

DESARROLLO DE APLICACIONES WEB PARA LA ALCALDÍA DE ITAGÜÍ

DANIEL HERNÁNDEZ OSORIO

UNIVERSIDAD CATÓLICA DE PEREIRA
FACULTAD DE CIENCIAS BÁSICAS E INGENIERÍAS
PROGRAMA DE INGENIERÍA DE SISTEMAS Y TELECOMUNICACIONES
PEREIRA
2021

DESARROLLO DE APLICACIONES WEB PARA LA ALCALDÍA DE ITAGÜÍ

DANIEL HERNÁNDEZ OSORIO

Informe de Práctica Profesional para optar por el título de Ingeniero en Sistemas y
Telecomunicaciones

Ing. Line Yasmin Becerra Sánchez, PhD
Directora de informe final y Monitora de práctica profesional

UNIVERSIDAD CATÓLICA DE PEREIRA
FACULTAD DE CIENCIAS BÁSICAS E INGENIERÍAS
PROGRAMA DE INGENIERÍA EN SISTEMAS Y TELECOMUNICACIONES
PEREIRA
2021

Nota de aceptación.

Firma del jurado

Firma del jurado

Firma del jurado

Pereira, 8 de junio de 2021

CONTENIDO

	Pág.
RESUMEN	3
ABSTRACT	4
INTRODUCCIÓN	5
1 DESCRIPCIÓN DEL ESCENARIO DE PRÁCTICA	7
2 DESCRIPCIÓN DE LAS ÁREAS DE INTERVENCIÓN	8
2.1 Secretaría de Hacienda	8
2.2 Secretaría General	9
3 JUSTIFICACIÓN	10
4 OBJETIVOS	12
4.1 Objetivo general	12
4.2 Objetivos específicos	12
5 MARCO TEÓRICO	13
5.1 Aplicación Web	13
5.2 Fronted	14
5.3 Backend	15
5.4 Tecnologías web	16
5.4.1 HTML.	16
5.4.2 CSS.	17
5.4.3 JavaScript.	17
5.5 Lenguaje Python	17

6 METODOLOGÍA	18
6.1 Actividades desarrolladas	18
6.2 FASES	18
6.3 Cronograma	19
7 DESARROLLO DE ACTIVIDADES DE PRÁCTICA PROFESIONAL	21
7.1 Contextualización	21
7.2 Descripción de metodologías y protocolos utilizados	21
7.3 Desarrollo Portal de pagos	22
7.4 Desarrollo Planoteca	29
8 RESULTADOS	35
8.1 Pruebas realizadas	35
8.1.1 Pruebas portal de pagos.	35
8.2 Pruebas planoteca.	36
8.3 Recomendaciones finales del proceso de práctica	37
9 CONCLUSIONES	38
10 RECOMENDACIONES	40
11 REFERENCIAS	41

LISTA DE TABLAS

	Pág.
Tabla 1. Cronograma de la intervención	25

LISTA DE FIGURAS

	Pág.
Figura 1. Organigrama de la DATIC [4].	7
Figura 2. Fases del Ciclo de Vida de software. VIEWNEXT [6].	8
Figura 3. Representación gráfica de una aplicación web [11]	14
Figura 4. Representación gráfica de un frontend [13].	15
Figura 5. Flujo web simple [15].	16
Figura 6. Modelo de la base de datos Portal Pagos	23
Figura 7. Diagrama de flujo Plataforma de pagos.	24
Figura 8. Diagrama de flujo proceso sonda.	25
Figura 9. Incidencias en portal pagos.	26
Figura 10. Commits de Portal de Pagos.	26
Figura 11. parte (a). Código Portal Pagos.	27
Figura 12. Modelo de base de datos Planoteca	29
Figura 13. Componentes Planoteca.	30
Figura 14. Incidencias Planoteca.	31
Figura 15. Commits Planoteca.	31
Figura 16. Aportes de código del proyecto Planoteca.	33
Figura 17. Pruebas Portal Pagos.	35
Figura 18. Pruebas Planoteca	36

RESUMEN

Los avances tecnológicos en los últimos años han sido grandes propulsores de la competitividad de los sectores productivos en el país, aportando a la productividad y eficiencia de los procesos, sin embargo desde los organismos públicos, este crecimiento se ha venido dando a un ritmo menos acelerado, existiendo en la actualidad una mayor necesidad por emprender una ruta hacia el mundo digital, en este sentido y específicamente la Alcaldía de Itagüí como ente público, busca alinearse a las necesidades actuales de los ciudadanos, implementando plataformas y aplicaciones web que mejoren la experiencia en la atención al usuario.

Por tanto, el presente informe de práctica profesional tiene como objetivo describir las diferentes actividades realizadas en el desarrollo de una aplicación web en la parte de backend para las diferentes dependencias de la alcaldía de Itagüí que permitan mejorar sus procesos y servicios, mediante el desarrollo backend basado en el lenguaje de programación Python haciendo uso de Django, Framework de Python de alto nivel que fomenta el desarrollo rápido y un diseño limpio. Además, implementando las reglas de API REST con el framework Django REST Framework, ideales para el Internet de las Cosas (IoT) y el desarrollo de aplicaciones para dispositivos móviles, en la parte de desarrollo frontend se usa el framework Quasar basado en Vue JS y como motor de base de datos se implementa PostgreSQL.

El desarrollo de este aplicativo web dio como resultado una plataforma interactiva que mejora la experiencia relacionada con el portal de pagos electrónico en zona de pagos de la alcaldía para el recaudo por concepto de impuestos, también se intervino la planoteca, portal en la que se realiza la administración de planos a cargo de la administración municipal y la consulta de planos arquitectónicos de curadurías, con la realización de esta intervención se concluye que el uso de software backend, es una herramienta versátil que permite el desarrollo de aplicativos web, los cuales en este caso se enfocan en el mejoramiento de la experiencia al usuario y la reducción en el tiempo de gestión por parte de éste, escenario que resalta la necesidad de hacer uso del software para aumentar la eficiencia de los procesos.

Palabras claves: aplicativo web, python, plataforma web, software.

ABSTRACT

Technological advances in recent years have been great promoters of the competitiveness of the productive sectors in the country, contributing to the productivity and efficiency of the processes, however, from public organizations, this growth has been taking place at a slower pace accelerated, there is currently a greater need to undertake a route to the digital world, in this sense and specifically the Mayor's Office of Itagüí as a public entity, seeks to align itself with the current needs of citizens, implementing platforms and web applications that improve the experience in user care.

Therefore, the objective of this professional practice report is to describe the different activities carried out in the development of a web application in the backend part for the different dependencies of the city hall of Itagüí that will allow improving their processes and services, through backend development based on the Python programming language using Django, a high-level Python framework that encourages rapid development and clean design. In addition, implementing the REST API rules with the Django REST Framework, ideal for the Internet of Things (IoT) and the development of applications for mobile devices, in the frontend development part the Quasar framework based on Vue JS is used, and PostgreSQL as database engine is implemented.

The development of this web application resulted in an interactive platform that improves the experience related to the electronic payment portal in the payment area of the mayor's office for the collection of taxes, when carrying out a series of commit, the plan library was also intervened, portal in which the administration of plans is carried out by the municipal administration and the consultation of architectural plans of curatorships, with the completion of this intervention it is concluded that the use of backend software is a versatile tool that allows the development of web applications, which in this case focus on improving user experience and reducing user management time, a scenario that highlights the need to make use of software to increase the efficiency of the Processes.

Keywords: web application, python, web platform, software.

INTRODUCCIÓN

En la actualidad, el sector público se ha venido enfrentado a grandes desafíos ante los acelerados cambios y desarrollos de carácter tecnológico y sociocultural de la actualidad, por lo que empresas de este tipo necesitan urgentemente ser remodeladas y modernizadas para dar respuesta a la creciente demanda de sus usuarios de soluciones transparentes, integrales y efectivas para un gobierno inteligente como pilar de un ecosistema de ciudad inteligente [1]. En este sentido el presente proyecto, aborda una problemática a nivel local desde la práctica profesional del ingeniero de sistemas y telecomunicaciones que es consistente con la necesidad de transformar la administración pública, mediante un desarrollo de la arquitectura empresarial a nivel web, visto como una solución alternativa a la necesidad de mejorar los procesos de seguimiento y control de los ingresos desde un enfoque estratégico en TI que se aplicará a la alcaldía de Itagüí con el fin de garantizar el mejoramiento del recaudo de la Secretaria de Hacienda y la gestión de planos arquitectónicos de los predios del municipio en la Secretaria General.

Sumado a lo anterior, este proyecto apunta a los lineamientos de construir una ciudad innovadora que utilice las tecnologías de la información y la comunicación (TIC) y otros medios para mejorar la toma de decisiones y la eficiencia de las operaciones, los servicios urbanos y su competitividad: “la innovación en la administración pública ha sido reconocida como un eje de nuevos esfuerzos para reactivar y fortalecer el papel de los gobiernos y administraciones públicas ante las complejidades y nuevos escenarios del siglo XXI [2]. Por lo tanto, se busca aportar a la mejora de los procesos enfocados en la atención a la ciudadanía y que a su vez generan un ingreso para la Alcaldía de Itagüí.

Como es de anotar, la implementación de una arquitectura empresarial se basa en el establecimiento de un conjunto de pautas arquitectónicas que aseguren un desarrollo armónico entre los modelos y necesidades de la empresa con los procesos de negocio y las tecnologías de la información [3], su aplicación se considera idónea para mejorar y optimizar los procesos de seguimiento y control del plan de desarrollo municipal actual y futuro propuesto por las distintas administraciones en el momento de la jurisdicción de la alcaldía, el municipio de Itagüí sumado, a la puesta en práctica y habilidades del futuro ingeniero de sistemas, el cual debe de estar capacitado para afrontar los retos que se presentan relacionados con el mejoramiento tecnológico del sector público.

Por ello, durante el desarrollo de este trabajo de práctica, se recogió la información administrativa, la experiencia y los procesos propios de la institución, que ya estaban definidos como insumos, para su posterior maquetación y estructuración con el fin de obtener un modelo confiable, objetivo y claramente apegado a la cadena de valor de la Alcaldía de Itagüí en línea con sus funciones administrativas de gestión y seguimiento del plan de desarrollo local, las dependencias intervenidas

siempre mantienen las condiciones de interoperabilidad y alineación con un enfoque tecnológico.

Por ello, este documento ha sido estructurado con el fin de darle repuesta a las necesidades latentes, tomando como base la misión de las secretarías desde un nivel empresarial, además, del reconocimiento de las estrategias y actividades de negocio que intervienen en la gestión. Esta actividad se apoya en la articulación del software backend con el frontend para su posterior desarrollo, análisis del grado de madurez de las capacidades y adecuación a la estrategia del gobierno digital, dando como resultado un aplicativo web que permita mejorar los procesos de recaudo de impuestos y gestión de planos, a la vez cuenta con una arquitectura moderna y un nivel de seguridad propio de un aplicativo del estado.

