21.1

FECHA	Martes, 9 de junio de 2020

Señores
UNIVERSIDAD DE CUNDINAMARCA
BIBLIOTECA
Ciudad

UNIDAD REGIONAL	Sede Fusagasugá.
TIPO DE DOCUMENTO	Pasantía.
FACULTAD	Ingeniería.
NIVEL ACADÉMICO DE FORMACIÓN O PROCESO	Pregrado.
PROGRAMA ACADÉMICO	Ingeniería de sistemas

El Autor(Es):

APELLIDOS COMPLETOS	NOMBRES COMPLETOS	No. DOCUMENTO DE IDENTIFICACIÓN
Chavez Chavarro	Sergio Julian	1069760295
Quintero Torres	Ivan Dario	1003519769

Director(Es) y/o Asesor(Es) del documento:

APELLIDOS COMPLETOS	NOMBRES COMPLETOS
Reyes Alvarez	Jorge Julio
Hernandez Molano	Eduard Norberto

TÍTULO DEL DOCUMENTO
Análisis, diseño, desarrollo e implementación de una aplicación con los módulos de facturación, flujos de caja y gestión de clientes; tipo “pos” con base a los requerimientos establecidos por la empresa Ice Company sas.

SUBTÍTULO (Aplica solo para Tesis, Artículos Científicos, Disertaciones, Objetos Virtuales de Aprendizaje)

TRABAJO PARA OPTAR AL TÍTULO DE: Aplica para Tesis/Trabajo de Grado/Pasantía
Ingeniero de sistemas.

AÑO DE EDICION DEL DOCUMENTO	NÚMERO DE PÀGINAS
05/06/2020	88

DESCRIPTORES O PALABRAS CLAVES EN ESPAÑOL E INGLÉS (Usar 6 descriptores o palabras claves)	
ESPAÑOL	**INGLÉS**
1. Patrón de arquitectura	Architecture pattern
2. Modelo, Vista, Vista Modelo (MVVM)	Model, View, Model View (MVVM)
3. Scrum.	Scrum
4.Desarrollo iterativo.	Iterative development
5.Base de Datos.	Database
6.Servicios web.	Web service.

RESUMEN DEL CONTENIDO EN ESPAÑOL E INGLÉS

(Máximo 250 palabras – 1530 caracteres, aplica para resumen en español):

Resumen

Este documento evidencia el desarrollo de la aplicación “FLYKE”, la cual se centra en los procesos operativos de cualquier tipo de restaurante o punto de venta (POS- point of sale), como objetivo busca optimizar las tareas realizadas en los establecimientos. En el presente trabajo se describe la lista de requerimientos del proyecto, el planteamiento del problema, especificaciones de Scrum como metodología de trabajo, además se especifican los diagramas UML para brindar una mayor comprensión del flujo de la aplicación, debido a que por términos de confidencialidad no es permitida la divulgación de algunos aspectos que comprometan los derechos de autor de la empresa con el software desarrollado, también se especifican las tareas de programación junto con las herramientas implementadas en el apoyo del desarrollo y la interacción entre el Scrum Team para trabajar sinérgicamente, con el objetivo de cumplir los requerimientos establecidos por el cliente.
Es importante resaltar el trabajo en conjunto con el proyecto de pasantía titulado “Análisis, diseño, desarrollo e implementación de una aplicación con los módulos de control de inventarios y seguridad; tipo “pos” con base a los requerimientos establecidos por la empresa ICE COMPANY ”que denominaremos como proyecto 2 ,el cual es realizado por el estudiante Juan David Gutierrez Agudelo.

Abstract

This document evidences the development of the “FLYKE” application, which focuses on the operational processes of any type of restaurant or point of sale (POS point of sale), with the aim of optimizing the tasks performed in the establishments. This work describes the list of project requirements, the problem statement, Scrum specifications as work methodology, and the UML diagrams are also specified to provide a better understanding of the application flow, due to confidentiality terms. The disclosure of some aspects that compromise the copyright of the company with the developed software is not allowed, the programming tasks are also specified along with the tools implemented in support of development and the interaction between the Scrum Team to work synergistically, In order to meet the requirements established by the client.
It is important to emphasize the work in conjunction with the internship project entitled "Analysis, design, development and implementation of an application with the modules of inventory control and security; type "pos" based on the requirements established by the company ICE COMPANY "that we will call as project 2 ,which is made by the student Juan David Gutierrez Agudelo.

AUTORIZACIÓN DE PUBLICACIÒN

Por medio del presente escrito autorizo (Autorizamos) a la Universidad de Cundinamarca para que, en desarrollo de la presente licencia de uso parcial, pueda ejercer sobre mí (nuestra) obra las atribuciones que se indican a continuación, teniendo en cuenta que, en cualquier caso, la finalidad perseguida será facilitar, difundir y promover el aprendizaje, la enseñanza y la investigación.

En consecuencia, las atribuciones de usos temporales y parciales que por virtud de la presente licencia se autoriza a la Universidad de Cundinamarca, a los usuarios de la Biblioteca de la Universidad; así como a los usuarios de las redes, bases de datos y demás sitios web con los que la Universidad tenga perfeccionado una alianza, son:
Marque con una "X":

AUTORIZO (AUTORIZAMOS)	SI	NO
1. La reproducción por cualquier formato conocido o por conocer.	X	
2. La comunicación pública por cualquier procedimiento o medio físico o electrónico, así como su puesta a disposición en Internet.	X	
3. La inclusión en bases de datos y en sitios web sean éstos onerosos o gratuitos, existiendo con ellos previa alianza perfeccionada con la Universidad de Cundinamarca para efectos de satisfacer los fines previstos. En este evento, tales sitios y sus usuarios tendrán las mismas facultades que las aquí concedidas con las mismas limitaciones y condiciones.	X	
4. La inclusión en el Repositorio Institucional.	X	

De acuerdo con la naturaleza del uso concedido, la presente licencia parcial se otorga a título gratuito por el máximo tiempo legal colombiano, con el propósito de que en dicho lapso mi (nuestra) obra sea explotada en las condiciones aquí estipuladas y para los fines indicados, respetando siempre la titularidad de los derechos patrimoniales y morales correspondientes, de acuerdo con los usos honrados, de manera proporcional y justificada a la finalidad perseguida, sin ánimo de lucro ni de comercialización.

Para el caso de las Tesis, Trabajo de Grado o Pasantía, de manera complementaria, garantizo(garantizamos) en mi(nuestra) calidad de estudiante(s) y por ende autor(es) exclusivo(s), que la Tesis, Trabajo de Grado o Pasantía en cuestión, es producto de mi(nuestra) plena autoría, de mi(nuestro) esfuerzo personal intelectual, como consecuencia de mi(nuestra) creación original particular y, por tanto, soy(somos) el(los) único(s) titular(es) de la misma. Además, aseguro (aseguramos) que no contiene citas, ni transcripciones de otras obras protegidas, por fuera de los límites autorizados por la ley, según los usos honrados, y en proporción a los fines previstos; ni tampoco contempla declaraciones difamatorias contra terceros; respetando el derecho a la imagen, intimidad, buen nombre y demás derechos constitucionales. Adicionalmente, manifiesto (manifestamos) que no se incluyeron expresiones contrarias al orden público ni a las buenas costumbres. En consecuencia, la responsabilidad directa en la

elaboración, presentación, investigación y, en general, contenidos de la Tesis o Trabajo de Grado es de mí (nuestra) competencia exclusiva, eximiendo de toda responsabilidad a la Universidad de Cundinamarca por tales aspectos.

Sin perjuicio de los usos y atribuciones otorgadas en virtud de este documento, continuaré (continuaremos) conservando los correspondientes derechos patrimoniales sin modificación o restricción alguna, puesto que, de acuerdo con la legislación colombiana aplicable, el presente es un acuerdo jurídico que en ningún caso conlleva la enajenación de los derechos patrimoniales derivados del régimen del Derecho de Autor.

De conformidad con lo establecido en el artículo 30 de la Ley 23 de 1982 y el artículo 11 de la Decisión Andina 351 de 1993, *"Los derechos morales sobre el trabajo son propiedad de los autores"*, los cuales son irrenunciables, imprescriptibles, inembargables e inalienables. En consecuencia, la Universidad de Cundinamarca está en la obligación de RESPETARLOS Y HACERLOS RESPETAR, para lo cual tomará las medidas correspondientes para garantizar su observancia.

NOTA: (Para Tesis, Trabajo de Grado o Pasantía):

Información Confidencial:

Esta Tesis, Trabajo de Grado o Pasantía, contiene información privilegiada, estratégica, secreta, confidencial y demás similar, o hace parte de la investigación que se adelanta y cuyos resultados finales no se han publicado. **SI _X_ NO __**.
En caso afirmativo expresamente indicaré (indicaremos), en carta adjunta tal situación con el fin de que se mantenga la restricción de acceso.

LICENCIA DE PUBLICACIÒN

Como titular(es) del derecho de autor, confiero(erimos) a la Universidad de Cundinamarca una licencia no exclusiva, limitada y gratuita sobre la obra que se integrará en el Repositorio Institucional, que se ajusta a las siguientes características:

a) Estará vigente a partir de la fecha de inclusión en el repositorio, por un plazo de 5 años, que serán prorrogables indefinidamente por el tiempo que dure el derecho patrimonial del autor. El autor podrá dar por terminada la licencia solicitándolo a la Universidad por escrito. (Para el caso de los Recursos Educativos Digitales, la Licencia de Publicación será permanente).

b) Autoriza a la Universidad de Cundinamarca a publicar la obra en formato y/o soporte digital, conociendo que, dado que se publica en Internet, por este hecho circula con un alcance mundial.

c) Los titulares aceptan que la autorización se hace a título gratuito, por lo tanto, renuncian a recibir beneficio alguno por la publicación, distribución, comunicación pública y cualquier otro uso que se haga en los términos de la presente licencia y de la licencia de uso con que se publica.

d) El(Los) Autor(es), garantizo(amos) que el documento en cuestión, es producto de mi(nuestra) plena autoría, de mi(nuestro) esfuerzo personal intelectual, como consecuencia de mi (nuestra) creación original particular y, por tanto, soy(somos) el(los) único(s) titular(es) de la misma. Además, aseguro(aseguramos) que no contiene citas, ni transcripciones de otras obras protegidas, por fuera de los límites autorizados por la ley, según los usos honrados, y en proporción a los fines previstos; ni tampoco contempla declaraciones difamatorias contra terceros; respetando el derecho a la imagen, intimidad, buen nombre y demás derechos constitucionales. Adicionalmente, manifiesto (manifestamos) que no se incluyeron expresiones contrarias al orden público ni a las buenas costumbres. En consecuencia, la responsabilidad directa en la elaboración, presentación, investigación y, en general, contenidos es de mí (nuestro) competencia exclusiva, eximiendo de toda responsabilidad a la Universidad de Cundinamarca por tales aspectos.

e) En todo caso la Universidad de Cundinamarca se compromete a indicar siempre la autoría incluyendo el nombre del autor y la fecha de publicación.

f) Los titulares autorizan a la Universidad para incluir la obra en los índices y buscadores que estimen necesarios para promover su difusión.

g) Los titulares aceptan que la Universidad de Cundinamarca pueda convertir el documento a cualquier medio o formato para propósitos de preservación digital.

h) Los titulares autorizan que la obra sea puesta a disposición del público en los términos autorizados en los literales anteriores bajo los límites definidos por la universidad en el “Manual del Repositorio Institucional AAAM003”

i) Para el caso de los Recursos Educativos Digitales producidos por la Oficina de Educación Virtual, sus contenidos de publicación se rigen bajo la Licencia Creative Commons: Atribución- No comercial- Compartir Igual.

j) Para el caso de los Artículos Científicos y Revistas, sus contenidos se rigen bajo la Licencia Creative Commons Atribución- No comercial- Sin derivar.

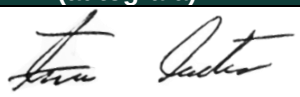
Nota:
Si el documento se basa en un trabajo que ha sido patrocinado o apoyado por una entidad, con excepción de Universidad de Cundinamarca, los autores garantizan

que se ha cumplido con los derechos y obligaciones requeridos por el respectivo contrato o acuerdo.

La obra que se integrará en el Repositorio Institucional, está en el(los) siguiente(s) archivo(s).

Nombre completo del Archivo Incluida su Extensión (Ej. PerezJuan2017.pdf)	Tipo de documento (ej. Texto, imagen, vídeo, etc.)
1. Análisis, diseño, desarrollo e implementación de una aplicación con los módulos de facturación, flujos de caja y gestión de clientes; tipo “pos” con base a los requerimientos establecidos por la empresa Ice Company sas.pdf	Texto e Imágenes.

En constancia de lo anterior, Firmo (amos) el presente documento:

APELLIDOS Y NOMBRES COMPLETOS	FIRMA (autógrafa)
Quintero Torres Ivan Dario	
Chavez Chavarro Sergio Julian	

21.1-51-20

ANÁLISIS, DISEÑO, DESARROLLO E IMPLEMENTACIÓN DE UNA APLICACIÓN CON LOS MÓDULOS DE FACTURACIÓN, FLUJOS DE CAJA Y GESTIÓN DE CLIENTES; TIPO “POS” CON BASE A LOS REQUERIMIENTOS ESTABLECIDOS POR LA EMPRESA ICE COMPANY SAS.

AUTORES:

SERGIO JULIAN CHAVEZ CHAVARRO

IVAN DARIO QUINTERO TORRES

DIRECTOR: JORGE JULIO REYES ALVAREZ

UNIVERSIDAD DE CUNDINAMARCA

FACULTAD DE INGENIERÍA

INGENIERÍA DE SISTEMAS

FUSAGASUGÁ

2020

Resumen

Este documento evidencia el desarrollo de la aplicación “FLYKE”, la cual se centra en los procesos operativos de cualquier tipo de restaurante o punto de venta (POS- point of sale), como objetivo busca optimizar las tareas realizadas en los establecimientos. En el presente trabajo se describe la lista de requerimientos del proyecto, el planteamiento del problema, especificaciones de Scrum como metodología de trabajo, además se especifican los diagramas UML para brindar una mayor comprensión del flujo de la aplicación, debido a que por términos de confidencialidad no es permitida la divulgación de algunos aspectos que comprometan los derechos de autor de la empresa con el software desarrollado, también se especifican las tareas de programación junto con las herramientas implementadas en el apoyo del desarrollo y la interacción entre el Scrum Team para trabajar sinérgicamente, con el objetivo de cumplir los requerimientos establecidos por el cliente.

Es importante resaltar el trabajo en conjunto con el proyecto de pasantía titulado “Análisis, diseño, desarrollo e implementación de una aplicación con los módulos de control de inventarios y seguridad; tipo “pos” con base a los requerimientos establecidos por la empresa ICE COMPANY ”que denominaremos como proyecto 2 ,el cual es realizado por el estudiante Juan David Gutierrez Agudelo.

Abstract

This document evidences the development of the “FLYKE” application, which focuses on the operational processes of any type of restaurant or point of sale (POS point of sale), with the aim of optimizing the tasks performed in the establishments. This work describes the list of project requirements, the problem statement, Scrum specifications as work methodology, and the UML diagrams are also specified to provide a better understanding of the application flow, due to confidentiality terms. The disclosure of some aspects that compromise the copyright of the company with the developed software is not allowed, the programming tasks are also specified along with the tools implemented in support of development and the interaction between the Scrum Team to work synergistically, In order to meet the requirements established by the client. It is important to highlight the work in conjunction with the internship project entitled “Analyze, design, develop and implement an application with product management, security components and application adjustments; Type 'pos' based on the requirements established by the ICE COMPANY 'that we will call project 2, which is carried out by the student Juan David Gutierrez Agudelo. It is important to emphasize the work in conjunction with the internship project entitled "Analysis, design, development and implementation of an application with the modules of inventory control and security; type "pos" based on the requirements established by the company ICE COMPANY "that we will call as project 2 ,which is made by the student Juan David Gutierrez Agudelo.

Contenido

1. Planteamiento del problema

Los procesos de los restaurantes se dividen por 3 áreas fundamentales, área de mesas, cocina y caja. Dentro del área de mesas se desarrolla la interacción entre el cliente y el mesero para realizar y tomar nota del pedido (lista de productos con comentarios de preferencias), en ocasiones durante este proceso surgen inconvenientes ya que el mesero no escribe en la orden todos los requerimientos expresados por el cliente.

Dentro del área de cocina se ven involucrados los meseros y el personal de cocina, participando en el flujo de la notificación de pedidos y la preparación de cada producto en la orden respectiva, en ocasiones la mala comunicación entre estos dos actores impide un funcionamiento correcto del servicio; el mesero puede estar a cargo del despacho donde existe la posibilidad de brindar atención a varias mesas al mismo tiempo, por esta razón el inicio de la preparación de los alimentos no es eficiente ya que la notificación de cada pedido no es instantánea.

El área de caja (facturación) carece de control del personal, los clientes y mesas dentro del restaurante ya que no se tiene un conocimiento claro respecto a los pedidos, órdenes activas y órdenes por cobrar realizadas por los clientes.

En el municipio de Fusagasugá y en general la mayoría de restaurantes procuran ofrecer el mejor servicio posible para aumentar su demanda y generar más rentabilidad, uno de los mayores retos es la comunicación y el funcionamiento eficiente de cada proceso perteneciente a las áreas de trabajo. En consecuencia surge la siguiente pregunta de investigación, ¿Cómo desarrollar un software que permita optimizar los procesos internos de los restaurantes potenciando el servicio al cliente?

2. Objetivo general

Analizar, diseñar, desarrollar e implementar una aplicación con los módulos de facturación, toma-pedidos y gestión de empleados; tipo “pos” con base a los requerimientos establecidos por la empresa Ice company sas.

2.1 Objetivos específicos

- Analizar, diseñar y crear la base de datos para la aplicación ‘Flyke’
- Desarrollar el módulo de facturación que permita gestionar diferentes tipos de restaurantes con opción a distintos medios de pago e impresiones.
- Desarrollar el módulo de toma pedidos para gestionar las órdenes en los diferentes tipos de restaurantes
- Desarrollar el módulo de gestión de empleados para administrarlos desde el aplicativo móvil.
- Implementar pruebas unitarias a los módulos desarrollados

3. Requerimientos funcionales

1. El sistema permite el registro de nuevos usuarios que a futuro serán clasificados como empleados o propietarios de los restaurantes.
2. El sistema adaptara una función especial al módulo “Toma pedidos” dependiendo el tipo de restaurante registrado (restaurante con toma pedidos en mesas, restaurante con toma pedidos como rompe filas, restaurante/Food Truck con toma pedidos con número cíclico de orden).
3. Para los restaurantes con toma pedido en mesas, el sistema mostrará las mesas disponibles y permitirá hacer el pedido por medio de tablets asignadas a cada mesa.
4. En el caso de los restaurantes rompe filas, el sistema dará acceso directamente al módulo de productos para efectuar el pedido y asignará automáticamente una mesa al usuario.
5. En el tipo de restaurante Food Truck el sistema permitirá realizar el pedido directamente y luego asignará un número en orden cíclico para determinar el turno del cliente.
6. El sistema admitirá comentarios en los productos y en la orden en general para una mejor atención al cliente.
7. El sistema mostrará la cantidad y precio de productos seleccionados, los ingredientes y las adiciones elegidas y el valor total de la factura en el momento de efectuar la toma de un pedido.
8. El sistema permitirá visualizar una lista de los empleados del restaurante con diferentes opciones como agregar, modificar y eliminar
9. El sistema permitirá elegir entre tres tipos de impresiones: productos a cocina, impresión de orden, impresión de factura, el proceso se efectúa por medio de una impresora térmica.
10. El sistema permitirá seleccionar el medio de pago y posible división de la cuenta si el usuario lo desea

11. El sistema permitirá realizar pago de la orden con ‘créditos Dely’ el cual será generado por medio de un código QR y posteriormente escaneado desde la aplicación ‘Dely domicilios’.

4. Estado del arte

Hoy en día el uso de software es un pilar fundamental para la gestión de todos los procesos que se llevan a cabo en una empresa, lo que hace que las tecnologías de la información y comunicación (TIC) estén presentes en todo momento a nivel global; debido a esto se genera competitividad tecnológica en el campo empresarial, lo que implica visualizar un panorama de las aplicaciones dentro del mismo segmento de mercado, a continuación, se mencionan las diferentes aplicaciones encontradas:

Globalsoft: En la página web de Globalsoft encontramos la siguiente descripción acerca de su software:

Es un sistema de información fácil, práctico y seguro, desarrollado bajo un ambiente Windows que permite a sus usuarios obtener sus informes de manera oportuna y facilitar el desarrollo diario de la empresa. Además, nuestro servicio de desarrollo sobre medida permite la creación de sistemas que cubran todas las necesidades específicas de cualquier empresa de cualquier sector económico. Nuestro software lo hemos enlazado con las últimas tendencias en cuanto a telecomunicaciones y redes informáticas, brindándole al usuario, control absoluto de su negocio desde cualquier parte del mundo y con todas las directrices de seguridad y control de accesos. Historia GLOBALSOFT T&S S.A.S. recuperado de: https://tysglobalsoft.com/empresa/

Odoo: En la página web de Odoo encontramos la siguiente información acerca de su software Odoo POS:

Basado en una interfaz inteligente que cualquier empresa minorista puede utilizar sin ninguna dificultad. ya que es extremadamente flexible, puede configurarlo para que se ajuste a sus necesidades específicas. La aplicación cuenta con manejo de módulos tales como facturación, control de inventarios y además está integrada con otras aplicaciones lo cual hace que Odoo

tenga una mayor cantidad de funcionalidades. Odoo Pos (2020). Recuperado de: https://www.odoo.com/es_ES/page/point-of-sale-shop

Tablet SP + Lite: En la página web de Tablet SP encontramos la siguiente información acerca de su software:

El más innovador sistema de punto de venta diseñado para operar en dispositivos con plataforma operativa Android. Listo para funcionar en tablets y equipos "all in one" permitiéndole a su negocio realizar el registro de las operaciones de compra -venta en mostrador de forma ágil y sencilla, gracias a su interface gráfica intuitiva. Dentro de las funcionalidades que ofrece se encuentra el registro de horario de entrada del usuario del punto de venta, permite múltiples usuarios del punto de venta, registro de ventas mediante una interfaz muy sencilla de utilizar con imágenes de sus productos, manejo de peso de productos, permite la administración completa de las actividades comerciales del día a día, cuenta con la funcionalidad de facturación electrónica , venta de tiempo aire , pago de servicios y aceptación de pago mediante tarjeta bancaria.
Tablet SP Lite está pensado tanto para organizaciones con tiendas propias como para franquiciatarios, permitiendo llevar la gestión de cada unidad de franquicia de forma individual o compartiendo datos a fin de disponer de una visión global consolidada en tiempo real de la operación de cada una de las tiendas. El uso de un dispositivo Tablet como punto de venta lo hace especialmente útil en entornos donde la relación con el cliente requiere de mayor interacción. Tablet Salepoint Software. (2016). App Tablet SP + Lite. 2020, de Tablet Salespoint Software Sitio web: https://www.tabletsp.com/producto/tablet-sp-lite/

5. Justificación

En la actualidad existen diversos tipos de restaurantes agrupados según su modelo de negocio, dentro de este grupo encontramos 3 tipos de restaurantes más importantes; los restaurantes convencionales cuyo proceso operativo se efectúa desde la atención al cliente directamente desde las mesas, por otro lado están los restaurantes food truck cuyo proceso es similar a las plazoletas de comida en donde primero se paga el pedido y después el cliente espera su turno, por último están los restaurantes rompe filas que siguen un proceso de tipo autoservicio en donde el cliente gestiona su propio pedido, la empresa ICE company SAS ya contaba con un nicho de mercado relacionado con los pedidos en un restaurante.

