Facultad de Estadística e Informática

Licenciatura en Ingeniería en Sistemas y Tecnologías de la información

Documento de diseño

CARRETO BARRIENTOS JOSÉ MANUEL

GARCIA ARMAS SERGGI RAFAEL

HERNANDEZ PIEDRA JOSE MANUEL

TORRES SARMIENTO JORGE JESÚS

Xalapa, Ver., 18 de Junio de 2025

Índice

1. Introducción 3
   1.1 Propósito 4
   1.2 Alcance 4
   1.3 Audiencia 5
   1.4 Definiciones 5
2. Descripción general 6
   Arquitectura del sistema: 6
3. Diseño funcional 7
   3.1 Diagrama de caso de uso 7
4. Componentes del sistema 8
   4.1 Diseños de clases 8
   4.2 Paquetes 9
5. Componentes dinámicos 12
   5.1 Interacciones 12
6. Diseños de datos 13
7. Diseño de la interfaz de Usuario 15
8. Restricciones y Suposiciones 16
9. Anexos 17
   Anexo A: Diagramas de despliegue 17
   Anexo B: Diagramas de casos de uso 18
   Anexo C: Tabla para los procesos 19
   Anexo D: Diagrama de clases 20
   Anexo E: Diagrama de paquetes 21
   Anexo F: Diagrama de entidad-relación 22

1. Introducción

Hoy en día, la manera en cómo se promocionan las empresas ha evolucionado con el pasar de los años, y ellos se han dado cuenta que una de las tecnologías que ha tomado más impulso en estas últimas décadas son el internet y las páginas web. Estas mismas, han revolucionado la manera en cómo obtenemos información y de cómo nos avisan de algunos productos más rápido y a la orden del día. Con eso en mente, empresas de renombre han cambiado la manera en cómo ofrecían sus productos y de cómo pueden llegar a un mayor publico alrededor del mundo. Pero para las empresas pequeñas y medianas (como las reposterías), no lo han implementado de manera efectiva en sus locales.

Es común ver cómo estas gestionan sus insumos, inventarios y ventas de forma manual o con sistemas genéricos que no se ajustan a sus necesidades reales. Esto, lejos de ayudar, puede generar errores, pérdidas de dinero y procesos poco eficientes. Imagina llevar el conteo de harina, huevos o azúcar en libretas, o calcular las ventas del día sin una herramienta que simplifique el trabajo. Los riesgos de equivocaciones son altos, y el tiempo invertido podría aprovecharse mejor.

Por eso, este proyecto nace de una necesidad concreta: ayudar a una pastelería local a modernizar su control interno mediante un sistema interactivo y personalizado. La idea no es imponer tecnología por moda, sino crear una solución práctica que se adapte a su día a día (desde la gestión de recetas hasta el seguimiento de ventas), haciendo su operación más ágil y precisa.

1.1 Propósito

El propósito del proyecto es hacer un sistema en donde se lleve un control de inventario de los ingredientes que se ocupan en una repostería, adema s de poder tener una gestión de productos que se venden en ella y que puede usar para llevar un control de las ventas, fecha de elaboración, fecha de caducidad, cuantos quedan disponibles y de eliminar los postres que ya están caducados o que ya no están disponibles.

El cliente quiere que el sistema puede ser usado de manera fácil, que sea intuitiva para que los usuarios no necesiten estar llamando a un supervisor para explicar cómo funciona, adema s de no ser difícil de usar, con acciones que sean simples y funcionales para no dar una mala experiencia con él y dejen de usarlo, por eso pide que sea de funcional, que no sea difícil de entender y que pueda tener acceso con un usuario y contraseña que se almacena en una base de datos y si el empleado es nuevo se registre en el sistema para poder usarlo.

1.2 Alcance

El alcance del sistema de gestión de la repostería puede variar dependiendo si la misma se encuentre de manera local con solo una sucursal o en caso de contar con más establecimientos, se puedan enviar los datos de cada venta hacia la zona en donde se encuentra la sede principal y con ello, tener un control de cada sucursal que se encuentre operando para poder checar el manejo de cada una.

1.3 Audiencia

Para este trabajo, se espera que la audiencia principal que va a tomarse en cuenta para esta primera versión es para los empleados y el personal (incluye el gerente de área, trabajadores de mostrador, cocineros, entre otros). Esto anterior, puede variar dependiendo de cuantos empleados están trabajando actualmente en dicha empresa o para este caso repostería. Se espera que en un futuro pueda ser usada para que los clientes puedan hacer pedidos en línea y puedan seleccionar si su pedido sea entregado en mostrador o si quieren que se lo lleven a su domicilio, además del método de pago que usen (Efectivo, tarjeta de débito o crédito o incluso usar transferencia).

