



---

# TECNOLOGICO DE ESTUDIOS SUPERIORES DE ECATEPEC

DIVISION DE INGENIERIA EN SISTEMAS COMPUTACIONALES

REFACTORIZACIÓN BACKOFFICE SYSTEM PAGO DIARIO

T E S I S

QUE PARA OBTENER EL TÍTULO DE  
INGENIERO EN SISTEMAS COMPUTACIONALES

P R E S E N T A N:

**PÉREZ ROJAS ISSAC**

DIRECTOR: M. EN C. ISRAEL ISAAC GUTIÉRREZ VILLEGAS  
CO-DIRECTOR: M. EN C. JAVIER NORBERTO GUTIÉRREZ VILLEGAS  
CO-DIRECTOR: I.S.C FRANCISCO JAVIER HERNÁNDEZ BARAJAS

ECATEPEC DE MORELOS, EDO. DE MÉXICO, 2022.

## **Agradecimientos**

### **A Dios**

*Por la fortaleza que nos da para superar cada obstáculo y por las ganas de luchar día a día por nuestros sueños.*

### **A mis padres**

*Gracias, por sus consejos y sabiduría, brindándome cualquier apoyo y animo posible. Por los valores que me han inculcado y por brindarme la oportunidad de tener una excelente educación en el transcurso de mi vida.*

### **A mi hermano**

*Por estar conmigo durante esas largas noches, acompañándonos mutuamente durante el desarrollo de nuestros proyectos, por brindarme consejos cuando tenía dudas.*

### **Amigos y compañeros**

*Gracias por este viaje a su lado, hoy culmina esta maravillosa aventura el cual estuvo lleno de horas de trabajo y alegría a lo largo de nuestra formación. Sus aportes fueron valiosos y necesarios para hacer mi etapa universitaria un trayecto de vivienda que nunca olvidare.*

**Pérez Rojas Issac**

---

 <p>GOBIERNO DEL ESTADO DE MÉXICO</p>	<p><b>DICTAMEN DE LIBERACIÓN Y CUMPLIMIENTO DE REQUISITOS ACADÉMICOS PARA TITULACIÓN</b></p> <p>FO-TESE-DA-78 TIT05</p>	
--	---	---

**DIRECCIÓN ACADÉMICA**  
DIVISIÓN INGENIERÍA EN SISTEMAS COMPUTACIONALES

**TES\_PRF-20221-004**  
17 DE JUNIO DEL 2022

**Datos generales**

Nombre del egresado:

<b>PÉREZ</b>	<b>ROJAS</b>	<b>ISSAC</b>
Apellido paterno	Apellido materno	Nombres (s)

Matrícula: 201720929 Teléfono: 55-8871-6928 Móvil: 55-2562-9664 Correo Electrónico: issacp130@gmail.com 201720929@tese.edu.mx

Opción de titulación: **TESIS PROFESIONAL**  
Únicamente en las opciones I, II, VII y X

PROYECTO: **REFACTORIZACIÓN BACKOFFICE SYSTEM PAGO DIARIO**

ASESOR (ES): **M. EN C. ISRAEL ISAAC GUTIÉRREZ VILLEGAS, M. EN C. JAVIER NORBERTO GUTIÉRREZ VILLEGAS, I.S.C. FRANCISCO JAVIER HERNÁNDEZ BARAJAS**

**Dictamen:**

Con base en la solicitud de opción de titulación No. ( 1 ) correspondiente al periodo escolar ( 2022 - 1 ), presentada por el alumno de referencia en fecha **06 DE ENERO DEL 2022** y después de haber sido desarrolladas las actividades requeridas en dicha opción, el Consejo Académico Divisional, conformado por los integrantes que firman al calce, ha determinado que Si ( **X** ) NO ( ) han sido cubiertos en su totalidad los requisitos que permitan a esta División Académica liberar al alumno de los trámites académicos para tal efecto e iniciar el trámite de titulación en la Unidad de Registro y Certificación de este Tecnológico de Estudios Superiores de Ecatepec.

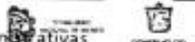
**Observaciones:**

**Consejo Académico Divisional**

 M. EN C. IRVING GARDUÑO ALCOCER GUILLERMO Presidente de la Academia de Ciencias Básicas	 LIC. MARIA REBECA GARCIA PONCE Presidenta de la Academia de Ciencias Sociales
 M. EN D. J. ANAHÍ FLORES NUÑEZ Presidenta de la Academia de Ciencias Económico y Administrativas	 M. EN C. ROSA MARÍA HERNÁNDEZ ROJAS Presidente de la Academia de Ciencias de la Ingeniería
 I.S.C. FRANCISCO JAVIER HERNÁNDEZ BARAJAS Presidente de la Academia de Diseño de la Ingeniería	 M. EN C. XÓCHITL RAQUEL WONG COHÉN Academia de Ingeniería Aplicada



**LICENCIATURA**  
DIVISIÓN DE INGENIERÍA EN SISTEMAS COMPUTACIONALES



DIRECCIÓN ACADÉMICA

Elaboró:  M. EN C. MERCEDES FLORES FLORES SECRETARIO DEL CONSEJO ACADÉMICO DIVISIONAL	Revisó y Visto Bueno:  ING. JOSÉ LUIS CHÁVEZ ROJAS ENCARGADO DE LA DIVISIÓN ISC
---	--

## Resumen

Con más de 5 años de experiencia “*Productos para Empleados SAPI DE C.V.*” provee de instrumentos financieros, naciendo como una empresa completamente tradicional se ha tenido que transformar para ser una empresa 100% tecnológica. Hoy en día los bancos, así como la banca electrónica han incorporado el concepto de “Salary On Demand”. Que contempla el desarrollo de un aplicativo que ofrece a colaboradores y trabajadores la posibilidad de consultar y disponer de su saldo de manera anticipada. Permitiendo a la empresa, gestionar las transacciones que realizan los trabajadores.

El trabajo pretende mejorar al sistema actual, incorporando nuevos módulos denominados como: Cálculo de Saldo, Pago de Servicios y Suscripciones.

La metodología SCRUM se emplea para identificar problemas de usabilidad, análisis completo de la plataforma web de PULPI y juzgar si este cumple con los principios de usabilidad definidos en la heurística y recopilando información obtenida por el usuario con la finalidad de encontrar oportunidades de mejora.

Como resultado, de la aplicación del SCRUM se desarrollaron e implementaron al aplicativo el pago de servicios, créditos de nómina, mejora en el cálculo de saldo de los colaboradores entre otros. Con el objetivo de brindarle a colaboradores y trabajadores una herramienta que permita mejorar el control de sus finanzas.

## Abstract

With more than 5 years of experience "Products for Employees SAPI DE C.V." provides financial instruments, born as a completely traditional company has had to transform to be a 100% technological company. Today banks, as well as electronic banking, have incorporated the concept of "Salary On Demand". That contemplates the development of an application that offers collaborators and workers the possibility of consulting and disposing of their balance in advance. Allowing the company to manage the transactions carried out by the workers.

The work aims to improve the current system, incorporating new modules called: Balance Calculation, Payment of Services and Subscriptions.

The SCRUM methodology is used to identify usability problems, complete analysis of the PULPI web platform and judge whether it complies with the usability principles defined in the heuristics and collecting information obtained by the user in order to find opportunities for improvement.

As a result, from the SCRUM application, the payment of services, payroll credits, improvement in the calculation of the balance of the collaborators, among others, were developed and implemented. With the aim of providing collaborators and workers with a tool that allows them to improve the control of their finances.

## Índice de Tablas y Figuras

### Índice de Tablas

	<b>Pág.</b>
Tabla 1 Roles principales en Scrum	22
Tabla 2 Reuniones en Scrum	22
Tabla 3. Nuevo cálculo de saldo	26
Tabla 4. Ejemplo nuevo cálculo de saldo primera quincenal	26
Tabla 5. Ejemplo nuevo cálculo de saldo segunda quincena	26
Tabla 6. Pruebas de QA	40
Tabla 7. Prueba de sanidad	41

### Índice de Figuras

	<b>Pág.</b>
Figura 1. Diagrama de Ishikawa	3
Figura 2. Visión general de Scrum	23
Figura 3. Cálculo de saldo anterior	25
Figura 4. Nuevo cálculo de saldo	25
Figura 5. Flujo de retiros	27
Figura 6. Pantalla pago de servicios	28
Figura 7. Pantalla telefonía	28
Figura 8. Pantalla ingresar número	29
Figura 9. Flujo suscripciones	30
Figura 10. Flujo general suscripciones	30
Figura 11. Pantalla suscripción	31
Figura 12. Pantalla inicio con suscripción	32
Figura 13. Estado 200	33
Figura 14. Tarjeta suscripción	33
Figura 15. Modal suscripción	34
Figura 16. Servicio	34
Figura 17. Funcionamiento por consola	35
Figura 18. Empresa con suscripción	35
Figura 19. Actualizar suscripción	35
Figura 20. Update consola	36
Figura 21. Actualización suscripción	36
Figura 22. Campos en formulario	37
Figura 23. Prueba unitaria	37
Figura 24. Ejemplo prueba unitaria backend	38
Figura 25. Prueba por medio de Karma	38
Figura 26. Tablero Jira	39
Figura 27. Criterios tareas	39
Figura 28. Pantallas en figma	41

## Índice de Figuras

	<b>Pág.</b>
Figura 29. Pago de servicio en dashboard	42
Figura 30. Recibo pago de servicio	43
Figura 31. Activar suscripción	43
Figura 32. Usuario con suscripción	44
Figura 33. Detalle colaborador suscripción	44

---

## Índice

	Pág.
<b>Introducción</b>	1
<b>Capítulo 1. Marco de Contexto</b>	2
1.1 Introducción	2
1.2 Definición del problema	2
1.3 Justificación	3
1.4 Objetivos	4
1.5 Alcances y limitaciones	4
<b>Capítulo 2. Marco teórico</b>	6
2.1 Introducción	6
2.2 Smartphone	6
2.3 Sistema de información	6
2.4 Sistemas transaccionales	8
2.5 Sistemas de apoyo la toma de decisiones	9
2.6 Sistemas de información estratégicos	9
2.7 Tecnologías de la información	11
2.8 Sistema Operativo	12
2.9 Sistema Operativo “Android”	13
2.10 Lenguaje de programación	13
2.11 Java	14
2.12 Bases de datos	15
2.13 Sistemas gestores de bases de datos	15
2.14 Modelo relacional	16
2.15 Modelo entidad relación	16
2.16 Modelo entidad relación extendida	16
2.17 Normalización de bases de datos	16
2.18 Dispositivos móviles	17
2.19 Aplicaciones Móviles	17
2.20 Metodologías de desarrollo ágil	17
2.21 Administración de Proyectos	19
2.22 SCRUM	19
2.23 Sprint	19
2.24 Aplicación Móvil	19
2.25 Cronograma de Actividades	19
2.26 Retrospectiva	19
2.27 Estatus	20
2.28 El panel de Tareas	20
<b>Capítulo 3. Marco Metodológico</b>	21
3.1 Introducción	21
3.2 Scrum: una metodología ágil	21
3.3 Scrum: Jira Software	23

---

## Índice

	<b>Pág.</b>
<b>Capítulo 4. Desarrollo Metodológico</b>	24
4.1 Introducción	24
4.2 Desarrollo Metodológico	24
4.3 Procedimiento	24
4.4 Diseño	24
4.4.1 Cálculo de saldo	24
4.4.2 Pago de servicios	27
4.4.3 Suscripciones	29
4.5 Pruebas unitarias y pruebas de QA	37
4.5.1 Pruebas unitarias	37
4.5.2 Pruebas QA	39
<b>Capítulo 5. Resultados</b>	42
5.1 Introducción	42
5.2 Análisis e interpretación de resultados	42
<b>Conclusiones</b>	45
<b>Recomendaciones</b>	46
<b>Referencias</b>	47
<b>Anexos</b>	52
<b>Glosario</b>	54

## Introducción

Con más de 5 años de experiencia “*Productos para Empleados SAPI DE C.V.*” provee de instrumentos financieros, naciendo como una empresa completamente tradicional se ha tenido que transformar para ser una empresa 100% tecnológica. Hoy en día los bancos, así como la banca electrónica han incorporado el concepto de “Salary On Demand”. Que contempla el desarrollo de un aplicativo que ofrece a colaboradores y trabajadores la posibilidad de consultar y disponer de su saldo de manera anticipada. Permitiendo a la empresa, gestionar las transacciones que realizan los trabajadores.

El trabajo pretende mejorar al sistema actual, incorporando nuevos módulos denominados como: Cálculo de Saldo, Pago de Servicios y Suscripciones.

Por lo tanto, se plantea la siguiente estructura de la tesis para tratar la problemática antes mencionada:

El *Marco de Contexto* es tratado en el *Capítulo 1*, tiene como objetivo mostrar el problema a resolver, la justificación del desarrollo de dicho proyecto con la finalidad de brindarle a los colaboradores una mejora y nuevos aplicativos, objetivo general y objetivos específicos donde se define las metas a realizar, buscando resolver un área de importancia para la empresa e implementar nuevas herramientas financieras (suscripciones y pago de telefonía móvil), por último se cuenta con alcances y limitaciones del proyecto en el cual se mencionan puntos a beneficiar y limitantes del mismo.

El *Capítulo 2* denominado *Marco Teórico*, se mencionan los conceptos que apoyan al desarrollo de la página web, así como mencionar aplicaciones ya existentes e instituciones que respaldan el proyecto. Además de contar con la definición de conceptos para el desarrollo de los nuevos aplicativos de la APP, haciendo uso de la metodología ágil de Scrum el cual es un marco de trabajo para desarrollo ágil de software.

La metodología ágil de Scrum es trata en el *Capítulo 3* que corresponde al *Marco Metodológico* para el desarrollo de proyectos que requieren mayor rapidez y adaptabilidad en sus resultados. Dando un breve resumen de la historia de Scrum, roles, palabras claves y fases (planificación del sprint, etapa de desarrollo, revisión del sprint y retroalimentación). Además de incluir el software utilizado durante el desarrollo de este proyecto.

En el *Capítulo 4. Desarrollo Metodológico*, se desarrollan las fases de Scrum para el desarrollo del proyecto donde incluye los apartados de procedimiento, diseño, pruebas unitarias de backend y frontend, pruebas de QA y sanidad.

Los resultados obtenidos son mostrados en el *Capítulo 5. Resultados*, asimismo el correcto funcionamiento de la aplicación y el super dashboard.

Finalmente, esta tesis se culmina con las conclusiones y recomendaciones pertinentes.

# Capítulo 1

## Marco de Contexto

### 1.1 Introducción

El presente capítulo denominado Marco de contexto tiene el objetivo de mostrar el problema a resolver, la justificación del desarrollo de dicho proyecto con la finalidad de brindarle a los colaboradores una mejora y nuevos aplicativos, objetivo general y objetivos específicos donde se define las metas a realizar, buscando resolver un área de importancia para la empresa e implementar nuevas herramientas financieras (suscripciones y pago de telefonía móvil), por último se cuenta con alcances y limitaciones del proyecto en el cual se mencionan puntos a beneficiar y limitantes del mismo.

### 1.2 Definición del problema

México es una de las industrias financieras más diversas y robustas de Latinoamérica, con cientos de participantes que van desde grandes bancos internacionales y locales, hasta una explosiva presencia de Fintech.

No obstante, la mitad de la población adulta no tiene una cuenta bancaria y la penetración del crédito sobre Producto Interno Bruto (PIB) alcanza un 4.7, el cual fue 0.4 que el año pasado.

Esto provoca que los empleados no cuenten con una forma de pagar servicios, deudas o algún entretenimiento hasta que llegue la quincena, gracias a la evolución tecnológica y a las plataformas móviles es posible el desarrollo en el sector financiero mexicano con el aterrizaje de un concepto relativamente nuevo en México: salario on demand.