Muchos de los restaurantes cuentan con varios espacios de trabajo que integran toda su estructura, como lo son las mesas, la parte de cocina y zona de facturación; en donde se opera a través de procesos manuales (lápiz y papel) para notificar una novedad o un cambio entre sus partes. Los procesos manuales demandan de mucho más tiempo y en ocasiones surgen errores o inconvenientes en la parte operativa (personal encargado de atención al cliente) además la implementación de un software contribuye al medio ambiente evitando el uso del papel para las órdenes, la carta de productos y los registros contables, debido a esto es pertinente el uso una herramienta tecnológica para optimizar la parte operativa, mejorando la productividad y el servicio a los clientes.

6. Impacto del proyecto

El impacto generado a partir de la realización de este proyecto se puede ver reflejado en muchos ámbitos tales como, la productividad de los restaurantes, la satisfacción del cliente, el desarrollo tecnológico en el municipio de fusagasugá, la eficiencia de los funcionarios del negocio, entre otros:

Productividad en restaurantes: El impacto que se genera en los restaurantes es tal vez el más destacado, pues los procesos en su mayoría serán los que más se optimicen, por ejemplo, el procedimiento para tomar un pedido es personalizado para el cliente quien podrá visualizar una imagen, nombre, descripción y precio de cada producto, seleccionar el producto y agregar un comentario si así lo desea, desde el mismo aplicativo también podrá hacer un llamado al mesero del restaurante para consultarle cualquier duda o comentario, por otro lado los cajeros tienen un apartado para realizar tomar el pedido y al mismo tiempo gestionar la facturación e impresión de la orden, entre otros procesos que hacen que en general el establecimiento tenga un mayor rendimiento y fluidez en cada uno de sus servicios.

Clientes del restaurante: realizar sus pedidos de forma digital a través del software (app), en donde puede visualizar cada producto junto con respectiva descripción, además existe la posibilidad de realizar comentarios sobre cada producto por ejemplo si no se quiere algún tipo de salsa, también puede seleccionar en los productos que aplica sus respectivos ingredientes y adiciones, adicional a esto el cliente tiene la posibilidad de hacer un comentario sobre toda la orden como por ejemplo todo bajo en azúcar. Esto brinda una mejor experiencia de usuario.

Meseros: existen dos formas para realizar la orden, el cliente realiza su propia orden a través del software o el mesero toma la orden del cliente en el software, esto impide que el mesero utilice papel y lápiz , se agiliza el tiempo en el que se realiza un pedido, además se tiene un mayor

control sobre todas las órdenes (visualizar en tiempo real), ya no hay que ir hasta la cocina para entregar los pedidos sino que el pedido se imprime directamente en la cocina lo que permite realizar una labor más organizada que mitiga al máximo los errores que pueden surgir.

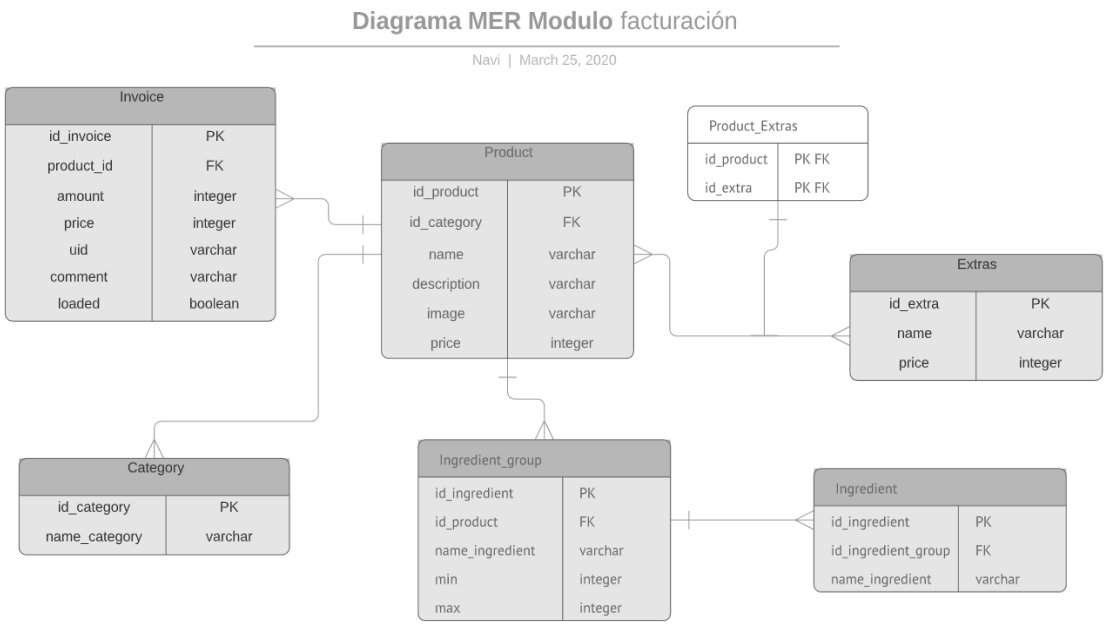
Cajero: De una forma óptima el cajero busca en el software la orden correspondiente a la mesa en la que estuvo el cliente o la orden asociada al cliente y finaliza la orden para que el cliente pague, se generan todos los datos importantes en la factura (impuestos, productos, valores adicionales) y la orden queda almacenada en el sistema junto con el dinero que entró. Esto optimiza el proceso de facturación en un menor tiempo posible tanto para el cliente como para el cajero.

Administrador: el administrador lleva un control sobre toda la parte financiera ya que cada orden que se despacha se almacena en el sistema, incluyendo el pago si se hice en efectivo o en medios digitales, esto permite generar los informes de acuerdo al intervalo de tiempo requerido (de tal fecha a tal fecha o de tal semana), de acuerdo a esos informes el administrador sabrá cuántas ganancias tuvo y según eso realizar promociones o invertir para generar más ganancias.

Debido a los anteriores efectos se observa que el software tiene un impacto alto positivo sobre cada uno de los usuarios relacionados con los procesos internos y externos del restaurante ya que se logra optimizar cada tarea y en un mayor porcentaje mejorará la satisfacción del cliente.

7. Especificación del diseño: Modelo entidad relación

En este diagrama MER se describen las entidades que intervienen en el modulo de facturacion, con sus respectivas relaciones, propiedades, tipos y dependencias. tal como se puede observar, la entidad de producto hace parte fundamental en la construcción del modulo de facturacion, ya que todas las otras entidades tienen algún tipo de relación o abstracción del producto.



Modelo entidad relacion: facturación, diagrama creado con la herramienta lucidchart

8. Marco teórico

Aunque se utiliza la metodología scrum como marco de trabajo en todo el desarrollo de la aplicación, se decide implementar aspectos de la metodología tradicional como los diagramas UML para brindar una mayor comprensión al lector con respecto al flujo de datos y funciones dentro del software, esto se tuvo en cuenta ya que por temas de confidencialidad no fue permitido evidenciar el aplicativo en su totalidad.

8.1 Modelo 4+1

Este modelo diseñado por el profesor Philippe Kruchten es utilizado para describir la arquitectura de un software basándose en la utilizacion de multiples puntos de vista.
El profesor Kruchten propone que: "un sistema se ha de documentar y describir con 4 vistas bien diferenciadas pero al mismo tiempo relacionadas para construir una quinta vista" *(Kruchten,1995, Noviembre)* vista a la que se le denomina "vista +1", dichas vistas Kruchten las definió de la siguiente manera:

• **La vista lógica** describe el modelo de objetos del diseño cuando se usa un método de diseño orientado a objetos. Para diseñar una aplicación muy orientada a los datos, se puede usar un enfoque alternativo para desarrollar algún otro tipo de vista lógica, tal como diagramas de entidad-relación.

• **La vista de procesos** describe los aspectos de concurrencia y sincronización del diseño.

• **La vista física** describe el mapeo del software en el hardware y refleja los aspectos de distribución.

La vista de desarrollo describe la organización estática del software en su ambiente de desarrollo.

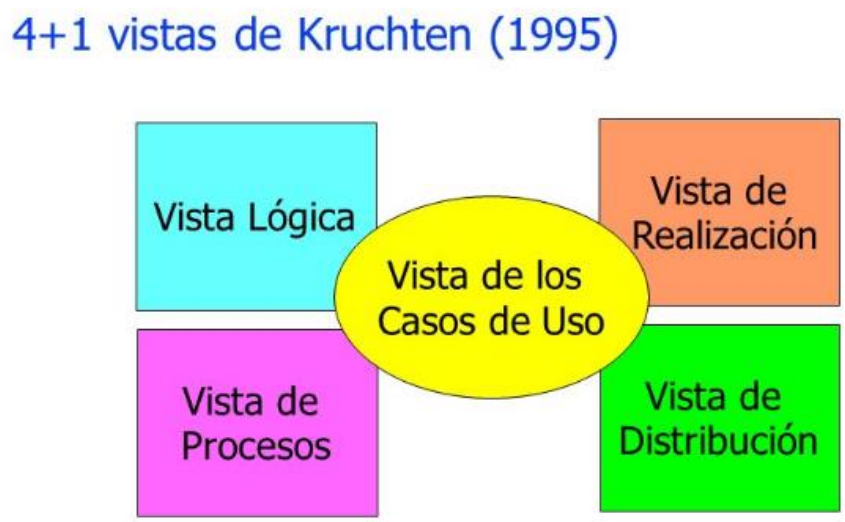


Imagen del modelo 4+1, extraída de: https://slideplayer.es/slide/1107404/

8.2 Lenguaje de programación kotlin

Kotlin es un nuevo lenguaje de programación desarrollado por JetBrains y está dirigido a la plataforma Java. Es un lenguaje conciso, seguro, pragmático y está centrado en la interoperabilidad con Java, por lo cual es usable en casi todos los ámbitos donde es usado Java, como el desarrollo en el lado del servidor, aplicaciones de Android, entre otros. *(Jemerov & Isakova, 2018)*. Es Estáticamente tipado, es decir, no hay que especificar los tipos de variables ya que la máquina virtual infiere en el tipo. Es un lenguaje que corre bajo la Máquina Virtual de Java, por lo que tiene el mismo rendimiento que Java. Es interoperable con JAVA, es decir que los módulos desarrollados con Java se pueden acoplar sin problema con kotlin. Las librerías que existen para trabajar en Java están migradas a kotlin por lo tanto se pueden utilizar.

8.3 Arquitectura MVVM

Según la plataforma de educación OpenWebinar hace referencia a MVVM como: "un arquitectura de desarrollo *(Microsoft)* aunque en los últimos años se ha adoptado por varios lenguajes de programación, su fundamento es desacoplar la parte de la vista (View), de la parte de la lógica de negocio (ViewModel) y la parte lógica de los datos (Modelos)" con el fin de asignar responsabilidad a cada componente de esta estructura, respetando el principio de responsabilidad única; esta arquitectura ofrece un fácil mantenimiento al desarrollo ya que los cambios que se hagan en la lógica o en los datos no afectarán el funcionamiento de la vista, gracias a esto permite que sea escalable y muy práctica para los equipos de desarrollo.

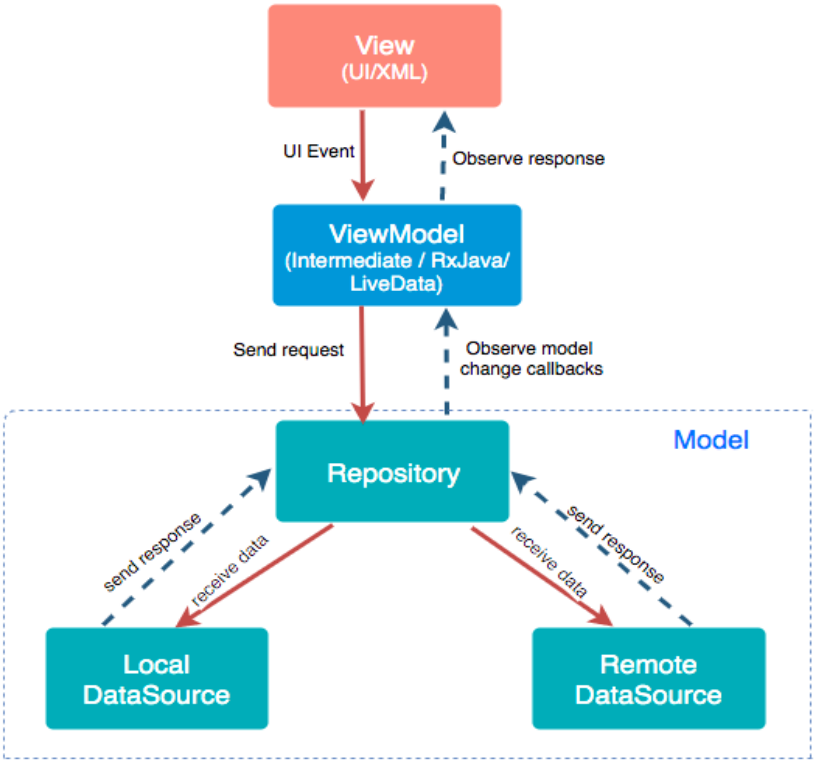


Imagen del arquitectura MVVM, extraída de: https://www.sovereignconsult.com/blog/know-mvvm-clean-architecture-android-app/

8.4 Jira software

Es una herramienta online para la administración de tareas de un proyecto, para que el equipo de trabajo pueda planificar, supervisar y public el software.

planifica: En la metodología scrum permite crear historias de usuario e incidencias, planifica sprints y distribuye tareas

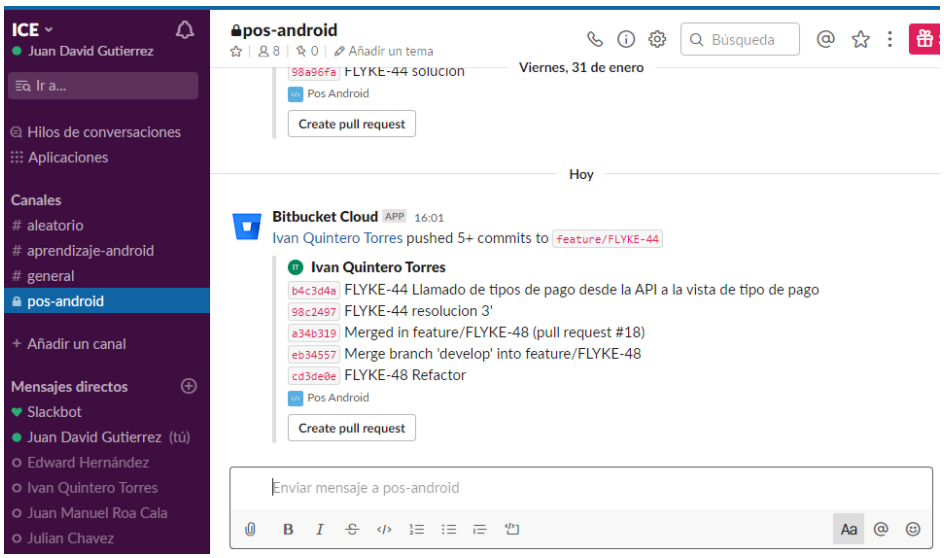
Supervisa: Prioriza y analiza el trabajo de tu equipo en su contexto

Crea informes: Mejora el rendimiento del equipo con datos visuales en tiempo rea
Jira software es compatible con bitbucket, slack.

8.5 Slack

Slack es una herramienta que sirve como centro de colaboración, puede reemplazar el correo electrónico y permite el trabajo colaborativo de una empresa. Tiene un diseño que respalda un trabajo natural en conjunto para trabajar en línea de forma eficiente.

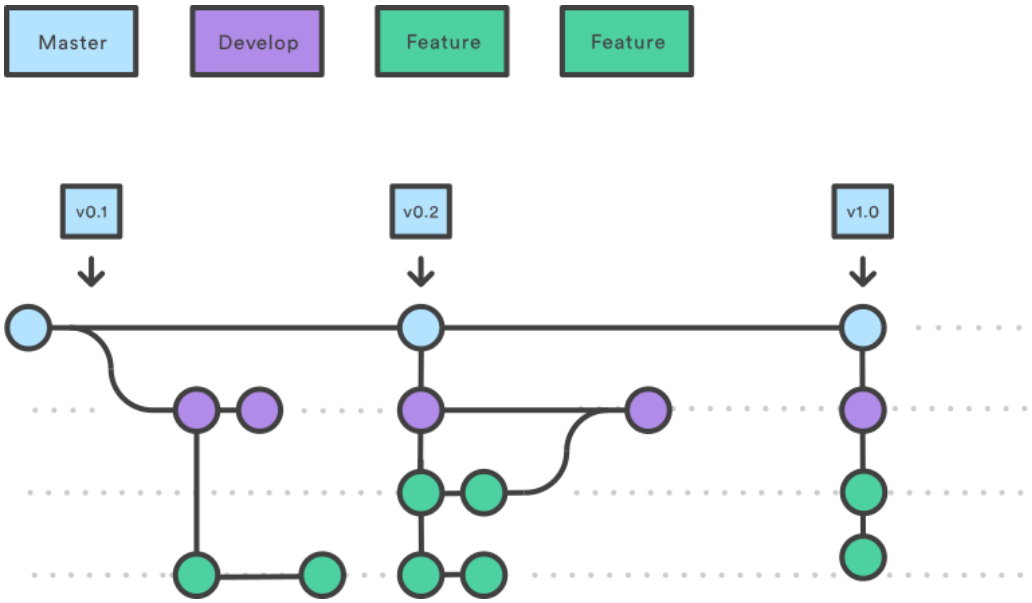
El espacio de trabajo está conformado por canales donde los miembros del equipo trabajan y se comunican; también están los chats (mensajes directos) donde cada integrante se puede comunicar con otro de forma personal



8.6 Git

Es un software de control de versiones para aplicaciones creado por Linus Torvalds, es decir gestiona los diferentes cambios que se realizan a lo largo del desarrollo de una aplicación (software), git funciona como un repositorio de manera local se puede utilizar a través de la línea de comandos

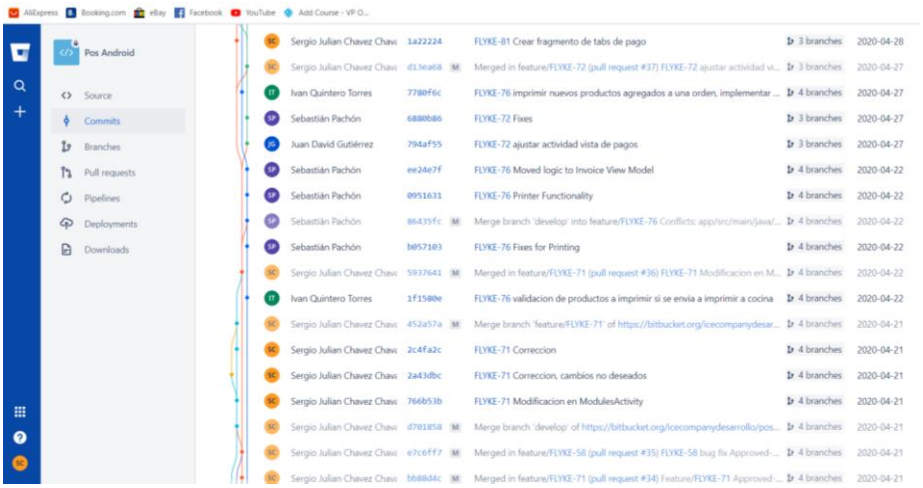
Rama (Branch): una rama es un espacio en donde se divide el código de la aplicación, una rama puede servir para almacenar el código de una determinada funcionalidad



En esta imagen podemos observar 3 tipos de ramas, la rama principal Master, una rama secundaria Develop y las demás ramas denominadas Feature cada rama específica parte del código de la aplicación, en la rama master se puede observar el versionamiento desde una versión 0.1 hasta una versión 1.0.

