

**PROYECCIÓN Y APLICACIÓN DE MEJORAS
REQUERIDAS POR LA EMPRESA CİNNOV S.A.S EN EL
ÁREA DE DESARROLLO Y TECNOLOGÍA**

Autor:

Jhorman Andrés Galindo Polania

Director:

Cristhian Ivan Riaño Jaimes

TRABAJO DE GRADO

Junio, 2022

UNIVERSIDAD DE PAMPLONA
FACULTAD DE INGENIERÍA Y ARQUITECTURA
2022

Dedicatoria

Este proyecto final me lo dedico a mí mismo, por ser capaz de llegar a este punto de la vida, creer siempre en mí y por las capacidades y conocimiento que he desarrollado por mi esfuerzo y ganas de salir adelante.

Agradecimientos

Primeramente, agradecer a la vida por ponerme en esta situación y facilitarme salir adelante a pesar de todos los obstáculos, siempre han salido las cosas de la mejor manera para cumplir todos mis objetivos. Mi inteligencia, responsabilidad y ganas de emprender han sido pioneros a nivel personal para poder cumplir todo lo que me propongo.

A mis familiares, especialmente a mi abuelita María Jovita por velar por mí en los tiempos más difíciles para poder sacar a adelante mis estudios.

Mi madre Yenny Polanía y mi padre Helbert Galindo por siempre estar para mí cuando más lo necesito y darme todas las herramientas y consejos para cumplir todo lo que me proponga en la vida. A ellos le debo todo lo que soy hoy en día.

Agradecer a todos mis docentes de la Universidad de Pamplona, especialmente al Ingeniero Cristhian Riaño por su paciencia y vocación por la enseñanza. Su papel fue muy importante en mi proceso de formación y ha contribuido para que este proyecto bajo su dirección, saliera adelante de la mejor manera.

Por último, quiero agradecer a mi pareja, amigos y colegas de la Universidad por su aporte en mi proceso de formación de vida y mi carrera universitaria.

Índice

1. Introducción	10
1.1. Resumen	10
1.2. Abstract	11
1.3. Contexto y condiciones del entorno de desarrollo	12
1.3.1. Ingreso a la empresa y conocimientos básicos.	12
1.3.2. Capacitaciones y adaptación de entorno para desarrollar las tareas.	12
1.4. Producto de la pasantía	13
1.4.1. Vinculación a un entorno laboral	13
1.4.2. Página WEB de la empresa	13
1.5. Definición del problema	13
1.6. Contexto y condiciones del problema	14
1.6.1. Implementar la condición de batería del dispositivo Bigo en la aplicación móvil	14
1.6.2. Investigar actualización de un sketch de forma inalámbrica	14
1.6.3. Desarrollo de una página web informativa	14
1.7. Objetivos	15
1.7.1. Objetivo general	15
1.7.2. Objetivos específicos	15
2. Estructura del documento	16
2.1. Capítulo I, Introducción	16
2.2. Capítulo II, Estado del Arte	16
2.3. Capítulo III, Metodología y plan de trabajo	16
2.4. Capítulo IV, Requerimientos de diseño	16
2.5. Capítulo V, Modelo de implementación	16
3. Estado del arte y Marco teórico	17
3.1. Aplicación móvil informativa de apoyo académico de la Universidad Politécnica Sa-	
lesiana	17
3.2. Implementación de una aplicación WEB con servicio de chatbot con inteligencia	
artificial que permita la autogestión de cuentas por pagar de los proveedores de la	
Universidad Autónoma de Bucaramanga	23
4. Metodología y plan de trabajo	28
4.1. Esquema metodológico	28
4.2. Descripción de esquema metodológico	28
4.3. Recursos necesarios	30
5. Requerimientos de diseño	31
5.1. Inicio	32
5.2. Conócenos	35
5.3. Nuestros servicios	36
5.4. Tecnologías	41
5.5. Contacto	41
6. Modelo de implementación	43
6.1. Rutas de navegación	43
6.2. Menú de navegación	45
6.3. Estructura de <i>Views</i>	48
6.4. <i>Widgets</i> reutilizables	50
7. Conclusiones	53

Índice de figuras

1.	Arquitectura del sistema	18
2.	Funcionamiento de manejo de estado con BloC	20
3.	-UI- Mockup de la Aplicación.	20
4.	-UI- Pantallas finales de la Aplicación	22
5.	-UI- Tabs bar, Noticias y Eventos	22
6.	Anatomía de un ChatBot	24
7.	Creación del proyecto en <i>Dialogflow</i>	25
8.	Tablero Kanban	26
9.	Ventana principal de la aplicación web en Flutter	26
10.	Resultado de consulta de facturas a través del chatbot	27
11.	Resultado de consulta de historial de pagos en <i>dashboard</i>	27
12.	Esquema metodológico.	28
13.	Esquema de modelado funcional	31
14.	Página de bienvenida	32
15.	Página informativa, ¿Quiénes somos?	33
16.	Página informativa, Aliados	34
17.	Menú de la página	34
18.	Línea de tiempo	35
19.	Equipo de trabajo	35
20.	Servicios ofrecidos por la empresa	36
21.	Servicios ofrecidos por la empresa: Asesoría	37
22.	Servicios ofrecidos por la empresa: Consultoría	37
23.	Servicios ofrecidos por la empresa: Acompañamiento Técnico	38
24.	Servicios ofrecidos por la empresa: Generación de Productos	38
25.	Servicios ofrecidos por la empresa: Formación y Desarrollo	39
26.	Servicios ofrecidos por la empresa: Investigación	39
27.	Proceso de producción	40
28.	Productos	40
29.	Tecnologías	41
30.	Pestaña de contacto	42
31.	Diagrama de secuencia: Contacto	42
32.	Estructura de código: Cuerpo	43
33.	Barra de navegación	44
34.	Estructura de código: Rutas de navegación	44
35.	Estructura de código: Diseño principal del código	45
36.	Representación del cuerpo de la página	46
37.	Estructura de código: Menú	47
38.	Estructura de código: Inicio	48
39.	Pestaña de Inicio completa	49
40.	Pestaña de Contacto: Redes sociales	50
41.	Estructura de código: <i>Widget ImageUrl</i>	51
42.	Estructura de código: Llamado de <i>Widget ImageUrl</i>	51

Siglas

API Application Programming Interfaces. 25

HTTP Hypertext Transfer Protocol. 19, 24

IoT Internet de las cosas. 15

JEE *Java Enterprise Edition*. 19

JSON JavaScript Object Notation. 21, 24

OTA Over The Air. 10, 11, 30

PC computador personal. 14

PHP Hypertext Preprocessor. 24

REST Representational State Transfer. 18

SDK Kit de Desarrollo de Software. 14, 24

UI Interfaz de usuario. 10, 11, 19, 29

UPS Universidad Politécnica Salesiana. 18

URL Localizador de Recursos Uniforme. 24

1. Introducción

1.1. Resumen

El área de Tecnología e innovación para el desarrollo de software, es un departamento de la empresa CiNNoV S.A.S, el cual se encarga de analizar, diseñar y desarrollar los sistemas internos que se requieren para su crecimiento. Mediante el presente escrito se describe cómo a través de la práctica empresarial de manera virtual se realizaron las funciones de ingeniero junior, brindando apoyo en la implementación del dispositivo Bigo, participar en la investigación para ejecutar el sistema Over The Air (OTA) y para la creación de la página web de CiNNoV S.A.S, con ello se pudo adquirir conocimiento en el desarrollo de diseño Interfaz de usuario (UI) de aplicaciones y el uso del módulo ESP32, entre otras. Inicialmente, la empresa ofreció material de estudio y capacitaciones que permitían conocer las herramientas y los lenguajes que se ejecutaban en el cargo. De acuerdo a esto, se me asignó la tarea de llevar a cabo el intercambio de datos entre el módulo y la aplicación móvil, para que, mostrará el porcentaje de batería en tiempo real del dispositivo Bigo, además, se tuvo participación en la investigación de la tecnología OTA para ser implementada dentro del dispositivo Bigo y así poder realizar la actualización del *sketch* de forma inalámbrica sin tener que conectar el dispositivo por algún medio físico. Finalmente, culminadas estas tareas, la empresa al no contar con página web para ofrecer información y sus productos a los clientes, designó la elaboración desde cero basadas en la herramienta de *framework* de Flutter.

PALABRAS CLAVE: *Over The Air*, *frontend*, desarrollo de software, *framework*, *sketch*, Flutter.

1.2. Abstract

The area of Technology and innovation for software development is a department of the company CiNNoV S.A.S, which is responsible for analyzing, design and develop the internal systems required for its advancement. This paper describes how, through virtual business practice, the functions of junior engineer were carried out, providing support in the implementation of the Bigo device, participating in the research to execute the OTA system and for the creation of the CiNNoV S.A.S website, with this it was possible to acquire knowledge in the development of design UI of applications and the use of the ESP32 module, among others. Initially, the company offered study material and training that allowed knowing the tools and languages that were executed in the position, according to this, I was assigned the task of carrying out the data exchange between the module and the mobile application. , so that, it will show the battery percentage in real time of the Bigo device, in addition, I had participation in the investigation of the OTA technology to be implemented within the Bigo device and thus be able to update the sketch wirelessly without have to connect the device by some physical means. Finally, after completing these tasks, the company, not having a website to offer information and its products to customers, designated the development from scratch based on the Flutter framework tool.

KEY WORDS: Over The Air, frontend, software development, sketch, framework, Flutter.

1.3. Contexto y condiciones del entorno de desarrollo

Para poner en contexto el entorno de desarrollo que se trabajó cuando se inició en la empresa, se dividió en dos fases. La primera hará referencia del proceso de ingreso a la empresa y sus requerimientos y la segunda fase establecerá cómo fue el inicio de la pasantía desarrollando las tareas asignadas.

1.3.1. Ingreso a la empresa y conocimientos básicos.

El ingreso a la empresa fue mediante una convocatoria en donde se buscaban ingenieros que tuvieran conocimientos en el área de tecnología y desarrollo de dispositivos, para abarcar tareas de programación. En ese entonces me postulé como pasante ya que mis conocimientos como estudiante de Ingeniería Mecatrónica podrían dar solución al perfil que estaban buscando. Luego de pasar por una serie de preguntas en la empresa, solo quedaba esperar. Días después recibo una llamada donde me informan que he sido aceptado por mi perfil para el cargo de Ingeniero Junior en la empresa CiNNoV S.A.S por mis habilidades como el razonamiento, la lógica y capacidad de tomar decisiones de calidad ingenieril para tomar el puesto. Posterior a esto me reúno con el equipo para realizar una reunión donde me explicaron cómo sería el trabajo y las condiciones laborales. En la reunión me dicen que el trabajo será remotamente virtual, ya que, el cargo para el que aplicaba, se podía desempeñar con un computador y una conexión a internet. Entonces, me dieron la facilidad de trabajar desde casa, además, me enviaban el material necesario desde Bogotá para por llevar a cabo mis funciones como Ingeniero en el área de desarrollo desde casa.

1.3.2. Capacitaciones y adaptación de entorno para desarrollar las tareas.

Para empezar el trabajo se necesitaba contar con un computador que tuviera ciertas capacidades para correr estos programas, ya que, cualquier computador de uso básico no podría correrlo de forma estable y también, era necesario una conexión a internet para la descarga de los programas de uso libre. La empresa no nos proporcionaba estas herramientas, puesto que, se asumía que cada uno contaba con esto o podríamos buscarlas. Seguido de esto, se procedió a una investigación e instalación previa de los programas y lenguajes que se usarían en el transcurso de la pasantía. Principalmente, tuvimos que informarnos del uso de software como GIT y *Visual Studio Code* y capacitarnos para entender su entorno de trabajo y comandos. Dentro de *Visual Studio* manejamos la programación del módulo ESP32 en el lenguaje C++ para el dispositivo Bigo y la programación de Flutter para el desarrollo de la aplicación y la página web en lenguaje DART.

1.4. Producto de la pasantía

Gracias a la oportunidad que me brindó la empresa CiNNoV S.A.S de trabajar con ellos, me permitió aprender muchas cosas que contribuyeron a mi crecimiento personal, encontrar nuevas rutas y expandir mis conocimientos en el campo laboral de esta área que terminaron gustándome como el mundo de la programación y el desarrollo de software enfocado en Flutter. Mi pensado a futuro será seguir estudiando este *framework* y aprendiendo más de estos lenguajes para llegar a ser un gran programador y así, llegar a construir mi propia empresa de soluciones tecnológicas.

1.4.1. Vinculación a un entorno laboral

Dentro del proceso de mi práctica, tuve que trabajar en equipo con mis compañeros, codificar para una sola rama y eso llevó a darme cuenta, que una empresa no la lleva una sola persona, sino que todos trabajando en conjunto se pueden crear grandes cosas. Aprender, como funciona este mundo, cumplir con un horario o una meta que te motiva a realizar tu trabajo fueron los factores que me hicieron aprender, como puede ser mi futuro como Ingeniero.

1.4.2. Página WEB de la empresa

Una de las tareas más importantes me fue asignado, la cual consistía en el desarrollo de la página web de la empresa para tener su carta de presentación frente a potenciales clientes. De esta pasantía quedaría mi registro como el creador de la página desde su inicio, para que futuros trabajadores puedan seguir implementándola o agregándole contenido haciendo uso de este código.

1.5. Definición del problema

CiNNoV S.A.S está empezando su proceso como empresa y a raíz de esto nació la necesidad de ampliar su equipo de desarrollo para avanzar de manera más rápida en sus productos e investigaciones. Es por eso que, requería de personas que tuvieran un perfil creativo, autodidacta, innovador, de liderazgo e iniciativa para cubrir el cargo de Ingeniero Junior en el área de Desarrollo e Innovación. Todo esto con la finalidad de abarcar las problemáticas de mejorar el producto Bigo, tanto en programación como en el desarrollo de la aplicación Móvil para tener una mejor experiencia con el cliente y la creación de una página web donde se pudiera evidenciar la información de la empresa.

