



# Universidad Nacional Autónoma de México Facultad de Ingeniería

*BASE DE DATOS  
Grupo:01*

*Reporte Proyecto Final  
Base de Datos*

**Integrantes:**

Mata Ramírez Andrea  
Rojo Ramirez Luis Edgar  
Rosas Cañada Abraham  
Vargas Luna José Ángel  
Zuckerman Cisneros Stephan

# ÍNDICE

1. INTRODUCCIÓN.....	3
2. PLAN DE TRABAJO.....	7
3. DISEÑO.....	10
4. IMPLEMENTACIÓN.....	20
5. CONCLUSIONES.....	22

# INTRODUCCION

En este proyecto utilizaremos los recursos y habilidades vistas en clases de teoría y laboratorio de bases de datos, desde el desarrollo y creación de un modelo Entidad-Relación hasta el desarrollo del Modelo Relacional del mismo, además de buscar soluciones efectivas y eficaces de la implementación de una base de datos desde cero.

El desarrollo de este trabajo escrito mostrará el proceso de análisis, diseño, descripción, manejo e implementación del mismo visto desde una fase inicial hasta una implementación más realista al mundo laboral, también se desarrolla desde la idea de los desarrolladores, desde la implementación y asignación de roles de los miembros del equipo hasta su desarrollo como programa y recurso de un futuro y posible establecimiento o empresa que busque una solución efectiva en el manejo de datos y como buscar opciones de soluciones ante diversos escenarios, así como visualizar las distintas características y fases que pasa por una base de datos antes de ser implementada en algún emprendimiento, negocio o comercio que necesite un manejo de datos

extenso, organizado o que se ajuste a las necesidades del solicitante.

También implementaremos el uso de diversas herramientas y tareas en Python que pueden ayudar bastante al manejo de disponibilidad y seguridad de nuestra base, que permita una buena interacción entre los usuarios que deseen utilizarla entre cliente y mesero, además de mostrar el gran avance a nivel de programación que se tuvo para poder llegar a una base de datos más realista y con posibilidad de campo laboral a futuro.



## Objetivos:

- Crear una base de datos estructurada y segura en PostgreSQL para almacenar y gestionar la información de los empleados del restaurante.
- Implementar funciones para el registro, búsqueda y visualización de datos de empleados, asegurando la integridad y accesibilidad de la información.
- Mejorar la eficiencia operativa del restaurante mediante la integración de herramientas digitales que faciliten la gestión y reduzcan el tiempo dedicado a tareas manuales.
- Diseñar el sistema de manera que pueda escalarse fácilmente para acomodar futuros crecimientos en el número de empleados y funcionalidades adicionales.
- Asegurar que el código sea mantenible y documentado para facilitar futuras actualizaciones y mejoras.

# Soluciones propuestas

- Utilizar Python y Tkinter para desarrollar una interfaz gráfica de usuario (GUI) que permita una fácil interacción con el sistema.
- Diseñar la interfaz para que sea intuitiva y fácil de navegar, permitiendo a los usuarios realizar operaciones comunes con mínima capacitación.
- Incorporar funcionalidades que automatizan tareas administrativas, como el registro y actualización de datos de empleados.
- Implementar medidas de seguridad en la base de datos para proteger la información sensible de los empleados.
- Garantizar que solo personal autorizado tenga acceso a datos críticos, manteniendo la confidencialidad y seguridad de la información.



# PLAN DE DISEÑO

Es este apartado contaremos nuestro modo de trabajo, desde asignación de roles y cómo fueron seleccionados para cada uno de los integrantes del equipo.

- Diseñadores: Mata Ramírez Andrea & Rosas Cañada Abraham
- Programadores: Rojo Ramírez Luis Edgar & Zuckerman Cisneros Stephan
- Estadísticos: Vargas Luna José Ángel & Mata Ramírez Andrea

## Diseñadores

Nuestro equipo de diseñadores se dieron a la tarea de crear y elaborar los modelos de Entidad-Relación de nuestra base de datos, al igual que analizarla y llegar al Modelo Relacional.

