

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. MARCO TEÓRICO 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. 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.

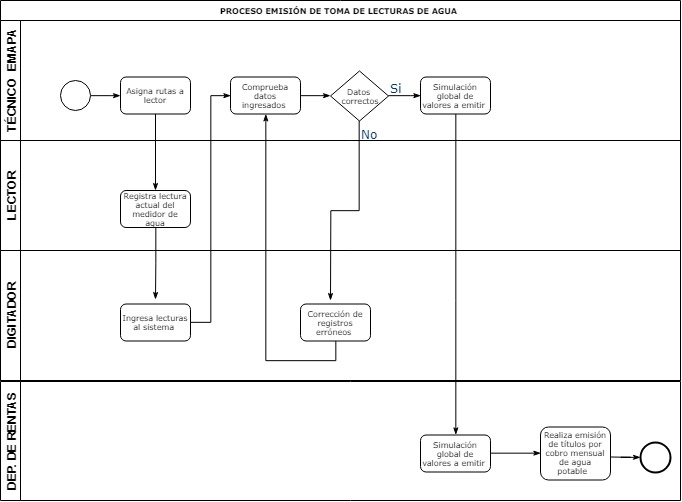
El proceso actual es deficiente porque la recolección de datos de lecturas retrasa la emisión global de planillas hasta que los digitadores concluyan el ingreso manual al sistema. Al mismo tiempo se genera la probabilidad de errores de digitación y perdida o deterioro de la hoja de ruta. Finalmente,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

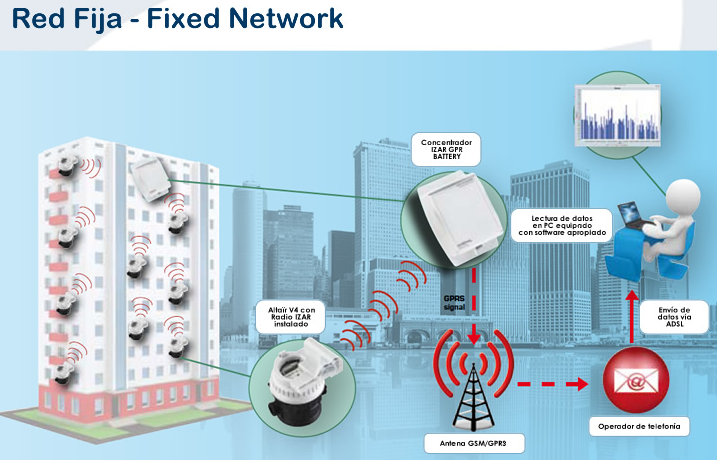


Figura 2: Medidores con red fija (Geoconta, 2018)

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 georeferenciado 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 georeferenciados.
* 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

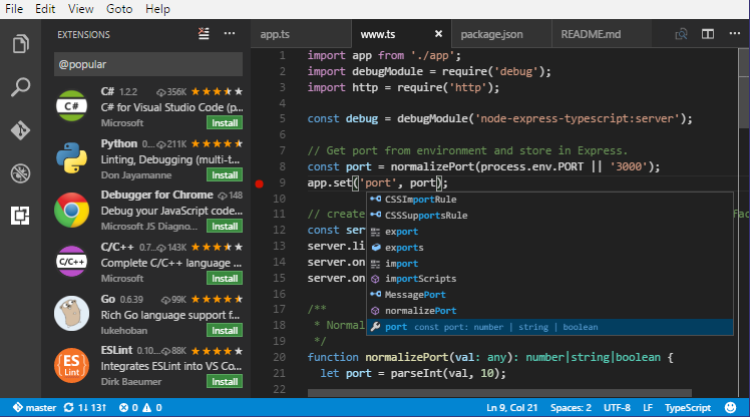


Figura 3: Interfaz gráfica de Visual studio code (VSC, 2018)

Visual Code[[1]](#footnote-1) 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.

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.

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[[2]](#footnote-2) 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.

## 

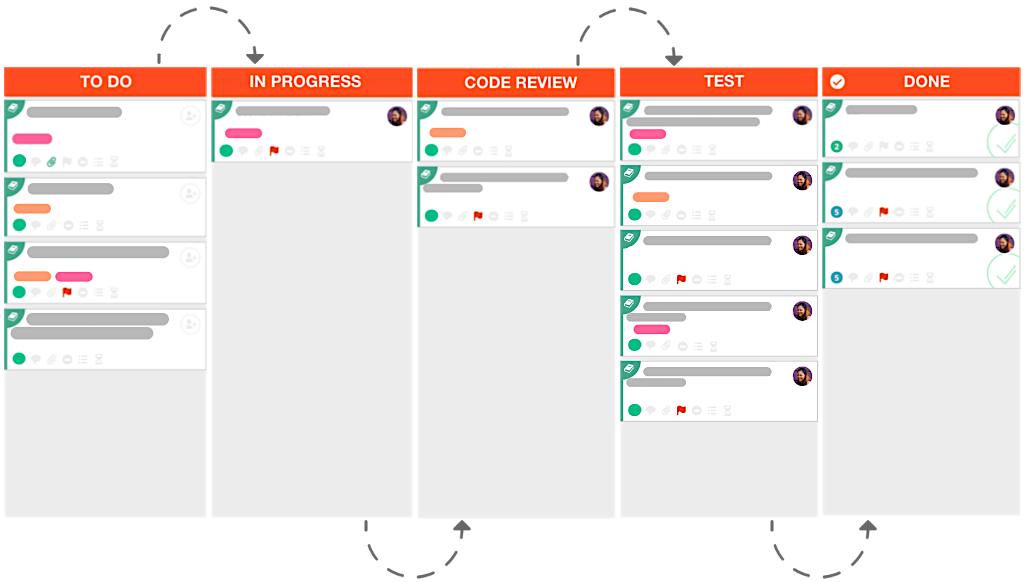


Figura 4: Tablero kaban de la herramienta vivifyScrum (Vivify, 2018)

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 nota. (la herramienyat permite componer mas columnas sin embargo kamban propone 3)

Características:

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

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

**1.3 Herramienta de prototipado de interfaces**

Las aplicaciones de prototipos de interfaz permiten al desarrollador generar un diseño previo de su interfaz gráfica de usuario, esto permite proponer al usuario diseños visualmente atractivos e intuitivos antes del desarrollo final, esto representa una ventaja ya que el usuario también podría proponer cambios que se ajusten de mejor manera a su necesidad.

**1.3.1 Marvel**

Marvel[[3]](#footnote-3) 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.

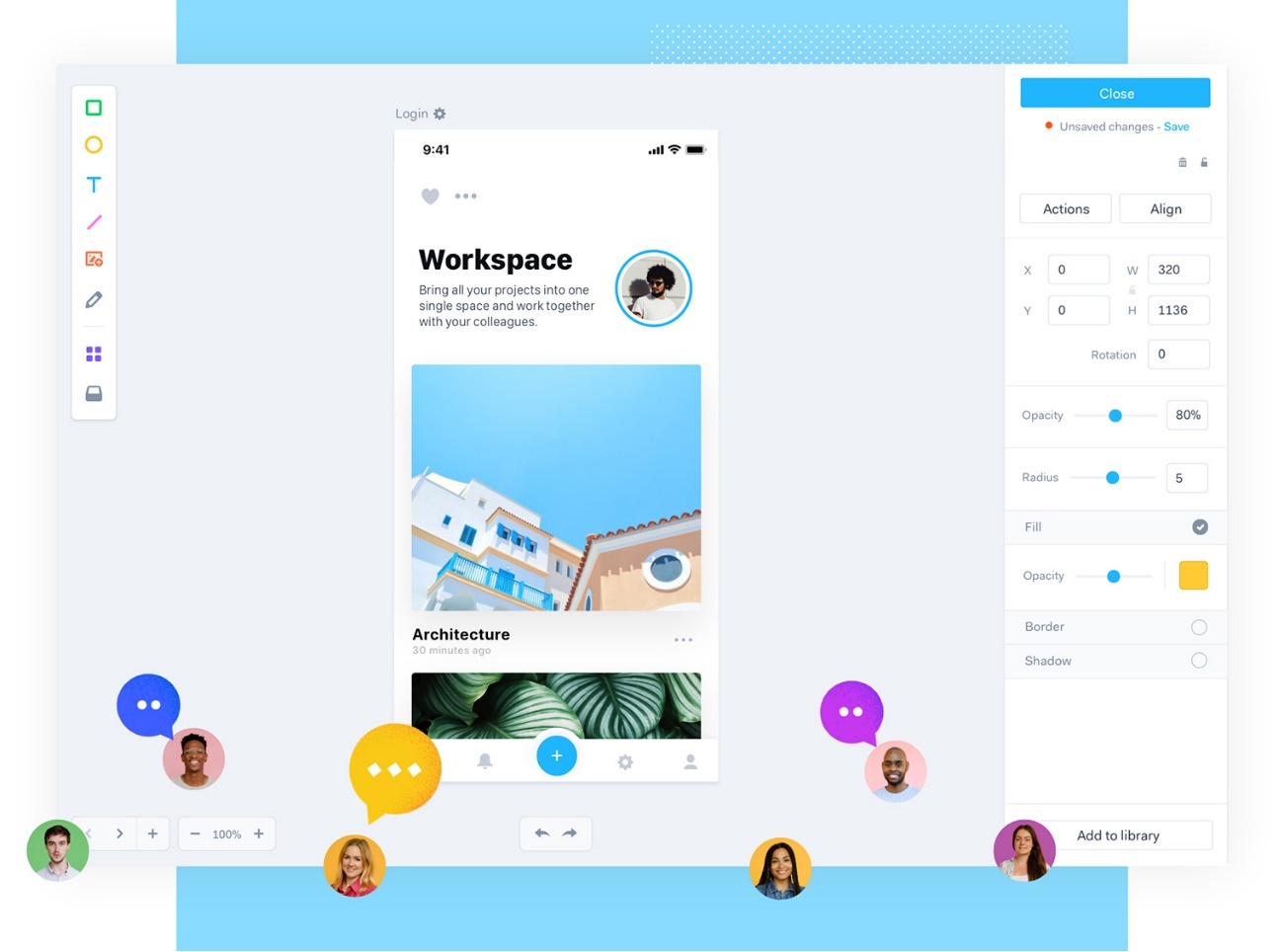


Figura 5: Canvas de diseño de interfaces Marvel App (MarvelApp, 2017)

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.

**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 6, 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.

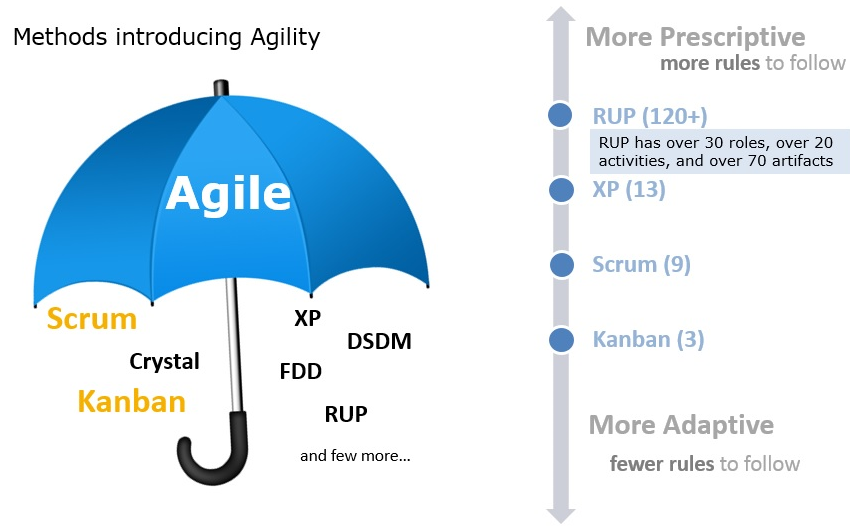


Figura 6: Principales metodologías agiles (Muñoz, 2016)

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

## 

Figura 7: Diagrama conceptual de la metodología Kanban ( Kirovska & Koceski, 2015)

## citaEn 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 manera 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 8.

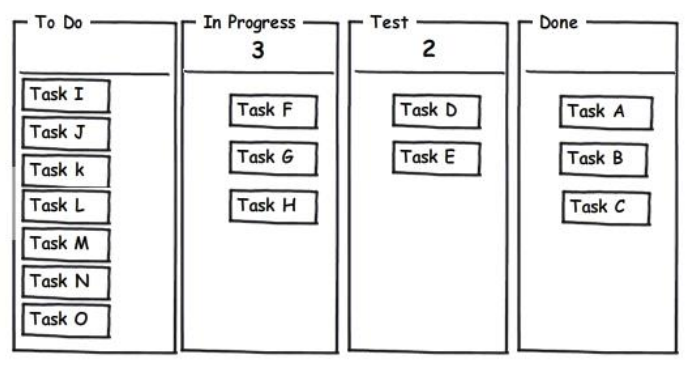
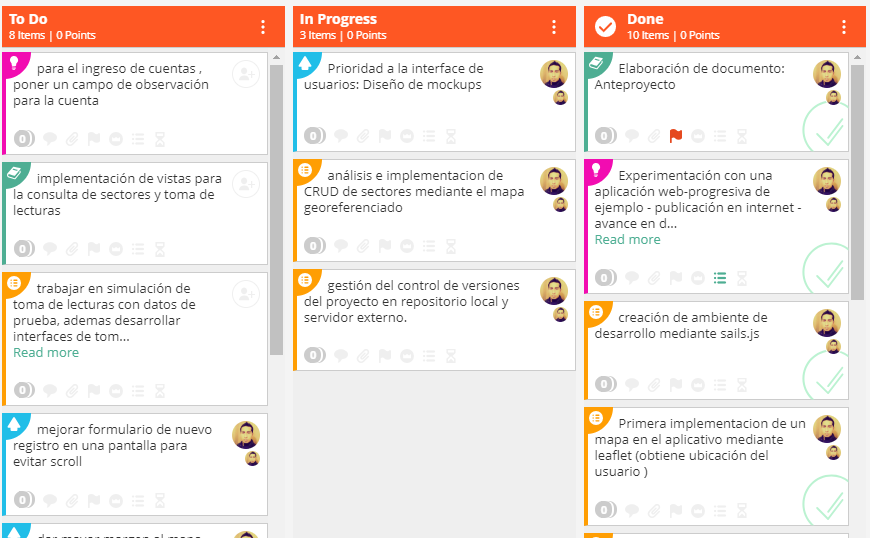


Figura 8: Tablero Kanban ( Kirovska & Koceski, 2015)



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 con flexibilidad (interacción -agrega valor al producto-flexibilidad, permite ver una vista general del avance del proyecto). Es una metodología que no tiene problemas para cambiar los requisitos del cliente, los cambios en 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 y puntualmente es el objetivo principal.

