



# MANUAL TÉCNICO

## PRÁCTICA 1

ALDO SAÚL VÁSQUEZ MOREIRA  
CARNET 202109754  
LAB. LENGUAJES FORMALES Y DE PROG.

## ENCABEZADO

Nombre: Aldo Saúl Vásquez Moreira

Carnet: 20109754

Laboratorio Lenguajes Formales y de Programación.

Nombre de Sistema: Gestor de Créditos Facultad de Ingeniería Universidad San Carlos de Guatemala

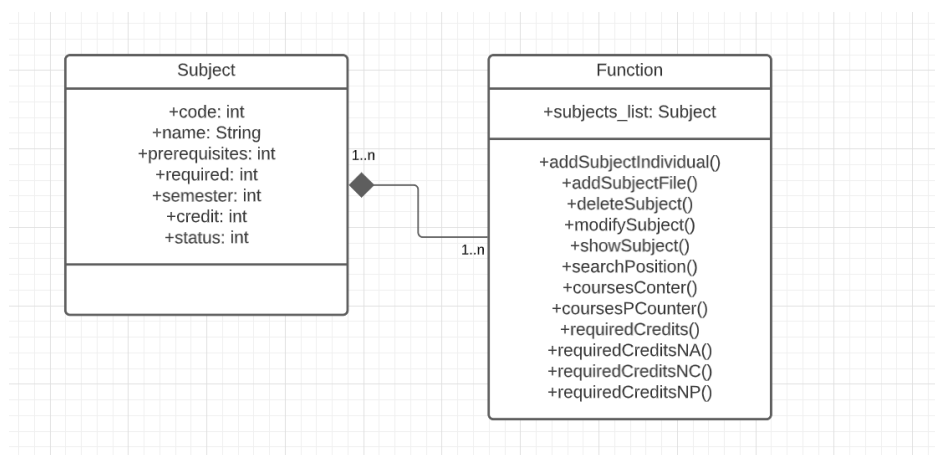
## PRINCIPIO, TÉCNICA O PARADIGMA APLICADO DE PROGRAMACIÓN

Se utilizó el paradigma de Programación Orientado a Objetos con Python.

## CONVENCIONES DE NOMENCLATURA

- Se declararon las clases con letra inicial mayúscula.
- Se declararon los métodos con letra inicial minúscula y aplicando la convención Camel Case.
- Se declararon las variables con letra inicial minúscula y con un guion bajo en caso fueran necesarias más de dos palabras.

## DIAGRAMA DE CLASES



## MÉTODOS PRINCIPALES

- **addSubjectIndividual:** Este método agrega cursos de manera individual al arreglo de cursos y valida que los datos estén correctos. Recibe como parámetros los datos del curso.

```
def addSubjectIndividual(code, name, prerequisites, required, semester, credit, status):
    longitude = len(Subject.subjects_list)
    answer = False
    for i in range(longitude):
        if Subject.subjects_list[i].code == code:
            answer = True
            break

    if answer == True:
        messagebox.showerror(message="El curso que desea agregar ya existe.", title="Operación no realizada.")
    elif (required != "0" and required != "1"):
        messagebox.showerror(message="Dato no aceptado en el campo 'Obligatorio'.", title="Operación no realizada.")
    elif (semester != "1" and semester != "2" and semester != "3" and semester != "4" and semester != "5" and semester != "6"):
        messagebox.showerror(message="Dato no aceptado en el campo 'Semestre'.", title="Operación no realizada.")
    elif (status != "1" and status != "0" and status != "-1"):
        messagebox.showerror(message="Dato no aceptado en el campo 'Estado'.", title="Operación no realizada.")
    else:
        newSubject = Subject(code, name, prerequisites, required, semester, credit, status)
        Subject.subjects_list.append(newSubject)
        messagebox.showinfo(message="Curso agregado correctamente.", title="Operación realizada con éxito")
        print(len(Subject.subjects_list))
```

- **addSubjectFile:** Este método agrega el archivo (únicamente csv) al sistema. Recibe como parámetros los datos del curso.

```
def addSubjectFile(code, name, prerequisites, required, semester, credit, status):
    added = False
    longitude = len(Subject.subjects_list)
    for i in range(longitude):
        if Subject.subjects_list[i].code == code:
            added = True
            break

    if (added == False):
        newSubject = Subject(code, name, prerequisites, required, semester, credit, status)
        Subject.subjects_list.append(newSubject)
        print(len(Subject.subjects_list))
```

