

# Proyecto final de la asignatura de bases de datos

Ing. Fernando Arreola

Marzo 2024

## Objetivo

El alumno analizará una serie de requerimientos y propondrá una solución que atienda a los mismos, aplicando los conceptos vistos en el curso.

## Descripción del problema

El problema se divide en dos partes:

### Parte uno:

Consiste en el diseño de una base de datos.

Un restaurante desea digitalizar su forma de operación, para ello se desarrollará un sistema informático que constará de varios módulos. El que corresponde a la implementación de la base de datos deberá atender el siguiente requerimiento:

Se debe almacenar el RFC, número de empleado, nombre, fecha de nacimiento, teléfonos, edad, domicilio, sueldo; de los cocineros su especialidad, de los meseros su horario y de los administrativos su rol, así como una foto de los empleados y considerar que un empleado puede tener varios puestos. Es necesario tener registro de los dependientes de los empleados, su curp, nombre y parentesco. Se debe tener disponible la información de los platillos y bebidas que el restaurante ofrece, una descripción, nombre, su receta, precio y un indicador de disponibilidad, así como el nombre y descripción de la categoría a la que pertenecen (considerar que un platillo o bebida sólo pertenece a una categoría). Debe tenerse registro del folio de la orden, fecha (con hora), la cantidad total a pagar por la orden y registro del mesero que levantó la orden, así como la cantidad de cada platillo/bebida y precio total a pagar por platillo/bebida contenidos en cada orden. Considerar que es posible que los clientes soliciten factura de su consumo, por lo que debe almacenarse su RFC, nombre, domicilio, razón social, email y fecha de nacimiento. Adicional al almacenamiento de información, la base de datos debe atender los siguientes puntos:

- Cada que se agregue un producto a la orden, debe actualizarse los totales (por producto y venta), así como validar que el producto esté disponible
- Crear al menos, un índice, del tipo que se prefiera y donde se prefiera. Justificar el porqué de la elección en ambos aspectos.
- Dado un número de empleado, mostrar la cantidad de órdenes que ha registrado en el día así como el total que se ha pagado por dichas órdenes. Si no se trata de un mesero, mostrar un mensaje de error.
- Vista que muestre todos los detalles del platillo más vendido.
- Permitir obtener el nombre de aquellos productos que no estén disponibles.
- De manera automática se genere una vista que contenga información necesaria para asemejarse a una factura de una orden.
- Dada una fecha, o una fecha de inicio y fecha de fin, regresar el total del número de ventas y el monto total por las ventas en ese periodo de tiempo. Nota: Con producto se hace referencia a los alimentos y bebidas.

Tomar en cuenta las siguientes consideraciones:

- Puede haber distintas soluciones al problema.
- Los requerimientos enlistados anteriormente, deberán ser realizados por medio de PostgreSQL, con los elementos que se consideren adecuados para resolverlos.
- El folio de la orden debe tener un formato similar a ORD-001, prefijo ORD, seguido de un guión y un número secuencial.
- Donde este presente el atributo *domicilio*, está compuesto por estado, código postal, colonia, calle y número.
- Donde este presente el atributo *nombre*, está compuesto por nombre, apellido paterno y materno.
- El diseño debe satisfacer todos los principios de diseño, los requerimientos anteriores y un buen manejo de información.

## Parte dos:

Una vez diseñada y lista la base de datos, deberá elegirse la implementación de alguno de los siguientes puntos:

1. Consultar la información general de los empleados, incluyendo su fotografía por medio de una app móvil o web.
2. Generar un dashboard que permita visualizar, al menos:
  - Ingresos del mes

- Platos sin existencia
  - Horas pico en que se registran más órdenes
3. Programar en C o en Java, un objeto que simule ser una tabla de una base de datos, lo que implica que permita ingresar, borrar, actualizar y consultar información.
  4. Ingresar, a partir de archivos de texto, la información registrada durante el día a otra base de datos. Se debe orquestar la inserción de cada archivo y tener validaciones del flujo en caso de errores.

## Entregables

Es necesario entregar lo que se solicita a continuación:

### Documento

Se deberá entregar un documento FORMAL elaborado en Latex, que contenga las siguientes secciones:

- Introducción: Breve descripción del análisis del problema, su(s) objetivos, su propuesta(s) de solución.
- Plan de trabajo: Queda a criterio de cada equipo el grado de detalle que se le dará a esta sección, pero mínimo debe contener descripción general de las actividades a realizar y su correspondiente plan de actividades, así como un breve detalle de qué hizo cada miembro del equipo.
- Diseño: Descripción de lo realizado en las correspondientes fases de diseño de las bases de datos, agregando los resultados de cada una de ellas.
- Implementación: Descripción de funcionamiento y código de los stored procedures, triggers, funciones, etc. Empleados para cumplir con los requerimientos del problema. También debe incluirse el DDL de la base de datos.
- Presentación: Descripción de lo que hace la modalidad seleccionada como forma de conexión hacia la base de datos.
- Conclusiones: Personales, detallando las dificultades, retos, aciertos, etc. que se presentaron en el proyecto.

### Código

A través de github, se deberá crear una carpeta por equipo donde se anexe lo siguiente, asignando nombres ADECUADOS para cada documento/archivo:

- El documento del punto anterior

- Códigos fuente de MER, MR
- Script de creación de la base de datos y tablas
- Script para el agregado de información
- Script de toda la programación a nivel BD
- Códigos de lo implementado como parte de la etapa de presentación
- La presentación a emplear a la hora de exponer

## Exposición

Como parte de la evaluación del curso, se deberá realizar una presentación del proyecto. La idea es simular la presentación de una solución de software a un posible cliente, lo que implica descripción del software, muestras de funcionalidad, solución de dudas, etc. Cada equipo es libre de decidir cómo hacer su presentación, el unico detalle es que debe ser TOTALMENTE formal.

## Puntos a considerar

Se tomará en cuenta lo siguiente, partiendo del porcentaje de calificación estipulado al inicio del semestre:

- Equipos que se copien en alguna parte del proyecto, su calificación del curso será 5.
- Equipo(s) que reprueben el proyecto o la parte de exposición, deberán presentar examen final sin importar lo obtenido en los otros rubros del curso.
- La calificación del primer examen final será 30 % lo obtenido en el proyecto, 70 % lo obtenido en el examen.
- La seriedad, formalidad, presentación y calidad del proyecto, si bien no tendrán un valor numérico específico, pueden jugar a favor o en contra de la calificación de los rubros correspondientes.
- Cualquier punto no contemplado en el presente documento será determinado por el profesor.

## Recomendaciones

No es obligatorio, pero se sugieren las siguientes puntos:

- Delegar correctamente responsabilidades
- No demorar en formar sus equipos y agregar su carpeta en GitHub

- Empleo de herramientas colaborativas
- Empleo de arquitecturas cliente servidor remotas o servicios en la nube
- Redacción y presentación en inglés