El desarrollo del presente informe inicia en el capítulo I con la definición del escenario de práctica punto en el que se indica la ubicación geográfica y empresa en la que se lleva a cabo la práctica profesional, seguido de esto en el capítulo II, se definen el área de intervención en las que se ejecuta la práctica, la cual es el desarrollo backend y web, en el capítulo III, se presenta la justificación de la realización de la práctica en estas dependencias, resaltando la necesidad de realizar una intervención en estas, en el capítulo IV se definen los objetivos de la práctica que le dan un norte a la realización de la presente práctica profesional, seguido de esto se realiza en el capítulo V el marco teórico, el cual aborda teorías relacionadas con aplicaciones web, backend, frontend, tecnologías web y lenguaje de programación Python que buscan sustentar la intervención realizada, en el capítulo VI, se estipula la metodología seleccionada para la realización de las diferentes aplicaciones web en las dependencias de la alcaldía de Itagüí, en el capítulo VII se relaciona la descripción del desarrollo de las actividades en la empresa y la intervención realizada, presentando en el capítulo VIII los resultados de los aplicativos web que se generaron, pasando al capítulo IX con conclusiones relacionadas con los resultados y el desarrollo general de la práctica.

1. DESCRIPCIÓN DEL ESCENARIO DE PRÁCTICA

La práctica se desarrolla en la empresa Ponce Asesores y Consultores S.A.S. constituida en el municipio de Itagüí Antioquia con dirección calle 75 AB SUR # 52-336. Entre su portafolio de servicios cuenta con desarrolladores de software para realizar desarrollos a la medida, asesores en gobierno en línea principalmente para entidades públicas con el fin de garantizar que los desarrollos realizados cumplan con los lineamientos y normatividad de gobierno en línea.

Según lo anterior, la alcaldía del municipio de Itagüí es una de las entidades donde se prestan los servicios por lo tanto la práctica se desarrolla en las instalaciones de la alcaldía mencionada en la Dirección Administrativa de las Tecnologías y Sistemas de la Información y las Comunicaciones (DATIC) formando parte del equipo de desarrollo del municipio. A continuación, se presenta un diagrama donde se describe la distribución jerárquica en la dependencia, En la figura 1 se muestra el organigrama de la dirección administrativa de las TIC's.

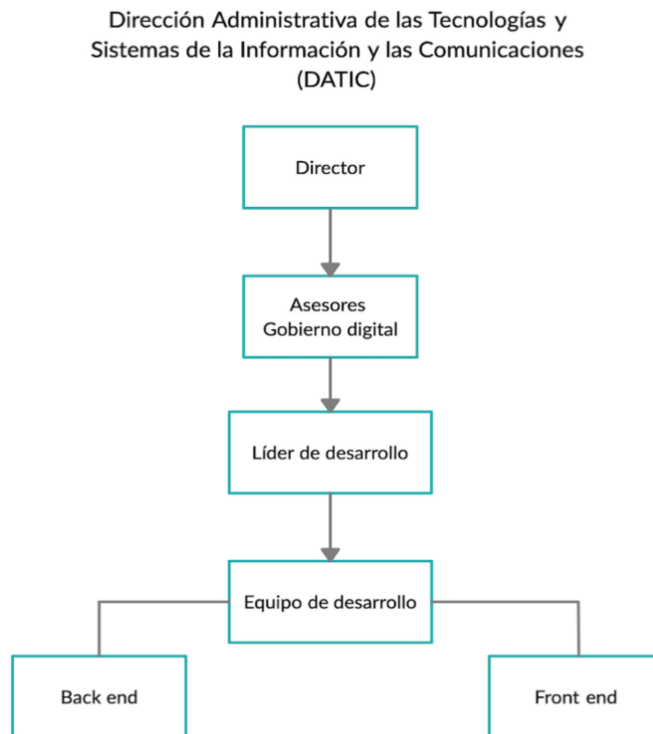


Figura 1. Organigrama de la DATIC [4].

2. DESCRIPCIÓN DE LAS ÁREAS DE INTERVENCIÓN

El equipo de desarrollo se encarga tanto de desarrollar aplicaciones para solventar las necesidades de las demás secretarías y dependencias de la alcaldía, como facilitar trámites para los ciudadanos, llevar procesos físicos a medios virtuales entre otros. El levantamiento de los requerimientos para el desarrollo de las aplicaciones se realiza en reuniones con los representantes del área que requiere el desarrollo con presencia de los asesores de gobierno en línea y el líder de desarrollo y cuando sea necesario también tendrá presencia de algunos desarrolladores. Luego se realizan reuniones internas en la dependencia (DATIC) donde se definen prioridades entre las aplicaciones a desarrollar, se desarrolla un cronograma y se crean y se asignan las tareas en la plataforma para la administración de tareas y gestión de proyectos de GitLab donde se realiza seguimiento al avance en las tareas.

2.1. SECRETARÍA DE HACIENDA

La secretaría de hacienda de la alcaldía de Itagüí actualmente tiene una necesidad ya que cuentan con una plataforma muy inestable para el recaudo de impuestos predial, industria y comercio, el pago de estos impuestos representa la mayor fuente de ingresos para el municipio, los problemas que tiene la plataforma generan un soporte constante y ocasiona que los contribuyentes no puedan realizar los pagos oportunamente afectando la recaudación para el municipio. Debido a lo anteriormente mencionado, la secretaría de hacienda ha manifestado su necesidad y ha solicitado que se desarrolle una nueva plataforma de recaudo que garantice estabilidad, que reduzca considerablemente el soporte y que permita un pago oportuno por parte de los contribuyentes. En la siguiente figura 2 se puede ver las fases del Ciclo de Vida del desarrollo de programas [5].



Figura 2. Fases del Ciclo de Vida de software. VIEWNEXT [6].

Con lo anterior presente, se propone intervenir en las fases de diseño, desarrollo y pruebas y puesta en marcha de la siguiente manera:

- Revisión de documentación de la pasarela de pagos Zona Pagos - plataforma de pagos externa - con el fin de facilitar la integración con el Portal de pagos planteado para la alcaldía.
- Realización de diagramas de flujo, para facilitar la comprensión.
- Integración backend con la pasarela Zona Pagos.

2.2. SECRETARÍA GENERAL

La secretaría general de la alcaldía de Itagüí tiene una necesidad latente relacionada con la forma en que realizan la gestión de planos arquitectónicos de los predios del municipio, actualmente toda la información sobre los planos la almacenan de manera física y los trámites sobre solicitudes que se obtienen sobre los planos se registran mediante el sistema Microsoft Access y en otras ocasiones en Excel, por todo esto es complicado que se puedan obtener indicadores de manera oportuna sobre los planos más solicitados etc. Además, de que se trabaje sobre los planos almacenados físicamente genera que toda la gestión sobre los planos sea manual muy limitada y propensa a errores.

Por las razones expuestas anteriormente la secretaría general ha solicitado el desarrollo de una aplicación web que les permita realizar la gestión de planos de manera digital, donde se puedan registrar los planos, las solicitudes a los planos y con la información almacenada pueda generar gráficos y reportes fácilmente. Para la puesta en marcha de este software fue indispensable la articulación del backend con el frontend, por tal motivo se propusieron las siguientes intervenciones:

- Desarrollo de la primera versión del software (backend).
- Apoyo al frontend en la realización de pruebas e integración.

3. JUSTIFICACIÓN

Gracias a los adelantos tecnológicos y uso masivo de la información por medio de internet, se ve la necesidad de llevar a cabo una aplicación web que posibilite a la ciudadanía el uso de los servicios brindados por la secretaria de hacienda y la secretaria general, relacionados con el recaudo de impuestos de manera virtual y el otro referente a la consulta de planos, para los diferentes establecimientos que se hallan en la jurisdicción del municipio de Itagüí. Sumado a lo anterior, la alcaldía posee una alta demanda de servicios por parte de la población la cual va en aumento y que requiere de procesos de transformación en su modelo de atención para asegurar un mejor grado de satisfacción al ciudadano que garantice su derecho y que complete sus expectativas para mejorar su percepción de la administración gubernamental a nivel local.

Cabe destacar que el motivante de este proyecto es la demostrada imposibilidad de brindar atención ciudadana óptima y de calidad a los servicios públicos municipales que se proponen las secretarías de la Alcaldía de Itagüí por la centralización de los mismos en este diminuto espacio; por consiguiente, desarrollar centros virtuales para la atención de los servicios municipales en este recinto es fundamental pues se resuelve una problemática que tendría año con año un aumento exponencial sino se atendía oportunamente.

En la actualidad es de saltar que en Colombia está en un proceso de revolución tecnológica y administrativa en sus diferentes entes del sector público, prueba de eso, tácticas como Régimen online (GEL) adoptada por medio del decreto 2573 de 2014 [7] en el que se establecen sus lineamientos primordiales proyectándose en 4 ejes temáticos (Servicios, Régimen abierto, TIC para la Administración, Estabilidad y Privacidad) y la Ley de Transparencia Publica e Ingreso a la Información [8], siendo estas varias de las pruebas de diversas iniciativas y políticas públicas adoptadas por el Régimen De Colombia que realizan la modernización y la mejora de las TIC al interior de las instituciones públicas nacionales y territoriales como es la situación de la Alcaldía de Itagüí, una necesidad de las novedosas administraciones y una posibilidad para interiorizar cambios de procesos y proyecciones a futuro apoyadas en herramientas de tecnologías de la información.

Sumado a lo anterior, el uso de aplicaciones web posibilita un conveniente uso y procedimiento de la información que a las diferentes dependencias de la Alcaldía llegan, entre los cuales se hallan los pagos de predial e impuesto a los ciudadanos, con su respectivo formato y certificados. Con la utilización de este sistema se pretende eludir el funcionamiento inadecuado de la información, la duplicidad de los datos, el cobro indebido de los impuestos que corresponden, las demoras en los tramites; además, de considerar puntos de vista relevantes como procesos bajo términos de habeas data, el funcionamiento de la información centralizada, lo que posibilita un acceso apropiado a los datos necesarios y evitando perderlos, y

además, proyectándose hacia servicios de régimen online, realizando uso de la reglamentación y los términos legales que corresponden y teniendo presente tecnología de código abierto para el desarrollo del aplicativo.

4. OBJETIVOS

4.1. OBJETIVO GENERAL

Desarrollar una aplicación web en la parte de backend para las diferentes dependencias de la alcaldía de Itagüí que permitan mejorar sus procesos y servicios.

