15 de septiembre de 2024

PROYECTO Tokyo bites

ALEJANDRO POIRIER 21.728.263-2

ALEJANDRO GATICA 18.608.055-6

JOSÉ REYES 21.385.393-7

INGENIERÍA DE SOFTWARE I

PTEC105 654

2do Semestre 2024

|  |  |  |  |
| --- | --- | --- | --- |
| Versión | Descripción | Responsable | Fecha |
| 1.0 | Versión inicial | Alejandro Poirier, Alejandro Gatica, José Reyes | 12/09/2024 |

Tabla de contenido

[Planteamiento del problema. 3](#_Toc177299063)

[Planteamiento de la solución. 4](#_Toc177299064)

[Funciones del software: 4](#_Toc177299065)

[Justificación del proyecto. 5](#_Toc177299066)

[Relevancia del proyecto para la sociedad 6](#_Toc177299067)

[Relevancia para los integrantes del proyecto. 6](#_Toc177299068)

[Definición de los objetivos el proyecto. 7](#_Toc177299069)

[Objetivo General: 7](#_Toc177299070)

[Objetivos Específicos: 7](#_Toc177299071)

[Metodología. 9](#_Toc177299072)

[Comparación de metodologías 9](#_Toc177299073)

[Propuesta de metodología mixta 10](#_Toc177299074)

[Pasos de la metodología mixta 11](#_Toc177299075)

[Especificación de Requerimientos de Software. 12](#_Toc177299076)

[Descripción de los Usuarios del Sistema 12](#_Toc177299077)

[Administrador 12](#_Toc177299078)

[Cliente 12](#_Toc177299079)

[Cajero Virtual 12](#_Toc177299080)

[Encargado de Despacho 12](#_Toc177299081)

[Lista de Requerimientos de Usuario 13](#_Toc177299082)

[Lista de Requerimientos del Sistema 14](#_Toc177299083)

[Requerimientos Funcionales 14](#_Toc177299084)

[Requerimientos No Funcionales 15](#_Toc177299085)

[Recursos consultados. 17](#_Toc177299086)

# Planteamiento del problema.

El restaurante Tokyo Bites, especializado en comida japonesa, ha enfrentado cambios significativos en su modelo de negocio tras la pandemia. Antes, la atención era mayormente presencial, donde los clientes ordenaban y pagaban directamente en el local. Sin embargo, el flujo de clientes presenciales ha disminuido considerablemente, y la mayoría de los pedidos ahora se realizan a través de una página web básica que solo muestra el menú y un número telefónico de contacto.

El proceso actual presenta varias ineficiencias:

1. **Pedidos manuales y confirmaciones:** Los encargados deben recibir y gestionar los pedidos por teléfono, confirmando manualmente la disponibilidad de los platos o combos. Esto puede generar demoras si el plato solicitado no está disponible, lo que genera pérdida de tiempo tanto para el cliente como para el representante del restaurante.
2. **Problemas de pago:** El sistema de pago actual depende de confirmaciones manuales de transferencias bancarias. Los encargados deben verificar si el pago se ha realizado, si el monto es correcto y asegurarse de que coincida con el total del pedido antes de procesar la entrega. Por la cantidad de pedidos, se dificulta confirmar en horas de alta demanda y es usual que se pierdan pedidos/dinero.
3. **Logística de entrega:** La gestión de entregas a domicilio también es manual, lo que puede generar retrasos o errores al calcular tiempos y disponibilidad de entrega para ciertas ubicaciones.
4. **Análisis de data:** El restaurante necesita de herramientas para realizar un análisis detallado de las ventas, preferencias de los clientes y patrones de consumo. Esto impide identificar con precisión los platos más populares, las ubicaciones geográficas clave con mayor demanda, las horas punta para pedidos y otras métricas clave que podrían optimizar las operaciones del restaurante. Se necesitan estos datos para ajustar la oferta, realizar promociones y gestionar de manera efectiva la logística.

# Planteamiento de la solución.

Se propone implementar una plataforma de ventas en línea que automatice y optimice los procesos de pedidos, pagos y logística, así como de generación de informe de ventas diarios, semanales y mensuales para posterior análisis.

Esta plataforma permitirá gestionar los pedidos de manera más eficiente, ofrecer un sistema de pagos en línea y mejorar la gestión de entregas a domicilio, eliminando los problemas derivados de la gestión manual.

