

**UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**

**FACULTAD DE INGENIERIA**

**ESCUELA DE CIENCIAS Y SISTEMAS**

**INTRODUCCIÓN A LA PROGRAMACIÓN Y COMPUTACIÓN 1**

**SECCIÓN E**

**SEGUNDO SEMESTRE 2022**

**ING. NEFTALÍ DE JESÚS ALDANA CALDERÓN**

**AUX. SERGIO FERNANDO OTZOY GONZALEZ**



**SNAKE**

## CONTENIDO

Objetivos .....	2
General .....	2
Específicos .....	2
Descripción.....	3
Flujo de la aplicación .....	3
Ventana .....	3
Juego .....	4
Frutas .....	6
Velocidad de la serpiente .....	6
Fin del juego .....	7
Consideraciones.....	9
Restricciones.....	9
Entregables .....	10

## OBJETIVOS

### GENERAL

- Usar hilos, algoritmos y programación orientada a objetos para desarrollar una solución informática compleja con el lenguaje de programación Java

### ESPECÍFICOS

- Desarrollar y mejorar la capacidad de comprensión y resolución de problemas a través del uso de algoritmos
- Construir una aplicación de ventanas usando el lenguaje de programación Java como herramienta de desarrollo de software
- Comprender la ejecución de procesos en paralelo implementando hilos en java.



## DESCRIPCIÓN

En esta práctica el estudiante deberá desarrollar el juego Snake en java. El juego consiste en controlar una criatura con forma de serpiente que se mueve en un área delimitada, recolectando fruta y evitando chocar con las paredes que rodean el área de juego. Cada vez que la serpiente se come una fruta, su cuerpo crecerá.

Las reglas del juego son las siguientes:

- Una vez iniciado el movimiento, el jugador no podrá detener la marcha de la serpiente.
- Cada vez que la serpiente se coma una fruta, su cuerpo crecerá.
- Cada vez que la serpiente se coma una fruta, la velocidad de movimiento aumentará.
- No se podrá iniciar un nuevo juego si hay uno activo.

La única forma de perder este juego es si la serpiente choca con una pared. El juego **no** se perderá si la serpiente se topa consigo misma.

La única forma de ganar este juego es alcanzar el largo máximo de la serpiente: **25 espacios**.

