

**Problema de programación, Restaurant.**

**Luis Javier Robles Topete**

**Programación Orientada a Objetos**

**Proyecto Final**



**Problema.**

Un restaurante llamado "Restaurant Navidad" necesita un programa para gestionar su menú, realizar pedidos y manejar las ventas. El restaurante desea ingresar un menú de platillos y cafés que incluya el nombre del platillo, su precio y la cantidad disponible en la cocina. Además, quiere poder mostrar el menú completo a los clientes y proporcionar comentarios específicos de un platillo cuando sea solicitado, para poder realizar un mejor manejo del restaurante completo, se solicita tener tambien los datos de las mesas, como el numero de comezales, el numero de mesa y el mesero que atiende, de la misma manera, se debe llevar un control del mesero, en el que se incluya su nombre, un codigo distintivo, una contraseña, su horario, y un control de cuantas mesas atendió en el día.

**Escenario de ejemplo.**

1. **Gestión del Menú:**

* El chef principal ingresa los platillos disponibles en el sistema. Cada platillo incluye su nombre y precio.
* Por ejemplo, se ingresan platillos como:
  + Nombre: "Pavo Navideño", Precio: $250.00, Existencia: si.
  + Nombre: "Lomo de Res", Precio: $180.00, Existencia: si.
  + Nombre: "Ensalada César", Precio: $80.00, Existencia: si.
* El barista principal registra en el sistema las bebidas y productos disponibles. Cada ítem incluye su nombre, precio y disponibilidad.
* Por ejemplo, se ingresan productos como:
  + Nombre: "Café Espresso", Precio: $45.00, Disponibilidad: Sí.
  + Nombre: "Capuchino Vainilla", Precio: $65.00, Disponibilidad: Sí.
  + Nombre: "Té Chai Latte", Precio: $60.00, Disponibilidad: Sí.

1. **Atención al Cliente:**

* Un grupo de 4 personas llega al restaurante y es asignado a la Mesa 5. El mesero Juan (código: 33) los atiende.
* El mesero les presta el menú, donde pueden ver todos los platillos disponibles, sus precios y pueden consultar al mesero si hay disponibilidades.
* Uno de los clientes pregunta por el "Pavo Navideño". El mesero selecciona el platillo en el sistema y muestra los detalles: Precio: $250.00, Existencia: 15 porciones, Ingredientes principales: Pavo, relleno, salsa de arándanos.

1. **Realización del Pedido:**

* Los clientes deciden ordenar:
* 1 "Pavo Navideño".
* 2 "Lomo de Res, Sin champiñones, que sea el primer platillo que traigan".
* 1 "Ensalada César".
* El mesero ingresa el pedido en el sistema, que verifica automáticamente la disponibilidad de los platillos y actualiza las existencias.

1. **Control de Mesas:**

* El sistema registra que la Mesa 5 está ocupada por 4 comensales y es atendida por el mesero Juan.
* El mesero Juan tiene un código distintivo (33) y una contraseña para acceder al sistema.

1. **Gestión del Mesero:**

* Después de atender la Mesa 5, el sistema actualiza el registro de mesas atendidas por Juan en el día. Ahora ha atendido 3 mesas en total.

1. **Cierre de la Mesa:**

* Al finalizar la comida, el sistema genera la cuenta para la Mesa 5, que incluye:
* 1 "Pavo Navideño": $250.00.
* 2 "Lomo de Res": $360.00.
* 1 "Ensalada César": $80.00.
* Total: $690.00.
* Los clientes pagan la cuenta, y el sistema marca la Mesa 5 como disponible para nuevos comensales.

**Encontrar los objetos principales.**

Los objetos principales identificados en el problema son:

* Restaurante: El lugar donde se gestiona el menú y se realizan los pedidos.
* Platillos: Cada uno de los elementos del menú.
* Cafés: Bebidas calientes y frias de la barra de cafetería.
* Mesa: La persona que realiza el pedido.
* Mesero: El empleado que toma los pedidos y los comunica a la cocina.

**Definir los datos miembro.**

Los datos miembro (atributos) para cada objeto son:

* **Alimento**:
  + Nombre: El nombre del platillo (por ejemplo, "Pasta Alfredo").
  + Precio: El costo del platillo (por ejemplo, $120.00).
  + Existencia: Si hay disponibilidad para preparar el pedido o no.
  + Ingredientes: Los ingredientes principales del platillo (opcional).

