

Planificación y Desarrollo de una Aplicación

Desafío:

El objetivo de este desafío es desarrollar una aplicación para resolver un problema determinado. Antes de comenzar con la programación, es necesario crear un plan en forma de pseudocódigo que sirva como guía. Esta planificación es esencial para estructurar correctamente la solución y facilitar su implementación en código.

Etapas del Desafío:

1. Generación del Pseudocódigo:

- Describir, en pasos lógicos, cómo funcionará la aplicación.
- Incluir las operaciones principales que realizará la aplicación de acuerdo al problema que se debe resolver.
- Asegurarse de que cada funcionalidad esté clara y bien definida en el pseudocódigo para evitar confusión durante la fase de implementación.
- Anticipar posibles interacciones del usuario con el sistema y cómo se manejarán los datos.

2. Desarrollo del Código:

- Basarse en el pseudocódigo previamente generado para escribir el código que implemente la solución.
- Seguir el plan estructurado en el pseudocódigo y ajustarlo según sea necesario durante el desarrollo.
- Garantizar que todas las funcionalidades y casos previstos en el pseudocódigo estén reflejados en el código final.

Este enfoque sistemático asegura que cualquier problema que se esté abordando tenga una solución bien planificada y organizada.

Ejemplo 1: Seguidor de Gastos Mensuales

Declaración del Problema:

Imagina que formas parte de un pequeño equipo de negocio, y tu gerente se ha acercado a ti con una tarea. Quieren que crees un programa fácil de usar para ayudar a rastrear los gastos mensuales. Tu programa debe permitir a los usuarios ingresar gastos diarios y, al final del mes, calcular sin esfuerzo el gasto total.

```
# INICIO
# DECLARAR gastos COMO LISTA VACÍA
# DECLARAR totalGasto COMO NÚMERO
# totalGasto <- 0
```

```

# IMPRIMIR "Bienvenido al Seguidor de Gastos Mensuales"
# PARA cada día EN el mes HACER
# IMPRIMIR "Ingrese el gasto del día ", día, " (o escriba 'fin' para
terminar):"
# LEER gasto
# SI gasto ES igual a 'fin' ENTONCES
# SALIR DEL BUCLE
# FIN SI
# AÑADIR gasto A la lista gastos
# totalGasto <- totalGasto + gasto
# FIN PARA
# IMPRIMIR "Gasto total del mes: ", totalGasto
# IMPRIMIR "Detalles de los gastos: ", gastos
# FIN

# INICIO
# DECLARAR gastos COMO LISTA VACÍA
gastos = []
# DECLARAR totalGasto COMO NÚMERO
totalGasto = 0

# IMPRIMIR "Bienvenido al Seguidor de Gastos Mensuales"
print("Bienvenido al Seguidor de Gastos Mensuales")

# PARA cada día EN el mes HACER
for dia in range(1, 31): # Asumiendo un mes de 30 días
    # IMPRIMIR "Ingrese el gasto del día ", día, " (o escriba 'fin'
para terminar):"
    gasto = input(f"Ingrese el gasto del día {dia} (o escriba 'fin'
para terminar): ")

    # SI gasto ES igual a 'fin' ENTONCES
    if gasto.lower() == 'fin':
        # SALIR DEL BUCLE
        break

    # Intentar convertir el gasto a un número
    try:
        gasto = float(gasto) # Convertir a float
        # AÑADIR gasto A la lista gastos
        gastos.append(gasto)
        # totalGasto <- totalGasto + gasto
        totalGasto += gasto
    except ValueError:
        print("Por favor, ingrese un número válido o 'fin' para
terminar.")

# FIN PARA
# IMPRIMIR "Gasto total del mes: ", totalGasto
print(f"Gasto total del mes: {totalGasto:.2f}")

```

```
# IMPRIMIR "Detalles de los gastos: ", gastos
print("Detalles de los gastos:", gastos)
# FIN
```

