

Introducción al rup

El Proceso Unificado Racional (RUP) es un enfoque de desarrollo de software que se basa en ciclos de vida iterativos e incrementales. Esta metodología permitiría al equipo abordar el proyecto en etapas manejables y realizar entregas incrementales, adaptándose a las necesidades cambiantes del sistema.

En la industria de restaurantes, donde los requisitos pueden evolucionar debido a las tendencias del mercado y las preferencias de los clientes, esta flexibilidad sería crucial para garantizar la relevancia del software. Además, el énfasis de RUP en la comunicación constante con stakeholders y la gestión efectiva de riesgos serían particularmente beneficiosos en el contexto de un software de gestión de restaurantes, donde la colaboración con el personal del restaurante y la anticipación de desafíos serían esenciales para el éxito del proyecto.

Dimensiones del Rup

En este trabajo se aplicaron las dimensiones horizontales y verticales para así poder tener un control de las fases, procesos, iteraciones, incrementos, roles, módulos y todo lo relacionado con la metodología rup.

A continuación, se presentan las dimensiones de la metodología rup:

Dimensión de Flujos de Trabajo (Workflows):

Modelado de Negocio:

Identificar los procesos de negocio del restaurante, tanto en el local como en línea.

Analizar los requisitos específicos del negocio para la gestión de pedidos, inventario, facturación, etc.

Requerimientos:

Definir los requisitos del sistema para la plataforma web y el sistema local.

Identificar las características esenciales, como la integración de pedidos en línea y la gestión de inventario.

Diseño:

Desarrollar la arquitectura del sistema teniendo en cuenta la integración entre la plataforma web y el sistema local.

Diseñar la interfaz de usuario para garantizar una experiencia coherente tanto en línea como en el local.

Implementación:

Codificar las funcionalidades necesarias para la gestión de pedidos, procesamiento de pagos y actualización de inventario.

Desarrollar módulos específicos para la plataforma web y el sistema local.

Pruebas:

Realizar pruebas para garantizar que el sistema cumpla con los requisitos tanto en el entorno local como en línea.

Implementar pruebas de integración para asegurar la interoperabilidad entre las dos plataformas.

Despliegue:

Planificar y ejecutar la implementación del sistema en el entorno local y en la web.

Coordinar la transición suave desde el sistema existente al nuevo.

Gestión de Proyectos: Planificar y supervisar el proyecto, teniendo en cuenta las peculiaridades de la implementación en línea y local.

Dimensión de Actividades (Activities):

Comunicación:

Facilitar la comunicación entre el equipo de desarrollo, el personal del restaurante y los usuarios finales.

Garantizar una comunicación efectiva durante el proceso de implementación.

Planificación:

Desarrollar planes detallados para cada fase del proyecto, considerando los aspectos específicos de la implementación en línea y local.

Modelado:

Crear modelos que representen tanto la arquitectura del sistema como los procesos de negocio específicos del restaurante.

Construcción:

Desarrollar y probar las funcionalidades del sistema, priorizando las características críticas para la operación diaria del restaurante.

Despliegue:

Coordinar las actividades de despliegue tanto para el sistema local como para la plataforma web.

Dimensión de Roles (Roles):

Roles en el Proceso: Identificar roles específicos, como el personal del restaurante, el equipo de desarrollo y el personal de soporte técnico.

Responsabilidades de los Roles: Definir las responsabilidades de cada rol en relación con la implementación y operación del sistema.

Habilidades de los Roles: Identificar las habilidades necesarias para cada rol, como conocimientos técnicos para el equipo de desarrollo y habilidades operativas para el personal del restaurante.

Dimensión de Artefactos (Artifacts):

Artefactos del Proceso: Documentar los resultados de cada fase del proceso, incluyendo especificaciones de requisitos, modelos de diseño y casos de prueba.

Modelos: Crear modelos que representen la arquitectura del sistema, la interfaz de usuario y los procesos de negocio.

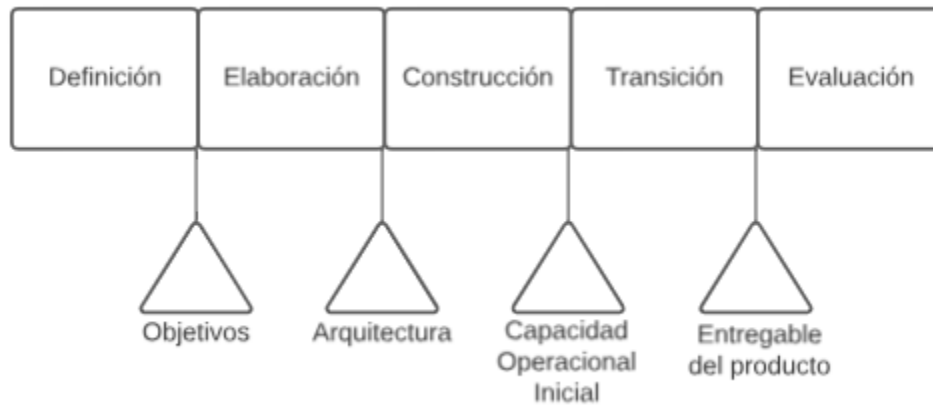
Código Fuente: Desarrollar y mantener el código fuente tanto para el sistema local como para la plataforma web.

Pruebas: Documentar los casos de prueba y los resultados de las pruebas, asegurando la calidad del sistema.

Documentación: Crear documentación detallada para usuarios finales y personal del restaurante.

Fases del rup

Figura 1



El ciclo de vida del software del RUP se descompone en cuatro fases secuenciales (figura 1). En cada extremo de una fase se realiza una evaluación (actividad: Revisión del ciclo de vida de la finalización de fase) para determinar si los objetivos de la fase se han cumplido. Una evaluación satisfactoria permite que el proyecto se mueva a la próxima fase.

