

Universidad De San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas

Lenguajes Formales y de Programación
Sección "B-"



MANUAL TÉCNICO:

Reporte de notas

Estudiante: José Ernesto Pajoc Raymundo

Carné: 201115455

Segundo semestre del 2,021

Presentación de la práctica: 19/08/2021

Objetivos

General:

Presentar las clases, métodos y funciones que conforman la práctica para el funcionamiento adecuado del software.

Específicos:

- Mostrar la descripción sobre la plataforma en la cual fue desarrollado el software.
- Identificar las librerías, funciones, métodos y atributos que conforman la parte principal del programa.

Introducción

El programa corresponde a la práctica 1, el cual es una aplicación desarrollada en el lenguaje de programación Python, este cuenta con un menú principal manejado a través de valores numéricos enteros, ingresados por consola, en él se puede cargar un archivo con extensión LFP el cual cuenta con el nombre de un curso y un grupo de estudiantes con sus respectivas notas, luego estos datos cargados al sistema pueden ser mostrados en dos tipos de reportes con sus respectivos parámetros para la toma de decisiones.

Información técnica

El software se desarrolló utilizando las siguientes tecnologías:

- Sistema operativo: Windows 10 (64 bits).
- Lenguaje utilizado: Python 3.8
- IDE: Visual Code, Versión: 1.56.2
- Navegador web: Google Chrome Versión 92.0.4515.159

Descripción de la Solución

Para poder desarrollar la práctica se analizo primero la información y el tipo de archivo que se debe cargar al sistema, con ello se crearon luego objetos que permiten gestionar y administrar la información de forma sencilla. El software se encuentra dividido en dos archivos los cuales cuentan con lo siguiente:

Tabla I

Componentes del archivo alumnos.

Alumnos.py	
Clase	estudiante: se utiliza para crear los objetos de tipo estudiante con los atributos nombre y nota, los métodos que posee son verNombre y verNota los cuales son utilizados para obtener la información de cada alumno.

Fuente: elaboración propia.

Tabla II

Componentes del archivo inicio.

inicio.py	
Librerías utilizadas	<p>Tkinter: esta librería permite utilizar entorno gráfico, lo cual es necesario para crear la ventana que permite seleccionar el archivo con los datos lfp.</p> <p>io: La función de esta librería es poder leer y cargar archivos externos.</p> <p>webbrowser: Permite controlar el navegador web predeterminado en el sistema operativo, el cual se abre automáticamente al momento de que el usuario seleccione exportar reporte.</p>
Módulos	Se importa la clase estudiante del módulo alumnos, esto con el fin de poder crear objetos para almacenar la información que se extrae

	del archivo cargado lfp.
Atributos	<p>opcion: se utiliza para almacenar el número de opción que ingrese el usuario.</p> <p>gano: la variable se utiliza como acumulador para llevar el control de los alumnos que ganaron.</p> <p>perdio: la variable se utiliza como acumulador para llevar el control de los alumnos que perdieron.</p> <p>promedio: esta variable permite guardar el calculo del promedio de todas las notas cargadas.</p> <p>ruta: almacena la ruta exacta de donde se encuentra el archivo externo lfp.</p> <p>archivoCargado: se utiliza como variable booleana para restringir la creación de reportes, ya que si esta no es VERDADERA no permite continuar.</p> <p>archivoLFP: almacena en una lista cada línea que se encuentra dentro del archivo lfp.</p> <p>curso: almacena el nombre del curso que viene en el archivo lfp.</p> <p>parametro: almacena los parámetros del curso que vienen en el archivo lfp.</p> <p>alumnos: esta lista permite almacenar los objetos de tipo estudiante.</p> <p>listaParametros: en esta lista se almacena cada parámetro que se encuentra en la última fila de información del archivo lfp.</p> <p>repConsola: se utiliza como variable booleana para verificar de que se ha llamado a la función mostrarConsola().</p>
Funciones	<p>separador(): esta función muestra una línea de guiones en consola, es utilizada como separador ya que su fin es separar cada proceso de forma gráfica.</p> <p>cargarArchivo(): la función permite crear una venta en la cual es usuario selecciona el archivo lfp, si el usuario cancela la ventana se muestra un mensaje de que no se cargo el archivo al sistema, en el caso de que si se cargo abre el archivo en un bufer y extrae cada línea del lfp para almacenarlo en la lista llamada archivoLFP, luego hace el llamado de la función procesarArchivo().</p> <p>procesarArchivo(): esta función toma como base la lista archivoLFP ya que de la posición 0 elimina los caracteres "=", "{" y luego almacena la información de esa posición en la variable curso. De la</p>

	<p>última posición de la lista extrae la información y elimina el carácter “}” para almacenarlo en la variable parámetros. Por último se utiliza un ciclo el cual permita recorrer la lista desde la posición 1 hasta la penúltima para poder eliminar los caracteres que no son necesarios y solo dejar los nombres con sus notas, los cuales son almacenados en una lista de objetos llamada alumnos, al finalizar el ciclo cambia e estado de la variable archivoCargdo.</p> <p>ascendenteNotas(): utiliza como base la lista de objetos alumnos para poder aplicar el método burbuja, el cual ordena en forma ascendente las notas de los alumnos.</p> <p>descendenteNotas(): permite recorrer la lista de objetos alumnos desde la última posición hasta la primera, en cada iteración se muestra el nombre y la nota de cada alumno en consola.</p> <p>ganoPerdioPromedio(): la función utiliza un ciclo que recorre toda la lista de objetos alumnos en donde se verifica la nota de cada alumno para hacer el conteo de los que aprobaron, reprobaron y cálculo del promedio.</p> <p>mostrarConsola(): en esta función primero se hace el llamado de las funciones ascendeteNotas(), ganaPerdioPromedio() y separador() con el fin de tener listo los resultados según los parámetros solicitados. Luego cambia el estado de la variable repConsola, después se utiliza un ciclo el cual recorre cada posición de la lista listaParametros y así poder mostrarlo en consola, en cada iteración se hace el cambio a mayúsculas para evitar inconvenientes en los parámetros.</p> <p>mostrarReporteHTML(): se evalúa si el estado de la variable repConsola es verdadero, si este es falso entonces se hace el llamado de las funciones ascendenteNotas() y ganaPerdioPromedio() con el fin de no tener inconvenientes al crear el archivo HTML. Luego se crea el archivo CSS para los estilos de la página web, después se crea el archivo HTML con su estructura básica, tabla de estudiantes y listados según los parámetros solicitados, por último se muestra un mensaje en consola de que se ha creado el archivo y se abre automáticamente en el navegador web.</p>
Proceso principal	Se utiliza un ciclo mientras para poder mostrar el menú en consola, en el cual el usuario puede ingresar el número que opción que desea, los reportes no se pueden crear sin antes haber cargado y procesado la información del archivo.

Fuente: elaboración propia.

Observaciones

- La información de los estudiantes dentro del archivo lfp deben venir en diferentes líneas, ya que si hay más de un estudiante en la misma línea se producirá un error en el proceso de datos.
- Los parámetros pueden estar escritos en mayúsculas o minúsculas ya que no producen ningún tipo de error.
- Los estudiantes dentro del archivo lfp pueden tener o no al inicio una tabulación ya que esto no produce ningún inconveniente.
- Las notas de los alumnos pueden ser números enteros o decimales.

Diagrama de actividades