1.6. Contexto y condiciones del problema

1.6.1. Implementar la condición de batería del dispositivo Bigo en la aplicación móvil

La empresa tenía un código en donde ya había un intercambio de datos de la aplicación con dirección a el dispositivo en tiempo real, pero no existía en sentido contrario, es decir, del dispositivo en dirección a la aplicación, para lo cual, mi trabajo en ese momento era desarrollar ese intercambio de datos por medio de código y así, ver reflejado el porcentaje de la batería dentro de la aplicación.

1.6.2. Investigar actualización de un sketch de forma inalámbrica

Los dispositivos de Bigo hacen en grandes cantidades desde China y al llegar a Colombia es necesario programarlo uno por uno. Este era un proceso tedioso y demorado, ya que, tocaba soldarle unos cables para poder programarlo desde un computador personal (PC), transferirle el código y luego volver a desoldarlo, así sucesivamente con todos los dispositivos. La empresa buscaba mejorar este proceso y, que mejor manera de hacerlo que de forma inalámbrica, es por eso que, en el transcurso de la investigación, me di cuenta que existía algo llamado transferencia por aire que más adelante les explicaré a detalle.

1.6.3. Desarrollo de una página web informativa

La empresa tenía una página ya creada, pero solo del dispositivo Bigo que es su producto más fuerte en el mercado, como tal, la empresa CiNNoV no tenía una carta de presentación frente a sus clientes en la internet diferente de sus redes sociales. Es por eso que me fue asignada la tarea realizar la página desde cero con el *framework* de flutter en el lenguaje de DART, ya que, este Kit de Desarrollo de Software (SDK) de Google estaba tomando mucha fuerza en los últimos años para el desarrollo de páginas y aplicaciones móviles. Aparte de eso, implementé la condición de que la página fuera *Responsive*, es decir, que sea agradable para cualquier tamaño de pantalla y dispositivo, ya sea escritorio o teléfono.

1.7. Objetivos

1.7.1. Objetivo general

- Proyectar incorporación de mejoras en aspectos relacionados a integración, conexión y operación por medio de la aplicación con tecnologías del Internet de las cosas (IoT).

1.7.2. Objetivos específicos

- Identificar las variables y parámetros del proyecto relacionados con conectividad, integración y comunicación del dispositivo.
- Proyectar estrategia para mejorar tecnológicamente el dispositivo en aspectos de conectividad, integración y operación a través de la aplicación.
- Ejecutar plan proyectado para desarrollar y aplicar las mejoras tecnológicas referentes a integración, comunicación y programación bajo el conocimiento adquirido en función de las necesidades de la empresa.
- Realizar pruebas detalladamente del trabajo realizado en funcionamiento para verificar su cumplimiento con los resultados esperados.
- Elaboración un manual de funcionamiento de la aplicación con todas sus nuevas implementaciones.

2. Estructura del documento

2.1. Capítulo I, Introducción

En este capítulo se aborda de manera general una conversación con el lector de acuerdo al planteamiento de la propuesta donde de manera resumida se pondrá en contexto con el trabajo realizado en la práctica, la temática, el por qué se desarrollará y el nivel de complejidad que podrá tomar realizarlo para generar un buen resultado de calidad ingenieril.

2.2. Capítulo II, Estado del Arte

En este capítulo se aborda de manera general el propósito de transmitir al lector, de manera resumida y evaluativa, el estado actual del tema y los conocimientos de proyectos previos que tienen similitud al trabajo de grado realizado.

2.3. Capítulo III, Metodología y plan de trabajo

En este capítulo se aborda de manera general todo el plan desarrollado en la empresa por medio de un esquema metodológico que representa la organización secuencial de los objetivos propuestos que ayuda a comprender el paso y las actividades ejecutadas en la empresa.

2.4. Capítulo IV, Requerimientos de diseño

En este capítulo se aborda de manera general los requerimientos y parámetros de diseño que se planearon para la realización de la página web, los temas a tener en cuenta y la estructura visual que se requiere, para que, la experiencia del usuario sea agradable cuando se esté navegando en esta.

2.5. Capítulo V, Modelo de implementación

En este capítulo se aborda de manera general cómo fue realizada la programación de la página web y los casos más importantes que se tuvieron en cuenta para la construcción y estructuración del código.

3. Estado del arte y Marco teórico

En la actualidad existen distintos proyectos que están enfocados en el desarrollo de software del *framework* de Flutter que han servido de mucha ayuda para guiarme a la hora de realizar este libro. Gracias a esto, se realizó una investigación por varios repositorios encontrados en la web, sirviendo como apoyo para la investigación sobre el tema propuesto.

3.1. Aplicación móvil informativa de apoyo académico de la Universidad Politécnica Salesiana

En el año 2021, los ingenieros de sistemas realizan su trabajo de titulación previo a la obtención del título que lleva por nombre “Aplicación móvil informativa de apoyo académico de la Universidad Politécnica Salesiana” [1]. Estos estudiantes en ese entonces, hicieron una encuesta en la página institucional de su Universidad para poder escuchar a sus compañeros y ver las necesidades para empezar a desarrollar la aplicación. Los estudiantes fueron escuchados y el resultado de la encuesta fue un levantamiento de necesidades como: información de pagos o deudas, información del estudiante, carnet virtual, horarios, noticias, eventos, estado de solicitudes y un chat interno.

Para el diseño de la aplicación hicieron un levantamiento de requerimientos para aclarar ciertas funcionalidades, sistemas y atributos, ya que, su finalidad era ser entregada para la comunidad estudiantil de la universidad, priorizando sus necesidades. Luego de analizar toda la información, definieron una lista de requisitos funcionales que estarían dentro de la aplicación:

- Inicio de sesión por medio del correo institucional.
- Mensajería.
- Generación de grupos automáticos según las materias matriculadas.
- Notificaciones.
- Comunicados.
- Noticias y eventos.
- Información de la Universidad.
- Solicitudes.
- Notas estudiantiles.
- Récord académico de estudiantes.
- Estado financiero.
- Horario estudiantil.
- Carnet digital por medio de código QR.

Después de tener estas necesidades claras, procedieron analizar las arquitecturas del sistema, como se ilustra en la siguiente figura.

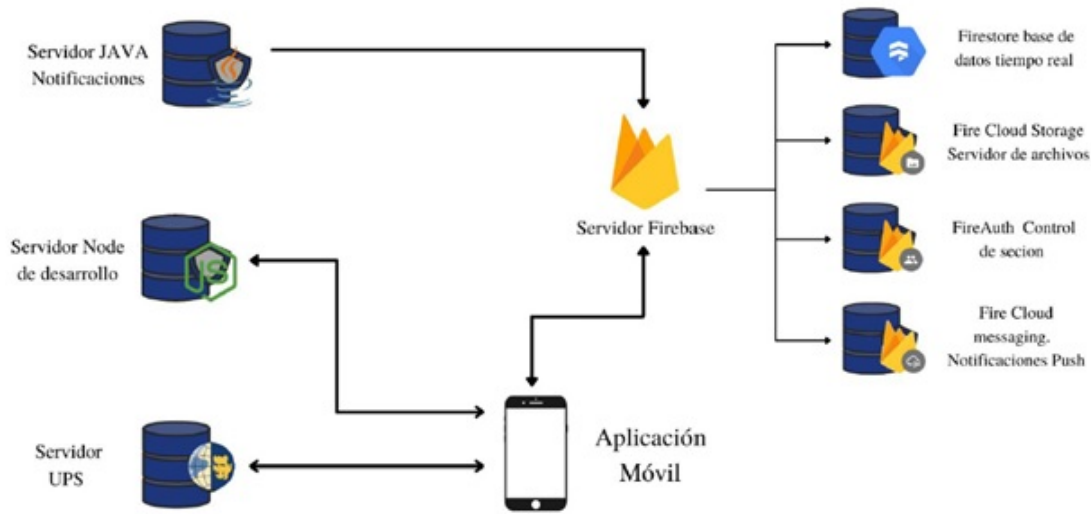


Figura 1: Arquitectura del sistema

Fuente: [1]

Como se puede apreciar en la Figura 1, los ingenieros necesitaron el uso interno de los servidores de la Universidad Politécnica Salesiana (UPS) y también servidores externos como los es *Firebase* [2], así también, algo de ayuda del personal de la universidad que hacía parte del equipo de *BackEnd* [3] para poder brindar la información institucional.

Haciendo análisis a los servidores de la UPS, el cual, aloja todos los servicios web que conforman la información interinstitucional y los datos de los estudiantes, definieron la siguiente información estructurada que necesitarían para mostrar en la aplicación:

1. Portal: Este servicio les ofrecía información sobre las noticias y eventos de la universidad.
2. SIGAC: Este servicio ofrecía información económica sobre el estado financiero del estudiante con la institución.
3. SOL: Servicio web que les permitía obtener toda la información sobre solicitudes de la universidad.
4. PED: Servicio que ofrecía información académica del estudiante.
5. ORG: Este servicio ofrecía información organizacional de la universidad.
6. INS: Servicio de información personal del estudiante.
7. GTH: Servicio de información de personal colaborativo como docentes, administrativos, etc.

El uso del servidor Representational State Transfer (REST) con Node.JS también fueron de gran ayuda, debido a que, les devolvía información estática de datos de la universidad y también

información de usuarios predefinidos. Para poder testear el almacenamiento y las notificaciones en la base de datos de una nube, crearon un servidor de Java con la arquitectura *Java Enterprise Edition* (JEE). Por último, hicieron uso de los servicios de *Firebase* aprovechando todos los beneficios que brinda.

En la creación de la aplicación se basaron en el *framework* de flutter por sus grandes ventajas y características de multiplataforma (iOS, Android y WEB) para su ejecución. Dentro del proyecto hicieron uso de las denominadas librerías externas para mejorar la experiencia del programador y así, evitar extender el código usando métodos que ya están implementados dentro de la comunidad de flutter. Entre estos se destacaron las siguientes librerías de las que se hicieron uso:

- Utilidades.
- *Date Format*.
- UI.
- *Firebase*.
- Peticiones *Hypertext Transfer Protocol (HTTP)*.
- Almacenamiento Interno.
- Verificación de estado de red.
- Manejadores de estado .

Eligieron como sistema de gestión de estado el patrón de Flutter BloC gracias a las ventajas que ofrece en su estructura y manejo de estados, como se ve representado en la siguiente figura:

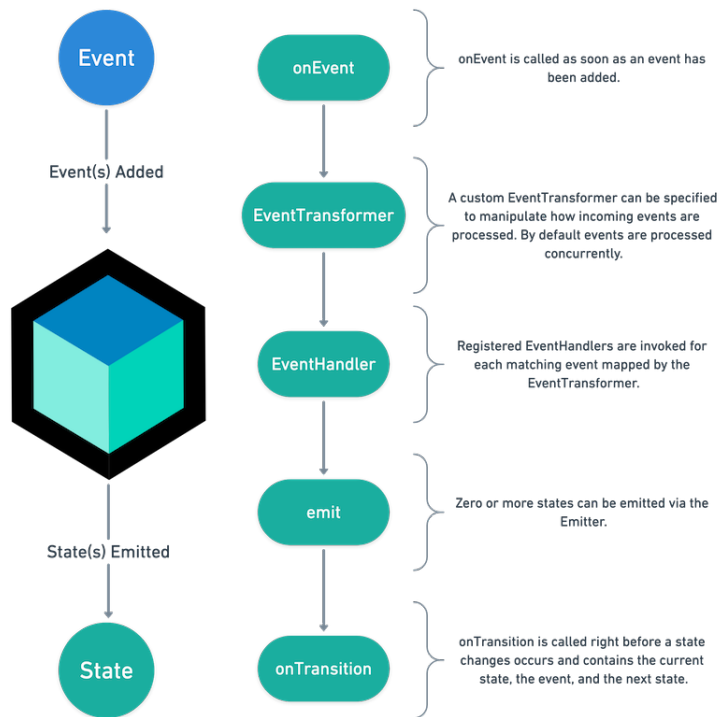


Figura 2: Funcionamiento de manejo de estado con BloC

Fuente: [4]

Finalmente, después de planear, estructurar y definir objetivos para la realización, procedieron con la codificación de la aplicación bajo el manejador de estado Bloc, como podemos apreciar en la Figura 2. El primero paso que hicieron fue producir una especie de diseño denominado “*mockup*” para tener una principal idea en su interfaz y de las animaciones que se fueran ejecutando dentro de la aplicación, como vemos en la Figura 3.

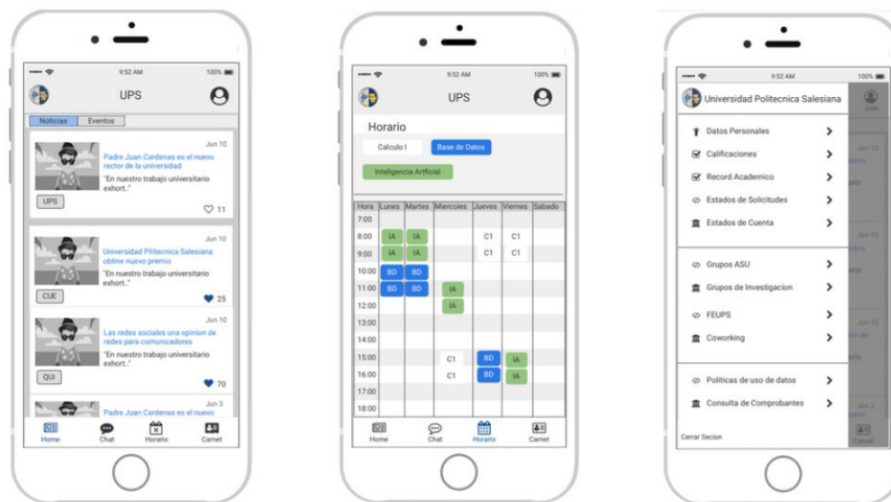


Figura 3: -UI- Mockup de la Aplicación.

