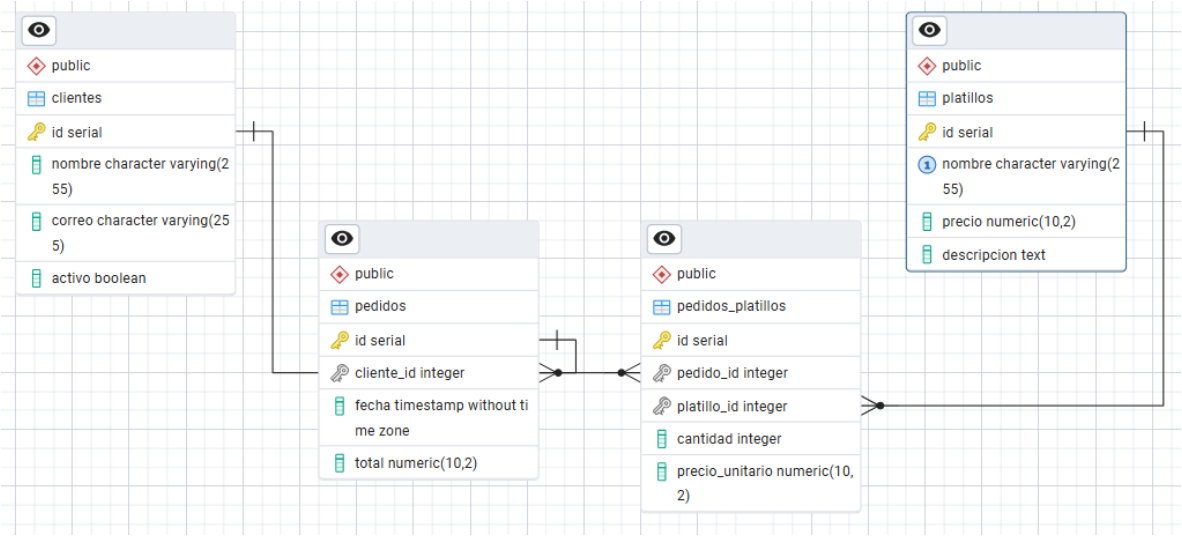
REPORTE DE BD

Esmeralda Bautista Martínez

Problema:

El restaurante “Delicias Gourmet” desea un sistema de escritorio que permita gestionar su menú de platillos, registrar a sus clientes y permitir que cada cliente realice pedidos que incluyan múltiples platillos. A su vez, cada platillo puede estar presente en varios pedidos.

Modelo de base de datos:



Explicación del código:

1) Importaciones y documento

- `from docx import Document` → clase principal para crear/editar un .docx
- `from docx.shared import Inches, Pt` → medidas/tamaños (no las usamos mucho aquí, pero útiles)
- `Document()` → instancia el documento en blanco

```
doc = Document()
```

2) Título principal (Heading nivel 1)

```
doc.add_heading('📄 Manual de Usuario - ... “Delicias Gourmet”', level=1)
```

- `level=1` = Título grande (estilo de Word “Heading 1”).

3) Secciones con `Heading` y `Paragraph`

- Para cada sección: `add_heading(..., level=2)` y después `add_paragraph(...)`.
- Ejemplo: **Introducción** y **Contenido**.

```
doc.add_heading('📜 Introducción', level=2)
doc.add_paragraph("Bienvenido(a) ... qué validaciones ...")
```

4) Lista numerada del índice

- Itera sobre una lista de strings y los agrega con `style='List Number'` para numeración automática:

```
for item in contenido:
    doc.add_paragraph(item, style='List Number')
```

5) Bloques de texto con viñetas "manuales"

- Para secciones como "Objetivo del sistema" y "Requisitos", se concatenan líneas con "- " o saltos de línea `\n`.
- (Otra opción sería usar estilos de lista, pero aquí se simplificó con texto plano y guiones.)

```
doc.add_paragraph(
    "El sistema fue diseñado para:\n"
    "- Reducir el tiempo...\n"
    "- Llevar un control...\n"
)
```