Tomando en cuenta las especificaciones del proyecto y lo solicitado, tuvieron la tarea de realizar los modelos que permitan que tanto el mesero como el cliente tengan acceso a la aplicación. Por parte del mesero registrarse y poder tomar ordenes, así como el cliente poder pedir

y ver la información de lo que pide. Regularmente por medio de lo visto en clase tanto teoría como laboratorio se buscaba y apoyaban de material visto que pudiera aminorar la carga de trabajo o reducir la dificultad en esta labor.

## Programadores

Nuestro equipo de programadores, junto con el de los diseñadores buscará implementar de manera optativa el código generado por medio de las especificaciones de las tablas, atributos y restricciones que tengan nuestra base creada, además de mantener en constante actualización cada movimiento que llegara a surgir, al igual que ideas para mantener la seguridad y limitaciones entre clientes y mesero. También entre los comentarios y opiniones donde iban surgiendo dudas y cuestiones que pusieran en tela de juicio algunos aspectos de la base, como utilizar una interfaz gráfica para volver más fácil el trabajo.

## Estadísticos

Est parte del equipo se encargó de registrar y documentar la programación y evaluación de la base de datos, además de notificar cuando se detectaron errores de ejecución, relación o incluso si se llegó a tener un posible caso de afectación a la seguridad de la base. Como se

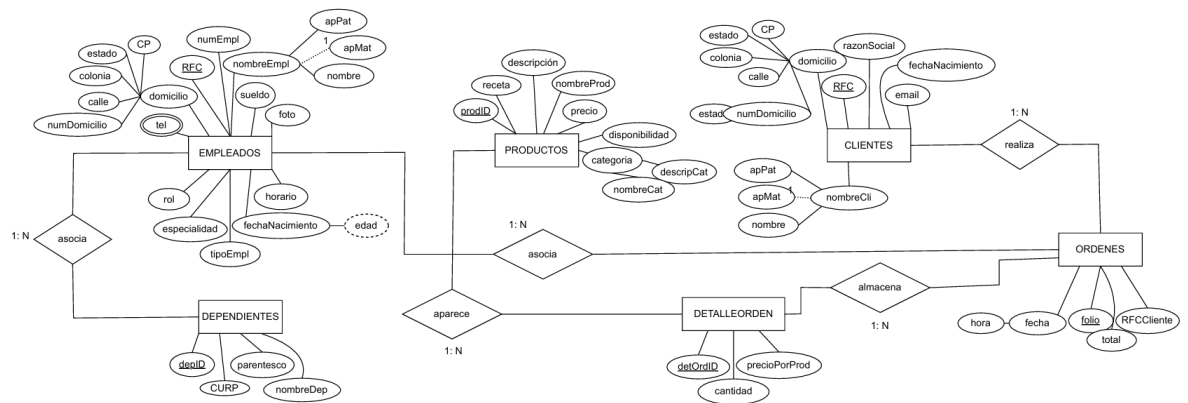


notará, ellos en la documentación también expondrán cuales fueron tantos los errores y aciertos que tuvo el equipo, su relación e interacción entre los miembros, además de fomentar el apoyo mutuo entre todos. Adicionalmente fueron los desarrolladores de la presentación que se expondrá en clase y detallaran los aspectos más importantes de este proyecto, además de desarrollar un poco más las habilidades de la buena escritura, redacción y relevancia en el dictado y elaboración de proyectos escritos como reportes y presentaciones mediante el uso de herramientas digitales.

# DISEÑO

Comenzaremos con el paso principal de una base de datos, diseñar el Modelo Entidad-Relación, comenzando con las principales entidades y sus relaciones entre ellas, además de especificar de los atributos de los mismos. Aquí comparamos como principales entidades a:

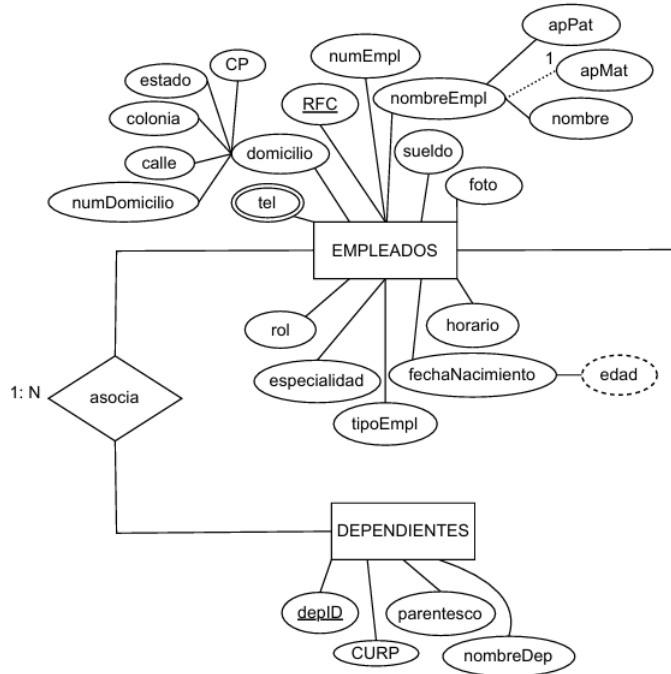
- EMPLEADO (PK RFC)
- DEPENDIENTE (PK Dep\_ID)
- PRODUCTO (PK PROD\_ID)
- DETALLEORDEN (PK DETORD\_ID)
- ORDEN (PK FOLIO)
- CLIENTE (PK RFC)



Comenzaremos con las relaciones.  
Primero con la relación Empleado-Dependiente.

Tipo de relación: Uno a Muchos (1:M)

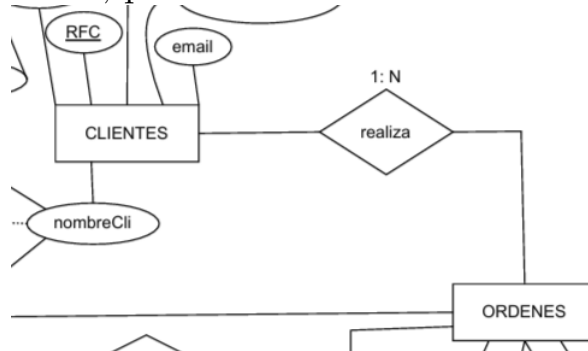
Descripción: Un empleado puede tener múltiples dependientes, pero un dependiente solo está asociado a un solo empleado.



Relación Cliente-Orden

Tipo de relación: Uno a Muchos (1:M)

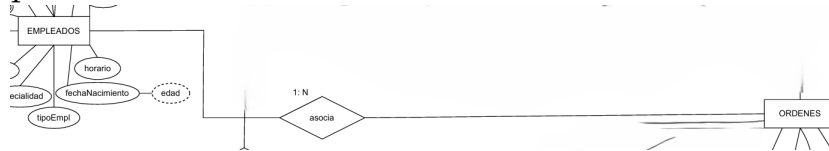
Descripción: Un cliente puede realizar múltiples ordenes, pero una orden está asociada a un solo cliente.



### Relación Empleado-Orden

Tipo de relación: Uno a Muchos (1:M)

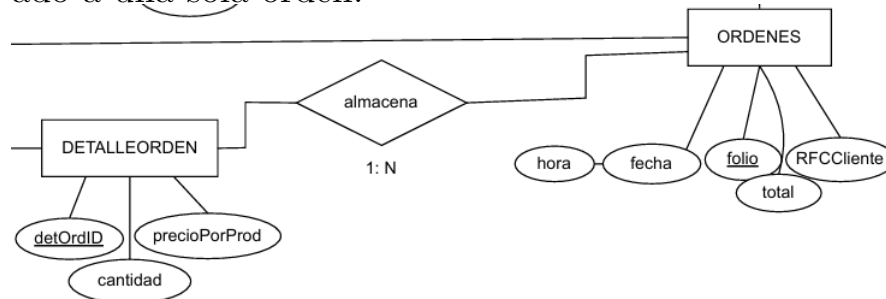
Descripción: Un empleado puede estar asociado a múltiples órdenes (como ejemplo, el empleado que tomó la orden), pero una orden esta asociada a un solo empleado.