**Diseño centrado en el usuario**

Es un método agil de desarrollo de software interactivo centrado en el usuario. El elemento principal de este enfoque son las interfaces de usuario (Perez Medina & Vanderdonckt, 2018).

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 5.

El diseño centrado en el usuario permite el desarrollo de software de manera colaborativa y propone 4 configuraciones posibles. El esquema de la figura 9, 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.

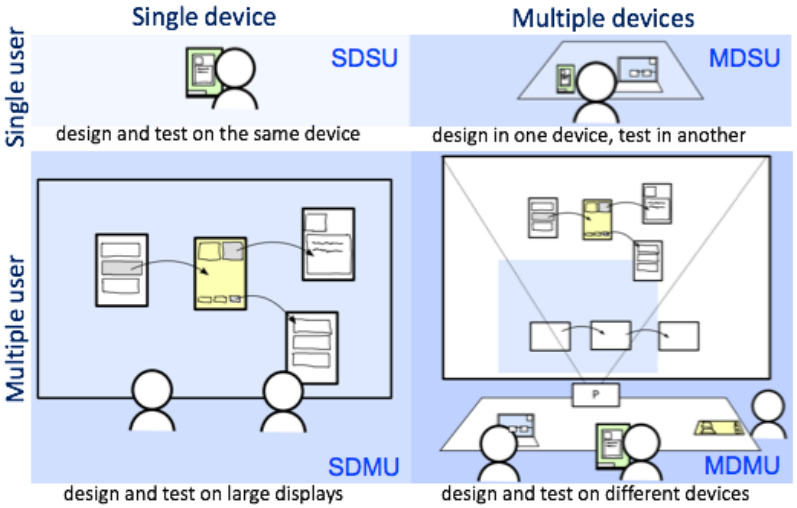


Figura 9: configuraciones físicas del diseño centrado en el usuario (Perez Medina & Vanderdonckt, 2018)

El flujo de trabajo de la figura 10, 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.



Figura 10: flujo de trabajo de prototipos de interfaces (Perez Medina & Vanderdonckt, 2018)

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

Se eligió este método porque su característica principal es involucrar al usuario haciéndolo participe del diseño de interfaces. Esto se convierte en una ventaja cuando 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 que nos permitio capturar requerimientos para la construcción de interfaces (precentar bocetos de interfaces ) en acuerdo con el diseñador, usuario final y el autor del presente trabajo ..(historia de usuario)

**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.

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

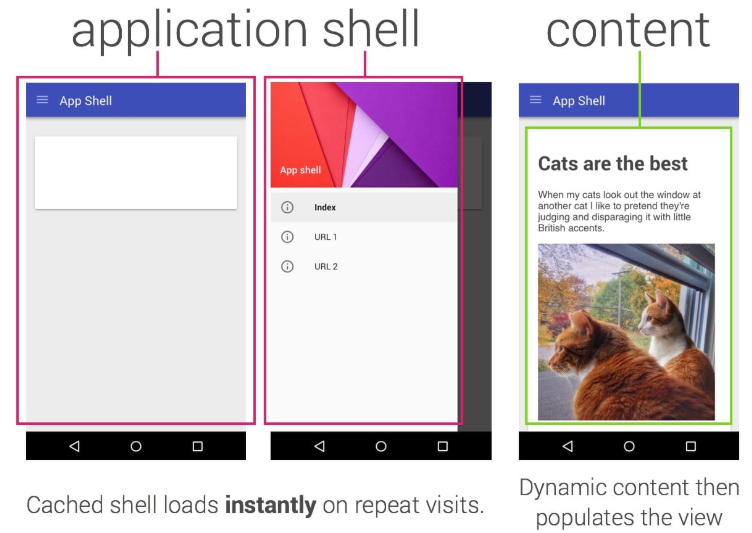


Figura 11: Shell de la aplicación web progresiva.

**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.



Figura 12

**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. 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.

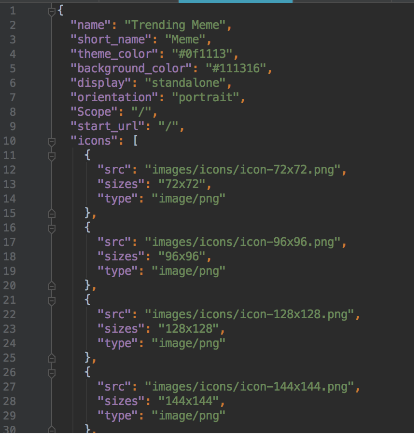
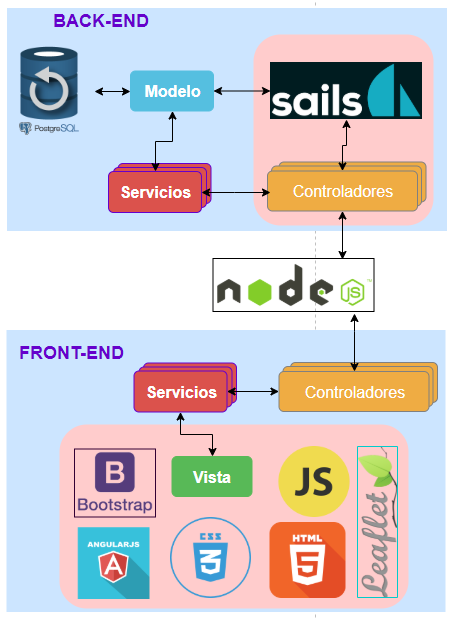
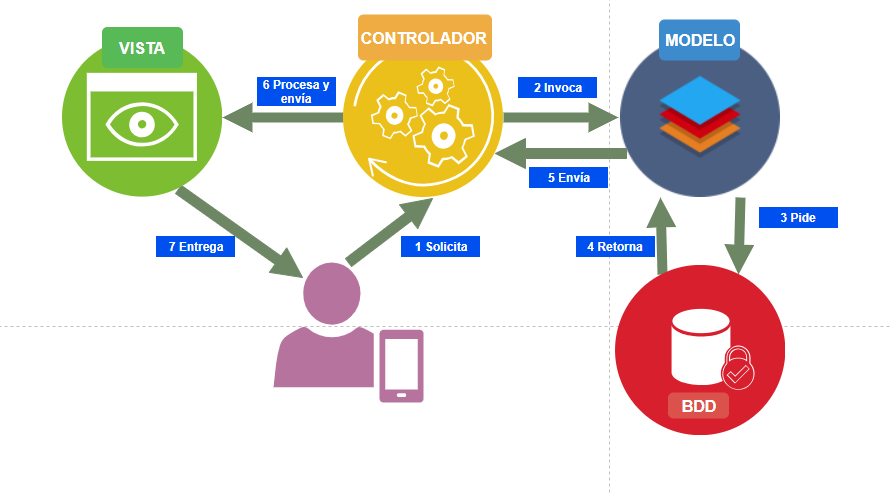


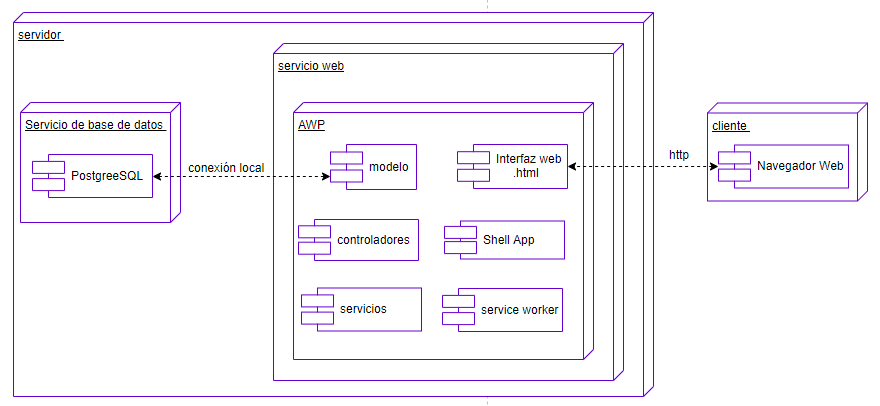
Figura 13

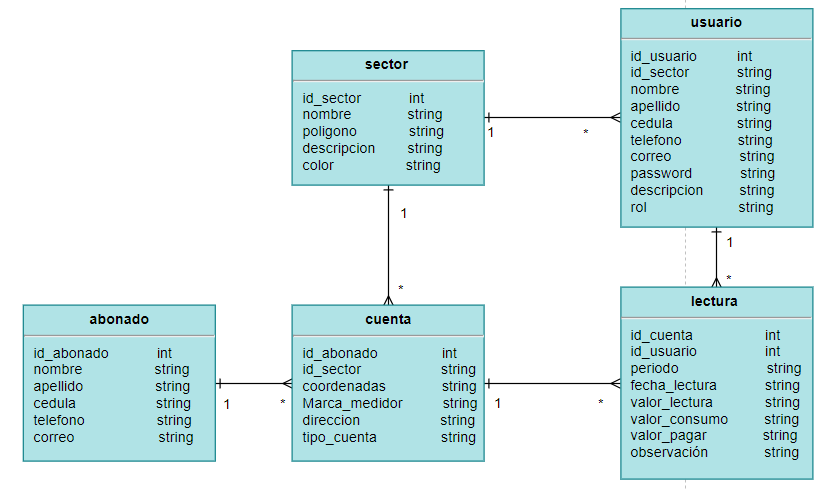
Cambiarel contraste de imágenes de codifo para q sean mas legibles

En definitiva, una aplicación web progresiva es una alternativa para la implementación de soluciones









Usabilidad

## FRONT-END

## Es un término que hace referencia a la capa de presentación en un proyecto de software, este siendo la parte que puede ver el usuario final al usar la aplicación.

## JavaScript

“JavaScript (JS) es un lenguaje ligero e interpretado, orientado a objetos con funciones de primera clase, más conocido como el lenguaje de script para páginas web, pero también usado en muchos entornos sin navegador, tales como node.js o Apache CouchDB”. (Ismail Bina, 2017)

Este lenguaje de script se ejecuta del lado del cliente del navegador, lo que significa que cualquier código de JavaScript se entrega desde el servidor hacia el cliente tal como las páginas web.



Figura 14. Pequeña porción de código JavaScript.

Tomado de: (Rand-Hendriksen, 2017)

Ventajas:

* Es muy sencillo de utilizar.
* Es veloz, por lo que tiende a ejecutar funciones inmediatamente.
* Cuenta con muchas opciones de efectos visuales.
* Es soportado por la mayoría de navegadores más populares, además es compatible con los más modernos.
* Es muy versátil, lo que hace que sea una excelente opción para desarrollar aplicaciones web.

Desventajas:

* Puesto a que está en el Front-End sus códigos son visibles, por lo tanto, pueden ser leídos por cualquier usuario, que tenga conocimientos del mismo.
* Sus scripts son muy limitados por razones de seguridad lo que hace que no sea posible realizar todo con JavaScript, por lo tanto, es necesario se debe complementar con otros lenguajes evolucionados y más seguros.

## Vue.js

Vue es un framework progresivo que sirve para construir interfaces de usuario, está diseñado desde cero lo que lo permite ser adoptable incrementalmente.



Figura 15. Comparativo Angular vs React vs Vue.

Tomado de: (Gorka, 2017)

Métodos y Datos:

Vue tiene ordenado sus distintos elementos, por lo cual, dentro de un objeto llamado “methods”, se define todas las funciones dentro de la instancia de Vue, de igual manera en un objeto llamado “data” se definen todas las variables globales para la instancia de Vue.



Figura 16. Componente Vue.

Tomado de: (Azaustre, 2017)

Componentes:

A pesar de que Vue puede utilizarse y verse como una librería muy simple, su potencia viene cuando utilizamos componentes. La orientación a componentes en arquitecturas web, por lo cual en Vue se puede crear nuevos componentes al igual que otros famosos framewroks.

Plantillas:

Vue cuenta con una ventaja en la cual en una misma plantilla podemos tener tanto el código de HTML y JavaScript de una forma muy ordenada, y al mismo tiempo esta puede ser separada si es que el usuario no desea tenerla así.

Ventajas:

* Ligero: El tamaño de este es de 18-21 KB por lo que el usuario no tarda en descargarlo y usarlo.
* Fácil de entender: Debido a que cuenta con simple estructura muy intuitiva.
* Simple al integrar: Al ser basada en JavaScript esta tiende integrarse fácilmente a otras aplicaciones web.
* Extensa documentación detallada: Existe mucha documentación altamente detallada, y con una gran cantidad comunidad de usuarios que la respaldan.

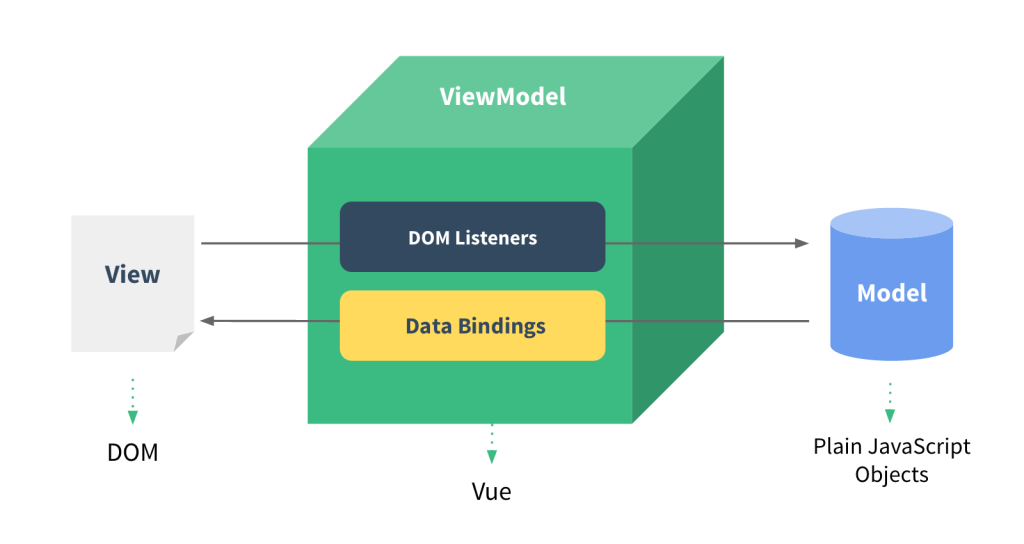


Figura 17. Diagrama de arquitectura de Vue.js

Tomado de: (Palacios, 2016)

Conclusión:

Se eligió este framework, ya que, el software al cual se debe incluir el módulo se maneja con este framework por lo que es 100% compatible, y es un estándar en la empresa para implementarlo.

