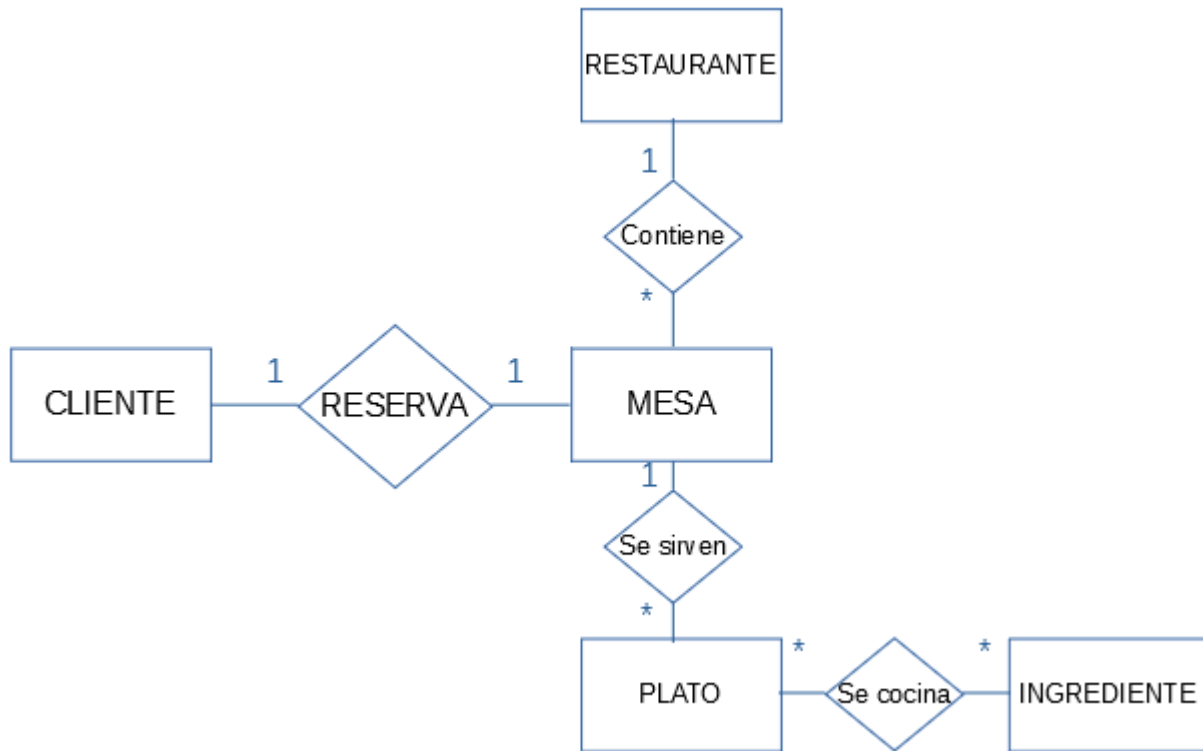


**EJERCICIO 131 – ejer131.java**

Implementa en Java el diagrama de clases de un Restaurante:



Cada una de las clases tendrá que implementar un interfaz. De esta forma, cada clase estará obligada a implementar los métodos cuya cabecera se declara en su interfaz.

**+Clase Cliente**

-Atributos: nombre y teléfono

-Métodos: implementará el interfaz “Interface\_Cliente”

```

//Devuelve el atributo nombre
public interface Interface_Cliente {
    public String getNombre();
}

```

### +Clase Mesa

-Atributos: número de Mesa, número de Personas de la mesa, estado (“LIBRE” o “OCUPADA”), ubicación (“terrazza”, “comedor interno”, etc..), factura y lista de platos que esa Mesa ha pedido.

