



## Proyecto Final

Universidad del Valle

Ingeniería en Sistemas

Desarrollo de Software I

### Integrantes:

Alejandro Sierra Betancourt - 2259559

Juan Pablo Castaño Arango - 2259487

Juan Manuel Ramírez Agudelo - 2259482

David Santiago Velasco Triana - 2259479

Cristian David Rivera Torres - 2259742

William Alexander Franco Otero - 2259715

## **Introducción a la problemática:**

En este documento, se abordará la problemática que le surge a un restaurante, el cual presenta tardanza en la toma de pedidos, causada por factores como congestión, falta de personal y complejidad en el proceso de toma de pedidos.

## **Objetivos generales:**

### **1. Comprender y aplicar los diferentes modelos de ciclo de vida del desarrollo de software en un contexto práctico.**

Se ha optado por la implementación de Kanban en el proyecto, pues este ayuda no solo a la gestión del flujo de manera correcta, sino que también permite adaptarse fácilmente a los cambios.

### **Principios de Kanban Aplicados:**

#### **1. Visualizar el Trabajo:**

- El tablero Kanban será la herramienta principal para visualizar todas las tareas y su estado actual. Esto ayuda a todos los miembros del equipo a tener una visión clara del progreso del proyecto.

#### **2. Limitar el Trabajo en Progreso (WIP):**

- Establecer límites en la cantidad de trabajo que puede estar en progreso en cualquier momento para evitar sobrecarga y mejorar el enfoque del equipo.

#### **3. Gestionar el Flujo de Trabajo:**

- Monitorizar y gestionar el flujo de trabajo para identificar y eliminar cuellos de botella y asegurar un progreso constante.

#### **4. Hacer las Políticas Explícitas:**

- Definir y acordar reglas claras para cómo se mueve el trabajo a través del tablero y cómo se gestionan las tareas.

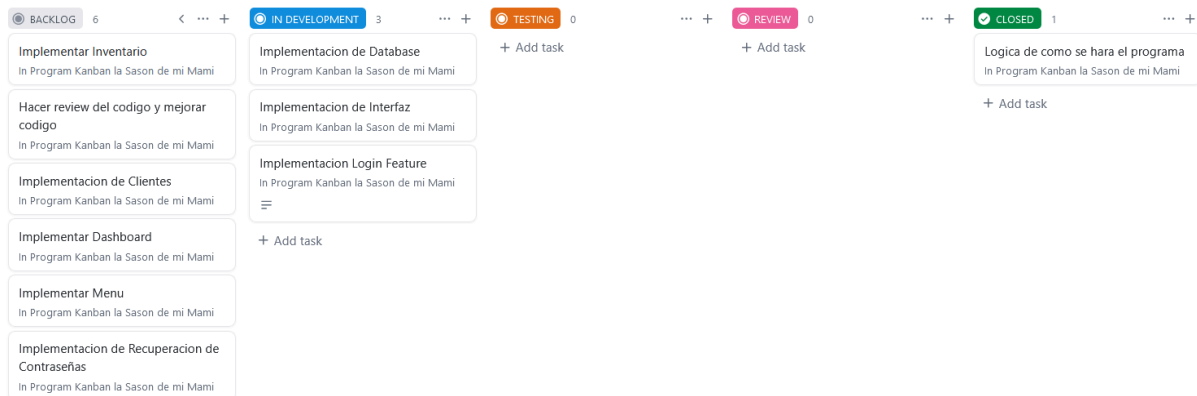
## 5. Implementar Circuitos de Feedback:

- Establecer revisiones regulares y retrospectivas para discutir el progreso, recibir feedback y mejorar continuamente los procesos.

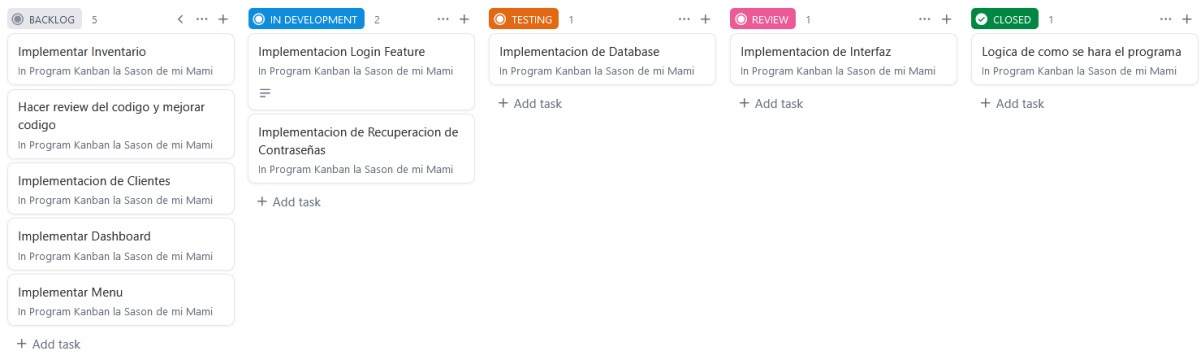
## 6. Mejorar Colaborativamente y Evolucionar Experimentalmente:

- Fomentar una cultura de mejora continua donde el equipo colabore para experimentar con nuevas ideas y prácticas.

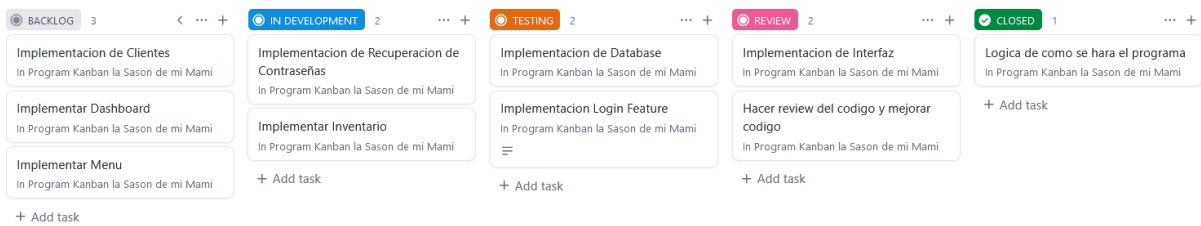
## Semana 1



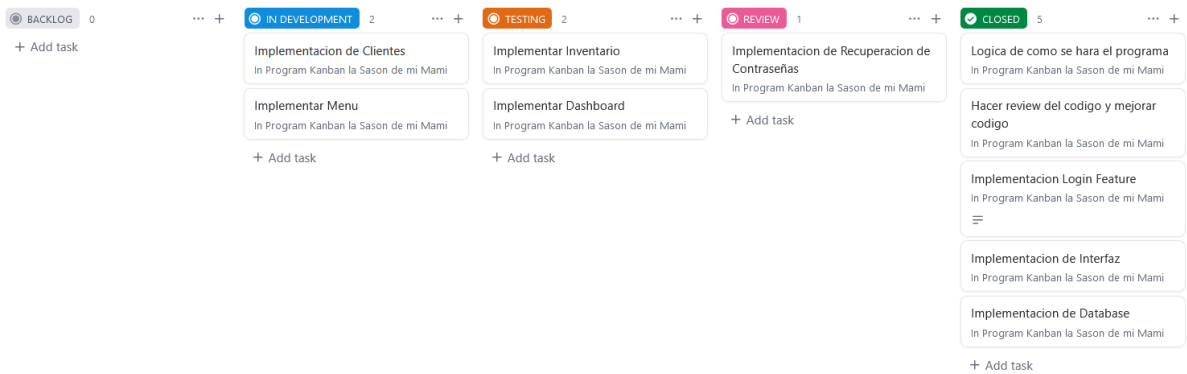
## Semana 2



## Semana 3



## Semana 4



**2. Utilizar eficazmente el enfoque orientado a objetos para diseñar y desarrollar software.**

La aplicación fue diseñada utilizando principios de orientación a objetos (OO), lo que facilita la modularidad, reutilización y mantenimiento del código. La estructura OO será clave en la organización del software, permitiendo representar entidades como pedidos, menús, clientes y mesas de manera clara y manejable.

### **3. Desarrollar habilidades en el uso de UML para el análisis y diseño de Sistemas de software.**

Utilizamos UML (Unified Modeling Language) para crear los diagramas necesarios que ayudarán a visualizar y documentar el diseño del sistema de una forma mucho más sencilla.

### **4. Implementar la metodología RUP en todas las fases del desarrollo de un proyecto de software.**

La metodología Rational Unified Process (RUP) es un marco de desarrollo de software iterativo e incremental que se centra en una entrega de alta calidad a través de fases bien definidas y la utilización de prácticas de desarrollo rigurosas. A continuación, se detalla cómo se implementó RUP en el desarrollo de la aplicación para la toma de pedidos en el restaurante que plantea el proyecto.

#### **Fases de RUP**

##### **Detalles de Implementación**

##### **1. Inicio:**

- **Reuniones con Stakeholders:** Reuniones iniciales con los propietarios del restaurante y otros interesados clave para entender sus necesidades y expectativas.

