



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**Desarrollo Front-End
en la Creación de Tableros de
Negocios**

INFORME DE ACTIVIDADES PROFESIONALES

Que para obtener el título de
Ingeniero Mecatrónico

P R E S E N T A

Jesús Alejandro Arzate Gómez



ASESOR DE INFORME

M.A. Luis Yair Bautista Blanco

Ciudad universitaria, Cd. Mx., 2025

ÍNDICE

1. EMPRESA EN LA QUE SE LABORA	4
1.1. Instituciones financieras y su repercusión en el país	4
1.2. La empresa	5
1.3. La Empresa en México	6
1.4. Estructura de la empresa en México	7
1.5. Misión y prioridades estratégicas	8
1.5.1. Misión	8
1.5.2. Prioridades Estratégicas	9
1.6. Dirección general engineering mx	10
2. DESARROLLADOR FRONT END	11
2.1. Desarrollador	11
2.2. Desarrollador web	11
2.3. Desarrollador Front-End	12
2.3.1. Comunicación entre back y front	13
2.4. Desarrollador Front End en la empresa	14
3. METODOLOGÍA AGILE	17
3.1. Introducción a la metodología Agile	17
3.1.1. Orígenes y evolución de la metodología	17
3.2. Metodología Agile	19
3.3. Principales Metodologías Ágiles en el Proyecto	20
3.3.1. Scrum	20
3.3.2. Extreme Programing (XP)	23
3.3.3. Kanban	24
3.4. Efectos de la metodología en la empresa y en la sociedad	25
3.4.1. La metodología 'agile': cambio la cultura en la empresa	25
4. CRM	27
4.1. CRM (Customer Relationship Manager)	27
4.2. Servicios en la nube	29
4.2.1. Sales Cloud	30
4.2.2. Desarrollo personalizado (Components Aura)	34
4.3. Configuraciones y permisos personalizados en el servicio de la nube	34
4.3.1. Configuraciones personalizadas	35
4.3.2. Metadata personalizada	36
4.3.3. Conjuntos de permisos	37
5. DESARROLLO FRONT END PARA TABLEROS DE NEGOCIO	38
5.1. Introducción al proyecto	38
5.1.1. Historias de usuarios	39
5.1.2. Servicios de la información	41
5.2. Necesidad del cliente	42
5.2.1. Sección de filtros	42
5.2.2. Sección gráfica	48
5.3. Marco y diagrama de flujo de trabajo	51

5.4. Configuraciones y metadata personalizada	55
5.4.1. Configuraciones personalizadas	56
5.4.2. Metadata personalizada	56
5.5. Desarrollo picklist	57
5.6. Desarrollo componentes contenedor, comisiones, facturación	61
5.7. Desarrollo componente gestor e implementación de componentes reutilizables	62
5.8. Despliegue de desarrollo a ambientes productivos	63
6. RESULTADOS	64
6.1. Resultados de sección de filtros	64
6.2. Resultados de sección gráfica	66
6.3. Resultados del aplicativo para la empresa	70
7. CONCLUSIONES	72
8. REFERENCIAS	76

1. EMPRESA EN LA QUE SE LABORA

El objetivo en este capítulo será introducir al lector en el mercado financiero y las instituciones que se encuentran en el país al día de hoy. Se mencionará la historia y las repercusiones más relevantes que estas han tenido sobre la sociedad. Asimismo, se plasmará la misión y visión de la empresa financiera donde se lleva a cabo la actividad profesional, así como la gran importancia que ha tenido el área de ingeniería en México y en los países donde la empresa milita, permitiendo al lector comprender la importancia de la sede en México para la corporación multinacional.

1.1. Instituciones financieras y su repercusión en el país

Las instituciones financieras son entidades cuya actividad principal es prestar servicios de índole económico a los agentes de una comunidad, dicho en otras palabras, son individuos económicamente productivos que requieren un servicio financiero.

Las instituciones financieras son partícipes del mercado capital al captar recursos del público para luego invertirlos en activos como depósitos bancarios, valores, títulos, acciones, por mencionar las más comunes, con la finalidad de obtener un rendimiento de este activo económico. A la vez se les puede identificar como un intermediario, asesor, proveedor de créditos, seguros, entre otras atribuciones. Por ello es que resultan tan importantes en la sociedad, pues sus labores son esenciales para el correcto funcionamiento de ésta.

En México las personas que realizan una actividad económica, consideran primordial una institución financiera para el manejo de sus bienes. Otro punto a resaltar es la seguridad que pueden otorgar a la sociedad con los productos que ofrecen, como seguros de vida o seguros de gastos médicos mayores.

Estas entidades no han dejado de reinventarse a lo largo del tiempo, actualmente se encuentran inmersos en la innovación tecnológica. Es así que han surgido las herramientas digitales, donde las aplicaciones de banco han sido hasta el día de hoy las más representativas. Éstas permiten que los usuarios accedan a los servicios financieros a través de sus dispositivos móviles, sin la necesidad de acudir a una sucursal para atender sus necesidades.

En 2019, de acuerdo con la Asociación de Bancos de México (ABM), había 51 millones de usuarios de banca por internet, un crecimiento de más de 100% en los últimos cinco años, y 17 millones de personas ya usaban apps bancarias, considerando que hasta hace poco no había tanto acceso a este tipo de servicios desde el celular. [1]

El presidente de la Asociación de Bancos de México (ABM), Daniel Becker, explicó que, de marzo del 2021 a febrero del 2022, se realizaron 2,081 millones de operaciones a través de la banca móvil, lo que representó un crecimiento de 184% anual; mientras que en banca por internet hubo 1,586 millones de operaciones en el periodo, 14.4% más. [2].

Las instituciones financieras han estado presentes en la sociedad desde hace mucho tiempo y, gracias a su capacidad de reinventarse, seguirán vigentes en el futuro.

1.2. La empresa

La historia de la empresa en la que trabajé para efectos de este reporte es la historia de muchas personas que, desde mediados del siglo XIX, han formado parte de más de un centenar de entidades financieras que se han ido uniendo para ampliar su proyecto empresarial. Al día de hoy, la empresa trabaja por un futuro mejor para las personas, buscando relaciones duraderas con sus clientes que con el paso del tiempo van cambiando la forma de ver el negocio. Gracias a todo lo mencionado anteriormente la empresa se ha convertido en todo un referente mundial y en uno de los bancos más reconocidos y premiados.

Tres fueron los sucesos que marcaron el rumbo de BBVA (La empresa) en el mercado financiero multinacional, el primer punto de inflexión fue la creación de dos financieras como pilares importantes, siendo el primer pilar el Banco de Bilbao como banco de emisión y descuento, liderada por la Junta de Comercio en 1857 en Bilbao, una ciudad situada al norte de España. Se trató de una iniciativa privada, impulsada por un contexto de crecimiento económico de la región. Por otra parte, el segundo pilar, el Banco de Vizcaya se funda en 1901; realiza sus primeras operaciones en Bilbao y poco a poco se extiende por toda España.

En 1960, Banco de Bilbao aumenta su dimensión integrando otros bancos y comienza a crear un grupo financiero; mientras, Banco de Vizcaya continúa su crecimiento, se consolida como un banco universal moderno y va configurándose como una importante institución financiera. Pero no fue hasta el año de 1998 que se establece como tal el segundo punto de inflexión, cuando se firma el acuerdo de fusión de Banco de Bilbao y Banco de Vizcaya para crear BBV.

En el año de 1991 se constituye la Corporación Bancaria de España como sociedad estatal y entidad de crédito con estatuto de banco. Inicia su historia con un modelo de banca federada, pero en 1998 se integran Corporación Bancaria de España, BEX (fusionado con BCI), BHE y Caja Postal en un solo banco que opera con la marca de Argentaria.

El tercer punto de inflexión fue la fusión entre BBV y Argentaria el 19 de octubre de 1999, con el objetivo de seguir creando valor. El nuevo banco (BBVA) nace con un tamaño relevante, una fuerte solvencia patrimonial y gran estructura financiera, una adecuada diversificación geográfica de los negocios y de los riesgos y, como consecuencias de todo ello, un mayor potencial de crecimiento de beneficios. [3]

1.3. La Empresa en México

La empresa en México tiene sus principios 1932, con el Sistema de Bancos de Comercio, luego su paso por una serie de financieras como Probursa, Multibanco Mercantil de México, Promex, Banca Cremi y Banco de Oriente. Finalmente, en el año 2000 se unen los objetivos de dos importantes grupos financieros en México y España: Bancomer y BBVA, una fusión que ha permitido establecer con firmeza a la empresa en el país. En 2019, BBVA inicia un proceso de unificación de su nombre en todo el mundo y en 2020 se elimina la marca "*Bancomer*" para llamarse únicamente BBVA.

La empresa en México tiene como objetivos primordiales, otorgar créditos que puedan impulsar el desarrollo de las familias y las empresas. La prioridad es que los clientes sean el centro de todas las decisiones, además de contribuir con el crecimiento económico al

participar activamente en proyectos que, como efecto principal, es la generación de nuevos empleos que benefician a la economía de la sociedad y del país.

La empresa da servicio a más de 27 millones de clientes en todo el país y cuenta con una base de 671 mil clientes Pyme. Al cierre de septiembre del año 2022 la cartera empresarial registró 598 mil millones de pesos, así como 636 mil millones de pesos en crédito a personas físicas y sus familias, para llegar a un monto que ha superado los 1.4 billones de pesos de cartera total. [4]

La innovación y modernidad son dos características que colocan a la empresa como una de las más relevantes de la transformación digital. La empresa en México es parte de la constante evolución de la industria bancaria, lo que ha lo catapultado a ser parte de un sistema altamente competitivo, donde la clave es la innovación. En la actualidad se realizan aproximadamente 105 millones de transacciones al mes en los canales digitales (web y móvil) y en los primeros nueve meses de 2022 las ventas digitales representaron 74.6% del total. Esta tendencia alista en la implementación de estas nuevas herramientas digitales, favorece la inclusión financiera y acerca a las personas a los servicios bancarios a través de los teléfonos celulares, que cuentan con millones de usuarios que se encuentran en segmentos que se están bancarizando. [4]

En 90 años la empresa en México se ha caracterizado por incorporar objetivos que marquen una diferencia positiva en la sociedad. La sostenibilidad se ha vuelto parte de las prioridades de la empresa, con la meta de mejorar la calidad de vida de las personas en un entorno respetuoso para el medio ambiente y llevar a sus clientes a una transición hacia un futuro más verde.

1.4. Estructura de la empresa en México

Los numerosos logros obtenidos por la innovación y el uso de nuevas formas de trabajo ágiles para entregar al cliente innovadoras soluciones en tiempo récord, le han permitido al Banco ser hoy una referencia en transformación digital. Por lo que la empresa al tener dicha necesidad de asumir este cambio de la manera más transparente posible, ha adquirido la siguiente estructura a nivel global.

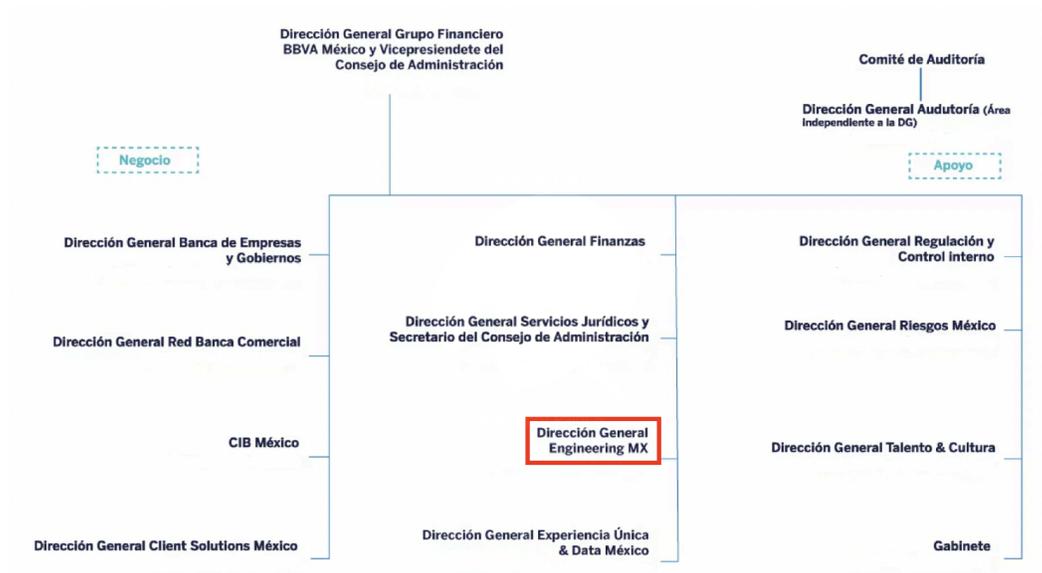


Imagen 1. Estructura de la Empresa en México
(En el recuadro en rojo se marca el área en el que me desempeñe). [1]

1.5. Misión y prioridades estratégicas

1.5.1. Misión

La misión de la empresa es **“Poner al alcance de todos, las oportunidades de esta nueva era”**.

Esta misión refleja el papel de la empresa, cómo facilitador para ofrecer a los clientes las mejores soluciones bancarias, ayudándoles a tomar las mejores decisiones financieras e impactar positivamente en su vida.

Ya que vivimos en la era de las oportunidades, lo que facilita el acceso a la educación a través de la tecnología; abre la posibilidad de acometer proyectos y perseguir los sueños a un colectivo mucho más amplio de lo que tradicionalmente era posible.

1.5.2. Prioridades Estratégicas

- **Mejorar la salud financiera de nuestros clientes**
Ayudar a nuestros clientes, mediante un asesoramiento personalizado, en la toma de decisiones y en la gestión de sus finanzas, para que puedan alcanzar sus objetivos vitales y de negocio.
- **Ayudar a nuestros clientes en la transición hacia un futuro sostenible**
Alinear paulatinamente nuestra actividad al Acuerdo de París y usar nuestro papel para ayudar a nuestros clientes en su transición hacia un futuro más sostenible inspirando por los Objetivos de Desarrollo Sostenible seleccionados.
- **Crecer en clientes**
 - *Acelerar el crecimiento rentable posicionándonos allí donde están nuestros clientes, apoyándonos en nuestros propios canales digitales y los de terceros.*
- **Buscar la excelencia operativa**
 - *Proporcionar la mejor experiencia de cliente, con procesos sencillos y automatizados, y manteniendo nuestro enfoque en la gestión robusta de los riesgos y la asignación óptima de capital.*
- **El mejor equipo y el más comprometido**
 - *Un equipo diverso y empoderado, guiado por nuestro propósito, valores y comportamientos e impulsado por un modelo de desarrollo de talento que proporciona oportunidades de crecimiento para todos.*
- **Datos y tecnología**
 - *Capacidades avanzadas de análisis de datos, junto con una tecnología segura y confiable que nos permita crear soluciones diferenciales de gran calidad y cumplir con nuestra estrategia.*

1.6. Dirección general engineering mx

Dentro de la estructura de la empresa se encuentra la Dirección General de Engineering MX, siendo esta el área tecnológica del banco. Dicha dirección alberga a las siguientes subdirecciones:

- Dirección de Infrastructure and Communication
- Dirección de Transformation Solutions
- Dirección de Arquitectura Engineering
- Dirección de Software Engineering

La dirección de Software Engineering es la palanca de transformación de la empresa, ya que facilita los desarrollos tecnológicos que el negocio utiliza diariamente. Esta dirección es responsable de la Banca de Empresa y Gobierno, dentro de la cual se encuentra el área de *Digital Work Place*. Es en esta última donde se realizó específicamente el desarrollo profesional en la subárea de desarrollo.

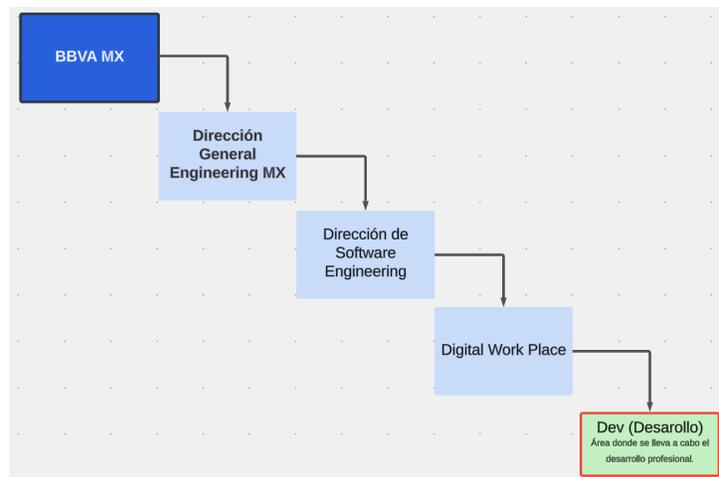


Imagen 2. Área donde se lleva a cabo el desarrollo Profesional.

2. DESARROLLADOR FRONT END

En este capítulo se hablará de una de las áreas (Desarrollador Front End) más importantes en la rama de desarrollo web en programación, esto le permitirá al lector saber más sobre el rol e importancia que esta figura tiene en la resolución de proyectos en equipos multidisciplinarios en la empresa y el mundo. A la par se abordará, el cómo es que es tipo de área interactúa con los diversos desarrollos que actualmente la empresa está llevando a cabo, generando productos de calidad y altamente competentes, con la finalidad de crear una mayor satisfacción en los usuarios y clientes.