4.2. OBJETIVOS ESPECÍFICOS

- Implementar el módulo de seguridad al iniciar el desarrollo de cada proyecto.
- Desarrollar la estructura para procesos en segundo plano según las necesidades de cada proyecto.
- Realizar las tareas definidas para cada proyecto según los requerimientos obtenidos y aplicando la lógica de negocio en cada uno de los proyectos.
- Codificar los módulos y funcionalidades según los estándares definidos para cada proyecto.
- Documentar el código aportado al proyecto siguiendo la especificación PEP 258 para el lenguaje de programación Python.

5. MARCO TEÓRICO

En el presente apartado, se definen los conceptos claves que conforman la base para el desarrollo de la práctica, soportando de esta forma la solución de los objetivos planteados en el informe, con el fin de entender cada uno de estos y tener mayor claridad sobre el tema tratado.

5.1. APLICACIÓN WEB

Una aplicación web puede describirse como un tipo de programa que se codifica en un lenguaje de programación que logre ser ejecutado por los navegadores de internet o por una red local o una intranet, estas se ejecutan mediante un navegador web y no requieren ser instaladas en el dispositivo del cliente, debido a que los datos o archivos usados permanecen almacenados en una red o en la nube

Las aplicaciones de esta índole tienen la posibilidad de ser de diferentes tipos, dependiendo de las necesidades de sus creadores y dueños, Hay las aplicaciones para administración interna, que se enfocan en los procesos más relevantes de la gestión de una organización como la facturación, inventarios, consumidores, usuarios, aliados, contabilidad, recursos humanos. Otras son herramientas de trabajo, como intranets, gestores de documentos, servicios compartidos, herramientas de comunicación como mails, boletines, comunicaciones personalizadas, chats, etc. Además, hay tiendas virtuales, gestores de ventas en línea, repositorios y buscadores, sitios turísticos, mapas, escuelas, foros y muchas más [9].

Esta clase de aplicaciones resultan muy convenientes ya que los consumidores o usuarios no necesitan instalarlas en sus dispositivos para entrar a ellas y usarlas, solo es preciso de un navegador web, por lo cual tienen la posibilidad de ser usadas en distintas plataformas y dispositivos, y que la aplicación se aloje y use en un servidor permite su mantenimiento y optimización. Comúnmente permanecen estructuradas en 3 capas interconectadas donde el navegador web da la primera capa, interpretando el código; el servidor que carga este código y toda la información conforma la segunda capa y una base de datos constituye la tercera y última capa. El navegador web envía pedidos a la capa intermedia, la cual da servicios usando consultas y actualizaciones a la base de datos a la vez que da una interfaz gráfica para la relación del cliente [10].

Para el desarrollo de las capas previamente mencionadas se tornan relevantes los conceptos de Frontend y Backend, donde el primero conforma lo cual puede ver el comprador, con lo cual puede interactuar, como la interfaz de cliente y el segundo tiene relación con eso que está ajeno a su manipulación, que está del lado del

servidor, como los archivos, programa de procesamiento y la permanencia, provista por la base de datos, la figura 3, describe gráficamente el frontend de una aplicación web.



Figura 3. Representación gráfica de una aplicación web [11]

5.2. FRONTEND

El Frontend es la porción de una aplicación con la que los usuarios tienen la posibilidad de interactuar, es famosa además, como el lado del comprador y es fundamentalmente todo lo que se ve en la pantalla una vez que se accede a un website o una aplicación, ciertos recursos que le componen son los tipos de letra, colores, adaptación para diversas pantallas (diseño web responsivo), los efectos del ratón, teclado, movimientos, desplazamientos, efectos visuales y muchos más que permiten navegar en una página web y desde dichos recursos se crea la vivencia del cliente [12], en la figura 4, se evidencia gráficamente una aplicación web en uso.

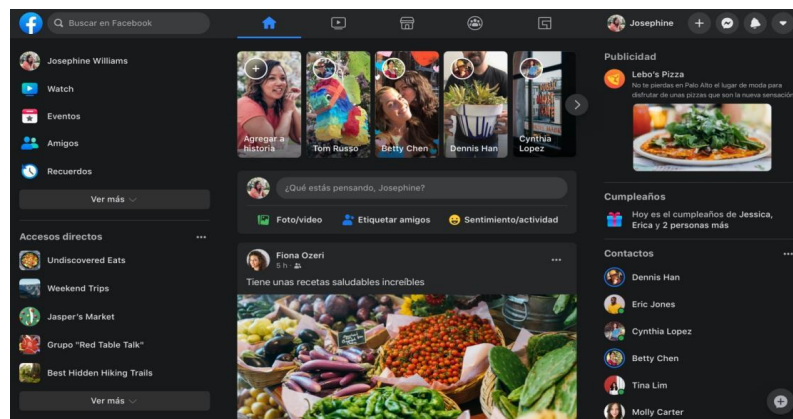


Figura 4. Representación gráfica de un frontend [13].

Se apoya en la porción de una página web que interactúa con los usuarios que la visitan. Es decir, es la parte visible, la que muestra los contenidos, el diseño y los instrumentos que le permiten al cliente navegar por la página web como lo deseen. Además, podría ser determinado como la parte que tiene todo el diseño web, contrario, paralelamente, al backend, que es el elemento de la web referido a los datos y al programa para su comunicación. El frontend es de gran relevancia en tanto representa una identidad corporativa de una organización en la Internet, algo que no es sencillo si se estima que el frontend de una página web de una compañía puede atraer o repeler consumidores. Además, este componente frecuenta agrupar 3 idiomas diferentes de programación: el HTML, el CSS y el JavaScript, descritos posteriormente [13].

Estos tres lenguajes de programación acostumbran a ser los más usados para la construcción y caracterización del frontend de una página web. El lenguaje HTML en concreto, siglas que corresponden a HyperText Markup Language, es el lenguaje más utilizado para llevar a cabo páginas web. Ahora, el HTML no es realmente un lenguaje que se encuentre creado para planificar, sino que usa etiquetas para ordenar y dar cuerpo al sinfín de portales de la web. Sin embargo, el lenguaje nombrado CSS se utiliza para conceptualizar un estilo y un aspecto a documentos escritos en lenguajes como el HTML [13].

Lo anterior supone que el lenguaje CSS dota al archivo de determinados colores, tipo y tamaño de fuentes e inclusive establece el espaciado entre letras y párrafos de dicho archivo. Al final, el lenguaje JavaScript es un lenguaje de programación que posibilita realizar ocupaciones tanto primordiales como complicadas en páginas web. Es un sistema que disfruta de la peculiaridad de no necesitar compilación alguna, debido a que son los propios navegadores a los que se delegan de leer el código para asimilarlo y realizar las actividades que este sugiere. Actualmente es el estándar una vez que se desean proponer mecanismos más complejos de lo común en una web [14].

5.3. BACKEND

Por su lado, el backend hace referencia al área lógica de toda página web, que no es visible a los ojos del cliente, a la arquitectura interna del lugar que garantiza que todos los recursos desarrollen la funcionalidad idónea. Está constituida por un servidor, una aplicación y una base de datos. Esta área es la encargada, además, de la funcionalidad del lugar, de la estabilidad y la mejora de los recursos. En el backend se hacen todos los procesos que posibiliten el desarrollo del sistema, como las actividades lógicas, las conexiones con bases de datos y la mejora de recursos y funcionamiento de archivos. En este se utilizan además, librerías distintas para

obtener funciones concretas como la compresión de imágenes o su pre visualización, incrementar la eficiencia de procesos, entre otras [15]. En la figura 5, se describe el flujo web conformado por el frontend y el backend.

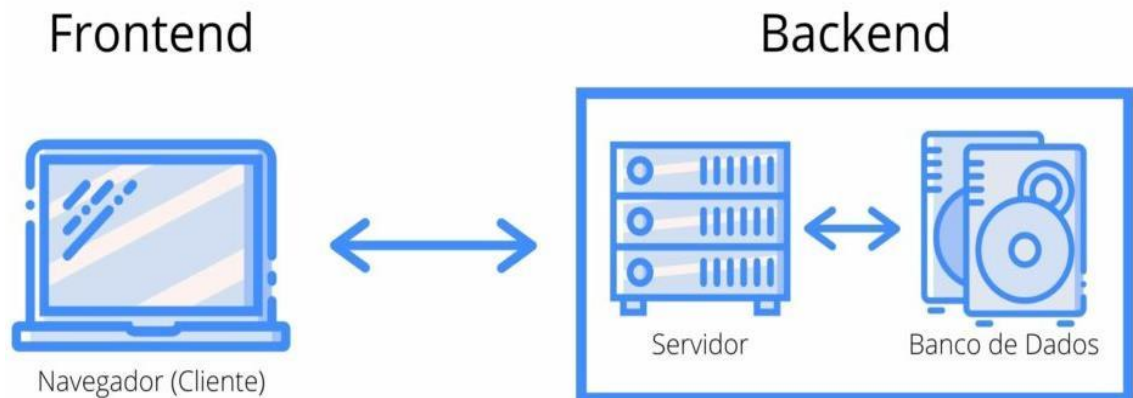


Figura 5. Flujo web simple [15].

En este sentido, la capa de ingreso a datos backend es la parte del desarrollo web que se ocupa de que toda la lógica de una página web funcione. Se trata del grupo de actividades que pasan en una web pero que no se ve, ejemplificando, la comunicación con el servidor [16]. Las funcionalidades de la capa de ingreso a datos se fundamentan en el desarrollo de funcionalidades que simplifiquen el proceso, conexión con base de datos y se ocupa de la estabilidad del ciber sitio, en si el plan va a constar en el desarrollo de todo en cuanto a la capa de ingreso a datos backend.

5.4. TECNOLOGÍAS WEB

5.4.1. HTML. HTML, o Lenguaje de marcado de hipertexto (en inglés HyperText Markup Language), es el elemento más fundamental de la Web. Define el sentido y la composición del contenido web. Además, de HTML, se aplican principalmente, otras tecnologías para mostrar la apariencia/presentación de una página web (como CSS) o la funcionalidad/comportamiento (como JavaScript). "Hipertexto" tiene relación con enlaces que conectan páginas web entre sí, así sea en un exclusivo portal web o entre otros HTML usa "marcado" para etiquetar escrito, imágenes y otro contenido para mostrarlo en un navegador web [17].

5.4.2. CSS. CSS es el segundo lenguaje más importante y básico para la creación de páginas web. El primero es HTML, que describe el contenido de su sitio. CSS es un lenguaje que describe la parte de la presentación: cómo mostrar elementos de la página, posiciones, características, distancias, colores y finalmente, la sección de apariencia general [18].

5.4.3. JavaScript. JavaScript es un lenguaje de programación o de secuencias de comandos que permite implementar funciones complejas en páginas web, cada vez que una página web hace algo más que solo mostrar información estática para ser vista, muestra oportunas actualizaciones de contenido, mapas interactivos, animación de Gráficos 2D/3D, desplazamiento de máquinas reproductoras de vídeo, etc. muy probablemente JavaScript está involucrado [19].

