Práctica Única

Manual Técnico

201906588 Byron Estuardo Solís González

Primer Semestre – Febrero 2021

1. Introducción

Desarrolló una aplicación que permite la lectura de un archivo de texto plano que contiene diferentes listas de valores enteros positivos separados por comas, una por línea. Siendo capaz de realizar dos operaciones con la información proporcionada por cada fila, ordenar los valores de la lista o buscar un valor en la lista.
En detalle las funciones de la aplicación son las siguientes: Cargar archivo de entrada, desplegar listas ordenadas, desplegar búsquedas, desplegar todas y desplegar todas a archivo HTML.

2. Objetivos

General: Dar una solución de software con base en los distintos paradigmas de programación, aplicando conocimientos adquiridos en clase.

Específicos

- Aplicar los conceptos generales sobre lenguajes formales, tales como alfabeto, símbolos, cadenas y reglas.

- Conocer las características principales del lenguaje de programación Python.

- Aplicar los algoritmos de búsqueda y ordenamiento.

3. Datos técnicos

- Lenguaje utilizado para la aplicación: Python
- DE utilizado: Visual Studio Code Herramientas de subida a la nube: Heroku
- Lenguaje utilizado para la generación del archivo: HTML

4. Requerimientos

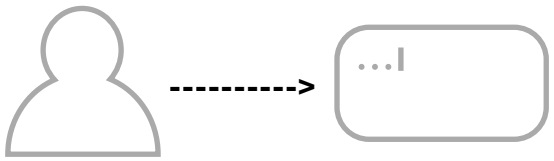
La aplicación puede ser ejecutada en cualquier sistema operativo que cumpla con los siguientes requerimientos:
- Python instalado
- Software con la capacidad de crear archivos de texto plano y/o archivos de texto plano.
- Navegador Web

Estructura del arcvido de entrada

Nombre de lista = Contenido , Nombre y especificaciones de la opercación

LISTA1 =1,4, 3,5, 7, 5 ORDENAR, BUSCAR 8

4. Flujo de trabajo



5. Paradigas utilizados

POO Se utilizó este paradigma para crear objetos de clase lista, para poder llevar de una forma más organizada el control de los datos provenientes del archivo, etos a ser leidos iban siendo ingresados a un objeto de tipo lista, y estos a su vez se registraban en una lista de objetos de clase lista.

Objeto Lista:

Atributos:

- id: (str) Nombre de la lista
- contenido: (list) Contenido de la lista
- o_ordenar: (boolean) Vendadero si se desea un ordenamiento
- o_buscar: (boolean) Verdadero si se desea realizar una búsqueda
- numeros_busqueda: (str) Valor a buscar

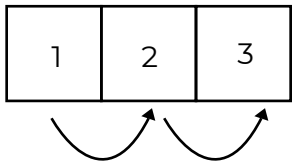
Funciones

- getId( ): Retorna un str con el nombre de la lista.
- getContenido_lista( ): Retorna una lista con el contenido de la lista.
- getContenido( ): Retorna un str con el contenido de la lista, separado por espacios.
- getO_ordenar( ) :Retorna un valor booleano
- getO_buscar( ) : Retorna un valor booleano
- getNumeros_busqueda( ) :Retorna un str con el datoa buscar
- getPosiciones( ) : Retorna las posiciones donde se encontro el dato buscado

6. Algoritmos utilizados

Busqueda Secuencial: Recorremos todos los elementos de la lista, simplemente nos trasladamos de un ítem a otro, siguiendo el orden secuencial subyacente hasta que encontremos lo que buscamos o nos quedemos sin ítems.

Se utilizó este algoritmo de busqueda porque era neseario comparar todos los datos del contenido de la lista, para determinar si eran iguales a los que se buscaban, y si lo eran obteníamos también su ubicación.



Ordenamiento por selección: Este busca el valor mayor a medida que hace una pasada y, después de completar la pasada, lo pone en la ubicación correcta. Después de la primera pasada, el ítem mayor está en la ubicación correcta. Después de la segunda pasada, el siguiente mayor está en su ubicación. Este proceso continúa y requiere n−1 pasadas para ordenar los n ítems, ya que el ítem final debe estar en su lugar después de la (n−1)-ésima pasada.

Se utilizó este algoritmo por ser de eficiencia media, ya que ahorra recursos al hacer solo un cambio de posición por pasada