



FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

DESARROLLO DE UN SISTEMA WEB PROGRESIVO PARA EL
CONTROL Y TOMA DE LECTURAS DE AGUA POTABLE

AUTOR

William Alex Marcillo Matute

AÑO

2019



FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

DESARROLLO DE UN SISTEMA WEB PROGRESIVO PARA EL
CONTROL Y TOMA DE LECTURAS DE AGUA POTABLE

Trabajo de Titulación presentado en conformidad a los requisitos
establecidos para optar por el título de Ingeniero en Sistemas de
Computación e Informática.

Profesor Guía

Dr. Jorge Luis Pérez Medina

Autor

William Alex Marcillo Matute

Año

2019

DECLARACIÓN DEL PROFESOR GUÍA

"Declaro haber dirigido el trabajo, desarrollo de un sistema web progresivo para el control y toma de lecturas de agua potable, a través de reuniones periódicas con el estudiante William Alex Marcillo Matute, en el semestre 201910, orientando sus conocimientos y competencias para un eficiente desarrollo del tema escogido y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación".

Jorge Luis Pérez Medina

Doctor Especialidad Informática

C.I. 1758993123

DECLARACIÓN DEL PROFESOR CORRECTOR

“Declaro haber revisado este trabajo, desarrollo de un sistema web progresivo para el control y toma de lecturas de agua potable, del estudiante William Alex Marcillo Matute en el semestre 201910 dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación”.

Carlos Andrés Muñoz Cueva

Master en Gerencia de Sistemas

C.I. 1712981511

DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE

"Declaro que este trabajo es original, de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de los autores vigentes".

William Alex Marcillo Matute

C.I. 1722704374

AGRADECIMIENTOS

A mi Dios, me sigue bendiciendo cada día, a mi Familia por su incondicional apoyo, a mis amigos, juntos hemos superado muchos retos, a los docentes que me forjaron.

DEDICATORIA

Dedico este trabajo a mis padres, Icela y Pato, su gran ejemplo es mi fuente de motivación. A Ángel M., Paola G., Nick G., Paul G., Tito M., Marco L., la mejor familia los mejores amigos, lo mejor de mi vida, los amo.

RESUMEN

Este proyecto pretende optimizar los procesos de control y toma de lecturas de medidores de agua potable, para la Empresa Municipal De Agua Potable (EMAP) administrada por el GAD (Gobierno Autónomo Descentralizado) Municipal de Pimampiro. El proyecto se desarrolló mediante el concepto de aplicaciones web progresivas, esta tecnología permitió el desarrollo de un módulo web donde un usuario con rol administrativo registra los medidores con marcadores, sobre un área dibujada por el usuario dentro de un mapa georreferenciado. El módulo web administra información de usuarios, abonados, cuentas, sectores y permite verificar el recorrido de los lectores en una jornada.

Lo progresividad permitió al mismo tiempo el desarrollo de una aplicación móvil donde un usuario con el rol lector tiene acceso a un mapa georreferenciado. El aplicativo especifica la ubicación geográfica de un medidor de agua, el usuario debe dirigirse al punto de medición para identificar a un abonado mediante un código QR para registrar la lectura, determinar los valores de consumo y pago.

Esta solución se propuso para evitar la recolección manual de lecturas, evitar la digitación de lecturas, mejorar tiempos de toma de lecturas, controlar lectores, favorecer tiempos de emisión y mejorar la productividad en general. El desarrollo de este proyecto de basa en el uso de herramientas con licenciamiento libre, una arquitectura definida y la aplicación de metodologías ágiles.

El estudio de usabilidad del proyecto recogió resultados favorables, los usuarios expresaron satisfacción con la interfaz, la información, la facilidad de uso y la interacción, para completar sus tareas eficientemente. Este proyecto, es un estudio que plantea la automatización de procesos de gestión de lecturas de agua potable, y puede ser viable para futuras implementaciones en otros municipios del país.

ABSTRACT

This project aims to optimize the processes of control and drinking water meter reading readings, for the Municipal Drinking Water Company (EMAP) administered by the GAD (Decentralized Autonomous Government) of Pimampiro. The project was developed through the concept of progressive web applications, this technology allowed the development of a web module where a user with an administrative role registers the meters with markers, over an area drawn by the user within a georeferenced map. The web module manages information of users, subscribers, accounts, sectors and allows to verify the readers' journey in a day.

The progressivity allowed at the same time the development of a mobile application where a user with the reader role has access to a georeferenced map. The application specifies the geographical location of a water meter, the user must go to the measurement point to identify a subscriber using a QR code to record the reading, determine the consumption and payment values.

This solution was proposed to avoid the manual collection of readings, to avoid the typing of readings, to improve reading times, to control readers, to favor emission times and to improve productivity in general. The development of this project is based on the use of tools with free licensing, a defined architecture and the application of agile methodologies.

The study of usability of the project gathered favorable results, the users expressed satisfaction with the interface, the information, the ease of use and the interaction, to complete their tasks efficiently. This project is a study that raises the automation of drinking water readings management processes and may be viable for future implementations in other municipalities of the country.

ÍNDICE

INTRODUCCIÓN	1
GAD Municipal Anfitrión	2
Problemática.....	2
Objetivo general.....	5
Objetivos específicos.....	5
1. CAPÍTULO I. MARCO TEÓRICO	6
1.1. IDE.....	6
1.1.1 Visual Studio Code	6
1.2. Herramienta De Gestion de Proyecto	8
1.2.1. VivifyScrum	8
1.3. Herramienta de prototipado de interfaces.....	9
1.3.1. Marvel	10
1.4. Front End.....	11
1.4.1. Diseño y Estilo.....	12
1.4.1.1. HTML.....	12
1.4.1.2. HTML5.....	13
1.4.1.3. CSS.....	13
1.4.1.4. CSS3.....	15
1.4.1.5. Bootstrap	16
1.4.2. Java Script.....	18
1.4.2.1. Librerías JavaScript	12
1.4.2.1.1. Angular	20
1.4.2.1.2. Leaflet.....	21
1.4.2.1.3 Draw	22
1.4.2.1.4 Instascan	23
1.5. Back-End.....	25

1.5.1. Node Js	25
1.5.2. Sails	26
1.6. Base de Datos	28
1.6.1. PostgreSQL	28
1.6.2. TeamSQL	30
1.7. Control de versiones de código fuente	31
1.7.1. Git	32
1.7.2. GitHub	32
1.8. OpenSSL	33
1.9. Ngrok	34
1.10. Arquitectura	37
1.10.1. Modelo Vista Controlador MVC	37
1.11. Metodologías de desarrollo	39
1.11.1. Metodologías de desarrollo ágil	39
1.11.1.1. Metodología Kanban	40
1.11.1.2. Diseño centrado en el usuario	43
1.12. Usabilidad	47
1.12.1. Cuestionario de Usabilidad	48
1.11.1. Tarjetas de reacción de productos de Mricrosoft	50
1.13. Conclusión de capítulo	51
2. CAPÍTULO II. ANÁLISIS Y DISEÑO	51
2.1. Aplicaciones Web progresivas	51
2.1.1. Shell de la App	52
2.1.2. Service Workers	53
2.1.3. Manifiesto de las aplicaciones web progresivas	54
2.2. Especificación de los requerimientos	56
2.2.1. Historias de usuario	58
2.2.1.1. Gestión de lecturas	59
2.2.1.2. Gestión de sectores	63

2.2.1.3. Gestión de abonados.....	64
2.2.1.4. Gestión de usuarios.....	65
2.2.1.5. Gestión de Cuentas.....	66
2.3. Diseño de Arquitectura.....	67
2.3.1. Diagrama lógico de la base de datos.....	67
2.3.2. Diagrama de arquitectura.....	69
2.3.3. Diagrama de componentes.....	71
2.4. Conclusión de Capítulo.....	72
3. CAPÍTULO III. DESARROLLO Y CODIFICACIÓN.....	72
3.1. Flujo de trabajo.....	72
3.1.1. Tarjeta:LEC-01.....	73
3.1.2. Tarjeta:LEC-02.....	74
3.1.3. Tarjeta:LEC-03.....	75
3.1.4. Tarjeta:LEC-04.....	76
3.1.5. Tarjeta:LEC-05.....	77
3.1.6. Tarjeta:LEC-06.....	78
3.1.7. Tarjeta:LEC-07.....	79
3.1.8. Tarjeta:LEC-08.....	80
3.1.9. Tarjeta:ADM-01.....	81
3.1.10. Tarjeta:ADM-02.....	82
3.1.11. Tarjeta:ADM-03.....	83
3.1.12. Tarjeta:ADM-04.....	84
3.1.13. Tarjeta:ADM-05.....	85
3.1.14. Tarjeta:ADM-06.....	86
3.1.15. Tarjeta:ADM-07.....	87
3.1.16. Tarjeta:ADM-08.....	88
3.1.17. Tarjeta:ADM-09.....	89
3.1.18. Tarjeta:ADM-10.....	90
3.1.19. Tarjeta:ADM-11.....	91
3.1.20. Tarjeta:ADM-12.....	92

3.1.21. Tarjeta:ADM-13	93
3.1.22. Tarjeta:ADM-14	94
3.1.23. Tarjeta:ADM-15	95
3.2. Conclusión de capítulo	96
4. CAPÍTULO IV. ESTUDIO DE USABILIDAD	96
4.1. Participantes	97
4.2. Tareas	98
4.2.1. Tareas usuario administrador	98
4.2.1. Tareas usuario lector	99
4.3. Materiales y métodos	99
4.4. Procedimiento	114
4.5. Resultados	114
4.5.1. Resultados GLAP Web.....	103
4.5.2. Resultados Glap Móvil.....	108
4.6. Conclusión del capítulo	1144
5. CONCLUSIONES Y RECOMENDACIONES	115
5.1. Conclusiones	115
5.2. Recomendaciones.....	116
REFERENCIAS.....	117

Introducción

A finales del año 2015 google nombró como Aplicaciones Web progresivas, en inglés Progressive Web Apps (PWA) (Steiner, 2018), para mencionar a aquellas aplicaciones web que son indistinguibles de las aplicaciones móviles, promoviendo su desarrollo y fijando características que una aplicación debe tener para ser parte de esta categoría sin embargo, los verdaderos inicios se darían en el año 2005 por el recién creado Web Hypertext Application Technology Working Group (WHATWG) la normativa de aplicaciones web 1.0 que más adelante se convirtió en el standard HTML5. (Anthes, 2012).

Un sistema georreferenciado es la integración de hardware y software diseñado para mostrar, analizar, almacenar y manipular información geográfica con el fin de resolver problemas complejos de planificación y gestión.

Considerando que el manejo responsable de los recursos hídricos se ha convertido en una de las principales prioridades de los municipios de Ecuador (GAD Municipal Pimampiro, 2018) es importante contar con una infraestructura tecnológica que ayude a optimizar recursos para su gestión.

A nivel Nacional existen pocos sistemas o aplicativos orientados a los procesos de toma de lecturas de agua potable, entre las tecnologías existentes en algunos municipios se encuentran sistemas ERP, donde el proceso actual se limita a la recolección manual de mediciones en hojas de registro donde posteriormente se digitan las lecturas para el preprocesamiento de datos y emisión de planillas.

Por su mayor presupuesto solo las municipalidades más grandes tienen sistemas de medición mediante dispositivos móviles, a pesar de las aplicaciones existentes, no hay herramientas con un enfoque gráfico que permita mostrar la gestión de la información mediante mapas georreferenciados.

GAD Municipal Anfitrión

Posterior a la fundación de Ibarra, Se establecen las autoridades civiles con el nombramiento de los alcaldes en las parroquias. Por decreto establecido por la Gran Colombia Pimampiro es considerada como parroquia el 25 de junio de 1824.

Mas tarde el 21 de mayo de 1981 el Gobierno del Dr. Jaime Roldós Aguilera ejecutaría un decreto legislativo que constituiría como cabecera cantonal la Parroquia de Pimampiro.

Problemática

Las entidades públicas como los Municipios del Ecuador, entre ellas el GAD Municipal Pimampiro, están en constante mejora de sus procedimientos para una mejor atención a sus contribuyentes. El GAD Municipal Pimampiro busca una mejor distribución de recursos y optimización de sus procesos. Dentro de este margen el proveer el servicio de agua potable es una de sus principales competencias.

La figura 1, muestra el modelo actual del proceso de emisión de planillas del GAD Municipal de Pimampiro. El proceso de emisión inicia con el técnico especializado de la empresa de agua potable, asigna a cada lector la ruta que debe recorrer en el mes, las rutas son fijas y los lectores llevan un registro a mano donde realizan las anotaciones de lecturas actuales, al finalizar la rutina el lector entrega el registro al personal de digitación para el ingreso de valores en el sistema donde se determina el consumo y el valor presuntivo de la factura según la categoría de la cuenta del abonado.

En un paso posterior el técnico del agua potable comprueba si los valores ingresados son los correctos en base al consumo promedio en base a meses anteriores determinado por un sistema. Cuando todas las lecturas han sido ingresadas el técnico crea un proceso de simulación de emisión global de títulos.

El proceso continúa con el jefe de rentas que toma el mismo proceso para volver a simular y comprobar que todos los valores sean correctos y finalmente se realiza la emisión global de títulos del mes correspondiente para que el contribuyente pueda pagar en la caja.

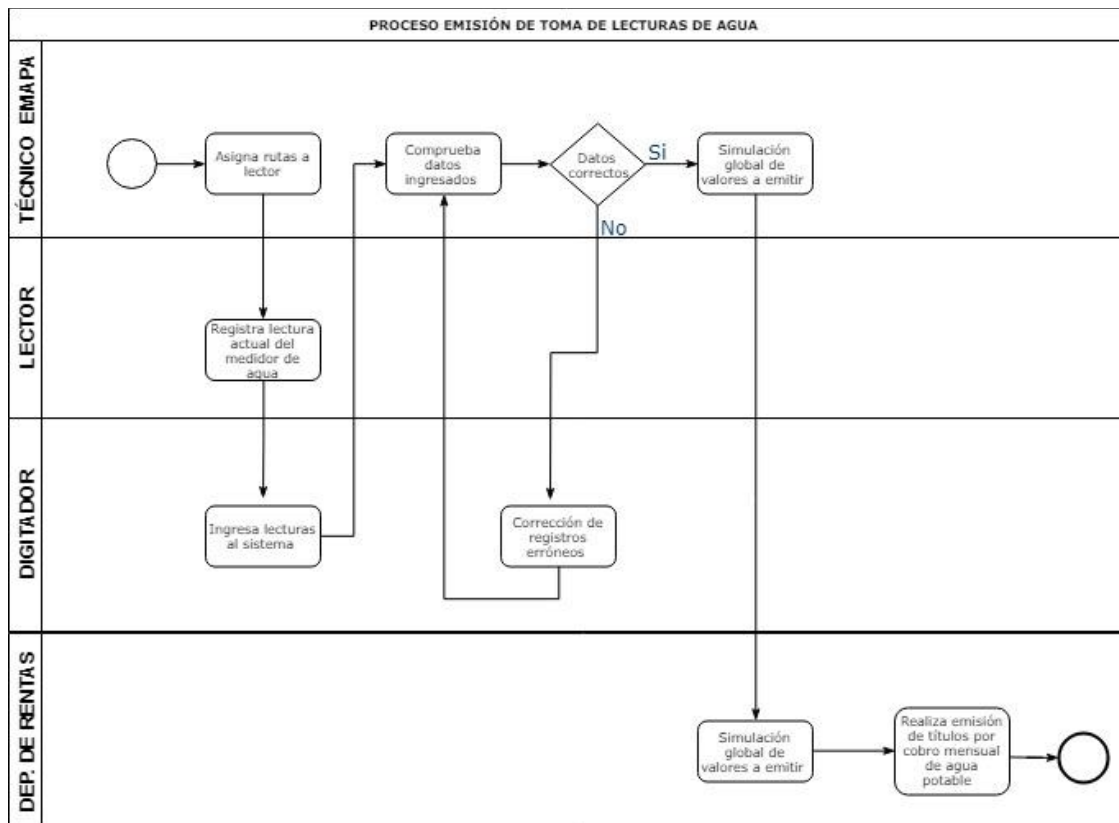


Figura 1. Proceso de toma de lecturas de agua potable.

El proceso actual es deficiente porque la recolección de datos de lecturas retrasa la emisión global de planillas hasta que los digitadores concluyan el ingreso manual al sistema. Al mismo tiempo se genera la probabilidad de errores de digitación y pérdida o deterioro de la hoja de ruta. Finalmente, no existe un control eficaz del recurso humano porque el sistema actual no muestra tiempos ni rutas recorridas por los lectores.

Actualmente existen sistemas automáticos de extracción de envíos de datos de lecturas de agua potable comprendidos por hardware y software.

Por ejemplo, la empresa española Geconta cuenta con medidores con envío de datos a través de la red móvil (ver figura 2), además el sistema incorpora una válvula de apertura-cierre que puede ser operada remotamente para intervenir en cortes de servicios por mora.

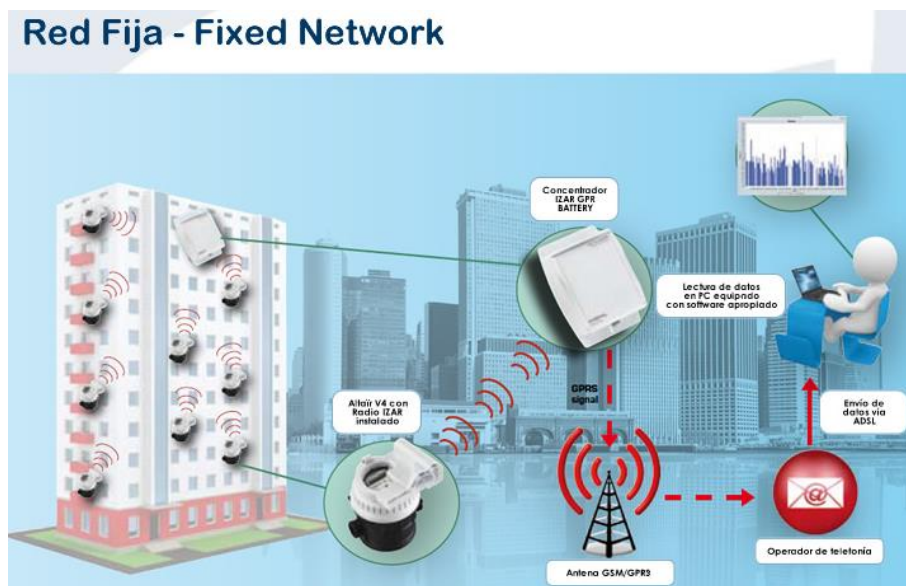


Figura 2. Medidores con red fija.

Tomado de: (Geoconta, 2018).

A pesar de las diferentes alternativas que ofrece el mercado internacional en cuanto a sistemas de integrados de medición de consumo de agua está claro que estas tecnologías aun tardarán en llegar a la realidad ecuatoriana, donde la primera dificultad tiene que ver con su elevado costo. Solo los municipios más grandes superarían este inconveniente porque manejan presupuestos mayores, pero dejarían fuera del alcance de las municipalidades más pequeñas. El segundo inconveniente tiene que ver con la infraestructura de red de agua potable, ya que para implementar dichas tecnologías se tendría que cambiar todos los medidores mecánicos actuales y sus costes recaería sobre los contribuyentes.

Por tal motivo se propone una alternativa que optimice el proceso actual mediante el desarrollo de un sistema web progresivo un mapa georreferenciado que permite identificar un medidor y su abonado para el ingreso y cálculo de su consumo mensual, de esta manera el Municipio de Pimampiro se beneficiará de una mejor organización de la información, tendrá un mejor control de lectores y distribución de rutas, los controles de reportes por daños permitirán que inconvenientes por fugas de agua sean atendidos con mayor brevedad, todo esto con la implementación de tecnología libre de los costes de licenciamiento.

Objetivo general

Desarrollar un sistema web Progresivo que permita mejorar el proceso de toma de lecturas de los medidores de agua potable mediante la automatización de la administración de información de la Empresa Municipal de Agua Potable (EMAP) del GAD Municipal de Pimampiro.

Objetivos específicos

- Proveer a la administración del GAD Municipal Pimampiro una tecnología de bajo coste para adaptarla a sus necesidades de mejorar el control de abonados, rutas, lectores, estados de conexión, consumo y reporte de daños.
- Implementar un sistema web progresivo para el control y toma de lecturas de agua potable mediante herramientas de información geográfica para administrar y analizar los datos de medidores georreferenciados.
- Desarrollar un sistema web progresivo para la Empresa Municipal de Agua Potable (EMAP) del GAD Municipal de Pimampiro para automatizar los procesos de emisión de planillas de agua potable basada en la metodología de desarrollo ágil Kanban.
- Estudiar el impacto la eficiencia y la mejora de usar la plataforma en el proceso de emisión de planilla de agua potable.

1. CAPÍTULO I. MARCO TEÓRICO

Este capítulo especifica la arquitectura del sistema, metodología de desarrollo, gestión de versiones, librerías, herramientas back-end y front-end que se utilizaron en el proyecto de titulación.

1.1. IDE

Aplicativo que ofrece servicios integrales para la codificación del desarrollo de sistemas informáticos, provee al usuario herramientas de construcción automáticas, un depurador autocompletado y manejo de extensiones.

1.1.1. Visual Studio Code

Visual Code es un editor de código fuente ligero pero potente con la capacidad de ejecutarse en plataformas macOS, Windows y Linux. Incorpora soporte para Node.js, JavaScript, TypeScript y tiene la capacidad de agregar una extensa gama de complementos para admitir lenguajes, entre ellos PHP, Go, C++, Python (VSC, 2018) . La figura 3 muestra una ventana de ejecución de Visual Studio Code.

Características:

- **Intellisense:** resalta y autocompleta la sintaxis, es decir proporciona terminaciones inteligentes basadas en los distintos tipos de variables, módulos importados y funciones definidas.
- **Depuración:** mediante una consola interactiva la depuración se puede realizar directamente desde el editor con pilas de llamadas y puntos de ruptura.

- Git Incorporado: visual studio code trabaja con proveedores SCM para revisar las diferentes etapas de los archivos manejando un adecuado control de versiones, directamente desde el editor se puede realizar “push and pull” de cualquier servicio.
- Extensiones: las funcionalidades se pueden ampliar con la instalación de extensiones para agregar nuevos idiomas, depuradores específicos, conexión a servicios adicionales. La ventaja principal es que las aplicaciones se ejecutan por procesos separados garantizando que no se ralentice el editor.

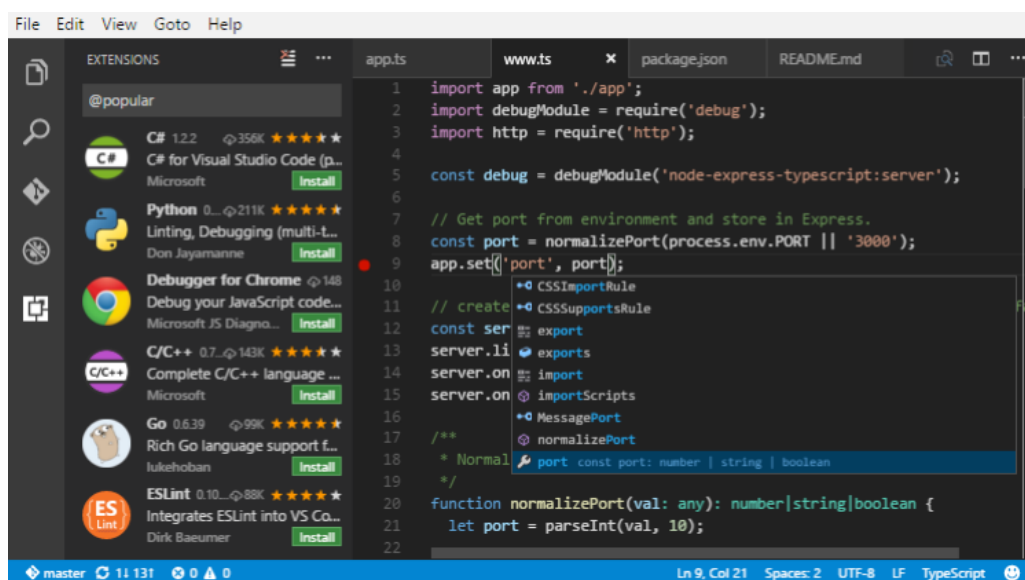


Figura 3. Interfaz gráfica de Visual studio code

Tomado de: (VSC, 2018).

Se eligió este editor porque a diferencia de otros, integra una terminal para manejo de comandos de nodejs y git. Visual Code es una herramienta de uso libre con una gran comunidad que genera colaboraciones, aporta con una extensa documentación y manuales de usuario.

Otro aspecto a notar es que desde la misma interfaz maneja el control de versiones en repositorio local o servicio remoto.

1.2. Herramienta de gestión de Proyecto

Una herramienta de gestión de proyectos es una aplicación que sirve para organizar y planificar el trabajo de un equipo de desarrollo, además permite al líder de un proyecto ajustar los lineamientos y parámetros de una metodología de desarrollo.

1.2.1. VivifyScrum

VivifyScrum crea el entorno apropiado para la administración de proyectos orientados idealmente a metodologías ágiles como scrum o Kanban.

Al seleccionar un proyecto enfocado en Kanban, vivifyScrum crea el tablero de tareas característico de la metodología para agregar tareas, de esta manera le da al equipo de trabajo la oportunidad de colaborar, permite realizar un seguimiento de su desempeño y se puede personalizar el tablero según las necesidades.

Luego de la creación del tablero se puede importar los miembros del equipo y dar acceso a los tableros existentes. El tablero Kanban maneja por defecto tres estados de tareas: por hacer, en progreso y listo.

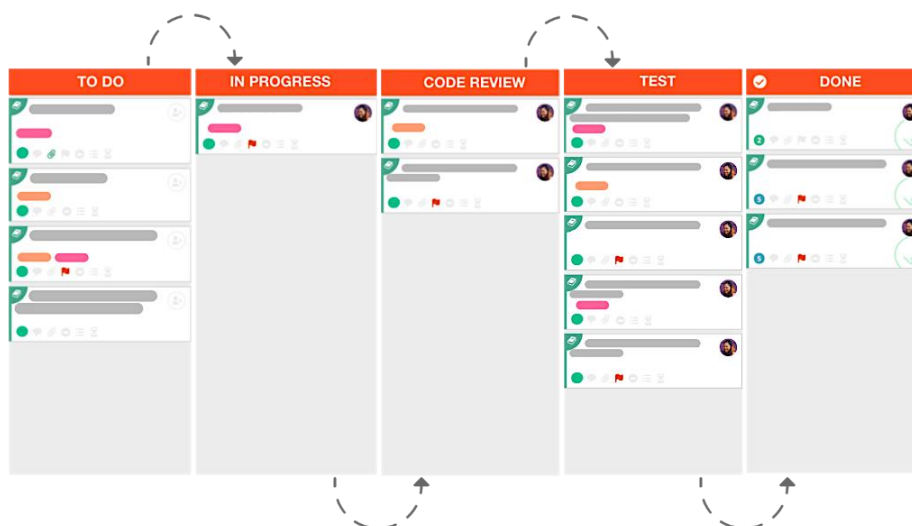


Figura 4. Tablero kanban de la herramienta vivifyScrum.

Tomado de: (Vivify, 2018).

La figura 4, muestra una representación de la interfaz principal de la herramienta vivifyScrum. En esta figura podemos observar las diferentes tareas clasificadas en un estado del tablero Kanban como por ejemplo el estado “To Do” (tarea pendiente por hacer), “In Progress” (Realizándose ahora mismo) y “Done” (Tareas cumplidas). La herramienta vivifyScrum también permite categorizar las tarjetas mediante colores e iconos por mejora, bugs, ideas, tareas, historia y notas. La herramienta permite personalizar el tablero con columnas adicionales sin embargo Kanban sugiere 3.

Características:

- Proporciona una representación visual del proceso del trabajo del equipo, donde se mueven los elementos de un estado a otro permitiendo identificar los cuellos de botella.
- Para representar de mejor manera las tareas el tablero es totalmente personalizable y se pueden añadir etiquetas de diferentes tipos de trabajo.
- Proporciona cuadros estadísticos a nivel individual y de equipo para medir el progreso y realizar ajustes al proyecto para alcanzar las metas.
- Incorpora un rastreador de tiempo para comparar las listas de trabajo frente a los “break point”.

Se eligió esta aplicación porque comparada con otras herramientas VivifyScrum ofrece la capacidad de realizar un seguimiento más especializado de los progresos un proyecto, tiene un seguidor de tiempo que determina cuanto duro una tarea, también incorpora un log de cambios que se hayan generado en el tablero.

1.3. Herramienta de prototipado de interfaces

Las aplicaciones de prototipos de interfaz permiten al desarrollador generar un diseño previo de su interfaz gráfica de usuario, esto permite proponer al usuario diseños visualmente atractivos e intuitivos antes del desarrollo final,

esto representa una ventaja ya que el usuario también podría proponer cambios que se ajusten de mejor manera a su necesidad.

1.3.1. Marvel

Marvel es una herramienta de diseño de interfaces digitales para aplicaciones iPhone, Android, web, iPad, Apple TV y Apple watch. Nació como un proyecto desde el 2013 y ahora cuenta con más de 2 millones de usuarios (MarvelApp, 2017). Mediante una plataforma web crea aplicaciones interactivas sin requerir de código. Permite la retroalimentación de ideas desarrollar productos novedosos. La figura 5 muestra la interfaz para el diseño de interfaces.

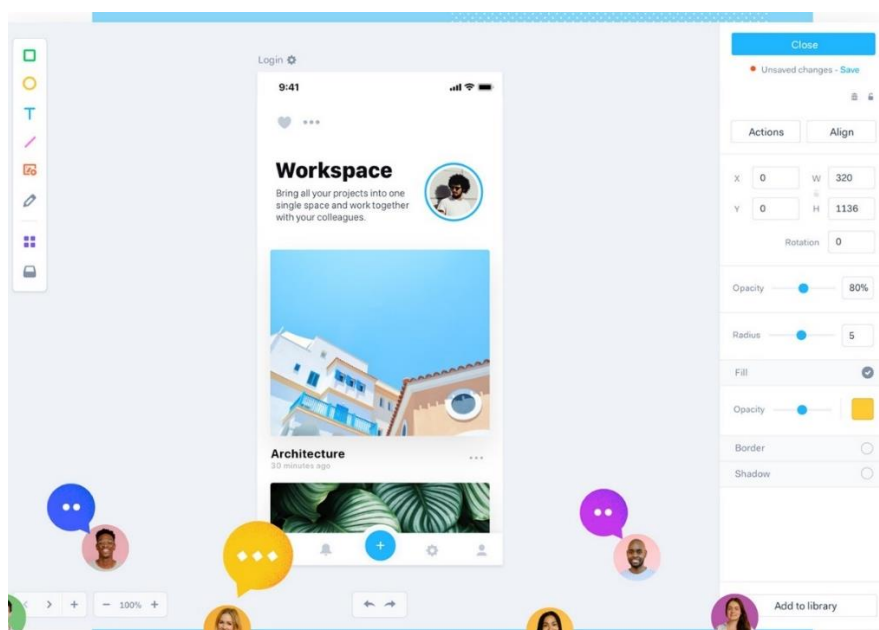


Figura 5. Canvas de diseño de interfaces Marvel App.

Tomado de: (MarvelApp, 2017)

Características:

- Compatible con formatos creados en Photoshop y sketch mediante la descarga de complementos, también se puede sincronizar con plataformas Google drive y Dropbox que facilita la gestión de versiones y el control de cambios.

- Genera prototipos que se adaptan a cualquier dispositivo o sistema operativo.
- Obtiene un resultado más realista con la incorporación de animaciones, gestos y transiciones.
- Marvel permite la colaboración en línea a distintos niveles, esto ofrece al equipo discutir procesos de desarrollo para luego discutirlos con el cliente final.
- Aumenta la experiencia con un aplicativo móvil con la cual se pueden revisar prototipos en dispositivos móviles.

Se tomo en cuenta a Marvel App, porque su versión gratuita ofrece todas las funcionalidades de la versión paga, excepto la colaboración en línea, que para el propósito del desarrollo de este proyecto no es requerida. La elaboración de prototipos es rápida ya que ofrece plantillas preelaboradas totalmente editables, entre ellas están: botones, formularios, tarjetas, mapas.

1.4. FRONT-END

La palabra Front-End se asocia a los desarrolladores con los principios de diseño y estructura de las páginas web, es decir la interfaz gráfica que un usuario puede ver e interactuar. Para la composición de elementos Front-End un desarrollador debe tomar en cuenta varias tecnologías como; HTML, CSS, JavaScript, e involucrar aspectos como la usabilidad y la legibilidad de la página.

