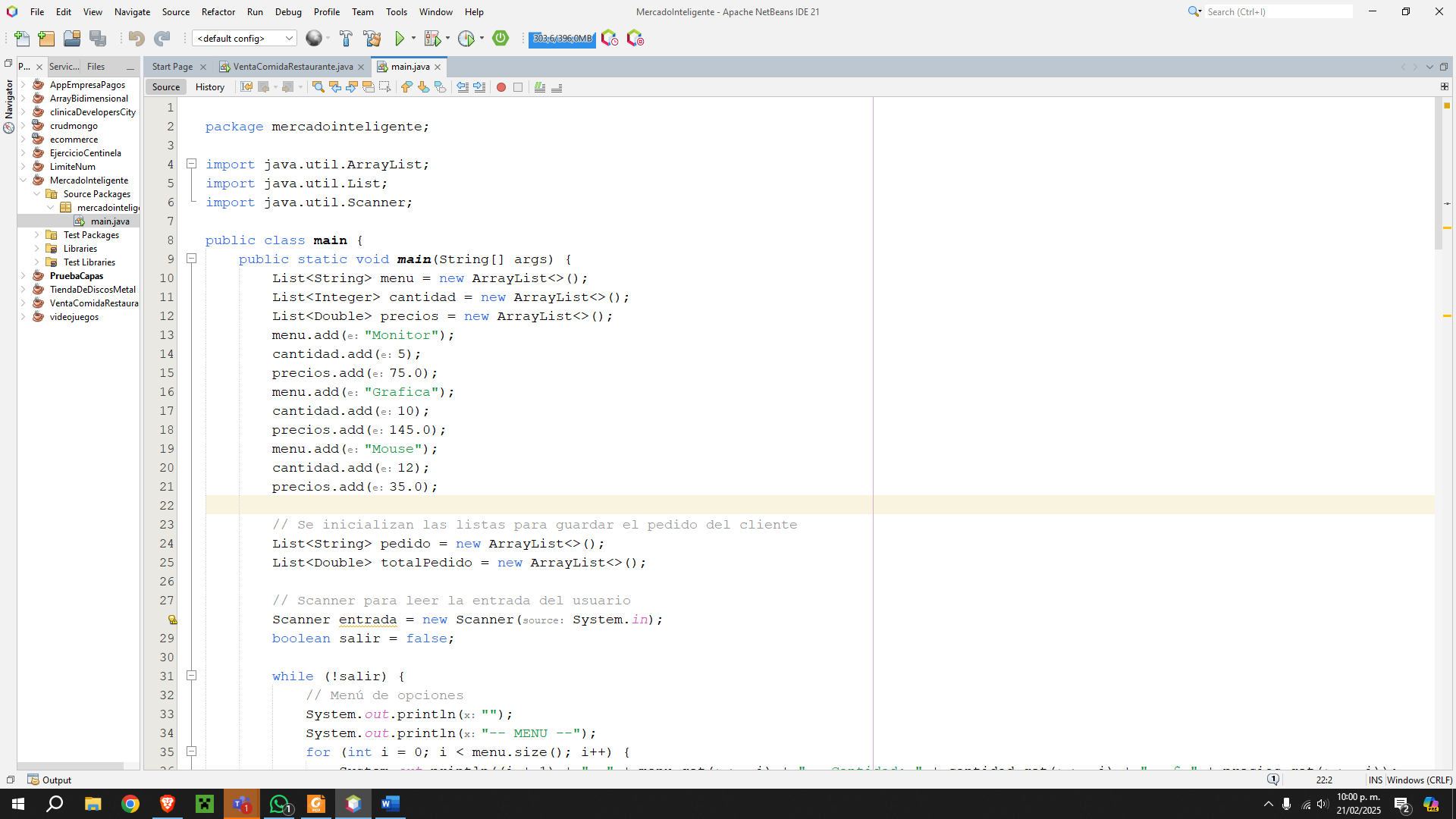
MERCADO INTELIGENTE



**1. Definición de listas para el inventario**

javaCopiarEditarList<String> menu = new ArrayList<>();  
List<Integer> cantidad = new ArrayList<>();  
List<Double> precios = new ArrayList<>();

* Se crean **tres listas**:
  + menu: almacena los nombres de los productos.
  + cantidad: guarda la cantidad disponible de cada producto.
  + precios: almacena el precio de cada producto.

javaCopiarEditarmenu.add("Monitor");  
cantidad.add(5);  
precios.add(75.0);

* Se agregan **tres productos** con su cantidad y precio.