## Funciones del software:

* **Visualización de productos en tiempo real:** Permite a los clientes acceder a un menú actualizado, mostrando los platos y promociones en tiempo real según su disponibilidad.
* **Registro y cuenta de usuario:** opción de creación de cuenta en la plataforma para facilitar el proceso de compra y permitir guardar preferencias, ver historial de pedidos y direcciones de entrega asociadas.
* **Gestión de pedidos en línea:** se realizarán los pedidos directamente de la plataforma, añadiendo los productos al carrito. La plataforma notificará automáticamente al equipo de cocina sobre los pedidos confirmados, con la posible opción de cancelación y devolución de dinero para el cliente, eliminando así la necesidad de gestión manual por parte del personal.
* **Procesamiento de pagos:** la plataforma procesará los pagos en línea de manera automática y segura para ambas partes implementando servicios de pago externos establecidos en el mercado. Esto evitará la necesidad de verificaciones manuales, garantizando una experiencia fluida para el cliente y gestión eficiente para el restaurante.
* **Optimización de logística de entrega:** se proporcionará información precisa sobre la ubicación del cliente y tiempos estimados de entrega. También podrá calcular la disponibilidad del servicio de entrega según zona geográfica y demanda en tiempo real. Además, permitirá la integración con servicios de reparto externos.
* **Generación de reportes y análisis de ventas:** El software generará informes diarios, semanales y mensuales de ventas, detallando los productos más vendidos, las horas de mayor demanda y el rendimiento financiero general. Un científico de datos podrá analizarla después esta información con las herramientas necesarias. ￼

# Justificación del proyecto.

La solución planteada responde a las necesidades urgentes del restaurante Tokyo Bites, que ha visto un cuello de botella en su crecimiento, esto debido a la falta de una plataforma digital que permita gestionar pedidos en línea, un sistema robusto y eficiente de entrega a domicilio y funcionalidades de estudio de clientes/mercado. Estos factores son clave en el mercado actual, ya que los consumidores buscan comodidad y rapidez en la adquisición, aspectos que ya tiene la competencia.

Implementar un sistema innovador y disruptivo en comparación con el actual flujo de trabajo, permitirá al restaurante adaptarse a las nuevas tendencias del mercado y mejorar su competitividad. Al optimizar sus operaciones, el restaurante podrá ofrecer una experiencia más eficiente y atractiva, lo que ayudará a captar nuevos clientes y retener a los actuales, así como la posibilidad de crecimiento futuro y apertura de nuevas sucursales.

## Relevancia del proyecto para la sociedad

Entre los beneficios sociales más considerables, se pueden destacar:

* **Accesibilidad**: Permitirá a más clientes realizar pedidos desde la comodidad de sus hogares, ampliando así el alcance del restaurante.
* **Generación de empleo**: Surgimiento de nuevas oportunidades laborales en el área de delivery y logística, así como posterior apertura de empleos relacionados con marketing y ciencia de datos.
* **Apoyo a negocios locales**: Se fomentará el consumo de un negocio local y toda la red de distribución que permite el funcionamiento de este. Esto contribuirá al desarrollo económico nacional.
* **Satisfacción del cliente**: El proceso de pedidos y pagos será más rápido y expedito, mejorando la experiencia del cliente.
* **Mejor comunicación:** Facilitará retroalimentación y mejora continua del servicio.

## Relevancia para los integrantes del proyecto.

Este proyecto brinda a los estudiantes de la oportunidad invaluable para aplicar los conocimientos estudiados en la asignatura en un entorno real, ganando experiencia práctica en el desarrollo de software. Al diseñar e implementar una plataforma digital integral, se trabajará con tecnologías y metodologías actuales y se aprenderá a resolver problemas operativos complejos con soluciones escalables para el negocio de comida, pero que puede replicarse también en otras áreas.

El proyecto permitirá seguir todas las etapas del proceso de **ingeniería de software**, desde la recopilación de requisitos/requerimientos, análisis, modelado, diseño, implementación, pruebas y despliegue.

Esta experiencia ayudará a fortalecer la comprensión sobre la importancia y procesos que tiene cada fase, mientras se desarrollan habilidades esenciales como la planificación, trabajo en equipo y la gestión de proyectos.

