

Documetación de proyecto final de la asignatura de bases de datos

JEDAI Industry S.A de C.V

November 2023

1 Introducción

El desafío de digitalizar las operaciones de un restaurante involucra el desarrollo de un sistema informático multifacético. En la primera parte, la cual es fundamental de este sistema es el diseño de una base de datos que gestione eficientemente la información relevante. Se plantea el almacenamiento y manejo de datos de empleados, platillos y bebidas ofrecidos, así como la administración de órdenes y clientes.

La base de datos requerirá la captura detallada de información personal de los empleados, incluyendo su RFC, número de empleado, datos personales, roles laborales, sueldo y detalles específicos según su puesto (especialidad para cocineros, horario para meseros, rol para administrativos). Además, se debe contemplar la relación entre empleados y sus dependientes, registrando su CURP, nombres y parentesco.

En cuanto a los platillos y bebidas ofrecidos por el restaurante, se necesitará un registro que incluya descripción, nombre, receta, precio, disponibilidad y su categorización. La gestión de órdenes requerirá un seguimiento detallado, incluyendo información del mesero, folio de orden, fecha, productos solicitados, cantidades y monto total.

Además, la base de datos debe cumplir con funcionalidades específicas, como la actualización de totales al agregar productos a la orden, generación de índices justificados por su relevancia y eficiencia, consulta de órdenes por número de empleado con validación específica, entre otras.

El diseño de la base de datos debe obedecer a los principios de diseño, considerar la estructura de datos detallada y asegurar un manejo óptimo de la información, garantizando su integridad y coherencia.

Mientras que en la segunda parte, posterior al diseño, se contempla la implementación de funcionalidades adicionales. Se plantean opciones como la visualización de información mediante una aplicación móvil o web que permita acceder a datos generales de los empleados, incluyendo sus fotografías. Además, se propone el desarrollo de un dashboard para visualizar ingresos mensuales, productos sin existencia y horarios pico de órdenes.

Otra posibilidad es la integración de archivos de texto para cargar información diaria a otra base de datos, asegurando un flujo controlado y validaciones en caso de errores durante el proceso de inserción.

El reto abarca desde el diseño y la implementación de una base de datos robusta para gestionar la complejidad de la operación de un restaurante hasta la creación de funcionalidades complementarias que optimicen la visualización y gestión de la información.

2 Plan de Trabajo

En la presente sección del trabajo, explicaremos de manera detallada la descripción general del proyecto, también describiremos el plan de acción que emplearemos para llevarlo a cabo.

Fase 1: Diseño de la Base de Datos

Análisis de requerimientos: Detallar necesidades específicas para empleados, platillos, bebidas, órdenes y clientes.

Estructura de la base de datos: Definir tablas para empleados, platillos, bebidas, órdenes y relaciones entre ellas.

Diseño detallado de campos: Especificar campos clave, como RFC, número de empleado, recetas, precios, etc.

Integridad y coherencia: Garantizar la cohesión de datos y mantener la integridad con relaciones y restricciones.

Fase 2: Implementación y Funcionalidades Adicionales

Desarrollo de Funcionalidades Específicas: Implementar la lógica para actualizar totales, generar índices relevantes y validar consultas.

Exploración de Opciones de Visualización: Evaluar viabilidad de una app/web para datos generales de empleados y fotografías.

Dashboard para Métricas Clave: Desarrollar un dashboard para visualizar ingresos mensuales, productos sin existencia y horarios pico.

Integración con Otros Sistemas: Evaluar y desarrollar la integración de archivos de texto para cargar datos a otra base con validaciones.

Fase 3: Evaluación y Optimización Continua

Pruebas y Retroalimentación: Realizar pruebas exhaustivas para garantizar funcionalidad y corrección.

Optimización de Rendimiento: Identificar áreas para mejorar la eficiencia y rendimiento del sistema.

Capacitación y Documentación: Preparar material para capacitar al personal y documentar el sistema para futuras referencias.

Este plan abarca desde la concepción de la base de datos hasta la implementación de características adicionales para mejorar la gestión y visualización de información en el restaurante. Cada fase se complementará con pruebas y ajustes para garantizar un sistema sólido y eficiente.

3 Diseño

1. Entendimiento del Problema: Analizar detalladamente los requisitos del restaurante. Identificar las entidades y relaciones clave.