### Relación Orden-DetalleOrden

Tipo de relación: Uno a Muchos (1:M)

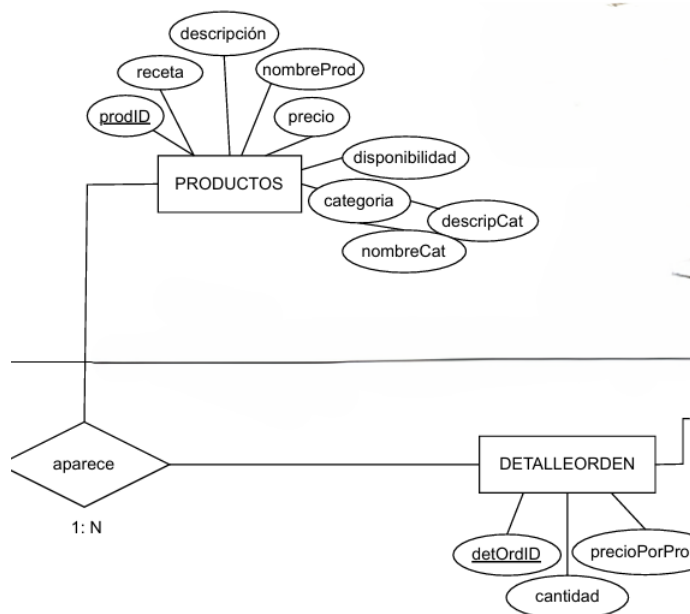
Descripción: Una orden puede tener múltiples detalles de orden, pero un detalle de orden solo está asociado a una sola orden.



### Relación Producto-DetalleOrden

Tipo de relación: Uno a Muchos (1:M)

Descripción: Un producto puede aparecer en múltiples detalles de orden, pero un detalle de orden está asociado solo a un producto.



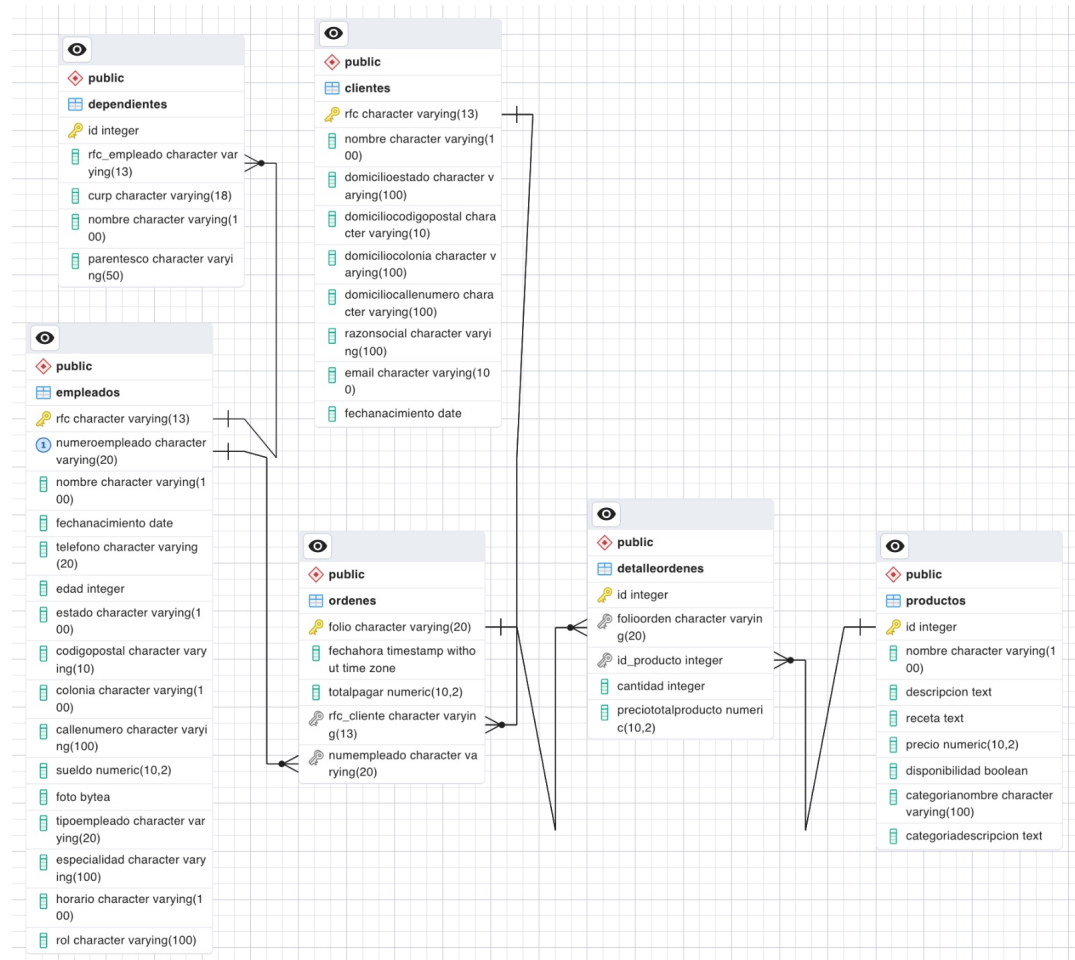
Dado que ya determinamos nuestras principales relaciones entre nuestras entidades y asignamos la cardinalidad, nuestras relaciones quedarían de la siguiente manera.

