

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**

**DECLARACIÓN DEL PROFESOR CORRECTOR**

**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

---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------.

William Marcillo.

DEDICATORIA

--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------.

William Marcillo.

**RESUMEN**

**ABSTRACT**

ÍNDICE

[INTRODUCCIÓN 1](#_Toc520477885)

[GAD Municipal Anfitrión 2](#_Toc520477886)

[Problemática 2](#_Toc520477887)

[Objetivo general 5](#_Toc520477888)

[Objetivos específicos 5](#_Toc520477889)

[1. CAPÍTULO I. MARCO TEÓRICO 6](#_Toc520477890)

* 1. [IDE 6](#_Toc520477891)

[1.1.1 Visual Studio Code 6](#_Toc520477893)

* 1. [Herramienta De Gestion de Proyecto 8](#_Toc520477894)

[1.2.1. VivifyScrum 8](#_Toc520477895)

* 1. [Herramienta de prototipado de interfaces 10](#_Toc520477898)

[1.3.1. Marvel 10](#_Toc520477900)

* 1. [Front End 11](#_Toc520477898)

[1.4.1. Diseño y Estilo 12](#_Toc520477902)

[1.4.1.1. HTML 12](#_Toc520477904)

[1.4.1.2. HTML5 13](#_Toc520477905)

[1.4.1.3. CSS 13](#_Toc520477906)

[1.4.1.4. CSS3 15](#_Toc520477907)

[1.4.1.5. Bootstrap 16](#_Toc520477908)

[1.4.2. Java Script 18](#_Toc520477910)

[1.4.2.1. Librerias JavaScript 20](#_Toc520477911)

[1.4.2.1.1. Angular 20](#_Toc520477912)

[1.4.2.1.2. Leaflet 21](#_Toc520477913)

[1.4.2.1.3 Draw 22](#_Toc520477914)

[1.4.2.1.4 Instascan 23](#_Toc520477914)

* 1. [Back-End 25](#_Toc520477915)

[1.5.1. Node Js 25](#_Toc520477916)

[1.5.2. Sails 26](#_Toc520477917)

* 1. [Base de Datos 28](#_Toc520477919)

[1.6.1. PostgreSQL 28](#_Toc520477920)

[1.6.2. TeamSQL 30](#_Toc520477921)

* 1. [Control de versiones de código fuente 31](#_Toc520477923)

[1.7.1. Git 31](#_Toc520477924)

[1.7.2. GitHub 32](#_Toc520477924)

* 1. [OpenSSL 33](#_Toc520477923)
  2. [Ngrok 34](#_Toc520477923)
  3. [Arquitectura 37](#_Toc520477925)

[1.10.1. Modelo Vista Controlador MVC 37](#_Toc520477926)

* 1. [Metodologias de desarrollo 39](#_Toc520477929)

[1.11.1. Metodologias de desarrollo ágil 39](#_Toc520477930)

[1.11.1.1. Metodologia Kanban 40](#_Toc520477911)

[1.11.1.2. Diseño centrado en el usuario 43](#_Toc520477911)

* 1. [Conclusión de capítulo 47](#_Toc520477931)

[2. CAPÍTULO II. ANÁLISIS Y DISEÑO 47](#_Toc520477933)

[2.1. Aplicaciones Web progresivas 47](#_Toc520477934)

[2.1.1. Shell de la App 48](#_Toc520477937)

[2.1.2. Service Workers 49](#_Toc520477937)

[2.1.3. Manifiesto de las aplicaciones web progresivas 50](#_Toc520477937)

[2.2. Especificación de los requerimientos 51](#_Toc520477935)

[2.2.1. Historias de usuario 52](#_Toc520477937)

[2.2.1.1. Gestión de lecturas 52](#_Toc520477911)

[2.2.1.2. Gestión de sectores 55](#_Toc520477911)

[2.2.1.3. Gestión de abonados 56](#_Toc520477911)

[2.2.1.4. Gestión de usuarios 56](#_Toc520477911)

[2.2.1.5. Gestión de Cuentas 56](#_Toc520477911)

[2.3. Diseño de Arquitectura 57](#_Toc520477935)

[2.3.1. Diagrama lógico de la base de datos 57](#_Toc520477937)

[2.3.2. Diagrama de arquitectura 58](#_Toc520477938)

[2.3.3. Diagrama de componentes 60](#_Toc520477938)

[2.4. Conclusión de Capítulo 61](#_Toc520477939)

[3. CAPÍTULO III. No definido 47](#_Toc520477941)

[3.1. ND 1 47](#_Toc520477942)

[3.1.1. CAT-001 47](#_Toc520477943)

[3.1.9. MAP-008 57](#_Toc520477951)

[3.2. ND 2 59](#_Toc520477952)

[3.2.1. MAP-009 59](#_Toc520477953)

[3.2.9. MAP-013 69](#_Toc520477961)

[3.3. ND 3 70](#_Toc520477962)

[3.3.1. CAT-003 70](#_Toc520477963)

[3.3.4. ATR-003 74](#_Toc520477966)

[3.4. Conclusión de capítulo 79](#_Toc520477970)

[4. CAPÍTULO IV. CASO DE PRUEBAS 80](#_Toc520477972)

[4.1.1. CP\_MAP\_001 81](#_Toc520477973)

[4.1.6. CP\_MAP\_006 86](#_Toc520477978)

[4.2. Conclusión de capítulo 104](#_Toc520477996)

[5. CONCLUSIONES Y RECOMENDACIONES 105](#_Toc520477997)

[5.1. Conclusiones 105](#_Toc520477998)

[5.2. Recomendaciones 106](#_Toc520477999)

REFERENCIAS…………………………………………………………….. 107

# Introducción

A finales del año 2015 google nombró como Aplicaciones Web progresivas, en ingles 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 georeferenciado 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 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 georrefenciados.

## 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.

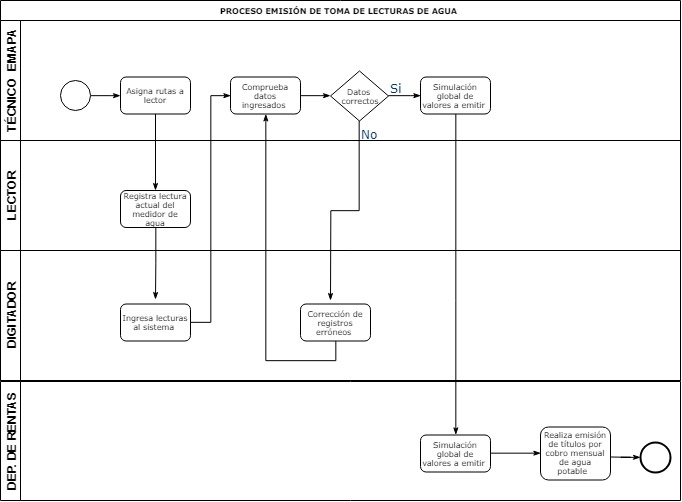
El proceso actual es deficiente porque la recolección de datos de lecturas retrasa la emisión global de planillas hasta que los digitadores concluyan el ingreso manual al sistema. Al mismo tiempo se genera la probabilidad de errores de digitación y perdida o deterioro de la hoja de ruta. Finalmente,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

Figura 1:Proceso de toma de lecturas de agua para emisión.

Actualmente existen sistemas automáticos de extracción de envió de datos de lecturas de agua potable comprendidos por hardware y software. Por ejemplo, la empresa española Geconta cuenta con medidores con envió 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.

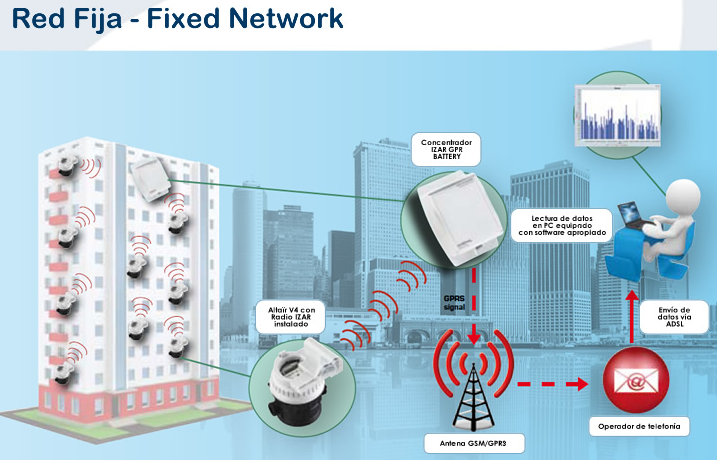


Figura 2: Medidores con red fija (Geoconta, 2018)

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.

