

Sistema de Gestión y Control de Panadería

Lautaro Eberhardt

Prácticas Profesionalizantes I

Nicolas Kappes

Índice

Introducción:	5
Fundamentación:	5
Alcance y límites:	5
Relevamiento de información:	6
Requisitos funcionales:	7
Requisitos no funcionales:	8
Diagrama de contexto:.....	9
Diagrama de casos de uso (Gerente):	10
Diagrama de casos de uso (Empleado):	10
Casos de uso:	11
1. Inicio de sesión:.....	11
1.2 Verificación de contraseña:.....	11
2. Pedidos:.....	12
3. Cargar pedidos:	12
3.2 Validar disponibilidad de producto:.....	13
4. Modificar pedidos:	14
5. Borrar pedidos:	15
6. Visitar historial:	16
7. Carga de Materia Prima:	16
7.1 Consultar stock materia prima:.....	17

7.2 Cargar materia prima nueva:	18
7.2.1 Eliminar materia prima:	18
8. Carga de Producto:.....	19
9. Modificar producto:	20
9.1 Borrar producto:	20
Diagrama de entidad-relación:	22
Diagrama de clases:	23
Arquitectura del sistema.....	23
Entrevista:	26
Conclusión.....	27
1. Prototipo de Interfaces	28
1.1 Herramienta utilizada:.....	28
Pantalla 1: Login	28
Pantalla 2: Pantalla principal	29
Pantalla 3: Carga de pedidos	30
Pantalla 4: Carga de materia prima	31
Pantalla 5: Carga de materia prima nueva	31
Pantalla 6: Eliminar materia prima	32
Pantalla 7: Carga de producto	33
Pantalla 8: Eliminación de producto	33

Pantalla 9: Consultar Stock	34
Pantalla 10: Modificar Producto	35
2. Estructura de la Base de Datos	36
2.1 Modelo Lógico.....	36
2.2 Modelo Físico	37

Introducción:

En este trabajo detalla el desarrollo de un sistema de gestión y control de pedidos para una empresa familiar de panificación y repostería. Actualmente, la empresa utiliza un método tradicional para recibir y gestionar pedidos, esto ha generado ineficiencias debido a la falta de control y organización. Con el fin de optimizar la gestión, el sistema propuesto permitirá registrar, gestionar y controlar los pedidos y materias primas de manera automatizada, brindando una solución integral a las problemáticas actuales.

Fundamentación:

Con el fin de optimizar la gestión, el sistema propuesto permitirá registrar, gestionar y controlar los pedidos de manera automatizada, brindando una solución integral a las problemáticas actuales. El sistema ofrecerá una interfaz intuitiva que permitirá al usuario registrar los pedidos de manera rápida y visual, asignar estados a cada pedido (encargado, realizado o cancelado), y llevar un control detallado de los pedidos y clientes, entre otras funciones.

Este proyecto nace de mi interés personal, gracias a mi experiencia en el sector de la panificación y repostería, lo que ha permitido identificar de manera precisa las necesidades operativas del negocio. Se espera que la implementación de este sistema no solo reemplace el método actual, sino que además mejore significativamente la organización interna, facilitando la gestión de pedidos y brindando un mayor control sobre las operaciones diarias.

Alcance y límites:

El sistema podrá realizar las siguientes acciones:

Registrar pedidos en una base de datos

Tendrá una interfaz creativa e intuitiva con imágenes de los productos de la panadería para seleccionar

Asignarle un estado al pedido (encargado, realizado, cancelado)

Asignarle a cada pedido su cliente

Registro de cantidad de pedidos en un mes (...año, ...semana, etc...)

Cargar Productos

Cargar materia prima para control de stock

Objetivos:

Mi objetivo general es realizar un sistema para llevar un mejor control y organización de una panadería mediante el cargado de pedidos mediante un usuario. De esta manera lograr que la empresa deje de utilizar el sistema actual y tenga un mejor entendimiento de sus ganancias y pedidos para próximos eventos.

Resultados esperados:

Lograr un sistema efectivo y amigable para la empresa, que les sirva para su función, reemplace el sistema actual de manera efectiva y perdure en el tiempo.

Como resultado esperado personal, me gustaría que el sistema sea recomendado por parte de la empresa ya sea para un proyecto parecido o bien me recomiende para otro campo.

Relevamiento de información:

Resumen:

La empresa familiar sobre la que trabajaré actualmente tiene un sistema de pedidos muy arcaico, se basa en el pedido de los clientes vía WhatsApp y luego trasladar esos pedidos a una pizarra para que así los panaderos y reposteros comiencen a trabajar.

Este método de trabajo tiene muy poco control, ya que al ser por WhatsApp si se realiza una gran cantidad de pedidos puede llegar a quedar algún mensaje perdido o abrir el mensaje y nunca llegar a la pizarra.

Mi solución es crear un sistema de gestión y control de pedidos, este sistema contara con una carga de pedidos mediante un usuario, entonces al momento de que realicen un pedido este pueda ser cargado en el sistema, seleccionando el producto como un ítem y seleccionar la cantidad. También podrá seleccionarse la fecha del pedido tipo calendario, y tendrá recordatorio con 1 día de anticipación, Por otra parte, también se podrá cancelar pedidos. También contará con un apartado para cargar materia prima para control de stock.

Tipo de estado de pedido: Cada pedido tendrá un tipo de estado, este tendrá un casillero a la derecha de la descripción del pedido y existen 3 tipos de estados encargado, cancelado y realizado.

Requisitos funcionales:

Inicio:

Permitir el registro mediante contraseña.

Registro y gestión de pedidos:

Permitir el registro de pedidos con información como nombre, descripción, precio, cantidad, etc.

Permitir la edición y eliminación de pedidos.

Validar la disponibilidad de productos.

Generar reportes de pedido.

Mantener un historial de pedidos.

Cargar materia prima

Consulta de pedidos pasados:

Permitir al usuario realizar consultas sobre pedidos pasados

Permitir el seguimiento y actualización del estado del pedido (encargado, realizado y cancelado).

Requisitos no funcionales:

Usabilidad:

Interfaz de usuario intuitiva y fácil de usar.

Respuesta rápida a las acciones del usuario.

Seguridad:

Autenticación y autorización de usuarios.

Cifrado de datos sensibles como información de clientes y pagos.

Disponibilidad:

Disponibilidad del sistema las 24 horas

Escalabilidad:

Capacidad de manejar un crecimiento en el número de clientes, productos y pedidos.

Mantenibilidad:

Documentación técnica detallada.

Facilidad para realizar cambios y actualizaciones en el sistema.

