

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)

[Empresa Anfitriona 2](#_Toc520477886)

[Problemática 2](#_Toc520477887)

[Objetivo general 3](#_Toc520477888)

[Objetivos específicos 3](#_Toc520477889)

[1. CAPÍTULO I. HERRAMIENTAS DE DESARROLO 4](#_Toc520477890)

* 1. [IDE 4](#_Toc520477891)

[1.1 Atom 4](#_Toc520477893)

* 1. [Gestor de Proyecto 6](#_Toc520477894)

[1.2.1. Trello 6](#_Toc520477895)

[1.2.2. Asana 8](#_Toc520477896)

* 1. [Front-End 9](#_Toc520477898)

[1.3.1. JavaScript 9](#_Toc520477900)

[1.3.2. Vue.js 11](#_Toc520477901)

[1.3.3. NPM 13](#_Toc520477902)

[1.3.4. Librerías de JavaScript 14](#_Toc520477903)

[1.3.4.1. JQuery 14](#_Toc520477904)

[1.3.4.2. Leaflet.js 15](#_Toc520477905)

[1.3.4.3. Marker Cluster 15](#_Toc520477906)

[1.3.4.4. Leaflet Draw 17](#_Toc520477907)

[1.3.4.5. DataTables 17](#_Toc520477908)

[1.3.4.6. Axios.js 19](#_Toc520477909)

[1.3.5. Diseño y Estilo 20](#_Toc520477910)

[1.3.5.1. HTML 20](#_Toc520477911)

[1.3.5.2. CSS3 21](#_Toc520477912)

[1.3.5.3. SASS 22](#_Toc520477913)

[1.3.5.4. Foundation Zurb 24](#_Toc520477914)

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

[1.4.1. Python 25](#_Toc520477916)

[1.4.2. Django 27](#_Toc520477917)

[1.4.3. GeoDjango 29](#_Toc520477918)

* 1. [Base de Datos 29](#_Toc520477919)

[1.5.1. PostgreSQL 29](#_Toc520477920)

[1.5.2. PostGIS 30](#_Toc520477921)

[1.5.3. PGAdmin 31](#_Toc520477922)

* 1. [Control de versiones de código 32](#_Toc520477923)

[1.6.1. Git 32](#_Toc520477924)

* 1. [Arquitectura 34](#_Toc520477925)

[1.7.1. N Capas 34](#_Toc520477926)

[1.7.2. Patrones de diseño 35](#_Toc520477927)

[1.7.2.1. Singleton 35](#_Toc520477928)

* 1. [Marco de Trabajo 35](#_Toc520477929)

[1.8.1. SCRUM 35](#_Toc520477930)

* 1. [Conclusión de capítulo 40](#_Toc520477931)

2.

[3. CAPÍTULO II. ANÁLISIS Y DISEÑO 41](#_Toc520477933)

[2.1. Product Backlog 41](#_Toc520477934)

[2.2. Diseño de Arquitectura 44](#_Toc520477935)

[2.2.1. Diagrama de Arquitectura 45](#_Toc520477937)

[2.2.2. Diagrama E-R de Base de Datos 45](#_Toc520477938)

[2.3. Conclusión de Capítulo 46](#_Toc520477939)

[3. CAPÍTULO III. DESARROLLO Y CODIFICACIÓN 47](#_Toc520477941)

[3.1. Sprint 1 47](#_Toc520477942)

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

[3.1.2. CAT-002 48](#_Toc520477944)

[3.1.3. MAP-001 50](#_Toc520477945)

[3.1.4. MAP-002 51](#_Toc520477946)

[3.1.5. MAP-003 52](#_Toc520477947)

[3.1.6. MAP-004 54](#_Toc520477948)

[3.1.7. MAP-006 55](#_Toc520477949)

[3.1.8. MAP-007 56](#_Toc520477950)

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

[3.2. Sprint 2 59](#_Toc520477952)

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

[3.2.2. MAP-011 60](#_Toc520477954)

[3.2.3. ATR-001 61](#_Toc520477955)

[3.2.4. ATR-002 63](#_Toc520477956)

[3.2.5. ATR-006 64](#_Toc520477957)

[3.2.6. ATR-007 65](#_Toc520477958)

[3.2.7. MAP-010 66](#_Toc520477959)

[3.2.8. MAP-012 68](#_Toc520477960)

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

[3.3. Sprint 3 70](#_Toc520477962)

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

[3.3.2. CAT-004 72](#_Toc520477964)

[3.3.3. CAT-005 73](#_Toc520477965)

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

[3.3.5.  ATR-004 75](#_Toc520477967)

[3.3.6. ATR-005 77](#_Toc520477968)

[3.3.7. MAP-005 78](#_Toc520477969)

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

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

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

[4.1.2. CP\_MAP\_002 82](#_Toc520477974)

[4.1.3. CP\_MAP\_003 83](#_Toc520477975)

[4.1.4. CP\_MAP\_004 84](#_Toc520477976)

[4.1.5. CP\_MAP\_005 85](#_Toc520477977)

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

[4.1.7. CP\_MAP\_007 87](#_Toc520477979)

[4.1.8. CP\_MAP\_008 88](#_Toc520477980)

[4.1.9. CP\_MAP\_009 89](#_Toc520477981)

[4.1.10. CP\_MAP\_010 90](#_Toc520477982)

[4.1.11. CP\_MAP\_012 91](#_Toc520477983)

[4.1.12. CP\_MAP\_013 92](#_Toc520477984)

[4.1.13. CP\_ATR\_001 93](#_Toc520477985)

[4.1.14. CP\_ATR\_002 94](#_Toc520477986)

[4.1.15. CP\_ATR\_003 96](#_Toc520477987)

[4.1.16. CP\_ATR\_004 96](#_Toc520477988)

[4.1.17. CP\_ATR\_005 97](#_Toc520477989)

[4.1.18. CP\_ATR\_006 98](#_Toc520477990)

[4.1.19. CP\_ATR\_007 99](#_Toc520477991)

[4.1.20. CP\_CAT\_002 100](#_Toc520477992)

[4.1.21. CP\_CAT\_003 101](#_Toc520477993)

[4.1.22. CP\_CAT\_004 102](#_Toc520477994)

[4.1.23. CP\_CAT\_005 102](#_Toc520477995)

[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 georefenciados.

## 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.

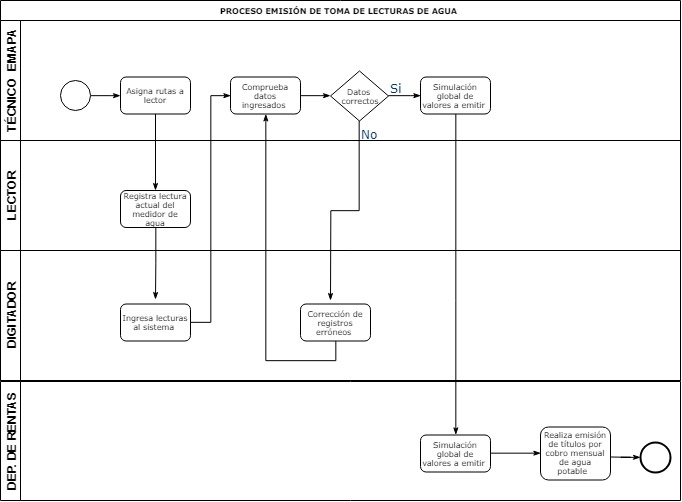
El proceso actual es deficiente porque la recolección de datos de lecturas retrasa la emisión global de planillas hasta que los digitadores concluyan el ingreso manual al sistema. Al mismo tiempo se genera la probabilidad de errores de digitación y perdida o deterioro de la hoja de ruta. Finalmente, 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.

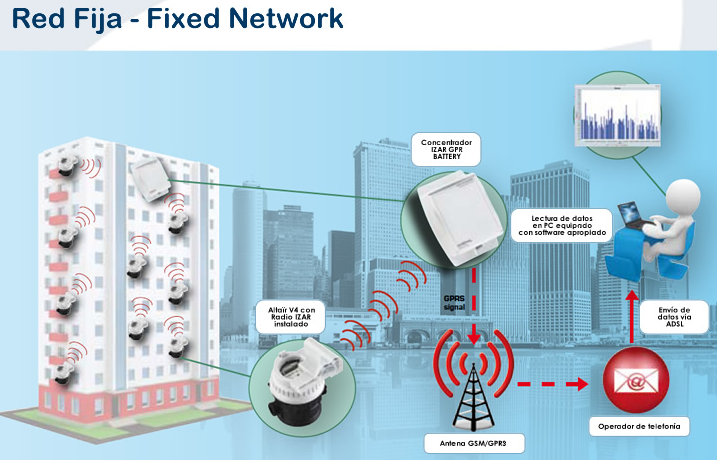


Figura 2: Medidores con red fija (Geoconta, 2018)

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.

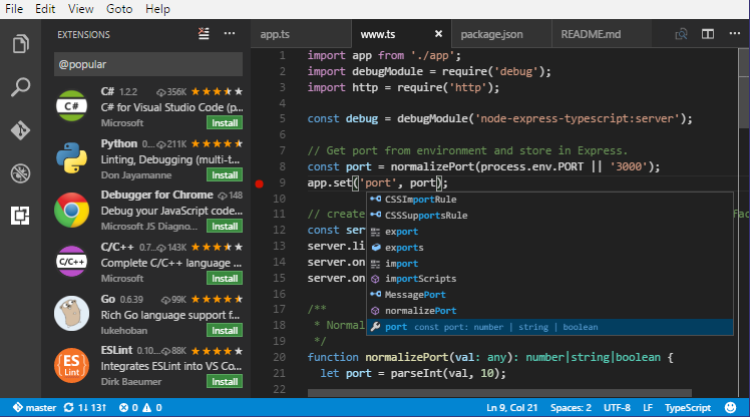


Figura 3: Interfaz gráfica de Visual studio code (VSC, 2018)

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.

## 

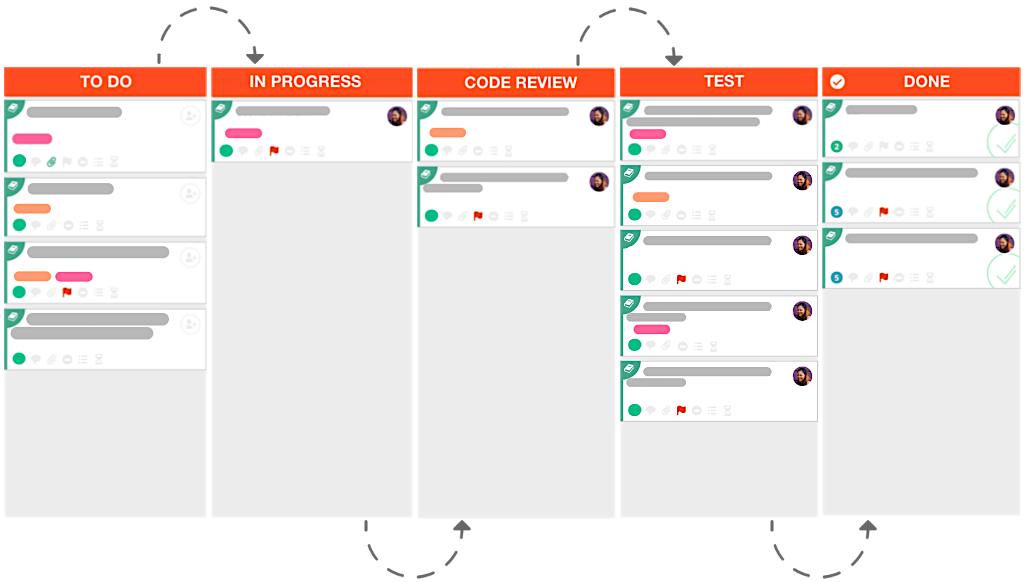


Figura 4: Tablero kaban de la herramienta vivifyScrum (Vivify, 2018)

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.

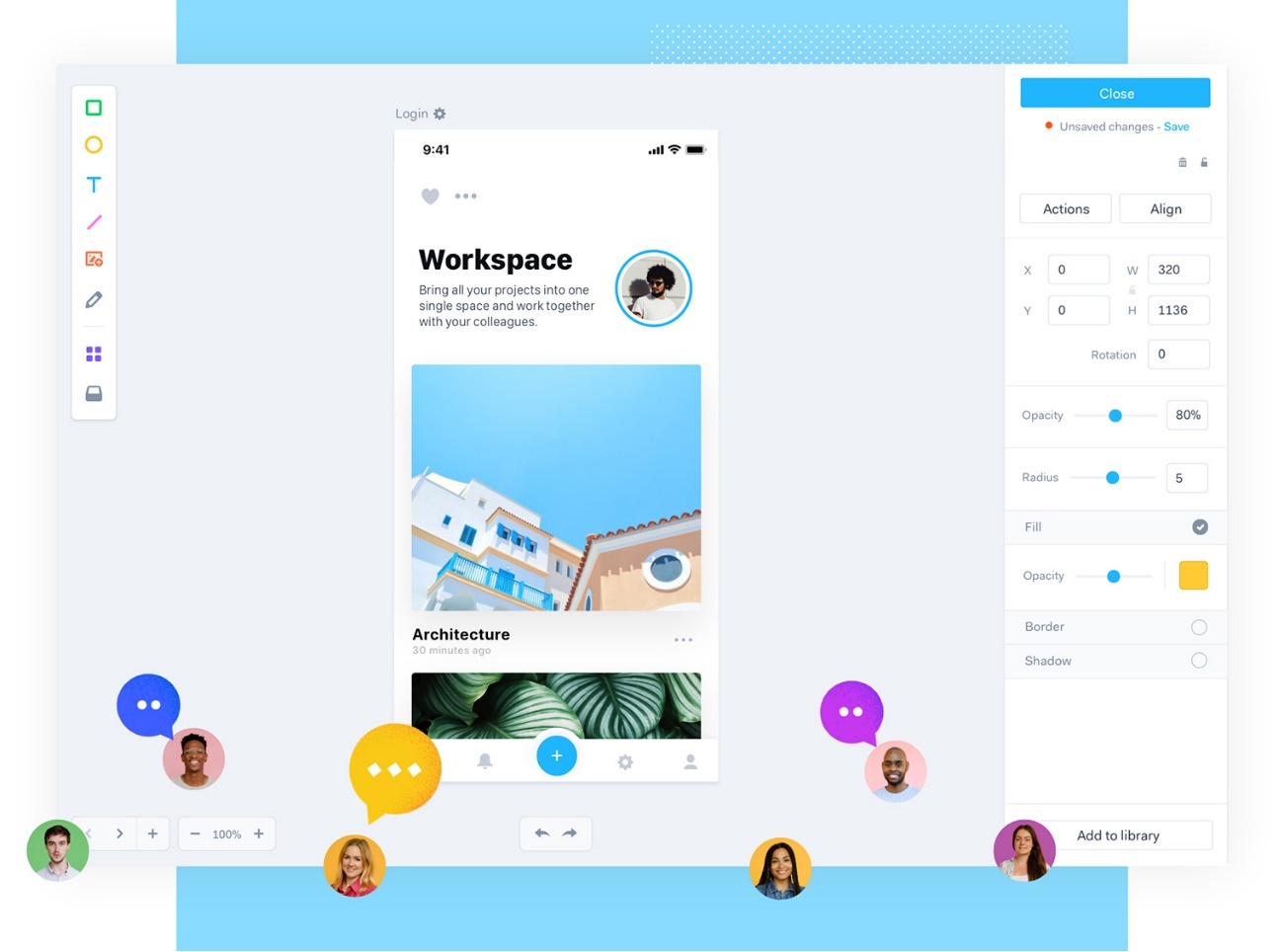


Figura 5: Canvas de diseño de interfaces Marvel App (MarvelApp, 2017)

**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.

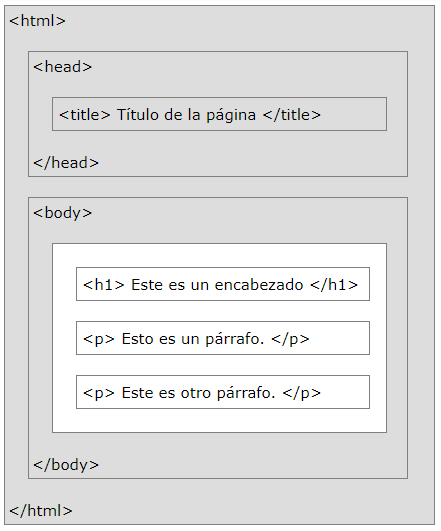


Figura 6: estructura básica y etiquetado HTML (Mozilla, 2018)

**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.

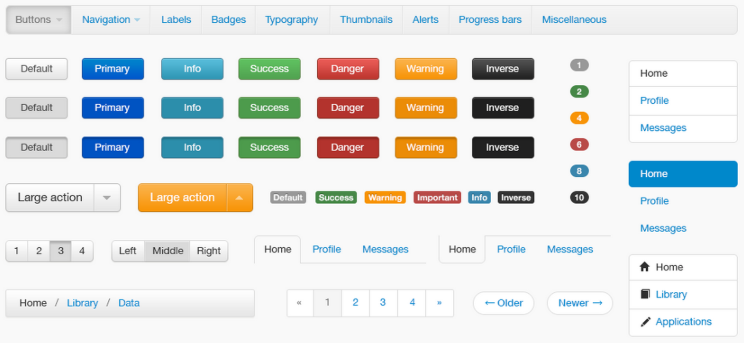


Figura 9: componentes Bootstrap (Bootstrap, 2018).