### Diagramación de las Relaciones

Visualizando las relaciones:

- **\*Empleados\***  $\leftarrow (1:N) \rightarrow$  **\*Dependientes\***
  - Un empleado tiene múltiples dependientes.
- **\*Clientes\***  $\leftarrow (1:N) \rightarrow$  **\*Ordenes\***
  - Un cliente realiza múltiples órdenes.
- **\*Empleados\***  $\leftarrow (1:N) \rightarrow$  **\*Ordenes\***
  - Un empleado está asociado a múltiples órdenes.
- **\*Ordenes\***  $\leftarrow (1:N) \rightarrow$  **\*DetalleOrdenes\***
  - Una orden contiene múltiples detalles de orden.
- **\*Productos\***  $\leftarrow (1:N) \rightarrow$  **\*DetalleOrdenes\***
  - Un producto aparece en múltiples detalles de orden.

# MODELO RELACIONAL



# Código

Creación de la tabla Empleado

## Creación de la tabla Empleados

```
CREATE TABLE Empleados (  
    RFC VARCHAR(13) PRIMARY KEY,  
    NumeroEmpleado VARCHAR(20),  
    Nombre VARCHAR(100),  
    FechaNacimiento DATE,  
    Telefono VARCHAR(20),  
    Edad INTEGER,  
    Estado VARCHAR(100),  
   CodigoPostal VARCHAR(10),  
    Colonia VARCHAR(100),  
    CalleNumero VARCHAR(100),  
    Sueldo DECIMAL(10, 2),  
    Foto BYTEA,  
    TipoEmpleado VARCHAR(20),  
    Especialidad VARCHAR(100),  
    Horario VARCHAR(100),  
    Rol VARCHAR(100)  
);
```

Tomaremos como llave primaria el RFC, que además debe ser único e irrepetible.

Creación de la tabla Dependiente

### Creación de la tabla Dependientes

```
CREATE TABLE Dependientes (  
    ID SERIAL PRIMARY KEY,  
    RFC_Empleado VARCHAR(13) REFERENCES Empleados(RFC),  
    CURP VARCHAR(18),  
    Nombre VARCHAR(100),  
    Parentesco VARCHAR(50)  
);
```

Tomaremos como llave primaria el ID del dependiente.

Creación de la tabla Producto

### Creación de la tabla Productos

```
CREATE TABLE Productos (  
    ID SERIAL PRIMARY KEY,  
    Nombre VARCHAR(100),  
    Descripcion TEXT,  
    Receta TEXT,  
    Precio DECIMAL(10, 2),  
    Disponibilidad BOOLEAN,  
    CategoriaNombre VARCHAR(100),  
    CategoriaDescripcion TEXT  
);
```

Tomaremos como llave primaria el ID del producto.