- **deleteSubject:** Este método elimina el curso del arreglo de cursos, recibiendo como parámetro el código del curso a eliminar.

```
def deleteSubject(codeParameter):
    answer = False
    longitude = len(Subject.subjects_list)
    print(longitude)
    for i in range(longitude):
        if Subject.subjects_list[i].code == codeParameter:
            Subject.subjects_list.pop(i)
            answer = True
            break

    if answer == True:
        messagebox.showinfo(message="Curso eliminado correctamente.", title="Operación realizada con éxito")
    else:
        messagebox.showerror(message="Curso no encontrado.", title="Operación no realizada.")
```

- **modifySubject:** Este método edita el curso recibiendo como parámetros los nuevos datos del curso.

```
def modifySubject(position, code, name, prerequisites, required, semester, credit, status):
    longitude = len(Subject.subjects_list)
    answer = False

    if (required != "0" and required != "1"):
        messagebox.showerror(message="Dato no aceptado en el campo 'Obligatorio'.", title="Operación no realizada.")
    elif (semester != "1" and semester != "2" and semester != "3" and semester != "4" and semester != "5" and semester != "6"):
        messagebox.showerror(message="Dato no aceptado en el campo 'Semestre'.", title="Operación no realizada.")
    elif (status != "1" and status != "0" and status != "-1"):
        messagebox.showerror(message="Dato no aceptado en el campo 'Estado'.", title="Operación no realizada.")
    else:
        Subject.subjects_list[position].code = code
        Subject.subjects_list[position].name = name
        Subject.subjects_list[position].prerequisites = prerequisites
        Subject.subjects_list[position].required = required
        Subject.subjects_list[position].semester = semester
        Subject.subjects_list[position].credit = credit
        Subject.subjects_list[position].status = status
        messagebox.showinfo(message="Curso modificado correctamente.", title="Operación realizada con éxito")
```

- **showSubject:** Este método sirve para verificar si un curso existe entre en el arreglo de los cursos, recibiendo como parámetro el código del curso a buscar.

```
def showSubject(codeParameter):
    longitude = len(Subject.subjects_list)
    for i in range(longitude):
        if Subject.subjects_list[i].code == codeParameter:
            var = 1
            break
        else:
            var = 0

    if var == 1:
        messagebox.showinfo(message="Curso encontrado con éxito.", title="Operación realizada con éxito")
        return True
    else:
        messagebox.showerror(message="Curso no encontrado.", title="Operación no realizada")
        return False
```

- **searchPosition:** Este método se encarga de buscar la posición de un curso entre el arreglo de los cursos, recibiendo como parámetro el código del curso a buscar.

```
def searchPosition(codeParameter):
    longitude = len(Subject.subjects_list)
    for i in range(longitude):
        if Subject.subjects_list[i].code == codeParameter:
            return i
```

- **coursesCounter:** Este método muestra la sumatoria de los cursos con un estado determinado. Como parámetro recibe el estado de los cursos.

```
def coursesCounter(status):
    counter = 0
    longitude = len(Subject.subjects_list)
    for i in range(longitude):
        if Subject.subjects_list[i].status == status:
            counter += int(Subject.subjects_list[i].credit)
    return counter
```

- **requiredCredits:** Este método hace la sumatoria de los créditos de los cursos obligatorios de determinado semestre. Recibe como parámetro el número de semestre.

```
def requiredCredits(semester):
    counter = 0
    longitude = len(Subject.subjects_list)
    for i in range(longitude):
        if Subject.subjects_list[i].required == "1" and not Subject.subjects_list[i].semester > semester:
            counter += int(Subject.subjects_list[i].credit)
    return counter
```

- **requiredCreditsNA:** Este método hace la sumatoria de los créditos de los cursos aprobados de determinado semestre. Recibe como parámetro el número de semestre.

```
def requiredCreditsNA(semester):
    counter = 0
    longitude = len(Subject.subjects_list)
    for i in range(longitude):
        if Subject.subjects_list[i].semester == semester and Subject.subjects_list[i].status == "0":
            counter += int(Subject.subjects_list[i].credit)
    return counter
```

- **requiredCreditsNC:** Este método hace la sumatoria de los créditos de los cursos que están siendo cursados de determinado semestre. Recibe como parámetro el número de semestre.

```
def requiredCreditsNC(semester):
    counter = 0
    longitude = len(Subject.subjects_list)
    for i in range(longitude):
        if Subject.subjects_list[i].semester == semester and Subject.subjects_list[i].status == "1":
            counter += int(Subject.subjects_list[i].credit)
    return counter
```

