



# **Universidad Politécnica de Tlaxcala Región Poniente**

*Ingeniería en Sistemas Computacionales*

Materia: Programación Orientada a Objetos

Docente: Ing. Vanesa Tenopala Zavala

Alumnos: 22SIC008

Isaac Brandon Martínez Ramírez

Tema: Reportes Ejercicios Hilos

Ciclo escolar: mayo-agosto 2024

Fecha: 07 de julio de 2024

# Reporte del Programa Ejercicio1

## 1. Descripción del Programa

El programa "Ejercicio" permite capturar los nombres de dos usuarios, el día y la hora en que ingresaron a su centro de labores. Luego, el programa determina si llegaron temprano o tarde basándose en una hora de ingreso esperada de las 8:00 AM. La interfaz gráfica desarrollada en Swing mejora la experiencia del usuario al proporcionar una forma intuitiva de ingresar los datos y ver los resultados.

## 2. Objetivos

- **Capturar datos de entrada:** Nombres, días y horas de ingreso de dos usuarios.
- **Determinar puntualidad:** Comparar las horas de ingreso con la hora esperada y determinar si los usuarios llegaron temprano o tarde.
- **Mostrar resultados:** Presentar los resultados en una interfaz gráfica de usuario (GUI).

## 3. Componentes del Programa

### 3.1. Interfaz Gráfica (GUI)

- **JFrame:** Ventana principal del programa.
- **JPanel:** Panel para organizar los componentes de la GUI en una cuadrícula de 6 filas y 2 columnas.
- **JLabels:** Etiquetas para los campos de entrada.
- **JTextFields:** Campos de texto para capturar los nombres, días y horas de ingreso.
- **JButton:** Botón para iniciar el cálculo de puntualidad.
- **JTextArea:** Área de texto para mostrar los resultados.
- **JScrollPane:** Panel de desplazamiento para la JTextArea.

### 3.2. Clases

- **Ejercicio1:** Clase principal que extiende JFrame y contiene la lógica de la GUI.
- **IngresoUsuario:** Clase interna que extiende Thread y se encarga de procesar la información de cada usuario en paralelo.

#### 4. Funcionamiento del Programa

1. **Inicialización:** Al ejecutar el programa, se inicializa la ventana principal Ejercicio1 con los componentes de la interfaz gráfica.
2. **Entrada de Datos:** El usuario ingresa los nombres, días y horas de ingreso para dos empleados.
3. **Ejecución de Hilos:** Al presionar el botón "Calcular", se crean y ejecutan dos hilos (IngresoUsuario), uno para cada empleado.
4. **Determinación de Puntualidad:** Cada hilo compara la hora de ingreso con la hora esperada (8:00 AM) y determina si el empleado llegó temprano o tarde.

#### 5. Resultado

Nombre del empleado 1:	Isaac
Día del ingreso 1:	1
Hora del ingreso 1 (HH.MM):	09:02
Nombre del empleado 2:	Diego
Día del ingreso 2:	5
Hora del ingreso 2 (HH.MM):	09:30

Calcular

#### 6. Conclusión

El programa "Ingreso a Centro de Labores" ofrece una solución intuitiva y eficiente para determinar la puntualidad de dos empleados mediante el uso de hilos y una interfaz gráfica. La implementación de hilos permite procesar la información de cada empleado en paralelo, mejorando la eficiencia del programa. Además, la interfaz gráfica desarrollada en Swing mejora la experiencia del usuario al facilitar la entrada de datos y la visualización de los resultados en tiempo real.

# Reporte del Programa Ejercicio2

## 1. Descripción del Programa

El programa "Ejercicio2" muestra los números del 1 al 20 con un retardo de 1500 milisegundos entre cada número. Se ha mejorado el programa original agregando una interfaz gráfica de usuario (GUI) utilizando Swing, lo que facilita su uso y mejora la presentación de los resultados.

## 2. Objetivos

- **Mostrar Números:** Imprimir los números del 1 al 20 con un retardo de 1500 milisegundos.
- **Interfaz Gráfica:** Proporcionar una GUI para iniciar el conteo y mostrar los números.

## 3. Componentes del Programa

### 3.1. Interfaz Gráfica (GUI)

- **JFrame:** Ventana principal del programa.
- **TextArea:** Área de texto para mostrar los números.
- **Button:** Botón para iniciar el conteo.
- **ScrollPane:** Panel de desplazamiento para la TextArea.

### 3.2. Clases

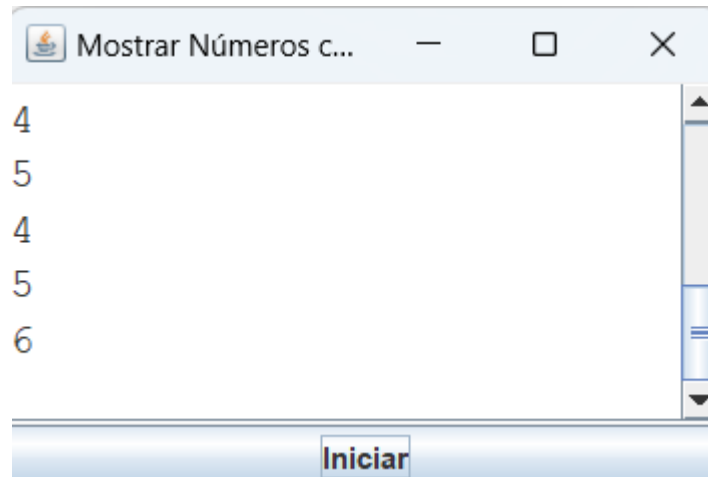
- **Ejercicio2:** Clase principal que extiende JFrame y contiene la lógica de la GUI.
- **ContarNumeros:** Clase interna que extiende Thread y se encarga de contar y mostrar los números con el retardo especificado.

## 4. Funcionamiento del Programa

1. **Inicialización:** Al ejecutar el programa, se inicializa la ventana principal Ejercicio2 con los componentes de la interfaz gráfica.
2. **Iniciar Conteo:** Al presionar el botón "Iniciar", se crea y ejecuta un hilo (ContarNumeros) para manejar el conteo de los números.
3. **Mostrar Números:** El hilo cuenta del 1 al 20, mostrando cada número en el área de texto con un retardo de 1500 milisegundos.