5.5 LENGUAJE PYTHON

El lenguaje Python en particular es popular en dominios como la enseñanza, la creación de prototipos y más recientemente, la ciencia de datos. La falta de garantías de seguridad de Python puede ser problemática cuando, como es cada vez más el caso, se utiliza para aplicaciones críticas con altas exigencias de corrección. La comunidad de Python ha reaccionado a esta tendencia integrando anotaciones de tipo y verificación de tipo estático opcional en el lenguaje [20]. Sin embargo, actualmente prácticamente no hay soporte de herramientas para razonar sobre los programas de Python más allá de la seguridad. Es un lenguaje muy eficiente a la hora de consumir recursos de maquina en parte a esto se debe su gran popularidad procesos de Big Data y Machine Learning.

6. METODOLOGÍA

En el siguiente capítulo, se describe la metodología utilizada para el desarrollo de la práctica profesional, describiendo las actividades a desarrollar dentro de la empresa y el método de trabajo, en este caso se utiliza la metodología Scrum, para gestionar el desarrollo de los proyectos.

6.1. ACTIVIDADES DESARROLLADAS

Para la ejecución de la práctica profesional en la empresa Ponce Asesores y Consultores S.A.S. se desarrollaron cuatro fases ligadas al proyecto que se lleva a cabo en las dependencias de la Alcaldía de Itagüí bajo la metodología Scrum, el cual es un método de trabajo en equipo basado en iteraciones o sprints, que tiene como objetivo controlar y planificar proyectos con gran número de cambios de última hora, donde la incertidumbre es alta.

Esta metodología está programada por semanas, al final de cada sprint o iteración, se verifica el trabajo validado de la semana anterior. Sobre esta base, se priorizan y planifican las actividades en las que se invierten los recursos en el próximo sprint. La metodología Scrum se centra en adaptar los resultados y responder a los requisitos reales y precisos del cliente, por lo tanto, se comprueban todos los resultados, ya que los requisitos varían con poca antelación, el tiempo mínimo para un sprint es una semana y el tiempo máximo es cuatro semanas.

6.2. FASES

Fase I (product back log): en esta fase se establece el producto final, para ello se hace un grupo de labores que permiten el inicio del plan, mediante la identificación en grupo con el cliente de los requerimientos necesarios que la aplicación debería cubrir. Se llevan a cabo un grupo de reuniones en donde se logra una recolección de datos en la cual el cliente manifiesta las necesidades que debe cumplir el desarrollo, usando un lenguaje natural comprensible para él.

Habiendo reconocido el grupo de historias del cliente se procede a crearlas usando una plantilla iniciativa por la metodología, el objetivo es que cada una de ellas lleven el mismo orden diseño y sea comprendida tanto por el cliente como por los líderes del proyecto. Se necesita llevar a cabo una lista de los requerimientos cada uno se designa en historia del cliente las cuales se rotulan con una característica del producto escrita a partir de la perspectiva del cliente garantizando que cada funcionalidad le brinde una salida a cada necesidad.

Fase II (sprint back log): en este punto se lleva a cabo la idealización de cada sprint para eso el equipo selecciona las funcionales a realizar basándose en la clasificación de las historias del cliente. Para cada período es primordial estipular un tiempo y una labor a realizar. Se establecen los roles que cada uno debe llevar a cabo en cada sprint.

Fase III (sprint): esta etapa tiene una duración máxima de 4 semanas, en cada aumento del website. El sprint tiene un principio y un final donde se toman como base las prioridades y necesidades de la Alcaldía Itagüí y se definen cuáles y cómo son las funcionalidades que se incorporaran a la aplicación web en siguiente sprint.

Fase IV (crecimiento): se evalúa el resultado del grupo de sprint conseguidos, e incluido como un producto del todo finalizado. En grupo con el cliente se concluye si la aplicación web cumple con la funcionalidad esperada o si se necesita hacer una revisión en la cual se modifica el producto con el objetivo de llevar a cabo lo acordado.

6.3. CRONOGRAMA

A continuación en la tabla 1 se presenta el cronograma de la intervención

Tabla 1. Cronograma de la intervención

OBJETIVO ESPECIFICO	ACTIVIDAD	MES					
		1	2	3	4	5	6
Implementar el módulo de seguridad al iniciar el desarrollo de cada proyecto.	Generar los códigos de seguridad.						
	Agregar los módulos de seguridad necesarios.						
	Realizar pruebas a cada uno de los módulos.						
Desarrollar la estructura para procesos en segundo plano según las necesidades de cada proyecto.	Generar un boceto de estructura con el fin de realizarle cambios.						
	Realizar la estructura de proceso para el portal de pagos						
	Realizar la estructura de proceso para planoteca.						
Realizar las tareas definidas para cada proyecto según los requerimientos obtenidos y aplicando la lógica de negocio en cada uno de los proyectos.	Aplicar tareas para para portal de pagos						
	Aplicar tareas para planoteca.						

Codificar los módulos y funcionalidades según los estándares definidos para cada proyecto.	Codificación de módulos para portal de pagos.							
	Codificación de módulos para planoteca.							
Documentar el código aportado al proyecto siguiendo la especificación PEP 258 para el lenguaje de programación Python.	Generar la documentación bajo los lineamientos de Python PEP 258.							

7. DESARROLLO DE ACTIVIDADES DE PRÁCTICA PROFESIONAL

En este capítulo se desarrolla cada uno de los objetivos planteados en el proyecto, iniciando desde la contextualización de la práctica y las funciones que se realizaron dentro de la empresa, evidenciando las metodologías utilizadas para el desarrollo del backend y la puesta en marcha del desarrollo del aplicativo web para la zona de pagos y la planoteca.

7.1. CONTEXTUALIZACIÓN

Dentro de la empresa en la que se realiza la práctica profesional, el cargo desempeñado fue el de Desarrollador Backend, el cual tiene las siguientes funciones:

1. Soportar a las plataformas y aplicaciones desarrolladas en la Dirección Administrativa de las TIC.
2. Apoyar el Desarrollo de nuevas aplicaciones de acuerdo con los requerimientos dados por los usuarios y cumpliendo los lineamientos y estándares definidos por la Dirección Administrativa TIC.
3. Capacitar en el uso de las aplicaciones desarrolladas.
4. Participar en la definición de estándares y normas aplicables al proceso de desarrollo software.
5. Realizar la documentación del código fuente, según los estándares establecidos por la Dirección administrativa de las TIC.

7.2. DESCRIPCIÓN DE METODOLOGÍAS UTILIZADAS

Para la ejecución de la práctica profesional en la empresa Ponce Asesores y Consultores S.A.S. se implementa el desarrollo bajo la metodología Scrum, al inicio de cada proyecto se realiza una reunión entre los funcionarios de las dependencias involucradas, también está presente un asesor de gobierno en línea que actúa con el rol de product owner y el líder de desarrollo actuando como scrum master, esta reunión permite obtener las necesidades de los funcionarios para desarrollar el proyecto y luego de pasar por una depuración de estas necesidades, se alimenta el backlog del proyecto, dónde se listan todas las tareas sin una prioridad definida.

Luego de lo anterior, el asesor de gobierno en línea, el líder de desarrollo y el equipo de desarrollo tienen una reunión para realizar la planeación del siguiente sprint que tiene una duración definida de dos semanas, se definen las prioridades de las tareas y se realiza la asignación de éstas a cada miembro del equipo.

Al finalizar cada sprint se agenda la reunión sprint retrospective dónde el equipo se reúne para evaluar el avance en el sprint finalizado e identificar situaciones a mejorar para ser implementadas en los siguientes sprint, una vez finalizada esta reunión se inicia con la planeación del nuevo sprint.

En la parte del desarrollo backend se usa el lenguaje de programación Python haciendo uso de framework -estructura base utilizada como punto de partida para elaborar un proyecto con objetivos específicos [21].

Para el proyecto se usa Django: Framework de Python de alto nivel que fomenta el desarrollo rápido y un diseño limpio [22] y además, implementando las reglas de API REST con el framework Django REST Framework, que son más rápidas y ligeras y resultan ideales para el desarrollo de aplicaciones web [23]. En la parte de desarrollo frontend se usa el framework Quasar basado en Vue JS. Como motor de base de datos se implementa PostgreSQL.

Como plataforma para control de versiones de los proyectos y para la administración de tareas, seguimiento de errores e incidencias se trabaja sobre la plataforma GitLab: un proyecto de código libre que se puede instalar en un servidor propio y que permite tener repositorios privados, sin costo [24].

7.3. DESARROLLO PORTAL DE PAGOS

Aplicación web que es el portal de recaudos de impuestos del municipio donde se reciben los pagos de impuestos predial, industria y comercio etc. Se realiza la integración en la parte backend con la pasarela Zona Pagos realizando la revisión de la documentación suministrada por la pasarela, también se relacionan diagramas de flujo para definir claramente el proceso de pago y de esta manera definir qué endpoints se deben crear y los modelos con los campos necesarios, luego se realiza la integración y se realizan pruebas en el proceso de pago.

Luego de la planificación del programa y la definición de requisitos en el diseño del programa se define el modelo de la base de datos como se especifica en la figura 6, y la figura 7 especifica el flujo del portal de pago; así:

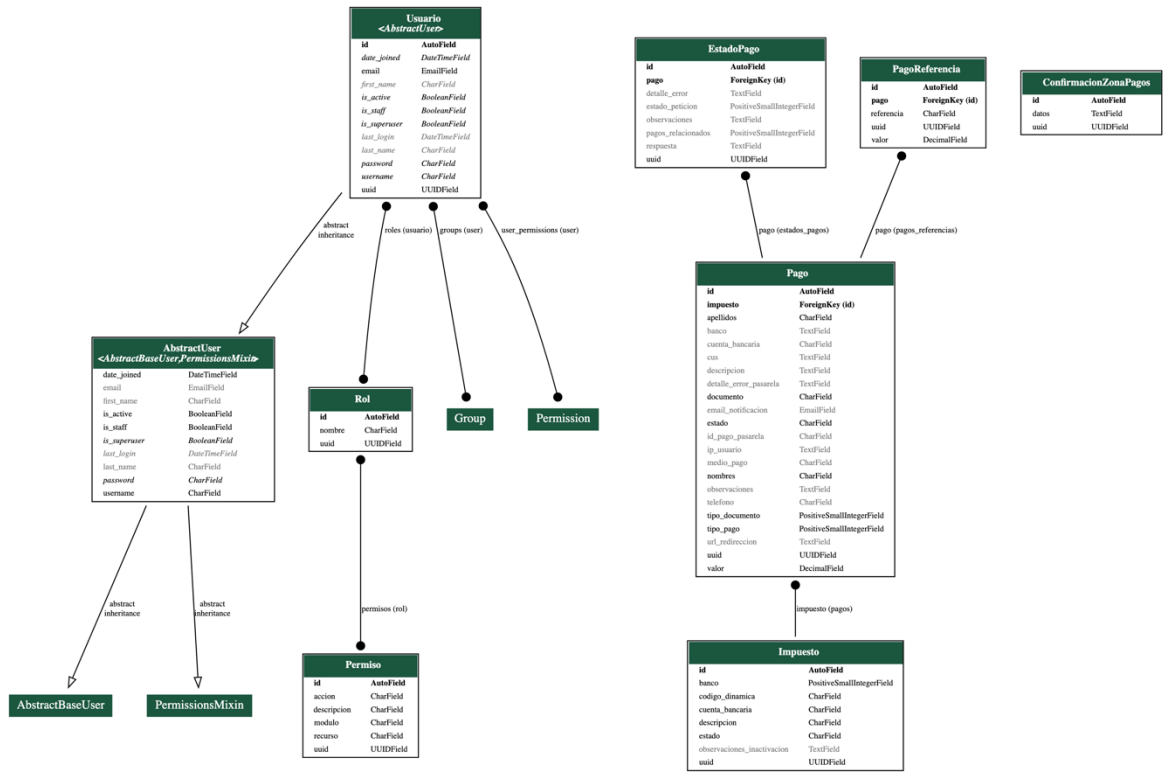


Figura 6. Modelo de la base de datos Portal Pagos

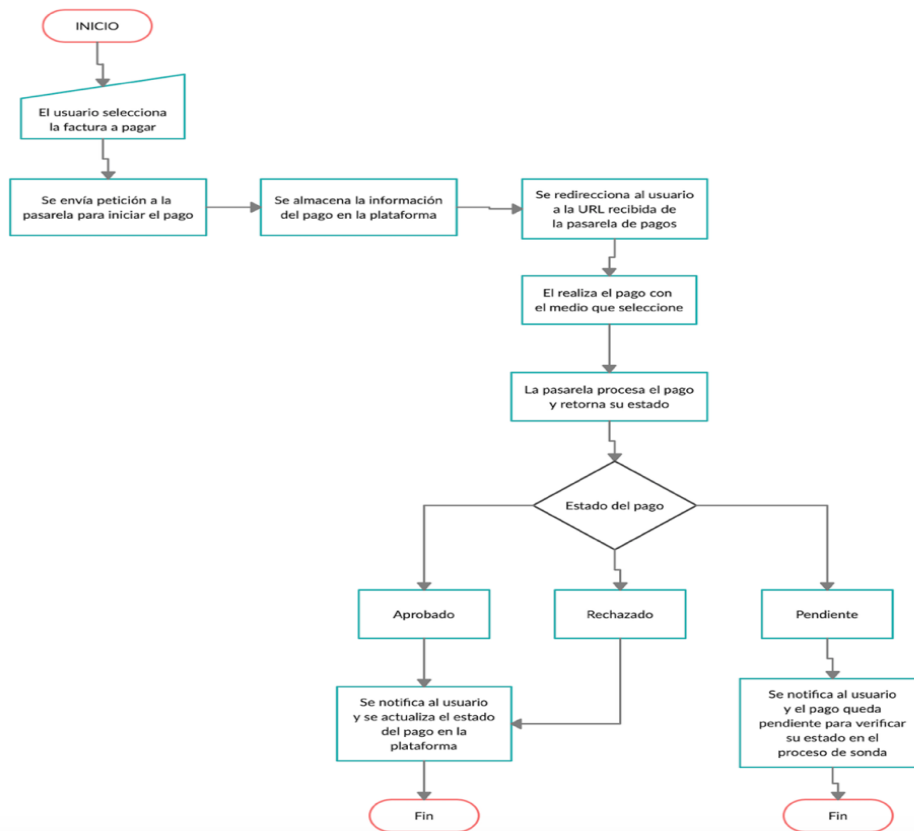


Figura 7. Diagrama de flujo Plataforma de pagos.

En la figura 8, se describe el proceso sonda, el cual reúne los pasos del proceso de verificación de pagos.

DIAGRAMA DE FLUJO DEL PROCESO SONDA (VERIFICACIÓN DE PAGOS)

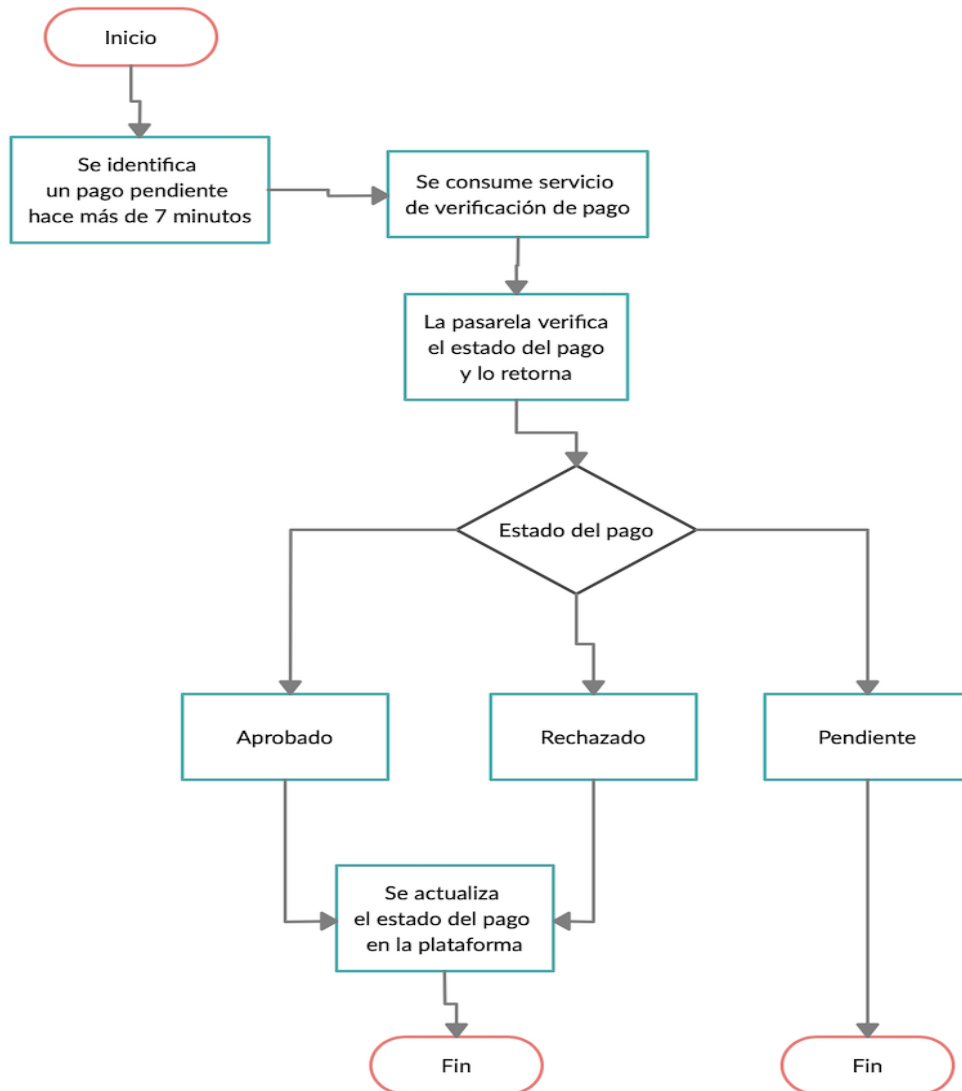


Figura 8. Diagrama de flujo proceso sonda.

Las integraciones o cambios de gran importancia se comparten al grupo de trabajo como Incidencias. En el cumplimiento de las actividades propuestas se comparten las siguientes incidencias en la figura 9.

Recent searches ▾ Assignee Daniel Hernandez ✕ Last updated ▾ ↓

Agregar endpoint de URL de retorno cliente en proceso de pago 0
 #55 - opened 1 week ago by Daniel Hernandez © P PAGOS: Primera versión Backend Funcional updated just now

Integración con zona de pagos 1
 #20 - opened 1 month ago by Julián Hernández © P PAGOS: Primera versión Backend En Revisión updated 1 week ago

Figura 9. Incidencias en portal pagos.

Para mantener control de versiones, se comparten, con el grupo de trabajo, cambios (preferiblemente pequeñas modificaciones) llamados commit. En la siguiente imagen se pueden ver los commits compartidos y una breve descripción de cada uno en la figura 10:

	feat: se agrega ajustes y se inicia el proceso de afctorización en integración con pasarela de pagos Daniel Hernandez authored 23 hours ago	e0289ff7		
	feat: agrega validaciones para verificar un pago en la pasarela. Daniel Hernandez authored 1 day ago	b387dda9		
	feat: actualiza migraciones y modelo para proceso de pagos. Daniel Hernandez authored 1 day ago	512c7f35		
	feat: agrega ajutes en el proceso de inicio de pago. Daniel Hernandez authored 2 days ago	f2fc1f90		
	feat: agrega nuevos ajustes en integración con pasarela de pagos. Daniel Hernandez authored 2 days ago	1461d9cf		
	feat: actualiza validaciones para conexión con Zona Pagos Daniel Hernandez authored 2 days ago	857d5661		
	feat: actualiza requirements Daniel Hernandez authored 2 days ago	9199118f		
	feat: agrega ajuste en modelos y migraciones para el proceso de pagos Daniel Hernandez authored 2 days ago	f4532e24		
	feat: agrega parte de estructura inicial de integración con Zona Pagos, pendien te refactorizar Daniel Hernandez authored 3 days ago	6a0fc435		
	feat: agrega ajustes uniciales para realizar pruebas en la integración con Zona Pagos Daniel Hernandez authored 3 days ago	720d77d0		
	feat: agrega modelos y migraciones para proceso de pago Daniel Hernandez authored 4 days ago	f9b96016		

Figura 10. Commits de Portal de Pagos.

Con los commits relacionados en la figura 10, y el flujo definido en las figuras 7 y 8, se evidencia la participación en gran parte de los módulos y componentes del software.

