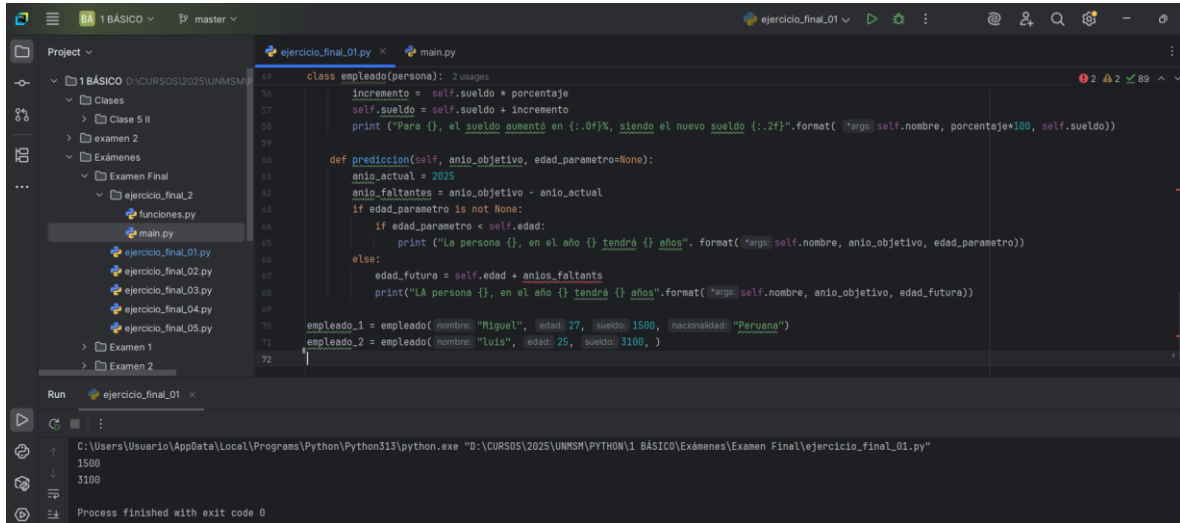


EXAMEN FINAL

Ejercicio 1



```
class Empleado(persona):
    def __init__(self, nombre, edad, sueldo, nacionalidad):
        super().__init__(nombre, edad)
        self.sueldo = sueldo
        self.nacionalidad = nacionalidad

    def calcular_incremento(self, porcentaje):
        incremento = self.sueldo * porcentaje
        self.sueldo = self.sueldo + incremento
        print(f"Para {self.nombre}, el sueldo aumentó en {incremento:.0f}%, siendo el nuevo sueldo {self.sueldo:.2f}")

    def prediccion(self, anio_objetivo, edad_parametro=None):
        anio_actual = 2025
        anio_faltantes = anio_objetivo - anio_actual
        if edad_parametro is not None:
            if edad_parametro < self.edad:
                print(f"La persona {self.nombre}, en el año {anio_objetivo} tendrá {anio_faltantes} años")
            else:
                edad_futura = self.edad + anio_faltantes
                print(f"La persona {self.nombre}, en el año {anio_objetivo} tendrá {edad_futura} años")
        else:
            print(f"La persona {self.nombre}, en el año {anio_objetivo} tendrá {anio_faltantes} años")

# Crear objetos
empleado_1 = Empleado(nombre="Riguel", edad=27, sueldo=1500, nacionalidad="Peruana")
empleado_2 = Empleado(nombre="Luis", edad=25, sueldo=3100, nacionalidad="Argentina")

# Ejecutar prediccion
empleado_1.calcular_incremento(10)
empleado_1.prediccion(2030)
empleado_2.calcular_incremento(10)
empleado_2.prediccion(2030)
```