4. **Actualización en Tiempo Real:** Los números se muestran en tiempo real en la interfaz gráfica.

## 5. Resultado



## 6. Conclusión

El programa "Ejercicio2" ofrece una solución intuitiva y eficiente para mostrar los números del 1 al 20 con un retardo de tiempo. La implementación de una interfaz gráfica mejora la experiencia del usuario y proporciona una forma clara y organizada de visualizar los resultados. La utilización de hilos permite mantener la interfaz gráfica receptiva mientras se realiza el conteo en segundo plano.

# Reporte del Programa Ejercicio3

## 1. Descripción del Programa

El programa "Ejercicio3" permite ingresar el nombre de un atleta y simular que corre una carrera de 30 km. La simulación muestra el progreso del atleta cada kilómetro con un retardo de 3500 milisegundos entre cada kilómetro. Se ha mejorado el programa original agregando una interfaz gráfica de usuario (GUI) utilizando Swing.

## 2. Objetivos

- **Simular Carrera:** Imprimir el progreso del atleta cada kilómetro con un retardo de 3500 milisegundos hasta completar 30 km.
- **Interfaz Gráfica:** Proporcionar una GUI para ingresar el nombre del atleta y mostrar el progreso de la carrera.

## 3. Componentes del Programa

### 3.1. Interfaz Gráfica (GUI)

- **JFrame:** Ventana principal del programa.
- **JPanel:** Panel para organizar los componentes de la GUI en una cuadrícula de 2 filas y 2 columnas.
- **JLabel:** Etiqueta para el campo de entrada del nombre del atleta.
- **TextField:** Campo de texto para ingresar el nombre del atleta.
- **Button:** Botón para iniciar la carrera.
- **TextArea:** Área de texto para mostrar el progreso de la carrera.
- **ScrollPane:** Panel de desplazamiento para la JTextArea.

### 3.2. Clases

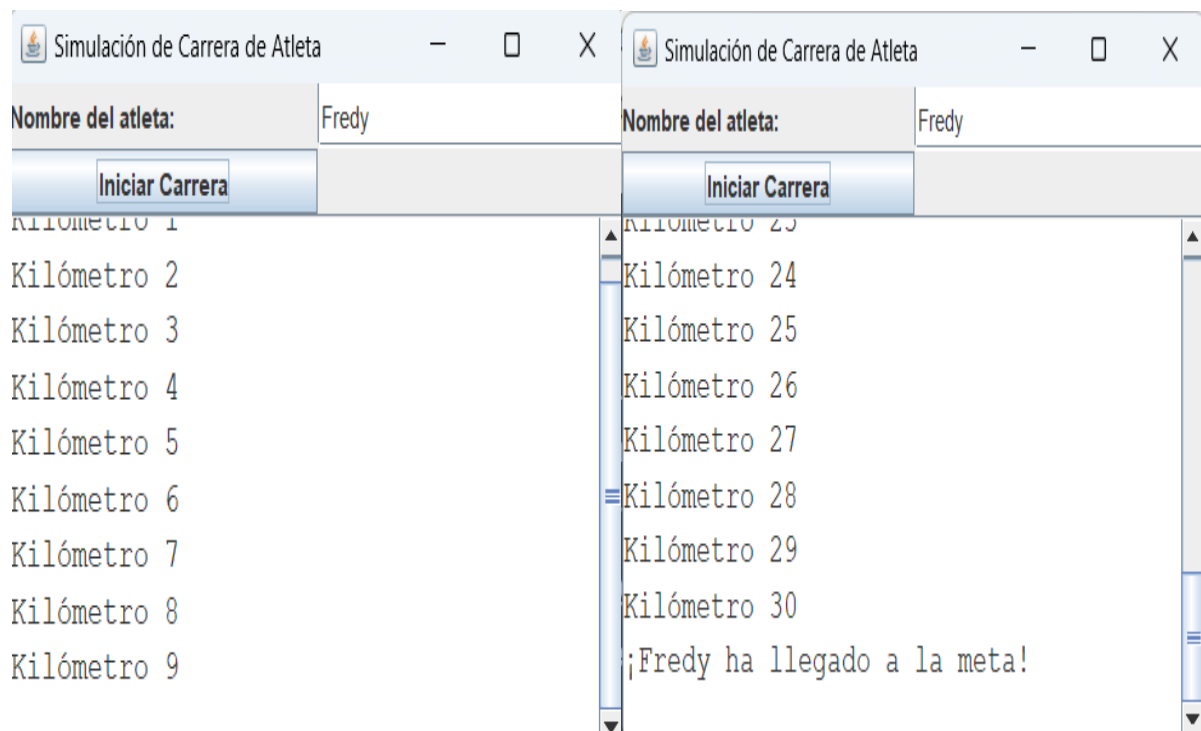
- **Ejercicio3:** Clase principal que extiende JFrame y contiene la lógica de la GUI.
- **CarreraAtleta:** Clase interna que extiende Thread y se encarga de simular la carrera del atleta.

## 4. Funcionamiento del Programa

1. **Inicialización:** Al ejecutar el programa, se inicializa la ventana principal Ejercicio3 con los componentes de la interfaz gráfica.
2. **Ingresar Datos:** El usuario ingresa el nombre del atleta en el campo de texto.

3. **Iniciar Carrera:** Al presionar el botón "Iniciar Carrera", se crea y ejecuta un hilo (CarreraAtleta) para manejar la simulación de la carrera.
4. **Simulación de Carrera:** El hilo cuenta los kilómetros del 1 al 30, mostrando cada kilómetro en el área de texto con un retardo de 3500 milisegundos.
5. **Mostrar Resultados:** Al completar los 30 km, se muestra un mensaje indicando que el atleta ha llegado a la meta.

## 5. Resultado



## 6. Conclusión

El programa "Ejercicio3" ofrece una solución intuitiva y eficiente para simular la carrera de un atleta que corre 30 km. La implementación de una interfaz gráfica mejora la experiencia del usuario y proporciona una forma clara y organizada de visualizar el progreso y los resultados de la carrera. La utilización de hilos permite mantener la interfaz gráfica receptiva mientras se realiza la simulación en segundo plano.

# Reporte del Programa Ejercicio4

## 1. Descripción del Programa

El programa "Ejercicio4" permite ingresar los nombres de tres mascotas y simular una carrera de 30 metros, indicando quién llegó a la meta primero. La simulación muestra el progreso de cada mascota a lo largo de la carrera con tiempos de avance aleatorios.

