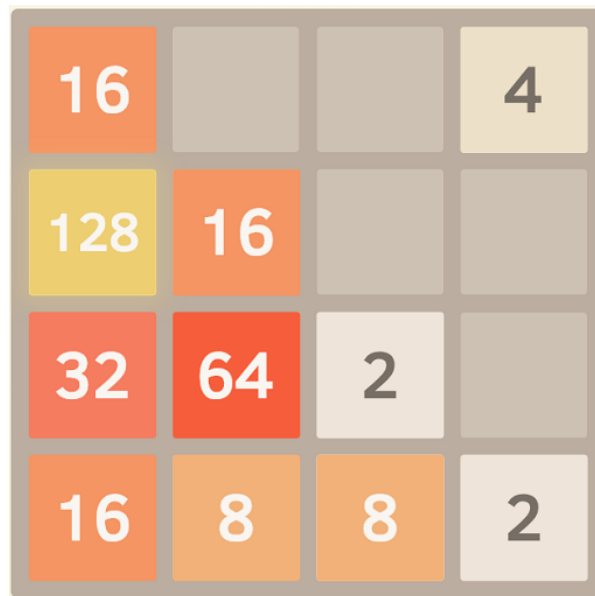




## Programación III – Comisión 1

### Trabajo Práctico N°1: “2048”



#### **Integrantes:**

Gamboggi, Martín

DNI: 32765431

Godoy, Ignacio Gabriel

DNI: 44097529

Gutiérrez, Diego

DNI: 38152049

Rivas Gutiérrez, Nicolás

DNI: 43570562

# Trabajo Práctico “2048”

## Introducción

El juego 2048 es un rompecabezas numérico que se juega en un tablero de 4x4 en el que el jugador combina baldosas con números múltiplos de 2 para finalmente obtener una baldosa con el número 2048. La interfaz proporciona una experiencia interactiva para el usuario al mostrar el tablero de juego, lo que le permite realizar movimientos mediante teclas.

## Descripción General

La implementación que utilizamos se basa en la programación orientada a objetos en Java. La interfaz hace uso de componentes de Swing para la construcción de la ventana principal y elementos interactivos como botones y cuadros de texto.

## Clase Interfaz

La clase **Interfaz** es la clase principal que contiene la lógica para construir y manejar la interfaz gráfica del juego. A continuación, se describen algunas de las principales funcionalidades implementadas en esta clase:

- **Inicialización de la Ventana Principal:** Se establece el aspecto y comportamiento de la ventana principal del juego, incluyendo el título, tamaño, y elementos del menú.
- **Gestión del Juego:** Se implementa la lógica para iniciar un nuevo juego, reiniciar el juego actual y gestionar la salida del juego.
- **Tablero de Juego:** Se crea y se muestra el tablero de juego, que es representado visualmente mediante baldosas con números.
- **Actualización de la Interfaz:** Se actualiza la interfaz gráfica para reflejar los cambios en el estado del juego, como movimientos del jugador y actualización del puntaje.
- **Manejo de Eventos de Teclado:** Se capturan y procesan las pulsaciones de teclas del usuario para permitir movimientos en el juego.
- **Mensajes al Usuario:** Se proporcionan mensajes informativos al usuario, como mensajes de inicio, mensajes de juego nuevo y mensajes de finalización del juego.

## Clase Tablero

### 1. Atributos:

- **tablero:** Una matriz que representa el estado actual del tablero del juego.
- **puntaje:** Almacena el puntaje actual del jugador.
- **records:** Una matriz que almacena los registros de los puntajes más altos y los nombres de usuario asociados.
- **usuario:** Almacena el nombre de usuario actual.

## 2. Métodos de Inicialización:

- **Tablero ()**: El constructor inicializa la matriz del tablero con ceros, coloca dos números aleatorios (2 o 4) en posiciones aleatorias del tablero y establece la matriz de registros con valores predeterminados.

## 3. Métodos de Control de Juego:

- **finPartida ()**: Verifica si la partida ha terminado, ya sea porque el jugador ha alcanzado 2048 o porque el tablero está lleno y no hay más movimientos válidos.
- **Ganador ()**: Comprueba si el jugador ha alcanzado el valor de 2048 en alguna posición del tablero.
- **jugadorPerdio ()**: Indica si el jugador ha perdido la partida, lo cual ocurre cuando no hay más movimientos válidos y el tablero está lleno.
- **movimientoValido ()**: Verifica si hay movimientos válidos disponibles en el tablero.
- **haySumaPosible ()**: Comprueba si es posible realizar alguna combinación en el tablero al mover las casillas.