Bienvenido al Seguidor de Gastos Mensuales

```
Ingrese el gasto del día 1 (o escriba 'fin' para terminar): 1000
Ingrese el gasto del día 2 (o escriba 'fin' para terminar): 20000
Ingrese el gasto del día 3 (o escriba 'fin' para terminar): 3000
Ingrese el gasto del día 4 (o escriba 'fin' para terminar): 40000
Ingrese el gasto del día 5 (o escriba 'fin' para terminar): 500000
Ingrese el gasto del día 6 (o escriba 'fin' para terminar): fin
```

Gasto total del mes: 564000.00

Detalles de los gastos: [1000.0, 20000.0, 3000.0, 40000.0, 500000.0]

Inventario de Tienda en Línea

Declaración del Problema:

Se te ha encargado crear un programa simple para una tienda en línea. El programa debería ayudar a gestionar el inventario de productos disponibles en la tienda. El inventario consta de varios productos con sus respectivos nombres, precios y cantidades en stock. Tu programa debe tener las siguientes características:

- Inicializar un inventario vacío para empezar.
- Permitir al usuario añadir nuevos productos al inventario. Cada producto debe tener un nombre, precio y cantidad inicial en stock.
- Mostrar el inventario actual, mostrando el nombre, precio y cantidad de cada producto.
- Permitir al usuario buscar un producto específico por nombre y ver sus detalles (precio y cantidad).
- Habilitar al usuario para actualizar la cantidad de un producto específico en stock.
- Implementar una función para calcular y mostrar el valor total de todo el inventario.

```
# INICIO
# DECLARAR inventario COMO LISTA VACÍA
# IMPRIMIR "Bienvenido al sistema de gestión de inventario"

# MIENTRAS VERDADERO HACER
#   IMPRIMIR "Seleccione una opción:"
#   IMPRIMIR "1. Añadir nuevo producto"
#   IMPRIMIR "2. Mostrar inventario"
#   IMPRIMIR "3. Buscar producto"
#   IMPRIMIR "4. Actualizar cantidad de producto"
#   IMPRIMIR "5. Calcular valor total del inventario"
#   IMPRIMIR "6. Salir"
#   LEER opcion
```

```

# SI opcion ES igual a 1 ENTONCES
#     IMPRIMIR "Ingrese el nombre del producto:"
#     LEER nombre
#     IMPRIMIR "Ingrese el precio del producto:"
#     LEER precio
#     IMPRIMIR "Ingrese la cantidad del producto:"
#     LEER cantidad
#     AÑADIR (nombre, precio, cantidad) A la lista inventario
# FIN SI

# SI opcion ES igual a 2 ENTONCES
#     IMPRIMIR "Inventario actual:"
#     PARA cada producto EN inventario HACER
#         IMPRIMIR "Nombre: ", producto.nombre, ", Precio: ",
producto.precio, ", Cantidad: ", producto.cantidad
#     FIN PARA
# FIN SI

# SI opcion ES igual a 3 ENTONCES
#     IMPRIMIR "Ingrese el nombre del producto a buscar:"
#     LEER nombreBuscado
#     ENCONTRADO <- FALSO
#     PARA cada producto EN inventario HACER
#         SI producto.nombre ES igual a nombreBuscado ENTONCES
#             IMPRIMIR "Producto encontrado - Nombre: ",
producto.nombre, ", Precio: ", producto.precio, ", Cantidad: ",
producto.cantidad
#             ENCONTRADO <- VERDADERO
#             SALIR DEL BUCLE
#         FIN SI
#     FIN PARA
#     SI ENCONTRADO ES FALSO ENTONCES
#         IMPRIMIR "Producto no encontrado."
#     FIN SI
# FIN SI

# SI opcion ES igual a 4 ENTONCES
#     IMPRIMIR "Ingrese el nombre del producto a actualizar:"
#     LEER nombreActualizar
#     ENCONTRADO <- FALSO
#     PARA cada producto EN inventario HACER
#         SI producto.nombre ES igual a nombreActualizar ENTONCES
#             IMPRIMIR "Ingrese la nueva cantidad:"
#             LEER nuevaCantidad
#             producto.cantidad <- nuevaCantidad
#             IMPRIMIR "Cantidad actualizada."
#             ENCONTRADO <- VERDADERO
#             SALIR DEL BUCLE
#         FIN SI