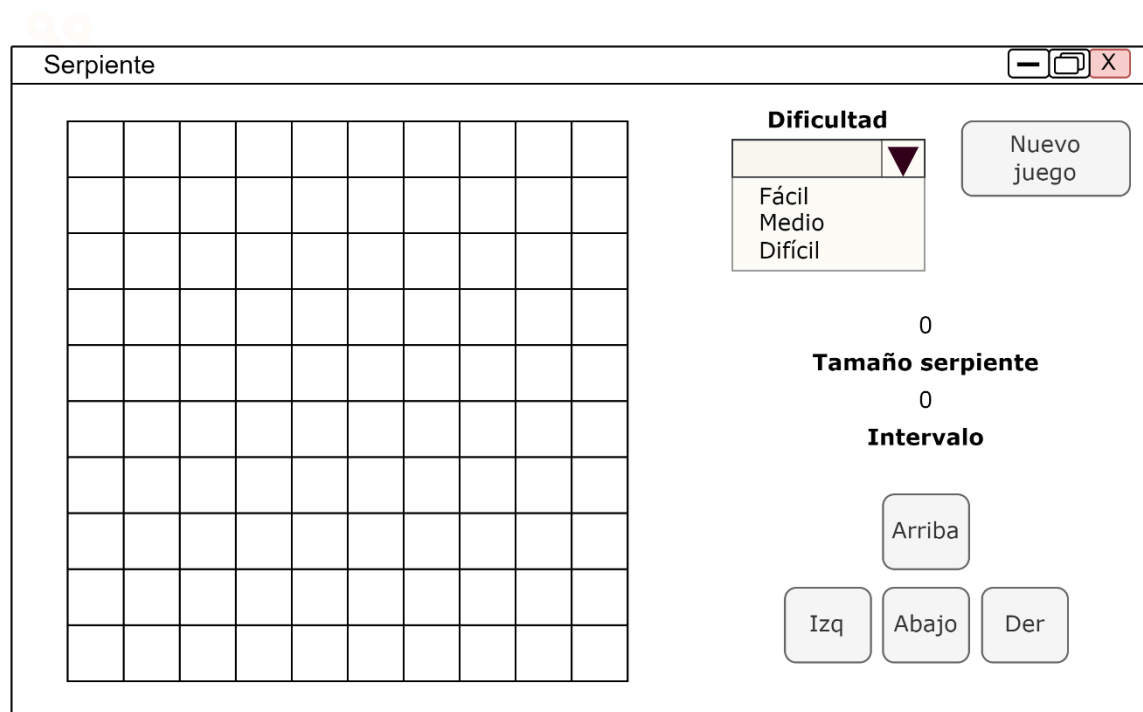
## FLUJO DE LA APLICACIÓN

### VENTANA

Al iniciar la aplicación se visualizar una ventana en donde se deberá visualizar:

- Un control que permita seleccionar la dificultad del juego:
  - Fácil
  - Medio
  - Difícil
- Un botón que permita iniciar un nuevo juego con la dificultad seleccionada
- Una etiqueta que indique y que muestre el tamaño actual de la serpiente. Conforme la serpiente vaya creciendo el valor de la etiqueta deberá crecer también.

- Una etiqueta que indique y que muestre el intervalo de movimiento de la serpiente. Conforme la serpiente vaya creciendo el valor de la etiqueta deberá disminuir (ya que se moverá más rápido).
- Cuatro botones que permitan controlar la serpiente:
  - Botón para que se mueva hacia arriba
  - Botón para que se mueva hacia abajo
  - Botón para que se mueva hacia la izquierda
  - Botón para que se mueva hacia la derecha
- El tablero del juego
  - El tablero consistirá en una matriz de controles que permitan el cambio de color de fondo (setBackground). Se recomienda usar etiquetas (JLabel).
  - El tamaño será de 10 x 10.



## JUEGO

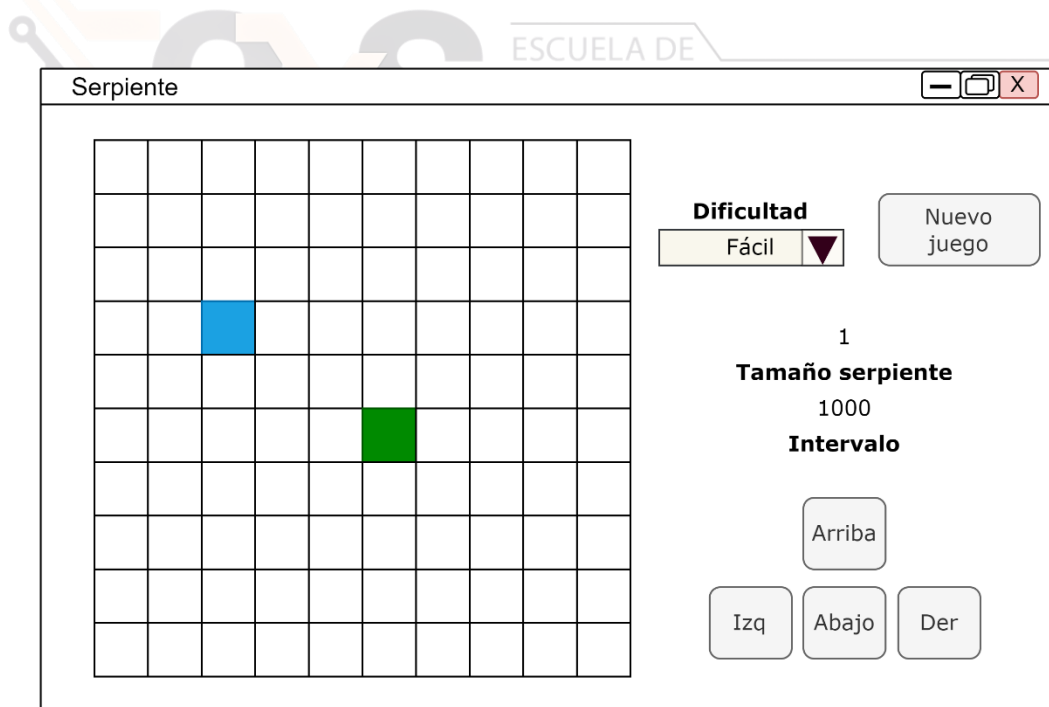
Para iniciar un nuevo juego, se deberá seleccionar la dificultad y luego presionar el botón que permite iniciar un nuevo juego.