Creación de la tabla Ordenes

### Creación de la tabla Ordenes

```
CREATE TABLE Ordenes (  
    Folio VARCHAR(20) PRIMARY KEY,  
    FechaHora TIMESTAMP,  
    TotalPagar DECIMAL(10, 2),  
    RFC_Cliente VARCHAR(13) REFERENCES Clientes(RFC),  
    NumEmpleado VARCHAR(20) REFERENCES Empleados(NumeroEmpleado)  
);
```

Tomaremos como llave primaria el FOLIO de la orden.

Creación de la tabla DetalleOrden

### Creación de la tabla DetalleOrdenes

```
CREATE TABLE DetalleOrdenes (  
    ID SERIAL PRIMARY KEY,  
    FolioOrden VARCHAR(20) REFERENCES Ordenes(Folio),  
    ID_Producto INTEGER REFERENCES Productos(ID),  
    Cantidad INTEGER,  
    PrecioTotalProducto DECIMAL(10, 2)  
);
```

Tomaremos como llave primaria el ID de los Detalles.

Creación de la tabla Cliente

## Creación de la tabla Clientes

```
CREATE TABLE Clientes (  
    RFC VARCHAR(13) PRIMARY KEY,  
    Nombre VARCHAR(100),  
    DomicilioEstado VARCHAR(100),  
    DomicilioCodigoPostal VARCHAR(10),  
    DomicilioColonia VARCHAR(100),  
    DomicilioCalleNumero VARCHAR(100),  
    RazonSocial VARCHAR(100),  
    Email VARCHAR(100),
```

Tomaremos como llave primaria el RFC, que además debe ser único e irrepetible.

Ahora implementaremos las llaves foráneas.

## Agregar clave foránea a la tabla Dependientes

```
ALTER TABLE Dependientes  
ADD CONSTRAINT fk_empleado_rfc  
FOREIGN KEY (RFC_Empleado)  
REFERENCES Empleados(RFC);
```

## Agregar clave foránea a la tabla Ordenes

```
ALTER TABLE Ordenes
ADD CONSTRAINT fk_cliente_rfc
FOREIGN KEY (RFC_Cliente)
REFERENCES Clientes(RFC);

ALTER TABLE Ordenes
ADD CONSTRAINT fk_empleado_num
FOREIGN KEY (NumEmpleado)
REFERENCES Empleados(NumeroEmpleado);
```

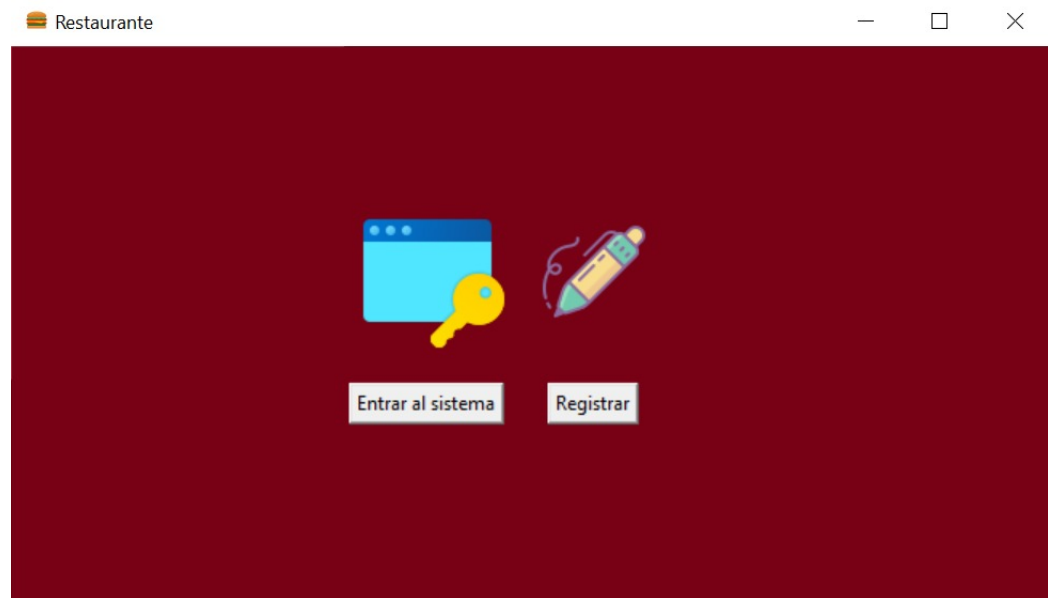
Y finalmente creamos los índices.