1.4 Definiciones

- **Inventario:** Conjunto de ingredientes y productos disponibles en la repostería. Incluye materias primas (harina, azúcar, leche, etc.) y productos terminados (pasteles, galletas, postres individuales).
- **Producto:** Cualquier bien elaborado por la repostería que esté disponible para su venta.
- **Sistema:** Aplicación web que permite gestionar el inventario, productos, ventas y usuarios de la repostería.
- **Usuario:** Persona que tiene acceso al sistema mediante un nombre de usuario y contraseña. Puede ser un empleado nuevo o actual.
- **Venta:** Registro de la transacción donde un cliente adquiere uno o más productos.
- **Fecha de elaboración:** Día en que fue creado un producto para la venta.
- **Fecha de caducidad:** Día límite en que un producto puede ser vendido o consumido. Una vez pasada, debe eliminarse del inventario.
- **Sucursal:** Establecimiento físico de la repostería. Puede haber más de una, en cuyo caso se coordinarán mediante una sede central.
- **Base de datos:** Almacenamiento estructurado de la información del sistema, como usuarios, productos, inventario y ventas.

2. Descripción general

Arquitectura del sistema:

El sistema propuesto está basado en una arquitectura web utilizando el framework **ASP.NET Core WebApp con Razor Pages**. Esta arquitectura permite una separación clara entre las capas del sistema, facilitando el mantenimiento y la escalabilidad de la aplicación.

El usuario interactúa directamente con las **Razor Pages**, que funcionan como la interfaz gráfica o **frontend** del sistema. Desde ahí, las solicitudes del usuario (como registrar una venta, consultar productos, actualizar inventario, etc.) son enviadas al **backend**, donde se procesan mediante los siguientes componentes:

- **PageModel:** Actúa como intermediario entre la interfaz y la lógica de negocio. Se encarga de manejar eventos y solicitudes HTTP provenientes del usuario.
- **Services:** Contienen la lógica de negocio. Aquí se define el comportamiento del sistema, como verificar caducidad de productos o calcular cantidades disponibles.
- **DAO (Data Access Object):** Se encarga de gestionar la comunicación directa con la base de datos. Esta capa abstrae las operaciones CRUD.
- **DbContext (Entity Framework Core):** Componente que permite interactuar con la base de datos SQL a través de objetos de entidad. Gestiona las consultas y persistencia de datos de forma estructurada.

Toda la información del sistema (usuarios, productos, ventas, inventario) se almacena en una **base de datos SQL Server**, la cual puede estar alojada localmente o en un servidor según se necesite.

El diagrama de despliegue **(Anexo A)** muestra cómo los distintos componentes del sistema interactúan entre sí y con el hardware, permitiendo una visualización clara de la distribución y comunicación dentro de la aplica

3. Diseño funcional

3.1 Diagrama de caso de uso

En este modelo, lo primero que se hizo es identificar a los actores que estarán presentes, además de los casos de uso, los limitantes del sistema, así como también la asociación que puede estar en cada uno de los casos de uso. Estos casos pueden estar asociados por comunicación, generalización, inclusión o extensión para cada caso.

- **Actor:** Son las personas que estarán involucradas en dicha práctica o actividad
- **Casos de uso:** son los elementos que están conectados a los atores.
- Límite de sistema: para este elemento, limita el espacio que da para cada caso de uso.
- **Asociación de comunicación (Línea simple):** actor de un solo caso
- Inclusión (Línea punteada con flecha): Un caso de uso simple incluye a otro.
- Extensión (Línea punteada con flecha mas delgada): Un caso de uso puede opcionalmente extenderse.
- Generalización (flecha vacía): Un actor o caso de uso hereda de otro.

Anexo B.- Diagrama de casos de uso

4. Componentes del sistema

4.1 Diseños de clases

Las clases que se crear a partir de la idea principal son: Usuario, Venta, Producto, Método de Pago, Categoría, Ingrediente y Receta. Cada una de ellas tiene sus atributos y métodos que los representan, además de que cada una está relacionada entre sí para que puedan usar la información que tiene cada uno en el momento en que el usuario este usando la interfaz para interactuar con ellos.

Las relaciones que tiene cada clase son las siguientes:

Usuario -> Venta: De un usuario a varias ventas de productos

Venta -> Método de Pago: De una venta a un método de pago

Venta -> Producto: De cero – varias ventas a una o más productos

Producto -> Categoría: De un producto a cero o una categoría de producto

Receta -> Categoría: De un postre puede tener de 0 a una categoría.