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

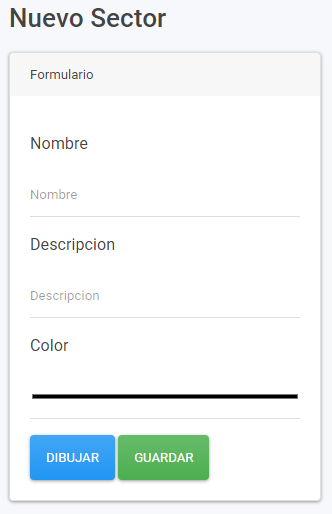


Figura 10: formulario Sectores

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.

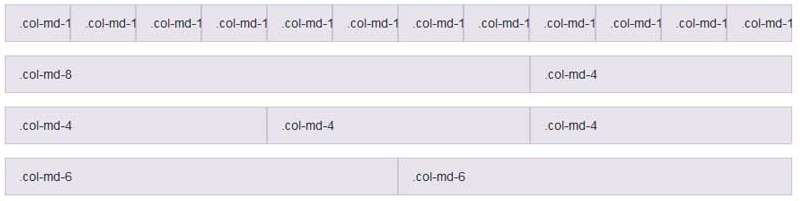


Figura 11: sistema de columnas Bootstrap (Bootstrap, 2018).

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.

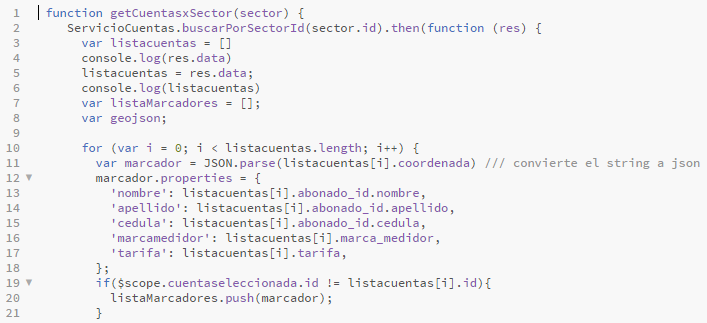


Figura 12: ejemplo código JavaScript implementado en este proyecto.

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 de JavaScript

**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.

**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.

## 

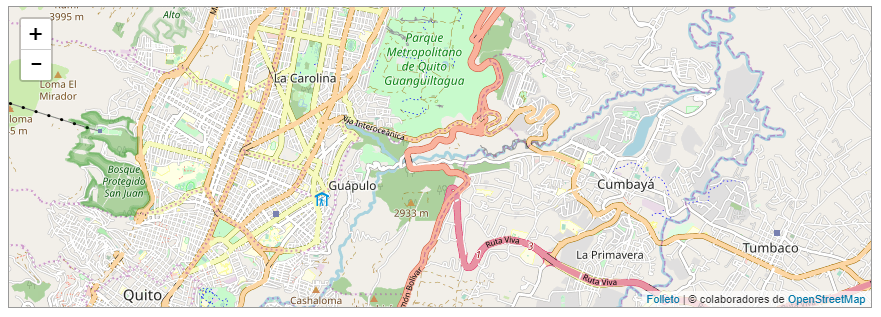


Figura 13: Mapa de la Ciudad de quito con Leaflet implementado en el mapa general de este proyecto.



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.

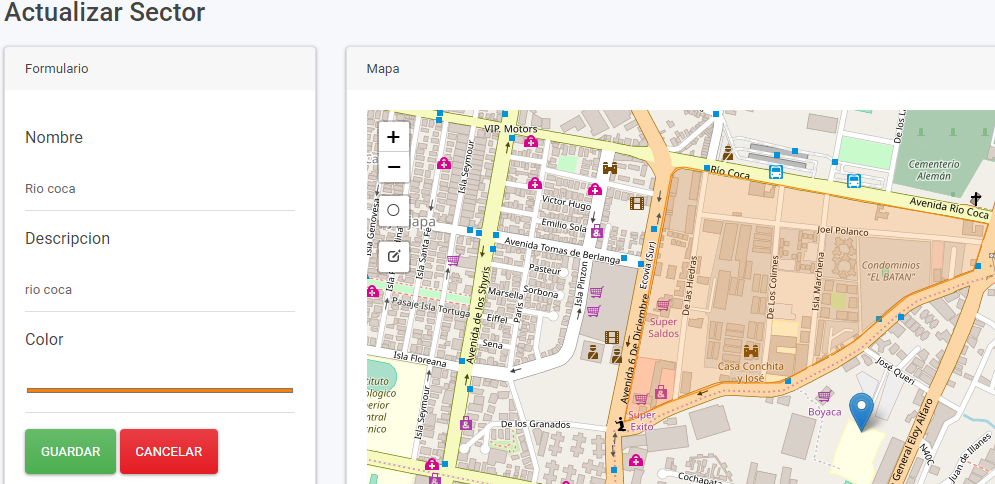


Figura 15: Dibujo del polígono del sector Rio Coca con Leaflet Draw.

**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.

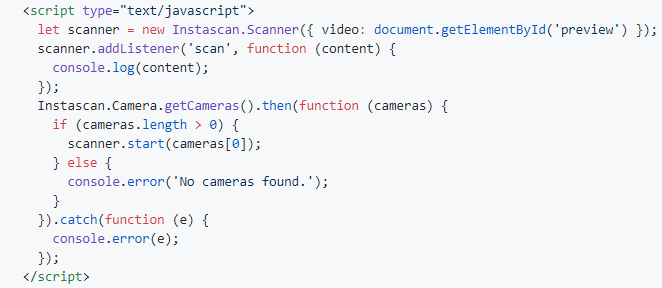


Figura 16: código de implementación (Schmich, 2017)

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

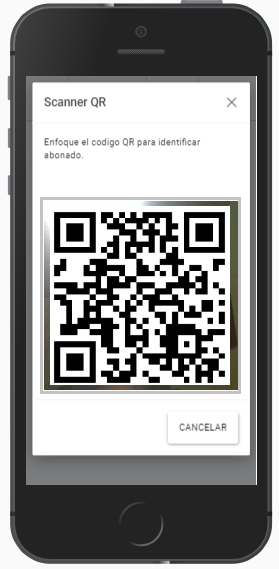


Figura 17: Modo de funcionamiento de Instascan para identificar abonados.

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.

## 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.

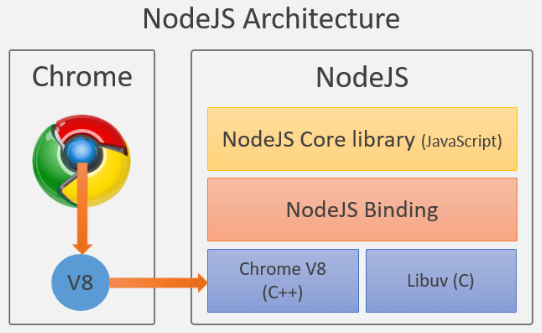


Figura 18: Arquitectura NodeJs (NodeJs, 2018).

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

## BACK-END

## 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.

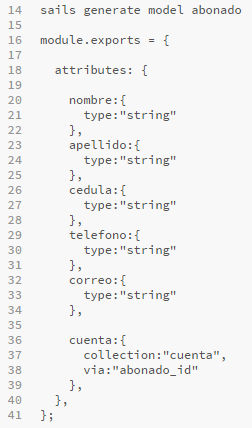


Figura 19: creación de entidad y atributos de la tabla abonado.

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.

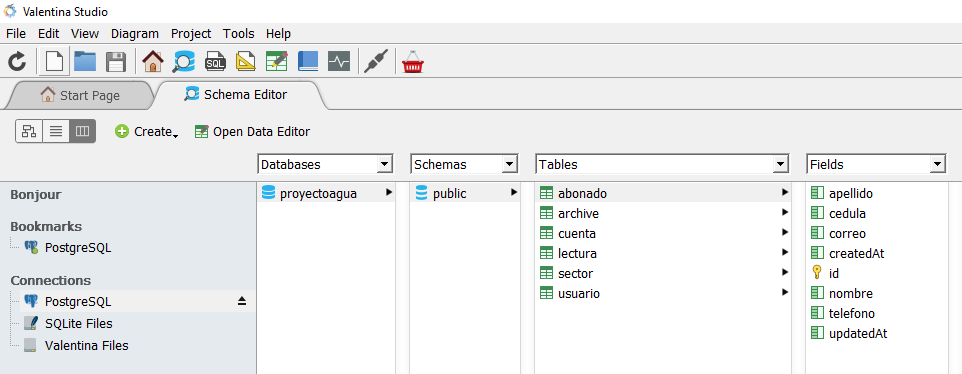


Figura 20: comprobación de creación de entidades a partir de Sails.

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.

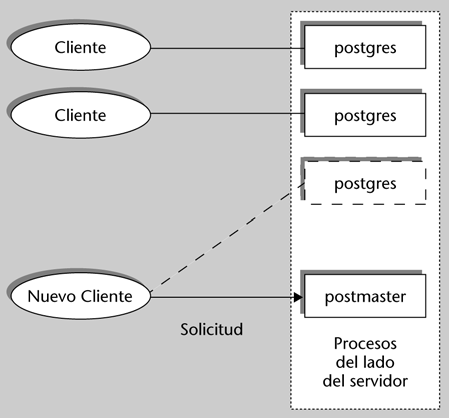


Figura 21: PostgreSQL, Arquitectura cliente servidor (The PostgreSQL Global Development Group, 2018).

**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.

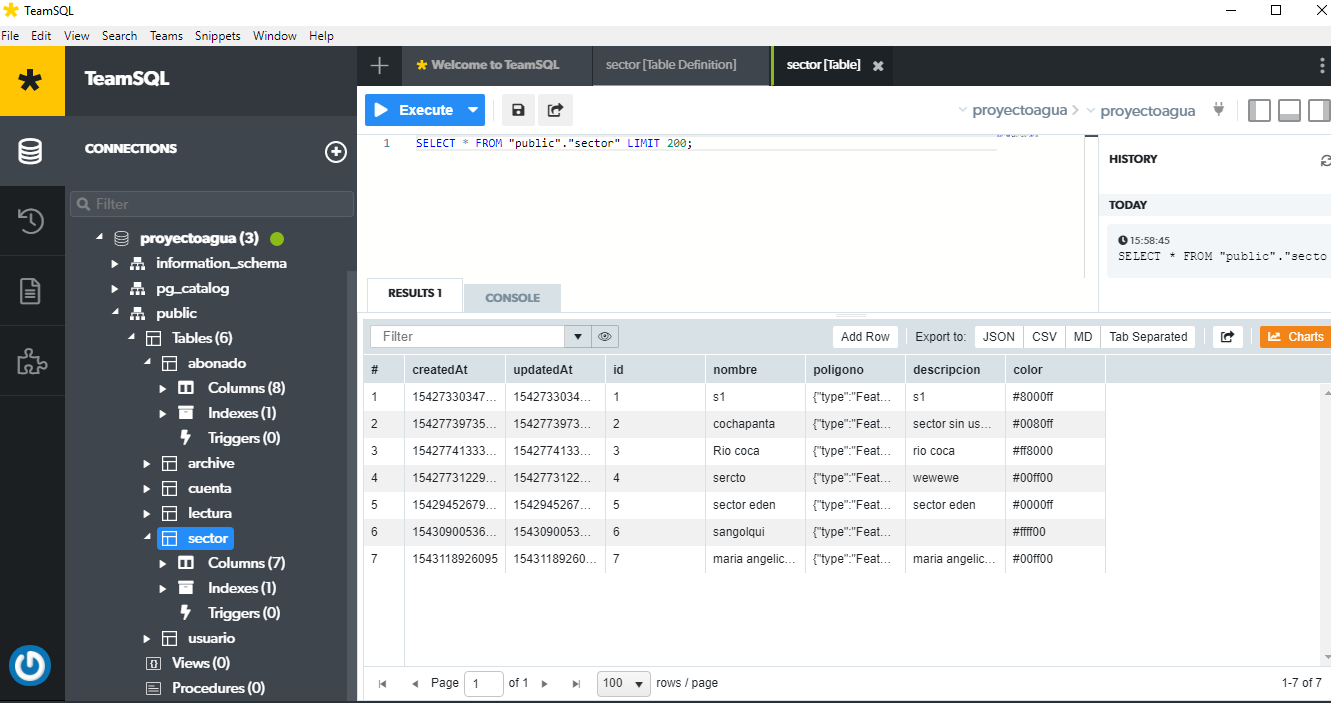


Figura 22: Consulta de la tabla lector mediante la interfaz grafica TeamSQL.

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.

## Git hub

## 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.

## 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 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.

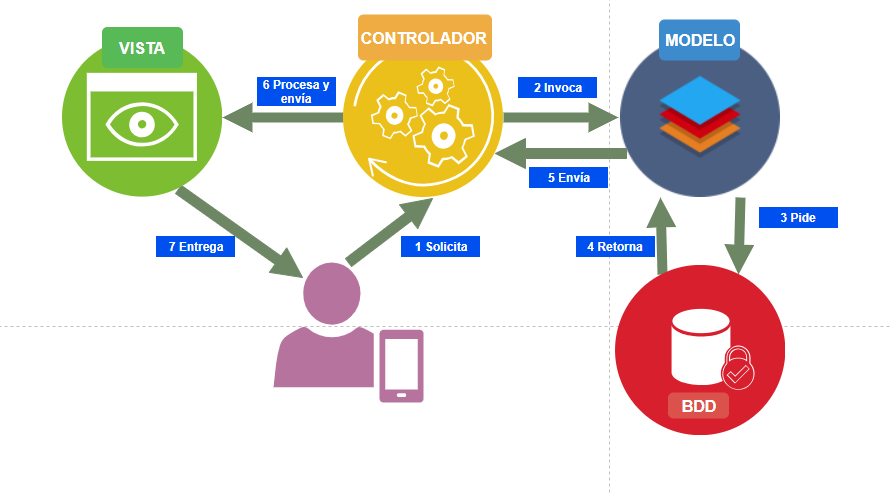


Figura 24: Esquema modelo vista controlador.