Planeando las fases

Fase 1: Definición

Objetivos: Definir los objetivos iniciales, como la gestión de pedidos locales y la captura de datos de clientes.

Stakeholders: Identificar a los principales stakeholders, como el equipo del restaurante y los clientes locales.

Visión: Crear una visión inicial del sistema CRM para el restaurante.

Fase 2: Elaboración

Requerimientos: Capturar los requerimientos esenciales, como la gestión de pedidos, la base de datos de clientes y la integración de pagos locales.

Fase 3: Construcción

Desarrollo Iterativo: Implementar la funcionalidad local, incluyendo la gestión de pedidos y la base de datos de clientes.

Pruebas Continuas: Realizar pruebas unitarias y de integración para garantizar la calidad y la coherencia en el nivel local.

Fase 4: Transición

Pruebas de Aceptación: Realizar pruebas de aceptación con el personal del restaurante para asegurar que las características locales funcionen según lo esperado.

Formación: Capacitar al personal del restaurante en el uso del sistema CRM local.

Lanzamiento Local: Lanzar la versión local del sistema CRM.

Fase 5: Evaluación

Recopilación de Feedback: Obtener comentarios de los usuarios sobre ambas versiones (local y web) del sistema CRM.

Análisis de Resultados: Evaluar el rendimiento y la satisfacción del usuario en ambas plataformas.

Mejoras Futuras: Planificar mejoras basadas en el feedback y los resultados de la evaluación.

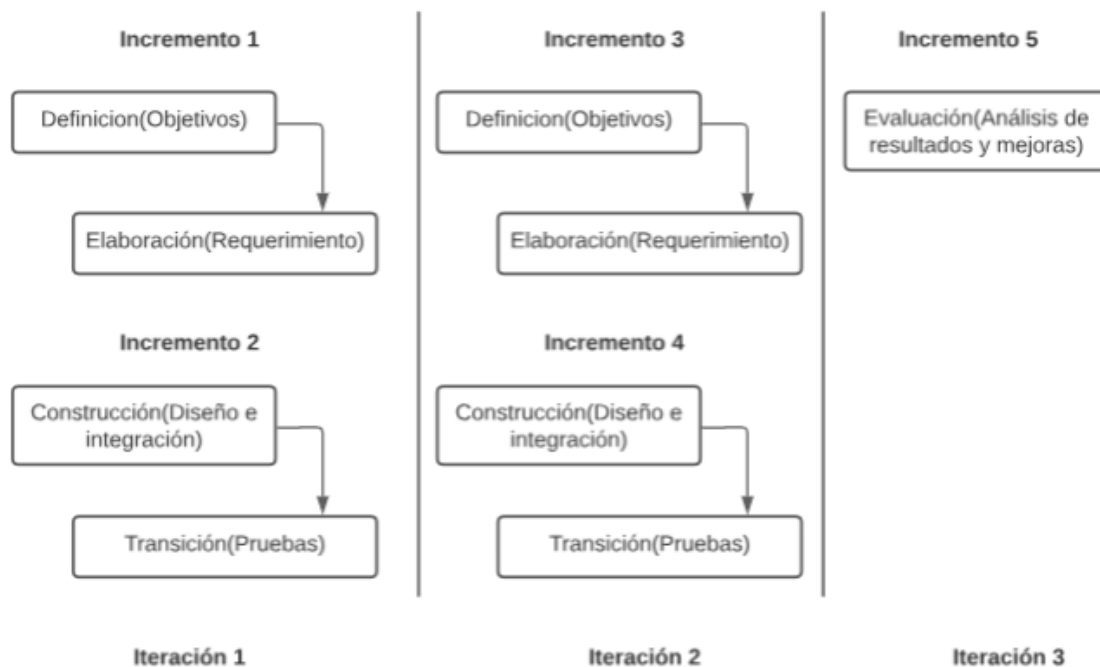
Iteraciones

El RUP maneja el proceso Iterativo Incremental para el desarrollo de las aplicaciones o proyectos, por tal motivo es de suma importancia explicar brevemente en que consiste este proceso.

Proceso Iterativo e Incremental

Este proceso se refiere a la realización de un ciclo de vida de un proyecto y se basa en la evolución de prototipos ejecutables que se muestran a los usuarios y clientes. En este ciclo de vida iterativo a cada iteración se reproduce el ciclo de vida en cascada a menor escala, estableciendo los objetivos de una iteración en función de la evaluación de las iteraciones precedentes y las actividades se encadenan en una minicascada con un alcance limitado por los objetivos de la iteración. En la figura 2 se muestran los pasos a realizar para seguir el ciclo de vida iterativo incremental, hasta la realización de una fase.