Se trata de un producto financiero que permite a los empleados de una empresa acceder al salario que ganan antes del día del pago quincenal o mensual. El 80% de los empleados mexicanos se gastan su salario quincenal en los primeros siete días provocando una inestabilidad financiera para el empleado.

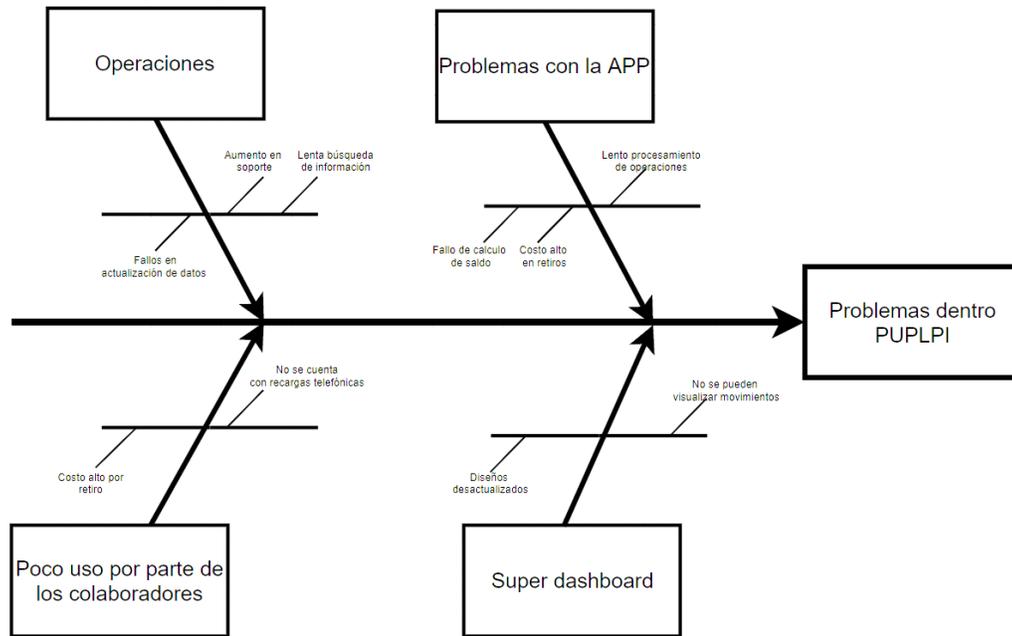
PULPI cuenta con varias características como lo son plataformas y aplicaciones, las cuales hoy en días son utilizados por cientos de usuarios, pero debido a su tráfico con diferentes tipos de usuarios, integraciones con múltiples empresas, integración con otros sistemas de nómina. Debido a esto la plataforma ha crecido demasiado alrededor de un 30% mensual.

Hoy en día el procesamiento de los datos es demasiado lento, además que algunos de los servicios que se ofrecen al usuario carecen de detalle, la plataforma no es tan detallada para el usuario provocando que se suela perder cuando trata de utilizar algún aplicativo.

Mientras que en operaciones el uso de la plataforma ha presentado algunos problemas en términos de actualización de datos y búsqueda de información.

Se desea mejorar los procesos de cálculo de saldo, darles un segundo uso a los adelantos del usuario a través de un pago de servicios, el poder reducir los costos por retiro que hoy día en promedio son de MXN\$ 36 a través de suscripciones y mejorar la validación de cuentas por parte de operaciones.

**Figura 1. Diagrama de Ishikawa**



*Fuente: Diseño Propio del Autor*

### 1.3 Justificación

PULPI cuenta con cientos de colaboradores y empresas que hacen uso hoy en día de sus instrumentos financieros ocupando un concepto conocido como Salary On Demand y, además brinda la opción a trabajadores de acceder a su salario ya trabajado, pero hoy en día existen unos problemas con el cálculo de saldo el cual ocasiona problema a colaboradores como a la empresa, ya que los colaboradores cuando quieren disponer de su salario cuentan con un saldo negativo provocándoles un estrés financiero.

Mientras que por parte de operación se incrementa el soporte puesto que una gran parte del soporte que se brinda es por el error de saldo negativo en las cuentas de los colaboradores. Además de implementar nuevos servicios como pago de servicios (telefonía móvil) y suscripciones.

Esta implementación ayudara a reducir el estrés financiero para el colaborador cuando quiera disponer de su periodo, mientras la implementación de nuevos servicios es un gran paso para la empresa dejando que el colaborador pueda realizar recargas telefónicas o comprar paquetes, esto complementa con la nueva implementación que permite comprar una suscripción mensual brindándole al usuario el veneficio de realizar operaciones con comisiones de MXN \$0.

## 1.4 Objetivos

### *Objetivo general*

Mejorar el cálculo de saldo por parte del colaborador para que este nunca cuente con saldo negativo y a que su vez estos puedan ser empleado para tanto adelantos como para pagar servicios. Incluyendo la implementación de suscripciones y pago de servicios.

### *Objetivos específicos*

- Suscripciones: Permitirá a los colaboradores realizar disposición de su sueldo sin tener que pagar comisión.
- Pago de servicios: Los colaboradores podrán realizar recargas telefónicas y compra de planes telefónicos.
- Cálculo de saldo: No se debe de contar con saldos negativos por parte del colaborador.
- Operaciones: Reducir el soporte.
- Validación de cuentas: automatizar la validación de cuentas.

## 1.5 Alcances y limitaciones

### Alcances:

- Aumento en uso de la aplicación por medio de pagos de servicios (recargas telefónicas).
- Brindar la posibilidad de ahorrar dinero a los colabores por medio de una suscripción.
- Reducción de soporte en estos se incluye envió de códigos de verificación, saldo negativo y cambio de datos personales.
- Al brindar mejoras al super dashboard permitirá que el equipo de operaciones pueda realizar sus tareas más rápidas.
- La búsqueda de información será más rápida y precisa.

#### Limitaciones:

- Con la arquitectura que hoy en día se tiene se reflejara limitado el cambio, debido a un microservicio que es la base principal de la aplicación y el cual su cambio tomara mas tiempo de lo esperado.
- Por políticas de empresa, algunas funciones de la aplicación como microservicios, códigos y cierta información que consideran importante de resguardar, no serán mencionadas en este proyecto.
- La información base, es la proporcionada por la empresa, a la fecha de inicio del proyecto, y sus proyecciones están en base a la misma.

## Capítulo 2 Marco teórico

### 2.1 Introducción

En este apartado se mencionan los conceptos que apoyan al desarrollo de la página web, así como mencionar aplicaciones ya existentes e instituciones que respaldan el proyecto. Además de contar con la definición de conceptos para el desarrollo de los nuevos aplicativos de la APP, haciendo uso de la metodología ágil de Scrum el cual es un marco de trabajo para desarrollo ágil de software.

### 2.2 Smartphone

Hoy en día la sociedad mexicana vive un fenómeno de exposición y uso creciente de la tecnología celular, en la mayoría de ámbitos. Esto se debe en gran medida a la disminución de costos en los equipos de telefonía celular y al incremento de sus capacidades tecnológicas.

El smartphone o también llamado “teléfono” inteligente, es un dispositivo que cabe en la palma de la mano y cuya aceptación se debe a su gran capacidad de comunicación, procesamiento y almacenamiento, al manejo de contenidos multimedia y a la facilidad de integrarse a las redes inalámbricas (Cruz & López Mortero, 2007).

En México el 57% de personas hace uso de un teléfono inteligente (INEGI, 2016), por lo que es fácil estar en contacto con el reconocimiento de voz, debido a que prácticamente la totalidad de los teléfonos inteligentes incorporan esta tecnología. Por una parte, la aceptación que han tenido los teléfonos inteligentes se debe a que estos cuentan con una pantalla táctil, lo cual facilita la navegación en internet y en general el uso de las diferentes aplicaciones disponibles. Pero una pantalla táctil no lo es todo, en ocasiones los usuarios tienen dificultades al escribir texto, puesto que la pantalla no es lo suficientemente grande como para que las teclas tengan un tamaño considerable. Es aquí donde el reconocimiento de voz surge como una alternativa para redactar cualquier texto de una manera más rápida y eficiente.

### 2.3 Sistema de información

Un sistema de información es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio. En un sentido amplio, un sistema de información no necesariamente incluye equipo electrónico (hardware). Sin embargo, en la práctica se utiliza como sinónimo de sistema de información computarizado (Kohen karen & Asín Lares, 2009).

Un sistema de información se compone de los siguientes elementos:

- El equipo computacional: Es el hardware necesario para que el sistema de información opere. Lo constituyen las computadoras y el equipo periférico.
- El recurso humano: Este interactúa con el sistema de información; son las personas que utilizan el sistema, lo alimentan con datos o utilizan los resultados que genera.
- Los datos o información fuente: Son todas las entradas que el sistema necesita para generar la información que se desea.
- El software: Son los programas que procesan los datos de entrada y generan los resultados que se esperan.
- Las telecomunicaciones: Básicamente el hardware y el software que transmiten en forma electrónica texto, datos, imágenes y voz.
- Procedimientos: Incluyen las políticas y reglas de operación, tanto en la parte funcional del proceso de negocio, como los mecanismos para hacer trabajar una aplicación en la computadora.

Cuando los elementos anteriores interactúan entre sí, son capaces de producir información de utilidad para una determinada organización. Para ello, se deben de cumplir cuatro actividades básicas:

1. Entrada de información: Es el ingreso de los datos en el sistema de información. Las entradas pueden ser manuales, las proporciona el usuario, o automáticas, datos o información que provienen o se toman de otros sistemas o módulos. Los típicos dispositivos de entrada de datos a los sistemas de información son las cintas magnéticas, las unidades de disquete, los lectores de código de barras, las plumas ópticas, el escáner, los monitores sensibles al tacto, los lectores de CD-ROM, los lectores de DVD, los USB, el mouse y el teclado.
2. Almacenamiento de información: Es una de las capacidades más importantes de una computadora, puesto que permite al sistema recordar la información guardada en la sesión anterior. Esta información se almacena en estructuras de información denominadas archivos, en su versión simple, y bases de datos, en su modalidad compleja. Los dispositivos de almacenamiento son los discos magnéticos o duros, los discos flexibles o disquetes, los discos compactos (CD-ROM), los discos de alta capacidad (zip), los discos de video (DVD) y los USB.
3. Procesamiento de información: Es la capacidad del sistema de información para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecida. Estos cálculos trabajan con datos de recién ingreso o con datos ya almacenados.

Esta característica de los sistemas permite la transformación de datos fuente en información útil.

4. Salida de información: Es la capacidad de un sistema de información para convertir la información procesada o los datos de entrada en información para el exterior. Las típicas unidades de salida son las impresoras, disquetes, cintas magnéticas, la voz, los plotters, entre otros. Es importante aclarar que la salida de un sistema de información puede constituir la entrada a otro sistema de información o módulo.

## **2.4 Sistemas transaccionales.**

Los sistemas transaccionales tienen como principales características:

- Permitir ahorros significativos de mano de obra, debido a que automatizan tareas operativas de la organización.
- Con frecuencia son el primer tipo de sistemas de información que se implanta en las organizaciones. Primero sirven para apoyar las tareas a nivel operativo de la organización y, a medida que evolucionan, continúan con los mandos intermedios para posteriormente apoyar a la alta administración.
- Tienen una intensa entrada y salida de información, aunque sus cálculos y procesos suelen ser poco complejos. Estos sistemas requieren de un arduo manejo de datos para poder realizar sus operaciones y, como resultado, generan grandes volúmenes de información.
- Tienen la propiedad de ser recolectores de información, es decir, a través de ellos se cargan las grandes bases de datos para su posterior utilización. Estos sistemas integran cantidades grandes de la información que se maneja en la organización, la cual será empleada después para apoyar a los mandos intermedios y altos.
- Son fáciles de justificar ante la dirección general, dado que sus beneficios son visibles y cuantificables en el corto plazo. El proceso de justificación se realiza con el cotejo de ingresos y costos.
- Son fácilmente adaptables a paquetes de aplicación que se encuentran en el mercado, porque automatizan los procesos básicos que, por lo general, son similares o iguales en otras organizaciones.
- Ejemplos de este tipo de sistemas son facturación, nóminas, cuentas por cobrar, cuentas por pagar, contabilidad general, conciliaciones bancarias, inventarios, abastecimientos, control de distribución, producción, etcétera (Kohen karen & Asín Lares, 2009).

## 2.5 Sistemas de apoyo la toma de decisiones.

El éxito de cada organización depende en gran medida de la calidad de las decisiones que toman sus empleados. Cuando la toma de decisiones implica grandes cantidades de información y mucho procesamiento, las tecnologías de la información hacen más eficiente y eficaz este proceso. Un sistema de soporte de decisiones (DSS), es un sistema de información basado en una computadora diseñado para ayudar a los administradores a elegir una de muchas soluciones alternas a un problema (Oz, 2008). Este tipo de sistemas ofrecen análisis sofisticados y rápidos de inmensas cantidades de datos e información.

Los DSS se deben implementar después de los sistemas transaccionales puesto que estos últimos constituyen su plataforma de información. Por otro lado, la información que generan los DSS suelen ser intensivos en cálculos y escasos en entradas y salidas de información de información, la cual toman de los sistemas transaccionales. Por ejemplo, un modelo de planeación financiera requiere poca información de entrada, pero realiza muchos cálculos durante su proceso.

Los sistemas de apoyo a las decisiones se clasifican en:

- DSS (decision support systems), sistemas de apoyo a la toma de decisiones.
- GDSS (group decision support systems), sistemas para la toma de decisiones en grupo.
- EIS (executive information systems), sistemas de información para ejecutivos.
- EDSS (expert decision support systems), sistemas expertos de apoyo a la toma de decisiones (Kohen karen & Asín Lares, 2009).

Cualquiera de la clasificación de DSS tienen la finalidad de brindar a los individuos encargados de la toma de decisiones es una organización de brindar las bases necesarias con información oportuna y verídica para el logro de sus objetivos planteados.

## 2.6 Sistemas de información estratégicos.

Su función es lograr ventajas que los competidores no poseen, tales como ventajas en costos y servicios diferenciados con clientes y proveedores. En este sentido, los sistemas estratégicos son creadores de barreras de entrada al negocio. Por ejemplo, el comercio electrónico es un sistema estratégico, debido a que da a las empresas elementos para diferenciar sus productos/servicios y así tener ventajas sobre otras compañías que ofrecen productos similares y no cuentan con este servicio. Si una empresa nueva decide entrar al mercado, tendrá que ofrecer este servicio para poder competir.

Los sistemas de información estratégicos tienen las siguientes características:

- Su función no es automatizar procesos operativos ni proporcionar información para apoyar la toma de decisiones. Sin embargo, este tipo de sistemas puede llevar a cabo esas funciones.
- Su desarrollo es dentro de la organización, por tanto, no se adaptan fácilmente a paquetes disponibles en el mercado.
- Su desarrollo típico es en incrementos. Su evolución dentro de la organización inicia con un proceso o función particular y a partir de ahí se agregan nuevas funciones o procesos.
- Apoyan el proceso de innovación de productos y procesos dentro de la empresa, debido a que buscan ventajas respecto de los competidores (Kohen karen & Asín Lares, 2009).

Es importante que se tome en cuenta que las ventajas que se obtienen con estos sistemas no son eternas, es decir, existe un periodo de vigencia que equivale al tiempo que tardan los competidores en alcanzar las diferencias o ventajas obtenidas por el sistema de información estratégico. Cuando esto sucede, los beneficios generados por el sistema de información estratégico se convierten en estándares de la industria, como en el caso del comercio electrónico, que serán requisitos de infraestructura tecnológica que una compañía debe poseer.

Los negocios, las personas y las organizaciones buscan utilizar la información de manera específica para tomar decisiones sólidas y para resolver problemas. Estas dos últimas prácticas muy relacionadas, son la base de toda empresa exitosa.