## NPM

“Npm es el manejador de paquetes por defecto para Node.js, un entorno de ejecución para JavaScript.” (Npm, 2016)

Esta herramienta tiene como usos más comunes: publicar, instalar, descubrir y desarrollar programas en Node.js.

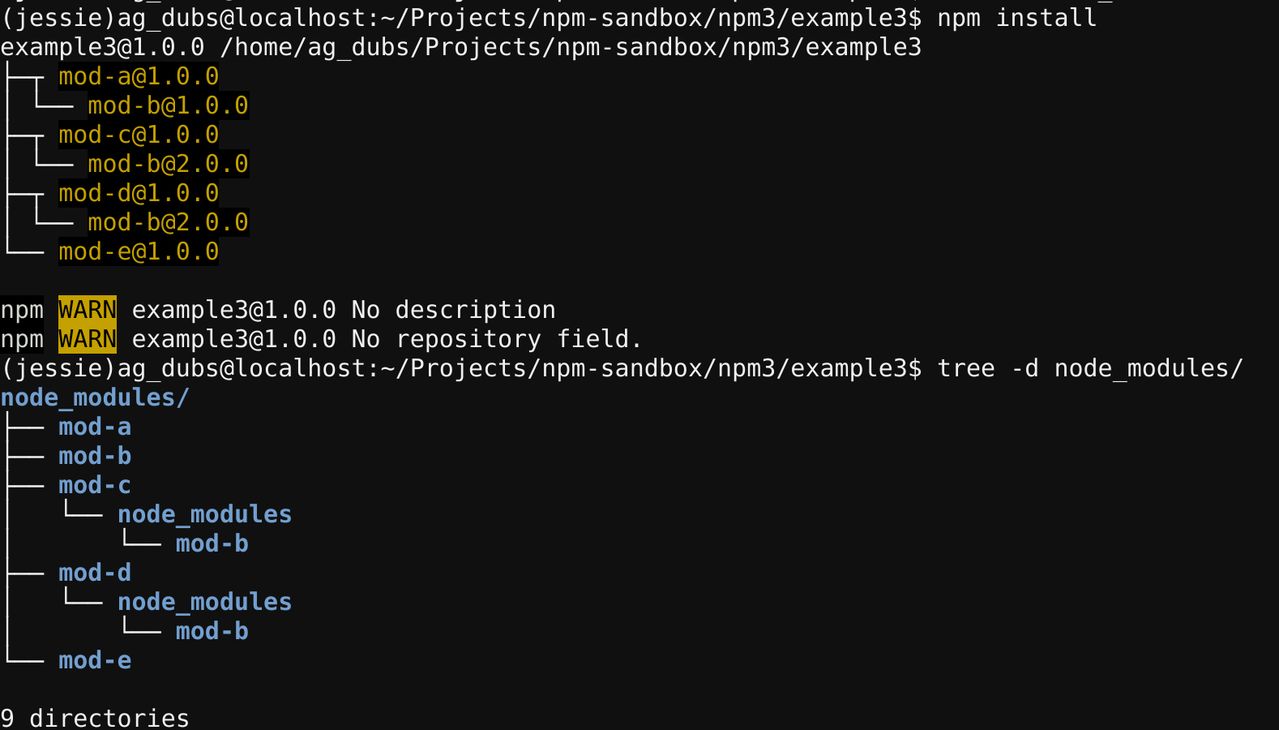


Figura 18. Consola ejecutando npm install.

Tomado de: (NPM, s.f.)

Conclusión:

Se eligió esta herramienta, ya que, el software al cual se debe incluir el módulo se maneja esta herramienta para instalar paquetes, por lo que se debe manejar el mismo esquema.

## Librerías de JavaScript

## JQuery

jQuery es una librería de JavaScript, que funciona en diversos navegadores, y que es compatible con CSS3. Su objetivo principal es hacer la programación “scripting” mucho más fácil y rápida del lado del cliente.

jQuery consiste en un único fichero JavaScript que contiene las funcionalidades comunes de DOM, eventos, efectos y AJAX.

La característica principal de la biblioteca es que permite cambiar el contenido de una página web sin necesidad de recargarla, mediante la manipulación del árbol DOM y peticiones AJAX. Para ello utiliza las funciones $()

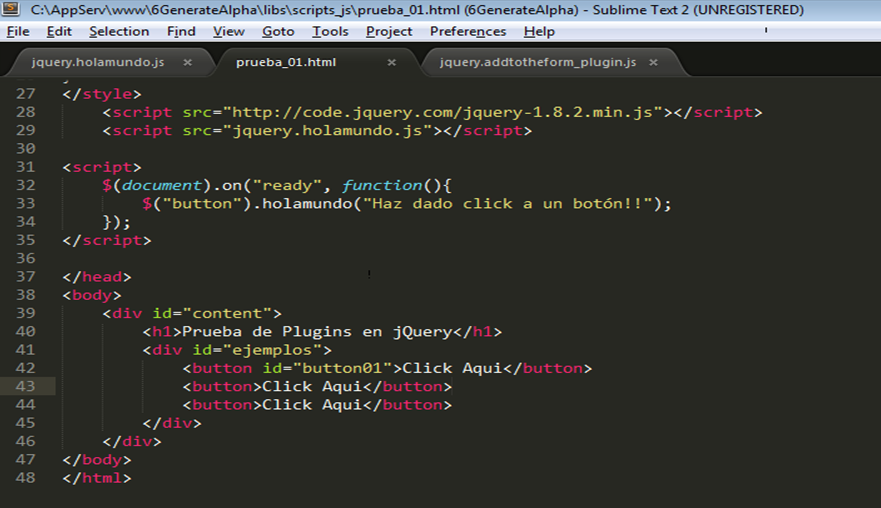


Figura 19. Script en JQuery.

Tomado de: (Serrato, 2013)

Ventajas:

* jQuery es flexible y rápido para el desarrollo web.
* Viene con licencia MIT y es open-source.
* Tiene una excelente comunidad de soporte.
* Tiene plugins.

Desventajas:

* Una de las principales desventajas de jQuery es la gran cantidad de versiones publicadas en el corto tiempo, constantemente salen nuevas versiones. No importa si usted está corriendo la última versión de jQuery, usted tendrá que conseguir y descargar la librería usted mismo y actualizarla constantemente.

## Leaflet.js

Leaflet es la biblioteca líder de código abierto de JavaScript para el uso de mapas interactivos y optimizados para dispositivos móviles.

“Leaflet está diseñado con la simplicidad, el rendimiento y la facilidad de uso en mente. Funciona de manera eficiente en todas las principales plataformas de escritorio y dispositivos móviles, se puede ampliar con muchos complementos, tiene una API hermosa, fácil de usar y bien documentada, y un código fuente simple y legible con el que es un placer contribuir.” (Leaflet, 2017).

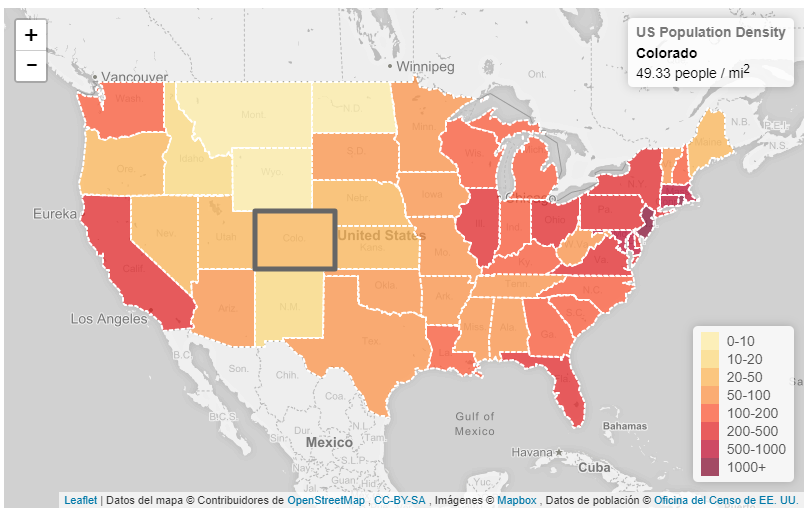


Figura 20. Demo Choropleth.

Tomado de: (Agafonkin, 2017)

## Marker Cluster

Es un plugin de Leaflet, el mismo que sirve para agrupar cientos de punto y visualizar tal y como si fuera uno.

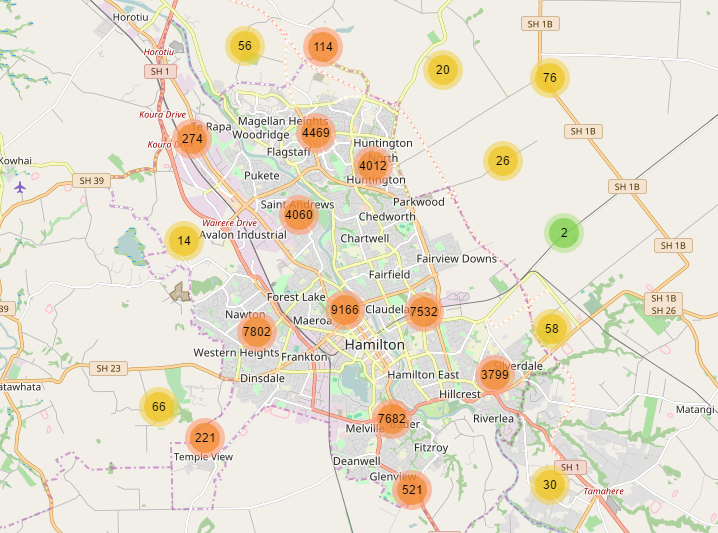


Figura 21. Demo Marker Cluster.

Tomado de: (Leaver, 2012)

Este plugin nos ayuda a optimizar y renderizar cientos y miles de puntos en la pantalla, para que así el navegador no colapse al tratar de renderizar demasiada información, por lo cual en eta librería podemos dar un nivel de zoom en el cual podemos activar y desactivar esta opción para que a la final se vea así:



Figura 22. Clúster de marcadores sin agrupación.

Tomado de: (Inidicum S.A, 2018)

## Leaflet Draw

Es un API de Leaflet que permite dibujar en el mapa cualquier tipo de polígono, línea, o punto:



Figura 23. Demo de Leaflet Draw.

Tomado de: (Leaflet, 2017)

Este API nos brinda un menú en el cual podemos agregar las opciones de dibujo para el mapa de Leaflet, en este podemos poner opciones de colores para cada tipo de gráfico, para tener una mejor experiencia de usuario.

## DataTables

Es una librería de JavaScript que junto con JQuery nos proporciona una forma interactiva de crear tablas en nuestro proyecto, con una gran cantidad de características que podemos personalizar:

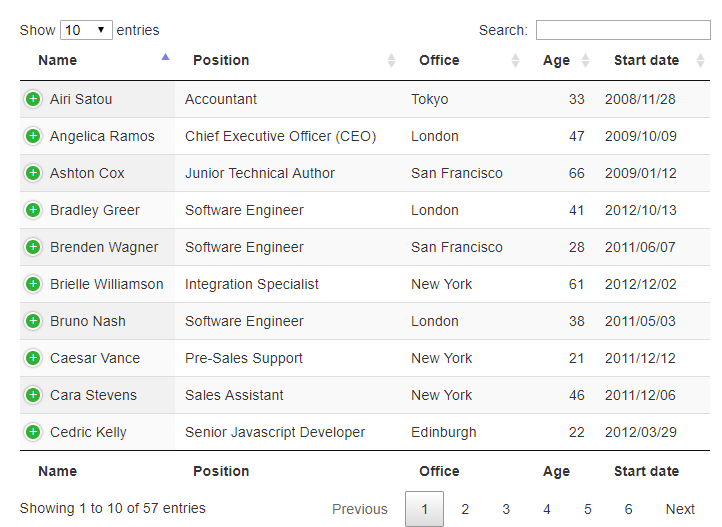


Figura 24. Ejemplo de uso de DataTables.

Tomado de: (SpryMedia Ltd, 2018)

“DataTables es un complemento para la biblioteca jQuery JavaScript. Es una herramienta altamente flexible, construida sobre los fundamentos de la mejora progresiva, que agrega todas estas funciones avanzadas a cualquier tabla HTML.” (datatblaes.net, 2017)

Características principales:

* Paginación: Navegación previa, siguiente y de página.



Figura 25. Paginación en DataTables.

Tomado de: (SpryMedia Ltd, 2018)

* Búsqueda instantánea: Filtrar los resultados por búsqueda de texto.



Figura 26. Búsquedas en DataTables.

Tomado de: (SpryMedia Ltd, 2018)

* Pedidos de múltiples columnas: Ordenar datos por múltiples columnas a la vez.

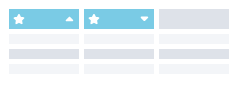


Figura 27. Filtros en DataTables.

Tomado de: (SpryMedia Ltd, 2018)

* Usa distintas fuentes de datos: DOM, JavaScript, Ajax y procesamiento del lado del servidor.
* Extensiones: Editor, botones, sensible y más.

## Axios.js

“Axios js es una librería de JavaScript construida con el objetivo de gestionar la programación asíncrona con promesas.” ([Cecilio](https://www.arquitecturajava.com/axios-js-una-libreria-de-promesas/) Álvarez, 2016)

Axios es un cliente del protocolo HTTP, basado en promesas para JavaScript, el mismo que puede ser utilizado tanto en el Front-end, como en el Back-end con Node.js. Utilizando Axios, es muy sencillo enviar peticiones a endpoints REST y realizar operaciones CRUD. Además, Axios puede ser utilizada desde una aplicación desarrollada con JavaScript plano, al igual que utilizando un Framework como Vue.

Características:

* Realizar peticiones XMLHttpRequest desde el navegador de una manera sencilla.
* Realizar peticiones HTTP desde Node.js.
* Soporta el API de Promesas.
* Intercepta peticiones y respuestas.
* Captura y transforma la información de las peticiones y respuestas.
* Cancela peticiones.
* Transforma automáticamente la información en formato JSON.

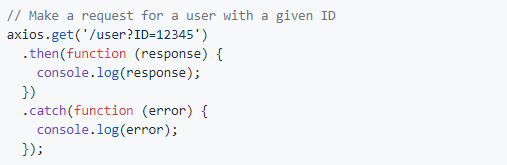


Figura 28. Ejemplo de código JS utilizando Axios.js

Tomado de: (Zabriskie, 2014)

## Diseño y Estilo

## HTML

HTML que significa lenguaje de marcado de hipertexto, es un lenguaje estándar, el mismo que se muestra en un navegador web para mostrar contenidos de sitios y aplicaciones web.

