Mostrar lista de pedidos

En este caso, se mostrara al usuario un listado de los pedidos que ah echo, donde se mostrara, la cantidad que pago/pagara, la fecha que se efectuó el pedido, productos pedidos, entre otra información.

|  |
| --- |
| **Producto** |
| * ID\_Product : int * Nombre : String * Precio : float * Cantidad : int |
| * getID\_Product() : int * getNombre() : String * getPrecio() : float * getCantidad() : int |

|  |
| --- |
| **Pedido** |
| * ID\_Pedido : int * Fecha\_pedido : String * Lista\_Productos[] : Producto * Total : float * Pagado : boolean * Entregado : boolean |
| * getID\_Pedido() : int * getFecha\_pedido() : String * getLista\_Productos() : Producto[] * getTotal() : float * getPagado() : boolean * getEntregado() : boolean * anadir\_producto() * Calcular\_Total() * setPagado() * setEntregado() * nuevo\_Total() * toString() |

|  |
| --- |
| **Usuario** |
| * ID\_Usuario : int * Nombre : String |
| * getID\_Usuario() : int * getNombre() : String |

|  |
| --- |
| **Pedidos\_Total** |
| * Lista\_Pedidos[] : Pedido * Cliente : Usuario |
| * getLista\_Pedidos() : Pedido[] * getCliente() : Usuario * Impr\_Todo() |

**Clases**

**Usuario**

Esta clase contendrá al usuario en cuestión, para así poder efectuar con mas facilidad todo.

**Producto**

En un principio quería crearle tres clases las cuales hereden esta clase (bebidas,palomitas y dulces), pero dado que en producto existe el Id, esto hace inesesario crear clases de herencia.

Dentro de esta clase se mostrara la información principal de cada producto, así como el monto solicitado.

**Pedido**

En este se mostrara el conjunto de productos solicitados por el usuario, así como el monto apagar, y si fue o no pagado.

Cree un procedimiento llamado, nuevo\_total, con la finalidad, de que, si en dado caso que el usuario no pague a tiempo, tomando como referencia la fecha del pedido, el coste de este se incrementara hasta que sea pagado.

**Pedidos\_total**

Este se encarga de mostrar en pantalla los pedidos con toda la información necesario.

**Relaciones de clase**

**Usuario-Pedidos\_total**

Tome en cuenta que para poder mostrar los pedidos es vital que existe un usuario, por ende, relacione pedidos\_total con usuario, para que así, si no hay usuario, no hay pedidos.

**Producto-Pedido**

Para este, obtuve la relación haciéndome la siguiente pregunta, si no hay productos, no existe un pedido, por esto mismo, relacione esas clases de esa forma, así si no existen productos, no existe el pedido.

**Pedido-Pedidos\_total**

Para este, tome como referencia, un listado de carrito de mercado libre, donde, en dado caso de existir un producto, se muestra, si no existe, simplemente no se muestra nada, con esta lógica puse una relación, donde no es vital que exista pedidos.