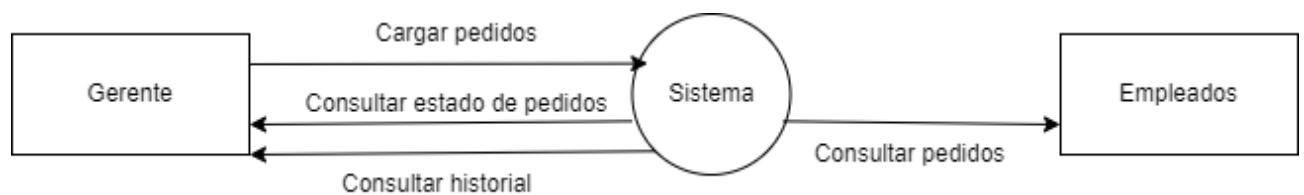
Diagrama de contexto:

Diagrama de casos de uso (Gerente):

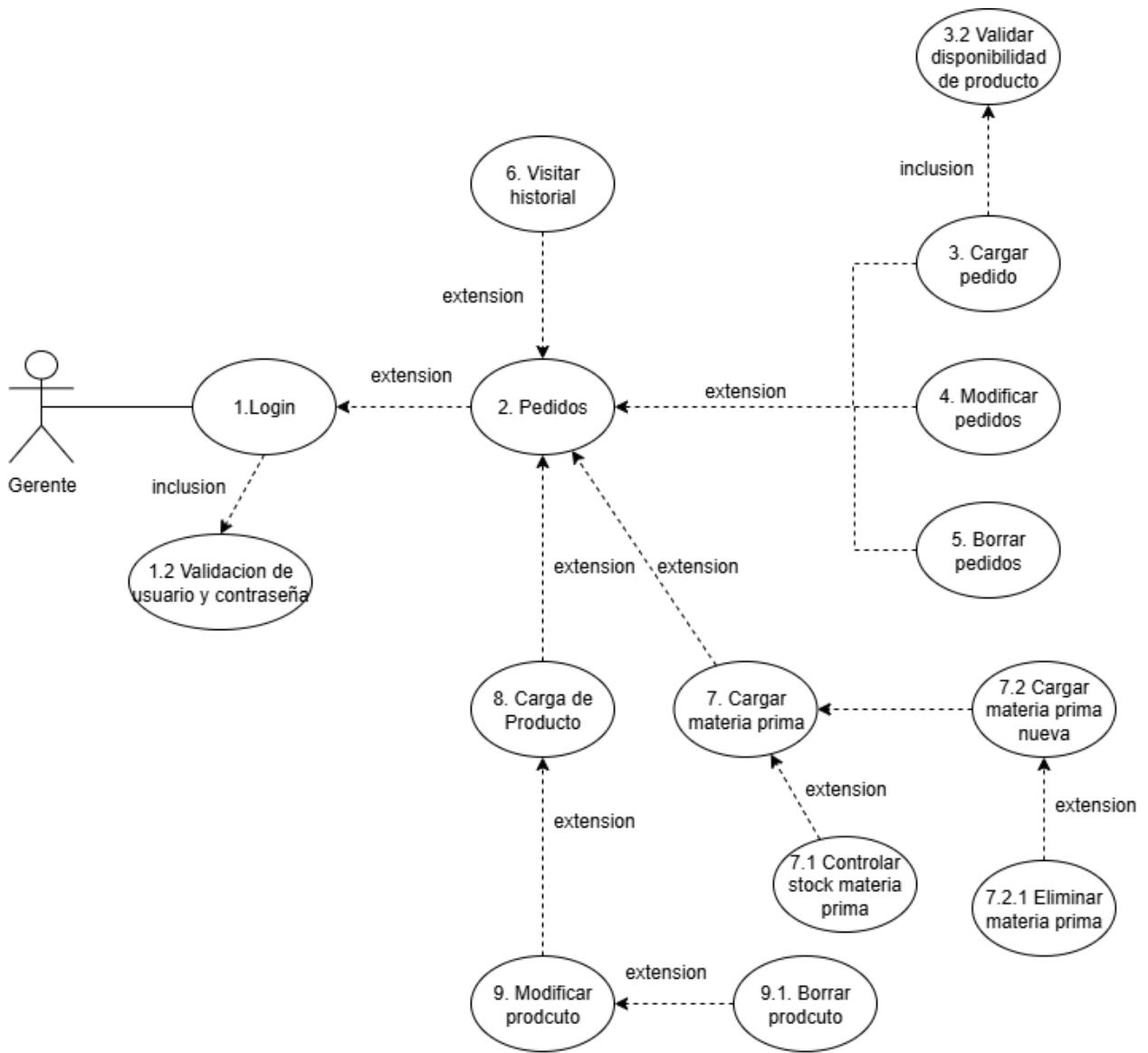
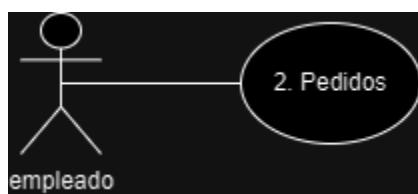


Diagrama de casos de uso (Empleado):



Casos de uso:

1. Inicio de sesión:

Descripción general del caso de uso: Este caso de uso se ingresa contraseña para que el gerente pueda ingresar al login.

Actor Principal: Gerente

Precondiciones:

1-Gerente introduce contraseña

2- Gerente presiona botón de “ingresar”

Postcondiciones:

1. Si contraseña correcta Sistema ingresa a pantalla principal

Relaciones:

1.2 Verificación de contraseña:

Descripción general del caso de uso: Este caso de uso verifica si la contraseña ingresada es correcta buscando el usuario en la base de datos.

Actor Principal: Gerente

Precondiciones:

1 - Sistema verifica si contraseña es correcta consultando la base de datos.

2 – Sistema encuentra usuario de gerente en base de datos.

Caminos Alternativos:

1- Sistema verifica que la contraseña es incorrecta.

Relaciones: Incluido en “Login”

2. Pedidos:

Descripción general del caso de uso: Este caso de uso puedes ver una grilla de los pedidos actuales, modificarlos- eliminarlos (botón modificar pedidos), y agregar otros pedidos nuevos (botón que dice “agregar pedido” arriba a la derecha de la grilla).

Actor Principal: Gerente

Precondiciones:

1- Sistema ingresa a pantalla principal y muestra todos los pedidos actuales

2- Gerente tiene 7 opciones que muestra el sistema (Cargar pedido, Modificar pedido, ver historial filtrado por fecha, cargar materia prima, consultar stock, cargar producto, modificar producto)

3- Gerente ingresa una opción

Postcondiciones:

1- Sistema ingresa a opción seleccionada por el usuario

Relaciones:

Extendido de “Login”

3. Cargar pedidos:

Descripción general del caso de uso: Este caso de uso permite cargar pedidos al sistema mediante entradas, el sistema para cargar el pedido va a requerir nombre del pedido, fecha de

entrega del pedido, productos a usar, cliente y cantidad. Además de un botón para confirmar y otro para volver a la pantalla anterior.

Actor Principal: Gerente

Precondiciones:

- 1- Sistema ingresa a pantalla de cargar pedidos
- 2- Gerente completa los campos del pedido
- 3- Gerente presiona el botón de “ingresar pedido”

Postcondiciones:

- 1- Sistema carga el pedido a la lista de pedidos actuales
- 2- Sistema permite volver a pantalla principal o seguir agregando pedidos

Relaciones: Extendido de “Pedidos”

3.2 Validar disponibilidad de producto:

Descripción general del caso de uso: Este caso de uso verifica si en el sistema se cuenta con la materia prima necesaria para poder cargar el pedido a la lista consultándolo en una base de datos.

Actor Principal: Gerente

Precondiciones:

- 1- Sistema verifica si cuentan con los productos para poder realizar el pedido

Postcondiciones:

1- Si Sistema cuenta con materia prima se agrega el pedido a lista de pedidos actuales

2- Sistema muestra cartel de confirmación de carga de pedido

Caminos Alternativos:

1- Sistema no cuenta con materia prima

2- Sistema muestra cartel con aviso que no se cuenta con materia prima, especificando cual es el producto que falta.

Relaciones:

Incluido en “Cargar pedido”

[4. Modificar pedidos:](#)

Descripción General: Este caso de uso permite modificar las características de los pedidos que están cargados en el sistema, específicamente permite modificar nombre del pedido, fecha de entrega del pedido, materia prima a usar, cliente y cantidad.

Actor Principal: Gerente

Precondiciones:

1- Sistema carga todos los pedidos actuales

2- Gerente selecciona un pedido para modificar

Postcondiciones:

1- Sistema lanza una ventana emergente con entradas de las características del pedido

2- Gerente realiza los cambios deseados en los campos y da al botón de “guardar cambios”

3- Sistema actualiza el pedido.