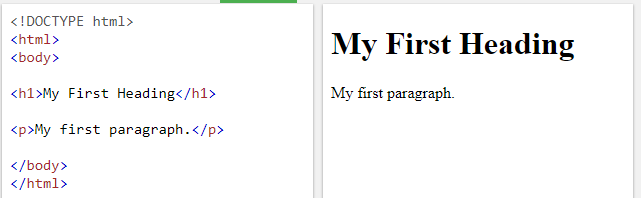


Figura 29.Ejemplo código HTML.

Tomado de: (W3schools, 2018)

Ventajas:

* Es nativo y de código abierto.
* Es compatible con todos los navegadores, incluso con los de los celulares.
* Es semánticamente fácil de interpretar, por lo que es fácil de aprender y utilizar.
* Se pueden hacer paginas ligeras sin tener que tener una conexión con un servidor.

## CSS3

CSS que significa hojas de estilo en cascada, tiene el propósito de definir estética en un sitio web, es decir que sirve como herramienta para dar colores, diseños, tamaños a diferentes componentes de HTML.

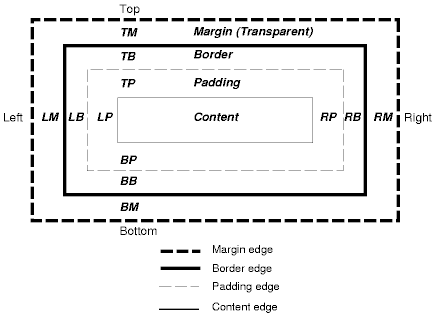


Figura 30. Estructura de elemento en HTML.

Tomado de: (Posenato, 2008)

CSS puede ser implementado dentro del código HTML, como puede ser utilizado desde un archivo externo, el mismo que debe ser referenciado en el HTML para poder utilizarlo.

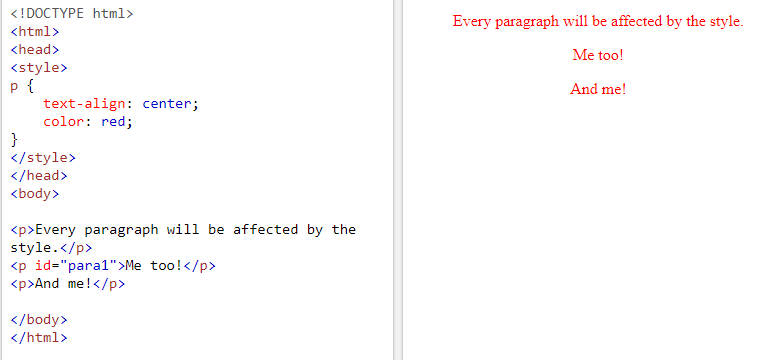


Figura 31. Estructura de Css.

Tomado de: (W3schools, 2018)

## SASS

SASS que significa hojas de estilo sintácticamente increíble, es un metalenguaje de CSS que a su vez es un script que es traducido a CSS. SASS permite junto a funciones y variables evitar menos código CSS, lo que hace que este sea más entendible que el CSS puro.



Figura 32. Diferencia entre estructura SASS y CSS.

Tomado de: (Vasović, 2016)

Ventajas:

* Se puede agrupar las reglas de CSS, por lo que se puede anidar clases de padres e hijos y reducir código CSS, tal y como se puede observar en la figura anterior.
* Se puede crear variables para evitar repetir código, y así, reducir el mismo.
* La sintaxis nos permite manejar tanto llaves y punto y comas, como a su vez una indentación, igual que en Python.



Figura 33. Proceso de conversión Sass.

Tomado de: (Davarnia, 2014)

## Foundation Zurb

Foundation es un framework para interfaz de usuario, con la característica de que es responsivo. Este framework nos proporciona una cuadricula que es responsivo, así como incluye componentes de interfaz de usuario HTML, CSS, formularios, barras de navegación, etc.

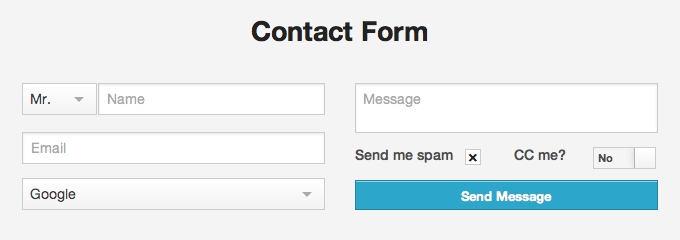


Figura 34. Ejemplo de formulario con Foundation.

Tomado de: (Pilott, 2013)

Características:

* Foundation viene con un conjunto de hojas de estilo por defecto, en el cual ya se tendrán diseños para varios de los componentes HTML, como botones, cuadros de texto, títulos, etc.
* Nos proporciona componentes en JavaScript, usando JQuery, además que una serie de plugins.
* Nos brinda un sistema de cuadricula y diseño responsivo, el mismo que no es únicamente para web, sino para dispositivos móviles.

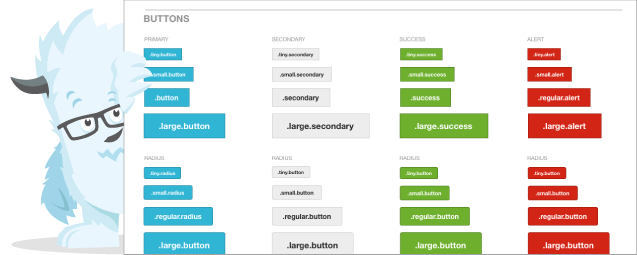


Figura 35. Ejemplo de estilos de Foundation.

Tomado de: (Zurb, 2018)

## BACK-END

## Python

Python, es un lenguaje (programación) que tiene como filosofía que su sintaxis debe favorecer a un código legible. Este lenguaje a diferencia de algunos es multiparadigma, lo que quiere decir que soporta programación orientada objetos, programación imperativa y hasta programación funcional. Así que se puede decir que es un lenguaje interpretado, multiplataforma y que usa un tipado dinámico.

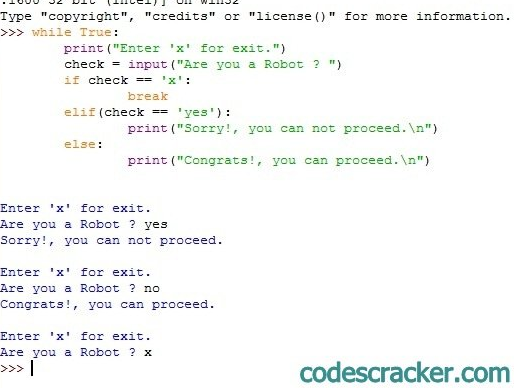


Figura 36. Ejemplo código Python.

Tomado de: (codescracker, s.f.)

Características:

* Simple.
* Código libre.
* Orientado a objetos.
* Lenguaje de alto nivel.
* Fácil de integrar.
* Extensas librerías.
* Sintaxis claras.

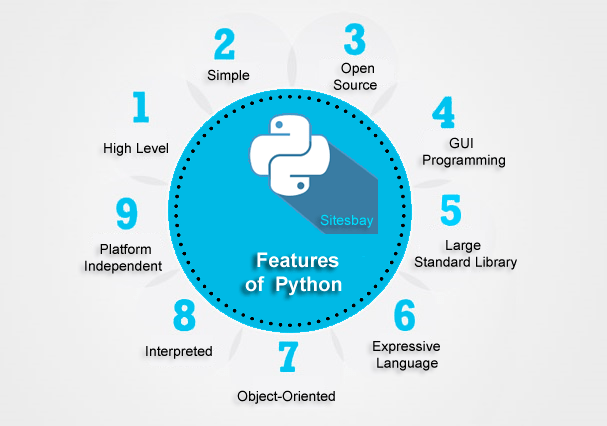


Figura 37. Características de Python.

Tomado de: (Steiner, 2018)

## Django

Django es un framework web que funciona junto con Python, siendo este de alto nivel, fomenta un rápido desarrollo y limpio diseño.

Este framework cuenta con 3 capas dentro de su arquitectura, las mismas que son:

1.) Capa Modelo:

Esta es una capa de abstracción la misma que sirve para estructurar y manipular los daros de la aplicación web. En esta capa se definen modelos, los que son la fuente definitiva de información de los datos.

2.) Capa Vista:

Esta capa sirve para encapsular la lógica que es responsable de procesar la solicitud del usuario y de responder la misma.

3.) Capa Plantilla:

Es la capa que interactúa directamente con el usuario, esta proporciona una sintaxis amigable con el diseñador.

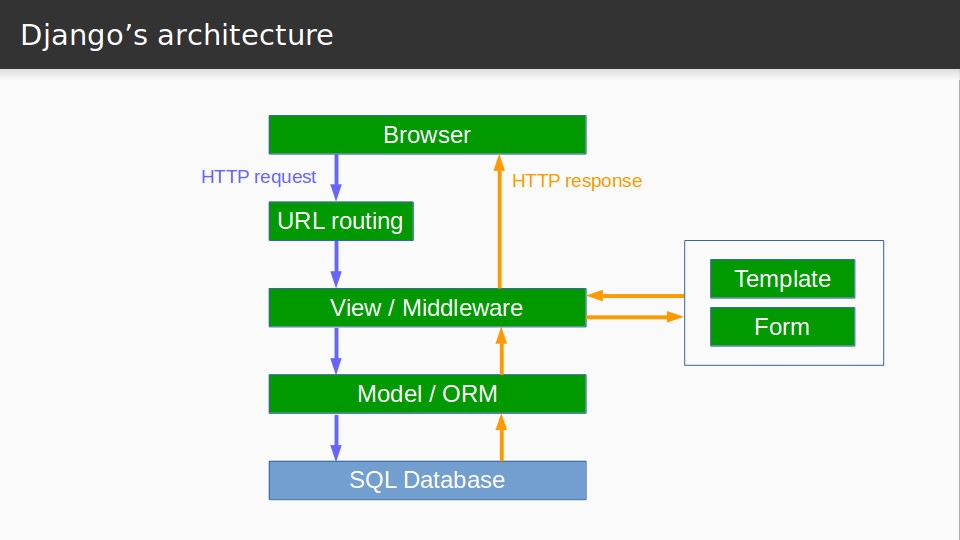


Figura 38. Diagrama de arquitectura de Django.

Tomado de: (GARHWAL, 2009)

Características:

* Rápido
* Seguro.
* Escalable.
* Versátil.
* Robusto.

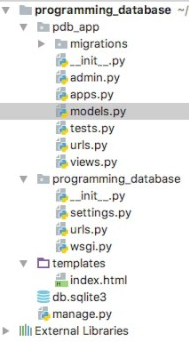


Figura 39. Estructura de archivos de Django.

Tomado de: (Steiner, 2018)

## GeoDjango

GeoDjango, es un módulo incluido para Django, y tal como su nombre lo dice es un marco web geográfico a nivel mundial. Este módulo tiene como objetivo crear aplicaciones web geográficas utilizando servicios basados en la georreferenciación.

Características:

* Posee campos para los modelos de Django que son de tipo geométricos y datos ráster.
* Contiene extensiones para el ORM, que sirven para hacer consultas personalizadas con datos espaciales.



Figura 40. Creación de modelo con GeoDjango.

Tomado de: (Pochechuev, 2011)

## Base de Datos

## PostgreSQL

PostgreSQL es uno de los más poderosos sistemas de base de datos relacional de código abierto, teniendo más 15 años de desarrollo y con una arquitectura comprobada, que le hace tener una alta reputación de fiabilidad e integridad de datos.

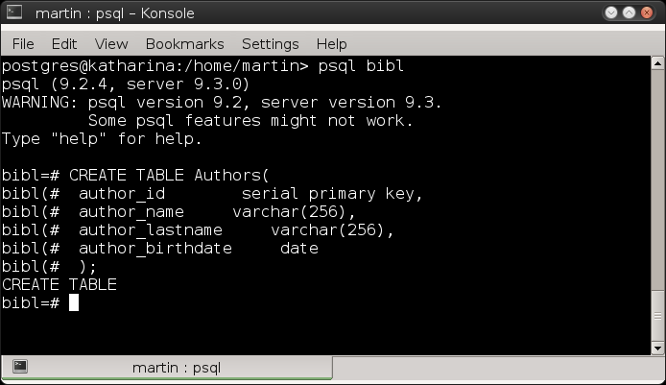


Figura 41. Consola de PostgreSQL.

Tomado de: (Marquez, 2014)

Características:

* Se ejecuta en distintos sistemas operativos (Windows, Linux, MacOS).
* Contiene Triggers.
* Contiene Assertions.
* Permite transacciones.
* Permite sacar respaldos.
* Alta integridad.
* Permite definir funciones den distintos lenguajes (PL/Java, PL/Python, etc).

## PostGIS

PostGIS es una extensión de datos espaciales para la base de datos PostgreSQL, esta, agrega soporte para muchos objetos geográficos, lo que permite hacer consultas con estos tipos de datos y que se ejecuten con SQL.



Figura 42. Consulta de datos geométricos con PostGIS.

Tomado de: (Alcaraz, 2016)

Características:

* Nuevos tipos de datos espaciales como geometría, ráster, etc.
* Funciones para seleccionar datos vectoriales y geográficos.
* Funciones SQL para proyección de la data espacial.

## PGAdmin

Es la herramienta oficial de PostgreSQL para administrar la base datos. Con una interfaz gráfica muy amigable con el usuario, nos permite hacer la mayoría de cosas que haríamos directamente desde la consola.