Fuente: [1]

Seguido de esto, empezaron a realizar la programación en flutter guiándose del diseño que tenían previamente, pero a medida que avanzaba la aplicación tuvieron que utilizar los colores de la universidad y logos y se iba cambiando un poco al diseño original que se tenía de guía.

En la creación de la base de datos se implementó un almacenamiento remoto en *Firebase*, dado que, este servicio le permitía almacenar los datos en la nube y luego ser solicitados dentro de la aplicación según su necesidad en un archivo *JavaScript Object Notation (JSON)*. Esto y muchos más servicios se usaron para la solicitud de datos y poder manipular todo en tiempo real desde la aplicación, tales como: chat, grupos, información, interacción y usuarios.

Como resultado, obtuvieron una aplicación muy interesante y completa para la universidad que cubría los siguientes módulos:

1. **INFORMACIÓN INSTITUCIONAL:** La finalidad es exponer al usuario información de carácter organizacional.
2. **PORTAL DE NOTICIAS Y EVENTOS:** La finalidad es mantener informados a los estudiantes sobre noticias y eventos que son publicados en su página web oficial.
3. **CARNET DIGITAL:** La finalidad era mostrar un carnet digital en código QR para poder ser validado en actividades dentro de la institución.
4. **INFORMACIÓN INSTITUCIONAL ACADÉMICA:** La finalidad de este apartado era mostrar toda la información privada del estudiante.
5. **COMUNICADOS:** Este servicio es muy importante para la aplicación, ya que provee toda clase de notificaciones para el usuario.
6. **CHAT INSTITUCIONAL:** La finalidad de este servicio, era poder tener contacto con estudiantes y docentes de la institución.

Finalmente, lograron su objetivo principal, el cual era crear una aplicación móvil informativa de apoyo académico para su universidad, la cual, les otorgó el título de ingenieros en sistemas. A continuación, se anexan imágenes de la aplicación ya terminada en la Figura 4 y la Figura 5.

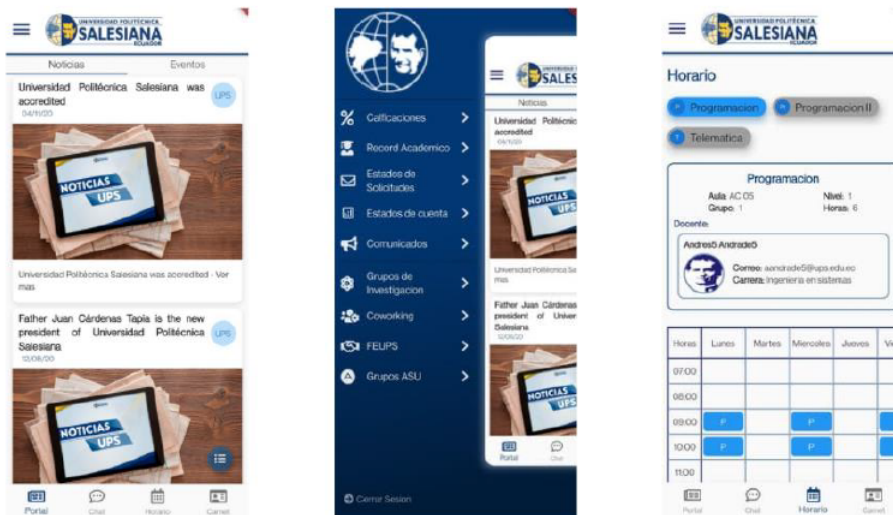


Figura 4: -UI- Pantallas finales de la Aplicación
Fuente: [1]

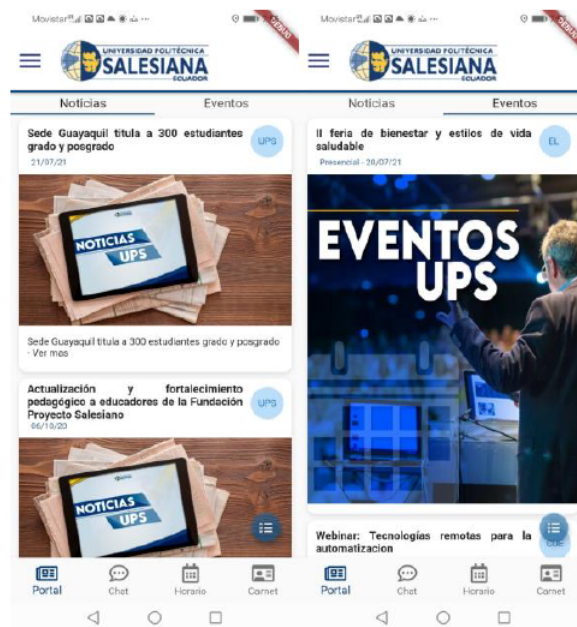


Figura 5: -UI- Tabs bar, Noticias y Eventos
Fuente: [1]

3.2. Implementación de una aplicación WEB con servicio de chatbot con inteligencia artificial que permita la autogestión de cuentas por pagar de los proveedores de la Universidad Autónoma de Bucaramanga

En el año 2020, un ingeniero realizó su trabajo en la modalidad de profundización para optar por el título de Magíster en Gestión, Aplicación y Desarrollo de Software, llamado “Implementación de una aplicación WEB con servicio de chatbot con inteligencia artificial que permita la autogestión de cuentas por pagar de los proveedores de la Universidad Autónoma de Bucaramanga” [5]. Con la finalidad de optimizar los procesos a la hora de atender a los proveedores mediante un Chat Bot que permitiera realizar consultas de sus estados financieros para reducir esfuerzo y tiempo por parte del personal de la empresa. Para realizar este proyecto se basó en etapas que vamos a mostrar a continuación:

1. INFORMACIÓN Y REQUERIMIENTOS

Inicialmente, se realizó una recolección de datos en la web y bibliografías sobre las tecnologías que incluían un chat automatizado en aplicaciones, ejemplos funcionales de la actualidad y su proceso de construcción para llevar a cabo su objetivo. Así mismo, información sobre en qué plataforma podría desarrollar la aplicación web y concluyeron que por sus ventajas la desarrollaría en el *framework* de Flutter. Del mismo modo, fue necesario una investigación sobre repositorios y proyectos de aplicaciones creadas en Flutter para su construcción.

2. ANÁLISIS DE TECNOLOGÍA CHAT BOT

Realizó una comparativa entre las tecnologías actuales que se encontraban disponibles para su uso, haciendo análisis de sus características, buscando cuál de ellas podría cubrir todos los requerimientos que exigía el proyecto. Dentro de la información recopilada, construyó una lista de características evaluadas de cada tecnología:

- a) LICENCIA: Para uso y costos asociados.
- b) LENGUAJE/INTEGRACIÓN: Lenguaje o método de integración con otras aplicaciones y servidores externos.
- c) IDIOMA: Idioma soportado.
- d) CANALES: Canales o vías en los que se puede implementar y poder interactuar con las personas.
- e) CLIENTES: A quién va dirigido.

3. SELECCIÓN DE TECNOLOGÍA CHAT BOT

Después de un análisis minucioso y tratando de indagar las mejores opciones, se concluyó que se usaría la tecnología de *Dialogflow* por su mayor compatibilidad con el *framework* de flutter y menores costos asociados. Para usarlo, solo basta con registrarse en *Google Cloud Platform*. Esta tecnología fue desarrollada por Google y está disponible para cualquier usuario que tenga un correo asociado a su empresa. Tiene un gran motor de inteligencia artificial y procesamiento que permite tener un flujo de conversación con muy pocos pasos como se puede apreciar en la Figura 6 y también posee un panel administrativo donde se puede configurar de muchas formas permitiendo un servicio confiable y una buena seguridad en la aplicación. La

plataforma *Dialogflow* ofrece un servicio de prueba en su versión más reciente que te permite hacer testeos mediante un Localizador de Recursos Uniforme (URL) con intercambio de datos por POST pero también te ofrece una licencia *Enterprise* que te cobra por interacción hasta tener un cupo gratuito de 300 usd por noventa días [6].

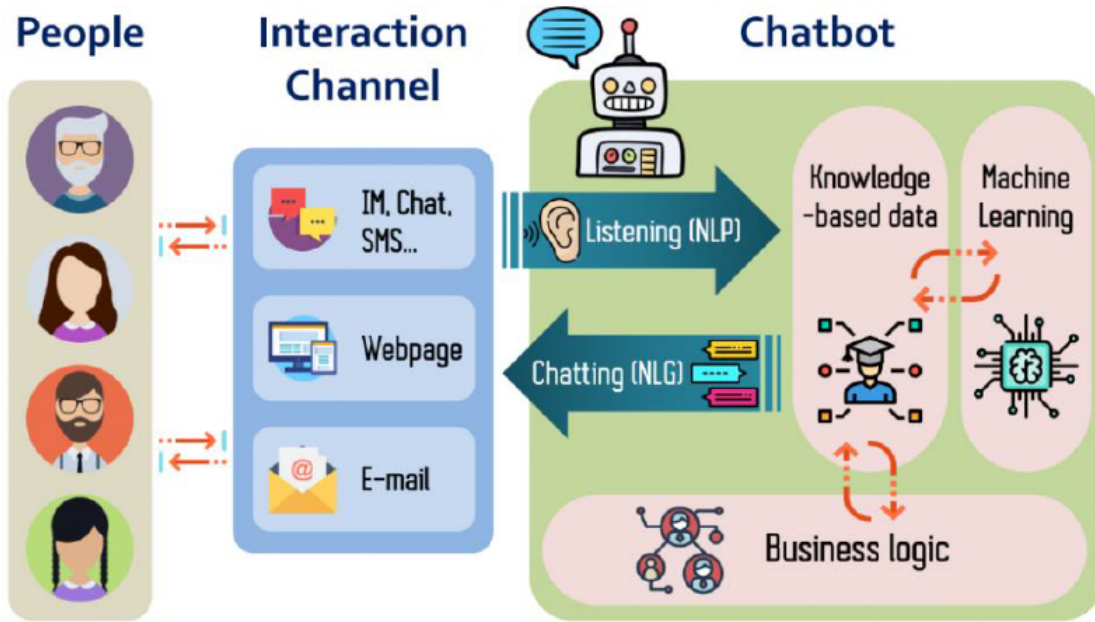


Figura 6: Anatomía de un ChatBot

Fuente: [5]

4. DESARROLLO DE APLICACIÓN

Para el desarrollo de la aplicación web se tuvo en cuenta un orden para poder unir todas las piezas al final y tener un buen resultado.

- a) **FRAMEWORK DE FLUTTER:** El uso de esta herramienta fue todo un reto para el autor, debida a que, este SDK apenas se encontraba en sus inicios lo cual no fue un trabajo sencillo de implementar el Chat Bot dentro de esta aplicación. Todo el diseño, interfaz, despliegue de menús, ventana de chat fue construida en el lenguaje Dart de Google.
- b) **APLICACIÓN PHP PARA *DIALOGFLOW*:** En el proceso de creación de la aplicación fue necesario crear un script en *Hypertext Preprocessor (PHP)* que accede por medio de credenciales a una cuenta configurada con *Google Cloud Platform* para el consumo de *Dialogflow* de modo que pudiera ser solicitado desde la aplicación de Flutter mediante HTTP y poder ser transmitido en formato JSON al chat bot [7]. Igualmente, se construyen paginas PHP para que estas pudieran procesar peticiones y realizar consultas en la base de datos de modo que no quedaran alojadas en la aplicación web ofreciendo más seguridad a la información.

5. **CHAT BOT *DIALOGFLOW*:** Para poder llevar a cabo el chatbot en *Dialogflow* se realizó una guía de pasos, empezando por registrar la cuenta en *Google Cloud Platform*, hasta realizar

todo el proceso de creación. Seguido de esto, la configuración del proyecto y las cuentas de acceso para el uso de la aplicación, como observamos en la Figura 7.

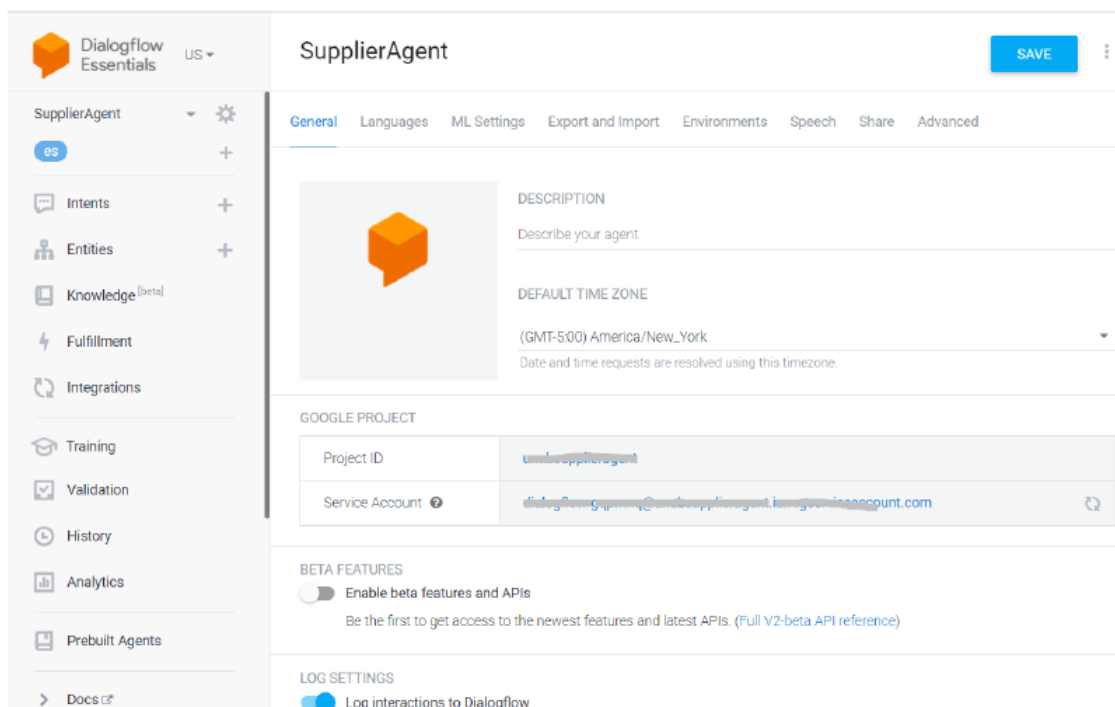


Figura 7: Creación del proyecto en *Dialogflow*
Fuente: [5]

Para poder compilar el chatbot, se debía realizar un análisis de datos para crear los temas de conversación y así, poder extraer y guardar los sinónimos asociados para dichos temas. Las principales entidades que se crearon para el chatbot fue: Factura, Servicio e Historial. Ya definidos estas entidades, se cerraron los *intents* en el agente que asignan las entradas por teclado del usuario a determinadas respuestas. Cuantos más ejemplos se le agreguen al *intent* tendrá más probabilidad de reconocer el servicio y dar un buen resultado. Para finalizar, se personalizó la interfaz de usuario que interactúa con el chatbot mediante la *Application Programming Interfaces (API)* de *JavaScript* en *Dialogflow*.