- **requiredCreditsNP:** Este método hace la sumatoria de los créditos de los cursos pendientes de determinado semestre. Recibe como parámetro el número de semestre.

```
def requiredCreditsNP(semester):
    counter = 0
    longitude = len(Subject.subjects_list)
    for i in range(longitude):
        if Subject.subjects_list[i].semester == semester and Subject.subjects_list[i].status == "-1":
            counter += int(Subject.subjects_list[i].credit)
    return counter
```

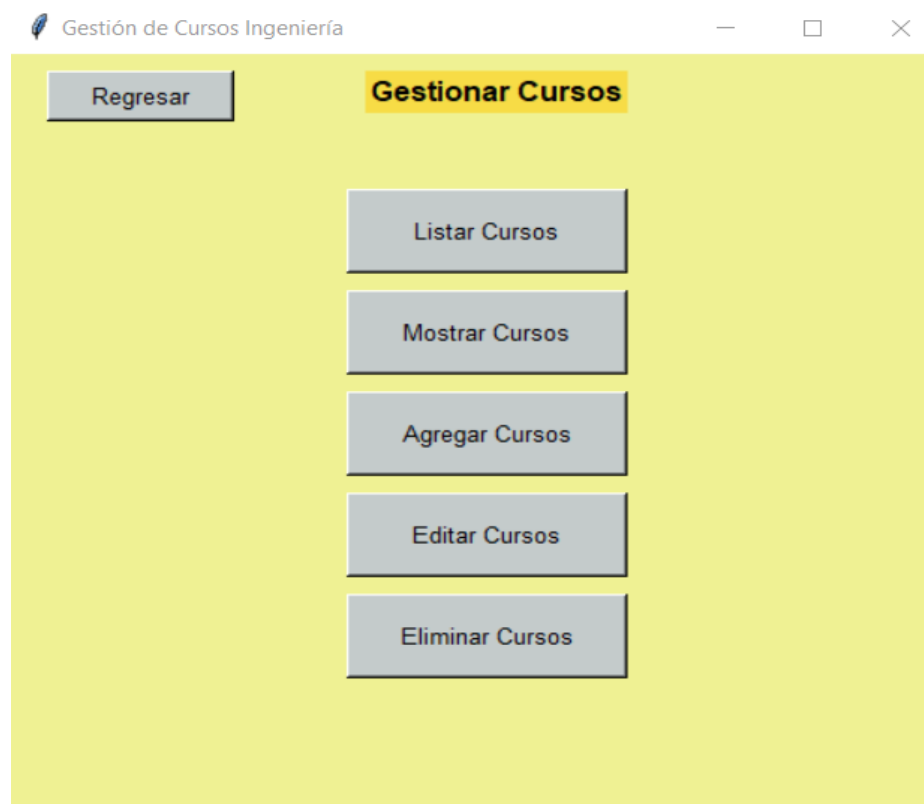
## DESCRIPCIÓN

Se utilizó el Paradigma de Programación Orientado a Objetos debido a la capacidad de abstracción que se tiene para modelar objetos de la vida real en el código. Asimismo, se cuenta con una clase que contiene todas las funciones para apartar del código principal todos estos procedimientos para facilitar y mejorar la visibilidad del código.

Adicionalmente, se utilizó la biblioteca Tkinter debido a que viene por defecto en las bibliotecas de Python y ofrece muchas opciones para crear una buena interfaz para los usuarios. También se utilizó la biblioteca csv de Python para el manejo exclusivo de archivos con dicha extensión.

## INTERFACES PRINCIPALES

- **Ventana Gestionar Cursos:**



Esta ventana contiene los botones que redirigen a cada una de las funcionalidades pertenecientes al apartado *Gestionar Cursos*.

- **Ventana Listar Cursos**

Código	Nombre	Pre	Obligatorio	Semestre	Créditos	Estado
114	Mate Intermedia 3	107	1	4	5	0
112	Mate Intermedia 2	107	1	4	5	0
010	Lógica	019	0	4	2	1
771	Introducción a la programación 2	107;770;795;960	1	4	5	0
962	Mate Computo 2	960;770;795	1	4	5	1
796	Lenguajes Formales	770;795;960	1	4	3	1
732	Estadística 1	107;005	1	4	5	0
0009	Idioma Técnico 3	0008	0	3	2	-1
150	Física 1	103;147	1	3	6	0
107	Mate Intermedia 1	103	1	3	10	0

Regresar

Acá se muestra de manera ordenada por medio de una tabla los cursos ingresados al sistema.