2.1. Desarrollador

Un desarrollador podríamos describirlo como el encargado de transmitir requerimientos, necesidades e ideas de los usuarios para finalmente plasmarlas en un lenguaje que sea capaz de comunicarse con una herramienta tecnológica (computadora), capaz de realizar órdenes para efectuar cálculos y operaciones lógicas en muy poco tiempo. Con esta herramienta se pueden crear desde algoritmos, aplicaciones, sitios web, entre muchas otras opciones. Existen muchas áreas en las que un desarrollador puede desempeñarse, entre los cuales nos encontramos algunas de la más comunes como lo son:

- **Desarrollador web**
- **Desarrollador móvil**
- **Desarrollador de Videojuegos**
- **Cloud Computing**
- **Data Science**
- **Desarrollo en Ciberseguridad**

2.2. Desarrollador web

Para el fin de este escrito nos enfocaremos exclusivamente en el **Desarrollo Web**, qué es el proceso de diseñar y desarrollar sitios y aplicaciones en esta área, este tipo de desarrollo se divide a la par en dos subáreas las cuales son **backend** (encargada de todo el procesamiento de datos detrás de un aplicativo) y **frontend** (encargada de la parte visible y la interacción directa con los usuarios con los desarrollos), siendo este último la subárea en la que se enfoca el desarrollo profesional y que se abordará en siguientes capítulos.

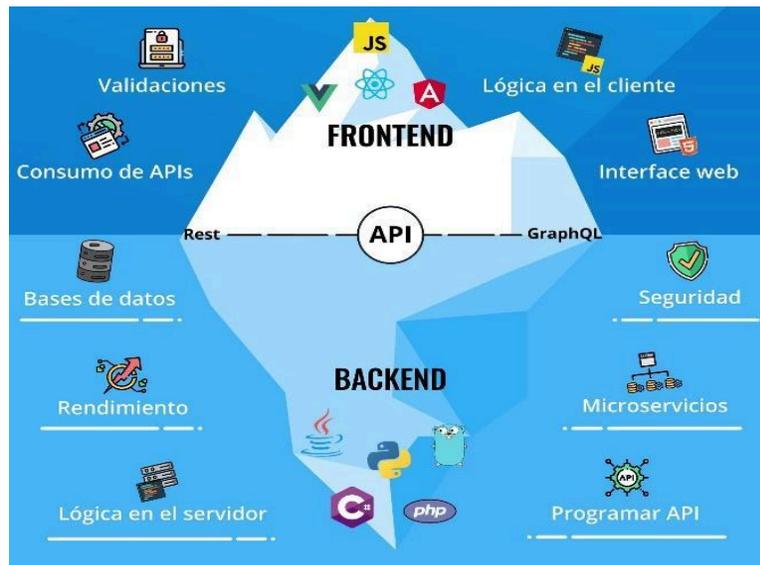


Imagen 3. Desarrollo Web. [2]

2.3. Desarrollador Front-End

Un desarrollador frontend es un experto de la programación y diseño web, encargado de crear e implementar la interfaz visual de un sitio web o aplicación. En este proyecto, mi contribución profesional se centró en el diseño y desarrollo de una interfaz visual, con la que los usuarios interactúan día a día para poder llevar a cabo sus actividades de manera eficiente. Mi trabajo como desarrollador frontend fue clave para ofrecer una experiencia de usuario funcional, intuitiva y, sobre todo, atractiva.

Como desarrollador frontend, utilicé la combinación de lenguajes de programación, como HTML, CSS y JavaScript. El HTML (HyperText Markup Language) define la estructura del contenido; CSS (Cascading Style Sheets) se emplea para dar estilo y diseño a la página; y, por último, JavaScript se encarga de la lógica que permite la funcionalidad de la interfaz.

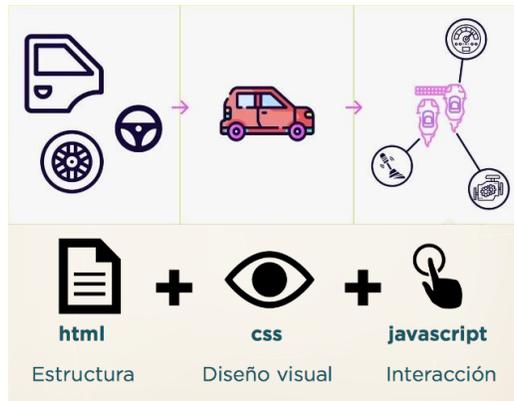


Imagen 4. Desarrollo Frontend. [3]

Mi trabajo como desarrollador frontend consistió en traducir diseños, requisitos y necesidades del negocio en aplicaciones funcionales y eficientes, garantizando así una experiencia de usuario intuitiva y satisfactoria.

2.3.1. Comunicación entre back y front

Para poder lograr la visualización del aplicativo, primero debe realizarse una etapa de procesamiento de la información, tarea que le corresponde al back. Una vez procesada, esta información se transmite al **front** y existen diversas formas de establecer esta conexión. A continuación, se presentan algunos de los métodos más comunes:

- **Middleware:** Actúa como una capa intermedia entre el backend y el frontend, y se encarga de proporcionar funciones adicionales para procesar y manipular los datos que se envían y reciben. Este método puede realizar tareas tales como la autenticación de usuarios, el registro de solicitudes y respuestas, manejo de errores, la validación de datos, compresión de datos, enrutamiento de solicitudes y caché de respuestas por mencionar las más comunes. Intercepta las solicitudes enviadas desde el frontend al backend y

viceversa, con la finalidad de procesar la información de acuerdo a las reglas definidas, de tal forma que esta llegue tanto al frontend y al backend de manera óptima.

- **API (Application Programming Interface):** Para este método el backend proporciona una API que permite al desarrollador frontend solicitar y recibir datos desde el servidor. La API actúa como un intermediario entre el Frontend y el Backend, permitiendo la comunicación y el intercambio de información estructurada. El Front realiza peticiones a la API para obtener los datos necesarios y luego los utiliza para mostrar la información en la interfaz de usuario.
- **Endpoints específicos:** El backend puede exponer endpoints específicos que el frontend puede utilizar para realizar solicitudes HTTP y obtener datos en un formato determinado, como JSON. Estos endpoints pueden estar diseñados para proporcionar datos específicos, como listas de usuarios, productos o mensajes, y el frontend puede hacer solicitudes a estos endpoints para obtener la información necesaria.

Estos métodos de comunicación permiten al frontend obtener los datos procesados por el backend y mostrarlos adecuadamente en la interfaz de usuario. La comunicación efectiva entre el frontend y el backend es fundamental para crear aplicaciones que ofrezcan una experiencia satisfactoria.

2.4. Desarrollador Front End en la empresa

En la empresa un desarrollador frontend desempeña un papel fundamental, ya que es el encargado de desarrollar y darle mantenimiento a las diversas aplicaciones y sitios web con los que cuenta la empresa. La principal responsabilidad es trabajar en la capa final de un desarrollo o aplicativo multidisciplinario, es decir, en la parte visible y manipulable por los usuarios.

Por mencionar algunas de las actividades en las que interviene un desarrollador frontend en la empresa se enlistan los siguientes puntos:

- **Desarrollo de interfaces de usuario:** Consiste en el desarrollo de interfaces de usuario de las aplicaciones web y móviles de la empresa. Asegurándose de que la interfaz sea intuitiva, fácil de usar y accesible para los usuarios, considerando también la optimización del rendimiento del aplicativo.
- **Integración con el backend:** Para un buen funcionamiento de un aplicativo o desarrollo ambas partes del desarrollo web, front y back, deben de colaborar de manera eficiente. Ambas sub áreas trabajan en conjunto para integrar las funcionalidades y los servicios ofrecidos por la empresa en la interfaz de usuario.
- **Mejoras y actualizaciones:** También participan en la mejora continua de las aplicaciones y sitios web de la empresa. Actualizándose continuamente de las últimas tendencias y mejores prácticas en el mundo del desarrollo frontend para implementar buenas prácticas, garantizar la seguridad de los aplicativos y mejorar la experiencia de usuario.
- **Implementación de diseños:** El desarrollador frontend trabaja en estrecha colaboración con los diseñadores para convertir los diseños en código funcional y satisfacer las necesidades de los clientes. Utilizando tecnologías como HTML, CSS y JavaScript para traducir los conceptos de diseño en elementos visuales interactivos.
- **Depuración y pruebas:** Los desarrolladores frontend son responsables de realizar pruebas exhaustivas en las aplicaciones y solucionar la funcionalidad, la optimización de rendimiento y la compatibilidad con diferentes navegadores y dispositivos.

Las responsabilidades o actividades exactas que un desarrollador frontend realiza en la empresa, pueden llegar a variar dependiendo del área, proyecto y equipo en el que se esté laborando.

Para la posición en específico en la cual se realizó el desarrollo profesional, la empresa pide los siguientes requisitos para poder ser considerados para la vacante:

- Desarrollador en lenguaje JAVA
- Capacidad para análisis y diseño de soluciones
- Gestión de proyectos
- Experiencia en metodologías ágiles
- Trabajo en Equipo
- Experiencia en CRM
- Habilidades blandas y comunicación efectiva
- Habilidad para detectar problemas y resolverlos de la manera más eficiente

3. METODOLOGÍA AGILE

A lo largo de este capítulo se plasmará la definición de una de las metodologías que, al día de hoy, tiene una mayor presencia en el sector de desarrollo de software y en muchas otras áreas de la ingeniería para la resolución de proyectos multidisciplinarios, así como la historia y la importancia que esta ha tenido sobre la sociedad, la Metodología Agile. A la par se abordará la estructura con la que cuenta la metodología y los beneficios que se pueden obtener si se decide implementar esta forma de resolución de problemas.

3.1. Introducción a la metodología Agile

La metodología Agile surgió como una respuesta a los diferentes desafíos y limitaciones de los enfoques tradicionales de desarrollo de software. Durante décadas, los proyectos de software se basan en métodos de gestión y planificación muy rígidos como el modelo en cascada (waterfall), donde las fases del proyecto se seguían secuencialmente: requisitos, diseño, implementación, pruebas y mantenimiento. Este enfoque tenía ciertas limitaciones, ya que no permitían adaptar el proyecto de manera ágil a los cambios o corregir errores en etapas avanzadas.

A medida que la industria del software evolucionó, surgieron procesos más flexibles, como el modelo de desarrollo iterativo y el modelo en espiral. Dentro de los beneficios más relevantes de estos modelos se tuvo una mayor interacción con el cliente y la posibilidad iterar y mejorar continuamente el producto. Sin embargo, aún se requería una metodología más ágil y adaptativa para afrontar la creciente complejidad y rapidez de los proyectos de software.

3.1.1. Orígenes y evolución de la metodología

Para poder hablar sobre el inicio de la metodología como tal, tenemos que remontarnos al siglo pasado, donde un par de jóvenes japonés Hiroataka Takeuchi e Ikujiro Nonaka completaban sus estudios en EEUU, país donde sobresalieron en el ámbito formativo como pensadores y divulgadores.

En enero de 1986 escriben un artículo para la Harvard Business Review titulado “The new new product development game”, en el que establecen

los conceptos clave de la metodología Agile:

- El proceso rápido y flexible (para la creación de productos).
- Equipos auto organizados, multifuncionales.
- Empoderamiento.
- Cómo construir software de forma iterativa.
- Aprendizaje y tolerancia al error.
- Desafío al status quo.

En paralelo, en la industria del software también estaban surgiendo nuevas formas de programación y desarrollo, como respuesta a la inquietud provocada por la rigidez del sistema. [5]

En 1986 Scrum Framework fue ideado por Ken Schwaber y Jeff Sutherland, quienes lo presentaron en la conferencia OOPSLA 95 (Object-Oriented Programming, Systems, Languages & Applications) en Austin Texas. Scrum se basa en la idea de entregas iterativas e incrementales, con equipos auto organizados y roles bien definidos, cabe mencionar que Scrum es prácticamente el estándar de facto para el desarrollo ágil.[6]

En el año de 1991 con Alistair Cockbur el método Crystal, marcó un punto de inflexión en la evolución de las metodologías de desarrollo de software, que posteriormente se integraron para formar lo que hoy en día se conoce como el marco de trabajo ágil. Este tipo de metodología se aplica para grupos pequeños de entre 6 y 8 desarrolladores localizados en la misma área, trabajando a la par en sistemas no críticos para el desarrollo, que, dicho de otra forma, es tolerable a fallos. [7]

Pero no fue hasta febrero del 2001, cuando un grupo de diecisiete expertos en el área de software se reunieron en la localidad de Snowbird, Utah, para discutir y definir un enfoque más eficaz para el desarrollo de software. Como resultado de esta reunión, se redactó el “ Manifiesto Ágil”. Dicho

manifiesto resalta 4 puntos muy importantes:

- Individuos e interacciones sobre procesos y herramientas.
- Software funcionando sobre documentación extensiva.
- Colaboración con el cliente sobre negociación contractual.
- Responder al cambio sobre seguir un plan rígido.

Estos valores se centran en la flexibilidad , la adaptabilidad y la orientación al cliente como elementos clave para el éxito del desarrollo de software en proyectos. Este acontecimiento generó un cambio total de paradigma en la industria y estableció las bases para la metodología ágil.

3.2. Metodología Agile

La metodología ágil es más que un marco de trabajo para el desarrollo de proyectos, donde el enfoque es tener una mayor flexibilidad y rapidez a la hora de gestionar proyectos, podría describirse como una filosofía que supone una forma distinta de abordar problemas, e implementa una forma de trabajo y organización bien estructurada. El objetivo es desarrollar productos y servicios de calidad que respondan a las necesidades de los clientes cuyas prioridades cambian a una velocidad cada vez mayor.

Inicialmente, la metodología ágil tuvo una amplia adopción en compañías de desarrollo de software, y a lo largo del tiempo ha demostrado su capacidad para mejorar significativamente los niveles de calidad, así como impulsar la motivación y productividad de los equipos de tecnología (TIC). Sin embargo, se ha reconocido la relevancia de incorporar la metodología ágil en cualquier tipo de empresa, ya que esto conlleva a una mayor agilidad, rapidez y enfoque en el cliente. Al implementar métodos ágiles, las empresas generan una mayor dinámica y adaptación a las demandas cambiantes del mercado, logrando así una ventaja competitiva significativa.

La metodología ágil establece que las fases secuenciales no son la mejor opción para abordar un problema, por otro lado, el desarrollo de actividades simultáneas e incrementales en varias áreas es la más óptima.

3.3. Principales Metodologías Ágiles en el Proyecto

Tras la publicación del Manifiesto Ágil, fueron apareciendo diversas metodologías que poco a poco ganaron popularidad en la comunidad de desarrollo de software, siendo **Scrum** la que se profundizará con mayor enfoque con efectos en el escrito.

3.3.1. Scrum

Scrum se convirtió en uno de los marcos de trabajo ágiles más conocidos y ampliamente utilizados. Proporciona un enfoque estructurado para el trabajo en equipo, con roles claros y ciclos de desarrollo llamados “sprints”. Esta



técnica permite abordar problemas de gran dificultad, a la par que se desarrollan productos de excelente valor de una forma productiva y creativa.

Los roles que llegamos a encontrar dentro del Scrum son los siguientes:

- **Product Owner** (Propietario del producto)- Es la persona responsable de conseguir el máximo valor de negocio y de escuchar las peticiones del cliente.

Algunas de las actividades que desempeña este rol son:

- Transmite al equipo la visión de negocio, explicar el porqué de las funcionalidades que se realizan.
- Responsable del retorno de la inversión del producto en construcción y prioridades por iteración de acuerdo al feedback recibido por los clientes.
- Acepta o rechaza incrementos en productos durante el sprint.

- Garantiza los requerimientos y prioridades por iteración de acuerdo al feedback proporcionado por los clientes.
- Define , prioriza y clarifica el backlog del equipo.

- **Scrum Master** - Es la persona que facilita el ambiente óptimo para que el equipo pueda implementar el proyecto de manera eficiente y eficaz.



Algunas de las actividades que desempeña este rol son:

- Responsables de que Scrum se entienda y se adopte correctamente en el equipo, facilitando ceremonias y velando por que se realicen.
- Le soporte al Product Owner en las ceremonias para gestionar el Product Backlog.
- Líder orientado al servicio.
- Facilita la adopción de Scrum en el equipo mediante coaching, mentorización y formación.
- Resuelve impedimentos.
- Garantiza el proceso de mejora continua(Inspect & Adapt), accionando los puntos resultantes de las ceremonias.
- Ayuda y orienta al equipo hacia la autoorganización.
- Protege al equipo de distracciones externas.

- **Development Team** (Equipo de desarrollo) - Son las personas responsables designadas por el Product Owner que deben entender los requisitos planteados por el Product Owner y crear los entregables del producto/servicio.



Algunas de las actividades que desempeña este rol son:

- Multidisciplinar y con visión end-2-end del proyecto.
- Responsable de cómo realizar los desarrollos del producto en construcción.
- Colaboración con el Product Owner para partir las funcionalidades en parte más pequeñas que se puedan desarrollar en una iteración.
- Responsable de estimar las funcionalidades del producto.
- Auto-Organización.

La metodología Scrum siempre trabaja con un mismo ciclo o flujo de trabajo para alcanzar el objetivo en común y dicho flujo se estructura en torno a 5 eventos principales.