```

```

#     FIN PARA
#     SI ENCONTRADO ES FALSO ENTONCES
#         IMPRIMIR "Producto no encontrado."
#     FIN SI
# FIN SI

# SI opcion ES igual a 5 ENTONCES
#     totalValor <- 0
#     PARA cada producto EN inventario HACER
#         totalValor <- totalValor + (producto.precio *
producto.cantidad)
#     FIN PARA
#     IMPRIMIR "El valor total del inventario es: ", totalValor
# FIN SI

# SI opcion ES igual a 6 ENTONCES
#     IMPRIMIR "Saliendo del sistema."
#     SALIR DEL BUCLE
# FIN SI
# FIN MIENTRAS
# FIN

# INICIO
# DECLARAR inventario COMO LISTA VACÍA
inventario = []
print("Bienvenido al sistema de gestión de inventario")

# MIENTRAS VERDADERO HACER
while True:
    # IMPRIMIR "Seleccione una opción:"
    print("Seleccione una opción:")
    print("1. Añadir nuevo producto")
    print("2. Mostrar inventario")
    print("3. Buscar producto")
    print("4. Actualizar cantidad de producto")
    print("5. Calcular valor total del inventario")
    print("6. Salir")

    # LEER opcion
    opcion = input("Ingrese su opción: ")

    # SI opcion ES igual a 1 ENTONCES
    if opcion == '1':
        #     IMPRIMIR "Ingrese el nombre del producto:"
        nombre = input("Ingrese el nombre del producto: ")
        #     IMPRIMIR "Ingrese el precio del producto:"
        precio = float(input("Ingrese el precio del producto: "))
        #     IMPRIMIR "Ingrese la cantidad del producto:"
        cantidad = int(input("Ingrese la cantidad del producto: "))
        #     AÑADIR (nombre, precio, cantidad) A la lista

```

```

inventario
    inventario.append({'nombre': nombre, 'precio': precio,
'cantidad': cantidad})

    #   FIN SI

    #   SI opcion ES igual a 2 ENTONCES
    elif opcion == '2':
        #       IMPRIMIR "Inventario actual:"
        print("Inventario actual:")
        for producto in inventario:
            print(f"Nombre: {producto['nombre']}, Precio:
{producto['precio']}, Cantidad: {producto['cantidad']}")

        #   FIN SI

        #   SI opcion ES igual a 3 ENTONCES
        elif opcion == '3':
            #       IMPRIMIR "Ingrese el nombre del producto a buscar:"
            nombreBuscado = input("Ingrese el nombre del producto a
buscar: ")
            encontrado = False
            #       PARA cada producto EN inventario HACER
            for producto in inventario:
                #           SI producto.nombre ES igual a nombreBuscado
                ENTONCES
                    if producto['nombre'] == nombreBuscado:
                        #           IMPRIMIR "Producto encontrado -
Nombre: ", producto.nombre, ", Precio: ", producto.precio, ",
Cantidad: ", producto.cantidad
                        print(f"Producto encontrado - Nombre:
{producto['nombre']}, Precio: {producto['precio']}, Cantidad:
{producto['cantidad']}")
                        encontrado = True
                        break
            #       FIN PARA
            #       SI ENCONTRADO ES FALSO ENTONCES
            if not encontrado:
                print("Producto no encontrado.")
            #   FIN SI

            #   SI opcion ES igual a 4 ENTONCES
            elif opcion == '4':
                #       IMPRIMIR "Ingrese el nombre del producto a
actualizar:"
                nombreActualizar = input("Ingrese el nombre del producto a
actualizar: ")
                encontrado = False
                #       PARA cada producto EN inventario HACER
                for producto in inventario:

```

```

# SI producto.nombre ES igual a nombreActualizar
ENTONCES
    if producto['nombre'] == nombreActualizar:
        # IMPRIMIR "Ingrese la nueva cantidad:"
        nuevaCantidad = int(input("Ingrese la nueva cantidad:
"))
        producto['cantidad'] = nuevaCantidad
        print("Cantidad actualizada.")
        encontrado = True
        break
    # FIN PARA
    # SI ENCONTRADO ES FALSO ENTONCES
    if not encontrado:
        print("Producto no encontrado.")
# FIN SI

# SI opcion ES igual a 5 ENTONCES
elif opcion == '5':
    totalValor = 0
    # PARA cada producto EN inventario HACER
    for producto in inventario:
        totalValor += producto['precio'] * producto['cantidad']
    # FIN PARA
    # IMPRIMIR "El valor total del inventario es: ",
totalValor
    print(f"El valor total del inventario es: {totalValor:.2f}")
# FIN SI

# SI opcion ES igual a 6 ENTONCES
elif opcion == '6':
    # IMPRIMIR "Saliendo del sistema."
    print("Saliendo del sistema.")
    break
# FIN SI
# FIN MIENTRAS
# FIN

```

Bienvenido al sistema de gestión de inventario

Seleccione una opción:

1. Añadir nuevo producto
2. Mostrar inventario
3. Buscar producto
4. Actualizar cantidad de producto
5. Calcular valor total del inventario
6. Salir

Ingrese su opción: 6

Saliendo del sistema.

Gestor de Tareas

Declaración del Problema:

Se te ha encargado crear un programa que ayude a gestionar una lista de tareas. El programa debe permitir a los usuarios agregar, eliminar y visualizar tareas, así como marcar las tareas como completadas. a aplicación debe tener las siguientes características:

- Inicializar una lista vacía para las tareas.
- Permitir al usuario agregar una nueva tarea con una descripción.
- Mostrar todas las tareas actuales, indicando cuáles están completadas y cuáles no.
- Permitir al usuario marcar una tarea como completada.
- Permitir al usuario eliminar una tarea de la lista.
- Implementar una opción para salir del programa.

El programa debe ser fácil de usar y manejar la entrada del usuario de manera adecuada.

```
# INICIO
# DECLARAR tareas COMO LISTA VACÍA
# IMPRIMIR "Bienvenido al Gestor de Tareas"

# MIENTRAS VERDADERO HACER
#     IMPRIMIR "Seleccione una opción:"
#     IMPRIMIR "1. Agregar nueva tarea"
#     IMPRIMIR "2. Mostrar tareas"
#     IMPRIMIR "3. Marcar tarea como completada"
#     IMPRIMIR "4. Eliminar tarea"
#     IMPRIMIR "5. Salir"
#     LEER opcion

#     SI opcion ES igual a 1 ENTONCES
#         IMPRIMIR "Ingrese la descripción de la tarea:"
#         LEER descripcion
#         AÑADIR (descripcion, FALSO) A la lista tareas
#     FIN SI

#     SI opcion ES igual a 2 ENTONCES
#         IMPRIMIR "Lista de tareas:"
#         PARA cada tarea EN tareas HACER
#             SI tarea está completada ENTONCES
#                 IMPRIMIR "Tarea: ", tarea.descripcion, " - Completada"
#             SINO
#                 IMPRIMIR "Tarea: ", tarea.descripcion, " - No
completada"
#             FIN SI
#         FIN PARA
#     FIN SI
```



```

# SI opcion ES igual a 3 ENTONCES
#     IMPRIMIR "Ingrese el número de la tarea a marcar como
completada:"
#     LEER numeroTarea
#     SI numeroTarea es válido ENTONCES
#         tarea <- tareas[numeroTarea - 1]
#         tarea.completada <- VERDADERO
#         IMPRIMIR "Tarea marcada como completada."
#     SINO
#         IMPRIMIR "Número de tarea no válido."
#     FIN SI
# FIN SI

# SI opcion ES igual a 4 ENTONCES
#     IMPRIMIR "Ingrese el número de la tarea a eliminar:"
#     LEER numeroTarea
#     SI numeroTarea es válido ENTONCES
#         ELIMINAR tarea en tareas[numeroTarea - 1]
#         IMPRIMIR "Tarea eliminada."
#     SINO
#         IMPRIMIR "Número de tarea no válido."
#     FIN SI
# FIN SI

# SI opcion ES igual a 5 ENTONCES
#     IMPRIMIR "Saliendo del Gestor de Tareas."
#     SALIR DEL BUCLE
# FIN SI
# FIN MIENTRAS
# FIN

# INICIO
# DECLARAR tareas COMO LISTA VACÍA
tareas = []
print("Bienvenido al Gestor de Tareas")

# MIENTRAS VERDADERO HACER
while True:
    # IMPRIMIR "Seleccione una opción:"
    print("Seleccione una opción:")
    print("1. Agregar nueva tarea")
    print("2. Mostrar tareas")
    print("3. Marcar tarea como completada")
    print("4. Eliminar tarea")
    print("5. Salir")

    # LEER opcion
    opcion = input("Ingrese su opción: ")

    # SI opcion ES igual a 1 ENTONCES