1.4.1. Diseño y Estilo

1.4.1.1. HTML

HTML se conoce como el Lenguaje Marcado de Hiper Texto y es la tecnología fundamental que se utiliza para definir la estructura de una página web. HTML está compuesto de varios elementos que se declaran mediante etiquetas de texto para incrustar contenido (Mozilla, 2018). Es un lenguaje interpretado y no compilado, de manera que los navegadores no muestran las etiquetas, pero las utilizan para mostrar el contenido. En la figura 6, se muestra una estructura básica del etiquetado HTML para una página web. La figura 6 muestra la configuración de las etiquetas que conforman la cabecera y el cuerpo de una página web.

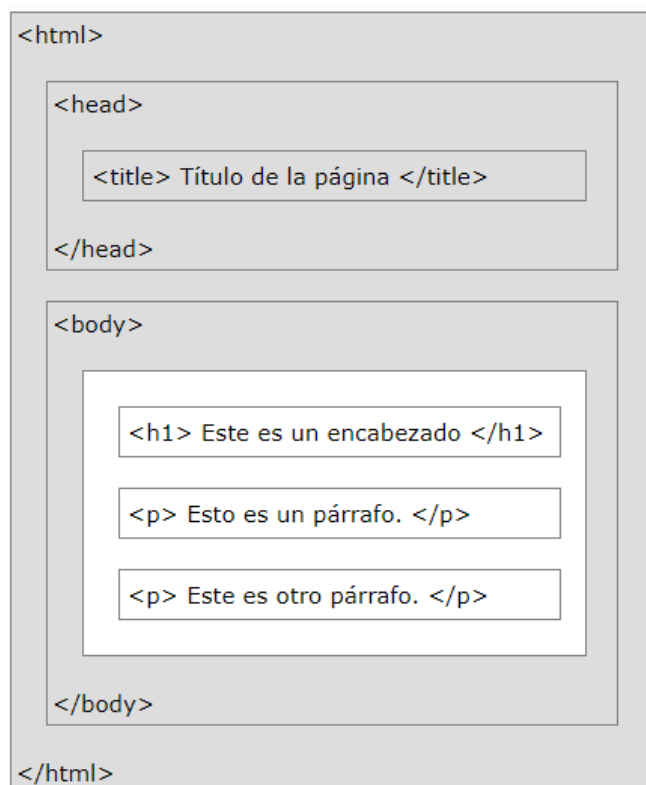


Figura 6. Estructura básica y etiquetado HTML

Tomado de: (Mozilla, 2018).

1.4.1.2. HTML5

HTML5 es la última evolución del estándar que define HTML. El término representa dos conceptos diferentes. Es una nueva versión del lenguaje HTML, con nuevos elementos, atributos y comportamientos. Además, es un conjunto más amplio de tecnologías que permite la construcción de aplicaciones web más diversas y potentes (w3schools, 2018).

HTML5, podría clasificarse en varios grupos según su función:

- **Conectividad:** le permite comunicarse con el servidor de formas nuevas e innovadoras.
- **Sin conexión y almacenamiento:** permite que las páginas web almacenen datos en el lado del cliente localmente y operen fuera de línea de manera más eficiente.
- **Multimedia:** Streamig de audio y videos en la web.
- **Gráficos y efectos 2D / 3D:** permiten una gama mucho más diversa de opciones de presentación.
- **Rendimiento e integración:** proporciona una mayor optimización de la velocidad y un mejor uso del hardware de la computadora.
- **Acceso a dispositivos:** permite el uso de varios dispositivos de entrada y salida.

1.4.1.3. CSS

CSS es la abreviatura que define a las Hojas de Estilo en Cascada (en inglés, "Cascading Style Sheets"). CSS se implementa junto con HTML para describir la presentación de una página web, esto incluye entre varias cosas los colores, el diseño y fuentes. CSS es independiente de HTML y se puede utilizar con cualquier lenguaje de marcado basado en XML (W3C, 2018).

En la figura 7, muestra una porción de código ejemplo de una hoja de estilos para la cabecera y títulos de HTML.

Características:

- CSS ahorra tiempo: se escribe una vez y luego reutilizar la misma hoja en varias páginas HTML. Puede definir un estilo para cada elemento HTML y aplicarlo a tantas páginas web como se desee.
- Las páginas más rápidas: Al escribir una regla de CSS de una etiqueta y aplicarla a todas las apariciones de esa etiqueta, implica menos código, traduciéndolo a tiempos de carga más rápidos.
- Fácil mantenimiento: para realizar un cambio global, simplemente se cambia el estilo, y todos los elementos en todas las páginas web se actualizarán automáticamente.
- Estilos superiores a HTML: CSS tiene una gama de atributos mucho más amplia que HTML, por lo que permite dar un mejor aspecto a una página en comparación con los atributos HTML.
- Múltiples dispositivos: las hojas de estilo permiten optimizar el contenido para más de un tipo de dispositivo. Al utilizar el mismo documento HTML, se pueden presentar diferentes versiones de un sitio web tanto para dispositivos móviles, como para versiones de escritorio.
- Estándares web globales: junto a HTML, CSS es una tecnología con un estándar que se mantiene definido por las especificaciones de W3C.

```
1  #header {  
2  background-color:#4C0000;  
3  height:120px;  
4  width:100%;  
5  padding:20px;  
6  }  
7  
8  h1.blog-title a {  
9  font-color:#FFFFFF;  
10 font-size:16px;  
11 font-family: Georgia, "Times New Roman", serif;  
12 text-decoration:none;  
13 }
```

Figura 7. Ejemplo CSS utilizado en este proyecto.

1.4.1.4. CSS3

CSS3 es la última actualización del lenguaje de codificación de las hojas de estilo en cascada. Esta última evolución del lenguaje viene con cientos de innovaciones importantes, lo que permite a los programadores manipular sus sitios de formas que eran imposibles o mucho más difíciles de lograr en versiones anteriores de CSS (Wpbeginner, 2018). La figura 8, muestra las principales diferencias entre las versiones 1,2.1 y 3.

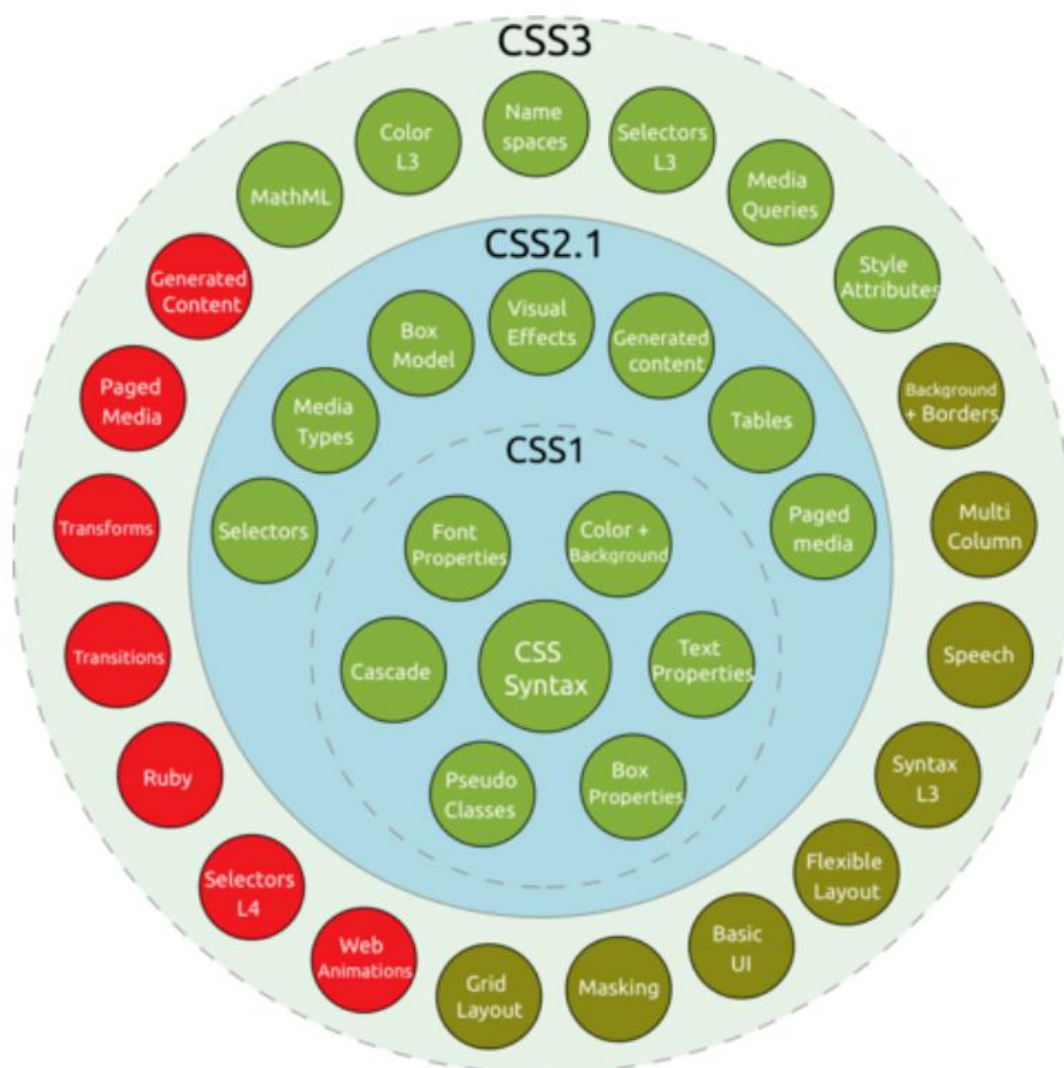


Figura 8. Versiones 1,2.1 y 3 de CSS

Tomado de: (Wpbeginner, 2018).

Algunos de los principales módulos de CSS3 son:

- Modelo de caja
- Valores de imagen y contenido reemplazado.
- Efectos de texto
- Selectores
- Fondos y fronteras
- Animaciones
- Interfaz de usuario (UI)
- Diseño de columna múltiple
- Transformaciones 2D / 3D

1.4.1.5. Bootstrap

Bootstrap 4, es el marco HTML, JavaScript y CSS en la versión más reciente del framework de desarrollo para aplicativos web responsivos y móviles. Es una biblioteca Front-End gratuita y de código abierto, contiene plantillas de diseño como tipografía, formularios, botones, barras de navegación y otros componentes de interfaz de usuario (Bootstrap, 2018). En la figura 9, se muestran algunos de los elementos más comunes de Bootstrap entre ellos botones, paneles, menús.

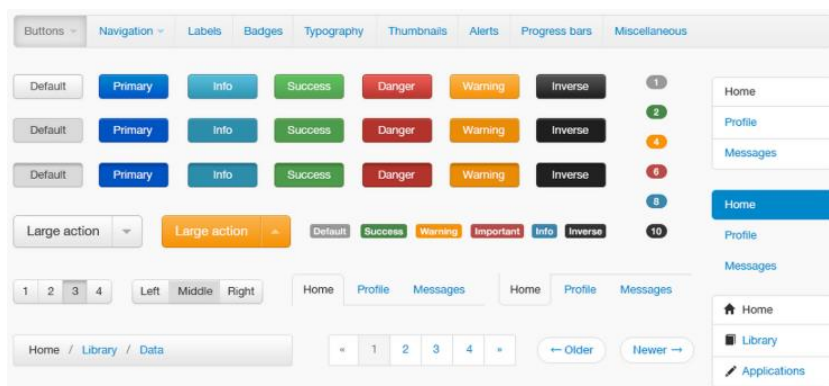


Figura 9. Componentes Bootstrap

Tomado de: (Bootstrap, 2018).

La figura 10, muestra el diseño del formulario Bootstrap de sectores de este proyecto.

The image shows a Bootstrap form titled "Nuevo Sector". It is enclosed in a light blue border. At the top, there is a header "Formulario" in a light gray box. Below this, the form contains three input fields: "Nombre", "Descripcion", and "Color". Each field has a label above it and a text input area. At the bottom of the form, there are two buttons: "DIBUJAR" (blue) and "GUARDAR" (green).

Figura 10. formulario Sectores

Para integrar Bootstrap a un proyecto web existen dos formas, la primera consiste en descargar la librería e incorporar su llamado en la cabecera del documento HTML. La segunda, consiste en pegar la etiqueta del CDN de Bootstrap en la cabecera del documento HTML. El segundo método pudiera ser óptimo para mantener el framework actualizado.

Para manejar las responsividad y que un sitio sea totalmente adaptativo a cualquier dispositivo en cualquier pantalla Bootstrap maneja un dominio de 12 columnas que pueden ser agrupadas de distintas maneras siempre y cuando estas sumen 12, como se muestra en la figura 11.

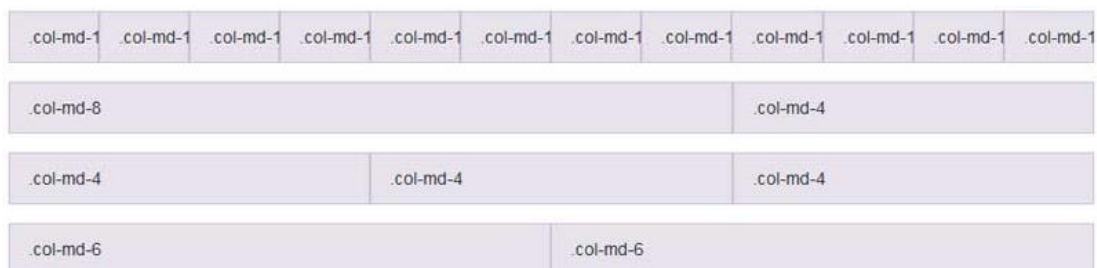


Figura 11. Sistema de columnas Bootstrap.

Tomado de: (Bootstrap, 2018).

Características:

- Fácil de usar e implementar.
- Amigable con dispositivos móviles.
- Bootstrap permite su personalización.
- Lista extensa de componentes prediseñados listos para usar.
- Características de respuesta rápida.
- Compatibilidad con todos los navegadores.
- Maneja JavaScript empaquetado.
- Tiene una extensa documentación y foros de la comunidad.
- Estilo base para la mayoría de los elementos HTML.

Se eligió Bootstrap como el marco de diseño para este proyecto porque es un framework que tiene variedad de componentes elaborados listos para usar, su implementación es fácil y casi no requiere cambios en las hojas de estilo para personalizarlo, esto reduce el tiempo de desarrollo en Front-End permitiendo al desarrollador enfocarse más en el desarrollo Back-End. Además, incluye variedad de temas para el caso que se quiera cambiar todo el diseño de una página.

1.4.2. JavaScript

Es un lenguaje de programación para la web. Es compatible con la mayoría de los navegadores web, incluidos Chrome, Firefox, Safari, Internet Explorer, Edge, Opera. La mayoría de los navegadores móviles para teléfonos inteligentes también admiten JavaScript (Mozilla , 2018).

Se utiliza principalmente para mejorar las páginas web para proporcionar una experiencia más fácil de usar.

Estos incluyen la actualización dinámica de páginas web, mejoras en la interfaz de usuario, como menús y cuadros de diálogo, animaciones, gráficos 2D y 3D, mapas interactivos, reproductores de video y más.

La figura 12, muestra una porción de código de ejemplo implementado en este proyecto, el cual contiene una función para recuperar los datos de las cuentas por sector.

JavaScript es un lenguaje interpretado por lo tanto no necesita ser compilado.

```
1 | function getCuentasxSector(sector) {
2   ServicioCuentas.buscarPorSectorId(sector.id).then(function (res) {
3     var listacuentas = []
4     console.log(res.data)
5     listacuentas = res.data;
6     console.log(listacuentas)
7     var listaMarcadores = [];
8     var geojson;
9
10    for (var i = 0; i < listacuentas.length; i++) {
11      var marcador = JSON.parse(listacuentas[i].coordenada) /// convierte
12 ▼   marcador.properties = {
13     'nombre': listacuentas[i].abonado_id.nombre,
14     'apellido': listacuentas[i].abonado_id.apellido,
15     'cedula': listacuentas[i].abonado_id.cedula,
16     'marcamedidor': listacuentas[i].marca_medidor,
17     'tarifa': listacuentas[i].tarifa,
18   };
19 ▼   if($scope.cuentaseleccionada.id != listacuentas[i].id){
20     listaMarcadores.push(marcador);
21   }
```

Figura 12. Ejemplo código JavaScript implementado en este proyecto.

Características:

- Es un lenguaje de scripting basado en objetos.
- Da al usuario más control sobre el navegador.
- Es un lenguaje de scripting basado en intérpretes.
- JavaScript es un lenguaje basado en objetos, ya que proporciona objetos predefinidos.
- tiene la capacidad de crear nuevas funciones dentro de los scripts.

1.4.2.1 Librerías JavaScript

1.4.2.1.1. Angular

Angular es un marco estructural para dar dinamismo a las aplicaciones web. Permite extender la sintaxis HTML para agregar sus propios componentes de manera clara. La declaración de las dependencias de Angular reduce el código en funcionalidades complejas que de otro modo se tendrían que codificar. Angular se ejecuta del lado del cliente en el navegador, lo cual hace que sea compatible con cualquier tecnología de servidor (AngularJS , 2018).

Angular infiere como una nueva sintaxis en el navegador mediante la declaración de constructores dominados directivas.

Características:

- Enlace de datos con la sintaxis “{{ }}”.
- Tiene estructuras de control para repetir, mostrar y ocultar fragmentos.
- Soporte para formularios y validación de formularios.
- Adjuntar un nuevo comportamiento a los elementos, como el manejo de eventos.
- Agrupación de HTML en componentes reutilizables.

AngularJS simplifica el desarrollo de aplicaciones al presentar un mayor nivel de abstracción al desarrollador, y es útil para las operaciones CRUD (Create Read Update Delete) de un sistema. Una de sus ventajas más representativas es que cada una de sus directivas tienen una funcionalidad específica que ahorra tiempo de programación. Por ejemplo, se puede implementar el ordenamiento de datos, se puede buscar elementos de una tabla, manejar repeticiones para extraer una lista de la base de datos, incluso manejar eventos tipo slide para dispositivos táctiles, en una sola línea de código con la declaración de una directiva.

1.4.2.1.2. Leaflet

Leaflet es una librería de JavaScript utilizada para creación de mapas interactivos en aplicaciones web y móviles. Provee a los desarrolladores todas las funcionalidades de un mapa con posicionamiento georreferenciado (Leaflet, 2018). La figura 13, muestra un ejemplo de un mapa web implementado en base a Leaflet.

Está diseñado para una fácil implementación, toma en cuenta la facilidad de uso, el rendimiento y simplicidad. Dispone de una gran documentación y su uso es libre porque está basado en las propiedades de Open Street Maps. La figura 14, muestra el código java script mínimo necesario para la implementación de un mapa con un marcador de coordenadas fijas.

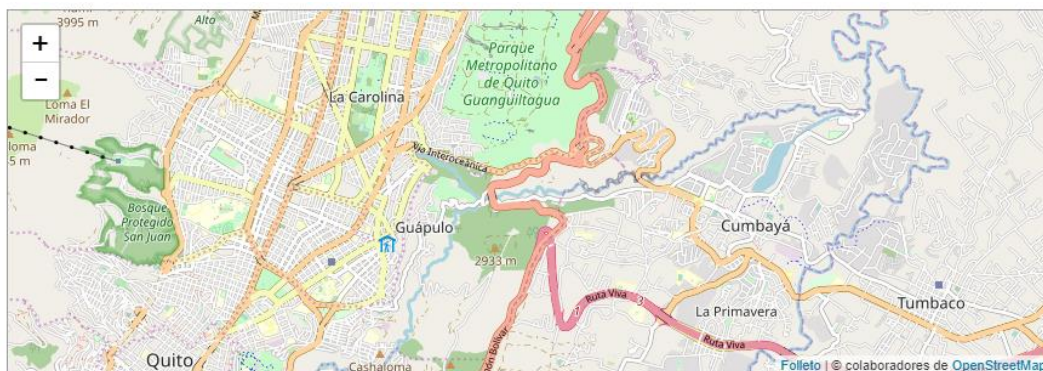


Figura 13. Mapa de la Ciudad de Quito con Leaflet implementado en el mapa general de este proyecto.

```
var map = L.map('map').setView([51.505, -0.09], 13);

L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
  attribution: '&copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors'
}).addTo(map);

L.marker([51.5, -0.09]).addTo(map)
  .bindPopup('A pretty CSS3 popup.<br> Easily customizable.')
  .openPopup();
```

Figura 14. Código Java Script que contiene el API Leaflet para implementación web.

Tomado de: (Leaflet, 2018).

Características:

- Tiene variedad de complementos para explorar múltiples funcionalidades.
- Distribuido con licenciamiento libre.
- Se puede personalizar el mapa con la selección de varios estilos de la librería.
- Permite agregar capacidades graficas como estilizar marcadores, dibujar polígonos y vincularlos a una base de datos.

Se eligió a Leaflet porque permite integra mapas a una aplicación web de manera rápida y sencilla mediante pocas líneas de código JavaScript y un CDN en la cabecera del código HTML. Una ventaja que destaca sobre otras APIs, es que su uso es totalmente libre, y no requiere de pagos cuando supera cierto límite de conexiones por usuario como lo hace GoogleMaps.

1.4.2.1.3. Draw

Draw es una librería JavaScript de Leaflet que permite la integración de controles para la edición de dibujos de figura geométricas y marcadores sobre el mapa.

Draw dibuja líneas, rectángulos, polígonos, círculos, marcadores y marcadores circulares. Draw permite modificar las propiedades de los gráficos como como la opacidad, color de borde, densidad de borde, color de relleno. (Leaflet, 2018) La librería draw retorna cada uno de los puntos generados en el dibujo en formato Json para ser guardados o extraídos en una base de datos. La figura 15 muestra el resultado obtenido de dibujar mediante los puntos de un polígono el sector Rio Coca georreferenciado, dentro de sus propiedades tiene una línea de borde color naranja, relleno de color naranja y transparencia en 0.2.

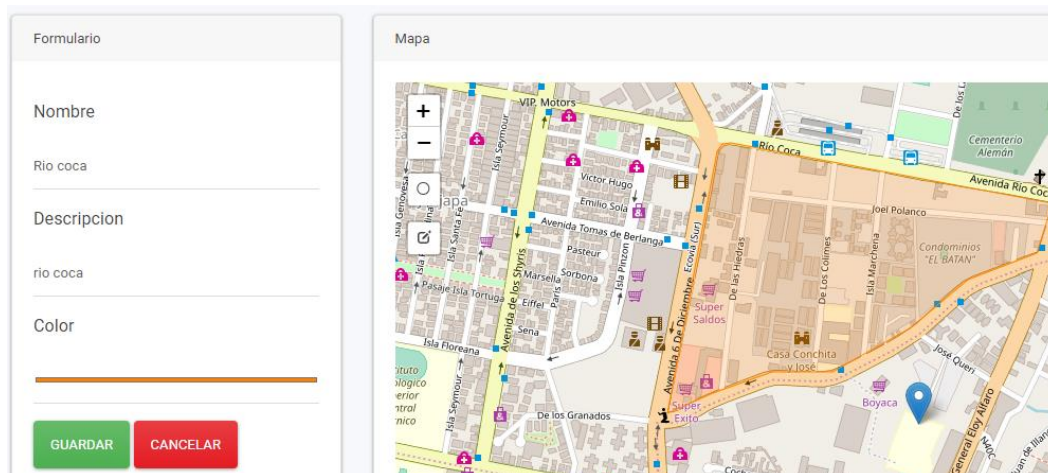


Figura 15. Dibujo del polígono del sector Río Coca con Leaflet Draw.

1.4.2.1.4. Instascan

Instascan es una librería JavaScript que permite la lectura en tiempo real de códigos QR mediante la activación de la cámara fotográfica de un dispositivo, puede ser implementada en un proyecto mediante NPM. Su uso y distribución es open Source (Schmich, 2017).

La figura 16, contiene el código JavaScript mínimo necesario para su implementación sobre HTML.

```
<script type="text/javascript">
  let scanner = new Instascan.Scanner({ video: document.getElementById('preview') });
  scanner.addListener('scan', function (content) {
    console.log(content);
  });
  Instascan.Camera.getCameras().then(function (cameras) {
    if (cameras.length > 0) {
      scanner.start(cameras[0]);
    } else {
      console.error('No cameras found.');
```

Figura 16. Código de implementación.

Tomado de: (Schmich, 2017)

La figura 17, muestra el funcionamiento del lector QR InstaScan en este proyecto, usado para la identificación de la cuenta de un abonado y ejecutado bajo el depurador de dispositivos de Google Chrome.



Figura 17. Modo de funcionamiento de Instascan para identificar abonados.

Características:

- Requiere HTTPS para su despliegue
- En el caso de smartphones detecta cámara frontal y delantera
- Compatible con android en navegadores Chrome, Firefox, Opera y Edge.
- Lectura rápida y confiable, no importa la orientación del código sobre la cámara, es decir las tomas de lectura de código pueden hacerse de manera vertical u horizontal.
- Maneja funciones predefinidas para resultados de lecturas, activación o cierre de cámaras.

1.5. BACK-END

1.5.1. NodeJs

NodeJs es un entorno de ejecución de java script del lado del servidor de código abierto y multiplataforma, se usa para el desarrollo de aplicaciones de red rápidas y escalables. NodeJs también proporciona bibliotecas y módulos JavaScript que simplifican el desarrollo de aplicaciones web (NodeJs, 2018).

NodeJs utiliza la biblioteca NPM (Node Package Manager) como un administrador de paquetes de descarga. La figura 18, presenta los componentes de operación.

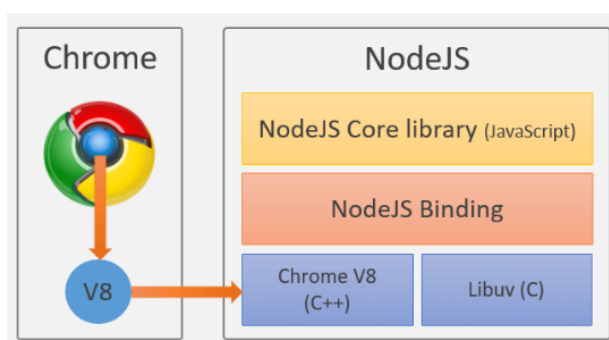


Figura 18. Arquitectura NodeJs.

Tomado de: (NodeJs, 2018)

Características:

- Node.js utiliza un modelo de E / S sin bloqueo controlado por eventos que lo hace liviano y eficiente.
- Perfecto para aplicaciones de tiempo real con uso intensivo de datos que se ejecutan en dispositivos distribuidos.
- Node es distribuido mediante licencia MIT.

- NodeJs no requiere de almacenar datos de las aplicaciones en el buffer, porque los datos siempre se emiten en tramas.
- Utiliza un programa de un solo subproceso y el mismo programa puede brindar servicio a un número mucho mayor de solicitudes que los servidores tradicionales.
- Permite programación orientada a eventos.

1.5.2. Sails

Sails es un framework de desarrollo para aplicaciones web con el marco MVC (Modelo Vista Controlador) para Node.js. Maneja conceptos como REST, HTTP, WebSockets, API, y tecnologías como Java, o Ruby, o Node.js. (SailsJs, 2018)

Sails proporciona acceso simple a la capa de datos ya que incorpora adaptadores compatibles para MongoDB, PostgreSQL, MySQL, Redis y acceso a datos del disco local. Sails provee seguridad y control de acceso basado en roles y políticas de forma predeterminada.

Características:

- Basado y construido sobre Express.js.
- Soporta web sockets en tiempo real.
- Toma un enfoque de “convención sobre configuración”.
- Cuenta con blue prints para una potente generación de código.
- La gestión de las bases de datos es independiente gracias a su potente línea ORM (Mapeador de Objetos Relacionales).
- Soporta múltiples bases de datos en el mismo proyecto.
- Crea un entorno para diferentes lenguajes de programación.
- Tiene una gran comunidad de colaboradores y basta documentación.

Creación de modelos:

Un modelo, de manera general es la representación de una abstracción en un objeto o una clase en un lenguaje de programación de propósito general. Sails crea y mapea las tablas de en bases de datos SQL y NoSQL.

Para la creación de la tabla abonados de este proyecto en la base, se simplifica en ejecutar la siguiente línea en el CLI de node:” sails generate model abonado”.

Para agregar atributos a la tabla abonados es necesario crear un archivo .js dentro del directorio “models” creado por sails y componer su estructura como lo muestra la figura 19.

```
14 sails generate model abonado
15
16 module.exports = {
17   attributes: {
18     nombre:{
19       type:"string"
20     },
21     apellido:{
22       type:"string"
23     },
24     cedula:{
25       type:"string"
26     },
27     telefono:{
28       type:"string"
29     },
30     correo:{
31       type:"string"
32     },
33     cuenta:{
34       collection:"cuenta",
35       via:"abonado_id"
36     },
37   },
38 },
39 };
```

Figura 19. Creación de entidad y atributos de la tabla abonado.

Sin embargo, estos cambios solo tendrán efecto en la base de datos solamente cuando el servidor web se haya levantado tras la ejecución del comando “sails lift” en la consola de node.

Cuando finaliza la operación PostgreSQL ha creado la tabla abonado con todos sus atributos, estos cambios se pueden comprobar mediante la herramienta cliente de PostgreSQL, figura 20.

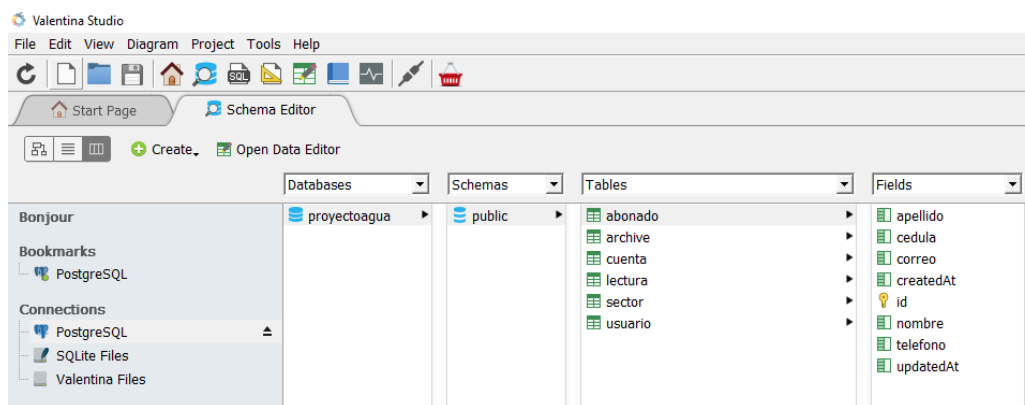


Figura 20. Comprobación de creación de entidades a partir de Sails.

Una de las razones principales por la que se eligió sails, es que brinda todas las cualidades de un servidor web, con la ventaja adicional que desde el mismo framework se pueden manejar los DDL y DML de la base de datos.

1.6. Base de Datos

1.6.1. PostgreSQL

PostgreSQL es un Sistema Manejador de Bases de Datos (DBMS) relacional de código abierto, cuenta y con más de 30 años de desarrollo activo en su comunidad. “Se ha ganado una sólida reputación de confiabilidad, funcionalidad y rendimiento” (The PostgreSQL Global Development Group, 2018).

PostgreSQL es compatible con todos los sistemas operativos y maneja transacciones de manera ACID (Atomicidad, Consistencia, Aislamiento Durabilidad).

PostgreSQL tiene una arquitectura cliente - servidor y en la figura 21 se muestran los sujetos involucrados en el marco de trabajo de operación normal del sistema gestor de base de datos.

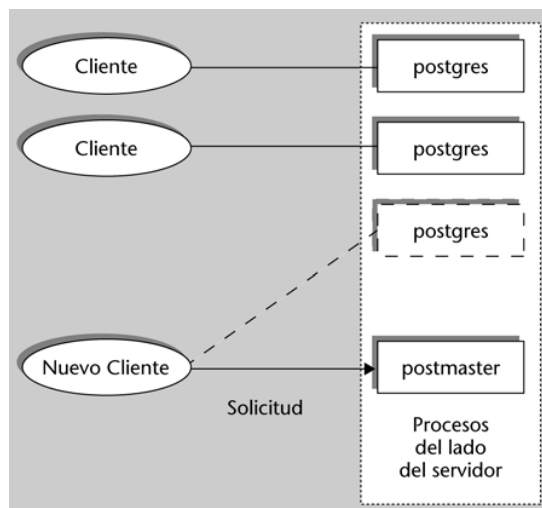


Figura 21. PostgreSQL, Arquitectura cliente servidor.

Tomado de: (The PostgreSQL Global Development Group, 2018)

Características:

- Manejo de rutinas complejas y avanzada funcionalidad de consultas declarativas que requieren control de concurrencia.
- Soporte multiusuario, extensible para métodos y tipos de datos definidos por el usuario.
- Proporciona soporte de desarrollo para múltiples lenguajes de programación gracias a su flexibilidad.
- Licenciamiento libre y altamente adaptable a las necesidades.
- Acceso al log de la base de datos que registra todas las transacciones para reponer el servicio en caso de una posible caída
- Documentación bien organizada, pública y libre.