## Metodología 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.

**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 25, 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.

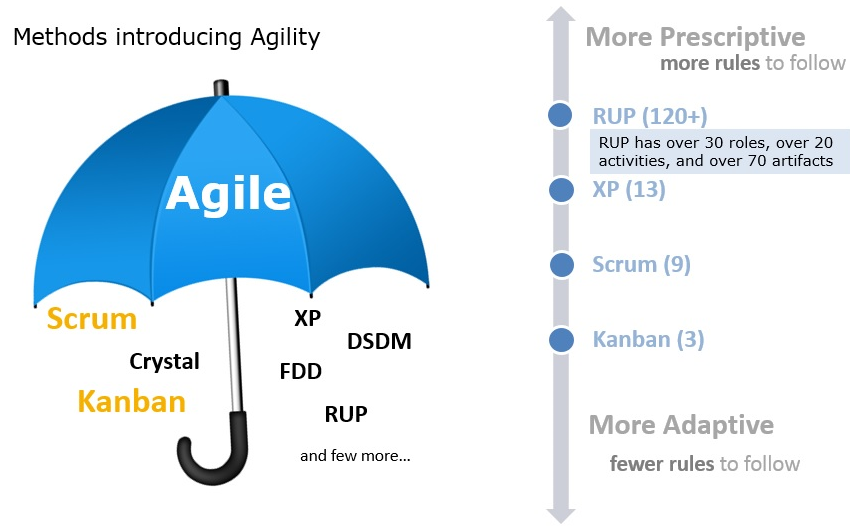


Figura 25:Principales metodologías agiles (Muñoz, 2016)

## 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 26 muestra el diagrama conceptual del modelo Kanban dividido en dos partes, producción y recuperación de piezas.

## 

Figura 26: 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 27 en el tablero Kanban de Vivifyscrum usado en este proyecto.

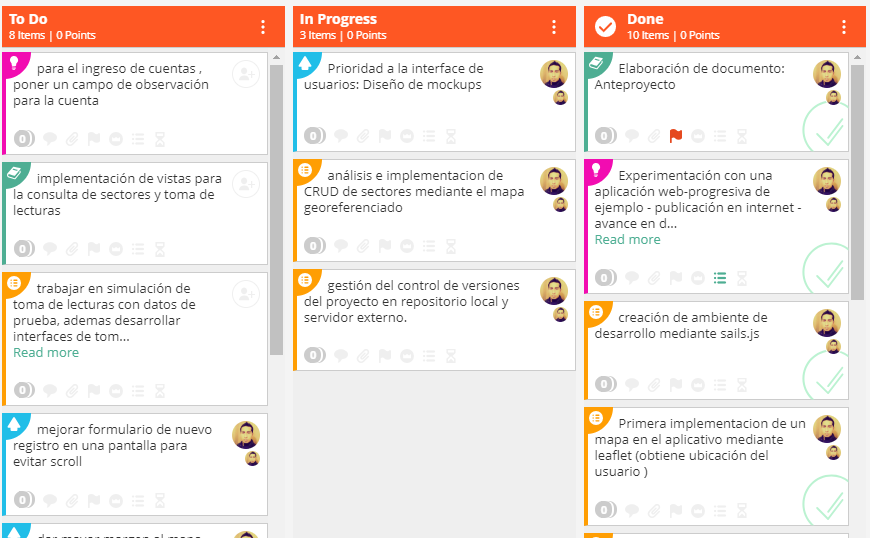


Figura 27: organización de tareas en Tablero Kanban de la herramienta VivifyScrum.

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.

**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 de la figura 27.

El diseño centrado en el usuario permite el desarrollo de software de manera colaborativa y propone 4 configuraciones posibles. El esquema de la figura 28, 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.

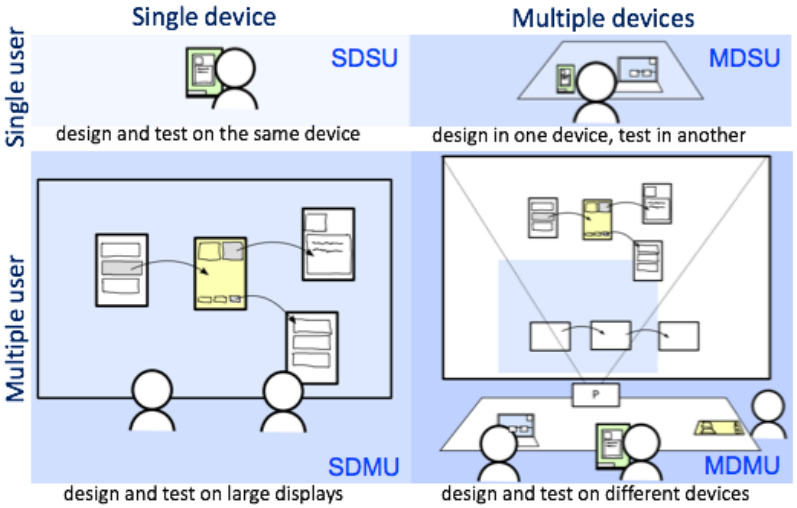


Figura 28 configuraciones físicas del diseño centrado en el usuario (Perez Medina & Vanderdonckt, 2018)

El flujo de trabajo de la figura 29, 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 29: 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.

**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

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

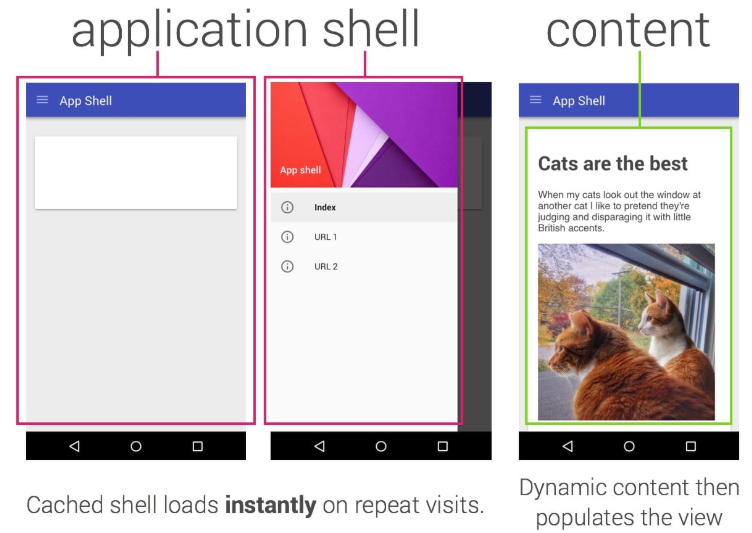


Figura 30: Shell de la aplicación web progresiva.

**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 31 contiene una porción del código del service worker implementado en el proyecto.



Figura 31: Service worker implementado en el proyecto.

**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 13 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 32: 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

## Desarrollo centrado en el usuario (mock ups)

## Tablero Kanban

## Diseño de Arquitectura

## La arquitectura de este proyecto se diseñó en base al estudio preliminar de la metodología centrada en el usuario donde se pudieron obtener varios requerimientos, según estas necesidades la arquitectura modelo vista controlador es una alternativa viable.

## Diagrama lógico de la de Base de Datos

La figura 33, presenta el diagrama lógico de la base de datos, 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: la entidad abonado 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.

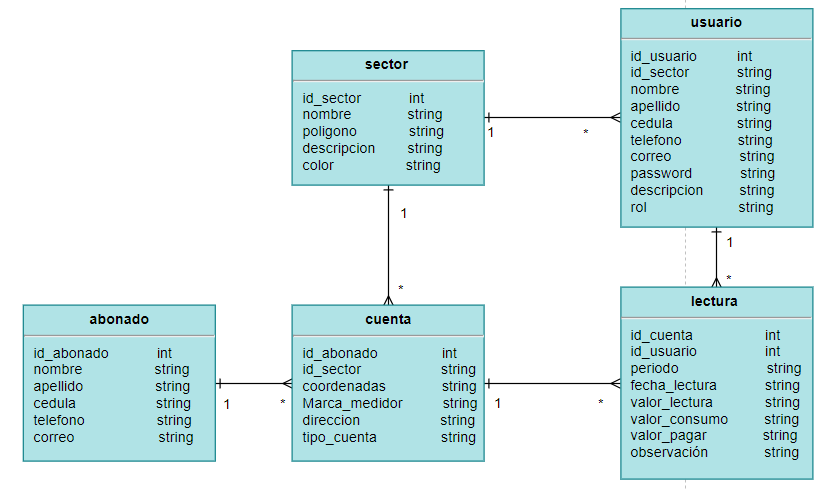


Figura 33: diagrama lógico de la base de datos.

## Diagrama de Arquitectura

La arquitectura que presenta la figura 34, 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.

Salis 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.

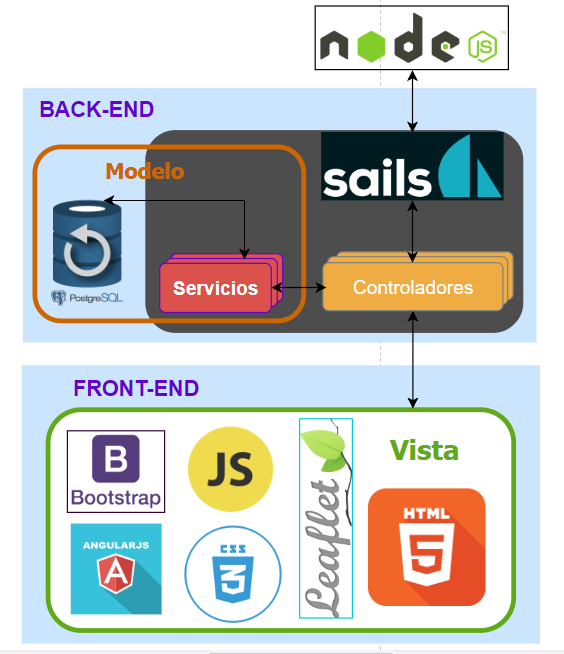


Figura 34: Arquitectura del sistema web progresivo para la toma de lecturas de agua potable

**Diagrama de Componentes**

El diagrama de componentes de la figura 35, 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. Finalmente, 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.

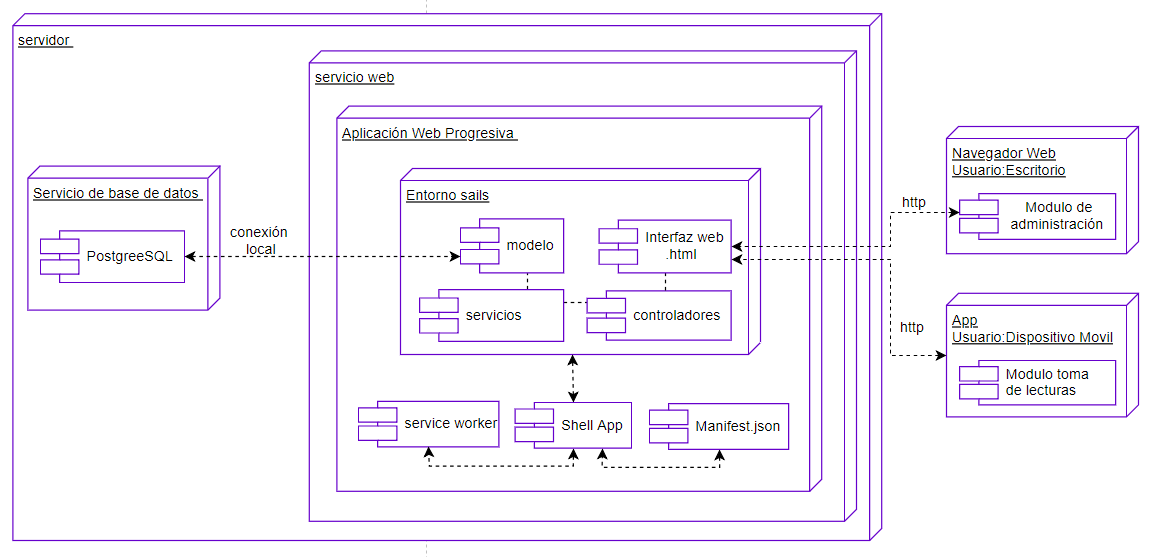


Figura 35: diagrama de componentes des sistema web progresivo para la toma de lecturas de agua potable.

+++++++++++++++++++++++++++++++++++++++++++++

## Conclusión de Capítulo

## En este capítulo se ha podido recolectar las historias de usuario y generar el Product Backlog del mismo, por lo que al tener en cuenta todas las historias de usuario recolectadas, se pudo generar un modelo entidad relación de la base de datos para poder ejercer el siguiente capítulo.

# CAPÍTULO III. DESARROLLO Y CODIFICACIÓN

En este capítulo se desarrollan y codifican todas las historias de usuario que fueron agregadas en el Product Backlog, además, se realizaron pruebas al final de cada Sprint para obtener funcionalidades al 100% listas para ser usadas por el usuario final.

## Sprint 1

En este Sprint se separaron las historias de usuario que tienen prioridad alta, esta prioridad se dio en base a las historias de usuario que son más importantes y necesarias para el usuario final, en la cual en su mayoría son historias de usuario que tienen funcionalidades independientes y de las cuales luego derivan otras historias de usuario.

## CAT-001

Se necesita mostrar un menú en el cual el usuario pueda interactuar para poder ingresar datos, visualizar, eliminar y modificar. Por lo que la pantalla debe contar con botones para que se puedan realizar dichas acciones.

Tabla 2

Historia de usuario CAT-001 detallada.

|  |  |
| --- | --- |
| Historia de Usuario: CAT-001 | |
| Numero: 1 | Usuario: Cliente |
| Nombre historia: Menú de catálogo para tipo de áreas | |
| Prioridad en negocio: Alta | Riesgo en desarrollo: Bajo |
| Puntos estimados: 2 | Iteración asignada: 1 |
| Programador responsable: Marco Lozano | |
| Descripción: Yo como jefe de producción, necesito definir un catálogo de geometrías, para así separar por tipo a cada una de ellas. | |
| Validación: El cliente debe poder seleccionar ver opciones de ingreso, modificación, eliminación y consulta, para el catálogo. | |

Figura 36. Menú de catálogo del proyecto.

## CAT-002

Esta historia de usuario, se requiere ingresar un tipo de área en el catálogo, por lo cual es necesario que cumpla con las propiedades que son: tipo de geometría, que quiere decir si es una línea, un polígono o un punto. También, se debe ingresar el ancho de la línea dado en pixeles en un rango de 1 a 4, el color de la línea y color de fondo si es que es polígono o punto.

Tabla 3

Historia de usuario CAT-002 detallada.

|  |  |
| --- | --- |
| Historia de Usuario: CAT-002 | |
| Numero: 2 | Usuario: Cliente |
| Nombre historia: Ingreso de catálogo de tipo de áreas | |
| Prioridad en negocio: Alta | Riesgo en desarrollo: Bajo |
| Puntos estimados: 1 | Iteración asignada: 1 |
| Programador responsable: Marco Lozano | |
| Descripción: Yo como jefe de producción, necesito asignar propiedades de color, grosor de línea, color de relleno a cada tipo del catálogo, para así poder ver en simbología. | |
| Validación: El cliente debe poder ingresar un nombre, y asignar un color de relleno, un color de línea, un grosor de línea, un tipo de geometría, y si el tipo de línea es entrecortada. | |

Figura 37. Ingreso de tipo de área, en catálogo.

## MAP-001

Se debe poder subir archivos csv que contengan puntos georreferenciados, es decir, que tengas coordenadas en formato UTM, las mismas que servirán para marcar en el mapa los límites de la geometría.

Tabla 4.

Historia de usuario MAP-001 detallada.

|  |  |
| --- | --- |
| Historia de Usuario: MAP-001 | |
| Numero: 3 | Usuario: Cliente |
| Nombre historia: Subida de archivos para el ingreso de áreas | |
| Prioridad en negocio: Alta | Riesgo en desarrollo: Bajo |
| Puntos estimados: 3 | Iteración asignada: 1 |
| Programador responsable: Marco Lozano | |
| Descripción: Yo, como jefe de producción, necesito subir archivos con puntos georreferenciados de la plantación para poder visualizar en el mapa. | |
| Validación: El cliente a través de una interfaz, debe poder subir archivos csv, los mismos que contienen puntos georreferenciados. | |

Figura 38. Subida de archivo para perímetro de la finca.

## MAP-002

Se debe poder observar figuras correspondientes a cada archivo subido, por lo que estas pueden ser líneas que representan canales o vías, también polígonos, que representan áreas como lotes u otras estructuras.

Tabla 5.