-Métodos: implementará el interfaz “Interface\_Mesa”

```
public interface Interface_Mesa {  
  
    //En el constructor de la clase se incrementará el numMesa con un contador estático.  
    //Además, se indicará el estado por defecto de una Mesa que es "LIBRE".  
  
    //Devuelve los datos de la Mesa (numMesa, numPersonas, ubicacion y estado)  
    public String obtenerDatosMesa();  
  
    //Devuelve el atributo numMesa  
    public int getnumMesa();  
  
    //Pone el atributo estado de la Mesa a 'LIBRE'  
    public void liberarMesa();  
  
    //Pone el atributo estado de la Mesa a 'OCUPADA'  
    public void ocuparMesa();  
  
    //Añadir al ArrayList de Platos el plato p  
    public void pedirPlato(Plato p);  
  
    //Informar de todos los platos pedidos por la Mesa.  
    //Para ello, recorrer el ArrayList de Platos y llamar a getNombre() en cada uno.  
    public void detalle_platos();  
  
    //Calcular la factura de la Mesa  
    //Para ello, recorrer el ArrayList de Platos y llamar a getPrecio, acumulando el total.  
    public double calcular_factura();  
  
}
```

### +Clase Reserva

-Atributos: fecha, hora, numero de Personas de la reserva, cliente que ha realizado la reserva (por tanto, del tipo Cliente) y mesa que se le ha asignado (por tanto, del tipo Mesa).

-Esta clase NO dispondrá de un interfaz, ya que solo tendrá un constructor. Este constructor, además de informar los atributos, pondrá la Mesa asignada como “OCUPADA” llamando al método ocuparMesa() de la clase Mesa.

### +Clase Restaurante

-Atributos: nombre del Restaurante, teléfono y lista de Mesas.

-Métodos: implementará el interfaz “Interface\_Restaurante”

```
public interface Interface_Restaurante {  
    //Añade la mesa m al ArrayList de mesas.  
    public void montarMesa(Mesa m);  
  
    //Buscar en el ArrayList de mesas aquella con el NumMesa de m y borrarla.  
    public void recogerMesa(Mesa m);  
  
    //Por cada mesa del ArrayList de mesas, llamar a obtenerDatosMesa().  
    //Sy el ArrayList está vacío indicar 'Todas las mesas estan recogidas'.  
    public void listarMesas();  
  
    //Recorrer el ArrayList de mesas acumulando las facturas de todas ellas  
    //Para ello, por cada mesa llamaremos a calcular_factura().  
    public double calcular_caja();  
}
```

### +Clase Plato

-Atributos: nombre del Plato, precio y lista de ingredientes del plato.

-Métodos: implementará el interfaz “Interface\_Plato”

```
public interface Interface_Plato {  
  
    //En el constructor de la clase se indicará el precio del plato de la siguiente forma:  
    //Si la primera letra del plato es una vocal (mayuscula o minuscula) -> el plato vale 10 €  
    //Si es cualquier otra letra -> el plato vale 5 €  
  
    //Añadir ingrediente i al ArrayList de ingredientes del plato  
    public void anyadir_ingrediente(Ingrediente i);  
  
    //Mostrar todos los ingrediente del Plato  
    //Para ello, recorrer el ArrayList de ingredientes y llamar a detalleIngrediente();  
    public String consultar_ingredientes();  
  
    //Devuelve el atributo precio  
    public double getPrecio();  
  
    //Devuelve el atributo nombre_plato  
    public String getNombre();  
}
```

**+Clase Ingrediente**

-Atributos: nombre del ingrediente y cantidad del mismo.

-Métodos: implementará el interfaz “Interface\_Ingrediente”

```
public interface Interface_Ingrediente {

    //Devuelve la informacion del nombre y la cantidad del ingrediente.
    public String detalleIngrediente();

}
```