Figura 2



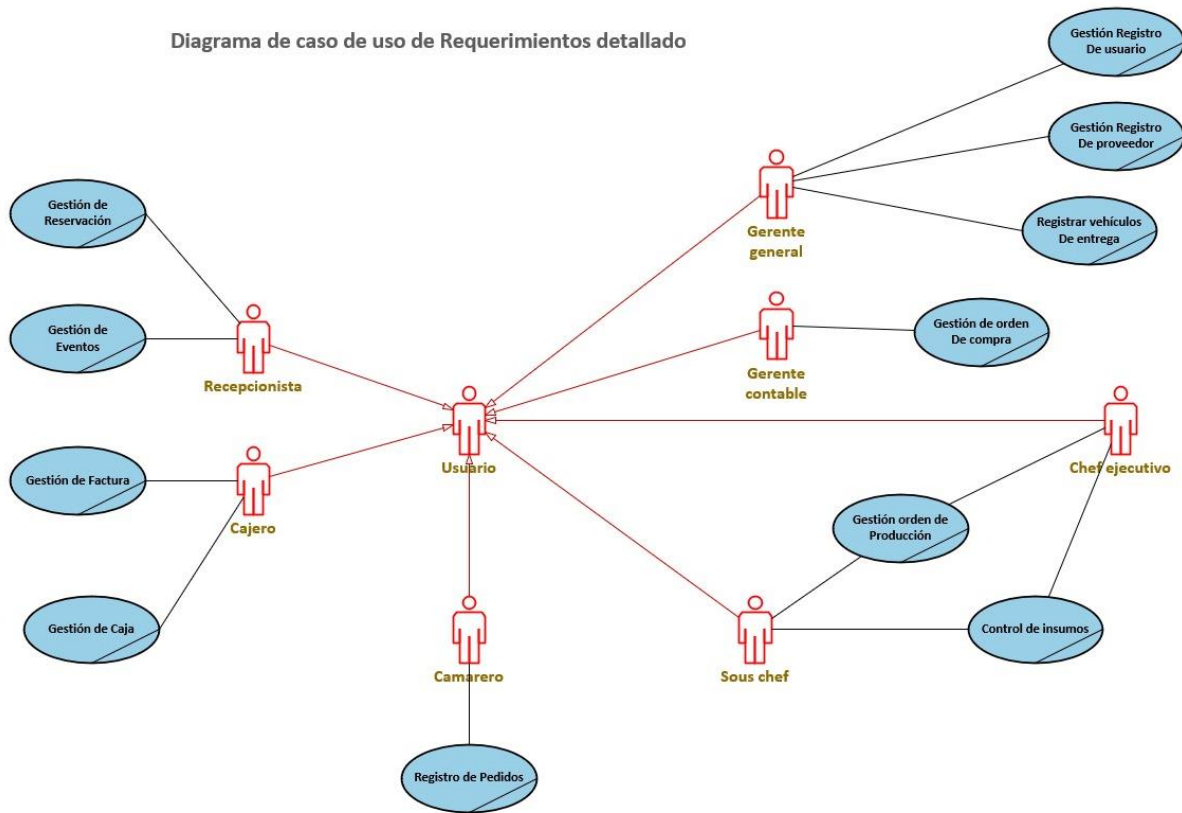
Modelado del negocio

El modelamiento del negocio es una disciplina en la metodología RUP que se ve en el ámbito más amplio de la empresa. Se utiliza para entender el negocio actual, procesos y determinan la forma en que se pueden mejorar. Identificar oportunidades para la automatización es una forma en que puede mejorar los procesos de negocio.

Para poder modelar el negocio se utilizaron diferentes diagramas mediante los cuales se encuentran:

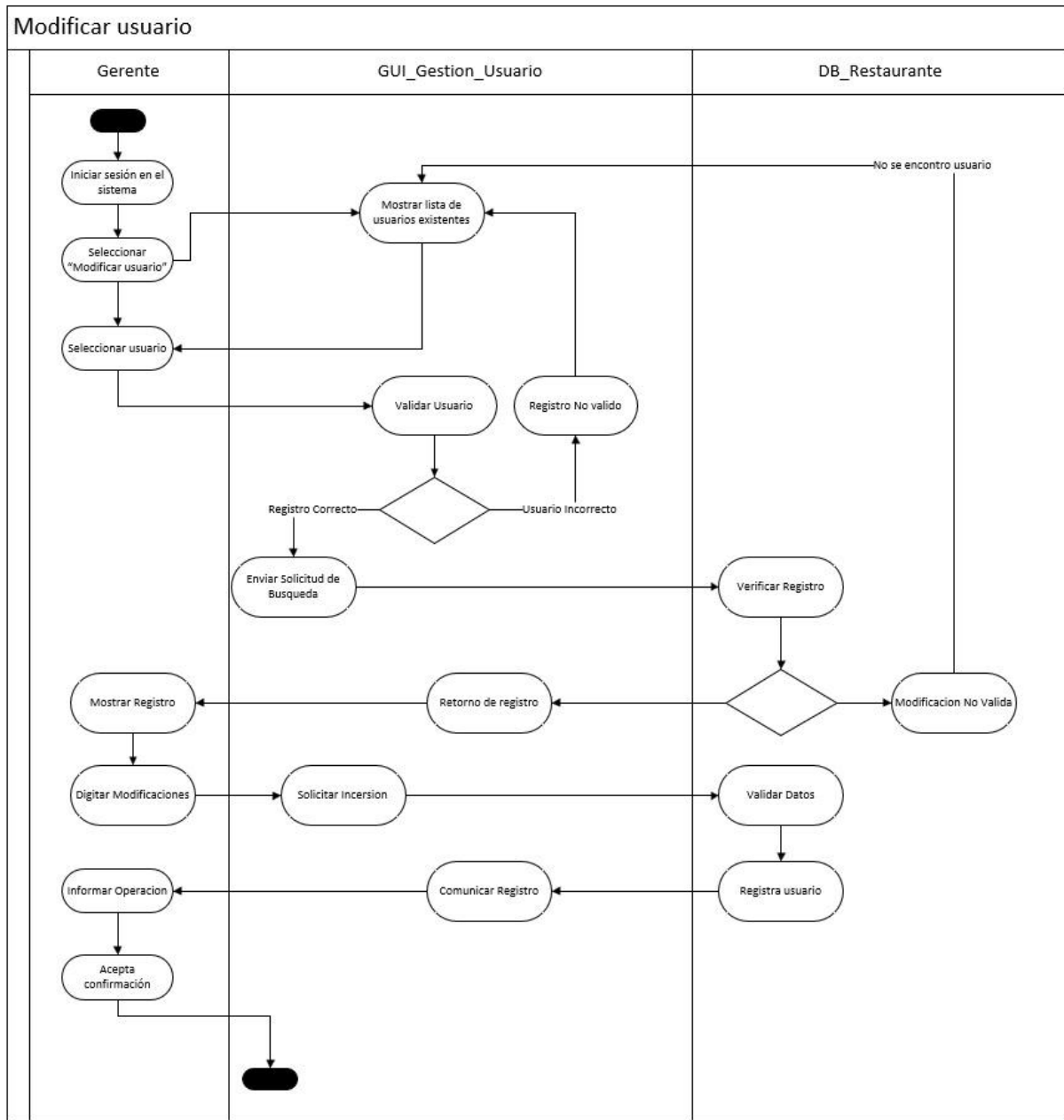
Diagramas de Caso de uso: Los diagramas de clases son diagramas estructurales del lenguaje unificado de modelado o UML (unified modeling language). Este lenguaje de modelado visual es un estándar ISO para visualizar los sistemas de la programación orientada a objetos, aunque también permite visualizar procesos de negocio.