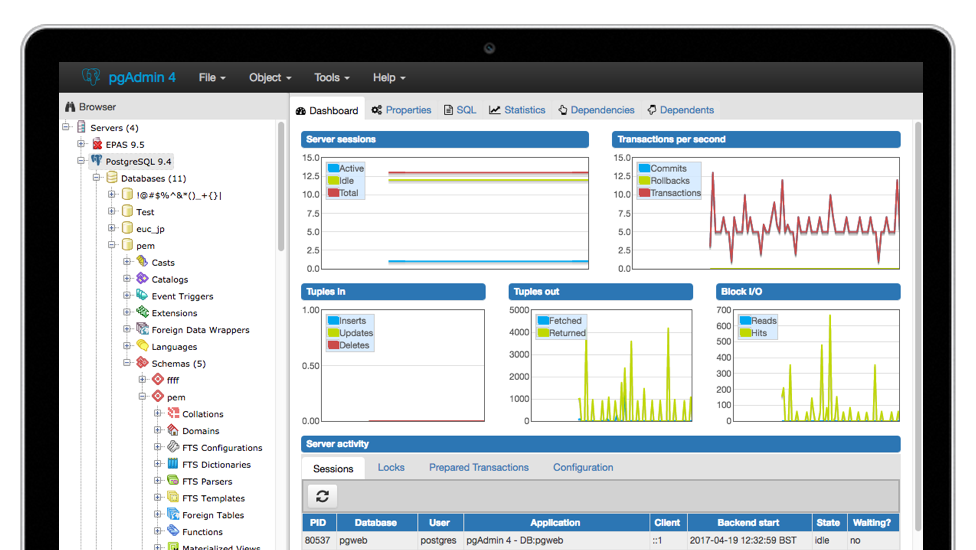


Figura 43. Dashboard de PgAdmin4.

Tomado de: (pgAdmin, s.f.)

Características:

* Multiplataforma.
* Funciona en múltiples versiones de PostgreSQL.
* Contiene una extensa documentación.
* Múltiples modos de implementación.
* Contiene distintas herramientas potentes.
* Contiene herramientas de mantenimiento de rutina.

## Control de versiones de código

## **Git**

“Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente. Su propósito es llevar registro de los cambios en archivos de computadora y coordinar el trabajo que varias personas realizan sobre archivos compartidos.” (Wikipedia, 2018)

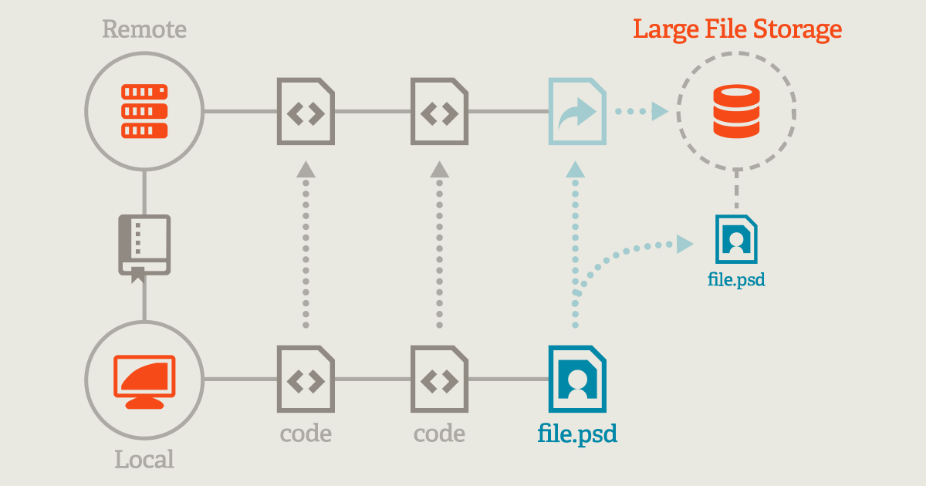


Figura 44. Proceso de Git.

Tomado de: (Github, s.f.)

Características:

* Gestión eficiente.
* Gestión distribuida.
* Rapidez en la administración de ramales.
* Re almacenamiento periódico.

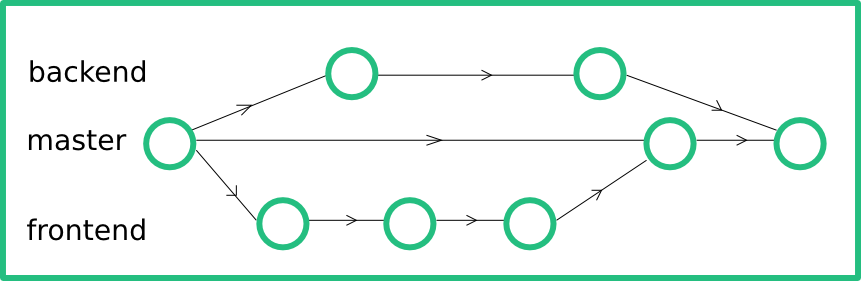


Figura 45. Ejemplo de ramificaciones.

Tomado de: (Ochoa, 2015)

## Arquitectura

## N Capas

La arquitectura de n-capas o de n-niveles, es una arquitectura en la cual se separa las funciones de presentación, integración, procesamiento, gestión de datos en diferentes capas físicamente.

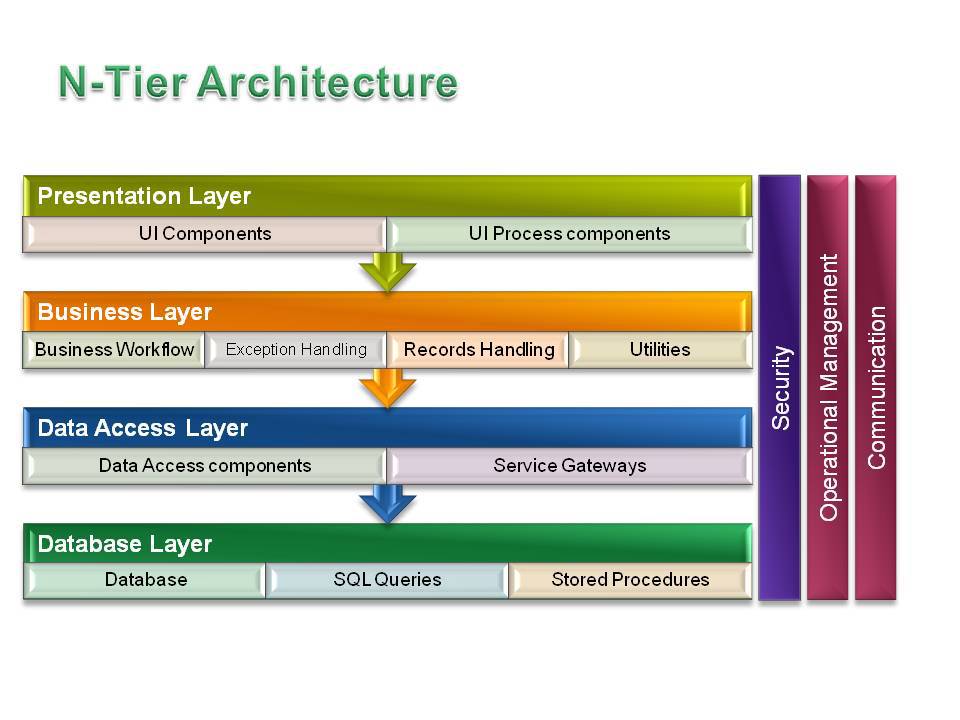


Figura 46. Diagrama de arquitectura N-Capas.

Tomado de: (GARHWAL, 2009)

Características:

* Se puede agregar o quitar capas dependiendo de la necesidad del proyecto.
* Cada una de las capas funcionan como cajas negras, por lo cual cada capa no sabe cómo procesa la información con la capa que está relacionada.
* Al modificar una capa, no tiene por qué afectarse el resto, dado a que son independientes, siempre y cuando se envié la información de la forma en la que fue diseñado.

## Patrones de diseño

## Singleton

Singleton que tiene su traducción en español como “instancia única”, es un patrón de diseño que tiene como objetivo restringir la creación de diversos objetos de una clase, lo que quiere decir que nos garantiza que una clase tenga solamente una instancia, a la cual proporciona un punto de acceso de forma global a través de un método o variable.

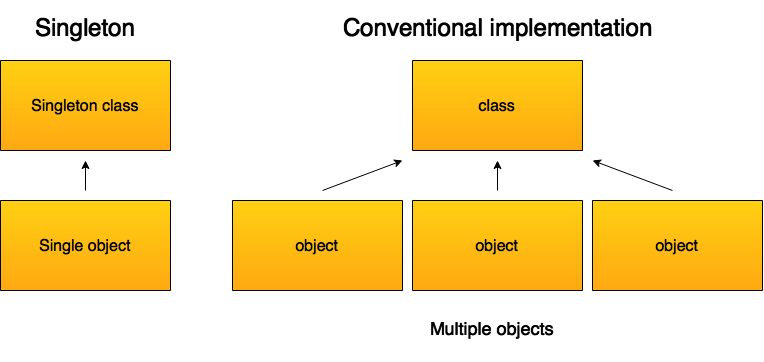


Figura 47. Patrón Singleton vs implementación convencional.

Tomado de: (Benharosh, 2015)

## Marco de Trabajo

## SCRUM

Scrum es uno de los marcos de trabajo más usadas en el mundo, está prácticamente compuesta de algunas características principales:

* Adopta una estrategia de desarrollo incremental.
* Se basa la calidad de los resultados en el conocimiento de las personas que se encuentran en equipos que ya son auto organizados,
* Se realizan cambios rápidos y frecuentes.

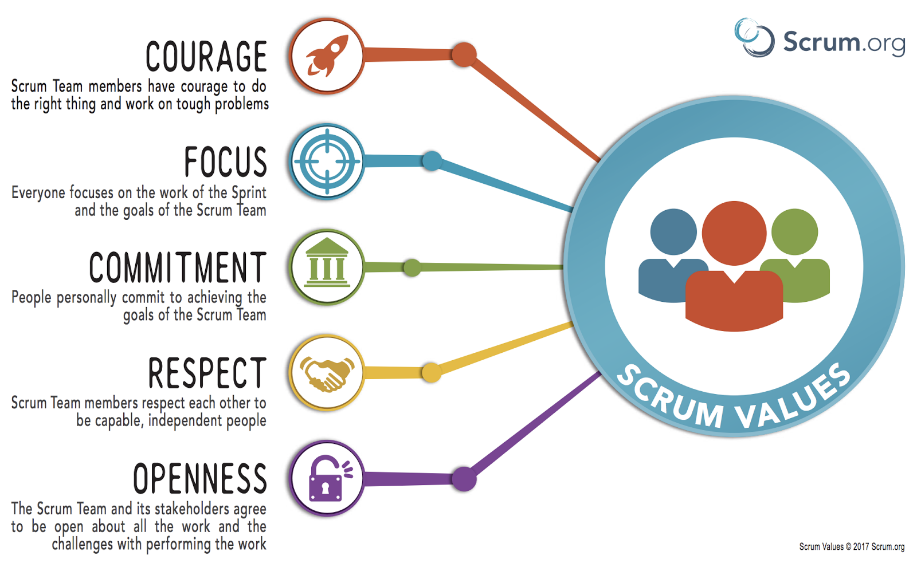


Figura 48. Valores de Scrum.

Tomado de: (Revueltas, 2017)

Roles:

* Personas comprometidas en el proyecto y en el proceso de SCRUM:
* Product Owner: Es la persona que tiene la visión final del producto final y la que más conoce sobre el negocio del cliente. Esta persona tiene el rol de tomar las decisiones y poner tomar las ideas del cliente y ponerlas en el Product Backlog.
* Scrum Master: Es el encargado de ver que todas las tareas se cumplan, y el encargado de resolver los problemas con el desarrollo.
* Equipo de desarrollo: Es el grupo de desarrollo.

Personas que dan la retroalimentación de salida del proceso, estas no están en el proceso de SCRUM:

* Usuarios: Son los destinatarios del producto.
* Stakeholders: Son personas que participan en las revisiones del sprint, y tienen algún beneficio con el proyecto.
* Managers: Son los que toman las decisiones a la hora de definir los objetivos del proyecto.

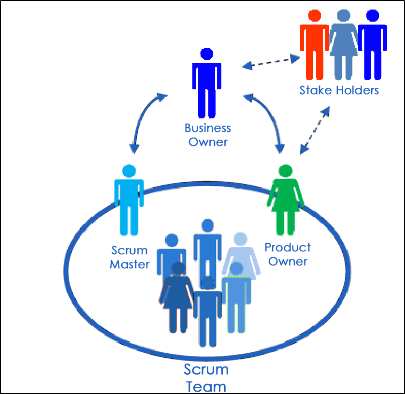


Figura 49. Roles en Scrum

Tomado de: (Gupta, s.f.)

Artefactos:

* Product Backlog: Es la lista de necesidades del cliente.
* Sprint Backlog: Es la lista de actividades o tareas que se van a realizar en un Sprint.
* Incremento: Es el resultado de cada Sprint, el mismo que debe ser operativo.

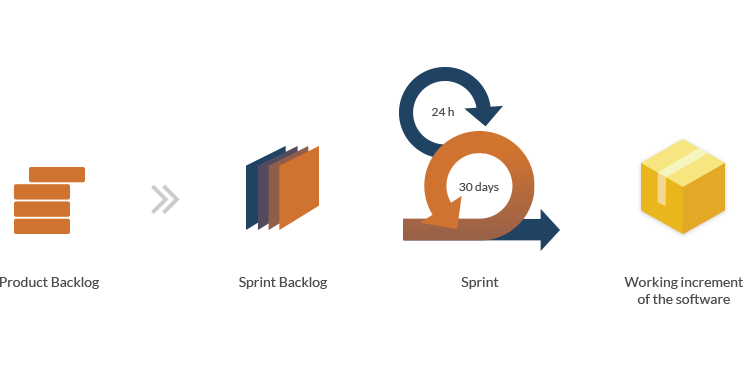


Figura 50. Proceso de artefactos Scrum

Tomado de: (softwarecamp, 2015)

Reuniones o Eventos:

* Sprint: Es cada una de las iteraciones del desarrollo.
* Planificación de sprint: Es la reunión previa al inicio del sprint, en esta se determina los objetivos y las tareas del sprint.
* Scrum diario: Es una reunión que se hace diariamente para ver los avances del día anterior y ver si se produjeron problemas.
* Revisión del sprint: Se revisa el trabajo del sprint, este puede analizar lo completado y lo no completado, en lo cual se muestra únicamente la parte completada a los interesados.
* Retrospectiva del sprint: Se revisa lo sucedido durante el sprint, y en este se crea planes de mejoras para los siguientes Sprints.

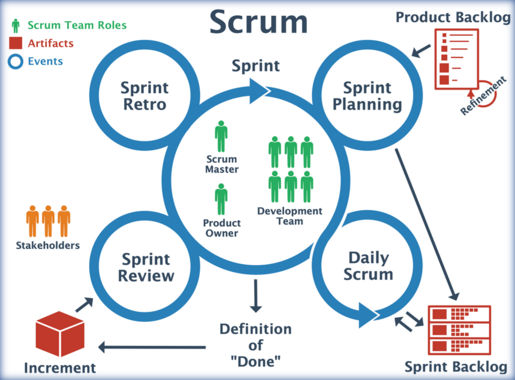


Figura 51. Ciclo de Scrum.