Además, se aprenderá a organizar e integrar arquitectura de software para poder garantizar la escalabilidad y eficiencia del sistema, asegurando que las distintas partes del software (UI, componentes, BD) funciones de manera cohesionada.

# Definición de los objetivos el proyecto.

## Objetivo General:

Desarrollar un sistema de ventas en línea para Tokio Bites que permita a los clientes realizar pedidos de forma fácil y eficiente, automatizando los procesos de compra, pago y entrega a domicilio, con el fin de mejorar la experiencia del cliente y optimizar la operación del restaurante, incrementando así las ventas, competitividad y presencia en el mercado nacional.

## Objetivos Específicos:

* **Implementar** una plataforma web interactiva que permita a los usuarios registrarse, explorar el menú y realizar pedidos, optimizando la experiencia del usuario, tomando como prioridad los dispositivos móviles. **Plazo de entrega: 21 de octubre, 2024 (3 sprint)**.
* **Integrar** una solución de pago externa segura (por ejemplo, Servipag, PayPal) que garantice la protección de los datos de los clientes y reduzca el tiempo de procesamiento de pagos en un 70% en comparación con el proceso actual. **Plazo de entrega: 28 de octubre, 2024 (mitad de un sprint).**
* **Desarrollar** un sistema de gestión de pedidos y entregas integrado con el equipo de cocina y el área de despacho, asegurando que el 95% de las órdenes se procesen y entreguen dentro del tiempo estimado de preparación y envío. **Plazo de entrega: 4 de noviembre de 2024 (1 sprint)**.
* **Implementar** un sistema de generación de informes que recopile datos de ventas, tiempos de entrega, y preferencias de los clientes, permitiendo un análisis posterior para estudios de mercado y la aplicación de ciencia de datos, con la capacidad de generar reportes automáticos al finalizar cada día/semana/mes. Plazo de entrega: 2 de diciembre, 2024 (2 sprint).

# Metodología.

## Comparación de metodologías

Las metodologías ágiles, como Scrum y Kanban, ofrecen una gran flexibilidad y constante retroalimentación, lo que ayuda al equipo de trabajo en su capacidad de adaptación rápida a los cambios. Este enfoque en entregas iterativas y constantes permite mantener una comunicación fluida entre el equipo de trabajo y el cliente inmediato, esto facilita la mejora continua del proyecto.

Sin embargo, una de sus desventajas es la falta de formalidad en la documentación y planificación a largo plazo, lo que puede generar confusión o falta de claridad, especialmente en proyectos grandes y complejos donde es necesario tener una visión más estructurada.

Por otro lado, las metodologías tradicionales, como Cascada y el Modelo en V, se caracterizan por su enfoque estructurado y lineal. Estas metodologías dividen claramente las fases de planificación, diseño, implementación y pruebas, lo que resulta ideal cuando los requisitos del proyecto están bien definidos desde el principio. No obstante, su principal desventaja es la falta de flexibilidad, lo que dificulta la adaptación a cambios durante el desarrollo.

Además, tienden a generar entregas más lentas y reducen la interacción con el cliente durante el proceso, lo que puede llevar a desajustes entre las expectativas iniciales y el producto final.

## Propuesta de metodología mixta

Se propone una metodología mixta que una **Scrum** con el enfoque más estructurado de **Cascada**. Esta combinación aprovecha la flexibilidad de Scrum y la formalidad en planificación de Cascada.

¿Por qué elegir un enfoque mixto?

* Mayor flexibilidad y adaptabilidad, uso de Scrum: Se usará para gestionar los ciclos cortos e iterativos de desarrollo, permitiendo entregas frecuentes y revisiones regulares del progreso al finalizar cada sprint. Esto permitirá ajustar el sistema de ventas y logística del restaurante a tiempo, según se va recolectando feedbwack del cliente de manera frecuente.
* Mejor estructura y control, uso de Cascada: Se aplicará en las fases iniciales más estructuradas que darán una idea general de la estructura del proyecto a largo plazo, esto incluye fases como Recopilación de requerimientos, Diseño del sistema y arquitectura del software, entre otras. Esto evitará confusiones y permitirá una alineación del equipo de las fases y fechas finales de entrega de cada parte del software.

