

```

# Practica 4 Herencia

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

    def __str__(self):
        return f"Ticket {self.ID} | Tipo: {self.tipo} | Prioridad: {self.prioridad} | Estado: {self.estado}"

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 ticket no puede ser resuelto por un desarrollador.")

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 ticket no puede ser resuelto por un tester.")

class projectmanager(empleado):
    def asignar_ticket(self, ticket, empleado):
        print(f"{self.nombre} asignó el ticket {ticket.ID} al empleado {empleado.nombre}")
        empleado.trabajar_en_ticket(ticket)

# instancias
tickets = []
developer1 = desarrollador("Carlitos")
tester1 = tester("Juanillo")
pm = projectmanager("Marianita")

# menu
while True:
    os.system("cls" if os.name == "nt" else "clear") # limpia pantalla
    print("--- MENÚ ---")

```

```

print("1. Crear un ticket")
print("2. Ver tickets")
print("3. Asignar un ticket")
print("4. Salir")
opcion = input("Elige una opción: ")

if opcion == "1":
    os.system("cls" if os.name == "nt" else "clear")
    id_ticket = len(tickets) + 1
    tipo = input("Tipo de ticket (software/prueba): ").lower()
    prioridad = input("Prioridad (alta/media/baja): ").lower()
    nuevo_ticket = ticket(id_ticket, tipo, prioridad)
    tickets.append(nuevo_ticket)
    print("Ticket creado con éxito.")
    input("Presiona Enter para continuar...")

elif opcion == "2":
    os.system("cls" if os.name == "nt" else "clear")
    if not tickets:
        print("No hay tickets creados.")
    else:
        for t in tickets:
            print(t)
        input("Presiona Enter para continuar...")

elif opcion == "3":
    os.system("cls" if os.name == "nt" else "clear")
    if not tickets:
        print("No hay tickets creados.")
        input("Presiona Enter para continuar...")
        continue
    for t in tickets:
        print(t)
    tid = int(input("Ingresa el ID del ticket a asignar: "))
    seleccionado = next((tk for tk in tickets if tk.ID == tid), None)
    if not seleccionado:
        print("Ticket no encontrado.")
        input("Presiona Enter para continuar...")
        continue
    print("1. Desarrollador\n2. Tester")
    emp = input("Asignar a: ")
    if emp == "1":
        pm.asignar_ticket(seleccionado, developer1)
    elif emp == "2":
        pm.asignar_ticket(seleccionado, tester1)
    else:
        print("Opción inválida.")
        input("Presiona Enter para continuar...")

elif opcion == "4":
    os.system("cls" if os.name == "nt" else "clear")
    print("Saliendo del programa...")
    break

else:
    print("Opción no válida.")
    input("Presiona Enter para continuar...")

```

```
# ejecucion del codigo
--- MENÚ ---
1. Crear un ticket
2. Ver tickets
3. Asignar un ticket
4. Salir
Elige una opción:
```