6) "Módulos" con subtítulos y bullets

- Se arma un diccionario `modulos = {...}` y se recorre:
    - Cada **clave** es un heading (nivel 2).
    - Cada **valor** es una lista de bullets (como párrafos con "- ").

```
for modulo, funciones in modulos.items():
    doc.add_heading(modulo, level=2)
    for func in funciones:
        doc.add_paragraph(f"- {func}")
```

7) Tabla de "Problemas comunes"

- `add_table(rows=1, cols=3)` crea la tabla.
- Primera fila = encabezados.
- Luego, por cada problema, se agrega una fila con `tabla.add_row().cells`.

```
tabla = doc.add_table(rows=1, cols=3)
hdr_cells = tabla.rows[0].cells
hdr_cells[0].text = 'Problema'
hdr_cells[1].text = 'Posible Causa'
hdr_cells[2].text = 'Solución'

for problema, causa, solucion in problemas:
    row_cells = tabla.add_row().cells
    row_cells[0].text = problema
    row_cells[1].text = causa
    row_cells[2].text = solucion
```

8) Guardar el archivo

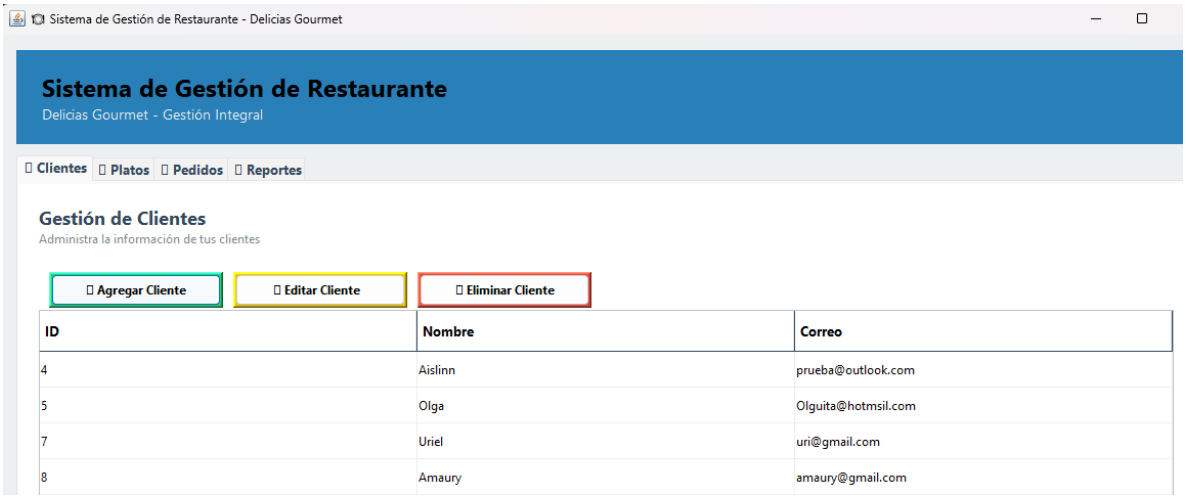
- Define la **ruta** y guarda el documento:

```
file_path = "/mnt/data/Manual_Usuario_Delicias_Gourmet.docx"
doc.save(file_path)
```

- Esa ruta es la que te compartí para descargar.