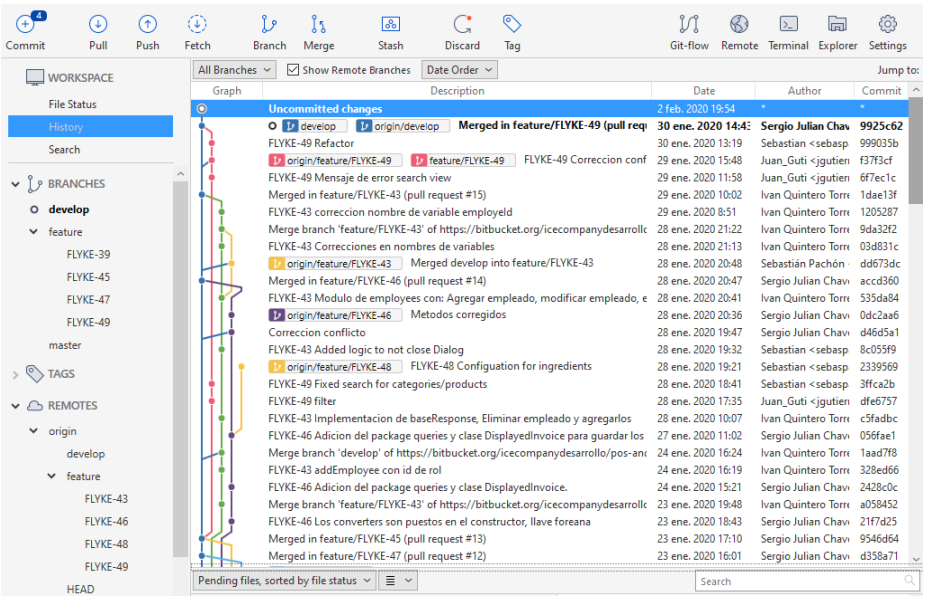
8.7 Bitbucket

Es una herramienta para el control de versiones utilizadas en git, es un repositorio remoto y sirve para planificar proyectos y colaborar en el código.



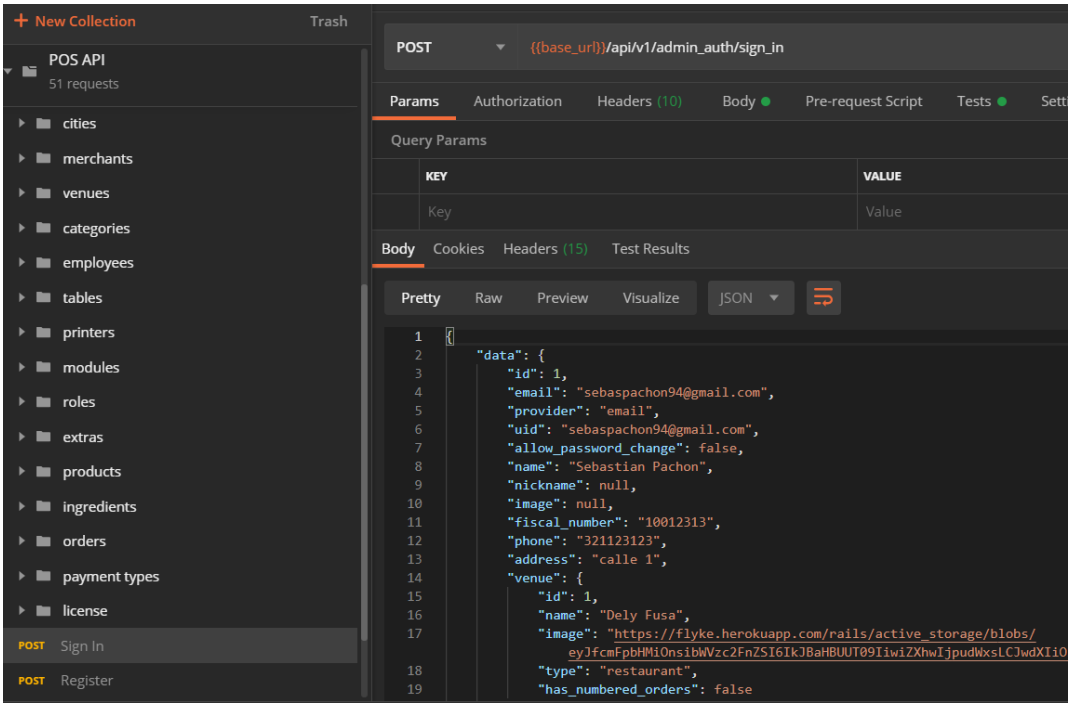
8.8 SourceTree

Es el sistema de control de versiones local, es más interactivo que utilizar la consola de git, pero funciona de la misma manera, en esta herramienta se pueden realizar operaciones como pull, git add, commits, push, merge, creación de ramas y entre otras funciones fundamentales para el trabajo en cooperativo, la siguiente es una imagen del entorno de la herramienta y con un historial en un punto aleatorio del proyecto:



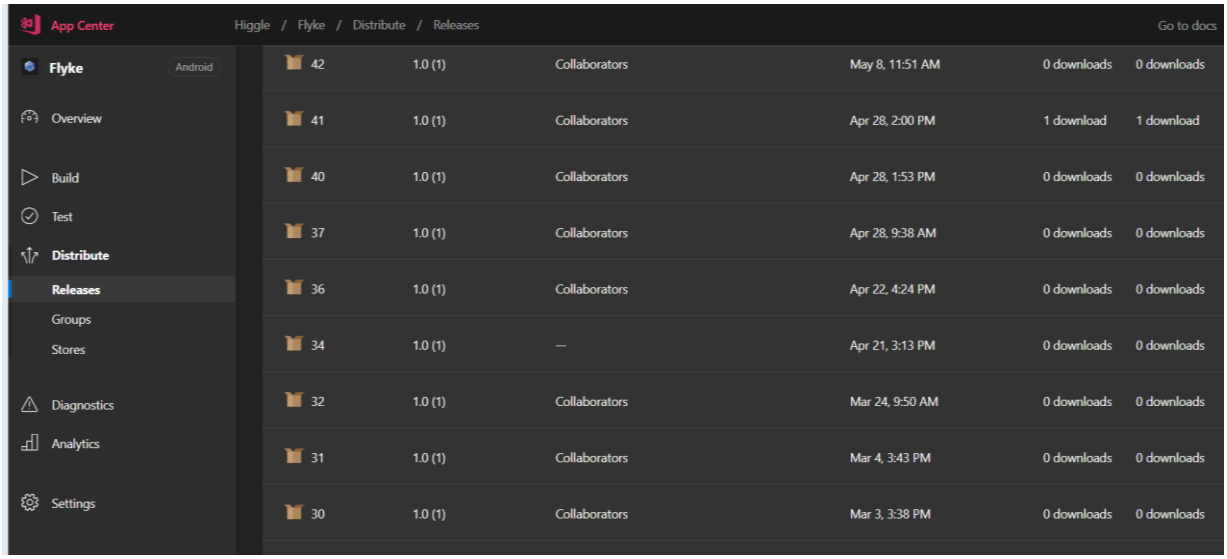
8.9 Postman

Postman es una herramienta creada para los desarrolladores con el fin de realizar peticiones HTTP a cualquier tipo de API. Es muy eficiente para el proyecto porque permite que la programación sea ágil al promover la verificación de las respuestas del servidor, este programa permite organizar las peticiones en colecciones, automatizar pruebas y lo mejor aún, sincronizar todos los equipos que estén unidos en el equipo de trabajo.



8.10 App Center

App center es un centro de control que nos permite subir las versiones de las tareas desarrolladas en forma de archivo instalable con la extensión: .apk, con el fin de que el QA team pueda descargar e instalar la versión para probar las nuevas funcionalidades y cambios realizados durante esta tarea y así poder determinar si cumple con los criterios de aceptación o no:



9. Metodología scrum

A pesar de haber utilizado la metodología scrum como marco de trabajo en todo el desarrollo de la aplicación, se decidió implementar aspectos de la metodología tradicional como los diagramas UML para brindar una mayor comprensión al lector con respecto al flujo de datos y funciones dentro del software, esto se tuvo en cuenta ya que por temas de confidencialidad no fue permitido evidenciar el aplicativo en su totalidad.

Para entender mejor el funcionamiento y la esencia de scrum a continuación se definirán algunos conceptos básicos de los actores y procesos que normalmente se involucran en los equipos de trabajo cuando se utiliza esta metodología ágil.

9.1 Scrum

Scrum es "un proceso de gestión que reduce la complejidad en el desarrollo de productos para satisfacer las necesidades de los clientes. La gerencia y los equipos de Scrum trabajan juntos alrededor de requisitos y tecnologías para entregar productos funcionando de manera incremental ..." (Joel Francia, 2017), ¿Qué es Scrum?. 27/05/2020, extraida de scrum.org Sitio web: https://www.scrum.org/resources/blog/que-es-scrum). Dicho en otras palabras la metodología scrum es un marco de trabajo conformado por grupos altamente productivos aliados entre sí para realizar entregas periódicas de un proyecto con el fin de brindar un valor agregado al cliente.

9.2 Roles scrum

Scrum Team: Está conformado por todos las partes interesadas del proyecto (product owner, scrum master y development team).

Product Owner: Edward Hernandez CEO Ice company sas.

También conocido como dueño del producto, es la persona que toma las decisiones y tiene conocimiento real del negocio del cliente junto a su visión; se encarga de tomar nota de cada idea del cliente para luego priorizarlas y plasmar el Product Backlog (lista del producto). Sus funciones principales son:

- Expresar claramente los elementos de la Lista del Producto
- Ordenar los elementos en la Lista del Producto para alcanzar los objetivos y misiones de la mejor manera posible.
- Optimizar el valor del trabajo que el Equipo de Desarrollo realiza

Scrum Master: Sebastian Pachón Programador senior**.**

Es el encargado de garantizar que la metodología scrum se realice de una manera adecuada y precisa, asegura que el dueño del producto conozca cómo ordenar la Lista de Producto para maximizar el valor y ofrece ayuda a todo el scrum team para mitigar cada obstáculo que retrase el proyecto.

Development team: Ivan Quintero, Julian Chavez, Juan Gutiérrez(Proyecto complementario)

Es el equipo delegado para realizar el desarrollo incremental de cada parte del producto "terminado" (tareas del product backlog incluidas en el sprint).

9.3 Elementos scrum

Product backlog: Es el inventario donde se almacena todo lo que podría ser necesario en el producto, se ve reflejado como una lista de necesidades del cliente qué son priorizadas por el product owner con el fin de estimar los requerimientos del proyecto. Un product backlog nunca

está completo, está abierto a la adición de nuevas tareas a medida que los incrementos son entregados y puestos en producción.

En el proyecto se utilizó la herramienta Jira software para administrar el product backlog, en dicha herramienta se le conoce como “hoja de ruta”.

Sprint backlog: Es la lista de tareas que elabora el equipo para cada sprint.

El sprint backlog de todo el proyecto fue manejado a través de la herramienta Jira Software.

Historia de usuario: Es la descripción de cada funcionalidad que va a tener el software, la historia de usuario se crea mediante la colaboración del cliente y el equipo. Su estructura estándar es la siguiente:

Historia de Usuario	
Número: 1	**Usuario:** Cliente
Nombre historia: Cambiar dirección de envío	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Baja
Puntos estimados: 2	**Iteración asignada:** 1
Programador responsable: José Pérez	
Descripción: Quiero cambiar la dirección de envío de un pedido.	
Validación: El cliente puede cambiar la dirección de entrega de cualquiera de los pedidos que tiene pendientes de envío.	

Imagen Historia de usuario extraida de: https://www.scrummanager.net/bok/index.php?title=File:Historia_usuario_ejemplo_1.jpg

9.4 Interacciones scrum

Sprint: Es el intervalo de tiempo en el cual se desarrollan tareas establecidas por el equipo de trabajo en el sprint planning, generando una iteración o incremento del producto final (al finalizar un sprint, se debe iniciar automáticamente uno nuevo).

En Flyke se realizaron en total 7 sprints, donde se tenía muy presente las tareas por hacer, su tiempo estimado y la duración de todo el sprint; la mayoría de los sprints duraron un tiempo de tres semanas, a excepción de otros con una duración de 1 mes o mes y medio. La organización del sprint se manejaba por medio de la herramienta Jira Software, donde se visualizaba un tablero con 4 secciones, "por hacer" (sprint backlog de tareas pendientes), "en curso"(tareas en desarrollo con puntos de historia y desarrollador responsable), "QA" (tareas realizadas que estaban en proceso de testing por otros dos pasantes Miguel Herrera y Juan Roa) y "listo" (tareas ya desarrolladas y aprobadas por el equipo de QA).

Sprint Planning: Es la reunión del scrum team donde se asignan las tareas a trabajar durante el sprint, definiendo los puntos de historia (tiempo empleado para realizar una tarea, 1 punto = 4 horas de trabajo), el desarrollador responsable y el tiempo límite del sprint (un sprint planning máximo puede durar 8 horas para un sprint de 1 mes).

Para realizar el sprint planning durante toda la pasantía se designó como herramienta de comunicación Skype, ya que el scrum master residía en Bogotá, era la manera más fluida para hacer este encuentro; el sprint planning duraba aproximadamente de dos a tres horas en las que se reunía el Scrum team que se conformaba por el Scrum master: Sebastian Pachon, Product owner: Eduard Hernandez, developer team: Julian Chavez Chavarro, Ivan Quintero Torres y el estudiante Juan David Gutierrez del proyecto que complementa a este (proyecto 2), QA team (Testing): Miguel Herrera, Juan Manuel Roa. Las tareas para el sprint backlog eran asignadas por el scrum master, dichas tareas se determinan tomando como referencia un tiempo estimado fundamentado en puntos de historia, también previamente acordado al inicio de todo el desarrollo del software, el cual en nuestro caso cada punto de historia es de aproximadamente 4 horas de trabajo, para la asignación de dichas tareas utilizamos una herramientas de software llamadas BITBUCKET la cual es una herramienta muy intuitiva y muy útil para llevar un control de todas las tareas que se pretenden realizar durante el desarrollo del software, es decir, bitbucket

entra a la metodología scrum como nuestro product backlog, todas las herramientas de software que utilizamos para llevar a cabo el desarrollo del proyecto están conectadas entre sí, lo que hace que el trabajo realizado sea tanto transparente como compacto y también nos brinda la posibilidad de monitorear todo el trabajo realizado durante los sprints.

Daily Scrum: Es una reunión diaria donde se reúne el development team para sincronizar sus actividades del día y hacer un feedback de lo desarrollado hasta el momento, respondiendo a los interrogantes: ¿Que hice la sesión anterior?, ¿Impedimentos encontrados? ¿Qué haré hoy?.Tiene una duración de aproximadamente 15 minutos.

A lo largo del proyecto el daily scrum se realizaba por medio de Slack, una herramienta muy potente para la comunicación entre equipos de trabajo a nivel mundial. Esta aplicación brinda la opción de programar un chatbot, el cual se asignó para realizar todos los días a las 9 am las 3 preguntas básicas de daily (¿Que hice la sesión anterior?, ¿Impedimentos encontrados? ¿Qué haré hoy?) las respuestas recolectadas por el chatbot eran visualizadas por el scrum master con el fin de notificar algún impedimento o avance del development team.

Sprint Review: Al finalizar un sprint se hace una reunión para revisar el incremento realizado, en caso de algún inconveniente específico el grupo brinda colaboración para solucionarlo y poder avanzar. De igual manera se actualiza la lista del producto para tener conocimiento qué tareas faltan por ejecutar.

En Ice Company sas este espacio era tomado en el momento del sprint planning, se aprovechaba la reunión por Skype donde se realizaba la revisión de las tareas desarrolladas por cada pasante y las respectivas observaciones.

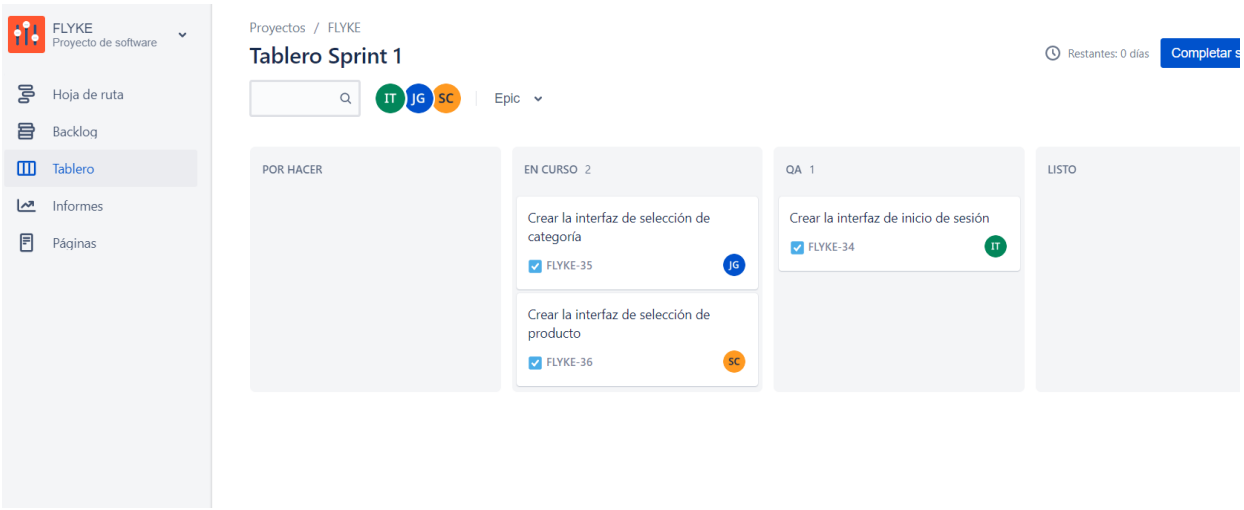
Sprint Retrospective: Es un bloque de tiempo donde el scrum team tiene la oportunidad de autoevaluarse, revisar cómo fue el desempeño en el anterior sprint e identificar mejoras para crear cultura de trabajo que brinde el crecimiento continuo.

En el transcurso del desarrollo de la aplicación Flyke el sprint retrospective también era realizado en el momento del sprint planning, se hacía un feedback a todo el personal de desarrollo y QA(development team, QA team), a los incrementos generados y las herramientas utilizadas.

9.5 Desarrollo Sprint 1

Para el primer sprint planning se utilizó la herramienta skype acordamos comenzar a crear las interfaces principales, como lo son las de inicio de sesión, selección de categoría que luego cambiaría su nombre a la de selección de módulos y la interfaz de selección de producto;

En la siguiente imagen se utiliza Scrum Board dividido por varias secciones. La sección “por hacer” representa, las tareas propuestas que aún no se han asignado, mientras que en la segunda sección “en curso” estarán las tareas que ya se están realizando, lo que implica que ya tienen un desarrollador responsable de esta, en la tercera sección “QA” se ubican aquellas tareas que se encuentran en revisión o técnicamente hablando en la etapa de testing, de la cual se encarga el QA team, quienes después de revisar deciden si la tarea se devuelve a la sección anterior en caso de que haya un bug o esté incompleta, si por el contrario la tarea cumple con los criterios de aceptación pasa a la sección final “listo” lo que quiere decir que la tarea fue realizada con éxito.