Ejemplo de diagrama de caso de uso a nivel de RUP Figura 3



Diagramas de actividad: Un diagrama de actividad UML ayuda a visualizar un determinado caso de uso a un nivel más detallado. Es un diagrama de comportamiento que ilustra el flujo de actividades a través de un sistema. Los diagramas de actividad UML también pueden utilizarse para representar un flujo de eventos en un proceso comercial.

Ejemplo de diagrama de actividad a nivel de RUP Figura 4



Requerimientos

Un requerimiento define la funcionalidad que se espera que un determinado SW tenga. Un requerimiento expresa lo que el cliente quiere. El Análisis de requerimientos es el proceso mediante el cual se identifican y se comprenden los requerimientos de un software.

Requerimientos Funcionales: representan funciones que tiene que realizar el sistema.

Requerimientos No Funcionales: representan elementos complementarios a las funciones del sistema como interfaces, entorno de operación, seguridad, rendimiento, etc.)

Requerimientos Funcionales

Usuarios

RFU-1) Registrar, modificar y eliminar a los usuarios

El sistema permitirá registrar, modificar y eliminar a los diversos usuarios del sistema, brindando sus respectivos permisos y accesos al sistema.

RFU-2) Permitir al cliente crear un usuario propio

El sistema permitirá al cliente crear un usuario propio en el sistema, detallando los datos necesarios para realizar dicho usuario .

Reservas

RFU-3) Realizar y gestionar reservas de mesas en el restaurante

El sistema permitirá al usuario realizar y gestionar reservas de mesas, detallando los datos necesarios para realizar dicho proceso.

RFU-4) Modificar el estado de las reservas

El sistema permitirá al usuario modificar el estado de una reserva, si ha sido cancelada, efectuada o incumplida.

RFU-5) Realizar y gestionar programaciones de eventos en el restaurante

El sistema permitirá al usuario realizar y gestionar programaciones de eventos en el restaurante, detallando los datos necesarios, especificaciones, pedidos, entre otros.

Venta/Producción

RFU-6) Registrar, modificar y eliminar a los clientes

El sistema permitirá registrar, modificar y eliminar toda la información relacionada con los clientes, detallando toda la información requerida como nombre, dirección, teléfono, entre otros.

RFU-7) Agregar, modificar y eliminar elementos del menú del restaurante

El sistema permitirá al usuario agregar, modificar y eliminar elementos del menú del restaurante, permitiendo al usuario realizar cambios y actualizaciones de manera rápida y precisa.

RFU-8) Registrar y gestionar las órdenes de los clientes

El sistema permitirá al usuario registrar y gestionar las órdenes de los clientes, detallando todos los tipos de productos, la cantidad y especificaciones que el cliente desea.

RFU-9) Generar un orden de producción

El sistema permitirá generar una orden de producción, para poder ser visualizada por el área de cocina, a partir de las órdenes de los clientes, detallando todos los tipos de productos, la cantidad y especificaciones que el cliente desea.

RFU-10) Registrar y controlar los insumos de cocina

El sistema permitirá al usuario registrar los insumos que se adquieran en el restaurante, con datos, como nombre, cantidad, descripción, entre otros. Además, permitirá llevar un control en tiempo real de la cantidad de los insumos en existencias.

RFU-11) Calcular automáticamente la cantidad de ingredientes necesarios para preparar los pedidos

El sistema permitirá modificar en tiempo real las existencias de los insumos, en base a las órdenes de producción. Estandarizando la cantidad de los insumos empleada para la creación de los diversos pedidos.

RFU-12) Registrar, modificar y eliminar a los proveedores

El sistema permitirá al usuario registrar, modificar y eliminar toda la información relacionada con los proveedores, permitirá llevar un mejor control de ellos.

RFU-13) Generar y gestionar las órdenes de compra

El sistema permitirá al usuario generar y gestionar las órdenes de compra de insumos, detallando los datos necesarios para realizar dicho proceso, relacionándolas con los proveedores.

RFU-14) Registrar las facturas de compras

El sistema permitirá al usuario ingresar las facturas de compras que se llevan a cabo, para tener un mejor control de ellas. Relacionándolas con las órdenes de compra antes emitidas.

RFU-15) Generar facturas de los pedidos

El sistema generará las facturas de los pedidos de manera automática, en base a los pedidos realizados por los clientes, tomando posibles descuentos u ofertas que existan en ese momento.

RFU-16) Calcular el impuesto y la propina

El sistema calculará los impuestos y las propinas que se reflejan en las facturas, en base a la normativa a seguir.

RFU-17) Registrar los vehículos de entrega

El sistema permitirá al usuario registrar los vehículos de entrega que el restaurante dispone, detallando toda la información necesaria.

RFU-17) Modificar y controlar el estado de los vehículos de entrega

El sistema permitirá al usuario modificar y controlar el estado de los vehículos de entrega que el restaurante dispone en tiempo real.

RFU-18) Registrar la informacion los repartidores

El sistema permitirá al usuario registrar la información de los repartidores.

RFU-19) Modificar y controlar el estado de los repartidores

El sistema permitirá al usuario modificar y controlar el estado de los repartidores de los que el restaurante dispone en tiempo real.

