

Problema 1:

```
NOTA_APROBATORIA = 11

def procesar_notas(estudiantes): 1 usage
    resultado = {}

    for nombre, notas in estudiantes.items():
        if len(notas) < 3:
            raise ValueError("El estudiante {} debe tener al menos 3 notas".format(nombre))

        promedio = sum(notas) / len(notas)
        estado = "aprobado" if promedio >= NOTA_APROBATORIA else "desaprobado"

        resultado[nombre] = {"promedio": promedio, "estado": estado}

    mejor_estudiante = None
    mejor_promedio = -1

    for nombre, datos in resultado.items():
        if datos["promedio"] > mejor_promedio:
            mejor_promedio = datos["promedio"]
            mejor_estudiante = nombre

    print("El estudiante con mayor promedio es {} con {:.2f}".format(*args: mejor_estudiante, resultado[mejor_estudiante]['promedio']))

    return resultado

estudiantes = {"Bruno": [15, 17, 16], "Fernando": [9, 10, 8], "Bryan": [18, 17, 19], "Jose": [11, 12, 10]}

resultado = procesar_notas(estudiantes)

print("Resultados:")
for estudiante, datos in resultado.items():
    print("{}: Promedio = {:.2f}, Estado = {}".format(*args: estudiante, datos['promedio'], datos['estado']))
```

Resultado 1:

```
El estudiante con mayor promedio es Bryan con 18.00
Resultados:
Bruno: Promedio = 16.00, Estado = aprobado
Fernando: Promedio = 9.00, Estado = desaprobado
Bryan: Promedio = 18.00, Estado = aprobado
Jose: Promedio = 11.00, Estado = aprobado
```

Problema 2:

```
def normalizar_nombres(nombres): 1 usage

    if len(nombres) < 6:
        raise ValueError("La lista debe contener al menos 6 nombres")

    resultado = []
    vistos = set()

    for nombre in nombres:
        partes = nombre.strip().title().split()
        for parte in partes:
            if parte not in vistos:
                resultado.append(parte)
                vistos.add(parte)

    return resultado

nombres = [" bruno ", "renzo", "María rosa", "MABEL", "nicolas", " bobbi", "jose antonio"]

try:
    normalizados = normalizar_nombres(nombres)
    print("Lista original: {}".format(nombres))
    print("Lista normalizada: {}".format(normalizados))
except ValueError as e:
    print("Error:", e)
```

Resultado 2:

```
Lista original: [' bruno ', 'renzo', 'María rosa', 'MABEL', 'nicolas', ' bobbi', 'jose antonio']
Lista normalizada: ['Bruno', 'Renzo', 'María', 'Rosa', 'Mabel', 'Nicolas', 'Bobbi', 'Jose', 'Antonio']

Process finished with exit code 0
```

Problema 3:

```
def convertir_precio(texto: str) -> float: 1 usage

    if not texto.strip():
        raise ValueError("El precio no puede estar vacio")

    try:
        precio = float(texto)
    except ValueError:
        raise ValueError("Formato de precio inválido")

    if precio < 0:
        raise ValueError("El precio no puede ser negativo")

    return precio

precios = []

for i in range(1, 4):
    entrada = input("Ingresar precio {}: ".format(i))
    try:
        valor = convertir_precio(entrada)
        precios.append(valor)
    except ValueError as e:
        print("Error: {}".format(e))


if precios:
    promedio = sum(precios) / len(precios)
    print("Precios válidos ingresados: {}".format(precios))
    print("Promedio: {:.2f}".format(promedio))
else:
    print("No hay precios válidos")
```

Resultado 3:



```
Ingresar precio 1: 100.69
Ingresar precio 2: -45
Error: El precio no puede ser negativo
Ingresar precio 3: abc
Error: Formato de precio inválido
Precios válidos ingresados: [100.69]
Promedio: 100.69


Process finished with exit code 0
|
```


Problema 4:

 BrunoMDH Add files via upload





c4ee8c0 · 1 minute ago 1 Commit


 1era práctica	Add files via upload	1 minute ago
 2da práctica	Add files via upload	1 minute ago


bruno-diaz-examenes / 1era práctica / 

 BrunoMDH Add files via upload





c4ee8c0 · 2 minutes ago History

Name	Last commit message	Last commit date
 ..		
 ejercicio1_bdh.py	Add files via upload	2 minutes ago
 ejercicio2_bdh.py	Add files via upload	2 minutes ago
 ejercicio3_bdh.py	Add files via upload	2 minutes ago

bruno-diaz-examenes / 2da práctica / 

 BrunoMDH Add files via upload

c4ee8c0 · 2 minutes ago History

Name	Last commit message	Last commit date
 ..		
 ejercicio1_bdh.py	Add files via upload	2 minutes ago
 ejercicio2_bdh.py	Add files via upload	2 minutes ago
 ejercicio3_bdh.py	Add files via upload	2 minutes ago