

Universidad Mariano Gálvez

Centro Regional: Boca del monte

Facultad: Ingeniería en sistemas

Curso: Algoritmos

Sección: C



Nombre: Elmer Yúnior Roblero Santos

Numero de carnet: 7690-23-14847

Fecha: 13/10/2023

Guatemala, de 2023

Documentación Externa del código en python

Documentación del Sistema de Gestión de Productos

Descripción

Este sistema de gestión de productos está diseñado para ayudar a administrar una lista de productos. Permite agregar productos, buscar productos por código o nombre, modificar los datos de un producto existente y guardar la información en un archivo para futuras sesiones.

Clases

LosProductos

Esta clase define la estructura de un producto. Cada producto tiene los siguientes atributos:

- ``losNombres``: El nombre del producto.
- ``losCodigos``: El código único del producto.
- ``losPrecios``: El precio del producto.
- ``Proveedores``: El proveedor del producto.
- ``Existencias``: La cantidad de existencias disponibles.
- ``LosEstados``: El estado del producto (A = Aprobado, R = Reprobado).
- ``LosDescuentos``: El descuento aplicado al producto.

Funciones

`agregar_LosProductos(LosProductoss, LosProductos)`

Esta función agrega un producto a la lista de productos ``LosProductoss``. Verifica si el código del producto ya existe en la lista y muestra un mensaje de error si es el caso.

`buscar_LosProductos(LosProductoss, busqueda)`

Esta función busca productos en la lista ``LosProductoss`` por código o nombre. Devuelve una lista de productos que coinciden con la búsqueda.

`modificar_LosProductos(LosProductoss, losCodigos, Nuevos_losPrecios, Nuevos_Proveedores, Nueva_Existencias, Nuevos_LosEstados, Nuevos_LosDescuentos)`

Esta función permite modificar los datos de un producto existente en la lista ``LosProductoss``. Busca el producto por su código y actualiza sus atributos con los nuevos valores proporcionados.

-Uso del Programa

Ejecuta el programa y selecciona una opción del menú:

- para agregar un producto. Debes proporcionar los detalles del producto.
- para buscar un producto por código o nombre. Ingresas la búsqueda y muestra los resultados.
- para modificar los datos de un producto existente. Proporciona el código del producto y los nuevos valores.
- para salir del programa. Los datos se guardarán en un archivo "LosProductoss.pkl".

2. Si eliges la opción

- 4 los datos se guardarán en un archivo "LosProductoss.pkl" para su uso posterior.

Ejemplo de Uso

- Agregar un producto:

- losNombres: Producto1
- Código de LosProductos: 001
- losPrecios de LosProductos: 19.99
- Proveedores: Proveedor1
- Existencias: 50
- LosEstados (A = Aprobado, R = Reprobado): A
- LosDescuentos: 5.0

- Buscar un producto:

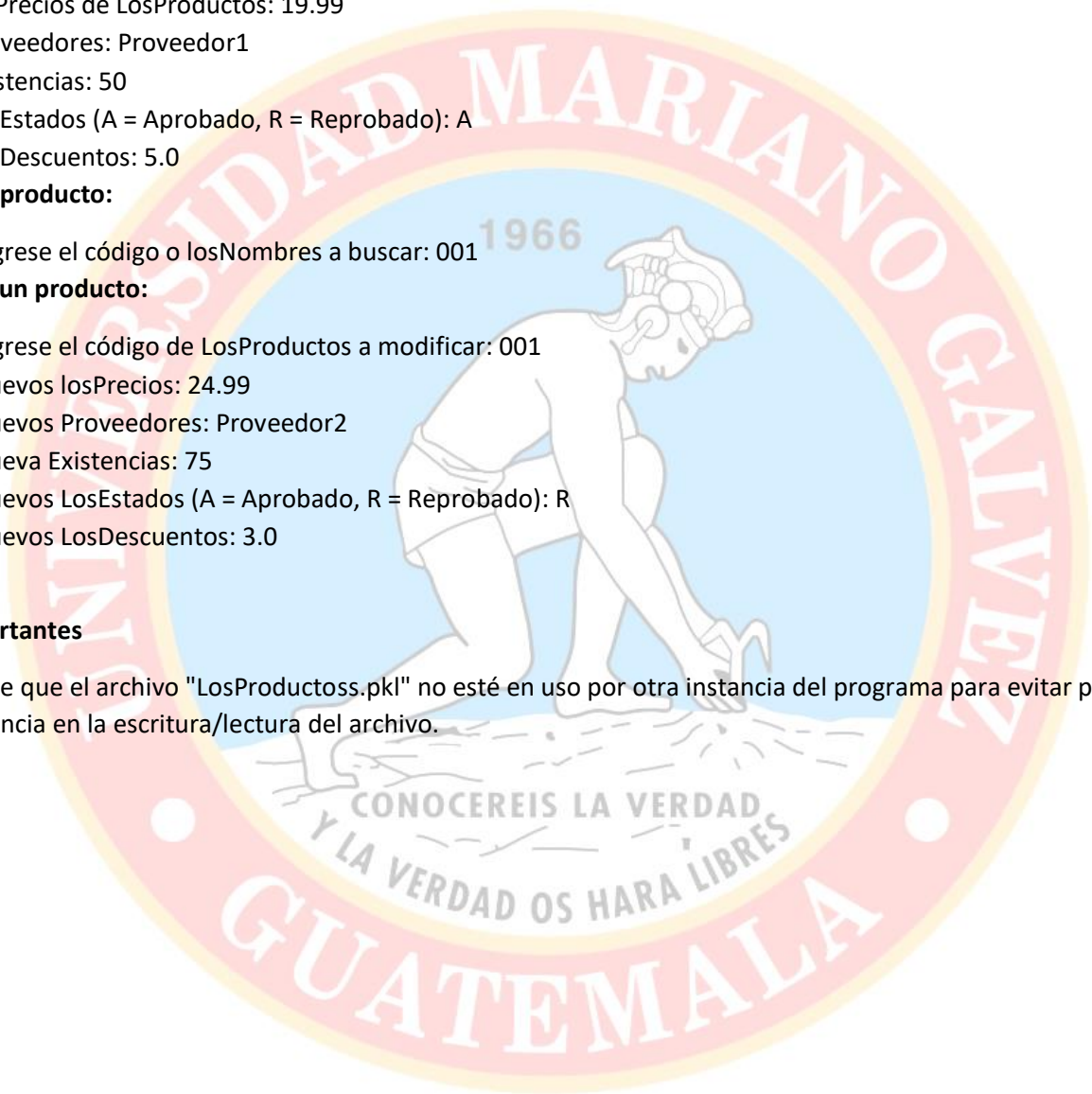
- Ingrese el código o losNombres a buscar: 001

- Modificar un producto:

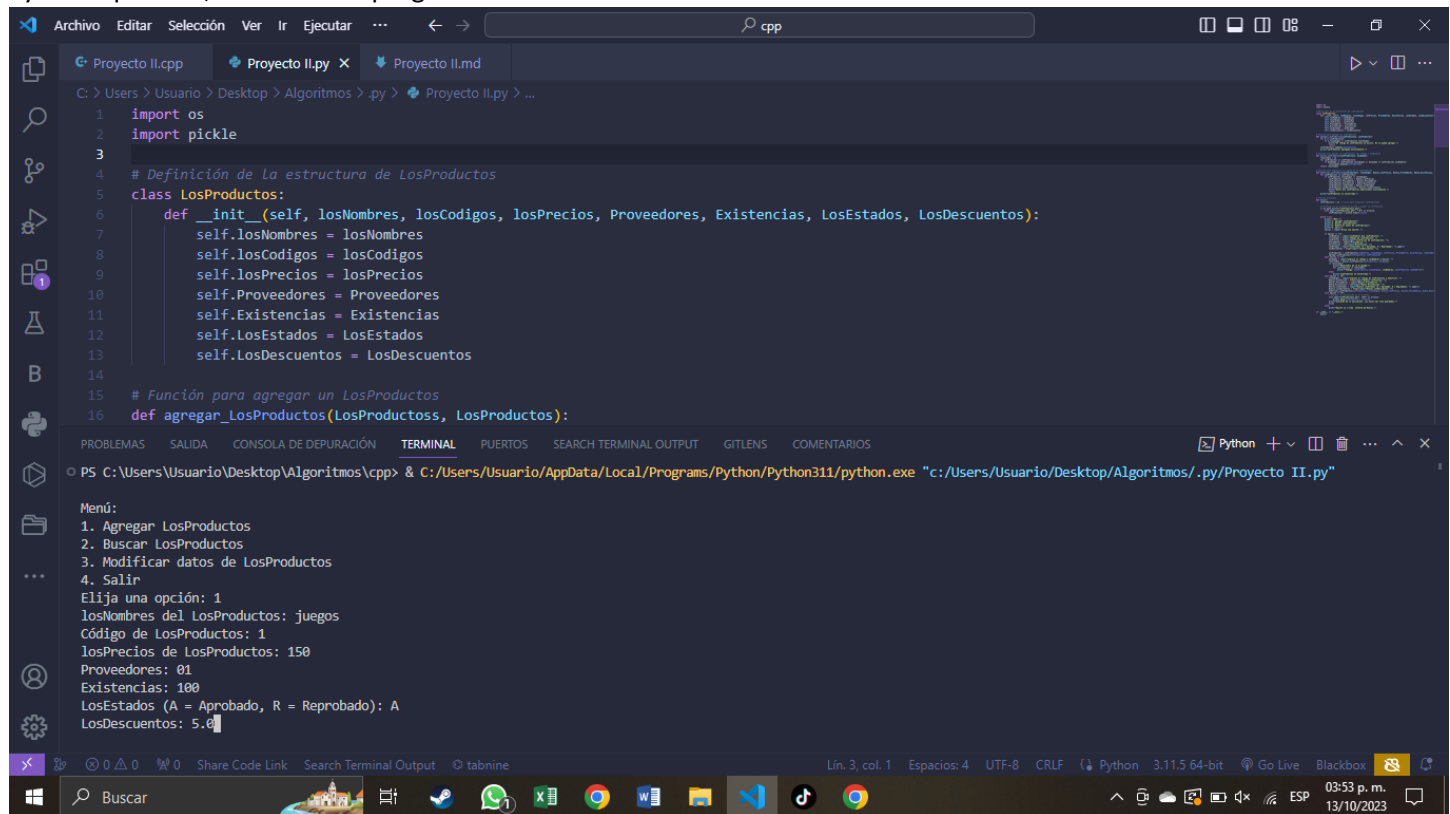
- Ingrese el código de LosProductos a modificar: 001
- Nuevos losPrecios: 24.99
- Nuevos Proveedores: Proveedor2
- Nueva Existencias: 75
- Nuevos LosEstados (A = Aprobado, R = Reprobado): R
- Nuevos LosDescuentos: 3.0

Notas Importantes

Asegúrate de que el archivo "LosProductoss.pkl" no esté en uso por otra instancia del programa para evitar problemas de concurrencia en la escritura/lectura del archivo.



Esta es la documentación básica del sistema de gestión de productos. Si tienes alguna pregunta adicional o necesitas ayuda específica, no dudes en preguntar.



```
Archivo  Editor  Selección  Ver  Ir  Ejecutar  ...  cpp
Proyecto II.cpp  Proyecto II.py  Proyecto II.md
C:\Users\Usuario\Desktop> Algoritmos\ .py > Proyecto II.py > ...
1  import os
2  import pickle
3
4  # Definición de la estructura de LosProductos
5  class LosProductos:
6      def __init__(self, losNombres, losCodigos, losPrecios, Proveedores, Existencias, LosEstados, LosDescuentos):
7          self.losNombres = losNombres
8          self.losCodigos = losCodigos
9          self.losPrecios = losPrecios
10         self.Proveedores = Proveedores
11         self.Existencias = Existencias
12         self.LosEstados = LosEstados
13         self.LosDescuentos = LosDescuentos
14
15     # Función para agregar un LosProductos
16     def agregar_LosProductos(LosProductoss, LosProductos):
17
18 PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS  SEARCH TERMINAL OUTPUT  GITLENS  COMENTARIOS
PS C:\Users\Usuario\Desktop\Algoritmos\cpp> & C:/Users/Usuario/AppData/Local/Programs/Python/Python311/python.exe "c:/Users/Usuario/Desktop/Algoritmos/.py/Proyecto II.py"
Menú:
1. Agregar LosProductos
2. Buscar LosProductos
3. Modificar datos de LosProductos
4. Salir
Elija una opción: 1
losNombres del LosProductos: juegos
Código de LosProductos: 1
losPrecios de LosProductos: 150
Proveedores: 01
Existencias: 100
LosEstados (A = Aprobado, R = Reprobado): A
LosDescuentos: 5.0
```