Durante todo el proceso de la realización del proyecto, implementaron la metodología Kanban para tener un orden de las actividades que se debían desarrollar, dando por terminada cada vez que era ejecutada [8]. Este método le permitió llevar a cabo un orden para poder ver el camino que iba recorriendo a medida que avanzaba y no sufriera retraso en alguna actividad que se haya quedado sin desarrollar, como se puede ver representado en la Figura 8.

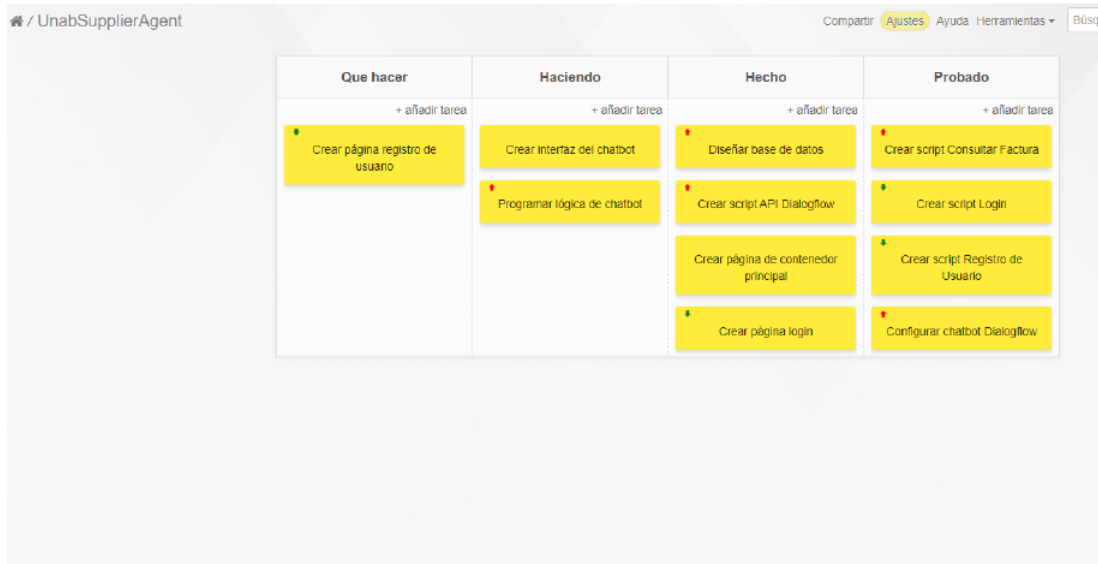


Figura 8: Tablero Kanban

Fuente: [5]

6. PRUEBAS: En la fase final, realizó un documento de pruebas que se llevaron a cabo por personas que no hacían parte del proyecto seleccionadas al azar para que le pudieran dar la opinión. Estas pruebas permitieron que pusieran en evidencia algunos errores y bugs como primera versión de la aplicación que luego serían corregidos de manera rápida. A continuación, se presenta los resultados de la aplicación hecha en Flutter vista desde navegador en la Figura 9.



Figura 9: Ventana principal de la aplicación web en Flutter

Fuente: [5]

Este proyecto le ofrecía al usuario la oportunidad de consultar información privada de sus

pagos mediante una conversación con el chat bot muy natural, de tal manera que, le permite mostrarle los datos solicitados en un tiempo promedio de cinco minutos como se puede apreciar en la pestaña de la Figura 10.

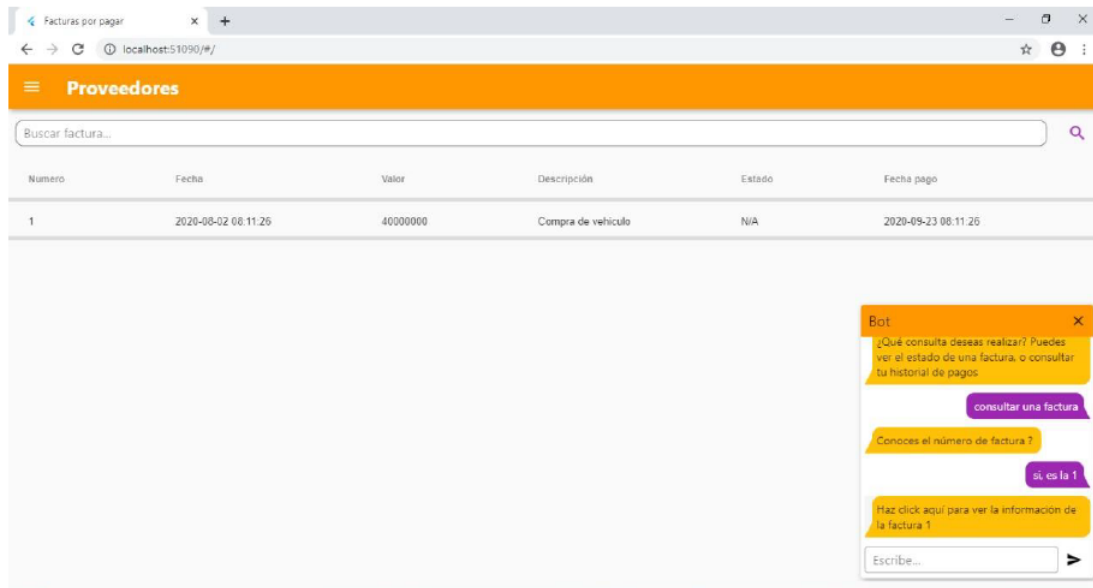


Figura 10: Resultado de consulta de facturas a través del chatbot
Fuente: [5]

Los resultados de la consulta financiera también se podían visualizar gráficamente y resumido en un *dashboard* con un diseño agradable como podemos apreciar en la Figura 11.

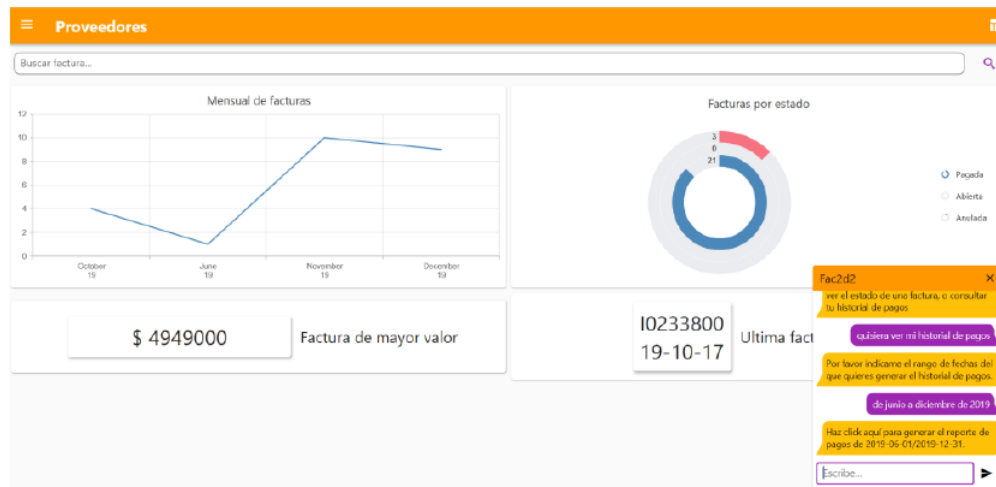


Figura 11: Resultado de consulta de historial de pagos en *dashboard*
Fuente: [5]

Al igual, la aplicación cuenta con su formulario de registro que solicita los datos básicos (Nombre, Email, Clave, Documento) que permite al usuario crear sus credenciales de acceso para poder consultar sus datos dentro de la aplicación.

4. Metodología y plan de trabajo

4.1. Esquema metodológico

Todo objetivo tiene siempre una previa planeación, la cual, ayuda a estructurar el proceso por el cual se quiere llevar a cabo una finalidad, es por eso que realizar un esquema nos ayuda a guiarnos en un recorrido más organizado para poder visualizar el camino de la meta a la que se quiere llegar.

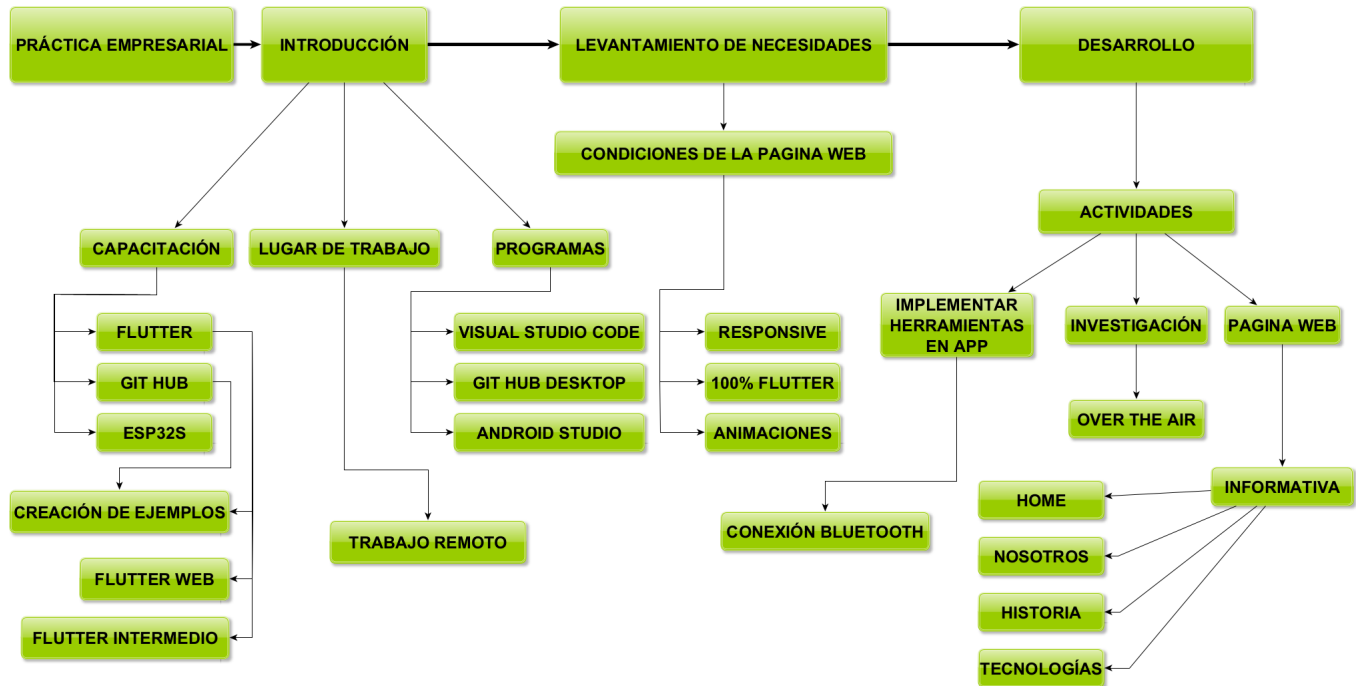


Figura 12: Esquema metodológico.

Fuente: Autor

4.2. Descripción de esquema metodológico

Como se puede apreciar en la Figura 12, el plan de trabajo se resumió mediante un diagrama teniendo en cuenta las distintas tareas y fases por las cuales se fueron realizando para poder concluir los proyectos que se tenían planeados dentro de la empresa. Uno a uno de estos ítems fueron organizados de tal manera que se pueda representar de forma estructurada el paso a paso y las tareas.

En la INTRODUCCIÓN se realizaron investigaciones, capacitaciones, adaptación de espacio de trabajo e instalación de programas para poder dar inicio a la práctica y comenzar con las tareas asignadas.

Para las CAPACITACIONES, fue algo nuevo y de manera rápida, ya que, tuvimos un plazo máximo de dos a tres semanas para aprender todo sobre los lenguajes que requerían y manejo de programas para empezar a trabajar, entre los cuales se encuentran los siguientes:

- **MÓDULO ESP32:** El dispositivo de Bigo estaba programado bajo este módulo que nos ofrecía muchas ventajas, una de ellas era el manejo de *Bluetooth* 4.2 que permitía la conexión con la

aplicación que fue diseñada y el intercambio de datos, ideal para poder desarrollar productos de IoT. Todo esto lo ofrecía el módulo en tiempo real para tener una experiencia de usuario muy cómoda según su programación [9].

- GIT HUB: Herramienta que servía para trabajar los códigos en grupo y tener un mejor control de las versiones a las cuales se les iba haciendo modificaciones y subiendo en la nube. [10]
- FLUTTER: Este aprendizaje fue uno de los más largos, ya que, todos los días, se aprendía algo nuevo. El kit de herramientas de UI de Google nos permitía realizar hermosas aplicaciones para web y móvil compiladas nativamente desde una única base de código. Bajo este *framework* se basó la mayoría de la pasantía, la cual, hizo parte de la construcción de la Aplicación móvil y la página web. Para poder profundizarlo muy bien me basé en cursos que ofrecía la plataforma de Udemy [11] bajo la dirección del ingeniero de software Fernando Herrera. Estos cursos que llevan por nombre “Flutter Web: Aplicaciones y páginas web profesionales” con un intensivo de 20.5 horas de video y “Flutter Intermedio: Diseños profesionales y animaciones” con un intensivo de 15.5 horas, fueron los que contribuyeron en mi aprendizaje y experiencia por medio de clases y ejemplos para poder realizar los trabajos requeridos por la empresa [12] .

