



---

# MANUAL TÉCNICO

---

Juan José Rodas Mansilla



202200389

LENGUAJES FORMALES Y DE PROGRAMACIÓN  
Universidad de San Carlos de Guatemala

## Explicación de Funciones:

### 1. Student.py

Llevará la lógica de la clase Estudiante. En donde tendrá un id (carnet), name (nombre), rating (notas).

La función `add_ratings` se encargará de almacenar las notas en relación a cada uno de los alumnos que se manejaran dentro del programa.

La función `calc_prom` llevará el promedio de todas las notas que pertenezcan a x alumno, sumando cada una de las notas registradas y luego siendo esta suma dividida entre la cantidad de notas del alumno en específico.

La función `approved` se encargará de verificar de si el promedio de cada uno de los alumnos es mayor a 61, esto ayudara a luego mostrar si el alumno tiene notas satisfactorias al generar el reporte de alumnos aprobados.

### 2. School.py

**Load\_students:** la función abrirá el documento seleccionado, este deberá de tener la extensión .est para que los datos sean cargados. Los datos entre estudiante y estudiante se verán separados por dos puntos (:), además de distinguir entre las comillas para cargar el nombre de cada alumno.

```
def load_students(self, nFile):
    with open(nFile, 'r') as f:
        for l in f:
            id, name = l.strip().split(':')
            self.students.append(Student(id, name.strip('"')))
    print('### Estudiantes añadidos ###')
```

**Add\_ratings:** dicha función obtendrá los datos de un documento con la extensión .cal e ira verificando entre cada una de las entradas si el carnet del estudiante en el documento exista, de ser así le agregara la nota de manera automática, si no encuentra al alumno simplemente descarta las notas y continuará con el siguiente alumno.

```
def add_ratings(self, nFile):
    with open(nFile, 'r') as f:
        for l in f:
            id, ratings = l.strip().split(':')
            ratings = list(map(int, ratings.split(',')))
            for student in self.students:
                if student.id == id:
                    student.add_ratings(ratings)
```

**Top3:** generará un archivo HTML luego de ordenar la lista de alumnos de forma ascendente con forme al promedio de cada promedio de los alumnos. Escribirá el documento HTML e ira creando la lista, únicamente escribirá como máximo a 3 alumnos los cuales serán los que mejor nota tendrán, si existen 2 o 1 alumno dentro del registro total de alumnos mostrara los que se hallen agregados. Al finalizar mostrara un mensaje que confirmara la creación de dicho archivo HTML.

```
def top3(self):
    sort = sorted(self.students, key = lambda x: x.calc_prom(), reverse=True)[:3]
    with open('top3.html', 'w') as file:
        file.write('<html><head><style>')

        file.write('p { margin-bottom: 15px }')
        file.write('</style></head><body><h1>Estudiantes con mejor promdeio</h1>')
        for est in sort:

            file.write(f'<p><strong>Carnet:</strong> {est.id}</p>')
            file.write(f'<p><strong>Nombre:</strong> {est.name}</p>')
            file.write(f'<p><strong>Promedio:</strong> {est.calc_prom()}</p><br>')

        file.write('</body></html>')
    print('Listado hecho!')
```

**Report:** crea un archivo HTML en donde se encontrarán todos los datos de cada uno de los alumnos, se irán mostrando en el orden en el que fueron agregados, por lo que el último alumno mostrado en la tabla del HTML será el ultimo alumno registrado en las listas del programa. Este archivo simplemente los mostrará, independientemente de si tendrá un promedio mayor o menor a cierta cantidad.

```
def report(self):
    with open('reporte.html', 'w') as file:
        file.write('<html><head><style>')
        file.write(' h1 { color:navy; }')
        file.write('p { margin-bottom: 15px }')
        file.write('</style></head><body><h1>Reporte de Estudiantes</h1>')

        for est in self.students:
            file.write(f'<p><strong>Carnet: </strong> {est.id}</p>')
            file.write(f'<p><strong>Nombre: </strong> {est.name}</p>')
            file.write(f'<p><strong>Carnet: </strong> {est.id}</p>')
            file.write('<p><strong>Calificaciones: </p>')
            file.write(', '.join(map(str, est.rating)))
            file.write('</p><br>')
        file.write('</body></html>')
    print('Listado hecho!')
```