Un sistema de información se puede entender como un conjunto de elementos que colaboran entre sí para procesar datos y producir información. En las empresas la mayoría de los sistemas de información con los que cuentan, están integrados por muchos subsistemas con metas secundarias, todas las cuales contribuyen a la meta principal de la organización. Es necesario tomar en cuenta que los elementos por los cuales están formados los sistemas de información pueden ser de diferentes tipos, tales como:

- Datos.
- Hardware.
- Software.
- Telecomunicaciones.
- Personas.
- Procedimientos.

Existen diferentes tipos de sistemas de información, de los cuales una empresa puede hacer uso de acuerdo a sus diferentes jerarquías.

- Sistemas transaccionales.
- Sistemas de apoyo a la toma de decisiones.
- Sistemas estratégicos.

Los sistemas transaccionales, tienen el propósito de realizar procesos que se llevan a cabo en la parte operativa de una empresa. Estos procesos tienen la característica de realizarse de manera rutinaria, además de que no son complejos.

Los sistemas de apoyo a la toma de decisiones se satisfacen de la información resumida que producen los sistemas transaccionales. Con la finalidad de poder brindar un panorama general a los directivos de la empresa, para que de esta forma se puedan tomar las decisiones más acertadas con base en hechos firmes y no en suposiciones o empíricamente.

Los sistemas estratégicos tienen la misión de dotar de alguna ventaja competitiva a una empresa, que sus competidores en el mercado no posean. Las diferentes ventajas pueden ser en costos y servicios diferenciados con clientes y proveedores para incrementar el volumen de ventas.

El uso de los sistemas de información por parte de las empresas dedicadas a la venta de alimentos tales como restaurantes y cadenas de comida rápida; han brindado una ventaja competitiva al ser utilizados en los procesos transaccionales. Cuando se habla de un sistema información basado en aplicaciones móviles que automatice los procesos de toma y cobro de pedidos en establecimientos dedicados a la venta de comida rápida, es imprescindible tener en cuenta los elementos que posee dicho sistema. De acuerdo al problema planteado figuran los siguientes elementos:

- Dispositivo móvil con sistema operativo Android
- Aplicación móvil
- Datos de entrada

## **2.7 Tecnologías de la información.**

Las tecnologías de la información se pueden ver como la base que soporta a los sistemas de formación, debido a que en ellas recaen sus elementos.

Las tecnologías de la información son todas aquellas tecnologías que permiten y dan soporte a la construcción y operación de los sistemas de información, y son tecnologías de hardware, software, de almacenamiento y de comunicaciones. Estas tecnologías forman la infraestructura tecnológica de la empresa, que provee una plataforma en la

cual la compañía construye y opera los sistemas de información (Kohen karen & Asín Lares, 2009).

En la actualidad los sistemas de información deben de cumplir con tres objetivos primordiales para las organizaciones:

- Automatizar los procesos operativos.
- Proporcionar información que sirve de apoyo en el proceso de toma de decisiones.
- Logran ventajas competitivas a través de su implantación y uso

Cuando los sistemas de información automatizan los procesos operativos dentro de una organización se les llama sistemas transaccionales, porque su función primordial consiste en procesar transacciones tales como: pagos, cobros, pólizas, entradas, salidas, etc. El complemento de los sistemas transaccionales son los sistemas enfocados a mejora de la toma de decisiones, cuya función es la manipulación de la información con el fin de apoyar y fundamentar la toma de decisiones. También existen los sistemas de automatización de oficinas, útiles en la mayoría de las empresas para trabajar con documentos. Algunos programas considerados en este tipo de sistemas son: hojas de cálculo, procesadores de texto, herramientas para hacer presentaciones, bases de datos y herramientas para el manejo de imágenes. Otro tipo de sistemas de acuerdo con los objetivos que cumple son los sistemas estratégicos; estos buscan desarrollar en las organizaciones ventajas competitivas a través del uso de las tecnologías de la información.

## **2.8 Sistema Operativo.**

Un sistema operativo es simplemente el intermediario entre la parte física de una computadora (hardware) y el usuario (Tanenbaum, 2009). Por otra parte, un sistema operativo se puede definir como la plataforma de software de un dispositivo, el cual soportará y administrará las aplicaciones para el usuario final.

Las computadoras modernas constan de procesadores, memorias, temporizadores, discos, ratones, interfaces de red, impresoras y una amplia variedad de otros dispositivos. En la perspectiva alterna, el trabajo del sistema operativo es proporcionar una asignación ordenada y controlada de los procesadores, memorias y dispositivos de E/S, entre los diversos programas que compiten por estos recursos (Tanenbaum, 2009).

En este sentido, es necesario mencionar que los dispositivos móviles actuales, tales como tabletas y smartphones pueden ser vistos como computadoras, por lo que estos dispositivos inteligentes cuentan con un sistema operativo que administra sus recursos. A inicios de la segunda década del siglo XXI fue cuando se empezaron a popularizar los smartphones y tabletas. En un inicio empezaron a competir diferentes sistemas operativos en el mercado como: Symbian, Blackberry OS, Windows Phone, iOS y

Android. Sin embargo, actualmente sobreviven estos dos últimos, debido a la preferencia de los usuarios a nivel mundial. De acuerdo a la empresa Gartner en el año 2016, el 80.7% de los smartphones distribuidos utilizaban el sistema operativo android, mientras que tan solo el 17.7% poseía el sistema operativo iOS. Las estadísticas son claras, y reflejan que el sistema operativo dominante actualmente es Android.

## **2.9 Sistema Operativo “Android”**

Android es un sistema operativo Open Source pensado para teléfonos móviles y desarrollado por Open Handset Alliance (OHA) bajo autorización de Google (Nazim, 2013). Android se basa en un kernel Linux y se distribuye bajo la licencia Apache License 2.0. Este sistema operativo separa la capa de hardware de la capa lógica; esto permite que cualquier teléfono Android pueda ejecutar la misma aplicación, lo cual hace posible que se genere un gran número de posibilidades para los fabricantes, desarrolladores y usuarios.

A pesar de su poco tiempo de vida, éste ha sido uno de los crecimientos más rápidos en lo que ha sistemas operativos se refiere, puesto que a principios de esta década sólo estaba presente en Smartphones, posteriormente también era posible encontrarlo en tabletas y televisores inteligentes. Hoy en día Android está presente en relojes y autos.

Existen diferentes motivos a los que se le deben su éxito. Una de las características más importantes de este sistema operativo reside en que es completamente libre. Es decir, ni para programar en este sistema ni para incluirlo en un teléfono hay que pagar nada (David & Clodoaldo, 2012). Por esta razón es muy popular entre los fabricantes de teléfonos y desarrolladores, visto que los costes para lanzar un teléfono o una aplicación son muy bajos.

Por otra parte, la amplia gama de aplicaciones que están disponibles en su tienda oficial llamada Google Play en donde es posible encontrar un gran número de aplicaciones para satisfacer necesidades específicas del usuario que van desde alarmas y relojes personalizados hasta cursos para aprender un nuevo idioma; además de que Android es muy flexible en cuanto a la personalización, debido a que el usuario tiene la capacidad de modificar e interactuar con el dispositivo Android para adaptarse a los gustos y preferencias del usuario.

## **2.10 Lenguaje de programación**

Para que las computadoras y demás dispositivos inteligentes puedan ser útiles para los seres humanos mediante aplicaciones, es necesario que éstas sean programadas mediante un lenguaje de programación. Un lenguaje de programación es un conjunto de reglas y estándares que es utilizado para escribir programas de computador (software), que puedan ser entendidos por él (Joyanes, Rodriguez, & Fernández, Fundamentos de Programación, Libro de problemas., 1996). Es importante considerar

que la programación se puede definir como el proceso de creación de un programa de computadora, mediante la aplicación de procedimientos lógicos (Wilson, 1993).

Hoy en día existen diferentes lenguajes de programación, los cuales son utilizados para programar el software que permite a las personas hacer uso de cualquier computadora. Estos lenguajes han sido creados para satisfacer alguna necesidad informática en específico, por ejemplo, para programar en un entorno web, para desarrollar software de computadoras de escritorio e incluso para desarrollar las aplicaciones que actualmente utilizan los dispositivos móviles. Para desarrollar software dedicado a dispositivos móviles, es importante considerar que prácticamente dos diferentes lenguajes de programación son utilizados. Por el lado de los dispositivos móviles con sistema operativo android el lenguaje java es utilizado para desarrollar sus aplicaciones móviles. Mientras que, por el lado de iOS, Objective-C es el lenguaje más popular en el que se desarrollan las aplicaciones que se encuentran en App Store.

### **2.11 Java**

Java es un lenguaje de programación creado para desarrollar programas que ejecuten tareas de diversos tipos, y se diseñó de manera concreta para realizar una amplia gama de actividades en internet (Joyanes & Zahonero, Programación en Java, 2011). Sin embargo, hoy en día el lenguaje java es utilizado para desarrollar las aplicaciones que se utilizan en los dispositivos móviles con sistema operativo android. Java fue elegido como lenguaje de programación para el entorno de desarrollo en android.

Haciendo uso del IDE oficial de desarrollo “android studio”. Es importante mencionar que este último, es un potente editor de códigos y ofrece diferentes herramientas para desarrolladores que aumentan la productividad de compilación de apps. A continuación, se describen las características más importantes:

- Un emulador rápido con varias funciones.
- Un entorno unificado en el que puedes realizar desarrollos para todos los dispositivos Android.
- “Instant Run” para aplicar cambios mientras tu app se ejecuta sin la necesidad de compilar un nuevo APK.
- Integración de plantillas de código y GitHub para ayudarte a compilar funciones comunes de las apps e importar ejemplos de código.
- Herramientas “Lint” para detectar problemas de rendimiento, usabilidad, compatibilidad de versión, etc.
- Compatibilidad con C++ y NDK
- Soporte incorporado para Google Cloud Platform, lo que facilita la integración de Google Cloud Messaging y App Engine.

## 2.12 Bases de datos.

Las bases de datos son un componente esencial de la vida cotidiana en la sociedad moderna. Actualmente la mayoría de las personas se enfrentan a diversas actividades que implican cierta interacción con una base de datos (Elmasri & Navathe, 2007). Una base de datos es una colección de datos lógicamente coherente con algún tipo de significado inherente, por lo que no es correcto llamar base de datos a un surtido aleatorio de datos (Elmasri & Navathe, 2007).

Entre los ejemplos más comunes del uso de una base de datos se encuentran: ir al banco a depositar o retirar dinero, realizar una reservación en un hotel de manera online, comprar algún artículo en una tienda de autoservicio, acceder al sistema de una biblioteca para encontrar un libro, o simplemente al interactuar con cualquier tipo de redes sociales. Todas estas actividades implican que la computadora acceda a una base de datos.

## 2.13 Sistemas gestores de bases de datos.

Un Sistema Gestor de Bases de Datos o SGBD se puede definir como una colección de datos relacionada entre sí, estructurados y organizados, y un conjunto de programas que acceden y gestionan esos datos. La colección de esos datos se denomina Base de Datos o BD (Ramos, Ramos, & Montero, 2006).

Los SGBD son paquetes de software muy complejos que deben proporcionar una serie de servicios que van a permitir almacenar y explotar los datos de forma eficiente. Como todo sistema, los SGBD se componen diferentes elementos; en este caso, son cuatro:

- a) **Lenguajes de los SGBD:** Todos los SGBD ofrecen lenguajes e interfaces apropiadas para cada tipo de usuario: administradores, diseñadores, programadores de aplicaciones y usuarios finales. Los lenguajes van a permitir al administrador de la BD especificar los datos que componen la BD, su estructura, las relaciones que existen entre ellos, las reglas de integridad, los controles de acceso, las características de tipo físico y las vistas externas de los usuarios.
- b) **El diccionario de datos:** Es el lugar donde se deposita información acerca de todos los datos que forman la BD. Es una guía en la que se describe la BD y los objetos que la forman. Ésta contiene las características lógicas de los sitios donde se almacenan los datos del sistema, incluyendo nombre, descripción, alias, contenido y organización. Identifica los procesos donde se emplean los datos y los sitios donde se necesita el acceso inmediato a la información.
- c) **Seguridad e integridad de datos:** Un SGBD proporciona los mecanismos para garantizar la seguridad e integridad de los datos.

- d) El administrador de la base de datos: En los sistemas de gestión de bases de datos actuales existen diferentes categorías de usuarios. Estas categorías se caracterizan por el nivel de privilegios que tiene cada usuario.

Actualmente existen diferentes sistemas gestores de bases de datos en el mercado, entre los más comunes se encuentran: MySQL, dBase, Informix, DB2, PostgreSQL, SQLite, entre otras. Actualmente SQLite es en SGBD más utilizado para almacenar datos en los dispositivos móviles actuales.

### **2.14 Modelo relacional.**

Para poder hacer uso de la estructura lógica de una base de datos es necesario representarla mediante un modelo. El modelo más utilizado para hacer este tipo de representaciones es el modelo relacional. En otras palabras, un modelo relacional permite representar la forma en que una base de datos está construida, además de cómo sus datos interactúan entre sí. El modelo relacional representa la base de datos como una colección de relaciones.

Informalmente, cada una de estas relaciones se parece a una tabla de valores o, de forma algo más extensa, a un fichero plano de registros (Ramez & Shamkant, 2007).

### **2.15 Modelo entidad relación.**

Para modelar una base de datos, es ideal contar con un diagrama que represente las entidades de una base de datos, así como sus interrelaciones de esta. Este modelo es llamado de “entidad-relación”. Dicho modelo intenta describir “el mundo real” como un conjunto de entidades y relaciones entre ellas. Está basado en una percepción de un mundo real que consiste en una colección de objetos básicos, denominados entidades, y de relaciones entre estos objetos (Ramez & Shamkant, 2007).

### **2.16 Modelo entidad relación extendida.**

Como se ha descrito anteriormente, un modelo entidad relación, intenta describir el mundo real mediante entidades y las relaciones que existen entre ellas, sin embargo, para efectos de programación es importante adaptar este modelo a uno que se parezca más a la programación orientada a objetos, Este modelo representa de forma más precisa los requerimientos de una base de datos (Mannino, 2007).

### **2.17 Normalización de bases de datos.**

La normalización es un proceso de eliminación de redundancias en una tabla para que sea más fácil de modificar (Mannino, 2007). A continuación, se muestran las tres formas normales para hacer que una base de datos funcione óptimamente.

- Primera forma normal. La primera forma normal ayuda a no repetir datos en las tablas que componen la base de datos.
- Segunda forma normal. La segunda forma normal intenta que cada columna de la tabla debe depender de la clave, esto significa que cada registro debe depender únicamente de la clave principal.
- Tercera forma normal. En cuanto a la tercera forma normal es necesario asegurarse de que ninguna columna dependa de una columna que no tenga una clave.

## **2.18 Dispositivos móviles.**

Un dispositivo móvil es un aparato electrónico que permite el procesamiento de datos mediante las aplicaciones que pueden ser instaladas en él, con el fin de cubrir alguna necesidad del usuario. Gracias a su pequeño tamaño y escaso peso son fáciles de transportar incluso en un bolsillo.

Debido a la fuerte competencia de este sector los cambios en estos dispositivos, su variedad y características evolucionan rápidamente. Sin embargo, se mantienen una serie de características comunes. La mayoría cuentan con una pantalla generalmente táctil y un teclado que hacen posible la entrada y salida de datos.

## **2.19 Aplicaciones Móviles.**

Las aplicaciones móviles se pueden concebir como el software o programas que poseen los dispositivos móviles con la finalidad de satisfacer una o varias necesidades de un usuario.

Las funciones de cada aplicación disponible van en función de las características de los dispositivos y sistema operativo de los dispositivos que se pretenda instalar. Estas características van desde el procesador, memoria RAM e incluso capacidad de almacenamiento.

## **2.20 Metodologías de desarrollo ágil.**

Durante el año 2001 nació el término “ágil” aplicado al área de desarrollo de software, la esencia de las metodologías ágiles se basa en valores y principios que deberían permitir a los equipos desarrollar software rápidamente y responder a los cambios que pueden surgir a lo largo del proyecto. Este tipo de metodologías se pueden concebir como una alternativa a los procesos de desarrollo tradicionales caracterizados por su total rigidez y muy dirigidos a la documentación que se genera tras cada una de las actividades desarrolladas.