Los commits relacionados en la figura 10, y las incidencias relacionadas en la figura 9; se sustentan y demuestran con algunos fragmentos de código aportado, donde se evidencia el uso de reglas API REST -de las que más adelante se hablará:

```

from django.urls import path

from views import (ImpuestoCambiaestadoAPIView, ImpuestoListAPIView,
                  ImpuestoRetrieveAPIView, ImpuestoActualizarCuentaAPIView, IniciarPagoAPIView, VerificarPagoAPIView,
                  ConfirmacionPagoAPIView, PagoListAPIView, ListarBancosListAPIView, ImpuestoListNoAuthAPIView,
                  ListarPagosPendientesAPIView)

urlpatterns = [
    path('impuestos/', ImpuestoListAPIView.as_view(), name='pagos-impuestos-api'),
    path('impuestos-no-auth/', ImpuestoListNoAuthAPIView.as_view(), name='pagos-impuestos-no-auth-api'),
    path('impuestos/<uuid:uuid>/', ImpuestoRetrieveAPIView.as_view(), name='pagos-impuestos-api'),
    path('impuestos/<uuid:uuid>/cambiar_estado/', ImpuestoCambiaestadoAPIView.as_view(),
         name='pagos-impuestos-cambiar_estado-api'),
    path('impuestos/<uuid:uuid>/actualizar_cuenta/', ImpuestoActualizarCuentaAPIView.as_view(),
         name='pagos-impuestos-actualizar_cuenta-api'),
    path('iniciar-pago/', IniciarPagoAPIView.as_view(),
         name='pagos-iniciar-pago-api'),
    path('verificar-pago/<uuid:uuid>/', VerificarPagoAPIView.as_view(),
         name='pagos-verificar-pago-api'),
    path('confirmacion-pago/', ConfirmacionPagoAPIView.as_view(),
         name='pagos-confirmacion-pago-api'),
    path('', PagoListAPIView.as_view(),
         name='pagos-api'),
    path('impuestos/listar_bancos/', ListarBancosListAPIView.as_view(),
         name='pagos-impuestos-listar-bancos-api'),
    path('impuestos/consultar_valor/', ListarPagosPendientesAPIView.as_view(),
         name='pagos-impuestos-consultar_valor-api'),
]

from rest_framework import serializers

from models import Impuesto, Pago

class ImpuestoListRetrieveSerializer(serializers.ModelSerializer):
    estado = serializers.SerializerMethodField()
    banco = serializers.SerializerMethodField()

    class Meta:
        model = Impuesto
        fields = ['uuid', 'descripcion', 'observaciones_inactivacion', 'estado', 'cuenta_bancaria', 'banco']

    def get_estado(self, obj):
        return {
            'codigo': obj.estado,
            'descripcion': obj.get_estado_display()
        }

    def get_banco(self, obj):
        return {
            'codigo': obj.banco,
            'descripcion': obj.get_banco_display()
        }

```

Figura 11. parte (a). Código Portal Pagos.

```

import logging
import json
from sentry_sdk import (
    capture_exception,
    capture_message,
    configure_scope
)
)
from django.conf import settings

from interoperabilidad.http.http_request import Client

logging.basicConfig(
    format='%(asctime)s - %(levelname)s - %(message)s',
    level=logging.DEBUG
)

client = Client()

class Cliente:

    def iniciar_pago(self, data):
        response = client._post(settings.ZONA_PAGOS_URL + 'Apis_CicloPago/api/InicioPago', data)
        return client._build_response(response)

    def verificar_pago(self, data):
        response = client._post(settings.ZONA_PAGOS_URL + 'Apis_CicloPago/api/VerificacionPago', data)
        return client._build_response(response)

from django.conf import settings
from config.pagination import Paginacion

from .models import Impuesto, Pago, EstadoPago
from .serializers import (ImpuestoCreateUpdateSerializer, ImpuestoListRetrieveSerializer, PagoListRetrieveSerializer,
                          ImpuestoListNoAuthRetrieveSerializer)
from interoperabilidad.zona_pagos.pagos import MetodosPagos
from apps.seguridad.mixins import PermisoRequeridoMixin
from interoperabilidad.http.http_request import Client

class ImpuestoListAPIView(PermisoRequeridoMixin, ListAPIView):
    """
    Se encarga de listar los impuestos, soporta los métodos:
    GET
    """
    queryset = Impuesto.objects.all()
    serializer_class = ImpuestoListRetrieveSerializer
    lookup_field = 'uuid'
    filter_backends = (filters.SearchFilter, filters.OrderingFilter)
    search_fields = ('descripcion',)
    ordering = ('descripcion',)
    pagination_class = Paginacion
    permisos_requeridos = {
        "GET": (settings.PERMISO_VER_IMPUESTOS, settings.PERMISO_CAMBIAR_ESTADO_IMPUESTOS,
               settings.PERMISO_CAMBIAR_CUENTA_IMPUESTOS),
    }
}

```

Figura 11 parte (b). Código Portal Pagos.

Con los fragmentos de código relacionados en la figura 11, se evidencia parte de la aplicación de Django como framework en los componentes del software, siguiendo la documentación de éste.

7.4. DESARROLLO PLANOTECA

La planoteca es la aplicación web para la administración de planos a cargo de la administración municipal, donde se registran los planos por parte de las curadurías municipales, se registran también las solicitudes de consulta de planos por parte de los ciudadanos con el fin de modernizar el sistema y facilitar el proceso de la gestión de planos.

Luego de la planificación del programa y la definición de requisitos, en el diseño del programa se definió el modelo de base de datos como se muestra en la figura 12, y a su vez que Planoteca se integrará por los componentes en la figura 13.

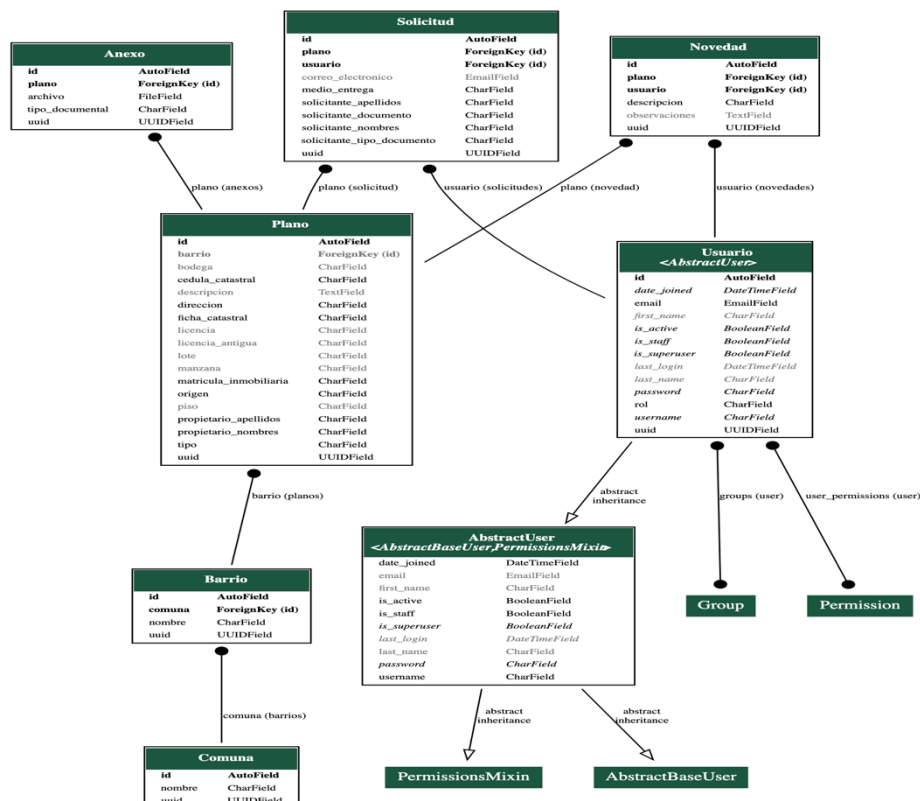


Figura 12. Modelo de base de datos Planoteca

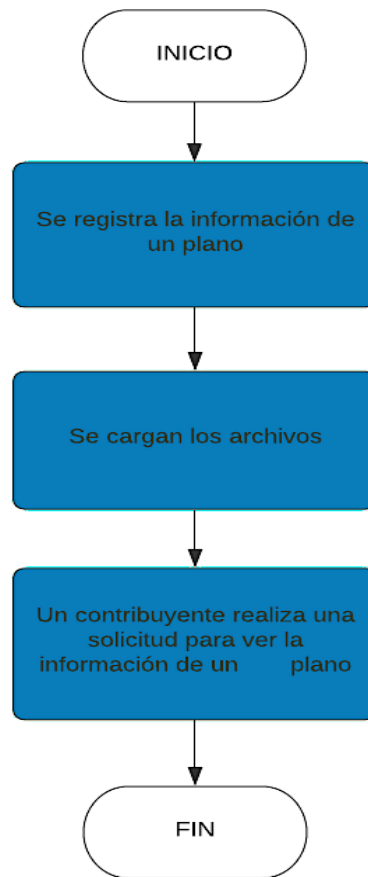


Figura 13. Componentes Planoteca.

Se realiza el desarrollo de la primera versión de la aplicación en la parte de backend trabajando muy de la mano con el equipo frontend para facilitar el proceso de pruebas e integración, donde desde el inicio del desarrollo se realiza la planeación, asignación y se definen las fechas límites para la entrega de las tareas.

Al igual que en el Portal de pagos, se relacionan las incidencias solucionadas durante el proyecto Planoteca en la figura 14.

Recent searches ▾
Assignee Daniel Hernandez ✕
Last updated ▾

Endpoints de la traza de consulta de planos

#9 · opened 2 weeks ago by Julián Hernández Versión 1 Backend

CLOSED 1

updated just now

Endpoints para gestionar documentos de un plano

#7 · opened 2 weeks ago by Julián Hernández Versión 1 Backend

CLOSED 1

updated 5 days ago

Endpoints para CRUD de planos #6 · opened 2 weeks ago by Julián Hernández · Versión 1 Backend	CLOSED 1 updated 1 week ago
Como administrador del sistema deseo administrar Usuarios, con fin de dar acceso al sistema a personas que los requieran. #3 · opened 2 weeks ago by Julián Hernández · Versión 1 Backend	CLOSED 2 updated 2 weeks ago
Como administrador del sistema deseo generar clave a los usuarios #4 · opened 2 weeks ago by Julián Hernández · Versión 1 Backend	CLOSED 2 updated 2 weeks ago
Como usuario deseo editar los datos de mi perfil con el fin de mantener actualizada mi cuenta y poder cambiar mi contraseña #5 · opened 2 weeks ago by Julián Hernández · Versión 1 Backend	CLOSED 2 updated 2 weeks ago
Como usuario deseo autenticarme con el fin de ingresar al sistema #2 · opened 2 weeks ago by Julián Hernández · Versión 1 Backend	CLOSED 2 updated 2 weeks ago
Creación de la estructura inicial del proyecto #1 · opened 2 weeks ago by Julián Hernández · Versión 1 Arquitectura	CLOSED 1 updated 2 weeks ago

Figura 14. Incidencias Planoteca.





































A continuación, en la figura 15 se relaciona un listado de todos los commits realizados.