Receta -> Ingrediente: Una receta pude tener 1 a más ingredientes

Ingrediente -> Categoría: Un ingrediente puede aparecer de 0 a 1 vez en alguna categoría

En el **Anexo D** se puede ver el diagrama con las clases y sus relaciones que tienen cada una de ellas.

4.2 Paquetes

En el diagrama de paquetes (**Anexo E**) muestra la estructura arquitectónica del sistema *ProyectPastel*, su objetivo es modelar de forma clara y ordenada los distintos módulos y componentes del sistema, agrupándolos por capas funcionales, de acuerdo con una arquitectura **en capas**.

Esta versión del diagrama se ha simplificado visualmente para facilitar su lectura y comprensión general, sin perder el enfoque modular del sistema.

Capas Principales del Sistema

1. Application Layer

- Contiene el archivo Program.cs, punto de entrada principal de la aplicación.
- Incluye un componente de configuración (appsettings.json) etiquetado como “Configuración”.
- Esta capa tiene la responsabilidad de **inicializar el sistema**, configurar servicios y preparar la ejecución general de la aplicación.
- Tiene una **dependencia directa hacia la capa de presentación y la de servicios**, ya que las inicializa y configura.

2. Application Layer (UI)

- Representa la capa de interacción con el usuario.
- Está dividida internamente en dos grupos importantes:
    - **Pages Comunes**, que agrupa vistas como Index, InicioSesion y Error.
    - **Módulo Inventario**, con páginas específicas para crear, editar e indexar productos de inventario.

- Esta capa se comunica directamente con los **servicios de la lógica de negocio** y **usa recursos compartidos del sistema**, como archivos JS, CSS e imágenes.

3. Application Layer (Services)

- Aquí se encuentran los servicios que contienen la **lógica del negocio**.
- Componentes como InventarioService y UserService representan operaciones relacionadas a inventario y usuarios.
- Esta capa recibe solicitudes desde la UI y las procesa de acuerdo a las reglas del negocio.
- También **utiliza DAOs** para acceder a los datos, y **opera sobre entidades del dominio**.

4. Domain Model (Entities)

- Agrupa los **modelos de datos** utilizados por todo el sistema, incluyendo las entidades inventario, postres, usuarios y ventas.
- También incluye el **Contexto de Base de Datos** (proyecto_pastelContext), que actúa como punto de conexión entre las entidades y la base de datos utilizando Entity Framework Core.
- Es utilizado tanto por los servicios como por los DAOs para representar la lógica del dominio del sistema.

5. Data Access Layer (DAOs)

- Los DAOs **acceden directamente a las entidades del dominio** y a los datos almacenados en la base de datos.
- Son responsables de realizar operaciones CRUD y **aislar la lógica de acceso a datos** del resto de la aplicación.

6. Infrastructure & External Libraries

- Aquí se muestran dependencias externas esenciales para el funcionamiento del sistema:
    - Entity Framework Core: ORM utilizado para mapear objetos a una base de datos relacional.
    - MySQL Connector: Driver que permite la comunicación entre el sistema y una base de datos MySQL.
- Estas librerías son usadas por el contexto de base de datos y por los DAOs.

7. Archivos SQL

- Representan scripts de base de datos como proyecto_pastel.sql y proyecto_pastel1.sql.
- Están vinculados al sistema como **recursos de gestión de datos**.
- Son manejados principalmente por la capa de acceso a datos.

Relaciones entre Paquetes (Dependencias)

- Application Layer **usa** la UI y configura los servicios.
- UI **llama** a los servicios de negocio.
- Services **operan sobre** entidades y **usan** DAOs.
- DAOs **acceden** a las entidades y usan bibliotecas externas para conectarse a la base de datos.
- UI **incluye** recursos estáticos como CSS, JS e imágenes.
- Entities depende de la infraestructura para la conexión con el ORM y la base de datos.

Los **archivos SQL** están relacionados como elementos de soporte para la gestión de datos del sistema.

5. Componentes dinámicos

5.1 Interacciones

El comportamiento dinámico del sistema está representado mediante uns **Diagrama de Secuencia (Anexo E)**, que describe cómo interactúan los distintos objetos involucrados en uno de los casos de uso clave: **el registro de un pedido por parte del empleado**.