## 4. Métodos de Movimiento:

- Métodos **moverArriba ()**, **moverAbajo ()**, **moverDerecha ()** y **moverIzquierda()**: Estos métodos implementan la lógica para deslizar y combinar números en el tablero en las cuatro direcciones posibles (arriba, abajo, derecha e izquierda).

## 5. Métodos Auxiliares:

- **vacias ()**: Cuenta la cantidad de ceros en el tablero.
- **ponerDosOCuatro ()**: Agrega un número aleatorio (2 o 4) en una posición aleatoria del tablero.
- **elegirNumero ()**: Elige aleatoriamente entre 2 y 4 con una probabilidad del 80% y 20%, respectivamente.
- **vaciasEnFila (int fila)**: Cuenta la cantidad de ceros en una fila específica del tablero.

## 6. Métodos de Gestión de Usuarios y Registros:

- Métodos **getUsuario ()**, **setUsuario ()**, **getRecord ()**, **setRecord ()**, **getUsuarioConRecord ()**, **setUsuarioConRecord ()**, **getRecordRealizado ()** y **setRecordRealizado ()**: Estos métodos permiten obtener y establecer el nombre de usuario y los registros de puntajes más altos.

## 7. Método de Control de Puntajes:

- **controlDePuntajes ()**: Gestiona la actualización de los puntajes más altos y los nombres de usuario asociados después de que termina una partida.

Esta clase proporciona todas las funcionalidades necesarias para controlar el estado del juego, gestionar los movimientos del jugador, mantener registros de puntajes y determinar cuándo ha terminado la partida

Por otro lado, las clases **Baldosa** y **ColorBaldosas** forman parte de la interfaz gráfica del juego 2048 en Java.

### Clase Baldosa:

Esta clase modela una baldosa individual en el tablero del juego.

#### 1. Atributos:

- **valor**: Representa el valor numérico de la baldosa.

#### 2. Constructores:

- **Baldosa ()**: Constructor que inicializa una baldosa vacía con valor cero.
- **Baldosa (int valor)**: Constructor que permite especificar el valor inicial de la baldosa.

#### 3. Métodos:

- **estaVacía ()**: Verifica si la baldosa está vacía (con valor cero).
- **obtenerColorBaldosa ()**: Devuelve el color asociado al valor de la baldosa mediante el uso de la clase **ColorBaldosas**.
- **getValor ()**: Obtiene el valor actual de la baldosa.
- **setValor (int valor)**: Establece el valor de la baldosa.

### Clase Enum ColorBaldosas:

Esta clase enumera los colores asociados a los diferentes valores de las baldosas.

#### 1. Atributos:

- **colorHex**: Representa el valor hexadecimal del color asociado.

#### 2. Constructores:

- **ColorBaldosas (int colorHex)**: Constructor que asigna un valor hexadecimal a cada color.

#### 3. Métodos:

- **obtenerColorHex ()**: Devuelve el valor hexadecimal del color asociado.

Estas clases se utilizan en la interfaz gráfica para representar visualmente las baldosas del juego 2048. La clase **Baldosa** se encarga de gestionar el valor y el color de cada baldosa individual, mientras que la clase **ColorBaldosas** enumera los colores específicos asociados a cada valor posible de las baldosas.

## **Detalles de Implementación**

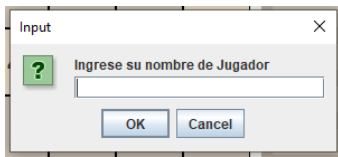
### **Componentes de Swing**

La interfaz hace uso extensivo de componentes de Swing, incluyendo **JFrame**, **JPanel**, **JBUTTON**, **TextField**, **JLabel**, y **JOptionPane**, para construir una interfaz gráfica interactiva y atractiva para el usuario.