Historia de usuario MAP-002 detallada.

|  |  |
| --- | --- |
| Historia de Usuario: MAP-002 | |
| Numero: 4 | Usuario: Cliente |
| Nombre historia: Subida de archivos para el ingreso de áreas | |
| Prioridad en negocio: Alta | Riesgo en desarrollo: Medio |
| Puntos estimados: 4 | Iteración asignada: 1 |
| Programador responsable: Marco Lozano | |
| Descripción: Yo, como jefe de producción, necesito dibujar en el mapa geometrías afines con distintas partes de la finca, para poder visualizarlas en el mapa. | |
| Validación: El cliente a través de una interfaz, debe poder visualizar todas las geometrías que han sido ingresadas en un mapa, teniendo en cuenta las configuración de cada geometría que fue elegida en el catálogo. | |

Figura 39. Polígonos dibujados en el mapa.

## MAP-003

En caso de subir archivos que contengan puntos representados para palmas, estos deben tener atributos como nombres, o fechas de plantación entre otras, para poder visualizarlas en la aplicación, por lo que es necesario que el modulo permita leer estos archivos.

Tabla 6.

Historia de usuario MAP-003 detallada.

|  |  |
| --- | --- |
| Historia de Usuario: MAP-003 | |
| Numero: 5 | Usuario: Cliente |
| Nombre historia: Subida de archivos para el ingreso de áreas | |
| Prioridad en negocio: Alta | Riesgo en desarrollo: Baja |
| Puntos estimados: 4 | Iteración asignada: 1 |
| Programador responsable: Marco Lozano | |
| Descripción: Yo, como jefe de producción, necesito poder subir atributos a cada palma georreferenciada, para poder saber que palma es en el mapa. | |
| Validación: El cliente debe subir un archivo, el mismo que además de las coordenadas georreferenciadas, debe incluir el título del atributo y su valor, por cada una de las palmas. | |

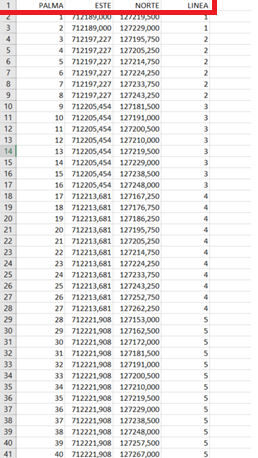


Figura 40. Archivo que contiene los atributos que son leídos por programa.

## MAP-004

A deferencia del a subida de archivo de perímetro de la finca, en esta historia se debe realizar una subida de archivos para cualquier tipo de geometría, está siendo línea, punto o polígono, con lo que el usuario debe seleccionar el tipo que pertenezca.

Tabla 7.

Historia de usuario MAP-004 detallada.

|  |  |
| --- | --- |
| Historia de Usuario: MAP-004 | |
| Numero: 6 | Usuario: Cliente |
| Nombre historia: Subida de archivos para el ingreso de áreas | |
| Prioridad en negocio: Alta | Riesgo en desarrollo: Bajo |
| Puntos estimados: 2 | Iteración asignada: 1 |
| Programador responsable: Marco Lozano | |
| Descripción: Yo, como jefe de producción, necesito poder dar un tipo geometría referente al catálogo al ingresar un área, para poder graficarla en base a sus propiedades. | |
| Validación: El cliente debe seleccionar el ítem del catálogo que ya ha sido ingresado previamente. | |

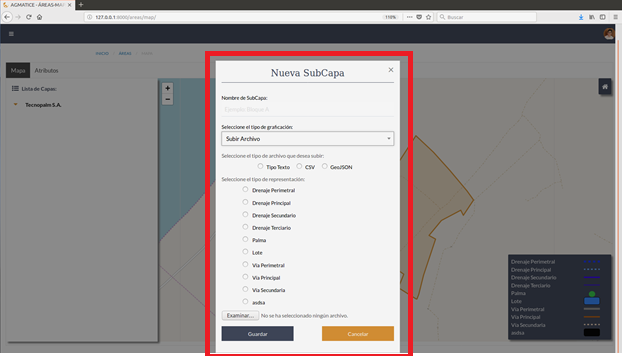


Figura 41. Ingreso de palmas, subiendo atributos.

## MAP-006

Para poder entender que representa daca figura en el mapa, es necesario que se tenga un cuadro de simbología o leyenda, por lo que el mapa debe constar de este para tener referencia de cada área dibujada.

Tabla 8.

Historia de usuario MAP-006 detallada.

|  |  |
| --- | --- |
| Historia de Usuario: MAP-006 | |
| Numero: 7 | Usuario: Cliente |
| Nombre historia: Simbología en mapa | |
| Prioridad en negocio: Alta | Riesgo en desarrollo: Bajo |
| Puntos estimados: 1 | Iteración asignada: 1 |
| Programador responsable: Marco Lozano | |
| Descripción: Yo, como gerente general, necesito visualizar una simbología en el mapa, para poder entender que representa cada figura. | |
| Validación: El cliente debe haber llenado datos en catálogo, para que así se muestre la simbología en base a los datos ingresados. | |

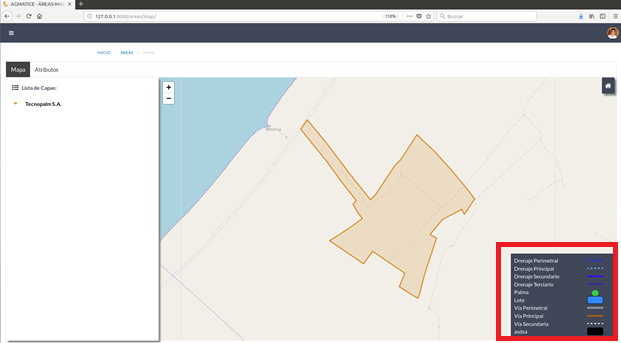


Figura 42. Mapa con su respectiva simbología.

## MAP-007

Como parte inicial de la parte del mapa, es necesario poder mostrar el perímetro de la finca dibujado, con lo que es necesario poder mostrarlo siempre en el mapa como fondo de todas las otras áreas.

Tabla 9.

Historia de usuario MAP-007 detallada.

|  |  |
| --- | --- |
| Historia de Usuario: MAP-007 | |
| Numero: 8 | Usuario: Cliente |
| Nombre historia: Gráfico del perímetro de finca. | |
| Prioridad en negocio: Alta | Riesgo en desarrollo: Baja |
| Puntos estimados: 2 | Iteración asignada: 1 |
| Programador responsable: Marco Lozano | |
| Descripción: Yo, como gerente general, necesito ver el perímetro de la finca pintada en el mapa, para poder tener un panorama del mismo. | |
| Validación: El cliente debe subir el archivo con las coordenadas georreferenciadas, o debe dibujar el perímetro de la finca. | |

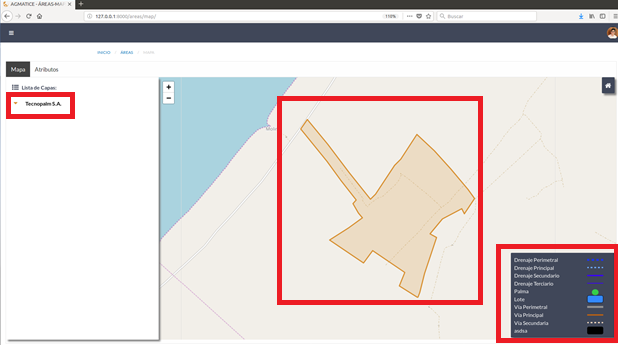


Figura 43. Mapa con el perímetro de la finca dibujada.

## MAP-008

Para poder organizar todas las figuras que se van a ingresar en el mapa, es necesario tener un manejo de capas, con lo cual se pueda agrupar por diferentes tipos de cosas cada área, así se podrá seleccionar y ver únicamente lo que se necesita y tener todo de forma ordenada.

Tabla 10.

Historia de usuario MAP-008 detallada.

|  |  |
| --- | --- |
| Historia de Usuario: MAP-008 | |
| Numero: 9 | Usuario: Cliente |
| Nombre historia: Visualización de capas. | |
| Prioridad en negocio: Alta | Riesgo en desarrollo: Bajo |
| Puntos estimados: 4 | Iteración asignada: 1 |
| Programador responsable: Marco Lozano | |
| Descripción: Yo, como gerente general, necesito poder ver a nivel de capas todas las estructuras que se tiene en la finca, para poder ver únicamente lo que quiera y no todo al mismo tiempo. | |
| Validación: El cliente debe ingresar una capa para poder agrupar varias de las geometrías, de las cuales estas también serán consideradas como capas si es que tienen geometrías hijas. Para definir este esquema se debe presentar las capas como un árbol de nodos. | |



Figura 44. Panel con capas de áreas separadas por lotes.

## Sprint 2

En este Sprint se separaron las historias de usuario que tienen prioridad media, estas historias de usuario tienen dependencia de las anteriores. Además, se agregaron algunas historias de usuarios que no se contemplaron en el capítulo 2 del presente documento.

## MAP-009

Al tener una interacción constante con el mapa, y poder mover y manejar el zoom del mismo, es necesario tener una opción de centrar el mapa al perímetro de la finca, con lo cual siempre se podrá regresar a la vista general del mapa.

Tabla 11.

Historia de usuario MAP-009 detallada.

|  |  |
| --- | --- |
| Historia de Usuario: MAP-009 | |
| Numero: 10 | Usuario: Cliente |
| Nombre historia: Centrar mapa. | |
| Prioridad en negocio: Media | Riesgo en desarrollo: Bajo |
| Puntos estimados: 1 | Iteración asignada: 2 |
| Programador responsable: Marco Lozano | |
| Descripción: Yo, como gerente general, necesito poder centrar el mapa en la finca, para no perder la vista si es que se mueve o se navega fuera del lugar. | |
| Validación: El cliente debe dar click en el botón de entrar mapa, para poder centrar el mapa en medio de todas las geometrías ingresadas, o visualizadas en ese instante. | |

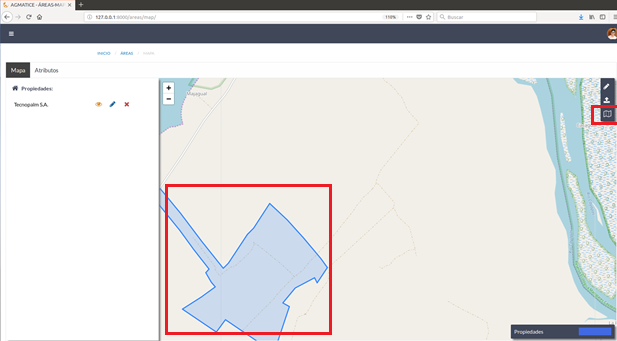


Figura 45. Mapa centrado, y botón de centrar mapa.

## MAP-011

Como parte fundamental del sistema, es necesario que se pueda ver cada punto o palma ingresada, y a más detalle, es necesario poder diferenciar cada palma, por lo que se tiene que poder visualizar cada atributo ingresado de cada punto.

Tabla 12.

Historia de usuario MAP-011 detallada.

|  |  |
| --- | --- |
| Historia de Usuario: MAP-011 | |
| Numero: 11 | Usuario: Cliente |
| Nombre historia: Mostrar punto o palmas. | |
| Prioridad en negocio: Media | Riesgo en desarrollo: Bajo |
| Puntos estimados: 5 | Iteración asignada: 2 |
| Programador responsable: Marco Lozano | |
| Descripción: Yo, como gerente general, necesito poder ver las palmas de finca con su respectiva información, para poder saber que palma es. | |
| Validación: El cliente dentro de las capas ingresadas, debe poder visualizar las palmas que se encuentran dentro de una capa, teniendo en cuenta que no se debe mostrar en el panel de capas todos los puntos. | |

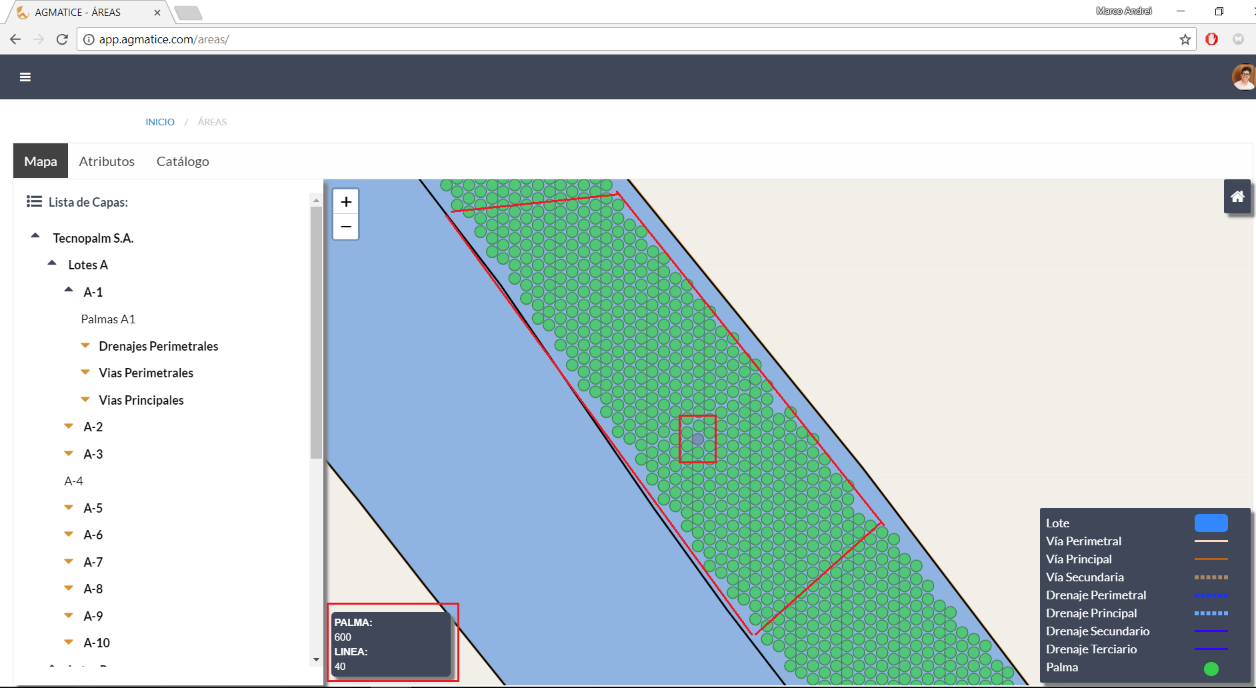


Figura 46. Mapa con palmas y su respectiva información.

## ATR-001

Se debe ingresar atributos, estos son referentes a características de cada área, estas sirven para poder tener información real y valida.

Tabla 13.

Historia de usuario ATR-001 detallada.

|  |  |
| --- | --- |
| Historia de Usuario: ATR-001 | |
| Numero: 12 | Usuario: Cliente |
| Nombre historia: Ingresar atributos. | |
| Prioridad en negocio: Media | Riesgo en desarrollo: Bajo |
| Puntos estimados: 1 | Iteración asignada: 2 |
| Programador responsable: Marco Lozano | |
| Descripción: Yo, como jefe de producción, necesito definir una serie de atributos, para poder asignar información a cada lote parcelario. | |
| Validación: El cliente debe poder ingresar una serie de atributos con unidades ya predefinidas a través de un formulario. | |

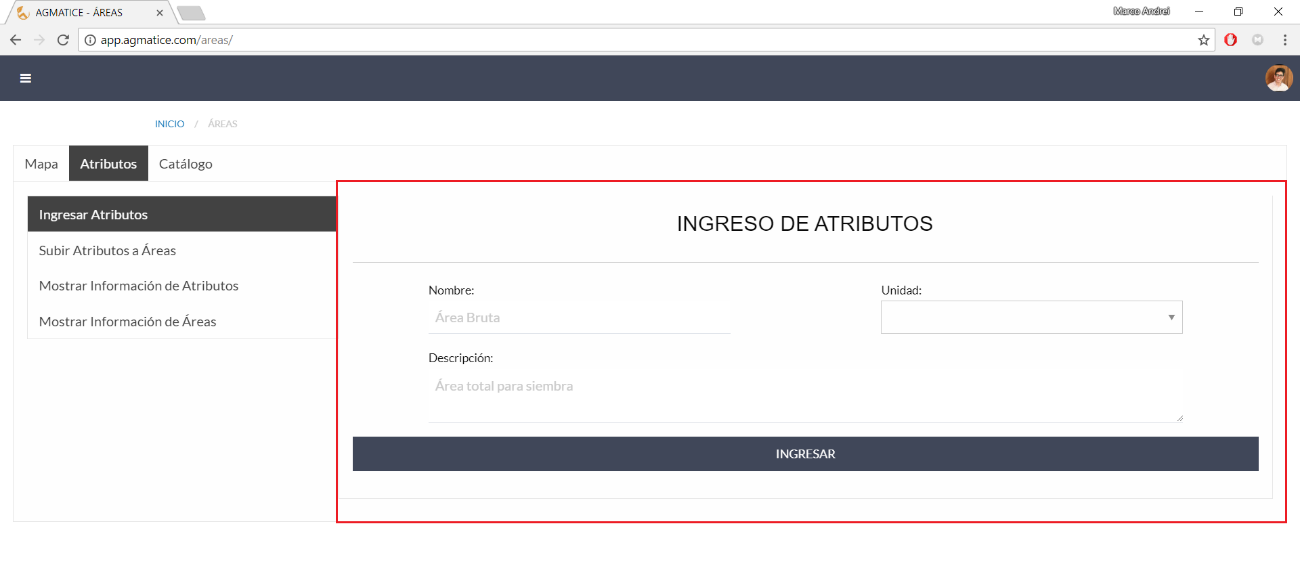


Figura 47. Formulario de ingreso de atributos.