- **Ventana Mostrar Cursos**

Gestión de Cursos Ingeniería

Regresar
Buscar

Código

Nombre

Pre-requisitos

Obligatorio

Semestre

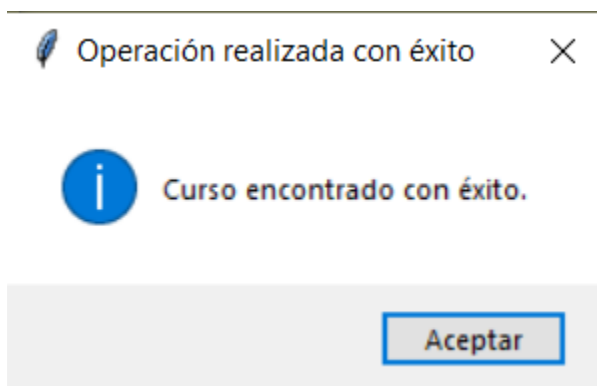
Créditos

Estado

Borrar




Esta ventana contiene en el tope una casilla en la cual se ingresa el código de curso que se quiere mostrar. Sí este se encuentra mostrará un mensaje como el que se muestra a continuación:




Y procederá a mostrar el detalle del curso de interés:

Una ventana de software titulada "Gestión de Cursos Ingeniería" con botones de minimizar, maximizar y cerrar. El formulario tiene un fondo amarillo claro. En la parte superior, hay un botón "Regresar" a la izquierda, un campo de texto con el valor "103" en el centro, y un botón "Buscar" a la derecha. Debajo, hay una lista de campos con etiquetas amarillas a la izquierda y campos de texto blancos a la derecha: "Código" (103), "Nombre" (Mate Basica 2), "Pre-requisitos" (101), "Obligatorio" (1), "Semestre" (2), "Créditos" (7), y "Estado" (0). En la parte inferior, hay un botón "Borrar".

En el caso de ingresar el código de un curso que no se encuentra en el sistema se mostrará una ventana emergente como la siguiente:


 Operación no realizada ×

 Curso no encontrado.

Aceptar

El botón borrar elimina los datos mostrados en las casillas para poder mostrar datos de otros cursos.

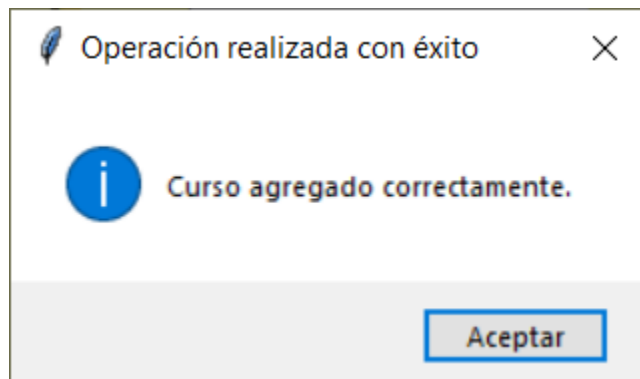
- **Ventana Agregar Curso:**

 Gestión de Cursos Ingeniería — □ ×

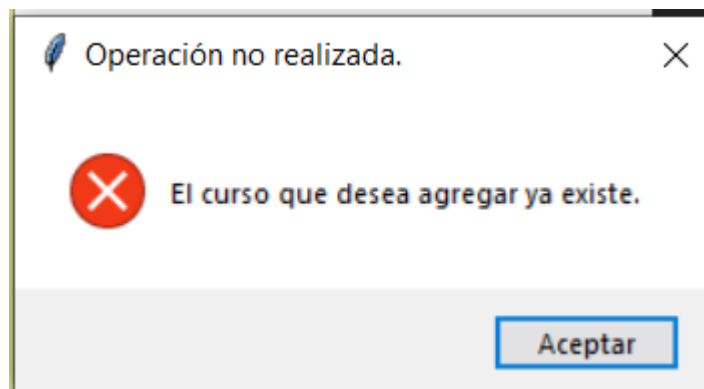
[Regresar](#) **Agregar Curso**

<b>Código</b>	<input type="text"/>
<b>Nombre</b>	<input type="text"/>
<b>Pre-requisitos</b>	<input type="text"/>
<b>Obligatorio</b>	<input type="text"/>
<b>Semestre</b>	<input type="text"/>
<b>Créditos</b>	<input type="text"/>
<b>Estado</b>	<input type="text"/>
<input type="button" value="Agregar Curso"/>	

