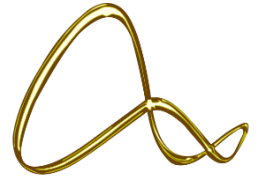




Instituto Politécnico Nacional



Unidad Profesional Interdisciplinaria en Ingeniería y  
Tecnologías Avanzadas

# Ingeniería Web

Alumnos

- Orozco Rojas Francisco
- Valdés García Karen Lilian

<b>Requerimientos funcionales</b>	<b>Descripción</b>
Registro de usuarios	
1. Gerente	Tendrá permitido acceso a la base de datos de los empleados y pedidos, así como podrá agregar nuevos empleados.
2. Mesero	Solo podrá ver el menú y acceder a los productos para comprarlos una vez que haya accedido a su cuenta registrada. No puede hacer pedidos si no está registrado el id en la base de datos.
3. Visitante	Solo puede ver la página de Login, pero sin poder acceder ya que no tiene una cuenta registrada.
Gestión de usuarios	
4. Registro de usuarios	Los meseros deben tener un registro en la base de datos para acceder a sus funciones. Deben contar con nombre, id, edad, sucursal y puesto.
Gestión de productos	
5. Registro de productos	Los productos deberán contar con información como precio y nombre.
Carrito de compras	
6. Visualización del carrito	El mesero debe poder agregar productos al carrito de compras. El mesero debe poder ver el contenido del carrito de compras. El mesero debe poder eliminar platillos de su carrito de compras. El mesero debe poder realizar la compra de todos los componentes en el carrito.

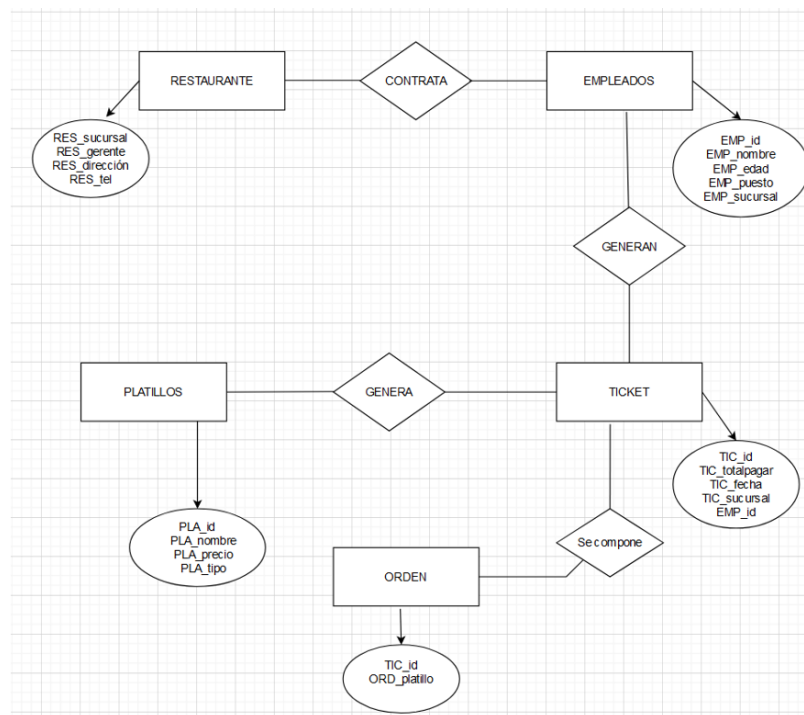
Tabla 1. Tabla de requerimientos funcionales (RF).