- Product Backlog

Es la primera fase y en el momento en el que se obtiene el máximo de información sobre el proyecto y se empiezan a definir las tareas y cuestiones de mayor relevancia como los objetivos a alcanzar, qué efectos tendrá el proyecto sobre la empresa, entre otros puntos.

- Sprint Planning

En esta fase se planifica quién realizará cada tarea, en cuánto tiempo y qué perfil es el adecuado para ello. Este plan se crea en colaboración de todo el equipo Scrum. Dicha planificación de sprints está limitada a un máximo de 4 horas por cada Sprint. Otro punto importante a destacar, es que el equipo de desarrollo sea capaz de entender claramente el alcance y la carga de trabajo que conlleva el próximo Sprint (Sprint Backlog), y con ello poder explicar cómo se van a lograr los objetivos establecidos y cómo se creará el valor buscado por el cliente.

- **Daily Sprint**
Son reuniones diarias que el equipo de desarrollo lleva a cabo durante el periodo del sprint, el enfoque de estas reuniones debe ser muy puntual, por lo que la duración de esta, no debe de pasar de los 15 minutos de duración. En la sesión se deben de realizar preguntas similares a las siguientes: ¿Qué hiciste ayer? ¿Qué te falta? ¿Qué tienes listo para hacer hoy? ¿Has tenido algún problema o impedimento? ¿Tienes dudas?, con el objetivo de dar un estatus de las tareas o progreso de cada uno de los integrantes que conforman el scrum y de esta forma que todos estén alineados del avance o problemas que el proyecto pueda presentar. De igual forma durante esta sesión se planifica la carga de trabajo para las próximas 24 horas.
- **Sprint Review**
Tiene lugar después de cada Sprint para valorar el avance y adaptar la lista de tareas si es necesario. Esta revisión debe de tener una duración máxima de 4 horas por cada Sprint.
- **Retrospectiva del Sprint**
Es una oportunidad para que el equipo de Scrum lleve a cabo una retroalimentación y cree un plan de mejora para implementar el siguiente Sprint. Dura como máximo 3 horas por cada Sprint.

3.3.2. Extreme Programing (XP)

XP se enfoca en la excelencia técnica y la entrega de software de alta calidad mediante prácticas como la programación en pareja, pruebas unitarias, integración continua y desarrollo basado en iteraciones cortas. Este método hace hincapié en respuesta a las necesidades cambiantes de los clientes.

Esta metodología implementa prácticas bien establecidas que se pueden clasificar en cuatro grupos: Retroalimentación a escala precisa, Proceso

continuo, Comprensión compartida y Bienestar del programador.

Hoy en día, es común que algunas de estas prácticas sean implementadas por los integrantes del Scrum, incluso realizan la combinación de metodologías ágiles como Scrum y Kanban a la par.

3.3.3. Kanban

Aunque Kanban se originó en la industria automotriz (Toyota), su aplicación en el desarrollo de software ha sido valiosa. Kanban es un método de gestión del flujo de trabajo que optimiza el valor ofrecido al cliente al mejorar la eficiencia, efectividad y previsibilidad general de un proceso siguiendo una serie de principios:

- Visualizar la carga de trabajo: el equipo usa una tabla Kanban para mostrar las tareas a realizar a través de la cadena de valor. El trabajo se divide en etapas que se representan en columnas donde se situará cada tarea dentro del flujo de trabajo establecido. Al crear un modelo visual el equipo puede observar la carga de trabajo, incluidos los obstáculos y las colas, y aumentar la colaboración y comunicación.
- Limitar el trabajo en curso: Asignar límites explícitos a la cantidad de elementos que pueden encontrarse en progreso en cada estado del flujo de trabajo.
- Gestionar y mejorar el flujo de trabajo: Implementar un flujo rápido y eficaz mediante la gestión y monitorización de su velocidad a través de métricas, KPIs y analíticas para garantizar transparencia y gestión activa.
- Establecer políticas explícitas: Con el objetivo de asegurar la eficiencia del proceso, es importante que los miembros del equipo entiendan el estatus del proyecto, a la vez que comprendan su asignación adecuadamente para garantizar el progreso. Para ello, se debe establecer el proceso de manera que esté bien definido, publicado y socializado. Esto se puede hacer mediante políticas, directrices o reglas de proceso.
- Mantener una mejora continua: Los equipos realizan propuestas para mejorar los procesos, buscando alcanzar la máxima eficiencia.

El tablero de Kanban más sencillo está compuesto por tres columnas, aunque se puede añadir “stories” y “testing” para completar el proceso:

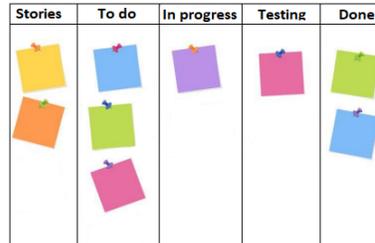


Imagen 5. Ejemplo de tablero Kanban.

3.4. Efectos de la metodología en la empresa y en la sociedad

La metodología Agile se está convirtiendo en una gran tendencia entre muchas empresas. Cada vez son más las grandes corporaciones que adoptan las prácticas Agile aprovechando la experiencia de los que fueron pioneros, o sus propias experiencias internas. Las empresas que tienen pensado implementar esta metodología, deben antes identificar la adaptación al contexto específico de la empresa, para validar si esta se apega a la cultura, objetivos de negocio o su background.

La experiencia demuestra que no existe un camino bien marcado para adoptar estas metodologías, sino que la transición debe generarse desde dentro, por y para la empresa, traducción todas las casuísticas de éxito y las mejores prácticas a un lenguaje y método que la empresa comprenda y pueda implementar.

3.4.1. La metodología ‘agile’: cambio la cultura en la empresa

El viaje de la empresa para convertirse en una organización ‘agile’ comenzó en 2014. Una transformación de una organización 100% digital que, a día de hoy, continúa avanzando. “En la empresa, el ‘agile’ ha llegado para quedarse”, afirmaba Ricardo Forcano, responsable del talento y cultura, recientemente.

La empresa se ha convertido en una de las primeras compañías con más de 100.000 empleados en transformarse en una organización ‘agile’. De hecho,

en la actualidad ya son más de 16.000 empleados los que están trabajando con estas metodologías, una cifra que crece semana a semana. [5]

La adopción de la filosofía 'agile' está ayudando a tener claras las prioridades, alineando las fuerzas de distintas áreas, pensando a nivel local y global. Y, por encima de todo, está permitiendo a los equipos colaborar íntegramente, poner al cliente en primer lugar, innovar, pensar en grande y aportar valor de forma global.

4. CRM

En el capítulo que se analizará a continuación tocará diversos puntos sobre el impacto e importancia de los CRM en la industria tecnológica de software, financiera, marketing y ventas por mencionar las que más influencia tienen sobre la empresa. Además se plasmarán las ventajas y las desventajas de utilizar estas potentes herramientas de desarrollo y los beneficios que la empresa obtiene al implementarlo en sus diversos aplicativos.

4.1. CRM (Customer Relationship Manager)

Un CRM es una estrategia, conjunto de prácticas y tecnologías diseñadas para ayudar a las empresas a mejorar y administrar las relaciones con los clientes. Almacena información de los clientes actuales y potenciales (como nombre, correo electrónico, dirección, teléfono, etc) e identifica sus actividades y puntos de contacto con la empresa.

Es importante establecer que el sistema CRM no es solo una lista de contactos simplemente, ya que sobre todo, recopila e integra datos valiosos para preparar y actualizar los equipos con información personal de los clientes, sus historiales de compra y sus preferencias, esto con la finalidad de que la empresa tenga un mejor entendimiento de las necesidades de sus clientes y ofrecerles un servicio más personalizado.

Una plataforma CRM considera relevantes los siguientes puntos que se muestran en la imagen siguiente:



Imagen 6. Actividades de un CRM. [4]

Los sistemas de CRM suelen incluir los siguientes beneficios:

1. Gestión de contactos y cuentas: Permite mantener un registro de todos los contactos y cuentas clientes, incluyendo información de contacto y detalle relevantes.
2. Automatización de marketing: Ayuda a crear y administrar campañas de marketing, segmentar audiencias y realizar seguimientos de resultados.
3. Servicio al cliente y soporte: Permite gestionar solicitudes de servicio al cliente, seguimiento de casos y realizar seguimientos de resultados.
4. Automatización de ventas: facilita la gestión de oportunidades de venta, seguimiento de ventas, pronóstico y gestión de pipelines.
5. Integración con otras herramientas: Puede integrarse con otras aplicaciones empresariales, como sistemas de contabilidad, correo electrónico y redes sociales, para una gestión más eficiente de datos y procesos.
6. Análisis y reporting: Ofrece herramientas para analizar datos de clientes, medir el rendimiento de las ventas y el marketing, y tomar decisiones informadas.

El objetivo principal de un CRM es mejorar la retención de clientes, aumentar la satisfacción de estos, optimizar las operaciones comerciales y, en última instancia, aumentar la rentabilidad de la empresa. Al mantener un registro detallado de las interacciones con los clientes y utilizar estos datos para personalizar las comunicaciones y estrategias, las empresas pueden ofrecer un mejor servicio y fortalecer las relaciones con sus clientes. Esto, a su vez, puede conducir a un aumento en las ventas y la lealtad del cliente.

La Empresa específicamente al tener convenio con el servicio en la nube (CRM), obtiene una mayor gestión de los indicadores medioambientales de sus centros corporativos y oficinas, y sin mencionar toda la capacidad que la nube le permite a la entidad financiera de simplificar el proceso de envío y validación de información. La empresa optimiza sus procesos para medir su huella de carbono y eliminar procedimientos manuales. En definitiva, se trata de una plataforma que permite realizar una gestión activa y basada en datos de sus políticas de sostenibilidad. Cabe mencionar que la empresa recibe cierto número de licencias, las cuales son repartidas entre los diferentes gremios que la forman. Un beneficio con el que cuenta la empresa al tener esta relación, es que el equipo técnico del servicio en la nube nos ayuda a dar mantenimiento a dicha herramienta a nivel general, mientras que la gestión y actualización de los desarrollos ya existentes en nuestro ambiente de producción, corren por cuenta de nuestra área de operaciones, también conocida como el área de BEX.

“En BBVA apostamos por las mejores soluciones que conduzcan hacia una transición sostenible. Incorporamos la dimensión de la sostenibilidad en nuestro día a día, tanto en relación con nuestros clientes como en nuestros procesos internos o en la gestión de todos los inmuebles corporativos”, afirma Desirée Granada, responsable global de inmuebles y servicios de la empresa.[8]

La idea de la empresa es cumplir con los objetivos del Plan Global de Ecoeficiencia del grupo bancario, que recoge entre sus objetivos reducir el consumo de electricidad por empleado un 10 % a un 7 % el consumo total, un 11% el consumo de agua y papel o un 4% los residuos netos.[8]

La Empresa recalca, que esta plataforma permite estabilizar los procesos de cálculo de emisiones de en los países donde tiene presencia para facilitar el seguimiento de los objetivos marcados por la entidad.

4.2. Servicios en la nube

Los servicios en la nube, también conocidos como servicios de computación en la nube, son soluciones tecnológicas que ofrecen una variedad de recursos y capacidades a través de Internet. Estos servicios se alojan en servidores remotos y se entregan a los usuarios y empresas a través de la web. Los servicios en la nube se han vuelto fundamentales en la industria de la tecnología de la información debido a:

- Flexibilidad - Su capacidad para adaptarse a las necesidades y metas cambiantes de las organizaciones, ofreciendo una amplia variedad de recursos informáticos bajo demanda. Permitiendo a las empresas ser ágiles, eficientes y centrarse en sus objetivos comerciales sin preocuparse por la gestión de la infraestructura subyacente.
- Escalabilidad - Permiten escalar recursos fácilmente según las necesidades. Ya que las empresas tienen la posibilidad de aumentar o reducir la capacidad de procesamiento, almacenamiento o ancho de banda de manera rápida y eficiente. Esto es especialmente útil y se refleja para aplicaciones y empresas que experimentan variaciones en la demanda de recursos, como las ventas de temporada o eventos especiales.
- Eficiencia - Su capacidad para escalar recursos bajo demanda, su modelo de pago por uso que evita la subutilización, la optimización de recursos, la automatización de tareas y aliviando a las organizaciones de tareas de mantenimiento. Además, ofrecen acceso global y tolerancia a fallos, lo que aumenta la eficiencia operativa y la continuidad del servicio. En

conjunto, estos atributos permiten a las empresas maximizar su eficiencia económica y operativa mientras se centran en sus objetivos.

La empresa no se queda atrás dentro de este rubro, y hoy en día cuenta con convenios con los servicios en la nube más importantes del momento. Y para poder llevar a cabo el CRM, la empresa implementa estos recursos para automatizar procesos comerciales, gestionar relaciones con los clientes y aprovechar los datos de manera efectiva. Algunos de los servicios utilizados por la empresa son:

- *Sales Cloud*
- *Service Cloud*
- *Marketing Cloud*
- *Commerce Cloud*
- *Platform y App Cloud*
- *Einstein Analytics*
- *Analytics*

4.2.1. Sales Cloud

Sales Cloud es un servicio en la nube, que gestiona las relaciones con los clientes (CRM). Incluye herramientas para la gestión de contactos, la automatización de la fuerza de ventas, la previsión de ventas, así como la productividad. Permite a los supervisores y los equipos de ventas gestionar el ciclo de ventas, priorizar tareas, gestionar las relaciones con los clientes y acceder a la información.[9]

Este servicio brinda la posibilidad de personalizar la herramienta dependiendo las necesidades específicas de la empresa, agregar campos personalizados, diseñar paneles, crear automatizaciones, crear flujos de trabajo y desarrollos personalizados. Para poder lograr esto, la plataforma cuenta con diferentes marcos de trabajo, pero, para fines del análisis de este escrito, se hablarán de los que más impacto tuvieron en el desarrollo del producto final, los cuales son:

- Lightning Component Framework - Marco de trabajo para el desarrollo de componentes personalizados en la interfaz de usuario de Salesforce. Permite a los desarrolladores crear componentes

reutilizables que se integran de manera nativa en Sales Cloud.

Algunos de los puntos más relevantes de este marco son los siguientes:

- ❖ Componentes Reutilizables: Los desarrolladores pueden crear componentes individuales que representan partes específicas de la interfaz de usuarios o funcionalidades de la aplicación, para posteriormente utilizarlos en múltiples sitios dentro de la aplicación o incluso en diferentes aplicaciones de Salesforce.



Imagen 7. Ejemplo de componentes reutilizables. [5]

- ❖ Desarrollo Declarativo y Programático: Los componentes se pueden desarrollar tanto de manera declarativa como programática. Los desarrolladores pueden utilizar la interfaz de usuario de Salesforce para crear componentes visualmente, arrastrando y soltando elementos, o pueden escribir código.

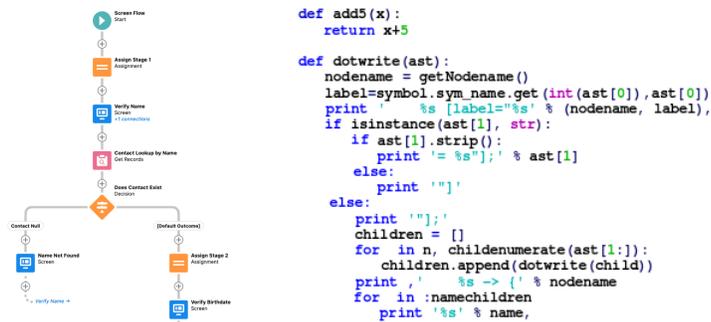


Imagen 8. Ejemplo Desarrollo Declarativo y Programático.

- ❖ Arquitectura Basada en Eventos: El framework utiliza una arquitectura basada en eventos lo que significa que los componentes interactúan y se actualizan en respuesta a acciones del usuario o cambios en los datos.
- ❖ Integración de Datos: Los componentes de Lightning pueden integrarse fácilmente con datos de Salesforce y otras fuentes externas. Esto permite que las aplicaciones personalizadas accedan y muestran datos de Salesforce y realicen operaciones CRUD (Crear, Leer, Actualizar, Eliminar) en registro.
- Apex - Es un lenguaje de secuencia centrado en objetos y muy definido que permite a los desarrolladores ejecutar declaraciones de control de transacciones y flujos en el servidor de la plataforma de Force.com junto con llamadas a la API de la plataforma Lightning. Gracias a una sintaxis parecida a Java y un funcionamiento similar a los procesamientos almacenados en una base de datos, Apex permite a los desarrolladores agregar la lógica comercial a la mayoría de eventos del sistema, entre los que se incluyen el hacer clic en los botones, las actualizaciones de registros relacionados y las páginas de visualforce. Los códigos de Apex pueden iniciarse mediante solicitudes de servicio Web y a partir de desencadenadores en objetos. En otras palabras, Apex se utiliza para crear componentes reutilizables que se integran de manera nativa en Sales Cloud. [10]

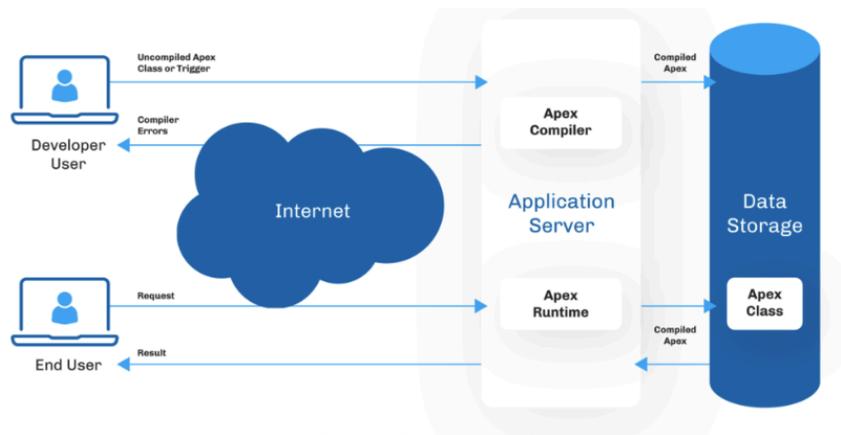


Imagen 9. Ejemplo Flujo Clases Apex. [6]

Dentro de la empresa, el servicio en la nube es gestionado por un área especializada que se dedica exclusivamente a los desarrollos realizados en esta plataforma. Esta área tiene la responsabilidad de satisfacer las necesidades de nuestros clientes internos, específicamente el área de negocios de la empresa.