Documentación Externa del Código en C++

Documentación del Sistema de Gestión de Productos

Descripción

Este código en C++ gestiona una lista de productos llamados `LosProductos`. Proporciona funciones para agregar, buscar y modificar productos. Además, los datos de los productos se almacenan en un archivo binario para su persistencia.

Estructura de LosProductos

La estructura `LosProductos` representa un producto y contiene los siguientes campos:

- `losNombres` (string): El nombre del producto.
- `losCodigos` (string): El código del producto.
- `losPrecios` (double): El precio del producto.
- `Proveedores` (string): El proveedor del producto.
- `Existencias` (int): La cantidad de existencias del producto.
- `LosEstados` (char): El estado del producto (A = Aprobado, R = Reprobado).
- `LosDescuentos` (double): El descuento aplicado al producto.

Funciones Principales

1. `agregar_LosProductos`: Agrega un producto a la lista de `LosProductoss`. Verifica que el código del producto no exista previamente.
2. `buscar_LosProductos`: Busca productos en la lista por código o nombre y devuelve una lista de resultados.
3. `modificar_LosProductos`: Modifica los datos de un producto en la lista utilizando su código.

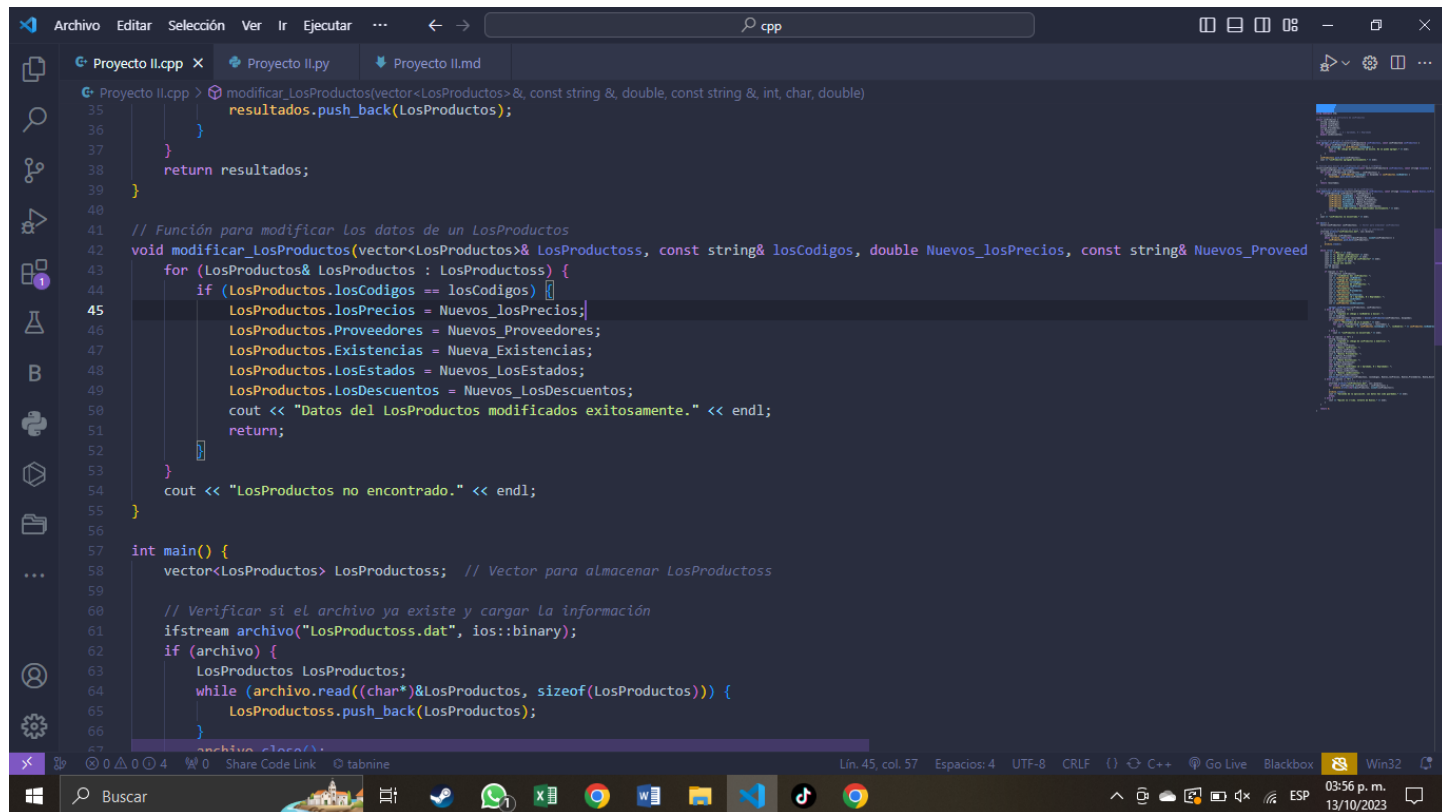
Flujo Principal

1. El programa carga datos de productos desde un archivo binario si existe.
2. Muestra un menú con opciones:
 - Agregar un producto
 - Buscar un producto por código o nombre
 - Modificar los datos de un producto
 - Salir del programa
3. Los datos ingresados por el usuario se procesan según la opción elegida.
4. Los productos y las modificaciones se almacenan en un archivo binario antes de salir del programa.

Uso

1. Compila el código utilizando un compilador C++.
2. Ejecuta el programa y sigue las opciones del menú para gestionar productos.