```

```

if opcion == '1':
    # IMPRIMIR "Ingrese la descripción de la tarea:"
    descripcion = input("Ingrese la descripción de la tarea: ")
    # AÑADIR (descripcion, FALSO) A la lista tareas
    tareas.append({'descripcion': descripcion, 'completada':
False})

# FIN SI

# SI opcion ES igual a 2 ENTONCES
elif opcion == '2':
    # IMPRIMIR "Lista de tareas:"
    print("Lista de tareas:")
    for i, tarea in enumerate(tareas):
        if tarea['completada']:
            print(f"{i + 1}. Tarea: {tarea['descripcion']} -
Completada")
        else:
            print(f"{i + 1}. Tarea: {tarea['descripcion']} - No
completada")

# FIN SI

# SI opcion ES igual a 3 ENTONCES
elif opcion == '3':
    # IMPRIMIR "Ingrese el número de la tarea a marcar como
completada:"
    numeroTarea = int(input("Ingrese el número de la tarea a
marcar como completada: "))
    # SI numeroTarea es válido ENTONCES
    if 1 <= numeroTarea <= len(tareas):
        tarea = tareas[numeroTarea - 1]
        tarea['completada'] = True
        print("Tarea marcada como completada.")
    else:
        print("Número de tarea no válido.")
# FIN SI

# SI opcion ES igual a 4 ENTONCES
elif opcion == '4':
    # IMPRIMIR "Ingrese el número de la tarea a eliminar:"
    numeroTarea = int(input("Ingrese el número de la tarea a
eliminar: "))
    # SI numeroTarea es válido ENTONCES
    if 1 <= numeroTarea <= len(tareas):
        tareas.pop(numeroTarea - 1)
        print("Tarea eliminada.")
    else:
        print("Número de tarea no válido.")
# FIN SI

```

```
# SI opcion ES igual a 5 ENTONCES
elif opcion == '5':
    # IMPRIMIR "Saliendo del Gestor de Tareas."
    print("Saliendo del Gestor de Tareas.")
    break
# FIN SI
# FIN MIENTRAS
# FIN
```

Bienvenido al Gestor de Tareas
Seleccione una opción:
1. Agregar nueva tarea
2. Mostrar tareas
3. Marcar tarea como completada
4. Eliminar tarea
5. Salir

Ingrese su opción: 5

Saliendo del Gestor de Tareas.