Periódicamente, llegan proyectos propuestos por los clientes que se consideran capaces de aportar un valor significativo a la empresa. Una vez priorizados estos proyectos, el área se encarga de asignarlos a los especialistas correspondientes, asegurando así la entrega de soluciones y aplicaciones de valor.

En mi caso particular, una vez que fui asignado al proyecto de Tableros Cash, se me proporcionaron ambientes de desarrollo exclusivos para este proyecto. Esto me permitió realizar los desarrollos y pruebas pertinentes sin afectar ni interferir con los trabajos de otros equipos.

Es de suma importancia mencionar que, para asegurar la calidad de nuestros desarrollos, es necesario pasar por diferentes ambientes. En cada uno de estos, se llevan a cabo distintos análisis para validar que el desarrollo sea lo más eficiente posible y no contenga vulnerabilidades. Algunas de las etapas por las cuales tuve que pasar para poder integrar el desarrollo en producción.

4.2.2. Desarrollo personalizado (Components Aura)

El desarrollo de Aura Componentes personalizados en Salesforce, es un proceso que implica la creación de elementos de interfaz de usuario altamente personalizados y funcionalidades específicas que se integran en la plataforma para utilizarlos en un aplicativo.

El marco de trabajo incluye un conjunto de componentes prediseñados. Estos componentes se conocen como componentes Lightning básicos. Puede ensamblar y configurar componentes para formar nuevos componentes en una aplicación. Los componentes se procesan para producir elementos HTML DOM dentro del navegador.

Un componente puede contener otros componentes, así como HTML, CSS, JavaScript o cualquier otro código habilitado para la Web. Esto permite crear aplicaciones con interfaces de usuario sofisticadas.

El servicio de la nube ayuda al grupo a gestionar de manera más eficiente la relación con sus clientes, apoya la puesta en producción de soluciones y optimiza las campañas de marketing para ofrecer una experiencia de cliente diferencial y de valor añadido. De esta manera la tecnología y el servicio en la nube que tiene como socio la empresa, permite poner al alcance de todos las oportunidades de esta nueva era.

4.3. Configuraciones y permisos personalizados en el servicio de la nube

Para poder tener el funcionamiento adecuado de un aplicativo cuando este se realiza mediante el marco de trabajo de Componentes Aura, no basta simplemente con el hecho de desarrollar código para crear la estructura, funcionalidad y el estilo de una aplicación, también se tienen que realizar ciertas configuraciones específicas en el ambiente (también conocido como Sandbox) donde se ejecuta el programa. En nuestro caso en específico en el servicio de la nube Sales Cloud, esto para permitirle obtener cierta

información, permisos y datos al programa, que previamente deben de ser seteados o configurados, para un adecuado funcionamiento.

Existen diferentes tipos de configuraciones en el servicio de la nube utilizado; sin embargo, para los fines de este escrito, se presentarán aquellas que tienen mayor impacto en el proyecto.

4.3.1. Configuraciones personalizadas

En la plataforma las configuraciones personalizadas (Custom Setting en Inglés) es un tipo especial de objeto que permite almacenar datos personalizados de configuración. Estos datos se pueden utilizar para configurar la funcionalidad de un aplicativo y se almacenan de manera similar a los registros de objetos estándar, pero están diseñadas específicamente para almacenar configuraciones. A diferencia de los objetos personalizados que tienen registros basados en ellos, la configuración personalizada le permite utilizar datos personalizados en toda su organización. La configuración personalizada le permite también distinguir usuarios o perfiles particulares basados en criterios personalizados.

Los datos de configuración personalizada se exponen en la memoria caché de la aplicación, lo que permite un acceso eficiente sin tener que enviar repetidas consultas a la base de datos. Luego estos datos podrán utilizarse en campos de fórmula, reglas de validación, flujos, Apex, y la API de SOAP.[11]

Algunas características clave de las configuraciones personalizadas en la plataforma son:

- Almacenamiento de Datos de Configuración (Servicios).
- Accesibles desde Apex.
- Visibles para Usuarios Administradores.
- Límites y Consideraciones.

En sí, las configuraciones personalizadas son una herramienta valiosa en la plataforma para almacenar datos de configuración que permiten la flexibilidad y personalización de las aplicaciones sin necesidad de escribir código personalizado para cada configuración. Están diseñadas para ser utilizadas por administradores y desarrolladores para adaptar la plataforma a las necesidades específicas de la organización.

4.3.2. Metadata personalizada

Los metadatos personalizados son metadatos de aplicación personalizables, implementables, que se pueden incluir en paquetes y actualizarlos. Estos tipos de metadatos permiten a los desarrolladores y administradores generar estructuras de datos personalizadas para almacenar información que no es apropiada para objetos estándar o configuraciones personalizadas, y pueden ser utilizadas con el objetivo de extender la funcionalidad de la plataforma según las necesidades específicas del desarrollo.

Algunos de tipos de metadatos que se pueden generar en este apartado en la plataforma son:

- Objetos de Metadatos Personalizados.
- Componentes de Lightning (*Lightning Components*).
- Páginas Visuales (*Visual Pages*).
- *Flows*.

Los tipos de metadatos personalizados son una parte integral de la plataforma que le permiten al desarrollador la personalización y extensión de las características y funcionalidad de esta para adaptarse a las necesidades que se presenten. Cada tipo de metadatos personalizados tiene su propio propósito y uso, y juntos forman una potente herramienta para la construcción de soluciones.

4.3.3. Conjuntos de permisos

Un conjunto de permisos es una colección de configuraciones y permisos que ofrece a los usuarios acceso a varias herramientas y funciones. Los conjuntos de permisos amplían el acceso funcional de los usuarios sin cambiar sus perfiles y son la forma recomendada de gestionar los permisos de sus usuarios, esto es útil para otorgar accesos a ciertas funciones o registros de manera excepcional sin afectar a todo un conjunto de usuarios. Son un mecanismo que permite otorgar permisos y derechos específicos a los usuarios de manera más granular y flexible que los perfiles estándar.

Algunas características importantes de los conjuntos de permisos en Salesforce son:

- Personalización de Permisos.
- Renovación de permisos.
- Flexibilidad.
- Auditoría de Permisos.
- Compatibilidad en Compartir Reglas.

Los conjuntos de permisos le dan el privilegio al administrador de seleccionar quien puede tener acceso al aplicativo o a los aplicativos del sandbox, dependiendo el perfil y la jerarquía con la que cuente.

5. DESARROLLO FRONT END PARA TABLEROS DE NEGOCIO

Durante este capítulo, el lector podrá identificar, paso a paso, el proceso de implementación llevado a cabo durante mi desarrollo profesional en la empresa, con el fin de comprender el impacto de mi participación en la organización. Se profundizará especialmente en la importancia de la aplicación de principios de ingeniería para poder resolver las necesidades y problemas surgidos a lo largo del proyecto, lo cual facilitará al lector identificar como apliqué las habilidades y conocimientos adquiridos durante mi formación universitaria. Asimismo, se relatarán los desafíos y soluciones encontrados en las distintas etapas, demostrando que la ingeniería fue fundamental en el desarrollo del proyecto.

5.1. Introducción al proyecto

Pará el área de tecnología de la empresa, es de suma importancia que los servicios tecnológicos y de soporte se realicen de manera eficiente, ya que son claves para un adecuado funcionamiento y desarrollo de la infraestructura de la institución financiera, para ello se crean una gran diversidad de desarrollos y soluciones para satisfacer las necesidades de la organización. Durante el año laboral se generan una gran cantidad de proyectos e iniciativas, con la intención de siempre encontrarse a la vanguardia de la innovación tecnológica y entregar productos de calidad a nuestros clientes, para poder lograr estos objetivos las diferentes áreas que forman parte del área de Engineering Mx, distribuyen al personal capacitado a cada uno de los diferentes proyectos formando a los equipos de desarrollo, los equipos Scrum.

El equipo Scrum en el que se llevó a cabo mi desarrollo profesional se denomina "Incentives Improvement Mayorista" y tuvo como objetivo construir un aplicativo conocido como Tableros Cash. Este aplicativo proporciona a la banca datos relevantes sobre los movimientos financieros de sus clientes a lo largo de un periodo de tiempo. Su función principal es ofrecer una herramienta que centralice la información necesaria para realizar un análisis específico y claro, lo cual permite a los usuarios tomar decisiones adecuadas y eficientes sobre los clientes. Este proyecto de desarrollo web se estructura en distintas capas, entre las cuales se encuentran:

- La capa del manejo de datos Datio.
- La capa del back GCP (Google Cloud Platform).
- La capa del middleware ASO.
- La capa del front Salesforce.

A continuación, profundizaremos en la última capa, ya que es el área de expertise que desempeña como integrante del equipo. Durante la explicación, también se mencionarán algunos aspectos de las demás capas debido a la relación que existe entre ellas.

Para iniciar con el análisis, la implementación y desarrollo de un proyecto dentro de un equipo scrum en la empresa, es fundamental contar con un conjunto de criterios necesarios para el DOR (*definition of ready*). Esto se refiere a los puntos o parámetros requeridos por parte del área de negocio, los cuales me permitieron realizar mis actividades de manera adecuada y precisa. En este caso, estos criterios incluyeron específicamente las historias de usuario y servicios funcionales.

5.1.1. Historias de usuarios

En un inicio, las historias de usuario deberían ser documentos donde el representante del área de negocio, en este caso el Product Owner (PO), establece los requisitos y necesidades del aplicativo, especificando la funcionalidad esperada. Este documento idealmente incluye todos los detalles necesarios sobre el funcionamiento y comportamiento del sistema para que, yo como desarrollador, pueda realizar un análisis adecuado y crear una solución alineada con las expectativas del cliente.

Sin embargo, en este proyecto las cosas se manejaron de manera distinta, ya que cuando me asignaron al equipo, las definiciones de estas historias aún estaban en desarrollo. Por la relevancia del proyecto, me vi en la necesidad de intervenir en la creación de estas historias de usuario, una práctica que, en principio, no es común en la empresa, ya que el rol del desarrollador suele concentrarse en el desarrollo en sí, sin involucrarse en la definición de las necesidades del negocio. A pesar de ello, debido a la urgencia del proyecto, asumí un papel más activo en la comprensión de las reglas de negocio y en la identificación de los puntos de dolor de nuestros usuarios. Esto me permitió comprender de una manera más profunda el negocio dentro de la empresa, lo cual me permitió poder visualizar mejoras o áreas de crecimiento dentro del mismo aplicativo e incluso la empresa.

Esta experiencia me permitió desarrollar una capacidad valiosa para detectar soluciones oportunas que resolvieran los problemas y necesidades de nuestros clientes con mayor precisión. Para adquirir esta habilidad, tuve que investigar a fondo, analizar y comprender con un enfoque minucioso el flujo de la banca, siendo este mi primer acercamiento a la industria. Este proceso de inmersión y aprendizaje me tomó cerca de tres meses, y fue clave para poder entender las expectativas de negocio y orientar mejor mis decisiones de desarrollo.

Durante las primeras semanas en el proyecto, me enfoqué en comprender el maquetado de diseño desarrollado por el equipo de UX (experiencia de usuario), el cual establece aspectos esenciales del aplicativo, como la estructura HTML, la lógica básica del procesamiento y algunas funciones elementales del tablero, incluyendo el manejo de errores en caso de fallos en el procesamiento de la información. Sin embargo, para desarrollar una historia de usuarios completa, trabajé en conjunto con el equipo de negocio en varias sesiones de análisis profundo, donde esclarecimos detalles clave, tales como: ¿qué tipo de usuarios utilizarían el aplicativo?, ¿todos los usuarios deberían tener los mismos privilegios de visibilidad del componente?, ¿cuál sería la volumetría de información a manejar?, ¿qué tipo de datos se manipularían en el tablero? y ¿cuál sería el impacto inmediato de este desarrollo para el sector bancario?

Estas preguntas me ayudaron a orientar el desarrollo del aplicativo de forma alineada con las necesidades de nuestros clientes y dentro de los límites de la factibilidad técnica. Además, esta comprensión detallada me permitió guiar a mi equipo de manera eficaz, logrando que las planeaciones y la detección de problemas fueran precisas y contribuyeran a una ejecución ágil. Este enfoque se reflejó en la satisfacción alta que los clientes expresaron con la entrega final del aplicativo.

5.2. Necesidad del cliente

El proyecto Incentives Improvement Mayorista incluye un aplicativo que concentra información clave para respaldar la toma de decisiones en el sector bancario. Este aplicativo, conocido en el área como el Tablero de Información, contiene un conjunto de componentes específicos que proporcionan diversas funcionalidades, adaptándose a los diferentes perfiles de usuarios que lo consultan o manipulan. El desarrollo del proyecto se estructuró en torno a tres hitos principales, denominados coloquialmente como “productos” a lo largo de este reporte, los cuales se detallarán en las secciones siguientes:

- 4 productos (SIT, DEM, CIE y Canales Digitales)
- Adquirente
- Multipagos

El Tablero Cash proporciona a nuestros cuatro tipos de usuarios (Asesores Cash, Subdirectores Cash, ASI, Staff) datos detallados de los clientes mediante una variedad de representaciones visuales, como filtros, gráficas, tablas y tarjetas, que varían en función al tipo de producto seleccionado por el usuario. La estructura de este tablero se divide en dos secciones principales: la primera, conocida en este reporte como la sección de filtros; y la segunda, la sección gráfica de la información.

5.2.1. Sección de filtros

La sección de filtros del tablero tiene como objetivo facilitar la selección de campos, permitiendo obtener diferentes niveles de filtrado al consultar la información en la sección gráfica. Para lograr esto, implementé un componente llamado “Picklist”, que consiste en un conjunto de listas de selección dinámicas con diversos campos de datos. Este componente permite a los usuarios elegir el nivel de filtrado adecuado según sus necesidades de análisis.

Uno de los principales retos al analizar e implementar este componente fue el alto grado de personalización solicitado, lo cual incrementó significativamente el tiempo de personalización solicitado, lo cual

incrementó significativamente el tiempo de desarrollo, ya que fue necesario construir el componente desde cero en lugar de utilizar componentes estandarizables ya existentes. A continuación, detallo algunos de los aspectos más relevantes que impactaron el desarrollo con mi contribución:

- El nivel de personalización aumentó el tiempo de creación e implementación a casi tres veces en comparación con otros proyectos similares.
- Desarrollé aproximadamente más de 2,000 líneas de código para este componente.
- Realicé iteraciones exhaustivas para asegurar un código limpio y el cumplimiento de buenas prácticas.
- Efectué pruebas de no afectación, verificando que mi desarrollo no interfiriera con la funcionalidad de aplicativos ya productivos al momento de su implementación.
- El componente consumió información de servicios externos en lugar de la base de datos del CRM, lo cual requirió pruebas adicionales en diferentes entornos de desarrollo para validarla funcionalidad de los servicios.
- Existieron dependencias con desarrollos de las etapas primarias del proyecto (DATIO-GCP-ASO).
- Configuré una gran variedad de metadatos personalizados para poder consumir servicios de manera adecuada.

Dentro de este componente, se establecieron dos reglas de negocio principales: la estructura de las listas de selección y el perfilamiento de visibilidad para cada tipo de usuario.

La primera regla, que aborda la estructura del componente, permite que este se presente al usuario de tres maneras diferentes. Dependiendo del tipo de producto seleccionado, la disposición, el tipo de campo y el número de listas se ajustan dinámicamente, como se ilustra en los tres diagramas siguientes:

- 4 Productos

Este diagrama muestra un conjunto de filtros para la categoría '4 Productos'. Los filtros están organizados en una cuadrícula de 2x5. La primera fila contiene los campos: Producto, Canal, Fecha Ini., Fecha Fin. y División. La segunda fila contiene: Sucursal, Ejecutivo, Grupo, N.Cliente y Contrato. En la esquina inferior derecha del contenedor, hay dos botones: 'limpiar' y 'filtrar'.

- Adquirente

Este diagrama muestra un conjunto de filtros para la categoría 'Adquirente'. Los filtros están organizados en una cuadrícula de 2x5. La primera fila contiene los campos: Producto, Tarjeta, Fecha Ini., Fecha Fin. y División. La segunda fila contiene: Sucursal, Ejecutivo, Grupo, N.Cliente y N.Afiliación. En la esquina inferior derecha del contenedor, hay dos botones: 'limpiar' y 'filtrar'.