- **Documento de Visión:** Descripción clara de la visión del proyecto, objetivos, beneficios esperados y alineación con las metas estratégicas del restaurante.
- **Caso de Negocio:** Análisis de costo-beneficio y justificación del proyecto.
- **Lista de Requisitos Iniciales:** Identificación de los requisitos clave, tanto funcionales como no funcionales, para establecer el alcance inicial del proyecto.

## 2. Elaboración:

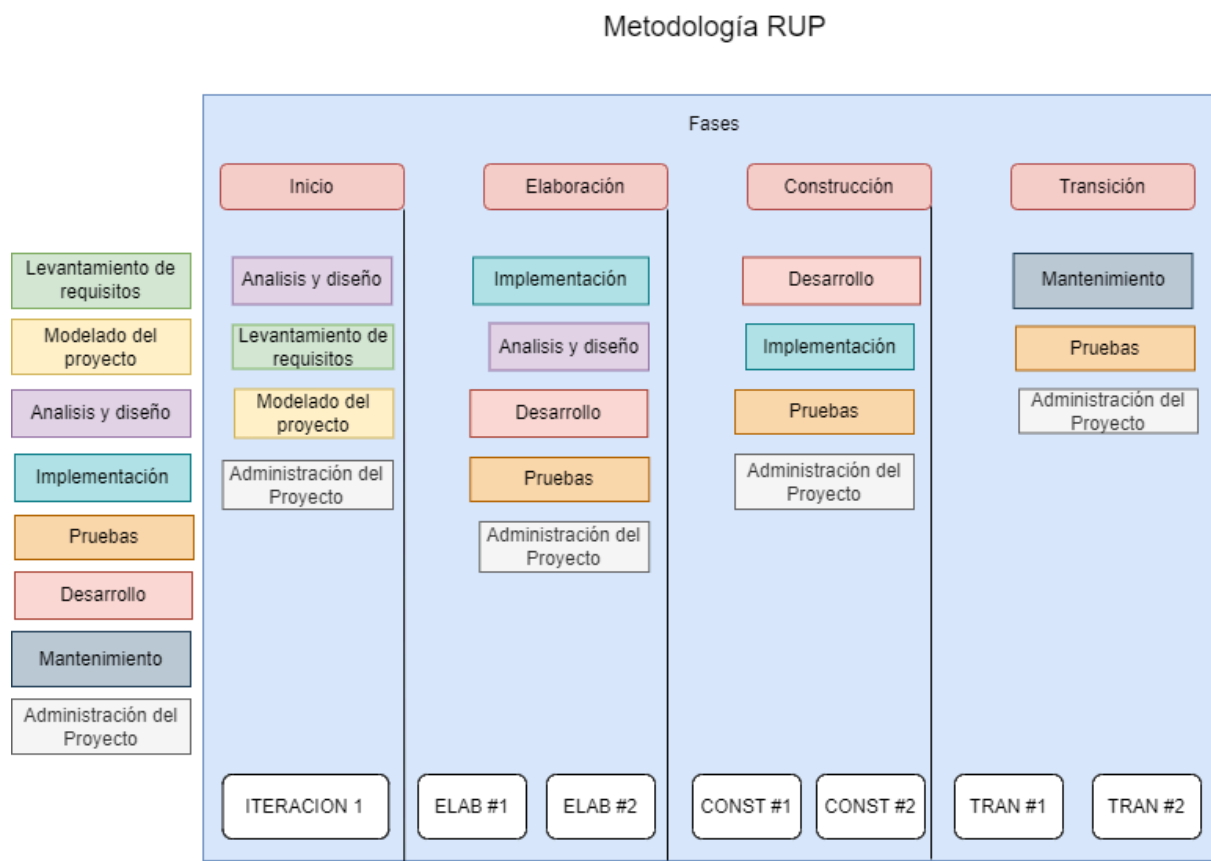
- **Análisis Detallado de Requisitos:** Recopilación y análisis exhaustivo de los requisitos, incluyendo entrevistas adicionales con stakeholders y análisis de procesos actuales.
- **Diseño de Arquitectura del Sistema:** Creación de diagramas UML para representar la estructura del sistema, incluidos los diagramas de casos de uso, clases y secuencia.
- **Planificación de Iteraciones:** Detallado plan de iteraciones para la fase de construcción, definiendo los incrementos del software y las prioridades de desarrollo.