## 2. Objetivos

- **Simular Carrera:** Imprimir el progreso de cada mascota cada metro con un retardo aleatorio hasta completar 30 metros.
- **Interfaz Gráfica:** Proporcionar una GUI para ingresar los nombres de las mascotas y mostrar el progreso de la carrera.

## 3. Componentes del Programa

### 3.1. Interfaz Gráfica (GUI)

- **JFrame:** Ventana principal del programa.
- **JPanel:** Panel para organizar los componentes de la GUI en una cuadrícula de 4 filas y 2 columnas.
- **JLabels:** Etiquetas para los campos de entrada de los nombres de las mascotas.
- **JTextFields:** Campos de texto para ingresar los nombres de las mascotas.
- **JButton:** Botón para iniciar la carrera.
- **JTextArea:** Área de texto para mostrar el progreso de la carrera.
- **JScrollPane:** Panel de desplazamiento para la JTextArea.

### 3.2. Clases

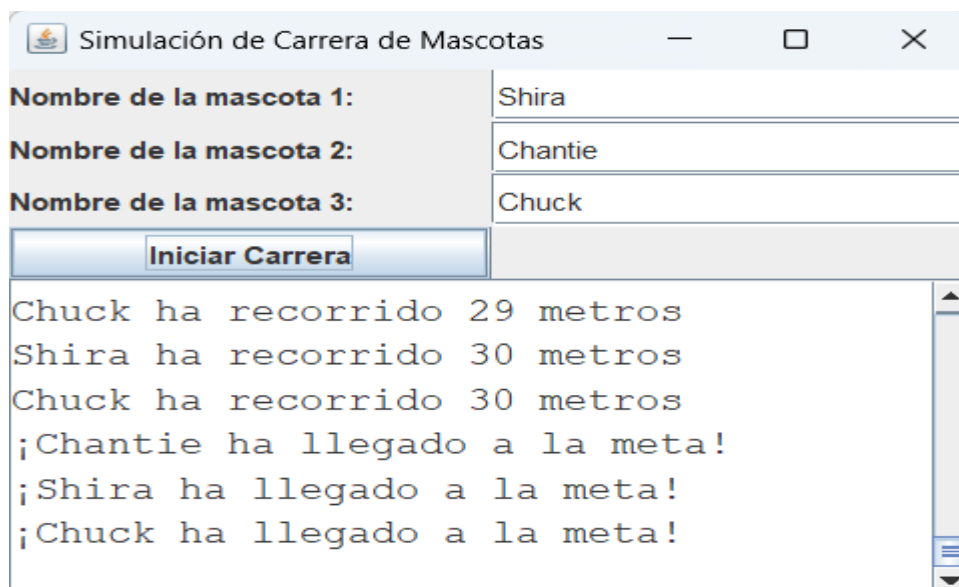
- **Ejercicio4:** Clase principal que extiende JFrame y contiene la lógica de la GUI.
- **CarreraMascota:** Clase interna que extiende Thread y se encarga de simular la carrera de cada mascota.



#### 4. Funcionamiento del Programa

1. **Inicialización:** Al ejecutar el programa, se inicializa la ventana principal Ejercicio4 con los componentes de la interfaz gráfica.
2. **Ingresar Datos:** El usuario ingresa los nombres de las tres mascotas en los campos de texto.
3. **Iniciar Carrera:** Al presionar el botón "Iniciar Carrera", se crean y ejecutan tres hilos (CarreraMascota) para manejar la simulación de la carrera de cada mascota.
4. **Simulación de Carrera:** Cada hilo cuenta los metros del 1 al 30, mostrando el progreso en el área de texto con un retardo aleatorio.
5. **Mostrar Resultados:** Al completar los 30 metros, se muestra un mensaje indicando que la mascota ha llegado a la meta.

#### 5. Resultado



#### 6. Conclusión

El programa "Ejercicio4" ofrece una solución intuitiva y eficiente para simular la carrera de tres mascotas que corren 30 metros. La implementación de una interfaz gráfica mejora la experiencia del usuario y proporciona una forma clara y organizada de visualizar el progreso y los resultados de la carrera. La utilización de hilos permite mantener la interfaz gráfica receptiva mientras se realiza la simulación en segundo plano. La adición de tiempos de sueño aleatorios hace que la carrera sea más realista y emocionante.

# Reporte del Programa Ejercicio5

## 1. Descripción del Programa

El programa "Ejercicio5" utiliza hilos independientes en clases diferentes para imprimir números pares del 1 al 10 y la suma de dichos números, y otro hilo que imprime números impares del 1 al 10 y su suma respectiva. La interfaz gráfica desarrollada en Swing mejora la experiencia del usuario al proporcionar una forma clara y organizada de visualizar los resultados.

## 2. Objetivos

- **Imprimir Números Pares e Impares:** Imprimir números pares e impares del 1 al 10 y sus sumas respectivas.
- **Interfaz Gráfica:** Proporcionar una GUI para iniciar los hilos y mostrar los resultados.

## 3. Componentes del Programa

### 3.1. Interfaz Gráfica (GUI)

- **JFrame:** Ventana principal del programa.
- **JTextArea:** Área de texto para mostrar los números y las sumas.
- **JButton:** Botón para iniciar los hilos.
- **JScrollPane:** Panel de desplazamiento para la JTextArea.

### 3.2. Clases

- **Ejercicio5:** Clase principal que extiende JFrame y contiene la lógica de la GUI.
- **HiloPares:** Clase interna que extiende Thread y se encarga de imprimir los números pares y su suma.
- **HiloImpares:** Clase interna que extiende Thread y se encarga de imprimir los números impares y su suma.