Las metodologías de desarrollo ágiles se basan en los siguientes principios (Letelier, 2006):

- Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas. La gente es el principal factor de éxito de un proyecto software. Si se sigue un buen proceso de desarrollo, pero el equipo falla, el éxito no está asegurado; sin embargo, si el equipo funciona, es más fácil conseguir el objetivo final, aunque no se tenga un proceso bien definido. No se necesitan desarrolladores brillantes, sino desarrolladores que se adapten bien al trabajo en equipo. Así mismo, las herramientas (compiladores, depuradores, control de versiones, etc.) son importantes para mejorar el rendimiento del equipo, pero el disponer más recursos que los estrictamente necesarios también puede afectar negativamente. En resumen, es más importante construir un buen equipo que construir el entorno. Muchas veces se comete el error de construir primero el entorno y esperar que el equipo se adapte automáticamente. Es mejor crear el equipo y que éste configure su propio entorno de desarrollo en base a sus necesidades.
- Desarrollar software que funciona más que conseguir una buena documentación. Aunque se parte de la base de que el software sin documentación es un desastre, la regla a seguir es “no producir documentos a menos que sean necesarios de forma inmediata para tomar una decisión importante”. Estos documentos deben ser cortos y centrarse en lo fundamental. Si una vez iniciado el proyecto, un nuevo miembro se incorpora al equipo de desarrollo, se considera que los dos elementos que más le van a servir para ponerse al día son: el propio código y la interacción con el equipo.
- La colaboración con el cliente más que la negociación de un contrato. Las características particulares del desarrollo de software hacen que muchos proyectos hayan fracasado por intentar cumplir unos plazos y unos costes preestablecidos al inicio del mismo, según los requisitos que el cliente manifestaba en ese momento. Por ello, se propone que exista una interacción constante entre el cliente y el equipo de desarrollo. Esta colaboración entre ambos será la que marque la marcha del proyecto y asegure su éxito.
- Responder a los cambios más que seguir estrictamente un plan. La habilidad de responder a los cambios que puedan surgir a lo largo del proyecto (cambios en los requisitos, en la tecnología, en el equipo, etc.) determina también el éxito o fracaso del mismo. Por lo tanto, la planificación no debe ser estricta puesto que hay muchas variables en juego, debe ser flexible para poder adaptarse a los cambios que puedan surgir. Una buena estrategia es hacer planificaciones detalladas para unas pocas semanas y planificaciones mucho más abiertas para unos pocos meses.

## 2.21 Administración de Proyectos

La Administración del Portafolio de Proyectos (PPM, por sus siglas en inglés) es un término que se refiere a varios enfoques para poder lidiar con proyectos que son parte de un portafolio general en una organización o equipo de desarrollo. ("Conexionesan", 2016)

## 2.22 SCRUM

SCRUM es una metodología ágil para el desarrollo de proyectos. Es un proceso de gestión que reduce la complejidad en el desarrollo de proyectos que ayuda a satisfacer las necesidades de los clientes o interesados. La gerencia y los equipos de Scrum trabajan juntos alrededor de requisitos y tecnologías para entregar productos funcionando de manera incremental usando el empirismo y la motivación en el equipo de desarrollo. (Francia, 2017)

## 2.23 Sprint

Es un intervalo prefijado durante el cual se crea un incremento de producto "Hecho o Terminado" utilizable, potencialmente entregable. A lo largo del desarrollo hay Sprints consecutivos de duración constante. (Bara, s.f.)

## 2.24 Aplicación Móvil

Una App es una aplicación de software que se instala en dispositivos móviles o tablets para ayudar al usuario en una labor concreta, ya sea de carácter profesional o de ocio y entretenimiento, El objetivo de una app es facilitarnos la consecución de una tarea determinada o asistirnos en operaciones y gestiones del día a día. (QODE, 2012)

## 2.25 Cronograma de Actividades

Es la transcripción a tiempos de los procesos y acciones para llevar a cabo un proyecto. En él se establece cuánto tiempo va a costar a la organización que sus recursos lleven a cabo cada proceso. (vilá, 2014)

## 2.26 Retrospectiva

Nombre de la reunión en la que el equipo analiza la forma de trabajo para su mejora continua. Las reuniones retrospectivas son por tanto un una "meta-práctica" ágil. (Manager, 2017)

## 2.27 Estatus

El estatus semanal y reporte mensual es parte del control del proyecto e implica comparar la ejecución con la planeación. (Sanchez, 2015)

## 2.28 El panel de Tareas

Este panel muestra las tareas que tienen asignadas los miembros del equipo. Esta tabla se divide en tres columnas que representan el estado de la actividad: (Walter.lara.37, 2016)

- Por hacer.
- Haciendo.
- Terminado.

## Capítulo 3

### Marco Metodológico

#### 3.1 Introducción

Durante esta parte se explica la metodología ágil de Scrum para el desarrollo de proyectos que requieren mayor rapidez y adaptabilidad en sus resultados. Dando un breve resumen de la historia de Scrum, roles, palabras claves y fases (planificación del sprint, etapa de desarrollo, revisión del sprint y retroalimentación). Además de incluir el software utilizado durante el desarrollo de este proyecto.

#### 3.2 Scrum: una metodología ágil

La palabra Scrum proviene del nombre de una jugada que ocurre en los partidos de rugby (Pressman, 2010). Scrum (Sims & Johnson, 2011) es una metodología ágil en la cual se llevan a cabo una serie de prácticas iterativas cuyo objetivo es que el grupo de desarrolladores trabaje unido, contribuyendo con sus habilidades individuales para la obtención de un software de buena calidad. Una de las características de Scrum es la entrega de porciones incrementales del producto final al término de cada iteración; de esta manera el cliente puede ir haciendo modificaciones o continuar con el desarrollo del software tal como se tenía previsto originalmente. Scrum es una metodología diseñada para el desarrollo de productos en ambientes complejos en donde se requiere un producto funcional rápidamente, con cambios constantes o con especificaciones ambiguas.

En un equipo de Scrum existen tres diferentes roles: el propietario del producto, el Scrum master y los miembros del equipo (Ver Tabla I). Dentro del marco general de Scrum existen reuniones que ayudan a la planeación del desarrollo en donde las experiencias adquiridas por los desarrolladores son muy relevantes, por lo que la retroalimentación y la recopilación de historias son fundamentales (Ver Tabla II).

El Sprint es el ciclo fundamental (iteración) del proceso de Scrum (Ver Figura 1). Scrum no especifica la longitud del sprint de su equipo, pero se considera que cuatro semanas es el tiempo máximo. Dos semanas es la longitud más frecuente, aunque también es común encontrar sprints con longitud de entre una y tres semanas.

**Tabla 1. Roles principales en Scrum**

<b>Rol</b>	<b>Descripción</b>
Propietario del producto (Product Owner)	Se asegura de que las necesidades de los clientes y usuarios finales son comprendidas por el equipo. Hace esto directamente creando, perfeccionando y comunicando los requisitos.
Maestro Scrum (Scrum Master)	Una de sus principales funciones es la implementación de cada una de las características de Scrum, también es un facilitador el cual se asegura de eliminar los problemas que pueda llegar a tener el equipo
Equipo de desarrollo	Los equipos son altamente colaborativos, también se auto-organizan. Los miembros del equipo tienen autoridad total sobre cómo se hace el trabajo. Las personas que hacen el trabajo son las más autorizadas para decidir la mejor forma de hacerlo

Fuente: Diseño Propio del Autor

**Tabla 2. Reuniones en Scrum**

<b>Reunión</b>	<b>Descripción</b>
Scrum diario (Daily Scrum)	Es una reunión diaria generalmente al inicio del periodo de trabajo, que se puede adaptar a las necesidades del equipo. De manera breve cada participante comparte: <i>Lo que ha completado desde el último Daily Scrum.</i> <i>Lo que espera completar para el siguiente Daily Scrum.</i> <i>Los obstáculos que ha tenido durante el desarrollo</i>
Scrum de Scrum	Esta reunión ocurre al finalizar el Daily Scrum; acuden los líderes de grupos especializados en un área de desarrollo. Cada participante comparte: <i>Lo que ha completado su equipo desde la última reunión.</i> <i>Lo que espera que complete su equipo para la siguiente reunión.</i> <i>Los obstáculos que ha tenido su equipo.</i>
Reunión de planeación del Sprint (Sprint Planning Meeting)	El Sprint Planning Meeting marca el principio del sprint. Comúnmente tiene dos partes. El objetivo de la primera parte es que el equipo se comprometa a un conjunto de metas para el sprint. Durante la segunda parte de la reunión, el equipo identifica las tareas que deben realizarse en el orden de las historias de usuario acordadas.
Reunión de revisión del Sprint (Sprint Review Meeting)	Al final del sprint, el equipo tiene la oportunidad de mostrar su trabajo en el Sprint Review. El equipo muestra las historias que fueron completadas y las que faltaron por completar. En esta etapa es cuando el propietario del producto toma nota del trabajo realizado, revisando las características solicitadas por el cliente y evaluando si es necesario realizar cambios o agregar nuevas características.
Retrospectiva (Retrospective)	La retrospectiva se lleva a cabo al final de cada sprint, consiste en dedicar tiempo para que el equipo se enfoque en lo que se ha aprendido durante el sprint; esta retroalimentación puede ser útil para hacer algunas mejoras.

Fuente: Diseño Propio del Autor

**Figura 2. Visión general de Scrum**

*Fuente: Diseño Propio del Autor*

### 3.3 Scrum: Jira Software

Para llevar a cabo la metodología se hizo uso de la herramienta de Jira la cual es una herramienta en línea para la administración de nuevas características que agreguen valor al sistema, la cual permite dar seguimiento de errores e incidencias, priorizar características y tener un enfoque más detallado del producto.

Si Scrum es el marco que guía a los equipos, sus roles, eventos, artefactos y reglas, entonces el tablero de Jira Scrum es una visualización de su progreso a lo largo del ciclo de desarrollo. El Jira Scrum Board es más que un tablero de tareas, ya que permite hacer lo siguiente:

- Mejorar la comunicación y la transparencia: El Jira Scrum Board es la única fuente de información para todo el trabajo que el equipo necesita hacer. Dado que es accesible para todos los miembros del equipo en cualquier momento, saben exactamente qué tareas deben realizar y pueden detectar rápidamente cualquier obstáculo.
- Liderar la planificación rápida y el desarrollo iterativo: En el corazón del marco de scrum se encuentra el sprint, que es una cantidad de tiempo establecida (generalmente dos semanas) para que los equipos creen un producto potencialmente relevante adicional. El Jira Scrum Board está diseñado para que los equipos puedan organizar su trabajo en los plazos de carrera.
- Mejorar el enfoque y la organización del equipo: Los equipos pierden los plazos de los proyectos cuando están sobrecargados o pierden la noción de los eventos clave. Los ensamblajes de Jira Scrum brindan transparencia al trabajo en equipo al dividir el trabajo en fases y usar informes de velocidad y trabajo pendiente.

## Capítulo 4 Desarrollo Metodológico

### 4.1 Introducción

En este capitulado se lleva a cabo las fases de Scrum para el desarrollo del proyecto donde incluye los apartados de procedimiento, diseño, pruebas unitarias de backend y frontend, pruebas de QA y sanidad.

### 4.2 Desarrollo Metodológico

La metodología ágil que se empleó para el desarrollo del proyecto fue SCRUM. Porque permite el trabajo colaborativo, en equipo y permite obtener el mejor resultado posible de un proyecto.

Durante el desarrollo del proyecto se realizaron entregas parciales y regulares del producto final, priorizando algunas tareas más que otras. Para llevar a cabo la metodología se hizo uso de la herramienta de Jira la cual es una herramienta en línea para la administración de nuevas características que agreguen valor al sistema, la cual permite dar seguimiento de errores e incidencias, priorizar características y tener un enfoque más detallado del producto.

### 4.3 Procedimiento

Para el desarrollo del proyecto se realizó una investigación del tipo cualitativa, ya que durante meses se ha dado a la situación donde el colaborador cuenta con un saldo negativo, quedándose sin el pago del periodo. Esto ocasiona dos grandes problemas, donde de un lado tenemos al colaborador disgustado por la pérdida de su periodo y del otro por parte de la empresa que por medio de un análisis se descubrió que un 70% del soporte brindado a usuarios era relacionado al saldo negativo.

#### 4.4 Diseño

##### 4.4.1 Cálculo de saldo

El nuevo cálculo de saldo tiene como propósito de reducir el soporte donde los colaboradores contaban con saldo negativo o pendiente, buscando el problema se logró identificar que los días donde el usuario necesitaba dinero, se disponía de poco sueldo.

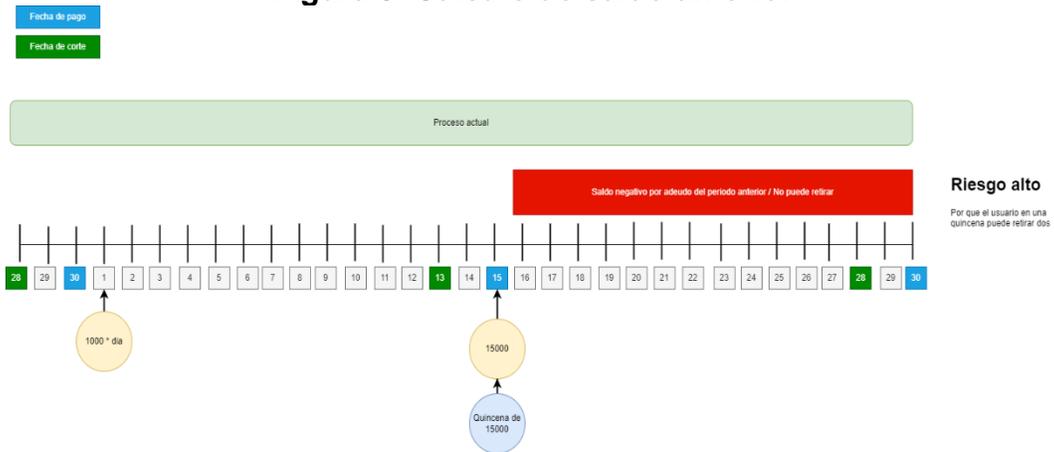
Además de que al usuario se le podía adelantar un periodo entero y se le descontaría hasta el próximo pago debido a que su retiro fue ejecutado días después de su fecha de corte.

Es decir que si el periodo empieza el día 1 contaba con su sueldo completo y la empresa realiza la fecha de corte el día 13, suponiendo que los días anteriores no se realizaba un retiro, para el día de pago la empresa debe de pagar completo su salario,

pero de lado de la plataforma aún puede retirar el adelanta a su salario si esto lo realizaba los días después de la fecha de corte con su empresa.

Provocando que el colaborador pudiera cobrar dos veces su salario, pero al llegar el siguiente periodo contaría con un saldo negativo.