La siguiente tabla hace relación a la historia de usuario número 1, que hace referencia a la construcción de la interfaz, resaltando que se construye el esqueleto principal de la aplicación y que sin embargo las demás interfaces se iban creando transversalmente durante el desarrollo:

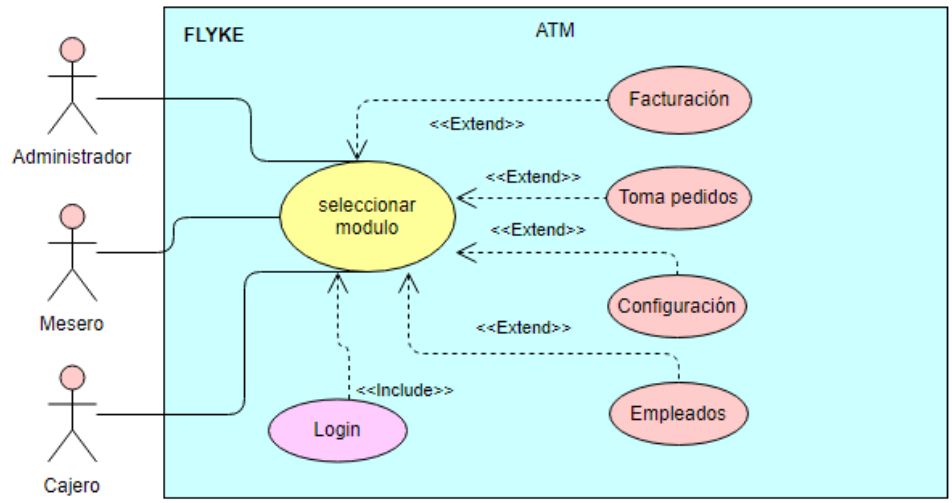
	Historia de usuario 001
Número: 001	**Usuario:** administrador - mesero - cajero
Nombre de historia: Interfaz de aplicación	
Prioridad: Alta	**Riesgo en desarrollo:** Medio
Puntos estimados: 10	**Iteración asignada:** 1
Programador responsable: Julian Chavez, Ivan Quintero	
Descripción: como administrador quiero una interfaz muy intuitiva y atractiva	
Validación: • se debe usar una gama de colores azules • Se deben crear una versión dark • Se deben crear una versión light • el contenido debe ser legible	

A continuación se detalla el caso de uso que representa la tarea de selección de módulo, realizada durante el primer sprint del proyecto y la cual es fundamental para acceder a cada uno de los módulos del aplicativo móvil:

Información de Catalogación			
Proyecto	Flyke		
Autor	Ivan Quintero Torres		
Versión	1,0	**Estado de Desarrollo**	Desarrollado
Definición del Caso de Uso			
Código	Caso de uso - 01		
Nombre	Selección de módulo		
Objetivo	El usuario logueado elegirá el módulo en el cual va a trabajar		
Descripción	Después de digitar el pin correctamente el usuario accede a la vista de módulos, en donde se visualizan 4 módulos. (Revisar cantidad de módulos)		
Actores	Administrador Empleados (Meseros, cajeros)		
Precondición	Digitar el pin correctamente		
Escenario Principal	**No.**	**Descripción de acciones**	
	1	El sistema muestra la vista de módulos según los permisos del usuario que inicio sesión	
	2	El usuario accede a alguno de los módulos.	
	3	El usuario accede al módulo seleccionado y termina el caso de uso	
	4		
Escenario Alternativo	**No.**	**Descripción de acciones alternas**	
	2a	El usuario accede al módulo de facturación	
	2b	El usuario accede al módulo de empleados	
	2c	El usuario accede al módulo de toma pedidos	
	2d	El usuario accede al módulo de cierre de caja	

	2e	El usuario accede a la opción de ajustes
	3a	El caso de uso continúa en el piso 3 del escenario principal
Escenarios de Excepción	**No.**	**Descripción de acciones de excepción**
		No existen excepciones para este caso de uso
Postcondición	El usuario accede a la vista del módulo seleccionado	

El siguiente diagrama representa cómo acceden los actores principales de la aplicación a esta sección y que otros procesos extienden de "seleccionar módulo", tenga en cuenta que este caso de uso puede cambiar dependiendo de los módulos que se le vayan agregando al proyecto:



Caso de uso realizado con la herramienta lucidchart

La siguiente tabla representa la historia de usuario correspondiente a la tarea correspondiente a la selección de módulo teniendo en cuenta que el acceso a dichos módulos va dependiendo de los permisos que se otorguen en el tema de seguridad:

	Historia de usuario 002
Número: 002	**Usuario:** administrador, cajero, mesero
Nombre de historia: Selección de módulo	

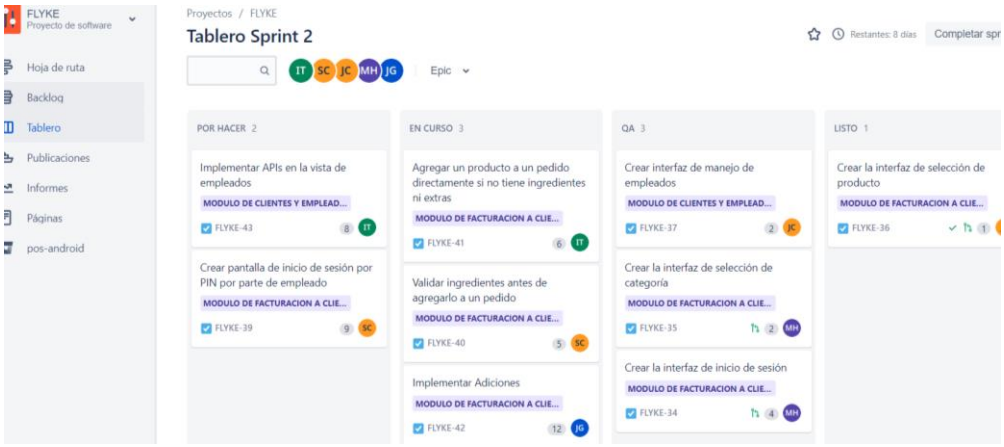
Prioridad: Media - Alta	**Riesgo en desarrollo:** Medio
Puntos estimados: 7	**Iteración asignada:** 1
Programador responsable: Ivan Quintero	
Descripción: como usuario quiero poder acceder a cada uno de los módulos de la aplicación, dependiendo de los permisos de cada rol.	
Validación: ● se debe tener una imagen en representación de cada módulo ● la interfaz debe tener una vista de cuadrícula ● el acceso a estos módulos varían dependiendo de los permisos de cada usuario	

Resumen sprint 1 (review, retrospective, planning):

Al final del primer sprint se tiene en primera instancia un conocimiento más a fondo de la metodología scrum adecuada a las herramientas manejadas por la empresa (Jira, Bitbucket, Git) que serán manejadas durante todo el proyecto. Pudimos notar que la construcción de interfaces iniciales es fundamental, al principio, en este sprint realizamos unas interfaces iniciales las cuales fueron planteadas por los desarrolladores, sin tener en cuenta ningún concepto de UI/UX por lo que la empresa decidió que era mejor realizar la construcción de interfaces en una herramienta dedicada a esto (Adobe XD) y tenerlas listas para cuando nosotros los desarrolladores las fuéramos a construir, también se pudo realizar la implementación de selección de módulo con información quemada, es decir, sin traer la información real que se debía traer de la API rest, finalmente se planeó empezar con tareas enfocadas al módulo de facturación.

9.6 Desarrollo Sprint 2

En el desarrollo del sprint 2 se implementaron una mayor cantidad de tareas, debido a que el scrum master puede calcular mediante el desarrollo del primer sprint que el tiempo estimado da para que el scrum team pueda ser más productivo y entregar una mayor cantidad de tareas:

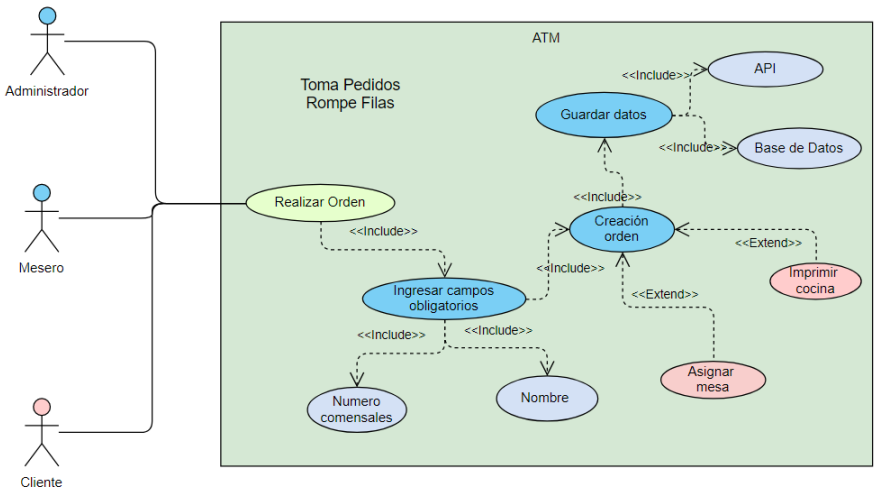


La siguiente tabla representa el caso de uso para cargar la orden en el tipo de restaurantes que contiene mesas, recordemos que los otros dos tipos de restaurantes (rompe-filas, foodtruck) no manejan el mismo flujo para su operación:

Información de Catalogación			
Proyecto	Flyke		
Autor	Julian Chavez Chavarro - Ivan Quintero		
Versión	1,0	**Estado de Desarrollo**	Desarrollado
Definición del Caso de Uso			
Código	Caso de uso - 02		
Nombre	Cargar Orden		
Objetivo	Cargar orden módulo de facturación - Toma pedido restaurante con mesas		
Descripción	El usuario carga ordenes desde el módulo de facturación y toma pedido		
Actores	Administrador y cajero		
Precondición	Seleccionar el módulo de facturación		
Escenario Principal	**No**	**Descripción de acciones**	
	1	El sistema muestra la vista de mesas	
	2	El usuario selecciona la mesa disponible	

	3	El sistema muestra la vista de los diferentes productos, la cantidad de productos y el resumen de la factura (si tiene productos)
	4	El usuario agrega los productos con las cantidades deseadas
	5	El sistema muestra el resumen de la factura
	6	El usuario realiza la orden
	7	El sistema almacena los datos de la orden en la API y la Base de Datos
	8	El sistema imprime la orden a cocina para su preparación
	9	El sistema muestra la vista de mesas con la orden cargada a la mesa correspondiente y termina el caso de uso
Escenario Alternativo	**No**	**Descripción de acciones alternas**
	1a	El usuario cierra la aplicación y termina el caso de uso
	1b	El usuario regresa a la vista de módulos y termina el caso de uso
Escenarios de Excepción	**N.**	**Descripción de acciones de excepción**
	7a	No hay conexión a la API y no se puede almacenar la orden.
	7b	El usuario se comunicará con los desarrolladores de la aplicación para solucionar el conflicto y finaliza el caso de uso
Postcondición	El usuario realiza, imprime a cocina y agrega la orden a la mesa correspondiente.	

El siguiente gráfico es para el caso de uso que explica la realización de una orden el un restaurante de rompe-filas y todos los procesos que implican realizar esta acción:

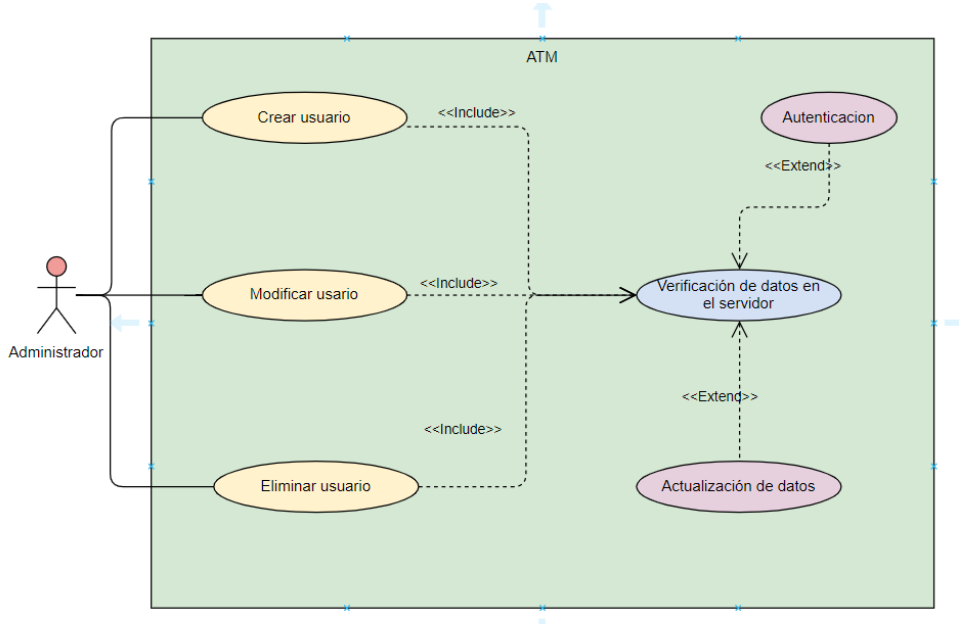


En la siguiente historia de usuario se nombran los criterios a tener en cuenta por parte del desarrollador responsable para dar vida y diseño al módulo de empleados, además se describen las acciones que el usuario (en este caso administrador) puede ejercer para la gestión de los empleados del negocio, incluyendo las validaciones que se deben tener en cuenta por parte del sistema para el correcto funcionamiento de este módulo:

Historia de usuario 003	
Número: 003	**Usuario:** Administrador
Nombre de historia: Implementar interfaz y crud de módulo de empleados	
Prioridad: Alta	**Riesgo en desarrollo:** Medio
Puntos estimados: 9	**Iteración asignada:** 1
Programador responsable: Ivan Quintero	
Descripción: Como administrador quiero un modulo que me permita ver la lista de los empleados vinculados al restaurante	
Validación: ● Se debe crear una interfaz siguiendo los colores establecidos por el diseñador. ● Se debe visualizar un listado de los empleados pertenecientes al negocio ● Cada ítem de la lista debe contener al final 2 iconos representando las opciones de editar y eliminar.. ● Para crear un empleado se debe contar con un botón representado con el icono (+), al seleccionarse abrirá modal el cual tendrá los campos de correo electrónico, rol a asignar y el pin de seguridad. ● Para la modificación de cada empleado debe aparecer en la interfaz un modal con sus datos básicos nombre, rol, pin de seguridad(solo puede ser modificado el rol y pin del empleado). ● En la opción eliminar empleado se debe abrir un modal con un enunciado de confirmación. ● El cajero tendrá el permiso de facturación y toma pedidos. ● El mesero tendrá el permiso de toma pedidos.	

Para el módulo de gestión de empleados es necesario contar con un esquema inicial de backend (desarrollado por Juan David Gutierrez del proyecto 2) y de esta manera acceder a los servicios

del API rest, el cual nos hará una verificación de datos y con una respuesta de servidor para el crud que se implementa desde la misma aplicación:

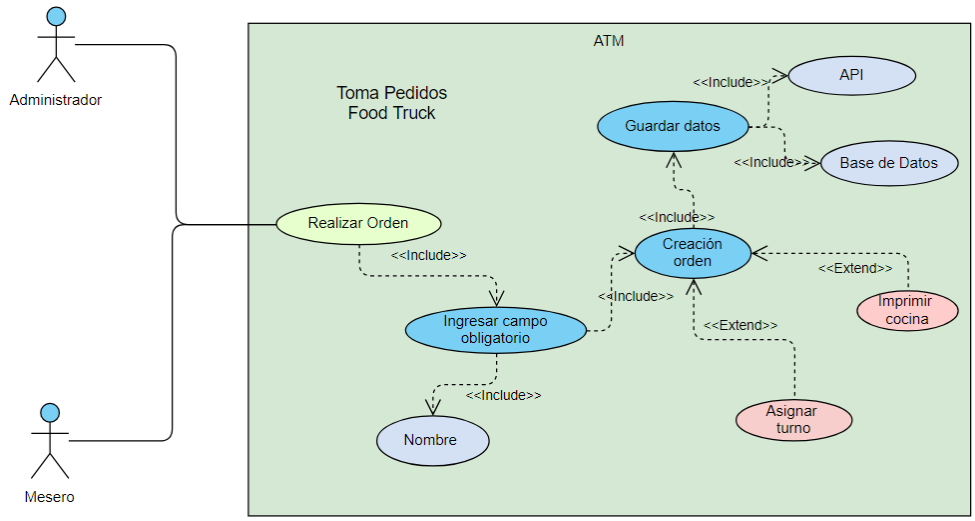


Esta historia de usuario describe las validaciones que debe tener el sistema para que el usuario pueda crear un pedido correctamente, teniendo en cuenta todos los subprocesos que la complementan:

	Historia de usuario 004
Número: 004	**Usuario:** administrador - Cajero
Nombre de historia: Realizar pedido	
Prioridad: Media - Alta	**Riesgo en desarrollo:** Alto
Puntos estimados: 10	**Iteración asignada:** 1
Programador responsable: Julian Chavez, Ivan Quintero	
Descripción: como usuario quiero crear un pedido con productos y cantidades respecto a lo ordenado por el cliente	
Validación: ● Se debe agregar a la orden adiciones e ingredientes de determinados productos ● las adiciones deben verse reflejadas y poder ser seleccionadas.	

- Si el producto tiene ingredientes obligatorios se debe validar que estén seleccionados correctamente
- Si un producto no tiene ingredientes ni adiciones debe agregarse directamente al pedido activo
- Al darle clic a una categoría se debe visualizar la lista de productos correspondiente y poder filtrar mediante un search.

Este gráfico de caso de uso muestra el proceso de realizar una orden desde el tipo de restaurante food truck, que cambia un poco respecto a los otros tipos de restaurante, en casos como agregar un nombre y asignar un turno al cliente:

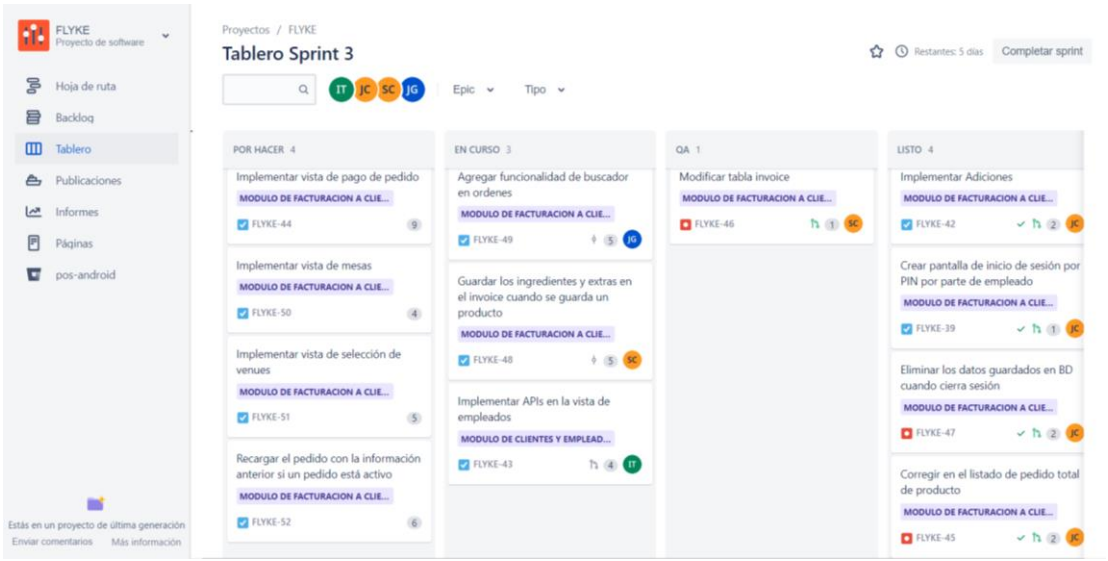


Resumen sprint 2 (review, retrospective, planning):

En la reunión del segundo sprint ya se trabajó algunas de las funcionalidades tanto de facturación como del toma-pedidos, enfocados a la creación de órdenes desde el API rest y utilizando esta información para dentro de la aplicación para poder visualizarla y modificarla según se requiera, el módulo de empleados quedó planteado en cuanto a interfaz y una primera funcionalidad que fue la visualización de empleados, el scrum master observó una mejora en el rendimiento del desarrollo de las tareas por lo que vio necesario aumentar la cantidad de tareas para el siguiente sprint, en las que se debía seguir trabajando con el módulo de facturación y los datos desde el API rest.

9.7 Desarrollo Sprint 3

En el desarrollo del sprint 3 se hizo énfasis en el módulo de facturación, debido a que este módulo desprende muchos otros subprocesos fundamentales para el flujo de la aplicación, las tareas de este sprint acaparan demasiado tiempo (puntos de historia) lo que hace que este sprint se extienda un poco más de lo esperado:

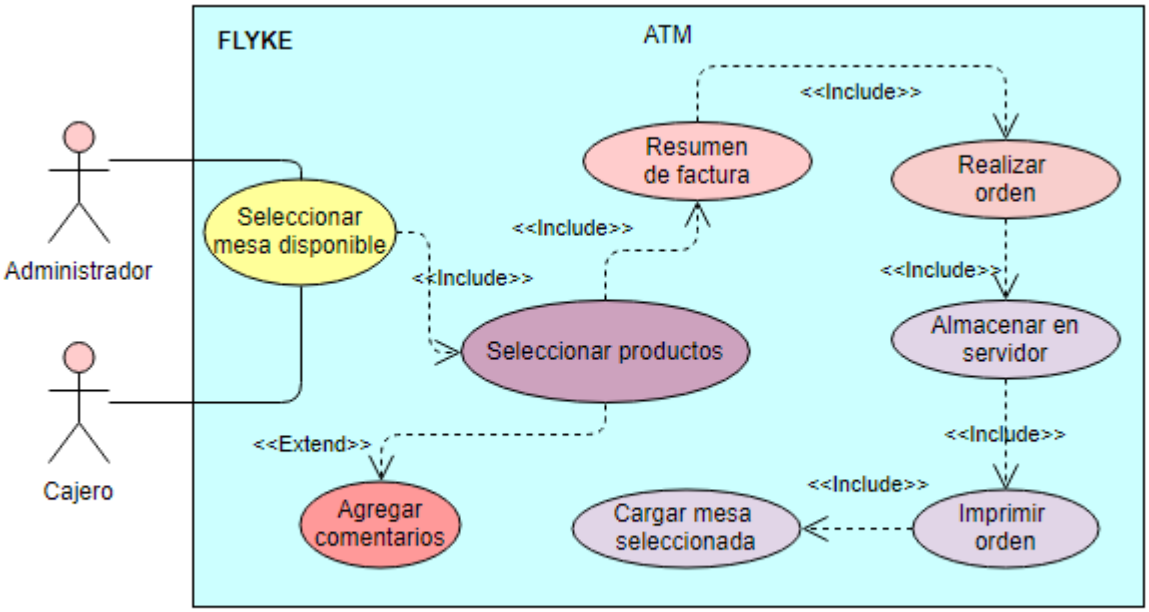


En esta historia de usuario se describen las diferentes validaciones que debe tener el apartado de las mesas, el cual permite ver un listado de mesas tanto disponibles como no disponibles y otras opciones necesarias para la gestión de las mesas del restaurante:

	Historia de usuario 005
Número: 005	**Usuario:** administrador - Cajero - Mesero
Nombre de historia: Seleccionar mesa	
Prioridad: Media	**Riesgo en desarrollo:** Medio
Puntos estimados: 5	**Iteración asignada:** 1
Programador responsable: Ivan Quintero	

Descripción: como usuario quiero seleccionar mesas para cargar, actualizar, imprimir y pagar los pedidos.
Validación: • Se debe mostrar un listado con las mesas del restaurante • Se debe mostrar el listado de mesas activas • sí una mesa tiene un pedido activo debe mostrarlo y un resumen del total • agregar 3 botones al lado de la mesa, botón de agregar o actualizar, botón de pagar y botón de imprimir

Este gráfico da a conocer el flujo que tienen los datos al seleccionar una mesa disponible en la vista de mesas, como podemos observar, incluye otro proceso fundamental (seleccionar productos) que incluye aún más subprocesos importantes:



Esta es una historia de usuario clave para la experiencia de usuario, puesto que describe la función de llamar al mesero mediante un botón dedicado con el fin de atender dudas personalmente al cliente:

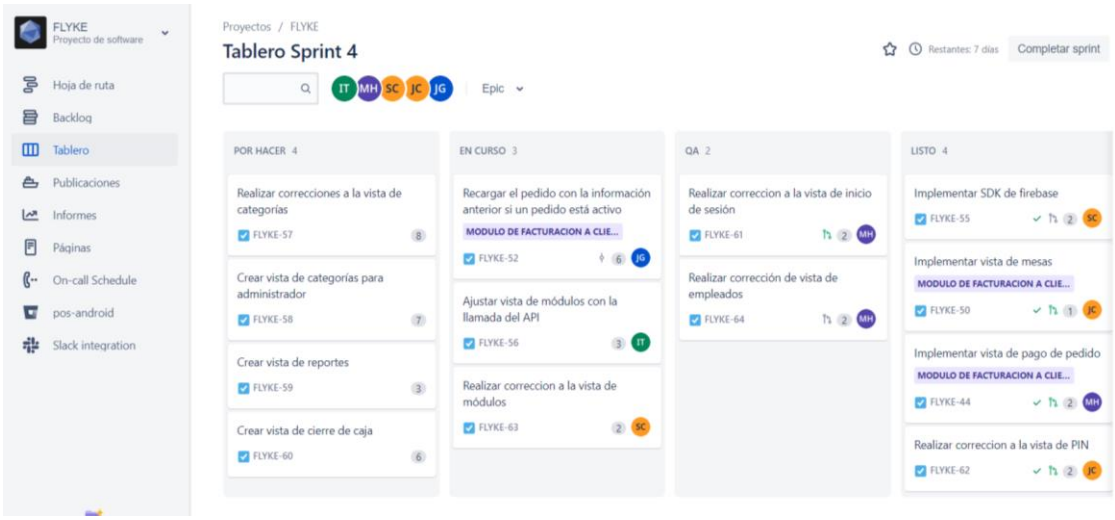
Historia de usuario 006	
Número: 006	**Usuario:** Comensal
Nombre de historia: Toma pedido restaurante, llamar mesero	
Prioridad: Baja - Media	**Riesgo en desarrollo:** Medio
Puntos estimados: 4	**Iteración asignada:** 1
Programador responsable: Julian Chavez	
Descripción: Como comensal necesito hacer un llamado al mesero encargado en caso de no saber cómo ejecutar alguna acción en este módulo.	
Validación: ● El llamado a meseros debe ser una opción visible y fácil de reconocer en la interfaz de usuario. ● Debe ser un botón de icono, ubicado en la parte superior ● Una vez presionado el botón se debe enviar una notificación a la tablet en caja para comunicar al encargado la llamada al mesero.	