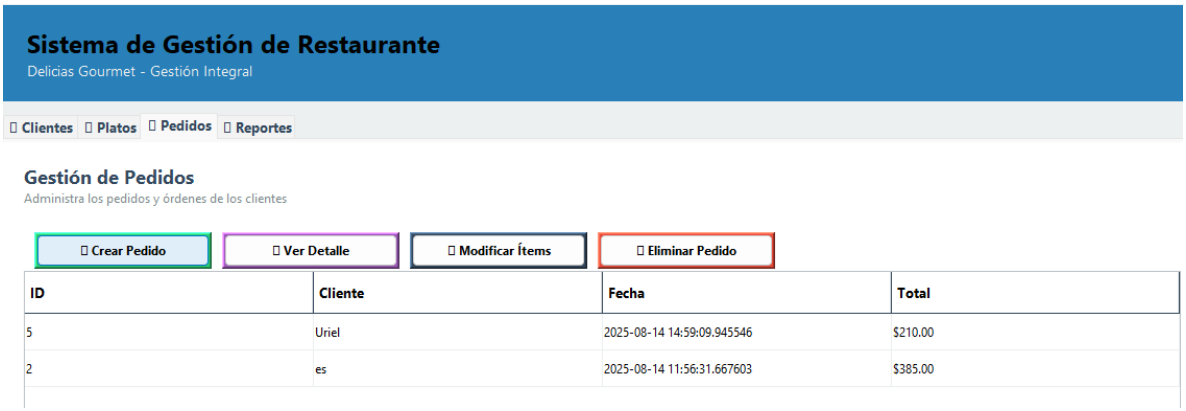
Casos de prueba:

Clientes:



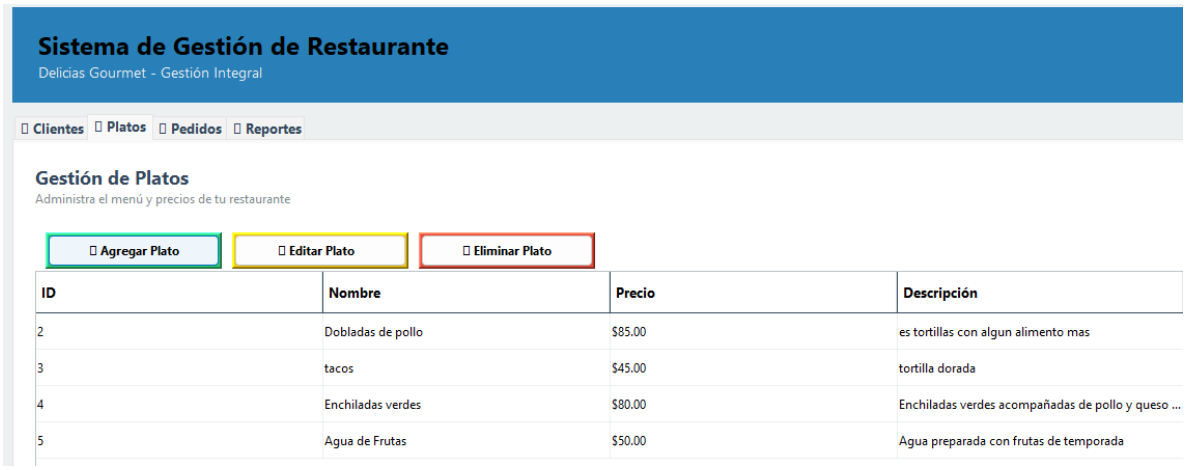
	id [PK] integer	nombre character varying (255)	correo character varying (255)	activo boolean
1	1	es	esme@gmail.com	false
2	2	Sam	sam@gmail.com	false
3	7	Uriel	uri@gmail.com	true
4	5	Olga	Olguita@hotmsil.com	true
5	4	Aislinn	prueba@outlook.com	true

Pedidos:



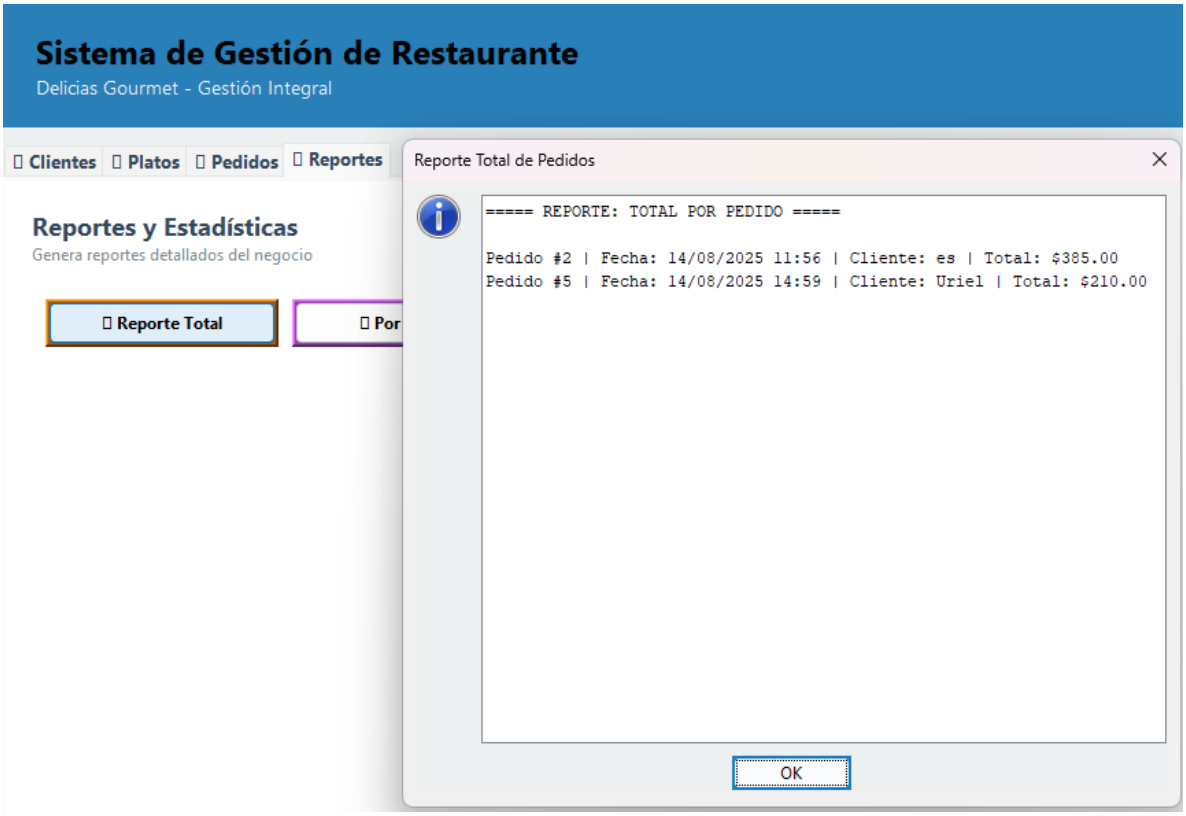
	id [PK] integer	nombre character varying (255)	precio numeric (10,2)	descripcion text
1	3	tacos	45.00	tortilla dorada
2	4	Enchiladas verdes	80.00	Enchiladas verdes acompañadas de pollo y queso fresco
3	5	Agua de Frutas	50.00	Agua preparada con frutas de temporada
4	2	Dobladas de pollo	85.00	es tortillas con algun alimento mas

Platillo:



	id [PK] integer	nombre character varying (255)	precio numeric (10,2)	descripcion text
1	3	tacos	45.00	tortilla dorada
2	4	Enchiladas verdes	80.00	Enchiladas verdes acompañadas de pollo y queso fresco
3	5	Agua de Frutas	50.00	Agua preparada con frutas de temporada
4	2	Dobladas de pollo	85.00	es tortillas con algun alimento mas

Reportes:



Manual de usuario:

Manual de Usuario - Sistema de Gestión de Restaurante “Delicias Gourmet”

Introducción

Bienvenido(a) al Manual de Usuario del Sistema de Gestión de Restaurante 'Delicias Gourmet'. Este documento le guiará paso a paso en el uso de todas las funcionalidades disponibles en el sistema. El objetivo es facilitar la administración de clientes, platillos y pedidos de manera rápida, segura y eficiente.