## Pasos de la metodología mixta

1. Fase de planificación y requisitos (Cascada):
   1. Identificación clara de todos los requisitos funcionales y no funcionales.
   2. Diseño de los diagramas generales, la arquitectura del sistema y la integración de las tecnologías necesarias.
2. Fase de desarrollo iterativo (Scrum):
   1. Dividir el proyecto en sprints de 2 – 3 semanas, con reuniones diarias de seguimiento (daily standups).
   2. Priorizar entregas incrementales y funcionales para mostrar al cliente la implementación de pequeñas funcionalidades.
   3. Durante este proceso de comunicación con el cliente al final de cada sprint, se pondrá foco en la retroalimentación constante entre el equipo y el cliente final.
3. Fase de pruebas y validación (Cascada + Scrum):
   1. Al termino de cada sprint, se realizará una fase de pruebas exhaustivas de las funcionalidades desarrolladas.
   2. Se utilizará la metodología tradicional de Cascada para la planificación y ejecución de pruebas exhaustivas al final del desarrollo del sistema completo.
4. Fase de despliegue y mantenimiento (Scrum):
   1. Despliegue gradual del sistema con soporte continuo, aplicando mejoras y ajustes según la retroalimentación de usuarios y/o clientes.

En resumen, este proceso se basa en una metodología híbrida que combina un enfoque ágil, principalmente en el desarrollo y la ejecución diaria del proyecto, con un enfoque tradicional en las etapas iniciales de planificación y control. Esto proporciona al equipo una visión general y unificada del proyecto y sus plazos, al tiempo que otorga la agilidad necesaria para adaptarse de manera eficiente a los cambios.

# Especificación de Requerimientos de Software.

## Descripción de los Usuarios del Sistema

### Administrador

Usuario encargado de la gestión del sistema, incluyendo la administración de productos, clientes, usuarios y reportes de ventas.

### Cliente

Usuario que accede a la plataforma para registrarse, realizar pedidos y consultar el menú. Tiene la posibilidad de anular pedidos y acceder a promociones exclusivas.

### Cajero Virtual

Funcionario del restaurante que gestiona las ventas asociadas a los pedidos realizados por los clientes en la web.

### Encargado de Despacho

Usuario encargado de recibir y gestionar las órdenes de despacho de los pedidos.

## Lista de Requerimientos de Usuario

1. El cliente debe poder registrarse en el sistema proporcionando información personal, incluyendo su dirección, teléfono y correo electrónico. El sistema debe validar la existencia del correo mediante un enlace de confirmación.
2. El cliente debe poder navegar por el menú, visualizar los productos disponibles y sus precios, y agregar productos al carrito de compra.
3. El cliente debe poder gestionar sus pedidos, incluyendo la posibilidad de anular un pedido antes de su preparación, especificando el motivo de la anulación.
4. El administrador debe poder gestionar los productos del menú, incluyendo agregar, editar y eliminar productos, así como actualizar su disponibilidad.
5. El cajero virtual debe poder gestionar los pagos realizados en línea, generar boletas electrónicas y enviarlas al cliente por correo electrónico.
6. El encargado de despacho debe poder recibir las órdenes de despacho y gestionarlas en el orden de llegada.

## Lista de Requerimientos del Sistema

### Requerimientos Funcionales

1. El sistema debe permitir a los clientes armar un pedido seleccionando productos disponibles en el menú
2. Cada producto debe incluir nombre, descripción, precio, y disponibilidad en tiempo real.
3. La orden de compra solo debe ser procesada una vez que el pago haya sido confirmado a través de una plataforma de pago externa integrada (por ejemplo, Servipag, PayPal, o transferencia bancaria).
4. El sistema debe permitir a los clientes registrados anular un pedido, siempre que la preparación no haya comenzado. Se debe solicitar el motivo de anulación y generar un registro.
5. El sistema debe permitir la venta únicamente a usuarios registrados.
6. Las ventas web deben asociarse a un "cajero virtual", representado por un usuario con privilegios específicos que pueda monitorear y gestionar las transacciones.
7. El sistema debe permitir el registro de nuevos clientes con un formulario que solicite los siguientes datos: RUN, nombre completo, dirección, comuna, provincia, región, fecha de nacimiento, sexo, correo electrónico, y teléfono.
8. La validación del correo electrónico debe realizarse mediante un enlace de verificación enviado al cliente.
9. El sistema debe permitir el registro tanto presencial en el local (a través del administrador) como en línea, con el mismo nivel de validación de la información proporcionada, especialmente el correo electrónico.
10. El sistema debe permitir la gestión de disponibilidad de productos en tiempo real, para que los administradores puedan agregar, editar o eliminar productos del menú según existencias.
11. Para cada venta realizada, el sistema debe generar una boleta electrónica en formato PDF, que será enviada automáticamente por correo electrónico al cliente.
12. El sistema debe proporcionar un módulo de reportes para el dueño del restaurante, que permita generar informes de ventas filtrados por rango de fechas, productos vendidos, y cantidad de transacciones realizadas.
13. El sistema debe contar con mantenedores de datos maestros que permitan la gestión de productos, usuarios y clientes, con roles de acceso específicos.
14. El sistema debe validar los correos electrónicos mediante una API de mensajería externa que verifique la existencia y estado activo del correo electrónico del cliente.
15. El sistema debe implementar un mecanismo de autenticación con roles definidos para limitar el acceso a funcionalidades según el perfil del usuario (administrador, cajero, cliente, etc.).
16. El sistema debe generar órdenes de despacho impresas para el área de cocina, con el detalle de productos solicitados, hora de pedido y orden de llegada.