**2. Definición de listas para el carrito de compras**

javaCopiarEditarList<String> pedido = new ArrayList<>();  
List<Double> totalPedido = new ArrayList<>();

* pedido: almacena los productos seleccionados por el usuario.
* totalPedido: guarda el precio de cada producto agregado.

**3. Uso de Scanner y la variable de control salir**

javaCopiarEditarScanner entrada = new Scanner(System.in);boolean salir = false;

* Se usa Scanner para leer la entrada del usuario.
* salir controla si el usuario quiere salir del programa.

**4. Bucle principal (Menú de opciones)**

javaCopiarEditarwhile (!salir) {  
    // Muestra el menú    System.out.println("-- MENU --");  
    for (int i = 0; i < menu.size(); i++) {  
        System.out.println((i + 1) + ". " + menu.get(i) + " - Cantidad: " + cantidad.get(i) + " - $ " + precios.get(i));  
 }

* Se muestra el menú con los productos disponibles y sus cantidades.

javaCopiarEditarSystem.out.println("1. AGREGAR PRODUCTO AL CARRITO");  
System.out.println("2. VER CARRITO Y TOTAL A PAGAR");  
System.out.println("3. ELIMINAR PRODUCTO DEL CARRITO");  
System.out.println("4. FINALIZAR COMPRA");  
System.out.println("5. SALIR");

* Se presentan las opciones disponibles.

**5. Leer opción del usuario**

javaCopiarEditarint opcion = entrada.nextInt();

* Se espera que el usuario ingrese una opción.

**Opciones del menú**

**Opción 1: Agregar producto al carrito**

javaCopiarEditarint seleccion = entrada.nextInt();if (seleccion > 0 && seleccion <= menu.size()) {  
    int index = seleccion - 1;  
    if (cantidad.get(index) > 0) {  
        pedido.add(menu.get(index));        totalPedido.add(precios.get(index));        cantidad.set(index, cantidad.get(index) - 1);  
        System.out.println(menu.get(index) + " \*\*\* Se ha añadido al pedido");  
 } else {  
 System.out.println("Lo siento, " + menu.get(index) + " está agotado.");  
 }  
}

* El usuario selecciona un producto.
* Se verifica si hay stock disponible.
* Si hay stock:
  + Se agrega al carrito (pedido y totalPedido).
  + Se reduce la cantidad en cantidad.

**Opción 2: Ver carrito y total a pagar**

javaCopiarEditarif (pedido.isEmpty()) {  
    System.out.println("El carrito está vacío.");  
} else {  
    double total = 0;  
    for (int i = 0; i < pedido.size(); i++) {  
        System.out.println((i + 1) + ". " + pedido.get(i) + " - $ " + totalPedido.get(i));  
 total += totalPedido.get(i);  
 }  
 System.out.println("Total a pagar: $" + total);  
}

* Se muestra el contenido del carrito y el total a pagar.

**Opción 3: Eliminar producto del carrito**

javaCopiarEditarSystem.out.println("Seleccione el número del producto que desea eliminar del carrito:");for (int i = 0; i < pedido.size(); i++) {  
    System.out.println((i + 1) + ". " + pedido.get(i) + " - $ " + totalPedido.get(i));  
}int seleccionEliminar = entrada.nextInt();if (seleccionEliminar > 0 && seleccionEliminar <= pedido.size()) {  
    System.out.println(pedido.get(seleccionEliminar - 1) + " \*\*\* Se ha eliminado del carrito");  
    pedido.remove(seleccionEliminar - 1);  
    totalPedido.remove(seleccionEliminar - 1);  
} else {  
 System.out.println("Opción inválida");  
}

* Se muestra el carrito y el usuario elige qué producto eliminar.
* Si la opción es válida, se elimina de pedido y totalPedido.

**Opción 4: Finalizar compra**

javaCopiarEditarif (pedido.isEmpty()) {  
    System.out.println("No tienes productos en el carrito.");  
} else {  
    System.out.println("Ticket de compra 1279");  
    double totalFinal = 0;  
    for (int i = 0; i < pedido.size(); i++) {  
        System.out.println((i + 1) + ". " + pedido.get(i) + " - $ " + totalPedido.get(i));  
 totalFinal += totalPedido.get(i);  
 }  
 System.out.println("Total a pagar: $" + totalFinal);  
 System.out.println("--- COMPRA FINALIZADA ---");  
 System.out.println("¡Gracias por tu compra!");  
}

* Se muestra el ticket de compra con el total a pagar.

**Opción 5: Salir del programa**

javaCopiarEditarsalir = true;

* Se cambia salir a true y se termina el bucle.

**6. Cerrar Scanner**

javaCopiarEditarentrada.close();

* Se cierra el objeto Scanner para evitar fugas de memoria.