



**UNIVERSIDAD NACIONAL AUTÓNOMA DE  
MÉXICO**

**FACULTAD DE ESTUDIOS SUPERIORES ARAGÓN**

**INGENIERÍA EN COMPUTACIÓN**

**Estructura de Datos**

**Profesor. Jesús Hernández Cabrera**

**Grupo: 1360**

**Castro Vázquez Luis Alfredo**

## Descripción

Este proyecto se centra en el desarrollo de un sistema de gestión de inventario para un negocio, en el que se pueden agregar, eliminar, buscar y listar productos, mediante una interfaz gráfica. La temática es de un sistema de inventario básico, pero útil en muchos tipos de comercios donde se manejan existencias de productos como lo podrían ser una tienda de abarrotes, artículos deportivos, etc, y es fundamental para llevar un control adecuado de las existencias de los productos manejados en el negocio.

Dicho sistema de inventario utiliza un Árbol Binario de Búsqueda (ABB) para organizar los productos y facilitar las búsquedas rápidas y eficientes. En lugar de almacenar los productos en otra estructura de datos, el ABB optimiza las operaciones de búsqueda, inserción y eliminación de productos a través de una estructura de datos jerárquica.

## Explicación de la estructura de la base de datos:

La estructura de la base de datos en este proyecto está basada en un ABB ya antes mencionado, que organiza los productos de acuerdo a un ID que es asignado de según como se vayan agregando productos al inventario. Al usar este tipo de estructura nos permite realizar operaciones de búsqueda, inserción y eliminación de manera eficiente en términos de tiempo, ya que el árbol se mantiene balanceado en la medida en que se agregan productos.

## Clases y Relaciones:

### 1. Clase Producto:

#### ○ Atributos:

- `id_producto`: Un identificador único para cada producto.
- `nombre`: El nombre del producto.
- `precio`: El precio del producto.
- `cantidad`: La cantidad disponible de ese producto.

#### ○ Métodos:

- `__str__`: Proporciona una representación en cadena de caracteres del producto.

### 2. Clase Nodo para el ABB:

#### ○ Atributos:

- `producto`: Es un objeto de la clase Producto que contiene la información del producto.
- `izquierdo`: Unión con el hijo izquierdo.
- `derecho`: Unión con el nodo hijo derecho.

- **Función:** Cada producto que se agrega es almacenado en un nodo diferente, que a su vez tiene nodos hijos que almacenan otros productos, los cuales se ordenan de manera jerárquica de acuerdo a su ID.

### 3. Clase ABB (Árbol Binario de Búsqueda):

- **Atributos:**
  - **raiz:** Es el nodo raíz del árbol.
- **Métodos:**
  - **insertar:** Inserta un nuevo producto en el árbol el cual se almacena en un nodo. Si se quiere agregar un producto ya existente (con el mismo ID) solo se actualiza la cantidad del producto.
  - **buscar:** Busca un producto por su ID. Utiliza recursión para encontrarlo en el árbol.
  - **eliminar:** Elimina un producto o reduce su cantidad en el inventario.
  - **in\_orden:** Recorre el árbol para listar los productos de forma ordenada de acuerdo a su ID.

### 4. Clase Interfaz:

- **Atributos:**
  - **arbol:** Un objeto de la clase ABB la cual almacena los productos.
  - **sig\_id:** Es un contador para asignar un ID a cada producto.
- **Métodos:**
  - **agregar():** Permite agregar un nuevo producto o actualizar la cantidad de un producto existente.
  - **eliminar():** Permite eliminar un producto o reducir su cantidad en el inventario.
  - **buscar():** Permite buscar un producto por su ID.
  - **listar():** Muestra todos los productos en el inventario.

## Funcionamiento

1. **Agregar Producto:** El usuario debe ingresar los datos del producto (nombre, precio y cantidad). El sistema se encarga de verificar si el producto ya existe. Si ya existe, solo se incrementa su cantidad; si no existe, se le asigna un nuevo ID al producto y se agrega al árbol.
2. **Eliminar Producto:** El usuario debe de ingresar el ID del producto y la cantidad a eliminar. El sistema busca el producto y reduce su cantidad o elimina el producto si la cantidad llega a cero.

3. **Buscar Producto:** El usuario ingresa el ID del producto y el sistema realiza su búsqueda. Si lo encuentra manda un mensaje con los datos del producto; si no lo encuentra, manda un mensaje diciendo que el producto no existe.
4. **Listar Productos:** El sistema recorre el árbol y muestra los productos en orden por su ID.