- Multipagos

Este diagrama muestra un conjunto de filtros para la categoría 'Multipagos'. Los filtros están organizados en una cuadrícula de 3x5. La primera fila contiene los campos: Producto, Subproducto, Medio Pago, Fecha Ini. y Fecha Fin. La segunda fila contiene: División, Sucursal, Ejecutivo, Grupo y N.Cliente. La tercera fila contiene el campo: Contrato. En la esquina inferior derecha del contenedor, hay dos botones: 'limpiar' y 'filtrar'.

Imagen 11. Casos Filtros-Producto

Dentro del flujo de la información en este apartado, se me señaló la importancia de establecer un orden en la selección de las listas, para asegurar un mayor control en el proceso y evitar posibles confusiones para el usuario al momento de interactuar con el tablero. A continuación, detalló el orden de los campos a seleccionar para alcanzar el nivel de filtrado más alto (suponiendo que el usuario tiene el perfil con mayor visibilidad). Este orden de selección sigue una secuencia de arriba hacia abajo y de izquierda a derecha, tal como se muestra en los tres diagramas mencionados anteriormente:

4 productos

- Selección de Tipo de Producto (Obligatorio y habilita la División)
- Selección de Canal (Opcional)
- Selección de Fecha de Inicio (Obligatorio)
- Selección de Fecha de Fin (Obligatorio)
- Selección de División (Obligatorio y habilita la Sucursal)
- Selección de Sucursal (Opcional y habilita al Ejecutivo)
- Selección de Ejecutivo (Opcional y habilita el Grupo)
- Selección de Grupo (Opcional y habilita el Número de Cliente)
- Selección de Número de Cliente (Opcional y habilita el Contrato)
- Selección de Contrato (Opcional)

Adquirente

- Selección de Tipo de Producto (Obligatorio y habilita la División)
- Selección de Tarjeta (Opcional)
- Selección de Fecha de Inicio (Obligatorio)
- Selección de Fecha de Fin (Obligatorio)
- Selección de División (Obligatorio y habilita la Sucursal)
- Selección de Sucursal (Opcional y habilita al Ejecutivo)
- Selección de Ejecutivo (Opcional y habilita el Grupo)
- Selección de Grupo (Opcional y habilita el Número de Cliente)
- Selección de Número de Cliente (Opcional y habilita el Contrato)
- Selección de Contrato (Opcional)

Multipagos

- Selección de Tipo de Producto (Obligatorio y habilita la División)
- Selección de Subproducto(Opcional)
- Selección de Medio de Pago(Opcional)
- Selección de Fecha de Inicio (Obligatorio)
- Selección de Fecha de Fin (Obligatorio)
- Selección de División (Obligatorio y habilita la Sucursal)
- Selección de Sucursal (Opcional y habilita al Ejecutivo)
- Selección de Ejecutivo (Opcional y habilita el Grupo)
- Selección de Grupo (Opcional y habilita el Número de Cliente)
- Selección de Número de Cliente (Opcional y habilita el Contrato)
- Selección de Contrato (Opcional)

La segunda regla de negocio se refiere a la visibilidad del componente, la cual impacta directamente en la capacidad de manipulación que tienen los cuatro tipos de usuarios en la aplicación. Esta regla define qué usuarios pueden visualizar y seleccionar valores de las listas, las cuales funcionan como parámetros de entrada en el primer servicio utilizado durante el desarrollo. Estos parámetros determinan la información que se reflejará posteriormente en el resto del tablero. A continuación, presento un

diagrama que ilustra la visibilidad asignada a cada tipo de usuario en relación con los diferentes tipos productos gestionados:

Listas de selección							
Perfil	Producto	División	Centro Responsable (CR)	Unidad Gestora (UG)	Grupo	Numero de Cliente	Convenio o Contrato o Afiliación
Staff Central	Completa	Completa	Completa	Completa	Completa	Completa	Completa
ASI	Completa	División asignada por default	Completa	Completa	Completa	Completa	Completa
Subdirector Cash	Completa	División asignada por default	Completa	Completa	Completa	Completa	Completa
Asesor Cash	Completa	División asignada por default	Conjunto de CRs asignados que le pertenecen al Asesor	Conjunto de UGs asignados que le pertenecen al Asesor	Completa	Completa	Completa

Visibilidad y manipulación

Completa
 Parcial
 Nula

Imagen 12. Tabla de visibilidad para los usuarios del Tablero Cash

En el diagrama anterior se muestra la visibilidad de los cuatro diferentes tipos de usuarios: el Staff Central, quien tiene la mayor capacidad de manipulación y acceso a todo el universo de información; el ASI y Subdirector Cash, quienes pueden visualizar únicamente la información de la división a la que están asignados; y finalmente, los Asesores Cash, que solo pueden ver datos de su división, centro responsable y unidad gestora, limitando considerablemente el universo de información disponible.

Para superar este reto, realicé múltiples sesiones de análisis en colaboración con los equipos de negocio y arquitectura, además de iterar con pruebas de usuario para afinar la lógica y garantizar que cada perfil accediera únicamente a la información relevante sin comprometer la seguridad ni la usabilidad. Esta estrategia me permitió desarrollar un flujo estándar para el componente, facilitando futuras implementaciones en otras áreas y beneficiando a la organización al reducir tiempos de desarrollo para proyectos similares.

Este primer componente consume el servicio de Filtros, el cual requiere un parámetro de entrada obligatorio: el tipo de producto. Para realizar consultas a este servicio, el usuario debe seleccionar al menos un valor en

dicho campo, y la respuesta del servicio devolverá un conjunto de datos correspondiente al siguiente campo seleccionado por el usuario. En el primer caso, sin embargo, se devuelven los canales disponibles y el rango de fechas de la información que se encuentra en la base de datos para el producto seleccionado.

En resumen, la implementación de este componente no solo resolvió la necesidad de la visibilidad y control de acceso, sino que también generó un beneficio importante al crear un componente estandarizado, reutilizable y adaptable para futuras aplicaciones. En la siguiente imagen, se detalla el funcionamiento del primer hito con los cuatro productos principales.

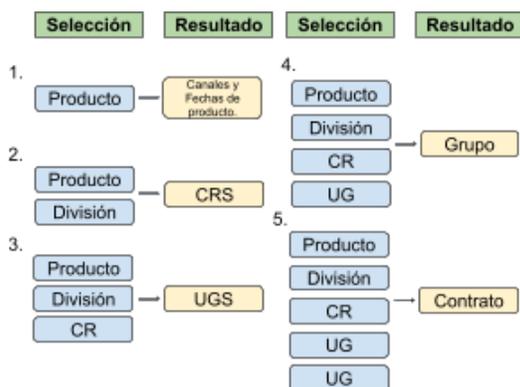


Imagen 13. Casos presentes en el consumo del servicio Filtros

En la imagen se pueden observar diversos llamados al servicio, numerados del uno al cinco. Los campos en color azul representan los parámetros de entrada que le usuario debe seleccionar en las listas desplegadas, mientras que los recuerdos de color amarillo indican las respuestas esperadas del servicio correspondiente. Este flujo se repite tanto para el hito de Adquirente como para Multipagos, aunque cada uno utiliza parámetros de entrada distintos en la consulta al servicio. A pesar de estas diferencias en los parámetros, el comportamiento del servicio permanece consistente al momento de consumirlo.

Implementar esta lógica uniforme en ambos hitos fue un desafío importante. Tuve que asegurarme de que cada variación en los parámetros

de entrada se ajustará a las necesidades específicas de cada hito sin perder la estandarización y funcionalidad general del servicio. Para lograrlo, realicé pruebas exhaustivas y ajustes iterativos en el flujo de llamadas al servicio, lo que no solo mejoró la precisión del filtrado para cada tipo de usuario, sino que también optimizó la experiencia del usuario final, haciendo que la interacción fuera intuitiva y eficiente.

Esta solución me permitió simplificar el proceso de consulta para usuarios, asegurando que, independientemente del hito o del producto seleccionado, se mantuviera un flujo homogéneo y fácil de seguir. Este enfoque estandarizado no solo facilita el mantenimiento y la actualización del sistema, sino que también incrementa la escalabilidad del componente, permitiendo su adaptación en futuros desarrollos de manera ágil.

5.2.2. Sección gráfica

La segunda sección del aplicativo Tablero Cash se compone de cinco tipos de tableros: comisiones, comisiones-adquirente, transacciones, transacciones-adquirente y facturación. El usuario puede visualizar cada uno de estos tableros a través de una interfaz de ventanas que puede manipular según sus preferencias. Cada tablero incluye un conjunto de gráficos, tablas y tarjetas, que varían en estructura y contenido de acuerdo al tipo de producto y el nivel de filtrado seleccionado previamente en el componente picklist. Esta flexibilidad permite que la sección gráfica presente un tipo de tablero con una estructura específica, adaptada para mostrar información crucial sobre los clientes.

Implementar esta variedad de tableros y asegurar que se adaptarían correctamente a cada nivel de filtrado y tipo de producto fue un proceso complejo. Fue necesario crear reglas de negocio claras para definir cómo y cuándo se debería desplegar cada estructura, lo cual implicó coordinar una lógica precisa entre el componente de picklist y la sección gráfica. Durante el desarrollo, enfrenté el reto de manejar múltiples combinaciones de filtros y estructuras, asegurándome de que cada una de ellas cumpliera con las necesidades específicas de cada perfil de usuario.

Esta sección gráfica sigue dos reglas de negocio principales que permitieron optimizar la personalización sin sacrificar el rendimiento. Estas reglas fueron fundamentales para lograr una experiencia de usuario coherente, manteniendo la organización y claridad de la información presentada en cada tablero, lo que facilitó la toma de decisiones para los usuarios. Además, este enfoque estructurado permitió crear una solución modular y escalable, que puede adaptarse fácilmente a futuras ampliaciones o ajustes sin necesidad de rehacer el diseño desde cero.

- 4 productos y multipagos mostrarán los tableros de comisiones y transacciones.
- Adquirente mostrará los tableros de comisiones-adquirente, facturación y transacciones-adquirente.

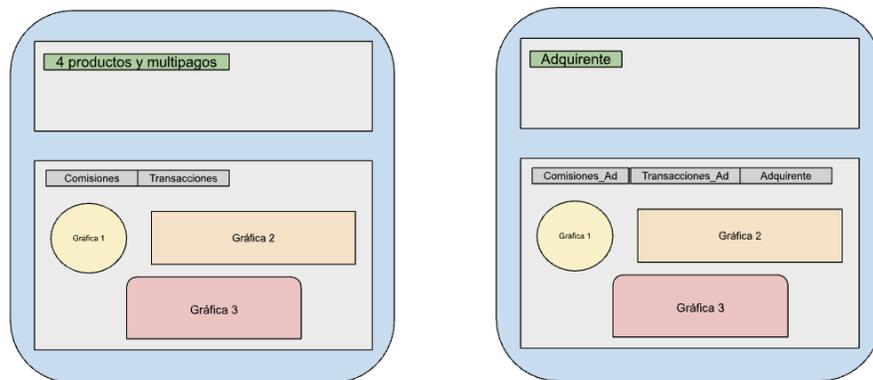


Imagen 14. Estructura del Tablero Cash

Esta sección del tablero realiza el consumo del servicio de summary, que recibe como parámetros de entrada los valores seleccionados por el usuario en la sección de filtros, además de otros parámetros predeterminados que se configuran automáticamente según el tipo de gráfico, tabla o tarjeta que se va a desplegar. Este flujo de datos facilita que el usuario obtenga información detallada y precisa de acuerdo a sus necesidades de análisis.

Para definir la estructura óptima de esta sección, organicé una serie de sesiones colaborativas con el equipo de UX (Experiencia de Usuario). En

estas reuniones, trabajamos en conjunto para encontrar un equilibrio entre la viabilidad técnica y la experiencia del usuario. Fue un proceso que requirió no solo comprender los principios de UX, sino también adaptar los requisitos técnicos a los estándares de diseño. Uno de los desafíos más importantes fue asegurar que las interacciones y visualizaciones fueran intuitivas, sin comprometer el rendimiento del tablero, lo que me llevó a profundizar en los detalles del diseño y adaptarlos a las limitaciones técnicas.

A lo largo de este proceso, también adquirí conocimientos básicos en diseño, los cuales me permitieron realizar ajustes en la maqueta utilizando Figma. Esta habilidad fue clave para ajustar rápidamente elementos visuales y proponer mejoras de diseño que optimizan tanto la apariencia como la usabilidad del tablero. Además, recibí asesoría en la creación de contenido, asegurándome de que los textos implementados no solo cumplieran con las normas internas de la empresa, sino que también fueran claros y útiles para el usuario final.

Estas experiencias de trabajo interdisciplinario con UX no solo contribuyeron a crear un tablero alineado con las expectativas del usuario, sino que también me proporcionaron herramientas valiosas para mejorar la comunicación y la interpretación de las necesidades del cliente en futuros desarrollos.

5.3. Marco y diagrama de flujo de trabajo

Para abordar el desarrollo y responder a las necesidades de negocio antes mencionadas, el primer paso que realicé fue un análisis exhaustivo para determinar el marco de trabajo más adecuado. Comencé por elaborar un cuadro comparativo que resumiera las ventajas y desventajas de los marcos de trabajo más utilizados en el entorno de servicios en la nube de la empresa. Este análisis incluyó aspectos clave como la escalabilidad, el rendimiento, la integración con otros servicios de la nube, y la facilidad de mantenimiento.

Marcos de Trabajo	Conexión con el Middleware	Facilidad de implementación	Desarrollo Personalizado	Reutilización de componentes	Necesidad de negocio
Aura components	SI	Media	SI	SI	NO
Lightning Web Components	NO	Media	SI	SI	NO
Flujos	NO	Media	NO	SI	NO
Analytics	NO	Media	NO	SI	NO

Imagen 15. Cuadro comparativo de los marcos de trabajo en Salesforce

Este análisis me permitió identificar las soluciones que mejor se alineaban con los objetivos del proyecto y las restricciones técnicas. La comparación fue particularmente útil al anticipar posibles problemas de integración y ajustar las expectativas del equipo de desarrollo en cuanto a los tiempos y los recursos necesarios. Además, al presentar este cuadro comparativo al equipo de arquitectura, pude justificar de manera clara y fundamentada la selección del marco de trabajo que optimizaría el rendimiento del tablero en la nube, asegurando que fuera una decisión alineada tanto con las expectativas de negocio como con las capacidades técnicas del equipo.

El cuadro comparativo me dio una visión clara sobre el marco idóneo para el desarrollo del aplicativo. Al analizar la conexión con el *middleware*, un aspecto crucial en el flujo de información, decidí que el marco de componentes Aura era la opción más adecuada. Esta elección se destacó debido a la necesidad de desarrollar una solución altamente

personalizada que pudiera gestionar de manera efectiva tanto la visibilidad de perfiles como los diversos elementos gráficos requeridos en el tablero.

Dado que esta fue mi primera experiencia trabajando con Aura, me enfrenté a un reto significativo, ya que el aprendizaje de este marco implicaba estudiar una gran variedad de recursos, consultar bibliografía especializada, y cumplir con una serie de certificaciones internas con calificaciones aprobatorias impuestas por el área. Este proceso de aprendizaje fue continuo durante todo el desarrollo del aplicativo. Al mismo tiempo, debía cumplir con las entregas programadas para el área de negocios, lo que me obligó a gestionar una curva de aprendizaje acelerada para obtener las habilidades necesarias y garantizar que la implementación técnica fuera óptima. Esta experiencia no solo fortaleció mi capacidad de adaptarme a nuevas tecnologías, sino que también me impulsó a mejorar en la gestión de mi tiempo y en la resolución de problemas bajo presión.

Una vez seleccionado el marco de trabajo, el siguiente paso fue revisar si ya existían componentes dentro del gremio que pudieran satisfacer las necesidades del proyecto, pues la empresa fomenta la reutilización de componentes con el fin de optimizar tiempos y dirigir los esfuerzos hacia innovaciones. En esta búsqueda, identifiqué tres componentes específicos que, para los fines de este informe, denominaremos de la siguiente manera:

- Componente Gráfica:** Este componente permite la creación de diversos tipos de gráficos, como gráficos de dona, barras apiladas, puntos y líneas, utilizando la información procesada de los servicios.
- Componente Imagen de Fondo:** Diseñado para aplicar distintos fondos en la interfaz frontal y empleado especialmente en la gestión de errores de los servicios. Utilizar este componente me permitió manejar de manera eficiente los mensajes de error visuales, asegurando que los usuarios tuvieran una experiencia intuitiva al enfrentarse a posibles problemas de carga de datos.
- Componente Tabla:** Capaz de crear tablas y tarjetas dinámicas a partir de la información procesada de los servicios. Este componente resultó crucial para el tablero, ya que permitía a los usuarios visualizar datos complejos de una manera

clara y organizada. Además, facilitó la implementación de tarjetas y tablas sin tener que desarrollar la funcionalidad desde cero, contribuyendo a una experiencia de usuario efectiva y optimizando los tiempos de desarrollo.

Cada uno de estos componentes me ofreció beneficios específicos, ayudándome a cumplir con los objetivos del proyecto y a minimizar el tiempo de desarrollo. A través de este proceso, no solo adquirí conocimientos avanzados en Aura y en la reutilización de componentes, sino que también desarrollé un enfoque analítico más profundo para evaluar la viabilidad de soluciones preexistentes y aplicarlas de manera estratégica dentro del proyecto.

