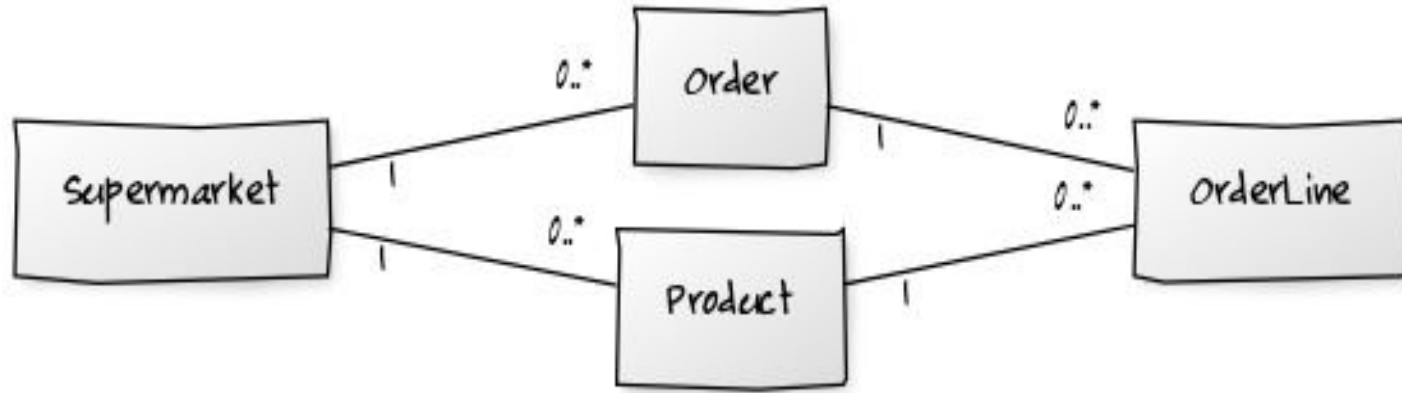


# **Práctica**

**Supermercado con listas, pilas y colas**

David Ayuso, Vladimir Rico

# El modelo

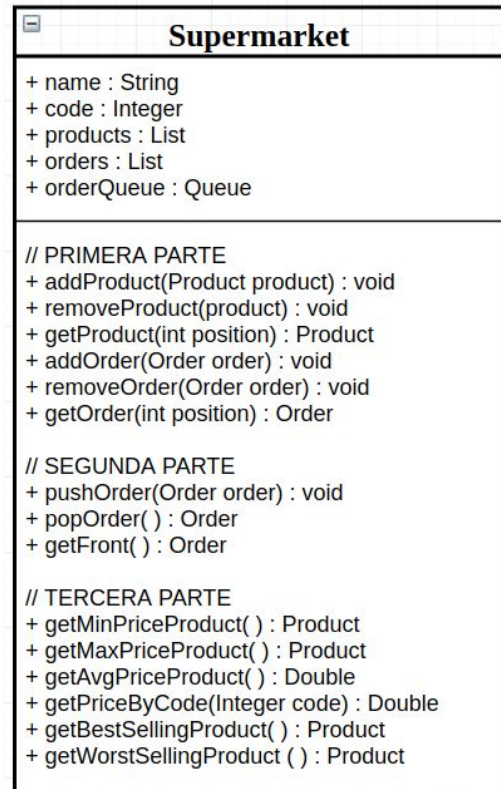


## Entidades

- Supermercado
- Pedido
- Producto
- Línea de pedido

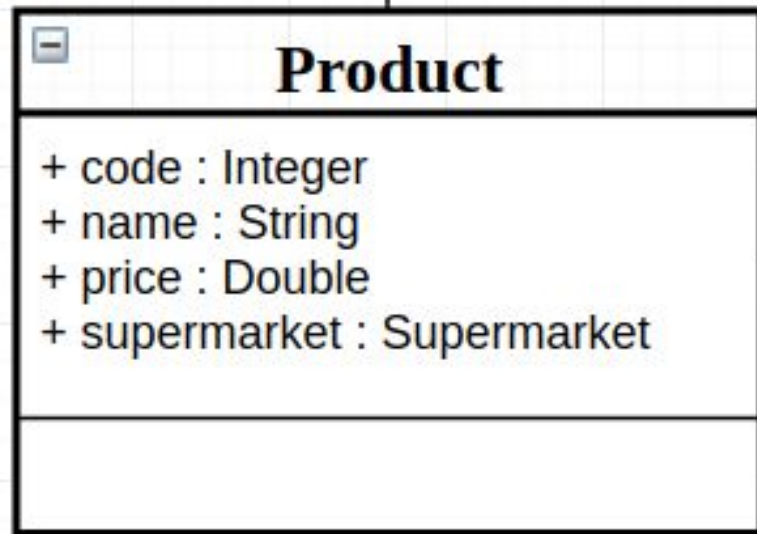
# Supermercado

- Se identifica mediante un código y un nombre
- Tiene una lista de productos y de pedidos relacionadas.
- Mantiene una cola de pedidos ordenados por orden de llegada
- Además posee métodos para calcular:
  - Producto de precio mínimo y máximo
  - Precio medio de los productos