Merge branch 'develop' into 7-feature/anexos Daniel Hernandez authored 5 days ago	695aea76
feat: actualiza archivo Insomnia.json con el endpoint para listar los tipos... ... Daniel Hernandez authored 5 days ago	449f6612
feat: agrega endpoint para listar los tipos documentales de los anexos Daniel Hernandez authored 5 days ago	958408e0
fix: realiza ajuste menor en endpoint para agregar anexos a un plano Daniel Hernandez authored 5 days ago	48526b5d
feat: actualiza archivo Insomnia.json agregando endpoint para eliminar anexos de un plano Daniel Hernandez authored 5 days ago	39834f55
feat: agrega endpoint para eliminar anexos de un plano Daniel Hernandez authored 5 days ago	c947ed82
Merge branch 'develop' into 7-feature/anexos Daniel Hernandez authored 6 days ago	8e881cb3
feat: actualiza archivo Insomnia.json con los endpoints de anexos Daniel Hernandez authored 6 days ago	e58896cc













Figura 15. Commits Planoteca.













Continuación figura 15. Commits Planoteca

feat: agrega endpoint con las respectivas valuaciones para cargar anexos a un plano Daniel Hernandez authored 6 days ago	3b638721
fix: realiza ajuste en metodos para convertir archivos Daniel Hernandez authored 6 days ago	6cc932aa
fix: realiza ajuste en serializador de Anexo Daniel Hernandez authored 6 days ago	767e9d60
feat: agrega valores en la configuración general Daniel Hernandez authored 6 days ago	3ed30885

 feat: agrega serilizador para el modelo Anexo Daniel Hernandez authored 1 week ago	832ddce9		
 feat: agrega migración para el modelo Anexo Daniel Hernandez authored 1 week ago	12400202		
 feat: agrega modelo Anexo y metodos para validar los archivos cargados Daniel Hernandez authored 1 week ago	b04378f0		
 feat: agrega migración y archivo para el cargue inicial de barrios y comunas del municipio Daniel Hernandez authored 1 week ago	26058442		
 fix: realiza ajuste menor en endpoint de listar planos Daniel Hernandez authored 1 week ago	9730e057		
 feat: actualiza archivo Insomnia.json con los endpoints del CRUD de planos Daniel Hernandez authored 1 week ago	0a91d11c		
 feat: registra las urls de las aplicaciones core y planos Daniel Hernandez authored 1 week ago	50c29082		
 feat: agrega endpoints para listar, crear, editar y eliminar planos, tambien... Daniel Hernandez authored 1 week ago	b9ba4daf		
 feat: agrega endpoints en la aplicación core con los listados de comunas y barrios Daniel Hernandez authored 1 week ago	4a06a256		
 feat: realiza ajuste den migración del modelo Plano Daniel Hernandez authored 1 week ago	1fba3808		
 feat: agrega vistas para los endpoints de Planos Daniel Hernandez authored 1 week ago	fcf84a15		
 feat: agrega los demás ficheros de la aplicación planos y registra las... Daniel Hernandez authored 1 week ago	82c1ddf2		
 feat: agrega los serializadores para iniciar la creación de los endpoints del CRUD de planos Daniel Hernandez authored 1 week ago	7b1d77f0		
 feat: agrega aplicación core Daniel Hernandez authored 1 week ago	4386833b		
 feat: agrega modelo y migraciones para el CRUD de planos. Daniel Hernandez authored 1 week ago	cc0ad21b		
 feat: actualiza archivo Insomnia.json Daniel Hernandez authored 2 weeks ago	41035c1c		
 feat: agrega endpoints para la gestión de usuarios Daniel Hernandez authored 2 weeks ago	d23b85ca		

Continuación figura 15. Commits Planoteca

 feat: agrega migración inicial Daniel Hernandez authored 2 weeks ago	1d9fc512		
 feat: agrega ajustes en los requirements Daniel Hernandez authored 2 weeks ago	2c62ccd1		
 feat: agrega ajustes en las configuración de Docker Daniel Hernandez authored 2 weeks ago	186c3650		
 feat: realiza ajustes en la configuración del proyecto Daniel Hernandez authored 2 weeks ago	0de04d2d		

 feat: agrega funcionalidad de login Daniel Hernandez authored 2 weeks ago	b14d41e7		
 fix: realiza ajuste en entorno de desarrollo Daniel Hernandez authored 2 weeks ago	2af510a5		
 Agrega versión inicial del proyecto. Daniel Hernandez authored 2 weeks ago	10efedc8		
 Adiciona README.md Daniel Hernandez authored 2 weeks ago	c79424de		

Pese a la gran cantidad de aportes, cambios, integraciones y pruebas realizadas en el proyecto Planoteca; a continuación, en la figura 16 se muestran solo algunos de los aportes de código.

```

from apps seguridad serializers import UsuariolistRetrieveSerializer
from apps core serializers import (ComunalistRetrieveSerializer, BarriolistRetrieveSerializer)

class PlanoListRetrieveSerializer(serializers.ModelSerializer):
    numero = serializers.SerializerMethodField()
    tipo = serializers.SerializerMethodField()
    origen = serializers.SerializerMethodField()
    barrio = BarriolistRetrieveSerializer()

    class Meta:
        model = Plano
        fields = ['numero', 'uuid', 'tipo', 'origen', 'cedula_catastral', 'matricula_inmobiliaria',
                 'ficha_catastral', 'licencia_antigua', 'licencia', 'propietario_nombres', 'propietario_apellidos',
                 'direccion', 'barrio', 'manzana', 'lote', 'piso', 'bodega', 'descripcion']

    def get_numero(self, obj):
        return obj.pk

    def get_tipo(self, obj):
        return {
            'codigo': obj.tipo,
            'descripcion': obj.get_tipo_display()
        }

    def get_origen(self, obj):
        return {
            'codigo': obj.origen,
            'descripcion': obj.get_origen_display()
        }

```

Figura 16. Aportes de código del proyecto Planoteca.

Continuación figura 16. Aportes de código del proyecto Planoteca.

```
from django.urls import path

from .views import (
    PlanoListCreateAPIView,
    PlanoRetrieveUpdateDestroyAPIView,
    PlanoTipoListAPIView,
    PlanoOrigenesTipoListAPIView,
    AnexoListCreateAPIView,
    AnexoDestroyAPIView,
    AnexoTipoDocumentalListAPIView,
    NovedadListAPIView,
    SolicitudListCreateAPIView,
    SolicitudMedioEntregaListAPIView,
    PlanoSolicitudesAPIView,
    SolicitanteTipoDocumentoListAPIView,
    SolicitanteSolicitudesAPIView,
    VerAnexoAPIView
)

urlpatterns = [
    path('', PlanoListCreateAPIView.as_view(), name='planos-listado-api'),
    path('<u>uid:uid/</u>', PlanoRetrieveUpdateDestroyAPIView.as_view(), name='planos-detalle-api'),
    path('tipos/', PlanoTipoListAPIView.as_view(), name='planos-tipos-api'),
    path('origenes/', PlanoOrigenesTipoListAPIView.as_view(), name='planos-origenes-api'),
    path('<u>uid:uid_plano/</u>anexos/', AnexoListCreateAPIView.as_view(), name='planos-anexos-api'),
    path('<u>uid:uid_plano/</u>anexos/<u>uid:uid/</u>', AnexoDestroyAPIView.as_view(), name='planos-anexos-eliminar-api'),
    path('anexos/tipos/', AnexoTipoDocumentalListAPIView.as_view(), name='planos-anexos-tipos-api'),
    path('<u>uid:uid_plano/</u>traza/', NovedadListAPIView.as_view(),
        name='planos-novedades-api'),
    path('<u>uid:uid_plano/</u>solicitudes/', SolicitudListCreateAPIView.as_view(), name='planos-solicitudes-api'),
    path('solicitudes/medios-entrega/', SolicitudMedioEntregaListAPIView.as_view(),
        name='planos-solicitudes-medios-entrega-api'),
    path('solicitudes-cantidad/', PlanoSolicitudesAPIView.as_view(),
```

8. RESULTADOS

8.1. PRUEBAS REALIZADAS

8.1.1. Pruebas portal de pagos. Durante la implementación del software se hacen pruebas del Portal de Pagos en la herramienta Insomnia, que sirve para gestionar y realizar pruebas a los endpoints [25] y así garantizar que estén respondiendo correctamente y entregando la información esperada. A continuación, en la figura 17 se relacionan algunas de las pruebas realizadas, y con las que se evidencia que en los endpoints tienen el funcionamiento esperado.

The figure displays three screenshots from the Insomnia API client, each showing a different test result for the Hacienda portal. Each screenshot includes a sidebar with a list of endpoints, a central area for the request details, and a right-hand panel for the response details.

- Top Screenshot:** A GET request to `server_host/hacienda/api/v1/pagos/` with a status of **200 OK**, 26.3 ms, and 541 B. The response body is a JSON object containing payment details such as `uid`, `impuesto`, `descripcion`, `estado`, `codigo`, `descripcion`, `referencia`, `documento`, `tipo_documento`, `codigo`, `descripcion`, `nombres`, `apellidos`, `telefono`, `descripcion`, `estado`, `codigo`, `descripcion`, `tipo_pago`, `codigo`, `descripcion`, and `valor`.
- Middle Screenshot:** A POST request to `server_host/hacienda/api/v1/pagos/iniciar-pago/` with a status of **200 OK**, 464 ms, and 139 B. The response body is a JSON object containing a `detail` field with a URL: `https://zonapagos.com/ciclonpago/Pago.aspx?ruta=5F9F827C23C3E6C41B17F69A149F5202C3470B139375C88884BE1A43A2A8998943603A6083CE2A`.
- Bottom Screenshot:** A GET request to `server_host/hacienda/api/v1/pagos/confirmacion-pago/` with a status of **200 OK**, 225 ms, and 60 B. The response body is a JSON object containing `id_comercio` (29296) and `id_pago` (2233445566). The `detail` field contains the message: "Se actualizó el estado del pago correctamente."

Figura 17. Pruebas Portal Pagos.

8.2. PRUEBAS PLANOTECA

De igual manera, para Planoteca también se realizan diversas pruebas para comprobar que las peticiones se resuelven de manera correcta. En la figura 18 se muestran algunas de las pruebas realizadas.

The figure displays three screenshots of the Planoteca API client interface, each showing a different HTTP request and its corresponding response.