El trabajo fue netamente REMOTO, es decir, todo se trabajó desde casa, donde nos enviaban el material y nosotros programábamos bajo los dispositivos que nos fueron suministrados para poder hacer pruebas [13].

En la instalación de PROGRAMAS, se requerían unas aplicaciones básicas que permitían realizar las actividades de manera exitosa. A continuación, mencionaré las más importantes:

- VISUAL STUDIO CODE: Fue uno de los programas más importantes dado que, desde este editor se podía realizar la codificación tanto en C++ [14] para el ESP32 y también en Dart para el *framework* de Flutter.
- GIT HUB DESKTOP: Esta extensión era instalada con la finalidad de poder conectar un editor y tener control de las versiones de código con la nube [15].
- ANDROID STUDIO: Era un requisito para poder correr Flutter en *Visual Studio Code*, ya que, podía realizar la simulación de dispositivos móviles para ver el código en tiempo real como se iba comportando. De igual forma también permitía simular desde un dispositivo móvil físico conectado por cable al computador, esto con la finalidad de no consumir tantos recursos del computador, debido a que, era muy pesado simularlo [16].

En el LEVANTAMIENTO DE NECESIDADES se condicionó de manera general los requerimientos que llevaría la PÁGINA WEB de la empresa. Principalmente, se sugirió que se desarrollara la versión de Escritorio para los visitantes desde una computadora, ya en un futuro se podría implementar que fuera *responsive*, es decir, que se adapte a cualquier tamaño de pantalla, ya sea Escritorio, Tablet o móvil [17]. Luego de un análisis de diferentes lenguajes, se tomó la decisión de que se haría cien por ciento en lenguaje de dart en el *framework* de flutter por las ventajas que ofrecía y la experiencia que ya se había adquirido en las capacitaciones [18]. Para finalizar, sugirieron que la página tuviera animaciones que fueran agradables con los lectores.

En el DESARROLLO de las actividades realizadas dentro de la empresa no solo se trabajó sobre la página web, también se hizo parte de la investigación en la que se quería realizar de forma

inalámbrica la actualización de un *Sketch* en un dispositivo. Esto es basado en una tecnología denominada OTA (*Over the Air*) que significa “Sobre el aire” [19] y también se hizo parte parte de la implementación del protocolo de comunicación *Bluetooth* entre el dispositivo Bigo y la aplicación móvil Bi Go para que realizara el intercambio de datos y poder ser visualizados de una u otra manera. Para la página WEB, nos reunimos con el equipo de diseño de la empresa y así se logró dar las pautas y finiquitar los ítems que llevaría esta. De manera resumida contendría un apartado de INICIO, donde se dará una breve introducción de la finalidad de la empresa. Un apartado de NOSOTROS, donde se hablará del equipo de trabajo y quienes somos. Un apartado de HISTORIA, donde se hablará los sucesos importantes de la empresa al pasar el tiempo. Por último, un apartado de TECNOLOGÍA donde se ofrecen los productos de la empresa a un público [20].

4.3. Recursos necesarios

1. Computador con licencia de Windows 11.
2. Conexión a internet 24/7.
3. Energía eléctrica.
4. Mesa de escritorio.
5. Silla de escritorio.
6. Dispositivo físico de Bigo (Pruebas).
7. Módulo ESP32 (Pruebas).
8. Teléfono Móvil (Simulaciones).

5. Requerimientos de diseño

Es importante tener clara la estructuración de una página WEB, para así, planear en debida forma el diseño para poder dibujar el camino y las rutas a seguir en el proceso de construcción de la forma más coordinada posible.

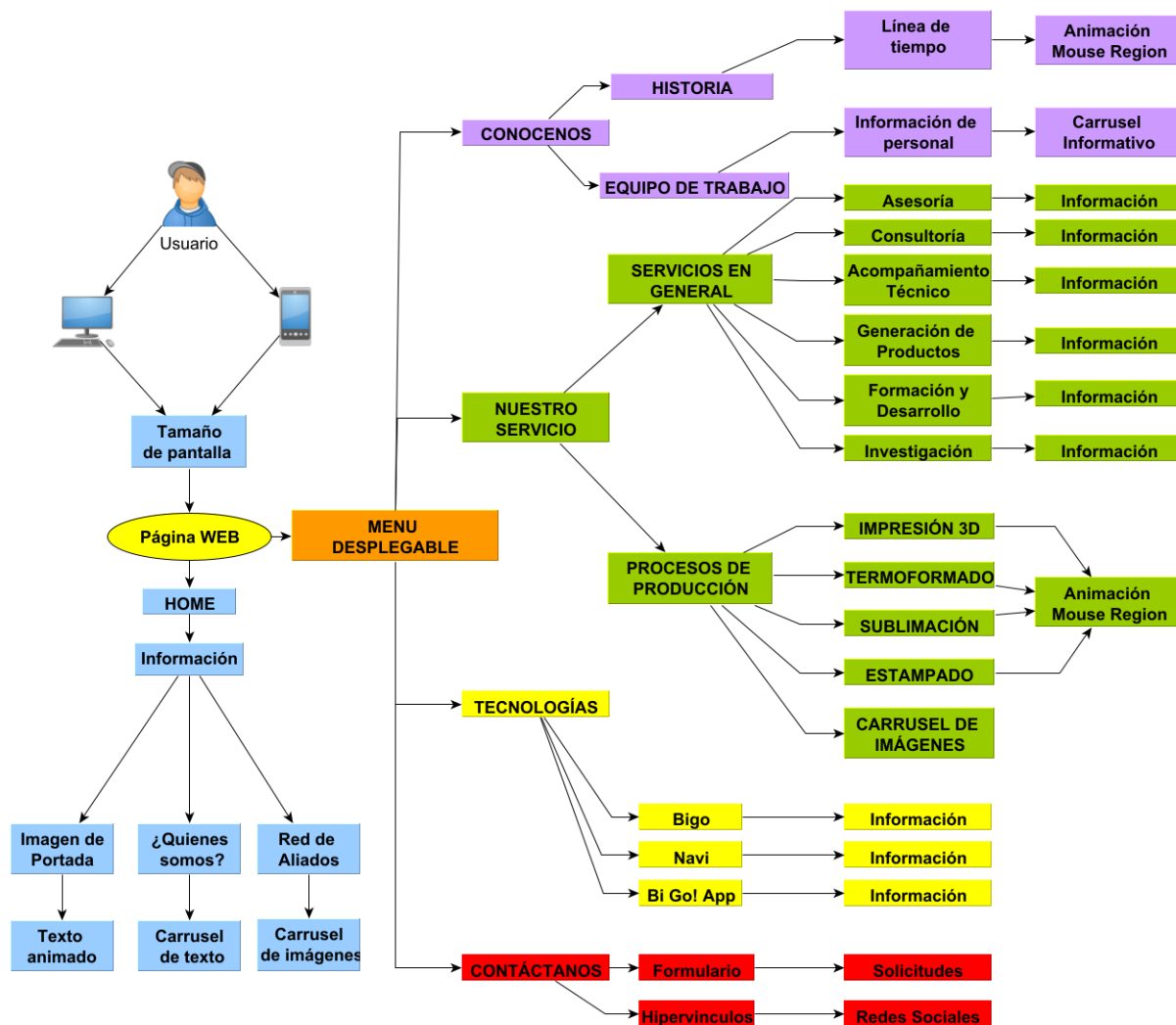


Figura 13: Esquema de modelado funcional

Fuente: Autor

La estructura interna y codificación que conforma la página web, se realizó de una forma organizada, como se aprecia en la Figura 13, con la finalidad de que los próximos ingenieros que quieran realizar implementaciones dentro de la página web, encuentren un orden por el cual, les va a ser más fácil identificar los caminos de las rutas. A continuación, se explicará de manera resumida como está conformada su respectiva estructura y animaciones.

Para comenzar, partiremos de un USUARIO cualquiera queriendo acceder a la página WEB alojada en el dominio www.cinnov.co, ya sea desde un dispositivo móvil o un computador de escritorio. El ancho de la página en pixeles definirá como mostrar el contenido visual para que siempre sea

agradable con el lector. Esto se realiza partiendo de un principio básico como lo es un condicional dentro del código. Dependiendo de que tan grande sea el ancho de la pantalla, automáticamente verá en formato escritorio o móvil, a esto lo denominamos hacer la página *responsive*.

5.1. Inicio

La primera impresión al ingresar a la página web será ver una página de inicio acompañado de un MENÚ, que le dará al lector diferente INFORMACIÓN de primera mano sobre la empresa. Esta información fue dividida por segmentos dentro del código, donde cada función es llamada hacia el inicio y se van apilando en forma de columna para ir mostrando el contenido. En el apartado de INICIO, tendremos información dividida en tres partes, para poder explicar cada una cómo funciona.

- IMAGEN DE PORTADA: Esta sería la primera impresión que tendría el lector al ingresar a la página desde algún lugar externo. Con un texto muy llamativo que entraría con una animación de izquierda a derecha como lo indica la flecha.



Figura 14: Página de bienvenida

Fuente: Adaptación [20]

La animación que indica la Figura 14 representada con una flecha negra, se realizó con ayuda del paquete existente del ingeniero Fernando Herrera llamado *Animate do* en la versión 2.1.0, donde simplemente el *Widget* que queramos animar para que realice una entrada es envuelto entre el *Widget* del paquete y se le asigna la animación, en este caso se hizo uso de *FadeInRight* con una duración de 1000 milisegundos que será el tiempo en el que se ejecute la animación.

- ¿QUIÉNES SOMOS?: Este *Widget* envolvía información que resumía el perfil de la empresa, pero también contaba con una animación denominada “Carrusel” donde un texto iba cambiando cada cierto tiempo para darle más significado al texto, como lo veremos a continuación:



Figura 15: Página informativa, ¿Quiénes somos?

Fuente: Adaptación [20]

Como se puede observar en la Figura 15, el texto iba cambiando de INNOVADORES a CREATIVOS y luego a INVESTIGADORES, el cual, funcionaba con un paquete externo de nombre *CarouselSlider* en la versión 4.1.1. Para que este efecto se viera de manera natural en el código había que inicializar un listado que contuviera estos 3 textos, y luego hacer uso del *Widget CarouselSlider*. Este contenía varias opciones que eran opcionales de elegir para ver la animación, en este caso, se configuró que fuera de manera automática, tuviera una duración de 4 segundos entre intervalos y que se viera únicamente la fracción de texto y fuera deslizándose.

- **RED DE ALIADOS:** En este apartado se utilizó el mismo paquete anterior para realizar un carrusel de imágenes para que fueran pasando en orden y que el usuario pudiera ver una agradable animación que también era controlada por unos botones que se anexaron con un paquete llamado *smooth page indicator* en la versión 1.0.0+2. Estos botones se controlaban pasándole el contexto de las imágenes y la posición en la que se encontraba, con esto se podía controlar de manera que fuera manipulando también las imágenes y los botones al tiempo.



Figura 16: Página informativa, Aliados
Fuente: Adaptación [20]

La anterior Figura 16, es una parte de lo que conforma el inicio y sus respectivas animaciones, de aquí saltamos a las rutas de los otros apartados, pero, ¿cómo hacíamos para navegar?, sencillo, con un menú como se puede observar en la siguiente imagen:

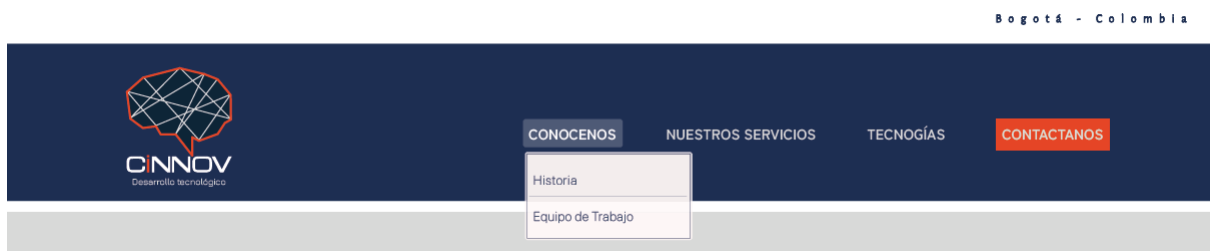


Figura 17: Menú de la página
Fuente: Adaptación [20]

Como se aprecia en la Figura 17, podíamos navegar de pestaña en pestaña, facilitando subsecciones entre ítem gracias a un *Widget* de Flutter llamado *MouseRegion*, el cual detecta y realiza alguna acción cuando dentro de su región se está pasando el puntero del *Mouse*, en este caso la acción que realizaba era desplegar el menú como un submenú.

5.2. Conócenos

El siguiente apartado se divide en dos partes, a continuación, explicaremos su contenido y estructura:

- **HISTORIA:** Esta parte de la página era muy importante, ya que, redactaba los hechos importantes desde que se inició la empresa. Llevaba una particularidad y era que contenía una línea de tiempo que mediante la acción de un *MouseRegion* iba mostrando un cuadro con un suceso en dicho año como se muestra en la Figura 18.