  - Consultar el precio por código de producto
  - Producto más y menos vendido



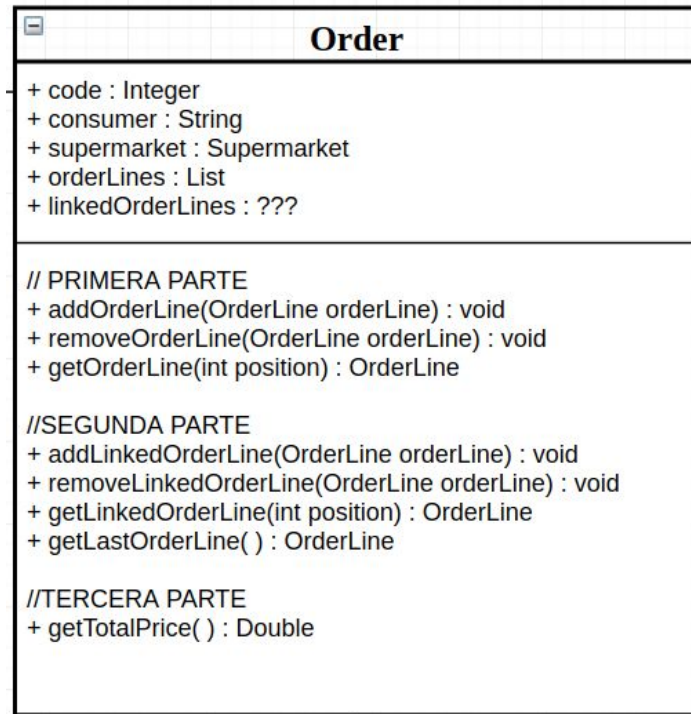
# Producto

- Se identifica mediante un código y un nombre
- Tiene asignado un precio
- Además está relacionado con un supermercado



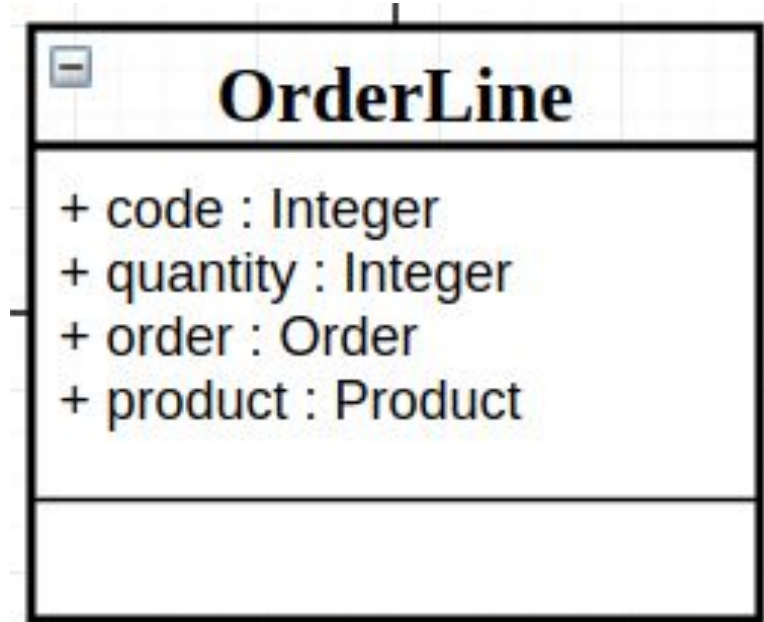
# Pedido

- Se identifica mediante un código
- Tiene una lista de *Líneas de pedido*, un cliente y un supermercado relacionados
- Además posee una lista enlazada de *Líneas de pedido*
- También contiene métodos para calcular:
  - Calcular precio total del pedido



