

# **MANUAL TECNICO**

## **PRACTICA 1**

Nathan Antonio Valdéz Valdéz

202001569

IPC1 – D

## INTRODUCCION

El presente documento ayuda a cualquier otro programador a entender mejor el algoritmo que se utilizó para poder realizar el juego, facilitará la edición, actualización o si el programador quisiera agregarle más detalles al juego.

# Requerimientos Mínimos

## Windows

- Windows 10 (8u51 y superiores)
- Windows 8.x (escritorio)
- Windows 7 SP1
- Windows Vista SP2
- Windows Server 2008 R2 SP1 (64 bits)
- Windows Server 2012 y 2012 R2 (64 bits)
- RAM: 128 MB
- Espacio en disco: 124 MB para JRE; 2 MB para Java Update
- Procesador: Mínimo Pentium 2 a 266 MHz
- Exploradores: Internet Explorer 9 y superior, Firefox

## Mac OS X

- Mac con Intel que ejecuta Mac OS X 10.8.3+, 10.9+
- Privilegios de administrador para la instalación
- Explorador de 64 bits

Se requiere un explorador de 64 bits (Safari, por ejemplo) para ejecutar Oracle Java en Mac.

## Linux

- Oracle Linux 5.5+<sup>1</sup>
- Oracle Linux 6.x (32 bits), 6.x (64 bits)<sup>2</sup>
- Oracle Linux 7.x (64 bits)<sup>2</sup> (8u20 y superiores)
- Red Hat Enterprise Linux 5.5+<sup>1</sup> 6.x (32 bits), 6.x (64 bits)<sup>2</sup>
- Red Hat Enterprise Linux 7.x (64 bits)<sup>2</sup> (8u20 y superiores)
- Suse Linux Enterprise Server 10 SP2+, 11.x
- Suse Linux Enterprise Server 12.x (64 bits)<sup>2</sup> (8u31 y superiores)
- Ubuntu Linux 12.04 LTS, 13.x
- Ubuntu Linux 14.x (8u25 y superiores)
- Ubuntu Linux 15.04 (8u45 y superiores)
- Ubuntu Linux 15.10 (8u65 y superiores)
- Exploradores: Firefox

## METODO MAIN

En el método main encontramos un menú en donde tenemos un sentencia switch y como primera opción nos indica iniciar el juego, para esto vamos a inicializar la posición del jugador que seria en la casilla numero 1 seguido de generar las penalizaciones que va tener nuestro tablero por medio de un arreglo secundario que en cada casilla se generan números aleatorios desde el 0 al 1 y este mismo nos va indicar que en las casillas que almacena un valor 1 hay penalizaciones, luego se ejecuta el método Juego(); que se explica a continuación.

### Juego();

Al inicio de este método lo primero que hace es verificar si en la casilla 1, que es donde inicia el juego, hay una penalización para poder verificar que se realice correctamente y poder continuar con lo siguiente que seria preguntarle al usuario si quiere continuar con el juego tirando el dado o regresar al menú principal. Si el jugador desea tirar el dado se generará un numero aleatorio desde el 2 hasta el 6 y este se le sumará a la posición del jugador

### MostrarTablero();

Este método a pesar de ser algo simple juega un papel importante en el programa puesto que además de mostrar en consola el tablero también verifica el tablero secundario de penalizaciones y si el jugador cae en una casilla en donde hay una penalización esta devuelve a la variable penitencia un valor true para que al inicio del método Juego(); verifique si hay penalización, luego de que la penalización se haya resuelto correctamente se le devuelve un valor false a la variable penitencia.

### Penitencias();

Para realizar una penitencia este método verifica si el jugador cayó en una casilla menor a la 64, se realizará una penitencia de nivel difícil, si cae en una casilla menor a 41 devuelve una penalización de nivel medio y si cae en una casilla menor a 17 penalización fácil. Luego se verifica si el usuario en ese nivel ya realizó 2 penalizaciones, si es así se exoneran todas las penalizaciones posteriores que toquen en ese nivel, si el usuario no ha realizado 2 operaciones en el nivel este tirará un dado para que el programa escoja aleatoriamente que cual de los 3 casos por nivel se resolverá