Al iniciar el juego, dos casillas del tablero deberán cambiar de color. Una casilla representará el cuerpo de la serpiente y otra representará una fruta. La posición inicial de la serpiente queda a discreción del estudiante, pero se recomienda colocarla en el centro del tablero. El posicionamiento de las frutas se discutirá más adelante.

Luego de iniciar el juego, la serpiente no se deberá mover hasta que el usuario le indique en qué dirección moverse con los botones de control. Una vez esto suceda, el movimiento de la serpiente no se deberá detener a menos que choque con alguna pared del tablero de juego.

En cualquier momento y mientras la partida esté activa, el usuario podrá cambiar la dirección de movimiento con los botones de control.

Cuando la serpiente “choque” o se coma una fruta, su cuerpo deberá crecer y la velocidad de movimiento deberá aumentar.



El color de la serpiente y de las frutas queda a discreción del estudiante, pero deberán ser distinguibles entre sí y del color de fondo del tablero.

---

## FRUTAS

Todas las frutas se deberán generar en coordenadas aleatorias del tablero. No darán punteo, únicamente harán crecer a la serpiente.

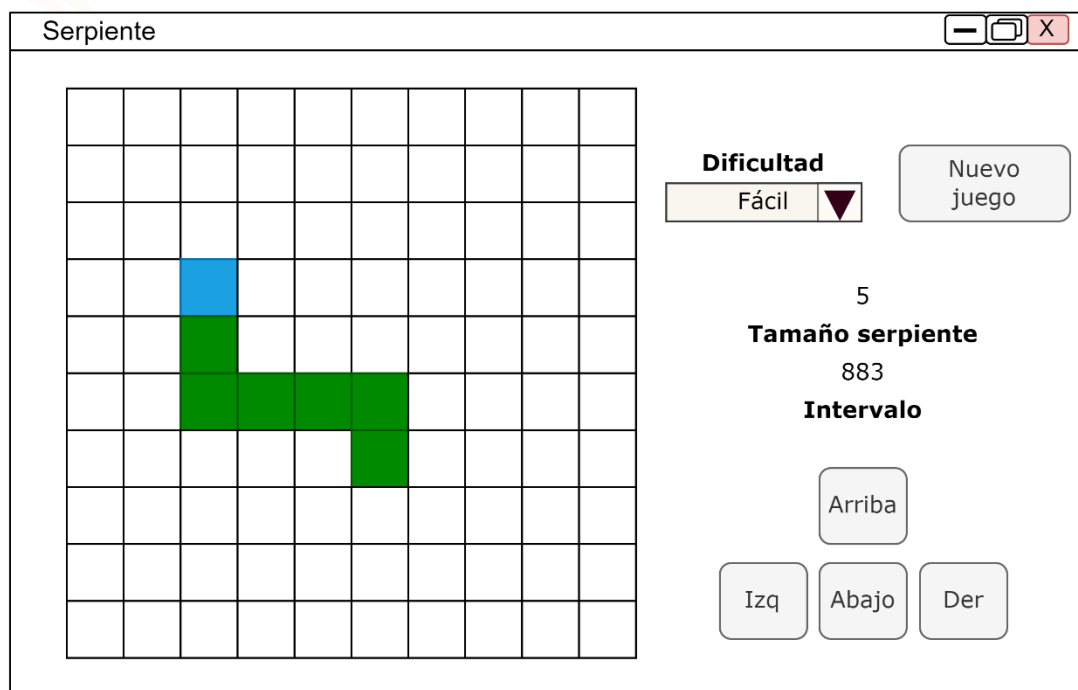
Cada vez que la serpiente se coma una fruta, una nueva deberá aparecer en el tablero.