Se optó por PostgreSQL porque no requiere de costos de licenciamiento. Posee características de integridad de datos en un entorno tolerante a fallas, de fácil administración e implementación. Cuenta con una extensa documentación y la comunidad constantemente lanza actualizaciones. Es compatible con el framework de desarrollo Sailsjs.

1.6.2. TeamSQL

TeamSQL es un cliente multiplataforma para la gestión de bases de datos, presenta una interfaz intuitiva y cómoda (TeamSQL, 2018). Se instala en macOS, Windows y Linux y puede gestionar bases de datos como:

- MySQL
- Microsoft SQL Server
- PostgreSQL
- AWS Redshift
- AWS Aurora
- Mariadb
- CitusData

Presenta una opción rápida de búsqueda lo cual permite encontrar datos y consultas de forma rápida y fácil. TeamSQL guarda un historial de consultas de manera local y también pueden guardarse en la nube (sincronización con Google drive) para ejecutarlas cuando se lo requiera. La figura 22, muestra la interfaz gráfica de TeamSQL.

Características:

- consultas guardadas en la nube.
- Busca todo, como tablas, vistas, consultas guardadas, funciones, procedimientos almacenados.

- Asigna color a la base de datos para distinguirlos fácilmente.
- Historial de ejecución de consultas.
- Permite conexión con múltiples bases de datos.
- Función grafica de análisis de datos en forma de barras, círculos, líneas, mapas de calor, entre otras.

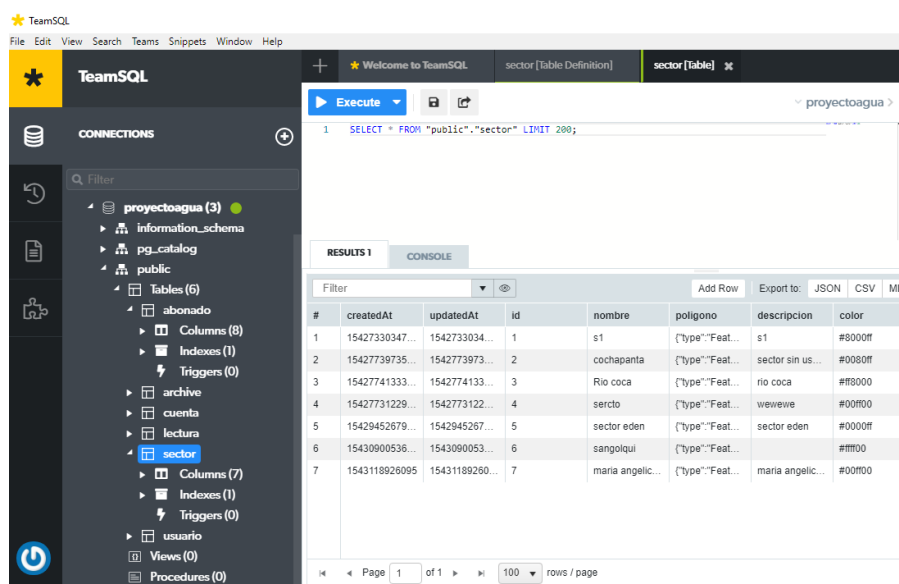


Figura 22. Consulta de la tabla lector mediante la interfaz gráfica TeamSQL.

Se eligió TeamSQL como herramienta para este proyecto porque su licenciamiento no implica costos. Simplifica el trabajo con la base de datos en una interfaz clara y funcional. Permite la descarga de complementos con funcionalidades adicionales.

1.7. Control de versiones de código fuente

Las herramientas de control de versiones son programas que registran todos los cambios realizados sobre los archivos a lo largo del tiempo, permite a un desarrollador comparar diferentes puntos de tiempo y volver a un estado anterior si así lo requiere.

1.7.1. Git

Git es un sistema de control de versiones, diseñado para todo tipo de proyectos con rapidez y eficiencia, es de código abierto y gratuito (Git-SCM, 2018).

Git tiene la capacidad de almacenar los cambios en un repositorio local y también permite la sincronización a repositorios externos como GitHub.

Git se centra en los archivos y su integridad por lo cual protege los envíos al historial de cambios con un algoritmo de criptográfico de hashing SHA1 para que las versiones no contengan cambios maliciosos o accidentales.

1.7.2. GitHub

Git hub es un repositorio en que se ofrece como un servicio en línea para alojar el código fuente de proyectos y controlar sus versiones. Puede contener carpetas y cualquier tipo de archivos. Gracias a su característica de colaboración y bifurcación se puede trabajar en dos versiones diferentes del mismo proyecto a partir del mismo punto de origen (GitHub, 2018). La figura 23 muestra el esquema de bifurcación a partir de la rama master.

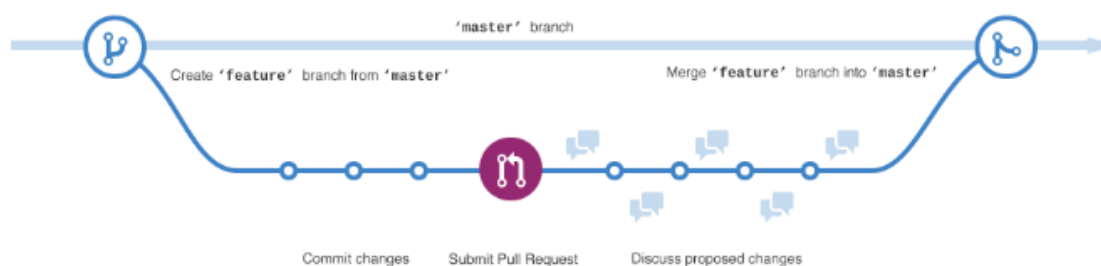


Figura 23. Esquema de bifurcación GitHub.

Tomado de: (GitHub, 2018)

Características:

- Su uso no requiere codificación.
- Permite trabajo colaborativo.
- Crea ramas y realiza cambios (Branch - commit).

- Clonación y sincronización hacia aplicativos de hosting para poner un proyecto a producción de inmediato.
- Vinculación a redes sociales para fomentar el crecimiento de proyectos y compartir conocimientos con otros desarrolladores.

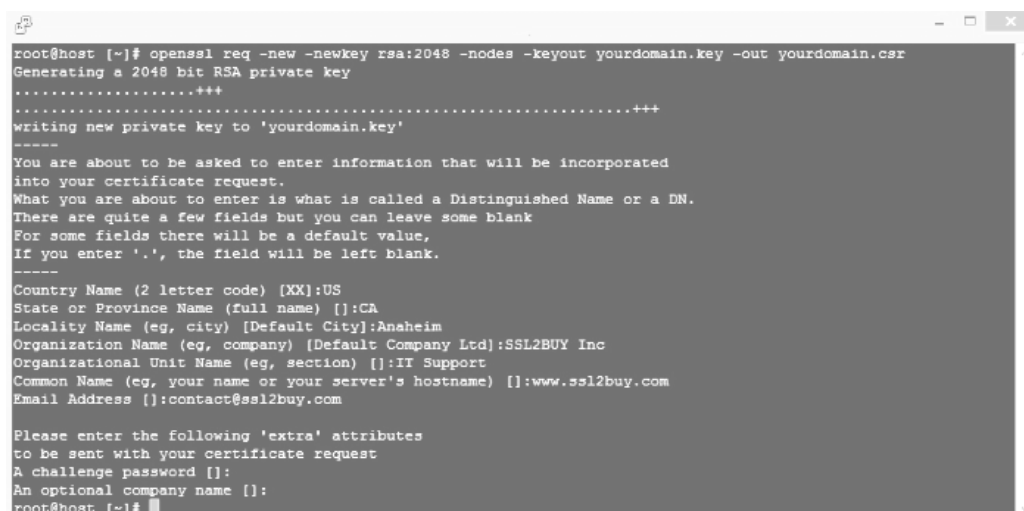
GitHub es una herramienta de gran utilidad para este proyecto ya que además de manejar el control de versiones, también mantiene una copia de respaldo en la nube y puede restituir el proyecto de inmediato en caso de cualquier contrariedad.

1.8. OpenSSL

Es un conjunto de herramientas robusto para proveer todas las funciones de los protocolos de seguridad de la capa de transporte (TLS) y la capa de sockets seguros (SSL) (OpenSSL, 2018).

Genera archivos de extensión .key .pem y .crt necesarios para agregarlos en un servidor web, ideal para generar certificados auto firmados en implementaciones de prueba dentro de localhost o un entorno LAN.

La figura 24 muestra un ejemplo del comando “-req” para obtener un archivo de extensión .key de un certificado auto firmado.



```
root@host [~]# openssl req -new -newkey rsa:2048 -nodes -keyout yourdomain.key -out yourdomain.csr
Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to 'yourdomain.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:US
State or Province Name (full name) []:CA
Locality Name (eg, city) [Default City]:Anaheim
Organization Name (eg, company) [Default Company Ltd]:SSL2BUY Inc
Organizational Unit Name (eg, section) []:IT Support
Common Name (eg, your name or your server's hostname) []:www.ssl2buy.com
Email Address []:contact@ssl2buy.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
root@host [~]#
```

Figura 24. Creación de certificado auto firmado.

Tomado de: (OpenSSL, 2018)

Características:

- Posee bibliotecas de criptografía para múltiples propósitos de seguridad.
- Se distribución es de licenciamiento libre.
- Entorno de ejecución bajo Windows, Linux y MacOS.
- Capacidad para crear archivos como autoridad certificadora dentro de localhost.
- Amplia comunidad de desarrolladores y documentación extensa.
- Agrega HTTPS a un servicio web debidamente configurado.

Gracias al mejoramiento de las seguridades cada vez más estrictas que usa Google Chrome en su versión más actual, surge la necesidad de implementar certificados para el servidor.

Una página web no está autorizada de acceder a la ubicación o cámara de un dispositivo, mientras no cuente con un certificado SSL. Dadas las características de este proyecto, Open SSL es la herramienta más apropiada para generar archivos de seguridad e incorporarlos a la configuración del servidor Sails, ya que el módulo de lecturas solicita habilitar el GPS y la cámara principal. De otro modo la aplicación web progresiva no se ejecutaría apropiadamente al no contar con el protocolo HTTPS.

1.9. Ngrook

Es un es una herramienta que expone servidores locales por medio de la “Transición de Direcciones de Red” (NAT) y detrás de firewalls a la internet pública a través de túneles seguros. (Ngrok, 2018). La figura 25, muestra un sencillo esquema del funcionamiento de Ngrok.

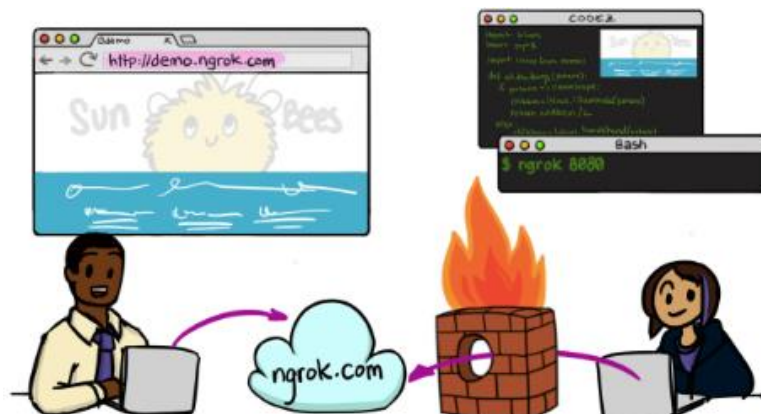


Figura 25. Esquema de funcionamiento Ngrok.

Tomado de: (Ngrok, 2018)

La aplicación se descarga y se ejecuta. Se debe proporcionar un puerto de servicio de red y un protocolo, para el caso del servidor web Sails de este proyecto, corresponde el protocolo HTTP y puerto 1337, tal como lo muestra la figura 26. Ngrok se conecta al servicio en la nube y acepta el tráfico para exponerlo en una dirección pública que se ejecuta desde la dirección local que se especificó.

```

C:\Windows\system32\cmd.exe - ngrok http 1337
ngrok by @inconshreveable

Session Status      online
Session Expires    7 hours, 59 minutes
Version             2.2.8
Region              United States (us)
Web Interface       http://127.0.0.1:4040
Forwarding           http://9a0f69e1.ngrok.io -> localhost:1337
                    https://9a0f69e1.ngrok.io -> localhost:1337

Connections
  ttl   opn   rt1   rt5   p50   p90
   1     5    0.02  0.00  13.28 13.28

HTTP Requests
-----
GET /bower_components/angular-aria/angular-aria.js
GET /bower_components/angular/angular.js
GET /bower_components/angular-animate/angular-animate.js
GET /bower_components/jquery/dist/jquery.js
GET /libs/scanqr/instascan.min.js
GET /js/modernizr.custom.js                200 OK
GET /js/classie.js                         200 OK
GET /js/gnmenu.js                          200 OK
GET /libs/leaflet/leaflet.js               200 OK
GET /css/component.css                     200 OK

```

Figura 26. Entorno de ejecución de ngrok.exe.

Posterior a una correcta configuración, la aplicación debería salir a una URL publica, como lo muestra la figura 27, que expone la aplicación web progresiva de este proyecto en “https://9a0f69e1.ngrok.io/...”

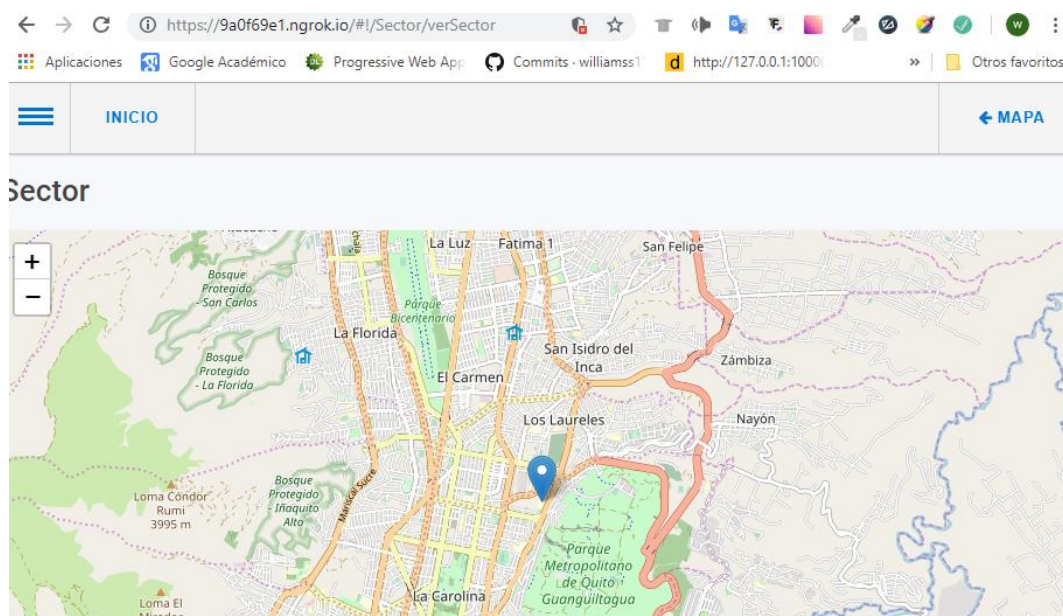


Figura 27. Exposición del servicio local-Back-End a IP pública.

Características:

- Túneles seguros: Crea instantáneamente una URL HTTPS pública para un sitio web que se ejecuta localmente en una máquina de desarrollo.
- Inspección: presenta una interfaz de inspección para verificar el tráfico de entrada y salida de datos.
- Soporte web sockets: comparte aplicaciones en tiempo real con websockets en tunes http sin ningún cambio en el aplicativo
- Contraseña: permite la configuración de credenciales de autenticación para proteger el túnel y las personas con quien se comparte el enlace
- Túneles TCP: se puede exponer cualquier servicio a internet incluso los que no usa HTTP.

La figura 28 muestra la página de control de tráfico de datos en entrada y salida, y se puede acceder a ella a través de localhost:4040.

The screenshot shows the Ngrok web traffic control interface. At the top, it indicates 'ngrok online' and 'Inspect' status. A yellow banner at the top states: 'You are using ngrok without an account. Your session will end in 7 hours, 58 minutes. Sign up for longer sessions.' Below this, there's a section for 'All Requests' with a 'Clear' button and a table of requests. The table lists various GET requests to different resources, all with a 200 OK status and response times ranging from 5ms to 16.93ms. To the right, there's a detailed view of a 'GET /' request, showing a '200 OK' status and a response body containing HTML code for a page titled 'GLAP'.

Request	Status	Duration
GET /images/icons/icon-144x144.png	200 OK	8ms
GET /favicon.ico	200 OK	16.99ms
GET /manifest.json	200 OK	10.99ms
GET /views/UploadMenu.html	200 OK	5ms
GET /scripts/controllers/usuarioCtrl.js	200 OK	5ms
GET /scripts/controllers/VerSectorCtrl.js	200 OK	5ms
GET /scripts/controllers/GestionCuentasCtrl.js	200 OK	15ms
GET /fonts/eicoicons/eicoicons.woff	200 OK	16.93ms
GET /fonts/codropsicons/codropsicons.woff	200 OK	14.99ms
GET /scripts/controllers/abonadosCtrl.js	200 OK	13.99ms

```

<!doctype html>
<html ng-app="proyectoMenu">
<head>
<meta charset="UTF-8" />
<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>GLAP</title>

```

Figura 28. Consola de entrada y salida de datos.

Se uso Ngrook para levantar un entorno web con una dirección publica conectado desde el Back-End local, con la cual se pudo probar la aplicación web progresiva desde el dispositivo móvil.

1.10. Arquitectura

1.10.1. Modelo Vista Controlador MVC

MVC es una arquitectura de software que separa los datos de una aplicación, de la interfaz de usuario y del controlador. La arquitectura MVC provee facilidad de mantenimiento debido a que los componentes tienen baja dependencia entre sí, los cambios que se realicen un componente no afectaran a otro (Mozilla, 2018).

La arquitectura MVC puede aumentar el rendimiento de un aplicativo escalándolo. Por ejemplo, a medida que este vaya aumentando su carga, se podría actualizar el hardware sin afectar otros componentes. Uno de los aspectos más valorados de la arquitectura MVC es la reutilización, un modelo puede ser consultado por múltiples vistas.

- Modelo: El modelo representa la forma de los datos y la lógica empresarial. Mantiene los datos de la aplicación. Los objetos del modelo recuperan y almacenan el estado del modelo en una base de datos.
- Vista: Ver es una interfaz de usuario. Ver datos de visualización utilizando el modelo para el usuario y también les permite modificar los datos.
- Controlador: Controlador maneja la solicitud del usuario. Normalmente, el usuario interactúa con la Vista, que a su vez genera una solicitud de URL apropiada, esta solicitud será manejada por un controlador. El controlador representa la vista apropiada con los datos del modelo como respuesta.

La figura 29, muestra el modo de interacción de MVC con el usuario, y el proceso se explica a continuación:

1. El usuario interactúa con la interfaz de usuario
2. El controlador recibe la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega a través de un gestor de eventos.
3. El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario.
4. El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario.
5. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo.
6. El controlador puede dar la orden a la vista para que se actualice. la vista no tiene acceso directo al modelo, dejando que el controlador envíe los datos del modelo a la vista.
7. La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

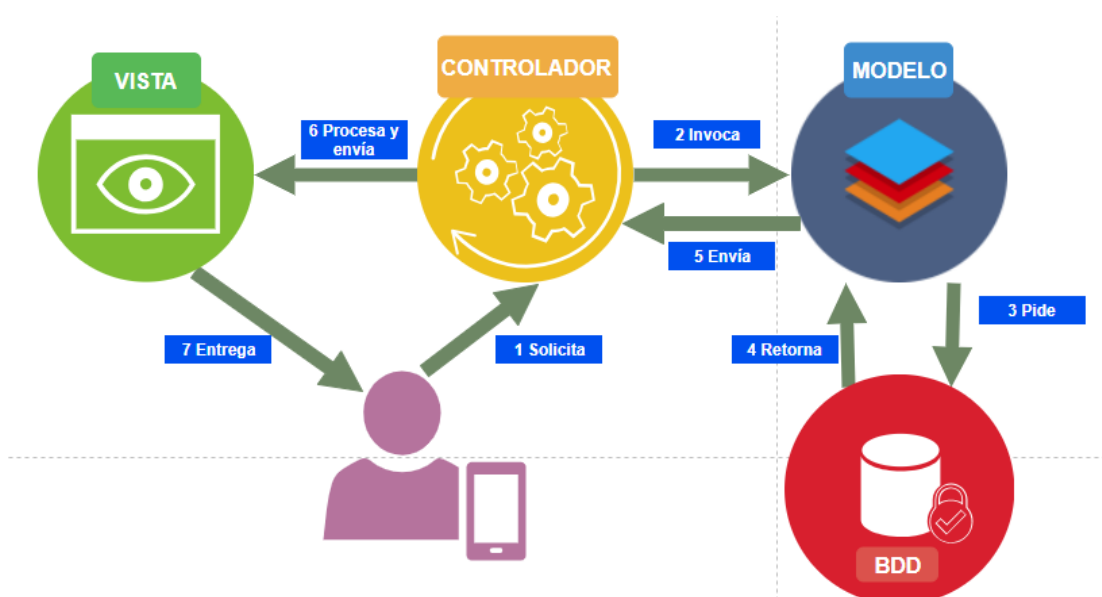


Figura 29. Esquema modelo vista controlador.

1.11. Metodologías de desarrollo

Son todas aquellas técnicas sistemáticas que proponen un modelo de proceso, compuestas por lineamientos o pautas a seguir para conseguir un objetivo. Una metodología de desarrollo es aplicable a casi todos los modelos de producción empresariales, su fin es conseguir la elaboración de productos de calidad en el menor tiempo posible con un gasto de recursos mínimo.

1.11.1. Metodologías de desarrollo ágil

En desarrollo de software, “las metodologías ágiles permiten adaptar la forma de trabajo a las condiciones del proyecto” (Rosselló Villán, 2018), de este modo se consigue flexibilidad e inmediatez. Permite que el proyecto y su desarrollo se adapte a las circunstancias específicas del entorno.

Surgieron como contra propuesta a las metodologías tradicionales, bajo la consolidación del manifiesto ágil (Beck, y otros, 2001), posterior a la reunión de un grupo de expertos de desarrollo de software en Utah en 2001.

La figura 30, muestra las metodologías ágiles más comunes para el desarrollo de software entre ellas se encuentran Scrum, Extreme Programming (XP), RUP, Kanban. Además, el cuadro sugiere una escala de las metodologías con más reglas a seguir respecto de las más adaptativas con menos reglas a seguir.

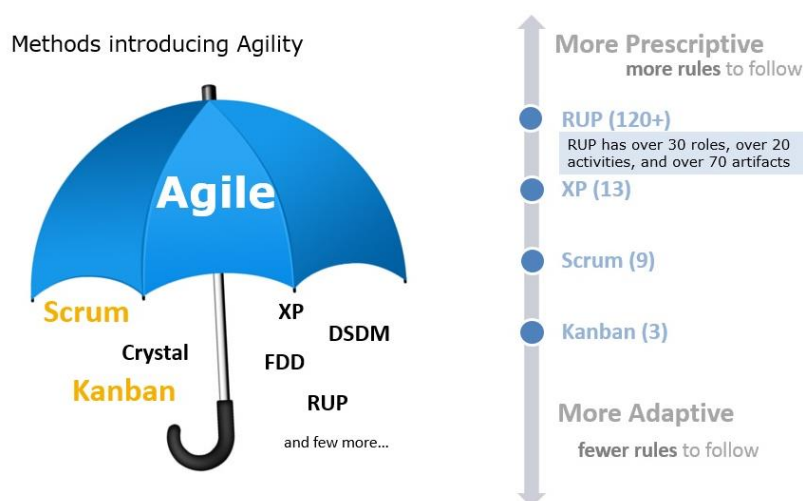


Figura 30. Principales metodologías ágiles.

Tomado de: (Muñoz, 2016)

1.11.1.1. Metodología Kanban

Para comprender el uso de Kanban en los procesos de desarrollo de software, es necesario revisar su uso original.

Fue creado originalmente por Toyota en 1950 donde fue usado para su sistema de producción. Kanban propone el concepto JIT (en inglés Just in Time), “justo a tiempo”, es decir hacer “solo lo que necesitas, cuando lo necesites y en la cantidad que necesites”. La figura 31 muestra el diagrama conceptual del modelo Kanban dividido en dos partes, producción y recuperación de piezas.

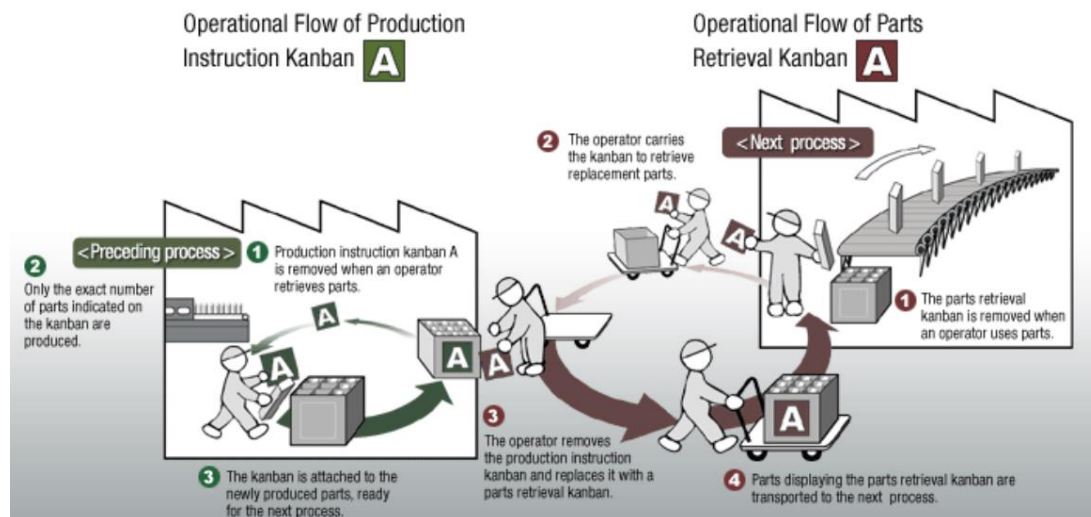


Figura 31. Diagrama conceptual de la metodología Kanban.

Tomado de: (Kirovska & Koceski, 2015)

En ingeniería de software los conceptos principales son similares a los usados para producción y son:

- Calidad perfecta
- Minimizar el desperdicio
- Mejora continua
- Flexibilidad
- Desarrollo y mantenimiento

Kanban en el desarrollo ágil de productos de software representa una forma de visualizar el proyecto presentado en tarjetas. Kanban se enfoca más en el trabajo que se realizará a tiempo en lugar de enfocarse en quién hizo qué.

En Kanban, el trabajo se organiza en tareas o procesos y permite detectar el flujo de trabajo de la manera más eficiente. Las reglas de Kanban son tales que no se necesita considerar los requerimientos del producto de software que no se necesita de inmediato.

No requiere escribir especificaciones, sino solo lo que se puede desarrollar. De esta manera solo se desarrolla lo que se puede probar e implementar.

La forma más común de visualizar el flujo del trabajo es mediante el uso de un tablero con columnas al que se le agrega notas, cada columna en el tablero representa un paso en el flujo y cada nota es una tarea, tal como lo muestra la figura 32 en el tablero Kanban de VivifyScrum usado en este proyecto.

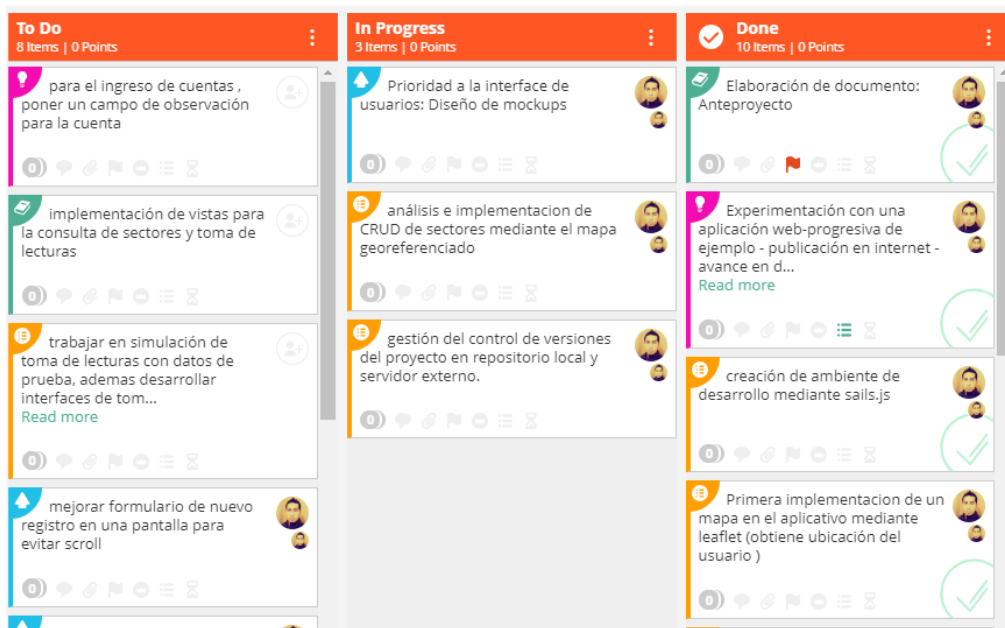


Figura 32. Organización de tareas en Tablero Kanban de la herramienta VivifyScrum.

Al crear un modelo visual, se puede limitar el trabajo en curso no terminado, podemos reducir el tiempo que necesita la tarea para moverse a través del sistema Kanban.

Esto puede evitar problemas como el cambio de tareas y reducir la priorización de tareas. Al utilizar los límites de trabajo en progreso, podemos optimizar el tablero Kanban para mejorar el flujo de trabajo sin problemas, recopilar métricas para analizar el flujo e incluso obtener indicadores de problemas futuros mediante el análisis del flujo de trabajo.

En conclusión, se optó por la metodología Kanban por ser uno de los métodos ágiles que podemos implementar con éxito, agrega valor al producto, flexibilidad, y permite ver una vista general del avance del proyecto. Es una metodología que no tiene problemas para cambiar los requisitos del cliente, la tecnología o los desarrolladores.

Kanban es idóneo para este proyecto porque es altamente flexible, pero al mismo tiempo hay reglas claramente definidas que regulan el proceso en especial la clasificación de tareas. La entrega del producto de software con menos errores posibles.

1.11.1.2. Diseño centrado en el usuario

Es un método ágil de desarrollo de software interactivo centrado en el usuario. El elemento principal de este enfoque son las interfaces de usuario (Pérez Medina & Vanderdonckt, 2018). Es iterativa en tempranas etapas de desarrollo donde se tiene la necesidad de capturar o refinar requerimientos.

El diseño centrado en el usuario propone la creación de interfaces de manera iterativa a través de prototipos o bocetos con múltiples superficies de interacción. El diseño centrado en el usuario pretende crear interfaces rápidas, flexibles e intuitivas para cualquier dispositivo.

A medida que los diseñadores avanzan en la construcción de interfaces se incorporan elementos de usabilidad y ergonomía de las interfaces.

El diseño centrado en el usuario permite la recopilación de requisitos de usuario de manera más objetiva. Para creación de prototipos existen varias técnicas y herramientas software, pero lo más común en la fase inicial del proyecto es la utilización de una pizarra. Ya que el boceto no requiere de ninguna habilidad de modelado avanzada, puede ser utilizada por cualquier parte interesada y los usuarios finales pueden ser partícipes del diseño hasta dar su aprobación.

Con el diseño anticipado de interfaces los desarrolladores pueden detectar posibles problemas de usabilidad antes de que se produzca la primera línea de código.

Se espera que una herramienta de prototipado provea la capacidad de trabajo colaborativo, tal como lo hace la herramienta Marvel utilizada en este proyecto.

El diseño centrado en el usuario permite el desarrollo de software de manera colaborativa y propone 4 configuraciones posibles. El esquema de la figura 33, propone 4 configuraciones posibles, el primero considera un dispositivo y un usuario. El segundo, un dispositivo con múltiples usuarios. El tercero, varios dispositivos con un usuario. El cuarto, varios dispositivos con varios usuarios.