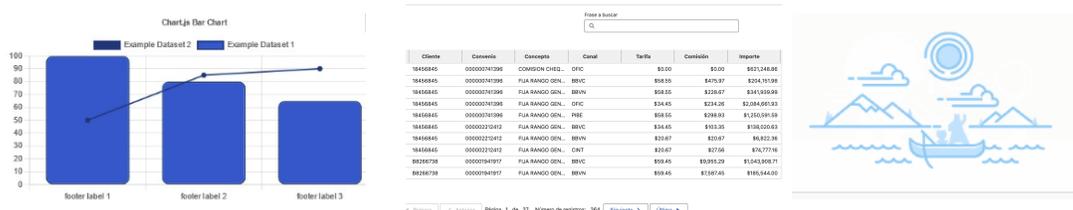


Imagen 16. Componentes Reutilizables (Gráfica,Tabla y Imagen de Fondo)

Teniendo en cuenta que estos componentes cumplían en su gran mayoría con la necesidades del desarrollo, realicé un diagrama de flujo de trabajo el cual estuviera preparado para poder utilizarlos dentro de nuestro aplicativo, considerando las implicaciones que esto conllevaba. Dicho diagrama después de un par de sesiones de análisis e implementación lo conformé de la siguiente forma:

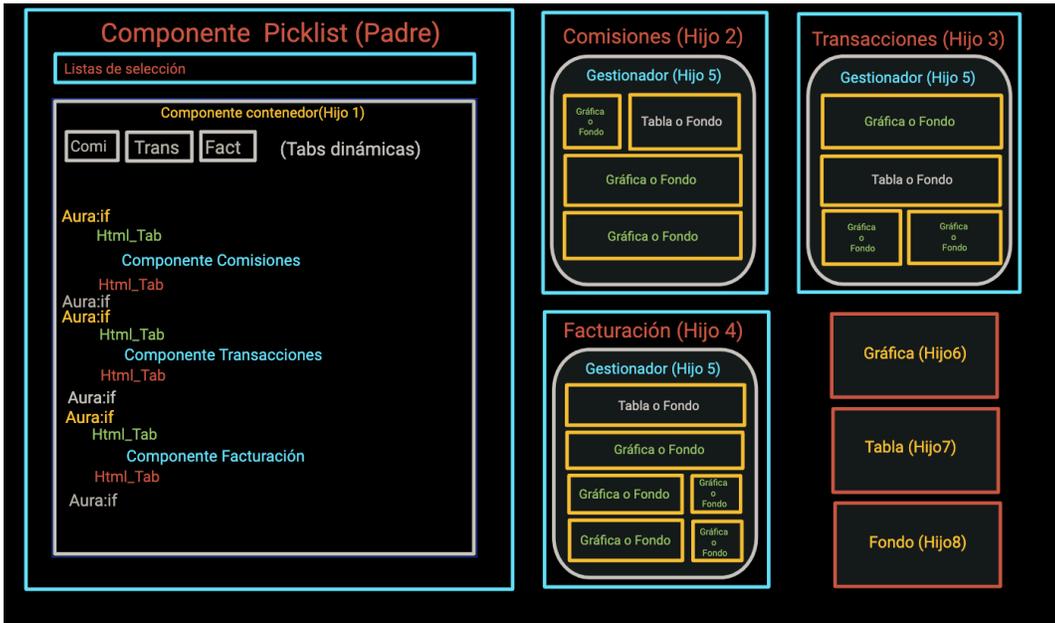


Imagen 17. Diagrama de flujo del Desarrollo

Como se puede apreciar en la imagen anterior, el flujo comienza con nuestro primer componente, el componente padre, que es el componente picklist. Esta elección la basé en dos puntos principales: primero, el nivel de filtrado que el usuario debe realizar, y segundo, como se mencionó anteriormente en este escrito, la visibilidad del aplicativo para los diferentes usuarios. Posteriormente, el conjunto de valores seleccionados se transmite en un objeto (`Map<String,String>`) al componente Contenedor (hijo 1) mediante un enlace de datos. Este componente tiene la capacidad de seleccionar el siguiente componente correspondiente (Hijo 2, Hijo 3 o Hijo 4) según el tipo de producto seleccionado en el padre. Este control se realiza mediante un bloque de `aura:if`, que nos permite validar el tipo de producto que tiene el objeto en proceso. Dentro de este componente, el usuario tiene la capacidad de seleccionar dinámicamente el tipo de tablero correspondiente al tipo de producto elegido, mediante un apartado que en la nube conocemos como `tabs`. Una vez que se ha identificado correctamente el producto y el tipo de tablero, el flujo puede llamar a cualquiera de los siguientes tres componentes: Comisiones, Transacciones y Facturación. Es importante recalcar que, independientemente del tipo de tablero elegido por el usuario, el flujo después de este punto es similar para los tres casos. Cualquiera de los componentes mencionados

anteriormente tendrá la función de establecer y organizar la interfaz de usuario donde se visualizarán las gráficas, tablas y tarjetas correspondientes al tipo de tablero seleccionado. Llegado a este punto, el siguiente componente que entra en acción es el Gestor, que será el encargado de procesar el consumo del servicio summary para transmitir, según el caso, la respuesta de dicho proceso a los tres diferentes componentes reutilizables (gráfica, tabla y fondo) y así completar el flujo de información y plasmar todos los elementos del tablero.

El desarrollo del diagrama de flujo resultó crucial para garantizar el funcionamiento óptimo del aplicativo, ya que requería implementar e integrar distintos componentes de manera que pudieran coexistir y comunicarse sin inconvenientes. Para lograr esto, el componente contenedor y el componente gestor desempeñaron roles fundamentales en el proceso, ya que su correcta implementación era clave para coordinar los demás elementos.

El análisis y desarrollo de estos componentes me tomó aproximadamente un mes, debido a la complejidad y precisión que requería este paso. Tuve que realizar múltiples pruebas exhaustivas para asegurarme de que cada componente respondiera correctamente en los distintos puntos del flujo. Esta fase inicial fue particularmente desafiante, ya que debía considerar las configuraciones específicas y los parámetros de entrada tanto de los componentes reutilizables existentes como de aquellos que diseñé desde cero.

Este trabajo no solo mejoró la eficiencia del flujo de datos en el tablero, sino que también me permitió garantizar que cada interacción del usuario con el aplicativo fuera ágil y libre de errores. Al superar estos retos, pude entregar un sistema robusto, optimizando así la experiencia del usuario final. Además, este proceso me ayudó a consolidar conocimientos técnicos sobre integración de componentes y me dio una perspectiva más amplia de cómo crear soluciones escalables y fáciles de mantener en el tiempo.

5.4. Configuraciones y metadata personalizada

Para poder utilizar los servicios y consultar la información registrada en metadata personalizada, es de suma importancia realizar y crear las configuraciones pertinentes.

Por lo que, lo primero que hice antes de empezar a crear cualquiera de los componentes que tiene el tablero, fueron las configuraciones y los metadatos personalizados, relativamente aquí lo que realicé fue configurar y setear los valores que necesitó mi desarrollo en la base de datos del CRM, con la intención de almacenarlos como variables estáticas y pudiera consultar la información al momento de ejecución de mi desarrollo.

5.4.1. Configuraciones personalizadas

Realicé configuraciones personalizadas principalmente para consumir los servicios utilizados en el tablero. Esto se debió a que los llamados a los servicios deben de tener ciertas características y configuraciones específicas. Aunque estos parámetros podrían establecerse desde el controlador del lado del servidor, opté por las configuraciones personalizadas para obtener un mayor control y gestión de los valores generados a futuro.

5.4.2. Metadata personalizada

Utilicé metadata personalizada para generar catálogos de información, así como para configurar componentes reutilizables, servicios y, sobre todo, abordé de manera más detallada la configuración de un "granting ticket" (GT) fiable. Estos tickets permiten acceder a servicios protegidos en la red y contienen parámetros que aumentan la seguridad al consumir servicios.

Para utilizar este GT, desarrollé un método en el controlador del servidor capaz de obtener los datos de los usuarios que realizan llamadas a cualquier servicio que requiera esta configuración. Es importante destacar que este proyecto fue pionero en el uso de este tipo de GT en el gremio, lo que me implicó una gestión minuciosa para identificar cómo debía implementarlo en el consumo de los servicios y cumplir con todas las medidas de seguridad establecidas por la empresa.

5.5. Desarrollo picklist

El primer componente que desarrollé fue el componente padre, el picklist, dicho componente como se ha mencionado con anterioridad, es el responsable de generar el nivel de filtrado para las consultas a los servicios y el encargado de ejecutar la lógica para la visibilidad de usuarios. Para la parte del filtrado, generé una serie de listas de selección (inputs) en un componente aura, mediante la siguiente línea de código en el archivo component:aura (html):

```
<lightning:select  
name="Nombre_de_lista"aura:id="Nombre_id_lista"label="Etiqueta  
_de_la_lista"disabled="Nombre_bandera_control"required="Boolean  
"onchange="Nombre_de_la_función_en_el_controlador" >
```

Esta línea de código recibe como parámetros el **name** y el **aura:id** que en un nuestro caso en particular sirven para poder identificar dicha lista de selección en el controller:aura (javascript), el parámetro **label** que contendrá la etiqueta que va contener nuestra lista, el parámetro **disabled** el cual habilita o inhabilita la lista al usuario dependiendo el caso en el que se encuentre, el parámetro **required** el cual le indicará al usuario si es necesario ingresar, o no, un valor en la lista y por último el parámetro **onchange**, el cual dependiendo un evento que se genera en la lista, este ejecutará una función dentro del controller:aura para generar un proceso o ejecutar un evento. Gráficamente la línea de código anterior se visualiza de la siguiente manera:



Imagen 18. Lista de selección implementada en la Interfaz

Una vez bien identifiqué todas las listas de selección que debía de contener el picklist apégandome a las necesidades de negocio, continué realizando la estructura adecuada

que debía contener el componente, conformándose de la siguiente manera:

*Tipo de producto Adquirente ▼	Tipo de tarjeta Todos ▼	*Fecha de inicio Selecciona 📅	*Fecha de término Selecciona 📅	*División Gobierno División Noreste ▼
*Centro Responsable (CR) Todos ▼	*Unidad Gestora (UG) Todos ▼	Grupo Todos ▼	Número de cliente Todos ▼	Número de afiliación Todos ▼

Imagen 19. Configuración del Picklist para el producto de Adquirente

Cabe mencionar que tuve que adquirir los conocimientos básicos y desarrollar mis habilidades en html y css, para poder dar la apariencia esperada del componente, con la finalidad de que cumpliera con las características del maquetado compartido por el equipo de UX.

Como se comentó en el apartado 5.2.1 el componente lo diseñé de tal manera, que dependiendo el tipo de producto que el usuario llegará a seleccionar, la estructura y las listas de selección que se le presentan al usuario, podrían cambiar de forma dinámicamente dependiendo el caso. Para poder llevar a cabo este proceso dentro del controller:aura realicé un método que fuera capaz de identificar el tipo de producto seleccionado por el usuario, y con ello habilitar o inhabilitar bloques aura:if que contienen las diferentes formaciones de listas de selección como se muestra a continuación:

```
Listas de selección que comparten todos los productos
<aura:if Tipo producto = 4 productos>
  Listas exclusivas para 4 productos
</aura:if>
<aura:if Tipo producto = adquirente>
  Listas exclusivas para Adquirente
</aura:if>
<aura:if Tipo producto = multipagos>
  Listas exclusivas para Multipagos
</aura:if>
```

Imagen 20. Método para mostrar el tipo de Picklist-Producto

Una vez realicé la estructura y la lógica de las listas dinámicas en el componente, proseguí con el llenado de datos para cada una de las listas. Para ello de igual forma mediante las reglas de negocio establecidas, identifiqué cuales iban a recibir la respuesta del servicio de filtros y cuales debían de obtener la información mediante metadata personalizada, esta decisión la tomé a partir del origen de la información y el tipo de dato que se iba a visualizar. En la siguiente tabla se establece lo antes mencionado:

Respuesta de Filtros (Servicio)	Metada Personalizada
<ul style="list-style-type: none"> ● Canal ● Subproducto ● Medio de Pago ● Sucursal ● Ejecutivo ● Grupo ● N.Cliente ● Contrato o Convenio o Afiliación 	<ul style="list-style-type: none"> ● Tipo de producto ● División ● Tipo de tarjeta

Imagen 21. Tabla con los picklist que su información la obtienen con Servicios y Metadatos

Para lograr una conexión efectiva con el *middleware* ASO y obtener respuestas precisas del servicio, desarrollé un controlador en el lado del servidor utilizando clases Apex. Implementé un método GET en estas clases, el cual realiza tanto el consumo del servicio como la gestión de errores en la respuesta. Aunque Apex comparte similitudes con JavaScript, me encontré con características y restricciones propias de esta tecnología, lo que requirió una curva de aprendizaje considerable para dominar su sintaxis y particularidades. Esto fue crucial para mantener un código funcional, limpio y alineado con las mejores prácticas de programación.

Otro reto fue el desarrollo de la metadata personalizada para las listas de selección, que servía como base para las consultas en la nube. Configuré la metadata personalizada en la plataforma, lo cual implicó la creación de clases Apex adicionales para realizar consultas eficientes y asegurar la correcta interacción con la información almacenada.

La complejidad de esta etapa radicó principalmente en el diseño y validación del flujo de trabajo dentro de las clases Apex, donde tuve que aplicar principios de programación

orientada a objetos y una adecuada estructura de datos. Estos principios no solo optimizaron la legibilidad y claridad del código, sino que también mejoraron la eficiencia de procesamiento de la información. Así, cada interacción, ya fuera con los servicios o con la metadata, se ejecutaba de manera óptima, reduciendo tiempos de carga y errores potenciales en la entrega de datos.

Una vez que creé el proceso en las clases apex, la creación de la matriz de pruebas fue clave para asegurar la calidad y estabilidad del desarrollo antes de su implementación en producción. Al configurar esta matriz, realicé diversos llamados a los servicios para generar diferentes tipos de respuestas, incluyendo posibles errores y casos límite. Esta estrategia me permitió probar a fondo la lógica de gestión de errores, verificando que todos los posibles escenarios respondieran de manera adecuada y eficiente.

Este enfoque de pruebas exhaustivas no solo permitió validar que cada casuística funcionara conforme a lo esperado, sino que también proporcionó un método efectivo para identificar rápidamente cualquier problema que pudiera surgir en producción. La implementación de esta matriz no fue un proceso sencillo, ya que requirió una planificación cuidadosa para cubrir todos los posibles escenarios y una comprensión profunda de la lógica de negocio y de los servicios que interactuaban con el sistema.

Gracias a esta matriz de pruebas, pude minimizar el riesgo de incidencias y asegurar que el flujo de la aplicación fuera robusto. Además, este proceso me dio una perspectiva valiosa sobre cómo anticipar posibles fallos y establecer mecanismos de detección temprana, habilidades fundamentales para el desarrollo de soluciones escalables y de alta calidad.

Para la parte de la visibilidad tuve que generar un proceso que fuera capaz de obtener la información del usuario activo en el aplicativo, de esta forma, una vez identificado el tipo de usuario, su información se procesaría mediante un método en el controller:aura, el cual dependiendo el caso aplicaría las reglas de negocio que se explican en el apartado 5.1. Con esto cada uno de los diferentes 4 tipos de usuarios tendrían un determinado nivel de manipulación dentro del tablero. Cabe mencionar que, para el tipo de usuario Asesor Cash diseñé un proceso un poco distinto a los demás, ya que este tipo de perfil debería

de obtener información específica de los campos Ejecutivo y Sucursal en comparación a sus homólogos que obtienen información general. Para poder determinar este nivel de perfilado, como se ha mencionado en subcapítulos anteriores, tuve que sumergirme en el área de negocio para poder entender cuales eran características y funciones que diferencian a cada perfil y con ello establecer el nivel de perfilado propicio con la intención de que los perfiles consulten y visualicen la información que les corresponde. Dicha definición no era algo que me correspondía a mi como desarrollador, pero al tomar una postura de liderazgo me ví comprometido a comprender con mayor precisión cómo se comporta el negocio.

5.6. Desarrollo componentes contenedor, comisiones, facturación

El componente contenedor como su nombre lo dice, lo creé de tal manera que fuera un componente que contuviera a los demás componentes en el flujo y pudiera realizar la gestión de la selección del tipo de tablero que desee desplegar el usuario. A pesar de que este componente no llega a tener una lógica tan fuerte como el componente padre, dicho componente cuenta con un método que le ayuda obtener, mediante un enlace datos, los parámetros de entrada seleccionados en el picklist para el consumo del servicio summary, de igual manera cuenta con un método que realiza el dinamismo de la selección de los diferentes tipos de tablero, para con esto, llamar dependiendo el tipo de tablero, al componente comisiones, transacciones o facturación. Otro método que diseñé en este componente, fue el método para la selección de los tipos tableros que se deben de habilitar o inhabilitar dependiendo el tipo de producto seleccionado por el usuario.

Una vez que el usuario ha identificado el tipo de tablero que desea visualizar, el flujo continúa conforme se ha mencionado anteriormente en uno de los siguientes tres componentes: comisiones, transacciones y facturación. La función de estos componentes se centra principalmente en definir la estructura que deben adoptar los componentes reutilizables y los parámetros de entrada específicos en sus distintas llamadas al servicio "summary". Para lograrlo, desarrollé los tres componentes junto con sus respectivas características y dimensiones.

