

**Desarrollo de videojuego para introducir a la  
resolución de problemas de programación,**

# **P-Learning**

Elihú Salcedo Ruiz



# 1 - Introducción



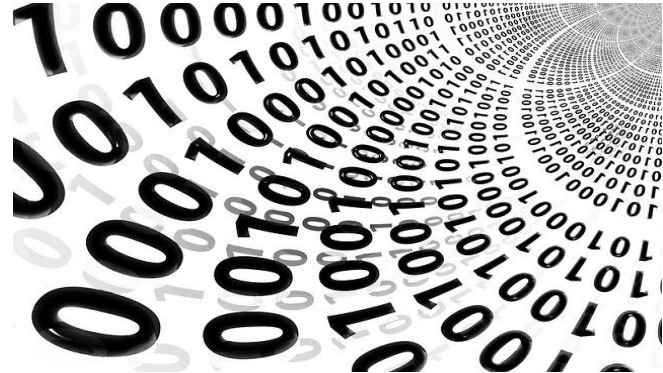
## 1.1 - ¿Programación para niños?

```

17 string input;
18 int iLength, iN;
19 double dblTemp;
20 bool again = true;
21
22 while (again) {
23     iN = -1;
24     again = false;
25     getline(cin, input);
26     system("cls");
27     stringstream(sinput) >> dblTemp;
28     iLength = sinput.length();
29     if (iLength < 4) {
30         again = true;
31         continue;
32     } else if (sinput[iLength - 3] != '.') {
33         again = true;
34         continue;
35     } while (++iN < iLength) {
36         if (isdigit(sinput[iN])) {
37             continue;
38         } else if (iN == (iLength - 3)) {
39             ve:

```

## Código fuente



## Código binario

## Compilación

## 1.2 - El juego como método de aprendizaje

"los juegos constituyen la fuente principal de desarrollo cultural en el niño, y en particular, del desarrollo de la actividad simbólica"



**Lev Vygotsky (1896-1934)**

Teórico de la psicología del desarrollo.

"El juego constituye una de las manifestaciones más importantes del pensamiento infantil ya que, a través de él, el niño desarrolla nuevas estructuras mentales a lo largo de sus distintas etapas evolutivas"



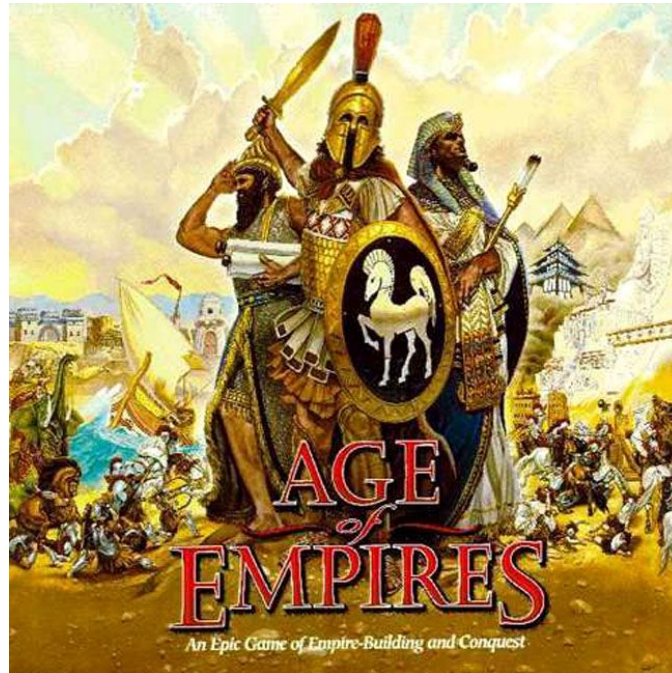
**Jean Piaget (1896-1980)**

Epistemólogo, psicólogo y biólogo.

