Explicación Detallada del Error de Productos Duplicados

¿Qué está pasando en el código original?

Vamos a analizar paso a paso lo que ocurre cuando el usuario ejecuta el programa:

java
```
public static void anadirPedido() {
    System.out.println("Mostrar Menu");
    datos.arrayProductos.add("Cafe");     // ← AQUÍ ESTÁ EL PROBLEMA
    datos.arrayProductos.add("Donuts");
    datos.arrayProductos.add("Té");
    // ... resto de productos
}
```

Simulación de Ejecución

Imagina que el usuario usa el programa así:

Primera vez que selecciona "1-Añadir pedido":
```
arrayProductos = ["Cafe", "Donuts", "Té", "Tostadas", "Churros",
                  "Te verde", "Mermelada", "Leche", "Cereales"]

Total de productos: 9 ✓
```

Segunda vez que selecciona "1-Añadir pedido":
```
arrayProductos = ["Cafe", "Donuts", "Té", "Tostadas", "Churros",
                  "Te verde", "Mermelada", "Leche", "Cereales",
                  "Cafe", "Donuts", "Té", "Tostadas", "Churros",  ← DUPLICADOS
                  "Te verde", "Mermelada", "Leche", "Cereales"]

Total de productos: 18 ✗
```

Tercera vez:

```
Total de productos: 27 X X X
```

¿Por qué sucede esto?

El método add() de ArrayList **SIEMPRE AÑADE** elementos, nunca reemplaza:

```
ArrayList<String> lista = new ArrayList<>();
lista.add("A"); // lista = ["A"]
lista.add("A"); // lista = ["A", "A"] ← NO elimina el anterior
lista.add("A"); // lista = ["A", "A", "A"]
```

Consecuencias Visibles para el Usuario

Cuando el usuario ve el menú después de seleccionar varias veces la opción 1:

```
Productos disponibles:
1 - Cafe
2 - Donuts
3 - Té
4 - Tostadas
5 - Churros
6 - Te verde
7 - Mermelada
8 - Leche
9 - Cereales
10 - Cafe      ← DUPLICADO
11 - Donuts    ← DUPLICADO
12 - Té        ← DUPLICADO
13 - Tostadas  ← DUPLICADO
...
```

¿Por qué es un problema grave?

1. **Confusión del usuario**: Ve productos repetidos
2. **Comportamiento incorrecto**: Si selecciona "10" piensa que elige otra cosa, pero es Café otra vez
3. **Consumo de memoria**: Cada vez crece la lista innecesariamente
4. **Bug lógico**: El programa no se comporta como debería

Solución Correcta

Opción 1: Inicializar en el Constructor (MEJOR PRÁCTICA)

java

```
public class CafeteriaDigital {
    private ArrayList<String> productosDisponibles;

    public CafeteriaDigital() {
        this.productosDisponibles = new ArrayList<>();
        inicializarProductos(); // ← Se ejecuta UNA SOLA VEZ
    }

    private void inicializarProductos() {
        productosDisponibles.add("Cafe");
        productosDisponibles.add("Donuts");
        productosDisponibles.add("Té");
        // ... resto
    }
}
```

¿Por qué funciona?

- El constructor se ejecuta **una sola vez** cuando creas el objeto
- Los productos se añaden **una sola vez**
- Cada llamada a anadirPedido() usa la lista ya inicializada

Opción 2: Verificar si la lista está vacía (PARCHE)

```
public static void anadirPedido() {
    // Solo inicializar si está vacía
    if (datos.arrayProductos.isEmpty()) {
        datos.arrayProductos.add("Cafe");
        datos.arrayProductos.add("Donuts");
        // ...
    }

    // Resto del código...
}
```

Problema de esta solución:

- Es un parche, no soluciona el diseño incorrecto
- Mezcla la lógica de inicialización con la de añadir pedidos

Opción 3: Inicializar en el main (ACEPTABLE para código simple)

```
public static void main(String[] args) {
    // Inicializar productos UNA SOLA VEZ al inicio
    datos.arrayProductos.add("Cafe");
    datos.arrayProductos.add("Donuts");
    datos.arrayProductos.add("Té");
    // ...

    int opcion;
    do {
        // ... menú
    } while (opcion != 0);
}

public static void anadirPedido() {
    // Ya NO añade productos aquí
    int opcion;
    do {
        for (int i = 0; i < datos.arrayProductos.size(); i++) {
            System.out.println((i + 1) + "-" + datos.arrayProductos.get(i));
        }
        // ... resto del código
    } while (opcion != 0);
}
```

Comparación Visual

CÓDIGO INCORRECTO:

```
Usuario selecciona opción 1 → añade 9 productos → lista tiene 9
Usuario selecciona opción 1 → añade 9 productos → lista tiene 18 ✗
```

Usuario selecciona opción 1 → añade 9 productos → lista tiene 27 ✗✗

```

### **CÓDIGO CORRECTO:**

```

Programa inicia → añade 9 productos → lista tiene 9

Usuario selecciona opción 1 → NO añade nada → lista tiene 9 ✓

Usuario selecciona opción 1 → NO añade nada → lista tiene 9 ✓

Regla General

Los datos de configuración (catálogos, menús, listas de productos) deben inicializarse UNA SOLA VEZ al inicio del programa, NO cada vez que se usan.

Este es uno de los errores más comunes en programadores principiantes: confundir la **inicialización de datos** con la **lógica de uso**.