Tomado de: (Leading Agile, 2017)

Historias de Usuario:

Las historias de usuario son descripciones de funcionalidades que deben ser incorporadas en el sistema, las mismas que al ser implementadas otorgan un valor al usuario. Las historias de usuario están compuestas por algunos componentes, los mismos que son relativos dependiendo del autor, pero en general existen los siguientes:

* Card: Es una breve descripción de lo que se desea.
* Conversation: Es una conversación en la cual se define el objetivo y sirve para asegurar que se a entendido la funcionalidad.
* Confirmation: Pruebas funcionales para encontrar detalles relevantes del funcionamiento.
* Id: Número de identificación de la historia de usuario.
* Title: Es el título que contendrá la historia de usuario.
* Priority: Es la prioridad que tendrá la funcionabilidad.
* Description: Es una descripción sintetizada de la historia de usuario.

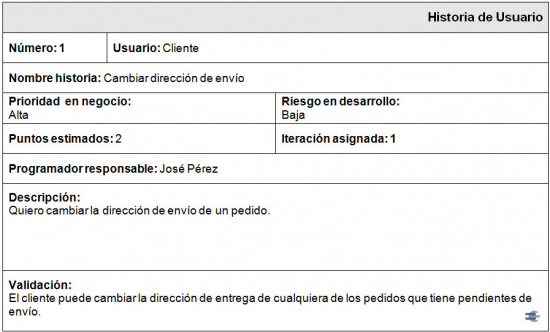


Figura 52. Ejemplo de historia de usuario.

Tomado de: (Scrum Manager Body Of Konwledge, 2014)

Para definir las historias de usuario se tiene que tener la sintaxis de:

* Como,
* Quiero,
* Para,

TaskBoard:

Es un tablero que sirve para poner las tareas del Sprint, en este se puede tener columnas de:

* Historia de Usuario.
* To Do: La lista de sub tareas por hacer.
* In Progress: Lista de sub tareas que se están realizando.
* Test o To Verify: Lista de pruebas o sub tareas por verificar.
* Done: La lista de sub tareas completadas.

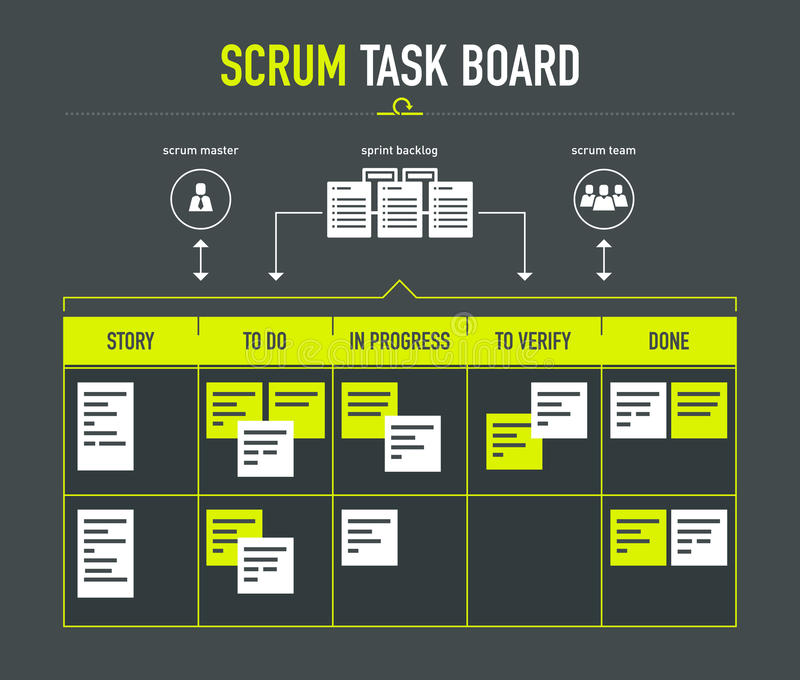


Figura 53. Ejemplo de tablero Scrum.

Tomado de: (Zabelava, s.f.)

## Conclusión de capítulo

## En este capítulo se ha podido definir la serie de herramientas que se han elegido para el desarrollo del módulo web, en la cual la mayoría de herramientas fueron seleccionadas debido a que tenían que ser compatibles con el sistema al cual se va a integrar, sin embargo, las herramientas que no tenían que ver con la integración fueron seleccionadas debido a sus grandes ventajas a diferencia de otras herramientas mencionadas.

# CAPÍTULO II. ANÁLISIS Y DISEÑO

En este capítulo se desarrollará el análisis y diseño de todo el proyecto, por lo que aquí, se recolectaran las historias de usuario, se realizaría el diagrama entidad relación, y se propondrá la arquitectura a seguir.

## Product Backlog

En esta sección se muestra la lista de requerimientos que han sido recolectados con los usuarios que interactuaran en el sistema, en la siguiente tabla se muestra cada historia de usuario, la misma que relata el usuario con su requerimiento.

Tabla 1

Product Backlog.

|  |  |  |  |
| --- | --- | --- | --- |
| Nº | Id | Pertenece | Historia de Usuario |
| 1 | MAP-001 | Mapa | Yo, **como** jefe de **producción**, **quiero** subir archivos con puntos georreferenciados de la plantación **para** poder visualizar en el mapa. |
| 2 | MAP-002 | Mapa | Yo, **como** jefe de producción, **quiero** dibujar en el mapa geometrías afines con distintas partes de la finca, **para** poder visualizarlas en el mapa. |
| 3 | MAP-003 | Mapa | Yo, **como** jefe de producción, **quiero** poder subir atributos a cada palma georreferenciada, **para** poder saber que palma es en el mapa. |
| 4 | MAP-004 | Mapa | Yo, **como** jefe de producción, **quiero** poder dar un tipo geometría referente al catálogo al ingresar un área, **para** poder graficarla en base a sus propiedades. |
| 5 | MAP-005 | Mapa | Yo, **como** jefe de producción, **quiero** poder eliminar las geometrías en el mapa, **para** así no tener las que no sirven, o están mal georreferenciadas. |
| 6 | MAP-006 | Mapa | Yo, **como** gerente general, **quiero** visualizar una simbología en el mapa, **para** poder entender que representa cada figura. |
| 7 | MAP-007 | Mapa | Yo, **como** gerente general, **quiero** ver el perímetro de la finca pintada en el mapa, **para** poder tener un panorama del mismo. |
| 8 | MAP-008 | Mapa | Yo, **como** gerente general, **quiero** 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. |
| 9 | MAP-009 | Mapa | Yo, **como** gerente general, **quiero** poder centrar el mapa en la finca, **para** no perder la vista si es que se mueve o se navega fuera del lugar. |
| 10 | MAP-010 | Mapa | Yo, **como** gerente general, **quiero** poder ocultar y mostrar los polígonos que quiera, **para** poder fijarme en lo que quiero ver. |
| 11 | MAP-011 | Mapa | Yo, **como** gerente general, **quiero** poder ver las palmas de finca con su respectiva información, **para** poder saber que palma es. |
| 12 | MAP-012 | Mapa | Yo, **como** gerente general, **quiero**, 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. |
| 13 | MAP-013 | Mapa | Yo, **como** gerente general, **quiero**, saber que geometría es la que está dibujada en el mapa, **para** diferenciar entre geometrías. |
| 14 | ATR-001 | Atributos | Yo, **como** jefe de producción, **quiero** definir una serie de atributos, **para** poder asignar información a cada lote parcelario. |
| 15 | ATR-002 | Atributos | Yo, **como** jefe de producción, **quiero** asignar atributos a cada lote parcelario, o a cada geometría, **para** así poder visualizar su información. |
| 16 | ATR-003 | Atributos | Yo, **como** jefe de producción, **quiero** poder editar la información de cada geometría, **para** actualizar si es necesario. |
| 17 | ATR-004 | Atributos | Yo, **como** jefe de producción, **quiero** eliminar atributos, **para** borrar aquellas que ya no se utilicen, o estén mal. |
| 18 | ATR-005 | Atributos | Yo, **como** jefe de producción, **quiero** modificar los atributos, **para** cambiar la información que haya sido mal ingresada. |
| 19 | ATR-006 | Atributos | Yo, **como** gerente general, **quiero** ver todos los atributos ingresados, **para** tener en cuenta los atributos que existen en las geometrías. |
| 20 | ATR-007 | Atributos | Yo, **como** gerente general, **quiero** ver por área, que atributos tienen, **para** poder saber la información correspondiente de cada área. |
| 21 | CAT-001 | Catálogo | Yo **como** jefe de producción, **quiero** definir un catálogo de geometrías, **para** así separar por tipo a cada una de ellas. |
| 22 | CAT-002 | Catálogo | Yo **como** jefe de producción, **quiero** asignar propiedades de color, grosor de línea, color de relleno a cada tipo del catálogo, **para** así poder ver en simbología. |
| 23 | CAT-003 | Catálogo | Yo **como** jefe de producción, **quiero** visualizar las propiedades de cada tipo en el catálogo, **para** ver cuáles han sido ingresados. |
| 24 | CAT-004 | Catálogo | Yo **como** jefe de producción, **quiero** eliminar los tipos del catálogo, **para** poder borrar los que tengan información incorrecta. |
| 25 | CAT-005 | Catálogo | Yo **como** jefe de producción, **quiero** modificar las propiedades del tipo en catálogo, **para** poder actualizar las propiedades en caso de que sean requeridas. |

Nota: El campo pertenece es una categoría de historia de usuario.

## Diseño de Arquitectura

## El diseño de la arquitectura es basado en base al software con el que se va a integrar, debido a que a la arquitectura tiene que ser compatible con el sistema heredado.

## Diagrama de Arquitectura

En base a los requerimientos que fueron recolectados, se decidió utilizar una arquitectura de 4 capas (n-capas), la misma que la capa de presentación será en Vue.js. El back-end será realizado con Python junto Django.

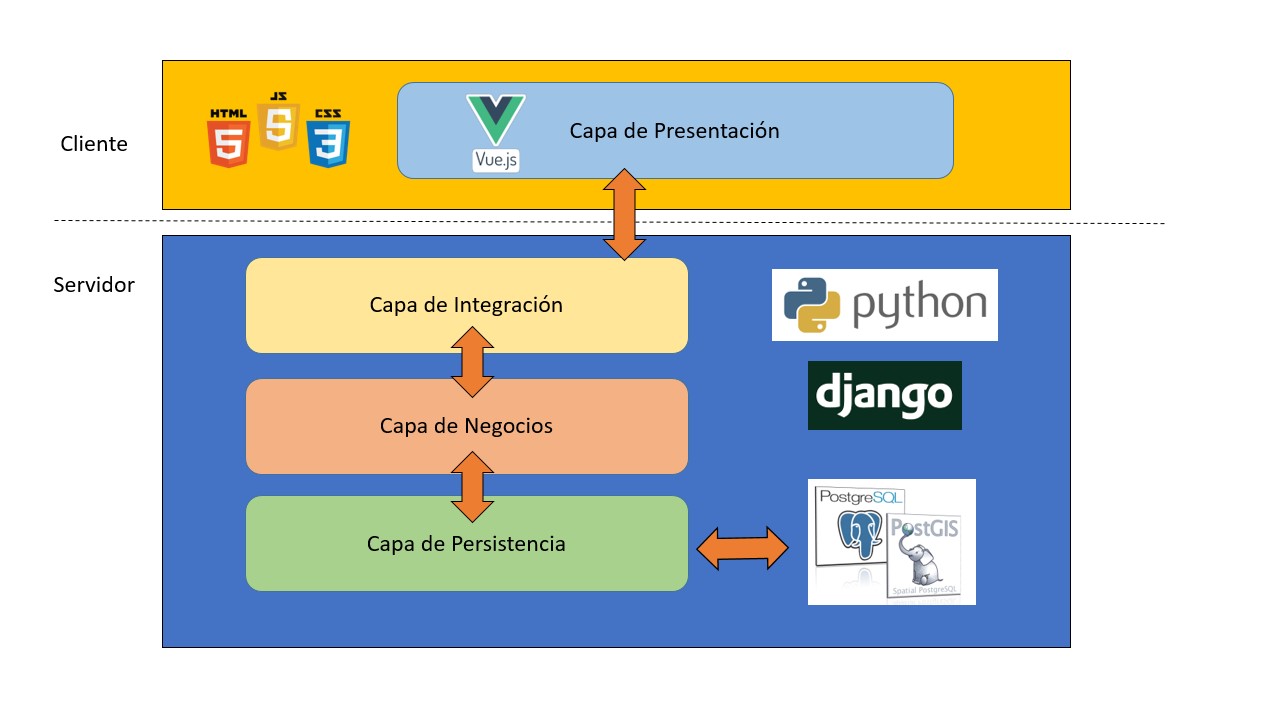


Figura 54. Diagrama de arquitectura del proyecto.

## Diagrama E-R de Base de Datos

De manera igual al diagrama de arquitectura, se diseñó el diagrama de base de datos, tomando en cuenta las 3 categorías de historias de usuario, las mismas que son, mapa (features o geometrías), catálogo y atributos.

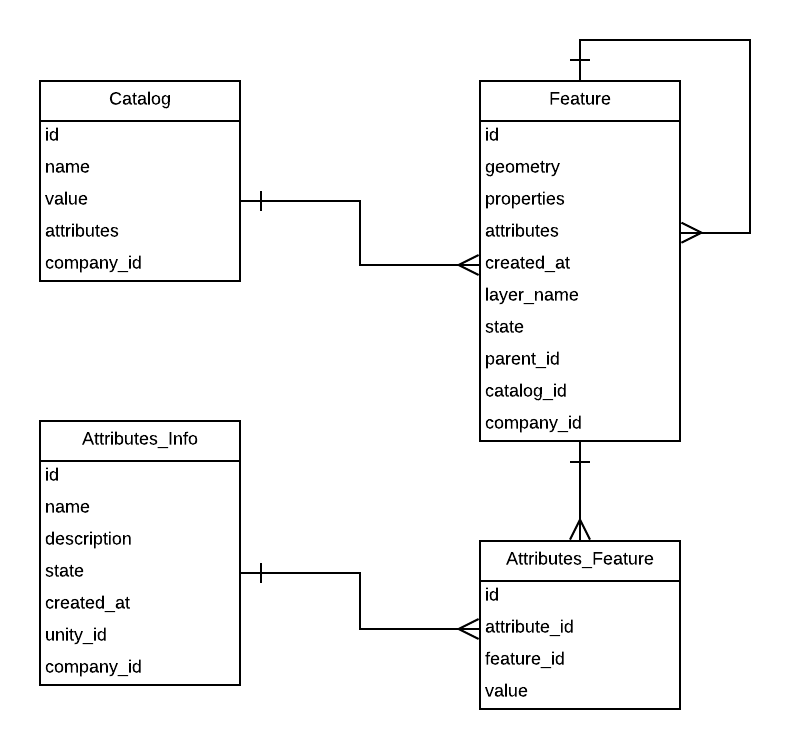


Figura 55. Diagrama de base de datos del proyecto.