# Línea de pedido

- Se identifica mediante un código y tiene una cantidad asignada
- Además está relacionado con un pedido y un producto



# Parte 1 - Operaciones con listas

- Métodos básicos de una lista
  - Obtener
  - Añadir
  - Eliminar
- Se localizan en las clases que tienen relacionadas listas de otros productos
  - Supermercado
  - Pedido

```
//Métodos a completar
//PRIMERA PARTE
public void addProduct(Product product) {
    //TODO Añadir producto
}

public void removeProduct(Product product) {
    //TODO Eliminar producto
}

public Product getProduct(int position) {
    //TODO Obtener producto

    return null;
}
```

src/main/java/model/**Supermarket.java**

## Parte 2 - Listas enlazadas

- Implementar una lista de *Líneas de pedido* como lista enlazada
- Implementar los métodos básicos como son:
  - Obtener
  - Añadir
  - Eliminar

```
//SEGUNDA PARTE
private ListaArticulos linkedOrderLines;

//SEGUNDA PARTE
public void addLinkedOrderLine(OrderLine orderLine) {
    //TODO Añadir artículo
}

public void removeLinkedOrderLine(OrderLine orderLine) {
    //TODO Eliminar artículo
}

public OrderLine getLinkedOrderLine(int position) {
    //TODO Coger artículo
    return null;
}
```

src/main/java/model/**Order.java**



## Parte 2 - Listas enlazadas

- Mediante las clases:
  - ListaArticulos (Lista)
  - Artículo (Nodo)
- Implementar los métodos básicos como son:
  - Obtener
  - Añadir
  - Insertar
  - Eliminar

```
public class ListaArticulos {  
    public void add(Articulo articulo) {  
        //TODO Añadir artículo  
    }  
    public void insert(Articulo articulo, int position) {  
        //TODO Insertar el artículo en la posición indicada  
    }  
    public void remove(Articulo articulo) {  
        //TODO Eliminar artículo de la lista  
    }  
    public void get(int position) {  
        //TODO Obtener artículo de la lista  
    }  
}
```

src/main/java/util/ListaArticulos.java

# Parte 3 - Colas y métodos complementarios

- Implementar una cola de pedidos en la clase Supermercado
- Mediante la interfaz *Queue*

```
//TERCERA PARTE  
private Queue<Order> orderQueue;
```

```
//TERCERA PARTE  
//Cola de pedidos  
public void pushOrder(Order order) {  
    //TODO Añadir pedido  
}  
  
public Order popOrder() {  
    //TODO Obtener pedido  
  
    return null;  
}  
  
public Order getFront() {  
    //TODO Devuelve el primer pedido de la cola  
    //OJO!! No hay que sacarlo, solo devolver el primer pedido  
  
    return null;  
}
```

src/main/java/model/Supermarket.java

# Parte 3 - Colas y métodos complementarios

- Parte opcional
- Implementar métodos para realizar operaciones o consultas sobre las listas que hemos realizado

```
//Operaciones
public void getMinPriceProduct() {
    //TODO Devolver el producto con el precio más bajo
}

public void getMaxPriceProduct() {
    //TODO Devolver el producto con el precio más alto
}

public void getAvgPriceProduct() {
    //TODO Devolver la media de precios de los productos
}

public void getPriceByCode(Integer productCode) {
    //TODO Devolver el precio del producto
}

public void getBestSellingProduct() {
    //TODO Devolver el producto más vendido junto con el número de ventas
}

public void getWorstSellingProduct() {
    //TODO Devolver el producto más vendido junto con el número de ventas
}
```

src/main/java/model/Supermarket.java

# Evaluación

- Parte 1 - Operaciones con listas
- Parte 2 - Listas enlazadas
- Parte 3 - Colas y métodos complementarios

# Criterios de evaluación

- Funcionalidad del método (70%)
- Claridad, comentarios y estructura (30%)

$$NP = P1 * 0.4 + P2 * 0.3 + P3 * 0.3$$