# Modulo Salón

Atributos:

* mesas: una lista que almacena el estado actual de las mesas en el restaurante, indicando si están ocupadas o no.
* pedidos: una lista que almacena los pedidos que se han realizado en cada mesa.
* cocina: una referencia al módulo de cocina para poder enviar los pedidos.
* simulacion: una referencia al módulo de simulación para poder recibir pedidos simulados.

Métodos:

* ingresarOrden(int numeroMesa, Pedido pedido): este método permite ingresar una orden en la mesa indicada. Recibe como parámetro el número de la mesa y el pedido realizado.
* notificarPedidoListo(Pedido pedido): este método se utiliza para recibir la notificación de que un pedido está listo para ser entregado. Recibe como parámetro el pedido que está listo.
* generarCuenta(int numeroMesa): este método se encarga de generar la cuenta para la mesa indicada, liberando la mesa para ser ocupada nuevamente. Recibe como parámetro el número de la mesa.
* conectarConSimulacion(): este método establece la conexión con el módulo de simulación para recibir pedidos simulados.
* desconectarDeSimulacion(): este método se encarga de cerrar la conexión con el módulo de simulación.

# Módulo Cocina

Atributos:

* ordenesPendientes: una lista que contiene las órdenes pendientes en la cocina.
* moduloSalon: una instancia del módulo Salón para poder notificar sobre las órdenes completadas.

Métodos:

* mostrarOrdenesPendientes(): muestra una lista de las órdenes pendientes en la cocina.
* marcarOrdenComoLista(orden): marca una orden como "lista" y notifica al módulo Salón cuál fue la orden completada.
* notificarSalon(orden): notifica al módulo Salón cuál fue la orden completada.
* completarOrdenes(): permite completar órdenes en orden arbitrario.

# Módulo Simulación

Atributos:

* salon: una referencia al módulo Salón para poder enviarle las órdenes generadas aleatoriamente.
* intervalo: el intervalo de tiempo entre la generación de órdenes aleatorias.
* mesas: un arreglo con las mesas existentes en el restaurante.
* hamburguesas: un arreglo con las hamburguesas predefinidas y sus ingredientes.

Métodos:

* conectarse(): método para establecer la conexión con el módulo Salón.
* desconectarse(): método para cerrar la conexión con el módulo Salón.
* generarOrdenAleatoria(): método para generar una orden aleatoria válida, seleccionando aleatoriamente una mesa y una hamburguesa predefinida, y permitiendo la adición de ingredientes personalizados.
* iniciarSimulacion(): método para iniciar la simulación, generando órdenes aleatorias válidas cada intervalo de tiempo y enviándolas al módulo Salón mediante la conexión establecida.
* detenerSimulacion(): método para detener la simulación.

# Productos

Atributos:

* nombre: el nombre de la hamburguesa.
* ingredientesOficiales: una lista que almacena los ingredientes oficiales de la hamburguesa predefinida.
* precioBase: el precio base de la hamburguesa.
* Adicionales: una lista que almacena los ingredientes adicionales que el cliente ha pedido para personalizar su hamburguesa.
* precioAdicionales: el precio total de los ingredientes adicionales que el cliente ha pedido.
* precioFinal: el precio final de la hamburguesa, que incluye tanto el precio base como el precio de los ingredientes adicionales.

Métodos:

* agregarIngrediente(String ingrediente): este método permite agregar un ingrediente adicional a la hamburguesa.
* quitarIngrediente(String ingrediente): este método permite quitar un ingrediente adicional de la hamburguesa.
* getPrecioBase(): este método devuelve el precio base de la hamburguesa.
* getPrecioAdicionales(): este método devuelve el precio total de los ingredientes adicionales que el cliente ha pedido.
* getPrecioFinal(): este método devuelve el precio final de la hamburguesa.
* getIngredientes(): este método devuelve una lista con los ingredientes oficiales y adicionales de la hamburguesa.
* getNombre(): este método devuelve el nombre de la hamburguesa.

# Ingredientes

Atributos:

* Nombre: nombre del ingrediente
* Precio: precio del ingrediente
* Adicionales: Nombre de los ingredientes adicionales
* PrecioAdicionales: precio por los ingredientes extras