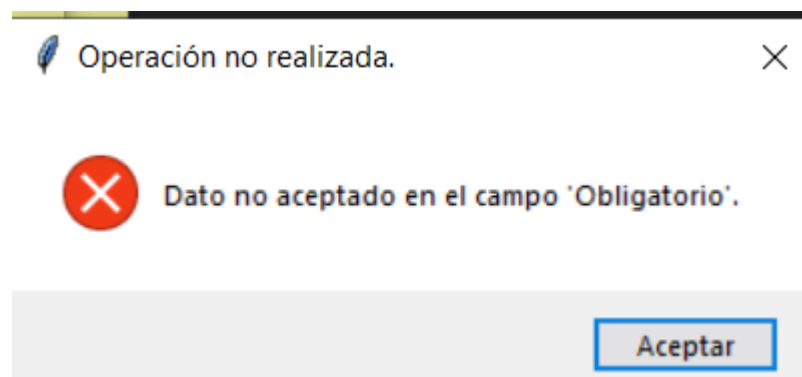
Esta ventana contiene las casillas para ingresar los datos respectivos al curso que se desea ingresar al sistema. Si esto se realiza de manera exitosa se mostrará un mensaje como el siguiente:



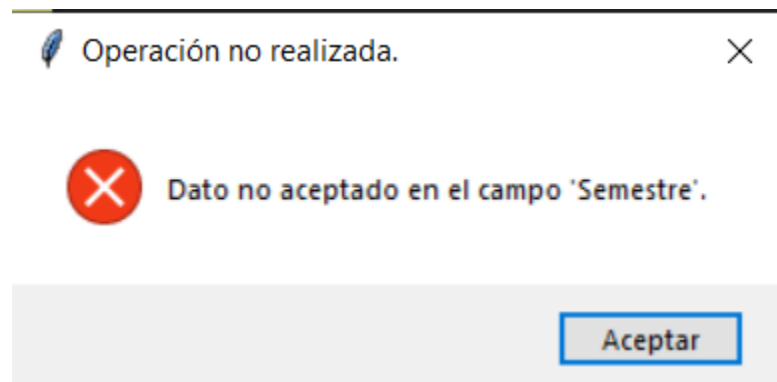
De no ser así, el sistema mostrará una casilla emergente con el error que ocurrió. Dichos errores están relacionados con las restricciones mencionadas en el apartado de *Requerimientos o Funcionalidades* descritos en este manual.



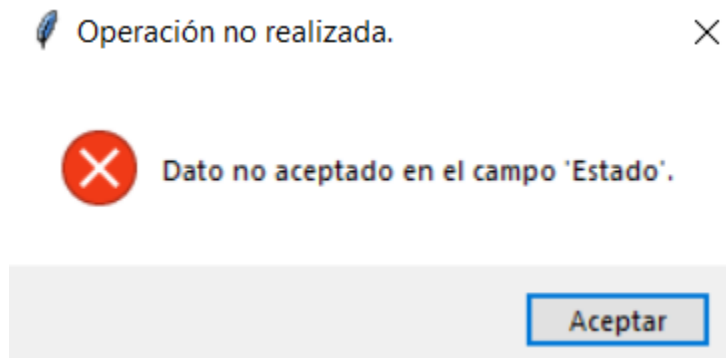
Si se desea ingresar un curso que ya existe.



Si en el campo Obligatorio se agregan datos diferentes a los especificados.



Si en el campo Semestre se agregan datos diferentes a los especificados.



Si en el campo Estado se agregan datos diferentes a los especificados

- **Ventana Editar Curso:**

The screenshot shows a window titled "Gestión de Cursos Ingeniería" with standard window controls (minimize, maximize, close). The window has a light yellow background. At the top left is a "Regresar" button. At the top right is a search bar with a "Buscar" button. Below these are seven input fields, each with a label to its left: "Código", "Nombre", "Pre-requisitos", "Obligatorio", "Semestre", "Créditos", and "Estado". The labels are in yellow boxes. At the bottom center is an "Editar" button.