2. Modelo de Entidad-Relación (MER): Crear un modelo MER que refleje las entidades y relaciones del restaurante. Definir atributos y cardinalidades. Identificar posibles redundancias y normalizar el modelo.

https://drive.google.com/file/d/1ODFRNCqOk6dcYo6lsZgiChc2FSz-RlQd/view?usp=drive_link

3. Diseño de la Base de Datos: Convertir el modelo MER a un modelo relacional MR. Definir tablas, atributos y claves primarias/foráneas. Considerar la integridad referencial y restricciones de la base de datos.

4. Implementación en PostgreSQL: Crear la base de datos en PostgreSQL. Ejecutar scripts SQL para definir tablas, relaciones y restricciones.

5. Inserción de Datos de Ejemplo: Insertar datos de prueba para validar la estructura de la base de datos.

6. Desarrollo de Funcionalidades: Implementar funciones en PgSQL para manejar la lógica de negocio. Crear vistas para las funcionalidades específicas (facturas, platillo más vendido, etc.).

7. Pruebas y Validación: Realizar pruebas exhaustivas para garantizar la integridad y funcionalidad de la base de datos. Validar las consultas y funciones desarrolladas.

4 Presentación

La modalidad de conexión hacia la base de datos PostgreSQL desde una aplicación en Python para este proyecto implica el uso de bibliotecas específicas como `psycopg2` o `SQLAlchemy` para establecer la conexión y realizar operaciones de lectura y escritura en la base de datos desde el código Python.

En primer lugar, se diseñará la base de datos en PostgreSQL según los requerimientos del restaurante, creando tablas para cada entidad y estableciendo relaciones entre ellas. La implementación de la base de datos se llevará a cabo utilizando consultas SQL ejecutadas desde Python a través de la conexión establecida.

Para interactuar con la base de datos desde la aplicación Python, se emplearán consultas SQL para insertar, actualizar, eliminar y consultar datos según sea necesario. Por ejemplo, al agregar un producto a una orden, se realizará una inserción en la tabla correspondiente y se actualizarán los totales pertinentes.

Las consultas de lectura se utilizarán para obtener información detallada según los requisitos, como mostrar la información de los empleados, los detalles del platillo más vendido, obtener los productos no disponibles, generar vistas similares a facturas de órdenes, entre otros.

El código está escrito en Python utilizando la biblioteca Tkinter para crear una interfaz gráfica de usuario (GUI). El programa tiene como objetivo principal gestionar pedidos y empleados en un restaurante, conectándose a una base de datos PostgreSQL.

Partes del código:

Importación de bibliotecas: Se importan varias bibliotecas necesarias para la interfaz gráfica (Tkinter), manejo de fechas (tkcalendar), manipulación de imágenes (PIL), y la conexión con PostgreSQL (psycopg2).

Conexión a la base de datos: Se establece la conexión con una base de datos PostgreSQL utilizando psycopg2.

Funciones: insertarorden(): Esta función se encarga de insertar órdenes en la base de datos, obteniendo los datos de la GUI. obteneralimentosdisponibles(): Realiza una consulta para obtener los alimentos disponibles en el restaurante. obtenernumeroempleado(): Realiza una consulta para obtener los números de empleado.

Creación de ventanas y elementos gráficos: Se definen diferentes funciones para crear ventanas y elementos gráficos específicos para el mesero y el administrativo. Por ejemplo, la función 'ventanamesero()' crea una ventana para el mesero donde puede registrar pedidos.

Interfaz gráfica principal: Se crea la ventana principal ('root') donde se coloca un menú lateral y se definen acciones para los botones ('btnmesero'). Se crea la función 'ventanaadministrativo()' que abre una ventana para la gestión de empleados por parte de un administrativo.

Este programa utiliza Tkinter para crear una interfaz gráfica que interactúa con una base de datos PostgreSQL mediante consultas SQL ejecutadas a través de la biblioteca psycopg2 en Python. Ofrece funcionalidades para registrar órdenes, consultar alimentos disponibles, y gestionar empleados, entre otras acciones propias de un sistema de restaurante.

5 Implementación

El código crea y modifica diversas tablas, vistas y funciones en una base de datos de un restaurante simulado.

Tablas: Se crean tablas como 'empleado', 'cocinero', 'mesero', 'administrativo', 'dependiente', 'categoria', 'platillo', 'bebida', 'orden', 'detalle_orden', 'cliente', entre otras, consusrespectivos atributos.