## Crear índices

```
CREATE INDEX idx_rfc_empleado ON Empleados(RFC);
CREATE INDEX idx_num_empleado ON Empleados(NumeroEmpleado);
CREATE INDEX idx_fecha_orden ON Ordenes(FechaHora);
```

# IMPLEMENTACIÓN

Aquí comenzamos con la parte de la aplicación móvil. Primero tendremos una pantalla donde nos permitirá ingresar al sistema, si ya estamos dados de alta (registrados) o poder registrarnos como nuevo usuario.



Como nosotros iniciaremos un nuevo empleado lo registraremos como una primera vez, llenaremos todos los campos y listo, o también podemos buscar a empleado anteriormente registrados mediante su ID.

Registro de empleados

Agregar empleado

Nombre:

Edad:

Dirección:

Registrar

Buscar por ID:  Buscar

- 1 ryan {San Francisco} 23
- 2 joe LA 30
- 4 cameron {Los Angeles} 28
- 5 Jhon SF 23
- 6 Marua NY 30
- 7 Roberto {San Francisco} 30

Una vez ingresado al sistema, tendremos tres opciones a elegir.

- Ordenar: el mesero tomara la orden.
- Platos: el personal podra ver la informacion de los platos (disponibiidad, descripcion y receta).
- Bebidas: similar a los platos.



Ordenar



Platillos



Bebidas

# CONCLUSIONES

- *Mata Ramírez Andrea: El diseño y consulta de bases de datos presentan varios desafíos y aprendizajes. Convertir los requisitos del negocio a modelos de datos robustos y funcionales es complejo, ya que requiere un entendimiento profundo de las relaciones entre entidades y de cómo representarlas adecuadamente. Manejar correctamente la notación es crucial para manejar las tablas con precisión, lo que puede llegar a ser desafiante. Diseñar tablas intermedias para manejar relaciones muchos a muchos, como la relación entre conductores y vehículos, implica una planificación cuidadosa para mantener la integridad referencial y la eficiencia de las consultas. El uso del álgebra relacional ayuda a estructurar el pensamiento y definir claramente los pasos necesarios para obtener los datos deseados, facilitando así la traducción a SQL. La distinción entre operaciones de selección y proyección en álgebra relacional contribuye a entender y descom-*

*poner mejor las consultas complejas. Por su parte, SQL proporciona una gama completa de funcionalidades para manejar restricciones, relaciones y agregaciones, permitiendo generar informes detallados y precisos. La capacidad de realizar joins y subconsultas es crucial para combinar datos de múltiples tablas y realizar cálculos complejos, como el cálculo de precios promedio históricos o el filtrado basado en condiciones agregadas. Finalmente, realizar pruebas exhaustivas del diseño de la base de datos asegura que los resultados sean precisos y que las consultas sean eficientes, ayudando a identificar y corregir errores tempranamente. En resumen, el proceso de diseñar y consultar bases de datos requiere un equilibrio entre la teoría y la práctica (SQL). Superar los retos iniciales en la comprensión de los requisitos y la construcción de modelos correctos lleva a la creación de bases de datos robustas y consultas eficientes, esenciales para el éxito de aplicaciones complejas como sistemas de movilidad y catálogos de libros. La claridad en la notación y la precisión en la imple-*



*mentación son claves para alcanzar resultados precisos y optimizados.*

- *Rojo Ramirez Luis Edgar: El desarrollo de este sistema de gestión para un restaurante representa un avance significativo en la digitalización y eficiencia operativa del negocio. Utilizando tecnologías modernas como PostgreSQL para la gestión de bases de datos y Python con la biblioteca Tkinter para la interfaz gráfica, se ha creado una solución robusta y fácil de usar que mejora la gestión de empleados y la operatividad diaria. Este proyecto no solo facilita el registro y seguimiento de los empleados, sino que también optimiza procesos clave, contribuyendo a una experiencia más fluida y profesional tanto para los empleados como para los clientes del restaurante.*
- *Rosas Cañada Abraham: La evolución de este proyecto ha sido todo menos sencilla; Una experiencia que nos ha enriquecido y desafiado a la vez, ha llegado a estar simbolizada por el flujo y reflujo de dificultades, desafíos y éxitos. Sin embargo, son estos*