Se espera que los resultados del presente estudio puedan ser aplicados a otros municipios del país.

## 

## Objetivo general

Desarrollar un sistema web Progresivo que permita mejorar el proceso 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 geografía 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.

# 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.

## 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.

## 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 4 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.

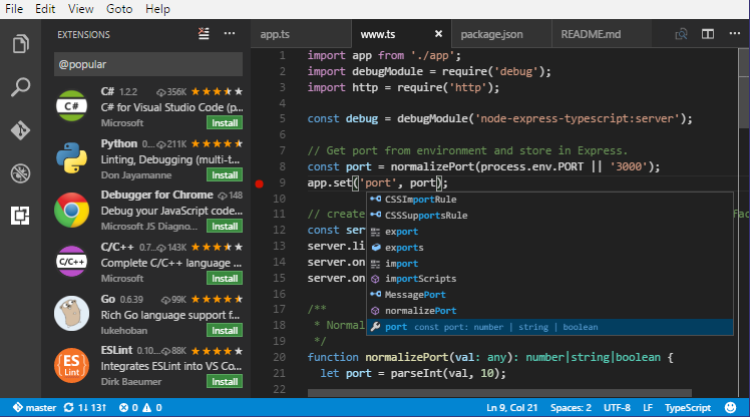


Figura 3: Interfaz gráfica de Visual studio code (VSC, 2018)

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.

## 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.

## 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.

## 

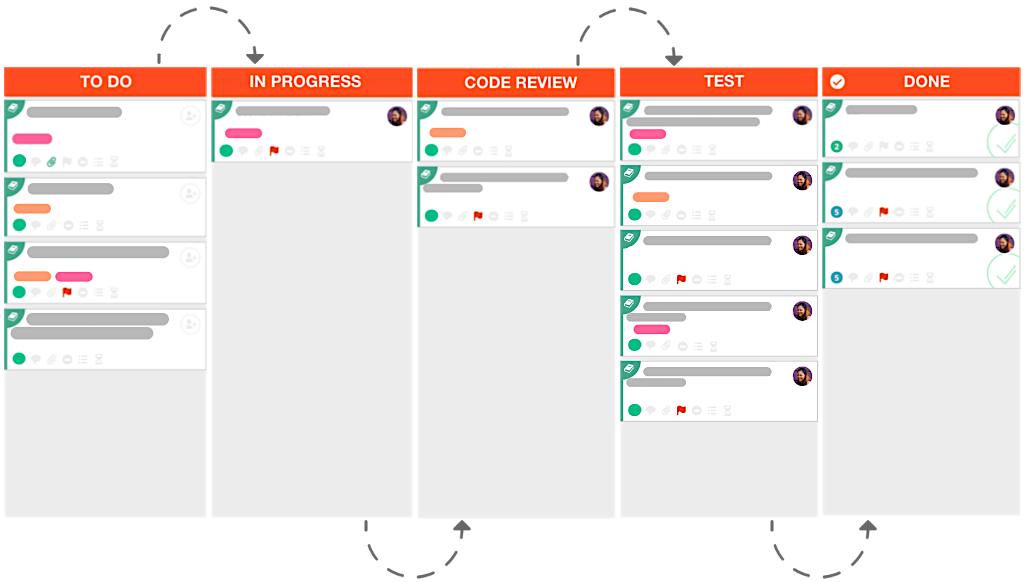


Figura 4: Tablero kaban de la herramienta vivifyScrum (Vivify, 2018)

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. **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.

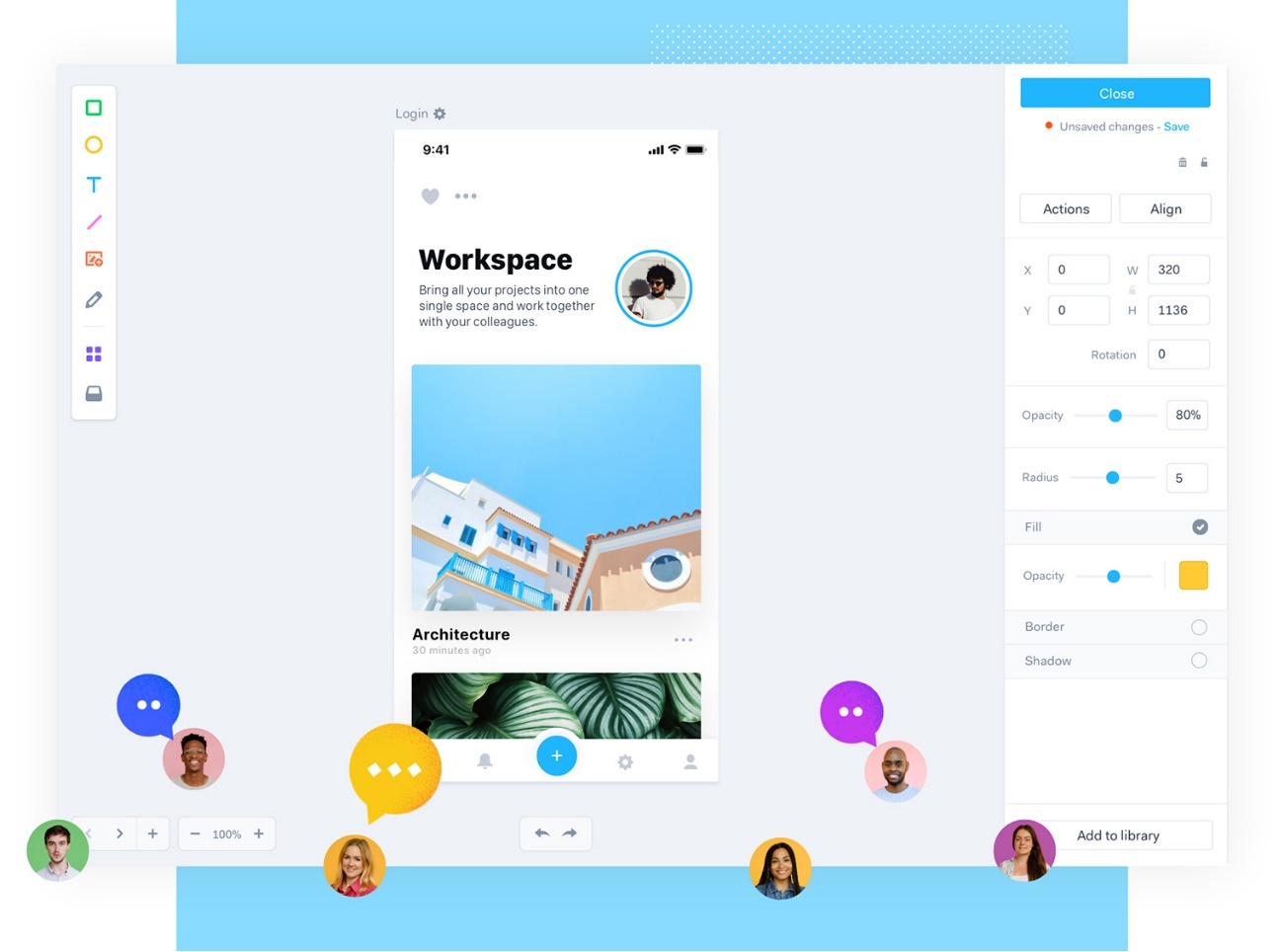


Figura 5: Canvas de diseño de interfaces Marvel App (MarvelApp, 2017)

**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 discutirlas 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.

## 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, el desarrollo Front-End

## Diseño y Estilo

## 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 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.

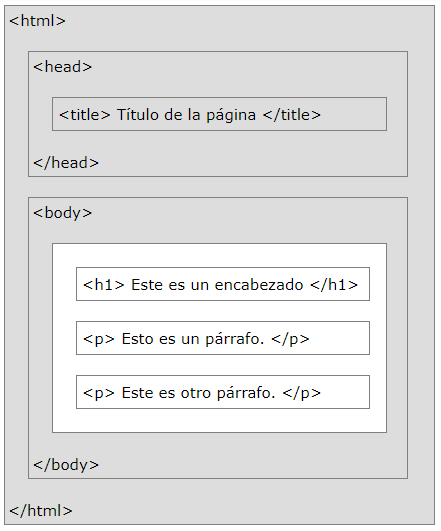


Figura 6: estructura básica y etiquetado HTML (Mozilla, 2018)

* + - 1. **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.

