

# MANUAL DEL PROGRAMADOR

## 1. INTRODUCCION

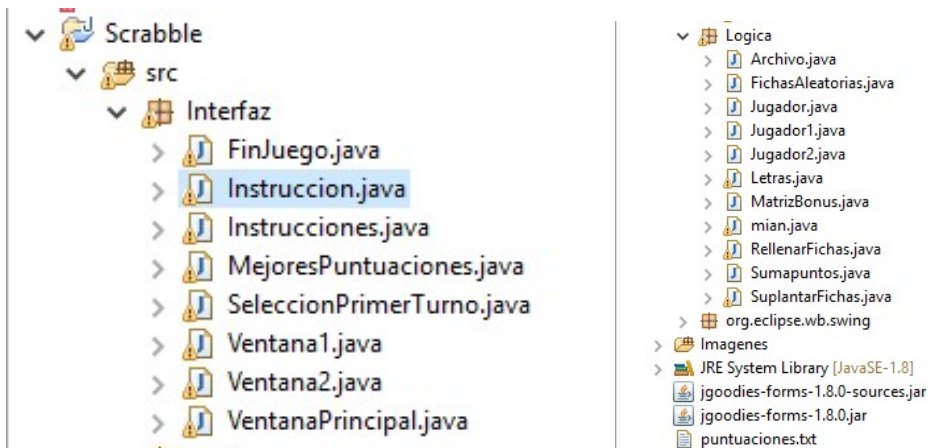
El sistema fue desarrollado bajo JAVA (JDK 1.2), y se usó Swing para la interfaz. Las librerías de Swing vienen integradas a partir del JDK 1.2.

Este proyecto cuenta con 3 partes, las partes de la interfaz, que es lo que el usuario vera, la parte lógica, en donde se utilizan clases para hacer cálculos y la lógica que llevara el programa.

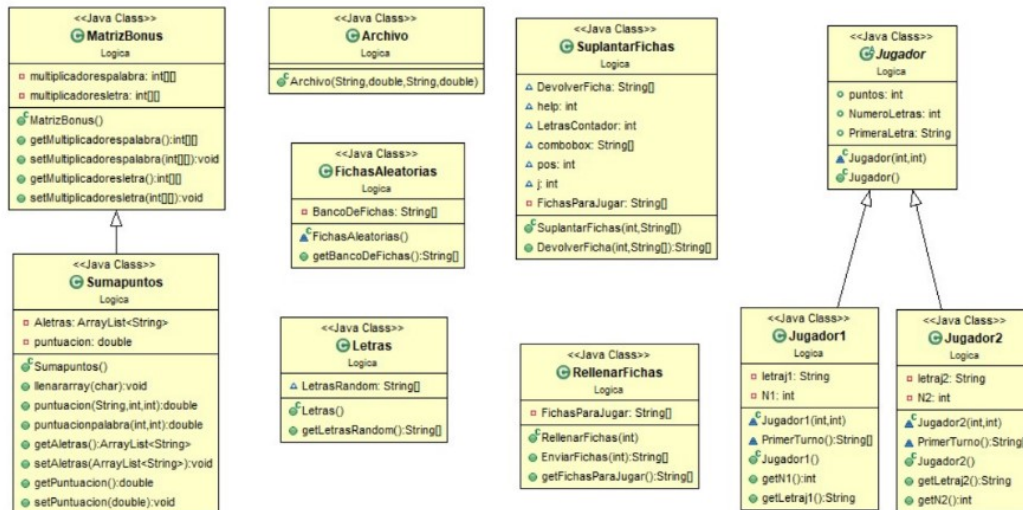
En las siguientes etapas de este manual, se explica el funcionamiento de los procesos críticos, y se enumeran sugerencias para optimizar los sistemas en futuras versiones.

## 2. ESTRUCTURA DEL PROGRAMA

El programa en su parte lógica cuantas con 10 clases y en su parte de la interfaz con 8 clases, cada una siendo una ventana diferente, utilizada la herramienta Window Builder por medio de Jframes.



### 3. ARQUITECTURA



#### 3.1 LA INTERFAZ:

En este espacio se enunciarán todas las clases y un pequeño resumen de lo que hace cada una y así llegar a tener una idea de cómo se estructura todo el código, en esta parte solamente la interfaz.

