

Proyecto: Gestor de Becas para Estudiantes

Este proyecto tiene como objetivo simular un sistema simple de gestión de becas para estudiantes, utilizando los conceptos fundamentales de Python: variables, operadores, estructuras de control, funciones, listas, diccionarios y manejo de errores.

Objetivos del proyecto

- Registrar estudiantes becarios.
- Consultar la lista de becarios registrados.
- Buscar becarios por nombre.
- Calcular el pago total mensual.
- Eliminar becarios.
- (Opcional) Editar información y guardar los datos en un archivo.

Elementos de programación que deberán usar

- Variables y operadores.
- Estructuras de control: if, for, while.
- Control de flujo: break, continue, pass.
- Funciones.
- Listas y diccionarios.
- Manejo de errores con try y except.

Estructura general del programa

El programa debe mostrar un menú interactivo con las siguientes opciones:

1. Agregar becario
2. Ver lista de becarios
3. Buscar becario por nombre
4. Calcular pago total
5. Eliminar becario
6. Salir

Ejemplo de estructura de datos

```
[{"nombre": "Ana Pérez", "nivel": "Licenciatura", "monto": 4000.0}, {"nombre": "Luis Gómez", "nivel": "Maestría", "monto": 6000.0}]
```

Instrucciones de desarrollo

1. Crea una lista vacía llamada `becarios`.
2. Define una función `mostrar_menu()` que imprima las opciones del menú.
3. Crea las funciones siguientes:
 - `agregar_becario()` → pide nombre, nivel y monto, y agrega un diccionario a la lista.
 - `ver_becarios()` → usa `enumerate()` para mostrar el número y datos de cada becario.
 - `buscar_becario()` → busca por nombre usando `in` y muestra los resultados.

- `calcular_total()` → suma los montos de todos los becarios y muestra el total.
 - `eliminar_becario()` → pide el nombre y elimina el registro correspondiente.
4. En la parte principal del programa, usa un ciclo `while True:` para mostrar el menú y pedir al usuario una opción.
5. Usa `break` para salir del ciclo cuando el usuario elija la opción de salir.

Conceptos clave

1. `try` y `except` sirven para manejar errores y evitar que el programa se detenga.
2. `return` devuelve un valor o termina una función anticipadamente.
3. `enumerate()` permite recorrer una lista y obtener el índice y el valor al mismo tiempo.

Extensiones opcionales

- Agregar opción para editar información de un becario.
- Calcular el promedio de monto por nivel académico.
- Guardar y cargar los datos en un archivo CSV o TXT.
- Validar los datos ingresados (por ejemplo, que el monto no sea negativo).

Mensaje final

Recuerda que este proyecto busca reforzar tu comprensión de los elementos básicos de Python. Concéntrate en escribir código claro, bien indentado y fácil de leer. Comenta las partes principales para explicar qué hace cada función.