



# **Practica 1**



Guatemala, 8 agosto del 2021

## Índice de contenido

---

|                                               |   |
|-----------------------------------------------|---|
| Índice de contenido                           | 2 |
| 1 Definición de la solución                   | 3 |
| 1.1 Objetivos                                 | 3 |
| 1.2 Alcances del proyecto                     | 3 |
| 1.3 Requerimientos del sistema                | 4 |
| <b>1.3.1 Requerimientos funcionales</b>       | 4 |
| <b>1.3.2 Descripción</b>                      | 4 |
| <b>1.3.3 Ejemplo de un archivo de entrada</b> | 4 |
| 1.4 Componentes                               | 5 |
| <b>1.4.1 Diagrama de clases</b>               | 5 |
| <b>1.4.2 Descripción de componentes</b>       | 5 |
| 1.4.2.1 helpers                               | 5 |
| 1.4.2.2 opciones                              | 5 |
| 1.4.2.3 Lectura de archivo                    | 5 |



## 1 Definición de la solución

---

### 1.1 Objetivos

---

Que el estudiante:

- Aplique los conceptos generales sobre lenguajes formales, tales como alfabeto, símbolos, cadenas y reglas.
- Conozca las características principales del lenguaje de programación Python.
- Aplique algoritmos de ordenamiento.

### 1.2 Alcances del proyecto

---

- Los entregables para este proyecto se adjunta en el repositorio de Git Hub con toda su documentación, este proyecto se sube a un repositorio privado, para que en un futuro los desarrolladores se le pueda dar mantenimiento al sistema, con toda su documentación.
- El sistema está diseñado para ser ejecutado en consola, utilizando archivos de entrada.
- El sistema permite realizar ciertas operaciones de interés, las cuales son:
  - Ordenar de forma ascendente
  - Ordenar de forma descendente
  - Calcular el promedio de notas
  - Obtener nota mínima
  - Obtener nota máxima
  - Obtener número de estudiantes aprobados
  - Obtener número de estudiantes reprobados
- La información es almacenada en cada ejecución en la memoria RAM.
- El sistema es capaz de generar reportes en HTML con todos los datos obtenidos.



## 1.3 Requerimientos del sistema

---

### 1.3.1 Requerimientos funcionales

---

- **Cargar archivo:** Muestra una ventana emergente que permite al usuario seleccionar un archivo LFP y carga el archivo a memoria, lo analiza y ejecuta el reporte.
- **Mostrar reporte en consola:** Muestra los datos obtenidos durante la ejecución del reporte. Se debe mostrar en pantalla el nombre del curso, el total de estudiantes que contiene el archivo y los datos propios del reporte que se solicite.
- **Exportar reporte:** Se debe de escribir un archivo HTML con los datos del reporte generado, la manera en que los datos son mostrados deben de ser agradables al usuario. Por lo menos, debe mostrar el nombre del curso, en una tabla la lista de estudiantes y su respectiva nota (en rojo las notas reprobadas y en azul las aprobadas) y al final los parámetros solicitados.

### 1.3.2 Descripción

---

Se le ha designado a usted estudiante de Ingeniería en Ciencias y Sistemas como colaborador en el área de reportes de notas en Control Académico en la Facultad de Ingeniería. La tarea que se le ha asignado es desarrollar una aplicación que permita la lectura de un archivo de texto plano con extensión lfp que contiene los datos de los estudiantes de un curso y la nota final que obtuvieron en dicho curso, finalmente el archivo contiene una palabra reservada que indica que tipo de reporte se quiere realizar.