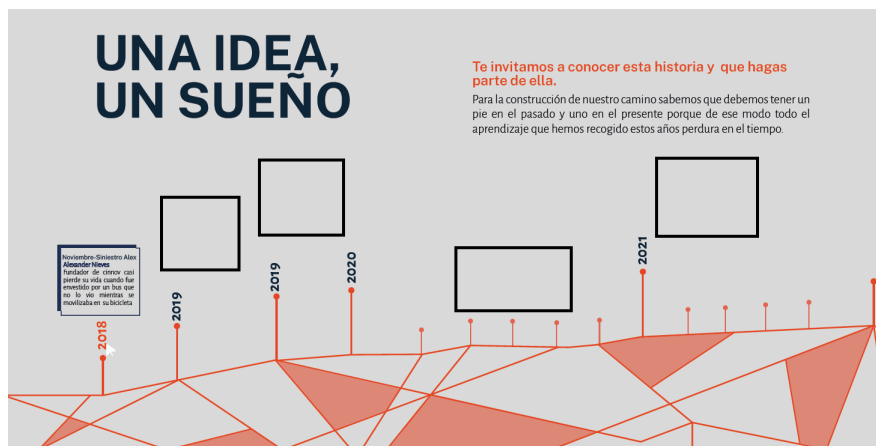


Figura 18: Línea de tiempo
Fuente: Adaptación [20]

- **EQUIPO DE TRABAJO:** Este apartado es netamente informativo y muy importante también. Su finalidad es mostrar todo el equipo que conforma la empresa clasificada por carreras. También contenía una animación tipo carrusel donde se iban pasando hacia la derecha controlados con unas flechas para ir en cualquier dirección, ya sea derecha o izquierda como se parecía en la Figura 19.



Figura 19: Equipo de trabajo
Fuente: Adaptación [20]

5.3. Nuestros servicios

La siguiente página contenía toda la información sobre los SERVICIOS que ofrecía la empresa en áreas de trabajo específicas o generales. De primera impresión tenía un resumen detallando cada ítem, pero al seleccionar el botón de “Quiero saber más”, era llevado a otra página donde daba información específica de ese apartado como se aprecia en la Figura 20.

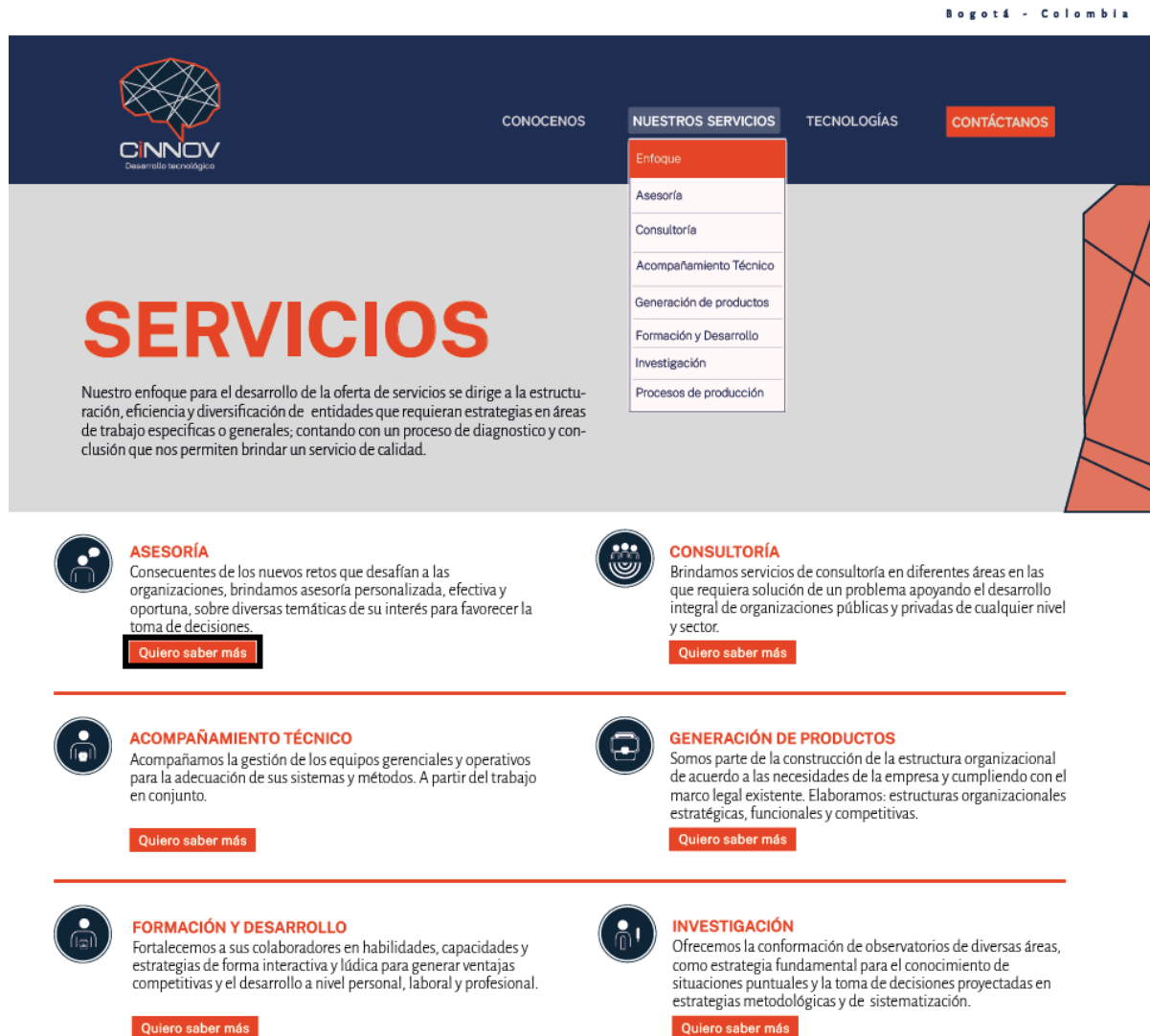


Figura 20: Servicios ofrecidos por la empresa

Fuente: Adaptación [20]

Para cambiar de página, simplemente basta con darle a la configuración del botón la ruta que se quería mostrar, dado que, es un requisito obligatorio, tener un botón y hacer que realice una acción al ser pulsado.

- En el ítem de ASESORÍA brinda información sobre temas de interés para favorecer la toma de alguna decisión dependiendo del proyecto que se tenga en mente como podemos ver en la Figura 21.

ASESORIAS



El informe de nuestros expertos le facilitará decidir la mejor alternativa según los requerimientos y necesidades de nuestros clientes.

Instruimos al personal interno sobre la optimización de procesos.

Entregamos apreciaciones acerca de cómo puede llevar a su organización a obtener altos estándares de rendimiento y

Figura 21: Servicios ofrecidos por la empresa: Asesoría
Fuente: Adaptación [20]

- En el ítem de CONSULTORÍA brinda información para poder dar solución a problemas apoyando el desarrollo integral de las organizaciones públicas o privadas del sector como podemos ver en la Figura 22.

CONSULTORÍA



Generamos diagnósticos organizacionales y presentamos estrategias de abordaje para resolver dichas situaciones.

Ayudamos al cumplimiento de los objetivos de su organización gracias al serio proceso de proximidad a sus necesidades e intervenciones específicas.

Abordamos el mapeo de puntos críticos y estrategias con las metodologías de vanguardia para el desarrollo organizacional para identificar nuevas áreas de mejora u oportunidades de ampliación.

Figura 22: Servicios ofrecidos por la empresa: Consultoría
Fuente: Adaptación [20]

- En el ítem de ACOMPAÑAMIENTO TÉCNICO brinda información sobre la gestión de los equipos gerenciales y operativos que permite adecuar sus sistemas y métodos de trabajo, como podemos ver en la Figura 23.



Figura 23: Servicios ofrecidos por la empresa: Acompañamiento Técnico

Fuente: Adaptación [20]

- En el ítem de GENERACIÓN DE PRODUCTOS brinda información y además ofrece sus servicios para la elaboración de estructuras organizacionales estratégicas, competitivas y funcionales, como podemos ver en la Figura 24.



Figura 24: Servicios ofrecidos por la empresa: Generación de Productos

Fuente: Adaptación [20]

- En el ítem de FORMACIÓN Y DESARROLLO brinda información para poder fortalecer a los colaboradores la capacidad, estrategia y habilidad de forma interactiva y lúdica para generar ventajas competitivas y el desarrollo a nivel profesional y personal como podemos ver en la Figura 25.



Figura 25: Servicios ofrecidos por la empresa: Formación y Desarrollo
Fuente: Adaptación [20]

- En el ítem de INVESTIGACIÓN brinda información sobre cómo observar las distintas áreas como estrategia fundamental para el conocimiento y toma de decisiones proyectadas en estrategias metodológicas y de sistematización, como podemos ver en la Figura 26.



Figura 26: Servicios ofrecidos por la empresa: Investigación
Fuente: Adaptación [20]

En el PROCESO DE PRODUCCIÓN se explica cómo funciona cada uno de los servicios que ofrece la empresa, mediante el *Widget MouseRegion* se animó de forma dinámica, un texto de tal forma que al pasar el puntero encima del recuadro se despliega otra información encima de ella tal como se puede apreciar en la Figura 27.



Figura 27: Proceso de producción
Fuente: Adaptación [20]

Pero también existe un Carrusel al final donde se puede evidenciar los productos y servicios que ofrece la empresa de forma física, como lo muestra la Figura 28.



Figura 28: Productos
Fuente: Adaptación [20]

5.4. Tecnologías

En esta pestaña se ofrece información de los tres principales productos de la empresa: Bigo, Navi y Bi Go! como podemos ver en la Figura 29, el cual, lo ha posicionado y generado inversiones por su desarrollo. Estos tres productos tienen cada uno su sección dentro de la página, donde es netamente informativa y da un breve resumen de que es cada producto con sus características y especificaciones. También en el apartado de Bigo contiene las redes sociales del producto y página donde se puede acceder por medio de hipervínculos, ya que, fue el primero producto que representaba la empresa, lo cual, estaba casi completo en su estructura, tienda y redes sociales.



Figura 29: Tecnologías
Fuente: Adaptación [20]

5.5. Contacto

Este último ítem de la página, llamado CONTÁCTO, era una parte de la página que iba acompañado en todos los demás ítems siempre al final. Su objetivo es que después de ver todo lo que ofrece, siempre aparezca en el final de cada pestaña. Pero tiene una particularidad y es que, aparte de mostrar todas las opciones de contacto tiene un formulario que las personas interesadas en hacer contacto con la empresa lo pueden hacer de una manera sencilla diligenciar los recuadros con su información y esta será enviada cuando se pulse el botón de “Enviar” que vemos encerrado en el recuadro negro, como podemos ver en la parte izquierda de la Figura 30. También tiene acceso por medio de hipervínculos que al pulsar encima de cada logo, será redireccionado a esa red social.



Figura 30: Pestaña de contacto
Fuente: Adaptación [20]

Para este modelo existía un parte de *Backend* que se encarga de recibir la información escrita en el formulario y al oprimir el botón será enviada a una base de datos que luego es redireccionada a la bandeja de entrada de la empresa para poder leer todas las solicitudes como se muestra en el siguiente diagrama de secuencia representado en la Figura 31.

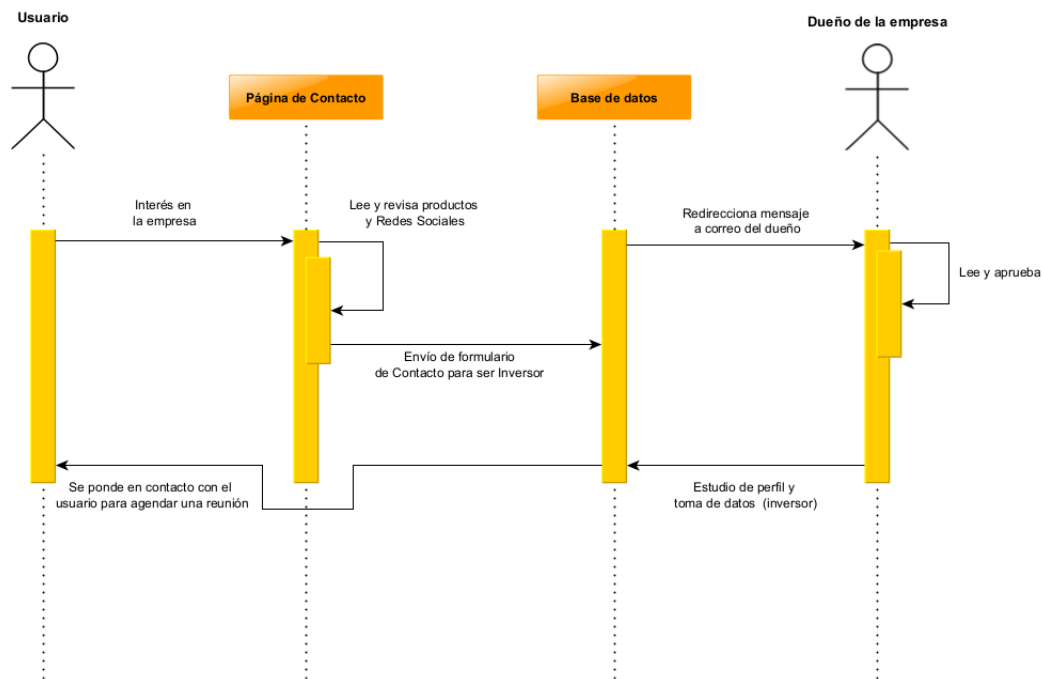


Figura 31: Diagrama de secuencia: Contacto
Fuente: Autor

6. Modelo de implementación

En el capítulo anterior analizamos como sería el modelo de vista para la página web de la empresa, pero para poder tener una estructura completa también tenemos que hablar de cómo está implementado su programación para poder controlar sus navegaciones, vistas y rutas. Dentro de estos parámetros, analizaremos los más importantes que se implementaron para la construcción de la página web.