Relaciones:

Extendido de “Pedidos”

[5. Borrar pedidos:](#)

Descripción General: Este caso de uso permite borrar los pedidos que están cargados en el sistema y confinando mediante una ventana emergente con un mensaje preguntando si quiere borrar el pedido definitivamente.

Actor Principal: Gerente

Precondiciones:

1- Sistema carga todos los pedidos actuales

2- Gerente selecciona un pedido para eliminar

Postcondiciones:

1- Gerente da a confirmar.

2- Sistema actualiza los pedidos.

Relaciones:

Extendido de “Pedidos”

6. Visitar historial:

Descripción General: Este caso de uso permite visualizar los pedidos pasados y sus características (nombre del pedido, fecha de entrega del pedido, tipo de pedido, materia prima a usar, cliente y cantidad) mediante un filtro de fecha que ingresa el gerente en una grilla.

Actor Principal: Gerente

Precondiciones:

1- Sistema carga los últimos pedidos realizados y permite ingresar una fecha específica de búsqueda

2- Sistema permite ingresar fecha

Postcondiciones:

1- Sistema filtra todos los pedidos a la fecha ingresada

2- Sistema actualiza grilla a pedidos filtrados

Relaciones:

Extendido de “Pedidos”

7. Carga de Materia Prima:

Descripción general del caso de uso: Este caso de uso permite hacer carga materia prima mediante una selección de “ítem” (estos pueden ser manteca, dulce de leche, harina, etc..) e ingresando la cantidad del mismo (especificando su unidad), además esto se almacena en una base de datos.

Actor Principal: Gerente

Precondiciones:

- 1- Sistema ingresa a pantalla de carga materia prima y muestra una entrada que sirve como filtro para seleccionar el ítem, una segunda entrada para seleccionar la unidad (litros, kilogramos,etc...) y un botón de “aceptar”.
- 2- Gerente completa las entradas y da al botón de aceptar.

Postcondiciones:

- 1- Sistema lanza una ventana emergente preguntando si quiere confirmar la carga de materia prima con un botón de confirmación.
- 2- Gerente da a confirmar y la materia prima se almacena en la base de datos.

Relaciones:

Extendido de “Pedidos”

[7.1 Consultar stock materia prima:](#)

Descripción general del caso de uso: Este caso de uso permite visualizar el stock actual de materia prima mediante una grilla. Se muestra el nombre, el distribuidor y el stock disponible.

Actor Principal: Gerente

Precondiciones:

- 1- Sistema ingresa a pantalla de control stock materia prima y muestra una grilla con las materias primas disponibles y sus cantidades.
- 2- Gerente puede ir a agregar materia prima o volver a la pantalla principal.

Postcondiciones:

Relaciones:

Extendido de “Carga de Materia prima”

7.2 Cargar materia prima nueva:

Descripción general del caso de uso: Este caso de uso permite cargar materia prima nueva completando los campos nombre, distribuidor y unidad.

Actor Principal: Gerente

Precondiciones:

1- Sistema ingresa a pantalla Cargar materia prima nueva y muestra los campos para completar.

2- Gerente completa los campos y da a agregar materia prima.

3- Gerente puede ir a eliminar materia prima o volver a la pantalla principal.

Postcondiciones:

Relaciones:

Extendido de “Carga de Materia prima”

7.2.1 Eliminar materia prima:

Descripción general del caso de uso: Este caso de uso permite eliminar materia prima seleccionándola en una lista y confirmándolo con un botón.

Actor Principal: Gerente

Precondiciones:

1- Sistema ingresa a pantalla eliminar materia prima y muestra lista de todas las materias primas.

2- Gerente selecciona la materia prima que desee eliminar y da a botón de confirmar.

Postcondiciones:

Relaciones:

Extendido de “Carga de Materia prima nueva”

8. Carga de Producto:

Descripción general del caso de uso: En este caso de uso se puede cargar un producto completando los campos nombre del producto, precio y cargando las materias primas y su debida unidad.

Actor Principal: Gerente

Precondiciones:

1- Sistema ingresa a pantalla principal y muestra los campos para completar.

2- Gerente completa los campos y da a cargar receta.

3- Gerente carga producto

Postcondiciones:

1- Sistema lanza una advertencia si la materia prima está asignada a algún producto.

Relaciones:

Extendido de “Pedido”

9. Modificar producto:

Descripción General: Este caso de uso permite modificar las características de los productos que están cargados en el sistema, específicamente permite modificar nombre del producto, el precio y su unidad.

Actor Principal: Gerente

Precondiciones:

- 1- Sistema carga todos los productos actuales
- 2- Gerente selecciona un producto para modificar

Postcondiciones:

- 1- Sistema lanza una ventana emergente con entradas de las características del producto
- 2- Gerente realiza los cambios deseados en los campos y da al botón de “guardar cambios”
- 3- Sistema actualiza el producto

Relaciones:

Extendido de “Carga de Producto”

9.1 Borrar producto:

Descripción General: Este caso de uso permite dar de baja los productos que están cargados en el sistema mediante una lista de una ventana emergente.

Actor Principal: Gerente

Precondiciones:

- 1- Sistema carga todos los productos actuales
- 2- Gerente selecciona un producto para eliminar

Postcondiciones:

- 1- Gerente da a confirmar
- 2- Sistema actualiza los productos

Relaciones:

Extendido de “Producto”

Diagrama de entidad-relación:

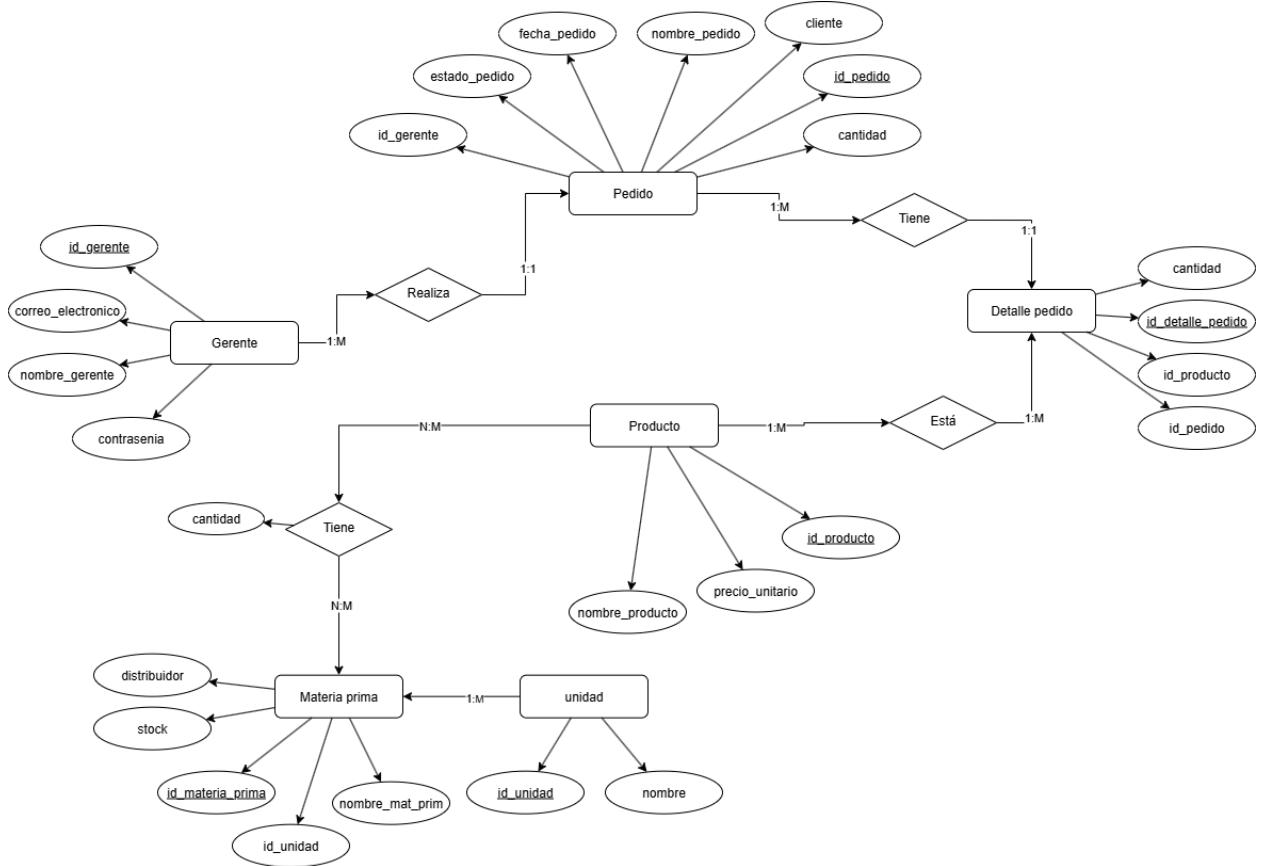
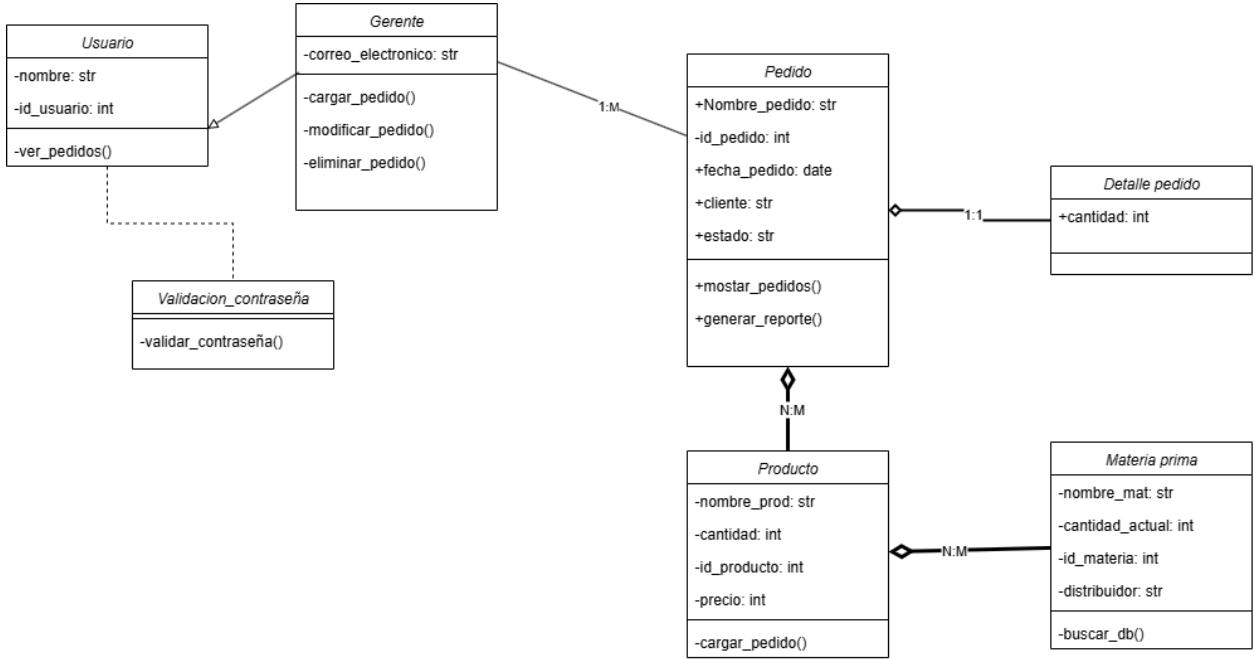


Diagrama de clases:



Arquitectura del sistema

El sistema está diseñado como una aplicación de escritorio con una arquitectura en capas, separando claramente la interfaz de usuario de la lógica de negocio y el acceso a datos.

1. Descripción General de Tecnologías

- Lenguaje de Programación:** Python es el lenguaje principal utilizado tanto para el *backend* (lógica de negocio) como para el *frontend* (interfaz de usuario).
- Interfaz de Usuario (Frontend):** Se utiliza Flet, un framework moderno que permite crear aplicaciones de escritorio (y web/móvil) interactivas en Python. Flet gestiona el renderizado de los componentes visuales y la captura de eventos del usuario.
- Base de Datos (Backend):** Se emplea MySQL como sistema gestor de base de datos relacional (SGBD) para almacenar toda la información persistente (pedidos, productos, stock, usuarios).

- **Conecividad de Base de Datos:** La librería mysql-connector-python se utiliza para establecer la conexión y ejecutar consultas SQL desde la lógica de Python hacia la base de datos MySQL.
- **Seguridad:** Se utiliza la librería bcrypt para el hash y la verificación segura de las contraseñas de los usuarios, evitando almacenarlas en texto plano.
- **Contenerización:** Se proporciona un archivo docker-compose.yml que utiliza Docker para desplegar el servicio de la base de datos MySQL de forma aislada y reproducible, facilitando la configuración del entorno de desarrollo y producción.

2. Estructura de Capas

La aplicación sigue un patrón arquitectónico que separa responsabilidades:

1. Capa de Presentación (Frontend):

- Compuesta por los archivos dentro de la carpeta frontend/.
- Cada archivo pantalla_*.py (ej. pantalla_login.py, pantalla_principal.py) define una vista específica de la aplicación usando componentes de Flet.
- Esta capa es responsable de mostrar la información al usuario y capturar sus interacciones (clics, ingreso de texto).
- Importa y utiliza funciones de la capa de lógica (backend/controladores_pana/) para ejecutar acciones.

2. Capa de Lógica de Negocio y Acceso a Datos (Backend):

- Compuesta por los archivos dentro de la carpeta backend/.

- backend/controladores_pana/: Contiene la lógica de negocio principal. Módulos como controlador_cargar_pedido.py o control_cargar_producto.py definen funciones para las operaciones CRUD (Crear, Leer, Actualizar, Borrar) y las validaciones (ej. verificar stock).
- backend/conexion_a_BD/: Gestiona la conexión con la base de datos MySQL. Los controladores utilizan esta conexión para ejecutar consultas SQL.

3. Capa de Datos (Base de Datos):

- Es la base de datos MySQL en sí misma, cuya estructura está definida en el script db_practicasprof.sql.
- Esta capa se desacopla del resto de la aplicación y es gestionada idealmente por Docker.