## ATR-002

Una vez que se ingresaron los atributos, es necesario poder vincularlo a cada área para que tenga un valor significativo, con lo que es necesario que se lo asigne a un área en específico.

Tabla 14.

Historia de usuario ATR-002 detallada.

|  |  |
| --- | --- |
| Historia de Usuario: ATR-002 | |
| Numero: 13 | Usuario: Cliente |
| Nombre historia: Asignar atributo a áreas. | |
| Prioridad en negocio: Media | Riesgo en desarrollo: Bajo |
| Puntos estimados: 4 | Iteración asignada: 2 |
| Programador responsable: Marco Lozano | |
| Descripción: Yo, como jefe de producción, necesito asignar atributos a cada lote parcelario, o a cada geometría, para así poder visualizar su información. | |
| Validación: El cliente debe poder asignar un atributo a un área específica, esto a través de formulario, o poder subir un archivo que contengan los atributos que van a ser asignados a diferentes áreas. | |

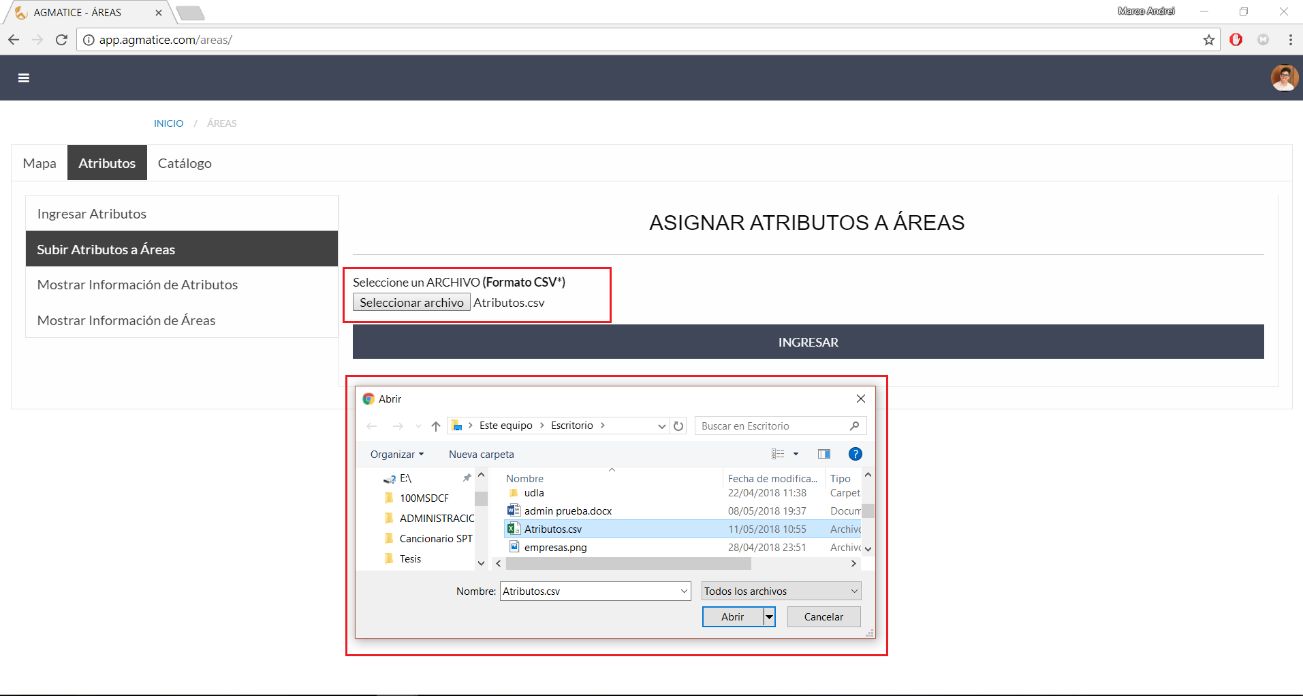


Figura 48. Subida de archivo para asignación de atributos.

## ATR-006

Para poder ver que todo se haya ingresado de forma correcta, es necesario poder tener una pantalla que muestre los atributos ingresados.

Tabla 15.

Historia de usuario ATR-006 detallada.

|  |  |
| --- | --- |
| Historia de Usuario: ATR-006 | |
| Numero: 14 | Usuario: Cliente |
| Nombre historia: Visualización de atributos. | |
| Prioridad en negocio: Media | Riesgo en desarrollo: Baja |
| Puntos estimados: 2 | Iteración asignada: 2 |
| Programador responsable: Marco Lozano | |
| Descripción: Yo, como gerente general, necesito ver todos los atributos ingresados, para tener en cuenta los atributos que existen en las geometrías. | |
| Validación: El cliente debe poder ver todos los atributos ingresados a través de una tabla, que muestre la respectiva información de la misma. | |

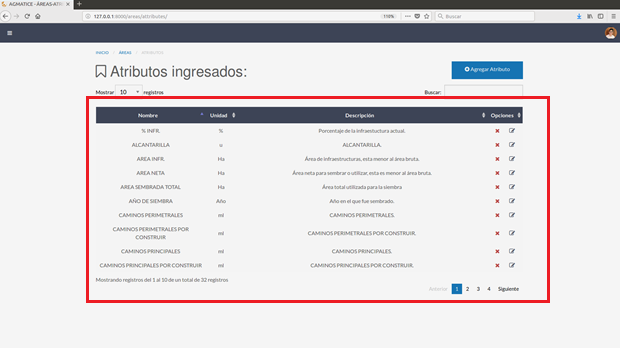


Figura 49. Tabla de atributos ingresados.

## ATR-007

De manera igual al de los atributos, es necesario poder ver por cada área que atributos tiene y su respectivo valor en una pantalla informativa.

Tabla 16.

Historia de usuario ATR-007 detallada.

|  |  |
| --- | --- |
| Historia de Usuario: ATR-007 | |
| Numero: 15 | Usuario: Cliente |
| Nombre historia: Visualización de atributos por área. | |
| Prioridad en negocio: Media | Riesgo en desarrollo: Bajo |
| Puntos estimados: 1 | Iteración asignada: 2 |
| Programador responsable: Marco Lozano | |
| Descripción: Yo, como gerente general, necesito ver por área, que atributos tienen, para poder saber la información correspondiente de cada área. | |
| Validación: El cliente debe poder ver los atributos de cada área. Estos siendo filtrados por el área seleccionada, además se debe poder visualizar todos los atributos asignados a todas las áreas. | |

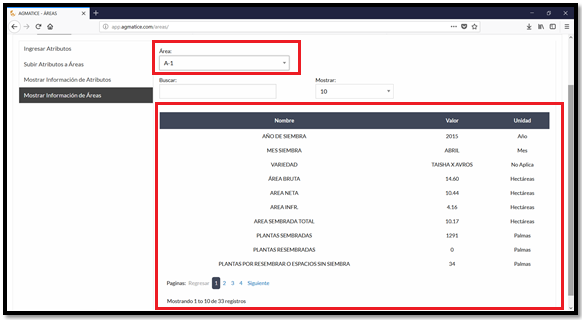


Figura 50.Tabla de atributos por área.

## MAP-010

Se necesita poder ocultar las figuras del mapa, para poder ver únicamente lo que se quiere, y de igual manera se debe poder observar luego de ocultar, esto debe manejarse en el panel de capas.

Tabla 17.

Historia de usuario MAP-010 detallada.

|  |  |
| --- | --- |
| Historia de Usuario: MAP-010 | |
| Numero: 16 | Usuario: Cliente |
| Nombre historia: Mostrar/Ocultar geometrías. | |
| Prioridad en negocio: Medio | Riesgo en desarrollo: Bajo |
| Puntos estimados: 1 | Iteración asignada: 2 |
| Programador responsable: Marco Lozano | |
| Descripción: Yo, como gerente general, necesito poder ocultar y mostrar los polígonos que quiera, para poder fijarme en lo que quiero ver. | |
| Validación: El cliente debe poder ocultar y mostrar una figura seleccionada en el panel de capas, además, estas pueden ser mostradas aunque se cierre la agrupación de capas. | |

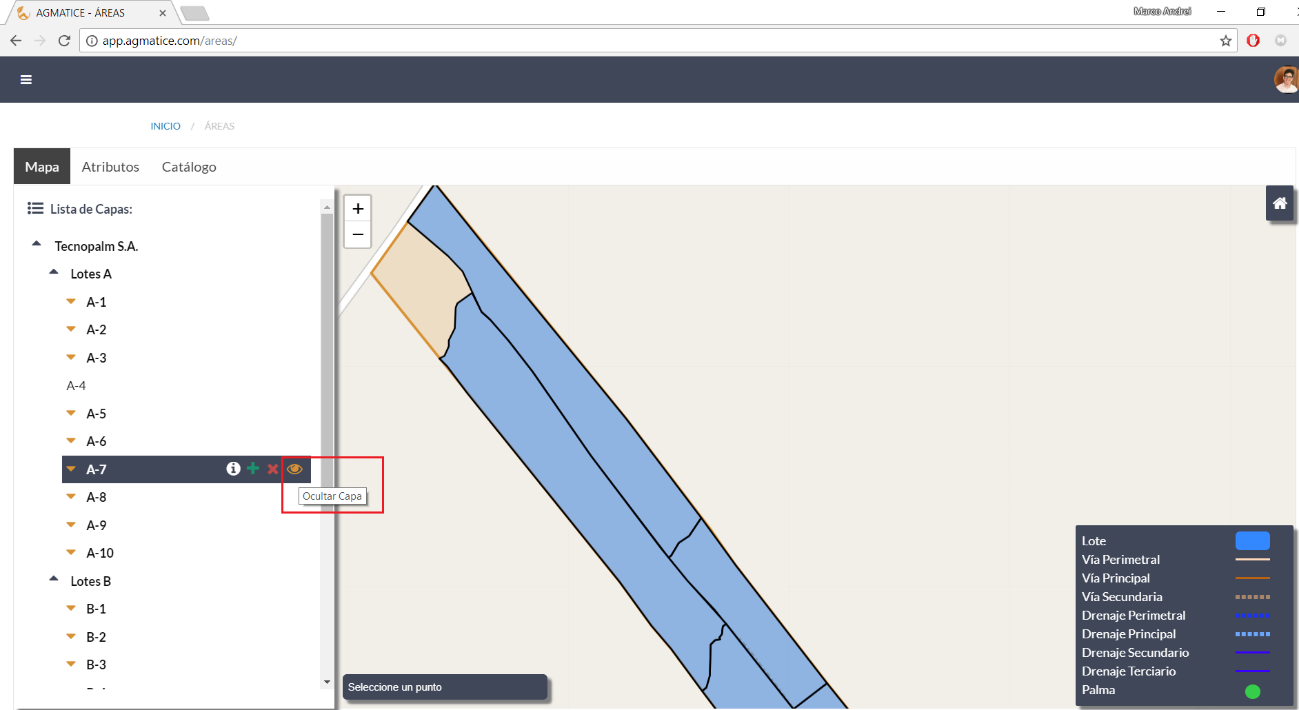


Figura 51. Botones de ocultar geometría.

## MAP-012

Es necesario que se muestre la lista de atributos que fueron asignados a cada lote, dentro del mapa, para tener comodidad del usuario, lo que permita que no salga de la página para poder visualizarlos.

Tabla 18.

Historia de usuario MAP-012 detallada.

|  |  |
| --- | --- |
| Historia de Usuario: MAP-012 | |
| Numero: 17 | Usuario: Cliente |
| Nombre historia: Mostrar atributos en mapa. | |
| Prioridad en negocio: Media | Riesgo en desarrollo: Bajo |
| Puntos estimados: 2 | Iteración asignada: 2 |
| Programador responsable: Marco Lozano | |
| Descripción: Yo, como gerente general, necesito, poder visualizar la información de los atributos correspondientes a cada a cada lote parcelario en el mapa, para no tener que ir a otro lado para ver la información. | |
| Validación: El cliente al dar click en el nombre del polígono, se centrará en el mapa y mostrará sus respectivos atributos. | |

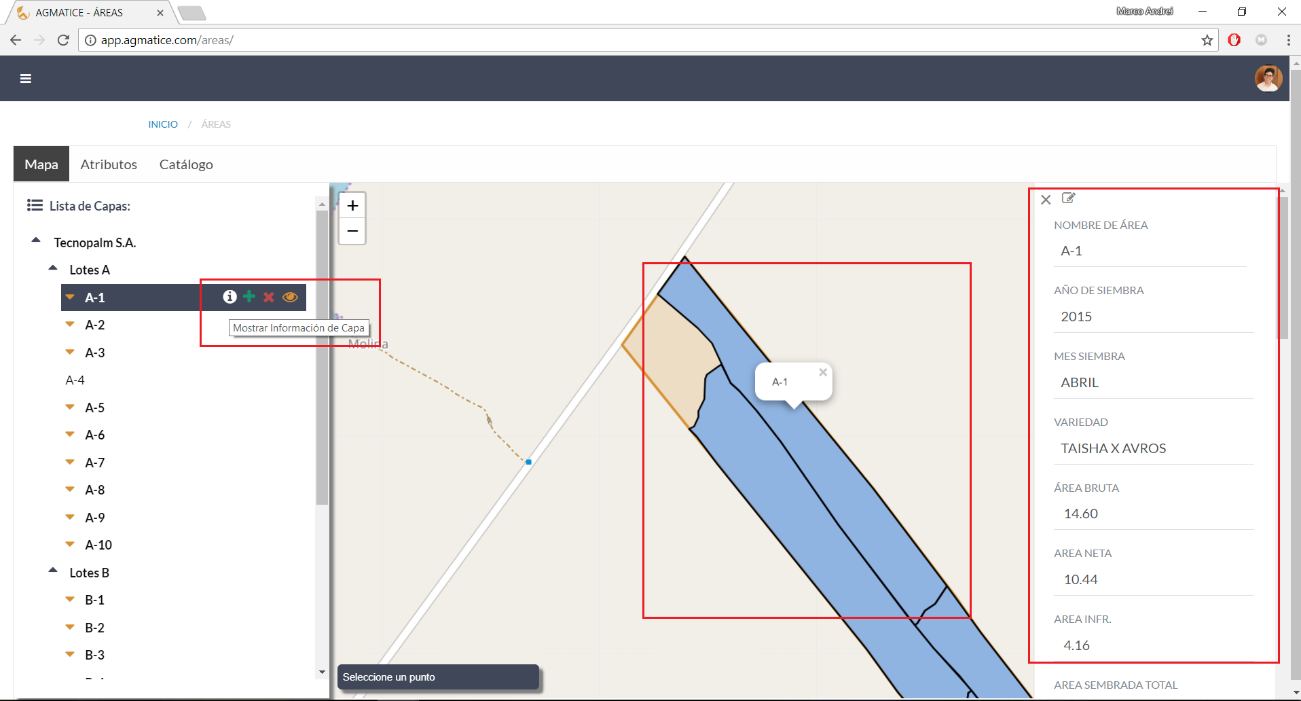


Figura 52. Atributos en el mapa.