RFU-20) Gestionar los envíos de los pedidos

El sistema permitirá al usuario gestionar los envíos de los pedidos, asignando a cada envío un vehículo y un repartidor.

Caja

RFU-21) Realizar la apertura de la caja

El sistema permitirá al usuario realizar la apertura de la caja y todas las actividades correspondientes a dicho proceso, brindando los detalles necesarios.

RFU-22) Realizar el arqueo de la caja

El sistema permitirá al usuario realizar el arqueo de la caja y todas las actividades correspondientes a dicho proceso, brindando los detalles necesarios.

RFU-23) Realizar el desglose de la caja

El sistema permitirá al usuario realizar el desglose de la caja y todas las actividades correspondientes a dicho proceso, brindando los detalles necesarios.

RFU-24) Realizar el cierre de la caja

El sistema permitirá al usuario realizar el cierre de la caja y todas las actividades correspondientes a dicho proceso, brindando los detalles necesarios.

RFU-25) Realizar, gestionar y controlar el depósito del día

El sistema permitirá al usuario realizar, gestionar y controlar el depósito del día y todas las actividades correspondientes a dicho proceso, brindando los detalles necesarios.

Requerimientos No Funcionales

A continuación, se presentan los requerimientos no funcionales:

Seguridad: El software debe garantizar la seguridad de los datos y la información confidencial del restaurante, como los registros de clientes y las transacciones financieras.

Rendimiento: El software debe ser capaz de manejar eficientemente un alto volumen de transacciones y consultas, especialmente durante los períodos de mayor demanda.

Debe tener una buena capacidad de respuesta y tiempo de carga rápido para evitar retrasos o tiempos de espera prolongados.

Escalabilidad: El sistema debe poder adaptarse y crecer a medida que el restaurante amplíe sus operaciones. Esto implica que el software debe ser capaz de manejar un mayor número de usuarios, transacciones y datos sin afectar su rendimiento.

Disponibilidad: El software debe estar disponible y accesible en todo momento, ya que la interrupción del servicio puede tener un impacto significativo en las operaciones del restaurante.

Usabilidad: El software debe ser intuitivo y fácil de usar para el personal del restaurante. Debe contar con una interfaz de usuario clara y bien diseñada, con instrucciones claras y funciones accesibles.

Compatibilidad: El software debe ser compatible con los sistemas operativos y dispositivos utilizados en el restaurante.

Debe ser compatible con otros sistemas o software utilizados, como sistemas de punto de venta.

Instalación: El sistema debe ser fácil de instalar.

Mantenibilidad: El sistema deberá estar documentado.

El sistema debe contar con una interfaz de administración que incluya: administración de usuarios, administración de perfiles y de accesos.

Validación de la Información: El sistema debe validar la información que se administra en todos los formularios, teniendo en cuenta elementos tales como: obligatoriedad de campos, longitud de caracteres permitida por campo, tipo de datos, entre otros.

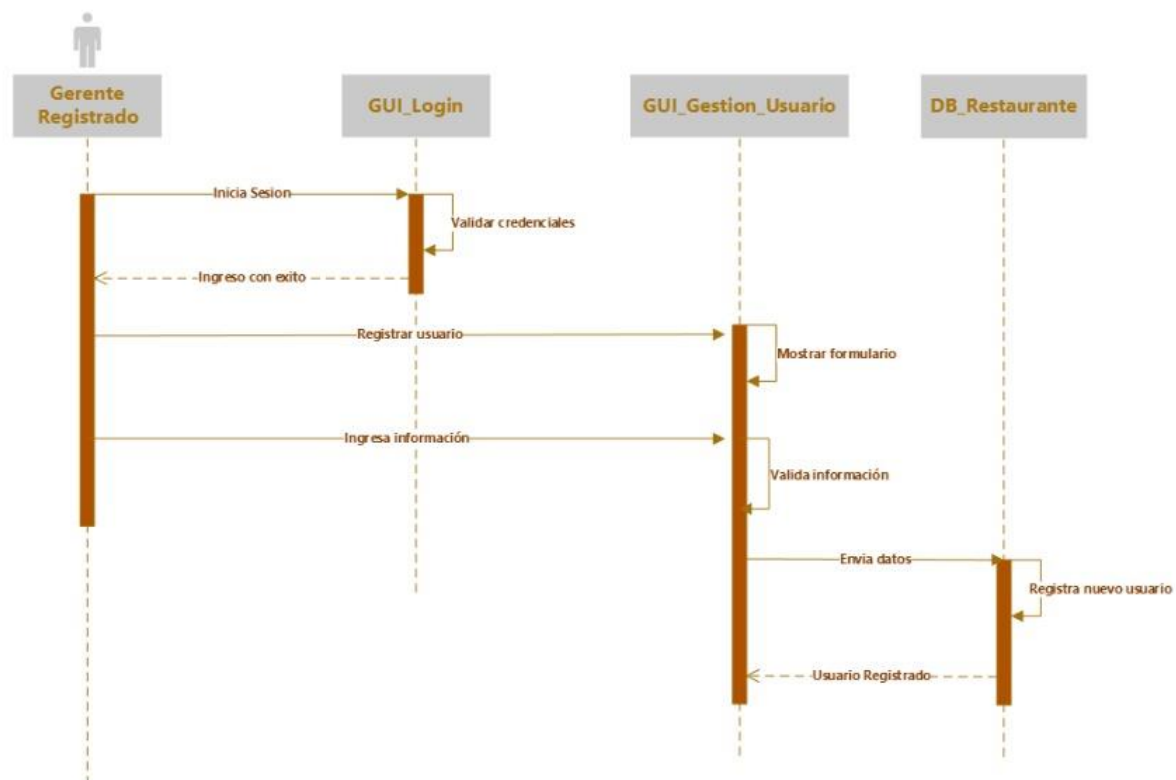
Backups: El sistema debe contar con funciones para generar copias de seguridad de manera periódica de toda la información almacenada en el sistema. La responsabilidad de crear, almacenar y restaurar estas copias de seguridad recae en el administrador del sistema.

