería
En este prototipo se implementan todas las capas de la arquitectura propuesta (figura 17). Implementando otros Service Providers y otros Middlewares diferentes a los que vienen con el proyecto, para cumplir las especificaciones de la aplicación.
Se hace el control de la autenticación en el sistema y el control de acceso por tipos de usuarios.
Se desarrollan los submódulos de administración de usuarios y de administración de solicitudes.

Etapa 4. Desarrollo de la guía de reingeniería
Se documenta la metodología de reingeniería utilizada, basándose en el proyecto al cuál se le aplica este proceso.
Igualmente se hace un documento con las características principales del framework laravel y se incluye como ejercicio práctico el desarrollo del prototipo de las funciones principales.

Figura 61. Lista de solicitudes modificada
Etapa 5. Socialización de la guía y del desarrollo de la aplicación al interior de la empresa

Se hace la socialización del desarrollo, principalmente con los programadores de la empresa y otros cercanos a ellos. Se hace el ejercicio práctico que aparece en el documento para desarrollo de aplicaciones con el framework laravel.
Logros y lecciones aprendidas

Los logros alcanzados y el aprendizaje obtenido va desde el entendimiento de lo que debe ser una metodología de desarrollo de software, la cual es en realidad la ejecución de diferentes actividades, en un orden establecido para llegar al desarrollo de un proyecto de software, en donde por ejemplo una metodología de desarrollo como la utilizada para esta aplicación, la metodología por prototipos, es sólo una de las actividades que hacen parte de la metodología de desarrollo de software.

Pasando también por el aprendizaje y práctica de un proceso de reingeniería de software y llegando al aprendizaje de un framework de desarrollo tan potente como lo es laravel.

A continuación se enumeran los logros y lecciones principales que se alcanzaron con gran satisfacción:

- Aprendizaje y aplicación de lo que es una metodología de desarrollo de software o de un proyecto.
- Aprendizaje y ejecución de una metodología para la reingeniería de software.
- Aprendizaje y uso de herramientas de desarrollo web del lado del cliente como son: framework bootstrap, librería de javascript jquery y otros plugin del mismo lenguaje, técnica y herramienta de programación web ajax, html5 y css.
- Aprendizaje y aplicación del framework laravel para aplicaciones web orientadas a objetos y modularizadas.
- Aprendizaje y utilización del gestor de base de datos PostgreSQL.
- Diseño de una base de datos.
- Creación y uso de repositorios git para el control y backups de versiones de una aplicación.
- Manejo herramienta putty para conexión a servidores.
- Creación de documentos de enseñanza y práctica acerca del trabajo realizado.
Limitaciones

Las limitaciones las encontramos en la sencillez de la aplicación que a pesar de cumplir con los requerimientos estéticamente es muy simple y tal vez desprovista del gusto de un diseñador especializado en esta área.

Encontramos limitaciones en la creación de nuevos estados, ya que no hay un acceso a esto desde la aplicación. Para agregar un nuevo estado se debe hacer desde la tabla estados en la base de datos.

También en cuanto al diseño responsivo, el cual se verá reflejado en dispositivos para escritorio, pero no para dispositivos móviles.
Conclusiones

La reingeniería del software no es tarea fácil, se necesita de dedicación y de trabajo detallado para entender un software desarrollado mucho tiempo atrás y por otro grupo de personas.

Sin embargo, la actualización de software debe ser, a mi parecer, una actividad que se le debe mencionar a los clientes que soliciten un programa o software particular. Desde hace tiempo atrás y probablemente a futuro toda la información será almacenada y administrada por sistemas de cómputo, así que cuando una persona o empresa tome la decisión de tener software propio debe saber que son sistemas que deben ser actualizados regularmente.