**Figura 3. Cálculo de saldo anterior**

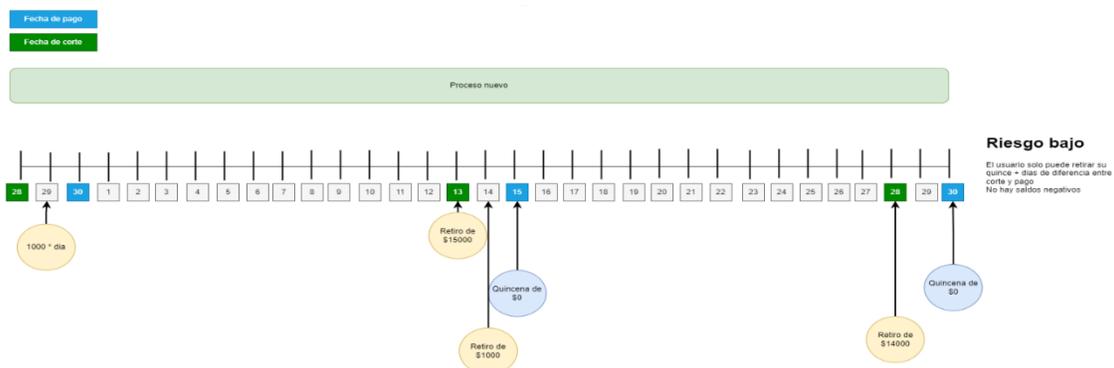


Fuente: Diseño Propio del Autor

La nueva implementación de cálculo de saldo es que el día 1 después de la fecha de corte por parte de la empresa del colaborador. Eso quiere decir que puede retirar hasta un 70% de su sueldo entre el día 1 y la fecha de corte dada por su empresa (ejemplo el día 13), dejando un 30% restante de su periodo suponiendo que este lo puede retirar el día 15 o dependiendo su tipo de pago (semanal, quincenal o mensual).

De esta forma nunca se presentará un saldo negativo y así PULPI solo adelante dos días al colaborador y no lo de todo un periodo.

**Figura 4. Nuevo cálculo de saldo**



Fuente: Diseño Propio del Autor

Ahora el cálculo de saldo se realiza de fecha de corte a fecha de corte como se ve en el siguiente ejemplo de nómina quincenal.

En el caso de nómina quincenal se debe calcular el sueldo solo dividiendo el sueldo neto mensual entre dos, ya que las compañías proporcionan un sueldo neto mensual, adicional a esto el sueldo diario se calcula según los días entre fecha de corte previo más un día y la fecha de corte actual.

**Tabla 3. Nuevo cálculo de saldo**

Descripción	Cantidad	Total
Sueldo mensual	\$18,000.00	
Restricción de empresa tomando como máximo el 100%	100% - 70%	30%
Restricción por tipo de colaborador	15%	15%
Restricción total	30% + 15%	45%
% de saldo disponible	100% - 45%	55%
Calculo mensual neto después de restricción	\$18,000.00 * 55%	\$9,900.00
Calculo por tipo de nómina quincenal	\$9,900.00 / 2	\$4,950.00
Sueldo disponible máximo por quincena en PULPI		\$4,950.00

Fuente: Diseño Propio del Autor

Este quedaría de la siguiente forma pagando en dos quincenas, simulando el resultado del nuevo cálculo de saldo.

Ejemplo 29 Noviembre - 13 Diciembre

**Tabla 4. Ejemplo nuevo cálculo de saldo primera quincenal**

Descripción	Cantidad	Total
Calculo mensual neto después de restricciones	\$18,000.00 * 75%	\$9,900.00
Calculo por tipo de nómina mensual	\$9,900.00 / 2	\$4,950.00
Días en el periodo	16	16
Sueldo diario	\$4,950.00 / 16	\$309.38
Sueldo al final de periodo	\$4,950.00	

Fuente: Diseño Propio del Autor

Ejemplo 14 Diciembre - 28 Diciembre

**Tabla 5. Ejemplo nuevo cálculo de saldo segunda quincena**

Descripción	Cantidad	Total
Calculo mensual neto después de restricciones	\$18,000.00 * 75%	\$9,900.00
Calculo por tipo de nómina mensual	\$9,900.00 / 2	\$4,950.00
Días en el periodo	15	15
Sueldo diario	\$4,950.00 / 15	\$330.00
Sueldo al final de periodo	\$4,950.00	

Fuente: Diseño Propio del Autor

#### 4.4.2 Pago de servicios

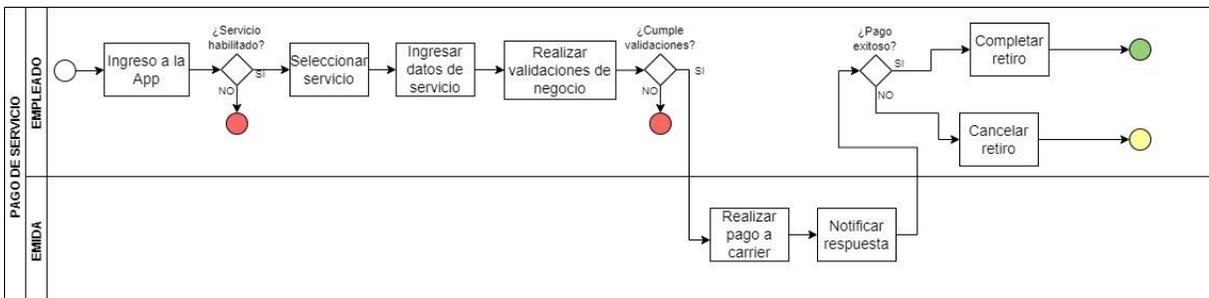
Los colaboradores podrán hacer uso del pago de servicio que les permite realizar recargas telefónicas y pago de plan telefónico. Se realizará con el saldo disponible del colaborador, por lo cual se aplicará la comisión que corresponde a la empresa donde está registrado el colaborador.

Para que un empleado pueda hacer uso de este servicio, la empresa a la que corresponde previamente debió haber activado el servicio, de lo contrario, no será visible la opción para realizar pagos de servicios.

La recarga solo se realizará siempre y cuando cuente con saldo suficiente para pagar el servicio más la comisión (la comisión no se aplica si cuenta con una suscripción o cupón) que aplique de acuerdo con su empresa.

Mientras que para PULPI, este se aplica como un retiro, por lo cual se aplica las validaciones existentes en el flujo de retiros.

**Figura 5. Flujo de retiros**



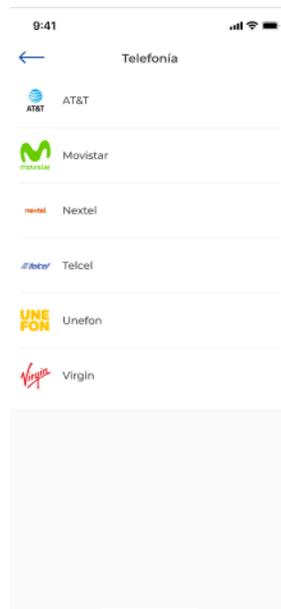
Fuente: Diseño Propio del Autor

Por parte de la aplicación tendrá el siguiente diseño al momento de querer hacer un pago de servicio. Primero se contará con la opción de servicio en la parte inferior y al seleccionar dicha opción se visualizará la opción de telefonía.

**Figura 6. Pantalla pago de servicios**

Fuente: Diseño Propio del Autor

Al seleccionar telefonía se visualizan las diferentes compañías de telefonía.

**Figura 7. Pantalla telefonía**

Fuente: Diseño Propio del Autor

Al elegir un proveedor de telefonía móvil se obtendrá la pantalla donde ingresa el número y el tipo de referencia.

**Figura 8. Pantalla ingresar número**

The screenshot shows a mobile application interface. At the top, the status bar displays the time 9:41, signal strength, and battery level. Below the status bar, there is a blue back arrow and the word 'Telefonía'. The Telcel logo is centered. Below the logo, the text reads 'Ingresa el número de referencia' followed by 'Introduce tu número de teléfono de 10 dígitos'. There are two input fields: the first is labeled 'No. de teléfono' and the second is labeled '\$00'. A blue 'Continuar' button is at the bottom.

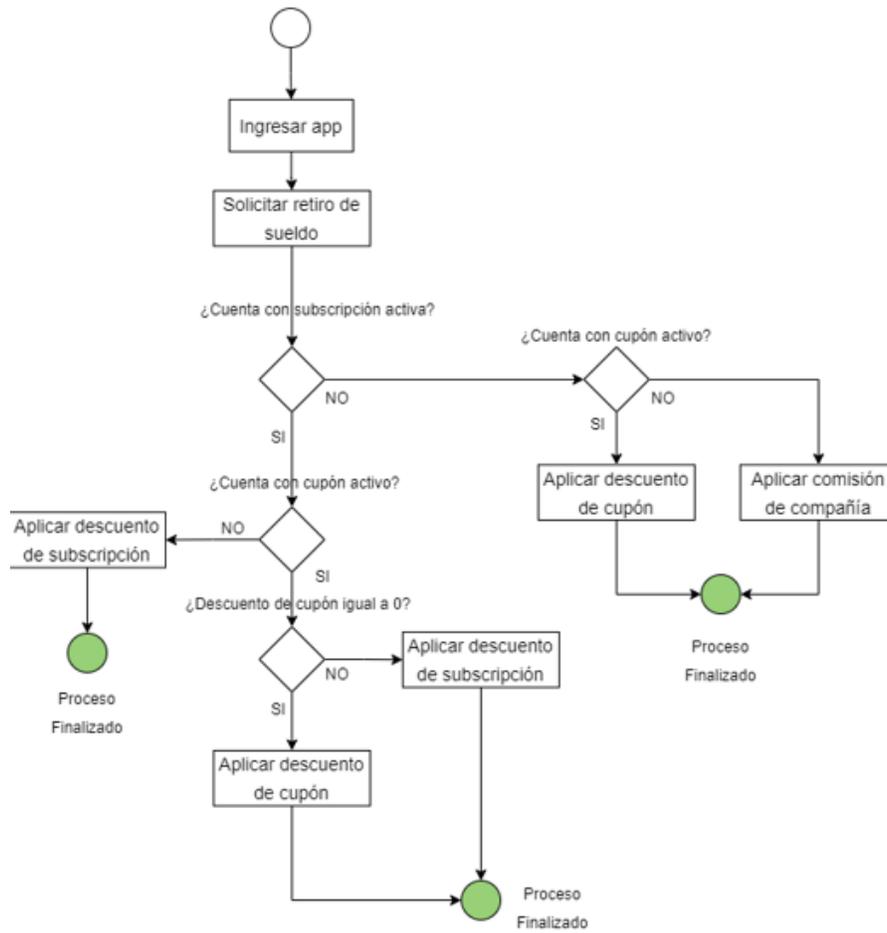
*Fuente: Diseño Propio del Autor*

#### 4.4.3 Suscripciones

La nueva funcionalidad de suscripciones pretende que los colaboradores dispongan de su sueldo sin tener que pagar la comisión que su empresa haya definido, realizando un pago mensual para contratar dicho beneficio el cual será renovado automáticamente siempre y cuando el colaborador cuente con sueldo disponible y no haya realizado una pre-cancelación del servicio durante el mes corriente.

La suscripción se definirá al crear una empresa o después en el detalle de la empresa, una vez ya se cuente con el servicio el colaborador podrá hacer uso de dicho servicio, pero en caso de que cuente con un cupón se aplicara el siguiente flujo.

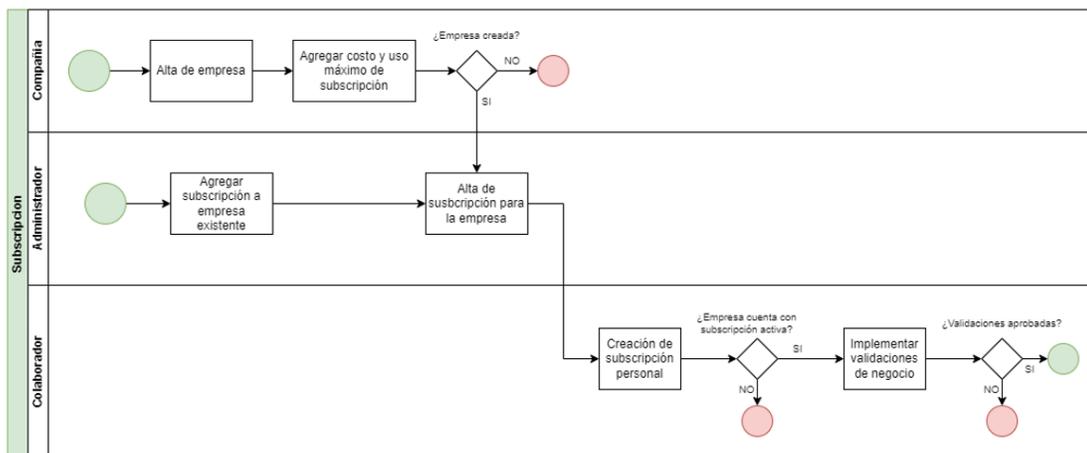
**Figura 9. Flujo suscripciones**



Fuente: Diseño Propio del Autor

Mientras que el flujo general de la suscripción se visualiza de la siguiente forma.

**Figura 10. Flujo general suscripciones**



Fuente: Diseño Propio del Autor

El primer requisito que se debe de cumplir es que la empresa realice la configuración del costo que tendrá este convenio para ser adquirido por sus colaboradores, al igual que el uso máximo permitido por colaborador.

Una vez cumplido el requisito anterior para que un colaborador de cualquier empresa pueda realizar la contratación del servicio, es necesario que su empresa tenga activado este beneficio.

Si la empresa cuenta con el beneficio, cada que el colaborador ingrese a la app se le mostrara la opción de realizar la contratación de servicio siempre y cuando no cuente con una suscripción activa.

**Figura 11. Pantalla suscripción**



*Fuente: Diseño Propio del Autor*

En esa pantalla el colaborador puede activar su suscripción, contando con una suscripción el inicio se mostrará de la siguiente forma.

**Figura 12. Pantalla inicio con suscripción**

Fuente: Diseño Propio del Autor

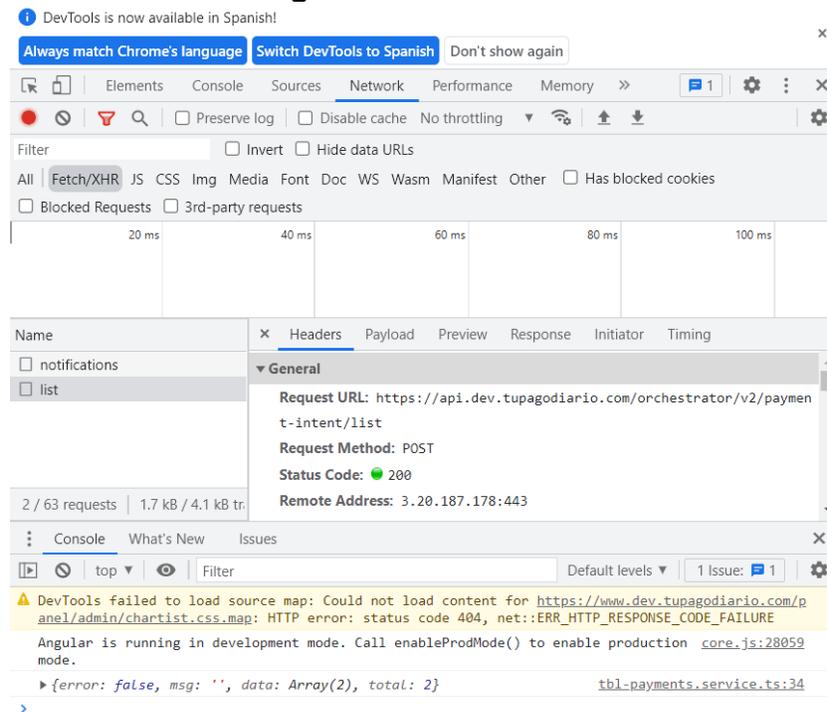
De esa forma al retirar el campo de comisión lo debe de marcar en \$0.

Para iniciar con la implementación primero se debe de crear un nuevo módulo y para consumir el servicio se debe de mandar a llamar la liga postman e implementar el funcionamiento del servicio para eso se debe de revisar si estamos trabajando con un POST o un GET. La diferencia radica en la forma de enviar los datos a la página cuando se presiona un botón. Mientras que el método GET envía los datos usando la URL, el método POST los envía de forma que no se pueden visualizar (en un segundo plano u ocultos al usuario).

Para estos casos se hace uso de los dos, pero depende si se desea que se muestre un dato o realizar una operación, creando un servicio donde se mande a llamar la liga de Postman, pero también se toma en cuenta la forma en la que se están mandado los datos ya sea en un arreglo u otro tipo de dato.