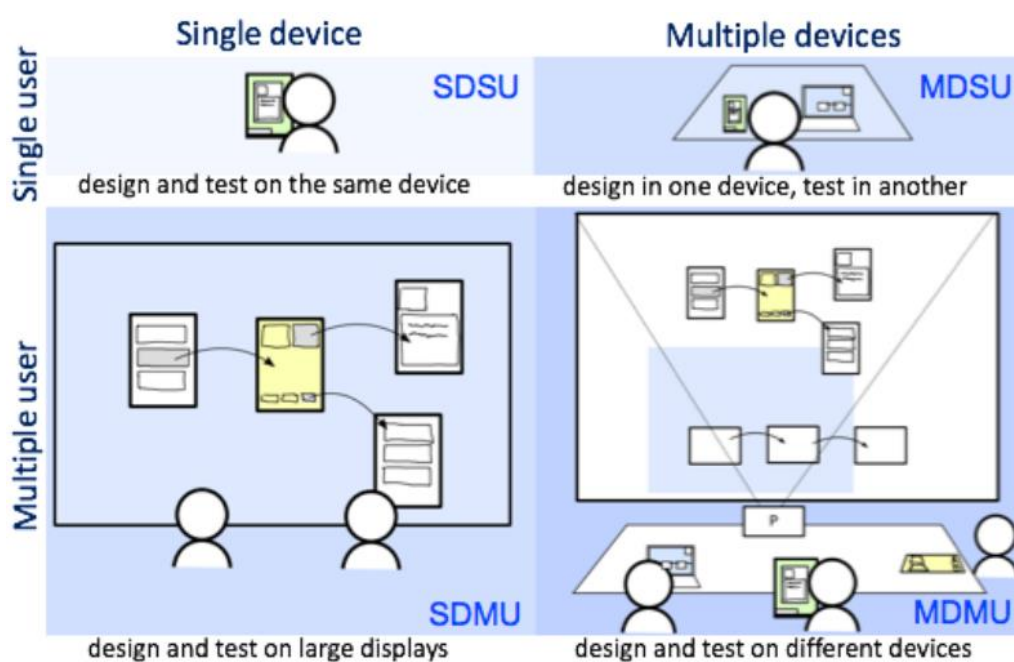


Figura 33. Configuraciones físicas del diseño centrado en el usuario.

Tomado de: (Pérez Medina & Vanderdonckt, 2018).

El flujo de trabajo de la figura 34, presenta 4 ejes principales sucesivos e iterativos en el cual siempre participan el diseñador y usuario en reuniones conjuntas.

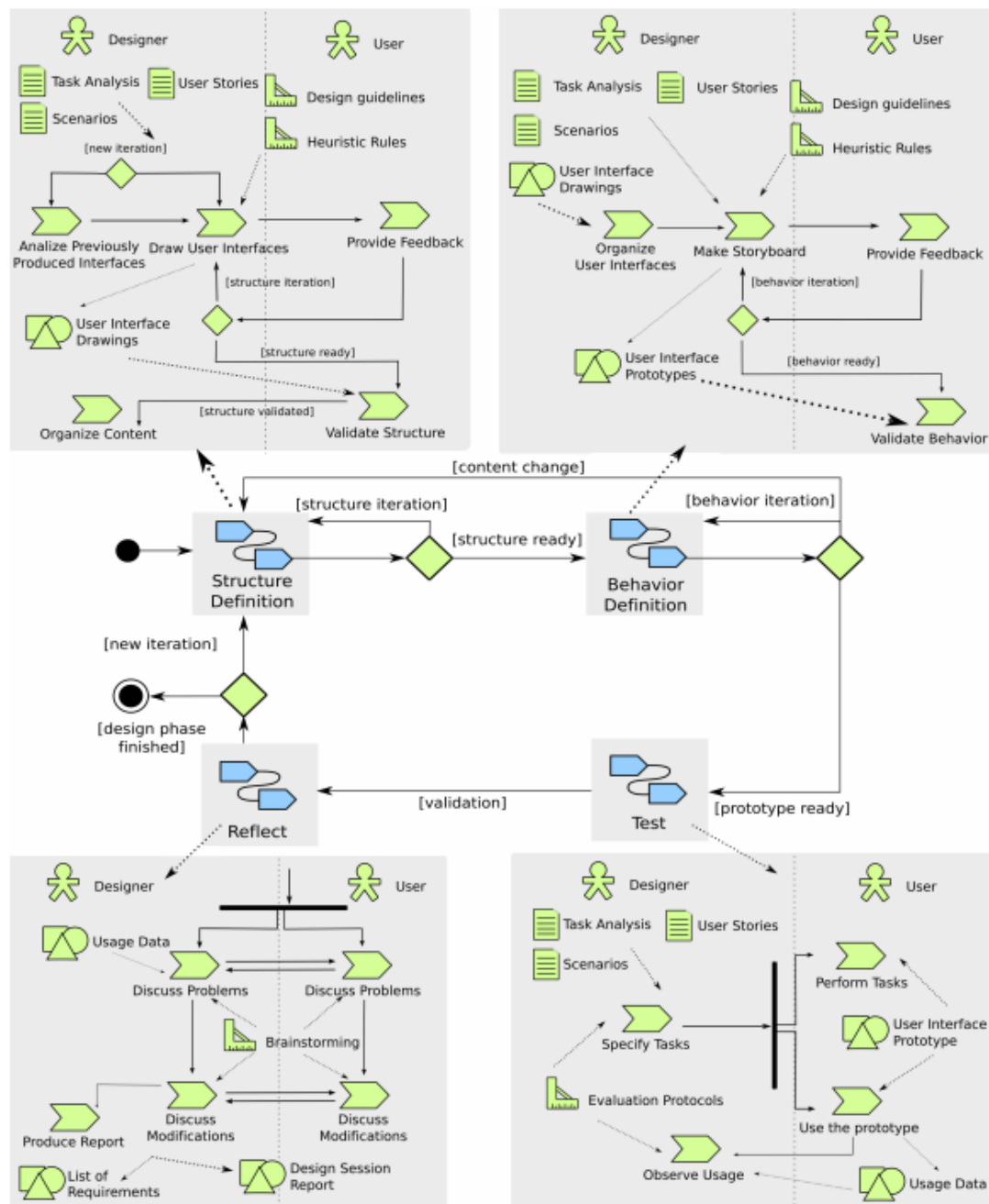


Figura 34. Flujo de trabajo de prototipos de interfaces.

Tomado de: (Pérez Medina & Vanderdonckt, 2018)

Proceso iterativo:

- Definición de la estructura: Mediante el análisis de requerimientos, el diseñador propone un boceto de la interfaz como punto de partida.
- El usuario puede proponer ideas o ajustes de la primera impresión del boceto. El diseñador realiza los ajustes hasta que el usuario valide el diseño.
- Definición de comportamiento: El diseñador organiza las interfaces para generar un tablero de historias que se crea en conjunto con las ideas y validación del usuario.
- Pruebas: la tarea del diseñador se centra en especificar tareas, observar el uso y definir protocolos. Mientras que el usuario prueba las interfaces de prototipo y uso de datos.
- Reflexión: el objetivo de esta etapa es generar un reporte de diseño y una lista de requerimientos, como consecuencia de las sesiones entre diseñador y usuario luego de discutir posibles problemas o modificaciones.

El software de diseño juega un papel fundamental en cada etapa, ya que en ella se registran todas las observaciones o detalles que se produjeron y se decidieron durante una reunión.

Se eligió este método porque su característica principal es involucrar al usuario desde el inicio de un proyecto haciéndolo participe del diseño de interfaces. Esto se convierte en una ventaja, el diseñador obtiene requisitos de manera más precisa mitigando ambigüedades cuando el usuario provee su retroalimentación. Además, este método permite evaluar las capacidades del software y se puede alinear con las metodologías ágiles. Finalmente se puede obtener un producto más personalizado con interfaces amigables al usuario, intuitivas, y adaptables a cualquier dispositivo.

En este proyecto se hizo uso de la metodología centrada en el usuario y permitió capturar requerimientos para la construcción de interfaces con acuerdos entre usuario final y el autor del presente trabajo.

1.12. Usabilidad

La norma ISO 9241-11 proporciona un marco para comprender el concepto de usabilidad y aplicarlo a diferentes escenarios donde las personas utilizan varios tipos de sistema (PÉREZ MEDINA, y otros, 2018). La definición oficial de usabilidad ISO 9241-11 (Stewart, 1998) es la medida en que un producto puede ser utilizado por usuarios específicos para lograr objetivos específicos con efectividad, eficiencia y satisfacción dentro de un contexto de uso específico.

La evaluación de la usabilidad puede ser formativa o sumativa según lo explica la figura 35.



Figura 35. Evaluación de la usabilidad.

1.12.1. Cuestionario de usabilidad

Dentro del conjunto de cuestionarios estándar para medir la usabilidad formativa, el estudio que realizó Tullis et al (Tullis & Stetson, 2004). Concluyó que el cuestionario más efectivo para medir la usabilidad formativa es el cuestionario elaborado por IBM, denominado Cuestionario de usabilidad de un sistema informático (CSUQ) (Lewis, 1995).

Las preguntas del cuestionario se clasifican en 4 categorías para evaluar diferentes aspectos:

- Sysuse (utilidad): Corresponde desde la pregunta 1 hasta la pregunta 8. Se refiere a la opinión de los usuarios sobre la facilidad de uso, el aprendizaje, la velocidad de operación, la eficiencia en la realización de tareas y el sentimiento subjetivo.
- Infoqual (calidad de la información): Corresponde desde la pregunta 9 hasta la pregunta 15. Estudia la subjetividad del usuario con respecto a la gestión de los errores del sistema, la claridad de la información.
- Interqual (calidad de la interfaz): Corresponde desde la pregunta 16 hasta la pregunta 18. Mide el componente afectivo de la actitud del usuario en el uso del sistema, es un componente importante para medir la interacción
- Overall (satisfacción global): Corresponde a la pregunta 19 y mide la satisfacción global del usuario al usar un sistema.

El cuestionario permite medir la satisfacción de los usuarios con la usabilidad de los sistemas informáticos. Las respuestas proporcionadas por los participantes a este cuestionario demuestran una alta correlación con la facilidad de uso del sistema que se está evaluando. El cuestionario CSUQ, consta de 19 preguntas y se responden en una escala de 1 a 7 puntos como lo muestra la figura 36.

(1 = muy en desacuerdo, 2 = muy en desacuerdo, 3 = en desacuerdo, 4 = neutral, 5 = de acuerdo, 6 = en gran medida de acuerdo, 7 = muy de acuerdo). Las escalas de calificación de 7 puntos permiten tres niveles de calificación positiva, negativa y neutral.

17. Me gustó usar la interfaz de este sistema *

	1	2	3	4	5	6	7	
No estoy de acuerdo en absoluto	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totalmente de acuerdo

Figura 36. Escala de calificación del cuestionario de usabilidad CSUQ de IBM.

Las preguntas del cuestionario CSUQ de IBM se describen a continuación:

1. En general, estoy satisfecho con lo fácil que es usar este sistema.
2. Es sencillo utilizar este sistema.
3. Puedo efectivamente completar mi trabajo usando este sistema.
4. Soy capaz de completar mi trabajo rápidamente usando este sistema.
5. Soy capaz de completar mi trabajo de manera eficiente utilizando este sistema.
6. Me siento cómodo usando este sistema.
7. Fue fácil aprender a usar este sistema.
8. Creo que me volví productivo rápidamente usando este sistema.
9. El sistema da mensajes de error que claramente me dicen cómo solucionar problemas.
10. Cada vez que me equivoco al usar el sistema, me recupero fácil y rápidamente.
11. La información (mensajes en pantalla y guía u otra documentación) provista con este sistema es clara.
12. Es fácil encontrar la información que necesito.
13. La información proporcionada con el sistema es fácil de entender.
14. La información es efectiva para ayudarme a completar mi trabajo.

15. La organización de la información en las pantallas del sistema es clara.
16. La interfaz de este sistema es agradable.
17. Me gusta usar la interfaz de este sistema.
18. Este sistema tiene todas las funciones y capacidades que espero que tenga
19. En general, estoy satisfecho con este sistema.

1.12.2. Tarjetas de reacción de productos de Microsoft

Las tarjetas de reacción de productos de Microsoft como parte de un estudio de usabilidad, permite validar los recursos estéticos y medir la respuesta emocional de un cliente frente a un producto software. (Benedek & Miner, 2002). La lista de palabras es extensa y busca abarcar una amplia variedad de reacciones posibles que se pueden aplicar a muchos aspectos diferentes de un producto software. Entre ellos se puede evaluar su atractivo visual, contenido y funcionalidad. La lista contiene 118 palabras diferentes que pueden interpretarse como positivas, negativas o neutrales.

Para el estudio de usabilidad de este proyecto solamente se han escogido 45 palabras para implementarlas en una encuesta breve.

De igual forma se han seleccionado 15 palabras positivas, 15 palabras negativas y 15 palabras neutras, según la tabla 1:

Tabla 1.

Selección de palabras para tarjetas de reacción.

Avanzado	Claro	Contradictorio	Eficaz
Ambiguo	Complejo	Conveniente	Eficiente
Atractivo	Exhaustivo	Creativo	Fácil
Aburrido	Confuso	Difícil	Rápido
Limpio	Consistente	Fácil de usar	Flexible
Poderoso	Pertinente	Rígido	Sencillo
Sofisticado	Comprensible	Ahorrar tiempo	Demasiado técnico
Usable	Útil	Vago	Inconsistente
Intuitivo	Irrelevante	Motivador	Lento

Simpático	Frustrante	Difícil de usar	Inadecuado
Incomprensible	Simplista	No atractivo	Ineficaz
Pérdida de tiempo			

1.13. Conclusión de capítulo

En el capítulo se mencionaron y describieron las herramientas, metodologías y frameworks que se usaron en este proyecto, su elección se dio porque dichas herramientas son compatibles con las demás tecnologías y se alinean con las características del proyecto, las herramientas propuestas en su mayoría son de licenciamiento libre o cuentan con una versión free, sus ventajas cumplen las expectativas para el propósito del trabajo de titulación.

2. CAPÍTULO II. ANÁLISIS Y DISEÑO

El capítulo de análisis y diseño muestra como el conjunto de herramientas, metodologías y frameworks ensamblan una arquitectura de una aplicación web progresiva. Este capítulo presenta diagramas de arquitectura, componentes y diagrama lógico de la base de datos.

2.1. Aplicaciones web progresivas

Es un conjunto de mejores prácticas para hacer que una aplicación web funcione de manera similar a una aplicación de escritorio o móvil. Una aplicación web progresiva procura dar una experiencia tan uniforme donde el usuario no pueda diferenciar entre una aplicación web progresiva y una aplicación móvil nativa (Google, 2018). Las aplicaciones web progresivas combinan lo mejor de la web y lo mejor de las aplicaciones nativas, los usuarios pueden acceder a ella desde el navegador.

Las aplicaciones web progresivas tienen la capacidad de enviar notificaciones y se cargan rápidamente incluso con redes débiles.

Características:

- **Adaptable:** se adapta a cualquier dispositivo de escritorio, móvil o Tablet.
- **Independiente de la conectividad:** permite trabajar con redes de mala calidad o sin conexión mediante el uso de Service Workers.
- **Segura:** las aplicaciones web progresivas solo soportan el protocolo HTTPS para evitar intromisiones y evitar que el contenido no sea manipulado.
- **Descubrible:** permite que los motores de búsqueda lo encuentren y se lo puede identificar como app gracias al manifiesto W3C.
- **Posibilidad de volver a interactuar:** permite la facilidad de interacción al implementar notificaciones.
- **Instalable:** las aplicaciones web progresivas se pueden instalar a la pantalla del usuario sin la molestia de una tienda de aplicaciones.
- **Vinculable:** las aplicaciones se pueden compartir fácilmente desde una URL.

2.1.1. Shell de la App

El Shell de la app impulsa la interfaz de usuario en la aplicación web progresiva, está compuesta del html, css y javascript mínimos para garantizar un rendimiento estable y confiable. De este modo la primera carga de la aplicación web progresiva debería ser rápida se almacenará en la memoria cache de inmediato, es decir los archivos Shell se descargan una vez mediante la red y luego se almacenan en el dispositivo local. Cada vez que el usuario inicie la

aplicación el recurso se obtendrá de la memoria cache obteniendo resultados de inicio muy rápidos.

La figura 37, muestra como el Shell de la app separa los componentes de la interfaz de usuario con la de los datos.

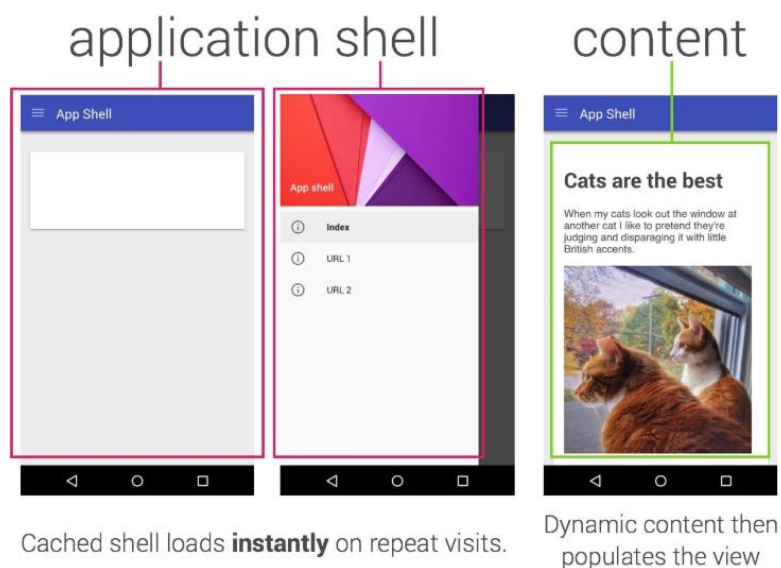


Figura 37. Shell de la aplicación web progresiva.

Tomado de: (Google, 2018)

2.1.2. Service Workers

Los service workers son eventos controlados por eventos que se ejecutan en segundo plano de una aplicación. Actúan como un proxy entre la red y la aplicación.

Pueden interceptar las solicitudes de red y la información de la caché en segundo plano. Esto se puede utilizar para cargar datos para uso sin conexión. Son una secuencia de instrucciones JavaScript que ejecuta eventos como buscar e instalar, y realizan tareas (Google, 2018). La figura 38 contiene una porción del código del service worker implementado en el proyecto.

```
self.addEventListener('fetch', event => {
  //caching for offline viewing
  const {request} = event;
  const url = new URL(request.url);
  if(url.origin === location.origin) {
    event.respondWith(cacheData(request));
  } else {
    event.respondWith(networkFirst(request));
  }
});

async function cacheData(request) {
  const cachedResponse = await caches.match(request);
  return cachedResponse || fetch(request);
}
```

Figura 38. Service worker implementado en el proyecto.

2.1.3. Manifiesto de las aplicaciones web progresivas

Se trata de un archivo de formato .json que permite que las aplicaciones web progresivas puedan controlar el modo de presentación al usuario, dirige lo que el usuario puede ejecutar y como puede hacerlo (Google, 2018).

La figura 39 muestra una porción de código del archivo .json utilizada en este proyecto.

Beneficios del manifiesto para la aplicación web progresiva:

- Ejecuta la aplicación sin la barra URL, es decir usa el modo de pantalla completa.
- Para optimiza la visualización de la pantalla controla su orientación.
- Mejora la experiencia de usuario mediante la ejecución de pantalla de presentación y un tema para el sitio.
- Ejecución de la aplicación de desde la barra URL o desde la pantalla de inicio.

```

54  {
55    "name": "Gestion De Lecturas De Agua Potable ",
56    "short_name": "GLAP",
57    "icons": [{
58      "src": "images/icons/icon-128x128.png",
59      "sizes": "128x128",
60      "type": "image/png"
61    }, {
62      "src": "images/icons/icon-144x144.png",
63      "sizes": "144x144",
64      "type": "image/png"
65    }, {
66      "src": "images/icons/icon-152x152.png",
67      "sizes": "152x152",
68      "type": "image/png"
69    }, {
70      "src": "images/icons/icon-192x192.png",
71      "sizes": "192x192",
72      "type": "image/png"
73    }, {
74      "src": "images/icons/icon-256x256.png",
75      "sizes": "256x256",
76      "type": "image/png"
77    }
78  ],
79  "start_url": "/index.html",
80  "display": "standalone",
81  "background_color": "#3E4EB8",
82  "theme_color": "#2F3BA2"
83  }

```

Figura 39. Manifiesto .json, contiene las características de inicio de la aplicación web progresiva del proyecto.

En conclusión, una aplicación web progresiva es una alternativa para brindar soluciones web y aplicaciones móviles con un solo desarrollo, no requiere de una tienda de aplicaciones para descargarla, en ella se pueden enviar notificaciones y siempre se mantienen actualizada. Es rápida y también puede trabajar en estados de desconexión, esta tecnología es apta para el proyecto propuesto ya que el aplicativo requiere del manejo de dos perfiles de usuario. El primero un usuario administrador quien trabaja con datos de lectores, abonados, sectores y cuentas, desde un dispositivo de escritorio. En el segundo perfil interviene un usuario lector, quien con la aplicación móvil se encargará de acudir a los domicilios de los abonados para tomar las lecturas de los medidores de agua.

2.2. Especificación de los requerimientos

Dado que una de las metodologías en la que se basa este proyecto es la del “diseño centrado en el usuario”, su enfoque obedece al diseño de interfaces con la creación de prototipos (mockups), quienes actúan como un medio de recopilación de requerimientos, se han obtenido juntamente con los usuarios y el autor de este proyecto.



Figura 40. Diseño centrado en el usuario, Refinamiento de interfaces.

En la figura 40, se puede apreciar el proceso iterativo de construcción para la evolución de interfaces. El ciclo refina el modelo y adquiere mayor funcionalidad cada vez, mediante el análisis de las historias de usuario, análisis de tareas y análisis de escenarios.

El primer prototipo puede representarse mediante un boceto dibujado en papel para plasmar ideas iniciales de la interfaz.

Para la construcción del segundo prototipo se ha implementado el uso de mockups con la herramienta Marvel App, el cual permite definir su comportamiento y brinda una perspectiva más clara de lo que se pretende alcanzar como producto.

En base a los prototipos y los requerimientos de usuario el desarrollador ya puede centrarse en la codificación de una primera versión y continuar con el ciclo hasta llegar a una versión definitiva.

En el mismo sentido se definió el comportamiento de la interfaz para obtener prototipos rápidos. El estudio con la participación con los usuarios es fundamental en el proceso, ya que sus recomendaciones permiten agregar nuevas funcionalidades para obtener mejores interfaces.

El esquema de comportamiento que muestra la figura 41, es un ejemplo lo que sucede cuando se da clic en un botón del sistema.

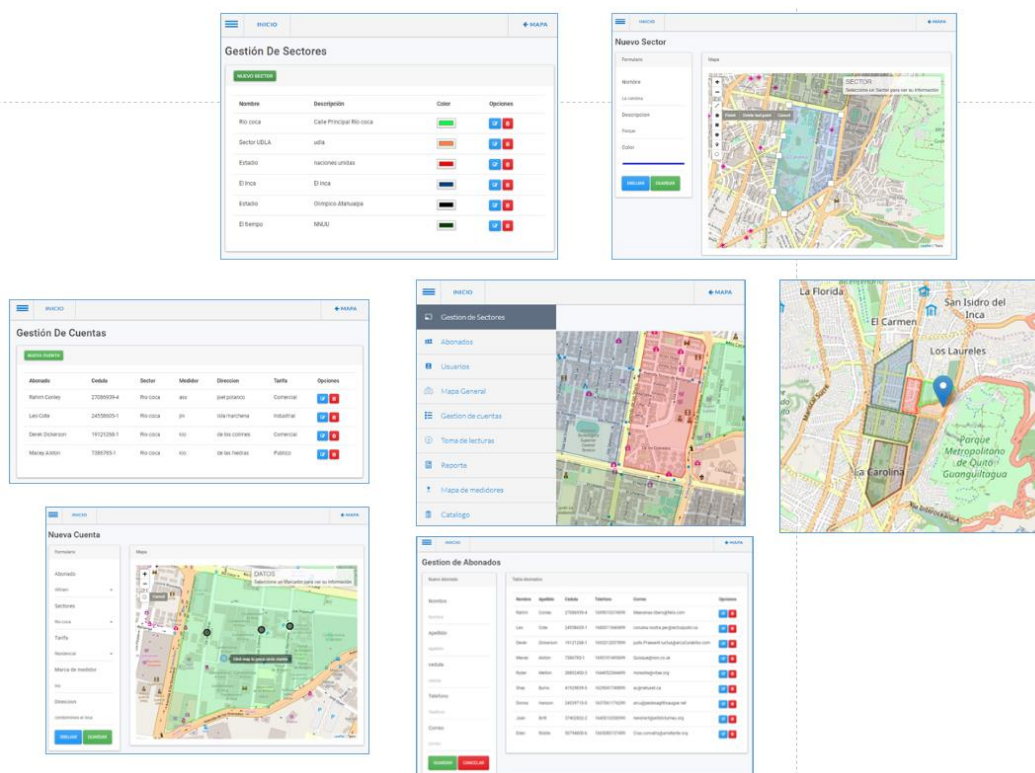


Figura 41. Comportamiento de interfaces.

Al mismo tiempo, también se han registrado las tareas en el tablero Kanban en la herramienta VivifyScrum como lo presenta la figura 42.

Finalmente se listarán los requerimientos mediante historias de usuario.

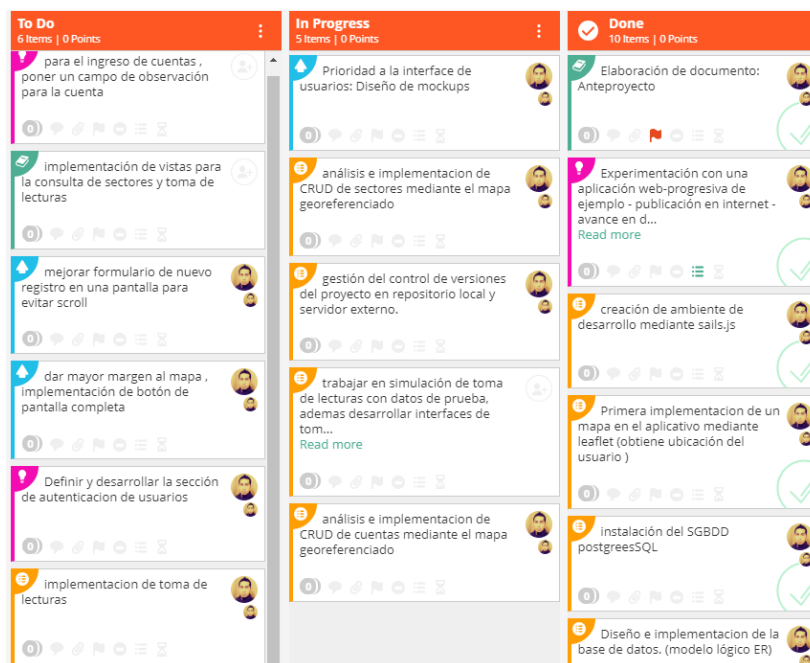


Figura 42. Ajuste de requerimientos en tablero Kanban.

2.2.1. Historias de usuario

Las historias de usuario de este proyecto se han definido con tarjetas del tablero Kanban, a las cuales se les ha asignado un nombre para su identificación, por ejemplo; “Tarjeta: LEC-01”, donde “LEC” hace referencia al usuario lector, y “01” indica el número de requerimiento. De la misma forma, las tarjetas nombradas como “ADM” son referentes al usuario administrador.

También se ha definido el tipo de acceso en “T1” y “T2”. El tipo de acceso “T1” corresponde solamente al usuario lector, quien ingresara al sistema únicamente desde el dispositivo móvil mediante la aplicación web progresiva descargada.

Lo propio sucede en el acceso “T2”, corresponde al usuario administrador e ingresa al sistema desde un navegador web desde un equipo de escritorio (más apropiado para el dibujo de mapas). Las historias de usuario que se describen a continuación presentan varios requerimientos para un prototipo o interfaz en la mayoría de los casos.

2.2.1.1. Gestión de lecturas

Tabla 2.

Tarjeta para el prototipo Login.

Usuario: Lector Acceso: T1
Tarjeta: LEC-01
Descripción: Yo como lector de medidores de agua potable, quiero que el sistema tenga un método de autenticación con usuario y contraseña, Para trabajar con mi perfil de usuario con las funciones habilitadas para realizar lecturas dentro del sector asignado.

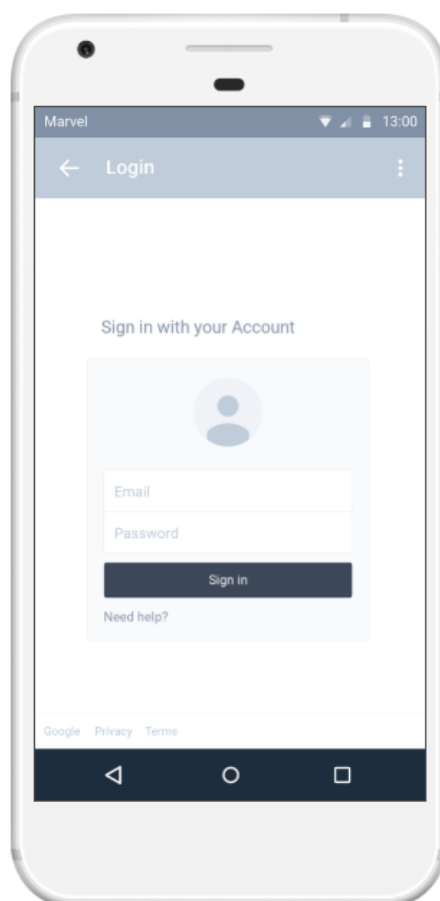


Figura 43. Mockup login de usuario.

Tabla 3.

Tarjeta para el prototipo mapa.

Usuario: Lector Acceso: T1
Tarjeta: LEC-02 Descripción: Yo como lector de medidores de agua potable, quiero obtener la posición del GPS del dispositivo móvil en tiempo real, para ubicarme en base al mapa georreferenciado.
Tarjeta: LEC-03 Descripción: Yo como lector de medidores de agua potable, quiero un mapa que muestre el sector y las cuentas de los abonados, para reconocer la ubicación de los medidores en las casas.
Tarjeta: LEC-04 Descripción: Yo como lector de medidores de agua potable, quiero un medio de identificación de la cuenta en el domicilio del abonado a través de un lector de código QR, para registrar la lectura.



Figura 44. Mockup Mapa.

Tabla 4.

Tarjeta del Prototipo para identificar usuario.

Usuario: lector Acceso: T1
Tarjeta: LEC-05
Descripción: Yo como lector de medidores de agua potable, quiero una pantalla que muestre los datos de la cuenta con datos del abonado, lectura anterior, consumo, teléfono, para registrar la lectura.



Figura 45. Mockup usuario consultado.

Tabla 5.

Tarjeta del prototipo para el ingreso de lectura.

Usuario: Lector	Acceso: T1	Prototipo: Ingreso Lectura
<p>Tarjeta: LEC-06</p> <p>Descripción: Yo como lector de medidores de agua potable, quiero una interfaz con los controles necesarios donde se realicen los cálculos de consumo y valor de factura automáticamente, para ingresar la lectura al sistema.</p>		
<p>Tarjeta: LEC-07</p> <p>Descripción: Yo como lector de medidores de agua potable, quiero editar las lecturas ingresadas, para corregir valores en caso de que se requiera realizar una corrección de lectura.</p>		
<p>Tarjeta: LEC-08</p> <p>Descripción: Yo como lector de medidores de agua potable, quiero describir una observación de lectura, para registrar novedades en el sistema cuando no se haya podido obtener la lectura por distintos motivos.</p>		



Figura 46. Mockup ingreso de lectura.

2.2.1.2. Gestión de sectores

Tabla 6.

Tarjeta para el prototipo de gestión de sectores.

Usuario: Administrador Acceso: T2
Tarjeta: ADM-01 Descripción: Yo como director de planificación de procesos EMAPA, quiero dibujar un polígono en un mapa georreferenciado, además deseo ingresar el nombre del sector y seleccionar un color, para asignarlo a un lector.
Tarjeta: ADM-02 Descripción: Yo como director de planificación de procesos EMAPA, quiero consultar una lista de sectores, para poder realizar operaciones de edición.
Tarjeta: ADM-03 Descripción: Yo como director de planificación de procesos EMAPA, quiero consultar una lista de sectores, para poder borrarlos del sistema.