3. Estructura de Carpetas

```
sistema_gestor_panificadora/
├── backend/           # Lógica de negocio y acceso a datos
│   ├── conexion_a_BD/ # Módulos de conexión a la BD
│   │   └── conexion_db.py
│   ├── controladores_pana/ # Controladores con lógica CRUD y validaciones
│   │   ├── control_cargar_materia_prima.py
│   │   ├── control_cargar_producto.py
│   │   ├── controlador_cargar_pedido.py
│   │   └── ... (otros controladores)
│   └── modelos_pana/     # Clases que modelan entidades (ej. Gerente)

├── frontend/          # Interfaz de usuario (vistas de Flet)
│   ├── controladores_front/ # Clases base o configuración de UI
│   │   └── configuracion_pantalla.py
│   ├── pantalla_cargar_materia_prima.py
│   ├── pantalla_cargar_producto.py
│   ├── pantalla_controlar_stock.py
│   ├── pantalla_login.py
│   ├── pantalla_principal.py
│   └── ... (otras pantallas)

└── documentacionYdiagramas/ # Documentación de soporte, diagramas y mockups
    ├── mockup/
    ├── Diagrama_de_clases.drawio.png
    └── diagrama_e_r_correccion_dic2.drawio.png

├── db_practicasprof.sql      # Script SQL para la creación de la base de datos
├── docker-compose.yml        # Archivo de configuración de Docker para la BD
├── logo_panaderia.png       # Recursos gráficos (assets)
└── main.py                   # Punto de entrada de la aplicación Flet
```

El archivo main.py actúa como el orquestador principal: inicializa la aplicación Flet, define las rutas de navegación y mapea cada ruta a una función constructora de las vistas definidas en la carpeta frontend/.

Entrevista:

Preguntas:

- a) ¿Cómo manejan actualmente los pedidos de la empresa?
- b) ¿Cuáles son las principales dificultades que enfrenta actualmente con el sistema de pedidos basado en WhatsApp y la pizarra?
- c) ¿Qué información considera imprescindible que se registre al momento de recibir un pedido?
- d) ¿Cómo le gustaría que el sistema visualice los pedidos actuales?
- e) ¿Tiene algún requisito especial en cuanto al acceso y seguridad del sistema?
- f) ¿Le gustaría una funcionalidad que permita ver los pedidos pasados como un historial?

Respuestas:

- a) Actualmente se manejan con pedidos mediante mensajes de WhatsApp y de allí se trasladan a una pizarra y así pueden verlo los empleados
- b) El mayor problema es la desorganización. A veces, los pedidos se pierden entre tantos mensajes, o nos olvidamos de pasarlos a la pizarra. Cuando hay muchos pedidos en un día, es fácil que se pierda alguno o que se haga un mal seguimiento.

- c) Lo básico sería: nombre del cliente, productos que está pidiendo, cantidad de cada producto, y la fecha y hora de entrega.
- d) Me gustaría que el sistema sea fácil de usar y que cada pedido tenga un estado claro, algo sencillo como 'encargado', 'en preparación' y 'completado', y que se pueda cambiar el estado fácilmente conforme vamos avanzando.
- e) Con una contraseña para el ingreso es suficiente

Eso sería muy útil ya que tendríamos un control más óptimo de las ganancias y cantidad de pedidos por mes, semana o año.

Conclusión

Este proyecto cierra exitosamente el ciclo formativo, logrando no solo entregar un producto de software que cumple con los objetivos de modernizar la gestión de la panificadora, sino también validando la capacidad de transitar desde el análisis abstracto hasta la resolución de problemas técnicos concretos. Mis principales avances se reflejan en el sistema funcional, con una arquitectura robusta y, fundamentalmente, en la experiencia adquirida al enfrentar y superar los desafíos reales del desarrollo de software, estoy muy orgulloso del trabajo que realicé.

A continuación voy a nombrar lo que fueron mis **mayores desafíos** en este trayecto:

El Desafío del Frontend: Uno de los retos más significativos fue la construcción de una interfaz de usuario que no solo fuera funcional, sino también compleja en su composición, sencilla y atractiva visualmente. La implementación de las distintas vistas con el framework Flet exigió un esfuerzo considerable para gestionar el estado, la navegación y la presentación de datos, logrando un equilibrio entre estética y usabilidad.

La Complejidad de la Modularización: La arquitectura del sistema se diseñó para ser altamente modular, separando estrictamente el frontend del backend, y este último a su vez en capas de controladores, modelos y conexión a datos. Si bien esta estructura probó ser fundamental para la organización y mantenibilidad del código, gestionar el flujo de información y las dependencias entre tantos archivos y carpetas constituyó un desafío de diseño de software en sí mismo.

La Iteración como Proceso de Desarrollo: El desarrollo en Prácticas II demostró que la planificación inicial, aunque crucial, es un punto de partida dinámico. El proyecto experimentó varias reestructuraciones de la arquitectura de la base de datos para optimizar las relaciones y el rendimiento. De igual manera, el alcance funcional fue objeto de un proceso iterativo constante, requiriendo recortes de funciones que resultaron ser secundarias y la agregación de nuevas características que emergieron como esenciales durante la implementación.

1. Prototipo de Interfaces

1.1 Herramienta utilizada:

MockFlow

1.2 Descripción general:

El prototipo representa el flujo del sistema de gestión de pedidos de una panadería, incluyendo el login, registro de pedidos, control de stock, gestión de productos y reportes.

1.3 Capturas de pantalla / Enlaces al prototipo:

Pantalla 1: Login

- Descripción: En esta pantalla el Gerente inicia sesión.

- Transiciones desde/hacia esta pantalla: Login – Pantalla principal.

Imagen 1.1:



Pantalla 2: Pantalla principal

- Descripción: En esta pantalla el Gerente visualizar los pedidos actuales y su estado, además la opción de elegir entre las opciones de cargar materia prima, cargar producto, consultar materia prima, modificar producto, filtrar pedidos por fecha, cargar un pedido y modificar un pedido.
- Transiciones desde/hacia esta pantalla: Pantalla principal – Opción elegida.

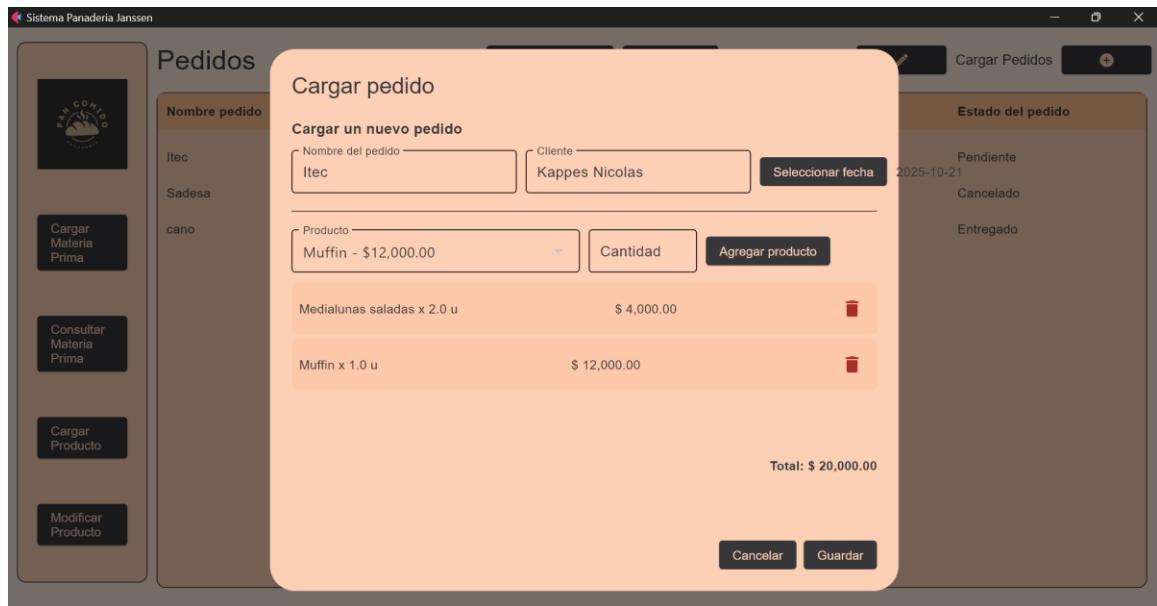
Imagen 1.2:

Pedidos				
Filtrar por fecha: <input type="button" value="Seleccionar fecha"/> <input type="button" value="Limpiar filtro"/> <input type="button" value="Modificar Pedidos"/> <input type="button" value="Cargar Pedidos"/> <input type="button" value="Nuevo"/>				
Nombre pedido	Cliente	Fecha del pedido	Precio total	Estado del pedido
Itec	Nicolas Kappes	2025-10-20	2000.0000	Pendiente
Sadesa	Eberhardt	2025-10-22	6000.0000	Cancelado
cano	cano	2025-10-24	3000.0000	Entregado

Pantalla 3: Carga de pedidos

- Descripción: En esta pantalla el gerente puede cargar un pedido completando los campos.
- Transiciones desde/hacia esta pantalla: Carga de pedidos – Pantalla principal.