Resumen sprint 3 (review, retrospective, planning):

Las tareas realizadas en este sprint tenían un mayor grado de complejidad, por lo que se extendió el sprint más de lo esperado, durante la reunión el scrum master nos llamó la atención por estas demoras, sin embargo se tuvo en cuenta que en la construcción de interfaces por parte de la empresa también habían tardado por parte de la empresa por lo tanto era justificable y finalmente se acordó continuar con lo restante de las tareas en el sprint 4 que se enfocan en detalles como la descripción del producto y los comentarios por producto y por pedido, mientras que la mayoría de tareas por ejemplo la de implementar la funcionalidad de llamado a mesero si quedo realizada en este mismo sprint como se había acordado.

9.8 Desarrollo Sprint 4

En el cuarto sprint se trabaja en gran parte la creación de interfaces que representan nuevas funcionalidades del aplicativo, así como también la funcionalidad de actualizar o recargar una orden activa, para el caso de que se requiera agregar un nuevo producto a una misma orden:

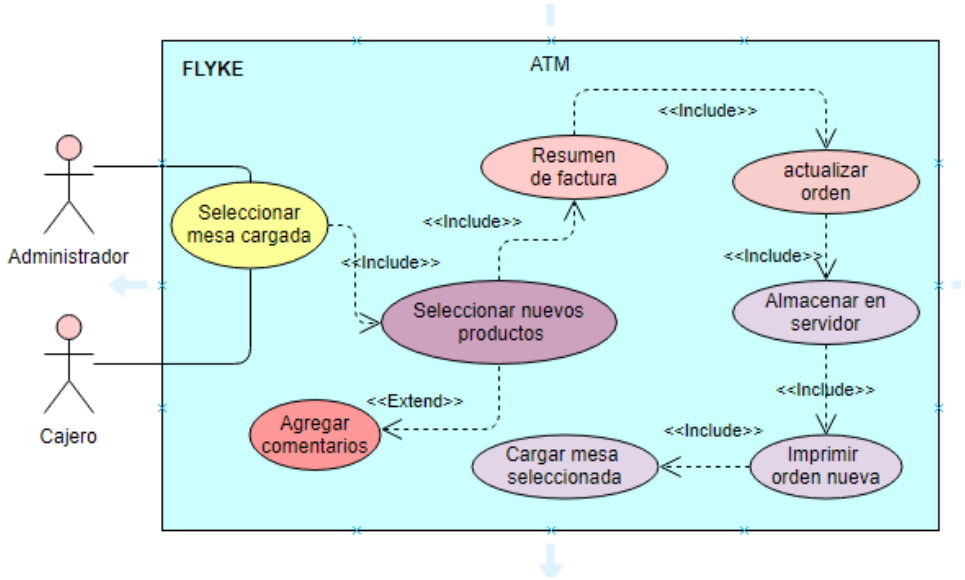


Este caso de uso tiene como nombre “Actualizar orden” y es pieza clave para que continúe el flujo de la información a través del módulo de facturación y toma-pedidos:

Información de Catalogación			
Proyecto	Flyke		
Autor	Ivan Quintero Torres		
Versión	1,0	**Estado de Desarrollo**	Desarrollado
Definición del Caso de Uso			
Código	Caso de uso - 03		
Nombre	Actualizar Orden		
Objetivo	Cargar orden módulo de facturación		
Descripción	El usuario actualiza órdenes desde el módulo de facturación para restaurante tipo mesas		
Actores	Administrador y cajero		
Precondición	Seleccionar el módulo de facturación		
	No	**Descripción de acciones**	

Escenario Principal	1	El sistema muestra la vista de mesas
	2	El usuario selecciona la mesa con una orden cargada
	3	El sistema muestra la vista de los diferentes productos, la cantidad de productos y el resumen de la factura
	4	El usuario agrega los productos con las cantidades deseadas
	5	El sistema muestra el resumen de la factura
	6	El usuario actualiza la orden
	7	El sistema actualiza los datos de la orden en la API y la Base de Datos
	8	El sistema imprime los productos nuevos a cocina para su preparación
	9	El sistema muestra la vista de mesas con la orden actualizada a la mesa correspondiente y termina el caso de uso
Escenario Alternativo	**No**	**Descripción de acciones alternas**
	1a	El usuario cierra la aplicación y termina el caso de uso
	1b	El usuario regresa a la vista de módulos y termina el caso de uso
Escenarios de Excepción	**No**	**Descripción de acciones de excepción**
	7a	No hay conexión a la API y no se puede almacenar la orden.
	7b	El usuario se comunicará con los desarrolladores de la aplicación para solucionar el conflicto y finaliza el caso de uso
Postcondición	El usuario agrega nuevos productos, imprime a cocina y actualiza la orden a la mesa correspondiente.	

Para actualizar una orden es necesario seleccionar una mesa que tenga una orden activa, para acceder a todos los datos de los productos previamente cargados y de esta manera poder agregar un nuevo producto como se denota en este gráfico:



Esta la historia de usuario que corresponde a la de actualizar un pedido, se describe las validaciones que debe tener el sistema para realizar esta acción, se debe recordar que el hecho de actualizar un pedido implica recuperar la información para agregarle los nuevos productos:

	Historia de usuario 007
Número: 007	**Usuario:** administrador - cajero
Nombre de historia: Actualizar pedido	
Prioridad: Media - Alta	**Riesgo en desarrollo:** Alta
Puntos estimados: 7	**Iteración asignada:** 1
Programador responsable: Ivan Quintero	
Descripción: como usuario quiero actualizar el pedido con nuevos productos, pero sin perder los productos anteriores	
Validación: • Si la mesa tiene un pedido activo se debe reflejar lo pedido anteriormente • Se deben poder agregar ítems nuevos al pedido sin agregar los ya existentes	

La historia de usuario a continuación describe cómo debe ser la interfaz para cierre de caja, que aunque no se entrega con funcionalidades, requiere de puntos de historia para su desarrollo:

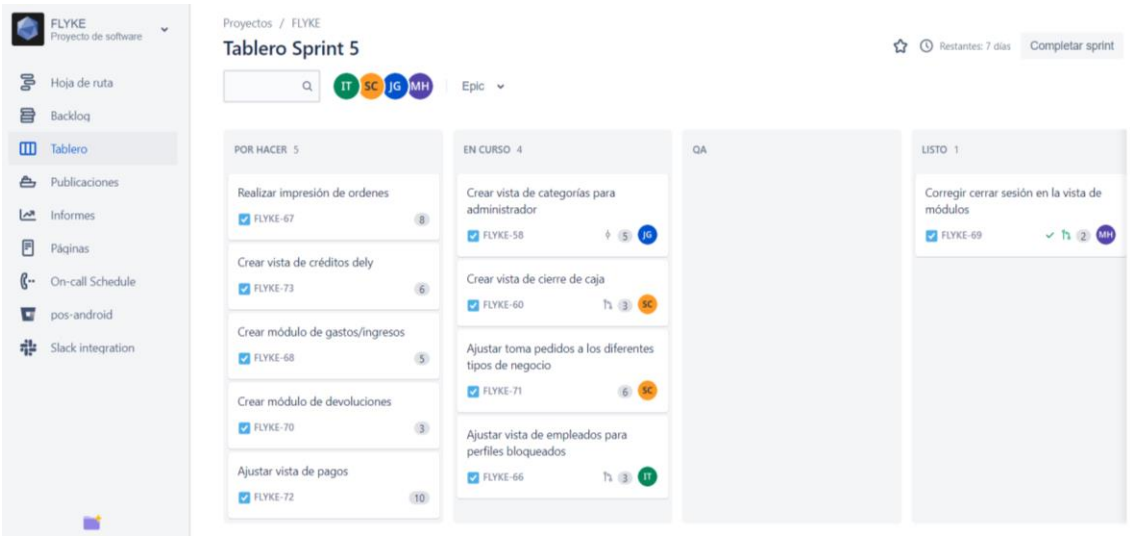
Historia de usuario 008	
Número: 008	**Usuario:** administrador - cajero
Nombre de historia: Interfaz cierre de caja.	
Prioridad: Baja	**Riesgo en desarrollo:** bajo
Puntos estimados: 3	**Iteración asignada:** 1
Programador responsable: Julian Chavez	
Descripción: Como administrador quiero una interfaz intuitiva donde se gestionen todos los flujos de dinero que se han involucrado en un dia trabajo, independientemente del día que necesite, quiero poder consultar el flujo de tarjetas, efectivo y créditos.	
Validación: ● La interfaz debe seguir en estándar de colores de toda la aplicación. ● La interfaz debe contener un menú fácil de identificar donde se puedan seleccionar los dispositivos, cajeros y fechas a consultar. ● Debe contener una tabla donde sea posible visualizar las ventas por usuario. ● Debe poseer una lista con todos los medios de pago recibidos por el restaurante y el total en ventas de cada uno.	

Resumen sprint 4 (review, retrospective, planning):

Durante la reunión al final de este sprint retroalimentamos respecto a las tareas de recargar y actualizar orden, pues se había cumplido a cabalidad dicha tarea que es pieza clave en el flujo de datos del aplicativo, también se terminó por completo el módulo de empleados y se trabajó en la construcción de una interfaz para el cierre de caja (únicamente interfaz), por lo que el scrum master definió este sprint como uno de los más productivos hasta este punto del desarrollo del proyecto y se acordó realizar tareas más generales para el siguiente sprint, entre ellas comenzar a trabajar con la funcionalidad de imprimir.

9.9 Desarrollo Sprint 5

En el sprint 5 vemos reflejadas las funcionalidades de impresión de órdenes, además se sigue con el desarrollo de algunas tareas que no fueron finalizadas en el sprint 4 como lo es la creación de la vista de categorías para el administrador y el ajuste de toma pedidos a distintos tipos de restaurante:

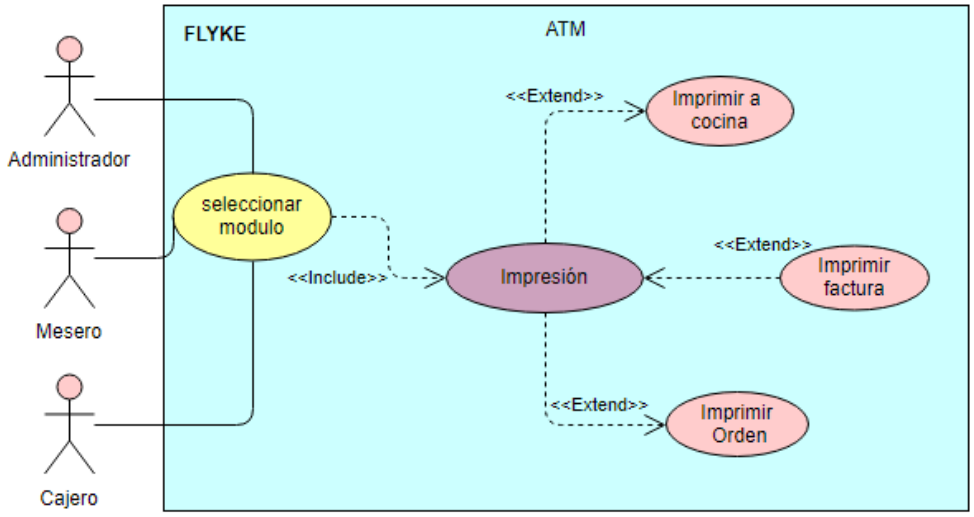


En este caso de uso se describe los lineamientos que debe poseer la funcionalidad de imprimir, esta operatividad del software es muy importante ya que es transversal para todo Flyke.

Información de Catalogación			
Proyecto	Flyke		
Autor	Ivan Quintero Torres		
Versión	1	**Estado de Desarrollo**	Desarrollado
Definición del Caso de Uso			
Código	Caso de uso - 04		
Nombre	Imprimir		
Objetivo	Realizar Impresiones		

Descripción	Desde el módulo el usuario puede realizar 3 tipos de impresiones: Imprimir a cocina, imprimir orden e imprimir factura	
Actores	Administrador, mesero y cliente	
Precondición	Seleccionar el módulo respectivo para impresión	
Escenario Principal	**No.**	**Descripción de acciones**
	1	El sistema muestra la vista del módulo
	2	El usuario oprime el botón de impresión
	3	El sistema muestra un menú con 3 tipos de impresión, cocina, orden y factura
	4	El usuario selecciona el tipo requerido
	5	El sistema realiza la impresión según los criterios del usuario y finaliza el caso de uso
Escenario Alternativo	**No.**	**Descripción de acciones alternas**
	2a	El usuario selecciona impresión a cocina
	2b	El sistema imprime únicamente cantidades y productos para su elaboración y finaliza el caso de uso
	2c	El usuario selecciona impresión de orden
	2d	El sistema imprime cantidades, productos y precios y finaliza el caso de uso
	2e	El usuario selecciona impresión de factura
	2f	El sistema imprime cantidades, productos, precios, impuestos e información de interés y finaliza el caso de uso.
Escenarios de Excepción	**No.**	**Descripción de acciones de excepción**
	5a	No se realizó la impresión
	7b	El usuario revisará el papel de la impresora o si tiene alguna cola de impresión, si no se soluciona debe comunicarse con algún experto o con los desarrolladores de la aplicación y finaliza el caso de uso.
Postcondición	El usuario realiza los 3 tipos de impresiones: Imprimir a cocina, imprimir orden e imprimir factura.	

A continuación en el esquema se evidencian las pautas y el flujo necesario para realizar el proceso de impresión en cualquiera de los módulos de la aplicación, teniendo en cuenta que la aplicación soporta la conectividad de varias impresoras:



En la siguiente historia de usuario nombrada “imprimir pedido” se nombran las propiedades que debe tener dicha funcionalidad y los criterios de aceptación válidos.

	Historia de usuario 009
Número: 009	**Usuario:** Administrador - cajero
Nombre de historia: Imprimir pedido	
Prioridad: Media - Alta	**Riesgo en desarrollo:** Alto
Puntos estimados: 8	**Iteración asignada:** 1
Programador responsable: Ivan Quintero	
Descripción: Como usuario quiero realizar las impresiones de los pedidos para enviarlos a cocina y para que una vez que se pague entregar la factura.	
Validación: ● la factura debe contener los productos del pedido, cantidades, precios y hora, la impresión debe tener la información básica del negocio como si declara impuesto o no	

● la Imprimir del pedido debe contener productos, cantidades y precios ● la impresión a cocina debe contener únicamente productos y cantidades ● posibilidad de reimpresión de factura

En la historia de usuario número 9 se especifican las funciones que debe desarrollar el modulo toma pedidos dependiendo el tipo de restaurante registrado.

Historia de usuario 010	
Número: 010	**Usuario:** Administrador
Nombre de historia: Seleccionar tipo de restaurante para adecuar el módulo toma pedidos.	
Prioridad: Alta	**Riesgo en desarrollo:** Alto
Puntos estimados: 8	**Iteración asignada:** 1
Programador responsable: Julian Chavez	
Descripción: Como usuario quiero poder modificar el tipo de restaurante para adecuar el módulo de toma pedidos y agilizar la facturación por medio de tablets.	
Validación: ● Como parámetro principal el usuario debe loguearse, de acuerdo a la respuesta de la API, si el tipo de restaurante registrado es Food Truck, la configuración local no se puede realizar, porque ese modelo de negocio tiene por defecto una configuración asignada. ● Si el restaurante es tipo restaurante con mesas, se habilita el módulo de configuración para poder elegir. ● En el módulo de configuración aparecerá un widget de tipo SwitchPreferenceCompat con la opción “rompe filas” deshabilitada. ● Si el usuario desea trabajar con tablets en cada una de las mesas, debe dejar esa configuración por defecto (rompe filas inhabilitado). ● Si el usuario desea trabajar solo con algunas tablets y destinar un lugar específico del restaurante para que desde allí el usuario haga su pedido, debe habilitar la opción Rompe filas.	

En la siguiente tabla se habla del proceso para crear un pedido en un restaurante de tipo rompe filas siguiendo los parámetros establecidos por el product owner:

	Historia de usuario 011
Número: 011	**Usuario:** Administrador - mesero - comensal - cajero
Nombre de historia: Toma pedido restaurante rompe filas, crear un pedido	
Prioridad: Media	**Riesgo en desarrollo:** medio
Puntos estimados: 7	**Iteración asignada:** 1
Programador responsable: Julian Chavez - Ivan Quintero	
Descripción: Como usuario quiero seleccionar los productos deseados para construir un pedido	
Validación: ● Se debe agregar a la orden adiciones e ingredientes de determinados productos ● las adiciones deben verse reflejadas y poder ser seleccionadas. ● Si el producto tiene ingredientes obligatorios se debe validar que estén seleccionados correctamente ● Si un producto no tiene ingredientes ni adiciones debe agregarse directamente al pedido activo ● Al darle clic a una categoría se debe visualizar la lista de productos correspondiente. ● Se debe poder filtrar categorías y productos a partir de lo escrito por el usuario.	

En esta tabla podemos identificar la unión y una parte complementaria de la historia de usuario número 10, agregando la funcionalidad de agregar comentarios a la orden.

	Historia de usuario 012
Número: 012	**Usuario:** Administrador - mesero - cajero
Nombre de historia: Toma pedido restaurante rompe filas, agregar comentario	
Prioridad: Baja	**Riesgo en desarrollo:** Baja
Puntos estimados: 3	**Iteración asignada:** 1

Programador responsable: Ivan Quintero
Descripción: Como usuario deseo agregar sugerencias o indicaciones sobre el pedido
Validación: ● Inicialmente se debe tener al menos un (1) producto seleccionado para permitir el ingreso de un comentario. ● El comentario debe estar relacionado a la preparación del producto. ● En caso de tener una lista de productos, se puede hacer un solo comentario especificando los cambios de varios productos. ● Los comentarios deben ser coherentes de acuerdo al producto, ingredientes y adiciones especificados.

La opción de ver la cantidad de productos seleccionados en el restaurante rompe filas está descrita en la siguiente tabla:

	Historia de usuario 013
Número: 013	**Usuario:** administrador - mesero - comensal
Nombre de historia: Toma pedido restaurante rompe filas, visualizar cantidad de productos cargados	
Prioridad: Media-Alta	**Riesgo en desarrollo:** Alto
Puntos estimados: 7	**Iteración asignada:** 1
Programador responsable: Julian Chavez	
Descripción: Como usuario quiero revisar la cantidad, nombre y descripción de los productos e ingredientes que se han seleccionado y así poder agregar o eliminar cantidades.	
Validación: ● Para acceder a la vista de resumen de pedido se debe tener al menos un (1) producto cargado. ● En la vista de resumen de pedido deben aparecer 2 botones diseñados como iconos (+) (-), para aumentar o disminuir cantidades a un producto en específico.	

En el restaurante rompe filas es necesario captar el nombre del comensal y el número de comensales total que se ingresara al restaurante para asignar una mesa adecuada, en la siguiente historia de usuario se explica de una manera mucho mejor:

Historia de usuario 014	
Número: 014	**Usuario:** administrador, mesero, comensal
Nombre de historia: Toma pedido restaurante rompe filas, ingresar datos del comensal a la orden	
Prioridad: Media-Alta	**Riesgo en desarrollo:** Bajo
Puntos estimados: 7	**Iteración asignada:** 1
Programador responsable: Julian Chavez	
Descripción: Como usuario necesito asignar el nombre y número de comensales a la orden creada.	
Validación: ● Para agregar los datos de los comensales a la orden, primero se debe verificar que tenga cargado como mínimo un (1) producto. ● Deben aparecer los campos obligatorios de nombre y número de comensales. ● Los dos campos deben ser registrados, de lo contrario aparecerá un mensaje error indicando que se deben completar los campos obligatorios.	