Top Screenshot: A POST request to `/planoteca/api/v1/planos/d7499a2d-42f0-47ae-81e0-8d4` with a status of `201 Created`. The response is a JSON object with the following structure:

```
1 {
2   "solicitante_tipo_documento": "CC",
3   "solicitante_documento": "1088",
4   "solicitante_nombres": "Luis",
5   "solicitante_apellidos": "Martinez",
6   "medio_entrega": "CO",
7   "correo_electronico": "luis@ttt.com"
8 }
```

Middle Screenshot: A GET request to `/planoteca/api/v1/planos/38fb60cc-cfa8-4c91-9e5f-a88e0` with a status of `200 OK`. The response is a JSON object with the following structure:

```
1 {
2   "uid": "da2726d2-29b3-41f7-80fc-83ef981c6e94",
3   "plano": "38fb60cc-cfa8-4c91-9e5f-a88e07578d0e",
4   "usuario": {
5     "uid": "5f6e9cf7-cf13-4e56-bae3-811b7bc293a5",
6     "username": "admin",
7     "first_name": "",
8     "last_name": "",
9     "email": "admin@test.com",
10    "rol": {
11      "codigo": "AD",
12      "descripcion": "Administrador"
13    }
14  },
15  "solicitante_tipo_documento": {
16    "codigo": "CC",
17    "descripcion": "Cédula de ciudadanía"
18  },
19  "solicitante_documento": "1088",
20  "solicitante_nombres": "Luis",
21  "solicitante_apellidos": "Martinez",
22  "medio_entrega": {
23    "codigo": "CO",
24    "descripcion": "Correo electrónico"
25  },
26  "correo_electronico": "luis@ttt.com",
27  "created": "2020-11-06T15:12:46.419218-05:00"
28 },
29 }
```

Bottom Screenshot: A GET request to `/planoteca/api/v1/planos/solicitudes-cantidad/` with a status of `200 OK`. The response is a JSON array with the following structure:

```
1 [
2   {
3     "fecha_inicio": "2020-11-01",
4     "fecha_fin": "2020-11-09",
5     "orden": "--cantidad"
6   },
7   {
8     "fecha_inicio": "2020-11-10",
9     "fecha_fin": "2020-11-18",
10    "orden": "--cantidad"
11  }
12 ]
```

Figura 18. Pruebas Planoteca

En conclusión, con las ilustraciones 17 y 18 que muestran algunas de las pruebas realizadas al Portal de Pagos y a Planoteca, respectivamente; se demuestra un correcto funcionamiento en los endpoints y en las áreas en las que se tuvo participación en ambos proyectos.

8.3. RECOMENDACIONES FINALES DEL PROCESO DE PRÁCTICA

Teniendo en cuenta las situaciones que limitaron o retrasaron el desarrollo y la entrega de los objetivos en los tiempos planeados principalmente se debió a falta de claridad de las necesidades por parte de las dependencias necesitadas, ya que al momento de realizar las reuniones y el levantamiento de requisitos muchas veces no se lograba aterrizar las necesidades fácilmente dando pie a ambigüedades.

Por lo tanto, cabe resaltar que, para la realización de proyectos a futuro, se debe de realizar un diagnóstico previo a los requisitos por parte de los clientes, con el fin de realizar una mejor gestión de tiempo y de recursos en pro del desarrollo de actividades más eficientes.

Tener en cuenta la cantidad de procesos repetitivos y usuarios que tendría la aplicación para que se pueda implementar en un equipo con los recursos adecuados para funcionar correctamente.

9. CONCLUSIONES

En conclusión, esta práctica empresarial permitió aterrizar los conceptos observados durante la carrera permitiendo ponerlos a prueba en un entorno empresarial, conociendo las tendencias en herramientas de desarrollo y las arquitecturas más usada en el momento de hacer un desarrollo para una empresa privada o pública en este caso. Sumado a lo anterior, el desarrollo de la práctica aportó al enriquecimiento de las habilidades blandas necesarias en el contexto laboral, además, de, poner a prueba todo los conocimientos adquiridos en la formación universitaria, colocando nuevos retos tanto en temas conocidos como desconocidos enfocados en el desarrollo de nuevas tecnologías que generan una exigencia personal para estudiar más a profundidad cada tema y de esta manera dar una solución más oportuna.

Adicional a lo anterior, la participación en un proyecto de la empresa Ponce Asesores y Consultores S.A.S. permitió adquirir conocimientos para la vida profesional y enriquecimiento a nivel personal, ya que se generaron aportes e ideas que para el inicio del proyecto fueron importantes, esto permitió elevar los niveles de afianzamiento en cuanto al desarrollo de planes de trabajo en un nivel administrativo, también, para la ejecución de las actividades enfocadas en el desarrollo de las aplicaciones para la Alcaldía de Itagüí resaltando la importancia de la seguridad de la información, proceso que es de suma importancia al tratarse de información sensible de funcionarios y contribuyentes del municipio lo que exigió de aptitudes y conocimientos relacionados con la seguridad para aplicarla a este tipo de proyectos que son sensibles y buscan garantizar la seguridad de los ciudadanos.

Por otro lado, para alinearse con los objetivos planteados para este proyecto, se utilizó la metodología de desarrollo Scrum que permitió diseñar y mejorar una aplicación web que mejoró el portal de pagos y gestión de recaudo de la secretaria de hacienda y la planoteca de la secretaria general de la Alcaldía de Itagüí. En este punto, se logró analizar toda la información suministrada por él para evaluar que hacía falta en la plataforma y proponer con la aplicación como se puede mejorar los procesos de tal manera que permitan la optimizar de los datos.

El resultado de cada objetivo ha permitido darle respuesta a la necesidad evidenciada por la institución solicitante, de esta forma se logró obtener una perspectiva más clara de lo que buscaban desarrollar por medio de historias de usuarios, diagramas, base de datos y prototipos funcionales, dando como resultado una aplicación web capaz de consumir con las necesidades planteadas en la obtención de requerimientos. Lo anterior definido dentro de un plan de trabajo que definió los procesos que la Alcaldía de Itagüí querían optimizar, por medio de este desarrollo se procedió a elaborar diferentes entornos, como lo son el frontend encargado del diseño y la parte gráfica, y el backend encargado de la funcionalidad y el servidor.

En el análisis que se llevó a cabo se encontró que la aplicación web presenta un desempeño adecuado en base a las pruebas realizadas, sin embargo, en el transcurso del desarrollo del proyecto, se reconoció el desconocimiento y uso de un sitio web porque, por un lado, es consciente de su existencia, pero no tienen idea de lo que pueden o no encontrar en él, siendo un elemento constante entre los funcionarios de la alcaldía.

10. RECOMENDACIONES

Se recomienda a estudiantes que están cercanos a iniciar una vida laboral, formarse complementariamente en el conocimiento de sistemas de control de versiones, el cual es indispensable en las competencias requeridas para un desarrollador de software en el mercado actual con el fin de conocer las nociones básicas sobre el cómo, el por qué y el para qué se hace.

Para el desarrollo de frontend, se recomienda a los estudiantes y desarrolladores en general, utilizar diferentes gamas de colores que resalten la utilidad de cada una de las herramientas que se colocan en las aplicaciones web, con el fin de mejorar la navegabilidad de los usuarios a su vez que se le da una identidad a esta.

También, se recomienda desarrollar los conocimientos y habilidades enfocados en las herramientas de lenguaje de programación como Ruby, Python y JavaScript, las cuales en la actualidad han venido teniendo gran acogida dentro del gremio por su versatilidad.

11. REFERENCIAS

- [1] M. Bouskela, M. Casseb, S. Bassi, C. De Luca, M. Facchina, «¿Qué es, al final de cuentas, una Smart City?,» La ruta hacia las Smart Cities, pp. 33-38, 2016.
- [2] A. Alujas, «Innovación en la Gestión Pública y Open Government (Gobierno Abierto): Una Vieja Nueva Idea,» Revista Buen Gobierno, vol. 2, 2010.
- [3] M. D. Arango Serna, J. E. Londoño Salazar, J. A. Zapata Cortés, «Arquitectura empresarial una vision general,» Revista Ingenierías de la Universidad de Medellín, vol. 9, nº 12, pp. 16-19, 2010.
- [4] DATIC, «Organigrama de la DATIC,» 2019. [En línea].
- [5] VIEWNEXT, «VIEWNEXT,» [En línea]. Available: <https://www.viewnext.com/el-ciclo-sdlc-en-7-fases/>.
- [6] VIEWNEXT, «Gestión y desarrollo de aplicaciones web,» 2019. [En línea]. Available: <https://www.viewnext.com/gestion-y-desarrollo-de-aplicaciones/>.
- [7] Ministerio de Tecnologías de la Información, «Decreto 2573 de 2014,» 2014. [En línea]. Available: https://www.mintic.gov.co/portal/604/articulos-14673_documento.pdf.
- [8] Congreso de la republica de Colombia, «Ley 1712 de 2014,» 2014. [En línea]. Available: <http://wsp.presidencia.gov.co/Normativa/Leyes/Documents LEY%201712%20DEL%2006%20DE%20MARZO%20DE%202014.pdf>.
- [9] L. Flores, «¿Qué Es Una Aplicación Web?,» Strapp. Web & Mobile Developers, 2019.
- [10] S. Luján, Programación De Aplicaciones Web, Alicante: Editorial Club Universitario, 2002.
- [11] H2tic, «Desarrollo de aplicaciones web,» 2018. [En línea]. Available: <https://www.h2tic.com/desarrollo-aplicaciones-web-malaga-jaen.html>.
- [12] P. Stefaniak, «¿Qué es Backend Y Frontend?,» Descubre Comunicación, 2020.
- [13] D. Villalba, «Desarrollo de una plataforma backend y frontend para la gestión de contenidos,» Universidad Carlos III de Madrid, Madrid, 2015.
- [14] K. E. Kendall, J. E. Kendall, Análisis y diseño de sistemas, México D.F.: Prentice Hall, 2005.
- [15] A. Fernández, «Estévez Fernández A. Adaptación de herramienta web para su correcta utilización por usuarios con discapacidad visual (Doctoral dissertation).,» 2017.
- [16] R. Arjonilla, «Backend,» 2016. [En línea]. Available: <https://rafarjonilla.com/que-es/backend/>.
- [17] MDN, «HTML: Lenguaje De Etiquetas De Hipertext,» 2020. [En línea]. Available: <https://developer.mozilla.org/es/docs/Web/HTML>.
- [18] Desarrolloweb.com, «CSS,» 2020. [En línea]. Available: <https://desarrolloweb.com/home/css>.

- [19] MDN, «¿Qué es Javascript?,» 2020. [En línea]. Available: https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/Qu%C3%A9_es_JavaScript.
- [20] G. van Rossum, J. Lehtosalo, L. Langa, «Python,» 2014. [En línea]. Available: <https://www.python.org/dev/peps/pep-0484/>.
- [21] Gabriela, «Rock Content,» [En línea]. Available: <https://rockcontent.com/es/blog/framework/>.
- [22] Django, «Djangoproject,» [En línea]. Available: <https://www.djangoproject.com/>.
- [23] Redhat, «Redhat,» [En línea]. Available: <https://www.redhat.com/es/topics/api/what-is-a-rest-api>.