

# Programación Web - Backend

## Trabajo Práctico: primer final

Profesores:

- 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 Martes 02/07

### Observaciones

- 1) Tecnología: Node JS
- 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 el punto (3 - Gestión del consumo del cliente en una mesa dada) 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 híbrido (react native, o similares), desktop (swing de java, c# de .net), etc es decir en la tecnología que uds escojan.

## **Enunciado: Extensión del sistema de reservas de mesas en restaurantes + gestión de consumo y emisión de tickets**

Se requiere la implementación de un módulo de gestión de consumo para ser asignado a las mesas.

Los módulos a desarrollar son los siguientes:

### **1) Administración de categoría de productos (CRUD: POST, PUT, DELETE, GET)**

Este módulo contempla la administración de categorías.

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

Ej: Bebida, Platos, Minutas, etc

### **2) Administración de datos de productos (CRUD: POST, PUT, DELETE, GET)**

Este módulo contempla la administración de los productos. Los datos a almacenar son: id, nombre del producto, id de la categoría a la que pertenece, precio de venta (ej: si es un Gaseosa Coca cola, debería asignarse a la categoría Bebidas).

### **3) Gestión del consumo del cliente en una mesa dada**

Gestiona el consumo para una mesa dada en un momento dado.

Se almacena:

Cabecera: id de la mesa, id del cliente, estado (abierto, cerrado<cerrado es cuando el cliente ya pagó su consumo y la mesa se libera para un nuevo cliente>), total, fecha y hora de creación de este consumo, fecha y hora de cierre del consumo de esa mesa

Detalles: id del producto, cantidad

Básicamente la funcionalidad que la pantalla debería permitir es la siguiente:

- 1) el usuario elige una mesa
- 2) si la mesa está ocupada:
  - a) automáticamente recupera el consumo actual de dicha mesa (todos los detalles)
  - b) se debe mostrar los datos de la cabecera: total actual, cliente asociado a la mesa
  - c) se debe permitir cambiar de cliente: si el cliente existe se selecciona de un listado (debe permitir filtrado para la búsqueda), de lo contrario se tiene que tener la funcionalidad para agregar un nuevo cliente
  - d) se debe permitir agregar nuevos detalles de consumo
- 3) si la mesa no está ocupada:
  - a) se muestra la pantalla con datos vacíos listo tanto para agregar detalles de consumo como seleccionar/agregar un cliente a esa mesa
- 4) cerrar una mesa ocupada:
  - a) cambia el estado de abierto a cerrado
  - b) se debe mostrar en pantalla en formato pdf un pequeño ticket con los datos del consumo (emulando una factura con los campos de fecha, cliente, y los detalles y el total abajo)