Se observa si se está consumiendo correctamente el servicio, alguno de los estados que puede regresar es un 200 que significa que se está consumiendo correctamente un servicio. Otro de los estados es un 400 donde indica que existe algún error al consumir el servicio, pero también puede darse que al momento de crear el microservicio falle el token o algún motivo relacionado con un bug o error de programación.

Figura 13. Estado 200



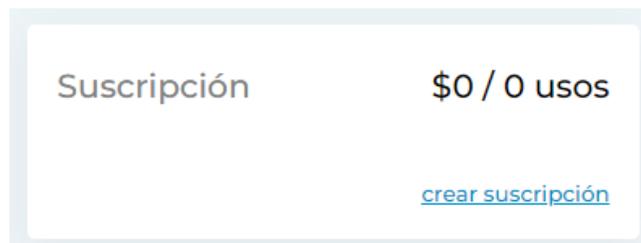
Fuente: Diseño Propio del Autor

Como se muestra en la consola el primer dato que manda es la URL del servicio que se consume después el tipo de dato que en este caso es un POST y por último el estado 200 que indica que se está consumiendo correctamente el servicio.

En Postman se puede ocupar varios entornos que pueden ser local, development y staging. Existen otros tipos de entorno, pero estos tres son los principales a utilizar.

Pasando al detalle de la empresa es donde se agrega una nueva tarjeta donde hablara el costo por suscripción. Este tendrá dos estados donde la empresa no cuenta con una suscripción desde el inicio entonces la tarjeta muestra \$0 en suscripción y en uso como se muestra en la figura 14.

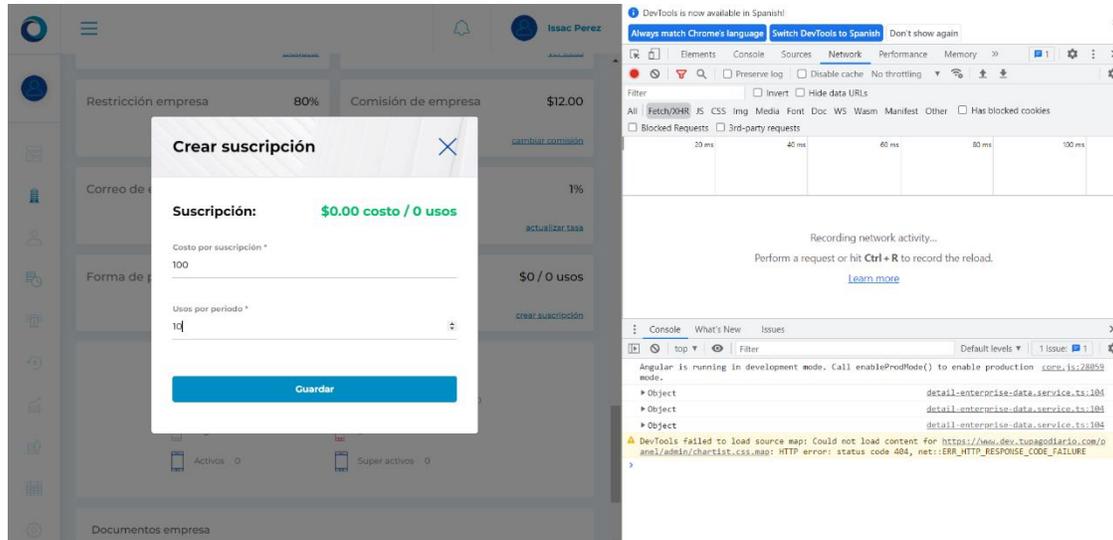
Figura 14. Tarjeta suscripción



Fuente: Diseño Propio del Autor

Se muestra la leyenda de crear suscripción en ese caso se puede crear a través de la modal, donde se asigna el costo y uso.

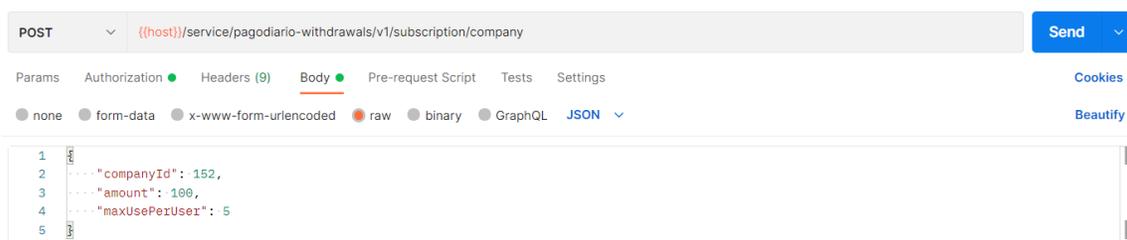
**Figura 15. Modal suscripción**



*Fuente: Diseño Propio del Autor*

Para crear una suscripción se hace uso de un Post por medio de Postman, este permite guardar datos para ello se envía a llamar el servicio y es necesario ver como se manda los datos en esta ocasión es del tipo array y se debe de ver que datos se hace llegar para que se cree la suscripción. Entonces en este servicio empleado para crear un elemento es necesario mandar 3 datos para la base de datos, principalmente que se haga referencia al id de la compañía.

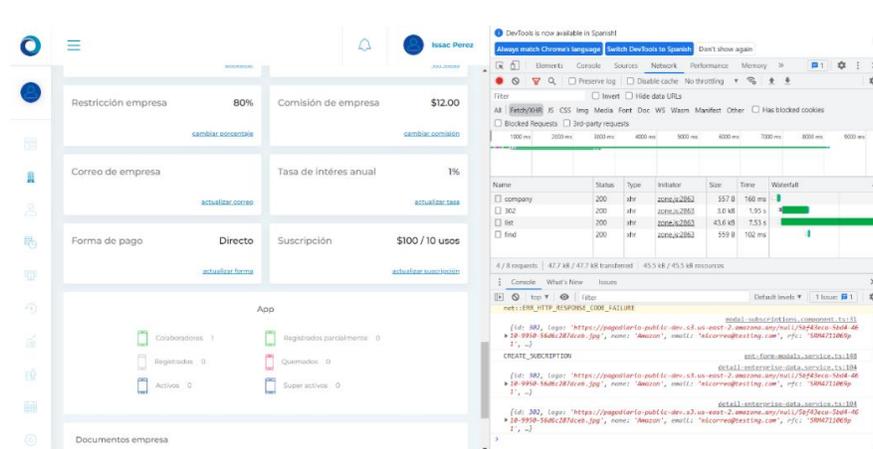
**Figura 16. Servicio**



*Fuente: Diseño Propio del Autor*

Una vez que ya se tiene hecho el servicio se puede probar el funcionamiento de este atreves de la consola del navegador.

Figura 17. Funcionamiento por consola



Fuente: Diseño Propio del Autor

Cuando una empresa ya cuenta con una suscripción la tarjeta cambia su leyenda por actualizar suscripción y además ya se muestra el costo y uso de la suscripción.

Figura 18. Empresa con suscripción



Fuente: Diseño Propio del Autor

Al presionar en actualizar suscripción se muestra que algunas cosas del modal cambiaron por actualizar suscripción, además que ya podemos ver el costo y uso.

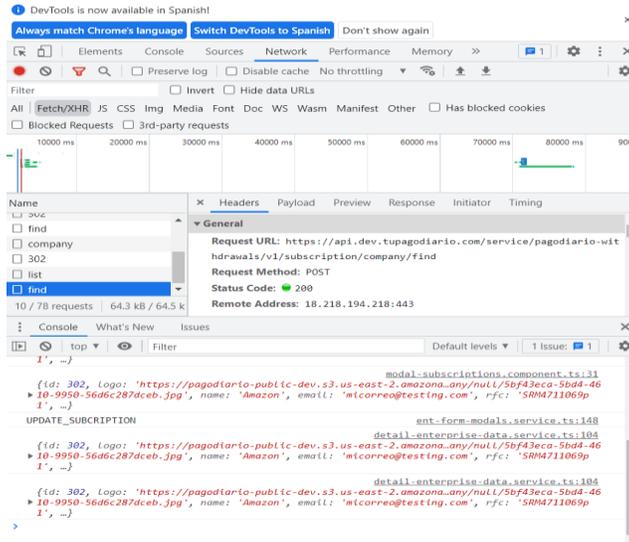
Figura 19. Actualizar suscripción



Fuente: Diseño Propio del Autor

Para los Update se ocupa otro tipo que es el PUT permite modificar algún registro para el funcionamiento debe de ocupar el tipo JSON de igual forma los principales estatus a mandar es un 200 o un 400.

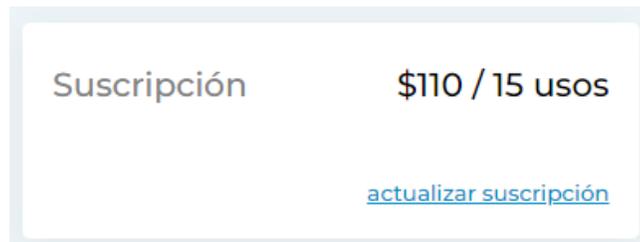
Figura 20. Update consola



Fuente: Diseño Propio del Autor

Y en la tarjeta como en el modal los cambios se ven reflejados.

Figura 21. Actualización suscripción



Fuente: Diseño Propio del Autor

Por último, en crear empresa se utiliza el mismo servicio de crear suscripción, pero ahora nada más va a agregar los campos de costo y uso en el formulario de nueva empresa.

Figura 22. Campos en formulario

Costo por suscripción	Usos por periodo de la suscripción
Ej. 100	Ej. 10

Fuente: Diseño Propio del Autor

De igual forma hace uso de un Post para crear una nueva suscripción al momento de crear una nueva empresa.

## 4.5 Pruebas unitarias y pruebas de QA

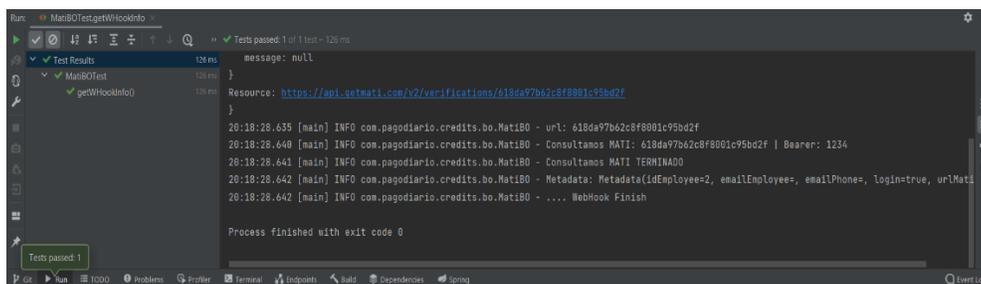
### 4.5.1 Pruebas unitarias

Las pruebas unitarias se basan en dos tipos diferentes eso depende si está trabajando en backend o fronted se realizaron las pruebas necesarias para ambos ambientes. En el caso de backend se ocupó principalmente Mockito, para crear objetos simulados, con el propósito de realizar las pruebas, se realizan pruebas a métodos que se crearon.

En la mayoría de las pruebas se ocupa la sintaxis de Mockito when, nos permitirá realizar pruebas del tipo condicional esto dará una comprobación de tipo del valor que está devolviendo, en el momento de la compilación. Ayuda a que si está realizando mal la prueba y se ejecuta regresara rápido el error.

Lo ideal es realizar todas las pruebas necesarias para cubrir con la historia de usuario, pero principalmente necesita hacer de los dos tipos de estados donde se ocupa un 200 y otro que sea fallido. En la figura 23 se observa una prueba unitaria exitosa.

Figura 23. Prueba unitaria



Fuente: Diseño Propio del Autor

Estas pruebas se fueron realizando por medio de IntelliJ idea junto con Mockito además de que para realizar unas pruebas de forma local se hizo uso de Insomnia, tiene un funcionamiento similar que Postman.

Ahora las pruebas unitarias para fronted se hace uso de Jasmine y Karma. Jasmine es un framework de Javascript para la definición de tests usando un lenguaje de tipo natural. Las pruebas se aplican a HTML, servicios y componentes, veremos las más importantes en servicios.

Las pruebas a servicios van a llamar los servicios que creamos y simular datos, es decir que si quiere guardar algo se deben de cumplir las condiciones de tener id o los campos que solicite, además manda a llamar al backend por medio de la liga del servicio.

**Figura 24. Ejemplo prueba unitaria backend**

```
it(
  'expectations: "should be able to get all Scheduled Subscriptions by collaborator id"',
  waitForAsync(
    inject(
      [ScheduledWithdrawalsService, HttpTestingController],
      function(
        service: ScheduledWithdrawalsService,
        backend: HttpTestingController
      ) => {
        const id = 1;
        const params = null;
        service
          .getAllScheduledSubscriptions(id, params)
          .subscribe(next: (next: []) => {
            expect(next).toBeTruthy();
          });
        backend
          .expectOne(
            url: `${service.$api__routes.PANEL_COLLABORATORS.WITHDRAWAL_LIST}`
          )
          .flush(
            body: { total: 0, data: [] },
            opts: { status: 200, statusText: 'Ok' }
          );
      }
    )
  );
```

Fuente: Diseño Propio del Autor

Cuando se tenga la prueba solo va a hacer falta ejecutarla con el motivo de verificar que este haciendo todo lo indicado además que veremos si esta prueba pasa o no.

**Figura 25. Prueba por medio de Karma**

Karma v 6.3.9 - connected; test: complete; DEBUG

Chrome 97.0.4692.71 (Windows 10) is idle

Jasmine 3.5.0 Open logs

Ran 1 of 124 specs - run all Finished in 1.30s

1 spec, 0 failures, randomized with seed 81276

```
RefundService
  • should be able to get the refunds of enterprises
  • should be able to get the refunds of enterprises but get status 500
  • should be able to change status of refund
  • should be able to get one refund by id
  • should be able to change status of refund but return 500
  • should be able to get one refund by id
  • should be able to change status of refund
  • should be able to get one refund by id
  • should be able to change status of refund
  • should be able to get one refund by id but get status 500
  • should be able to change status of refund
  • should be able to add description to refund
  • should be able to get history of refund
  • should be able to add description to refund but return status 500
  • should be able to get the history of refund but return status 500
  • should be able to construct last refunds
  • should be created

TorusandConditionComponent
  • should be created

AplicacionService
  • should be created

Tblind
  • should be created

TorusandConditionService
  • should be created
  • should be able to get all terms and condition but return status 500
  • should be able to create new terms and condition but return status 500
```

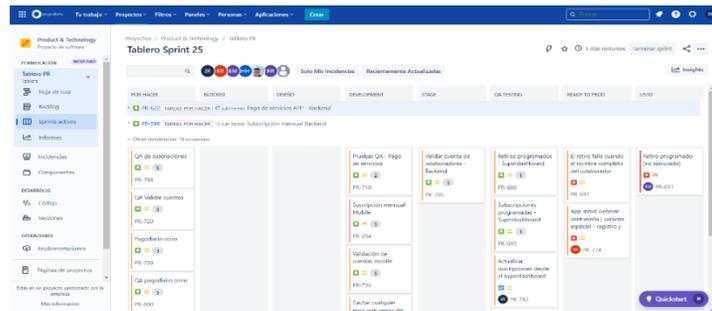
Fuente: Diseño Propio del Autor

Como resultado se obtiene unas pruebas correctas y funcionales.

### 4.5.2 Pruebas QA

Las pruebas de QA son un compromiso de la calidad en los productos digitales. Se realiza la comprobación del correcto funcionamiento del software, el objeto de QA es detectar los errores en el menor tiempo posible. Para realizar las pruebas de QA se debe consultar primero el tablero de Jira que es la herramienta que se ocupó para la metodología.