Esto es algo que cada día es más sencillo de lograr dadas las nuevas técnicas y herramientas como la programación orientada a objetos y el framework laravel, orientadas a la modularización del código y por ende la facilidad de actualización del sistema o de una parte de este, sin afectar el funcionamiento principal, por el contrario a la creación de nuevas y modernas funcionalidades.
Recomendaciones

El análisis y entendimiento de un código que ha sido desarrollado por otro programador, con técnicas en desuso y diferentes a las que una persona maneja, y sobre todo basado en un lenguaje del que no se tiene mucha experiencia, puede ser un gran reto incluso para un programador con experiencia.

Pero todas estas dificultades se solven cuando se estudia y se comprende las necesidades de las personas que hacen uso de la aplicación y de quienes la utilizarán a futuro. La comunicación entre desarrolladores y usuarios finales de un software marca la diferencia en cuanto a productividad, por tiempo, calidad y cumplimiento de requerimientos del software.

De igual forma entender que la ciencia de la programación está en constante y creciente evolución, así que el conocimiento debe ser renovado continuamente, para aprovechar las nuevas herramientas que agilizan el desarrollo. La idea no es volver a crear lo que ya existe, si no, basarse en lo que está hecho y así tod@s ir un paso adelante, hacia la evolución.
Bibliografía


7) Palacios, D. Por qué Laravel NO es un framework MVC y tú deberías olvidarte de MVC. URL https://styde.net/porque-laravel-no-es-mvc-y-tu-deberias-olvidarte-de-mvc/

8) Palacios, R. Curso de laravel 5. URL https://www.youtube.com/watch?v=toqljDT95wo&list=PLIdmSRJrEJ0u-5Nv2k6W8Vhe0wUP_7H5W


10)Refness, D. The world’s largest web developer site. URL https://www.w3schools.com/default.asp
11) Atlassian. The Git solution for professional teams. URL
   https://bitbucket.org

12) TuXSTONe Ltda. TuXSTONe Technologies, URL
   http://www.tuxstone.com/website/
Villavicencio, 31 de Agosto de 2017

EL SUSCrito DIRECTOR OPERATivo
TuXSTONe Technologies
NIT. 822.004.594-3

HACE CONSTAR

Que la estudiante DIANA CAROLINA TARAPUES CHIRIVI identificado con cédula de ciudadanía número 52.929.077, prestó sus servicios a nuestra empresa como PASANTE desde el 07 de marzo del año 2017, hasta el treinta de agosto del año 2017, dentro del proyecto de REINGENIERÍA PARA SISTEMAS LEGADOS DESARROLLADOS POR TUXTONE TECHNOLOGIES IMPLEMENTANDO UN FRAMEWORK PHP MEDIANTE EL REDISEÑO DEL SOFTWARE DE CONTRATACIÓN WEB.

Durante la prestación de sus servicios cumplió sus funciones y las ha desempeñado de manera honesta, responsable y profesional, por lo que se da como aprobado su trabajo.

La presente constancia se expide a solicitud de la interesada a los treinta (31) días del mes de Agosto de Dos Mil Dieciséis (2017) en VillavicenCio.

Firma Autorizada,

JUAN FAJARDO BARRERO
C.C. 86055406
DIRECTOR OPERATIVO
TuXSTONe Technologies Limitada
Villavicencio, 31 de Agosto de 2017

EL SUSCRITO DIRECTOR OPERATIVO
TuXSTONE Technologies
NIT. 822.004.594-3

HACE CONSTAR

Que la estudiante DIANA CAROLINA TARAPUES CHIRIVI identificada con cédula de ciudadanía número 52.929.077, dictó la capacitación a nuestra empresa como una de las actividades dentro del proyecto de REINGENIERÍA PARA SISTEMAS LEGADOS DESARROLLADOS POR TUXSTONE TECHNOLOGIES IMPLEMENTANDO UN FRAMEWORK PHP MEDIANTE EL REDESÉNIO DEL SOFTWARE DE CONTRATACIÓN WEB.

La presente constancia se expide a solicitud de la interesada a los treinta (31) días del mes de Agosto de Dos Mil Dieciséis (2017) en Villavicencio.

Firma Autorizada,

JUAN FAJARDO BARRERO
c.c. 86055406
DIRECTOR OPERATIVO
TuXSTONE Technologies Limitada