### Requerimientos No Funcionales

1. El sistema debe ser accesible desde cualquier navegador web estándar (Chrome, Firefox, Edge).
2. El sistema debe estar disponible las 24 horas del día.
3. Debe implementarse una arquitectura de tres capas (presentación, lógica de negocio y datos), garantizando la separación de responsabilidades y una fácil escalabilidad.
4. El sistema debe ser desarrollado utilizando bases de datos Oracle y lenguajes de programación orientados a objetos como .NET o Java EE, priorizando la integridad y seguridad de los datos.
5. Los procesos automáticos, como el envío de correos electrónicos y la emisión de boletas, deben ser manejados mediante procedimientos almacenados en la base de datos para optimizar el rendimiento y evitar errores.
6. La integración con sistemas externos, como la plataforma de pago y la API de mensajería, debe realizarse utilizando servicios web (REST o SOAP), garantizando estándares de comunicación seguros y eficientes.
7. El sistema debe cifrar las contraseñas de los usuarios y realizar control de sesiones para prevenir accesos no autorizados, siguiendo las mejores prácticas de seguridad informática (como OAuth 2.0 o JWT).
8. Todas las entradas de datos en formularios deben incluir validaciones que verifiquen la integridad y formato adecuado de los datos (por ejemplo, correos electrónicos válidos, números telefónicos en formato correcto, etc.).
9. La interfaz gráfica debe ser intuitiva y coherente con los estándares modernos de diseño de interfaces, incluyendo elementos visuales familiares y accesibles.
10. El tiempo de aprendizaje para un usuario estándar del sistema no debe exceder las 4 horas de capacitación básica, gracias a una interfaz amigable y clara.
11. La tasa de errores cometidos por el usuario al interactuar con el sistema debe ser inferior al 1% del total de transacciones procesadas.
12. El sistema debe incluir manuales de usuario claramente estructurados, tanto en formato digital como imprimible, que describan los procesos clave.
13. El sistema debe proporcionar mensajes de error claros y comprensibles para el usuario final, orientados a resolver problemas o redirigir acciones de manera efectiva.
14. El sistema debe contar con un módulo de ayuda en línea con preguntas frecuentes (FAQ) y asistencia en tiempo real a través de un chatbot o sistema de tickets.
15. La aplicación web debe ser responsiva, garantizando una correcta visualización y funcionalidad en dispositivos móviles (tabletas, teléfonos inteligentes) y pantallas de diferentes tamaños.
16. Las interfaces gráficas deben cumplir con principios de usabilidad, accesibilidad y diseño inclusivo, permitiendo a usuarios con diferentes capacidades utilizar el sistema sin problemas.

# Recursos consultados.

Sommerville, I. (2011). Software Engineering (9th ed.). Pearson.

Sommerville destaca la importancia de la ingeniería de software desde la perspectiva de los procesos iterativos y la importancia de la gestión de requerimientos, lo que fue clave para estructurar el análisis de las metodologías ágiles y tradicionales en este proyecto.

Pressman, R. (2014). Software Engineering: A Practitioner's Approach (8th ed.). McGraw-Hill.

Pressman proporciona un enfoque práctico para la planificación de proyectos de software, lo que ayudó a guiar la definición de objetivos SMART y la justificación de la metodología mixta utilizada para este proyecto.