## 4. Funcionamiento del Programa

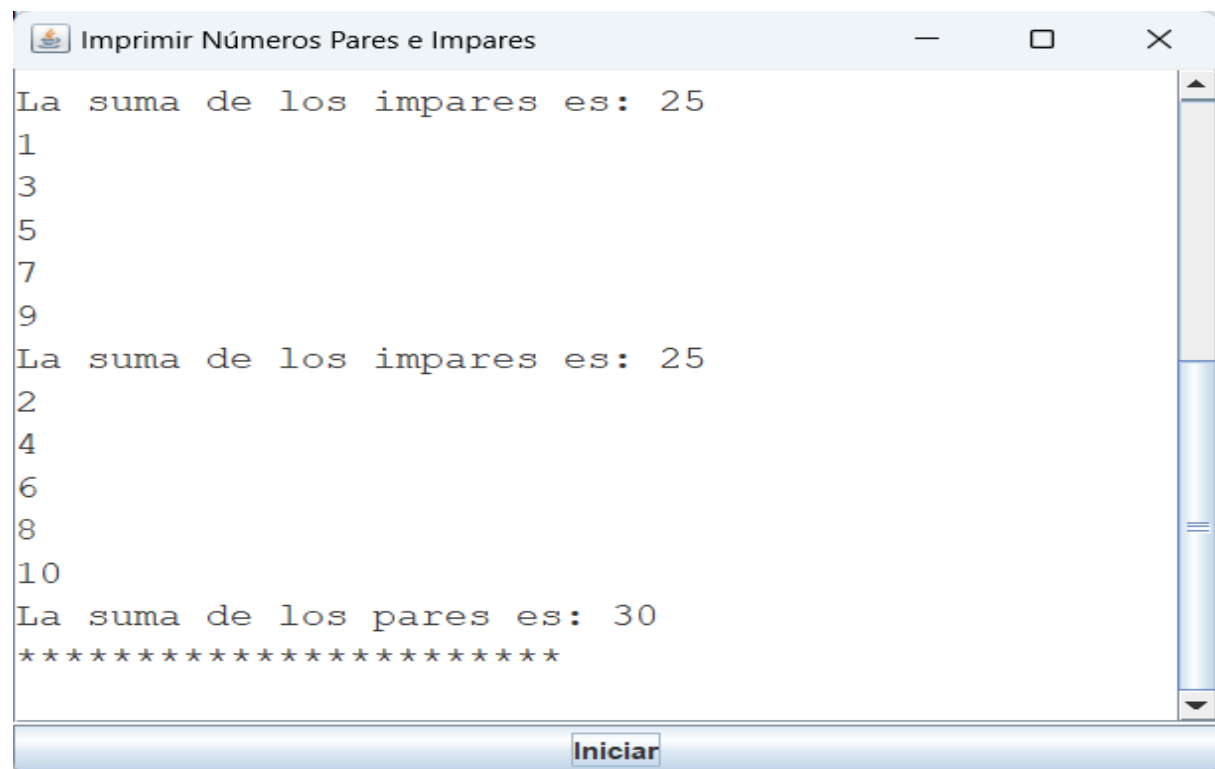
**Inicialización:** Al ejecutar el programa, se inicializa la ventana principal Ejercicio5 con los componentes de la interfaz gráfica.

**Iniciar Hilos:** Al presionar el botón "Iniciar", se crean y ejecutan cuatro hilos (dos de HiloPares y dos de HiloImpares) para manejar la impresión de números pares e impares y sus sumas respectivas.

**Imprimir Resultados:** Los hilos imprimen los números pares e impares y sus sumas respectivas en el área de texto.

**Actualización en Tiempo Real:** Los resultados se muestran en tiempo real en la interfaz gráfica.

## 5. Resultado



## 6. Conclusión

El programa "Ejercicio5" ofrece una solución intuitiva y eficiente para imprimir números pares e impares del 1 al 10 y sus sumas respectivas. La implementación de una interfaz gráfica mejora la experiencia del usuario y proporciona una forma clara y organizada de visualizar los resultados. La utilización de hilos permite mantener la interfaz gráfica receptiva mientras se realizan los cálculos en segundo plano.

# Reporte del Programa Ejercicio6

## 1. Descripción del Programa

El programa "Ejercicio6" permite seleccionar una estación del año de un arreglo y muestra los meses correspondientes a esa estación con un retardo de 1000 milisegundos entre cada mes. La interfaz gráfica desarrollada en Swing mejora la experiencia del usuario al proporcionar una forma clara y organizada de visualizar los resultados.

## 2. Objetivos

- **Mostrar Estaciones y Meses:** Imprimir los meses correspondientes a una estación seleccionada con un retardo de 1000 milisegundos entre cada mes.
- **Interfaz Gráfica:** Proporcionar una GUI para seleccionar una estación y mostrar los meses correspondientes.

## 3. Componentes del Programa

### 3.1. Interfaz Gráfica (GUI)

- **JFrame:** Ventana principal del programa.
- **JComboBox:** Combo box para seleccionar una estación.
- **JTextArea:** Área de texto para mostrar los meses.
- **JButton:** Botón para iniciar la visualización de los meses.
- **JScrollPane:** Panel de desplazamiento para la JTextArea.
- **JPanel:** Panel para organizar los componentes de la GUI.

### 3.2. Clases

- **Ejercicio6:** Clase principal que extiende JFrame y contiene la lógica de la GUI.
- **MostrarMesesRunnable:** Clase interna que implementa Runnable y se encarga de mostrar los meses correspondientes a la estación seleccionada.

## 4. Funcionamiento del Programa

**Inicialización:** Al ejecutar el programa, se inicializa la ventana principal Ejercicio6 con los componentes de la interfaz gráfica.

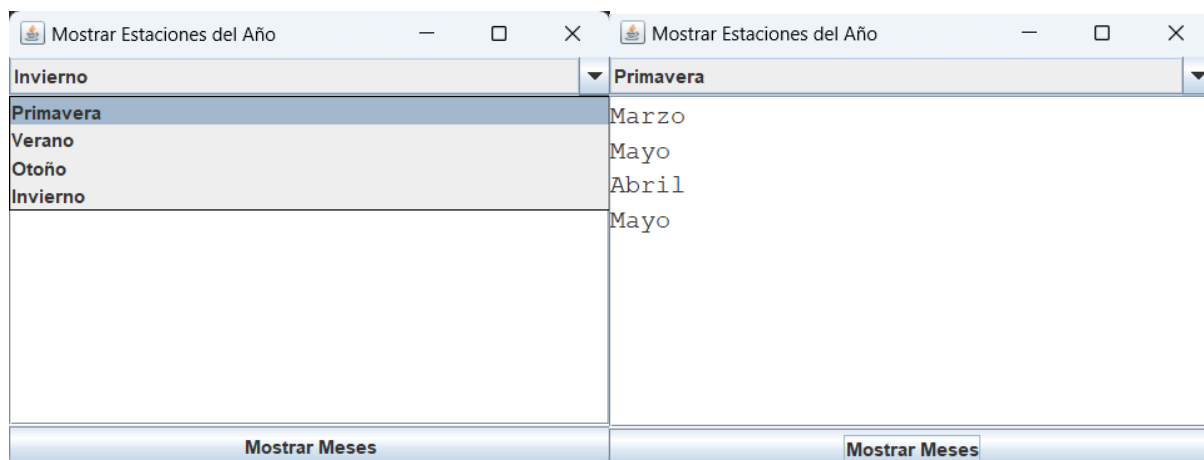
**Seleccionar Estación:** El usuario selecciona una estación del combo box.