## MAP-013

Para diferenciar entre cada figura dibujada en el mapa, es necesario mostrar su nombre cuando se de click en la figura. Ya que así se deberá mostrar un mensaje con el nombre de la figura seleccionada.

Tabla 19.

Historia de usuario MAP-013 detallada.

|  |  |
| --- | --- |
| Historia de Usuario: MAP-013 | |
| Numero: 18 | Usuario: Cliente |
| Nombre historia: Mostrar nombre de la geometría en mapa. | |
| Prioridad en negocio: Media | Riesgo en desarrollo: Bajo |
| Puntos estimados: 1 | Iteración asignada: 2 |
| Programador responsable: Marco Lozano | |
| Descripción: Yo, como gerente general, necesito, saber que geometría es la que está dibujada en el mapa, para diferenciar entre geometrías. | |
| Validación: El cliente al dar click en la geometría dibujada en el mapa, ya sea polígono, punto o línea, debe desplegar a través de un pop-up el nombre de la figura seleccionada. | |

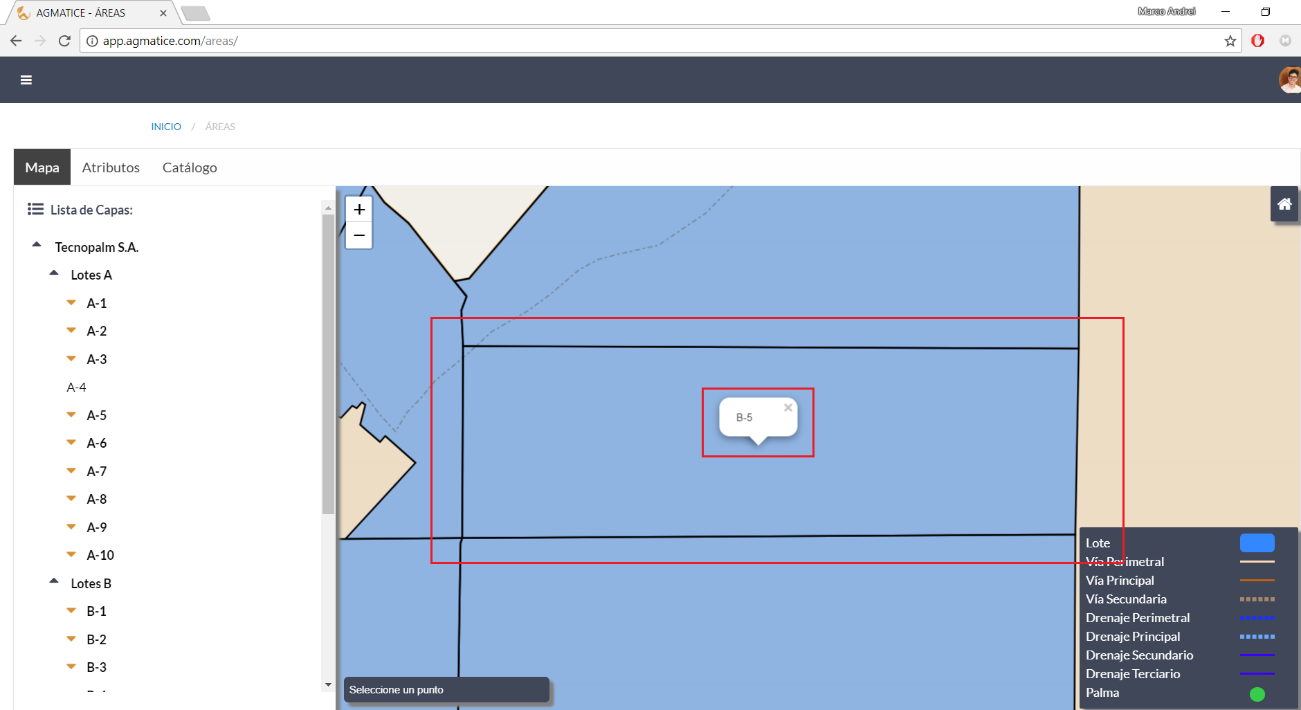


Figura 53. Atributos en el mapa.

## Sprint 3

En este Sprint se separaron las historias de usuario que tienen menos prioridad en el negocio, y las que dependen de las otras historias de usuario.

## CAT-003

Luego de poder ingresar, se debe poder observar una tabla que muestre la lista de datos ingresados en el catálogo, por lo que se de mostrar toda la información agregada a la misma, y además se debe poner botones de actualizar y eliminar.

Tabla 20.

Historia de usuario CAT-003 detallada.

|  |  |
| --- | --- |
| Historia de Usuario: CAT-003 | |
| Numero: 19 | Usuario: Cliente |
| Nombre historia: Mostrar datos de catálogo. | |
| Prioridad en negocio: Baja | Riesgo en desarrollo: Bajo |
| Puntos estimados: 2 | Iteración asignada: 3 |
| Programador responsable: Marco Lozano | |
| Descripción: Yo como jefe de producción, necesito visualizar las propiedades de cada tipo en el catálogo, para ver cuáles han sido ingresados. | |
| Validación: El cliente entrar en catálogo debe observar una tabla que contenga toda la lista de los datos ingresados previamente, además, se debe tener acceso a botones de eliminar y modificar, dentro de la tabla. | |

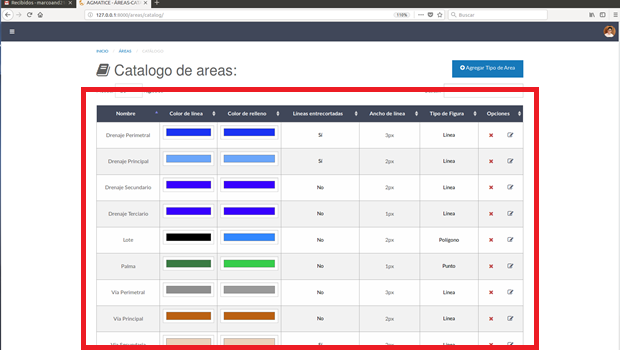


Figura 54. Lista de datos del catálogo.

## CAT-004

En caso de ingresar algún elemento mal, se debe poder eliminar, por lo que esta historia de usuario se enfoca en eliminar el dato del catálogo seleccionado.

Tabla 21.

Historia de usuario CAT-004 detallada.

|  |  |
| --- | --- |
| Historia de Usuario: CAT-004 | |
| Numero: 20 | Usuario: Cliente |
| Nombre historia: Eliminar dato de catálogo. | |
| Prioridad en negocio: Baja | Riesgo en desarrollo: Bajo |
| Puntos estimados: 1 | Iteración asignada: 3 |
| Programador responsable: Marco Lozano | |
| Descripción: Yo como jefe de producción, necesito eliminar los tipos del catálogo, para poder borrar los que tengan información incorrecta. | |
| Validación: El cliente al dar click en el botón eliminar que se encuentra en la tabla de catálogos, además se debe abrir un pequeño modal en el que se confirme la eliminación de dicho elemento. | |

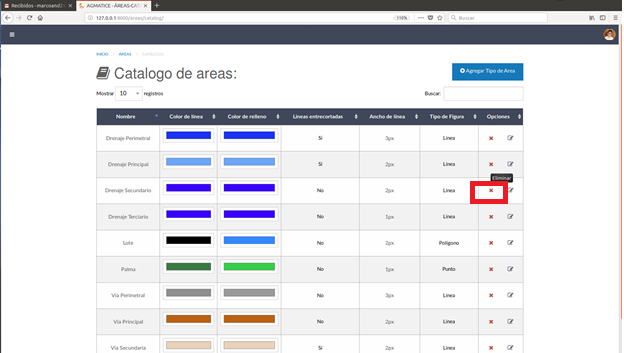


Figura 55. Botón de eliminar dato de catálogo.

## CAT-005

En caso de haber ingresado mal algún elemento o de actualizar alguno de los datos del elemento del catálogo, se debe poder modificar el elemento que se seleccione del catálogo.

Tabla 22.

Historia de usuario CAT-005 detallada.

|  |  |
| --- | --- |
| Historia de Usuario: CAT-005 | |
| Numero: 21 | Usuario: Cliente |
| Nombre historia: Modificar dato de catálogo. | |
| Prioridad en negocio: Baja | Riesgo en desarrollo: Baja |
| Puntos estimados: 2 | Iteración asignada: 3 |
| Programador responsable: Marco Lozano | |
| Descripción: Yo como jefe de producción, necesito modificar las propiedades del tipo en catálogo, para poder actualizar las propiedades en caso de que sean requeridas. | |
| Validación: El cliente al dar en el botón de modificar, de mostrar una modal que contenga los datos actuales del elemento a modificar, el mismo que permitirá actualizar el elemento. | |

Figura 56. Ventana de actualización de dato de catálogo.

## ATR-003

En caso de actualizar la información de un área, se debe poder actualizar la información del área seleccionada teniendo en cuenta que está será solo su nombre.

Tabla 23.

Historia de usuario ATR-003 detallada.

|  |  |
| --- | --- |
| Historia de Usuario: ATR-003 | |
| Numero: 22 | Usuario: Cliente |
| Nombre historia: Modificar información de área. | |
| Prioridad en negocio: Baja | Riesgo en desarrollo: Bajo |
| Puntos estimados: 1 | Iteración asignada: 3 |
| Programador responsable: Marco Lozano | |
| Descripción: Yo, como jefe de producción, necesito poder editar la información de cada geometría, para actualizar si es necesario. | |
| Validación: El cliente al mostrar los atributos dentro del mapa, podrá ver la información del área seleccionada, por lo que ahí podrá actualizar la misma, teniendo en cuenta que los limites georreferenciados no se podrán modificar. | |

Figura 57. Ventana de actualización de nombre de capa.

## ATR-004

Se necesita eliminar los atributos que estén desactualizados, o que ya no se usen.

Tabla 24.

Historia de usuario ATR-004 detallada.

|  |  |
| --- | --- |
| Historia de Usuario: ATR-004 | |
| Numero: 23 | Usuario: Cliente |
| Nombre historia: Eliminar atributo. | |
| Prioridad en negocio: Baja | Riesgo en desarrollo: Bajo |
| Puntos estimados: 1 | Iteración asignada: 3 |
| Programador responsable: Marco Lozano | |
| Descripción: Yo, como jefe de producción, necesito eliminar atributos, para borrar aquellas que ya no se utilicen, o estén mal. | |
| Validación: El cliente al dar click en eliminar, deberá borrar el atributo, este deberá borrarse en cascada si estaba vinculado a diferentes áreas. Además, deberá mostrar un aviso de confirmación para poder eliminarlo. | |

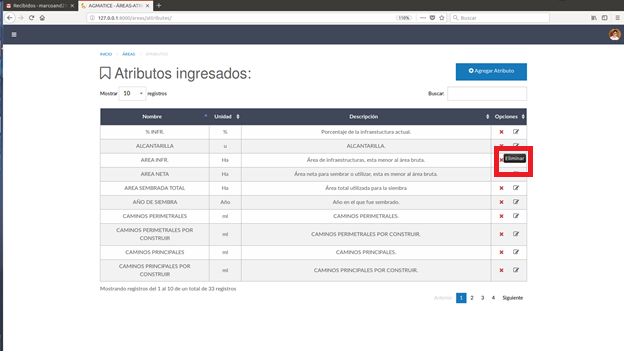


Figura 58. Botón de eliminación de atributo.

## ATR-005

Se debe poder actualizar la información de atributos en generales, teniendo en cuenta que estos no son los que están asignados a cada área.

Tabla 25.

Historia de usuario ATR-005 detallada.

|  |  |
| --- | --- |
| Historia de Usuario: ATR-005 | |
| Numero: 24 | Usuario: Cliente |
| Nombre historia: Actualizar atributos. | |
| Prioridad en negocio: Baja | Riesgo en desarrollo: Baja |
| Puntos estimados: 2 | Iteración asignada: 3 |
| Programador responsable: Marco Lozano | |
| Descripción: Yo, como jefe de producción, necesito modificar los atributos, para cambiar la información que haya sido mal ingresada. | |
| Validación: El cliente al dar click actualizar, debe permitir modificar la información del atributo, este al modificarlo, debe mostrar los cambios incluso en los atributos que fueron asignados a un área. | |



Figura 59. Ventana de actualización de atributos.

## MAP-005

En caso de hacer pruebas, o de desechar áreas que no sirven, se debe poder eliminar las áreas del mapa, por lo que al dar click en el icono de borrar estas se deben eliminar.

Tabla 26.

Historia de usuario MAP-005 detallada.

|  |  |
| --- | --- |
| Historia de Usuario: MAP-005 | |
| Numero: 25 | Usuario: Cliente |
| Nombre historia: Eliminar geometría. | |
| Prioridad en negocio: Baja | Riesgo en desarrollo: Bajo |
| Puntos estimados: 2 | Iteración asignada: 3 |
| Programador responsable: Marco Lozano | |
| Descripción: Yo, como jefe de producción, necesito poder eliminar las geometrías en el mapa, para así no tener las que no sirven, o están mal georreferenciadas. | |
| Validación: El cliente al dar click en el botón de eliminar, debe borrar la geometría seleccionada, teniendo en cuenta que se debe borrar todas las figuras que sean hijos de la misma, además al borrar una capa de puntos, se debe eliminar cada puntos. | |



Figura 60. Eliminar geometría o capa.

## Conclusión de capítulo

## 

## En este capítulo se desarrollaron 3 Sprints, los cuales tuvieron una duración de 2 semanas cada uno aproximadamente y por lo que se pudo rescatar, que, al final de cada sprint se desarrollaron pruebas, y se entregaron funcionalidades en las cuales el usuario final ya podían ir utilizando.

# CAPÍTULO IV. CASO DE PRUEBAS

Un caso de prueba es un proceso en el cual se evalúa a través de condiciones y variables, si el funcionamiento de una historia de usuario es completamente satisfactorio o cumple al 100% con lo requerido por el usuario.

En este capítulo se documenta los casos de prueba que fueron ejecutados al final de cada Sprint, por lo que, se tomó el formato de la siguiente tabla:

Tabla 27.

Ejemplo de caso de prueba.

|  |  |  |
| --- | --- | --- |
| Caso de prueba Nº: Número de caso de prueba | | |
| Objetivo de caso de prueba | Se describe el objetivo al cual se quiere llegar al cumplir el caso de prueba satisfactoriamente | |
| Identificador | Es código identificador del caso de prueba | |
| Nombre del caso | Es el nombre del caso de prueba | |
| Precondiciones | En caso de que haya precondiciones para que se cumpla el caso de prueba | |
| Paso | Resultado Esperado | Resultado Real |
| El número del paso y descripción del mismo | Lo que debería pasar en caso de que cumpla el paso. | Evidencia de lo que pasa, o resultado real. |

## CP\_MAP\_001

Tabla 28.

Caso de prueba perteneciente a la historia de usuario MAP-001.

|  |  |  |
| --- | --- | --- |
| Caso de prueba Nº: 1 | | |
| Objetivo de caso de prueba | Validar que, en el ingreso de subida de archivos georreferenciados de la propiedad, se muestre correctamente los puntos en el mapa. | |
| Identificador | CP\_MAP\_001 | |
| Nombre del caso | Validar ingreso de propiedad de finca. | |
| Precondiciones | * El archivo georreferenciado debe cumplir con el formato propuesto. | |
| Paso | Resultado Esperado | Resultado Real |
| 1. Dar click en la opción de subida de archivos. | Se debe desplegar una ventana con la información para subir archivos. | OK. |
| 2. Ingresar el nombre la propiedad. | Ninguno. | Ninguno. |
| 3.Dar click en el botón de subida de archivos. | Se debe cargar el archivo, y mostrar el nombre a lado del botón. | OK. |
| 4.Dar click en el botón guardar. | Se debe desplegar un mensaje, de ingreso correcto, y se debe mostrar en el mapa el área dibujada. | OK. |

## CP\_MAP\_002

Tabla 29.