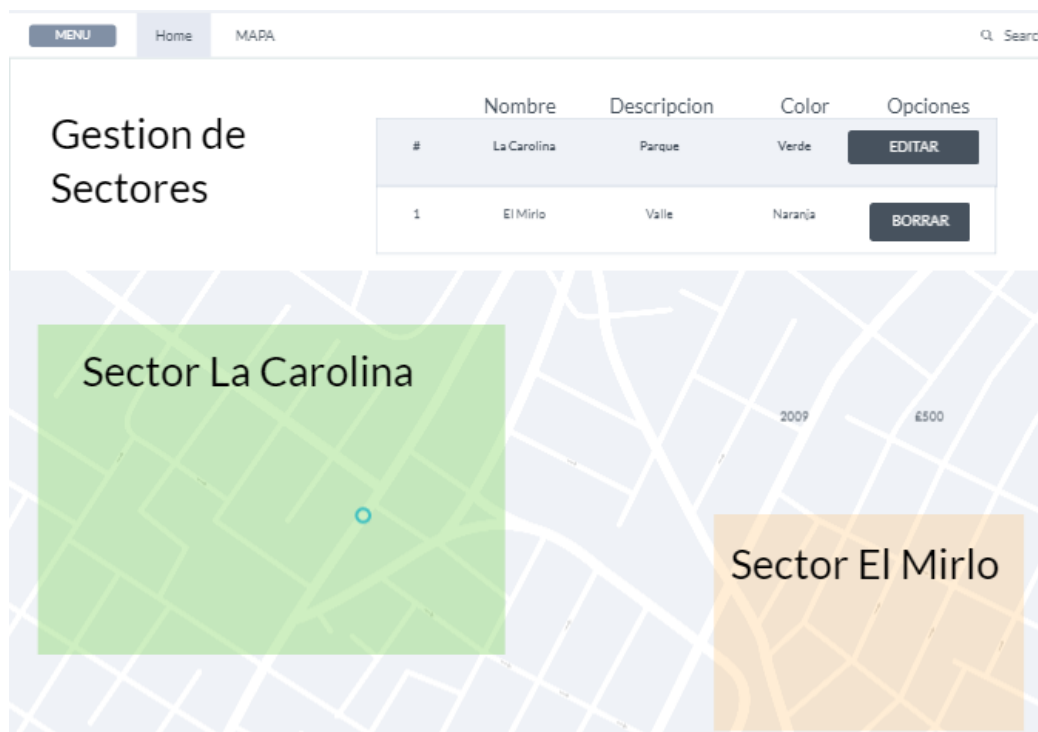


Figura 47. Mockup de gestión de sectores.

2.2.1.3. Gestión de abonados

Tabla 7.

Tarjeta para el prototipo de gestión de abonados.

Usuario: Administrador Acceso: T2
Tarjeta: ADM-04 Descripción: Yo como director de planificación de procesos EMAPA, quiero ingresar todos los datos pertinentes a un abonado, para asignarlo a una cuenta y guardar la información en el sistema.
Tarjeta: ADM-05 Descripción: Yo como director de planificación de procesos EMAPA, quiero , consultar una lista de abonados, para poder realizar operaciones de edición.
Tarjeta: ADM-06 Descripción: Yo como director de planificación de procesos EMAPA, quiero consultar una lista de abonados, para poder borrarlos del sistema.

#	Nombre	Apellido	Cedula	Correo	Opciones
	Alex	Bonilla	173345678-8	cds@gmail.com	EDITAR BORRAR
1	Alex	Bonilla	173345678-8	cds@gmail.com	EDITAR BORRAR
2	Alex	Bonilla	173345678-8	cds@gmail.com	EDITAR BORRAR
3	Alex	Bonilla	173345678-8	cds@gmail.com	EDITAR BORRAR
4	Alex	Bonilla	173345678-8	cds@gmail.com	EDITAR BORRAR
5	Alex	Bonilla	173345678-8	cds@gmail.com	EDITAR BORRAR

Figura 48. Mockup tabla de abonados.

2.2.1.4. Gestión de usuarios

Tabla 8.

Tarjeta para el prototipo de gestión de usuarios.

Usuario: Administrador Acceso: T2
Tarjeta: ADM-07 Descripción: Yo como director de planificación de procesos EMAPA, quiero ingresar todos los datos pertinentes a un usuario tipo lector, para asignarlo dentro de un sector y guardar la información en el sistema.
Tarjeta: ADM-08 Descripción: Yo como director de planificación de procesos EMAPA, quiero , asignar roles de usuario, para manejar perfiles de Lector o administrador en el inicio de sesión.
Tarjeta: ADM-09 Descripción: Yo como director de planificación de procesos EMAPA, quiero consultar una lista de usuarios, para poder realizar operaciones de edición.
Tarjeta: ADM-10 Descripción: Yo como director de planificación de procesos EMAPA, quiero consultar una lista de usuarios, para poder borrarlos del sistema.

#	Nombre	Rol	Cedula	Correo	Opciones
	Ruben Borja	Administrador	173345678-8	cds@gmail.com	EDITAR BORRAR
1	Carlos Yuqui	Lector	173345678-8	cds@gmail.com	EDITAR BORRAR
2	Ruben Borja	Administrador	173345678-8	cds@gmail.com	EDITAR BORRAR
3	Carlos Yuqui	Lector	173345678-8	cds@gmail.com	EDITAR BORRAR
4	Ruben Borja	Administrador	173345678-8	cds@gmail.com	EDITAR BORRAR
5	Carlos Yuqui	Lector	173345678-8	cds@gmail.com	EDITAR BORRAR

Figura 49. Mockup tabla de usuarios.

2.2.1.5. Gestión de cuentas

Tabla 9.

Tarjeta para el prototipo de gestión de cuentas.

Usuario: Administrador Acceso: T2
<p>Tarjeta: ADM-11</p> <p>Descripción: Yo como director de planificación de procesos EMAPA, quiero ingresar los datos de una cuenta, para vincularlos con un abonado existente.</p>
<p>Tarjeta: ADM-12</p> <p>Descripción: Yo como director de planificación de procesos EMAPA, quiero ingresar los datos de una cuenta, para vincularlos con un sector existente.</p>
<p>Tarjeta: ADM-13</p> <p>Descripción: Yo como director de planificación de procesos EMAPA, quiero insertar marcadores georreferenciados dentro de un sector del mapa, para tener una visualización grafica de la ubicación de la cuenta.</p>
<p>Tarjeta: ADM-14</p> <p>Descripción: Yo como director de planificación de procesos EMAPA, quiero consultar una lista de las cuentas, para poder realizar operaciones de edición.</p>
<p>Tarjeta: ADM-15</p> <p>Descripción: Yo como director de planificación de procesos EMAPA, quiero consultar una lista de las cuentas, para poder borrarlas del sistema.</p>

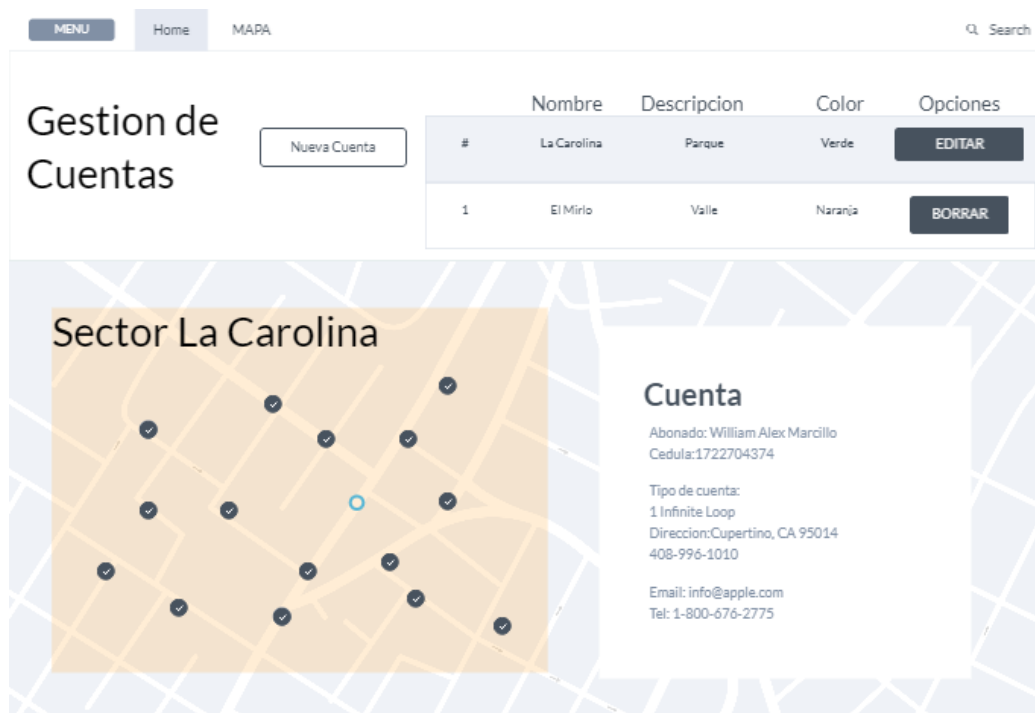


Figura 50. Mockup de gestión de cuentas.

2.3. Diseño de Arquitectura

La arquitectura de este proyecto se diseñó con base al estudio preliminar de la metodología centrada en el usuario donde se pudieron obtener varios requerimientos, según estas necesidades la arquitectura MVC es la solución más viable.

2.3.1. Diagrama lógico de la de Base de Datos

La figura 51, presenta el diagrama lógico de la base de datos. El diagrama cuenta con 5 entidades relacionadas:

- Sector: Es la entidad que rompe la relación de cardinalidad M:M entre las entidades "cuenta" y "usuario". El atributo más importante es "polígono" y su tipo de datos es String para almacenar el conjunto de coordenadas de formato Json que conformaran la figura geométrica en el mapa.

- Usuario: Es la entidad que registra 2 tipos de usuario en el atributo “rol”, un usuario administrador y un usuario lector. Al usuario se le asigna un sector para que recorra una ruta y registre las lecturas.
- Abonado: Un abonado es aquella entidad que registra todos atributos de un cliente para vincularlo a una cuenta.
- Cuenta: Dentro de sus atributos consta las claves foráneas de “sector” y “abonado”, es decir una cuenta pertenece a un sector y a un abonado, sin embargo, un abonado podría tener múltiples cuentas. La entidad cuenta también registra una “coordenada” que se representa geográficamente el mapa con un marcador.
- Lectura: Es la entidad que contendrá datos de la cuenta a la que se realizó la lectura y también los datos del usuario que registro la lectura, también contiene datos de valores de consumo, periodos, y valores a pagar.

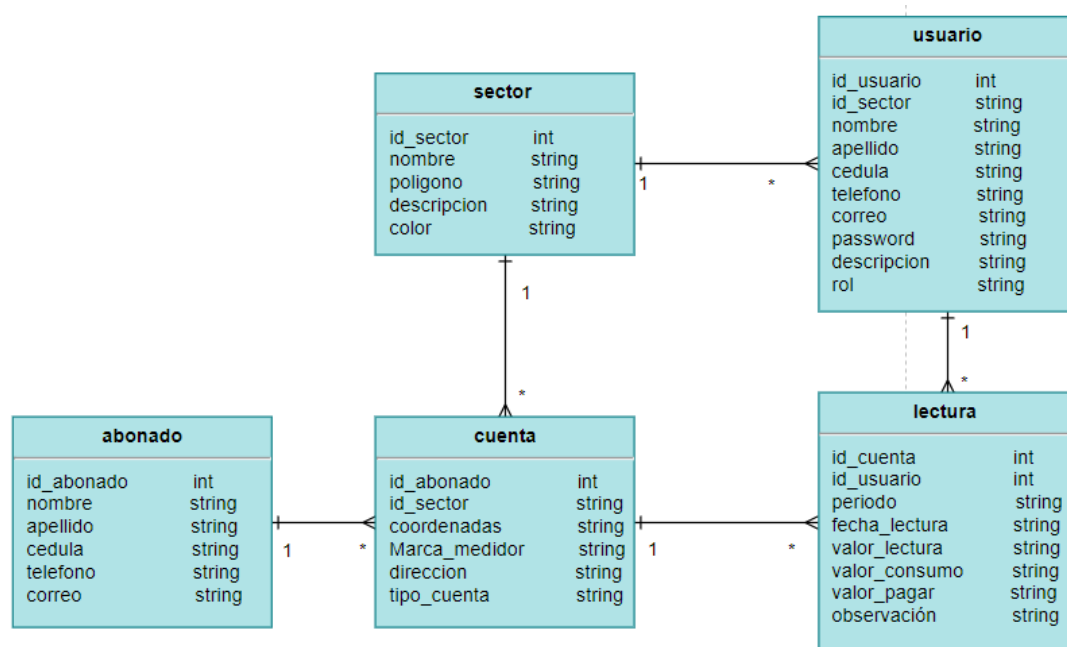


Figura 51. Diagrama lógico de la base de datos.

2.3.2. Diagrama de Arquitectura

La arquitectura que presenta la figura 52, se divide en dos módulos principales el Back-End y Front-End, cada uno integra su tecnología y sus componentes. Esto es transparente para el usuario ya que en conjunto funciona como un solo sistema.

Back-End: Está compuesto principalmente por Sails, quien dentro de su Framework provee una estructura definida para marco de desarrollo MVC. Su rol principal es actuar como servidor para publicar servicios web.

En esta arquitectura el dominio de Sails maneja los controladores quienes serán los encargados de recibir las peticiones del usuario. los controladores también interactúan con los servicios y son los intermediarios del envío de datos entre el modelo y la vista.

Los servicios son una serie de archivos JavaScript encargados de gestionar la conexión a la base de datos PostgreSQL mediante un adaptador.

Los servicios contienen los métodos para realizar las operaciones CRUD y debe crearse un servicio por cada entidad (tabla de la base de datos). Sails permite ejecutar operaciones DDL para definir el modelo sin intervención directa en la base de datos PostgreSQL, según las necesidades del desarrollador se puede redefinir el esquema cada vez, la operación tendrá efecto cuando se levanten los servicios de Sails. Para llevar a cabo la tarea simplemente se modifican los archivos que contienen la estructura del modelo.

Sails es manejado desde Node y controla su entorno, puede crear nuevos proyectos, levanta servicios puede crear entidades en la base de datos.

Font-End: como es común en los sistemas web, el Front-End está compuesto principalmente por Archivos HTML. El Front-End es la interfaz donde el usuario obtiene respuesta de las peticiones que se han realizado en el controlador. El Front-End también está compuesto por archivos CSS para dar diseño y una mejor apariencia.

Bootstrap se indexa a la página principal con el objetivo de desplegar un diseño uniforme a todas las páginas, y se encarga de proveer responsividad.

Dentro del Front-End también han agregado librerías JavaScript como Angular y Leaflet, la primera librería se implementa para usar sus directivas y generar mayor dinamismo a la página. La segunda librería se ha vinculado para conseguir la capa de presentación de los mapas, ofrece funcionalidades de geoposicionamiento, y grafica de figuras geométricas.

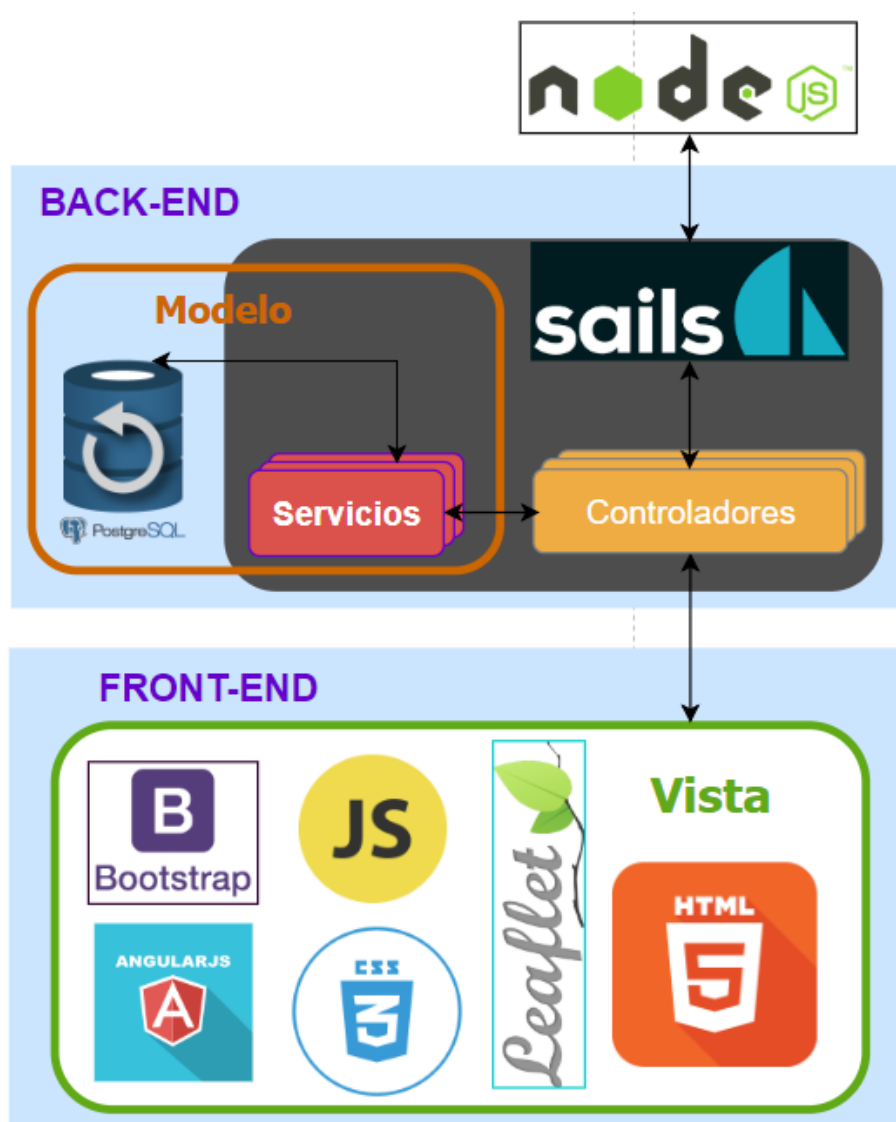


Figura 52. Arquitectura del sistema web progresivo para la toma de lecturas de agua potable.

2.3.3. Diagrama de Componentes

El diagrama de componentes de la figura 53, muestra un sistema cliente-servidor. Los usuarios acceden al servicio web mediante una conexión TCP (orientado a la conexión) con el protocolo HTTP. El servicio web está comprendido por un entorno sails con todos los componentes de modelos, servicios, controladores e interfaces.

Sails provee los medios de acceso hacia la base de datos PostgreSQL mediante una conexión local.

Los service worker, Shell App, manifest.json, son la base para que el sistema pase de ser una página web, a una aplicación web progresiva.

El diagrama de componentes define dos tipos de usuario. El primero es un usuario que accede al módulo de administración desde el navegador web de un dispositivo de escritorio. El segundo usuario accede al módulo de lecturas desde el dispositivo móvil por medio de la aplicación progresiva descargada.

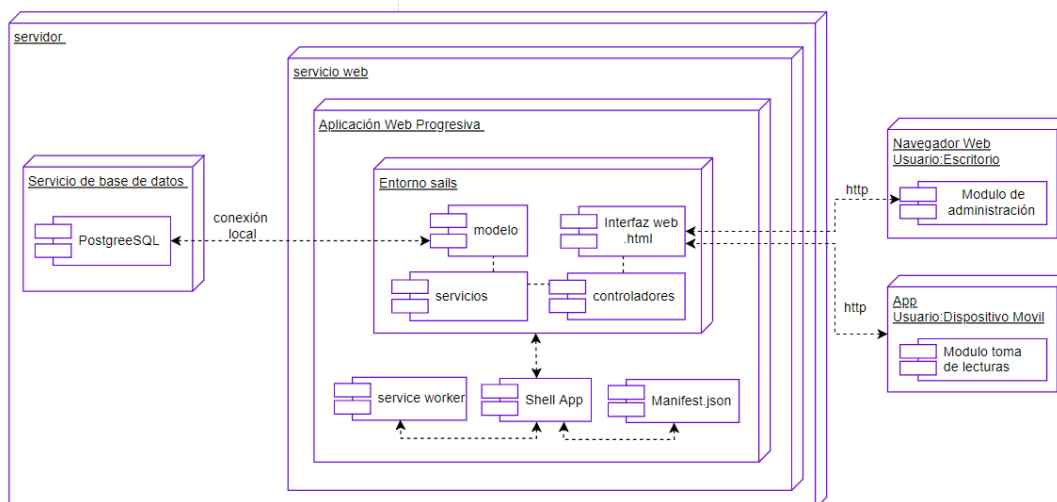


Figura 53. Diagrama de componentes del sistema web progresivo para la toma de lecturas de agua potable.

2.4. Conclusión de Capítulo

En el capítulo de análisis y diseño convergen los requerimientos de usuario realizados con la ayuda de la metodología centrada en el usuario, las herramientas y prácticas, para formar parte de un diseño centrado en el usuario y una arquitectura de software, de manera que sea una propuesta viable como solución para automatizar el proceso de toma de lecturas de medidores de agua potable.

El siguiente capítulo muestra la fase de desarrollo y codificación a partir de la especificación de diseño.

3. CAPÍTULO III. DESARROLLO Y CODIFICACIÓN

Este capítulo presenta la etapa de desarrollo y el despliegue de codificación del proyecto, con el objetivo de resolver las historias de usuario.

El nombre que se le ha atribuido al aplicativo es “GLAP”, que representan las siglas de “Gestión De Lecturas De Agua Potable”.

3.1. Flujo de trabajo

El flujo de trabajo está representado por el movimiento de las tarjetas (tareas) a través del tablero Kanban, son los entregables que alcanzan el estado “Done” (Hecho o Realizado) en un periodo corto de tiempo.

Las tarjetas indican el rol de usuario, su acceso y modulo al que pertenece. Además, muestra los objetos principales del código dentro de la arquitectura MVC con sus funciones principales. Finalmente se presenta las interfaces del sistema y su implementación.

3.1.1. Tarjeta: LEC-01

Tabla 10.

Implementación de historia de usuario LEC-01.

HISTORIA DE USUARIO: LEC-01			
Rol de usuario: Lector	Desarrollador: William Marcillo	Acceso: T1	Módulo: Gestión de lecturas
Descripción			
Yo como lector de medidores de agua potable, quiero que el sistema tenga un método de autenticación con usuario y contraseña, Para trabajar con mi perfil de usuario con las funciones habilitadas para realizar lecturas dentro del sector asignado.			
Elementos del código			
MVC		Funciones Principales	
Servicio	servicioUsuario.js	<ul style="list-style-type: none"> funcionLoginCtrl(); 	
Modelo	usuario		
Vista	Login.html		
Controlador	LoginCtrl.js		
Implementación			
Se debe ingresar usuario y contraseña registrados para acceder a las opciones, el sistema cargara el perfil de usuario previamente registrado.			



Figura 54. Interfaz login usuario GLAP web y móvil.

3.1.2. Tarjeta: LEC-02

Tabla 11.

Implementación historia de usuario LEC-02.

HISTORIA DE USUARIO: LEC-02			
Rol de usuario: Lector	Desarrollador: William Marcillo	Acceso: T1	Módulo: Gestión de lecturas
Descripción			
Yo como lector de medidores de agua potable, quiero obtener la posición del GPS del dispositivo móvil en tiempo real, para ubicarme en base al mapa georreferenciado.			
Elementos del código			
MVC		Funciones Principales	
Servicio	servicioLectura.js	<ul style="list-style-type: none"> • ObtenerSector(); • initMap(); • getPosition(); 	
Modelo	Lectura		
Vista	InicioLectura.html		
Controlador	InicioLecturaCtrl.js		
Implementación			
El sistema solicitará al usuario permisos para acceder a su ubicación mediante la habilitación del GPS, el usuario deberá conceder el permiso y podrá ver su ubicación mediante un marcador azul en el mapa.			

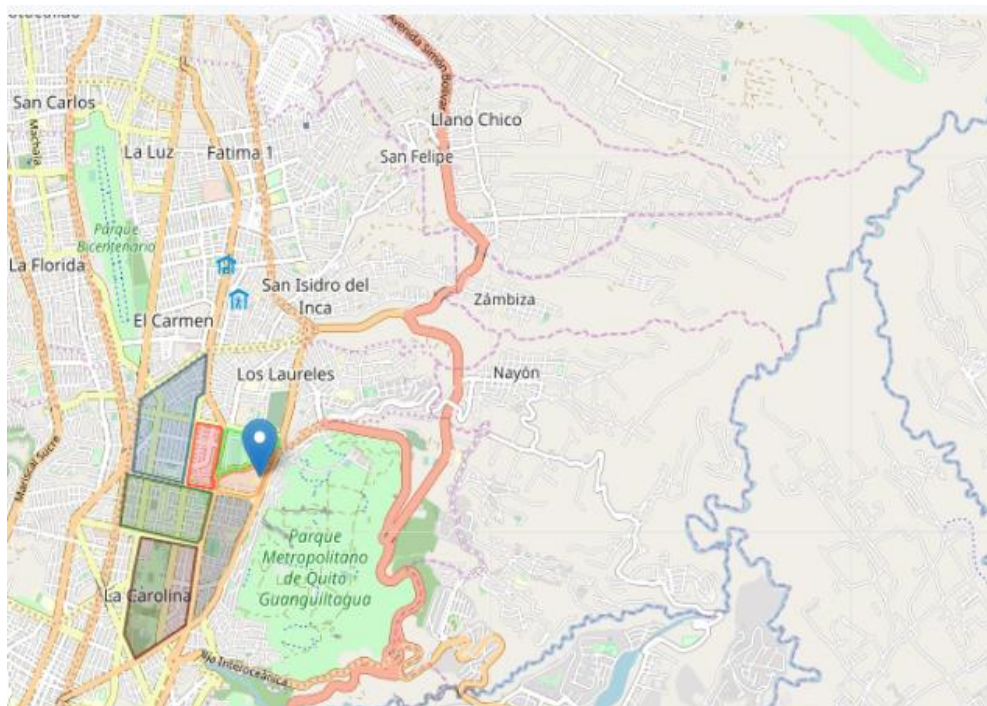


Figura 55. Interfaz mapa general de sectores.

3.1.3. Tarjeta: LEC-03

Tabla 13.

Implementación historia de usuario LEC-03.

HISTORIA DE USUARIO: LEC-03			
Rol de usuario: Lector	Desarrollador: William Marcillo	Acceso: T1	Módulo: Gestión de lecturas
Descripción			
Yo como lector de medidores de agua potable, quiero un mapa que muestre el sector y las cuentas de los abonados, para reconocer la ubicación de los medidores en las casas.			
Elementos del código			
MVC		Funciones Principales	
Servicio	servicioLectura.js	<ul style="list-style-type: none"> • ObtenerSector(); • initMap(); • getposition(); • buscarcuenta(); • getCuentasxSector(); 	
Modelo	Lectura		
Vista	InicioLectura.html		
Controlador	InicioLecturaCtrl.js		
Implementación			
Una vez que el usuario se ha autenticado correctamente, el sistema carga el sector, y los medidores en el mapa representados por marcadores circulares de color azul, naranja y verde. Los colores representan estados del medidor, el color azul significa un medidor sin leer, el color verde significa un medidor leído y el color naranja representa a todos los medidores que registran algún inconveniente de lectura. El usuario solo podrá ver los datos de su perfil y sector asignado.			

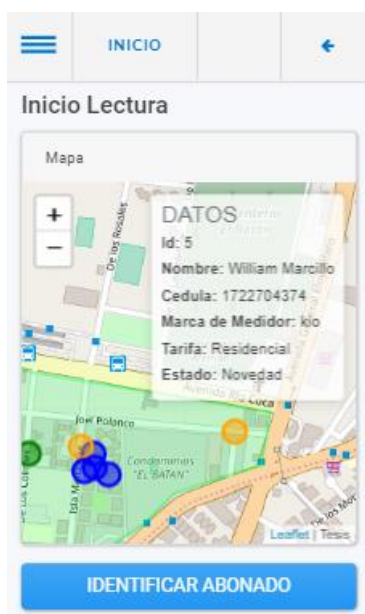


Figura 56. Interfaz de inicio de toma del lecturas.

3.1.4. Tarjeta: LEC-04

Tabla 14.

Implementación de historia de usuario LEC-04.

HISTORIA DE USUARIO: LEC-04			
Rol de usuario: Lector	Desarrollador: William Marcillo	Acceso: T1	Módulo: Gestión de lecturas
Descripción			
Yo como lector de medidores de agua potable, quiero un medio de identificación de la cuenta en el domicilio del abonado a través de un lector de código QR, para registrar la lectura.			
Elementos del código			
MVC		Funciones Principales	
Servicio	servicioLectura.js	<ul style="list-style-type: none"> • habilitarCamara(); • ocultarCamara(); • buscarcuenta(); 	
Modelo	Lectura		
Vista	InicioLectura.html		
Controlador	InicioLecturaCtrl.js		
Implementación			
Clic en el botón "Identificar Abonado". Cuando el sistema habilita la cámara, es necesario acercar el dispositivo al código QR para escanearlo. El sistema identifica al abonado mostrando sus datos en pantalla.			



Figura 57. Scanner QR para identificar un usuario.

3.1.5. Tarjeta: LEC-05

Tabla 15.

Implementación historia de usuario LEC-05.

HISTORIA DE USUARIO: LEC-05			
Rol de usuario: Lector	Desarrollador: William Marcillo	Acceso: T1	Módulo: Gestión de lecturas
Descripción			
Yo como lector de medidores de agua potable, quiero una pantalla que muestre los datos de la cuenta con datos del abonado, lectura anterior, consumo, teléfono, para registrar la lectura.			
Elementos del código			
MVC		Funciones Principales	
Servicio	servicioLectura.js	<ul style="list-style-type: none"> • buscarcuenta(); • getCuentasxSector(); • obtenerLecturaAnterior(); 	
Modelo	Lectura		
Vista	InicioLectura.html		
Controlador	InicioLecturaCtrl.js		
Implementación			
Cuando el sistema ha identificado al abonado después de la lectura QR, inmediatamente se muestra la información de nombres de abonado, lecturas anteriores, consumo, estado y otros datos.			

Scanner QR

Enfoque el código QR para identificar abonado.

Abonado: Ryder Melton

Estado

Leido

Nueva Lectura

21

Consumo

21

valor a pagar

5,25

Observacion

GUARDAR CANCELAR

Figura 58. Interfaz de ingreso de lectura.

3.1.6. Tarjeta: LEC-06

Tabla 16.

Implementación historia de usuario LEC-06.

HISTORIA DE USUARIO: LEC-06			
Rol de usuario: Lector	Desarrollador: William Marcillo	Acceso: T1	Módulo: Gestión de lecturas
Descripción			
Yo como lector de medidores de agua potable, quiero una interfaz con los controles necesarios donde se realicen los cálculos de consumo y valor de factura automáticamente, para ingresar la lectura al sistema.			
Elementos del código			
MVC		Funciones Principales	
Servicio	servicioLectura.js	<ul style="list-style-type: none"> • CalculoLectura(); • GuardarLectura(); • obtenerLecturaAnterior(); 	
Modelo	Lectura		
Vista	InicioLectura.html		
Controlador	InicioLecturaCtrl.js		
Implementación			
El sistema muestra el campo “Nueva lectura”, donde se ingresa un valor entero, mientras se digita o incrementa el valor de la lectura el consumo y valor a pagar se calculan de inmediato en los campos respectivos, cuando se desee confirmar los cambios se dará clic en el botón Guardar. El efecto inmediato de esta acción hará que los marcadores circulares de color azul se pinten a color verde.			

Scanner QR

Enfoque el código QR para identificar abonado.

Abonado: Ryder Melton

Estado

Leido

Nueva Lectura

21

Consumo

21

valor a pagar

5.25

Observacion

GUARDAR CANCELAR

Figura 59. Cálculo automático de consumo y valor a pagar.

3.1.7. Tarjeta: LEC-07

Tabla 17.

Implementación historia de usuario LEC-07.

HISTORIA DE USUARIO: LEC-07			
Rol de usuario: Lector	Desarrollador: William Marcillo	Acceso: T1	Módulo: Gestión de lecturas
Descripción			
Yo como lector de medidores de agua potable, quiero editar las lecturas ingresadas, para corregir valores en caso de que se requiera realizar una corrección de lectura.			
Elementos del código			
MVC		Funciones Principales	
Servicio	servicioLectura.js	<ul style="list-style-type: none"> • CalculoLectura(); • GuardarLectura(); • obtenerLecturaAnterior(); • buscarcuenta(); • obtenerLecturaAnterior(); 	
Modelo	Lectura		
Vista	InicioLectura.html		
Controlador	InicioLecturaCtrl.js		
Implementación			
Para editar o rectificar una lectura, el usuario simplemente tiene que volver a escanear el código QR para identificar el abonado, ingresar el valor de la lectura correcta y guardar los cambios.			