**Mostrar Meses:** Al presionar el botón "Mostrar Meses", se crea y ejecuta un hilo (MostrarMesesRunnable) para manejar la visualización de los meses correspondientes.

**Imprimir Resultados:** El hilo imprime los meses correspondientes a la estación seleccionada en el área de texto con un retardo de 1000 milisegundos entre cada mes.

**Actualización en Tiempo Real:** Los resultados se muestran en tiempo real en la interfaz gráfica.

## 5. Resultado



## 6. Conclusión

El programa "Ejercicio6" ofrece una solución intuitiva y eficiente para mostrar los meses correspondientes a una estación del año seleccionada. La implementación de una interfaz gráfica mejora la experiencia del usuario y proporciona una forma clara y organizada de visualizar los resultados. La utilización de hilos permite mantener la interfaz gráfica receptiva mientras se realiza la visualización de los meses en segundo plano.

# Reporte del Programa Ejercicio7

## 1. Descripción del Programa

El programa "Ejercicio7" permite simular que 10 nombres de los pitufos comen pan con queso (3 veces cada uno) y muestra quién termina de comer. La interfaz gráfica desarrollada en Swing mejora la experiencia del usuario al proporcionar una forma clara y organizada de visualizar el progreso y los resultados.

## 2. Objetivos

- **Simular Pitufos Comiendo:** Mostrar el progreso de los pitufos comiendo pan con queso y quién termina de comer.
- **Interfaz Gráfica:** Proporcionar una GUI para visualizar el progreso y los resultados de la simulación.

## 3. Componentes del Programa

### 3.1. Interfaz Gráfica (GUI)

- **JFrame:** Ventana principal del programa.
- **TextArea:** Área de texto para mostrar el progreso y los resultados.
- **Button:** Botón para iniciar la simulación.
- **ScrollPane:** Panel de desplazamiento para la TextArea.

### 3.2. Clases

- **Ejercicio7:** Clase principal que extiende JFrame y contiene la lógica de la GUI.
- **Pitufito:** Clase interna que implementa Runnable y se encarga de simular a cada pitufo comiendo pan con queso.

## 4. Funcionamiento del Programa

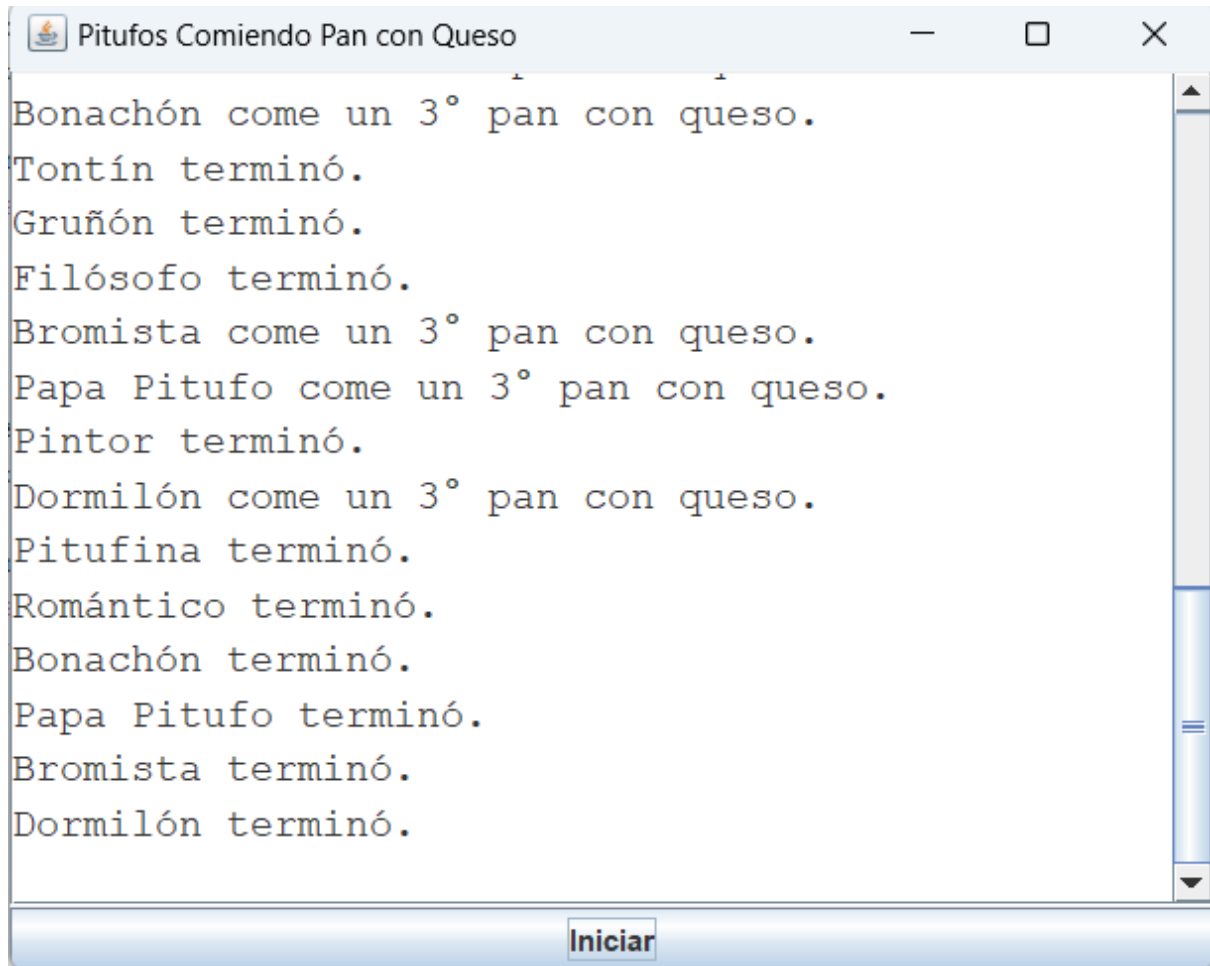
**Inicialización:** Al ejecutar el programa, se inicializa la ventana principal Ejercicio7 con los componentes de la interfaz gráfica.