## 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. | |

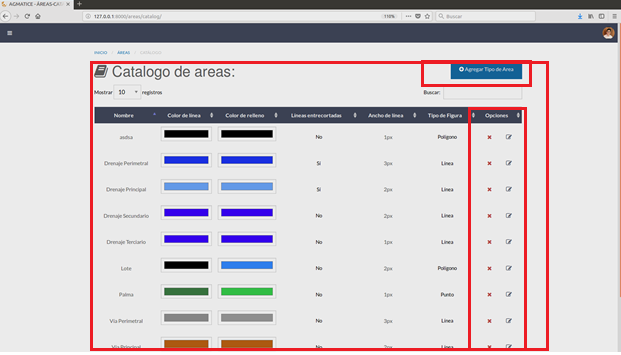


Figura 56. Menú de catálogo del proyecto.

## 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. | |

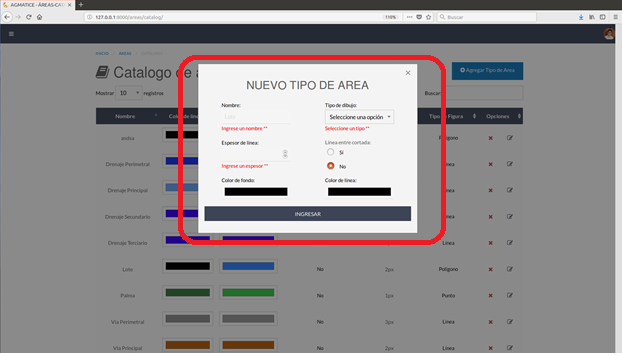


Figura 57. Ingreso de tipo de área, en catálogo.

## 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. | |

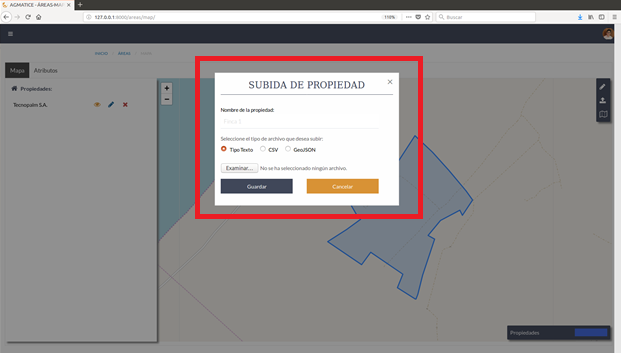


Figura 58. Subida de archivo para perímetro de la finca.

## 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 59. 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. | |

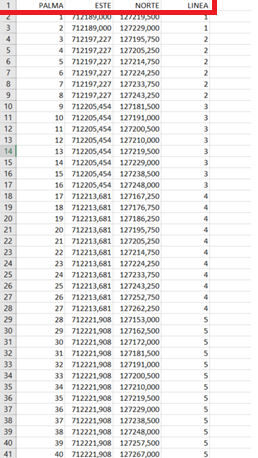


Figura 60. Archivo que contiene los atributos que son leídos por programa.

## 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. | |

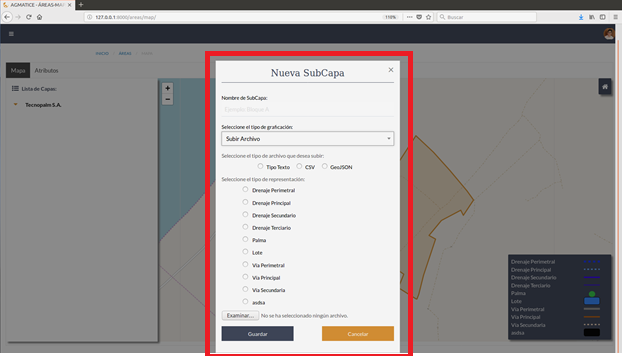


Figura 61. Ingreso de palmas, subiendo atributos.

## 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. | |

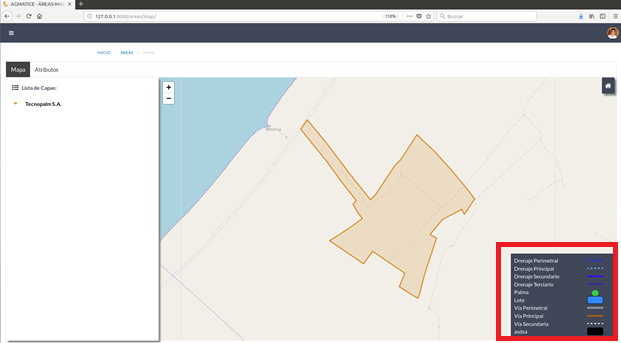


Figura 62. Mapa con su respectiva simbología.

## 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. | |

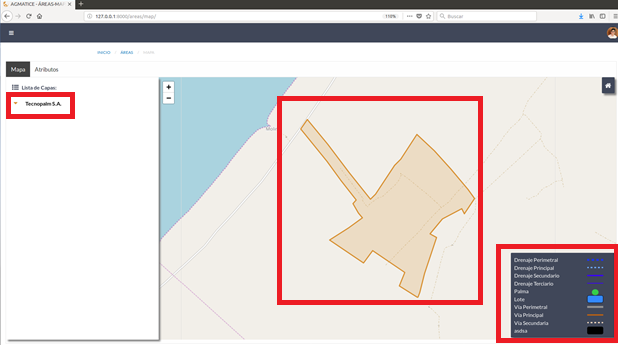


Figura 63. Mapa con el perímetro de la finca dibujada.

## 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 64. 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. | |

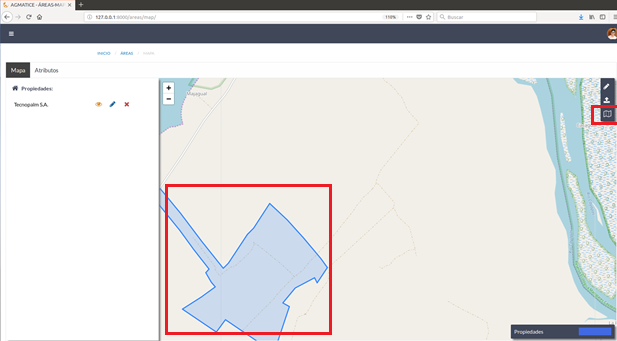


Figura 65. Mapa centrado, y botón de centrar mapa.

## 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. | |

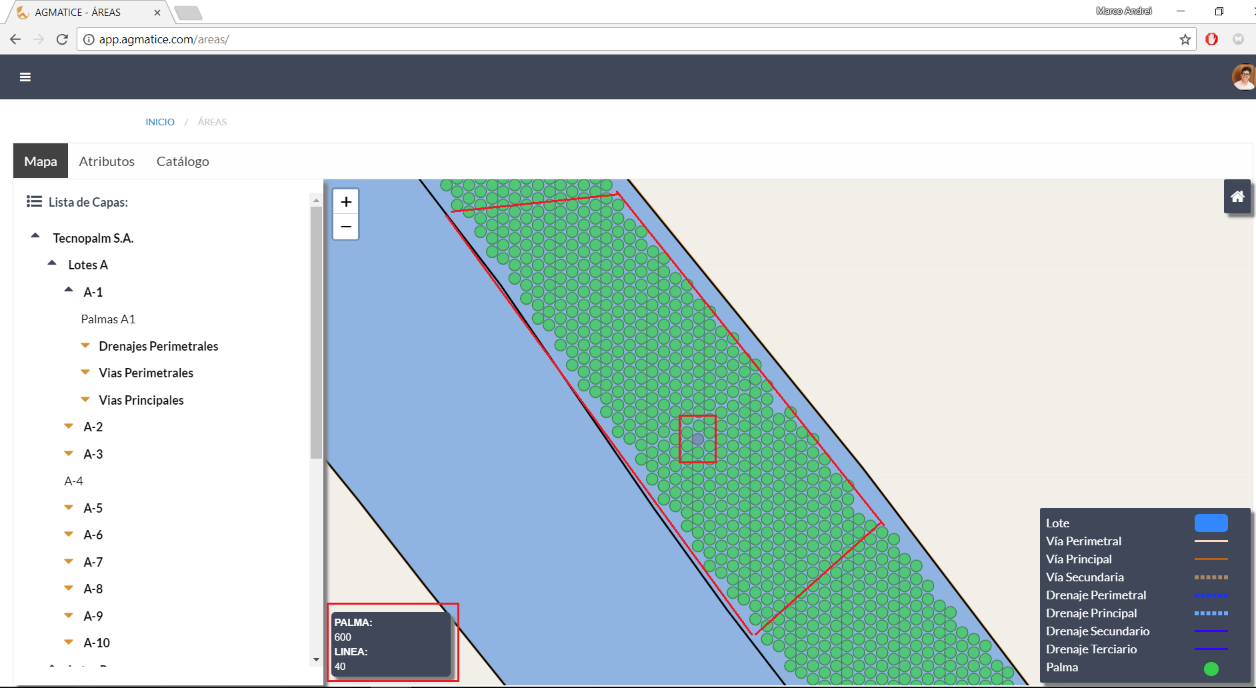


Figura 66. Mapa con palmas y su respectiva información.

## 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. | |

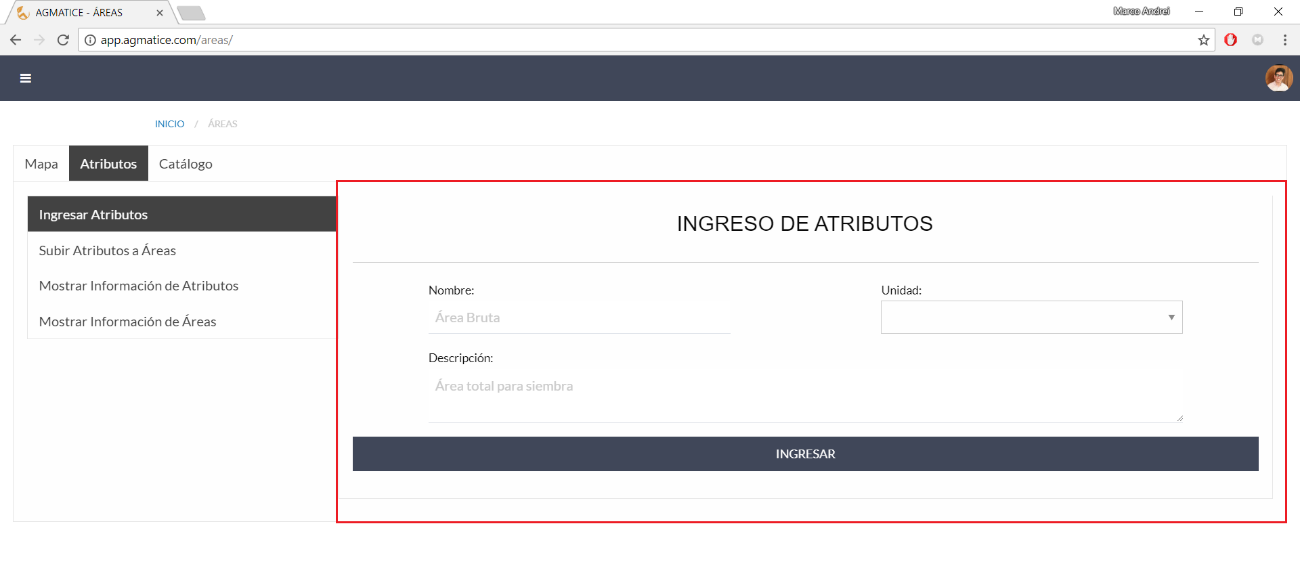


Figura 67. Formulario de ingreso de atributos.

## 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. | |

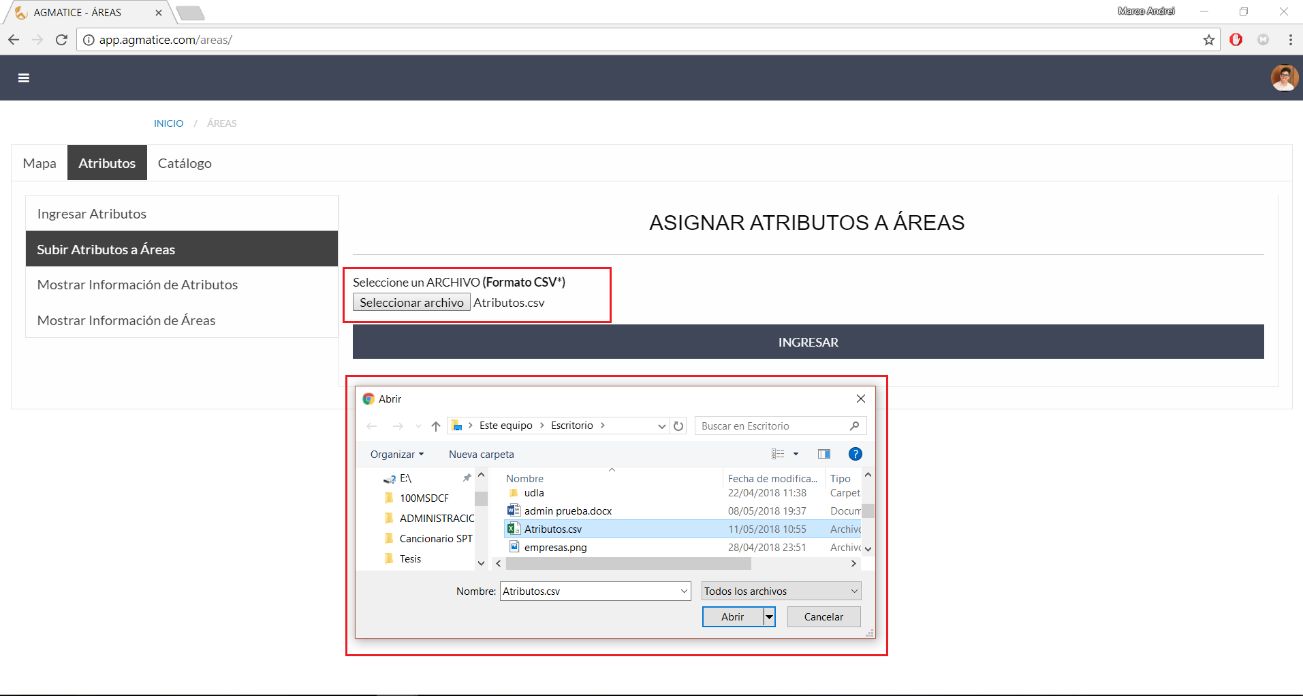


Figura 68. Subida de archivo para asignación de atributos.

