

1.

```
class Persona:
    def __init__(self, nombre="", edad=0, nacionalidad="peruana", saldo=0):
        self.nombre = nombre
        self.edad = edad
        self.nacionalidad = nacionalidad
        self.saldo = saldo

    def actualizar_nombre(self):
        self.nombre = input("Actualice su nombre: ")

    def actualizar_edad(self):
        while True:
            try:
                self.edad = int(input("Actualice su edad: "))
                if self.edad <= 0:
                    raise ValueError
            except ValueError:
                print("El edad no es valida")
            else:
                print(f"Su edad es: {self.edad}")
                break

    def cumplir_años(self):
        self.edad = self.edad + 1
        print(f"Su edad actualizada es: {self.edad}")

    def mostrar_saldo(self, saldo):
        self.saldo = int(input("Ingrese su saldo: "))
        print(f"Su saldo es: {self.saldo}")

    def transferir(self, destino):
```

```
Actualice su nombre: Sebastian
-----
Actualice su edad: 19
Su edad es: 19
-----
Su edad actualizada es: 20
-----
Elija el año de la predicción: 30
En el año 2035 tendra 30
-----

////Persona_2:
Actualice su nombre: Adrian
-----
Actualice su edad: 19
Su edad es: 19
-----
Su edad actualizada es: 20
-----
Elija el año de la predicción: 35
En el año 2040 tendra 35

////Transferencias:
Transferir la cantidad de: 240
Saldo insuficiente
```

2.

```
import random

def almacenar_numeros(): 3 usages
    while True:
        try:
            valor = int(input("indica los numeros aleatorios que quieres generar: "))
            lista = [random.randint(a: 1, b: 50) for _ in range(valor)]
            print(f"Tu lista es: {lista}")
            return lista
        except ValueError:
            print("Valor no valido")

def no_repetidos(lista_1): 3 usages
    no_repetidos = list(set(lista_1))
    print(f"La lista sin repetir es: {no_repetidos}")
    return no_repetidos

def orden_listas(lista): 3 usages
    mayor_menor = sorted(lista, reverse=True)
    menor_mayor = sorted(lista)
    print(f"Numeros de mayor a menor: {mayor_menor}")
    print(f"Numeros de menor a mayor: {menor_mayor}")
    return mayor_menor, menor_mayor

def mayor_par(lista): 3 usages
    num_par = [num for num in lista if num % 2 == 0]
    if num_par:
        mayor = max(num_par)
        print(f"El mayor numero par es: {mayor}")
        return mayor
    else:
        print("No hay números pares")
```

```
from modulo import almacenar_numeros, no_repetidos, orden_listas, mayor_par

lista1 = almacenar_numeros()
no_repetidos1 = no_repetidos(lista1)
orden1_mayor, orden1_menor = orden_listas(no_repetidos1)
mayor_par1 = mayor_par(no_repetidos1)

print("\n-----")

lista2 = almacenar_numeros()
no_repetidos2 = no_repetidos(lista2)
orden2_mayor, orden2_menor = orden_listas(no_repetidos2)
mayor_par2 = mayor_par(no_repetidos2)
```

```
D:\virtualenvs\env-examen\Scripts\python.exe D:\Python\Exámenes\Ex
indica los numeros aleatorios que quieres generar: 7
Tu lista es: [38, 19, 48, 12, 17, 5, 25]
La lista sin repetir es: [5, 38, 12, 48, 17, 19, 25]
Numeros de mayor a menor: [48, 38, 25, 19, 17, 12, 5]
Numeros de menor a mayor: [5, 12, 17, 19, 25, 38, 48]
El mayor numero par es: 48

-----

indica los numeros aleatorios que quieres generar: 5
Tu lista es: [9, 10, 22, 26, 33]
La lista sin repetir es: [33, 9, 10, 22, 26]
Numeros de mayor a menor: [33, 26, 22, 10, 9]
Numeros de menor a mayor: [9, 10, 22, 26, 33]
El mayor numero par es: 26
```

5. Repositorio:

The screenshot shows the GitHub interface for the repository 'sebastian-segovia-examenes', which is marked as 'Public'. At the top right, there are buttons for 'Pin' and 'Watch'. Below the repository name, the current branch is 'main', with '1 Branch' and '0 Tags' indicated. A search bar labeled 'Go to file' is present, along with 'Add file' and 'Code' buttons. The commit history shows a recent update by 'Sebastian Segovia' with the message 'Actualización: Examen final', commit hash '437e8f5', and '2 Commits' made '1 minute ago'. A list of files follows, including '.idea', '1ra_practica', '2da_practica', and 'Examen_final', all with update messages and timestamps. At the bottom, a 'README' file is listed with a red underline.

sebastian-segovia-examenes Public

main 1 Branch 0 Tags

Go to file Add file <> Code

Sebastian Segovia and Sebastian Segovia Actualización: Examen final 437e8f5 · 1 minute ago 2 Commits

.idea	Actualización examen	last week
1ra_practica	Actualización examen	last week
2da_practica	Actualización: Examen final	1 minute ago
Examen_final	Actualización: Examen final	1 minute ago

README