



PRÁCTICO 1

Jorge Luis Esteves Salas

Fernando Navia

Joel Dalton Montero

Ana Laura Cuellar

Weimar Valda

Leonel Eguez Camargo

Alejandro Hurtado Rodas

Dirigido por el docente:

JIMMY REQUENA LLORENTTY

Materia:

Programación II

MIS PRIMERAS PRUEBAS DE REPLIT

Después de seguir los pasos del texto guía en la plataforma pude configurar mi Replit con mi repositorio tuve unas complicaciones y todavía no puedo configurar el token de acceso correctamente pero la sincronización de archivos entre funciona bien:

```
~/workspace: git pull

~/workspace/pruebasProg2$ git commit -m "prueba de subida 2"
On branch main
Your branch is ahead of 'origin/main' by 5 commits.
  (use "git push" to publish your local commits)

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   ../.replit
    deleted:    ../PruebasProg2/clese02_prueba.py

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    clese02_prueba.py

no changes added to commit (use "git add" and/or "git commit -a")
~/workspace/pruebasProg2$ git push origin main
Enumerating objects: 18, done.
Counting objects: 100% (18/18), done.
Delta compression using up to 8 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (15/15), 1.76 KiB | 1.76 MiB/s, done.
Total 15 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), done.
To https://github.com/Alejandro26/prog2-alejandrohurtado
   2dab68a..d6c3f11  main -> main
~/workspace/pruebasProg2$ ls
clese02_prueba.py  pruebasubida.md
~/workspace/pruebasProg2$ cd..
bash: cd..: command not found
~/workspace/pruebasProg2$ cd.
```

Añadi y elimine archivos tanto en el repositorio como en el replit y funciona bien :

```
~/workspace: git pull

~/workspace$ git add linkUtiles.md
~/workspace$ git commit -m "update del archivo"
[main c161965] update del archivo
 1 file changed, 5 insertions(+)
 create mode 100644 linkUtiles.md
~/workspace$ git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 446 bytes | 446.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Alejandro26/prog2-alejandrohurtado
   d6c3f11..c161965  main -> main
~/workspace$ git pull
Already up to date.
~/workspace$ git pull
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 2 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (2/2), 874 bytes | 874.00 KiB/s, done.
From https://github.com/Alejandro26/prog2-alejandrohurtado
   c161965..39836c0  main -> origin/main
Updating c161965..39836c0
Fast-forward
 Links Utiles | 3 ---
 1 file changed, 3 deletions(-)
 delete mode 100644 Links Utiles
~/workspace$
```

APRENDIENDO A LLAMAR FUNCIONES

La clase de centro en aprender a usar las funciones y llamarlas para poder optimizar el código use de base los ejercicios que nos dio de tarea para modificarlos y crear funciones que posteriormente llame

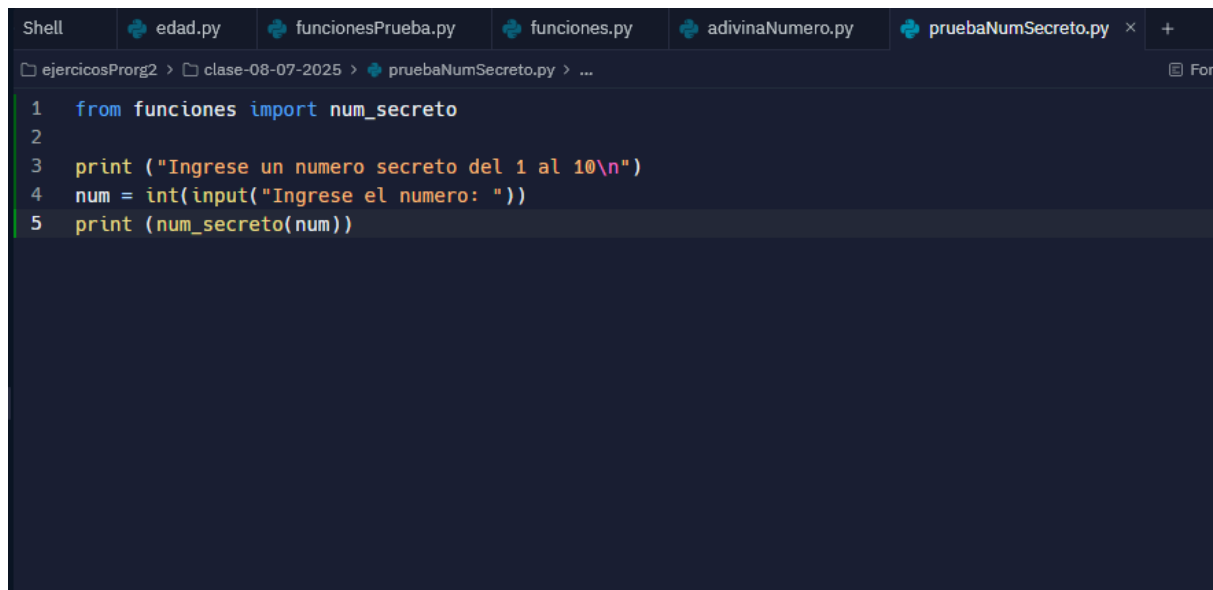
Versión original de la tarea

```
Shell  edad.py  funcionesPrueba.py  funciones.py  adivinaNumero.py x  pruebaNumSecreto.py  +  ⋮
ejerciciosProrg2 > acti1 > adivinaNumero.py > ...  Format
1  #Adivina el numero secreto
2
3  num = 7|
4
5  print("Adivina el numero secreto del 1 al 10\n")
6  num_user = int(input("\nIngrese el numero: "))
7  if num_user == num:
8      print("\nFelicidades adivinaste el numero secreto")
9  else:
10     print("\nLo siento no adivinaste el numero secreto\n")
11
12  print ("____Alejandro Hurtado____")
13
14
```