## 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.

## 

Figura 7: ejemplo CSS utilizado en este proyecto.

## 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 (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

## 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.

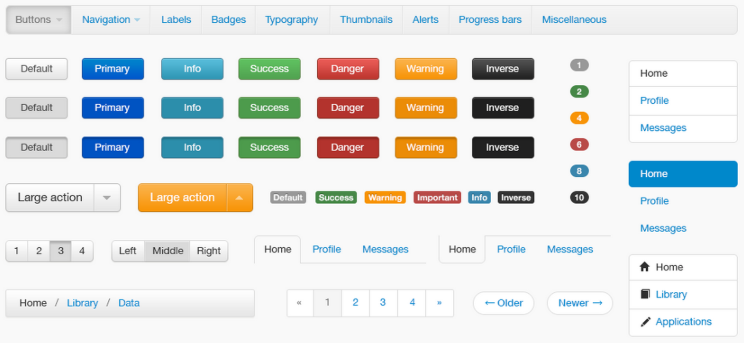


Figura 9: componentes Bootstrap (Bootstrap, 2018).

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

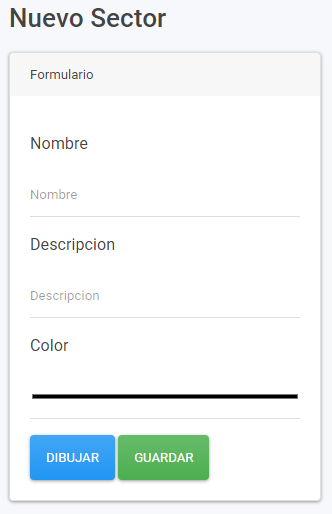


Figura 10: formulario Sectores

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 nanjea 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.

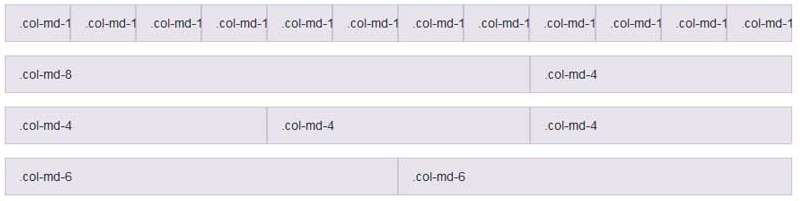


Figura 11: sistema de columnas Bootstrap (Bootstrap, 2018).

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.

## 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.

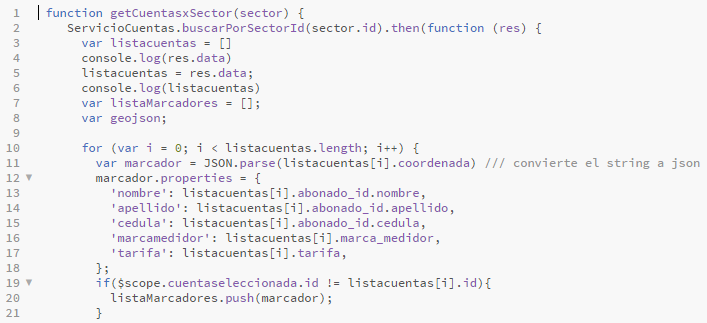


Figura 12: ejemplo código JavaScript implementado en este proyecto.

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.

## Librerías JavaScript

* + - * 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 especifica que ahora tiempo de programación. Por ejemplo, se puede implementar el ordenamiento de datos, se puede buscar elementos de una tabla, manejar repeticiones para extraes 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. **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.

## 

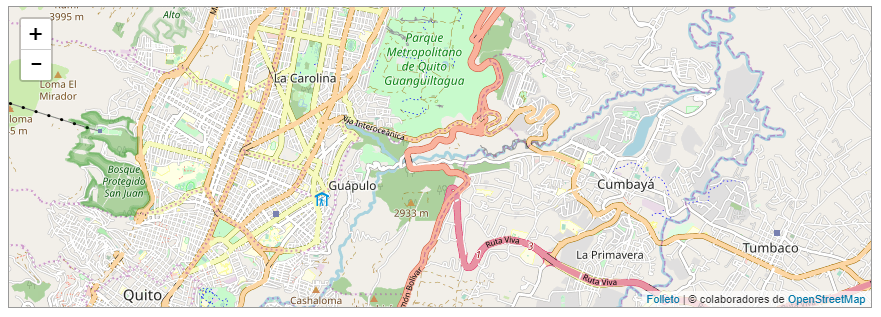


Figura 13: Mapa de la Ciudad de quito con Leaflet implementado en el mapa general de este proyecto.



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

## 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.

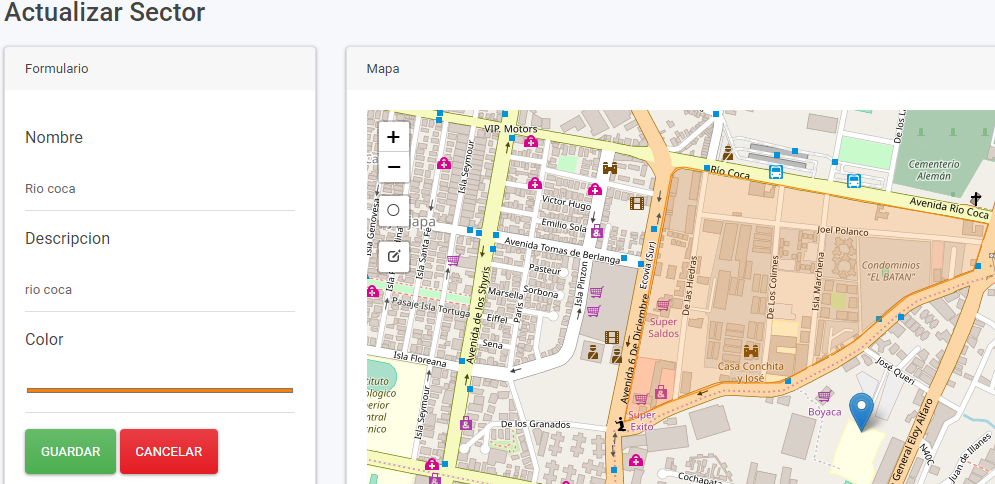


Figura 15: Dibujo del polígono del sector Rio Coca con Leaflet Draw.

* + - * 1. **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 sobre HTML.

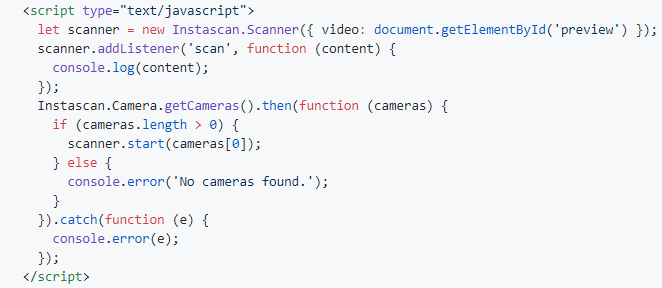


Figura 16: código de implementación (Schmich, 2017)

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

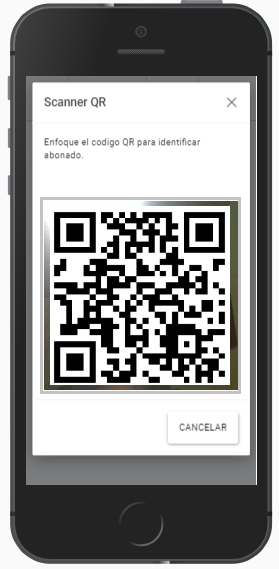


Figura 17: Modo de funcionamiento de Instascan para identificar abonados.

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 camaras.

## BACK-END

## 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.

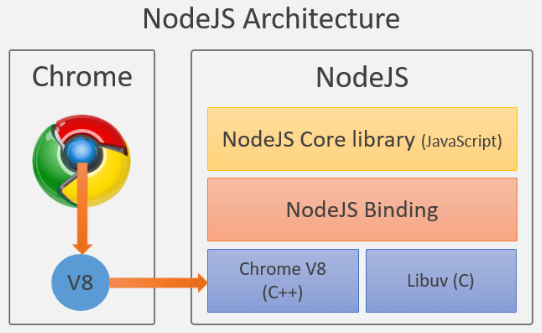


Figura 18: Arquitectura NodeJs (NodeJs, 2018).