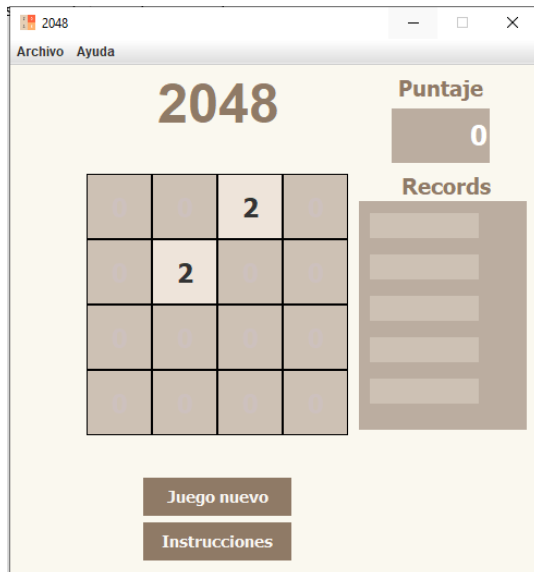
### **Interacción con el Usuario**

La interfaz proporciona una interacción fluida con el usuario, permitiéndole iniciar nuevos juegos, reiniciar el juego actual, salir del juego y mostrar instrucciones. Además, se muestran mensajes informativos y de estado durante el juego para guiar al usuario.

## Funcionalidad

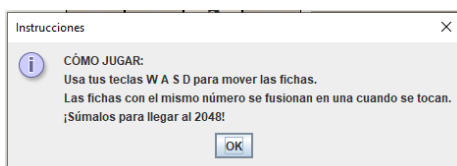


Cuando se inicia el juego se le pide al jugador que ingrese su nombre, que será guardado en caso de que supere algún puntaje record.



Luego de ingresar el nombre se dará inicio al juego, aparecerán aleatoriamente 2 números que pueden ser "2" o "4".

Además, se cuenta con las opciones "Juego nuevo" que comienza un nuevo juego e "Instrucciones" que nos da un breve tutorial.

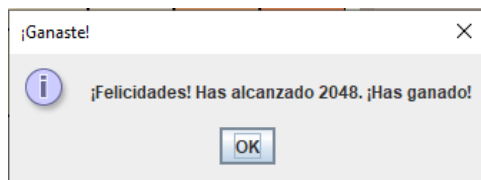


También se tiene en la parte superior otras opciones, incluyendo la opción de salir del juego, la cual nos pedirá una confirmación.

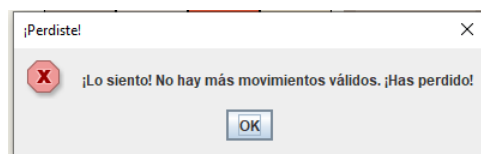


A medida que vamos jugando se puede ver como cada baldosa de distinto número tiene distinto color para diferenciarlos más rápidamente.

A su vez, hay un panel que nos muestra el puntaje que vamos acumulando.



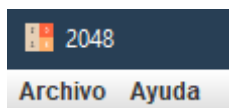
Finalmente, hay un cartel de victoria cuando se llega a 2048 en alguna baldosa.



O, por otro lado, salta un cartel de derrota en caso de que no se haya llegado a 2048 y nos quedamos sin movimientos para hacer.



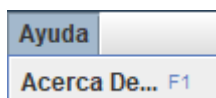
Cuando un jugador alcanza un nuevo record se almacena en la pestaña de Records históricos y a la hora de jugar nuevamente se puede visualizar los records obtenidos de otros jugadores en partidas anteriores, Además el record actual de nuestra partida en curso se muestra al lado del puntaje obtenido en tiempo real para poder verificar nuestra mejor puntuación.



En la esquina superior izquierda se encuentran 2 pestañas desplegables bajo el nombre “Archivo” y “Ayuda”, en estas podemos realizar las siguientes acciones:



La pestaña Archivo nos muestra un menú que consta de 3 opciones, “Juego Nuevo” para iniciar una partida desde 0, “Reiniciar” para volver a intentar una partida sin perder las puntuaciones y records y “Salir” para cerrar el juego. Estas opciones también poseen atajos del teclado de manera que podemos acceder a ellas sin abrir el menú “Archivo”.



La pestaña Ayuda nos muestra un menú que consta de 1 opción, “Acerca De...” nos permite acceder al informe de este trabajo realizado en grupo.