6.1. Rutas de navegación

En el apartado que denominamos cuerpo del código, es donde inicializamos datos y variables que serán la primer vista UI del código como podemos ver representado en la Figura 14.



```
1 void main() => runApp(MyApp());
2
3 class MyApp extends StatelessWidget {
4
5   @override
6   Widget build(BuildContext context) {
7     return MaterialApp(
8       debugShowCheckedModeBanner: false,
9       title: 'CiNNoV',
10      initialRoute: '/home',
11      onGenerateRoute: RouteGenerator.generateRoute,
12      navigatorKey: navigationService.navigatorKey,
13      builder: ( _, child){
14        return HomeLayoutPage(
15          child: child ?? CircularProgressIndicator(),
16        );
17      },
18    );
19  }
20 }
```

Figura 32: Estructura de código: Cuerpo

Fuente: Autor

Lo que representa la línea 10, es que al ejecutar la aplicación y desplegarla en la web, la ruta de inicio que se mostrará después de su dominio es un “/home” como se aprecia en la Figura 33 en el recuadro negro y será enviado a una Clase de nombre *RouteGenerator*, la cual, es la encargada de manipular los nombres de las rutas que se verán reflejadas en los *links* una vez son seleccionados desde la app.



Figura 33: Barra de navegación

Fuente: Autor

Dentro de código tenemos una clase que es la encargada de manipular las rutas una vez son seleccionadas por el usuario, así, de esa forma al cambiar de pestaña, también cambia la información de la ruta mostrada en la barra de navegación.

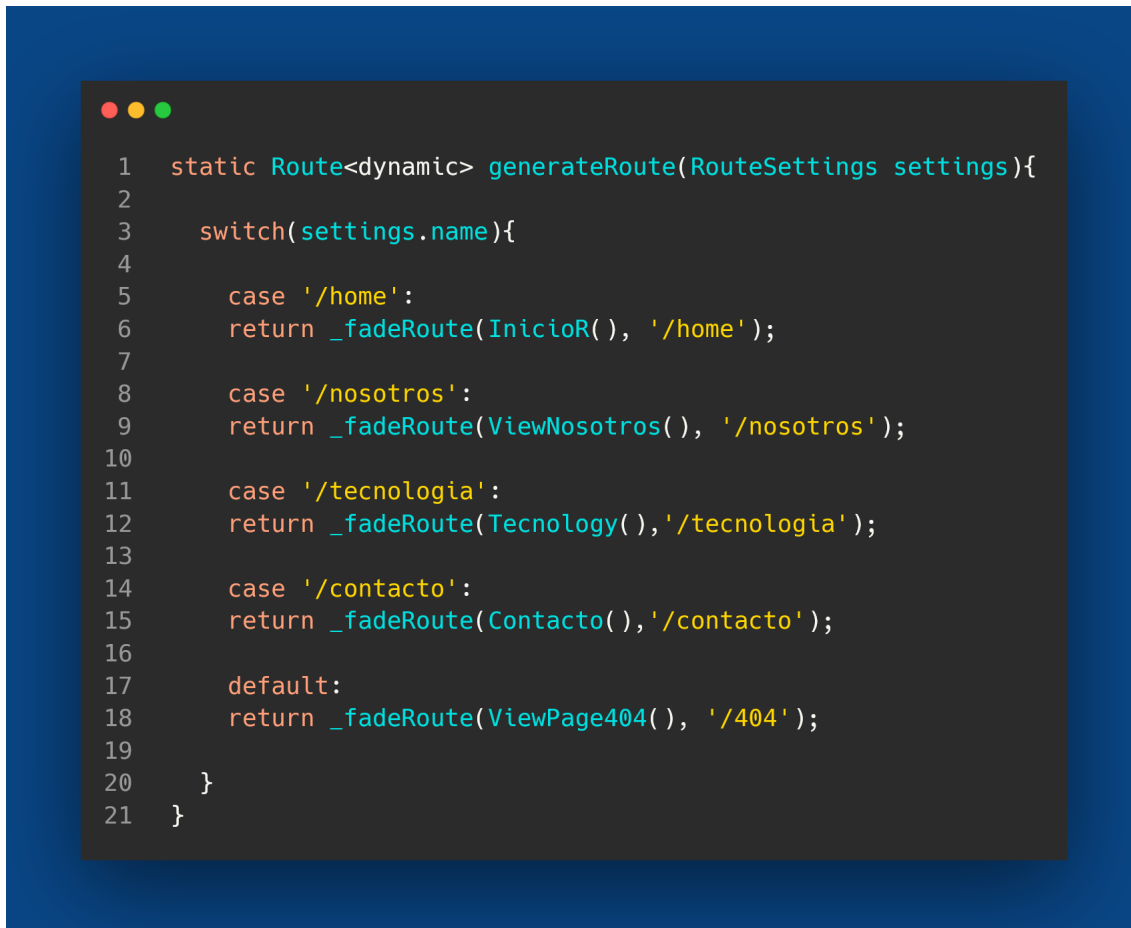


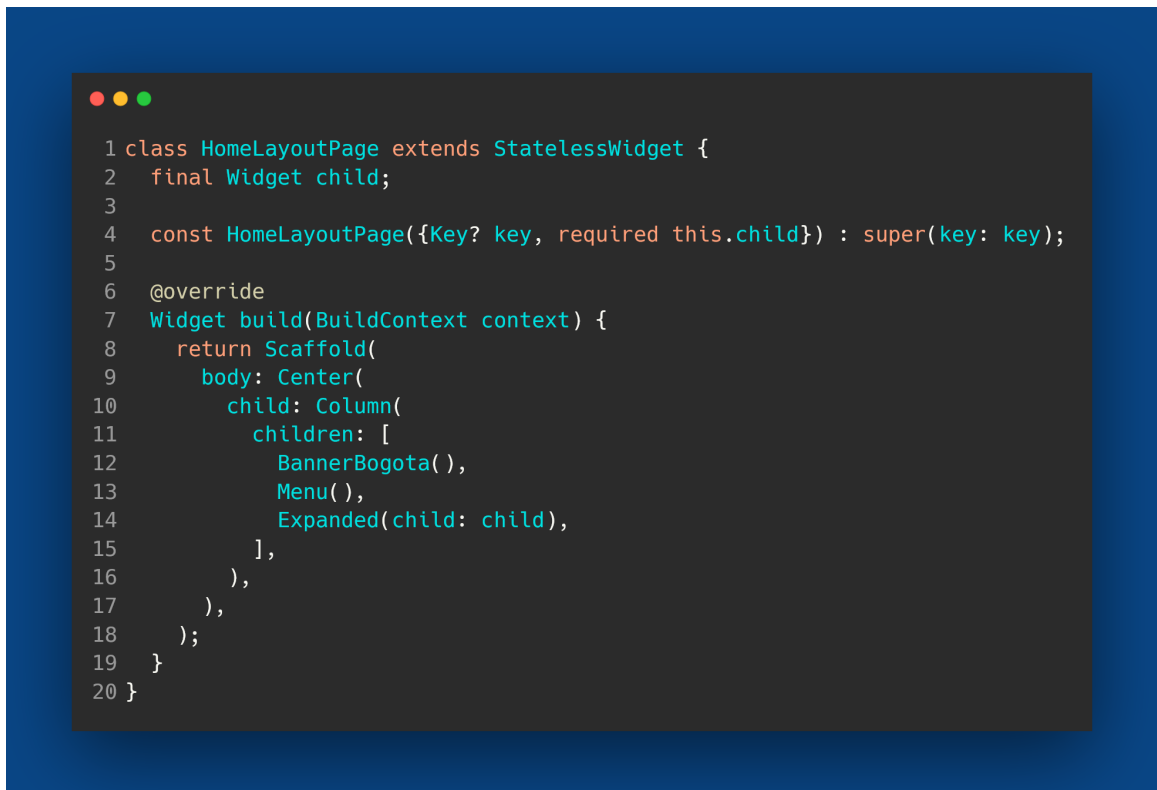
Figura 34: Estructura de código: Rutas de navegación

Fuente: Autor

Este apartado del código que podemos apreciar en la Figura 34, es el encargado de manipular los nombres que se verán reflejados en los *links*, en este caso eran manipulados desde un comparador tipo *case* que donde recibía el nombre de la ruta y devolvía, un *Widget* donde se encontraba toda la información de los apartados que conformaban la página web: Inicio, Nosotros, Tecnología, Contacto. En caso de no existir una ruta asociada con esos nombres anteriores, era devuelto el famoso *Error 404* que indica que no se encontró ese tipo de ruta para mostrar información. Una vez recibida esta información y cayendo en cierto caso de comparación, era devuelto un *Widget* llamado *PageRouteBuilder* que era el encargado de hacer la animación de la transición. Este *Widget* recibe como parámetro el *Widget* donde saltará al dar clic y el nombre de la ruta para modificar el link todo en el tiempo deseado y la animación de transición que queramos seleccionar.

6.2. Menú de navegación

Este apartado va de la mano con el mencionado anteriormente, a diferencia que aquí, explicaremos como navega en el menú, ya que, de él depende mandar los casos en forma de *String* como se ven reflejados en la Figura 34. Para comenzar, necesitamos una estructura donde fijaremos la barra de menú en la parte superior de la página mientras se hace *Scroll* dentro del contexto de la página. Seguido de esto, podemos observar en la línea de código número 14 de la Figura 32, que retorna y cae en el *Widget* de *HomeLayoutPage*. Este *Widget* tiene la siguiente estructura:



```
1 class HomeLayoutPage extends StatelessWidget {
2   final Widget child;
3
4   const HomeLayoutPage({Key? key, required this.child}) : super(key: key);
5
6   @override
7   Widget build(BuildContext context) {
8     return Scaffold(
9       body: Center(
10        child: Column(
11          children: [
12            BannerBogota(),
13            Menu(),
14            Expanded(child: child),
15          ],
16        ),
17      ),
18    );
19  }
20 }
```

Figura 35: Estructura de código: Diseño principal del código

Fuente: Autor

En la Figura 35 podemos apreciar que el único parámetro que recibe obligatoriamente es un *Widget* que cae renombrado como “*child*”. Luego de esto tenemos la estructura en forma de co-

lumna donde en primer lugar, tenemos un banner que dice la ubicación de la empresa, seguida del *Widget* dónde está toda la información del Menú, y después tenemos envuelto en un *Widget* llamado *Expanded*, el *Widget* que recibe para que ocupe todo el espacio posible.



Figura 36: Representación del cuerpo de la página

Fuente: Autor

El código de la Figura 35 se puede ver representado en forma de *View* en la Figura 36. Como podemos observar, está diseñado en forma columna, primeramente, el banner blanco de la ubicación de la empresa, seguido del menú y por último el *Widget* que se encuentra coloreado en AMARILLO como ejemplo, envuelto en el *Widget* de *Expanded* que su finalidad es expandir todo el *Widget* en el espacio que sea necesario para que se vea de una forma natural la página. Una vez explicado esto, ya podremos entender cómo funciona más fácil la navegación entre rutas, pero para entender esto de la mejor manera, daremos a conocer cómo funciona en la estructura de programación en el menú.

Una vez explicado esto, ya podremos entender cómo funciona de manera más fácil la navegación entre rutas, pero para entender esto, daremos a conocer cómo funciona en la estructura de programación en el menú.

```

1 class _MenuDesktop extends StatelessWidget {
2   @override
3   Widget build(BuildContext context) {
4     return Container(
5       child: Row(
6         children: [
7           CustomButton(
8             text: 'CONOCENOS',
9             onPressed: () => navigationService.navigateTo('/home'),
10            color: Colors.white),
11          CustomButton(
12            text: 'NUESTROS SERVICIOS',
13            onPressed: () => navigationService.navigateTo('/nosotros'),
14            color: Colors.white),
15          CustomButton(
16            text: 'TECNOLOGÍAS',
17            onPressed: () => navigationService.navigateTo('/tecnologia'),
18            color: Colors.white),
19          CustomButton(
20            text: 'CONTÁCTANOS',
21            onPressed: () => navigationService.navigateTo('/contacto'),
22            color: Colors.white),
23        ],
24      ),
25    );
26  }
27 }

```

Figura 37: Estructura de código: Menú
Fuente: Autor

Como podemos apreciar en la Figura 37, el menú estaba dirigido por el *Widget Row*, que es el encargado de organizar de forma horizontal los elementos que tenga como hijo, como se puede ver representado de en forma de *View* en la Figura 33. Dentro de la codificación se hizo un *Widget* reutilizable llamado *CustomButton* el cual recibía como parámetro obligatorio, el nombre que tendría ese botón, la acción que realizaría al ser oprimido, en este caso enviar el nombre de la ruta y el color del texto. Como podemos ver en la Figura 34, vemos que es devuelto el *String*, el cual, entra en comparación y dado su caso sea afirmado con alguno de los expuestos, éste saltaba a la pestaña dando así la orden de mostrar en pantalla el apartado y el *link* correspondiente.

6.3. Estructura de *Views*

Como todas las pestañas a las que navegaban tenían su misma estructura, vamos a explicar de forma general cómo funciona una de ellas, tomando como el ejemplo la pestaña de Inicio.



```
1 class InicioR extends StatelessWidget {
2   const InicioR({Key? key}) : super(key: key);
3
4   @override
5   Widget build(BuildContext context) {
6
7     return Container(
8
9       child: SingleChildScrollView(
10        child: Column(
11          children: [
12
13            Intro(),
14            Intro_somos(),
15            IntroLineTime(),
16            IntroAliados(),
17            IntroContacto()
18
19          ],
20        ),
21      ),
22    );
23  }
24 }
```

Figura 38: Estructura de código: Inicio

Fuente: Autor

Como podemos apreciar en la Figura 32 la ruta inicial que se manda es el caso de “/home”, que ésta cae directamente en la Figura 34 y llama al *Widget InicioR*. Este *Widget* mencionado anteriormente es el que vemos en la Figura 38 y cómo podemos apreciar es una clase de tipo *StatelessWidget*, ya que, es estática, a diferencia de la otra clase que existe dentro de la programación de Dart llamada *StatefulWidget* que es usada cuando necesitamos hacer cambios de estado e inicializarlos.

La programación que cae está sobre de la Figura 38 no es más que una *Column* que envuelve en sus hijos los *Widget* que serán mostrados en la Figura 35, la cual, recibe este parámetro en forma de *CHILD*. Esta columna es envuelta dentro de un *Widget* llamado *SingleChildScrollView*, para que, se puede hacer *Scroll* dentro de la página y así poder ver sus elementos mientras que la barra de menú permanece estática.



