

Práctica Clean Code

Mi programa es un contador de frutas y verduras de la nevera, escribimos en consola la cantidad de cada una y luego restamos las que vamos a usar en dos recetas, una receta solo con verduras y otra solo con frutas.

- Bloque 1: Nombres

```
static int n = 0;
static int p = 0;
static int m = 0;
static int b = 0;
static int z = 0;
static int c = 0;

public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    System.out.println("Ingrese la cantidad de naranjas que hay en la nevera:");
    int n2 = scanner.nextInt();
    n += n2;

    System.out.println("Ingrese la cantidad de plátanos que hay en la nevera:");
    int p2 = scanner.nextInt();
    p += p2;

    System.out.println("Ingrese la cantidad de manzanas que hay en la nevera:");
    int m2 = scanner.nextInt();
    m += m2;

    System.out.println("Ingrese la cantidad de berenjenas que hay en la nevera:");
    int b2 = scanner.nextInt();
    b += b2;

    System.out.println("Ingrese la cantidad de zanahorias que hay en la nevera:");
    int z2 = scanner.nextInt();
    z += z2;

    System.out.println("Ingrese la cantidad de calabacines que hay en la nevera:");
    int c2 = scanner.nextInt();
    c += c2;
}
```

En el código podemos ver que ninguna variable tiene un nombre descriptivo que nos ayude a entender a qué hace referencia, en este caso el println es lo único que nos da una pista de que se trata cada nombre. También, a la hora de buscar estas variables será mucho más complicado en un código más o menos grande y dificultará cualquier conversación que las incluya.

Para mejorar este código cambiamos el nombre de las variables por los nombres de cada fruta y verdura y a las que se encuentran en el scanner podemos mantener el nombre inicial y añadir algún adjetivo o sustantivo que defina ese nuevo estado.

Un ejemplo de buen nombramiento sería:

```
static int naranjas = 0;
static int platanos = 0;
static int manzanas = 0;
static int berenjenas = 0;
static int zanahorias = 0;
static int calabacines = 0;

public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    System.out.println("Ingrese la cantidad de naranjas que hay en la nevera:");
    int naranjasIniciales = scanner.nextInt();
    naranjas += naranjasIniciales;

    System.out.println("Ingrese la cantidad de plátanos que hay en la nevera:");
    int platanosIniciales = scanner.nextInt();
    platanos += platanosIniciales;

    System.out.println("Ingrese la cantidad de manzanas que hay en la nevera:");
    int manzanasIniciales = scanner.nextInt();
    manzanas += manzanasIniciales;

    System.out.println("Ingrese la cantidad de berenjenas que hay en la nevera:");
    int berenjenasIniciales = scanner.nextInt();
    berenjenas += berenjenasIniciales;

    System.out.println("Ingrese la cantidad de zanahorias que hay en la nevera:");
    int zanahoriasIniciales = scanner.nextInt();
    zanahorias += zanahoriasIniciales;

    System.out.println("Ingrese la cantidad de calabacines que hay en la nevera:");
    int calabacinesIniciales = scanner.nextInt();
    calabacines += calabacinesIniciales;
}
```

Las funciones también deben contar con un nombre que describa qué acción realiza, cuando hay varias funciones que hacen acciones similares, es importante definir las de manera que se diferencien entre ellas. En este caso tenemos una función que muestra la cantidad de cada fruta y verdura dentro de la nevera y la otra suma la nueva comprada a esa cantidad.

```
static void suma() {  
    System.out.println("Inventario en la nevera:");  
    System.out.println("Naranjas: " + n);  
    System.out.println("Plátanos: " + p);  
    System.out.println("Manzanas: " + m);  
    System.out.println("Berenjenas: " + b);  
    System.out.println("Zanahorias: " + z);  
    System.out.println("Calabacines: " + c);  
}
```

```
static void añadidas(Scanner scanner) {  
    System.out.println("Ingrese la cantidad de naranjas que ha comprado:");  
    int naranjasCompradas = scanner.nextInt();  
    naranjas += naranjasCompradas;  
  
    System.out.println("Ingrese la cantidad de plátanos que ha comprado:");  
    int platanosComprados = scanner.nextInt();  
    platanos += platanosComprados;  
  
    System.out.println("Ingrese la cantidad de manzanas que ha comprado:");  
    int manzanasCompradas = scanner.nextInt();  
    manzanas += manzanasCompradas;  
  
    System.out.println("Ingrese la cantidad de berenjenas que ha comprado:");  
    int berenjenasCompradas = scanner.nextInt();  
    berenjenas += berenjenasCompradas;  
  
    System.out.println("Ingrese la cantidad de zanahorias que ha comprado:");  
    int zanahoriasCompradas = scanner.nextInt();  
    zanahorias += zanahoriasCompradas;  
  
    System.out.println("Ingrese la cantidad de calabacines que ha comprado:");  
    int calabacinesComprados = scanner.nextInt();  
    calabacines += calabacinesComprados;  
  
    System.out.println(";Compra realizada con éxito! Frutas y verduras añadidas al inventario.");  
}
```

Una buena forma de definir una función es usando verbos que expliquen su funcionalidad.