---

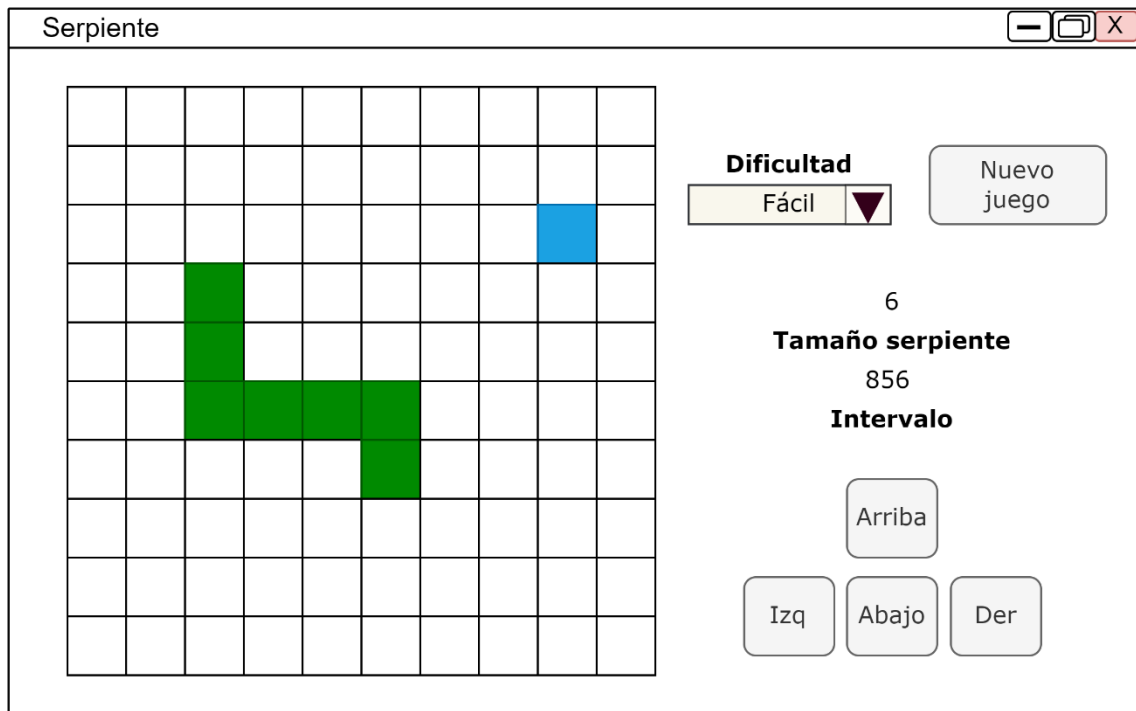
## VELOCIDAD DE LA SERPIENTE

La serpiente iniciará con un intervalo de 1000. Es decir, 1000 milisegundos le tomarán moverse a su siguiente posición.

Cada vez que la serpiente consuma una fruta la velocidad deberá aumentar, es decir, los milisegundos deberán disminuir, haciendo que los movimientos ocurran más a menudo.



En la imagen, la serpiente está a punto de comerse la fruta azul. El tamaño de la serpiente aumentará y el intervalo de movimiento disminuirá.



El intervalo de movimiento cambiará de acuerdo con la dificultad seleccionada. El intervalo disminuirá de la siguiente manera:

- Fácil, -3% de intervalo.
- Medio, -6% de intervalo.
- Difícil, -9% de intervalo.