Análisis y diseño

Para el mejoramiento del análisis de los CU se utilizaron diferentes métodos entre los cuales se pueden encontrar:

Diagramas de secuencia: Un diagrama de secuencia es una forma de diagrama de interacción que muestra los objetos como líneas de vida a lo largo de la página y con sus interacciones en el tiempo representadas como mensajes dibujados como flechas desde la línea de vida origen hasta la línea de vida destino.

Ejemplo de diagrama de secuencia a nivel de RUP Figura 5



Diagramas de colaboración: Un diagrama de colaboración presenta la organización estructural de los objetos que envían y reciben mensajes. Un diagrama de colaboración incluye los siguientes elementos: objeto: representa los diferentes objetos utilizados. actor: representa un elemento externo al sistema.

Ejemplo de diagrama de colaboración Figura 6

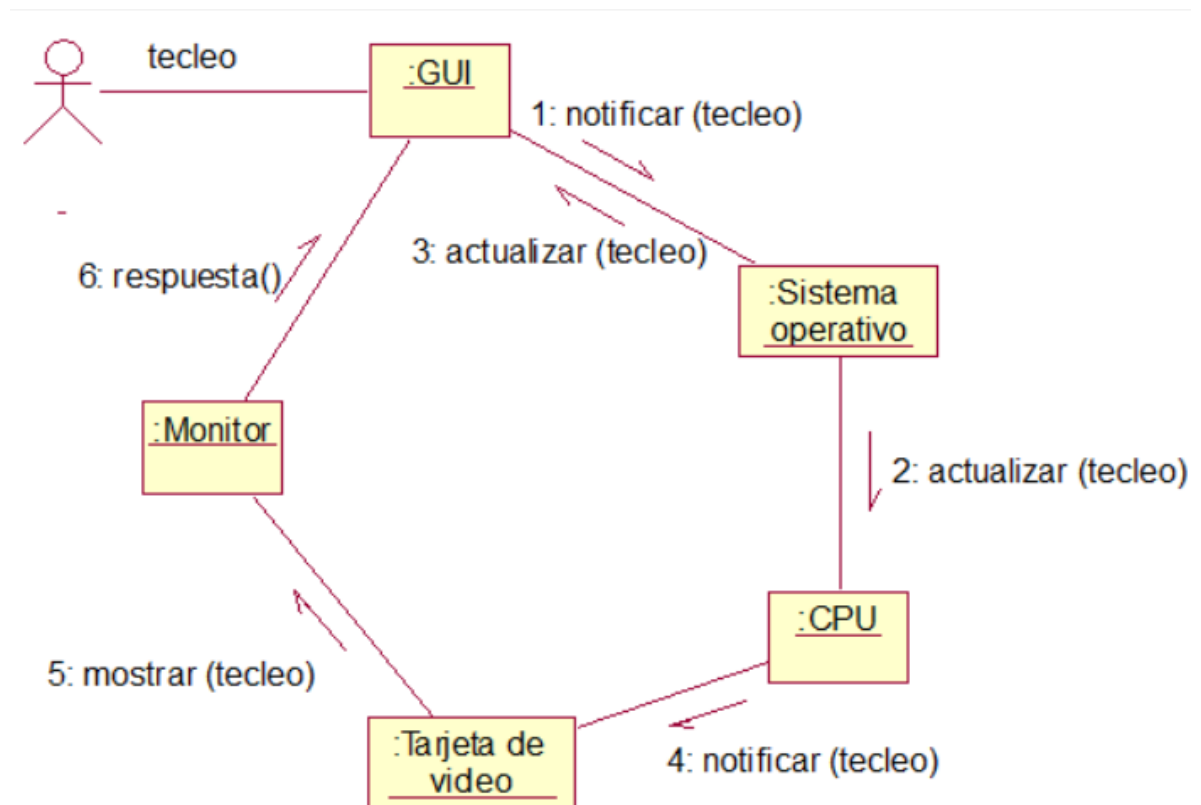
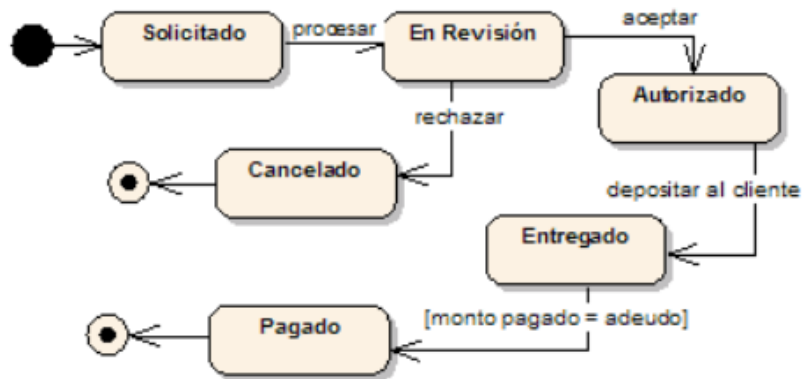


Diagrama de estados: Un diagrama de estado UML (también llamado diagrama de estado, diagrama de transición de estados o diagrama de máquina de estados) muestra los estados por los que pasa una máquina de estados finitos, es decir, un modelo de comportamiento que consiste en acciones y estados o transiciones a otros estados.