## 3. Construcción:

- **Desarrollo Iterativo:** Implementación de funcionalidades del sistema en ciclos iterativos, con revisiones regulares para asegurar que se cumplan los requisitos y se mantenga la calidad.
- **Pruebas Continuas:** Realización de pruebas a lo largo de cada iteración para identificar y corregir errores, asegurando que cada incremento del software es funcional y estable.
- **Documentación:** Creación de documentación técnica detallada y guías de usuario para facilitar la implementación y uso del sistema.

#### 4. Transición:

- **Implementación en Producción:** Despliegue del sistema en el entorno de producción, asegurando que todo el hardware y software necesario esté en su lugar.
- **Capacitación de Usuarios:** Programas de capacitación para los usuarios del restaurante, incluyendo demostraciones prácticas y soporte inicial.
- **Feedback y Ajustes:** Recopilación de feedback de los usuarios finales para realizar ajustes finales y asegurar que el sistema cumple con sus expectativas y necesidades.



### Documentación Específica del Proyecto

## **1. Proyecto y Justificación**

El sector de la restauración se enfrenta constantemente al desafío de mejorar la eficiencia operativa y la experiencia del cliente para mantenerse competitivo. La implementación de tecnología avanzada es una de las soluciones más efectivas para abordar estos desafíos. Este proyecto tiene como objetivo desarrollar e implementar una aplicación innovadora para la toma de pedidos en un restaurante, utilizando dispositivos instalados en cada mesa. Esta aplicación permitirá a los clientes realizar pedidos de manera directa, optimizando así los procesos internos del restaurante y proporcionando una experiencia de usuario superior.

La aplicación para la toma de pedidos es una solución tecnológica diseñada específicamente para restaurantes que buscan modernizar y mejorar su servicio. Cada mesa del restaurante estará equipada con un dispositivo, como una tablet, que permitirá a los clientes acceder al menú, realizar pedidos y personalizarlos según sus preferencias.

### **Justificación:**

- Innovación: Moderniza el proceso de toma de pedidos.
- Incremento de ganancias: Aumenta la eficiencia y potencialmente las ventas.
- Control de datos: Mejora la gestión operativa con datos en tiempo real.
- Calidad de servicio: Reduce errores y tiempos de espera.
- Reducción de costos: Optimiza el uso de personal.

## **2. Requerimientos Funcionales y No Funcionales**

### **Funcionales:**

- Permitir a los clientes realizar pedidos desde las mesas.
- Mostrar el menú actualizado en la aplicación.



- Confirmar disponibilidad de productos en tiempo real.

#### **No Funcionales:**

- Seguridad de datos: Protección de información sensible.
- Usabilidad: Interfaz intuitiva y fácil de usar.
- Rendimiento: Respuesta rápida del sistema.

### **3. Riesgos y Retos**

#### **Riesgos:**

- Error en el software.
- Fuga de datos.
- Acceso no autorizado.

#### **Mitigación:**

- Pruebas exhaustivas y mantenimiento continuo.
- Encriptación y autenticación robusta.
- Capacitación en seguridad para el personal.

### **4. Triángulo de Hierro**

#### **Alcance:**

- **Funcionalidades:** Desarrollo de la aplicación para la toma de pedidos, verificación de inventario en tiempo real, gestión del menú, integración con sistemas de pago.
- **Características específicas:** Implementación de seguridad robusta, interfaz de usuario intuitiva.

- **Objetivos específicos:** Mejorar la eficiencia operativa, reducir los tiempos de espera, aumentar la satisfacción del cliente.

#### **Costo:**

- **Recursos humanos:** Contratación de desarrolladores, diseñadores de UX/UI, especialistas en seguridad.
- **Tecnología:** Adquisición de tablets para las mesas, servidores para alojar la aplicación, licencias de software necesarias.
- **Mantenimiento:** Costos asociados al mantenimiento continuo y actualización del software.

#### **Tiempo:**

- **Inicio:** Definición del alcance y planificación inicial.
- **Elaboración:** Refinamiento de requisitos y desarrollo de prototipos.
- **Construcción:** Desarrollo iterativo del software y pruebas.
- **Transición:** Implementación en el entorno de producción, capacitación a usuarios finales.

#### **Calidad:**

- **Pruebas exhaustivas:** Realización de pruebas unitarias, de integración y de aceptación del usuario para garantizar la calidad del software.
- **Feedback continuo:** Recopilación de retroalimentación de los usuarios y ajustes basados en sus comentarios.
- **Mejora continua:** Actualización regular del software para mejorar la funcionalidad y la seguridad.

