

28-8-2021

# Manual Técnico

## Practica 1



Abner Martín Noj Hernández  
201801027

# Índice

Descripción del Proyecto.....	2
Datos Técnicos.....	2
Flujo de trabajo .....	2
Descripción de clases .....	2

## Descripción del Proyecto

Java es un lenguaje de programación, enfocado a la programación orientada a objetos, este tipo de programación permite al programador la reutilización de código. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo, lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra.

Se solicitó que se realizara un programa en consola con el lenguaje de java para poder simular un juego llamado: “Escaleras Matemáticas”.

A continuación se describirá cada una de las funcionalidades del programa.

## Datos Técnicos

Lenguaje utilizado: Java

IDE utilizado: Netbeans 8.2

Sistema Operativo: Windows 8 (64 bits)

## Flujo de trabajo

El flujo se describe de la siguiente manera:

1. Se abre el archivo .jar
2. Se ejecuta el menú.
3. Se ingresan los datos necesarios para el área seleccionada
4. El resultado se muestra en consola
5. Se puede generar reportes.

## Descripción de clases

### **Clase Main:**

En esta clase es la que permite que se ejecuta el programa.

### **Clase Menu:**

En esta clase esta almacenada el flujo del menú principal del juego

### **Clase Juego:**

En esta clase se encuentra le flujo del juego incluyendo las penalizaciones a realizar

#### **inicializar():**

este método crea el tablero, el tablero de posición del jugador, además del tablero de penalizaciones, y mantiene el flujo del juego.

#### **desplazamientoJugador():**

este método realiza el desplamiento del jugador dentro del juego.

**posicionJugadorInicio():**

este método es para ubicar al jugador en la casilla 1 al comienzo de la partida

**llenadoPenalizacion():**

este método es para llenar la matriz de penalizaciones, donde se colocan las posiciones aleatoriamente donde estará la penalización.

**condicionRandom():**

este método es para asignar aleatoriamente números en una matriz

**llenadoTablero():**

este método es para llenar la matriz del 1 al 64 en el orden de arriba para abajo

**horizontal():**

este método es para imprimir una línea horizontal en el tablero

**impresionTablero():**

este método es para imprimir la matriz tal y como se llenó

**impresionfinal():**

este método es para imprimir el tablero como se muestra en pantalla donde el 1 se encuentra en la parte inferior y el 64 en la parte superior.

**buscar():**

función para buscar un valor en una matriz

**penalizacionesFacil():**

método para escoger una penalización de las 3 que son fáciles.

**penalizacionIntermedia():**

método para escoger una penalización de las 3 que son intermedias.

**penalizacionDificil():**

método para escoger una penalización de las 3 que son difíciles..

**Clase Penalizaciones:**

En esta clase se encuentra el flujo para realizar las penalizaciones del juego.

**penalizacionDificil():**

método que realiza el flujo para calcular la división de matrices.

**penalizacionMedio():**

método para realizar el flujo para calcular la suma de matrices.

**penalizacionFacilparaB():**

método que realiza el flujo para calcular 1 de la tres opciones fáciles.

**penalizacionFacilparaA():**

método que realiza el flujo para calcular 1 de la tres opciones fáciles.

**penalizacionFacilparaC():**

método que realiza el flujo para calcular 1 de la tres opciones fáciles.

**Suma():**

función para realizar la suma de matrices, y de devolverá una matriz de tipo int.

**Multiplicacion():**

función para realizar la multiplicación de matrices, y de devolverá una matriz de tipo double.

**definirInversa():**

función para realizar la inversa de matrices, y de devolverá una matriz de tipo double.

**calculoDeterminante():**

función para calcular el determinante de una matriz

**calculoMatrizMenor():**

función para encontrar la matriz una matriz menor de otra matriz.

**matrizAdjunta():**

función para calcular la adjunta de una matriz

**calculoSigno():**

función para calcular el signo correspondiente en una posición para una matriz

**transpuesta():**

función para calcular la transpuesta de una matriz

**inversa():**

función para calcular la inversa de una matriz, asignando cada valor del resultado de la traspuesta sobre el determinante.

**ImpresiónMatriz():**

función para imprimir una matriz

**redondeoMatriz ():**

función para redondear a 3 decimales los elementos de una matriz