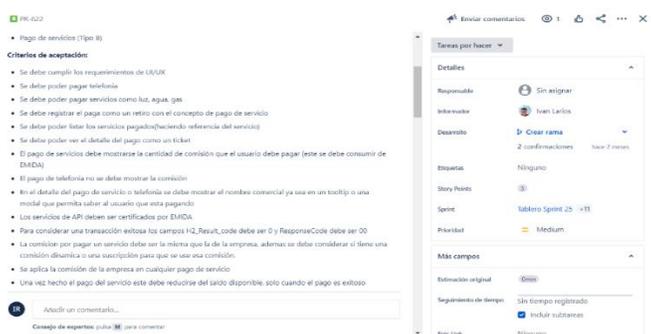
Figura 26. Tablero Jira



Fuente: Diseño Propio del Autor

Se revisa los criterios de las tareas y se revisa que el producto final cuente con todas las características.

Figura 27. Criterios tareas



Fuente: Diseño Propio del Autor

Este proceso se realiza con cada nueva versión que llegue a salir por cada cambio.

Se realizan casos de pruebas que son un conjunto de condiciones o variables las cuales determinan si una aplicación, software o una característica sea aceptable o no aceptada. Para la revisión de este servicio como la empresa cuenta con una versión de Android y IOS, se puede revisar la aplicación para ambos casos ya que puede contar con la misma funcionalidad pero los ambientes y diseño son diferentes pudiendo modificar

uno u otro al momento de implementar un nuevo servicio, en la siguiente figura veremos la revision para un Android.

**Tabla 6. Pruebas de QA**

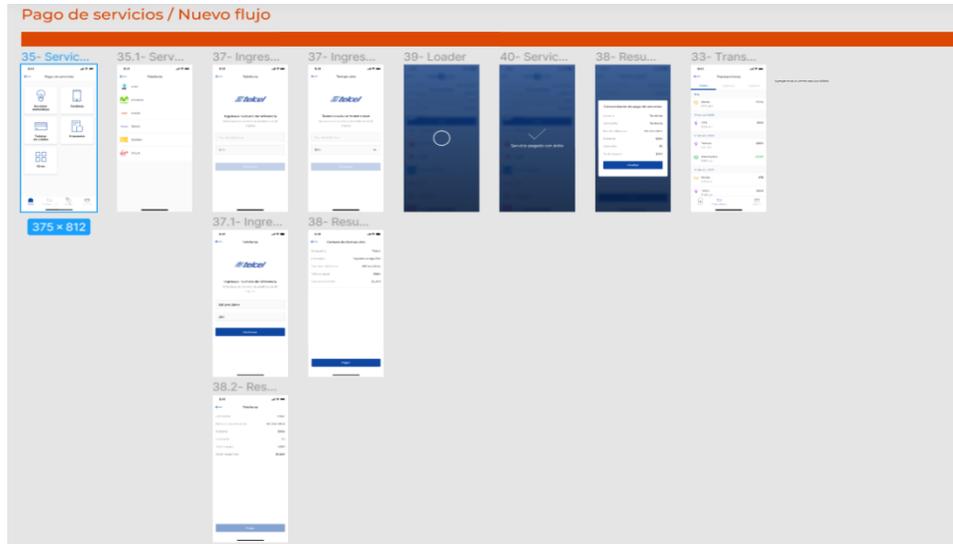
CP App movil Pago de servicios - 622					Primera fase (dev)	
NO		Nombre/Modulo	Precondicines	Pasos	estado	fecha
1	1.1	Pago de servicios (Telefonía - Telcel)	Ingresar a la app móvil	Se muestra el modulo "inicio" con las opciones: * Perfil * Notificaciones * El saldo acumulado * Monto a retirar * Concepto * Próxima echa de pago * Días trabajados * Retiro programado (en caso de tener activos)	Pasado	15/12/2021
	1.2		Ir al modulo Servicios	Se muestran los servicios a pagar como: * Servicios domésticos * Telefonía * Tarjetas de crédito	Pasado	15/12/2021
	1.3		Ingresar a la opción "telefonía"	Se muestra el listado de teléfonos disponibles: * Movistar * Telcel * Virgin	Pasado	15/12/2021
	1.4		Seleccionar "Telcel"	Se muestra: * Logo de Telcel * Cuadro de texto para ingresar el numero telefónico a 10 dígitos * Concepto a pagar * Botón "continuar"	Pasado	15/12/2021
	1.5		Ingresar un numero telefónico y opción "Seleccionar concepto"	Se muestran la opción de elegir entre una recarga o un paquete	Pasado	15/12/2021
	1.6		Seleccionar opción "recarga"	Se muestra la opción "recarga" seleccionada y se habilita una nueva opción llamada "Seleccionar Concepto"	Pasado	15/12/2021
	1.7		ir a la opción "Seleccionar Concepto"	Se muestran las cantidades a elegir para realizar la recarga: * \$30 * \$50 * \$100	Pasado	15/12/2021
	1.8		Seleccionar una de las cantidades a recargar	Se habilita el botón "Continuar"	Pasado	15/12/2021
	1.9		Digitar botón "Continuar"	Se muestra el resumen de la transacción con los datos: * Compañía * Numero de referencia * Subtotal * Comisión * Total a pagar * Saldo disponible * Botón "Pagar"	Pasado	15/12/2021
	1.10		Seleccionar la opción "Pagar"	Se muestra el mensaje de "Servicio pagado con éxito" y posteriormente el comprobante de pago de servicio con los datos: * Servicio * Compañía * Numero de referencia * Subtotal * Comisión * Total a pagar * Botón "Finalizar"	Pasado	15/12/2021
	1.11		Seleccionar botón "Finalizar"	Se muestran los servicios a pagar como: * Servicios domésticos * Telefonía * Tarjetas de crédito	Pasado	15/12/2021

Fuente: Diseño Propio del Autor

Se revisa cosa por cosa, en el ejemplo es pago de servicio telefonía móvil donde se tiene caso y se va desarrollando, con precondition, pasos, resultados esperados de cada paso y por ultimo las fases de prueba donde tenemos las de cada ambiente (staging, staging con suscripción, staging sin suscripción y producción).

Se comprueba que las pantallas realizadas en figma sean las mismas que se realizaron y se verifica que se cuente con un uso similar a las de figma.

**Figura 28. Pantallas en figma**



Fuente: Diseño Propio del Autor

En los casos de que se llegó a encontrar algún error, se crea en el tablero de Jira una subtarea donde se diga si existe un bug, error o algún cambio. Si esta fase ya está completa se pasa a realizar las pruebas de sanity o sanidad, son pruebas de software que se realizan después de la compilación del software con el motivo de verificar que los errores se hayan solucionado y que no se introduzcan problemas adicionales debido a estos cambios.

Tabla VII. Prueba de sanidad

ID	Modulo	Escenario	Plataforma	Fecha
25	Pago de servicios	Realizar un pago de servicio	Android / IOS	Pasado
26	Suscripciones	Que me deje hacer una suscripción	Android / IOS	Pasado

Fuente: Diseño Propio del Autor

Cuando el producto pasa todas las pruebas ya se puede realizar la liberación del producto final hasta el momento.

## Capítulo 5 Resultados

### 5.1 Introducción

El propósito de este capítulo llamado Resultados es mostrar los resultados obtenidos como lo son el correcto funcionamiento de la aplicación y el super dashboard.

### 5.2 Análisis e interpretación de resultados

Como resultado final se tiene el correcto funcionamiento de los dos nuevos servicios y la mejora del nuevo cálculo de saldo, el cual no cuenta con una vista como tal, pero el cambio se ve reflejado en el soporte, ya que un gran parte era sobre colaboradores que contaban con saldo negativo.

Mientras que pago de servicios se cuenta con su correcto funcionamiento donde se comporta como un retiro común y el cual se puede consultar a través del super dashboard.

**Figura 29. Pago de servicio en dashboard**

Folio	Monto total	Fecha	Hora	Concepto	Estatus	Tarjeta / Cuenta
PD1642168814035	\$210.00	14 enero 2022	08:00	imprevistos	En proceso	*****5256
PD1642168809329	\$160.00	14 enero 2022	08:00	pago de servicios	En proceso	*****5256
SUBS1642115409895-67	\$100.00	13 enero 2022	17:10	Subscripción	Pagado	*****7885
PD1642115354776	\$86.00	13 enero 2022	17:09	adelanto de nomina	En proceso	*****5287
1642115308180	\$96.00	13 enero 2022	17:08	Pago de Servicio Vig...	Pagado	*****5287
SUBS1642108958805-66	\$100.00	13 enero 2022	15:22	Subscripción	Pagado	*****4321
1642108564640	\$96.00	13 enero 2022	15:16	Pago de Servicio Vig...	Pagado	*****5287
PD1642108489538	\$46.00	13 enero 2022	15:14	adelanto de nomina	En proceso	*****5287

Fuente: Diseño Propio del Autor

Se observa en concepto como pago de servicio, mientras por medio de app se tiene la vista del recibo de la siguiente forma.

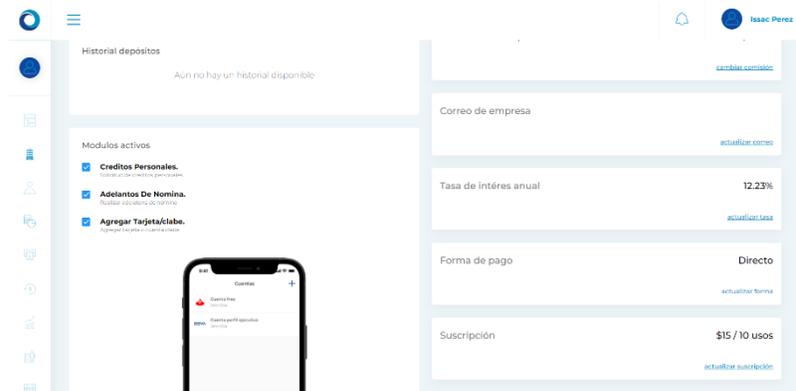
Figura 30. Recibo pago de servicio



Fuente: Diseño Propio del Autor

Por ultimo las suscripciones se pueden activar por medio del super dashboard o al momento de crear una nueva suscripción, cuando una empresa ya cuenta con dicho servicio activo se puede visualizar dentro del detalle de la empresa.

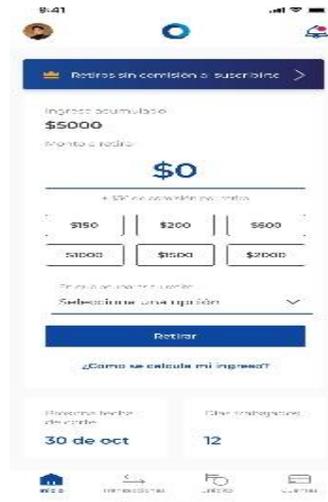
Figura 31. Activar suscripción



Fuente: Diseño Propio del Autor

Mientras por medio de la aplicación se visualiza como un usuario cuenta con la suscripción y con comisiones en \$0.

Figura 32. Usuario con suscripción



Fuente: Diseño Propio del Autor

También por medio del dashboard en el detalle del colaborador se puede ver los que cuentan con suscripción.

Figura 33. Detalle colaborador suscripción

Subscripciones programadas

Folio	Adquisición	Expiración	Renovación
190	28 diciembre 2021 10:21	28 diciembre 2021 10:21	\$0.00

Mostrando 1 - 1 retiros

Fuente: Diseño Propio del Autor

## Conclusiones

La mejora en cálculo de saldo es de mucha ayuda para la empresa, ya que presenta un riesgo bajo al momento de realizar los cortes de cada periodo permitiendo que la empresa no cuente con pérdidas monetarias además de que contribuyo a la reducción de soporte por parte de operaciones. Permitir que el usuario tenga disposición de su salario sin ningún problema alguno. Mientras por la parte de pago de servicios es un gran paso para la compañía permitiendo la compra de tiempo aire o paquetes por medio de su salario ya trabajado

Por último, la implementación de suscripciones es un gran beneficio para los colaboradores, permitiéndoles realizar sus retiros o pago de servicios sin comisiones, solo pagando una mensualidad que define cada empresa, la cual cuenta con límite de usos, suponiendo que tiene cinco usos y al pasar al sexto se le cobrara la comisión definida de su empresa y esta suscripción la puede renovar al siguiente mes.

El proyecto es muy grande y con un enorme potencial de ser unicornio, es impresionante con lo que hoy en día se cuenta, pero si hace falta mejorar algunos detalles mínimos, por parte de super dashboard hace falta encontrar una forma de que al momento de ir con los colaboradores sea un poco más rápido de igual forma al momento de buscar este puede tardar un rato en devolver una respuesta.

La empresa tiene un potencial de ocupar el modelo B2C que se refiere a la estrategia que desarrollan las empresas comerciales para llegar directamente al cliente o consumidor final. Es decir que ya no va a ser necesario contar con un convenio con empresas para que un empleado pueda hacer uso de este servicio financiero.

## Recomendaciones

En esta sección, se presenta aspectos pendientes de explorar que se identificaron como parte del desarrollo de la tesis y podrían ser motivo de futuras investigación.

El primer punto identificado es la necesidad de realizar experimentos controlados para medir el impacto cuantitativo de la integración de las técnicas ágiles con usabilidad y validar los hallazgos del presente estudio.

Por otra parte, dado que el caso presenta un contexto particular de un único desarrollador y una aplicación móvil que tiene un alcance pequeño, sería importante realizar más casos de estudio donde se utilicen aplicaciones de mayor alcance y se apliquen a equipos de desarrollo.

Hacer uso del modelo B2C para poder llegar a un público más amplio y no depender mucho de convenios con empresas, actualmente se hace uso del modelo B2B que son negocios de empresa a empresa.

Finalmente, un punto pendiente por explorar sería la forma de integrar técnicas de usabilidad con otros frameworks ágiles crecientes como Kanban.