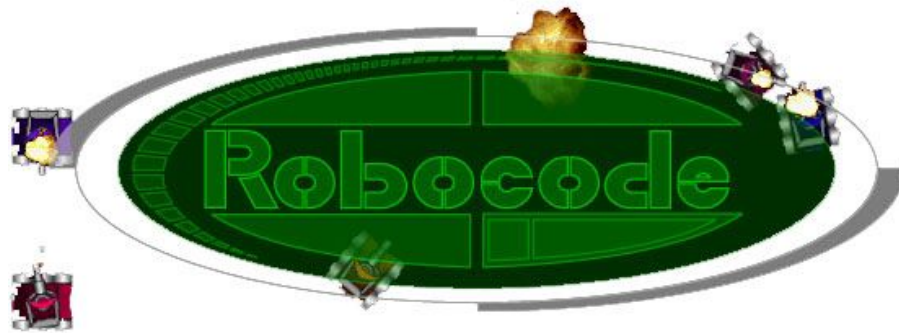
## 1.3 - Videojuegos para aprender



## 1.3 - Videojuegos para aprender



## 1.4 - Videojuegos para aprender a programar



## 2 - La idea P-Learning





## 2.1 - Resumen

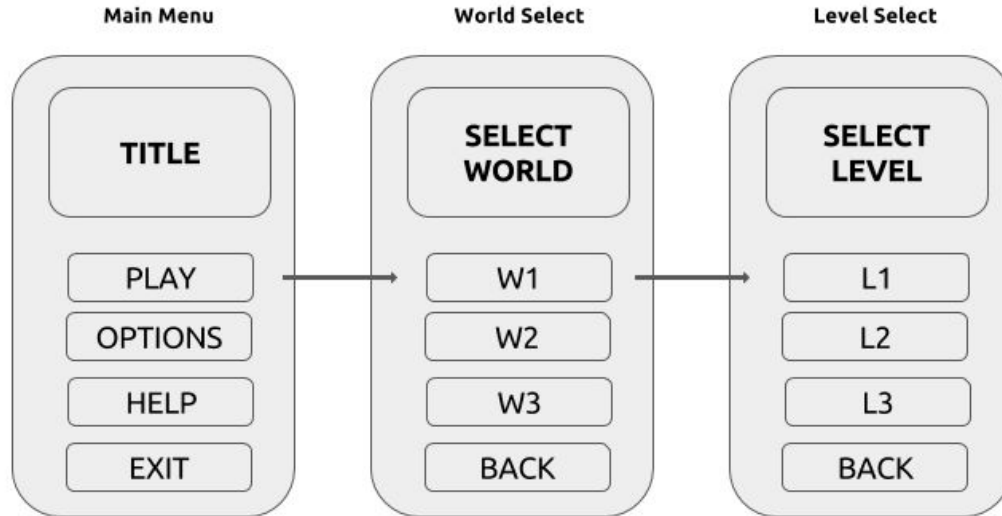
ENTRADA DADA

ESTRUCTURAS DE  
CONTROL

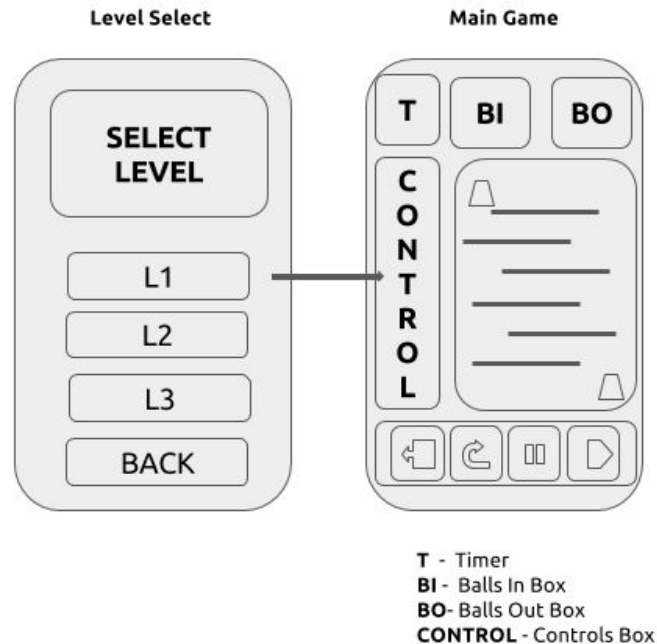
SALIDA DESEADA

SOFTWARE LIBRE