Caso de prueba perteneciente a la historia de usuario MAP-002.

|  |  |  |
| --- | --- | --- |
| Caso de prueba Nº: 2 | | |
| Objetivo de caso de prueba | Validar que, se despliegue en el mapa las figuras correspondientes al tipo de área seleccionada en catálogo. | |
| Identificador | CP\_MAP\_002 | |
| Nombre del caso | Validar áreas dibujadas en el mapa. | |
| Precondiciones | * Se debe tener ingresado datos en el catálogo de áreas. * Se debe tener ingresado áreas con respectivas relaciones con el catálogo. | |
| Paso | Resultado Esperado | Resultado Real |
| 1.Se debe dar click en la opción de mapa del menú. | Se debe abrir la pantalla de áreas. | OK. |
| 2.Se debe dar click en el nombre de la propiedad. | Se debe abrir el panel de capas y áreas que pertenecen a la propiedad. | OK. |
| 3.Se debe dar click en cualquiera de las capas de la propiedad. | Se debe abrir las subcapas del área, las mismas que si pertenecen a un tipo de catálogo deben pintarse en el mapa. | OK. |

## CP\_MAP\_003

Tabla 30.

Caso de prueba perteneciente a la historia de usuario MAP-003.

|  |  |  |
| --- | --- | --- |
| Caso de prueba Nº: 3 | | |
| Objetivo de caso de prueba | Validar que, se muestre en el mapa cada palma georreferenciada, y que al pasar el mouse sobre dicha palma, se muestre información de sus atributos. | |
| Identificador | CP\_MAP\_003 | |
| Nombre del caso | Validar graficación de palmas y atributos en mapa. | |
| Precondiciones | * Se debe haber subido un archivo con las palmas georreferenciadas, las mismas que debían tener sus atributos. | |
| Paso | Resultado Esperado | Resultado Real |
| 1.Se debe dar click en la opción de mapa del menú. | Se debe abrir la pantalla de áreas. | OK. |
| 2.Se debe dar click en el nombre de la propiedad. | Se debe abrir el panel de capas y áreas que pertenecen a la propiedad. | OK. |
| 3.Se debe dar click en la capa que contenga como subcapa las palmas. | Se debe mostrar en el mapa una agrupación de puntos correspondiente a cada palma. | OK. |
| 4.Se debe hacer zoom con el scroll. | Se debe ver cada palma con un circulo. | OK. |
| 5.Se debe pasar con el mouse sobre cualquier punto. | Se debe mostrar en la esquina inferior izquierda del mapa la información sobre esa palma. | OK. |

## CP\_MAP\_004

Tabla 31.

Caso de prueba perteneciente a la historia de usuario MAP-004.

|  |  |  |
| --- | --- | --- |
| Caso de prueba Nº: 4 | | |
| Objetivo de caso de prueba | Validar que, que se deba ingresar correctamente la subida de archivos para áreas dentro de la propiedad, en la cual se debe elegir un tipo de área referente al catálogo. | |
| Identificador | CP\_MAP\_004 | |
| Nombre del caso | Validar subida de archivos de sub áreas de la propiedad. | |
| Precondiciones | Se debe tener ingresado datos en el catálogo de áreas.  El archivo georreferenciado debe cumplir con el formato propuesto. | |
| Paso | Resultado Esperado | Resultado Real |
| 1.Se debe dar click en la opción de mapa del menú. | Se debe abrir la pantalla de áreas. | OK. |
| 2.Se debe dar click en el nombre de la propiedad. | Se debe abrir el panel de capas y áreas que pertenecen a la propiedad. | OK. |
| 3.Se debe dar click en agregar capa o subcapa. | Se debe desplegar una ventana para el ingreso de capa o subcapa. | OK. |
| 4.Se debe elegir la opción de subir archivo en el combo box de tipo de graficación. | Se debe desplegar debajo del combo box, dos opciones para elegir el tipo de archivo a subir, y el tipo de área referente al catálogo. Además se debe mostrar el campo para ingresar el nombre de capa y para subir el archivo. | OK. |
| 5.Se deben ingresar los datos mostrados en la ventana. | Ninguna. | OK. |
| 6.Se debe dar click en guardar. | Se debe desplegar un mensaje de ingreso correcto. | OK. |

## CP\_MAP\_005

Tabla 32.

Caso de prueba perteneciente a la historia de usuario MAP-005.

|  |  |  |
| --- | --- | --- |
| Caso de prueba Nº: 5 | | |
| Objetivo de caso de prueba | Validar que, se elimine la capa seleccionada, y a su vez todos las subcapas que pertenezcan a la misma. | |
| Identificador | CP\_MAP\_005 | |
| Nombre del caso | Validar eliminación de capas. | |
| Precondiciones | Se tiene que tener ingresada la capa a eliminar. | |
| Paso | Resultado Esperado | Resultado Real |
| 1.Se debe dar click en la opción de mapa del menú. | Se debe abrir la pantalla de áreas. | OK. |
| 2.Se debe dar click en el nombre de la propiedad. | Se debe abrir el panel de capas y áreas que pertenecen a la propiedad. | OK. |
| 3.Se debe dar click en el botón de eliminar. | Se debe desplegar un mensaje de que se eliminó correctamente la capa. | OK. |

## CP\_MAP\_006

Tabla 33.

Caso de prueba perteneciente a la historia de usuario MAP-006.

|  |  |  |
| --- | --- | --- |
| Caso de prueba Nº: 6 | | |
| Objetivo de caso de prueba | Validar que, se muestre correctamente la simbología en el mapa. | |
| Identificador | Es código identificador del caso de prueba | |
| Nombre del caso | CP\_MAP\_006 | |
| Precondiciones | Se debe tener ingresado datos en el catálogo de áreas. | |
| Paso | Resultado Esperado | Resultado Real |
| 1.Se debe dar click en la opción de mapa del menú. | Se debe abrir la pantalla de áreas. Ahí se debe observar la simbología perteneciente a las propiedades ingresadas. | OK. |
| 2.Se debe dar click en el nombre de la propiedad. | Se debe abrir el panel de capas y áreas que pertenecen a la propiedad. Además se debe de observar en la esquina inferior derecha la simbología ingresada a través del catálogo de áreas. | OK. |

## CP\_MAP\_007

Tabla 34.

Caso de prueba perteneciente a la historia de usuario MAP-007.

|  |  |  |
| --- | --- | --- |
| Caso de prueba Nº: 7 | | |
| Objetivo de caso de prueba | Validar que, se muestren correctamente todas las propiedades ingresadas en la pantalla de inicio de mapa. | |
| Identificador | CP\_MAP\_007 | |
| Nombre del caso | Validar áreas de propiedades. | |
| Precondiciones | Se debe haber ingresado distintas propiedades. | |
| Paso | Resultado Esperado | Resultado Real |
| 1.Se debe dar click en la opción de mapa del menú. | Se debe abrir la pantalla de áreas. Ahí se debe observar las propiedades pintadas en el mapa, además se debe poder observar los nombres en el panel de la izquierda. | OK. |

## CP\_MAP\_008

Tabla 35.

Caso de prueba perteneciente a la historia de usuario MAP-008.

|  |  |  |
| --- | --- | --- |
| Caso de prueba Nº: 8 | | |
| Objetivo de caso de prueba | Validar que, se despliegue correctamente el panel de capas pertenecientes a la propiedad o finca. | |
| Identificador | CP\_MAP\_008 | |
| Nombre del caso | Validar visualización de capas | |
| Precondiciones | Se debe haber ingresado capas dentro de la propiedad. | |
| Paso | Resultado Esperado | Resultado Real |
| 1.Se debe dar click en la opción de mapa del menú. | Se debe abrir la pantalla de áreas. | OK. |
| 2.Se debe dar click en el nombre de la propiedad. | Se debe abrir el panel de capas y áreas que pertenecen a la propiedad. | OK. |
| 3.Se debe dar click en el nombre de la capa o subcapa. | Se debe mostrar las capas hijas de la seleccionada. | OK. |
| 4. Se debe dar click en el nombre de la capa o subcapa. | Se debe cerrar las capas en el panel. | OK. |

## CP\_MAP\_009

Tabla 36.

Caso de prueba perteneciente a la historia de usuario MAP-009.

|  |  |  |
| --- | --- | --- |
| Caso de prueba Nº: 9 | | |
| Objetivo de caso de prueba | Validar que, se centre el mapa tomando en cuenta las figuras graficadas. | |
| Identificador | CP\_MAP\_009 | |
| Nombre del caso | Validar centrar mapa en propiedades o fincas. | |
| Precondiciones | Se debe tener ingresado propiedades. | |
| Paso | Resultado Esperado | Resultado Real |
| 1.Se debe dar click en la opción de mapa del menú. | Se debe abrir la pantalla de áreas. | OK. |
| 2.Se debe dar click en el botón de centrar mapa, que se encuentra en la parte superior derecha del mapa. | Se debe centrar el mapa en el centro de las propiedades ingresadas, de tal manera que se puedan visualizar todas. | OK. |

## CP\_MAP\_010

Tabla 37.

Caso de prueba perteneciente a la historia de usuario MAP-010.

|  |  |  |
| --- | --- | --- |
| Caso de prueba Nº: 10 | | |
| Objetivo de caso de prueba | Validar que, se puedan ocultar y mostrar las diferentes áreas en el mapa. | |
| Identificador | CP\_MAP\_010 | |
| Nombre del caso | Validar mostrar/ocultar áreas. | |
| Precondiciones | Se debe tener ingresado propiedades.  Se debe tener ingresado áreas dentro de las propiedades. | |
| Paso | Resultado Esperado | Resultado Real |
| 1.Se debe dar click en la opción de mapa del menú. | Se debe abrir la pantalla de áreas. | OK. |
| 2.Se debe dar click en el botón de ocultar área. | Se debe desaparecer el grafico perteneciente al área seleccionada en el mapa. | OK. |
| 3. Se debe dar click en el botón de mostrar área. | Se debe aparecer el grafico perteneciente al área seleccionada en el mapa. | OK. |
| 4.Se debe dar click en el nombre de la propiedad. | Se debe abrir el panel de capas y áreas que pertenecen a la propiedad. | OK. |
| 5.Se debe click en el botón ocultar de alguna capa. | Se debe desaparecer el grafico perteneciente al área seleccionada en el mapa. | OK. |
| 6. Se debe click en el botón de mostrar de la capa ocultada. | Se debe aparecer el grafico perteneciente al área seleccionada en el mapa. | OK. |

## CP\_MAP\_012

Tabla 38.

Caso de prueba perteneciente a la historia de usuario MAP-012.

|  |  |  |
| --- | --- | --- |
| Caso de prueba Nº: 12 | | |
| Objetivo de caso de prueba | Validar que, se pueda visualizar la lista de atributos de cada área dentro del mapa. | |
| Identificador | CP\_MAP\_012 | |
| Nombre del caso | Validar visualización de lista de atributos en mapa. | |
| Precondiciones | Se debe haber agregado datos de atributos.  Se debe haber agregado datos de atributos a un área. | |
| Paso | Resultado Esperado | Resultado Real |
| 1.Se debe dar click en la opción de mapa del menú. | Se debe abrir la pantalla de áreas. | OK. |
| 2.Se debe dar click en el nombre de la propiedad. | Se debe abrir el panel de capas y áreas que pertenecen a la propiedad. | OK. |
| 3.Se debe dar click en el nombre de la capa que tenga atributos. | Se debe desplegar un panel en el lado derecho del mapa, el mismo que debe contener la lista de atributos ingresados en dicha área. | OK. |

## CP\_MAP\_013

Tabla 39.

Caso de prueba perteneciente a la historia de usuario MAP-013.

|  |  |  |
| --- | --- | --- |
| Caso de prueba Nº: 13 | | |
| Objetivo de caso de prueba | Validar que, se muestre el nombre del área dibujada en el mapa. | |
| Identificador | CP\_MAP\_013 | |
| Nombre del caso | Validar despliegue de nombre de área. | |
| Precondiciones | Se debe haber agregado sub áreas a la propiedad. | |
| Paso | Resultado Esperado | Resultado Real |
| 1.Se debe dar click en la opción de mapa del menú. | Se debe abrir la pantalla de áreas. | OK. |
| 2.Se debe dar click en el nombre de la propiedad. | Se debe abrir el panel de capas y áreas que pertenecen a la propiedad. | OK. |
| 3.Se debe dar click en el nombre de alguna subcapa. | Se debe mostrar las áreas pertenecientes a esa capa, siendo estas dibujadas en el mapa. | OK. |
| 4.Se debe dar click sobre alguna área del mapa. | Se debe mostrar el nombre del área en un popup. | OK. |

## CP\_ATR\_001

Tabla 40.

Caso de prueba perteneciente a la historia de usuario ATR-001.

|  |  |  |
| --- | --- | --- |
| Caso de prueba Nº: 14 | | |
| Objetivo de caso de prueba | Validar que, se ingrese correctamente atributos. | |
| Identificador | CP\_ATR\_001 | |
| Nombre del caso | Validar ingreso de atributos. | |
| Precondiciones | Se debe tener ingresado unidades para el atributo. | |
| Paso | Resultado Esperado | Resultado Real |
| 1.Se debe dar click en la opción de atributos del menú. | Se debe abrir la pantalla de atributos. | OK. |
| 2.Se debe dar click en el botón de agregar atributos. | Se debe desplegar una ventana con un formulario de ingreso de atributos. | OK. |
| 3.Se debe llenar todos los campos. | Ninguno. | OK. |
| 4.Se debe dar click en guardar. | Se debe mostrar un mensaje de ingreso correcto. Además, se debe poder ver el atributo ingresado en la pantalla inicial de atributos. | OK. |

## CP\_ATR\_002

Tabla 41.

Caso de prueba perteneciente a la historia de usuario ATR-002.

|  |  |  |
| --- | --- | --- |
| Caso de prueba Nº: 15 | | |
| Objetivo de caso de prueba | Validar que, se ingrese un atributo a un área, el mismo que debe vincularse al área dándole un valor. | |
| Identificador | CP\_ATR\_002 | |
| Nombre del caso | Validar asignación de atributo a un área. | |
| Precondiciones | Se debe haber ingresado datos de atributos.  Se debe haber ingresado datos de áreas. | |
| Paso | Resultado Esperado | Resultado Real |
| 1.Se debe dar click en la opción de asignar atributos a áreas del menú. | Se debe abrir la pantalla de asignación de atributos a áreas. | OK. |
| 2.Se debe dar click en el botón de agregar atributo a área. | Se debe desplegar una ventana con un formulario de asignación a un área. | OK. |
| 3.Se debe llenar los campos del formulario. | Ninguna. | OK. |
| 4.Se debe dar click en guardar. | Se debe desplegar un mensaje asignación correcta. | OK. |

## CP\_ATR\_003

Tabla 42.

Caso de prueba perteneciente a la historia de usuario ATR-003.

|  |  |  |
| --- | --- | --- |
| Caso de prueba Nº: 16 | | |
| Objetivo de caso de prueba | Validar que, se actualice correctamente el nombre de un área, siendo este propiedad o capa. | |
| Identificador | CP\_ATR\_003 | |
| Nombre del caso | Validar actualización de nombre de área. | |
| Precondiciones | Se debe haber ingresado un área. | |
| Paso | Resultado Esperado | Resultado Real |
| 1.Se debe dar click en la opción de mapa del menú. | Se debe abrir la pantalla de áreas. | OK. |
| 2.Se debe dar click en el botón de actualizar. | Se debe abrir una ventana que contiene un campo de actualización de nombre. | OK. |
| 3.Se debe cambiar el nombre del área. | Ninguna. | OK. |
| 4.Se debe dar click en actualizar. | Se debe mostrar un mensaje de área actualizada. | OK. |

## CP\_ATR\_004

Tabla 43.