## Referencias

- A. (2019, 9 diciembre). Introducción a las Pruebas Unitarias (Unit Testing) con JUnit y Mockito. Java desde 0. Recuperado 17 de enero de 2022, de <https://javadesde0.com/introduccion-a-las-pruebas-unitarias-con-junit-y-mockito/>
- A., D. (2021, 29 julio). ¿Qué es Angular? - Una guía para principiantes. Tutoriales Hostinger. Recuperado 17 de enero de 2022, de <https://www.hostinger.mx/tutoriales/que-es-angular>
- Descripción general de Kotlin | Desarrolladores de Android |. (2019, 27 diciembre). Android Developers. Recuperado 17 de enero de 2022, de <https://developer.android.com/kotlin/overview?hl=es>
- Núñez-Torrón Stock, A. (2021, 11 mayo). ¿Qué es una API y para qué sirve? Todo lo que necesitas saber. Business Insider España. Recuperado 17 de enero de 2022, de <https://www.businessinsider.es/api-sirve-todo-necesitas-saber-861403>
- Qué es Fecha de CORTE y PAGO en Tarjeta de Crédito 2020. (2020, 5 febrero). Kardmatch. Recuperado 17 de enero de 2022, de <https://blog.kardmatch.com.mx/que-es-la-fecha-de-corte-de-tu-tarjeta-de-credito#como-funciona-la-fecha-de-corte-y-pago>
- ¿Qué son los microservicios? (2018, 13 abril). Red hat. Recuperado 17 de enero de 2022, de <https://www.redhat.com/es/topics/microservices/what-are-microservices>
- Y. (2021, 11 abril). ¿Qué es Spring Initializr? SACAViX Tech Blog. Recuperado 17 de enero de 2022, de <https://sacavix.com/2021/04/11/que-es-spring-initializr/#:%7E:text=Spring%20Initializr%20es%20una%20API,de%20arranque%20de%20nuevos%20proyectos.&text=API%3A%20Interfaz%20que%20permite%20se,Eclipse%20STS%2C%20Intellij%20y%20otros>
- "Conexionesan". (24 de 10 de 2016). conexionesan. Obtenido de <https://www.esan.edu.pe/apuntesempresariales/2016/10/la-administracion-delportafolio-de-proyectos/>
- Bara, M. (s.f.). OBS Bussines School. Obtenido de <https://www.obs-edu.com/int/bloginvestigacion/project-management/las-5-etapasen-los-sprints-de-un-desarrollo-scrum>
- Cisneros, R. (10 de Enero de 2019). "Scrum": agilidad, colaboración, iteración e incremento. Obtenido de El Periódico: <https://elperiodico.com.gt/opinion/2019/01/08/s-crum-agilidad-colaboracion-iteracion-eincremento/>
-

- Francia, j. (25 de 09 de 2017). Scrum.org. Obtenido de <https://www.scrum.org/resources/blog/que-esscrum>
- Management, R. e. (Diciembre de 2018). ¿Qué es una metodología ágil? Obtenido de Recursos en Project Management: <https://www.recursosenprojectmanagement.com/metodologia-agil/>
- Hopcroft, J., Motwani, R., & Ullman, J. (2008). Teoría de autómatas, lenguajes y computación. Pearson.
- 24, I. (07 de Febrero de 2018). Google Play. Obtenido de Google Play: <https://play.google.com/store/apps/details?id=com.andromo.dev597108.app690294>
- Alacarta. (07 de Febrero de 2018). Google Play. Obtenido de Google Play: <https://alacartamexico.com/>
- Almazán Morales, J. L., Méndez Salvatorio, A., & Lee kim, H. S. (2013). ¿UNA HERRAMIENTA INNOVATIVA PARA LAS PYMES? INCEPTUM, 85-109.
- Appstaurante. (07 de Febrero de 2018). Appstaurante. Obtenido de Appstaurante: <http://www.appstaurante.com>
- Bautista, G. B. (s.f.). Clasificación de las empresas en México. Obtenido de Universidad Autónoma del Estado de Hidalgo: <https://www.uaeh.edu.mx/scige/boletin/prepa4/n5/m14.html>
- Bellman, R. (1978). Functional equations in dynamic programming.
- Booch, G., Rumbaugh, J., & Jacobson, I. (1999). Unified Modeling Language. Madrid, España.
- Bucanek, J. (2009). Model-View-Controller Pattern. Apress.
- C. B., & G. K. (2005). Hibernate in action. Greenwich: Manning.
- Capó Vicedo, J., Exposito, Lanza, M., & Mesia Buades, E. (2007). La importancia de los clusters para la competitividad de las PYME en una economía global. Eure, 119-133.
- Cheung, B. (07 de Febrero de 2018). Google Play. Obtenido de Google Play: <https://play.google.com/store/apps/details?id=com.bong.BillCalculator3>
- Chomsky, G. (1956). Chomsky y los lenguajes formales. Ochoa.
-

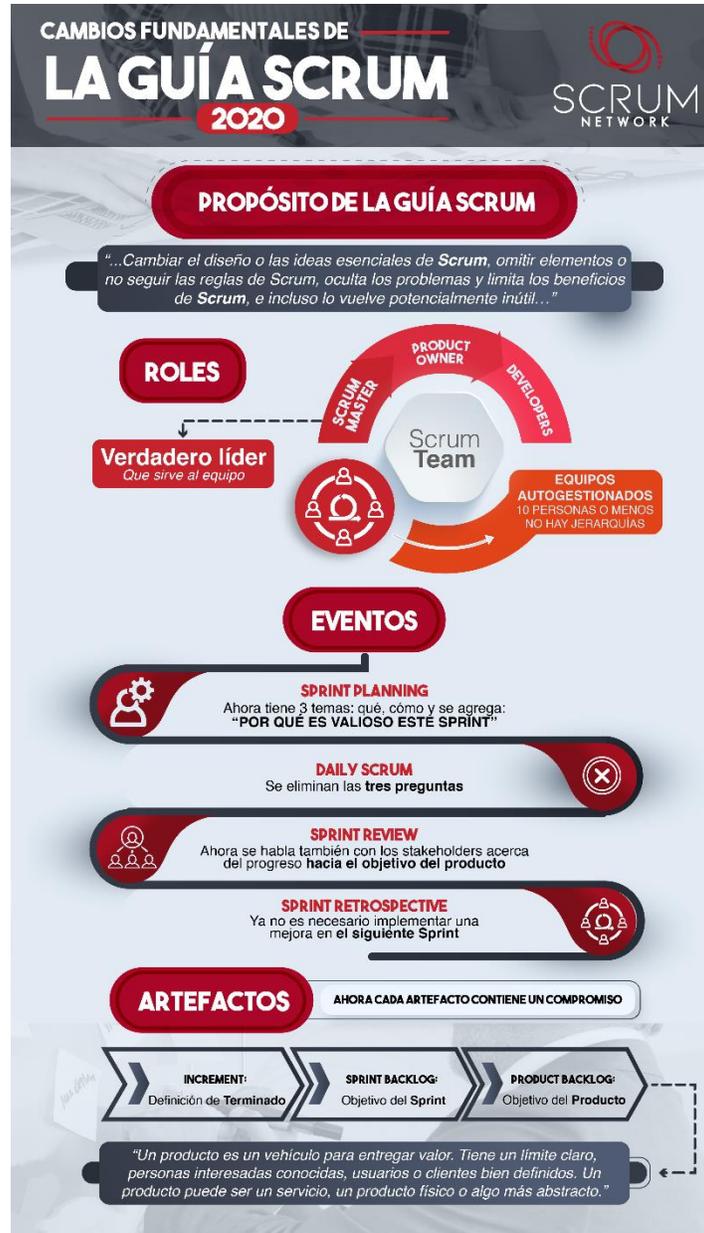
- Cleri, C. (2014). El libro de las PyMEs. Granica.
- COTEC. (1999). Pautas metodológicas en la gestión de la tecnología y la innovación para empresas. Fundación COTEC.
- Cruz, F., & López Mortero, G. (2007). Una visión general del m-learning y su proceso de adopción en el esquema educativo. Coloquio internacional Tendencias Actuales de Cómputo e Informática en México.
- D. R., & C. R. (2012). Programación en android. España: Aula Mentor.
- Domínguez. (2009). Innovación y competitividad de PYMES, la experiencia internaciona. Benemérita Universidad Autónoma de Puebla.
- Elmasri, R., & Navathe, S. (2007). Fundamentos de sistemas de bases de datos, Quinta Edición. Pearson
- Financiero, E. (12 de Febrero de 2014). El Financiero. Obtenido de El Financiero: <http://www.elfinanciero.com.mx/empresas/en-mexico-solo-de-las-pymes-usan-tecnologiasde-la-informacion.html>
- G. A., & M. O. (2010). The Definitive Guide to SQLite, Second Edition. New York: Apress.
- Garnica González, J., & Nuño de la Parra, J. P. (2011). Una visión de la innovación como elemento clave para mejorar la competitividad en las PyMEs mexicanas. Red Internacional de Investigadores en Competitividad.
- Google. (20 de Marzo de 2018). Google Cloud. Obtenido de Google Cloud: <https://cloud.google.com/speech-to-text/docs/basics>
- Grune, D., & Jacobs, C. (2008). Parsing Techniques. New York: Springer.
- GRUPO, I. S. (2003). Metodologías Ágiles en el Desarrollo de Software. Jornadas de Ingeniería del Software y Bases de Datos.
- Gutiérrez Suarez, C. (2017). Google play game services. Madrid, España. indalquiler. (07 de Febrero de 2018). Google Play. Obtenido de Google Play: [https://play.google.com/store/apps/details?id=com.mobincube.appdomicilio.sc\\_HWB3XU](https://play.google.com/store/apps/details?id=com.mobincube.appdomicilio.sc_HWB3XU)
- INEGI. (2014). Censos económicos 2014.
- INEGI. (2016). Censos económicos 2014, La industria restaurantera en México. Aguascalientes, México.: INEGI.
- J. K. (2010). Usung SQLite. Sebastopol, CA: O'Reilly.
-

- Joyanes, L., & Zahonero, I. (2011). Programación en Java. Mc Graw Hill.
- Joyanes, L., Rodriguez, L., & Fernández, M. (1996). Fundamentos de Programación, Libro de problemas. Mc Graw Hill.
- Kenneth, R. (2012). Essential Scrum. Addison-Wesley.
- Kinvey. (2016). How long does it take to build a mobile app? Obtenido de <https://www.kinvey.com/how-long-to-build-an-app-infographic/>
- Kohen karen, D., & Asín Lares, E. (2009). Tecnologías de información en los negocios. México, DF: Mc Graw Hill.
- Kurzwil, R. (1990). The age of intelligent machines. Cambridge: MIT Press.
- Letelier, P. (2006). Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP).
- López, F. (2008). El Lenguaje Objective-C para programadores C++ y Java.
- Maldonado Guzmán, G., Martínez Serna, M., Garcia Pérez, D., Aguilera Enríquez, L., & González Adame, M. (2010). La influencia de las TICs en el rendimiento de la PyME de Aguascalientes. Investigación y Ciencia de la Universidad Autónoma de Aguascalientes, 57-65.
- Mannino, M. (2007). Administración de bases de datos, diseño y desarrollo de aplicaciones. Mc Graw Hill.
- McLaughlin, B. (2002). Building Java Enterprise Applications, Vol I: Architecture. O Reilly.
- Moreno Aroca, A. M. (2014). USO DE TECNOLOGÍAS DE INFORMACIÓN Y SU INCIDENCIA EN EL CRECIMIENTO DE LAS PYMES DEL SECTOR TEXTIL DE GUAYAQUIL. Guayaquil, Ecuador.
- N. B. (2013). Android 4 Principios de desarrollo de aplicaciones Java. España: ENI Ediciones.
- Narvin. (07 de Febrero de 2018). Google Play. Obtenido de Google Play: <https://play.google.com/store/apps/details?id=com.narvin.android.easytaxcalculator>
- Oropeza, L., & Suárez, S. (2006). Algoritmos y métodos para el reconocimiento de voz en español mediante sílabas. Computación y sistemas, 270-286.

- Oz, E. (2008). Administración de los sistemas de la información. México, DF: CENGAGE Learning.
- Pantaleo, G., & Ludmila, R. (2014 ). INGENIERIA DE SOFTWARE. Alfaomega.
- Porter, M. (1990). The Competitive Advantage of Nations. New York: The Free Press.
- Pressman, R. (2010). Ingeniería del software, Un enfoque práctico. Mc Graw Hill.
- Pressman, R. (2010). Ingeniería del Software, un enfoque práctico. México DF: Mc Graw Hill.
- Proyectos Agiles. (26 de Julio de 2017). Proyectos Agiles. Obtenido de <https://proyectosagiles.org/que-es-scrum/>
- R. S., & S. T. (2015). Mobile Learning: Nuevas realidades en el aula. OCEANO.
- Ramez, E., & Shamkant, N. (2007). Fundamentos de sistemas de bases de datos. Pearson.
- Ramos, J., Ramos, A., & Montero, F. (2006). Sistemas Gestores de Bases de Datos. Madrid: Mc Graw Hill.
- Ravenna, M. C., & González, L. (2004). Las tecnologías de la información en las pequeñas y medianas empresas mexicanas. México: Scripta Nova.
- Ravi Sethi, A., & Ullman, J. (1986). Compiladores, principios, tecnicas y herramientas.
- Reyes Bascuñana, A., & Satofimia Ruiz, J. (s.f.). Inteligencia artificial aplicada a personas con discapacidad.
- Soriano. (2007). PYMES y EMPRENDEDORES. Obtenido de <http://www.pymesyemprendedores.com.mx/consultorio.html>
- Tanenbaum, A. S. (2009). Sistemas operativos modernos 3ra Edición. México: Pearson.
- TheFork, L. /. (07 de Febrero de 2018). Google Play. Obtenido de Google Play: <https://play.google.com/store/apps/details?id=com.lafourchette.lafourchette>
- Wilson, L. (1993). Comparative Programming Languages. Addison-Wesley

## Anexos

### Anexo 1. Guía Scrum



Fuente: <https://scrumcolombia.org/docs/Actualizacion-Guia-Scrum-2020.png>

## Anexo 2. Manifiesto Scrum



Fuente: <https://scrumcolombia.org/docs/Aniversario-Manifiesto-Agil.jpg>

## Glosario

<b><i>Burndown Chart</i></b>	El gráfico de Burndown muestra el trabajo restante en el tiempo. El trabajo restante está definido por los puntos de los PBI en eje Y y el tiempo en el eje X
<b><i>Capacidad del equipo</i></b>	Es una medida de la cantidad de trabajo que puede hacer el equipo en un Sprint. Se deben considerar la cantidad de personas, los días efectivos de trabajo, las reuniones y la disponibilidad de cada persona
<b><i>Daily Scrum (Evento)</i></b>	Scrum Diario. Reunión diaria del equipo de desarrollo, máximo 15 minutos preferiblemente en el mismo lugar, a la misma hora y al frente del Tablero de Scrum
<b><i>DoD</i></b>	Definition of Done. La definición de terminado es un entendimiento común de lo que significa que un PBI está terminado.
<b><i>DoR</i></b>	Definition of Ready. La definición de listo permite saber cuándo un PBI tiene las características necesarias para ser desarrollado por el Equipo y hacer parte del Sprint.
<b><i>Equipo de Desarrollo (Rol)</i></b>	Quienes ejecutan las tareas durante el Sprint
<b><i>Equipo Scrum</i></b>	Todos los integrantes del equipo: Scrum Master, Product Owner y Equipo de Desarrollo.
<b><i>Impedimento</i></b>	Es cualquier cosa que disminuye la velocidad del equipo o no lo deja avanzar hacia el objetivo del Sprint. Son gerenciados por el Scrum Master.
<b><i>Incremento (Artefacto)</i></b>	Es la suma de todos los elementos de la Lista de Producto completados durante un Sprint y el valor de los incrementos de todos los Sprints anteriores.
<b><i>Objetivo del Sprint</i></b>	Es una meta establecida para el Sprint que puede lograrse mediante la implementación de la Lista de Producto.
<b><i>PBI</i></b>	Product Backlog Item. Son los elementos que van en el Product Backlog. Con frecuencia son creados en formato de Historias de Usuario.

---

<b>Product Backlog (Artefacto)</b>		Lista de Producto. Es una lista ordenada de todo lo que se conoce que es necesario en el producto. Es la única fuente de requisitos para cualquier cambio a realizarse en el producto y es gerenciada por el Product Owner.
<b>Product Owner (Rol)</b>		Dueño de Producto. Define la visión del producto refinando el Product Backlog para tener PBI que el equipo pueda ejecutar.
<b>Puntos</b>		Es una medida de la complejidad de los PBI. Se definen en el refinamiento y debe ser estimados por el equipo de desarrollo que son los que ejecutan las tareas.
<b>Refinamiento</b>		Consiste detallar, ordenar, estimar y darles el tamaño adecuado a los PBI del Product Backlog.
<b>Scrum Master (Rol)</b>		Es quien debe verlar porque se haga un buen Scrum y es responsable de la productividad del equipo. Debe ser un lider que colabora y no que se impone.
<b>Spring Review (Evento)</b>		Revisión del Sprint. Reunión para inspeccionar el Incremento y adaptar la Lista de Producto si fuese necesario.
<b>Sprint (Evento)</b>		Es un período de tiempo fijo en el cual se realiza el trabajo y que contiene los otros eventos de Scrum.
<b>Sprint Backlog (Artefacto)</b>		Lista de pendientes del Sprint. Es el conjunto de elementos de la Lista de Producto seleccionados para el Sprint, más un plan para entregar el Incremento de producto y conseguir el Objetivo del Sprint.
<b>Sprint Planning (Evento)</b>		Planeación del Sprint. Se hace al inicio del Sprint para definir qué se hara y cómo se hará.
<b>Sprint Retrospective (Evento)</b>		Retrospectiva del Sprint. Reunión que le permite al Equipo Scrum de inspeccionarse a sí mismo y de crear un plan de mejoras que sean abordadas durante el siguiente Sprint.
<b>Velocidad del equipo</b>	<b>del</b>	Es una medida de la productividad del equipo. Se define normalmente como Puntos por Sprint (puntos/Sprint).

---