

Programación Web - Backend

Trabajo Práctico: Segundo Parcial.

Profesores:

- Ing. Ivan Ríos
- Ing. Gustavo Sosa Cataldo

Este trabajo involucra la implementación del Backend de un sistema informático que será especificado para cada grupo.

Revisiones y entrega

- Entrega del TP:
 - o Lunes 22/05 (día del segundo parcial)

Grupos de alumnos

Los grupos deberán mantenerse con respecto al trabajo del primer parcial.

Observaciones

- 1) Tecnología: Node JS (se puede usar JavaScript o TypeScript indistintamente)
- 2) Para la implementación puede utilizar el IDE de su elección (VSCode, Netbeans, IntelliJ IDEA, etc).
- 3) La base de datos: libre elección (recomendado Postgres)
- 4) Se tiene que utilizar obligatoriamente Sequelize u otro ORM en caso que la base de datos sea relacional.
- 5) Log: Imprimir en consola cada interacción
- 6) **Para los puntos 4 y 5 es obligatorio el desarrollo de una pantalla frontend en cualquier tecnología:** web (html plano con javascript puro, angular, react, vue, etc), mobile nativo (android en java o kotlin, iOS en swift), mobile hibrido (react native, o similares), desktop (swing de java, c# de .net), etc es decir en la tecnología que uds escojan.

Enunciado 1: Sistema de reservas de mesas en restaurantes

Se requiere la implementación de un sistema de reservas de mesas en restaurantes, donde el sistema tendrá que mantener un registro de las mesas con sus coordenadas en cada restaurante y la capacidad de cada una de ellas.

Los módulos a desarrollar son los siguientes:

1) Administración de datos del restaurante (CRUD: POST, PUT, DELETE, GET)

Este módulo contempla la administración de datos del restaurant.

Los datos a almacenar serán los siguientes: id, nombre, direccion.

2) Administración de datos de las mesas (CRUD: POST, PUT, DELETE, GET)

Este módulo contempla la administración de los datos de cada mesa en un restaurante dado. Los datos a almacenar son: id, nombre mesa, id del restaurante al que pertenece, posición (x, y), planta en la que se encuentra la mesa (nro de piso, valor por defecto es 1, y podría tener otros valores en caso que el restaurante cuente con más de un piso, esto ayudará a graficar el mapa de mesas), capacidad de comensales en la mesa.

3) Administración de datos del cliente (POST, GET)

Gestiona los datos del cliente: id, cedula (único), nombre, apellido

4) Reserva de mesas (hay que desarrollar el frontend)

Este módulo permite reservar una mesa. Los datos que se almacenan para una reserva son: id, id de restaurante, id de mesa, fecha, rango de hora, id de cliente, cantidad solicitada. Lo que debe permitir la pantalla es lo siguiente:

- a) el usuario escoge el restaurante
- b) el usuario selecciona la fecha y la hora de reserva. La lista de horas es estática con los siguientes valores: 12 a 13, 13 a 14, 14 a 15, 19 a 20, 20 a 21, 21 a 22, 22 a 23. Nótese que si un cliente quiere reservar por dos horas, entonces la aplicación debería ser capaz de permitir seleccionar más de una opción (por ejemplo para reservar de 12 a 14 debe ser posible seleccionar 12 a 13 y 13 a 14).
- c) al tener seleccionados esos tres parámetros (restaurante, fecha y rangos de hora), el sistema deberá listar todas las mesas aun no reservadas bajo esos criterios, con sus respectivas capacidades
- d) el usuario escoge la mesa que se adapta a la capacidad solicitada para reservar
- e) por la cédula del cliente busca si ya existe, y si no existe lo registra (solo nombre apellido y cédula como se estableció en el punto 3)
- f) guarda los datos

5) Lista de reservas (hay que desarrollar el frontend)

Pantalla para ver la lista de reservas registradas en una pantalla, las cuales se pueden filtrar por: restaurante (obligatorio), fecha (obligatorio) y cliente (opcional), y debe estar ordenados por horario (creciente) y mesa (creciente).

6) Opcional => hasta 15% extra para el final: graficar el mapa de mesas por plantas utilizando el nro de piso y las coordenadas de la posición