

Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ciencias y Sistemas

Introducción a la programación y Computación 1

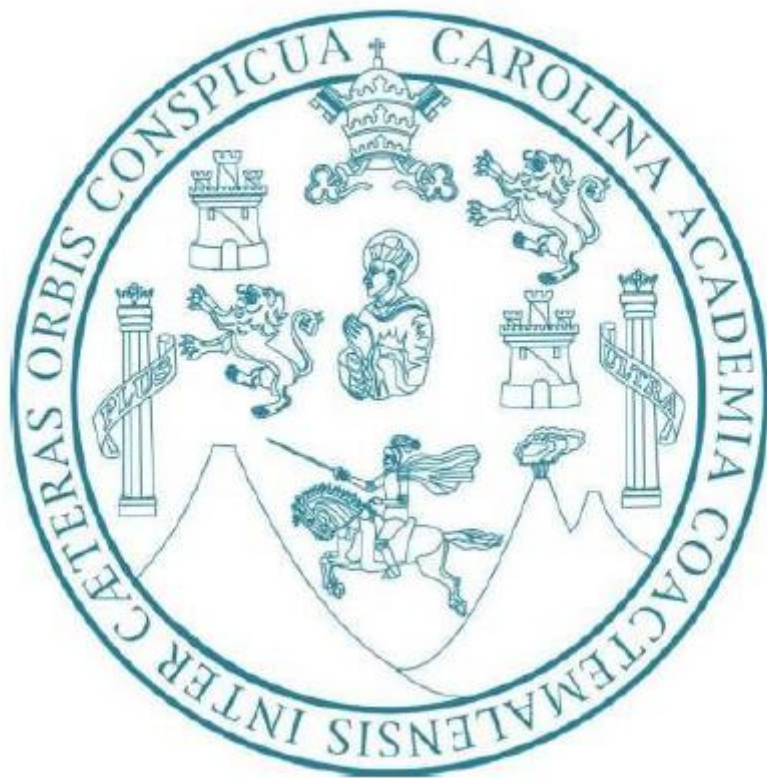
Segundo Semestre 2021

Catedrático: Moisés Eduardo Velásquez Oliva

Tutor académico: William Adolfo Corado Cabrera

Practica 1

Manual técnico



Nombre: Josue Daniel Solis Osorio

Carnet: 2020001574

Fecha: 28/08/2021

Introducción

El presente documento describe los aspectos técnicos informáticos del sistema de información. El documento familiariza las persona que utilicen el programa para su correcta ejecución.

Objetivo de los procedimientos

La elaboración de el juego denominado serpientes y escaleras.

Especificación técnicas

El sistema operativo con el que se trabajo con Windows 10, utilizando el lenguaje de programación denominado java

Lógica del programa

Constructores

- 1) Tablero: Este constructor se encarga de enumerar la parte superior de cada casilla del tablero
- 2) Calculadora: Este constructor se encarga de tener las variables “reporte1” y “reporte2”

Clase: Tablero

- dibujadoTablero: esta función se encarga dibujar el tablero por medio de ciclos for

```
public void dibujadoTablero() {...26 lines }
```

- convertirPosicion: esta función se encarga de definir las posiciones en las que estará el jugador a la hora de recorrer el tablero

```
public int convertirPosicion(int x, int y) {...165 lines }
```

- jugadorYPenalizaciones: Esta función se encarga de darles un valor de 0 a 64 para el jugador y las penalizaciones

```
public void jugadorYPenalizaciones() {...14 lines }
```

- dado: Esta función se encarga de crear un dado a partir de un numero aleatorio

```
public int dado() {...6 lines }
```

- juego: Esta función se encarga de asignar las posiciones “fácil, intermedio y difícil” a las penalizaciones

```
public void juego() {...46 lines }
```

- generarReporte2: Esta función se encarga de generar el reporte 2

```
public void generarReporte2() {...20 lines }
```

- generarReporte1: Esta función se encarga de generar el reporte 1

```
public void generarReporte1() {...21 lines }
```