El listado completo de las mismas es el siguiente:

- **Ventana1:** Bienvenida al juego.
- **Ventana2:** Ingresar nombre de los usuarios.
- **Instrucciones:** Reglas del juego con un esquema del tablero.
- **VentanaPrincipal:** Tablero del juego.
- **SeleccionPrimerTurno:** Selección de que jugador inicia primero.
- **MejoresPuntuaciones:** Tabla con jugadores con mas puntos .
- **Fin del juego:** Muestra el ganador y los puntos de los jugadores.
- **Instruccion:** Instrucción muy breve y poco compleja.

A continuación se habla mas detenidamente acerca de algunos de los aspectos más relevantes de algunas de las clases. Otras fueron omitidas dado que poco se puede aportar a lo ya escrito en su documentación respectiva.

## 1. SeleccionPrimerTurno

Esta clase se usa como un dado para saber quién inicia el juego, se tiene un botón el cual se pulsa para que salgan 2 fichas del juego totalmente aleatorias, y luego se evalúa su código ascii con respecto a la letra A, y la más cercana a esta ganara.

Para esta clase se crean 2 objetos en la lógica, uno llamado llamando a la clase Jugador1 y otro a Jugador2, que tienen los datos iniciales para poder iniciar el juego, y para empezar las clases Jugador1 y Jugador2 con clases que crean objetos de la clase Letras, en donde se tiene un vector con todas las fichas que se utilizaran en el juego, para sacar una al azar.

## 2. VentanaPrincipal

Esta clase implementa una ventana que muestra una tabla de botones de 15x15, en total son 225 botones que tienen su debido código, en el cual almacena el puntaje de cada botón, y tiene la información para poder modificar el banco de fichas de cada jugador, ya que cuando una letra se coloca en un botón esta automáticamente debe salir del comboBox en la cual esta almacenada ya que esta ya se uso.

En esta clase también se usa la clase RellenarFichas usada para poder simular un banco de fichas en donde cada jugar va robando cada vez que no tiene sus fichas completas, este se usa como un array en donde la posición que se quiere la nueva ficha la guardara, y se reemplazara con una nueva.

### 2.1 Arrays Trabajados en esta clase

Para llevar el control de los datos, se hace uso de arreglos, y solo se enunciaran los de importancia muy grande para el buen funcionamiento del programa:

- **comboBox:** Lleva 7 posiciones para almacenar las 7 fichas del combo Box dependiendo del turno
- **activarBotones:** de tamaño 15x15 es una matriz que funciona para activar o desactivar cada casilla dependiendo de donde se coloque una ficha.

### **3 MejoresPuntuaciones**

Esta funciona como una tabla que muestra el puntaje final de cada jugador con el puesto que ocupo, y muestra los mejores jugadores, estos resultados también se imprimen en un bloc de notas que va a guardar estos mismos datos.

Esta clase utiliza 2 arrayList para el manejo de datos más óptimo en donde se guarda el nombre del jugador y su puntuación final

#### **3.2 Clases Asociadas**

Otras clases involucradas en el control de puntos son Sumapuntos y MatrizBonus y estas clases son involucradas en el proceso de contar los puntos, MatrizBonus guarda las posiciones de los botones donde cada letra puede valer más de lo normal, y la clase Sumapuntos hereda de MatrizBonus ya que la clase Sumapuntos lo que hace es sumar los puntos de cada letra que coloque un jugador en cada turno, esto teniendo en cuenta que según la posición puede valer más cada letra.

### **4. LA APLICACION:**

Esta aplicación esta un formato .jar que puede ser leído por un sistema operativo como Linux, Windows o Mac.

### **5. LOS ARCHIVOS**

Los archivos manejados por este programa son las puntuaciones de cada jugador con su respectivo nombre, esto es implementado y guardado en eun bloc de notas.

### **6. POSIBLES EXTENSIONES O MEJORAS**

1.Cada botón en la matriz funciona totalmente independiente de otro, sin embargo muchos cumplen la misma función, se podría reutilizar código de alguna manera para hacerlo mucho mas optimo.