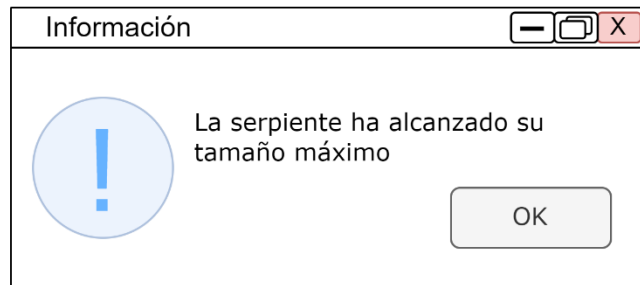
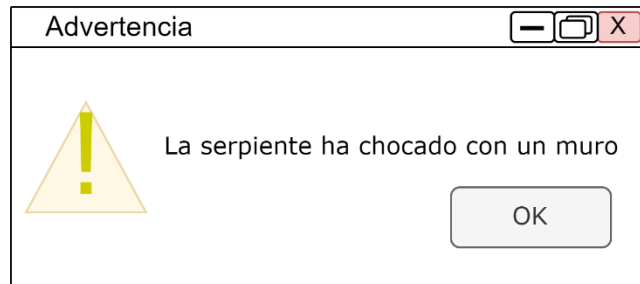
En las dos anteriores imágenes se visualiza el comportamiento del intervalo. En la primera el intervalo es de 883 y en la segunda es 856. Con la dificultad fácil se realiza el siguiente cálculo:  $883 * (1 - 0.03) = 856$ . Para los milisegundos únicamente se manejarán valores enteros.

## FIN DEL JUEGO

El juego terminará y notificará si ocurre cualquiera de los siguientes casos:

- La serpiente se choca con los bordes del tablero.
- La serpiente alcanza su tamaño máximo: 25.





Cuando una partida termine se podrá iniciar una nueva realizando el mismo procedimiento: elegir la dificultad y presionando el botón iniciar, etc.

## CONSIDERACIONES

- Se tomará en cuenta la tolerancia a fallos de la solución. Deberá procurar validar cualquier posible excepción de la solución desarrollada, independientemente si se mencionó en los requerimientos o no.
- Las imágenes que se muestran en este enunciado sirven de ejemplo. No es necesario que la solución generada sea exactamente igual, pero debe asegurarse que se cumpla con los requerimientos mencionados.
- Toda la aplicación deberá ser desarrollada completamente en Java. Se recomienda usar las librerías `java.awt` y `javax.swing` para construir la interfaz gráfica.
- El IDE para usar queda a discreción del estudiante. Se recomienda la herramienta de construcción de interfaz gráfica (drag & drop) de NetBeans.
- Se tendrán en cuenta detalles como:
  - Correcta indentación del código.
  - Uso de comentarios.
  - Declaración e identificación de variables
  - Declaración e identificación de funciones
  - Declaración e identificación de clases.
- No se tomará en cuenta el código como “procedimiento” para ponderar un ítem. Sin embargo, si se revisará para asegurarse que se cumpla con los requerimientos, restricciones y estándares de calidad de código.
- Se deberá usar o implementar al menos un hilo de Java para tener derecho a calificación.

## RESTRICCIONES

- No se podrá usar librerías que implementen la clase `java.util.Collection` o extiendan la clase `java.util.Iterable`. Es decir, no se podrá usar objetos como `LinkedList`, `Set`, `Vector`, `ArrayList`, etc. Únicamente deberá hacer uso de variables, constantes, arreglos, funciones, ciclos (`for`, `while`, `do-while`), estructuras de selección (`if-else`, `switch`), clases generadas por el estudiante,

entre otras. Cualquier otra librería que el estudiante quiera usar se deberá consultarlo primero con el auxiliar.

- Las clases creadas por el estudiante y usadas para modelar objetos deberán cumplir con las propiedades de la programación orientada objetos. Se penalizará el uso inadecuado de las variables y funciones de clase (static) cuando la información deba ser almacenada en variables de instancia o manejada por funciones de instancia, respectivamente.

## ENTREGABLES

- Fecha y hora límite de entrega: **21/09/2022 23:59 hrs.** No se aceptarán entregas luego de esa hora. No se aceptarán entregas por correo.
- Copias detectadas tendrán nota 0 y serán reportadas a las instancias correspondientes.
- Entregar únicamente el enlace de su repositorio de GitHub en UEDI. Se verificará que el repositorio no tenga actividades ni commits luego de la fecha de entrega. Si hay evidencia de modificación de código se realizará un rollback y desplegará la versión anterior más cercana a la fecha y hora de entrega.