

UNIVERSIDAD PRIVADA DOMINGO SAVIO



Actividad 01

DOCENTE: Jimmy N. Requena Llorentty

TURNO: Mañana

CARRERA: Ing. En Sistemas

MATERIA: Programación II

ESTUDIANTE: Maria F. Vidaurre Alvarado

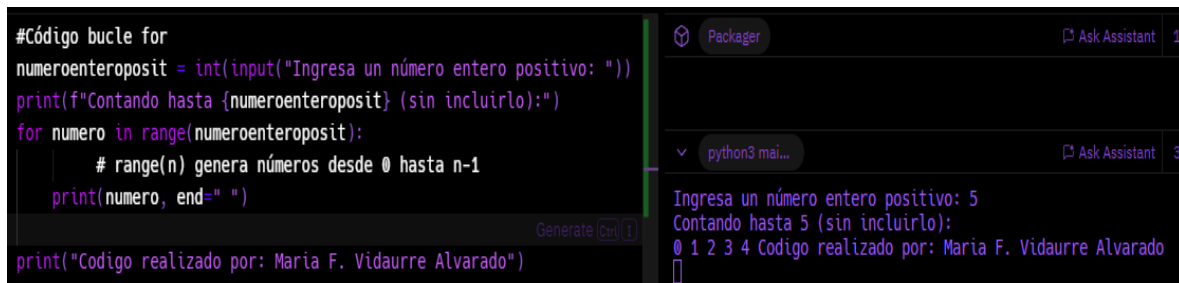
Santa Cruz- Bolivia

Fecha y hora actual: 2025-06-17 13:46:49

Captura #1 README.md



Captura #2 Bucle “for”



Captura #3 Bucle “while”



Captura #4 Clasificación películas en cine

```
# Código de que películas puedes ver en el cine
edad_str = input("Bienvenido al cine, ¿cuál es tu edad?: ")
edad = int(edad_str)

if edad < 0:
    print("Edad no válida. Por favor, ingresa un número positivo.")
elif edad >= 18:
    print("¡Puedes ver películas clasificadas R!")
elif edad >= 13: # Python llega aquí solo si edad NO es >= 18 y NO
es < 0
    print("Puedes ver películas clasificadas PG-13.")
else: # Si no es < 0, ni >= 18, ni >= 13, entonces debe ser < 13 y
>= 0
    print("Te recomendamos películas clasificadas G o P.")
print("Codigo realizado por: Maria F. Vidaurre Alvarado")
```

Package

python3 mai...

Bienvenido al cine, ¿cuál es tu edad?: 18
¡Puedes ver películas clasificadas R!
Codigo realizado por: Maria F. Vidaurre Alvarado

Package

python3 mai...

Bienvenido al cine, ¿cuál es tu edad?: 14
Puedes ver películas clasificadas PG-13.
Codigo realizado por: Maria F. Vidaurre Alvarado

Captura #5 Tabla multiplicar

```
# Código de la tabla de multiplicar
num_tabla = int(input("Ingresa un número para ver su tabla de
multiplicar: "))
print(f"--- Tabla del {num_tabla} ---")

for i in range(1, 11): # i tomará valores de 1 a 10
    resultado = num_tabla * i
    print(f"{num_tabla} x {i} = {resultado}")
print("Codigo realizado por: Maria F. Vidaurre Alvarado")
```

Package

python3 mai...

Ingresa un número para ver su tabla de multiplicar: 3
--- Tabla del 3 ---
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
3 x 6 = 18
3 x 7 = 21
3 x 8 = 24
3 x 9 = 27
3 x 10 = 30
Codigo realizado por: Maria F. Vidaurre Alvarado