<b>Requerimientos No funcionales</b>	<b>Descripción</b>
Seguridad	

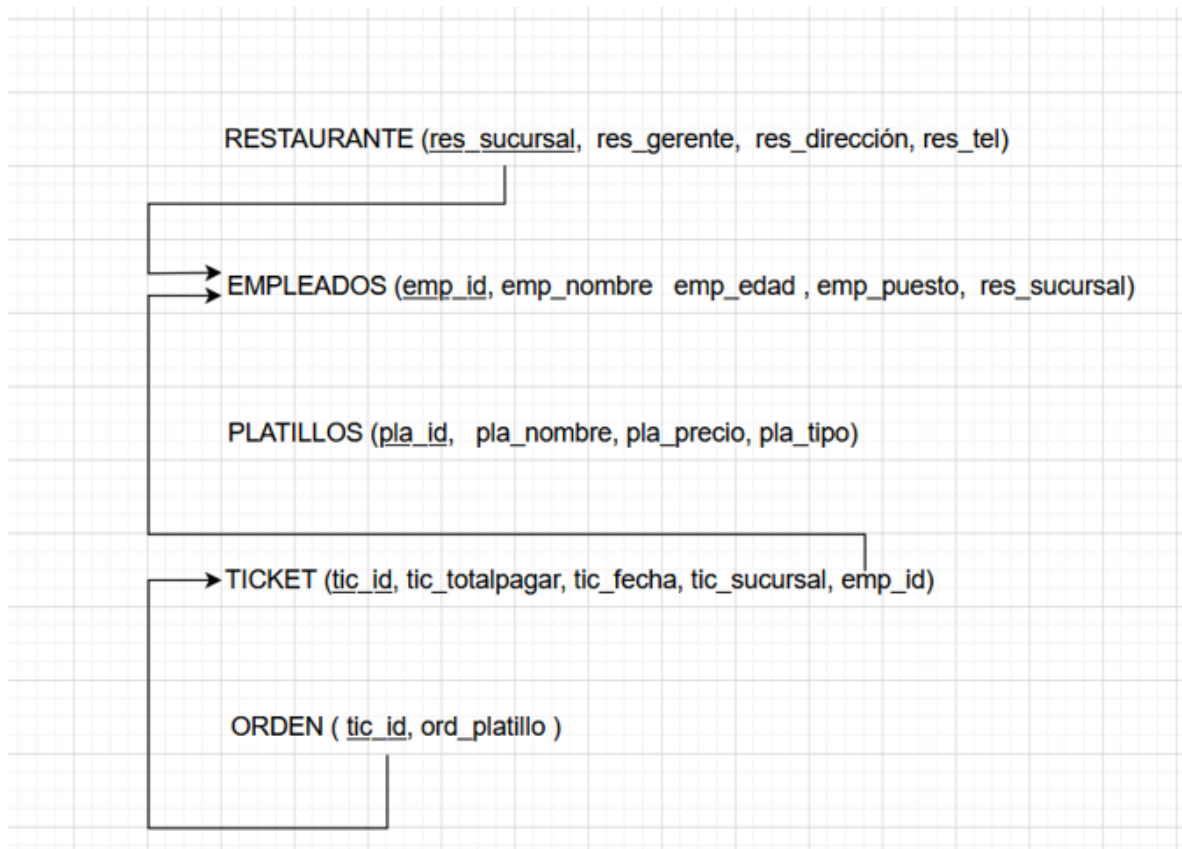
1. Autenticación	El sistema debe abrir diferentes vistas dependiendo del rol de quien lo usa.
2. Control de acceso	El sistema debe implementar un control de acceso, asegurando que los usuarios solo puedan acceder a funciones y datos según su rol (mesero, gerente).
Interfaz	
3. Manejo	La interfaz y el manejo del sitio web debe ser fácil e intuitivo para que el usuario pueda manejarlo.
4. Consistencia visual	El diseño visual de la interfaz debe ser consistente en todas las páginas, utilizando los mismos colores, fuentes y distribución para evitar confusiones al usuario.

Tabla 2. Tabla de requerimientos no funcionales (RNF).

## Modelo Entidad-Relación



## Modelo Relacional



## Arquitectura del sistema

### 1. Estructura General del Sistema Frontend

Utiliza React para construir la interfaz de usuario. React permite la renderización del lado del servidor (SSR), lo que mejora el rendimiento y la SEO de tu aplicación.

Backend: Se utilizó el lenguaje JavaScript

Base de Datos: Microsoft SQL Server se utilizará para almacenar datos estructurados a través de XAMPP.

### 2. Configuración de React: Páginas y Componentes

Crea páginas y componentes en Next.js que se conecten a tu backend para obtener y enviar datos. API Routes: Utiliza las rutas API de Next.js para manejar las solicitudes del frontend. Estas rutas pueden actuar como intermediarias entre tu aplicación y SQL Server.

### 3. Conexión a Microsoft SQL Server Conexión desde el Backend:

Para establecer una conexión desde el backend (como Node.js o Next.js) con Microsoft SQL Server utilizando XAMPP como entorno de desarrollo:

- i. XAMPP se utiliza para alojar el servidor backend (por ejemplo, Node.js), pero no interactúa directamente con SQL Server.
- ii. El backend se conecta a SQL Server mediante un cliente como el paquete mssql en Node.js, usando credenciales (usuario, contraseña, servidor, puerto).
- iii. El backend envía consultas SQL al SQL Server, que procesa la solicitud y devuelve los resultados.
- iv. El servidor web de XAMPP puede servir las solicitudes HTTP, pero la comunicación con SQL Server ocurre a través del backend, que maneja la lógica y consulta la base de datos.
- v. El flujo general es: el frontend hace una solicitud, el backend consulta SQL Server y luego devuelve los resultados al frontend.

Este proceso permite que la aplicación backend sirva de intermediaria entre el frontend y la base de datos, facilitando la gestión de datos.

## Modelado UML