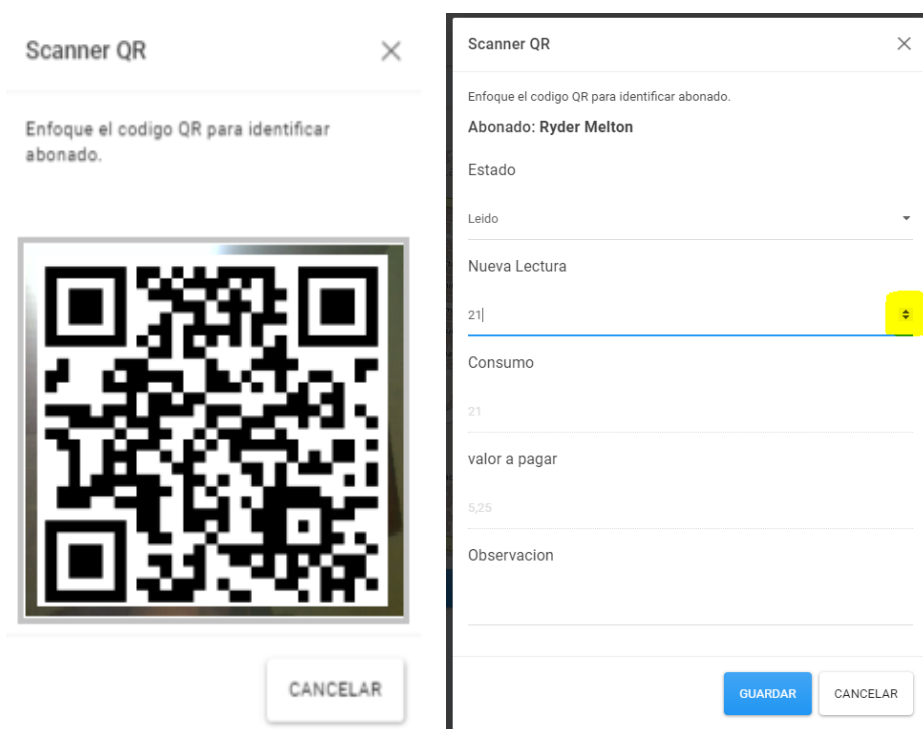


Figura 60. Interfaz de rectificación de lecturas.

3.1.8. Tarjeta: LEC-08

Tabla 18.

Implementación historia de usuario LEC-08.

HISTORIA DE USUARIO: LEC-08			
Rol de usuario: Lector	Desarrollador: William Marcillo	Acceso: T1	Módulo: Gestión de lecturas
Descripción			
Yo como lector de medidores de agua potable, quiero describir una observación de lectura, para registrar novedades en el sistema cuando no se haya podido obtener la lectura por distintos motivos.			
Elementos del código			
MVC		Funciones Principales	
Servicio	servicioLectura.js	<ul style="list-style-type: none"> • CalculoLectura(); • GuardarLectura(); • obtenerLecturaAnterior(); • buscarcuenta(); 	
Modelo	Lectura		
Vista	InicioLectura.html		
Controlador	InicioLecturaCtrl.js		
Implementación			
La interfaz de ingreso de lectura permite cambiar el estado de leído a pendiente y tiene un campo para registrar una observación sobre la lectura. Como efecto los marcadores del mapa que presenten observación se pintaran de color naranja.			

Figura 61. Campo de ingreso de observaciones Implementado.

3.1.9. Tarjeta: ADM-01

Tabla 19.

Implementación historia de usuario ADM-01.

HISTORIA DE USUARIO: ADM-01			
Rol de usuario: Administrador	Desarrollador: William Marcillo	Acceso: T2	Módulo: Gestión de Sectores
Descripción			
Yo como director de planificación de procesos EMAP, quiero dibujar un polígono en un mapa georreferenciado, además deseo ingresar el nombre del sector y seleccionar un color, para asignarlo a un lector.			
Elementos del código			
MVC		Funciones Principales	
Servicio	ServicioSector.js	<ul style="list-style-type: none"> recuperarSector(); initMap(); restartMap(); saveDraw(); 	
Modelo	sector		
Vista	NuevoSector.html		
Controlador	NuevoSectorCtrl.js		
Implementación			
El Sistema abre una página con un formulario en la parte izquierda y presenta un mapa del lado derecho. Se ingresan los datos del formulario en los campos Nombre y descripción. Se dibujan un sector, para terminar el polígono clic en botón finalizar.			
Cuando el usuario compruebe que los datos son correctos, el paso final es dar clic en el botón guardar para que el registro se guarde en la base de datos, finalmente recibirá un mensaje que confirma un ingreso exitoso.			

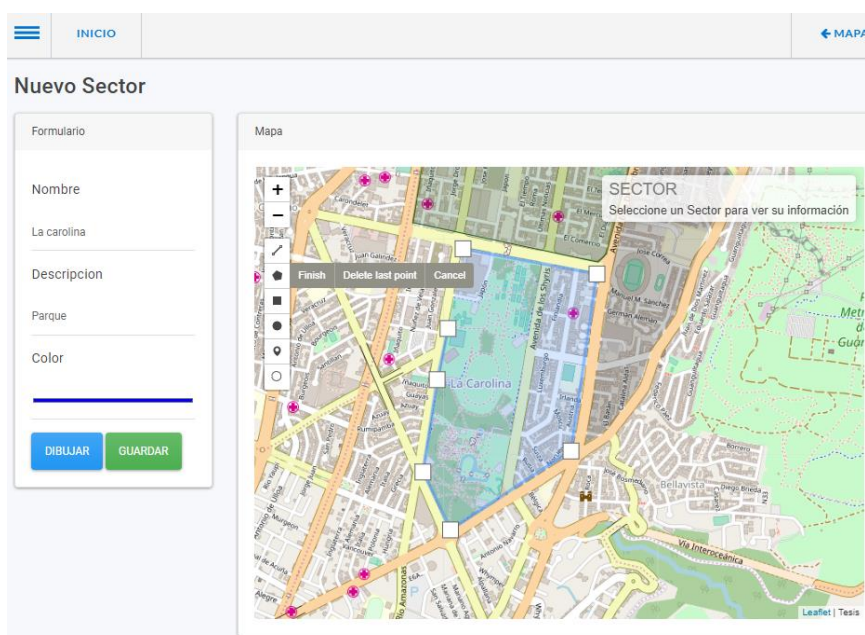


Figura 62. Interfaz de ingreso de nuevo sector.

3.1.10. Tarjeta: ADM-02

Tabla 20.

Implementación historia de usuario ADM-02.

HISTORIA DE USUARIO: ADM-02			
Rol de usuario: Administrador	Desarrollador: William Marcillo	Acceso: T2	Módulo: Gestión de Sectores
Descripción			
Yo como director de planificación de procesos EMAPA, quiero consultar una lista de sectores, para poder realizar operaciones de edición.			
Elementos del código			
MVC		Funciones Principales	
Servicio	ServicioSector.js	<ul style="list-style-type: none"> actualizarSector(); 	
Modelo	sector		
Vista	ActualizarSector.html		
Controlador	ActualizarSector.js		
Implementación			
La actualización del sector permite redefinir el área del polígono, para realizarlo es necesario dar clic en el botón editar del control del mapa. De inmediato aparecerán unos cuadros de color blanco en el margen del mapa. Arrastrar los cuadros para establecer el borde de una nueva área. Luego se dará clic en el botón guardar registrar los cambios en la base de datos.			

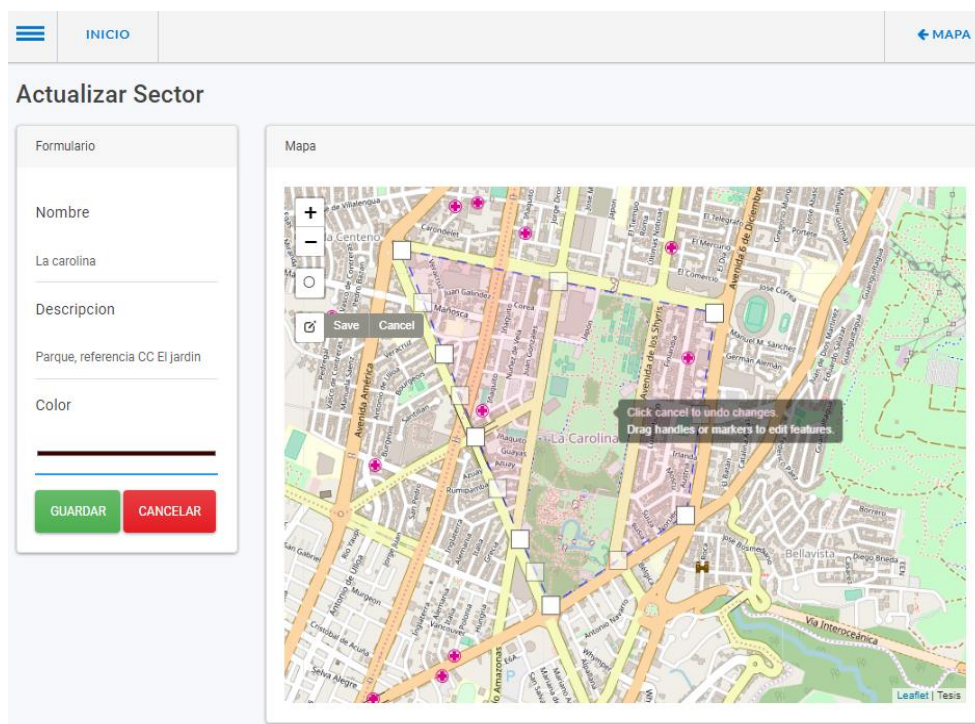


Figura 63. Interfaz de actualización de sector.

3.1.11. Tarjeta: ADM-03

Tabla 21.

Implementación historia de usuario ADM-03.

HISTORIA DE USUARIO: ADM-03			
Rol de usuario: Administrador	Desarrollador: William Marcillo	Acceso: T2	Módulo: Gestión de Sectores
Descripción			
Yo como director de planificación de procesos EMAPA, quiero consultar una lista de sectores, para poder borrarlos del sistema.			
Elementos del código			
MVC		Funciones Principales	
Servicio	ServicioSector.js	<ul style="list-style-type: none"> • obtenerSectores(); • BorrarSector(); 	
Modelo	sector		
Vista	GestionSectores.html		
Controlador	GestionSectores.js		
Implementación			
Para borrar los sectores el sistema muestra una lista con los registros, en la columna de opciones un botón de color rojo inicia la acción de eliminación, luego será necesario aceptar la acción en el mensaje para confirmar.			

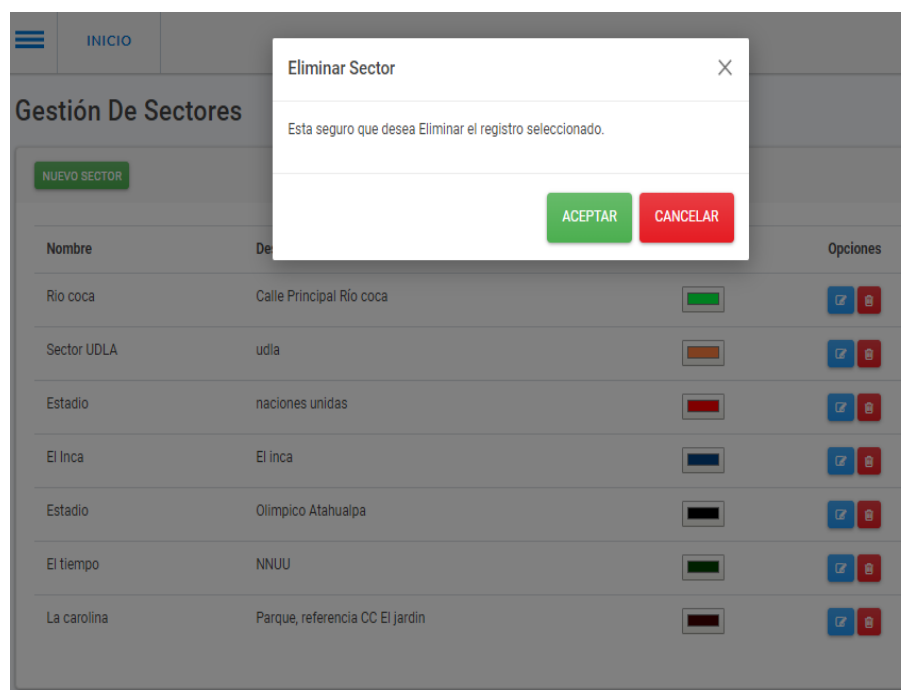


Figura 64. Controles de eliminación de registros implementado.

3.1.12. Tarjeta: ADM-04

Tabla 22.

Implementación historia de usuario ADM-04.

HISTORIA DE USUARIO: ADM-04			
Rol de usuario: Administrador	Desarrollador: William Marcillo	Acceso: T2	Módulo: Gestión abonados
Descripción			
Yo como director de planificación de procesos EMAPA, quiero ingresar todos los datos pertinentes a un abonado, para asignarlo a una cuenta y guardar la información en el sistema.			
Elementos del código			
MVC		Funciones Principales	
Servicio	servicioabonado.js	<ul style="list-style-type: none"> guardarAbonado(); obtenerAbonado(); 	
Modelo	Abonado		
Vista	ListaAbonados.html		
Controlador	abonadosCtrl.js		
Implementación			
El sistema muestra la interfaz de gestión de abonados, desde aquí Podemos agregar un nuevo abonado llenado los datos que solicita el formulario.			

Gestion de Abonados

Nuevo Abonado

Nombre
Nombre

Apellido
Apellido

cedula
cedula

Telefono
Telefono

Correo
correo

GUARDAR **CANCELAR**

Tabla Abonados

Nombre	Apellido	Cedula	Telefono	Correo	Opciones
Rahim	Conley	27086939-4	1699010319099	Maecenas.libero@felis.com	
Leo	Cote	24558605-1	1680011840499	conubia.nostra.per@lectusjusto.ca	
Derek	Dickerson	19121268-1	1692012057899	justo.Praesent.luctus@arcuCurabitur.com	
Macey	Alston	7386785-1	1695101495699	Quisque@non.co.uk	
Ryder	Meiton	38802400-3	1644052544499	molestie@vitae.org	
Shay	Burks	41929839-5	1629041749899	ac@netuset.ca	
Donna	Hanson	24539710-0	1637061176299	arcu@pedesagittisauque.net	
Joan	Britt	37402802-2	1645010358599	hendrerit@elitdictumeu.org	
Eden	Riddle	50794800-6	1665080157499	Cras.convallis@ametante.org	

Figura 65. Tabla de gestión de abonados.

3.1.13. Tarjeta: ADM-05

Tabla 22.

Implementación historia de usuario ADM-05.

HISTORIA DE USUARIO: ADM-05			
Rol de usuario: Administrador	Desarrollador: William Marcillo	Acceso: T2	Módulo: Gestión abonados
Descripción			
Descripción: Yo como director de planificación de procesos EMAPA, quiero, consultar una lista de abonados, para poder realizar operaciones de edición.			
Elementos del código			
MVC		Funciones Principales	
Servicio	servicioabonado.js	<ul style="list-style-type: none"> • actualizarAbonado(); • obtenerAbonado(); 	
Modelo	Abonado		
Vista	ListaAbonados.html		
Controlador	abonadosCtrl.js		
Implementación			
Para la edición el sistema muestra en la interfaz de gestión de abonados un botón de edición para traer los datos al formulario y realizar los cambios en los datos.			

The screenshot shows a web application interface for managing subscribers. At the top, there are navigation links for 'INICIO' and 'MAPA'. The main heading is 'Gestion de Abonados'. On the left, there is a form titled 'Nuevo Abonado' with fields for 'Nombre' (filled with 'William'), 'Apellido' (filled with 'Marcillo'), 'cedula' (filled with '1722704374'), 'Telefono' (filled with '0984848166'), and 'Correo' (filled with 'williamss00@hotmail.com'). Below the form are 'GUARDAR' and 'CANCELAR' buttons. On the right, there is a table titled 'Tabla Abonados' with columns: 'Nombre', 'Apellido', 'Cedula', 'Telefono', 'Correo', and 'Opciones'. The table contains 10 rows of subscriber data, each with a blue edit icon and a red delete icon in the 'Opciones' column.

Nombre	Apellido	Cedula	Telefono	Correo	Opciones
Rahim	Conley	27086939-4	1699010319099	Maecenas.libero@felis.com	[Edit] [Delete]
Leo	Cote	24558605-1	1680011840499	conubia.nostra.per@lectusjusto.ca	[Edit] [Delete]
Derek	Dickerson	19121268-1	1692012057899	justo.Praesent.luctus@arcuCurabitur.com	[Edit] [Delete]
Macey	Alston	7386785-1	1695101495699	Quisque@non.co.uk	[Edit] [Delete]
Ryder	Melton	38802400-3	1644052544499	molestie@vitae.org	[Edit] [Delete]
Shay	Burks	41929839-5	1629041749899	ac@netuset.ca	[Edit] [Delete]
Donna	Hanson	24539710-0	1637061176299	arcu@pedesagittisaugue.net	[Edit] [Delete]
Joan	Britt	37402802-2	1645010358599	hendrerit@elitdictumeu.org	[Edit] [Delete]
Eden	Riddle	50794800-6	1665080157499	Cras.convallis@ametante.org	[Edit] [Delete]

Figura 66. Edición de abonados.

3.1.14. Tarjeta: ADM-06

Tabla 23.

Implementación historia de usuario ADM-06.

HISTORIA DE USUARIO: ADM-06			
Rol de usuario: Administrador	Desarrollador: William Marcillo	Acceso: T2	Módulo: Gestión abonados
Descripción			
Yo como director de planificación de procesos EMAPA, quiero consultar una lista de abonados, para poder borrarlos del sistema.			
Elementos del código			
MVC		Funciones Principales	
Servicio	servicioabonado.js	<ul style="list-style-type: none"> BorrarAbonado(); obtenerAbonado(); 	
Modelo	Abonado		
Vista	ListaAbonados.html		
Controlador	abonadosCtrl.js		
Implementación			
Para borrar un registro, el sistema muestra en la interfaz de gestión de abonados un botón de Eliminar, luego el usuario deberá realizar clic en aceptar en el mensaje de confirmación para completar la acción.			

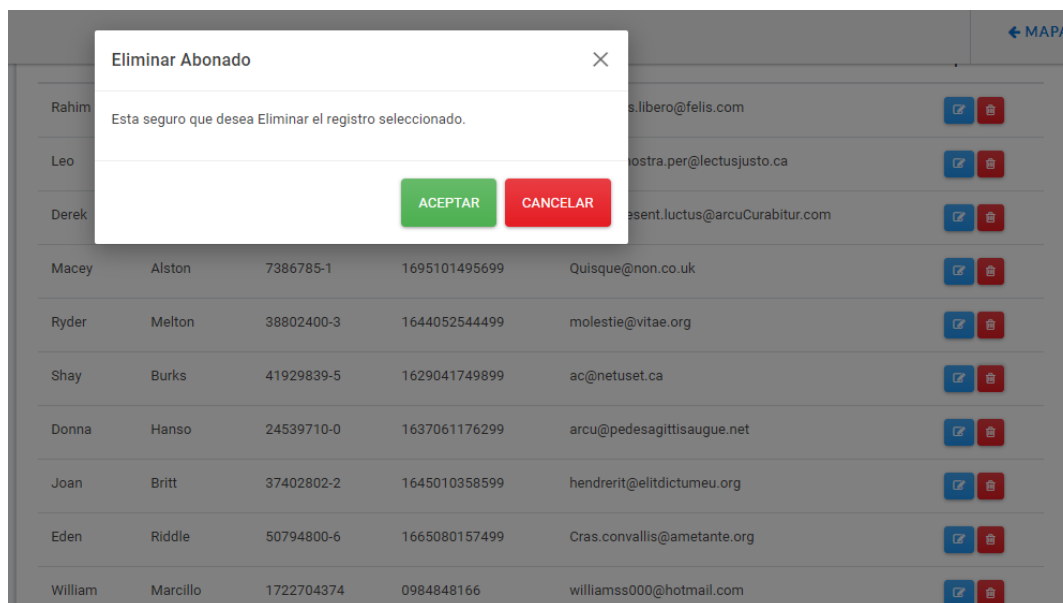


Figura 67. Control de eliminación de registros.

3.1.15. Tarjeta: ADM-07

Tabla 24.

Implementación historia de usuario ADM-07.

HISTORIA DE USUARIO: ADM-07			
Rol de usuario: Administrador	Desarrollador: William Marcillo	Acceso: T2	Módulo: Gestión de Usuarios
Descripción			
Yo como director de planificación de procesos EMAPA, quiero ingresar todos los datos pertinentes a un usuario tipo lector, para asignarlo dentro de un sector y guardar la información en el sistema.			
Elementos del código			
MVC		Funciones Principales	
Servicio	servicioUsuario.js	<ul style="list-style-type: none"> guardarUsuario(); 	
Modelo	Usuario		
Vista	ListaUsuarios.html		
Controlador	usuarioCtrl.js		
Implementación			
El sistema muestra la interfaz de gestión de usuarios, desde aquí Podemos agregar un nuevo usuario llenado los datos que solicita el formulario.			

The screenshot shows a web application interface for user management. At the top, there are navigation links for 'INICIO' and 'MAPA'. The main heading is 'Gestion de Usuarios'. On the left, there is a 'Nuevo Usuario' form with the following fields: Nombre (Alex), Apellido (Loor), Cedula (1755667896), Telefono (0984545633), Correo (Lector), Password (Lector), Descripcion (correspondiente a rio coca), Rol (1), and Sector (a dropdown menu). At the bottom of the form are 'GUARDAR' and 'CANCELAR' buttons. On the right, there is a 'Tabla de Usuarios' table with the following data:



Nombre	Apellido	Cedula	Telefono	Correo	Password	Descripcion	Rol	Sector	Opciones
Alex	Loor	1755667896	0984545633	Lector	Lector	correspondiente a rio coca	1	Rio coca	 

Figura 68. Tabla de gestión de usuarios.

3.1.16. Tarjeta: ADM-08

Tabla 25.

Implementación historia de usuario ADM-08.

HISTORIA DE USUARIO: ADM-08			
Rol de usuario: Administrador	Desarrollador: William Marcillo	Acceso: T2	Módulo: Gestión de Usuarios
Descripción			
Yo como director de planificación de procesos EMAPA, quiero, asignar roles de usuario, para manejar perfiles de Lector o administrador en el inicio de sesión.			
Elementos del código			
MVC		Funciones Principales	
Servicio	servicioUsuario.js	<ul style="list-style-type: none"> guardarUsuario(); 	
Modelo	Usuario		
Vista	ListaUsuarios.html		
Controlador	usuarioCtrl.js		
Implementación			
Dentro del formulario de ingreso, el usuario tiene un control de opciones Rol para seleccionar entre el perfil de administrador o seleccionar un perfil de lector.			

The screenshot shows a web application interface for user management. At the top, there is a navigation bar with a hamburger menu icon and the text 'INICIO'. Below this is a header for 'Gestion de Usuarios'. The main content area features a form titled 'Nuevo Usuario'. The form has the following fields and values: 'Nombre' (Alex), 'Apellido' (Loor), 'cedula' (1755667896), 'Telefono' (0984545633), 'Correo' (Lector), 'Password' (Lector), 'Descripcion' (correspondiente a rio coca), 'Rol' (1), and 'Sector' (a dropdown menu). At the bottom of the form, there are two buttons: a green 'GUARDAR' button and a red 'CANCELAR' button.

Figura 69. Ingreso de nuevo usuario.

3.1.17. Tarjeta: ADM-09

Tabla 26.

Implementación historia de usuario ADM-09.

HISTORIA DE USUARIO: ADM-09			
Rol de usuario: Administrador	Desarrollador: William Marcillo	Acceso: T2	Módulo: Gestión de Usuarios
Descripción			
Yo como director de planificación de procesos EMAPA, quiero consultar una lista de usuarios, para poder realizar operaciones de edición			
Elementos del código			
MVC		Funciones Principales	
Servicio	servicioUsuario.js	<ul style="list-style-type: none"> • actualizarUsuario(); • obtenerUsuarios(); 	
Modelo	Usuario		
Vista	ListaUsuarios.html		
Controlador	usuarioCtrl.js		
Implementación			
Para la edición el sistema muestra en la interfaz de gestión de usuarios mediante un botón de edición para traer los datos al formulario y realizar los cambios en los datos del registro seleccionado.			

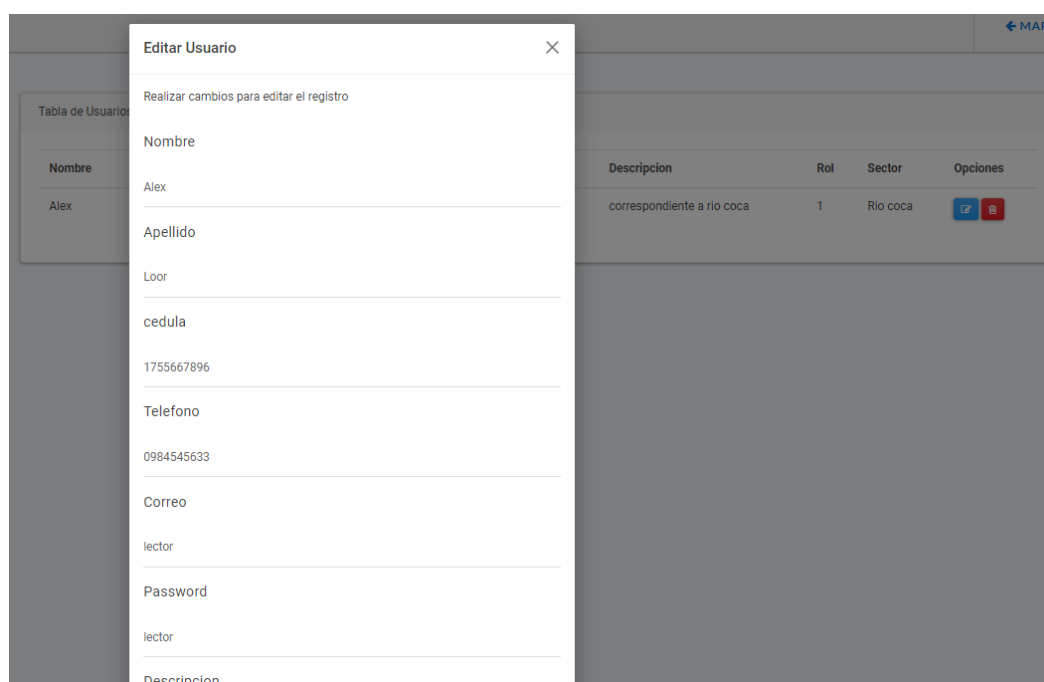


Figura 70. Edición de registros para usuarios.

3.1.18. Tarjeta: ADM-10

Tabla 27.

Implementación historia de usuario ADM-10.

HISTORIA DE USUARIO: ADM-10			
Rol de usuario: Administrador	Desarrollador: William Marcillo	Acceso: T2	Módulo: Gestión de Usuarios
Descripción			
Yo como director de planificación de procesos EMAPA, quiero consultar una lista de usuarios, para poder borrarlos del sistema.			
Elementos del código			
MVC		Funciones Principales	
Servicio	servicioUsuario.js	<ul style="list-style-type: none"> BorrarUsuario(); obtenerUsuarios(); 	
Modelo	Usuario		
Vista	ListaUsuarios.html		
Controlador	usuarioCtrl.js		
Implementación			
Para borrar un registro, el sistema muestra en la interfaz de gestión de usuarios un botón de Eliminar, luego deberá realizar clic en aceptar en el mensaje de confirmación para completar la acción.			

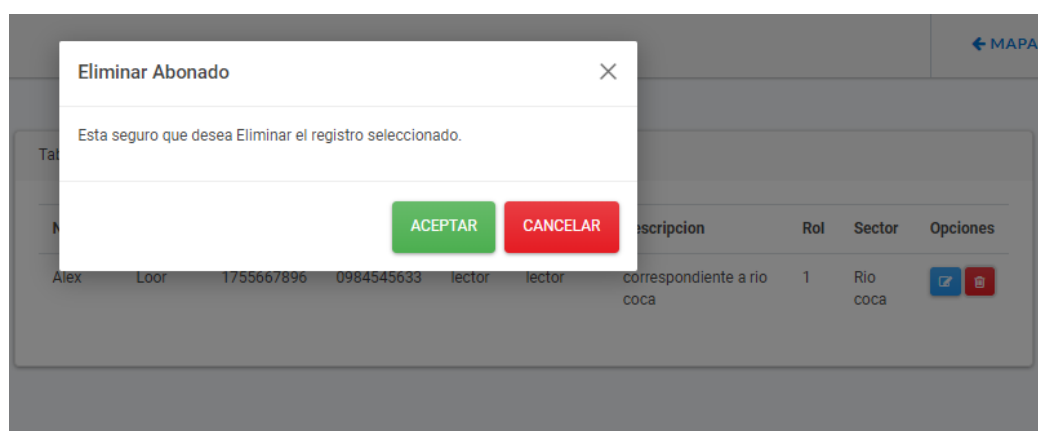


Figura 71. Control de eliminación de registros.

3.1.19. Tarjeta: ADM-11

Tabla 28.

Implementación historia de usuario ADM-11.

HISTORIA DE USUARIO: ADM-11			
Rol de usuario: Administrador	Desarrollador: William Marcillo	Acceso: T2	Módulo: Gestión de Cuentas
Descripción			
Yo como director de planificación de procesos EMAPA, quiero ingresar los datos de una cuenta, para vincularlos con un abonado existente.			
Elementos del código			
MVC		Funciones Principales	
Servicio	ServicioCuentas.js	<ul style="list-style-type: none"> • initMap(); • IngresarCuenta(); • obtenerSector(); • obtenerAbonados(); • getCuentasxSector(); 	
Modelo	Cuenta		
Vista	NuevaCuenta.html		
Controlador	NuevaCuentaCtrl.js		
Implementación			
Para vincular la cuenta con un abonado existente, el sistema trae una lista para seleccionar el abonado dentro del control "Abonado" que deberá ser ingresado junto con los demás campos solicitados.			

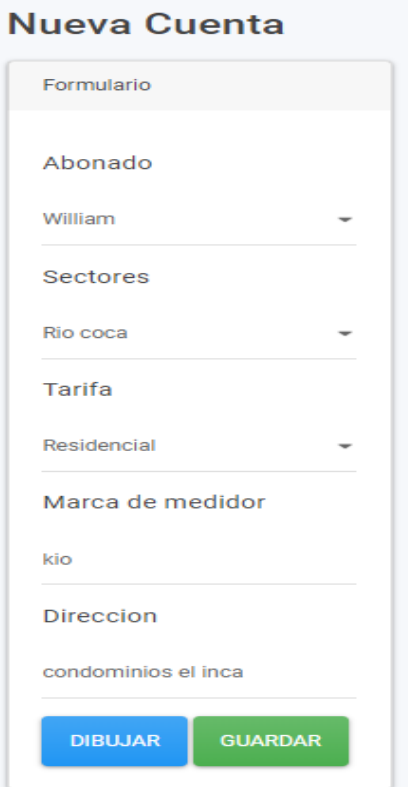


Figura 72. Interfaz de ingreso de nuevas cuentas.

3.1.20. Tarjeta: ADM-12

Tabla 29.

Implementación historia de usuario ADM-12.

HISTORIA DE USUARIO: ADM-12			
Rol de usuario: Administrador	Desarrollador: William Marcillo	Acceso: T2	Módulo: Gestión de Cuentas
Descripción			
Yo como director de planificación de procesos EMAPA, quiero ingresar los datos de una cuenta, para vincularlos con un sector existente.			
Elementos del código			
MVC		Funciones Principales	
Servicio	ServicioCuentas.js	<ul style="list-style-type: none"> • initMap(); • IngresarCuenta(); • obtenerSector(); • obtenerAbonados(); • getCuentasxSector(); 	
Modelo	Cuenta		
Vista	NuevaCuenta.html		
Controlador	NuevaCuentaCtrl.js		
Implementación			
Para vincular la cuenta con un sector existente, el sistema trae una lista para seleccionar el sector dentro del control "Sector" que deberá ser ingresado junto con los demás campos solicitados.			

The image shows a web form titled "Nueva Cuenta" (New Account). The form is contained within a light blue border and has a header "Formulario". It contains several input fields, each with a label and a dropdown menu:

- Abonado:** A dropdown menu with "William" selected.
- Sectores:** A dropdown menu with "Rio coca" selected.
- Tarifa:** A dropdown menu with "Residencial" selected.
- Marca de medidor:** A text input field with "kio" entered.
- Direccion:** A text input field with "condominios el inca" entered.

At the bottom of the form, there are two buttons: a blue button labeled "DIBUJAR" and a green button labeled "GUARDAR".

Figura 73. Vinculación de sectores a una nueva cuenta.

3.1.21. Tarjeta: ADM-13

Tabla 30.

Implementación historia de usuario ADM-13.