**Iniciar Simulación:** Al presionar el botón "Iniciar", se crean y ejecutan hilos para cada uno de los 10 pitufos, simulando que comen pan con queso.

**Simulación de Progreso:** Cada pitufo come tres veces, con un tiempo de sueño aleatorio entre cada comida para simular diferentes velocidades.

**Mostrar Resultados:** El progreso y el resultado de cada pitufo se muestran en tiempo real en el área de texto de la interfaz gráfica.

## 5. Resultado



## 6. Conclusión

El programa "Ejercicio7" ofrece una solución intuitiva y eficiente para simular que los pitufos comen pan con queso y mostrar quién termina de comer. La implementación de una interfaz gráfica mejora la experiencia del usuario y proporciona una forma clara y organizada de visualizar el progreso y los resultados. La utilización de hilos permite mantener la interfaz gráfica receptiva mientras se realiza la simulación en segundo plano. La adición de tiempos de sueño aleatorios hace que la simulación sea más realista y divertida.

# Reporte del Programa Ejercicio 8

## 1. Descripción del Programa

El programa "Ejercicio8" permite simular el proceso del ascensor, donde los usuarios suben y bajan de un piso con una escala de tiempo. La interfaz gráfica desarrollada en Swing mejora la experiencia del usuario al proporcionar una forma clara y organizada de visualizar el progreso del ascensor.

## 2. Objetivos

- **Simular Ascensor:** Mostrar el progreso del ascensor subiendo y bajando entre pisos con una escala de tiempo.
- **Interfaz Gráfica:** Proporcionar una GUI para visualizar el progreso del ascensor.

## 3. Componentes del Programa

### 3.1. Interfaz Gráfica (GUI)

- **JFrame:** Ventana principal del programa.
- **TextArea:** Área de texto para mostrar el progreso del ascensor.
- **Button:** Botón para iniciar la simulación.
- **ScrollPane:** Panel de desplazamiento para la TextArea.

### 3.2. Clases

- **Ejercicio8:** Clase principal que extiende JFrame y contiene la lógica de la GUI.
- **Ascensor:** Clase interna que implementa Runnable y se encarga de simular el ascensor subiendo y bajando entre pisos.

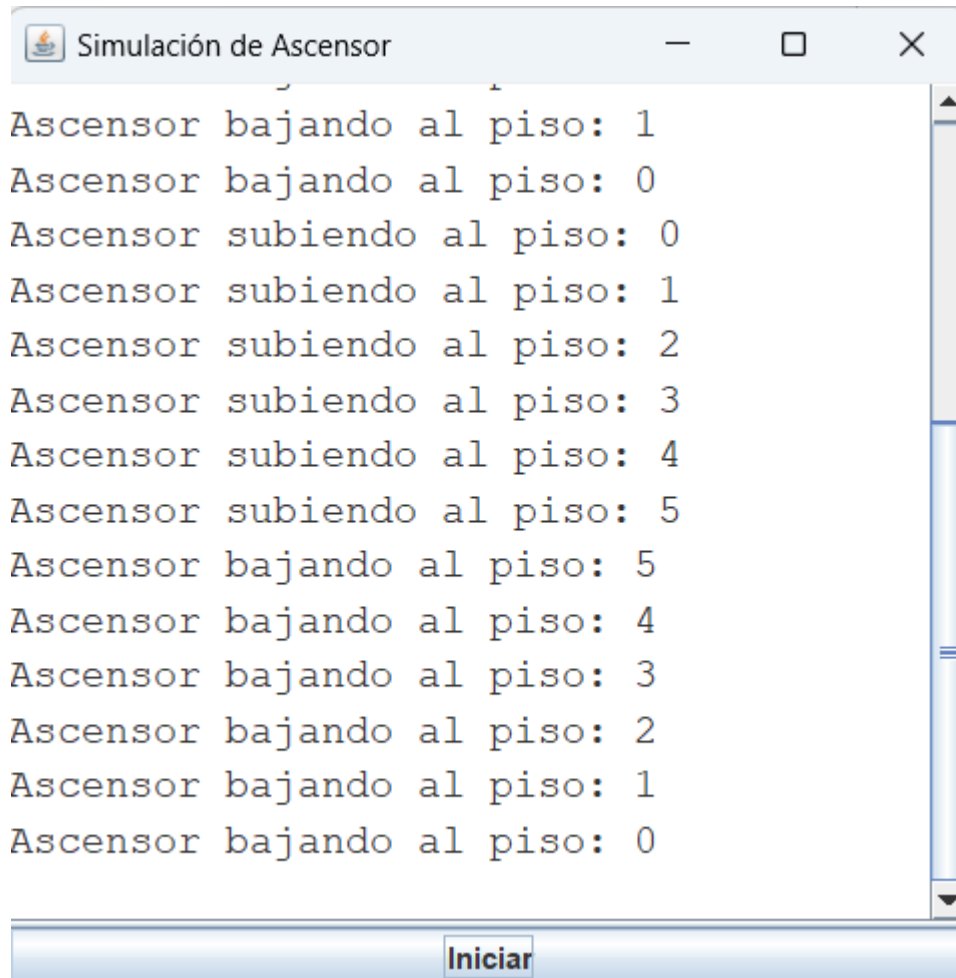
## 4. Funcionamiento del Programa

1. **Inicialización:** Al ejecutar el programa, se inicializa la ventana principal Ejercicio8 con los componentes de la interfaz gráfica.
2. **Iniciar Simulación:** Al presionar el botón "Iniciar", se crea y ejecuta un hilo (Ascensor) para manejar la simulación del ascensor subiendo y bajando entre pisos.



3. **Simulación de Progreso:** El ascensor sube y baja entre pisos con un tiempo de sueño de 1000 milisegundos para simular el tiempo que tarda en moverse entre pisos.
4. **Mostrar Resultados:** El progreso del ascensor se muestra en tiempo real en el área de texto de la interfaz gráfica.