Función creada a partir el código original


```
17
18  #Adivina el numero secreto
19
20  def num_secreto(num):
21      print("Adivina el numero secreto del 1 al 10\n")
22      num_user = int(input("Ingrese el numero: ")) |
23      if num_user == num:
24          return "\nFelicidades, adivinaste el numero secreto"
25      else:
26          return "\nLo siento, no adivinaste el numero secreto\n"
27
28  print ("____Alejandro Hurtado____")
```

Prueba llamando a la función desde otro archivo de código de Python



```
Shell  edad.py  funcionesPrueba.py  funciones.py  adivinaNumero.py  pruebaNumSecreto.py  +
ejercicosProrg2 > clase-08-07-2025 > pruebaNumSecreto.py > ...
1  from funciones import num_secreto
2
3  print ("Ingrese un numero secreto del 1 al 10\n")
4  num = int(input("Ingrese el numero: "))
5  print (num_secreto(num))
```

Demostración del Funcionamiento en la terminal de Replit



```
Shell  edad.py  funcionesPrueba.py  funciones.py  adivinaNumero.py  pruebaNumSecreto.py  +
~/workspace/ejercicosProrg2/clase-08-07-2025: python pruebaNumSecreto.py

~/workspace$ cd ejerciciosProrg2/clase-08-07-2025/
~/.../ejercicosProrg2/clase-08-07-2025$ python pruebaNumSecreto.py
____Alejandro Hurtado____
Ingrese un numero secreto del 1 al 10

Ingrese el numero: 3
Adivina el numero secreto del 1 al 10

Ingrese el numero: 7

Lo siento, no adivinaste el numero secreto
~/.../ejercicosProrg2/clase-08-07-2025$
```

PRUEBAS DE LISTAS O VECTORES

Alejandro Hurtado Rodas – 10/07/2025 – 17:34

Aprendí a utilizar y guardar datos en una lista para que sean más fáciles de manejar y cambiar utilizando las listas en python, para el primer ejercicio cree un código simple en el que una lista contiene mis tres comidas favoritas y luego las imprime para posteriormente solicitar al usuario que cambie la primera comida de la lista y la vuelva a imprimir en la terminal.

```
listaNombreEstu.py listaNotas.py listaComidaFav.py X
ejercicosProrg2 > acti2 > listaComidaFav.py > ...
1 #Lista donde se guardan las comidas
2 comida_fav = ["Salteña", "Cuñape", "Sopa de Mani"]
3 print(comida_fav)
4
5 print("\nMi segunda comida favorita es " + comida_fav[1])
6 print("\nCambiar tu primera comida favorita")
7 nueva_comida = input("\nIngresa tu nueva comida favorita:")
8 comida_fav[0] = nueva_comida
9 print(comida_fav)
10 cantidad_comida = len(comida_fav)
11 print(f"\nLa cantidad de comidas favoritas que tienes es de {cantidad_comida}")
```

```
PS C:\clone\prog2-alejandrohurtado> & C:/Users/HP/AppData/Local/Programs/Python/Python39-64/Scripts/python.exe C:\clone\prog2-alejandrohurtado\ejercicosProrg2\acti2\listaComidaFav.py
['Salteña', 'Cuñape', 'Sopa de Mani']

Mi segunda comida favorita es Cuñape

Cambiar tu primera comida favorita

Ingresa tu nueva comida favorita:milanesa
['milanesa', 'Cuñape', 'Sopa de Mani']

La cantidad de comidas favoritas que tienes es de 3
PS C:\clone\prog2-alejandrohurtado> █
```

El segundo ejercicio consiste en tener una lista de nombres e imprimir los nombres con un mensaje de bienvenida usando un bucle For

```
listaNombreEstu.py X listaNotas.py listaComidaFav.py
ejercicosProrg2 > acti2 > listaNombreEstu.py > ...
1 lista_nom = ["nom1", "nom2", "nom3", "nom4", "nom5"]
2
3 for i in lista_nom:
4     print(f"Bienvenido al equipo {i} ")
5
```

```

PS C:\clone\prog2-alejandrohurtado> & C:/Users/HP/AppData/Local/Programs/Python/Python313/Python.exe C:\clone\prog2-alejandrohurtado/ejercicosProrg2/acti2/listaNombreEstu.py
Bienvenido al equipo nom1
Bienvenido al equipo nom2
Bienvenido al equipo nom3
Bienvenido al equipo nom4
Bienvenido al equipo nom5
PS C:\clone\prog2-alejandrohurtado> 

```

El tercer ejercicio consiste en tener una lista de notas de estudiantes, en mi caso puse 10 en todo para que sea muy fácil saber los resultados, y con esas notas sumar para conseguir el total y sacar la media de las notas pero sin usar las funciones de suma de python en su lugar se debe usar un bucle for e imprimir los resultados

```

ejercicosProrg2 > acti2 > listaNotas.py > ...
1  mis_notas = [10, 10, 10, 10, 10]
2  sum_total = 0
3  for i in mis_notas:
4      sum_total = sum_total + i
5
6  promedio = sum_total / len(mis_notas)
7  print(f"La suma total de todas las notas es de: {sum_total}")
8  print(f"El promedio de las notas es de: {promedio}")

```

```

PS C:\clone\prog2-alejandrohurtado> & C:/Users/HP/AppData/Local/Programs/Python/Python313/Python.exe C:\clone\prog2-alejandrohurtado/ejercicosProrg2/acti2/listaNotas.py
La suma total de todas las notas es de: 50
El promedio de las notas es de: 10.0
PS C:\clone\prog2-alejandrohurtado> 

```