

# Avance Proyecto: Inventario Supermercado

Integrantes: Sofia Labra, Sebastian Soza, Maite Villalón

## **Gestión de Inventario de un supermercado:**

Sistema de compra y venta de productos de supermercado con manejo automático de stock, generando reportes de órdenes de compra por sección y proveedores.

### **1.1 Realizar un análisis de los datos a utilizar y principales funcionalidades a implementar que dan sentido a la realización del proyecto.**

En este proyecto se propone presentar un programa de sistema de inventario de un supermercado. Nuestro objetivo es facilitar el manejo de inventario, realizando diferentes funcionalidades hasta el momento, tales como agregar y mostrar los productos. Trabajaremos con dos colecciones de elementos, Marcas y Productos, donde cada marca tendrá su propia lista de productos, estas fueron escogidas esperando hacer del programa lo más amigable e intuitivo para el usuario. A continuación presentaremos las siguientes clases y funcionalidades que se han implementado hasta el momento.

#### **Funcionalidades:**

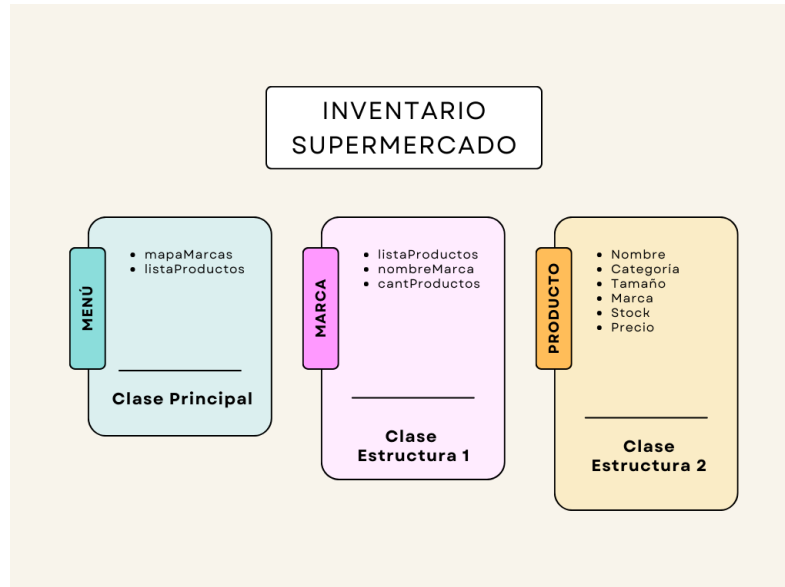
- Agregar marcas y productos, las marcas son guardadas en un HashMap, donde cada una de ellas guarda una lista de productos con sus atributos respectivos.
- Mostrar por pantalla todos los productos registrados.
- Mostrar por pantalla los productos registrados de una marca específica.

## 1.2 Diseño conceptual de clases del Dominio y su código en Java

El código con las clases, se encuentra en el repositorio github.

Existen 3 clases hasta el momento:

1. Clase Principal (Menú): En esta clase se basa el inicio de la aplicación.
2. Clase Estructura 1 (Marca): Clase que contiene los datos sobre cada marca.
3. Clase Estructura (Producto): Esta clase contiene los atributos de cada producto.



## 1.3 Todos los atributos de todas las clases deben ser privados y poseer sus respectivos métodos de lectura y escritura (getter y setter).

Clase marca contiene en el código sus respectivos getter y setter (Clase Marca, líneas 33-51)

Clase producto, contienen en el código sus respectivos getter y setter (Clase producto, líneas 38-91)

#### 1.4 Se deben incluir datos iniciales dentro del código.

En la clase Menú, en la línea 17 de main se hace llamado a el método encargado de inicializar las variables.

En el método agregar que recibe(HashMap, ArrayList, int) que comienza en la línea 57 y se encuentra en la clase anteriormente nombrada, la inicialización de estos datos están entre las líneas 63 y 74.

#### 1.5 Diseño conceptual y codificación de 2 colecciones de objetos, con la 2ª colección anidada como muestra la figura. Las colecciones pueden ser implementadas mediante arreglos o clases del Java Collections Framework

##### Diseño conceptual Estructura 1:

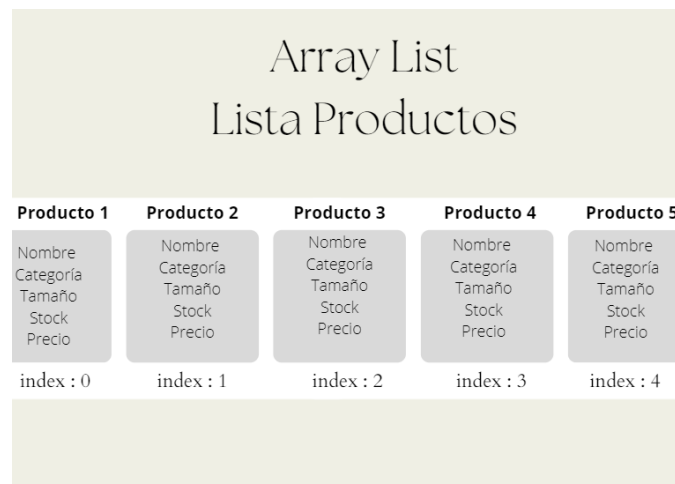
Contienen como **keys** los nombres de las marcas y los **values** serán la lista que contiene los productos de tal marca.