Contenido

1. 1. Objetivo del sistema
2. 2. Requisitos del sistema
3. 3. Acceso al sistema
4. 4. Descripción de la interfaz principal
5. 5. Gestión de Clientes
6. 6. Gestión de Platillos
7. 7. Gestión de Pedidos
8. 8. Reportes
9. 9. Manejo de errores

10. 10. Preguntas frecuentes

Objetivo del Sistema

El sistema fue diseñado para:
- Reducir el tiempo de gestión de clientes, platillos y pedidos.
- Llevar un control preciso de los pedidos realizados.
- Garantizar la integridad de los datos almacenados.
- Mejorar la experiencia del usuario al interactuar con la interfaz.

Requisitos del Sistema

• Sistema Operativo: Windows 7 o superior, Linux o macOS.
• Java 8 o superior instalado.
• PostgreSQL 12 o superior.
• Conexión a Internet para instalación y actualizaciones.
• 4 GB de RAM y 500 MB de espacio libre en disco.

Acceso al Sistema

El sistema puede iniciarse ejecutando el archivo .jar o desde un entorno de desarrollo como NetBeans o IntelliJ. Una vez iniciado, se mostrará la ventana principal con las pestañas de Clientes, Platillos, Pedidos y Reportes.

Gestión de Clientes

- Agregar nuevos clientes con nombre y correo.

- Listar clientes activos.

- Modificar datos de clientes.

- Eliminar clientes y sus pedidos asociados.

Gestión de Platillos

- Registrar platillos con nombre, precio y descripción.

- Validar que el nombre sea único y precio > 0.

- Editar o eliminar platillos del menú.

Gestión de Pedidos

- Crear nuevos pedidos seleccionando un cliente y uno o más platillos.

- Ver los pedidos de un cliente.

- Eliminar pedidos existentes.

- Calcular automáticamente el total a pagar.

Reportes

- Generar reporte total de pedidos.

- Obtener lista de pedidos por cliente con fecha.

Manejo de Errores

El sistema incluye validaciones para prevenir errores comunes:
- No se pueden crear pedidos sin platillos.
- Las cantidades deben ser mayores a 0.
- No se permiten nombres de platillos repetidos.
- El correo de cliente debe ser válido.

?Preguntas Frecuentes

• ¿Qué sucede si elimino un cliente?
Se eliminarán también sus pedidos asociados.

• ¿Puedo modificar un pedido después de creado?
Actualmente no, debe eliminarlo y crearlo nuevamente.

• ¿Es posible recuperar datos eliminados?
No, las eliminaciones son permanentes.

Conclusiones:

El desarrollo del sistema de escritorio **"Delicias Gourmet"** representa una solución integral para la gestión de clientes, platillos y pedidos dentro de un restaurante. A lo largo del proyecto, se logró implementar una interfaz intuitiva y amigable, respaldada por una base de datos sólida en **PostgreSQL**, que garantiza la integridad y disponibilidad de la información.

El uso de **Java con Swing** permitió construir un entorno visual ordenado y fácil de navegar, mientras que la aplicación de **JDBC** aseguró una conexión estable y eficiente con la base de datos. La estructura modular del código favorece el mantenimiento y escalabilidad futura, permitiendo que el sistema pueda adaptarse a nuevas funcionalidades como reportes avanzados, integración con sistemas de facturación o un módulo de reservas.

Gracias a las validaciones y al manejo de errores, el sistema previene registros inconsistentes, evita pedidos incompletos y protege la información. Además, la relación entre entidades asegura que las operaciones como la eliminación de clientes mantengan la coherencia de datos, borrando automáticamente los pedidos asociados.