Una de las funcionalidades más innovadoras es la asignación de la mesa dependiendo el número de comensales que son registrados por el usuario, a continuación se nombran los parámetro para que sean cumplidos los objetivos de dicha funcionalidad:

Historia de usuario 015	
Número: 015	**Usuario:** Administrador - cajero
Nombre de historia: Toma pedido restaurante rompe filas, nombre de la mesa a la cual dirigirse	
Prioridad: Media	**Riesgo en desarrollo:** Medio
Puntos estimados: 5	**Iteración asignada:** 1

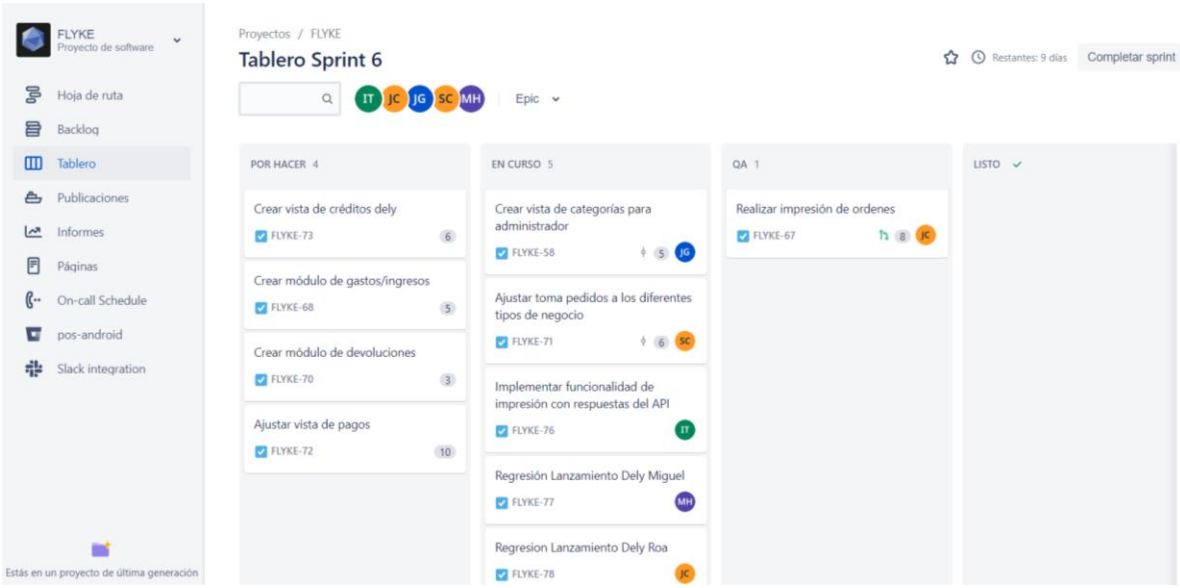
Programador responsable: Julian Chavez
Descripción: Como usuario quiero que al registrar todos los campos y requisitos de la orden me informe qué mesa es la asignada dependiendo el número de comensales.
Validación: ● Para realizar la orden se deben haber cumplido con todos los requisitos anteriormente nombrados. ● Debe existir un botón identificado como “ordenar” para qué la orden creada sea enviada al servidor y sea verificada. Si el proceso es correcto, la orden es guardada en la API y la base de datos. ● Si el proceso fue correcto en pantalla debe aparecer un mensaje de confirmación con el número de mesa asignado al comensal y sus acompañantes. ● Al esperar 30 segundos u oprimir el botón de Ok, el mensaje de confirmación debe desaparecer, dejando como nueva vista, la pantalla de menú sin ítems guardados y lista para ingresar una nueva orden. En caso de ser incorrecto aparecerá un mensaje donde se informa el tipo de error al usuario.

Resumen sprint 5 (review, retrospective, planning):

En el trabajo realizado en el sprint 5 se estableció la funcionalidad de impresión por medio de impresoras IP, ajustando flyke a impresoras térmicas de 88mm; es importante resaltar que se inició el desarrollo del módulo de facturación y toma pedidos pero debido a su magnitud no es posible su finalización, por esta razón se planea seguir con el desarrollo de estas tareas en el siguiente sprint; según el Sprint review del scrum team, se replantean más puntos de historia al no haber tenido en cuenta las funcionalidades totales de dichos módulos.

9.10 Desarrollo Sprint 6

El sprint 6 cuenta con la continuación de algunas tareas que fueron planteadas en el sprint 5, debido a que en los puntos de historia anteriormente asignados a estas tareas fueron muy bajos para el verdadero tiempo y dedicación qué se necesita, se aclara que el módulo de facturación es un módulo complejo y necesita de bastante tiempo, y por último el módulo toma pedidos falta ser configurado para restaurantes tipo Food Truck.

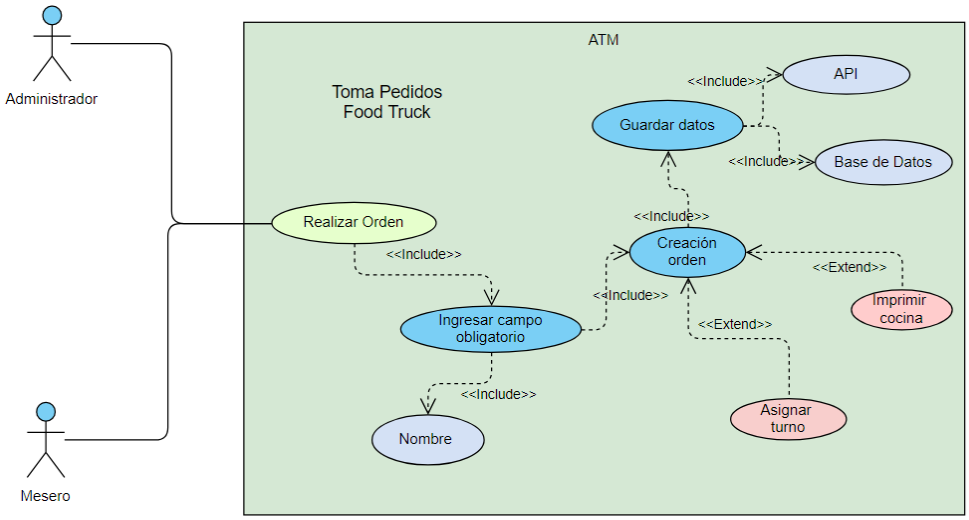


En el siguiente caso de uso nombrado “Cargar orden modulo Toma pedidos (restaurante tipo Food Truck)” se plantean las características que debe contener el modulo toma pedidos si el restaurante ingresado es de tipo Food Truck:

Información de Catalogación			
Proyecto	Flyke		
Autor	Julian Chavez Chavarro		
Versión	1	**Estado de Desarrollo**	Desarrollado
Definición del Caso de Uso			
Código	Caso de uso - 05		

Nombre	Cargar orden modulo Toma pedidos (restaurante tipo Food Truck)	
Objetivo	Cargar orden en el módulo Toma pedidos en un restaurante tipo Food Truck	
Descripción	El usuario carga ordenes desde el módulo de Toma pedidos	
Actores	Administrador, mesero y cliente	
Precondición	Seleccionar el módulo de toma pedidos	
Escenario Principal	**No.**	**Descripción de acciones**
	1	El sistema muestra la vista de menú
	2	El usuario selecciona los productos con cantidades, ingredientes y adiciones deseadas
	3	El sistema muestra el resumen de la factura
	4	El usuario agrega los datos obligatorios como lo son el nombre y número de comensales
	5	El usuario realiza la orden
	6	El sistema le asigna un número de turno
	7	El sistema almacena los datos de la orden en la API y la Base de Datos
	8	El sistema imprime la orden a cocina para su preparación
	9	Al esperar 30 segundos u oprimir el botón ok el sistema retorna al menú cómo un usuario nuevo sin ítems guardados
Escenario Alternativo	**No.**	**Descripción de acciones alternas**
	4a	El usuario no ingresa alguno de los campos obligatorios
	4b	El sistema muestra un error en pantalla “Digite campo obligatorio”
	4c	El usuario ingresa completamente los campos
	5a	El caso de uso continúa en el piso 5 del escenario principal
Escenarios de Excepción	**No.**	**Descripción de acciones de excepción**
	7a	No hay conexión a la API y no se puede almacenar la orden.
	7b	El usuario se comunicará con los desarrolladores de la aplicación para solucionar el conflicto y finaliza el caso de uso
Postcondición	El usuario realiza, imprime a cocina y agrega la orden, a su vez se le asigna una mesa automáticamente.	

El diagrama correspondiente a este caso de uso explica el flujo y los requisitos que posee la funcionalidad que va a tomar el modulo toma pedidos respecto a los restaurantes de tipo Food truck



En la historia de usuario número 15 se especifican las funciones que debe desarrollar el modulo toma pedidos dependiendo el tipo de restaurante registrado.

	Historia de usuario 016
Número: 016	**Usuario:** administrador
Nombre de historia: Seleccionar tipo de restaurante para adecuar el módulo toma pedidos.	
Prioridad: Alta	**Riesgo en desarrollo:** medio
Puntos estimados: 7	**Iteración asignada:** 2
Programador responsable: Julian Chavez	
Descripción: Como usuario quiero poder modificar el tipo de restaurante para adecuar el módulo de toma pedidos y agilizar la facturación por medio de tablets.	

Validación:
● Como parámetro principal el usuario debe loguearse, de acuerdo a la respuesta de la API, si el tipo de restaurante registrado es Food Truck, la configuración local no se puede realizar, porque ese modelo de negocio tiene por defecto una configuración asignada. ● Si el restaurante es tipo restaurante con mesas, se habilita el módulo de configuración para poder elegir. ● En el módulo de configuración aparecerá un widget de tipo SwitchPreferenceCompat con la opción "rompe filas" deshabilitada. ● Si el usuario desea trabajar con tablets en cada una de las mesas, debe dejar esa configuración por defecto (rompe filas inhabilitado). ● Si el usuario desea trabajar solo con algunas tablets y destinar un lugar específico del restaurante para que desde allí el usuario haga su pedido, debe habilitar la opción Rompe filas.

En la siguiente tabla se habla del proceso para crear un pedido en un restaurante tipo Food Truck siguiendo los parámetros establecidos por el product owner:

	Historia de usuario 017
Número: 017	**Usuario:** Administrador - cajero
Nombre de historia: Toma pedido restaurant food truck, crear un pedido	
Prioridad: Alta	**Riesgo en desarrollo:** Alto
Puntos estimados: 8	**Iteración asignada:** 2
Programador responsable: Julian Chavez, Ivan Quintero	
Descripción: Como usuario quiero seleccionar los productos deseados para construir un pedido	
Validación: ● Se debe agregar a la orden adiciones e ingredientes de determinados productos	

● las adiciones deben verse reflejadas y poder ser seleccionadas. ● Si el producto tiene ingredientes obligatorios se debe validar que estén seleccionados correctamente ● Si un producto no tiene ingredientes ni adiciones debe agregarse directamente al pedido activo ● Al darle clic a una categoría se debe visualizar la lista de productos correspondiente. Se debe poder filtrar categorías y productos a partir de lo escrito por el usuario

La opción de ver la cantidad de productos seleccionados en el restaurante Food Truck es muy importante, ya que este tipo de restaurante solo permite el ingreso de los productos una sola vez, luego le es entregado al comensal los alimentos cargados en la orden.

	Historia de usuario 018
Número: 018	**Usuario:** Administrador - mesero - cajero
Nombre de historia: Toma pedido restaurant food truck, visualizar cantidad de productos cargados	
Prioridad: Media-Alta	**Riesgo en desarrollo:** Medio
Puntos estimados: 7	**Iteración asignada:** 2
Programador responsable: Ivan Quintero	
Descripción: Como usuario quiero revisar la cantidad, nombre y descripción de los productos e ingredientes que se han seleccionado y así poder agregar o eliminar cantidades.	
Validación: ● Para acceder a la vista de resumen de pedido se debe tener al menos un (1) producto cargado. ● En la vista de resumen de pedido deben aparecer 2 botones diseñados como iconos (+), (-) para aumentar o disminuir cantidades a un producto en específico.	

En caso de disminuir las cantidades de un producto hasta llegar a 0, el producto automáticamente debe eliminarse de la orden y desaparecer de esta vista.

Es importante que el comensal ingrese su nombre para localizar su pedido y notificarle, como primera instancia el cajero dará un turno de orden cíclico para localizar e identificar el pedido, en todo caso el nombre es fundamental para localizar el comensal en caso de perder su número de turno.

Historia de usuario 019	
Número: 019	**Usuario:** Administrador - cajero
Nombre de historia: Toma pedido restaurant food truck, Ingresar datos del cliente a la orden	
Prioridad: Baja	**Riesgo en desarrollo:** Bajo
Puntos estimados: 4	**Iteración asignada:** 2
Programador responsable: Julian Chavez	
Descripción: Como usuario necesito asignar el nombre del comensal a la orden creada.	
Validación: ● Para agregar el nombre del comensal a la orden, primero se debe verificar que tenga cargado como mínimo un (1) producto. ● Debe aparecer el campo obligatorio nombre de comensal. El campo debe ser registrado, de lo contrario aparecerá un mensaje error indicando que se debe completar el campo obligatorio.	

La siguiente historia de usuario nombra los criterios de aceptación y la función que cumple la asignación de turnos en el restaurante tipo Food Truck.

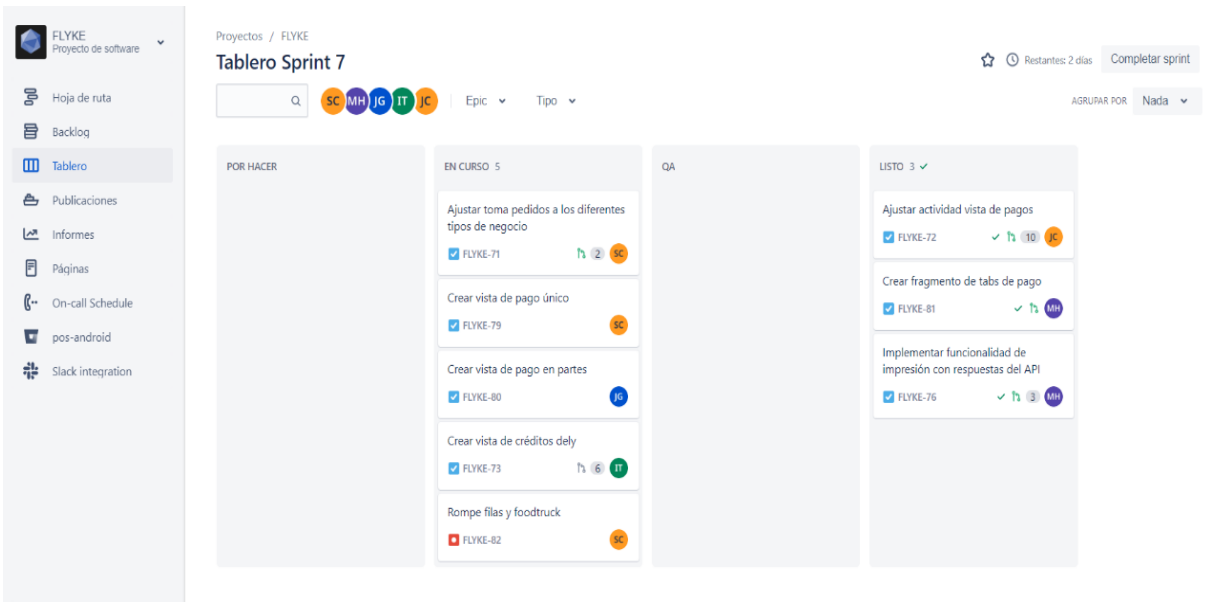
Historia de usuario 020	
Número: 020	**Usuario:** Administrador - cajero
Nombre de historia: Toma pedido restaurant food truck, número de turno asignado	
Prioridad: Baja - media	**Riesgo en desarrollo:** Media
Puntos estimados: 5	**Iteración asignada:** 2
Programador responsable: Julian Chavez	
Descripción: Como usuario quiero que al registrar todos los campos y requisitos de la orden me informe qué número de turno es el asignado al comensal	
Validación: ● Para realizar la orden se deben haber cumplido con todos los requisitos anteriormente nombrados. ● Debe existir un botón identificado como "ordenar" para qué la orden creada sea enviada al servidor y sea verificada. ● Si el proceso es correcto, la orden es guardada en la API y la base de datos. ● Si el proceso fue correcto en pantalla debe aparecer un mensaje de confirmación con el número de turno asignado al comensal. ● Al esperar 30 segundos u oprimir el botón de Ok, el mensaje de confirmación debe desaparecer, dejando como nueva vista, la pantalla de menú sin ítems guardados y lista para ingresar una nueva orden. En caso de ser incorrecto aparecerá un mensaje donde se informa el tipo de error al usuario.	

Resumen sprint 6 (review, retrospective, planning):

En el sprint 6 se finalizaron dos tareas propuestas en el sprint 5 debido a la magnitud de los módulos (facturación y toma pedidos), también se creó el módulo de ajustes con el objetivo de flexibilizar el modulo toma pedidos de acuerdo al tipo de restaurante registrado. Otra situación que es pertinente nombrar es el inicio de la emergencia sanitaria Covid-19 lo cual afectó el desplazamiento de los pasantes a la empresa y se tomó como medida el trabajo remoto para cada estudiante.

9.11 Desarrollo Sprint 7

En este sprint ocurrieron algunos inconvenientes debido a la emergencia sanitaria Covid-19, lo que ocasionó que se aplaza por un tiempo el sprint planning de este último sprint, además sucedieron inconvenientes con la empresa por cuestiones de disposición y tiempo de su parte; notificamos a la universidad la situación y seguimos el debido proceso. Gracias a la colaboración y acompañamiento de la universidad se logró llegar a un acuerdo con la empresa Ice Company sas para plantear las tareas finales que consolidaron un producto mínimo viable de la aplicación "Flyke". Por las circunstancias nombradas el sprint 7 tuvo un tiempo más corto en comparación a los demás sprints.

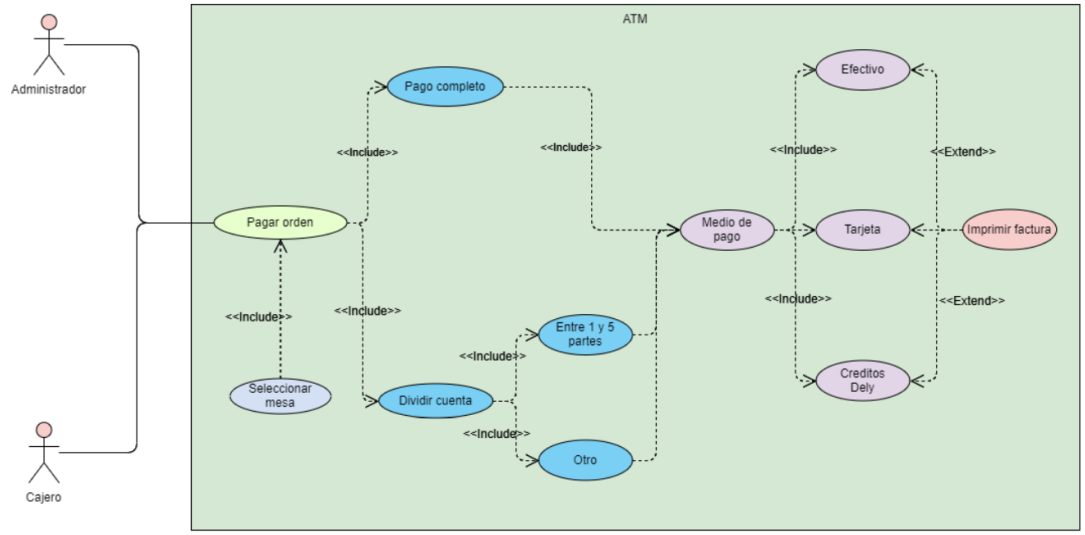


A continuación se detalla el caso de uso que representa las tareas correspondientes al pago de órdenes, donde participa la creación de la vista, funcionalidad de pagos completos, pagos en partes y pagos por medio de créditos Dely:

Información de Catalogación			
Proyecto	Flyke		
Autor	Ivan Quintero Torres – Julian Chavez Chavarro		
Versión	1,0	Estado de Desarrollo	Desarrollado
Definición del Caso de Uso			
Código	Caso de uso - 06		
Nombre	Pago de orden		
Objetivo	Pagar la orden realizada		
Descripción	El administrador o cajero realiza el proceso de pago de una orden		
Actores	Administrador - Cajero		
Precondición	Seleccionar la mesa que contiene cargada la orden a pagar. Seleccionar la opción de pago		
Escenario Principal	**No.**	**Descripción de acciones**	
	1	El administrador / cajero selecciona la opción pago de orden	
	2	El sistema carga la vista de pago, donde muestra el total de la orden (compuesta por el subtotal, cobro de servicio e impuestos) y las opciones de pago (pago completo y dividir pago).	
	3	El administrador selecciona la opción de pago	
	4	Se imprime la orden pagada y termina el caso de uso	

Escenario Alternativo	**No.**	**Descripción de acciones alternas**
	3a	Opción de pago completo
	3b	Opción de dividir pago
	3c	El caso de uso continúa en el ítem 4 del escenario principal
Escenarios de Excepción	**No.**	**Descripción de acciones de excepción**
	2a	El sistema no tiene conexión al servidor o a la DB
	2b	El sistema lo notifica con un mensaje de excepción
	2c	El usuario debe notificar a los administradores de la aplicación (soporte) y Termina el caso de uso
Postcondición	La orden es pagada.	