3. Los datos de los productos se guardarán en un archivo binario llamado "LosProductoss.dat" para su persistencia.

Esta es una documentación externa simplificada para comprender el propósito y las funciones del código en C++. Puedes expandirla según tus necesidades específicas y agregar detalles adicionales, como requisitos del sistema o ejemplos de uso.



```
Proyecto II.cpp > modificar_LosProductos(vector<LosProductos>&, const string& double, const string& int, char, double)
35     resultados.push_back(LosProductos);
36 }
37 }
38 return resultados;
39 }
40
41 // Función para modificar Los datos de un LosProductos
42 void modificar_LosProductos(vector<LosProductos>& LosProductoss, const string& losCodigos, double Nuevos_losPrecios, const string& Nuevos_Proveed
43 for (LosProductos& LosProductos : LosProductoss) {
44     if (LosProductos.losCodigos == losCodigos) {
45         LosProductos.losPrecios = Nuevos_losPrecios;
46         LosProductos.Proveedores = Nuevos_Proveedores;
47         LosProductos.Existencias = Nueva_Existencias;
48         LosProductos.LosEstados = Nuevos_LosEstados;
49         LosProductos.LosDescuentos = Nuevos_LosDescuentos;
50         cout << "Datos del LosProductos modificados exitosamente." << endl;
51         return;
52     }
53 }
54 cout << "LosProductos no encontrado." << endl;
55 }
56
57 int main() {
58     vector<LosProductos> LosProductoss; // Vector para almacenar LosProductoss
59
60     // Verificar si el archivo ya existe y cargar la información
61     ifstream archivo("LosProductoss.dat", ios::binary);
62     if (archivo) {
63         LosProductos LosProductos;
64         while (archivo.read((char*)&LosProductos, sizeof(LosProductos))) {
65             LosProductoss.push_back(LosProductos);
66         }
67     }
68     archivo.close();
69 }
```