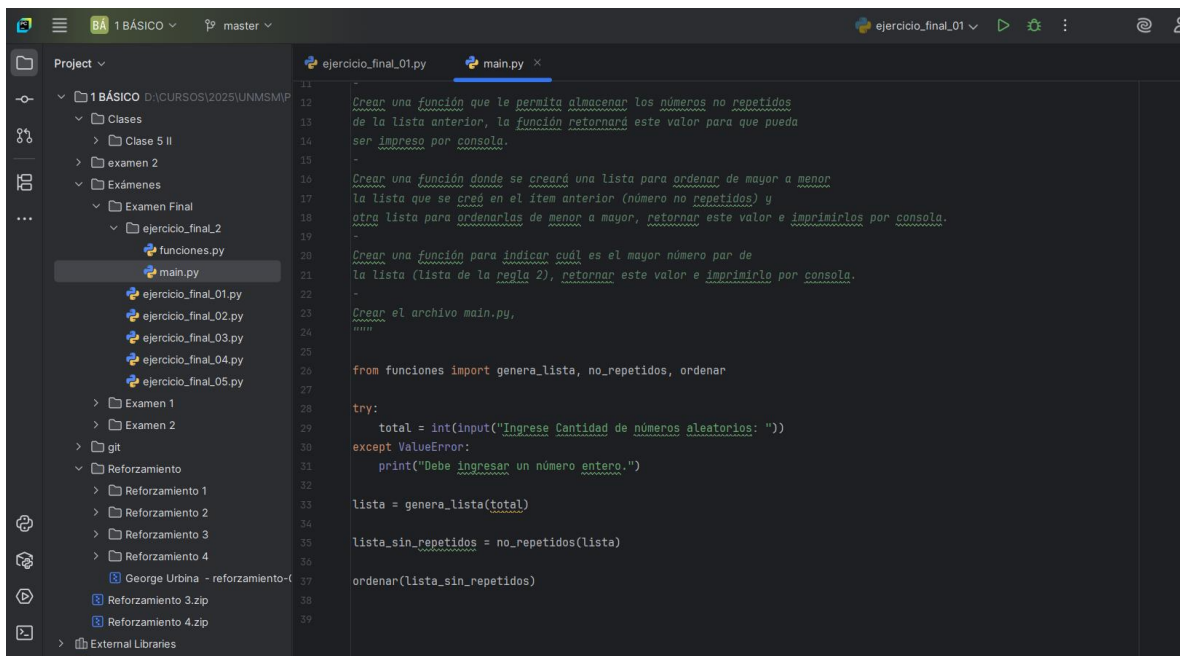
Run: ejercicio_final_01

C:\Users\User\AppData\Local\Programs\Python\Python313\python.exe "D:\CURSOS\2025\UNMSM\PYTHON\1 BÁSICO\Exámenes\Examen Final\ejercicio_final_01.py"

1500
3100

Process finished with exit code 0

Ejercicio 2



```
def genera_lista(tamaño):
    """Crear una función que le permita almacenar los números no repetidos de la lista anterior, la función retornará este valor para que pueda ser impreso por consola."""
    lista = []
    while len(lista) < tamaño:
        numero = random.randint(1, 100)
        if numero not in lista:
            lista.append(numero)
    return lista

def no_repetidos(lista):
    """Crear una función donde se creará una lista para ordenar de mayor a menor la lista que se creó en el ítem anterior (número no repetidos) y otra lista para ordenarlas de menor a mayor, retornar este valor e imprimirlos por consola."""
    lista_ordenada_desc = sorted(lista, reverse=True)
    lista_ordenada_asc = sorted(lista)
    return lista_ordenada_desc, lista_ordenada_asc

def mayor_numero_par(lista):
    """Crear una función para indicar cuál es el mayor número par de la lista (lista de la regla 2), retornar este valor e imprimirlo por consola."""
    numeros_pares = [numero for numero in lista if numero % 2 == 0]
    return max(numeros_pares) if numeros_pares else None

# Crear el archivo main.py
from funciones import genera_lista, no_repetidos, ordenar

try:
    total = int(input("Ingrese Cantidad de números aleatorios: "))
except ValueError:
    print("Debe ingresar un número entero.")

lista = genera_lista(total)
lista_sin_repetidos = no_repetidos(lista)
ordenar(lista_sin_repetidos)
```