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

## 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.

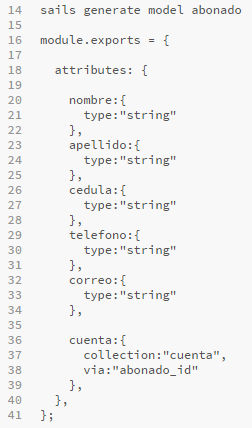


Figura 19: creación de entidad y atributos de la tabla abonado.

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.

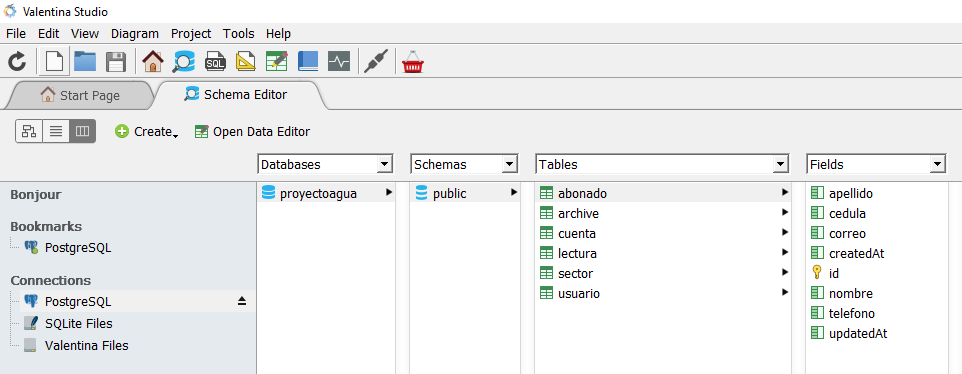


Figura 20: comprobación de creación de entidades a partir de Sails.

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.

## Base de Datos

## 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.

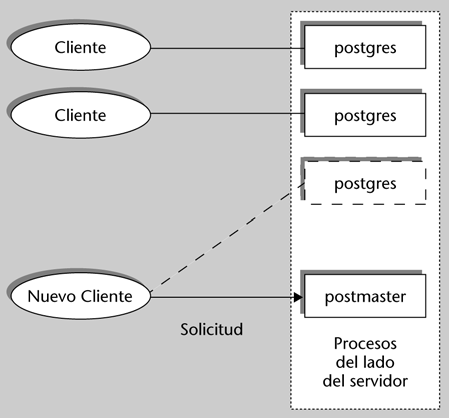


Figura 21: PostgreSQL, Arquitectura cliente servidor (The PostgreSQL Global Development Group, 2018).

**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 leguajes 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 opto 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.

## TeamSQL

TeamSQL es un cliente multiplataforma para la gestión de bases 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.

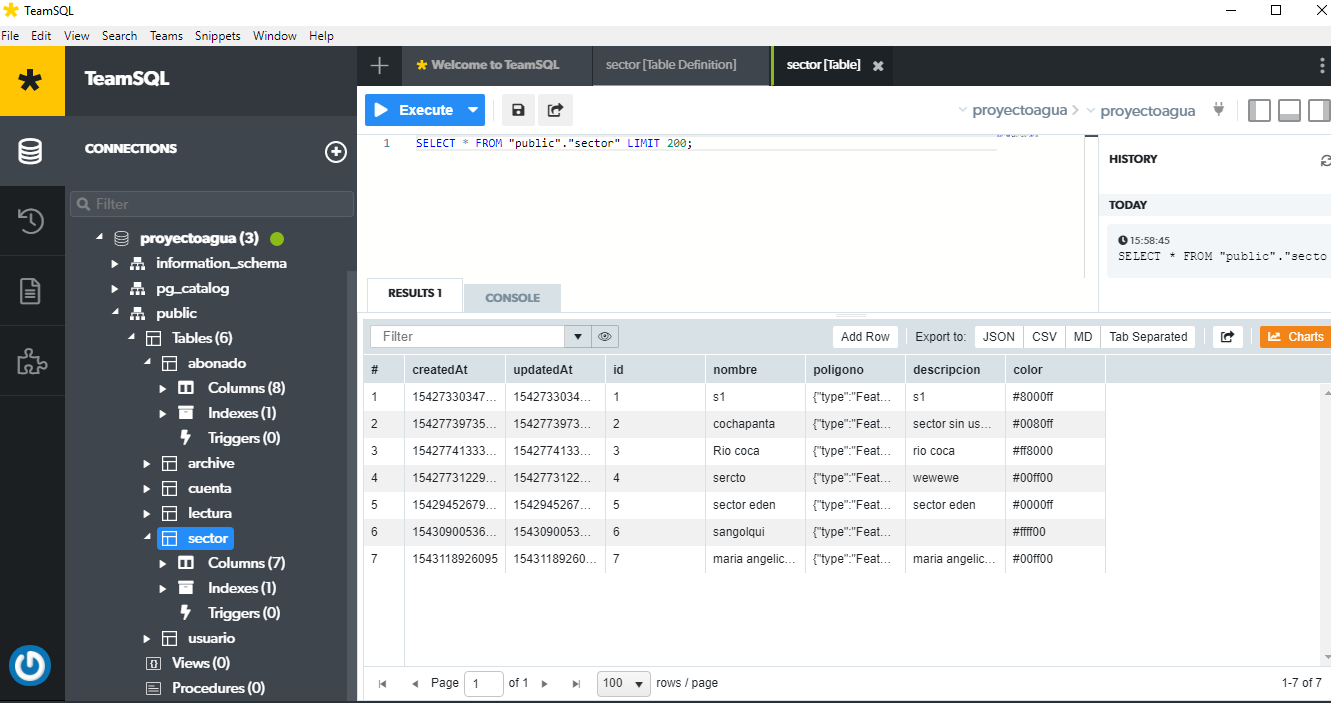


Figura 22: Consulta de la tabla lector mediante la interfaz gráfica TeamSQL.

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.

## 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.

## **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.

## 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 (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. **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.

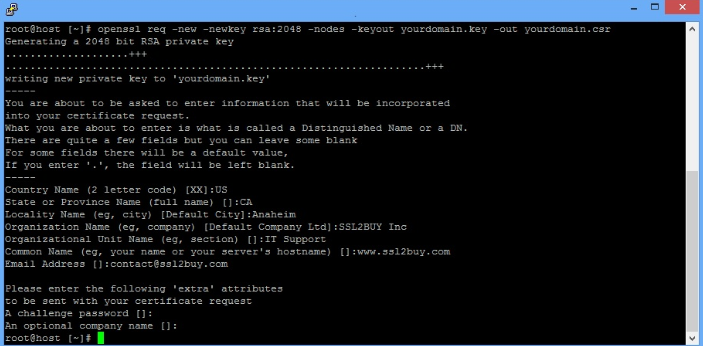


Figura 24: creación de certificado auto firmado (OpenSSL, 2018).

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. **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 (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. Ngrook se conecta al servicio en la nube y acepta el tráfico para exponerlo en una dirección publica que se ejecuta desde la dirección local que se especificó.

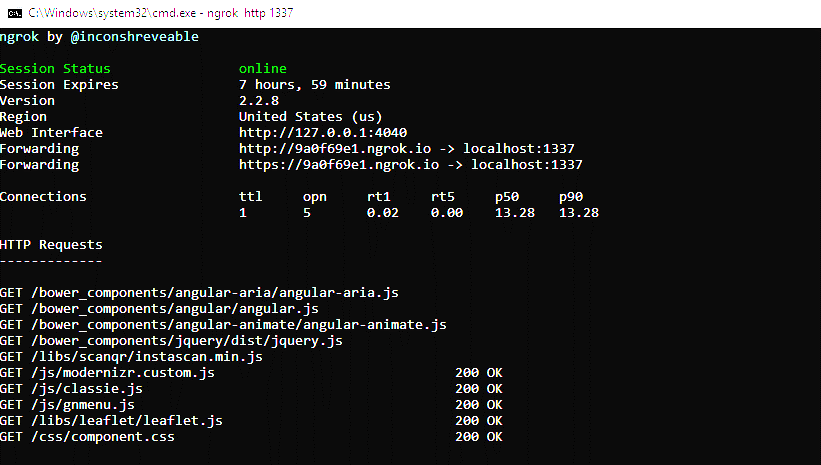


Figura 26: entorno de ejecución de ngrook.exe

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...”