HISTORIA DE USUARIO: ADM-13			
Rol de usuario: Administrador	Desarrollador: William Marcillo	Acceso: T2	Módulo: Gestión de Cuentas
Descripción			
Yo como director de planificación de procesos EMAPA, quiero insertar marcadores georreferenciados dentro de un sector del mapa, para tener una visualización grafica de la ubicación de la cuenta.			
Elementos del código			
MVC		Funciones Principales	
Servicio	ServicioCuentas.js	<ul style="list-style-type: none"> • initMap(); • IngresarCuenta(); • obtenerSector(); • obtenerAbonados(); • getCuentasxSector(); 	
Modelo	Cuenta		
Vista	NuevaCuenta.html		
Controlador	NuevaCuentaCtrl.js		
Implementación			
El Sistema abre una página con un formulario en la parte izquierda y presenta un mapa del lado derecho. El usuario tendrá que ingresar los datos del formulario en los campos solicitados.			
Para habilitar los controles es necesario dar clic en el botón “dibujar”, esto hará visible los controles de mapa para posicionar un medidor de agua georreferenciado con un marcador circular. Luego el usuario deberá Seleccionar el marcador circular y posicionarlo dentro del área del sector.			

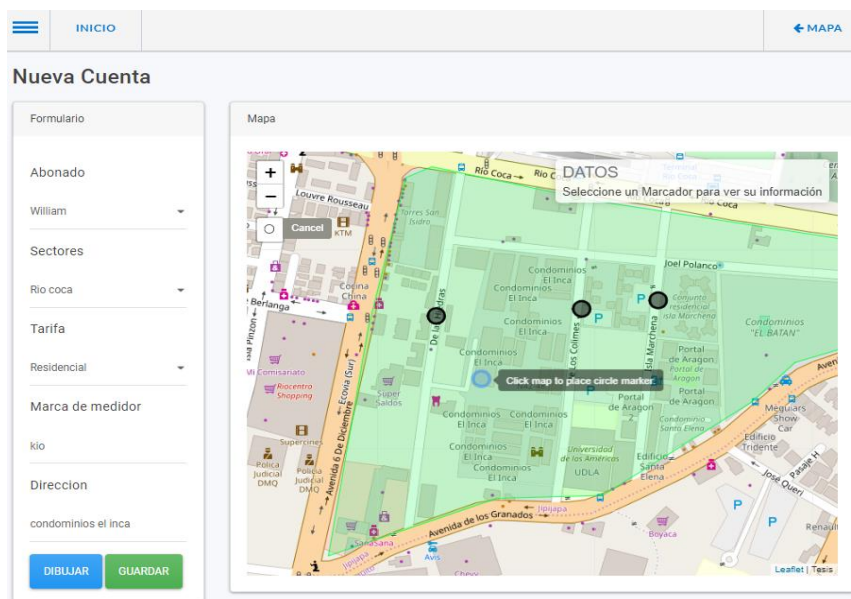


Figura 74. Mapa de medidores.

3.1.22. Tarjeta: ADM-14

Tabla 31.

Implementación Historia de usuario ADM-14.

HISTORIA DE USUARIO: ADM-14			
Rol de usuario: Administrador	Desarrollador: William Marcillo	Acceso: T2	Módulo: Gestión de Cuentas
Descripción			
Yo como director de planificación de procesos EMAPA, quiero consultar una lista de las cuentas, para poder realizar operaciones de edición.			
Elementos del código			
MVC		Funciones Principales	
Servicio	ServicioCuentas.js	<ul style="list-style-type: none"> • initMap(); • actualizarCuenta(); • obtenerSector(); • obtenerAbonados(); • getCuentasxSector(); 	
Modelo	Cuenta		
Vista	ActualizarCuenta.html		
Controlador	ActualizarCuentaCtrl.js		
Implementación			
<p>Para actualizar una cuenta el usuario deberá seleccionar un registro y dar clic en el botón editar que se encuentra dentro del mapa, con esta acción el marcador circular se resalta con color azul.</p> <p>Con esa acción el marcador está disponible para moverse.</p> <p>El usuario podrá reposicionar el marcador a la ubicación deseada dentro del área del mapa. Finalmente, los cambios permanecerán cuando el usuario de clic el botón "Guardar".</p>			

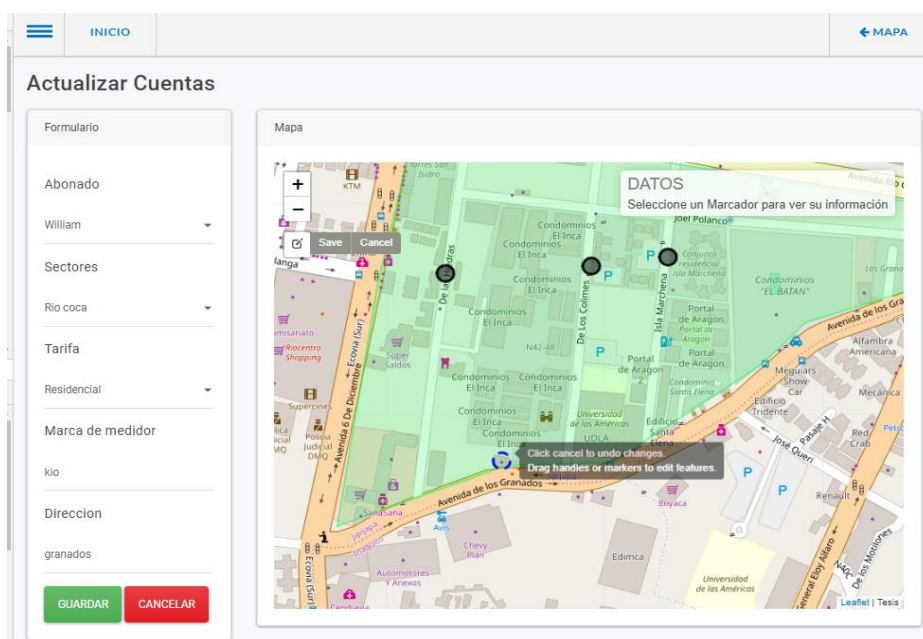


Figura 75. Edición de cuentas.

3.1.23. Tarjeta: ADM-15

Tabla 32.

Implementación historia de usuario ADM-13.

HISTORIA DE USUARIO: ADM-13			
Rol de usuario: Administrador	Desarrollador: William Marcillo	Acceso: T2	Módulo: Gestión de Cuentas
Descripción			
Yo como director de planificación de procesos EMAPA, quiero consultar una lista de las cuentas, para poder borrarlas del sistema.			
Elementos del código			
MVC		Funciones Principales	
Servicio	ServicioCuentas.js	<ul style="list-style-type: none"> • SeleccionarCuentas(); • obtenerCuentas(); • BorrarCuentas(); 	
Modelo	Cuenta		
Vista	GestionCuentas.html		
Controlador	GestionCuentasCtrl.js		
Implementación			
Para borrar un registro, el sistema muestra en la interfaz de gestión de cuentas un botón Eliminar, luego deberá realizar clic en aceptar en el mensaje de confirmación para completar la acción.			

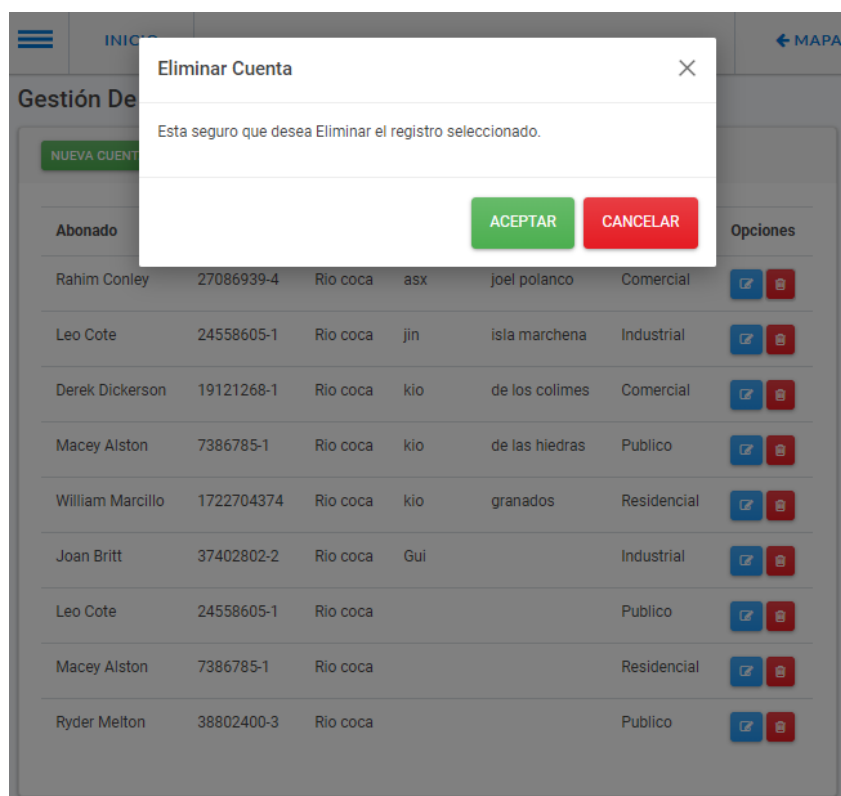


Figura 76. Control de eliminación de registros.

3.2. Conclusión del capítulo

En este capítulo se revisó el flujo de trabajo en el desarrollo del aplicativo respecto con la metodología Kanban. Se mostraron los principales componentes del software en el módulo web y móvil. Se mencionaron las funciones principales del código, la interacción con la arquitectura MVC y el diseño gráfico de las interfaces.

4. CAPÍTULO IV. ESTUDIO DE USABILIDAD DE USUARIO

La usabilidad sumativa se realiza mediante la experiencia de usuario UX (User Experience). UX es un concepto estratégico que pretende conectar al usuario con un producto a nivel emocional. UX es importante en el modelo de desarrollo de software para resolver problemas de usuario y promover el incremento de voluntad de aceptación.

UX reúne los requerimientos y las necesidades de usuario en contraste con las limitaciones tecnológicas, para brindar experiencias satisfactorias deseables y realizables.

Dentro de este marco, el estudio que se presenta en esta sección busca analizar el nivel satisfacción de usuario del “Sistema web progresivo para el control de toma de lecturas de agua potable”, mediante la obtención de datos por encuestas a usuarios que probaran el aplicativo. El objetivo es medir índices de satisfacción, calidad de la información y calidad de la interacción de usuario. A partir de las apreciaciones de los usuarios se pretende generar estadísticas de análisis de usabilidad y estudiar la eficacia del uso de la aplicación web progresiva GLAP.

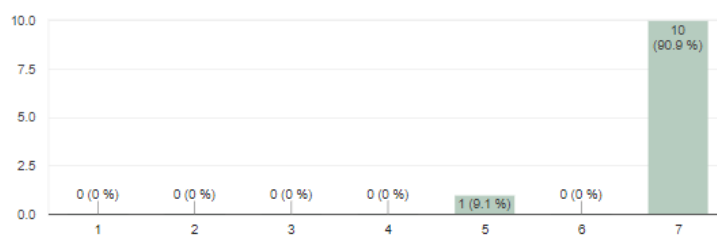
Los resultados obtenidos facilitarán el reconocimiento de las fortalezas y debilidades del diseño, dejando la pauta de los puntos que se deben optimizar para mejorar la experiencia de usuario.

4.1. Participantes

Dadas las limitaciones de reunir al personal que labora en la Empresa Municipal de Agua Potable (EMAP) se contó con la participación voluntaria de solo 11 personas. A pesar de que fueron conocedores del contexto de estudio no se requirió de ninguna habilidad específica de los participantes. Se realizó una encuesta demográfica cuyos datos sugieren que el promedio de edad de los participantes es de 32 años, de los cuales 8 personas fueron hombres y tres fueron mujeres. Todos los participantes pertenecen a la Empresa Municipal de Agua Potable. Dentro de los 11 participantes se pudo contar con tres lectores de medidores de agua. La figura 77, muestra el perfil de los usuarios en base a la frecuencia de uso de un computador, un Smartphone y el manejo de sistemas de lecturas de agua potable.

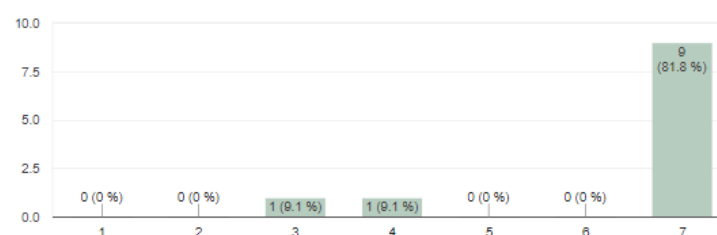
Uso una computadora frecuentemente:

11 respuestas



Uso un smartphone frecuentemente:

11 respuestas



Uso sistemas de que gestionen procesos de agua potable:

11 respuestas

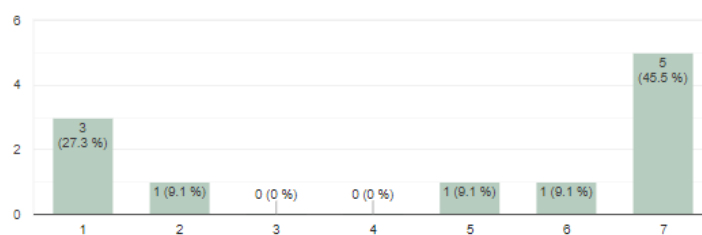


Figura 77. Resultados de encuesta demográfica.

El 90% de los consultados usa una computadora con frecuencia. De igual manera más del 80% de los entrevistados admitieron en uso frecuente de un teléfono inteligente. Respecto a la pregunta del uso de sistemas de gestión de lecturas, solo 5 de los 11 participantes respondieron esta pregunta con el valor 7 (muy de acuerdo). Sin embargo, tres participantes admitieron no haber usado sistemas para la gestión de lecturas asignado la calificación 1 y 2 de la escala.

4.2. Tareas

Dado que el aplicativo GLAP maneja un rol Administrador y un rol Lector se propusieron 4 tareas para cada perfil. Se debe tomar en cuenta que el usuario administrador realizó las tareas desde un equipo de escritorio, mientras que el usuario lector realizó las tareas desde un dispositivo móvil. Ambos perfiles tienen una etapa de familiarización con el aplicativo y las tareas se describen a continuación.

4.2.1. Tareas usuario administrador

Tabla 33.

Descripción de tareas del administrador.

#	Tarea	Descripción
0	Familiarización con el sistema	Esta tarea le permitirá familiarizarse con la aplicación y comprender las tareas a realizar. En este momento recibirá una formación sobre las diferentes funcionalidades del sistema. Por favor, haga las preguntas que considere necesarias.
1	Ingresar y editar un sector	En esta tarea, deberá visualizar los cambios de los polígonos que conforman los sectores en el mapa.
2	Ingresar y editar una cuenta	En esta tarea, deberá posicionar los marcadores circulares dentro de un sector en el mapa
3	Ingresar y editar un abonado	En esta tarea, deberá ver los cambios realizados en la tabla de abonados.

4.2.2. Tareas usuario lector

Tabla 34.

Descripción de tareas del usuario lector.

#	Tarea	Descripción
0	Familiarización con el sistema	Esta tarea le permitirá familiarizarse con la aplicación y comprender las tareas a realizar. En este momento recibirá una formación sobre las diferentes funcionalidades del sistema. Por favor, haga las preguntas que considere necesarias.
1	Descargar la aplicación al dispositivo móvil	En esta tarea, deberá conseguir la instalación del aplicativo mediante Google Chrome en la URL indicada.
2	Realizar una lectura	En esta tarea, deberá registrar una lectura identificando un abonado mediante la lectura de un código QR.
3	Editar una lectura y cambiar a estado pendiente.	En esta tarea, le permitirá visualizar el cambio realizado con un marcador circular de color naranja en el mapa.

4.3. Materiales y métodos

El ambiente de pruebas para llevar a cabo el estudio de usabilidad se implementó esencialmente con tres dispositivos, una laptop de desarrollo (utilizada como servidor web), un equipo de escritorio y un dispositivo móvil. La tabla 35 detalla las especificaciones técnicas para cada uno de estos dispositivos.

Tabla 35.

Lista de materiales usados en el experimento.

Materiales		
Servidor web	Cliente	Dispositivo móvil
Servicio: Sails Marca: Lenovo Modelo: G40 Disco duro: 500GB RAM: 8GB SO: Windows 10	Usuario: Administrador Marca: Dell Modelo: Optiplex 3060 Disco duro: 500GB RAM: 4GB SO: Windows 8.1 Sistema: 64 bits	Usuario: Lector Marca: Sony Modelo: Xperia Z5 Almacenamiento interno: 32GB SD: 16GB RAM: 3GB

Sistema: 64 bits Procesador: Intel Core i5 Pantalla: 14"	Procesador: Intel Core i3 Pantalla: 19" Navegador: Google Chrome 71.0.35	SO: Android Nougat 7.1.1 Sistema: 64 bits Procesador: snap dragon 810 Pantalla: 5.2" Cámara Principal: 23Mpx Navegador: Google Chrome 71.0.35
--	--	--

Se utilizó la infraestructura de red municipal para conectar a internet al servidor y el equipo de escritorio. El dispositivo móvil se conectó mediante la red 4G. El servicio se hizo visible para los usuarios a través del enlace temporal <https://2cbb4248.ngrok.io>, que se publicó y configuró previamente en un tunneling creado por la herramienta Ngrok. Esto se explica mejor en el esquema de pruebas de la figura 78.

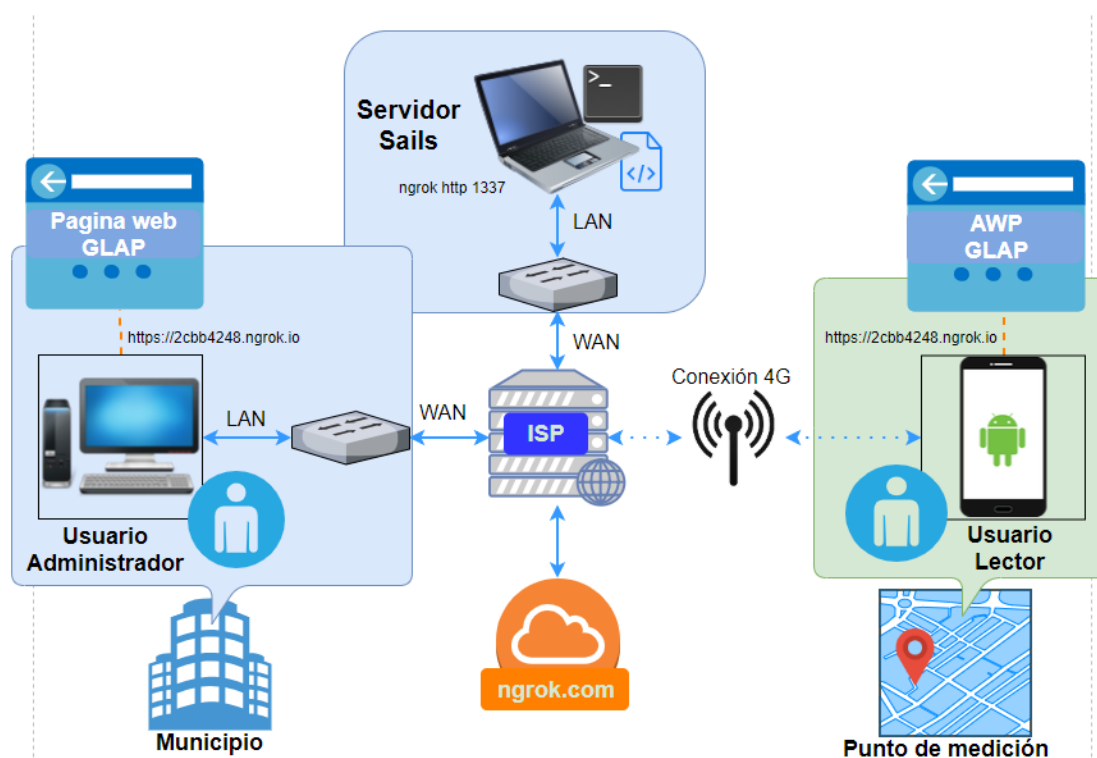


Figura 78. Esquema de pruebas.

El ejercicio de usabilidad se compuso de 2 fases para los 11 participantes, la primera consistía en que cada participante probara la aplicación web con el rol de administrador desde el edificio municipal.

La segunda fase consistía en que cada participante probara el aplicativo móvil. El usuario debía realizar un corto recorrido a través del “Parque Central De Pimampiro”, y caminar hacia cada uno de los 4 adhesivos impresos con códigos QR para simular la toma una lectura de un medidor de agua potable.

Los adhesivos fueron colocados antes de realizar el experimento.

La figura 79, muestra donde se ubicó de cada adhesivo QR y el recorrido que debía hacer cada participante desde el punto de partida hasta el punto de llegada, el trayecto no superaba los 132 metros.

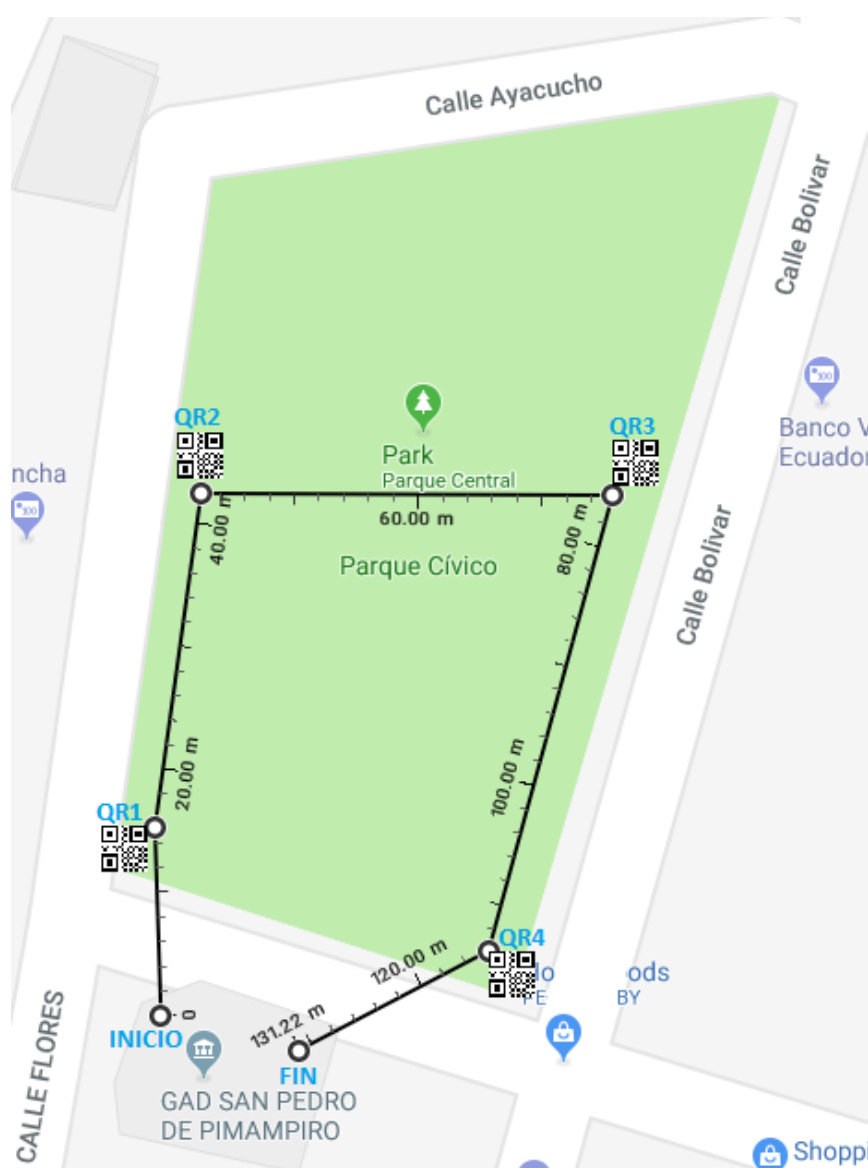


Figura 79. Recorrido de usuario al probar GLAP móvil.

4.4. Procedimiento

El estudio se llevó a cabo principalmente en la sala de reuniones del GAD Municipal Pimampiro con 11 participantes que pertenecen a la Empresa Municipal De Agua potable.

El protocolo experimental comenzó con la firma de un formulario de consentimiento que se otorga a cada participante antes de empezar el experimento, en este formulario se explican los términos, condiciones, riesgos, y se manifiesta que los datos obtenidos en el experimento son totalmente confidenciales y se utilizarán únicamente para motivos académicos.

Después a cada participante se le asignó un número aleatorio de formulario para preservar el anonimato. Este número aleatorio también es una referencia para identificar los resultados del experimento. A continuación, se pidió a los participantes que completaran un cuestionario con información demográfica. Luego, los participantes fueron instruidos en una capacitación no mayor a 15 minutos. Se les explicó las funcionalidades del sistema, sobre cómo usarlo.

Una vez que los participantes se familiarizaron con la aplicación de “Gestión De Lecturas De Agua Potable” (GLAP) y después de que confiaban en el aplicativo, se les pidió que completaran las tareas de administrador y lector 1, 2, y 3 descritas en la sección anterior.

El tiempo total para completar las tareas fue de 15 minutos (5 minutos rol Administrador, 10 minutos rol lector). Al final del experimento, cuando se completaron todas las tareas, se pidió a los participantes que respondieran el Cuestionario de Usabilidad del Sistema CSUQ de IBM. Adicionalmente se agregó una pregunta abierta opcional para obtener comentarios constructivos (¿Hay algún comentario que le gustaría hacer sobre la aplicación de medición de agua GLAP o su experiencia al usarla durante el experimento?).

Finalmente, a los participantes se les entregó una lista de 45 palabras obtenidas a partir de las tarjetas de reacción de productos de Microsoft. En la lista se encontraban palabras de apreciación positiva y negativa para que seleccionen tantas palabras como deseen. Con estas palabras los participantes describirían la experiencia con el aplicativo GLAP.

4.5. Resultados

En esta sección se describe los resultados obtenidos por el cuestionario CSUQ de IBM de acuerdo con la percepción y retroalimentación de los 11 participantes.

El estudio analizó GLAP web y GLAP móvil descritos a continuación.

4.5.1. Resultados GLAP Web

La tabla 36 presenta los resultados individuales por cada una de las preguntas del cuestionario CSUQ de IBM para el sistema GLAP web, contrasta los valores de la media con la desviación standard. Los valores serán analizados a detalle a lo largo de esta sección. El comportamiento de las preguntas 1 y 5 tiene resultados de desviación estándar con valores similares. Es decir, están correlacionadas con valores de desviación standard de 1,51.

La apreciación de los participantes en las preguntas 16,17 y 19 coinciden con los valores medios con un puntaje de 6.45, lo que quiere decir que las interfaces tienen gran influencia para aportar valor a la satisfacción general del sistema. Así mismo, las respuestas 9,10 y 18 son las más críticas y se observa diferencias significativas en las respuestas de los participantes, lo valores medios están por debajo de 4,45 y su desviación standard es la más alta con valores de 2.10.

Tabla 36.

Resultados de la evaluación individual de las preguntas GLAP web.

Categoría	#	Preguntas	Media	SD
SYSUSE	1	En general, estoy satisfecho con lo fácil que es usar este sistema	6,45	1,51
	2	Era simple usar este sistema	5,91	1,51
	3	Podría completar mi trabajo de manera efectiva usando este sistema	5,45	1,63
	4	Pude completar mi trabajo rápidamente usando este sistema	5,91	1,38
	5	Pude completar mi trabajo de manera eficiente usando este sistema	5,55	1,69
	6	Me sentí cómodo usando este sistema	6,27	1,19
	7	Fue fácil aprender a usar este sistema	5,82	2,04
	8	Creo que podría ser productivo rápidamente usando este sistema	6,18	1,60
INFOQUIAL	9	El sistema dio mensajes de error que claramente me dicen cómo solucionar problemas	4,27	2,10
	10	Cada vez que cometí un error al utilizar el sistema, podría recuperarme fácil y rápidamente	4,45	2,11
	11	La información (como ayuda en línea, mensajes en pantalla y otra documentación) proporcionada con este sistema era clara	5,27	1,85
	12	Fue fácil encontrar la información que necesitaba	5,73	1,56
	13	La información provista para el sistema fue fácil de entender	5,73	1,74
INTERQUIAL	14	La información fue efectiva para ayudarme a completar las tareas y los escenarios	5,45	1,69
	15	La organización de la información en las pantallas del sistema fue clara	6,00	1,67
	16	La interfaz de este sistema fue agradable	6,45	0,93
OVERALL	17	Me gustó usar la interfaz de este sistema	6,55	0,82
	18	Este sistema tiene todas las funciones y capacidades que espero que tenga	4,45	1,75
OVERALL	19	En general, estoy satisfecho con este sistema	6,45	1,04

Para que los resultados expresen una apreciación favorable al sistema, es necesario que el promedio obtenido en las métricas de CSUQ sea superior a 4. La figura 80, indica que todas las categorías evaluadas en el cuestionario han obtenido como promedios valores superiores a 5. Eso quiere decir que los usuarios han valorado positivamente el sistema en aspectos de la calidad de la de información, calidad de la interacción y calidad global.

Al mismo tiempo el grafico de la figura 80, muestra que el promedio 5,27 es el más bajo y corresponde a la categoría infoqual. Estos resultados sugieren que el sistema debe mejorar la forma de presenta su información (mensajes de error) al usuario para resolver posibles errores.

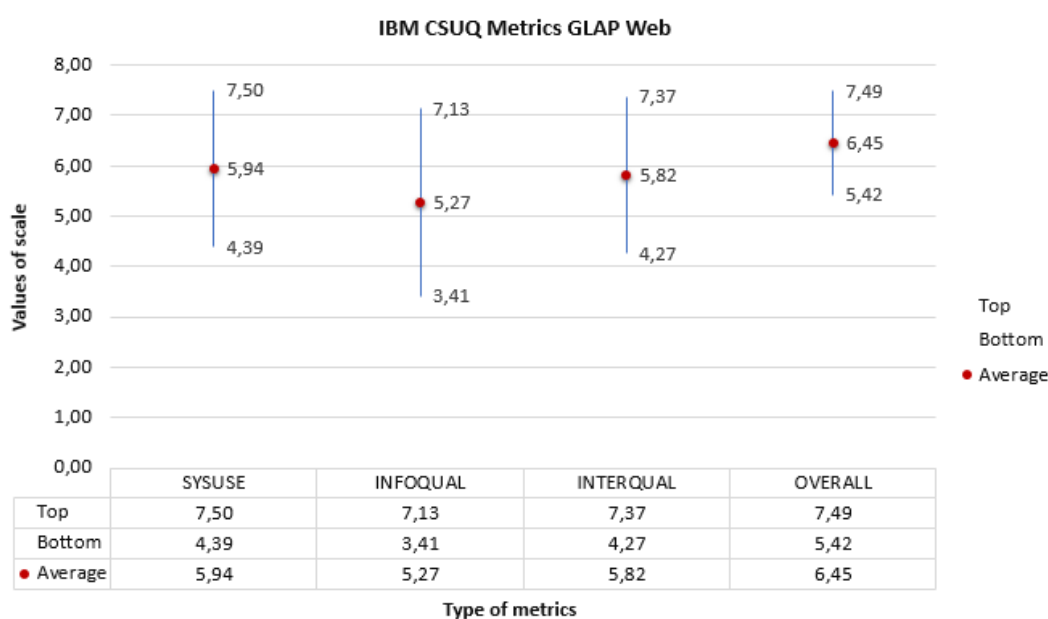


Figura 80. Métricas de CSUQ de GLAP web.

La figura 81, muestra las 19 respuestas individuales de los 11 participantes del cuestionario CSUQ, los resultados individuales indican que los participantes con el ID 4,10 y 11 fueron menos críticos. Por el contrario, los participantes 6 y 9 son los más críticos. Las preguntas con mayor promedio de calificación corresponden a Q16 y Q17 ambas con un promedio de 6.5, lo que infiere que para los usuarios la interfaz del sistema es agradable y gustan usar de ella.

La pregunta Q19 alcanza un promedio de 6.4 siendo también una de las más valoradas, esto indica que los usuarios en general están satisfechos con el sistema. La pregunta menos apreciada por los participantes del cuestionario de IBM CSUQ fue la Q9 con un promedio de 4.27. Este último resultado revela un importante problema de usabilidad para GLAP web ya que el sistema no proporcionó mensajes de error claros y suficientes para solucionar problemas.