Caso de prueba perteneciente a la historia de usuario ATR-004.

|  |  |  |
| --- | --- | --- |
| Caso de prueba Nº: 17 | | |
| Objetivo de caso de prueba | Validar que, se elimine correctamente el atributo seleccionado, este tiene que borrar también en cascada si tiene el atributo asignado a un área. | |
| Identificador | CP\_ATR\_004 | |
| Nombre del caso | Validar eliminación de atributo. | |
| Precondiciones | Debe haber ingresado atributos. | |
| Paso | Resultado Esperado | Resultado Real |
| 1.Se debe dar click en la opción de atributos del menú. | Se debe abrir la pantalla de atributos. | OK. |
| 2.Se debe dar click en el botón de eliminar atributo. | Se debe desplegar un mensaje de que se eliminó correctamente el atributo. | OK. |

## CP\_ATR\_005

Tabla 44.

Caso de prueba perteneciente a la historia de usuario ATR-005.

|  |  |  |
| --- | --- | --- |
| Caso de prueba Nº: 18 | | |
| Objetivo de caso de prueba | Validar que, se pueda actualizar la información de los atributos ingresados, estos deberían también mostrarse actualizados si están vinculados a un área. | |
| Identificador | CP\_ATR\_005 | |
| Nombre del caso | Validar actualización de atributo. | |
| Precondiciones | Debe haber ingresado atributos. | |
| Paso | Resultado Esperado | Resultado Real |
| 1.Se debe dar click en la opción de atributos del menú. | Se debe abrir la pantalla de atributos. | OK. |
| 2.Se debe dar click en el botón de actualizar atributo. | Se debe desplegar una ventana que contiene los datos actuales del atributo. | OK. |
| 3.Se debe actualizar los datos que sean requeridos en el formulario. | Ninguna. | OK. |
| 4.Se debe dar click en el botón de actualizar. | Se debe desplegar un mensaje de que se ha actualizado correctamente. | OK. |

## CP\_ATR\_006

Tabla 45.

Caso de prueba perteneciente a la historia de usuario ATR-006.

|  |  |  |
| --- | --- | --- |
| Caso de prueba Nº: 19 | | |
| Objetivo de caso de prueba | Validar que, se pueda visualizar la lista de todos los atributos ingresados, estos deben desplegar todos los campos del atributo. | |
| Identificador | CP\_ATR\_006 | |
| Nombre del caso | Validar visualización de atributos. | |
| Precondiciones | Debe haber ingresado atributos. | |
| Paso | Resultado Esperado | Resultado Real |
| 1.Se debe dar click en la opción de atributos del menú. | Se debe abrir la pantalla de atributos. Y se debe mostrar una tabla con la lista de atributos ingresados, con todos sus campos. | OK. |

## CP\_ATR\_007

Tabla 46.

Caso de prueba perteneciente a la historia de usuario ATR-007.

|  |  |  |
| --- | --- | --- |
| Caso de prueba Nº: 20 | | |
| Objetivo de caso de prueba | Validar que, se pueda visualizar cada serie de atributos que han sido asignados a un área. | |
| Identificador | CP\_ATR\_007 | |
| Nombre del caso | Validar visualización de atributos asignados a áreas. | |
| Precondiciones | Se debe haber asignado atributos a áreas. | |
| Paso | Resultado Esperado | Resultado Real |
| 1.Se debe dar click en la opción de atributos asignados a áreas del menú. | Se debe abrir la pantalla de atributos asignados a áreas. Y se debe mostrar un combo box para seleccionar el área del cual se requiere ver sus atributos. | OK. |
| 2.Se debe seleccionar el área en el combo box. | Se debe mostrar una tabla con todos los atributos asignados al área con sus respetivos valores. | OK. |

## CP\_CAT\_002

Tabla 47.

Caso de prueba perteneciente a la historia de usuario CAT-002.

|  |  |  |
| --- | --- | --- |
| Caso de prueba Nº: 22 | | |
| Objetivo de caso de prueba | Validar que, se ingrese un tipo de área dentro del catálogo de áreas. Se debe validar que se ingresen todos los campos del formulario. | |
| Identificador | CP\_CAT\_002 | |
| Nombre del caso | Validar ingreso de datos en catálogo de áreas. | |
| Precondiciones | Ninguna. | |
| Paso | Resultado Esperado | Resultado Real |
| 1.Se debe dar click en la opción de catálogo del menú. | Se debe abrir la pantalla de catálogos. | OK. |
| 2.Se debe dar click en el botón de agregar. | Se debe desplegar una ventana con el formulario de ingreso de catálogo. | OK. |
| 3.Se debe llenar todos los campos del formulario. | Ninguno. | OK. |
| 4.Se debe dar click en guardar. | Se debe desplegar un mensaje de ingreso correcto. | OK. |

## CP\_CAT\_003

Tabla 48.

Caso de prueba perteneciente a la historia de usuario CAT-003.

|  |  |  |
| --- | --- | --- |
| Caso de prueba Nº: 23 | | |
| Objetivo de caso de prueba | Validar que, se pueda visualizar los datos del catálogo ingresado, con todos sus respectivos campos. | |
| Identificador | CP\_CAT\_003 | |
| Nombre del caso | Validar visualización de datos del catálogo. | |
| Precondiciones | Se debe haber ingresado datos en el catálogo. | |
| Paso | Resultado Esperado | Resultado Real |
| 1.Se debe dar click en la opción de catálogo del menú. | Se debe abrir la pantalla de catálogos. Además, se debe poder visualizar una tabla que contiene la lista de datos ingresados con sus respectivos campos. | OK. |

## 

## CP\_CAT\_004

Tabla 49.

Caso de prueba perteneciente a la historia de usuario CAT-004.

|  |  |  |
| --- | --- | --- |
| Caso de prueba Nº: 24 | | |
| Objetivo de caso de prueba | Validar que, se pueda eliminar algún dato del catálogo. | |
| Identificador | CP\_CAT\_004 | |
| Nombre del caso | Validar eliminación de catálogo. | |
| Precondiciones | Se debe haber ingresado datos en el catálogo. | |
| Paso | Resultado Esperado | Resultado Real |
| 1.Se debe dar click en la opción de catálogo del menú. | Se debe abrir la pantalla de catálogos. | OK. |
| 2.Se debe dar click en el botón de eliminar. | Se debe mostrar un mensaje de que se eliminó correctamente el dato del catálogo. | OK. |

## CP\_CAT\_005

Tabla 50.

Caso de prueba perteneciente a la historia de usuario CAT-005.

|  |  |  |
| --- | --- | --- |
| Caso de prueba Nº: 25 | | |
| Objetivo de caso de prueba | Validar actualización de ítem del catálogo, este no debe permitir dejar datos en blanco. | |
| Identificador | CP\_CAT\_005 | |
| Nombre del caso | Validar actualización de ítem del catálogo. | |
| Precondiciones | Se debe haber ingresado datos en el catálogo. | |
| Paso | Resultado Esperado | Resultado Real |
| 1.Se debe dar click en la opción de catálogo del menú. | Se debe abrir la pantalla de catálogos. | OK. |
| 2.Se debe dar click en el botón de actualizar. | Se debe desplegar una ventana que contenga un formulario con los campos ingresados del ítem. | OK. |
| 3.Se debe actualizar los campos del formulario que se requieran. | Ninguno. | OK. |
| 4.Se debe dar click en el botón de actualizar. | Se debe desplegar un mensaje de actualización correcta. | OK. |

## Conclusión de capítulo

El ejecutar los casos de prueba nos ayudaron a validar el correcto funcionamiento de las historias de usuario, por lo que en algunos casos se tuvo que corregir algunas funcionalidades, y con lo cual se pudo evaluar la calidad de cada entrega.

# CONCLUSIONES Y RECOMENDACIONES

Como último capítulo del documento, se tiene las conclusiones que fueron el resultado de haber desarrollado el trabajo de titulación, además se obtuvieron algunas recomendaciones que serán expuestas en sus respectivos puntos.

## Conclusiones

Al iniciar el proyecto, la única manera de poder visualizar las áreas de interés de la finca era a través de archivos que se generaban sin exactitud a través de herramientas como AutoCAD, por lo que no era posible poder acceder a esta información sin estas herramientas, con lo que ahora el poder evidenciar las áreas de interés dentro de la finca a través de un mapa, permitió que el gerente pueda conectarse desde cualquier lugar con conexión a internet y ver el mapa con total exactitud, lo que permite mostrar a inversionistas o interesados de la finca ver como se está manejando el terreno sin estar físicamente en el lugar.

Como resultado de poder insertar cada área de la finca en la base de datos, nos proporcionó la facilidad de incluir atributos, los mismos que proporcionan información con valor a cada una de las áreas, por lo que ahora se puede sacar información sobre plantas sembradas por lote, plantas por sembrar, área bruta, entre otras de las características más importantes de cada área las mismas que ayudarán a restringir las ordenes de trabajo que se elaboren a futuro, para evitar robos a la hora de realizar planificaciones con valores erróneos y ficticios sobre las áreas de la finca.

Utilizar una base de datos con extensión GIS como lo fue PostGIS, ayudo mucho con el desarrollo del proyecto, esto debido a la agilidad con la que nos ayuda a gestionar los datos espaciales dentro de la base de datos, además de que este nos ayudaba con funciones para sacar la información como JSON y calcular distancias entre puntos, sin tener que utilizar librerías.

El aplicar casos de prueba durante el final de cada Sprint, permitió verificar el funcionamiento de cada historia de usuario realizada. Por lo cual, al seguir la estructura de pruebas y ver el correcto funcionamiento de las historias de usuario, se pudo dar un criterio de calidad del software, este teniendo buenos resultados para el usuario final.

## Recomendaciones

Si se va a leer archivos con puntos o coordenadas georreferenciadas es necesario tener en cuenta el sistema de coordenadas con los que son recibidos, ya que la herramienta utilizada para mostrar estos datos espaciales (Leaflet) funciona con el sistema de coordenadas GD que son latitud y longitud, y las que se recibían por parte de los archivos eran UTM, lo que generó problemas a la hora de graficar las áreas, así que es recomendable que se maneje por defecto un sistema de coordenadas para no tener que hacer conversiones del lado del servidor.

Es recomendable poner a lo largo del proyecto “tooltips” en botones, o componentes en los que interactúe el usuario, para brindarles información rápida y concisa sobre las acciones que pueden tomar en las vistas HTML. Esto se debe a que así podemos mejorar la experiencia de usuario al hacer aplicaciones web.

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

Agafonkin, V. (2017). *Interactive Choropleth Map*. Obtenido de http://leafletjs.com/examples/choropleth/

Alcaraz, M. (2016). *Your First Steps With the Geography Data Type*. Obtenido de http://www.vertabelo.com/blog/technical-articles/getting-started-with-postgis-your-first-steps-with-the-geography-data-type

Azaustre, C. (2017). *Qué es lo que me gusta de Vue.js*. Obtenido de https://carlosazaustre.es/que-es-lo-que-me-gusta-de-vue-js/

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

Benharosh, J. (2015). *The singleton pattern::the good, the bad, and the ugly*. Obtenido de https://phpenthusiast.com/blog/the-singleton-design-pattern-in-php

codescracker. (s.f.). *Python Programming Examples*. Obtenido de https://codescracker.com/python/program/

Davarnia, R. (2014). *Getting Started with Sass & Compass*. Obtenido de https://www.slideshare.net/robdvr/sass-compass-40996352

Fleschenberg, R. (2017). *A simple Django architecture diagram*. Obtenido de https://fleschenberg.net/django-architecture-diagram/

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

GARHWAL, T. (2009). *Logical building blocks in n-tier application architecture*. Obtenido de https://tgarhwal.wordpress.com/2009/05/07/logical-building-blocks-in-n-tier-application-architecture/

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

Github. (s.f.). *Git Large File Storage*. Obtenido de https://git-lfs.github.com/

Gorka, T. (2017). *Unit testing Vue.js : Challenges in legacy code*. Obtenido de https://blog.tuleap.org/unit-testing-vuejs-challenges-legacy-code

Gupta, S. (s.f.). *Roles of team members involved in an AGILE Scrum project*. Obtenido de http://www.quotium.com/performance/roles-team-members-involved-agile-scrum-project/

Inidicum S.A. (2018). *Módulo de áreas*. Obtenido de http://app.agmatice.com/areas/

Leading Agile. (2017). *5 Scrum Events*. Obtenido de https://leadagile.in/2017/12/18/5-scrum-events/

Leaflet. (2017). *Leaflet Draw API reference*. Obtenido de http://leaflet.github.io/Leaflet.draw/docs/leaflet-draw-latest.html

Leaver, D. (2012). *Leaflet.markercluster*. Obtenido de https://github.com/Leaflet/Leaflet.markercluster

Marquez, M. (2014). *Programming in the console*. Obtenido de http://xomalli.blogspot.com/2014/02/utilizando-los-comandos-grant-y-revoke.html

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

NPM. (s.f.). *npm3 Non-Determinism*. Obtenido de http://npm.github.io/how-npm-works-docs/npm3/non-determinism.html

Ochoa, J. (2015). *Sistema de ramificaciones en git*. Obtenido de https://styde.net/sistema-de-ramificaciones-en-git/

ochobitshacenunbyte. (2015). *Atom, un potente editor libre*. Obtenido de https://www.ochobitshacenunbyte.com/2015/04/29/atom-potente-editor-libre/

Palacios, D. (2016). *Instalación y primeros pasos con Vue.js*. Obtenido de https://styde.net/introduccion-a-vue-js/

pgAdmin. (s.f.). *pgAdmin, PostgreSQL Tools*. Obtenido de https://www.pgadmin.org/

Pilott, M. (2013). *Foundation for Beginners: Custom Forms and Switches*. Obtenido de https://webdesign.tutsplus.com/articles/foundation-for-beginners-custom-forms-and-switches--webdesign-13109

Pochechuev, I. (2011). *GeoDjango in a nutshell*. Obtenido de https://www.slideshare.net/DjangoStars/geodjango-in-a-nutshell

Posenato, R. (2008). *W3 - diseño y realización de hipertexto para sitios web*. Obtenido de http://profs.sci.univr.it/~posenato/courses/html2002/10-Css.html

Rand-Hendriksen, M. (2017). *JavaScript Essential Training*. Obtenido de https://www.linkedin.com/learning/javascript-essential-training-3

Revueltas, P. (2017). *Los 5+1 valores de equipos Scrum altamente efectivos*. Obtenido de https://www.paradigmadigital.com/techbiz/los-51-valores-de-equipos-scrum-altamente-efectivos/

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/

Scrum Manager Body Of Konwledge. (2014). *Historia de usuario*. Obtenido de http://www.scrummanager.net/bok/index.php/Historia\_de\_usuario

Serrato, M. (2013). *Cómo crear Plugins para jQuery*. Obtenido de http://html5facil.com/tutoriales/como-crear-plugins-para-jquery/

softwarecamp. (2015). *Proyectos atrasados o fuera de presupuesto*. Obtenido de http://softwarecamp.mx/capacitacion/proyectos-atrasados-o-fuera-de-presupuesto-porque-funciona-scrum/

SpryMedia Ltd. (2018). *Add advanced interaction controls to your HTML tables the free & easy way*. Obtenido de https://datatables.net/

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

Trello. (2018). *Nosotros*. Obtenido de https://trello.com/home

Vasović, Z. (2016). *How long time does it take to learn SASScss?* Obtenido de https://www.quora.com/How-long-time-does-it-take-to-learn-SASScss

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

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

W3schools. (2018). *CSS Tutorial*. Obtenido de https://www.w3schools.com/css/

W3schools. (2018). *HTML*. Obtenido de https://www.w3schools.com/html/tryit.asp?filename=tryhtml\_basic\_document

Zabelava, T. (s.f.). *Scrum task board*. Obtenido de https://www.dreamstime.com/stock-illustration-scrum-task-board-info-graphic-dark-grey-background-image70979307

Zabriskie, M. (2014). *axios*. Obtenido de https://github.com/axios/axios

Zurb. (2018). *Stencils for Foundation 4 are Now Here!* Obtenido de https://zurb.com/blog/stencils-for-foundation-4-are-now-here