



Proyecto Final

Semestre enero-junio 2021

1. Detalles: FAVOR DE LEER CON DETENIMIENTO

- El proyecto debe ser desarrollado por un equipo de 5 a 7 personas. Pueden ser menos personas, pero si hay un desbalance grande entre equipos, algunas personas de equipos con muchos integrantes serán movidas a equipos con menos integrantes.
- La fecha límite para subir el proyecto a Teams es el jueves 10 de junio de 2021 a las 23.59h. Si un equipo quiere hacer la entrega antes de esa fecha lo puede hacer.
- Se deben subir a Teams tres archivos. 1) El código del proyecto como un .zip con todos los directorios necesarios para su funcionamiento, incluyendo los códigos en JavaScript, Vue, HTML y SQL. Llamar a este archivo como **DAS_proyecto_final_codigo.zip** 2) Un archivo .txt con los nombres de los integrantes del equipo, en orden alfabético empezando por el nombre. Llamar a este archivo como **DAS_proyecto_final_equipo.txt**. 3) Una imagen JPG o PNG con el diagrama relacional de la base de datos obtenido a través de la ingeniería inversa con MySQL. Llamar a este archivo como **DAS_proyecto_final_diagrama.PNG**.
- Solo una persona (el responsable del equipo que ustedes nombren) debe subir los archivos.
- La presentación de proyectos será el viernes 11 de junio de 2021 a las 10.00h. Deben tener todo su sistema listo para mostrarse para esa hora. Esto incluye, tener abiertos el editor de código (Visual Studio Code), las terminales, MySQL Workbench y el navegador con la interfaz de usuario de su sistema. En las terminales deben tener sus proyectos *server* y *client* corriendo. Todo debe estar listo para que lo compartan a través de su pantalla en Teams. Si un equipo quiere presentar antes de la fecha, lo puede hacer, previo acuerdo con el profesor.
- La presentación se dará en dos etapas. Una demostración de las funcionalidades de su sistema y una sesión de preguntas a cada miembro (de forma individual).
- La demostración la pueden dividir entre 2 o 3 personas.
- En la sesión de preguntas individuales solo el miembro del equipo al que se le haga la pregunta puede responder. Las preguntas serán sobre la construcción e implementación de las funcionalidades de su sistema. Esto incluye explicaciones sobre el diagrama relacional, el código en SQL y el código en JavaScript
- Tanto el orden de presentación de los equipos como el orden en que se harán las preguntas individuales será de forma aleatoria.

2. Descripción

El proyecto consiste en crear un sistema para controlar la información de pedidos por mesa en un restaurante. El sistema debe incluir dos elementos: el diseño y construcción de una



base de datos con MySQL y la construcción de una aplicación web de tipo cliente-servidor que permita manipular los datos de la base. La aplicación web debe ser construida con Node JS y VUE.

Para la construcción del sistema, vamos a considerar lo siguiente:

- El restaurante tiene un local único (no hay sucursales)
- Hay un número fijo de mesas, las cuales deben tener una distribución (localización, orden o forma de ubicarla) específica dentro del local.
- Un grupo de personas puede ingresar y sentarse en cualquier mesa dependiendo de la capacidad de esta. No es posible juntar mesas.
- Una vez que las personas están a la mesa, el mesero toma la orden de comida y bebida para la mesa.
- Debe ser posible agregar, eliminar o modificar elementos de la orden.
- De esta forma, la orden para la mesa debe incluir la cuenta total de lo que se pidió.
- Debe haber una opción de pagar. Al pagar, el sistema debe generar una nota de compra con el listado de todo lo que se consumió y el total a pagar.
- Debe ser posible saber si la orden de una mesa está abierta o ya está pagada.
- La base de datos debe incluir al menos las siguientes tablas: **mesa, mesero, comida/bebida y orden.**
- Tanto para las comidas como para las bebidas, se debe incluir una categoría. Por ejemplo, para comidas: ensaladas, sopas, carnes, pescados, etc; para bebidas: refrescos, aguas, cervezas, etc.

Adicional a las tablas mínimas mencionadas, se deben agregar las tablas necesarias para que el sistema pueda funcionar.

Para cada tabla:

- a. Definir los atributos necesarios a su consideración. Definir los tipos de datos y aquellos que sean NOT NULL.
- b. Definir los atributos (o combinación de atributos) que puedan ser índices o únicos.
- c. Definir la llave primaria (natural o artificial) y comentar brevemente por qué se decide utilizar una u otra.
- d. Identificar y generar las ligas necesarias entre las tablas (uno-a-uno, uno-a-muchos, muchos-a-muchos). En caso de ligas muchos-a-muchos se deben generar las nuevas tablas correspondientes.
- e. Para cada liga, definir la llave foránea.
- f. Aplicar ingeniería inversa para obtener el diagrama relacional y guárdalo como una imagen con el nombre **DAS_proyecto_final_diagrama.PNG**



La aplicación web debe permitir hacer las operaciones CRUD a través de la interfaz/vista sobre cada tabla que sea necesaria para el funcionamiento del sistema, especialmente la de órdenes.

Adicionalmente, con el sistema debe ser posible realizar al menos las siguientes consultas con el sistema para las órdenes:

- a. Órdenes por mesero para una fecha
- b. Órdenes por mesa para una fecha
- c. Órdenes por fecha
- d. El total de ventas para una fecha

Incluir otras consultas que se consideren relevantes.