## 2.2 - Diseño conceptual



## 2.3 - Lógica principal de juego



## 3 - Desarrollo



## 3.1 - Tecnología y componentes

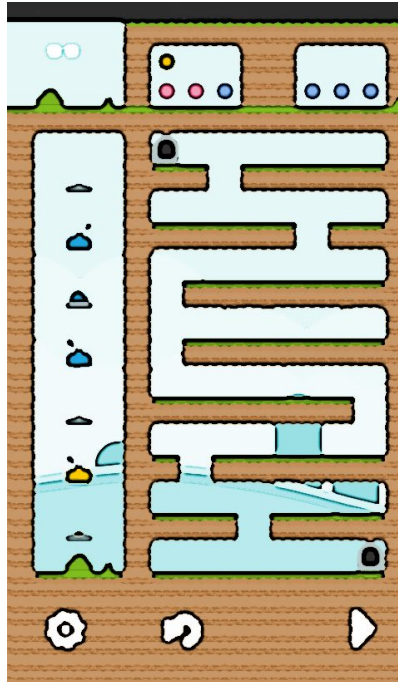
built with

**libGDX**

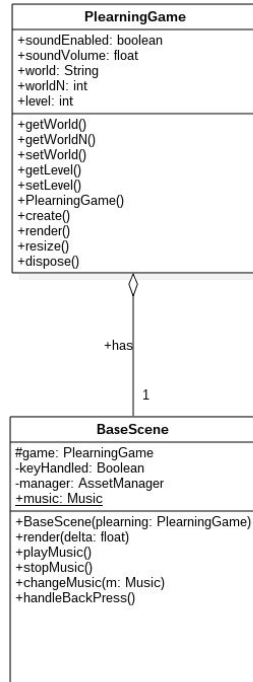
OpenJDK



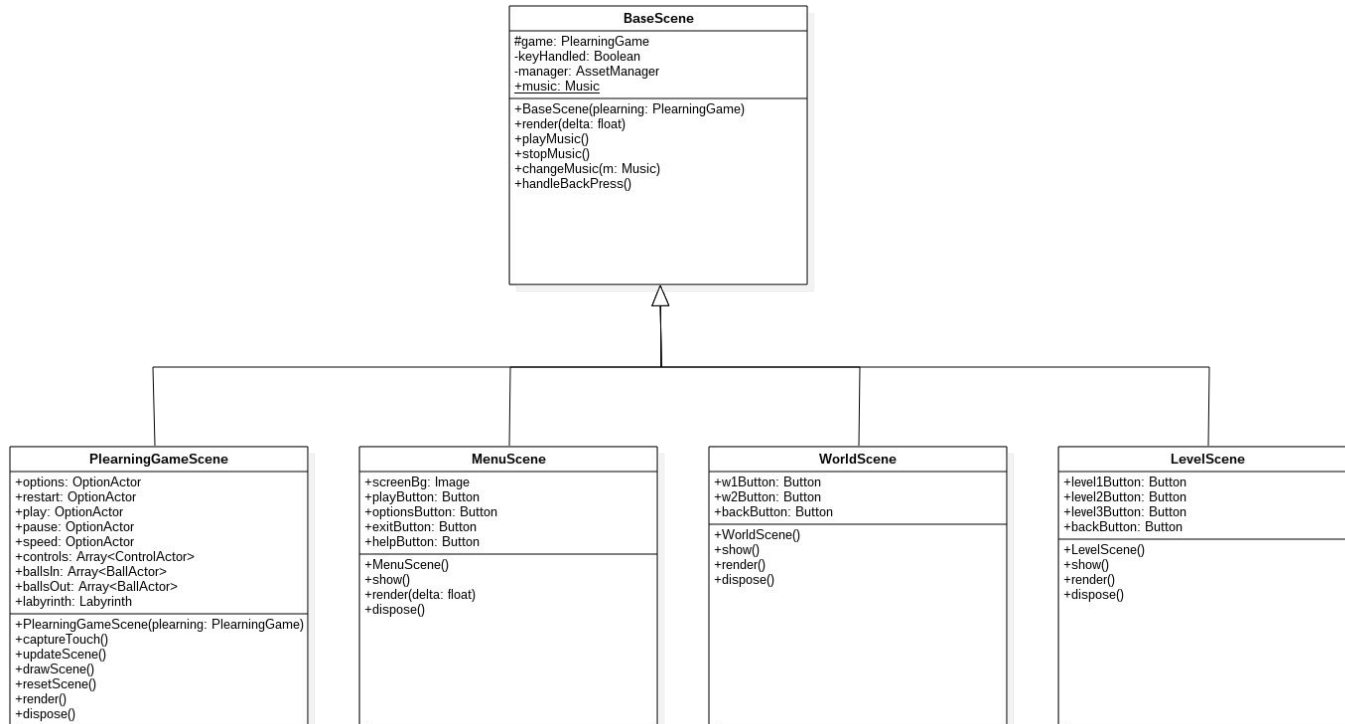
## 3.2 - Diseño detallado



## 3.3 - Classes

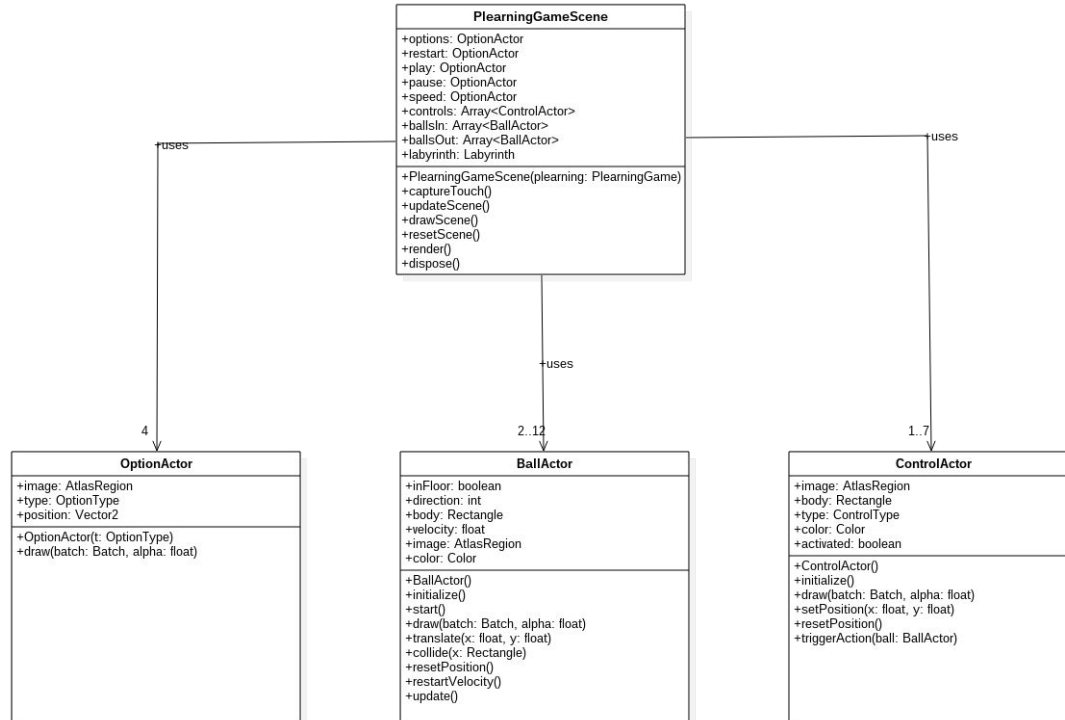


## 3.3 - Classes II





## 3.3 - Clases III



## 3.4 - Código fuente

Datos

Inicialización

Estados de  
juego

```
public Array<ControlActor> getControls(int level){  
    return controls.get(level);  
}  
public Array<BallActor> getBallsIn(int level){  
    return ballsIn.get(level);  
}  
public Array<BallActor> getBallsOut(int level){  
    return ballsOut.get(level);  
}
```