Nos podemos dar cuenta de que tanto los platillos como los cafés comparten similitudes, por lo que creamos primero un objeto Alimento y a partir de el crearemos los objetos platillo y café.

* **Platillo**:
  + - Orden Completa o media orden: Saber la porcion que gusta el comensal, tambien el platillo recibe descuento.
    - Prioridad Alta o Baja: Esto puede representar el orden que los comensales quieren que les traigan sus platos, por ejemplo si quieren que lo primero que llegue es la comida de un niño con este valor se agrega para que el platillo del niño sea preparado primero.
* **Café**:
  + - Cafeina: Importante para la preparacion.
    - Caliente o fria: Algunas bebidas pueden servirse tanto calientes como frias.
    - Leche: El tipo de leche que quiere el cliente o ninguna.
    - Tamaño: El cliente puede pedir sus bebidas de distintos tamaños.
* **Mesero**:
  + Nombre: El nombre del mesero.
  + Codigo: Su codigo distintivo
  + Contraseña.
  + Contador: La cantidad de pedidos que ha tomado.
* **Mesa**:
  + Numero de mesa: numero de mesa.
  + Comenzales: cantidad de personas en la mesa.
  + Mesero: Encargado de antender la mesa.
  + Pedido: Alimentos que ordenaron los comensales.
  + Total: Acumulado del precio de los pedidos.
  + Actividad: Si la mesa ya pidió la cuenta o no.

**Determinar el comportamiento deseado.**

Los comportamientos (métodos) para cada objeto son:

* **Alimento:**
  + Alimento(String,double,String,boolean):Constructor del alimento
  + getNombre():Muestra el nombre.
  + setNombre():Modifica el nombre.
  + getCosto():Muestra el costo.
  + setCosto():Modifica el costo.
  + getComentario():Muestra el comentario.
  + setComentario():Modifica el comentario.
  + setExistencia():Declara como disponible o no al alimento.
  + getExistencia():Muestra la existencia de los alimentos.
  + **Platillo:**
    - isOrdenCompleta():Muestra si la orden es completa o media.
    - setOrdenCompleta():Modifica la orden o media orden.
    - getPrioridadalta():Muestra si el platillo es prioritario o no.
    - setPrioridadalta():Modifica si el platillo es prioritario o no.
  + **Cafe:** 
    - hasCafeina():Muestra si la bebida lleva cafeina o no.
    - setCafeina():Modifica si la bebida lleva cafeina o no.
    - hasHielo():Muestra si la bebida lleva hielo o no.
    - sethielo():Modifica si la bebida lleva hielo o no.
    - printMilkList():Muestra la lista de leches.
    - getMilkListSize():Muestra la cantidad de leches.
    - setMilk():Modifica la leche del cafe.
    - getMIlk():Muestra la leche del cafe.
    - setSize():Modifica el tamaño de la bebida.
    - getSize():Muestra el tamaño de la bedida.
    - printSizeList():Muestra la lista de tamaños.
* **Mesero :**
  + Mesero():Constructor.
  + setNombre() void:agrega un nuevo nombre.
  + getNombre() String:Muestra el nombre del mesero.
  + setCode() void:Agrega un codigo de mesero nuevo.
  + getCode() int:Muestra el codigo de un mesero.
  + setPassword(contraseña)void:Configura su contraseña.
  + LogIn(contraseña) boolean:se utiliza para verificar si la contraseña ingresada en el sistema es correspondiente al mesero.
  + addContador() void:Agrega una mesa a su contador.
  + getContador() int:muestra su contador de mesas.
  + rebootContador() void:elimina el contador para el día siguiente.
* **Mesa:**
  + Mesa():Constructor
  + getNumero():muestra el numero de la mesa.
  + setComenzales():Ingresa el numero de comenzales en la mesa.
  + getComenzales():Muestra el numero de comenzales en la mesa.
  + setMesero():Ingresa el mesero que atiende temporalmente.
  + getMesero():Muestra el mesero que atiende.
  + setPedido():Crea el pedido.
  + getPedido():Muestra detalles del pedido.
  + Infomesa():proporciona informacion de la mesa.

**Asociación**