Ejemplo de diagrama de colaboración Figura 6



Implementación

Pruebas

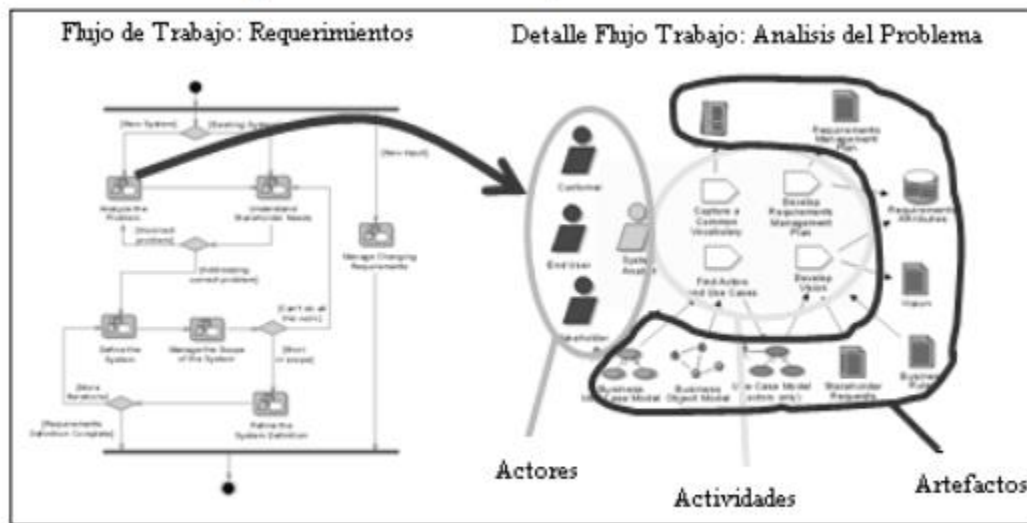
Despliegue

Gestión y configuración de cambios

Organización y elementos en RUP

Ya conociendo varias partes del RUP nos concentraremos ahora en los elementos que lo componen, entre estos se tienen: Flujos de Trabajo, Detalle de los Flujos de Trabajo, Actores, Actividades y Artefactos. En la figura 7 se muestran más claramente como se representan gráficamente cada uno de estos elementos y la interrelación entre ellos. Se puede observar que el Flujo de Trabajo de Requerimientos conlleva varios pasos, cada uno de estos pasos tiene asociado uno o varios actores, los cuales a su vez son los encargados de la ejecución de varias actividades, las cuales a la vez están definidas en artefactos o guías para su realización.

Figura 7



Actores o roles

Son los personajes encargados de la realización de las actividades definidas dentro de los flujos de trabajo de cada una de las disciplinas del RUP, estos actores se dividen en varias categorías: Analistas, Desarrolladores, Probadores, Encargados, Otros actores.

Jefe de Proyecto

Funciones:

- Planificar el proyecto
- Supervisar el desarrollo del proyecto
- Coordinar todas las etapas del proyecto
- Establecer un plan detallado
- Definir los plazos
- Servir como punto focal de comunicación
- Comunicar el estado del proyecto
- Informar sobre los avances

- Identificar posibles obstáculos
- Gestionar riesgos potenciales

Responsabilidades:

- Asegurarse de que el proyecto se desarrolle según los objetivos establecidos
- Designar tareas y responsabilidades al equipo
- Facilitar la colaboración entre stakeholders y clientes
- Comunicar de manera clara y concisa
- Evaluar y controlar el progreso del proyecto
- Tomar decisiones clave para el proyecto
- Asegurar que se cumplan los plazos
- Garantizar la entrega exitosa del software

Responsable de Investigación y Diseño

Funciones:

- Analizar las necesidades de los usuarios y stakeholders
- Diseñar la arquitectura y la estructura del software
- Crear diagramas de flujo y modelos de datos
- Desarrollar prototipos y wireframes de la interfaz de usuario
- Evaluar tecnologías y herramientas para el desarrollo
- Colaborar con el equipo de desarrollo en la implementación
- Mantenerse al tanto de avances en desarrollo de software

Responsabilidades:

- Traducir las necesidades de los usuarios en especificaciones técnicas
- Diseñar una arquitectura que cumpla con los objetivos del proyecto
- Crear representaciones visuales claras de la estructura del software
- Desarrollar prototipos de la interfaz de usuario para validación

- Aplicar principios de diseño de software efectivo

Responsable de Desarrollo

Funciones:

- Especificar los requisitos técnicos del proyecto
- Planificar la implementación del software
- Codificar y programar el software
- Definir e integrar componentes y módulos del software
- Realizar pruebas de unidad y pruebas de integración
- Asegurar la escalabilidad y eficiencia del software
- Colaborar con el equipo de diseño
- Capacitar a otros miembros del equipo en desarrollo

Responsabilidades:

- Garantizar que se satisfagan los requisitos técnicos del proyecto
- Elaborar un plan detallado de desarrollo
- Escribir código limpio y eficiente
- Integrar diferentes partes del software de manera coherente
- Realizar pruebas exhaustivas para garantizar la funcionalidad
- Optimizar el rendimiento y la escalabilidad del software.

Responsable de Pruebas

Funciones:

- Definir estrategias y planes de pruebas
- Diseñar casos de prueba y escenarios de prueba
- Ejecutar pruebas funcionales y de integración
- Identificar y documentar defectos y problemas
- Realizar pruebas de rendimiento y seguridad.
- Coordinar la ejecución de pruebas con el equipo de desarrollo
- Validar que el software cumple con los requisitos
- Proporcionar retroalimentación al equipo de desarrollo

- Garantizar la calidad y fiabilidad del software

Responsabilidades:

- Planificar y programar pruebas de acuerdo con el ciclo de desarrollo
- Crear casos de prueba que cubran todos los aspectos del software
- Ejecutar pruebas de manera sistemática y documentar resultados
- Reportar y priorizar defectos y problemas encontrados
- Asegurar que el software cumpla con los estándares de calidad.
- Proporcionar información sobre la estabilidad y calidad del software.