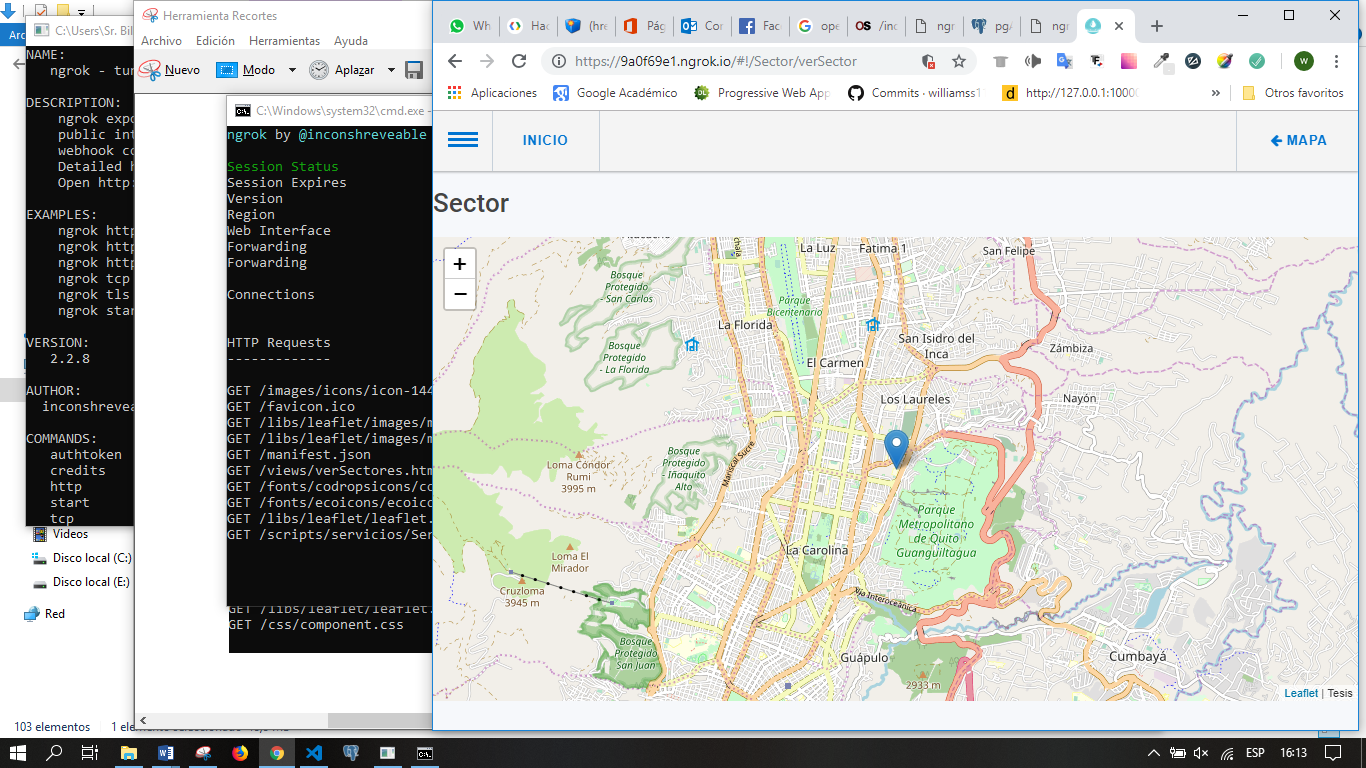


Figura 27: exposición del servicio local-Back-End a IP pública.

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 intenet 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

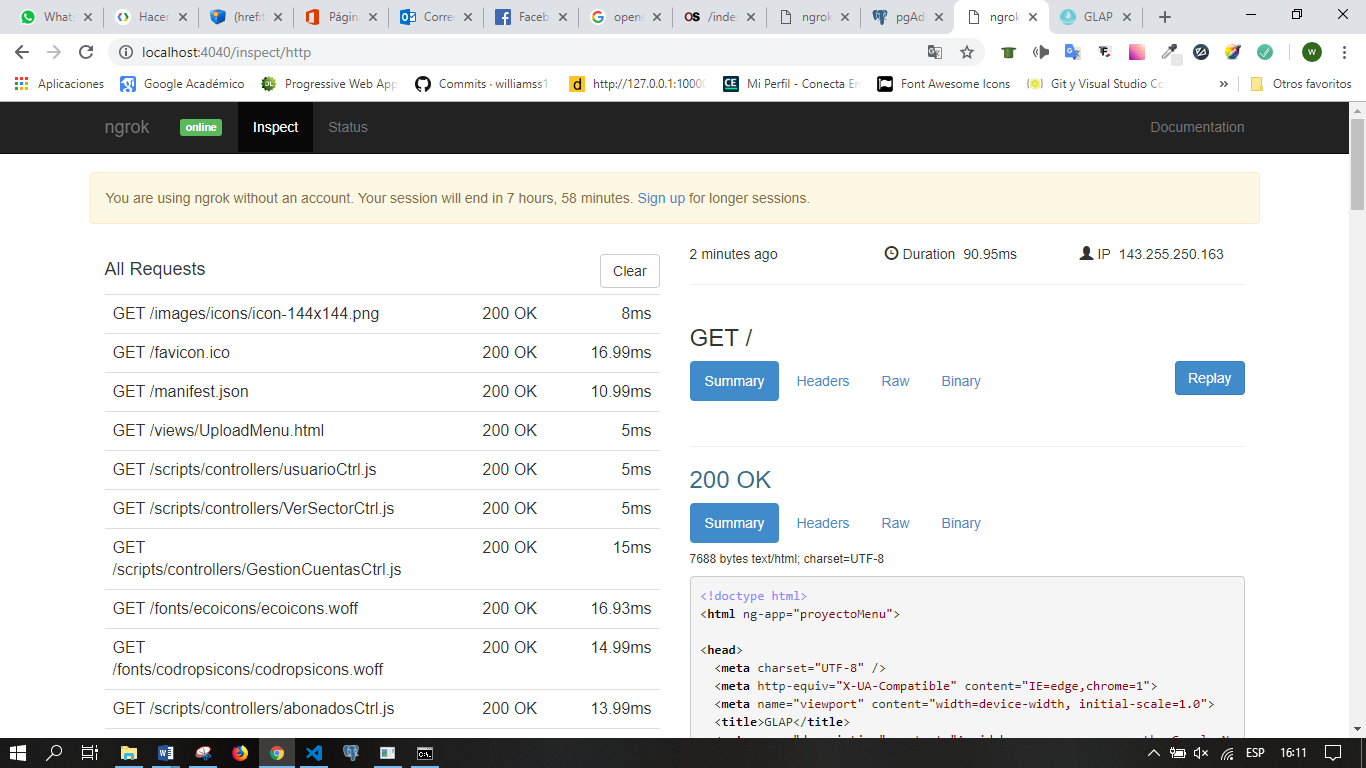


Figura 28: consola de entrada y salida de datos.

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.

## Arquitectura

## 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 aumentado 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:

## El usuario interactúa con la interfaz de usuario

## 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.

## El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario.

## El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario.

## La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo.

## 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.

## La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

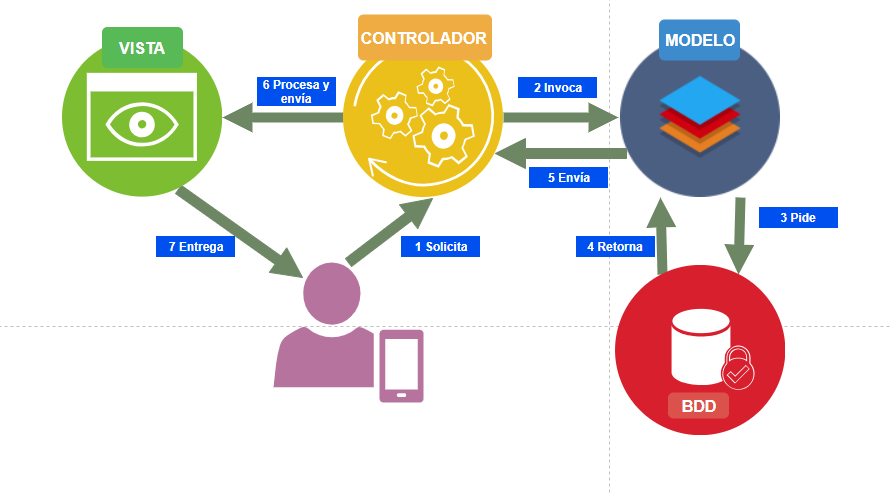


Figura 29: Esquema modelo vista controlador.