## 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. | |

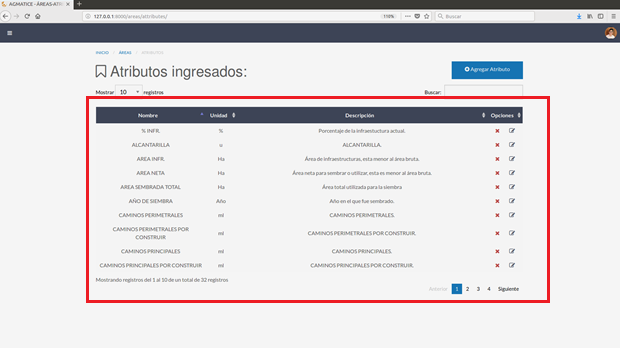


Figura 69. Tabla de atributos ingresados.

## 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. | |

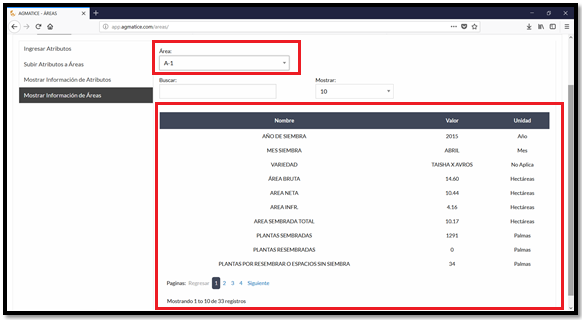


Figura 70.Tabla de atributos por área.

## 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. | |

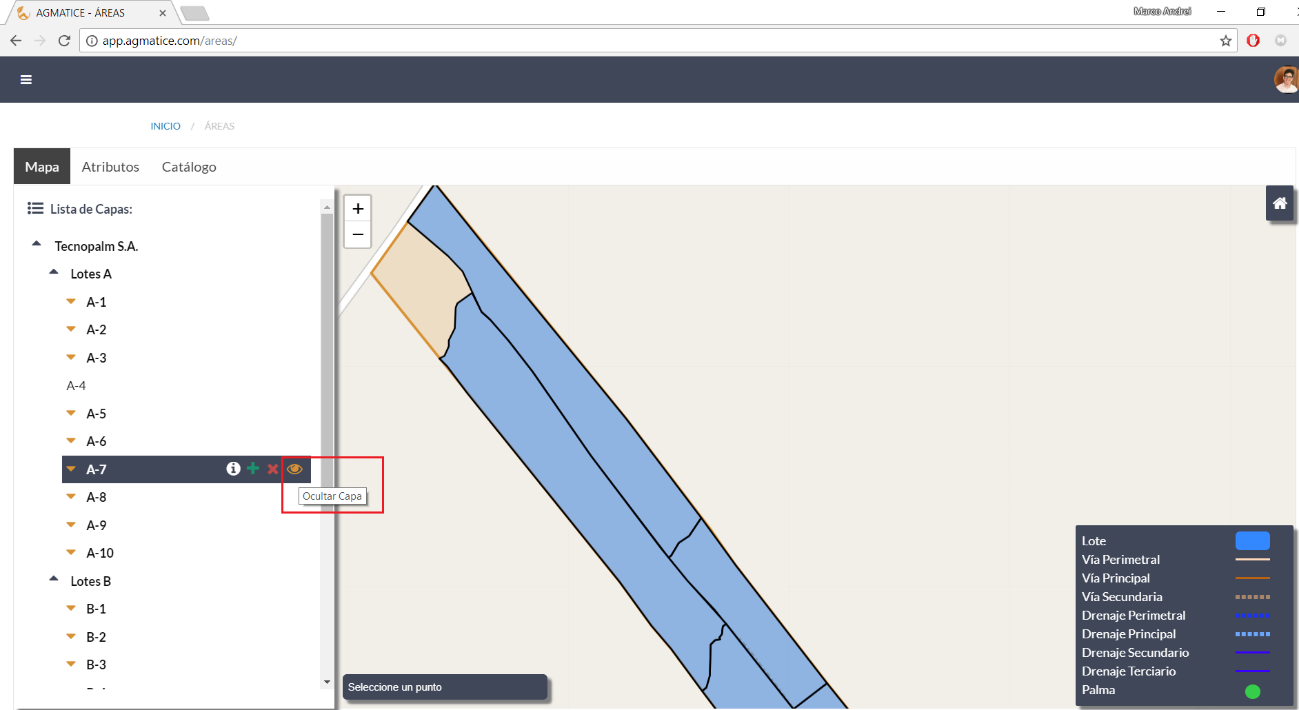


Figura 71. Botones de ocultar geometría.

## 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. | |

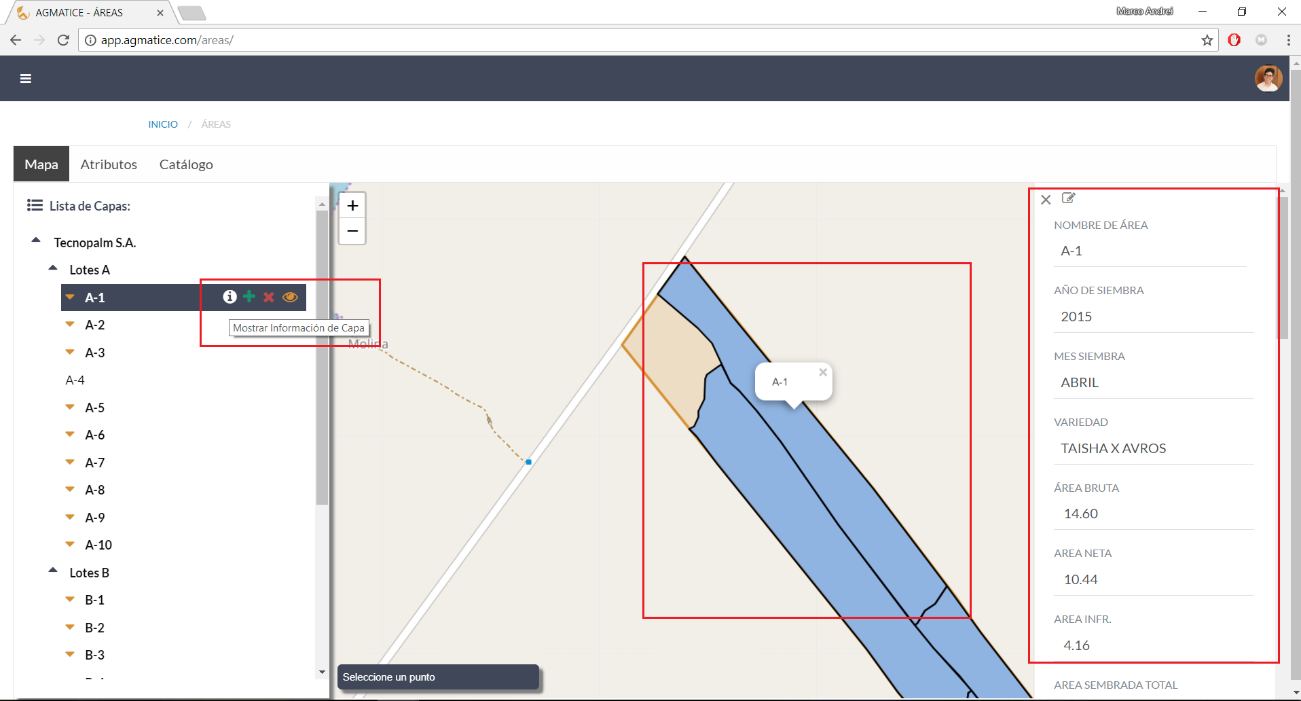


Figura 72. Atributos en el mapa.

## 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. | |

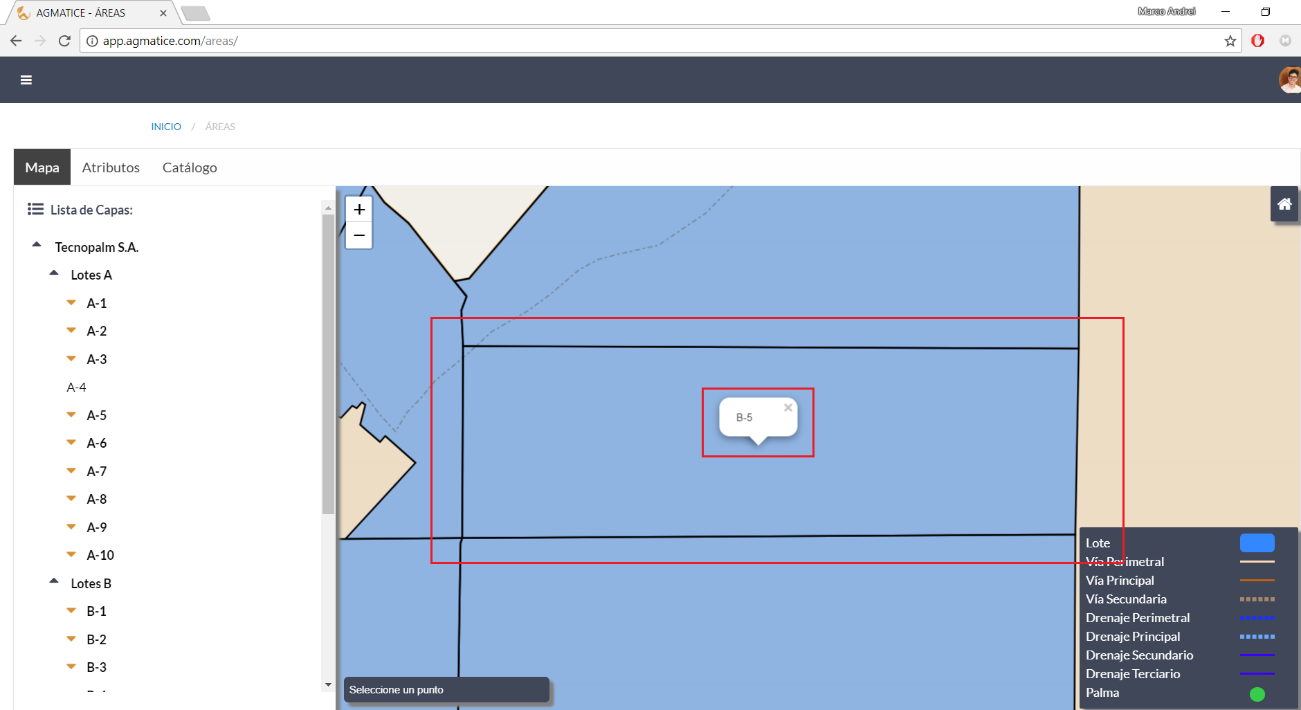


Figura 73. Atributos en el mapa.

## 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. | |

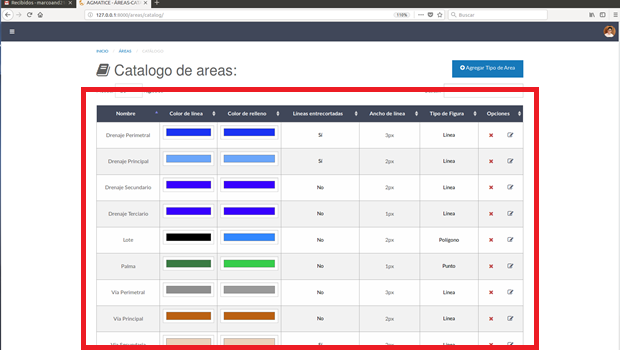


Figura 74. Lista de datos del catálogo.

## 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. | |

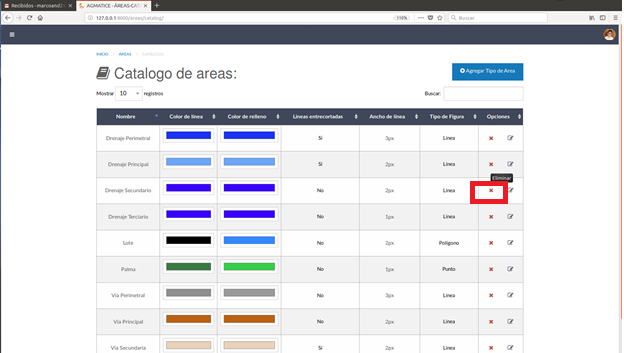


Figura 75. Botón de eliminar dato de catálogo.

## 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. | |

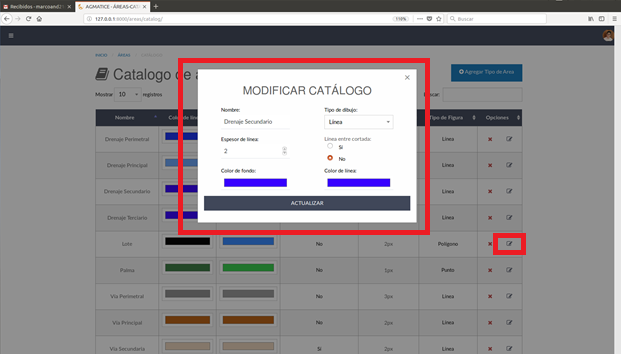


Figura 76. Ventana de actualización de dato de catálogo.

## 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 77. 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. | |

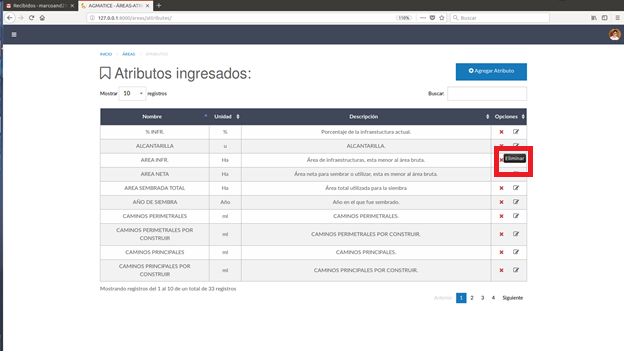


Figura 78. Botón de eliminación de atributo.

## 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 79. 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 80. 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

1. [↑](#footnote-ref-1)
2. [↑](#footnote-ref-2)
3. [↑](#footnote-ref-3)