Administrador de la Base de Datos

Funciones:

- Diseñar la estructura y esquema de la base de datos
- Implementar políticas y medidas de seguridad para proteger la integridad y confidencialidad de los datos
- Crear modelos de datos y diagramas para representar la relación entre los datos
- Asegurarse de que las consultas de base de datos sean eficientes y de alto rendimiento
- Ajustar el rendimiento de la base de datos para manejar la carga de trabajo
- Brindar asesoramiento y soporte técnico al equipo de desarrollo en cuestiones de base de datos

Responsabilidades:

- Asegurar que el diseño de la base de datos sea coherente con los objetivos del proyecto de software
- Desarrollar y mantener modelos de datos precisos y actualizados

- Configurar y crear la base de datos de manera que sea compatible con el software
- Optimizar consultas para garantizar un rendimiento eficiente del software

Diseñador

Funciones:

- Crear la apariencia visual de la interfaz de usuario del software
- Desarrollar prototipos de la interfaz de usuario para visualizar y probar conceptos
- Elegir iconos, imágenes y otros elementos visuales que se utilizarán en el software
- Garantizar que el diseño se mantenga coherente en todo el software

Responsabilidades:

- Crear una interfaz de usuario atractiva y funcional que satisfaga las necesidades del usuario
- Diseñar la experiencia del usuario de manera que sea intuitiva y efectiva
- Desarrollar prototipos para validar y refinar conceptos de diseño
- Elaborar wireframes detallados que guíen la implementación del diseño
- Seleccionar y adaptar elementos visuales para mantener la coherencia de marca
- Asegurarse de que la interfaz sea receptiva y se adapte a diferentes dispositivos
- Trabajar de cerca con los desarrolladores para garantizar una implementación precisa

Investigador

Funciones:

- Recopilar y analizar los requisitos del proyecto de software
- Identificar problemas y desafíos que el software debe abordar
- Recopilar datos y estadísticas relevantes para el desarrollo del software
- Realizar investigaciones de usuarios para comprender sus necesidades y preferencia
- Evaluar la viabilidad técnica y financiera del proyecto

Responsabilidades:

- Asegurarse de que se comprendan y documenten adecuadamente los requisitos del proyecto
- Identificar y documentar claramente los problemas que el software debe resolver
- Recopilar datos relevantes para respaldar la toma de decisiones durante el desarrollo
- Ayudar a diseñar el software teniendo en cuenta las preferencias y necesidades de los usuarios
- Ayudar a determinar si el proyecto es factible desde una perspectiva técnica y financiera

Programador Senior

Funciones:

- Escribir código de alta calidad y eficiente para el software
- Contribuir al diseño técnico del sistema y la arquitectura
- Diagnosticar y resolver problemas técnicos complejos
- Brindar orientación y mentoría a programadores junior

- Realizar revisiones de código para garantizar la calidad y coherencia del software
- Crear documentación técnica clara y detallada

Responsabilidades:

- Desarrollar código de alta calidad que cumpla con los estándares de la industria y los requisitos del proyecto
- Contribuir al diseño técnico del software, incluyendo decisiones de arquitectura y tecnología
- Identificar y abordar problemas de rendimiento para mejorar la eficiencia del software
- Escribir pruebas de unidad sólidas y contribuir a la estrategia de pruebas en general
- Mantenerse actualizado sobre las últimas tendencias tecnológicas y evaluar su aplicabilidad en el proyecto
- Asistir a reuniones de equipo y contribuir con ideas y actualizaciones de progreso

Programador Junior

Funciones:

- Escribir código siguiendo las pautas y estándares establecidos
- Identificar y solucionar problemas y errores en el código
- Documentar el código y los cambios realizados de manera clara y concisa
- Ayudar en la identificación y resolución de problemas técnicos sencillos

Responsabilidades:

- Trabajar en estrecha colaboración con otros miembros del equipo de desarrollo

- Aprender nuevas tecnologías y técnicas de programación
- Documentar el código y los cambios realizados de manera que sea fácilmente comprensible para otros
- Asistir a reuniones de equipo y contribuir con ideas y actualizaciones de progreso

Tester

Funciones:

- Diseñar estrategias y planes de prueba para el software
- Elaborar casos de prueba detallados basados en requisitos y especificaciones
- Identificar problemas y registrarlos en un sistema de seguimiento de defectos
- Evaluar la usabilidad del software desde la perspectiva del usuario final
- Realizar pruebas de rendimiento para evaluar el comportamiento del software bajo diferentes cargas

Responsabilidades:

- Crear documentación detallada de los resultados de las pruebas
- Trabajar estrechamente con desarrolladores y diseñadores para resolver problemas
- Mantener un enfoque constante en la calidad del software a lo largo del ciclo de desarrollo.

Artefactos

Los artefactos son el resultado parcial o final que es producido y usado por los actores durante el proyecto. Son las entradas y salidas de las actividades, realizadas por los actores, los cuales utilizan y van produciendo estos artefactos para tener guías. Un artefacto puede ser un documento, un modelo o un elemento de modelo.

Conjuntos de artefacto

Se tiene un conjunto de artefactos definidos en cada una de las disciplinas y utilizadas dentro de ellas por lo actores para la realización de las

mismas, a continuación, se definen cada una de estas categorías o grupos de

artefactos dentro de las disciplinas del RUP:

a) Modelado del negocio

- ✓ Diagrama de flujo
- ✓ Diagrama de caso de uso

b) Requerimientos

- Diagrama de caso de uso.
- Diagrama de flujo
- Diagrama de secuencia.
- Diagrama de componentes.

c) Análisis y diseño del sistema

- Diagrama de despliegue.
- Diagrama de componentes
- Diagrama de Paquetes.
- Diagrama de clase.

- Diagrama de estado.

d) Implementación

- Diagrama de clase.
- Diagrama de componentes.
- Diagrama de despliegue.

e) Pruebas

- Diagrama de secuencia.

f) Despliegue

- Diagrama de Estados de Transición