### 1.3.3 Ejemplo de un archivo de entrada

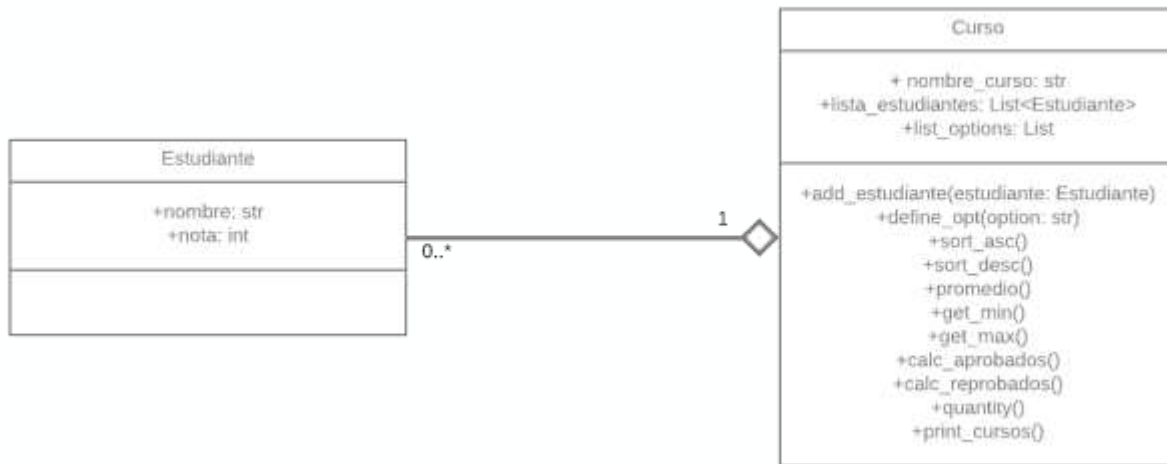
---

```
NOMBRE_DEL_CURSO = {  
    < "Estudiante 01" ; 65 >,  
    < "Estudiante 02" ; 89 >,  
    < "Estudiante 03" ; 36 >,  
    < "Estudiante 04" ; 65 >,  
    < "Estudiante 05" ; 99 >,  
    < "Estudiante 06" ; 71 >  
} PARAMETRO
```



## 1.4 Componentes

### 1.4.1 Diagrama de clases



### 1.4.2 Descripción de componentes

#### 1.4.2.1 helpers

Se utilizó un algoritmo de ordenamiento de burbuja (bubble sort), el cual crea una copia de la lista, sobre la cual itera, hasta lograr que el ordenamiento finalice, admite el parámetro **reverse** el cual le indica que debe ordenar en sentido inverso, es decir descendente, devuelve una lista como resultado sin alterar la lista original.

#### 1.4.2.2 opciones

El programa lee una serie de opciones, las cuales son almacenadas en una lista dentro de la clase **curso**, la cual es iterada en las operaciones de reporte, de manera que pueden aplicarse varios operadores al mismo conjunto de datos, todas las operaciones mantienen inmutable los datos, ya que no alteran el set de datos original.

#### 1.4.2.3 Lectura de archivo

El programa tiene la capacidad de leer el archivo de entrada, el cual está definido por una serie de reglas sintácticas, dentro del archivo existirá únicamente un curso, junto con su set de datos y las operaciones definidas a realizar, de la siguiente forma:



- 1 **NOMBRE DEL CURSO** es un identificador que representa el nombre del curso al cual pertenece el listado de estudiantes.  
A este identificador le prosigue un signo igual (=) y un par de llaves ({ }) dentro de las cuales se encuentra el listado de estudiantes del curso y su respectiva nota.
- 2 El listado de estudiantes se compone por elementos separados por coma (,). Estos elementos tienen la siguiente estructura:  
Inician con un corchete triangular (<). Les sigue una cadena de texto encerrada en comillas dobles ("). Seguidamente se coloca punto y coma (;) y luego la nota obtenida, dicha nota debe de ser un número entero positivo. Finalmente termina el elemento con un corchete triangular (>).
- 3 **PARAMETRO** puede venir uno o más de estos valores, separados por coma:  
**ASC** = Ordenar ascendentemente las notas de los estudiantes.  
**DESC** = Ordenar descendentemente las notas de los estudiantes.  
**AVG** = Obtener el promedio de los estudiantes del curso.  
**MIN** = Obtener la nota mínima de los estudiantes del curso.  
**MAX** = Obtener la nota máxima de los estudiantes del curso.  
**APR** = Obtener el número de estudiantes aprobados en el curso.  
**REP** = Obtener el número de estudiantes reprobados en el curso.