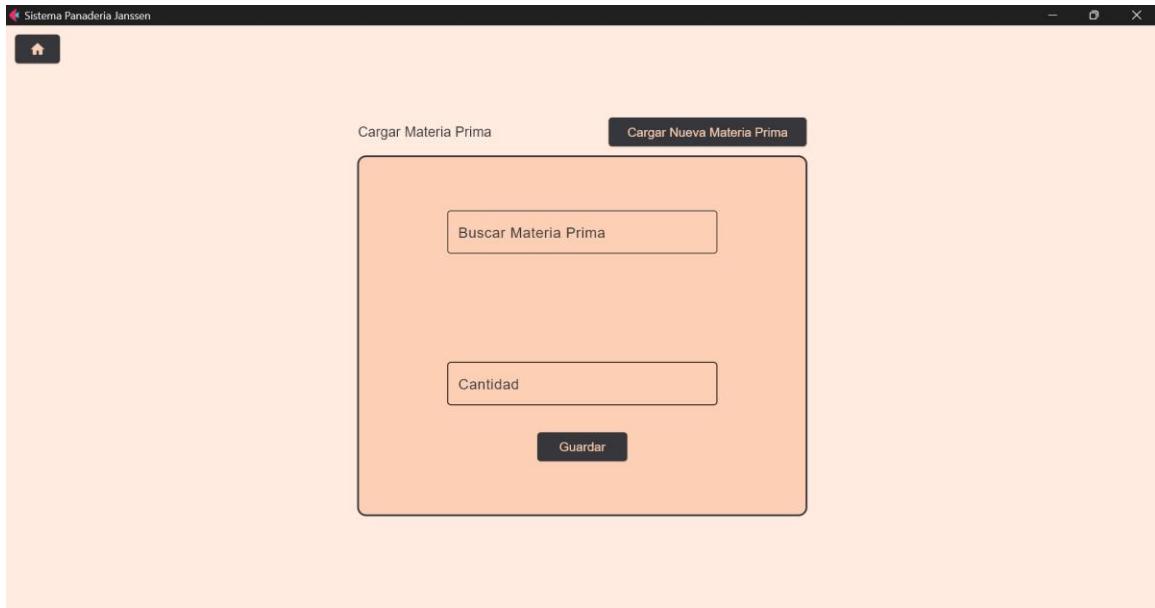
Imagen 1.3:



Pantalla 4: Carga de materia prima

- Descripción: En esta pantalla el gerente puede cargar materia prima completando los campos.
- Transiciones desde/hacia esta pantalla: Carga de materia prima – Pantalla principal.

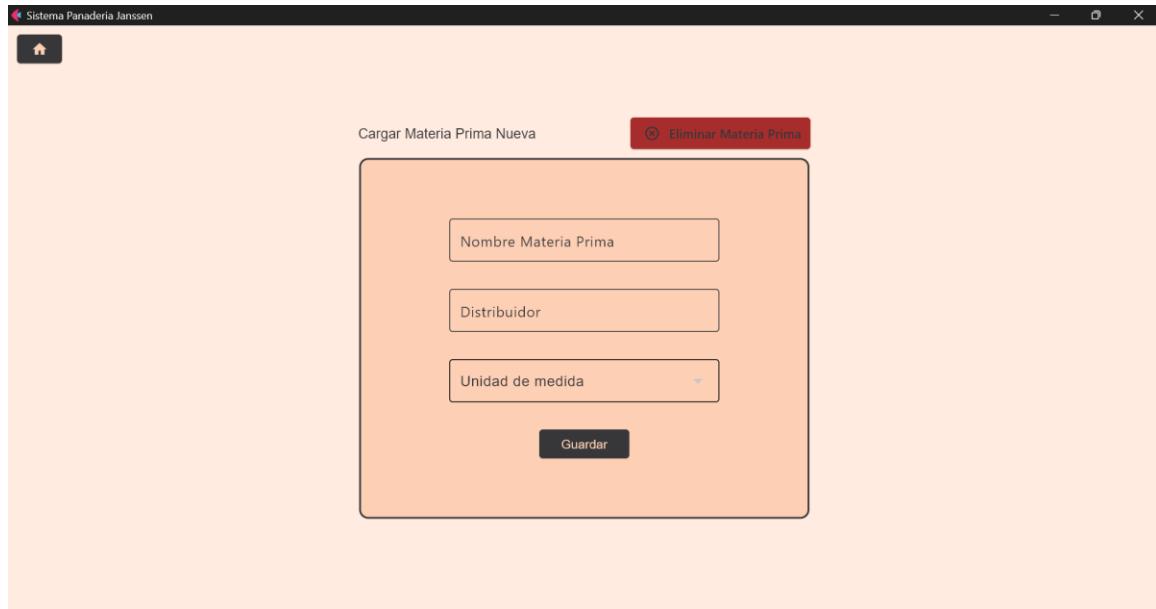
Imagen 1.4:



Pantalla 5: Carga de materia prima nueva

- Descripción: En esta pantalla el gerente puede cargar una materia prima nueva completando los campos.
- Transiciones desde/hacia esta pantalla: Carga de materia prima nueva – Pantalla principal.

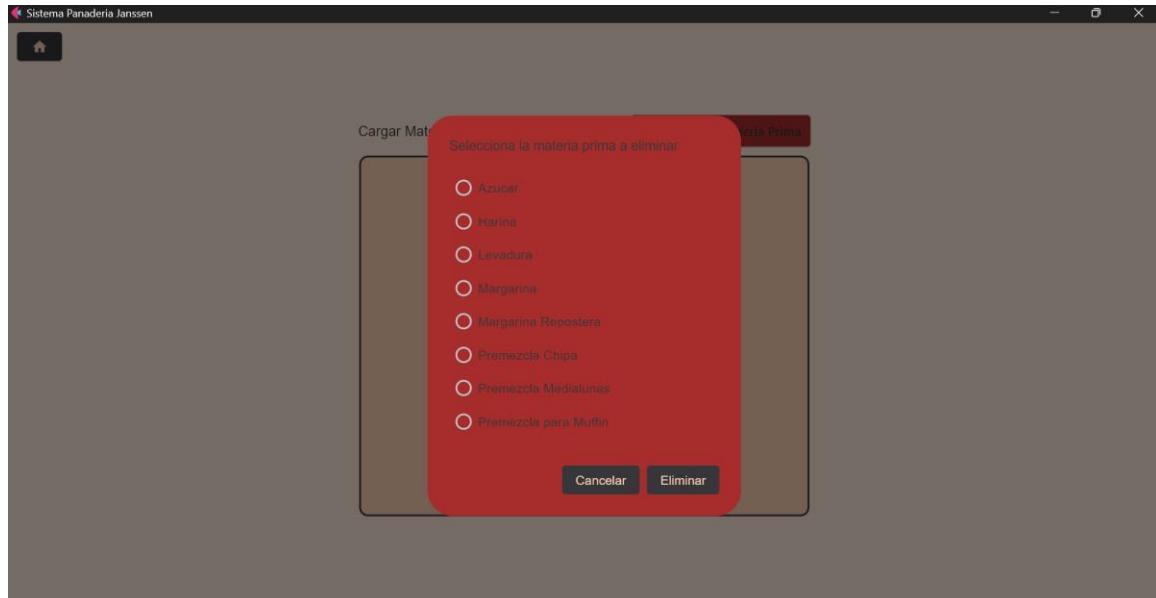
Imagen 1.5:



Pantalla 6: Eliminar materia prima

- Descripción: En esta pantalla el gerente puede eliminar una materia prima seleccionándola.
- Transiciones desde/hacia esta pantalla: Eliminar materia prima – Carga de materia prima nueva.