## 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. **Metodologías de desarrollo ágil**

En desarrollo de software, “las metodologías agiles 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 agiles 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.

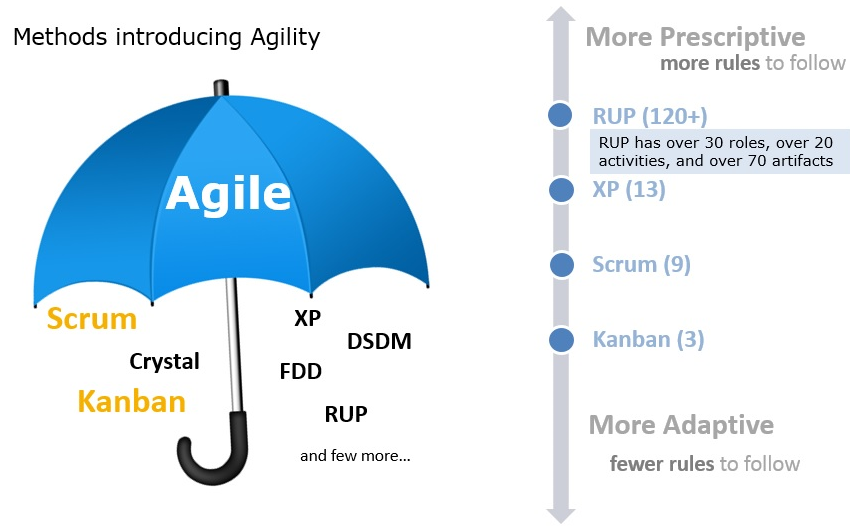


Figura 30:Principales metodologías agiles (Muñoz, 2016)

## 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 ( Kirovska & Koceski, 2015)

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

## Calidad perfecta

## Minimizar el despilfarro

## 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.

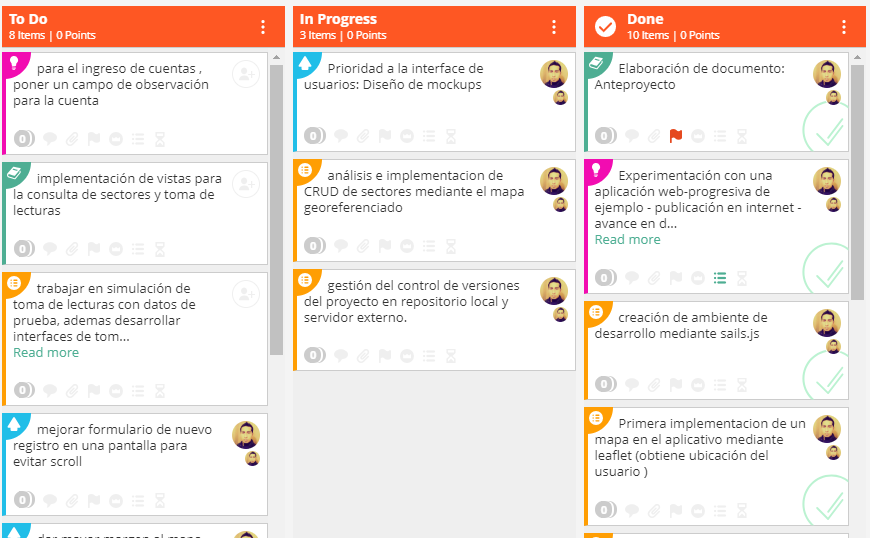


Figura 32: organización de tareas en Tablero Kanban de la herramienta VivifyScrum.

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. **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 (Perez 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.

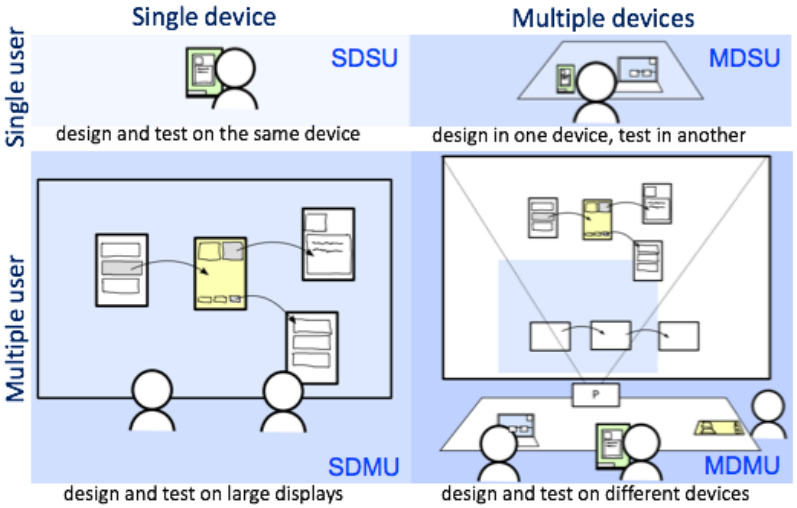


Figura 33 configuraciones físicas del diseño centrado en el usuario (Perez Medina & Vanderdonckt, 2018)

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:

* 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 a justes 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.



Figura 34: flujo de trabajo de prototipos de interfaces (Perez Medina & Vanderdonckt, 2018)

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 agiles. 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.

## 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.

# 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.

* 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.

Presentar Shell de la app-service workers y manifiesto

* + 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 35, muestra como el Shell de la app separa los componentes de la interfaz de usuario con la de los datos.

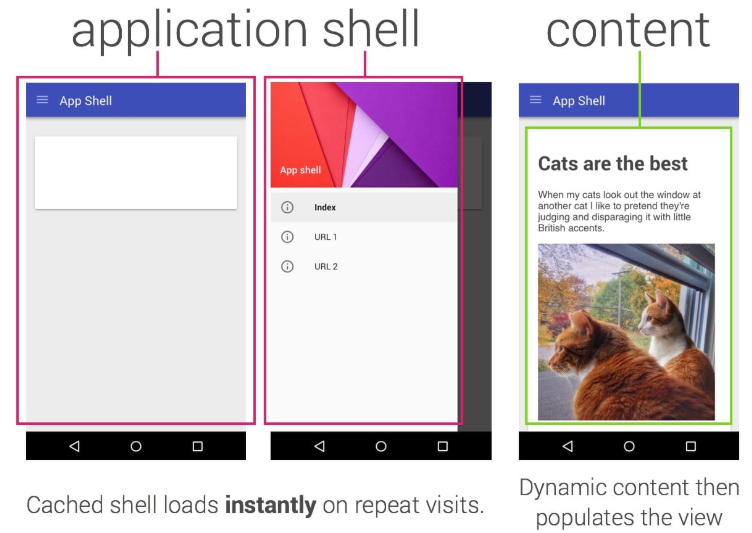


Figura 35: Shell de la aplicación web progresiva.

* + 1. **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 36 contiene una porción del código del service worker implementado en el proyecto.



Figura 36: Service worker implementado en el proyecto.

* + 1. **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 37 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.



Figura 37: 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.

## 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 (mock ups), quienes actúan como un medio de recopilación de requerimientos, se han obtenido juntamente con los usuarios y el autor de este proyecto.

## En la figura 38, 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.

## 

Figura 38: Diseño centrado en el usuario, Refinamiento de interfaces.

## 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 38, es un ejemplo lo que sucede cuando se da clic en un botón del sistema.

## Al mismo tiempo, también se han registrado las tareas en el tablero Kanban en la herramienta VivifyScrum como lo presenta la figura 40.

## Finalmente se listarán los requerimientos mediante historias de usuario.

## 

## 

Figura 39: Comportamiento de interfaces.