ID	Q01	Q02	Q03	Q04	Q05	Q06	Q07	Q08	Q09	Q10	Q11	Q12	Q13	Q14	Q15	Q16	Q17	Q18	Q19
1	7	7	7	6	6	6	5	7	5	4	5	6	7	5	7	7	7	2	7
2	7	6	5	7	7	7	6	7	3	6	7	6	5	5	5	7	7	4	7
3	7	6	4	6	6	7	7	6	5	4	4	5	6	6	7	7	6	4	6
4	7	6	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
5	6	6	5	7	6	6	7	7	4	6	5	6	6	5	6	6	7	4	7
6	7	3	2	4	3	4	3	3	1	1	3	4	5	5	4	6	5	3	5
7	7	7	6	5	5	7	7	7	3	5	7	7	6	6	7	7	7	7	7
8	7	7	6	6	5	7	7	7	3	5	4	6	6	6	7	7	7	4	7
9	2	3	4	3	2	4	1	3	2	1	2	2	1	1	2	4	5	3	4
10	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	6	7	7	7
11	7	7	7	7	7	7	7	7	7	3	7	7	7	7	7	7	7	4	7
Respuesta:	1	2	3	4	5	6	7												

Figura 81. GLAP web, Resultados individuales del cuestionario IBM CSUQ.

La figura 82 indica que los resultados obtenidos en el cuestionario IBM CSUQ, en general se observan tendencias positivas porque la mayoría de los participantes afirmaron estar muy de acuerdo en sus respuestas. Esto concuerda con la predominancia del color verde y en la mayoría de los casos el promedio por pregunta fue mayor a 5.5. El resultado de la pregunta Q18 confirma que los usuarios no estaban totalmente de acuerdo, pero tampoco en desacuerdo respecto a que el sistema tenga todas las funcionalidades y capacidades que esperan que tenga, el mayor porcentaje fue neutral y predomina el color amarillo. Para la pregunta Q16, a juicio del 90% de los participantes la interfaz del sistema fue agradable, mientras que el 10% se mantuvo en una posición neutral. Según la figura 82, uno de los puntos débiles del sistema es la dificultad para recuperarse fácil y rápidamente de un error, esto tendría relación directa con la evaluación de que el sistema debe mejorar la presentación de mensajes de errores que indiquen como solucionar problemas.

Esta afirmación es visible en las preguntas Q9 y Q10, en ambos casos alcanzan la desaprobación más alta del cuestionario por parte de los participantes donde alcanza un 20% de desacuerdo. Uno de los aspectos positivos del sistema según el estudio, es que la mayoría de los participantes creen que sí podrían ser productivos con GLAP web, específicamente un 80% afirmo estar muy de acuerdo. En el mismo sentido 9 de los 11 participantes estuvieron muy de acuerdo en que el sistema resulta fácil de aprender.

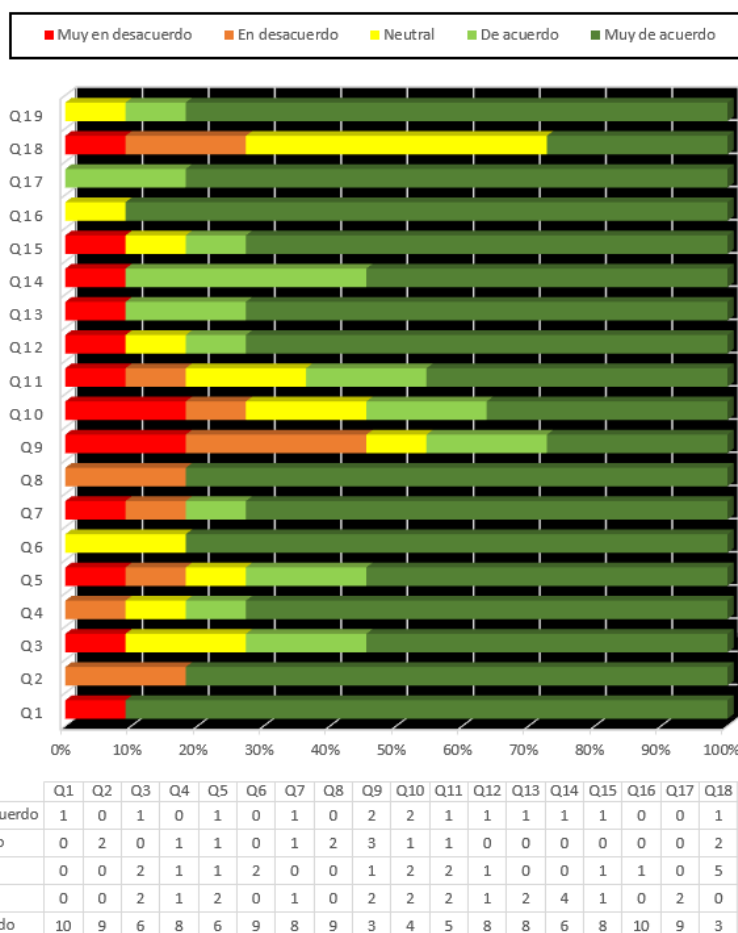


Figura 82. Resultados generales del cuestionario IBM CSUQ.

Los resultados de palabras de la figura 83 muestran la frecuencia con la que los participantes puedan describir mejor la experiencia después de usar GLAP web. 72% de los participantes han descrito su experiencia con la palabra “fácil”. Las palabras “fácil de usar,” “eficiente” y “ahorro de tiempo”, han sido seleccionadas por 6 ocasiones de 11 posibles.

Otro aspecto positivo del sistema respecto a la usabilidad es que el 45% (5 ocasiones de 11 posibles) de los usuarios calificaron al sistema como rápido, simple e intuitivo. Por el contra parte en porcentajes menores al 18% algunos usuarios mencionaron que el sistema les resultó confuso sofisticado y muy técnico. Sin embargo, en términos generales el sistema tiene una reacción más favorable sobre los participantes pues han seleccionado palabras como “amistoso”, “motivador”, “claro” y “poderoso”. La figura 83, también muestra como los participantes dejaron de lado muchas palabras desfavorables para catalogar al sistema GLAP Web.

Gracias a esta recopilación estadística de palabras, en base a la frecuencia de selección se pudo graficar un mapa de palabras para describir mejor la experiencia de usuario con el sistema GLAP web. Tal como lo indica la figura 84, las palabras más sobresalientes son: útil, fácil, eficiente y rápido.

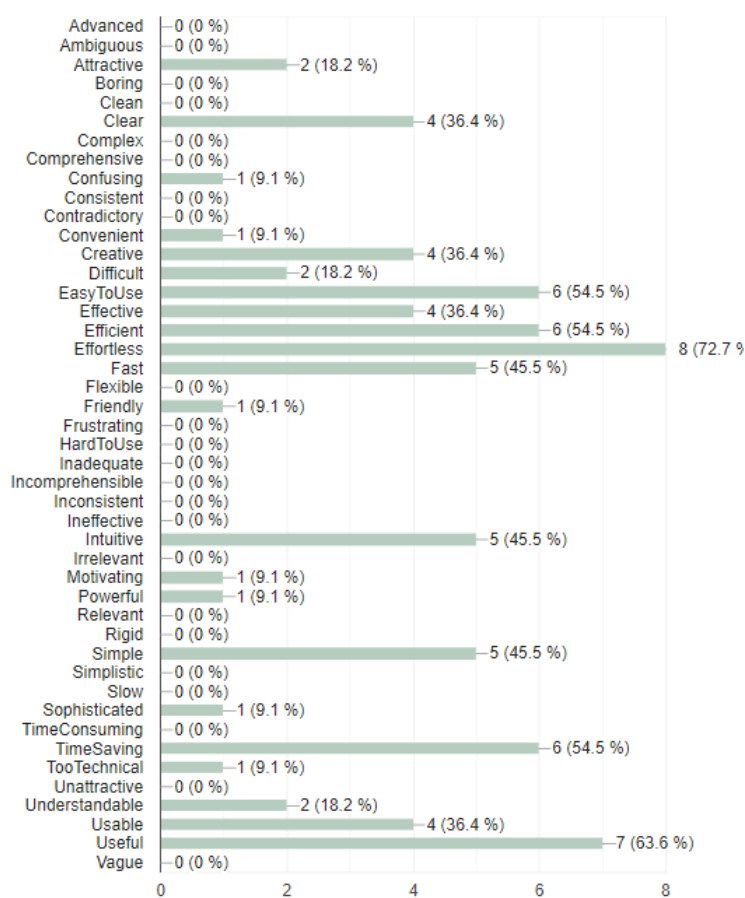


Figura 83. Resumen de selección de palabras GLAP web.

La desviación estándar más baja se encuentra en las preguntas 17(interquial) y 19 (overall) con valores medios que superan el 6.36, lo que indica un alto porcentaje de coincidencia de respuestas favorables con la interfaz y satisfacción general del sistema.

Tabla 37.

Resultados de la evaluación individual de las preguntas GLAP móvil.

Categoría	#	Preguntas	Media	SD
SYSUSE	1	En general, estoy satisfecho con lo fácil que es usar este sistema	6,27	1,27
	2	Era simple usar este sistema	6,09	1,22
	3	Podría completar mi trabajo de manera efectiva usando este sistema	6,27	1,42
	4	Pude completar mi trabajo rápidamente usando este sistema	6,36	0,92
	5	Pude completar mi trabajo de manera eficiente usando este sistema	6,09	1,22
	6	Me sentí cómodo usando este sistema	6,45	1,04
	7	Fue fácil aprender a usar este sistema	6,45	0,82
	8	Creo que podría ser productivo rápidamente usando este sistema	6,18	1,33
INFOQUIAL	9	El sistema dio mensajes de error que claramente me dicen cómo solucionar problemas	5,36	1,91
	10	Cada vez que cometí un error al utilizar el sistema, podría recuperarme fácil y rápidamente	5,45	1,63
	11	La información (como ayuda en línea, mensajes en pantalla y otra documentación) proporcionada con este sistema era clara	5,27	1,74
	12	Fue fácil encontrar la información que necesitaba	5,82	1,40
	13	La información provista para el sistema fue fácil de entender	5,91	1,22
	14	La información fue efectiva para ayudarme a completar las tareas y los escenarios	5,55	1,37
INTERQUIAL	15	La organización de la información en las pantallas del sistema fue clara	6,00	1,26
	16	La interfaz de este sistema fue agradable	6,45	1,04
	17	Me gustó usar la interfaz de este sistema	6,73	0,65
	18	Este sistema tiene todas las funciones y capacidades que espero que tenga	5,09	1,64
OVERALL	19	En general, estoy satisfecho con este sistema	6,36	0,81

La figura 85, de manera general también muestra resultados favorables, porque todos sus promedios presentan valores superiores a 4. A criterio de los participantes el sistema les pareció fácil de aprender y fácil de usar. Esto se refleja en el hecho de que la categoría Sysuse alcanza un promedio de 6.27. A pesar de que el Infoquial de GLAP móvil es superior a 5 es el promedio más bajo de las 4 categorías. Esta tendencia fue la misma en GLAP Web por lo cual se recomienda mejorar la calidad de la información. El aspecto más positivo de este análisis es que de manera global el usuario está satisfecho con el sistema, ya que en la categoría Overall tiene el promedio más alto con un 6.36.

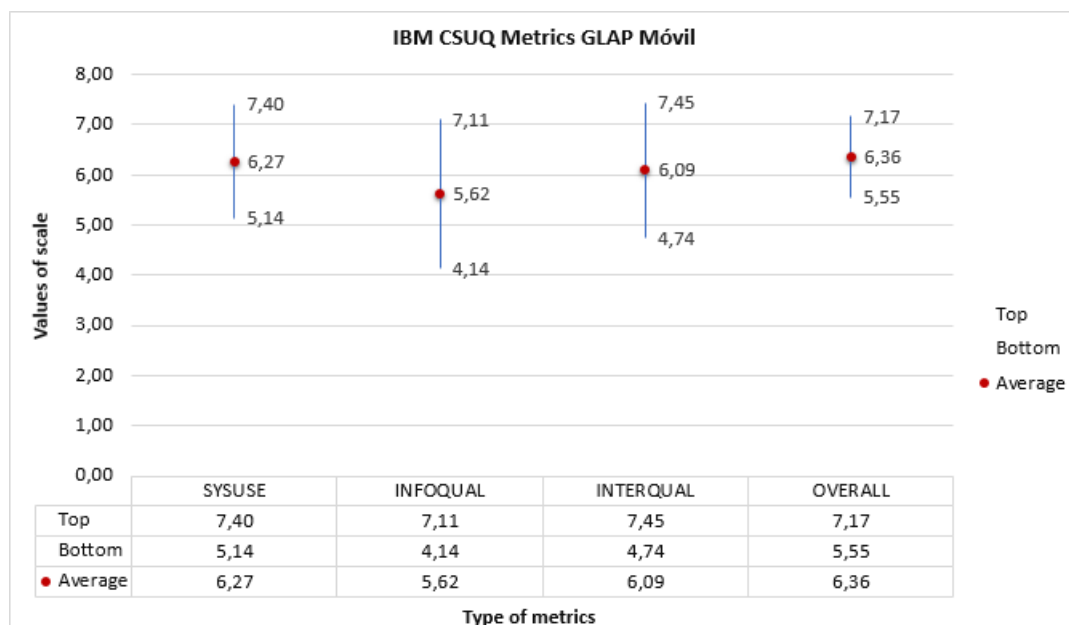


Figura 85. Métricas de CSUQ de GLAP móvil.

Los resultados que muestra el cuadro de la figura 86, indica criterios mucho más favorables para GLAP móvil en comparación con las respuestas individuales de GLAP web de la figura 81. En ningún caso un participante asignó la calificación más baja para las 19 preguntas. A pesar de que las respuestas son diferentes a los resultados obtenidos en GLAP web, los resultados individuales obtenidos para GLAP móvil, tienen patrones repetitivos por usuario, por ejemplo; los participantes con ID 4,10 y 11 son más positivos con el sistema. De igual forma como sucedió en GLAP web, los participantes con ID 6 y 9 parecen ser los más críticos.

La pregunta Q18 tuvo el promedio más bajo con 5.1, esto significa que los usuarios esperan funcionalidades adicionales. Tal como el caso anterior en GLAP móvil, los usuarios presentan las mismas dificultades para recuperarse de un error y requieren que el sistema emita más mensajes apropiados y oportunos de error. Esto se hace visible en el promedio de 5.45 y 5.27 de las preguntas Q10 y Q11. Los resultados más favorables, se encuentran en las preguntas Q07 y Q16, ambas alcanzan un promedio de 6.5. Infiere que los participantes aprendieron el sistema con facilidad y que la interfaz del sistema fue de su agrado.

En los resultados de la pregunta Q06 los participantes admitieron que se sienten cómodos usando el sistema y en la pregunta Q19 el resultado promedio de 6.4 indica que los participantes están satisfechos con el sistema.

ID	Q01	Q02	Q03	Q04	Q05	Q06	Q07	Q08	Q09	Q10	Q11	Q12	Q13	Q14	Q15	Q16	Q17	Q18	Q19
1	7	7	7	7	6	5	6	7	2	5	5	6	4	5	5	7	7	4	7
2	7	6	7	7	7	7	6	6	5	5	5	5	6	3	5	7	7	4	6
3	7	6	7	5	4	7	7	5	6	5	4	5	6	5	7	6	6	3	6
4	7	5	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
5	6	7	7	7	7	7	7	7	7	7	7	7	5	4	7	7	7	5	7
6	5	4	4	5	5	6	5	3	3	3	2	3	4	5	4	4	5	3	5
7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
8	7	7	7	7	7	7	7	7	7	7	6	7	7	7	7	7	7	5	6
9	3	4	3	5	4	4	5	5	3	3	3	4	5	5	4	5	7	4	5
10	6	7	6	6	6	7	7	7	5	4	5	6	7	6	6	7	7	7	7
11	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Respuesta:	1	2	3	4	5	6	7												

Figura 86. GLAP móvil, Resultados individuales del cuestionario IBM CSUQ.

En el esquema general de resultados de la figura 87 obtenidos en el cuestionario CSUQ de IBM, muestra una tendencia favorable ya que los participantes afirmaron estar muy de acuerdo en la mayoría de las preguntas. La pregunta Q1 indica que el 90% de los participantes estuvo satisfecho con lo fácil que es aprender el sistema. A pesar de que el 20% de los participantes se mantuvo neutral, el 80% afirmó estar de acuerdo respecto a la simplicidad de usar el sistema. Los resultados también sugieren que los participantes han apreciado la calidad de la información gestionada por GLAP web. Según la pregunta Q14, 9 de 11 participantes indicaron que la información proporcionada por el sistema era fácil de entender.

Así les ayudo a completar las tareas de los ejercicios propuestos. Nuevamente una de las fortalezas del sistema es la comodidad que brinda las interfaces a los usuarios según la pregunta Q06, esto también se corrobora con los resultados favorables de las preguntas Q16 y Q17 donde al menos 9 de 11 participantes estuvieron de acuerdo. Otro aspecto que notar según los resultados de la pregunta Q18, es que al menos el 18% de los participantes desean más funcionalidades o capacidades en el sistema, y un 27% tuvo una posición neutral.

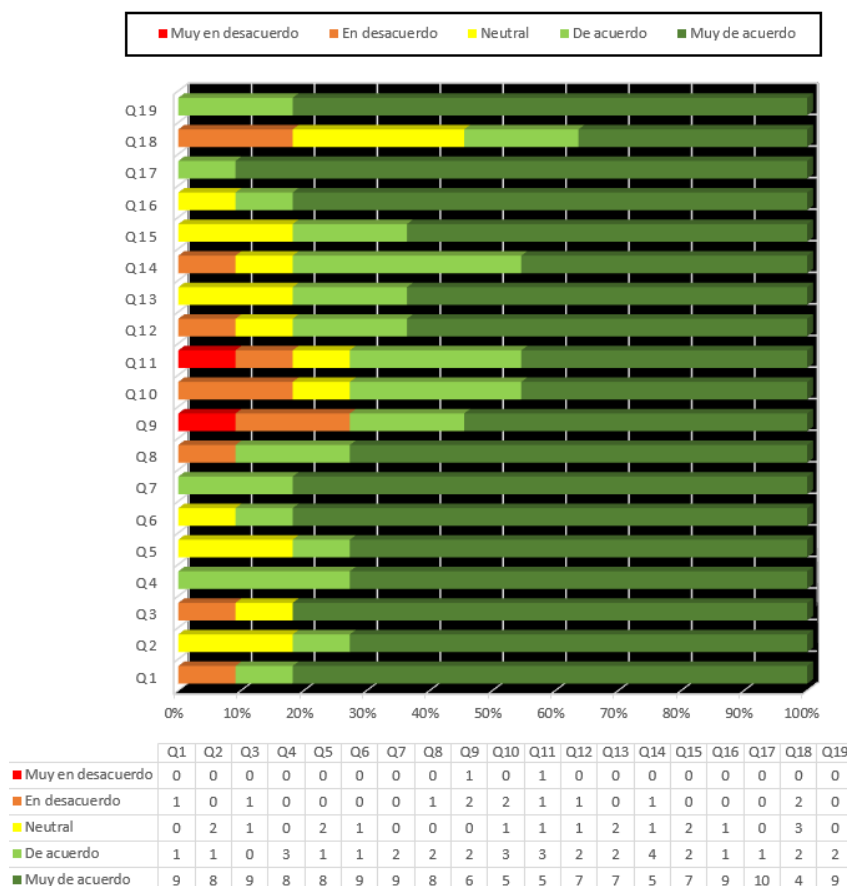


Figura 87. GLAP móvil, Resultados generales del cuestionario IBM CSUQ.

“Rápido” y “fácil” son las palabras que mejor describe la experiencia con GLAP web según la figura 88, porque han sido las palabras que se han seleccionado con mayor frecuencia. Las palabras “eficiente” y “fácil de usar” han sido seleccionadas por 6 ocasiones determinando un 56% de frecuencia.

Los participantes también se refirieron a GLAP móvil como un sistema “útil” e “intuitivo” por 5 ocasiones.

Según los resultados de la figura 88, solamente un usuario de los 11 posibles ha catalogado el sistema como “muy técnico” e incluso “lento”. Sin embargo, en términos generales GLAP móvil tienen una percepción favorable para los usuarios que probaron el aplicativo, mencionaron que el sistema les permitiría “ahorrar tiempo”, que es “claro” y “amigable”.

La figura 89, muestra el mapa de palabras generado como experiencia de usuario de GLAP móvil, las 4 palabras que más resaltan visualmente en el mapa son “útil”, “fácil”, “rápido” y “eficiente”.

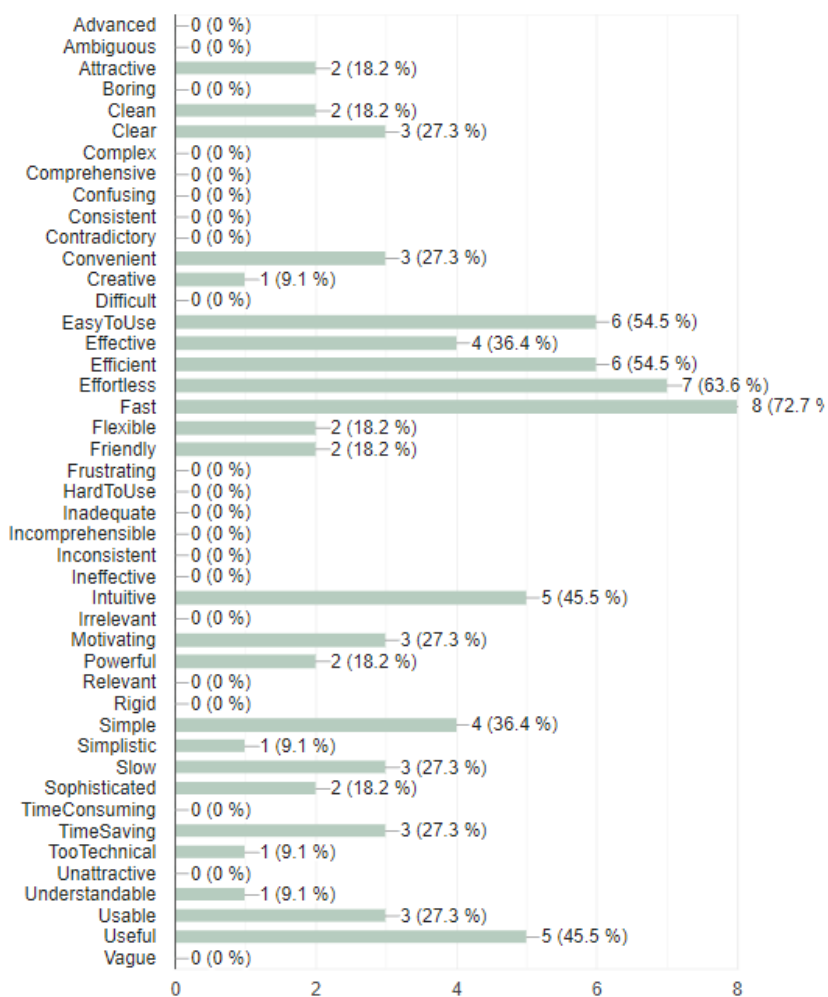


Figura 88. Resumen de selección de palabras GLAP móvil.



Figura 89. Mapa de palabras UX GLAP Móvil.

4.6. Conclusión de capítulo

Los resultados obtenidos en el estudio de usabilidad en general tienen una reacción positiva para los participantes. Las cuatro categorías del cuestionario CSUQ que analizan calidad de interfaces, calidad de información, calidad de interacción y satisfacción global presentan un balance positivo según los cuadros estadísticos analizados. La encuesta demográfica también revela que los usuarios experimentados en el uso de computadoras también tienen una alta experiencia en el uso de smartphones. Así mismo se determinó que la edad si afecta la experiencia de usuario, ya que los participantes de mayor edad con ID 6 (55 años) y 9 (46 años) son los usuarios más críticos calificando al sistema con bajos puntajes en sus respuestas. En general cuanto más jóvenes sean los participantes mejor será su experiencia con el sistema GLAP. Así mismo, los participantes que han interactuado con sistemas que tengan relación con procesos de agua potable, han calificado de mejor forma, respecto a los participantes que no han usado sistemas de agua potable. Los resultados obtenidos en GLAP web y GLAP móvil, infiere que el sistema debe mejorar la calidad de la información. Finalmente, los participantes se alinean con el hecho de que el sistema puede incorporar más funcionalidades.

5. CONCLUSIONES Y RECOMENDACIONES

En este último capítulo se mencionarán los resultados obtenidos a lo largo del desarrollo del proyecto de titulación, además se describe algunas recomendaciones que serán sugeridas a continuación en sus respectivos puntos.

5.1. Conclusiones

Este proyecto de titulación es una contribución para el desarrollo de aplicativos que gestionen el servicio de agua potable, orientados hacia la gobernabilidad digital de los municipios del país.

Las aplicaciones web progresivas brindan la facilidad de que con un mismo desarrollo se implementen sistemas web y sistemas para dispositivos móviles.

El desarrollo de este proyecto demostró que el método de desarrollo centrado en el usuario permitió la construcción de prototipos en un proceso iterativo que promueve la evolución de las interfaces y que se validan con la aprobación del usuario. El ciclo refino el modelo y agrego mayor funcionalidad cada vez. Por ejemplo, GLAP web inicio con mediante el dibujo de un boceto, luego se convirtió en un mockup para analizar su comportamiento y finalmente evoluciono con el desarrollo funcional de una primera versión que se mejoró en el tiempo con los lineamientos de la metodología Kanban.

Se realizo un estudio de usabilidad del sistema GLAP web y GLAP móvil con los usuarios de la Empresa Municipal De Agua Potable en Pimampiro, con 11 voluntarios.

Los resultados obtenidos de un estudio de usabilidad de usuario del sistema web progresivo para el control y toma de lecturas de medidores de agua potable, indicaron una reacción positiva de los participantes, los usuarios expresaron su satisfacción con el sistema en términos de Calidad de la información, calidad de interfaces y calidad de interacción.

Este proyecto permitió identificar aspectos específicos de usabilidad que deben implementarse para garantizar una experiencia de usuario eficiente.

Se obtuvieron elementos para afirmar que los usuarios con el sistema evitan procesos de digitación, optimizan tiempos de lectura y visualizan el recorrido de lectores.

5.2. Recomendaciones

Las aplicaciones web progresivas obligatoriamente necesitan de un protocolo HTTPS para su correcta ejecución. En el mismo contexto los navegadores web actualmente han reforzado sus políticas de seguridad, por lo cual se recomienda implementar certificados SSL en el servicio web antes de cualquier desarrollo web, especialmente si una aplicación va a interactuar con dispositivos GPS o cámaras.

Para el estudio de usabilidad se recomienda que la retroalimentación de un error debe ser lo más detallada posible ya que un juicio ambiguo puede representar una gran barrera para la aceptación de la plataforma.

A pesar de que los estudios de usabilidad dieron resultados favorables y la tendencia mostro resultados positivos en todas las categorías evaluadas, se recomienda mejorar la calidad de la información con mensajes claros y oportunos que le permitan a los usuarios superar un error fácilmente.

Así mismo se recomienda plantear un nuevo estudio de usabilidad que incorpore conceptos de tiempos de realización de tareas, promedios de efectividad del sistema, perfiles de usuario y tiempos de trabajo, para optimizar procesos de medición.

Para un estudio más eficiente se debería incorporar conceptos de tasa de error y tasa de efectividad, con manejadores de eventos que contabilicen tiempos de aciertos y equivocaciones de los usuarios con el sistema.

REFERENCIAS

- Anthes, G. (2012). *HTML5 leads a web revolution*. Recuperado de http://delivery.acm.org/10.1145/2210000/2209256/p16-anthes.pdf?ip=200.125.244.139&id=2209256&acc=OPEN&key=4D4702B0C3E38B35%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35%2E6D218144511F3437&__acm__=1541650630_7659fea4fe3e3a6a8c2f996088cb46d2
- Akirovska, N., & Koceski, S. (2015). *USAGE OF KANBAN METHODOLOGY AT SOFTWARE DEVELOPMENT TEAMS*.
- AngularJS . (2018). *Angular*. Recuperado de <https://docs.angularjs.org/guide/introduction>
- Beck, K., Beedle, M., Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., . . . Thomas, D. (2001). Manifiesto por el Desarrollo Ágil de Software. Recuperado de <http://agilemanifesto.org/iso/es/manifiesto.html>
- Benedek, J., & Miner, T. (2002). *Measuring Desirability: New methods for evaluating desirability in a usability lab setting*. Recuperado de <https://www.microsoft.com/usability/uepostings/desirabilitytoolkit.pdf>
- Bootstrap. (2018). *Getbootstrap*. Recuperado de <https://getbootstrap.com/docs/4.1/components/alerts/>
- GAD Municipal Pimampiro. (2018). BASE LEGAL QUE RIGE LA OPERACIÓN DEL GOBIERNO AUTÓNOMO DESCENTRALIZADO MUNICIPAL DE SAN PEDRO DE PIMAMPIRO. Pimampiro, Imbabura, Ecuador. Recuperado de [ttp://www.pimampiro.gob.ec/municipio-sp-2384/base-legal.html](http://www.pimampiro.gob.ec/municipio-sp-2384/base-legal.html)
- Geoconta. (2018). Medidores de agua. Recuperado de <http://www.geconta.net/sp/telelectura/redfija/redfija.html>
- GitHub. (2018). *Push git*. Recuperado de <https://guides.github.com/activities/hello-world/>
- Git-SCM. (2018). Control de versiones. Recuperado de <https://git-scm.com/>
- Google. (2018). *Google Developers*. Recuperado de <https://developers.google.com/web/fundamentals/codelabs/your-first-pwapp/?hl=es>
- Leaflet. (2018). *Leaflet* . Recuperado de <https://leafletjs.com/>
- Lewis, J. R. (Enero de 1995). *Ibm computer usability satisfaction questionnaires: Psychometric evaluation and instructions for use. VII*. Recuperado de <http://dx.doi.org/10.1080/10447319509526110>
- MarvelApp. (2017). *Marvel*. Recuperado de <https://marvelapp.com/about-us>
- Mozilla . (2018). *Developer Mozilla*. Recuperado de <https://developer.mozilla.org/bm/docs/Web/JavaScript>

- Mozilla. (2018). *MVC Architecture*. Recuperado de https://developer.mozilla.org/en-US/docs/Web/Apps/Fundamentals/Modern_web_app_architecture/MVC_architecture
- Mozilla. (2018). *MDN web docs*. Recuperado de <https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5>
- Muñoz, R. (19 de Junio de 2016). *Cantabriatic*. Recuperado de <http://www.cantabriatic.com/aprendiendo-tecnicas-de-desarrollo-agil/>
- Ngrok. (2018). *Ngrok*. Recuperado de <https://ngrok.com/product>
- NodeJs. (2018). *Node*. Recuperado de <https://nodejs.org/en/docs/>
- OpenSSL. (2018). *Open SSL*. Recuperado de <https://www.openssl.org/docs/>
- Perez Medina, J. L., & Vanderdonckt, J. (2018). *Sketching by Cross-Surface Collaboration*. Quito, Pichincha, Ecuador.
- PÉREZ MEDINA, J. L., GONZÁLEZ, M., PILCO, H. M., JIMENES1, K., ACOSTA VARGAS, P., SANCHEZ GORDON, S., . . . RYBARCZYK, Y. (2018). *Usability study of a Web-Based Platform for Home Motor Rehabilitation*.
- Rosselló Villán, V. (3 de Octubre de 2018). Las metodologías ágiles más utilizadas y sus ventajas dentro de la empresa. Recuperado de <https://www.iebschool.com/blog/que-son-metodologias-agiles-agile-scrum/>
- SailsJs. (2018). *Sails*. Recuperado de <https://sailsjs.com/documentation/reference>
- Schmich, C. (30 de 4 de 2017). *Github*. Recuperado de <https://github.com/schmich/instascan/blob/master/README.md>
- Steiner, T. (23 de Abril de 2018). *What is in a Web View? An Analysis of Progressive Web App Features When the Means of Web Access is not a Web Browser*. Lyon, Francia. Recuperado de <http://delivery.acm.org/10.1145/3190000/3188742/p789-steiner.pdf>
- Stewart, T. (1998). *Ergonomic requirements for office work with visual display terminals (vdts): Part 11: Guidance on usability*. 9241.
- TeamSQL. (2018). *TEAM SQL*. Recuperado de <https://teamsql.io/features/cloud>
- The PostgreSQL Global Development Group. (8 de 11 de 2018). PostgreSQL: la base de datos relacional de código abierto más avanzada del mundo. Recuperado de <https://www.postgresql.org/>
- Tullis, T. S., & Stetson, J. N. (2004). *A comparison of questionnaires for assessing website usability. I*.
- Vivify. (2018). *Vivify Scrum*. Recuperado de <https://vivifyscrum.com/features>
- VSC. (2018). *Visual Studio Code*. Recuperado de <https://code.visualstudio.com/>
- W3C. (2018). *w3.org*. Recuperado de <https://www.w3.org/standards/webdesign/htmlcss>

w3schools. (2018). *w3schools.com*. Recuperado de https://www.w3schools.com/html/html5_intro.asp

Wpbeginner. (2018). *Wpbeginner*. Recuperado de <https://www.wpbeginner.com/glossary/css/>