Caso de uso: Registro de Pedido

El flujo inicia cuando un **empleado** ingresa al sistema y finaliza cuando se confirma o se rechaza el pago de un pedido. A continuación, se detallan las interacciones clave entre los objetos involucrados:

1. **Inicio de sesión**
    - El empleado introduce sus credenciales en el sistema.
    - El sistema envía las credenciales al controlador para validarlas.
    - El controlador consulta la base de datos para buscar al usuario.
    - La base de datos devuelve el resultado (válido o inválido).
    - El sistema muestra las opciones disponibles solo si la autenticación es válida.
2. **Selección de módulo**
    - Una vez dentro, el empleado selecciona el módulo "Venta".
    - El sistema activa el módulo de venta e inicia el proceso.
3. **Registro del pedido**
    - El empleado procede a registrar el pedido.
    - El sistema solicita el método de pago.
    - El empleado introduce el método de pago, que es enviado al controlador para su validación.
    - Si el pago es válido, se registra en la base de datos.
    - Si el pago es aceptado, el sistema confirma y entrega el pedido.
    - Si el pago es rechazado, se notifica el error.
    - En caso de fallo en la autenticación, se muestra un mensaje de error.

Este flujo refleja una secuencia lógica y ordenada en la cual el sistema garantiza la integridad del proceso de ventas, asegurando que solo personal autorizado pueda acceder y realizar operaciones, y que cada venta quede debidamente registrada.

6. Diseños de datos

Para nuestra base de datos del programa se crearon 7 tablas las cuales son:

1. detalle_venta: Contiene un id propio para su identificación, un id venta el cual se obtiene desde la tabla venta, un id postre el cual se obtiene desde la tabla de postre para referencia el postre que se vendii, una variable cantidad para indicar cuantos postres se vendieron, una variable precio_unitario la cuál determina el precio por unidad, un subtotal que es coste de todo lo vendido.

2. Ingredientes_recetas: un identificador personal el cual es id_ingredientes_recetas, dos identificadores referenciadas de las tablas receta y la tabla ingredientes los cuales son id_receta y id_ingrediente, la cantidad que refiere a la cantidad de ingredientes necesarios para la receta y una variable unidad que se refiere a si es en kg, g, ml, l etc.

3. Inventario: Contiene un identificador personal para la tabla la cual es id_ingrediente, la variable nombre la cual se refiere al nombre del ingrediente, la variable cantidad la cual se refiere a la cantidad necesaria del ingrediente y la variable unidad que se refiere a si es en kg, g, ml, l etc.

4. Postres: la cuál contiene su identificador personal para la tabla la cual es id_postres, la variable nombre que contiene el nombre del postre, una variable descripción la cual contiene la descripción del postre, la variable precio_base que es el precio original del postre, una variable fecha_creacion la cuál contiene la fecha de creación del postre en la base de datos.

5. Recetas: la cual contiene el identificador personal de la tabla la cual es id_receta, un identificador referenciado de otra tabla la cual es id_postre para identificar de que postre es la receta, una variable descripción la cual contiene los pasos a seguir de la receta.

6. usuarios: contiene un identificador ede la tabla la cual es id_usuarios, la variable nombre la cual contiene el nombre del usuario registrado en la base de datos, la variable correo que contiene el correo con el cual se registró el usuario, la variable contraseña la cual contien la contraseña que se creo al momento de registrarse y la variable fecha_registro la cual contiene el dia el cual el usuario creo su cuenta.

7. ventas: la cuál contiene el identificador personal de la tabla la cual es id_venta, un identificador referenciado de otra tabla el cual es id_usuarios el cual sirve oara saber que usuario es el que esta haciendo la compra del producto y por último la variable fecha_venta la cuál contiene la fecha en la cual se realizó la venta del producto.

El Diagrama de entidad-relación se puede encontrar en el **Anexo F.**

7. Diseño de la interfaz de Usuario

Pantalla	Descripción
Login	Página de autenticación. Permite ingresar al sistema mediante usuario y contraseña.
Inicio	Panel principal tras iniciar sesión. Muestra estadísticas, reportes y accesos rápidos a módulos.
Privacidad	Sección donde se muestra la política de privacidad del sistema, uso de datos y condiciones.
Ventas	Permite registrar pedidos, seleccionar productos y procesar métodos de pago.
Inventario	Muestra el inventario de productos e ingredientes. Permite filtrar por disponibilidad, caducidad y tipo.
Crear/Editar Inventario	Formularios para registrar nuevos productos o modificar los existentes (nombre, cantidad, fechas, etc.).
Gestión de usuarios	Sección para registrar nuevos empleados, editar sus datos o eliminar accesos.
Detalle de elementos	Muestra información específica de productos, ventas o usuarios según se requiera.