El siguiente diagrama representa cómo acceden los actores principales de la aplicación a esta sección y que otros procesos extienden de “Pago completo”, “Dividir cuenta”, tenga en cuenta que este caso de uso puede cambiar al momento de añadir más opciones que parezcan convenientes para la empresa:



La siguiente tabla hace relación a la historia de usuario número 20, que hace referencia a la construcción de la interfaz y la implementación de pago de la orden con la opción de pago completo, resaltando el valor total y las posibles opciones de pago, con los medios de pago aceptados por el restaurante (Efectivo, tarjeta o créditos Dely) :

	Historia de usuario 021
Número: 021	**Usuario:** Administrador - cajero.
Nombre de historia: Pago de orden, opción pago completo.	
Prioridad: Media	**Riesgo en desarrollo:** Alto
Puntos estimados: 6	**Iteración asignada:** 1
Programador responsable: Ivan Quintero.	
Descripción: Como usuario deseo pagar la orden en un solo pago	
Validación: ● El sistema debe mostrar el valor total de la orden (compuesta por el subtotal, cobro del servicio e impuestos) ● El sistema debe mostrar los medios de pago (efectivo, tarjeta o créditos Dely) brindando en la opción de efectivo 4 opciones rápidas de pago. ● Debe existir una opción para ingresar otro monto a pagar por el comensal (number pad).	

● El botón de pago se debe habilitar una vez que se seleccione un medio de pago.

La siguiente tabla hace referencia a la historia de usuario número 21, que hace relación a la construcción de la interfaz y la implementación de pago de la orden con la opción de dividir cuenta, brindando al comensal pagar la orden en distintas partes:

Historia de usuario 022	
Número: 022	**Usuario:** Cajero
Nombre de historia: Pago de orden, opción dividir cuenta.	
Prioridad: Media	**Riesgo en desarrollo:** Alto
Puntos estimados: 7	**Iteración asignada:** 1
Programador responsable: Julian Chavez	
Descripción: Como usuario deseo dividir la orden en varios pagos	
Validación: ● El sistema debe mostrar el valor total de la orden (compuesta por el subtotal, cobro del servicio e impuestos) ● El sistema debe mostrar las opciones de pago completo junto con las opciones de dividir cuenta, se debe permitir dividir la cuenta hasta en 5 partes. ● Debe existir una opción para ingresar otro monto a pagar por el comensal (number pad). ● Al elegir una de las opciones de dividir pago y oprimir el botón pagar, el sistema debe actualizar la vista y mostrar los medios de pago con el nuevo total del primer pago. ● Se debe iterar cada pago hasta que el sistema no tenga ningún valor por cobrar.	

La siguiente tabla hace relación a la historia de usuario número 22, que hace énfasis a la construcción de la interfaz y la implementación de pago de la orden con la opción de créditos Dely :

Historia de usuario 023	
Número: 023	**Usuario:** Administrador - cajero
Nombre de historia: Pagar una orden por medio de créditos Dely	
Prioridad: Media - Alta	**Riesgo en desarrollo:** Alto
Puntos estimados: 7	**Iteración asignada:** 1
Programador responsable: Ivan Quintero	
Descripción: Como usuario deseo pagar la orden por medio de mis créditos Dely.	
Validación: ● El sistema debe mostrar el total de la orden (compuesta por el subtotal, cobro de servicio e impuestos) y la opción de pago con créditos Dely. ● El sistema debe mostrar un código QR con el total de la orden a pagar y el código alterno del QR e imprimir la orden pagada.	

Resumen sprint 7 (review, retrospective, planning):

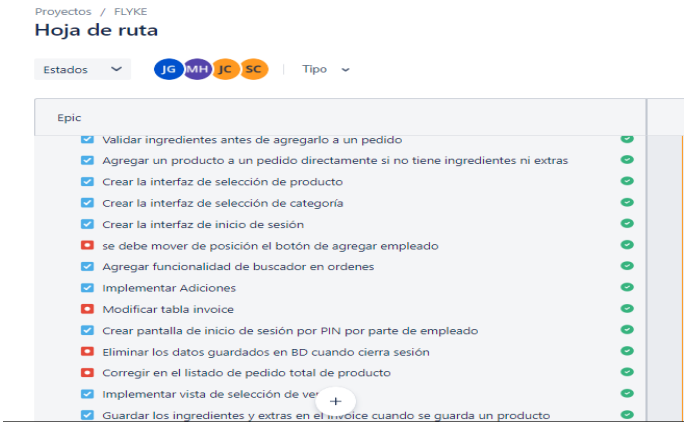
En este sprint final se consolidó el PMV (Producto mínimo viable) establecido por la empresa Ice Company sas. La empresa realizó la revisión exhaustiva de todo el desarrollo en busca de mejoras y cambios para lo creado; se establecieron algunos detalles respecto a las opciones de pago, con el fin de perfeccionar la experiencia de usuario y el flujo de los pagos de cada orden. Por último los pasantes hicieron los respectivos cambios, notificando al scrum master y al Product owner, por consiguiente Edward Hernandez Molano (Product Owner) aprueba el trabajo realizado por los estudiantes y se da fin a la pasantía.

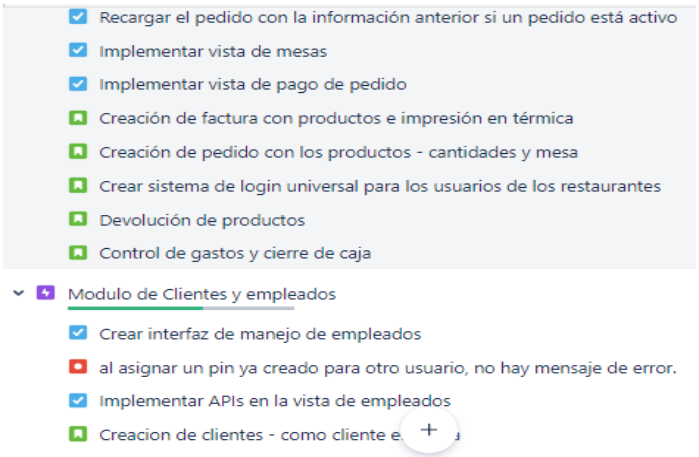
9.12 Product backlog

Es el inventario donde se almacena todo lo que podría ser necesario en el producto, se ve reflejado como una lista de necesidades del cliente qué son priorizadas por el product owner con el fin de estimar los requerimientos del proyecto. Un product backlog nunca está completo, está

abierto a la adición de nuevas tareas a medida que los incrementos son entregados y puestos en producción.

En el proyecto se utilizó la herramienta Jira software para administrar el product backlog, en dicha herramienta se le conoce como “hoja de ruta”.



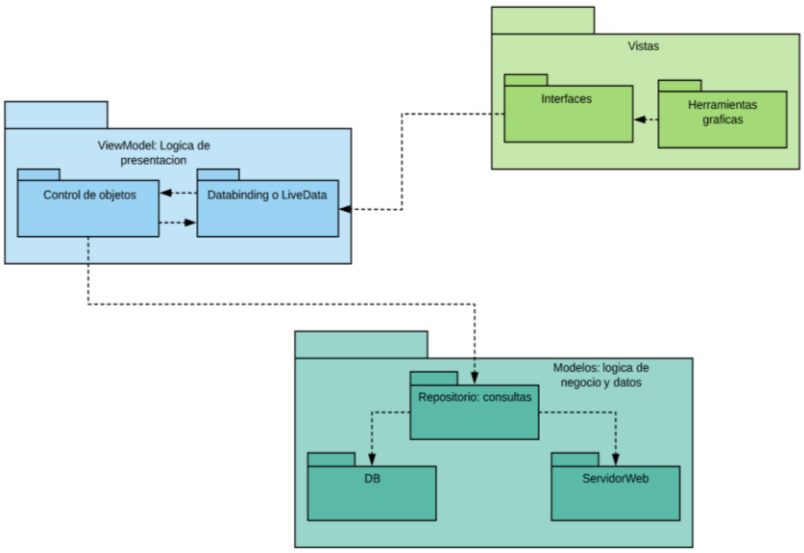


10. Diagramas

Anteriormente Se aclaró que la metodología principal del proyecto fue scrum, pero por motivos de confidencialidad relacionados con la empresa Ice Company sas no es posible compartir o mostrar interfaces, funcionalidades o código fuente relacionado al proyecto Flyke, por ello se decidió apoyarse en metodologías tradicionales para dar una orientación más acertada de todo el proyecto desarrollado durante la pasantía. Los siguientes diagramas hacen parte de Lenguaje Unificado de Modelado (UML), su principal función es demostrar, especificar y documentar el funcionamiento de la aplicación desarrollada:

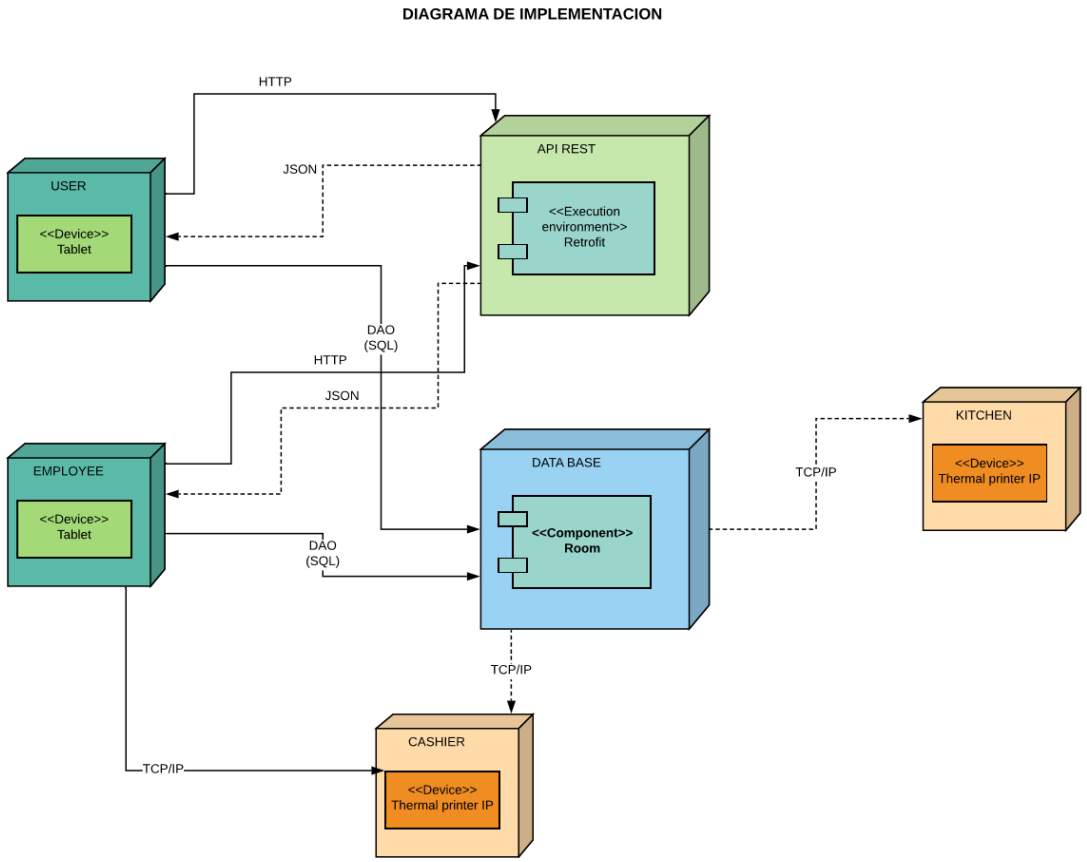
10.1 Diagrama de paquetes

El diagrama de paquetes provee una representación gráfica de la arquitectura qué es manejada en cada proyecto de desarrollo. En Flyke se utilizó la arquitectura MVVM la cual estipula una comunicación directa de cada paquete pero definiendo una responsabilidad única a cada componente, es decir separa la vista(interfaces) de la lógica de negocio(ViewModel, Live Data) y la lógica de los datos(Modelos, Repositorios, Servicios):



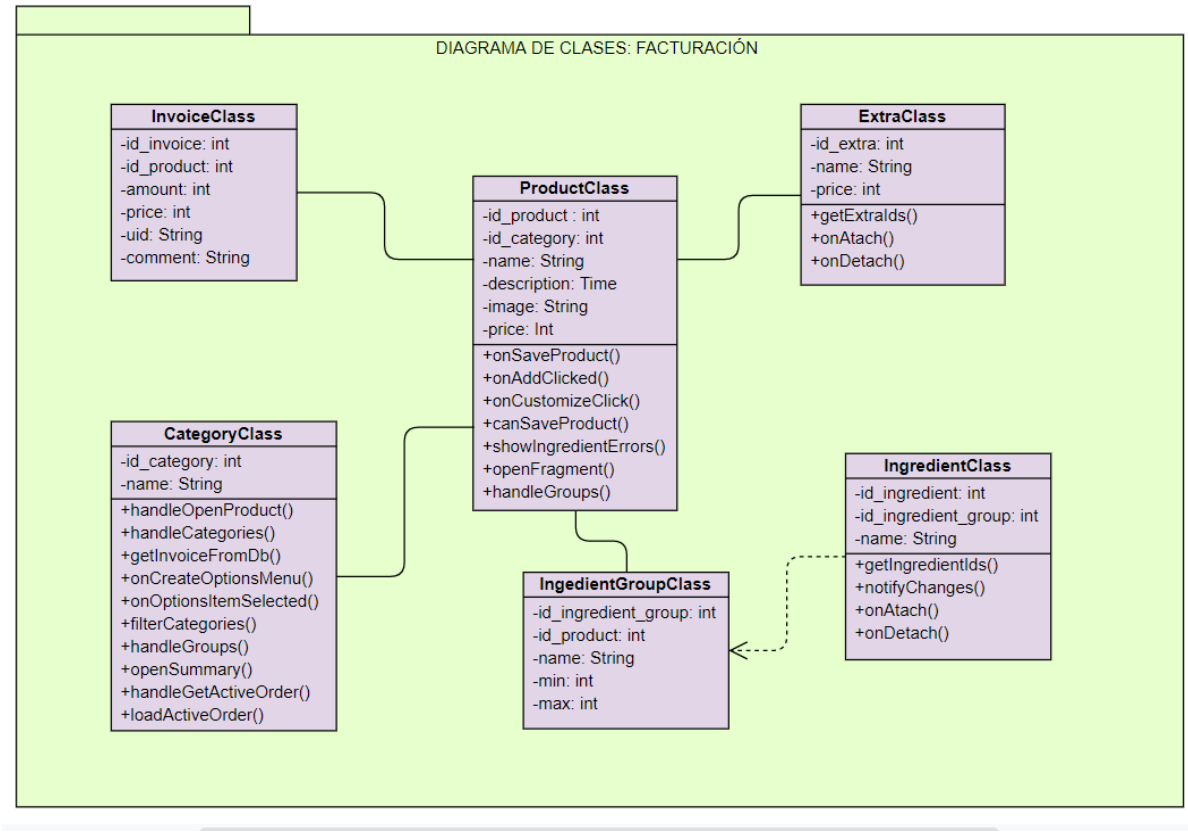
10.2 Diagrama de implementación

En este diagrama se representa toda la arquitectura del sistema, incluyendo hardware, software y demás componentes, en este caso podemos observar que la alimentación de los datos se da mediante una API rest (web service) y una DB (base de datos local) y también se reconocen los dispositivos físicos que intervienen en el flujo de datos de la aplicación como lo son las tablets y las impresoras térmicas:



10.3 Diagrama de clases

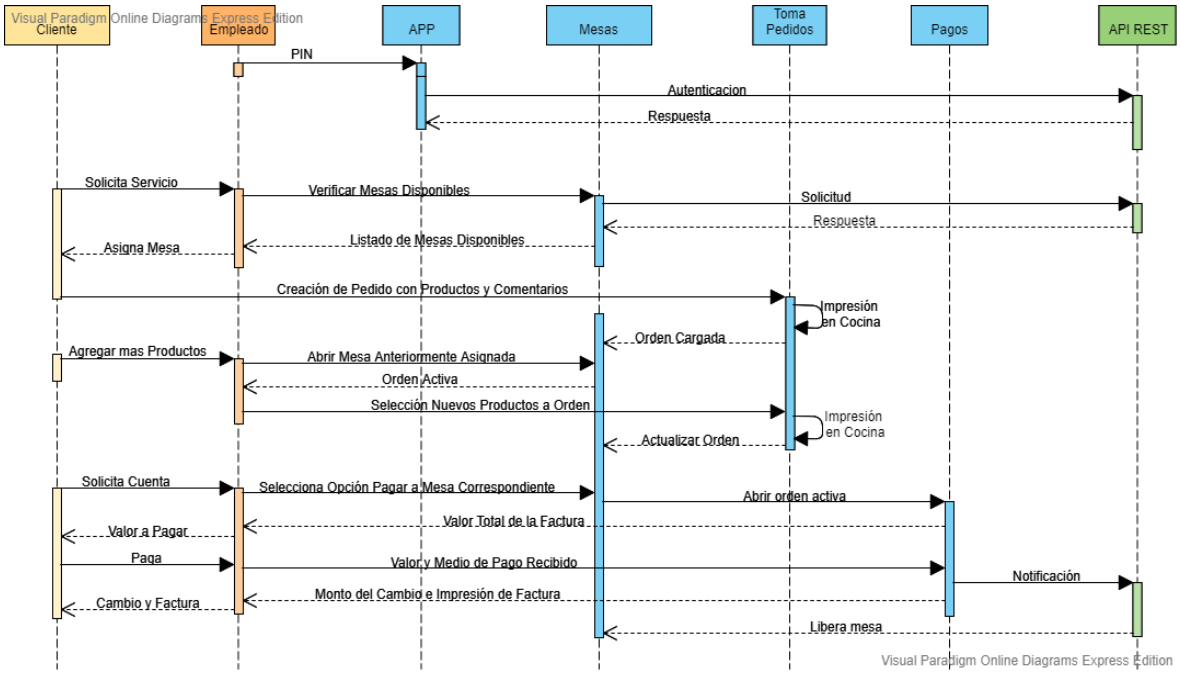
En el diagrama de clases podremos visualizar todas las clases construidas durante el proyecto, con sus respectivos parámetros o propiedades, métodos y claramente las relaciones entre sí, se grafican las clases que constituyen el módulo de facturación, el cual representa una parte fundamental de la aplicación:



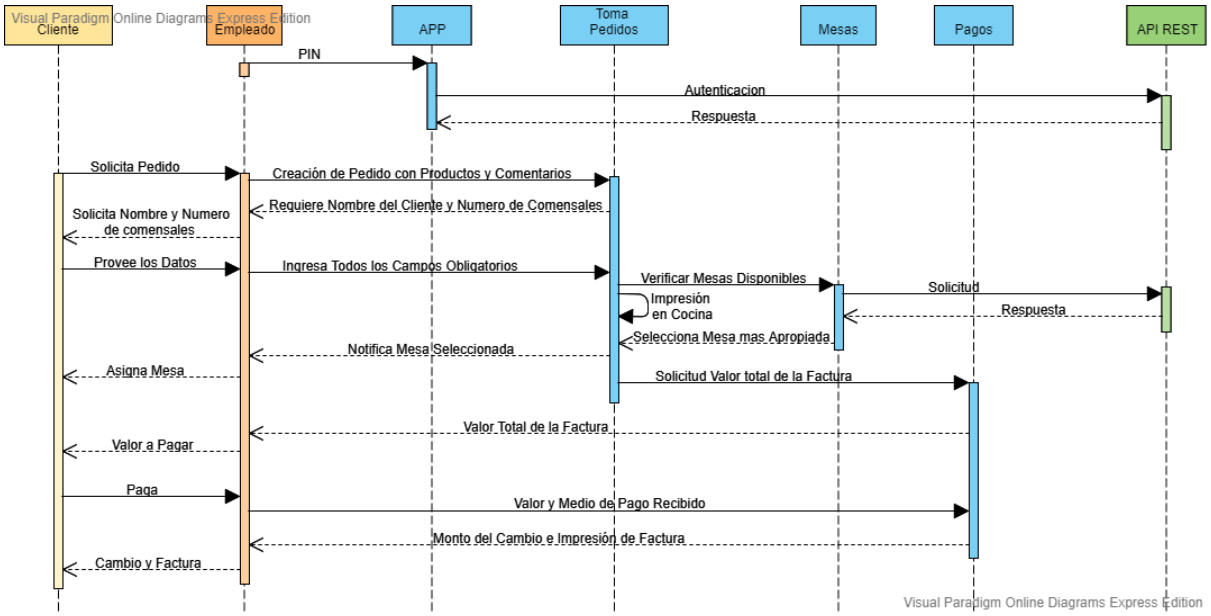
10.4 Diagrama de secuencia

El diagrama de secuencia ilustra la relación que tienen los objetos entre sí y cómo es su comunicación en un ambiente cotidiano, los objetos son nombrados dentro de las cajas principales (cliente, empleado, App, Mesas, Toma pedidos, Pagos y Api rest) y sus relaciones son manifestadas a través de cajas más delgadas unidas por medio de líneas continuas (peticiones o envío de información) y líneas punteadas (respuestas o acciones en segundo plano).

