

Universidad Autónoma de Baja California Facultad de Ingeniería Arquitectura y Diseño



Materia: Lenguaje de Programación Python

Alumno: FABIAN AGUIAR SERGIO

Carrera: Ingeniero en computación

Fecha: 21/05/2024

Matrícula: 374317

Maestro: Pedro Nuñez Yepiz

Grupo: 432

Práctica No.: #12

Tema - Unidad : Diccionarios

Introducción

En esta práctica se desarrolló un programa en Python que permite gestionar información básica de los trabajadores de una fábrica. El programa proporciona un menú interactivo con varias opciones, como agregar registros automáticamente, eliminar registros, buscar por ID, ordenar la lista, generar archivos en diferentes formatos y cargar/mostrar archivos existentes. La práctica está diseñada para demostrar el uso de listas, diccionarios y técnicas de formato de salida en Python.

Objetivos

- Implementar funciones para validar números enteros.
- Utilizar listas y diccionarios para almacenar y gestionar datos.
- Generar una salida formateada y legible.
- Crear y manejar archivos de texto.
- Validar las operaciones del programa asegurando su correcto funcionamiento.

Teoría

Listas en Python

Una lista en Python es una colección ordenada y mutable de elementos. Las listas permiten almacenar elementos de diferentes tipos de datos y ofrecen diversas operaciones como añadir, eliminar, y modificar elementos.

Ejemplo:

```
mi_lista = [1, 2, 3, 4, 5]
mi_lista.append(6)
print(mi_lista) # Salida: [1, 2, 3, 4, 5, 6]
```

Diccionarios en Python

Un diccionario en Python es una colección no ordenada de elementos. Cada elemento tiene una clave única que se asocia con un valor. Los diccionarios son mutables y permiten una búsqueda rápida de elementos.

Ejemplo:

```
mi_diccionario = {'nombre': 'Juan', 'edad': 30}
print(mi_diccionario['nombre']) # Salida: Juan
```

Impresión Bonita de Diccionarios

Para imprimir diccionarios de manera más legible y organizada, se puede utilizar la librería pandas para convertir el diccionario en un DataFrame y luego mostrarlo como una tabla.

Ejemplo:

```
import pandas as pd
data = [{'Id': 1, 'Nombre': 'Juan'}, {'Id': 2, 'Nombre': 'María'}]
df = pd.DataFrame(data)
print(df)
```

Resultados

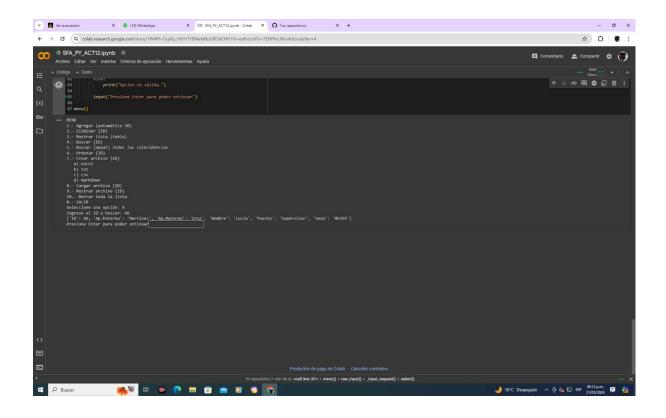
El programa permite gestionar eficientemente una lista de registros de trabajadores, con opciones para agregar, eliminar, buscar, ordenar, y guardar en distintos formatos. Además, asegura la integridad de los datos mediante validaciones y confirma las acciones críticas con el usuario.

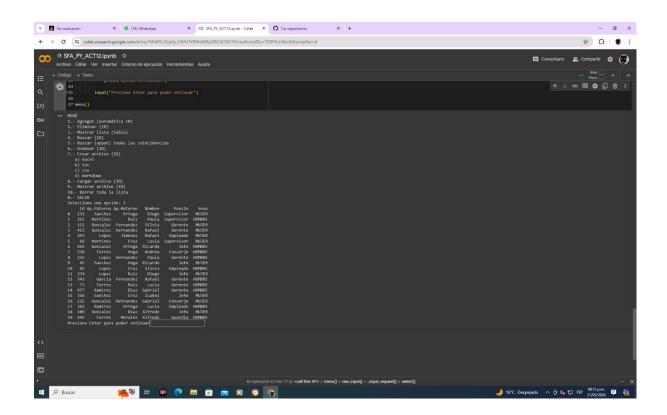
Conclusión

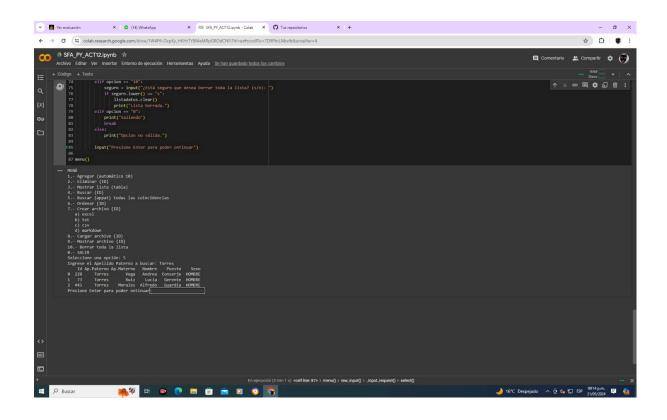
La práctica desarrollada ha permitido implementar y comprender diversas funcionalidades avanzadas relacionadas con la gestión de datos, utilizando conceptos fundamentales de programación en Python. La creación de un sistema de gestión de registros de trabajadores ha brindado una excelente oportunidad para aplicar el uso de listas y diccionarios, así como la manipulación de archivos en diferentes formatos. A lo largo del desarrollo del programa, se han abordado y resuelto varios desafíos significativos que han contribuido a un entendimiento más profundo de la programación estructurada y orientada a objetos.

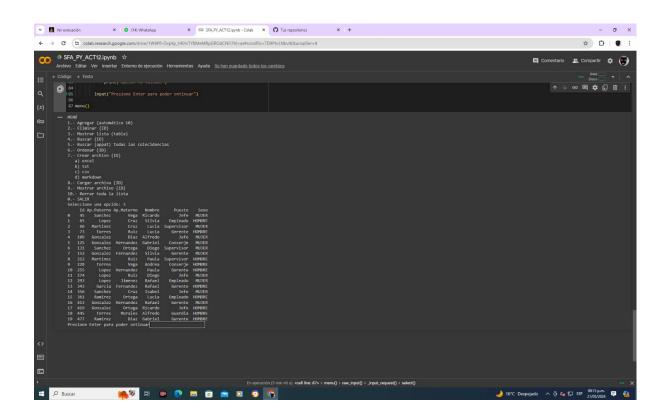
Esta práctica no solo refuerza los conceptos teóricos sobre estructuras de datos y técnicas de manipulación de información, sino que también proporciona una experiencia práctica en la creación de aplicaciones funcionales. permitiendo aplicar y consolidar conocimientos sobre listas, diccionarios y técnicas de impresión en Python. También se trabajó con la manipulación de archivos y se implementaron buenas prácticas de programación para validar entradas y confirmar acciones.

ANEXOS









GIT HUB

https://github.com/sergiofabian05/SFA_PY_ACTS.git

Aquí están todos los códigos..

EN ESTE ESTA SOLO EL REPORTE Y CÓDIGO

https://github.com/sergiofabian05/SFA_PY_ACT12.git