8. Restricciones y Suposiciones

Documentación de cualquier limitación

- **Falla al momento de cargar las imágenes a la página**
  Para este caso, al momento de añadir algunas imágenes a la página, estas no se cargan como corresponden. Aparece un icono que muestra que esta la imagen, pero no muestra la imagen en sí

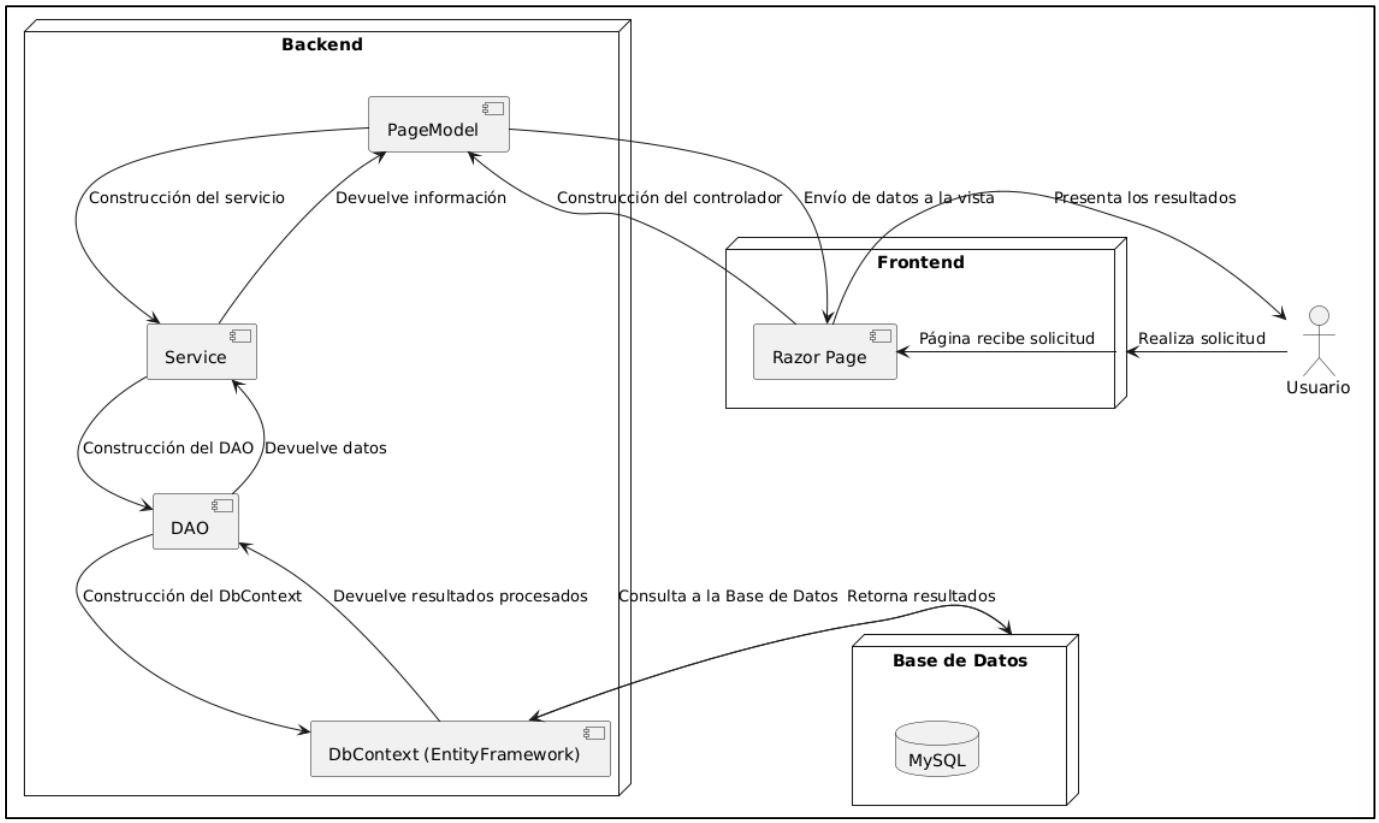
- **Fallas técnicas al momento de probar el inicio de sesión**
  Al momento de conectar la base de datos con el inicio de sesión, se produce un error arraigado con la selección de usuario root de la base de datos.

- **Uso limitado de algunos elementos al momento de crear las ventanas de la página**
  Esto debido a problemas al momento de crear algunas, se presentaron problemas de configuración, esto hizo que se ocurriera un plan para poder añadirlo más adelante algunas funciones que no se pudieron implementar en esta primera versión.