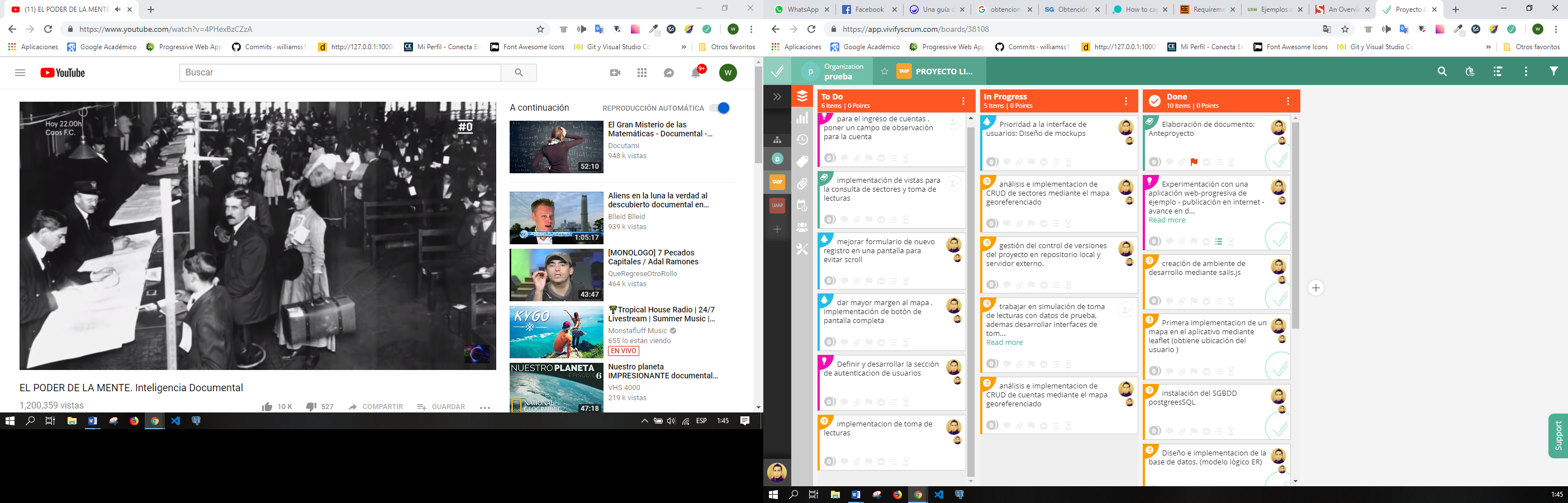


Figura 40. Ajuste de requerimientos en tablero Kanban.

## 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.

## 

## Gestión de lecturas

|  |  |
| --- | --- |
| **Usuario:** Lector **Acceso:** T1 | **Prototipo:** login |
| **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. |  |

|  |  |
| --- | --- |
| **Usuario:** Lector **Acceso:** T1 | **Prototipo:** Mapa |
| **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. |

|  |  |
| --- | --- |
| **Usuario:** lector **Acceso:** T1 | **Prototipo:** Identificar usuario |
| **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. |  |

|  |  |
| --- | --- |
| **Usuario:** Lector **Acceso:** T1 | **Prototipo:** Ingreso Lectura |
| **Tarjeta:** LEC-06**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. |  |
| **Tarjeta:** LEC-07**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. |
| **Tarjeta:** LEC-08**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. |

* + - 1. **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. |

* + - 1. **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. |

* + - 1. **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. |
| **Id:** ADM-10**Descripción: Yo como** director de planificación de procesos EMAPA, **quiero** consultar una lista de usuarios, **para** poder borrarlos del sistema. |

* + - 1. **Gestión de cuentas**

|  |
| --- |
| **Usuario:** Administrador **Acceso:** T2 |
|  |
| **Tarjeta:** ADM-11**Descripción: Yo como** director de planificación de procesos EMAPA, **quiero** ingresar los datos de una cuenta, **para** vincularlos con un abonado existente. |
| **Tarjeta:** ADM-12**Descripción: Yo como** director de planificación de procesos EMAPA, **quiero** ingresar los datos de una cuenta, **para** vincularlos con un sector existente. |
| **Tarjeta:** ADM-13**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. |
| **Tarjeta:** ADM-14**Descripción: Yo como** director de planificación de procesos EMAPA, **quiero** consultar una lista de las cuentas, **para** poder realizar operaciones de edición. |
| **Tarjeta:** ADM-15**Descripción: Yo como** director de planificación de procesos EMAPA, **quiero** consultar una lista de las cuentas, **para** poder borrarlas del sistema. |

## 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.

## Diagrama lógico de la de Base de Datos

La figura 41, 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 registe 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.

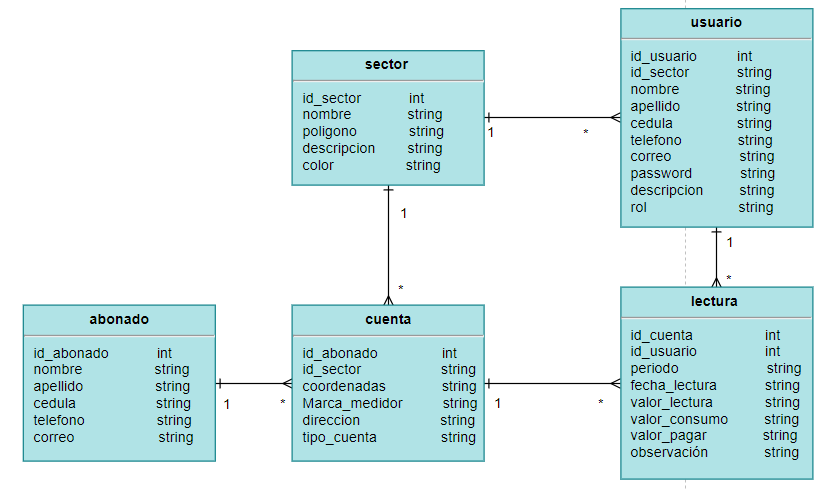


Figura 41: diagrama lógico de la base de datos.

## Diagrama de Arquitectura

La arquitectura que presenta la figura 42, 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 envió 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.

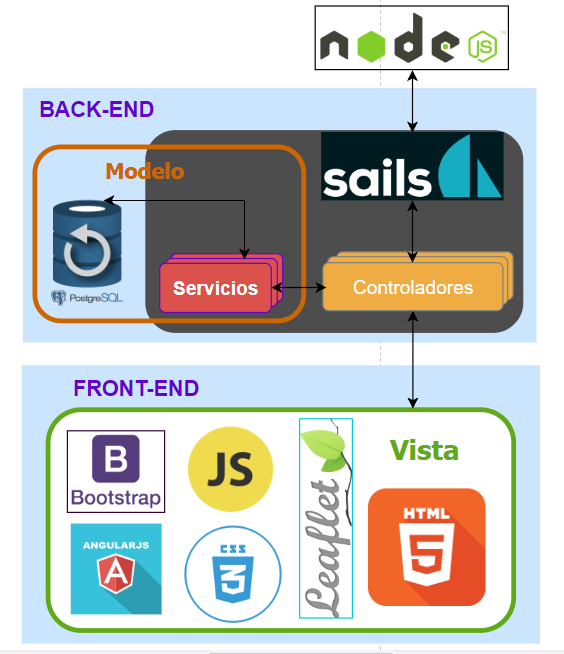


Figura 42: Arquitectura del sistema web progresivo para la toma de lecturas de agua potable

* + 1. **Diagrama de Componentes**

El diagrama de componentes de la figura 43, 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 salís 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.