## 5. Resultado



## 6. Conclusión

El programa "Ejercicio8" ofrece una solución intuitiva y eficiente para simular el proceso del ascensor, donde los usuarios suben y bajan de un piso con una escala de tiempo. La implementación de una interfaz gráfica mejora la experiencia del usuario y proporciona una forma clara y organizada de visualizar el progreso del ascensor. La utilización de hilos permite mantener la interfaz gráfica receptiva mientras se realiza la simulación en segundo plano.

# Reporte del Programa Ejercicio9

## 1. Descripción del Programa

El programa "Ejercicio9" permite ingresar los nombres de los usuarios que inician sesión e indicar el tiempo de sesión activa. La interfaz gráfica desarrollada en Swing mejora la experiencia del usuario al proporcionar una forma clara y organizada de visualizar el tiempo de sesión activa.

## 2. Objetivos

- **Indicar Tiempo de Sesión Activa:** Mostrar el tiempo de sesión activa de cada usuario que inicia sesión.
- **Interfaz Gráfica:** Proporcionar una GUI para ingresar los nombres de los usuarios y visualizar el tiempo de sesión activa.

## 3. Componentes del Programa

### 3.1. Interfaz Gráfica (GUI)

- **JFrame:** Ventana principal del programa.
- **TextField:** Campo de texto para ingresar el nombre del usuario.
- **TextArea:** Área de texto para mostrar el tiempo de sesión activa.
- **Button:** Botón para iniciar la sesión del usuario.
- **ScrollPane:** Panel de desplazamiento para la TextArea.
- **Panel:** Panel para organizar los componentes de la GUI.

### 3.2. Clases

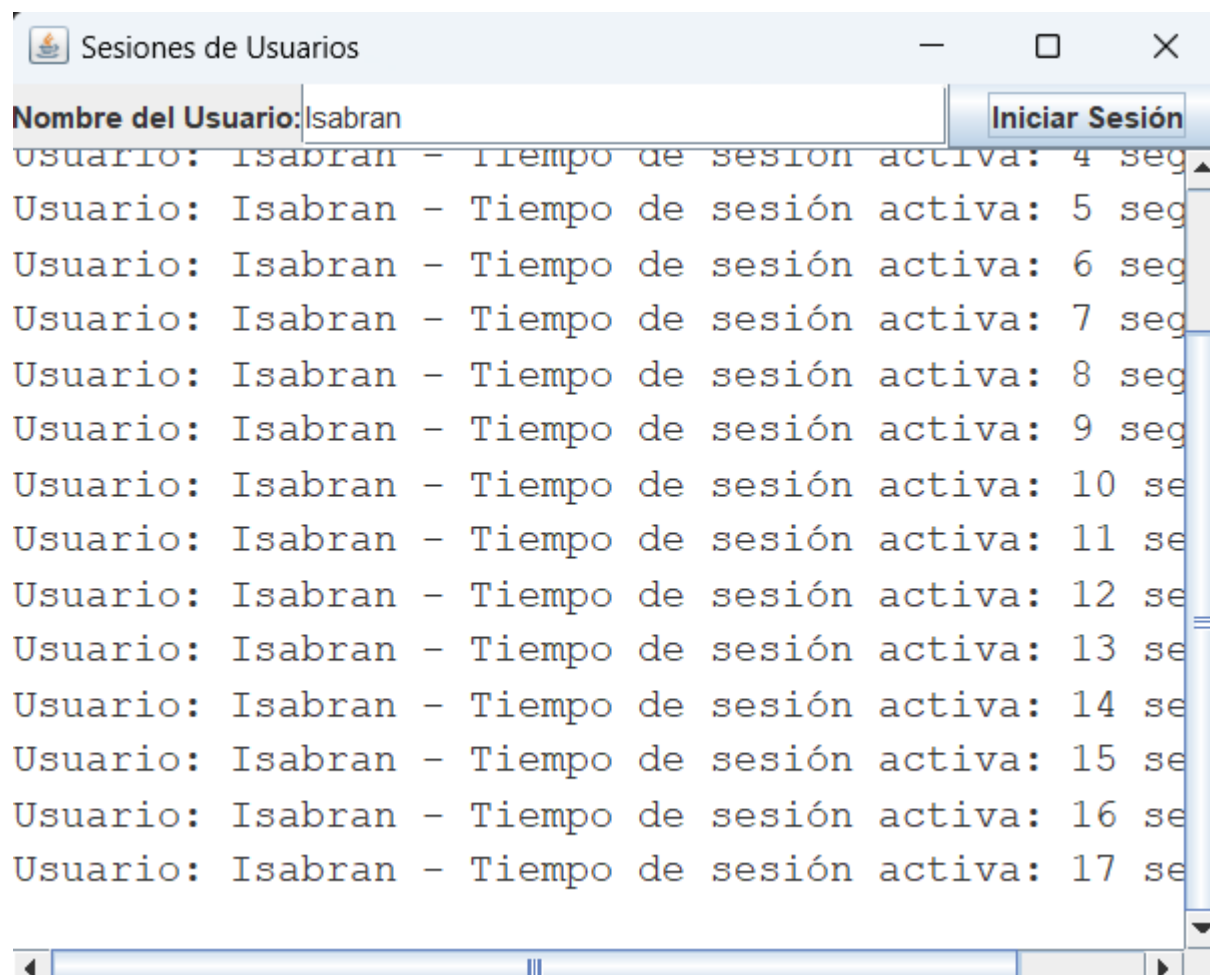
- **Ejercicio9:** Clase principal que extiende JFrame y contiene la lógica de la GUI.
- **SesionUsuario:** Clase interna que implementa Runnable y se encarga de manejar la sesión de cada usuario.

## 4. Funcionamiento del Programa

1. **Inicialización:** Al ejecutar el programa, se inicializa la ventana principal Ejercicio9 con los componentes de la interfaz gráfica.
2. **Iniciar Sesión:** Al presionar el botón "Iniciar Sesión", se crea y ejecuta un hilo (SesionUsuario) para manejar la sesión de cada usuario ingresado.

3. **Simulación de Progreso:** El tiempo de sesión activa se incrementa cada segundo y se muestra en tiempo real en el área de texto de la interfaz gráfica.
4. **Mostrar Resultados:** El progreso y el resultado del tiempo de sesión activa se muestran en tiempo real en la interfaz gráfica.