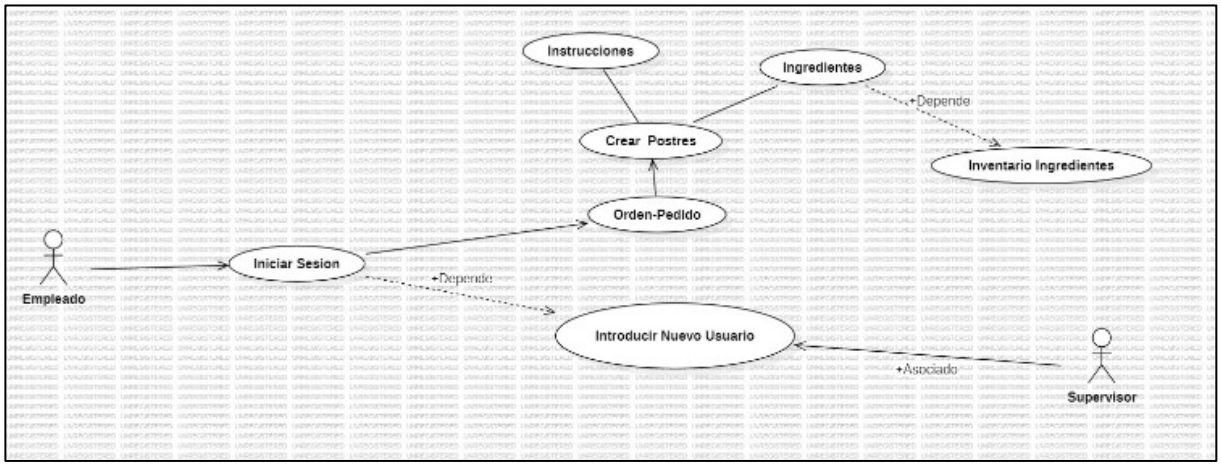
9. Anexos

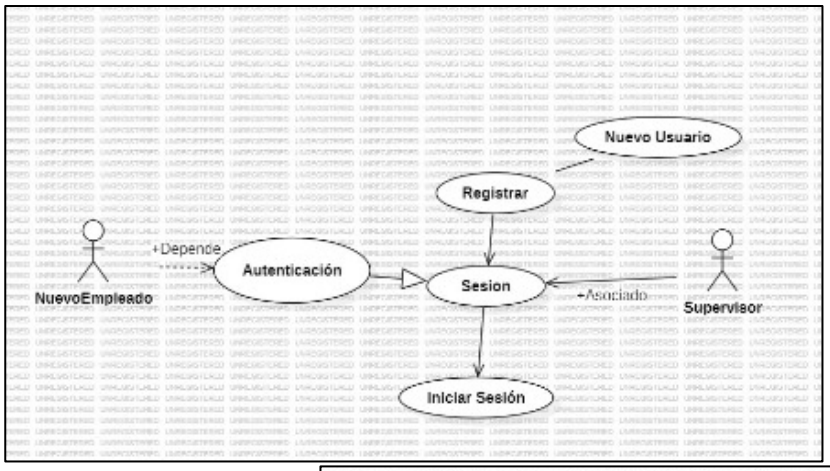
Material de referencia, glosario o documentación de soporte