## 4 - Futuro



## 4.1 - Objetivos alcanzados

Lógica principal  
de juego

Generación de  
niveles

Captura de  
objetivos de  
juego

Recorridos en  
laberinto



## 4.2 - Objetivos pendientes



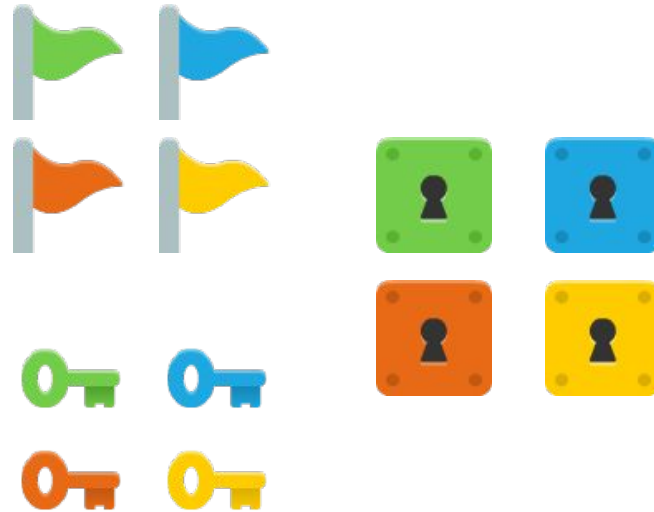
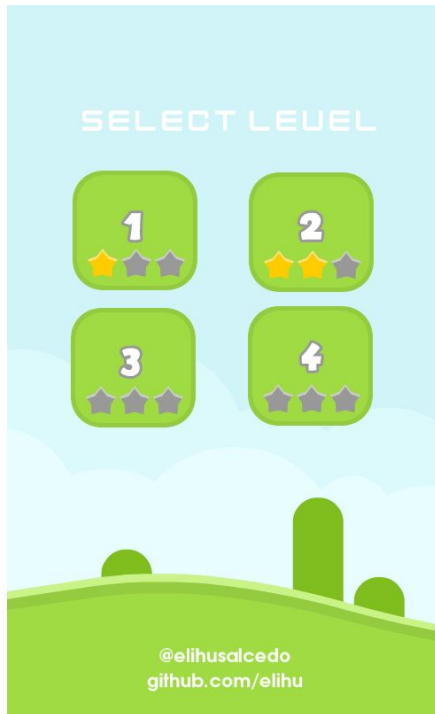
**Puntuaciones y  
recompensas de  
juego**

**Desarrollo de  
nuevos  
controles**

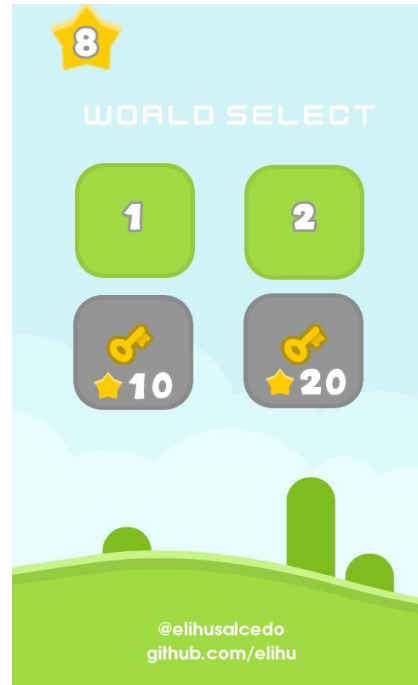
**Publicación en  
Google Play**

**Desarrollo de  
nuevos mundos  
y niveles**

## 4.3 - Siguiente iteración



## 4.4 - Aplicación comercial



## 5 - Demostración





## **5.1 - Demostración en vídeo**

## 5.2 - Demostración en PC

## 5.3 - Demostración en smartphone

¡Muchas gracias!

Elihú Salcedo Ruiz

**Grado en Ingeniería Informática**