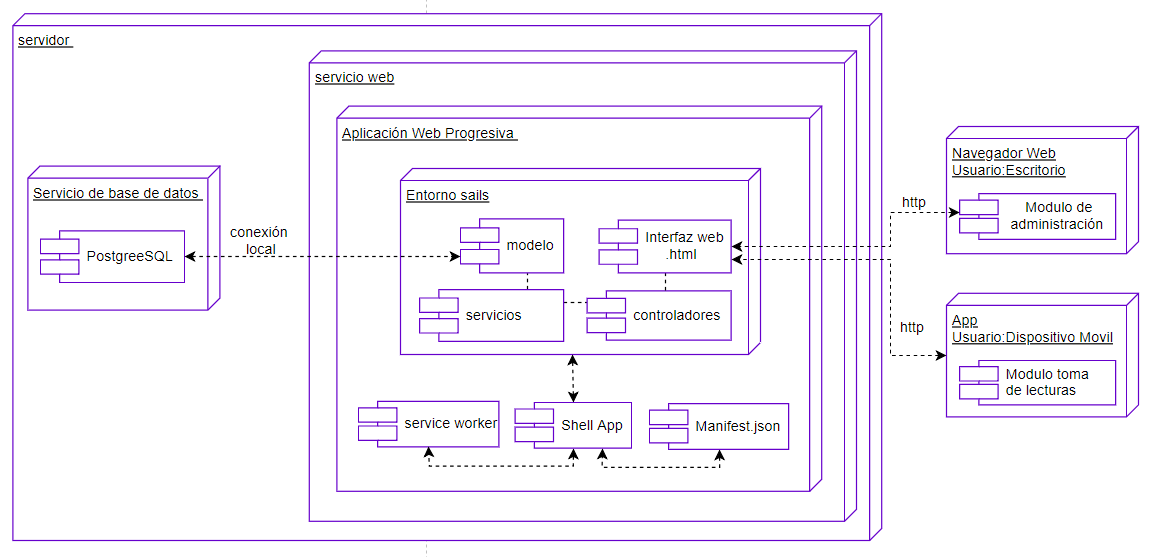


Figura 43: diagrama de componentes des sistema web progresivo para la toma de lecturas de agua potable.

## 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.

# CAPÍTULO III. DESARROLLO Y CODIFICACION

# 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”.

# 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.

# Tarjeta: 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 | | * 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. | | | | |
| Interfaz | | | | |
|  | | | | |

# Tarjeta: 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 | | * 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. | | | | |
| Interfaz | | | | |
|  | | | | |

# Tarjeta: 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 | | * 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. | | | | |
| Interfaz | | | | |
|  | | | | |

# Tarjeta: 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 | | * 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. | | | | |
| Interfaz | | | | |
|  | | | | |

# Tarjeta: 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 | | * 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. | | | | |
| Interfaz | | | | |
|  | | | | |

# Tarjeta: 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 | | * 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. | | | | |
| Interfaz | | | | |
|  | | | | |

# Tarjeta: 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 | | * 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. | | | | |
| Interfaz | | | | |
|  | | | | |

# Tarjeta: 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 | | * 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. | | | | |
| Interfaz | | | | |
|  | | | | |

# Tarjeta: 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 | | * 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 dibujar 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. | | | | |
| Interfaz | | | | |
|  | | | | |

# Tarjeta: 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 | | * 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. | | | | |
| Interfaz | | | | |
|  | | | | |

# Tarjeta: 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 | | * obtenerSectores(); * BorrarSector(); | |
| Modelo | sector | |
| Vista | GestionSectores.html | |
| Controlador | GestionSectores.js | |
| Implementación | | | | |
| Para borra 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. | | | | |
| Interfaz | | | | |
|  | | | | |

# Tarjeta: 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 | | * 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. | | | | |
| Interfaz | | | | |
|  | | | | |

# Tarjeta: 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 | | * 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. | | | | |
| Interfaz | | | | |
|  | | | | |

# Tarjeta: 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 | | * 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. | | | | |
| Interfaz | | | | |
|  | | | | |

# Tarjeta: 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 | | * 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. | | | | |
| Interfaz | | | | |
|  | | | | |

# Tarjeta: 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 | | * 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. | | | | |
| Interfaz | | | | |
|  | | | | |

# Tarjeta: 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 | | * 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. | | | | |
| Interfaz | | | | |
|  | | | | |

# Tarjeta: 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 | | * 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. | | | | |
| Interfaz | | | | |
|  | | | | |

# Tarjeta: 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 | | * initMap(); * IngresarCuenta(); * obtenerSectores(); * 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. | | | | |
| Interfaz | | | | |
|  | | | | |

# Tarjeta: 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 | | * initMap(); * IngresarCuenta(); * obtenerSectores(); * 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. | | | | |
| Interfaz | | | | |
|  | | | | |

# Tarjeta: 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 | | * initMap(); * IngresarCuenta(); * obtenerSectores(); * 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. | | | | |
| Interfaz | | | | |
|  | | | | |

# Tarjeta: 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 | | * initMap(); * actualizarCuenta(); * obtenerSectores(); * obtenerAbonados(); * getCuentasxSector(); | |
| Modelo | Cuenta | |
| Vista | ActualizarCuenta.html | |
| Controlador | ActualizarCuentaCtrl.js | |
| Implementación | | | | |
| 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.  Con esa acción el marcador está disponible para moverse.  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”. | | | | |
| Interfaz | | | | |
|  | | | | |

# Tarjeta: ADM-15

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 | | * 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. | | | | |
| Interfaz | | | | |
|  | | | | |

# CAPÍTULO IV. ESTUDIO DE USABILIDAD DE USUARIO

## La experiencia de usuario o UX (User Experience) 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 busca analizar el nivel de experiencia 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 de usuario, generar estadísticas de análisis de usabilidad y estudiar la eficacia del uso de la aplicación web progresiva GLAP.

## Los resultados obtenidos facilitaran 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.

## Conclusión de capítulo

# CONCLUSIONES Y RECOMENDACIONES

## Conclusiones

## Recomendaciones

REFERENCIAS

Anthes, G. (2012). HTML5 leads a web revolution. Obtenido 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

Kirovska, N., & Koceski, S. (2015). USAGE OF KANBAN METHODOLOGY AT SOFTWARE DEVELOPMENT TEAMS.

AngularJS . (2018). Obtenido 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. Obtenido de http://agilemanifesto.org/iso/es/manifesto.html

Bootstrap. (2018). *Getbootstrap*. Obtenido 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. Obtenido de http://www.pimampiro.gob.ec/municipio-sp-2384/base-legal.html

Geoconta. (2018). *geconta.net*. Obtenido de http://www.geconta.net/sp/telelectura/redfija/redfija.html

GitHub. (2018). Obtenido de https://guides.github.com/activities/hello-world/

Git-SCM. (2018). Obtenido de https://git-scm.com/

Google. (2018). *Google Developers*. Obtenido de https://developers.google.com/web/fundamentals/codelabs/your-first-pwapp/?hl=es

Leaflet. (2018). *Leaflet* . Obtenido de https://leafletjs.com/

MarvelApp. (2017). *Marvel*. Obtenido de https://marvelapp.com/about-us

Mozilla . (2018). *Developer Mozilla*. Obtenido de https://developer.mozilla.org/bm/docs/Web/JavaScript

Mozilla. (2018). Obtenido de https://developer.mozilla.org/en-US/docs/Web/Apps/Fundamentals/Modern\_web\_app\_architecture/MVC\_architecture

Mozilla. (2018). *MDN web docs*. Obtenido de https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5

Muñoz, R. (19 de Junio de 2016). *Cantabriatic*. Obtenido de http://www.cantabriatic.com/aprendiendo-tecnicas-de-desarrollo-agil/

Ngrok. (2018). *Ngrok*. Obtenido de https://ngrok.com/product

NodeJs. (2018). *Node*. Obtenido de https://nodejs.org/en/docs/

OpenSSL. (2018). *Open SSl*. Obtenido de https://www.openssl.org/docs/

Perez Medina, J. L., & Vanderdonckt, J. (2018). Sketching by Cross-Surface Collaboration. Quito, Pichincha, Ecuador.

Rosselló Villán, V. (3 de Octubre de 2018). Las metodologías ágiles más utilizadas y sus ventajas dentro de la empresa. Obtenido de https://www.iebschool.com/blog/que-son-metodologias-agiles-agile-scrum/

SailsJs. (2018). *Sails*. Obtenido de https://sailsjs.com/documentation/reference

Schmich, C. (30 de 4 de 2017). *Github*. Obtenido 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. Obtenido de http://delivery.acm.org/10.1145/3190000/3188742/p789-steiner.pdf

TeamSQL. (2018). Obtenido 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. Obtenido de https://www.postgresql.org/

Vivify. (2018). *Vivify Scrum*. Obtenido de https://vivifyscrum.com/features

VSC. (2018). *Visual Studio Code*. Obtenido de https://code.visualstudio.com/

W3C. (2018). *w3.org*. Obtenido de https://www.w3.org/standards/webdesign/htmlcss

w3schools. (2018). *w3schools.com*. Obtenido de https://www.w3schools.com/html/html5\_intro.asp

Wpbeginner. (2018). *Wpbeginner*. Obtenido de https://www.wpbeginner.com/glossary/css/