5.7. Desarrollo componente gestor e implementación de componentes reutilizables

Desarrollé un proceso en el componente gestor para administrar el consumo del servicio "summary" en sus diferentes puntos finales (endpoints). Esto era necesario ya que cada gráfica, tabla y tarjeta en los tableros requería una respuesta única. Dentro del gestor, además de enviar el objeto con los parámetros de entrada provenientes del "picklist", agregué parámetros específicos según el tipo de respuesta deseada.

Al igual que con el servicio de filtros, creé un controlador del lado del servidor utilizando clases Apex. La diferencia radica en que este servicio se debe consumir mediante un método POST, lo que me implicó realizar las configuraciones necesarias y desarrollar la lógica adecuada para su consumo. Es importante mencionar que también implementé la gestión de errores correspondiente para este proceso.

Otra cosa que se controla mediante el componente gestor es el llamado correspondiente a los componentes reutilizables, para ello desarrollé un método capaz de identificar el tipo de llamado que se le solicitaba al componente y así seleccionar al componente tabla, gráfica o imagen de fondo. Los componentes reutilizables a su vez requieren ciertos requisitos para poder utilizarlos, como el ordenamiento de la información (la respuesta del servicio acorde a la estructura que acepta el componente) que se lleva a cabo mediante un proceso implementado en las clases apex, esto debido a que todas las salidas del servicio summary devuelven la respuesta en una estructura tipo JSON donde los datos no se encuentran preparados para introducirlos directamente como parámetros de entrada para los componentes reutilizables, cabe mencionar que cada tipo de gráfica y tabla, requieren una estructura distinta y específica.

El componente de imagen de fondo será mostrado únicamente si, durante la gestión de errores del consumo del servicio (summary), identifica algún tipo de error. De esta manera, el usuario podrá percatarse de que ha ocurrido algún error durante el proceso de la información y de esta forma podrá solicitar la ayuda pertinente del administrador de la plataforma, si así lo requiere la situación.

5.8. Despliegue de desarrollo a ambientes productivos

La fase de migración y delegación del aplicativo al ambiente de producción fue esencial para garantizar un sistema confiable y alineado con los estándares de la empresa multinacional. Este proceso me exigió no solo una adaptación técnica, sino también un entendimiento detallado de los lineamientos y procedimientos de seguridad y calidad que deben cumplirse.

Al tratarse de un entorno compartido donde interactúan múltiples desarrollos de diversas áreas, comprendí la importancia de las pruebas de no afectación, las cuales aseguraban que mi implementación no alterara el funcionamiento de otros aplicativos existentes. Esto implicaba realizar pruebas funcionales exhaustivas y analizar las interacciones con otros componentes del CRM.

Además, esta fase fue una gran oportunidad para profundizar en herramientas de control de versiones y análisis de código, fundamentales para detectar problemas de calidad y seguridad. Con Chimera, pude identificar posibles vulnerabilidades en el código, mientras que Sonar me ayudó a detectar y corregir problemas de "small codes" o códigos poco óptimos. Esto me permitió garantizar un código robusto y seguro antes de su despliegue. Aunque técnicamente no fue la etapa más desafiante, su complejidad radicó en el cumplimiento de las múltiples validaciones y en obtener las autorizaciones necesarias para cada implementación en producción. Cada cambio en el sistema requería alrededor de un mes para completarse, lo que sumó considerablemente en el tiempo total del proyecto, extendiéndose a lo largo de siete meses en procesos de instalación técnica. Este período prolongado subrayó la importancia de la planificación meticulosa y la gestión de tiempos en proyectos de gran escala.

6. RESULTADOS

Durante este capítulo se abordarán los resultados obtenidos a lo largo de la estancia en el proyecto Incentives Improvement Mayorista, se mostrará y visualizará el tablero obtenido durante el proyecto y se abordará el valor que este aplicativo dio a la empresa y a los usuarios finales que van a utilizar dicha herramienta.

6.1. Resultados de sección de filtros

El picklist fue uno de los componentes que contaron con un nivel alto de complejidad, debido a todas las necesidades de negocio con las que debía contar. Al ser el primer componente que se realizó al momento de llegar al proyecto, dicho componente sufrió de una gran cantidad de iteraciones y replanteamientos para poder conseguir el dinamismo y la lógica que el PO esperaba de este. Cabe mencionar que dicho componente lo construí desde cero, debido al nivel de personalización requerida. El componente picklist que se usa actualmente en el tablero cuenta con las siguientes tres vistas:

- Adquirente

The screenshot shows a filter interface for 'Adquirente'. It features several dropdown menus: '* Tipo de producto:' (Adquirente), 'Tipo de tarjeta:' (Todos), '* Desde:' (Seleccione), '* Hasta:' (Seleccione), and '* División:' (Todos). Below these are 'Centro Responsable:' (Todos), 'Unidad Gestora:' (Todos), 'Grupo:' (Todos), 'Número de cliente:' (Todos), and 'Número de Afiliación:' (Todos). At the bottom right, there are buttons for 'Limpiar filtros' and 'Consultar'.

- 4 productos

The screenshot shows a filter interface for '4 productos'. It features several dropdown menus: '* Tipo de producto:' (CIE), 'Canal:' (Todos), '* Desde:' (Seleccione), '* Hasta:' (Seleccione), and '* División:' (Todos). Below these are 'Centro Responsable:' (Todos), 'Unidad Gestora:' (Todos), 'Grupo:' (Todos), 'Número de cliente:' (Todos), and 'Convenio:' (Todos). At the bottom right, there are buttons for 'Limpiar filtros' and 'Consultar'.

- Multipagos

The screenshot shows a filter interface for 'Multipagos'. It features several dropdown menus: '* Tipo de producto:' (Multipagos), 'Subproducto:' (Todos), 'Medios de Pago:' (Todos), '* Desde:' (Seleccione), and '* Hasta:' (Seleccione). Below these are 'División:' (Todos), 'Centro Responsable:' (Todos), 'Unidad Gestora:' (Todos), 'Grupo:' (Todos), 'Número de cliente:' (Todos), and 'Contrato:' (Todos). At the bottom right, there are buttons for 'Limpiar' and 'Consultar'.

De igual forma se pudo obtener la visibilidad para los diferentes usuarios que utilizarán el tablero, dándonos las siguientes vistas para cada uno de estos cuando los diferentes perfiles desean realizar una consulta a los tableros solamente con los parámetros

obligatorios, que dentro del picklist se marcan con un asterisco rojo en la parte superior izquierda de las listas:

- Perfil Cash Management

- Perfil ASI

- Perfil Subdirector Cash

- Perfil Staff

Podemos apreciar que el perfil con menos libertad para consumir información es el Asesor Cash, ya que este no podrá realizar consultas a los tableros, si no ingresa como mínimo el valor de un CR, una vez ingresado un valor en esta lista de selección por el usuario, el botón de consultar se habilita, dándole la capacidad de seguir con el proceso de consulta.

Para los perfiles ASI y Subdirector Cash a comparación del perfil antes mencionado el perfil Cash, cuentan con un nivel más alto de visibilidad. Para que puedan realizar alguna consulta, solamente será necesario que ingresen un valor en la lista de selección de división para que el botón de consultar se pueda habilitar como se muestra en las imágenes anteriores. Y por último pero no menos importante tenemos al perfil Staff, que es el perfil con mayor visibilidad, ya que este tipo de perfil podrá realizar consultas solamente seleccionando el tipo de producto.

Otra de las funcionalidades presentadas en dicho componente fue el consumo tanto del servicio de filtros como las consultas de la metadata personalizada. Con los cuales el usuario tendrá la capacidad elegir el nivel de filtrado que necesite en el segundo apartado del tablero. A continuación se muestra el caso en el que el usuario staff desee obtener el nivel de filtrado más alto eligiendo un valor hasta la lista de selección de “convenio”:

The screenshot shows the 'Tablero Cash' interface. At the top, it displays the user's role as 'STAFF'. Below this, there are several filter fields arranged in two rows. The first row includes: '*Tipo de producto:' (with a dropdown menu), 'Canal:' (with a dropdown menu), '*Desde:' (with a date input field showing '2/05/2023'), '*Hasta:' (with a date input field showing '4/09/2023'), and '*División:' (with a dropdown menu). The second row includes: 'Centro Responsable:' (with a dropdown menu), 'Unidad Gestora:' (with a dropdown menu), 'Grupo:' (with a dropdown menu showing '15'), and 'Número de cliente:' (with a dropdown menu). Below the filter fields, there are two buttons: 'Limpiar filtros' and 'Consultar'. At the bottom of the panel, there is a legend with a red square labeled 'Servicio' and a blue square labeled 'Metadata'.

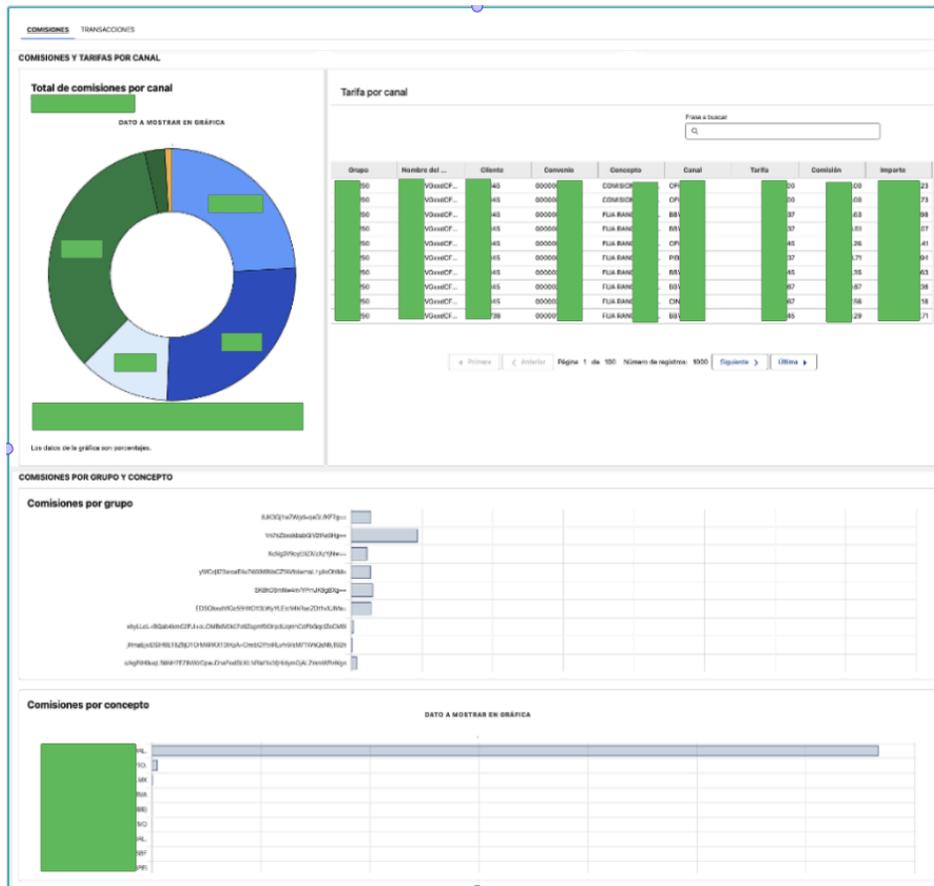
Cabe mencionar que el ejemplo mostrado anteriormente, se presenta cuando el usuario selecciona un producto que pertenezca al hito de 4 productos, pero sin importar este punto, el comportamiento del componente se ajusta al tipo de consulta que el usuario desee realizar.

6.2. Resultados de sección gráfica

Una vez el usuario selecciona el tipo de filtrado que desee realizar y ejecuta el proceso de consulta en el componente picklist, el usuario será capaz de visualizar la sección gráfica. Como se sabe, en esta sección coexisten los componentes; contenedor, comisiones, transacciones, facturación, gestor, tabla, gráfica e imagen de fondo. Todos estos

componentes en conjunto dieron como resultado las siguientes interfaces gráficas, que cabe recalcar, el tipo de interfaz que se le presente al usuario depende directamente del tipo de producto seleccionado.

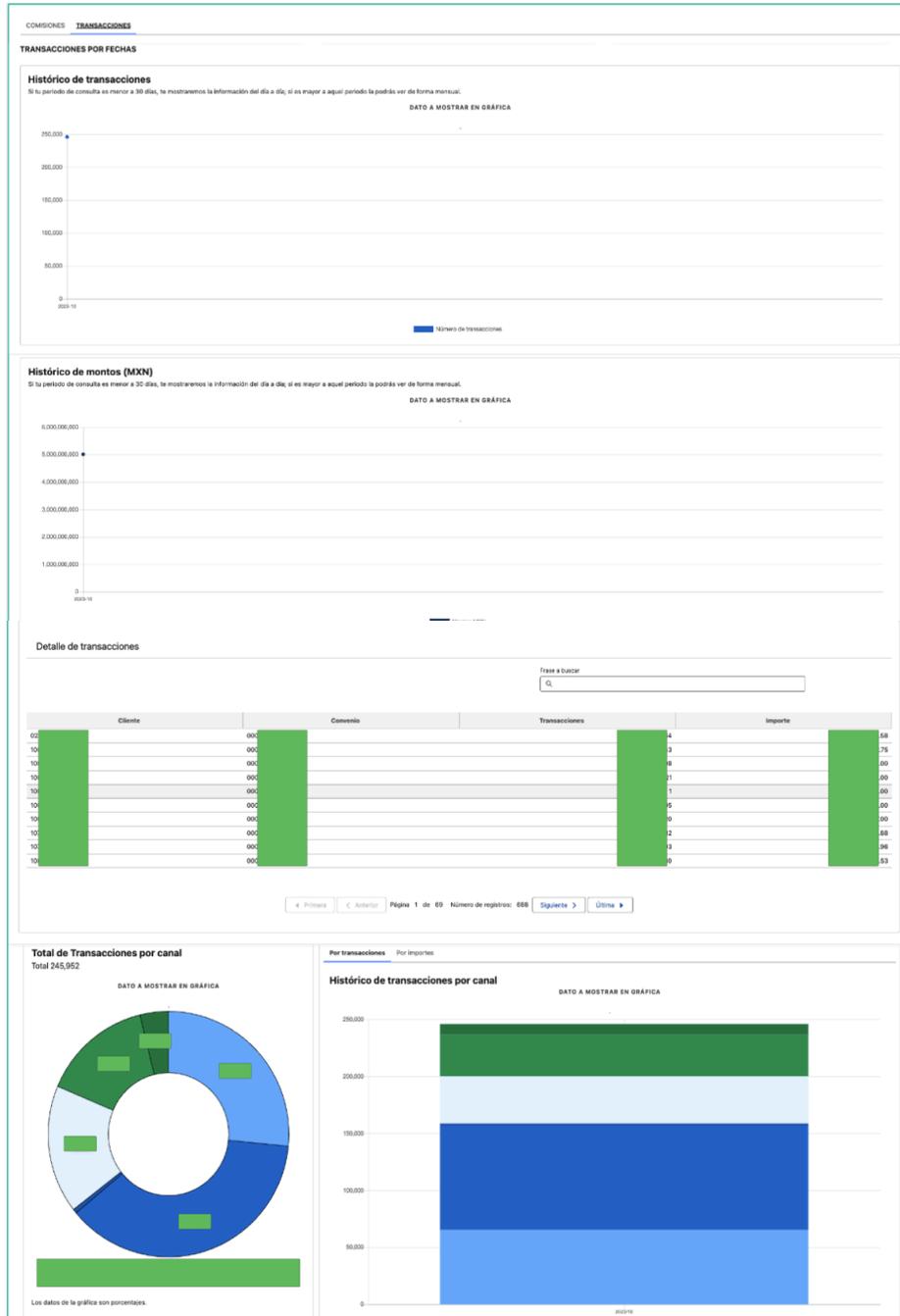
- ☐ Gráficas de comisiones presentes en el hito de Multipagos y 4 productos



