

# Practica 4

```
# Practica 4. Herencia
# 1. Crear una clase Ticket con los siguientes atributos:
# id
# tipo(eje: software, prueba)
# prioridad (alta, media,baja)
# estado (por defecto "pendiente")
# Crear dos ticket de ejemplo y mostrarlos por pantalla
import os
os.system("cls")

class ticket:
    def __init__(self, id, tipo, prioridad, estado="pendiente"):
        self.id = id
        self.tipo = tipo
        self.prioridad = prioridad
        self.estado = estado

# 2. Clase empleado
class empleado:
    def __init__(self, nombre):
        self.nombre = nombre
    def trabajar_en_ticket(self, ticket):
        print(f"El empleado {self.nombre} revisa el ticket {ticket.id}")

class desarrollador(empleado):
    def trabajar_en_ticket(self, ticket):
        if ticket.tipo == "software":
            ticket.estado = "resuelto"
            print(f"El Ticket {ticket.id} fue resuelto por {self.nombre}")
        else:
            print("Este tipo de ticket {ticket.tipo} no se puede resolver por este usuario")
```

```

class tester(empleado):
    def trabajar_en_ticket(self, ticket):
        if ticket.tipo == "prueba":
            ticket.estado = "resuelto"
            print(f"El Ticket {ticket.id} fue resuelto por {self.nombre}")
        else:
            print("Este tipo de ticket {ticket.tipo} no se puede resolver por este usuario")

class projecManager(empleado):
    def asignar_ticket(self, ticket, empleado):
        print(f"{self.nombre} asigno el ticket {ticket.id} al empleado {empleado.nombre}")
        empleado.trabajar_en_ticket(ticket)

# Crear tickets y empleados (Instancia de objetos)
ticket1 = ticket(1, "software", "alta")
ticket2 = ticket(2, "prueba", "baja")

developer1 = desarrollador("Carlitos")
tester1 = tester("Julio")
pm = projecManager("Marianita")

pm.asignar_ticket(ticket2, developer1)
pm.asignar_ticket(ticket1, tester1)

# Agregar un menu interactivo en la consola con while y con if para:
# 1. Crear un Ticket
# 2. Ver los Ticket
# 3. Asignar un Ticket
# 4. Salir del Programa

os.system("cls")
tickets = []
contador_id = 1
while True:
    print("-" * 40)
    print("\n\t MENÚ PRINCIPAL \n\t 1. Crear un Ticket \n\t 2. Ver los Tickets \n\t 3. Asignar un Ticket")
    print("-" * 40)
    opcion = input("Selecciona una opcion:")

    if opcion == "1":
        tipo = input("Ingrese el tipo de ticket (software/prueba): ").lower()
        prioridad = input("Ingrese la prioridad (alta/media/baja): ").lower()


```

```

ticket_nuevo = ticket(contador_id, tipo, prioridad)
tickets.append(ticket_nuevo)
print(f" Ticket creado con ID {contador_id}")
contador_id += 1

elif opcion == "2":
    if not ticket:
        print(" No hay tickets registrados.")
    else:
        print("\n--- LISTA DE TICKETS ---")
        for t in tickets:
            print(t)

elif opcion == "3":
    if not ticket:
        print(" No hay tickets para asignar.")
    else:
        try:
            id_ticket = int(input("Ingrese el ID del ticket a asignar: "))
            ticket_asignar = next((t for t in tickets if t.id == id_ticket), None)

            if ticket_asignar:
                print("Seleccione empleado:")
                print("1. Desarrollador")
                print("2. Tester")
                empleado_op = input("Opción: ")

                if empleado_op == "1":
                    pm.asignar_ticket(ticket_asignar, developer1)
                elif empleado_op == "2":
                    pm.asignar_ticket(ticket_asignar, tester1)
                else:
                    print(" Opción de empleado inválida")
            else:
                print(" Ticket no encontrado.")

        except ValueError:
            print(" ID inválido.")

elif opcion == "4":
    print(" Saliendo del programa...")
    break
else:

```

```
print(" Opción inválida. Intente nuevamente.")  
# Diagrama de Secuencia
```