Captura #6 Refactorizar cálculo área rectángulo

```
#Codigo de refactorizacion (incluimos variables y listas) rectangulo
def calcular_area_rectangulo(base, altura):
    return base * altura

rectangulos = [] #es una variable para almacenar los datos de los rectangulos
contador = 1 #es la variable para contar los rectangulos
while True:
    base = int(input("Ingrese la base del rectángulo: "))
    altura = int(input("Ingrese la altura del rectángulo: "))
    area = calcular_area_rectangulo(base, altura)
    rectangulos.append((contador, base, altura, area)) #el append te permite agregar elementos a la lista
    print(f"El área del rectángulo {contador} ({base}x{altura}) es: {area}")
    contador += 1
    continuar = input("¿Desea ingresar otro rectángulo? Presione Enter para continuar o 'n' para salir: ")
    if continuar.lower() == 'n':
        break
ver_datos = input("¿Desea ver los datos de los rectángulos guardados? Presione 1 para ver: ")
if ver_datos == '1':
    for rectangulo in rectangulos:
        numero, base, altura, area = rectangulo
        print(f"Rectángulo {numero}: Base = {base}, Altura = {altura}, Área = {area}")
print("Codigo realizado por: Maria F. Vidaurre Alvarado")
```

Captura #7 Adivina el número secreto

```
# Código del juego adivina el número secreto
# El número secreto ahora es fijo: 6 pero puede ser cambiado por cualquier otro número entero.
numero_secreto = 6
print("¡Bienvenido al juego Adivina el Número Secreto!")
print("Estoy pensando en el...")
# Iniciamos la variable adivinanza con un valor diferente al número secreto para que el bucle WHILE se ejecute al menos una vez.
adivinanza = 0
# Bucle que continua mientras la adivinanza no sea correcta
while adivinanza != numero_secreto:
    try:
        # Pide al usuario que adivine el número.
        adivinanza = int(input("Ingresa el número que crees que es el secreto: "))
        # Aqui le damos una pista si el número no es correcto
        if adivinanza > numero_secreto:
            print("Demasiado alto. Intenta de nuevo.")
        elif adivinanza < numero_secreto:
            print("Demasiado bajo. Intenta de nuevo.")
    except ValueError:
        # Maneja el error si el usuario no ingresa un número entero (int)
        print("Ingresa un número entero.")
# Mensaje (print) que aparece cuando el número fue adivinado
print(f"¡Correcto! El número era {numero_secreto}. ¡Felicidades, adivinaste!")
```

Captura #8 Comida boliviana favorita

```
# Crea una lista llamada comidas_favoritas con los nombres (strings) de tus 3 comidas bolivianas favoritas.
comidas_favoritas = ["Sopa de Mani", "Pique Macho", "Majadito"]

# Modificamos el primer elemento (índice 0) a "Patasca"
comidas_favoritas[0] = "Patasca"

# Imprimimos la lista completa para verificar que se creó bien.
print("Mi lista de comidas favoritas es:")
print(comidas_favoritas)

# Imprime un mensaje que diga: "Mi segunda comida favorita es: [el nombre de la segunda comida]".
print(f"\nMi segunda comida favorita es: {comidas_favoritas[1]}")

# Usamos len() al imprimir un mensaje
# len() nos da la cantidad de elementos en la lista.
print(f"Mi lista de comidas favoritas tiene {len(comidas_favoritas)} elementos.")
print("Codigo realizado por: Maria F. Vidaurre Alvarado")
```

Package Manager | Ask Assistant | ✓

python3 mai... | Ask Assistant | ✓

Mi lista de comidas favoritas es:
['Sopa de Mani', 'Pique Macho', 'Majadito']

Mi segunda comida favorita es: Pique Macho
Mi lista de comidas favoritas tiene 3 elementos.
Codigo realizado por: Maria F. Vidaurre Alvarado

Package Manager | Ask Assistant | ✓

python3 mai... | Ask Assistant | ✓

Mi lista de comidas favoritas es:
['Patasca', 'Pique Macho', 'Majadito']

Mi segunda comida favorita es: Pique Macho
Mi lista de comidas favoritas tiene 3 elementos.
Codigo realizado por: Maria F. Vidaurre Alvarado