**Approved:** mostrara en un archivo HTML los alumnos en color verde que tengan un promedio mayor a 61 y en color rojo a los que tengan un promedio menor a 61. Los mostrara de forma desordenada, por lo que la tabla se generara en el mismo orden que genera el reporte general de alumnos. Al finalizar mostrara un mensaje en consola el cual confirmara la creación de dicho archivo HTML.

```
def approved(self):
    with open('aprobados.html', 'w') as file:
        file.write('')
        file.write('<html><head><style>')
        file.write(' h1 { color:navy; }')
        file.write('p { margin-bottom: 15px }')
        file.write('.approved { color: green, font-weight: bold; }')
        file.write('.not { color: red, font-weight: bold; }')
        file.write('</style></head><body><h1>Reporte de Aprobados</h1>')

        for est in self.students:
            file.write(f'<p><strong>Carnet: </strong> {est.id} </p>')
            file.write(f'<p><strong>Nombre: </strong> {est.name} </p>')
            file.write(f'<p><strong>Carnet: </strong> {est.calc_prom()} </p>')
            if est.approved():
                file.write('<p class = "approved"> Aprobado</p><br>')
            else:
                file.write('<p class = "not"> Reprobado</p><br>')
        file.write('</body></html>')
    print('Reporte generado!')
```

### 3. Main.py

**Menú:** maneja de forma constante la elección de cada una de las funciones principales, luego de elegir una opción y de que esta se vea finalizada continuara en funcionamiento el programa para que la inserción de datos sea constante.

```
while True:
    print("")
    print("-----")
    print("#####          MENÚ PRINCIPAL          #####")
    print("#####          GOTITAS DEL SABER          #####")
    print("-----")

    print("-----")
    print("1. SALIR |")
    print("2. DATOS DEL ALUMNO |")
    print("3. CARGAR ALUMNOS |")
    print("4. AGREGAR CALIFICACIONES |")
    print("5. REPORTE ALUMNOS |")
    print("6. REPORTE APROBADOS |")
    print("7. TOP 3 ALUMNOS |")
    print("-----")

    print("")
    print("##      INGRESE UNA OPCION:      ##")
    opcion = input()
    print("")
```

Si se elige la opción 1 del menú se finalizará con el programa, ya que este funciona con un while la condición ya no se cumplirá mediante un Break. Luego mostrara un mensaje en donde confirme la detención del programa.

```
if opcion == '1':
    print('+++ Adios, esperamos volver a verte :) +++')
    break
```

Para la opción 2 únicamente se mostraran los datos del estudiante que manejo la lógica del programa.

```
def data():
    print("-----")
    print("#####          DATOS DEL ESTUDIANTE          #####")
    print("-----")
    print("-> JUAN JOSÉ RODAS MANSILLA |")
    print("-> 202200389 |")
    print("-> LFP Sección A- |")
    print("-> Ingenieria en Ciencias y Sistemas |")
```

Al ingresar la opción 3 o 4 se solicitará mediante la interfaz Tkinter el nombre del archivo que se cargará, en esta opción no se ingresará la extensión del documento, únicamente el nombre. Luego de que se escriba el nombre del documento se comenzara con la inserción de datos, independientemente de si se eligió ingresar alumnos o calificaciones. De preferencia se hará la inserción de alumnos primero para que no haya problemas con carnets no encontrados dentro del documento de calificaciones.

```
elif opcion == '3':  
    name = loadDocument('est')  
    if name:  
        program.load_students(name)  
  
elif opcion == '4':  
    name = loadDocument('cal')  
    if name:  
        program.add_ratings(name)
```

Luego con las opciones 5, 6 y 7 se llamarán a las funciones que crearán cada uno de los archivos HTML anteriormente explicados.

```
elif opcion == '5':  
    program.report()  
  
elif opcion == '6':  
    program.approved()  
  
elif opcion == '7':  
    program.top3()
```