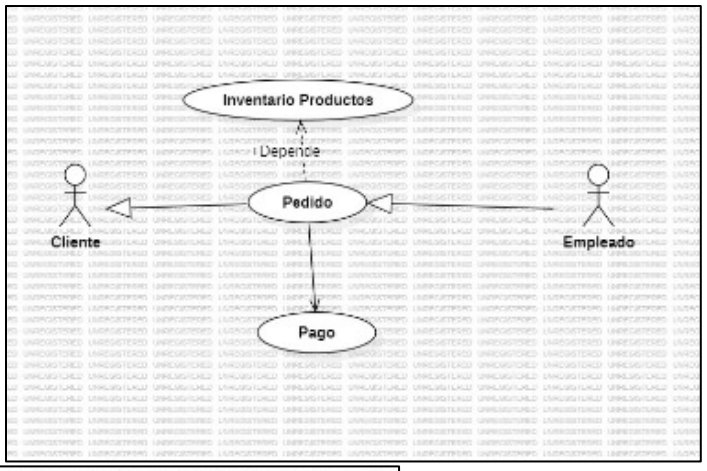
Anexo A: Diagramas de despliegue

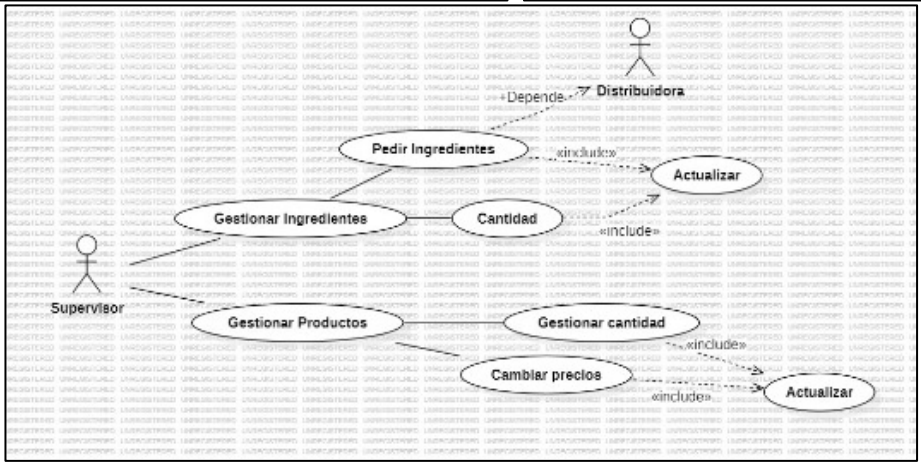


Anexo B: Diagramas de casos de uso







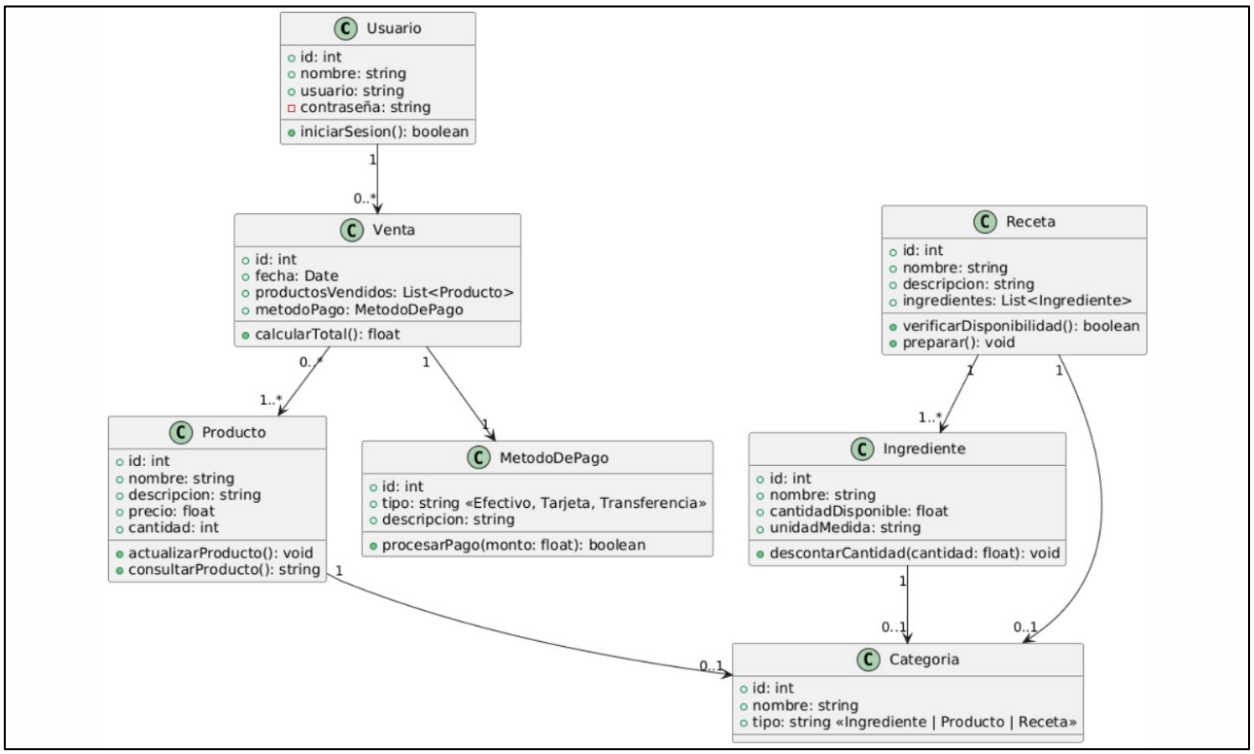


Anexo C: Tabla para los procesos

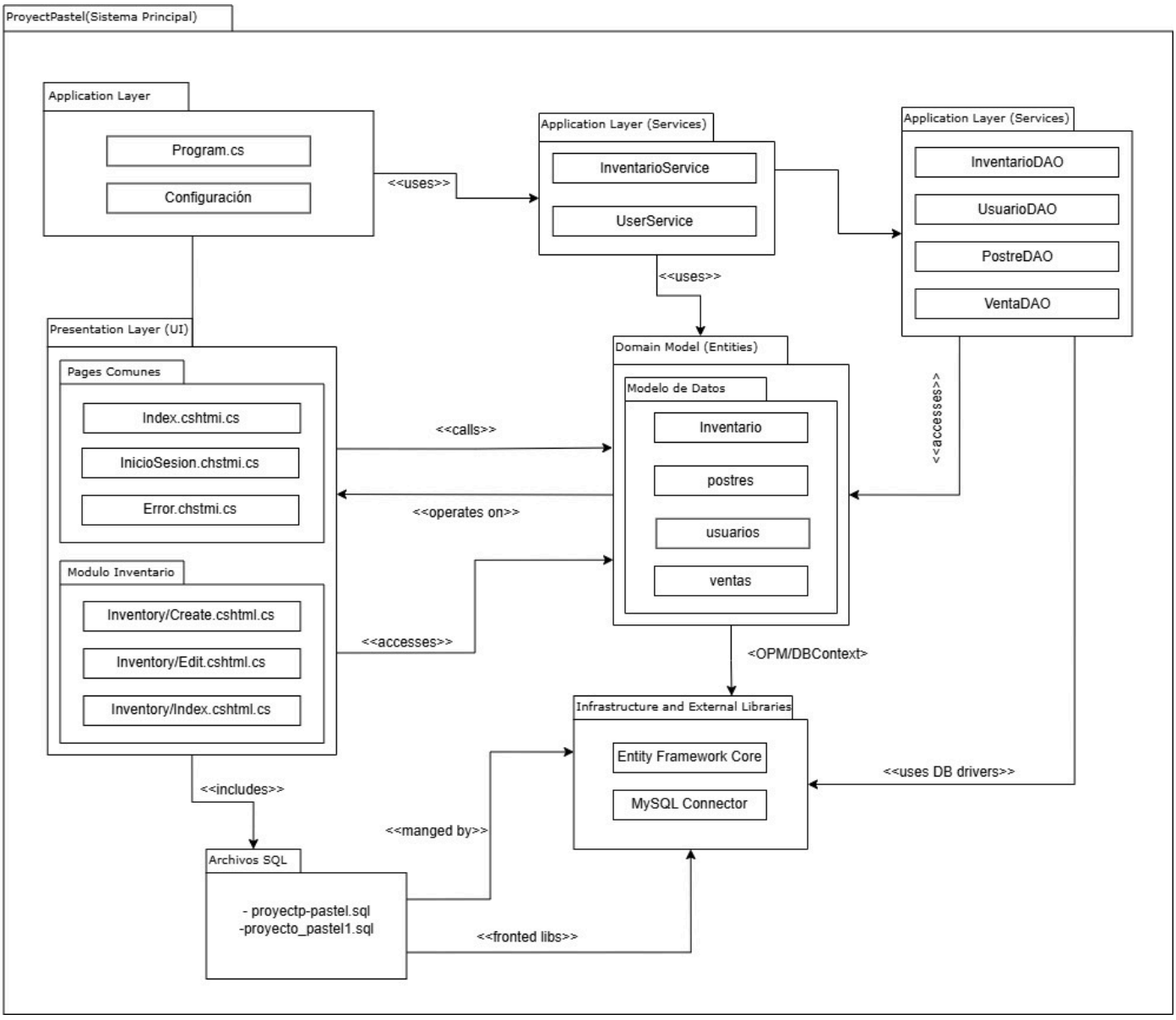
Tabla resumen de procesos

Proceso de Negocio	Objetivo	Actores	Actividades Clave
Gestión de Recetas	Control de recetas según ingredientes disponibles	Administrador, Empleado	Agregar, editar, consultar, eliminar, verificar ingredientes
Gestión de Ingredientes/Productos	Control de inventario	Administrador, Empleado	Registrar ingredientes, modificar productos, alertas
Gestión de Ventas	Registro de ventas y stock	Cajero, Cliente	Registrar venta, disponibilidad, comprobante
Gestión de Usuarios	Control de accesos	Administrador, Empleados	Autenticación, permisos, eliminación
Gestión de Reportes	Consultar estadísticas e inventario	Administrador	Ver ventas, aplicar filtros, identificar tendencias

Anexo D: Diagrama de clases



Anexo E: Diagrama de paquetes



Anexo F: Diagrama de entidad-relación