Programación Orientada a Objetos

Grupo 3

Taller #5

Pablo Vanegas

Juan José Grajales Alvarez

Juan Pablo Ramirez Betancur

UNIVERSIDAD NACIONAL DE COLOMBIA

Febrero 11 de 2025

TALLER #6

```
import os
import ipywidgets as widgets
from IPython.display import display, clear_output

class ContactManager:
    def __init__(self, file_name="/content/friendsContact.txt"):
        self.file_name = file_name
        if not os.path.exists(self.file_name):
            open(self.file_name, 'w').close()

    def add_friend(self, name, number):
        with open(self.file_name, "r") as file:
            for line in file:
                existing_name, existing_number = line.strip().split("!")
                if existing_name == name or existing_number ==
number:
                    return "Error: Nombre o número ya existen."

        with open(self.file_name, "a") as file:
```

```
            file.write(f"{name}!{number}\n")
        return "Contacto agregado."

    def display_contacts(self):
        contacts = []
        with open(self.file_name, "r") as file:
            for line in file:
                if line.strip():
                    try:
                        name, number = line.strip().split("!")
                        contacts.append(f"Nombre: {name}, Número: {number}")
                    except ValueError:
                        continue
        return "\n".join(contacts) if contacts else "No hay contactos guardados."

    def update_contact(self, name, new_number):
        found = False
        temp_file = "temp.txt"
        with open(self.file_name, "r") as file, open(temp_file, "w") as temp:
```

```
        for line in file:
            try:
                existing_name, existing_number = line.strip().split("!")
                if existing_name == name:
                    temp.write(f"{name}!{new_number}\n")
                    found = True
                else:
                    temp.write(line)
            except ValueError:
                continue

    if found:
        os.replace(temp_file, self.file_name)
        return "Contacto actualizado."
    else:
        os.remove(temp_file)
        return "Error: Contacto no encontrado."

def delete_contact(self, name):
    found = False
    temp_file = "temp.txt"
```

```
        with open(self.file_name, "r") as file, open(temp_file, "w") as temp:
            for line in file:
                existing_name, _ = line.strip().split("!")
                if existing_name == name:
                    found = True
                    continue
                temp.write(line)

        if found:
            os.replace(temp_file, self.file_name)
            return "Contacto eliminado."
        else:
            os.remove(temp_file)
            return "Error: Contacto no encontrado."

# Crear instancia del manejador de contactos
manager = ContactManager()

# Crear widgets de la interfaz gráfica
name_input = widgets.Text(placeholder="Nombre")
```

```
number_input = widgets.Text(placeholder="Número")
output = widgets.Output()

create_button = widgets.Button(description="Crear")
read_button = widgets.Button(description="Mostrar")
update_button = widgets.Button(description="Actualizar")
delete_button = widgets.Button(description="Eliminar")
clear_button = widgets.Button(description="Limpiar")

def create_contact(b):
    with output:
        clear_output()
        print(manager.add_friend(name_input.value,
number_input.value))

def read_contacts(b):
    with output:
        clear_output()
        print(manager.display_contacts())

def update_contact(b):
```

```
    with output:
        clear_output()
        print(manager.update_contact(name_input.value,
number_input.value))

def delete_contact(b):
    with output:
        clear_output()
        print(manager.delete_contact(name_input.value))

def clear_fields(b):
    name_input.value = ""
    number_input.value = ""
    with output:
        clear_output()

# Asignar funciones a los botones
create_button.on_click(create_contact)
read_button.on_click(read_contacts)
update_button.on_click(update_contact)
delete_button.on_click(delete_contact)
```

```
clear_button.on_click(clear_fields)


display(widgets.VBox([
    name_input,
    number_input,
    widgets.HBox([create_button, read_button, update_button, delete_button, clear_button]),
    output
]))
```

Interfaz Grafica

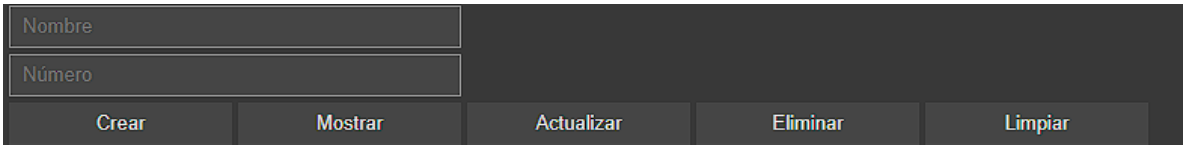

Diagrama de Flujo

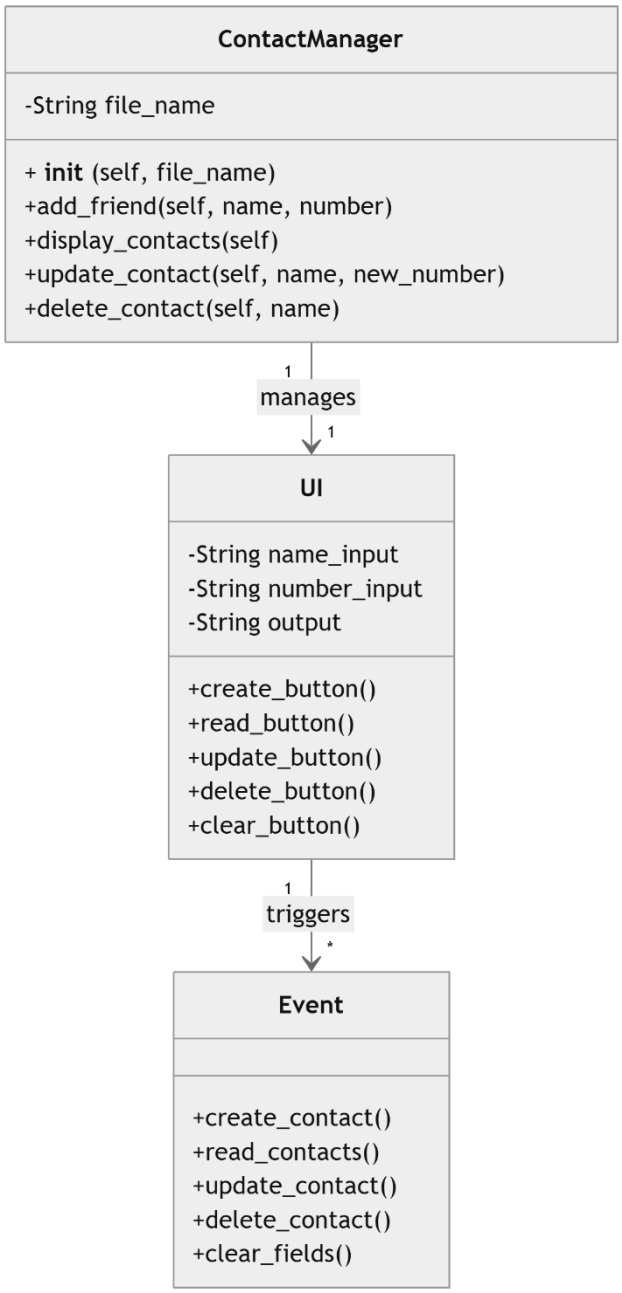

Diagrama de Casos de Uso