Figura 39: Pestaña de Inicio completa
Fuente: Adaptación [20]

La Figura 39, representa lo que mencionamos anteriormente en la programación de la Figura 38. Para una mejor estructura, se dividió en cinco *Widgets* que introdujeron en forma de columna, el cual, le da ese aspecto.

- El Widget llamado Intro() en la Figura 38, hace referencia al apartado donde se ve la portada y el título, la cual, será la primera impresión del usuario al visitar la página.
- El Widget llamado Introsomos() en la Figura 38, hace referencia al apartado donde está la información de la empresa, la que se puede ver en segundo lugar.
- El Widget llamado IntroLineTime() en la Figura 38, hace referencia al apartado que lleva por título “Ahora y en un futuro”, pues era el encargado de mostrar una serie de imágenes a la izquierda y a su derecha una línea donde brindaba más información.
- El Widget llamado IntroAlaidos() en la Figura 38, hace referencia a la red de apoyo que tiene una animación tipo carrusel, la cual, era la encargada de mostrar las empresas que tenían convenio con nosotros.
- El Widget llamado IntroContacto() en la Figura 38, era el encargado de mostrar todas las redes sociales y tener un formulario para poner en contacto al usuario con las directivas de la empresa.

De esta forma se construyó la primera pestaña haciendo referencia al Inicio, la cual, de esa misma manera, se hicieron los demás apartados. Todo esto siguiendo la misma estructura y pasando por los mismos *Widgets* que controlaban las rutas.

6.4. *Widgets* reutilizables

En el proceso de la construcción del código veremos que muchas de las cosas que estamos programando se pueden reducir al crear un *Widget* de carácter reutilizable, es decir, si necesitamos mostrar diez veces la misma imagen con el texto y las mismas funciones en diferentes situaciones, es mejor hacer un *Widget* y mandarle los parámetros que hacer nuevamente el código de la misma. Esto nos extiende bastante la escritura y por ende los recursos que consumirá el computador serán más pesado al ejecutarlo. Entonces, para no escribir cien líneas de código, es más útil crear un *Widget* reutilizable y simplemente mandarle los parámetros para que este nos devuelva el resultado o acción esperada y reducir la escritura lo que más podamos.



Figura 40: Pestaña de Contacto: Redes sociales
Fuente: Adaptación [20]

Vamos a tomar como ejemplo los logos que están en la parte de contacto como vemos en la Figura 40, ya que, tienen una particularidad y es que al dar clic encima del logo lo redirige a un hipervínculo que da información sobre esa red social. Fácilmente, podríamos hacer la codificación una por una entre pares de *Row* anidados en tres columnas, pero nos extenderíamos mucho en el código.

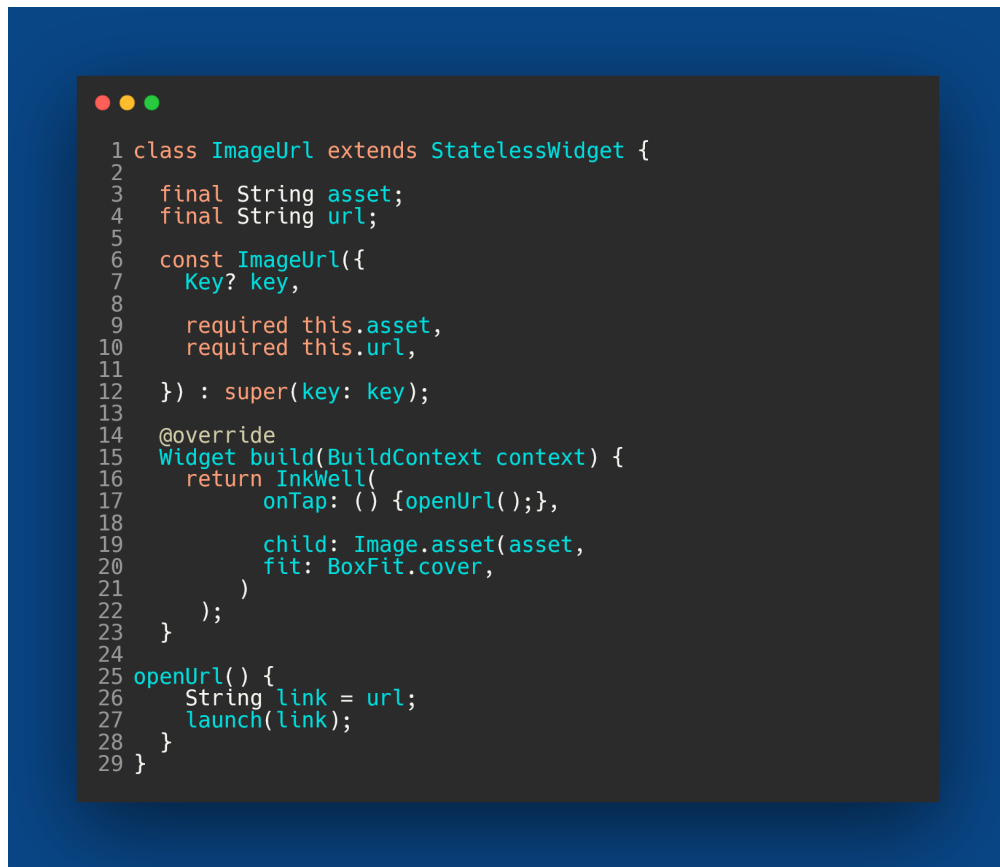


Figura 41: Estructura de código: *Widget ImageUrl*

Fuente: Autor

Para poder reducir el código en este caso, creamos una clase que se extiende de tipo *StatelessWidget*, ya que, no contaremos con cambios de estado. Los parámetros obligatorios que recibe este *Widget* reutilizable son dos *Strings*, uno para referir a la ubicación de la imagen en la variable “*asset*” y la otra para recibir el *link* en forma de texto almacenado en la variable “*url*” como podemos ver en las líneas 9 y 10 de la Figura 41. Estos datos en forma de *String* pasan por el código haciendo su trabajo de devolver una Imagen con hipervínculo que se mostraría en pantalla como se ve en la Figura 40.

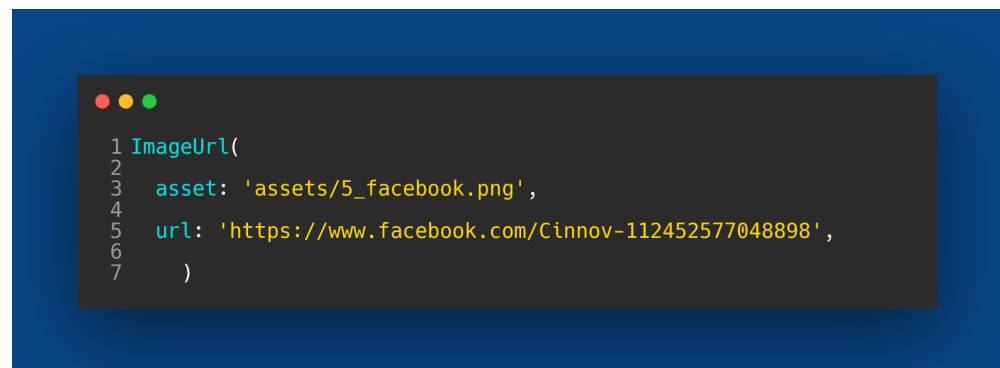


Figura 42: Estructura de código: Llamado de *Widget ImageUrl*

Fuente: Autor

Cuando estamos construyendo el código y necesitamos de este *Widget* reutilizable, solo basta con llamarlo y mandarle los parámetros que son obligatorios. En este caso, como podemos observar en la Figura 42, podemos ver que estamos construyendo el logo de Facebook y mandándole su hipervínculo. Por eso al mencionar este *Widget* le pasamos estos dos parámetros y él nos devuelve la imagen construida. Muchas de las cosas implementadas dentro del código fueron reutilizadas para evitar doblar el código y extenderlo en muchas líneas.

7. Conclusiones

- Con base en lo expuesto anteriormente, todo indica que, el realizar una página web cada vez es más necesario, debido a que, aumenta el poder llegar a las personas de una manera más formal y actualizada. El internet juega un papel muy importante y su competencia es enorme, es muy necesario que nuestra página web tenga un diseño moderno y agradable para captar la atención del usuario. Gracias al *framework* de Flutter podemos desarrollar este tipo de páginas web o aplicaciones móviles que de forma nativa se comportan de una manera excelente.
- En el proceso de estar laborando en la empresa CINNoV S.A.S, tal y como lo hemos podido comprobar, la tecnología y conectividad juega un papel muy importante en la industria a medida que avanzan los años. Participar en la investigación de la actualización sobre el aire (OTA) da pie para seguir avanzando en la actualización del dispositivo sin tener una conexión física, ya sea desde la nube o por transferencia directa desde algún protocolo de comunicación. Esto ahorrará mucho tiempo a la hora de cargar códigos a miles de dispositivos que adquiere la empresa para su distribución.
- En virtud de lo estudiado, el lenguaje de Dart en el *framework* de Flutter se convertirá en un próximo estándar de programación que evolucionará el desarrollo de páginas web y aplicaciones para sistemas operativos Android y iOS marcando un antes y un después de su desarrollo, esto a raíz de que posee muy buenas características para su uso y facilidad para la creación de aplicaciones con UI muy llamativos de alto rendimiento y sobresalientes que se ajustan a los requisitos y necesidades personales, también por su compatibilidad con servidores y líneas de comando no muy complejas de comprender.
- Tras el análisis expuesto con la construcción del código de Flutter, el tener siempre un orden estructurado y poder reutilizar *Widgets* facilita mucho la escritura del código y el ahorro de recursos a la hora de ejecutar el proyecto en simuladores, ya que, reduce el use de la máquina y por ende todo funcionará de manera más fluida.
- A modo de cierre, el haber expuesto en este trabajo de grado los parámetros de diseño y modelos de implementación, queda más fácil para futuros ingenieros, tener una guía para poder continuar y realizar cambios dentro del proyecto con el fin agregar nuevas funcionalidades en la Empresa.

Referencias

- [1] Torres Peña, Pablo Andrés y Bolívar Andrés Andrade Solórzano: *Aplicación móvil informativa y de apoyo académico de la Universidad Politécnica Salesiana*. B.S. thesis, 2021.
- [2] Albertengo, Guido, Fikru G Debele, Waqar Hassan y Dario Stramandino: *On the performance of web services, google cloud messaging and firebase cloud messaging*. Digital Communications and Networks, 6(1):31–37, 2020.
- [3] Pérez Ibarra, Susana Graciela, José Rolando Quispe, Felipe Fernando Mullicundo y Daniel Alberto Lamas: *Herramientas y tecnologías para el desarrollo web desde el FrontEnd al BackEnd*. En *XXIII Workshop de Investigadores en Ciencias de la Computación (WICC 2021, Chilecito, La Rioja)*, 2021.
- [4] Prayoga, Regawa Rama, Alifia Syalsabila, Ghifari Munawar y Rahil Jumiyan: *Performance Analysis of BLoC and Provider State Management Library on Flutter*. Jurnal Mantik, 5(3):1591–1597, 2021.
- [5] Nieto Cortés, Julián David: *Implementación de una aplicación web con servicio de chatbot con inteligencia artificial que permita la autogestión de cuentas por pagar de los proveedores de la Universidad Autónoma de Bucaramanga*. 2020.
- [6] Sabharwal, Navin y Amit Agrawal: *Introduction to Google dialogflow*. En *Cognitive virtual assistants using Google Dialogflow*, páginas 13–54. Springer, 2020.
- [7] Paredes Silva, Lucia Bernarda: *Chatbot basado en Dialogflow para mejorar la atención al cliente en un Colegio Odontológico del Perú*. 2021.
- [8] Castellano Lendínez, Laura: *KANBAN. METODOLOGÍA PARA AUMENTAR LA EFICIENCIA DE LOS PROCESOS*. 3C Tecnología, 8(1), 2019.
- [9] Benito Herranz, Álvaro y cols.: *Desarrollo de aplicaciones para IoT con el módulo ESP32*. 2019.
- [10] Astigarraga, Julen y Verónica Cruz-Alonso: *¿ Se puede entender cómo funcionan Git y GitHub! Ecosistemas*, 31(1):2332–2332, 2022.
- [11] Cetina, Iuliana, Dumitru Goldbach y Natalia Manea: *Udemy: a case study in online education and training*. Revista Economică, 70(3):46–54, 2018.
- [12] Faust, Sebastian: *Using Google s Flutter Framework for the Development of a Large-Scale Reference Application*. 2020.
- [13] Cortés Díaz, Gabriela, Nicolás Henao Godoy, Valentina Osorio Linero y cols.: *Trabajo remoto en tiempos de covid-19 y su impacto en el trabajador*. 2020.
- [14] Garrido Carrillo, Antonio: *Estructuras de datos avanzada con soluciones en C++*. Estructuras de datos avanzada con soluciones en C++, páginas 1–553, 2018.
- [15] Van Strien, Daniel: *An Introduction to Version Control Using GitHub Desktop*. The Programming Historian, 2016.

- [16] Montoya González, Diego: *Diseño y desarrollo de una aplicación móvil tipo cronometro mediante el uso del lenguaje Android Studio*. Tesis de Doctorado, Editorial Universitaria San Mateo, 2018.
- [17] Paz Díaz, María Cristina: *Diseño y desarrollo de un sitio web responsive para el taller de reparación de vehículos Celio Motor*. 2021.
- [18] Macias Vera, Erick Vicente: *Estudio comparativo de los frameworks del desarrollo móvil nativo"FlutterReact Native"*. B.S. thesis, BABAHOYO: UTB, 2021, 2021.
- [19] Nikolov, Neven: *Research firmware update over the air from the cloud*. En *2018 IEEE XXVII International Scientific Conference Electronics-ET*, páginas 1–4. IEEE, 2018.
- [20] S.A.S, CINNoV: *Diseño y Requerimientos de página WEB*. 2022.