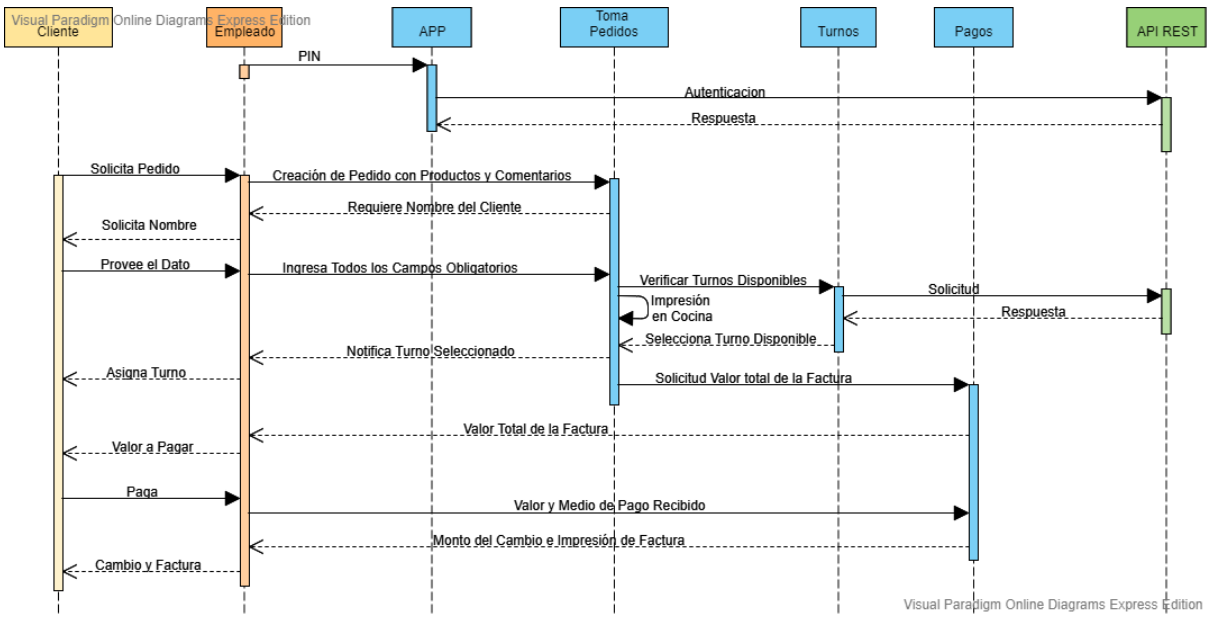
Módulo Facturación y toma pedidos para Restaurante con Mesas:



Modulo toma pedidos para Restaurante tipo Rompe Filas:

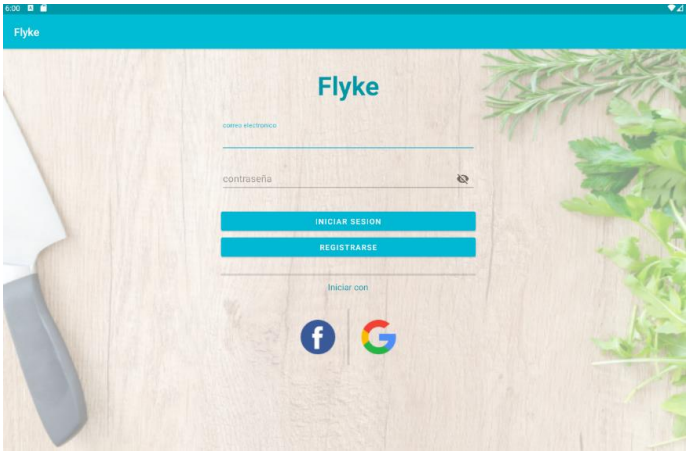


Modulo toma pedidos para Restaurante tipo Food Truck:

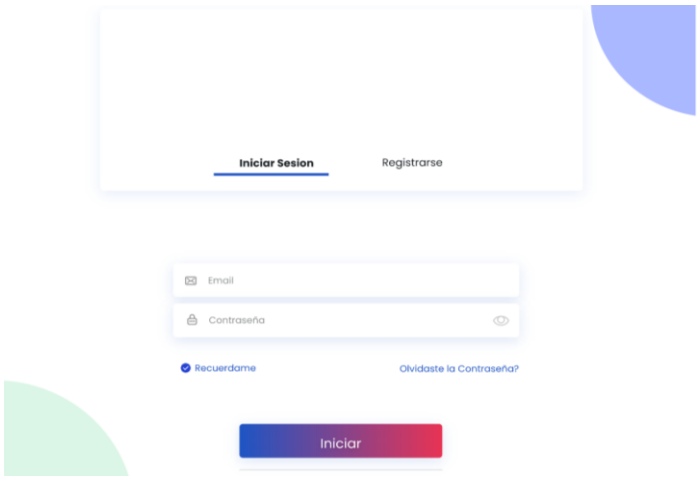


11. Cambios en el desarrollo

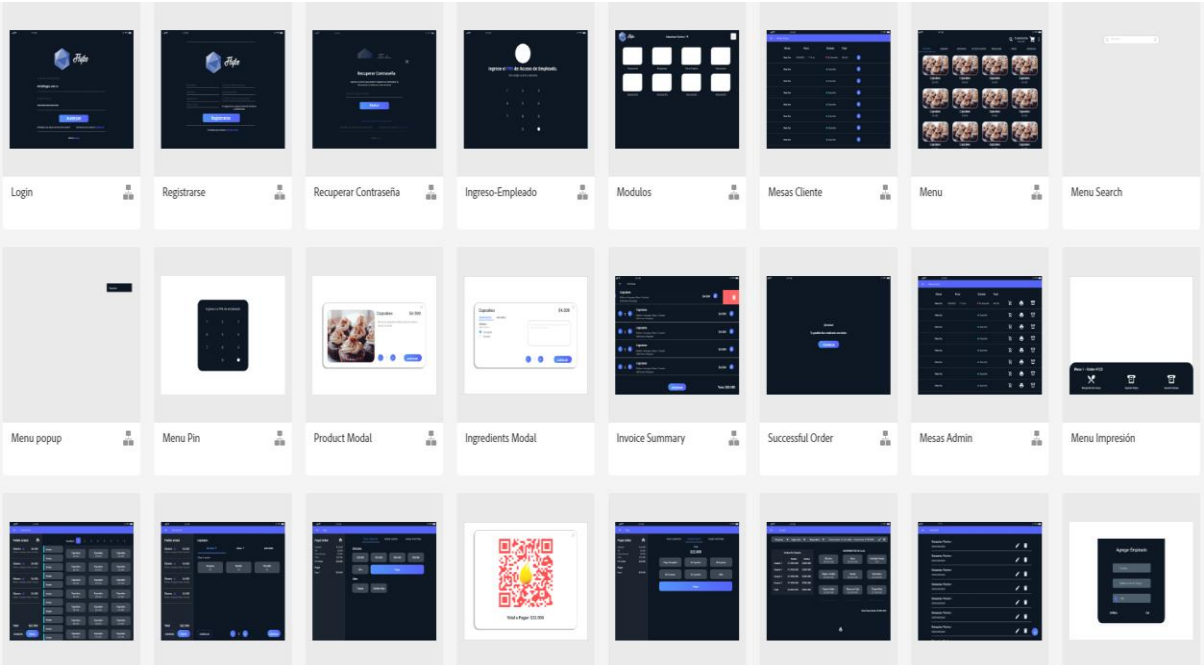
Durante el desarrollo existieron muchos cambios como por ejemplo el cambio en las interfaces, estas fueron construidas con la herramienta de Adobe XD, dichas interfaces al principio del proyecto eran planteadas por los pasantes, luego la empresa decidió que construía los prototipos de las interfaces, tal cual como las querían, para que el developer team las construyeran.Un ejemplo claro es la interfaz de inicio de sesión la cual al ser la interfaz inicial para de todo el proyecto, sufrió grandes cambios mientras la empresa buscaba el tema más adecuado esta interfaz fue la primera que se construyó:



Luego se construyó este otro tipo de interfaz, con un toque más personalizado y limpio pero que aún seguía sin convencer a la empresa para ser el tema principal del aplicativo:



Finalmente se decidió por trabajar primeramente con un tema oscuro, más personalizado, agradable a la vista del usuario, con detalles como la implementación del logo de Flyke, el cual es una figura en una gama de colores azul llamada dodecaedro, a continuación el esquema de todas las interfaces de la aplicación:



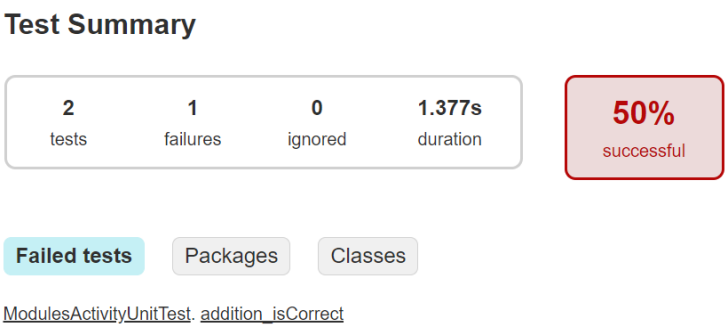
12. Pruebas unitarias

Debido al contrato de confidencialidad no es permitido evidenciar las pruebas unitarias con imágenes del código fuente, por esta razón se adjuntan los informes de pruebas unitarias proporcionados por el entorno de desarrollo Android Studio, dicho informe se genera después de ejecutar el código de testing implementado por el developer team, cabe destacar para una mayor comprension que la palabra clave: Activity representa el archivo que contiene todo el código relacionado a uno o más módulos.

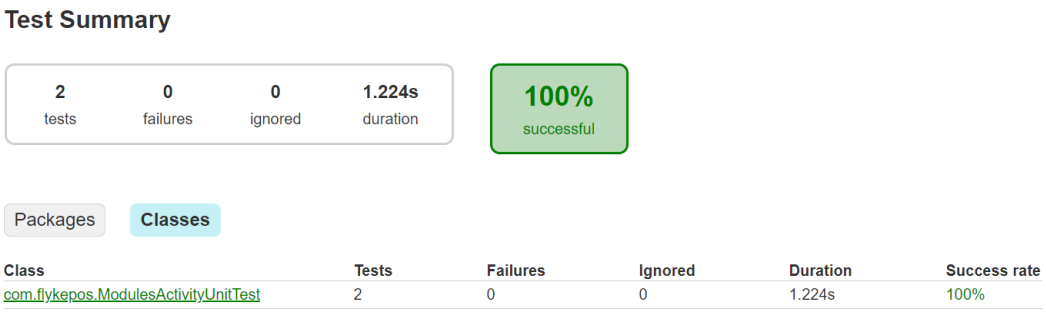
12.1 Prueba a clase de módulos (Modules Activity)

Para la prueba del activity de módulos(Modules activity) se diseñó primeramente un método que construyera un listado de módulos aleatoriamente, para que el método que construía los módulos lo reconociera y lanzará un resultado.

Para el primer resultado fallo en un 50% puesto que el método no contaba con un código escalable para construir más módulos:



Entonces al arreglar este inconveniente, el test arroja un resultado positivo en un 100% construyendo de manera efectiva los módulos enviados desde el método de la prueba unitaria:



Test Summary

2 tests	0 failures	0 ignored	1.224s duration

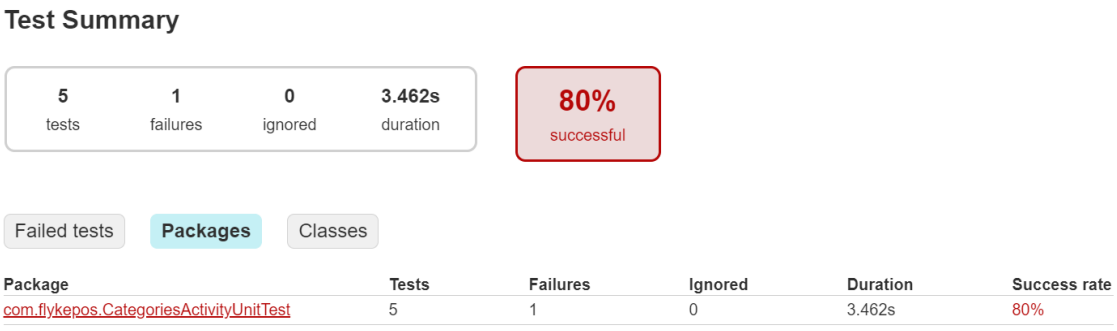
100% successful

Packages | Classes

Class	Tests	Failures	Ignored	Duration	Success rate
com.flykepos.ModulesActivityUnitTest	2	0	0	1.224s	100%

12.2 Prueba a clase de Categorías (Categories Activity)

En categories activity se realizaron 5 pruebas, las cuales estaban orientadas principalmente a: abstracción de categorías desde el API, posicionarlas, recuperar los productos pertenecientes a esta categoría, entre otros; como podemos observar el primer test tuvo una calificación del 80% ya que al momento de posicionar estas categorías, habian algunos paquetes de datos de tipo null:



Test Summary

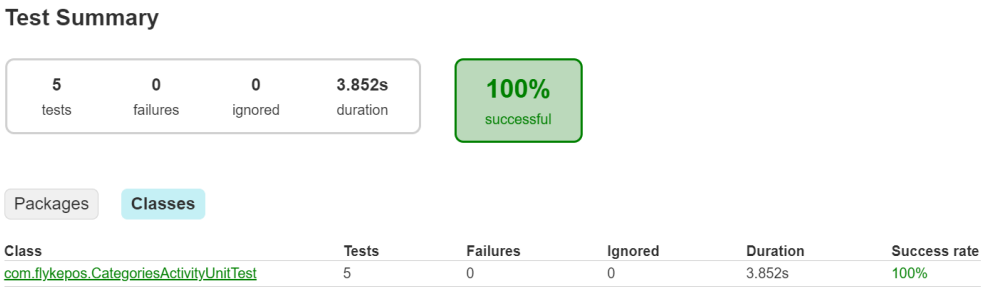
5 tests	1 failures	0 ignored	3.462s duration

80% successful

Failed tests | Packages | Classes

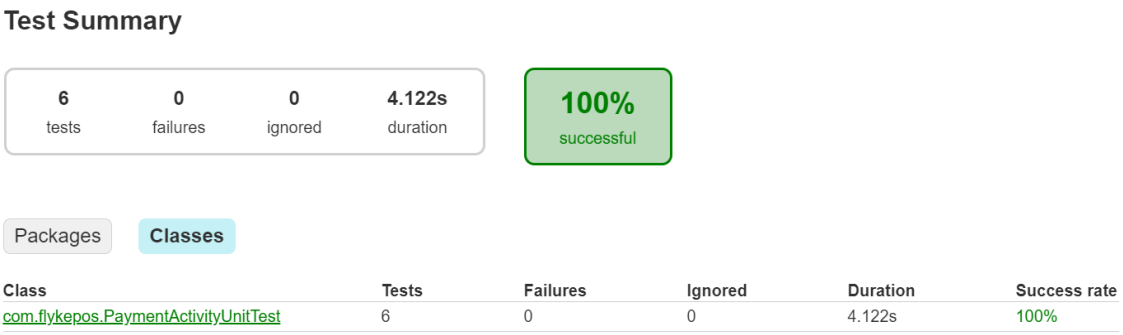
Package	Tests	Failures	Ignored	Duration	Success rate
com.flykepos.CategoriesActivityUnitTest	5	1	0	3.462s	80%

Corrigiendo el método que abstrae las categorías y las posiciona en su respectiva interfaz, para evitar que reciba información nula, obtenemos un resultado del test del 100%, que podemos ver en el siguiente gráfico:



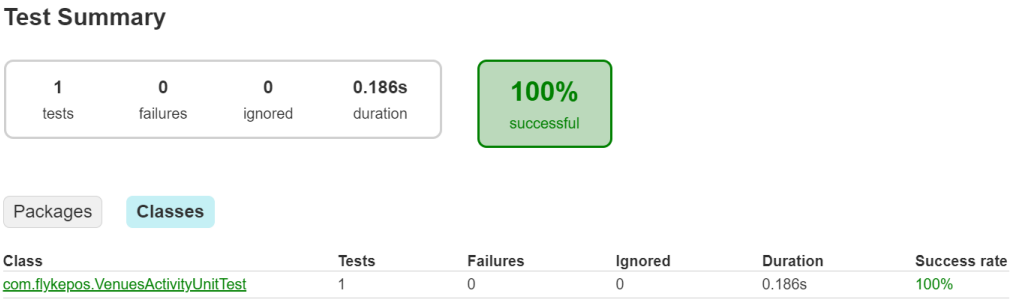
12.3 Prueba a clase de Pago (Payment Activity)

En las seis pruebas que se realizaron en el activity de pagos, se obtuvo un resultado favorable, ya que la implementación de los métodos para recuperar los tipos de pago y plasmarlos en la interfaz funcionaron correctamente:



12.4 Prueba a clase de Qr (Qr Payment Fragment)

La prueba al fragmento que contiene la lógica del código QR era crucial realizarla, puesto que esta representa la integración con la aplicación de domicilios: Dely. La prueba arrojó un resultado del 100% puesto que la aplicación graficaba correctamente el código QR correspondiente al ID de la orden actual para poder realizar el pago con créditos Dely:



Test Summary

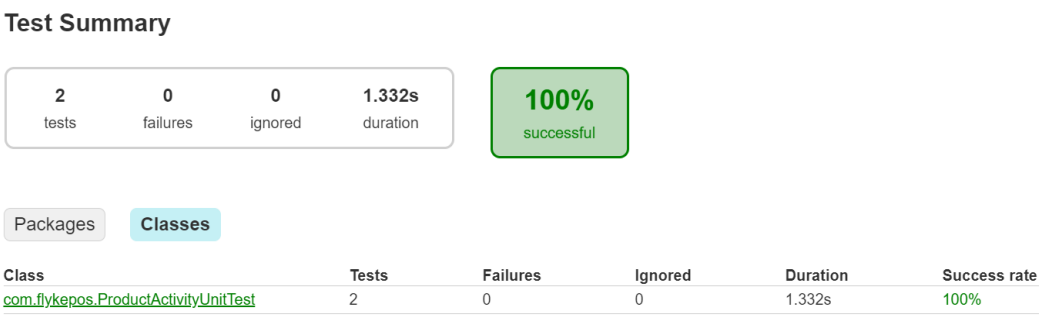
1 tests	0 failures	0 ignored	0.186s duration	100% successful

Packages | Classes

Class	Tests	Failures	Ignored	Duration	Success rate
com.flykepos.VenuesActivityUnitTest	1	0	0	0.186s	100%

12.5 Prueba a clase de Producto (Product Activity)

Por último pero no menos importante, se realiza un par de pruebas al activity que contiene el código fuente del producto, en la que se puede validar parámetros tales como, imagen del producto no nula, nombre y descripción correctos; finalmente validar si el producto si estaba en la categoría correcta:



Test Summary

2 tests	0 failures	0 ignored	1.332s duration	100% successful

Packages | Classes

Class	Tests	Failures	Ignored	Duration	Success rate
com.flykepos.ProductActivityUnitTest	2	0	0	1.332s	100%

13. Conclusiones

La utilización de herramientas tecnológicas durante el proceso de desarrollo de software representa un apoyo complementario para mitigar los posibles errores o fallas durante todas las fases del desarrollo, por ejemplo, herramientas como lucidchart que nos permiten la construcción de la base de datos mediante esquemas de tipo MER.

El módulo de facturación se creó especialmente para tener un mayor control de las ventas, manejo de clientes y órdenes pendientes, brindando al cajero o administrador del restaurante una optimización de los procesos de pago e impresión a través de esta herramienta tecnológica (Flyke).

Flyke es una herramienta escalable que permite la flexibilidad de ciertas funcionalidades dependiendo el capital del cliente o el tamaño de su restaurante, gracias al modulo “Toma pedidos” puede adaptarse a los principales modelos de negocio y facilitar la creación de las órdenes.

Podemos concluir también que en todo negocio es fundamental tener la facultad de gestionar procesos administrativos, por ejemplo en el caso de flyke, el módulo de gestión de empleados se convierte en un herramienta muy útil en la que se visualiza un listado de los empleados del negocio, con sus respectivos roles, nombres y claves de seguridad, para que el administrador pueda crear, modificar o eliminar cualquier funcionario de la lista.

El mundo ha sido afectado por cambios drásticos gracias a la industria tecnológica, además se ha reflejado que las compañías cada vez están migrando su filosofía y su visión corporativa a

desarrollar un modelo de trabajo donde la innovación y la recursividad sean la clave para diseñar un camino que logre ser escalable y mantenible para un futuro cambiante, en el caso de los restaurantes pudimos observar que el tema de la facturación puede variar dependiendo del tipo de restaurante, pero que la opción de imprimir a diferentes zonas si es una importante funcionalidad que añade un plus a la agilización de procesos.

Es sumamente importante implementar una arquitectura de software, en el desarrollo se utilizó MVVM que permite un código mantenible, escalable, limpio y además segmenta la arquitectura, este último permite que en el caso de que falle algo, no deje de funcionar toda la aplicación sino solo en el segmento del fallo y de esta manera realizar pruebas unitarias con mayor facilidad.

En el desarrollo de Software es importante brindar gran importancia al análisis de requerimientos ya que estos definen el rumbo del desarrollo, si se hace un mal análisis de los requerimientos el desarrollo estará sujeto a muchos cambios.

Es indispensable el correcto trabajo en equipo, durante el desarrollo de la aplicación la metodología (Scrum) nos permitió tener una buena comunicación y coordinación debido a herramientas como el daily y el sprint planning

14. Referencias

[1] SA. Atlassian | Software Development and Collaboration Tools. de Jira Software Sitio web: https://www.atlassian.com/

[2] SA. El motor de tu trabajo. de Slack sitio web : https://slack.com/intl/es-co/

[3] Kruchten, P. (1995). Planos Arquitectónicos: El Modelo de 4+ 1 Vistas de la Arquitectura del Software. IEEE Software, 12(6), 42-50.

[4] Fuentes, J. R. L. (2015). Desarrollo de Software ÁGIL: Extreme Programming y Scrum. IT Campus Academy.

[5] SA. Bitbucket | The Git solution for professional teams. Mayo 2, 2020, Sitio web: https://bitbucket.org

[6] SA. Agiliza el desarrollo de servicios REST con Postman. Abril 12, 2020 Sitio web: https://metadrop.net/articulos/agiliza-desarrollo-servicios-rest-postman

[7] Weitzenfeld, A. (2005). Ingeniería de software orientada a objetos con UML, Java e Internet. Thomson,.

[8] Developer, A. (2015). Android Developer. línea. Sitio web: https://developer. android.com

[9] Txema Rodríguez. (19 Mayo 2017). kotlin ya es un lenguaje oficial de android. 08 Mayo 2020, de xakata android Sitio web: https://www.xatakandroid.com/programacion-android/kotlin

[10] (Joel Francia, 2017), ¿Qué es Scrum?. 27/05/2020, scrum.org Sitio web: https://www.scrum.org/resources/blog/que-es-scrum)