**+Clase Principal**

En esta clase ejecutaremos en el main() la siguiente secuencia de acciones, que describirá como un par de clientes hacen una reserva, piden unos platos, comen, pagan y se van:

```
//Creamos 2 clientes
Cliente c1 = new Cliente("Paco Perez","666301220");
Cliente c2 = new Cliente("Gareth Bale","654122322");

//Creamos 2 mesas
Mesa m1 = new Mesa(2,"Terraza");
Mesa m2 = new Mesa(4,"Comedor interno");

//Creo un restaurante y monto las 2 mesas
Restaurante res1 = new Restaurante("Bazofia para todos","654312450");
res1.montarMesa(m1);
res1.montarMesa(m2);

//Listamos las 2 mesas y deberian estar libres
System.out.println("-Listado de las mesas:");
res1.listarMesas();

//Hago las reservas de las 2 mesas
Reserva r1 = new Reserva("01/06/2020",20,2,m1,c1);
Reserva r2 = new Reserva("01/06/2020",22,4,m2,c2);

//Obtenemos los datos de la mesa de la terraza, ahora está ocupada
System.out.println("-Estado de la mesa de la terraza:");
System.out.println(m1.obtenerDatosMesa());

//Obtenemos los datos de la mesa del comedor interno, ahora está ocupada
System.out.println("-Estado de la mesa del comedor interno:");
System.out.println(m2.obtenerDatosMesa());

//Damos de alta 3 ingredientes
Ingrediente in1 = new Ingrediente("tomate",100);
Ingrediente in2 = new Ingrediente("spaghetti",200);
Ingrediente in3 = new Ingrediente("queso",30);

//Creamos un plato con esos 3 ingredientes
Plato p1 = new Plato("Spaghetti al plato");
```

```
p1.anyadir_ingredient(in1);
p1.anyadir_ingredient(in2);
p1.anyadir_ingredient(in3);
System.out.println("-Ingredientes de los spaghetti:");
System.out.println(p1.consultar_ingredientes());

//Creamos otro plato con 2 ingredientes de antes y uno nuevo
Plato p2 = new Plato("Pizza barbacoa");
Ingrediente in4 = new Ingrediente("carne picada",300);
p2.anyadir_ingredient(in1);
p2.anyadir_ingredient(in3);
p2.anyadir_ingredient(in4);
System.out.println("-Ingredientes de la pizza:");
System.out.println(p2.consultar_ingredientes());

//Creamos un ultimo plato con un nuevo ingrediente
Plato p3 = new Plato("Entrecot con salsa de queso");
Ingrediente in5 = new Ingrediente("entrecot",500);
p3.anyadir_ingredient(in5);
p3.anyadir_ingredient(in3);
System.out.println("-Ingredientes del entrecot:");
System.out.println(p3.consultar_ingredientes());

//Desde la mesa m1 se piden los 3 platos creados anteriormente
m1.pedirPlato(p1);
m1.pedirPlato(p2);
m1.pedirPlato(p3);

//Resumimos los 3 platos que se han pedido en esa mesa
System.out.println("-Platos pedidos de la primera mesa");
m1.detalle_platos();

//Piden la factura de la mesa y se van
System.out.println("-La mesa ha gastado: " + m1.calcular_factura() + "€ y se van.");
m1.liberarMesa();

//La segunda mesa pide 4 entrecots
m2.pedirPlato(p3);
m2.pedirPlato(p3);
m2.pedirPlato(p3);
m2.pedirPlato(p3);
System.out.println("-Platos pedidos de la segunda mesa");
m2.detalle_platos();

//Piden la factura de la mesa y se van
System.out.println("-La mesa ha gastado: " + m2.calcular_factura() + "€ y se van.");
m2.liberarMesa();

//Comprobamos el total de la caja
System.out.println("El restaurante ha facturado: " + res1.calcular_caja() + "€.");

//Al final del dia desmontamos las mesas
//Empezamos recogiendo la mesa m1
res1.recogerMesa(m1);
```

```
//Si listamos las mesas solo nos aparece la mesa m2, ya que la m1 la hemos recogido  
res1.listarMesas();  
  
//Recogemos la mesa m2  
res1.recogerMesa(m2);  
  
//Comprobamos que todas las mesas estan recogidas.  
//Esta llamada devuelve el mensaje "Todas las mesas estan recogidas"  
res1.listarMesas();
```