

PROYECTO

MARIA JULIANA PENJA MENDEZ
CARLOS SEBASTIAN NARANTO ARIAS.
ESTUDIANTE

JUAN CARLOS MARTINEZ DIAZ
PROFESOR

LOGICA COMPUTACIONAL.
UNIVERSIDAD ANTONIO
NARINO.

2024.

3. Por cada requerimiento funcional desarrollado en el numeral 1-2, programa en un lenguaje de programación.

Nombre: Mostrar Productos

Parámetros: lista de productos de la tienda.

Retorno: Muestra la información de todos los productos.

Descripción: Función para visualizar la información de todos los productos de la tienda incluyendo nombre, tipo, cantidad actual, precio base y precio final.

`void mostrarProductos(ListaProductos productos)`

`{
 cout << "Información de productos:" <<
 endl;`

`for (const auto & producto: Productos)`

`{
 cout << "Nombre:" << producto.Nombre
 << endl;`

`cout << "Tipo:" << producto.tipo <<
 endl;`

`cout << "Cantidad Actual:" << producto.
 CantidadActual << endl;`

`cout << "Precio Base" << producto.
 PrecioBase << endl;`

`cout << "Precio Final:" << producto.
 precioFinal << endl;
}`

Nombre: Vender producto

Parámetros: Nombre del producto a vender y cantidad de unidades a vender

Retorno: El dinero obtenido por la venta o un mensaje si el producto no se encuentra.

Descripción: Función encargada de vender

`float venderProducto(string nombrePro
ducto, int cantidad, ListaProductos &
productos)`

`{
 for (auto & productos: productos)`

`{
 if (producto.nombre == nombreProduc
to).`

`if (producto.CantidadActual >=
 cantidad)`

`{
 producto.CantidadActual -=
 cantidad;`

`return producto.precioFinal *
 Cantidad;`

`else
}`

una cantidad específica de un producto y actualizar el inventario.

```
cout << "No hay suficiente stock para vender." << endl;
return 0;
```

```
cout << "Producto no encontrado." << endl;
return 0;
```

Nombre: Abastecer Producto

Void abastecerProducto(string nombreProducto, int cantidad, ListaProductos & productos)

Parámetros: Nombre del producto a abastecer y cantidad de unidades.

```
for (auto & producto : productos)
{
    if (producto.nombre == nombreProducto)
    {
```

Retorno: Muestra un mensaje indicando que el producto fue abastecido.

```
    producto.CantidadActual += cantidad;
    cout << "Producto " << nombreProducto << " Abastecido con éxito. Nueva cantidad: " << producto.CantidadActual << endl;
```

Descripción: Función que permite abastecer el inventario con una cantidad adicional de un producto especificado.

```
    return;
```

```
    cout << "Producto no encontrado." << endl;
```

Nombre: Cambiar Producto

Void cambiarProducto(string nombreProductoActual, string nuevoNombre, string nuevoTipo, int nuevaCantidad, int cantidadMinima, float nuevoPrecioBase, ListaProductos & productos)

Parámetro: Nombre del producto actual y nuevos datos del pro-

ducto (nombre, tipo, cantidad, mínima y precio base.

Retorno: Muestra un mensaje indicando que el producto ha sido actualizado.

Descripción: Función para modificar la información de un producto incluyendo nombre, tipo, cantidad, etc.).

```
for (auto & producto : productos)
```

```
{  
    if (producto.nombre == nombre  
        ProductoActual)
```

```
{  
    producto.nombre = nuevoNombre;
```

```
    producto.tipo = nuevoTipo;
```

```
    producto.cantidadActual =  
        nueva_cantidad;
```

```
    producto.cantidadMinima =  
        cantidadMinima;
```

```
    producto.precioBase = nuevoPre  
        cioBase
```

```
    cout << "Producto Actualizado  
        con éxito" << endl;
```

```
    return;
```

```
}
```

```
}  
cout << "Producto no encontrado."  
    << endl;
```