Captura #9 Lista de notas y promedio

```
#Codigo para calcular el promedio de notas
# Creamos una lista llamada lista_mis_notas con las notas numéricas.
lista_mis_notas = [60.5, 74, 80.5, 90.5]
# Creamos una variable suma_total inicializada en 0.
suma_total = 0

# Usamos un bucle for para recorrer lista_mis_notas.
# En cada iteración, añade la nota actual a suma_total.
for nota in lista_mis_notas:
    suma_total = suma_total + nota

# Calcula el promedio: promedio = suma_total / len(lista_mis_notas).
# len(lista_mis_notas) nos da la cantidad de elementos en la lista.
promedio = suma_total / len(lista_mis_notas)

# Imprime(print) la suma total y el promedio.
print(f"Mis notas son: {lista_mis_notas}")
print(f"La suma total de las notas es: {suma_total:.2f}")
print(f"El promedio de las notas es: {promedio:.2f}") # :.2f para mostrar dos decimales
print("Codigo realizado por: Maria F. Vidaurre Alvarado")
```

Package Manager | Ask Assistant | ✓

python3 mai... | Ask Assistant | ✓

Mis notas son: [60.5, 74, 80.5, 90.5]
La suma total de las notas es: 305.50
El promedio de las notas es: 76.38
Codigo realizado por: Maria F. Vidaurre Alvarado

Captura #10 Sumar elementos de una lista

```
#Codigo para sumar todos los elementos de una lista
def sumar_elementos(lista_numeros):
    acumulador_suma = 0
    for numero in lista_numeros:
        acumulador_suma += numero
    return acumulador_suma

# --- Casos de Prueba con assert y manejo de errores ---
print("Iniciando pruebas de la función 'sumar_elementos'...")
pruebas_pasadas = 0
total_pruebas = 0

# Prueba 1: Suma básica
total_pruebas += 1
try:
    assert sumar_elementos([1, 2, 3, 4, 5]) == 15
    print("✅ Prueba 1 (Suma básica): Pasó")
    pruebas_pasadas += 1
except AssertionError:
    print("❌ Prueba 1 (Suma básica): ¡Falló! El resultado no fue 15.)
```

Package

python3 mai...

Iniciando pruebas de la función 'sumar_elementos'...

- ✅ Prueba 1 (Suma básica): Pasó
- ✅ Prueba 2 (Con números negativos): Pasó
- ✅ Prueba 3 (Lista vacía): Pasó
- ✅ Prueba 4 (Un solo elemento): Pasó

Resumen de pruebas: 4 de 4 pruebas pasaron exitosamente.
¡Todas las pruebas pasaron exitosamente! Nuestra función es robusta.

--- Demostración con tu lista ---
La lista de números es: [10, 5, 20, 15, 8, 25]
La suma total es: 83
Codigo realizado por: Maria F. Vidaurre Alvarado

Captura #11 Encontrar número mayor en lista

```
#Codigo numero mayor en lista
def encontrar_mayor(lista_numeros):
    """
    Encuentra el número más grande en una lista de números.

    Args:
        lista_numeros (list): La lista de números a evaluar.

    Returns:
        int/float or None: El número más grande en la lista, o None si la lista está vacía.
    """
    # Manejo del caso especial: lista vacía
    if not lista_numeros:
        return None

    # Suposición Inicial: Asume que el primer elemento es el más grande hasta ahora.
    mayor_temporal = lista_numeros[0]

    # Iteración: Recorre la lista desde el segundo elemento
    for i in range(1, len(lista_numeros)):
        elemento_actual = lista_numeros[i]

        # Comparación: Si el elemento actual es más grande, actualiza mayor_temporal.
```

Package

python3 mai...

Iniciando pruebas de la función 'encontrar_mayor'...

- ✅ Prueba 1 (Positivos variados): Pasó
- ✅ Prueba 2 (Números negativos): Pasó
- ✅ Prueba 3 (Elementos iguales): Pasó
- ✅ Prueba 4 (Lista vacía): Pasó
- ✅ Prueba 5 (Un solo elemento): Pasó

Resumen de pruebas: 5 de 5 pruebas pasaron exitosamente.
¡Todas las pruebas pasaron exitosamente! Nuestra función es robusta.