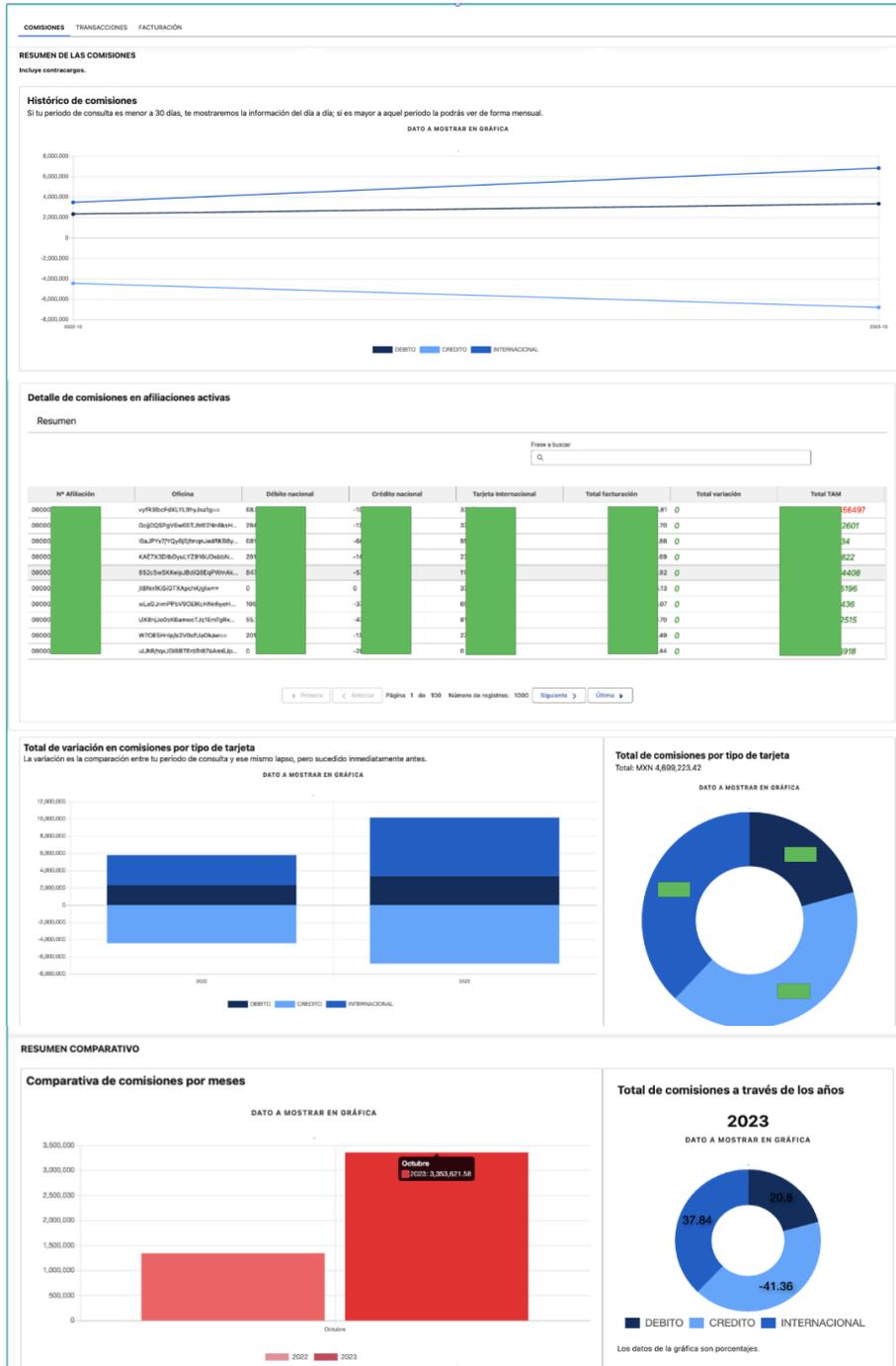
Como podemos observar este tablero, tablero de comisiones contiene un gráfica de tipo dona, dos gráficas de barras horizontales (componente Gráfica) y una tabla que contiene (componente Tabla). Este tablero se va a presentar, si y sólo si, el usuario elige un producto perteneciente a los 4 productos o el producto multipagos. Es importante mencionar, que el usuario tiene la capacidad dentro del tablero de buscar información específica y concreta de cualquier cliente y escoge la tab de comisiones que está

presente en el componente contenedor.

- Gráficas de transacciones presentes en el hito de Multipagos y 4 productos



- Gráficas de comisiones, transacciones y facturación presentes en el hito de adquirente



El tablero de transacciones contiene dos gráficas de puntos, una gráfica de dona, dos gráficas de barras apiladas (componente Gráfica) y una tabla de registros (componente Tabla) que de igual manera que el componente comisiones este tablero se va a presentar, si y sólo si, el usuario elige un producto perteneciente a los 4 productos o el producto multipagos y escoge la tab de transacciones.

Y los tableros de comisiones_adquirente, transacciones_adquierente y facturación que le presentarán al usuario dos gráficas de dona, dos gráficas de barras apiladas, una gráfica de puntos (componente Gráfica) y una tabla de registros (componente Tabla) que sólo se presentarán al usuario, si y sólo si, el usuario selecciona al producto adquirente. Cabe mencionar que los tres tipos de tablero presentes en esta casuística, contendrán los mismos elementos, pero se reflejarán con distinta información y el usuario tendrá la capacidad de elegir cualquiera de estos mediante las tabs presentes en el componente contenedor.

6.3. Resultados del aplicativo para la empresa

El aplicativo resultó tener un gran efecto en la banca desde los primeros meses, ya que este ayudó a tener un 35 % de mejora en la atención y seguimiento de los procesos de los clientes. Teniendo en cuenta esta particularidad, cumplí directamente con uno de nuestros tres estandartes dentro de la empresa “El cliente es primero”. Para poder realizar una estimación equivalente monetariamente, se realizó un análisis comparativo entre el porcentaje de las oportunidades generadas a partir de que el aplicativo estuvo en manos de los especialistas de negocios y los primeros tres trimestres pero del año anterior y se pudo visualizar un crecimiento del 14.5% en oportunidades ganadas.

Cabe mencionar que el aplicativo tuvo un impacto directo en la forma en la que los especialistas analizan la información de los clientes, ya que anteriormente para poder realizar un dictamen tenían que consultar la información desde diferentes aplicativos hasta concentrados de archivos en excel para poder dar una resolución, pero ahora con dicho tablero pueden visualizar la información precisa y concisa desde un mismo

aplicativo, lo que disminuye tiempos de operación de semanas a días de ejecución en el análisis.

Después de una larga gestión de casi un año y seis meses, donde, llevé a cabo una gran variedad de actividades en equipo bajo la metodología ágil, toma de decisiones, despliegues del desarrollo al ambiente de producción, resolución de problemas e implementación de mis habilidades técnicas y blandas en el proyecto Incentives Improvement Mayorista, se logró la meta de entregar el aplicativo Tablero Cash funcional con los tres hitos definidos por la necesidad de negocio.

7. CONCLUSIONES

A lo largo del proyecto Tableros Cash, tuve la oportunidad de enfrentar y superar una serie de desafíos técnicos, metodológicos y de gestión que fueron fundamentales para mi desarrollo profesional. Este proyecto, en particular, no solo me permitió aplicar los conocimientos adquiridos a lo largo de mi carrera, sino también expandirlos en nuevas áreas, con el fin de entregar una solución robusta que cumpliera con las necesidades del negocio. Al mirar hacia atrás, puedo decir con certeza que este fue un proceso enriquecedor que consolidó mis competencias tanto técnicas como blandas, las cuales fueron determinantes en la exitosa ejecución del proyecto. Sin embargo, no fueron únicamente estas habilidades las que contribuyeron al éxito, sino también las competencias blandas que adquirí en diferentes proyectos a lo largo de la carrera, como el concurso de Innovación de Nikola Tesla en mi primer semestre, donde en conjunto con un compañero de materia, diseñamos un motor stirling que obtenía su fuente de energía mediante los rayos del sol focalizados y unificados con la ayuda de una antena parabólica, este reto me ayudó a reafirmar mis primeros acercamientos al trabajo en equipo, lo que me permitió plasmar de una mejor forma mis ideas y propuestas para la solución de problemas y por primera vez en toda mi vida pude realizar y aplicar ingeniería para poder diseñar y crear un sistema basado en cálculos matemáticos, leyes y teoremas científicos, esto provocó en mí un interés muy genuino por afrontar la gran cantidad de problemas que suelen presentarse en nuestro entorno.

En tercer semestre ingresé a la escudería UNAM Motorsport, tomando el cargo de líder del sistema fuel and cooling tanto para el carro eléctrico y el de combustión por dos años y medio. Posteriormente a pesar de que mis últimos semestres los llevé a cabo en casa por la pandemia, gracias al gran trabajo y adaptabilidad que tuvieron mis profesores, logré mejorar tanto mi comunicación a distancia y trabajar en proyectos desde casa con equipos multidisciplinarios, haciendo mención por ejemplo de mi clase de temas selectos de Mecatrónica, donde se me introdujo por primera vez con el concepto "Scrum" y se nos dejó un proyecto que requirió de una gran variedad de conceptos adquiridos en las diferentes materias de la carrera.

Gracias a todo lo mencionado anteriormente, desarrollé habilidades fundamentales como liderazgo, trabajo en equipo, capacidad de adaptabilidad, gestión eficiente del tiempo, comunicación efectiva y una mentalidad de crecimiento. Estas habilidades fueron determinantes para asumir un rol protagónico dentro del equipo de desarrollo de la empresa, liderando el proyecto desde su inicio.

Desde el inicio, tuve que abordar el desarrollo de un componente altamente personalizado, el Picklist, que requería una lógica compleja de filtros dinámicos y personalización para adaptarse a las diversas necesidades de los diferentes usuarios. La dificultad principal residió en la demanda de crear un componente desde cero, lo que me obligó a dejar de lado componentes reutilizables y extender considerablemente los tiempos de desarrollo. Sin embargo, a pesar de este reto, el proceso de crear el componente me brindó la oportunidad de perfeccionar mi capacidad para resolver problemas y tomar decisiones sobre cómo manejar las interacciones entre los diferentes servicios y componentes del sistema. La implementación de los servicios externos, junto con la gestión de los metadatos en la plataforma, representó un paso significativo en el dominio de nuevas tecnologías y técnicas, lo que implicó superar un considerable desafío de aprendizaje relacionado con el uso de clases Apex, un área nueva para mí.

A lo largo del proyecto, mi enfoque no solo estuvo en la parte técnica, sino que también fue clave mantener una comunicación constante con el equipo y con el negocio. El uso del Scrum me permitió gestionar el proyecto de forma ágil y adaptable, realizando entregas continuas y asegurando que el desarrollo estuviera alineado con los objetivos y expectativas del negocio. El desafío de cumplir con los plazos, mientras gestionaba un proceso de aprendizaje acelerado, me exigió un manejo eficiente de mi tiempo y la capacidad de tomar decisiones rápidas, todo mientras mantenía el enfoque en entregar un producto de alta calidad.

El trabajo en conjunto con el equipo de UX también fue crucial para lograr una experiencia de usuario que no solo fuera intuitiva, sino también eficiente y alineada con los requerimientos del negocio. Estas sesiones me permitieron no solo entender la importancia de un diseño coherente, sino también aplicar nuevos conocimientos en herramientas como Figma para ajustar las interfaces y cumplir con las normativas internas

de la empresa. Además, mi participación activa en la definición de la arquitectura del sistema y los procesos de migración al ambiente de producción, me permitió profundizar en la gestión de versiones y en la implementación de pruebas funcionales para garantizar que los cambios no afectaran a otros sistemas productivos. Fue aquí donde comprendí la relevancia de la robustez del proceso de migración y cómo la metodología y las pruebas previas al despliegue son fundamentales para evitar posibles incidencias en producción.

Uno de los retos más significativos fue el proceso de migración y delegación del aplicativo al ambiente de producción. A pesar de que esta etapa no presentaba un reto técnico de alto nivel, involucraba una serie de procedimientos y validaciones necesarios para asegurar el correcto funcionamiento del sistema en un entorno productivo. Durante este proceso, aprendí a gestionar el control de versiones, a realizar pruebas de no afectación y a utilizar herramientas de análisis de código como Chimera y Sonar, las cuales fueron esenciales para identificar posibles vulnerabilidades y malas prácticas en el desarrollo. Si bien esta fase consumió mucho tiempo y esfuerzo, el aprendizaje fue fundamental para poder implementar cambios futuros de manera más ágil y con mayor certeza de que no se generarían problemas en el entorno productivo.

La colaboración constante con las diferentes partes del equipo y la supervisión de los avances en el proyecto también me permitió desarrollar habilidades clave como liderazgo, comunicación efectiva y capacidad de adaptación. El proyecto no solo implicaba una serie de retos técnicos, sino también la gestión de un equipo diverso, la toma de decisiones estratégicas y la capacidad de adaptarse rápidamente a las necesidades cambiantes del negocio. Como líder del proyecto, tuve que gestionar las expectativas del negocio, guiar al equipo en la implementación de las soluciones más adecuadas y asegurarme de que se cumplieran los plazos establecidos, lo que me brindó una valiosa experiencia en la gestión de proyectos ágiles.

Gracias a todo lo mencionado anteriormente, pude hacer frente a todos los desafíos que surgieron durante el proceso de desarrollo y entregar un producto de alta calidad. Este éxito no solo se reflejó en la calidad técnica del aplicativo, sino también en la manera en que logré involucrarme en todas las fases del proyecto, desde el análisis inicial hasta la migración a producción. Los reconocimientos recibidos en el equipo y los ascensos obtenidos en un tiempo relativamente corto, son una muestra del impacto positivo que

tuvo mi trabajo en el proyecto y en el desarrollo del equipo. Actualmente, como Gerente de Desarrollo de Soluciones de Software, puedo mirar atrás y ver cómo cada desafío enfrentado en el proyecto de Tablero Cash, que fue una experiencia transformadora, no solo a nivel técnico, sino también en cuanto a habilidades de gestión y liderazgo. Cada desafío fue una oportunidad para aprender, adaptarme y mejorar mis competencias. Hoy en día, no solo tengo una comprensión más profunda de las tecnologías con las que trabajé, sino también una mejor capacidad para liderar equipos y gestionar proyectos complejos de manera efectiva. Sin duda, este proyecto marcó un antes y un después en mi carrera, consolidando mis habilidades y sentando las bases para futuros desafíos en mi camino profesional.

8. REFERENCIAS

- [1] Edgar Juárez. (2021). *Banca Digital ya es un tema de educación financiera*. [Online] <https://www.eleconomista.com.mx/sectorfinanciero/Banca-digital-ya-es-un-tema-de-educacion-financiera-20210926-0036.html> [Acceso:].
- [2] Edgar Juárez. (2022). *Digitalización, clave para lograr una mejor inclusión financiera*. [Online] Disponible en: <https://www.eleconomista.com.mx/sectorfinanciero/Digitalizacion-clave-para-lograr-una-mayor-inclusion-financiera-20220522-0039.html> [Acceso:].
- [3] Eduardo Osuna. (2018). Información corporativa. [Online] Disponible en: <https://www.bbva.com/es/informacion-corporativa/#historia-de-bbva> [Acceso:].
- [4] Carlos Torres Vila (2022). BBVA México, 90 años de transformar la banca y crear oportunidades. [Online] Disponible en: <https://www.elfinanciero.com.mx/opinion/eduardo-osuna/2022/12/14/bbva-mexico-90-anos-de-transformar-la-banca-y-de-crear-oportunidades/> [Acceso:].
- [5] BBVA (2023). Origen de la metodología Agile. [Online] Disponible en: <https://www.bbva.es/finanzas-vistazo/agile/metodologia-agile/origen-metodologia-agile.html> [Acceso:].
- [6] Instituto Agile (2022). ¿Qué es Scrum? Las raíces del término Scrum [Online] Disponible en: <https://www.institutoagile.com/post/el-t%C3%A9rmino-scrum-aplicado-al-desarrollo-de-proyectos> [Acceso:].
- [7] Genaro Cuofano (2023). Crystal Agile Framework En Pocas Palabras. [Online] Disponible en: <https://fourweekmba.com/es/marco-%C3%A1gil-de-cristal/> [Acceso:].
- [8] Salesforce (2022). BBVA implanta una plataforma para gestionar su huella de carbono en la nube. [Online] Disponible en: https://cincodias.elpais.com/cincodias/2022/07/25/companias/1658771364_834056.html#:~:text=BBVA%20ha%20contratado%20a%20Salesforce,carbono%20y%20eliminar%C3%A1%20procedimientos%20manuales.[Acceso:].
- [9] Salesforce (2023). Acelera tu crecimiento en Sales Cloud. [Online] Disponible en: <https://www.salesforce.com/es/products/sales-cloud/overview/#:~:text=%C2%BFQu%C3%A9%20es%20Sales%20Cloud%3F,ventas%2C%20as%C3%AD%20como%20la%20productividad> [Acceso:].
- [10] Salesforce (2023). Apex Code Overview. [Online] Disponible en: https://help.salesforce.com/s/articleView?id=sf.code_about.htm&type=5 [Acceso:].
- [11] Salesforce (2023). Crear configuraciones personalizadas. [Online] Disponible en: https://help.salesforce.com/s/articleView?id=sf.cs_about.htm&type=5 [Acceso:].

IMÁGENES:

- [1] BBVA MX (2023). *Estructura Directiva* [Imagen] Disponible en:
<https://www.bbva.mx/personas/informacion-corporativa/estructura-directiva.html>
[Acceso:].
- [2] ED Team (2020). *¿Cuáles son las áreas de programación?* [Imagen] Disponible en:
<https://ed.team/blog/cuales-son-las-areas-de-la-programacion> [Acceso:].
- [3] Platzi (2020). *¿Qué es HTML, CSS, JavaScript?* [Imagen] Disponible en:
<https://platzi.com/blog/que-es-html-css-javascript/> [Acceso:].
- [4] Salesforce (2023). *Todo lo que necesitas saber sobre la Gestión de Relación con los Clientes.* [Imagen] Disponible en:
<https://www.salesforce.com/mx/crm/#que-hace-un-crm-scroll-tab> [Acceso:].
- [5] Salesforce (2023). *Skill up Lightning Fast* [Imagen] Disponible en:
<https://developer.salesforce.com/quests/web-components> [Acceso:].
- [6] Andrew Cook (2023). *Salesforce Flow Builder vs. Apex.* [Imagen] Disponible en:
<https://www.salesforceben.com/salesforce-flow-builder-vs-apex/> [Acceso:].