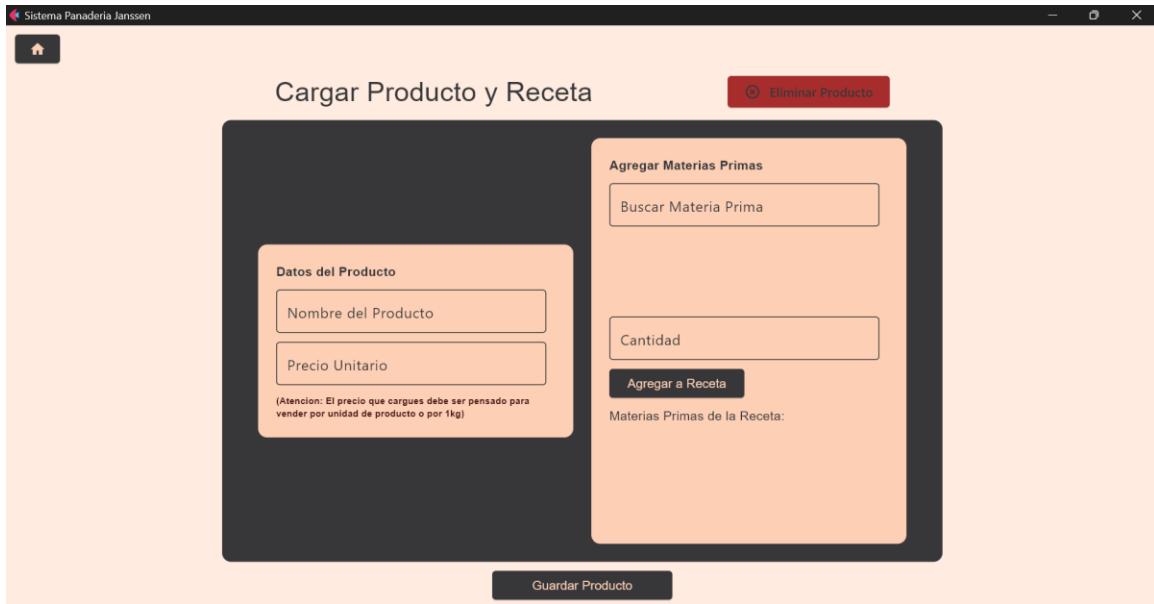
Imagen 1.6:



Pantalla 7: Carga de producto

- Descripción: En esta pantalla el gerente puede cargar un producto completando los campos.
- Transiciones desde/hacia esta pantalla: Carga de producto– Pantalla principal.

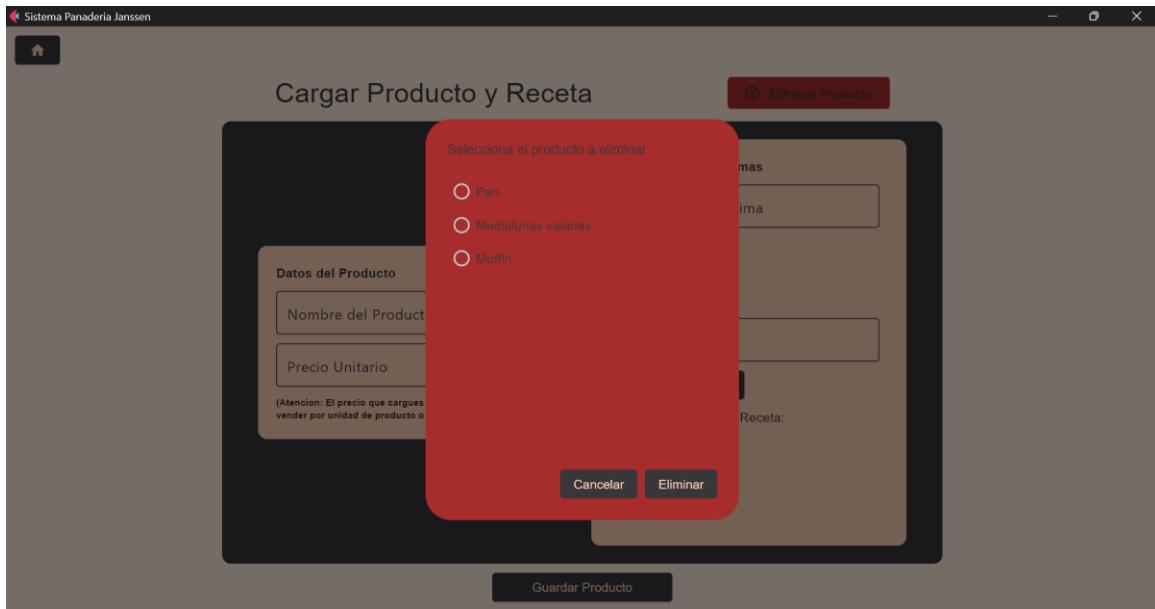
Imagen 1.7:



Pantalla 8: Eliminación de producto

- Descripción: En esta pantalla el gerente puede eliminar un producto seleccionándolo.
- Transiciones desde/hacia esta pantalla: Eliminación Producto– Carga de producto.

Imagen 1.8:



Pantalla 9: Consultar Stock

- Descripción: En esta pantalla el gerente puede visualizar el stock actual de materia prima
- Transiciones desde/hacia esta pantalla: Consulta stock – Pantalla principal o agregar stock.

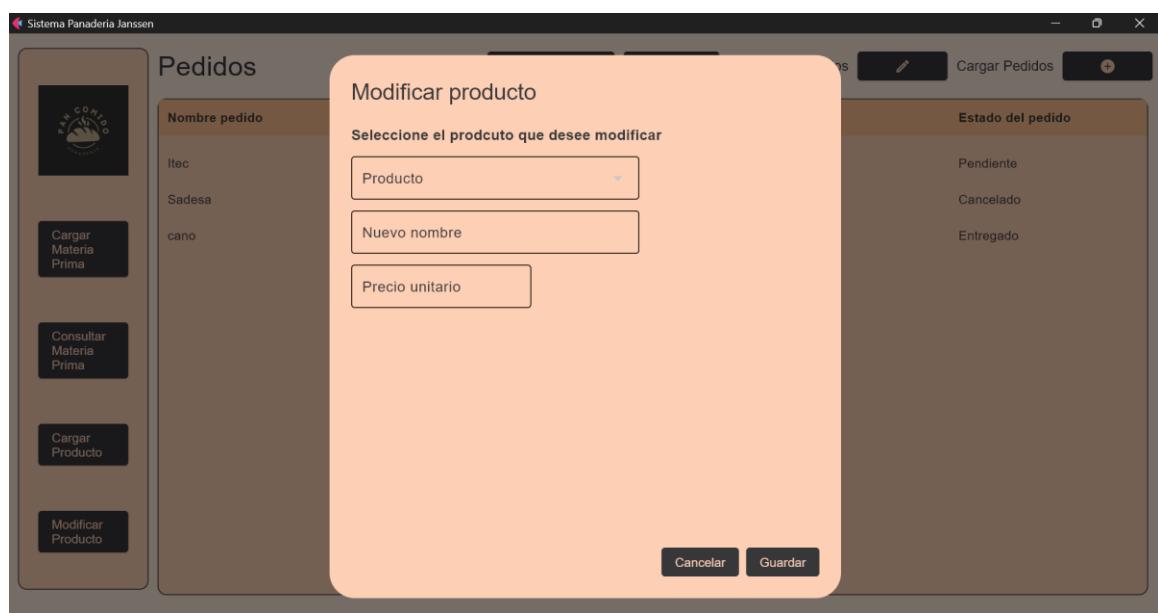
Imagen 1.9:

Control de Stock de Materia Prima			Agregar Stock
Nombre	Distribuidor	Stock Disponible	
Azucar	Arco	0.00 kg	
Harina	Nani	348.00 kg	
Levadura	Nani	2,074.00 g	
Margarina	Nani	50.00 kg	
Margarina Repostera	Nani	50.00 kg	
Premezcla Chipa	Calsa	20.00 kg	
Premezcla Medialunas	Calsa	0.00 kg	
Premezcla para Muffin	Calsa	1.00 kg	
Sal	Nani	909.00 g	

Pantalla 10: Modificar Producto

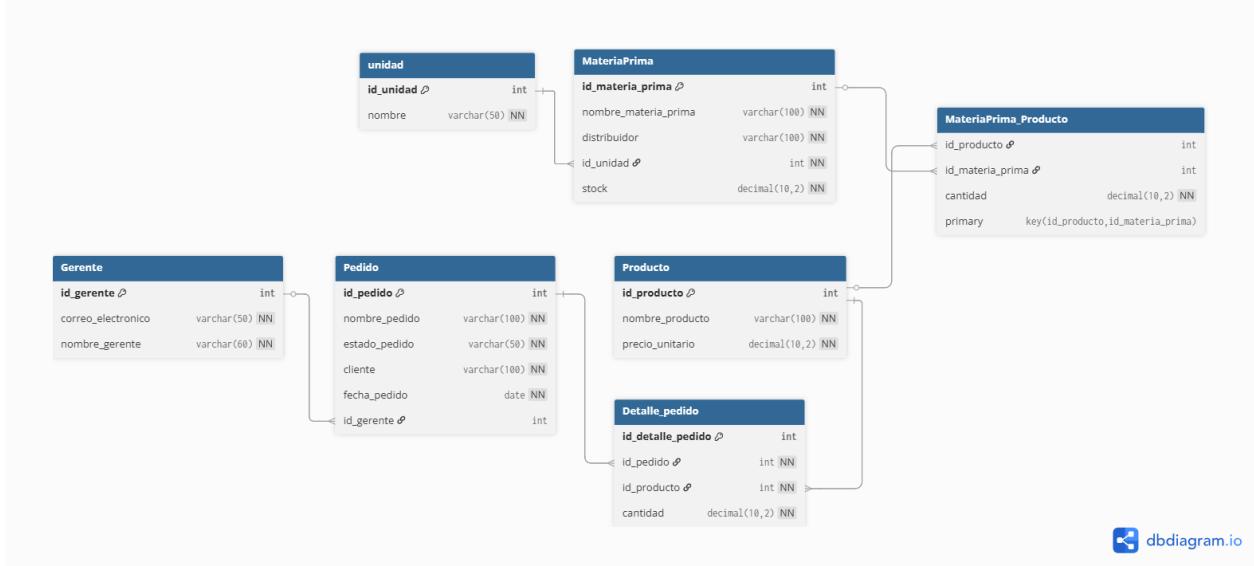
- Descripción: En esta pantalla el gerente modificar un producto.
- Transiciones desde/hacia esta pantalla: Modificar producto– Pantalla principal.

Imagen 2.1:



2. Estructura de la Base de Datos

2.1 Modelo Lógico



Relaciones entre tablas

Un Gerente puede gestionar muchos Pedidos → Pedido.id_gerente es clave foránea que referencia a Gerente.id_gerente.

Una Unidad (de medida) puede usarse en muchas Materias Primas → MateriaPrima.id_unidad es clave foránea que referencia a unidad.id_unidad.

Un Pedido puede tener muchos Detalles de pedido → Detalle_pedido.id_pedido es clave foránea que referencia a Pedido.id_pedido.

Un Producto puede aparecer en muchos Detalles de pedido → Detalle_pedido.id_producto es clave foránea que referencia a Producto.id_producto.

Un Producto puede estar hecho de muchas Materias Primas, y una Materia Prima puede ser usada en muchos Productos → Relación muchos a muchos representada por la tabla intermedia MateriaPrima_Producto.

- MateriaPrima_Producto.id_producto es clave foránea que referencia a Producto.id_producto.
- MateriaPrima_Producto.id_materia_prima es clave foránea que referencia a MateriaPrima.id_materia_prima.

2.2 Modelo Físico

```
CREATE TABLE Gerente (
    id_gerente INT AUTO_INCREMENT PRIMARY KEY,
    correo_electronico VARCHAR(50) NOT NULL UNIQUE,
    nombre_gerente VARCHAR(60) NOT NULL
) ENGINE=InnoDB;
```

```
CREATE TABLE unidad (
    id_unidad INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(50) NOT NULL UNIQUE
) ENGINE=InnoDB;
```

```
CREATE TABLE MateriaPrima (  
  
    id_materia_prima INT AUTO_INCREMENT PRIMARY KEY,  
  
    nombre_materia_prima VARCHAR(100) NOT NULL,  
  
    distribuidor VARCHAR(100) NOT NULL,  
  
    id_unidad INT NOT NULL,  
  
    stock DECIMAL(10,2) NOT NULL,  
  
    FOREIGN KEY (id_unidad) REFERENCES unidad(id_unidad)  
  
) ENGINE=InnoDB;
```

```
CREATE TABLE Producto (  
  
    id_producto INT AUTO_INCREMENT PRIMARY KEY,  
  
    nombre_producto VARCHAR(100) NOT NULL,  
  
    precio_unitario DECIMAL(10,2) NOT NULL  
  
) ENGINE=InnoDB;
```

```
CREATE TABLE MateriaPrima_Producto (  
  
    id_producto INT NOT NULL,
```

```

    id_materia_prima INT NOT NULL,
    cantidad DECIMAL(10,2) NOT NULL,
    PRIMARY KEY (id_producto, id_materia_prima),
    FOREIGN KEY (id_producto) REFERENCES Producto(id_producto),
    FOREIGN KEY (id_materia_prima) REFERENCES MateriaPrima(id_materia_prima)
) ENGINE=InnoDB;

```

```

CREATE TABLE Pedido (
    id_pedido INT AUTO_INCREMENT PRIMARY KEY,
    nombre_pedido VARCHAR(100) NOT NULL,
    estado_pedido VARCHAR(50) NOT NULL,
    cliente VARCHAR(100) NOT NULL,
    fecha_pedido DATE NOT NULL,
    id_gerente INT,
    FOREIGN KEY (id_gerente) REFERENCES Gerente(id_gerente)
) ENGINE=InnoDB;

```

```

CREATE TABLE Detalle_pedido (

```

```
id_detalle_pedido INT AUTO_INCREMENT PRIMARY KEY,  
id_pedido INT NOT NULL,  
id_producto INT NOT NULL,  
cantidad DECIMAL(10, 2) NOT NULL,  
FOREIGN KEY (id_pedido) REFERENCES Pedido(id_pedido),  
FOREIGN KEY (id_producto) REFERENCES Producto(id_producto)  
) ENGINE=InnoDB;
```