- operacionesFaci1: esta función se encarga de asignar almohadillas a dificultad fácil

```
public void operacionesFaci1() {...21 lines }
```

- operacionesFaci2: Esta función se encarga de asignar almohadillas a dificultad intermedia

```
public void operacionesFaci2() {...21 lines }
```

- operacionesFaci3: Esta función se encarga de asignar almohadillas a dificultad Difícil

```
public void operacionesFaci3() {...21 lines }
```

- escribirCss: Esta función se encarga de añadir css al html de los reportes

```
public void escribirCss() {...47 lines }
```

Clase: Calculadora

- leyDeCosenos: Esta función se encarga de generar un numero aleatorio para decidir cual de las 3 opciones ejecutar a la hora de caer en una almohadilla para la dificultad facil

```
public double leyDeCosenos() {...25 lines }
```

- SumaDeMatrices: Esta función se encarga de generar un numero aleatorio para decidir cual de las 3 opciones ejecutar a la hora de caer en una almohadilla para la dificultad intermedia

```
public double SumaDeMatrices() {...25 lines }
```

- divisionElegir: Esta función se encarga de generar un numero aleatorio para decidir cual de las 3 opciones ejecutar a la hora de caer en una almohadilla para la dificultad difícil

```
public double divisionElegir() {...25 lines }
```

- LeyDeCosenosopcion1: esta función ejecuta el problema 1 de ley de cosenos

```
public double LeyDeCosenosopcion1() {...111 lines }
```

- LeyDeCosenosopcion2: esta función ejecuta el problema 2 de ley de cosenos

```
public double LeyDeCosenosopcion2()
```

- LeyDeCosenosopcion3: esta función ejecuta el problema 1 de ley de cosenos

```
public double LeyDeCosenosopcion3()
```

- sumaYRestaDeMatricesOpcion1: esta función ejecuta el problema 1 de suma y resta de matrices

```
public void sumaYRestaDeMatricesOpcion1() {...104 lines }
```

- sumaYRestaDeMatricesOpcion2: esta función ejecuta el problema 2 de suma y resta de matrices

```
public void sumaYRestaDeMatricesOpcion2() {...104 lines }
```

- sumaYRestaDeMatricesOpcion3: esta función ejecuta el problema 3 de suma y resta de matrices

```
public void divisionDeMatricesOpcion3() {...78 lines }
```

- divisionDeMatricesOpcion1: esta función ejecuta el problema 1 de división de matrices

```
public void divisionDeMatricesOpcion1() {...80 lines }
```

- divisionDeMatricesOpcion2: esta función ejecuta el problema 2 de división de matrices

```
public void divisionDeMatricesOpcion2() {...77 lines }
```

- divisionDeMatricesOpcion3: esta función ejecuta el problema 3 de división de matrices

```
public void divisionDeMatricesOpcion3() {...78 lines }
```

- Cofactores: Esta función se encarga de calcular los cofactores de una matriz

```
public float[][] cofactores(float matriz[][]) {...38 lines }
```

- Determinante4x4: Esta función se encarga de obtener el determinante 4x4

```
public float determinante4x4(float matriz[][]) {...32 lines }
```

- Traspuesta: Esta función se encarga de transponer una matriz

```
public float[][] traspuesta(float matriz[][]) {...10 lines }
```

- Inversa: Esta función se encarga de dividir la adjunta con el determinante

```
public float[][] inversa(float matriz[][]) {...17 lines }
```

- Adjunta: Esta función se encarga transponer la matriz de los cofactores

```
public float[][] adjunta(float matriz[][]) {...5 lines }
```

- divisionDeMatrices: Esta función se encarga de multiplicar inversa con la matriz A

```
public float[][] divisionMatrices(float[][] matrizA, float matrizB[][]) {...8 lines }
```

- imprimir: Esta función se encarga de imprimir matrices

```
public void impresion(int[][] matrizA) {...16 lines }
```