--- Demostración con tu lista original ---
La lista de números es: [45, 12, 78, 33, 90, 67, 21]
El número más grande en la lista es: 90
Codigo realizado por: Maria F. Vidaurre Alvarado

Captura #12 Contador de elementos en lista

```
#Codigo de contador en lista (cuantas veces aparece un elemento en una lista)
def contar_repeticiones(lista, elemento_buscado):
    contador = 0
    for elemento in lista:
        if elemento == elemento_buscado:
            contador += 1
    return contador

mi_lista = [1, 2, 3, 2, 4, 5, 2, 6, 7, 2, 8, 9, 2, 10, 5, 5]
elemento_a_buscar = 5

resultado = contar_repeticiones(mi_lista, elemento_a_buscar)

print(f"El número {elemento_a_buscar} aparece {resultado} veces en la lista.")
print("Codigo realizado por: Maria F. Vidaurre Alvarado")
```

El número 2 aparece 5 veces en la lista.
Codigo realizado por: Maria F. Vidaurre Alvarado

El número 5 aparece 3 veces en la lista.
Codigo realizado por: Maria F. Vidaurre Alvarado

Captura #13 Invertir lista

```
#Codigo de Invertir lista con range, append
def invertir_lista(lista_original):
    lista_invertida = []
    for i in range(len(lista_original) - 1, -1, -1):
        lista_invertida.append(lista_original[i])
    return lista_invertida

lista_original = [1, 2, 3, 4, 5]
print ("Lista original: ", lista_original)
print ("lista_invertida ;" , invertir_lista(lista_original))
print("Codigo realizado por: Maria F. Vidaurre Alvarado")
```

Lista original: [1, 2, 3, 4, 5]
lista_invertida ; [5, 4, 3, 2, 1]
Codigo realizado por: Maria F. Vidaurre Alvarado

Captura #14 Búsqueda lineal

```
#Codigo de busqueda lineal
def busqueda_lineal(lista, elemento_buscado):
    for i in range(len(lista)):
        if lista[i] == elemento_buscado:
            return i
    return -1

mi_lista = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
elemento_buscado = 3
#sustitui clave por elemento_buscado
indice_encontrado = busqueda_lineal(mi_lista, elemento_buscado)

if indice_encontrado != -1:
    print(f"El elemento {elemento_buscado} se encontró en el índice: {indice_encontrado}")
else:
    print(f"El elemento {elemento_buscado} no se encontró en la lista.")
print("Codigo realizado por: Maria F. Vidaurre Alvarado")
```

El elemento 8 se encontró en el índice: 7
Codigo realizado por: Maria F. Vidaurre Alvarado

El elemento 3 se encontró en el índice: 2
Codigo realizado por: Maria F. Vidaurre Alvarado

Captura #15 Búsqueda Binaria

```
#Codigo de busqueda binaria
def busqueda_binaria(lista_ordenada, clave):
    izquierda = 0
    derecha = len(lista_ordenada) - 1
    while izquierda <= derecha:
        medio = (izquierda + derecha) // 2

        if lista_ordenada[medio] == clave:
            return medio
        elif clave > lista_ordenada[medio]:
            izquierda = medio + 1
        else:
            derecha = medio - 1
    return -1

# lista de prueba xd
lista_ordenada = [2, 5, 8, 12, 16, 23, 38, 56, 72, 91]
# Cuando la clave es encontrada o no
clave_buscada_1 = 24
indice_1 = busqueda_binaria(lista_ordenada, clave_buscada_1)
if indice_1 != -1:
    print(f"La clave {clave_buscada_1} se encontró en el índice: {indice_1}")
else:
    print(f"La clave {clave_buscada_1} no se encontró en la lista.")
```

⊞ Packager Ask Assistant ✓

python3 mai... Ask Assistant ✓

La clave 23 se encontró en el índice: 5
Codigo realizado por: Maria F. Vidaurre Alvarado

⊞ Packager Ask Assistant ✓

python3 mai... Ask Assistant ✓

La clave 24 no se encontró en la lista.
Codigo realizado por: Maria F. Vidaurre Alvarado