## 5. Resultado



## 6. Conclusión

El programa "Ejercicio9" ofrece una solución intuitiva y eficiente para ingresar los nombres de los usuarios que inician sesión e indicar el tiempo de sesión activa. La implementación de una interfaz gráfica mejora la experiencia del usuario y proporciona una forma clara y organizada de visualizar el tiempo de sesión activa. La utilización de hilos permite mantener la interfaz gráfica receptiva mientras se realiza la simulación en segundo plano.

# Reporte del Programa Ejercicio10

## 1. Descripción del Programa

El programa "Ejercicio10" permite simular a los pitufos corriendo y a Azrael atrapándolos uno a uno. La interfaz gráfica desarrollada en Swing mejora la experiencia del usuario al proporcionar una forma clara y organizada de visualizar el progreso y los resultados de la simulación.

## 2. Objetivos

- **Simular Pitufos Corriendo y Azrael Atrapándolos:** Mostrar el progreso de los pitufos corriendo y Azrael atrapándolos uno a uno.
- **Interfaz Gráfica:** Proporcionar una GUI para visualizar el progreso de la simulación.

## 3. Componentes del Programa

### 3.1. Interfaz Gráfica (GUI)

- **JFrame:** Ventana principal del programa.
- **TextArea:** Área de texto para mostrar el progreso de los pitufos y Azrael.
- **Button:** Botón para iniciar la simulación.
- **ScrollPane:** Panel de desplazamiento para la TextArea.

### 3.2. Clases

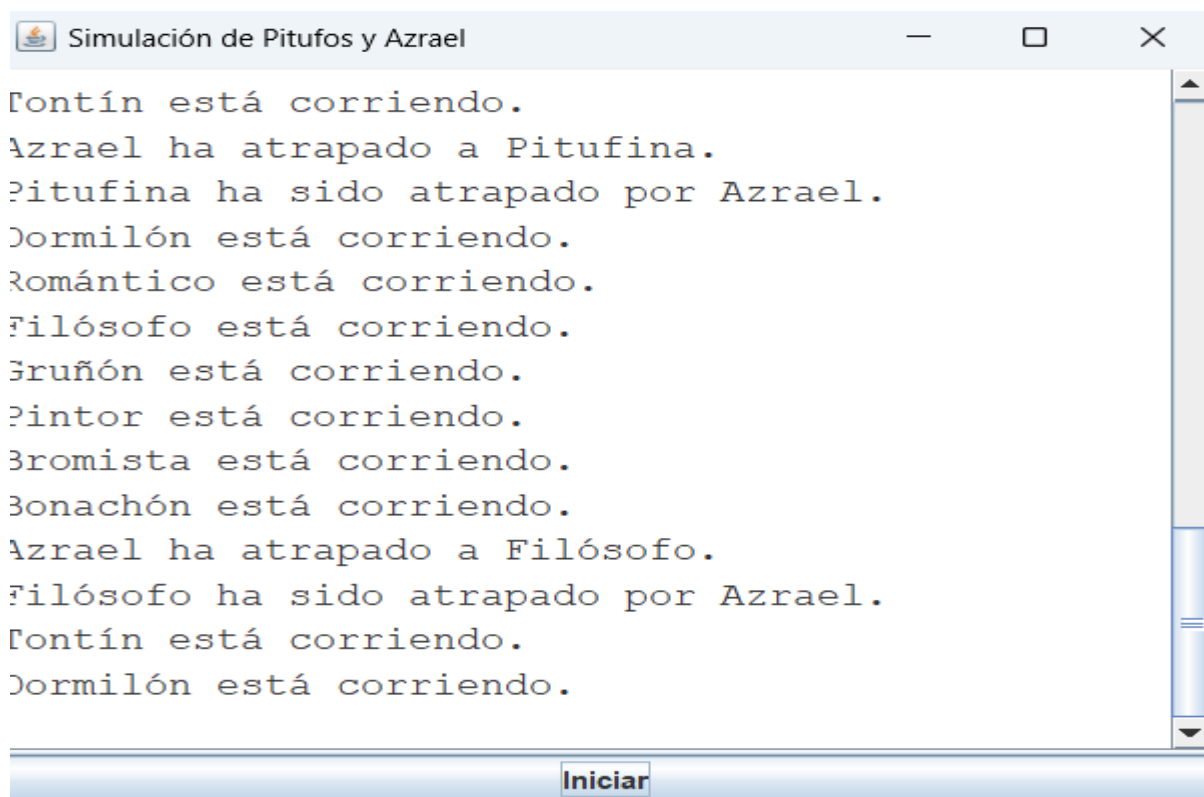
- **Ejercicio10:** Clase principal que extiende JFrame y contiene la lógica de la GUI.
- **Pitufos:** Clase interna que implementa Runnable y se encarga de simular a los pitufos corriendo.
- **Azrael:** Clase interna que implementa Runnable y se encarga de simular a Azrael atrapando a los pitufos.

## 4. Funcionamiento del Programa

1. **Inicialización:** Al ejecutar el programa, se inicializa la ventana principal Ejercicio10 con los componentes de la interfaz gráfica.

2. **Iniciar Simulación:** Al presionar el botón "Iniciar", se crean y ejecutan hilos para cada uno de los pitufos, simulando que corren, y un hilo para Azrael, simulando que los atrapa.
3. **Simulación de Progreso:** Los pitufos corren con un tiempo de sueño aleatorio entre cada acción para simular diferentes velocidades. Azrael atrapa a los pitufos uno a uno con un tiempo de sueño aleatorio entre cada captura.
4. **Mostrar Resultados:** El progreso y el resultado de la simulación se muestran en tiempo real en el área de texto de la interfaz gráfica.

## 5. Resultado



## 6. Conclusión

El programa "Ejercicio10" ofrece una solución intuitiva y eficiente para simular a los pitufos corriendo y a Azrael atrapándolos uno a uno. La implementación de una interfaz gráfica mejora la experiencia del usuario y proporciona una forma clara y organizada de visualizar el progreso y los resultados de la simulación. La utilización de hilos permite mantener la interfaz gráfica receptiva mientras se realiza la simulación en segundo plano. La adición de tiempos de sueño aleatorios hace que la simulación sea más realista y divertida.