



# **Actividad de Resolución de problemas:** **Sistema de citas**

M1. Fundamentos de Programación



**Problema por solucionar**

Has sido contratado en una agencia para agendar citas a los ciudadanos. Cada persona ingresa su nombre y el tiempo estimado que necesita para resolver su problema. Para mejorar la experiencia de los ciudadanos, se necesita minimizar el tiempo de espera organizando las solicitudes de manera eficiente.

**Objetivo del algoritmo (sistema de citas) por diseñar**

Deberás crear un sistema de citas con el objetivo de ordenar las solicitudes de manera lógica para reducir los tiempos de espera y mejorar la satisfacción de los ciudadanos.

**Actividades**

Para desarrollar el algoritmo para el sistema de citas, realiza uno a uno los siguientes pasos:

## 1. Planeación

Haz una lista de los pasos que seguirás para resolver el problema.

*Ejemplo de pasos básicos:*

- Solicitar la cantidad de ciudadanos que harán una solicitud.
- Registrar el nombre y tiempo estimado de cada solicitud.
- Ordenar las solicitudes de menor a mayor tiempo estimado.
- Mostrar la lista ordenada para su atención.

## 2. Análisis

Identifica los elementos clave del problema:

- Entrada de datos:
  - Nombre del ciudadano.
  - Tiempo estimado de atención (máximo 120 minutos).
- Proceso:
  - Ordenar las solicitudes de menor a mayor tiempo.
- Salida esperada:
  - Una lista organizada de solicitudes para su atención eficiente.

## 3. Diseño de Solución

Realiza en pseudocódigo la solución planteada:

- Pedir al usuario la cantidad de solicitudes a registrar.
- Para cada solicitud, pedir el nombre y el tiempo estimado.

## Módulo 1.

### Fundamentos de programación

- Guardar los datos en una lista.
- Ordenar la lista según el tiempo estimado.
- Mostrar la lista ordenada de solicitudes.

```
inicio Registro Solicitudes
    Definir n, i Como Entero
    Definir nombre Como Cadena
    Definir tiempo Como Entero
    Definir solicitudes Como Lista

    Escribir "Ingrese la cantidad de solicitudes a registrar:"
    Leer n

    solicitudes <- []

    Para i <- 1 Hasta n Hacer
        Escribir "Ingrese el nombre de la solicitud ", i, ":"
        Leer nombre
        Escribir "Ingrese el tiempo estimado de la solicitud ",
i, ":"
        Leer tiempo

        Agregar (nombre, tiempo) a solicitudes
    FinPara

    // Ordenar la lista por el tiempo estimado (menor a mayor)
    Ordenar solicitudes Por tiempo

    Escribir "Lista de solicitudes ordenadas por tiempo
estimado:"
    Para cada solicitud en solicitudes Hacer
        Escribir "Nombre: ", solicitud.nombre, " - Tiempo: ",
solicitud.tiempo
    FinPara
Fin
```

## 4. Desarrollo

Desarrolla la solución diseñada en lenguaje de programación (solo entrada manual).

*NOTA: Tu docente te indicará el lenguaje a utilizar.*

```
# Programa para registrar solicitudes con tiempo en horas y minutos
print("*** Bienvenido al Sistema de Citas ***")

n = int(input("Ingrese la cantidad de citas a registrar: "))

solicitudes = [] # lista vacía

for i in range(1, n + 1):
    name_solicitante = input(f"Ingrese el nombre del solicitante {i}: ")

    while True:
        hour = int(input(f"Ingrese las horas estimadas para la solicitud {i} (0-2): "))
        min = int(input(f"Ingrese los minutos estimados para la solicitud {i} (0-59): "))

        # Validación de minutos
        if min < 0 or min >= 60:
            print("⚠ Los minutos deben estar entre 0 y 59. Intente nuevamente.")
            continue

        time = hour * 60 + min # Convertir todo a minutos

        # Validación de tiempo máximo
        if 0 < time <= 120:
            break
        else:
            print("⚠ El tiempo total debe ser mayor a 0 y no superar los 120 minutos. Intente nuevamente.")

    solicitudes.append((name_solicitante, time))

# Ordenar lista por tiempo
solicitudes.sort(key=lambda x: x[1])

# Mostrar resultados
print("\n📋 Lista de solicitudes ordenadas por tiempo estimado:")
for solicitud in solicitudes:
    name_solicitante, time = solicitud
    hour = time // 60
    min = time % 60
    print(f"Nombre: {name_solicitante} - Tiempo: {hour}h {min}m")

print("\n✅ Citas generadas exitosamente")
```

## 5. Pruebas

Realiza tres pruebas con diferentes datos de entrada.



### Reflexión

Reflexiona sobre lo aprendido en esta actividad, respondiendo a las siguientes preguntas, relacionadas con el algoritmo desarrollado:

- ¿Cómo afecta el orden de las solicitudes en el tiempo de espera total?
- Si el tiempo de espera de una solicitud se duplicara, ¿cómo cambiaría la organización de las citas?
- ¿Cómo podrías mejorar este algoritmo sin complicarlo demasiado?