En este caso sería:

```
static void mostrarInventario() {  
    System.out.println("Inventario en la nevera:");  
    System.out.println("Naranjas: " + naranjas);  
    System.out.println("Plátanos: " + platanos);  
  
static void comprar(Scanner scanner) {  
    System.out.println("Ingrese la cantidad de naranjas que ha comprado:");  
    int naranjasCompradas = scanner.nextInt();  
    naranjas += naranjasCompradas;  
  
    System.out.println("Ingrese la cantidad de plátanos que ha comprado:");
```

- Bloque 2 : Funciones

Para que haya un buen clean code es importante tener funciones cortas que sean fáciles de modificar y testear. También tendremos que tener en cuenta la intención de una sola acción por función. En este código tenemos una función con muchas líneas.

```
static void comprar(Scanner scanner) {  
    System.out.println("Ingrese la cantidad de naranjas que ha comprado:");  
    int naranjasCompradas = scanner.nextInt();  
    naranjas += naranjasCompradas;  
  
    System.out.println("Ingrese la cantidad de plátanos que ha comprado:");  
    int platanosComprados = scanner.nextInt();  
    platanos += platanosComprados;  
  
    System.out.println("Ingrese la cantidad de manzanas que ha comprado:");  
    int manzanasCompradas = scanner.nextInt();  
    manzanas += manzanasCompradas;  
  
    System.out.println("Ingrese la cantidad de berenjenas que ha comprado:");  
    int berenjenasCompradas = scanner.nextInt();  
    berenjenas += berenjenasCompradas;  
  
    System.out.println("Ingrese la cantidad de zanahorias que ha comprado:");  
    int zanahoriasCompradas = scanner.nextInt();  
    zanahorias += zanahoriasCompradas;  
  
    System.out.println("Ingrese la cantidad de calabacines que ha comprado:");  
    int calabacinesComprados = scanner.nextInt();  
    calabacines += calabacinesComprados;  
  
    System.out.println(";Compra realizada con éxito! Frutas y verduras añadidas al inventario.");  
}
```

Dividimos en dos las funciones, por una parte el scanner y por otro la suma de los productos nuevos. Estas funciones permiten tener pocos argumentos y poder utilizarlas de forma independiente.

```
static void comprar(Scanner scanner) {  
    agregarProducto(scanner, "naranjas");  
    agregarProducto(scanner, "platanos");  
    agregarProducto(scanner, "manzanas");  
    agregarProducto(scanner, "berenjenas");  
    agregarProducto(scanner, "zanahorias");  
    agregarProducto(scanner, "calabacines");  
    System.out.println(";Compra realizada con éxito! Frutas y verduras añadidas al inventario.");  
  
static void agregarProducto(Scanner scanner, String producto) {  
    System.out.println("Ingrese la cantidad de " + producto + " que ha comprado:");  
    int cantidadComprada = scanner.nextInt();  
    switch (producto) {  
        case "naranjas" -> naranjas += cantidadComprada;  
        case "platanos" -> platanos += cantidadComprada;  
        case "manzanas" -> manzanas += cantidadComprada;  
        case "berenjenas" -> berenjenas += cantidadComprada;  
        case "zanahorias" -> zanahorias += cantidadComprada;  
        case "calabacines" -> calabacines += cantidadComprada;  
    }  
}
```

- Bloque 3: Comentarios

Es habitual cuando se comienza a programar utilizar los comentarios como forma de guía, sin embargo, para tener un código limpio debemos evitar esto, ya que un código tiene que explicarse por sí mismo. En este código tenemos varios comentarios explicativos, todos innecesarios ya que los nombres de las variables y funciones ya lo hacen. En el código final, estos comentarios son eliminados.

```
// Mostrar resultados
System.out.println("Cantidad de frutas que quedan en la nevera: " + frutasDespues);
System.out.println("Cantidad de verduras que quedan en la nevera: " + verdurasDespues);

// Verificar si hay 2 o menos frutas y verduras y enviar un mensaje
if (frutasDespues <= 2) {
    System.out.println("¡Atención! Quedan 2 o menos frutas en la nevera. ¡Compra más!");
}
if (verdurasDespues <= 2) {
    System.out.println("¡Atención! Quedan 2 o menos verduras en la nevera. ¡Compra más!");
}

// Comprar más frutas y verduras
comprar(scanner);

// Mostrar el inventario actualizado
mostrarInventario();
}

static void comprar(Scanner scanner) {
    agregarProducto(scanner, "naranjas");
    agregarProducto(scanner, "platanos");
    agregarProducto(scanner, "manzanas");
    agregarProducto(scanner, "berenjenas");
    agregarProducto(scanner, "zanahorias");
    agregarProducto(scanner, "calabacines");
    System.out.println("¡Compra realizada con éxito! Frutas y verduras añadidas al inventario.");
}

//Añade los productos comprados al inventario
static void agregarProducto(Scanner scanner, String producto) {
    System.out.println("Ingrese la cantidad de " + producto + " que ha comprado:");
    int cantidadComprada = scanner.nextInt();
    switch (producto) {
        case "naranjas" -> naranjas += cantidadComprada;
        case "platanos" -> platanos += cantidadComprada;
        case "manzanas" -> manzanas += cantidadComprada;
        case "berenjenas" -> berenjenas += cantidadComprada;
        case "zanahorias" -> zanahorias += cantidadComprada;
        case "calabacines" -> calabacines += cantidadComprada;
    }
}
```