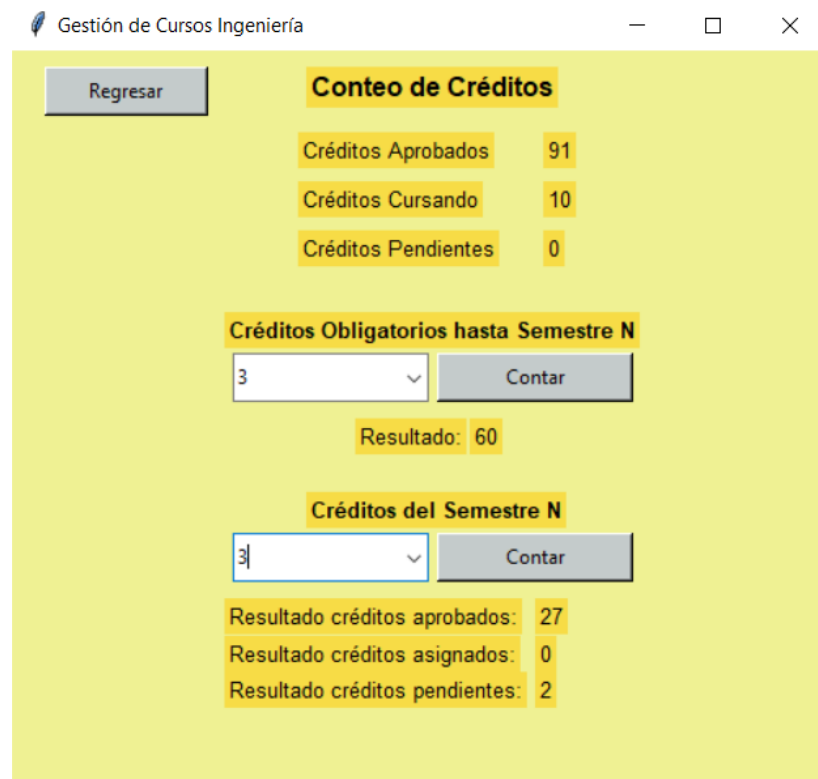
Contiene una casilla para ingresar el código del curso de interés. Si se encuentra se mostrará un mensaje de éxito y después de esto los datos del curso para que se puedan modificar. Si se modifica correctamente lo notificará, de caso contrario, desplegará una ventana con el error.

- **Ventana Eliminar Curso:**

The screenshot shows a window titled "Gestión de Cursos Ingeniería" with standard window controls (minimize, maximize, close). The window has a light yellow background. At the top left is a "Regresar" button. Below it is a large empty rectangular box for input. At the bottom center is a button labeled "Eliminar Curso".

Contiene una casilla para ingresar el código del curso a eliminar. Si esto se realiza correctamente lo notificará, de caso contrario indicará que el código del curso no existe dentro del sistema.

- **Ventana Contador de Créditos:**



The screenshot shows a web application window titled "Gestión de Cursos Ingeniería". It features a yellow background and a "Regresar" button in the top left. The main content is divided into three sections:

- Conteo de Créditos:** A summary section showing "Créditos Aprobados: 91", "Créditos Cursando: 10", and "Créditos Pendientes: 0".
- Créditos Obligatorios hasta Semestre N:** A section with a dropdown menu set to "3" and a "Contar" button. Below it, the "Resultado: 60" is displayed.
- Créditos del Semestre N:** A section with a dropdown menu set to "3" and a "Contar" button. Below it, three results are shown: "Resultado créditos aprobados: 27", "Resultado créditos asignados: 0", and "Resultado créditos pendientes: 2".

Muestra los datos de los créditos del semestre de interés.

## PLANIFICACIÓN O ESTIMACIÓN

- **Leer archivo CSV:** 15 minutos
- **Creación de clases:** 30 minutos
- **Creación de funciones para manipular los datos:** 2 días
- **Interfaz Gráfica:** 2 días

## **GLOSARIO**

1. Extensión de archivo: Las extensiones indican qué aplicación ha creado el archivo o puede abrirlo, y qué icono se debe utilizar para el archivo. Por ejemplo, la extensión docx indica al equipo que Microsoft Word puede abrir el archivo y que debe mostrar un icono de Word al verlo en el Explorador de archivos.
2. Explorador de archivos: El Explorador de archivos o Explorador de Windows, como fue nombrado hasta la edición de Windows 8, es el administrador de archivos oficial del sistema operativo Microsoft Windows.
3. Software: Conjunto de programas y rutinas que permiten a la computadora realizar determinadas tareas.
4. Sistema: En terminología informática el software de sistema, denominado también software de base, consiste en un software que sirve para controlar
5. Paradigma: El término paradigma es empleado para indicar un patrón, modelo, ejemplo o arquetipo. Por lo general hace referencia a una serie de teorías que son tomadas como modelo a seguir al momento de solucionar cualquier tipo de problemas que puedan surgir en determinadas situaciones.