## Diseño conceptual Estructura 2:

Cada marca tiene su lista de productos, donde estos se guardan en un arraylist, se guardan desde la posición 0 hasta (cantidad de productos - 1).

Además, por cada producto tenemos sus atributos, tales como su nombre, su tamaño (gr o ml), categoría (lácteo, verduras, aseo, etc), stock y precio.



## 1.6 Diseño conceptual y codificación de 2 clases que utilicen sobrecarga de métodos (no de constructores).

Se usa el método agregar en dos casos, cuando el mapa marca está sin datos y cuando ya tiene datos para ingresar los productos.

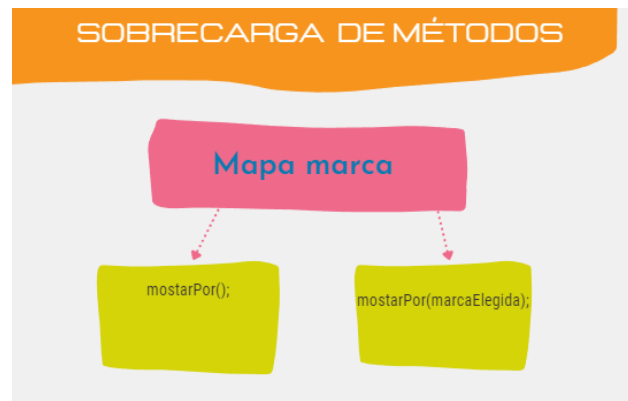
### Sobrecarga 1:

1. `agregar(HashMap, ArrayList, int)`: se encuentra en la clase “Menu” entre las líneas 57 y 107. Se encarga de agregar productos nuevos en el caso de existir la marca, sino, se agrega la marca y los productos.
2. `agregar(HashMap, ArrayList, Producto)`: se encuentra en la clase “Menu” entre las líneas 109 127. Se encarga de inicializar los atributos de cada producto agregado.



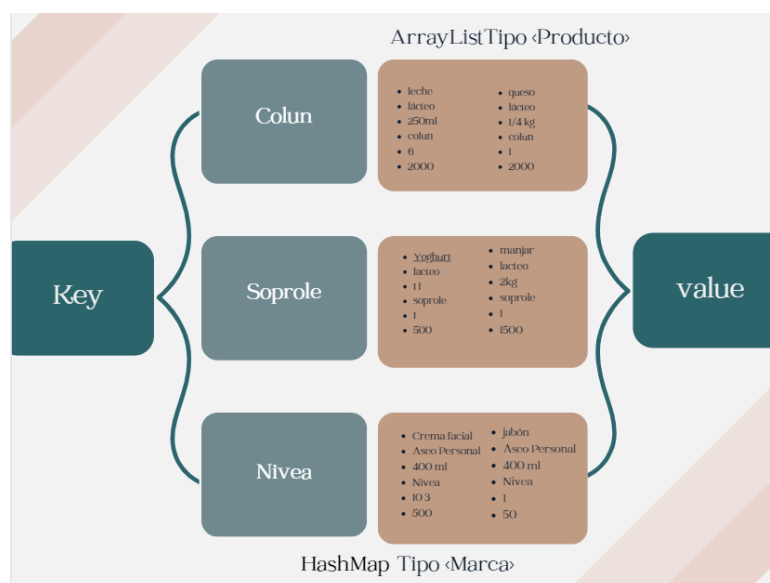
## Sobrecarga 2:

1. `mostrarPor()`: Este método se encuentra en la clase “Marca” desde la línea 62 y 72. Se encarga de mostrar todos las marcas con sus productos.
2. `mostrarPor(ArrayList)`: Este método, también se encuentra en la clase “Marca” entre las líneas 75 y 82. Se encarga de mostrar todos los productos de una marca en específica.



## 1.7 Diseño conceptual y codificación de al menos 1 clase mapa del Java Collections Framework.

Este es nuestro mapa que se compone por una key y por una ArrayList que sería el value.



## **1.8 Se debe hacer un menú para el Sistema**

Menú propuesto hasta el momento, contiene como opciones:

1. Agregar producto: Cuando el usuario desea agregar un producto, lo primero que se le consulta es el nombre de la marca del producto, en el caso que esta marca no esté registrada, se agrega a el mapa y se guarda el producto en la lista correspondiente.
2. Mostrar producto: Contiene dos opciones, la primera es que muestre todas las marcas y todos los productos que estas contienen y la segunda muestra solamente los productos de una marca específica.



## **1.9 Todas las funcionalidades deben ser implementadas mediante consola (Sin ventanas)**

Todas las funcionalidades implementadas hasta el momento son mediante consola y sin ventanas.

## **1.10 Utilización de GitHub (Realización de al menos 3 Commit)**

Se realizaron los 3 commit por grupo, se puede verificar en el repositorio de GitHub.