Vistas: Se definen vistas ('v_platillos_mas_vendidos', 'v_productos_disponibles', 'v_facturas_orden') que muestran información relevante.

Funciones: Se crean funciones almacenadas ('agregar_producto_orden', 'estadisticas_mesero') para realizar operaciones comunes.

Índices: Se agregan índices para mejorar el rendimiento en búsquedas.

Modificaciones: Se realizan modificaciones a la estructura de las tablas existentes, como añadir columnas ('rf_cliente_a_orden') y establecer restricciones de clave foránea.

Es un modelo de base de datos detallado para un restaurante que incluye la gestión de empleados, clientes, productos, órdenes y estadísticas relacionadas con las ventas y el personal. Cada componente está diseñado para gestionar de manera eficiente y efectiva las operaciones de un restaurante.

6 Conclusiones

Girón Reyes Ingrid Ailin: El proyecto enfrentó desafíos técnicos y de diseño al estructurar la base de datos y garantizar la conectividad y funcionalidad del sistema. Los logros incluyen una conexión estable con PostgreSQL, un diseño funcional de la base de datos y la implementación exitosa de ciertas funcionalidades. La colaboración entre los equipos de desarrollo y la comprensión profunda de las necesidades del restaurante fueron clave para el éxito del proyecto. Las partes que más se me dificultaron fueron las siguientes: Conexión y manejo de la base de datos: Garantizar una conexión estable con la base de datos y realizar operaciones de manera eficiente y segura puede ser un desafío técnico. Interfaz de usuario: Desarrollar una interfaz gráfica intuitiva y funcional para la interacción del personal del restaurante puede requerir tiempo y habilidades de diseño.

Méndez Maldonado Jorge: La complejidad de adaptar todos los datos, desde los detalles de los empleados hasta la gestión de órdenes y productos, implicó una cuidadosa estructuración y sincronización de la información en la base de datos. La transformación de estos datos para asegurar su coherencia y la correcta representación en el entorno de la base de datos fue un desafío técnico considerable. La correcta implementación de consultas SQL, actualizaciones y validaciones desde un entorno Python requirió un sólido entendimiento de ambas tecnologías y la capacidad de traducir adecuadamente las operaciones a la lógica de la base de datos. En lo personal, lo que se me complicó fue adaptar todos los datos y realizar las operaciones sql desde python.

Pérez Jiménez Erandi: En conclusión, las dificultades encontradas durante la conexión entre PostgreSQL y Python, así como en la implementación de la interfaz gráfica, destacan la naturaleza desafiante pero enriquecedora del desarrollo de software . Estos obstáculos, aunque inicialmente frustrantes, ofrecen valiosas lecciones y oportunidades de mejora tanto a nivel técnico como de trabajo en equipo, fue difícil solucionar estos pormenores pero gracias a la ayuda de todos los integrantes del equipo se le pudo dar solución.

Puente Estrada Rodrigo Abdiel: El desarrollo del proyecto enfrentó diversos retos técnicos y de diseño, especialmente al estructurar la base de datos y asegurar su conectividad para lograr un sistema completamente funcional. Se lograron hitos significativos, entre ellos una conexión sólida con PostgreSQL, un diseño efectivo de la base de datos y la implementación exitosa de varias funcionalidades clave. Garantizar una conexión estable con la base de datos y ejecutar operaciones de manera eficiente y segura se reveló como un desafío técnico destacado. Superar obstáculos relacionados con la optimización del rendimiento y la seguridad fue crucial para el éxito del proyecto. El desarrollo de una interfaz gráfica que fuera intuitiva y funcional para el personal del restaurante representó otro conjunto de desafíos. Se requirió tiempo y habilidades de diseño para lograr una interfaz que facilitara las operaciones diarias del personal y mejorara la experiencia general.

Serrano Martinez Diana: Se me dificulto mucho el manejo de la base de datos en Postgres ya que de alguna manera siento que Oracle es mucho más

fácil. La interpretación del problema desde el inicio se me hizo bastante sencilla, ya que te daba los puntos específicos a considerar, el manejo de mapear generalización, especialización, traslape, etc se me dificultó un poco pero con ayuda del equipo se logró, de igual manera la programación de la app se me hizo muy compleja. Tuvimos que ser muy muy cuidadosos al pioneer de manera correcta los constraints de las tablas para evitar redundancias y/o ambigüedades posibles. Concluimos de manera exitosa el proyecto.