*elementos (no importa lo dispares o abrumadores que sean a primera vista) los que han desempeñado papeles fundamentales en nuestra comprensión y eventual dominio en el ámbito del diseño e implementación de bases de datos. Uno de los mayores inconvenientes que encontramos tuvo su origen en la naturaleza intrincada de los requisitos del sistema. Un volumen insuperable de datos heterogéneos, que iban desde detalles granulares sobre los empleados hasta detalles específicos sobre productos y pedidos, se cernía sobre nosotros. Su gestión requirió una cuidadosa orquestación que condujo a lo que casi podría describirse como una creación ingeniosa: el esquema de la base de datos. Cada rincón tenía que estar imbuido de significado a través de una definición adecuada; cada conexión forjada necesitaba normalización en aras de la uniformidad. Esta no fue una tarea completada de una sola vez; más bien, se desarrolló como un proceso iterativo cargado de revisiones debido a errores en el camino. Uno de los principales obstáculos que en-*

frentó fue la integración de funciones adicionales en el sistema. Por ejemplo, actualizar automáticamente los totales de los pedidos y validar la disponibilidad del producto a través de PostgreSQL requirió un desarrollo y una depuración cuidadosa de los procedimientos, activadores y funciones almacenados. Entre los desafíos clave también estaba cómo administrar las imágenes de los empleados de manera efectiva; esto nos impulsó a explorar las mejores prácticas en el almacenamiento de archivos binarios en una base de datos que luego llevaron a aplicarlas de manera racional. De la misma manera, generar vistas y consultas específicas para necesidades de reporte y visualización como la vista de platos más vendidos o el reporte de ventas por periodo fueron esfuerzos fuertemente enfocados a la optimización y puesta a punto, buscando una postura perfeccionista de las consultas SQL diseñadas para estas tareas. La integración del sistema con una aplicación web para la consulta de información de empleados y la generación automática de vistas para la facturación también

*presentó retos técnicos y de coordinación, especialmente en cuanto a la sincronización de datos y la implementación de interfaces de usuario intuitivas y funcionales. Terminando este proyecto no solo nos permitió aplicar y consolidar los conocimientos adquiridos en el curso, sino que también nos proporcionó una experiencia práctica valiosa en el diseño, implementación y gestión de bases de datos en un entorno real. Las lecciones aprendidas y las habilidades desarrolladas durante este proceso serán fundamentales para nuestros futuros proyectos y carreras profesionales.*

- *Vargas Luna José Ángel:*  
*La elaboración de este proyecto fue una buena propuesta para el entendimiento de las bases de datos, además de ponernos como un reto la elaboración de una base en fase cero y poder implementarla y diseñarla a nuestro criterio, también contandolo como una posible venta de un cliente o manejo al mundo laboral, además de permitirnos salirnos de nuestra zona de confort en cuanto un diseño un poco más profundo a las bases de datos.*

*Uno de los principales problemas que tuvimos fue la asignación de roles y un poco la falta de motivación del comienzo del proyecto.*

- *Zuckerman Cisneros Stephan: Este proyecto nos brindó la oportunidad de reunir y aplicar todos los conocimientos adquiridos durante el curso, ya que se fue trabajando desde cero hasta un avance significativo, el cuál nos dió oportunidad de ver cómo se resuelven los problemas del mundo real (en materia de bases de datos). Este proyecto me ayudó a resolver un problema siguiendo un método tanto estratégico como estricto, hubo que tener mucho cuidado en cada una de las etapas, desde describir las restricciones de negocio hasta llegar a la implementación física, donde fue todo un reto aprender un nuevo lenguaje de programación para poder implementar nuestras relaciones definidas conceptualmente . Además de que resaltamos la importancia de las bases de datos y su importancia en la resolución de problemas que involucren sistemas de información.*