



**UNIVERSIDAD
MAYOR DE SAN SIMÓN**
Ciencia y Conocimiento desde 1832

UNIVERSIDAD MAYOR DE SAN SIMÓN
FACULTAD DE CIENCIAS Y TECNOLOGÍA
CARRERA INGENIERÍA DE SISTEMAS



Documentación del proyecto

Docente: Valentin Laime Zapata

Materia: Métodos, técnicas y taller
de programación

Estudiante: Cesia Alondra Triveño Moreno

Fecha: 15/10/24

Descripción General: Este proyecto consiste en un juego de Piedra, Papel o Tijeras diseñado. El juego permite a los usuarios participar en dos modos: jugar contra la computadora y jugar con otro usuario (modo Torneo).

Índice

1. Clase Crear_nuevo_usuario	página 2
2. Clase Historial	página 3
3. Clase Iniciar_Sesion	página 4
4. Clase Juego_PC	página 5
5. Clase Juego_Torneo	página 7
6. Clase Logro	página 9
7. Clase Logros	página 13
8. Clase Menu	página 14
9. Clase Partida	página 15
10.Clase Resultado_PC	página 17
11.Clase Resultado_Torneo	página 19
12.Clase Usuario	página 21

Clase Crear_nuevo_usuario

Paquete: `java.lang.Object`

Extiende: `javax.swing.JFrame`

Descripción

La clase `Crear_nuevo_usuario` representa una ventana que permite a los usuarios crear una nueva cuenta en el juego.

Constructores

`Crear_nuevo_usuario()`

Constructor de la clase que inicializa la ventana.

Métodos

`public static void main(String[] args)`

Método principal que inicia la aplicación.

`public void setModo(String modo)`

Establece el modo de juego.

- **Parámetros:**
 - `modo`: El modo de juego ("PC" o "Torneo").

`public void setUsuario1(String nombre)`

Establece el nombre del primer usuario en el modo Torneo.

- **Parámetros:**
 - `nombre`: El nombre del primer usuario.

Clase Historial

Paquete: `java.lang.Object`

Extiende: `javax.swing.JFrame`

Descripción

La clase `Historial` es una ventana que permite a los usuarios ver su historial de partidas.

Constructor

`Historial()`

Inicializa la ventana.

Métodos

`void cargar_historial()`

Carga el historial desde el archivo y lo muestra en la interfaz.

`void setDireccion(String direc)`

Establece la dirección del archivo de historial.

Parámetros:

- `direc`: La dirección del archivo de historial.

`void setUsuario(String usuario)`

Establece el nombre del usuario.

Parámetros:

- `usuario`: El nombre del usuario.

`void transparencia_BT_menu()`

Hace invisible el botón de menú.

`void transparencia_BT_salir()`

Hace invisible el botón de salir.

```
static void main(String[] args)
```

Método principal que inicia la aplicación.

Parámetros:

- **args**: Los argumentos de línea de comandos.

Clase Iniciar_Sesion

Paquete: `java.lang.Object`

Extiende: `javax.swing.JFrame`

Descripción

La clase `Iniciar_Sesion` es una ventana de interfaz gráfica que permite a los usuarios iniciar sesión en la aplicación.

Constructor

```
Iniciar_Sesion()
```

Inicializa la ventana.

Métodos

```
void setUsuario1(String usuario)
```

Establece el nombre del primer usuario en el modo torneo.

Parámetros:

- **usuario**: El nombre del usuario.

```
void setModo(String modo)
```

Establece el modo de juego.

Parámetros:

- `modo`: El modo de juego.

`static void main(String[] args)`

Método principal que inicia la aplicación.

Parámetros:

- `args`: Los argumentos de línea de comandos.

Clase Juego_PC

Paquete: `java.lang.Object`

Extiende: `javax.swing.JFrame`

Descripción

La clase `Juego_PC` representa una ventana de juego de "Piedra, Papel o Tijeras". Permite a los usuarios jugar contra la computadora.

Constructor

`Juego_PC()`

Inicializa la ventana del juego.

Campos

- `String direc_historial`: Dirección del historial de partidas.
- `String direc_logros`: Dirección de los logros del usuario.
- `int num_partida`: Número de la partida actual.
- `int racha`: Racha de victorias.
- `String usuario`: Nombre del usuario.

Métodos

`void setNum_partida(int num_partida)`

Establece el número de partida.

Parámetros:

- `num_partida`: El número de la partida.

`void setDirec_Historial(String direc_historial)`

Establece la dirección del historial de partidas.

Parámetros:

- `direc_historial`: La dirección del historial.

`void setDirec_Logros(String direc_logros)`

Establece la dirección de los logros del usuario.

Parámetros:

- `direc_logros`: La dirección de los logros.

`void setUsuario(String us)`

Establece el nombre del usuario y lo muestra en la etiqueta correspondiente.

Parámetros:

- `us`: El nombre del usuario.

`void setRacha(int racha)`

Establece la racha de victorias.

Parámetros:

- `racha`: La racha de victorias.

`int obtener_num_partida(String direc_historial)`

Obtiene el número de la partida a partir del historial.

Parámetros:

- `direc_historial`: La dirección del archivo de historial.

Returns:

- El número de la última partida.

`static void main(String[] args)`

Método principal que inicia la aplicación.

Parámetros:

- `args`: Los argumentos de línea de comandos.

Clase Juego_Torneo

Paquete: `java.lang.Object`

Extiende: `javax.swing.JFrame`

Descripción

La clase `Juego_Torneo` representa la ventana del juego de "Piedra, Papel o Tijeras" para dos jugadores.

Constructor

`Juego_Torneo()`

Inicializa la ventana del juego.

Métodos

`void cargar_datos()`

Carga los datos en la interfaz.

`String ganador(int usuario1, int usuario2)`

Determina el ganador basado en las jugadas de los dos jugadores.

Parámetros:

- `usuario1`: Jugada del jugador 1.
- `usuario2`: Jugada del jugador 2.

Returns:

- Resultado de la partida: "Ganaste", "Perdiste", o "Empate".

`int obtener_num_partida(String direc_historial)`

Obtiene el número de la partida a partir del historial.

Parámetros:

- `direc_historial`: La dirección del archivo de historial.

Returns:

- El número de la última partida.

`void setDirec_Historial(String direc_historial)`

Establece la dirección del historial del juego.

Parámetros:

- `direc_historial`: La dirección del historial.

`void setNum_partida(int num_partida)`

Establece el número de la partida.

Parámetros:

- `num_partida`: El número de la partida a establecer.

`void setRacha(int racha)`

Establece la racha de victorias del jugador.

Parámetros:

- `racha`: La racha a establecer.

`void setUsuario1(String usuario)`

Establece el nombre del jugador 1.

Parámetros:

- **usuario**: El nombre del jugador 1.

void setUsuario2(String usuario2)

Establece el nombre del jugador 2.

Parámetros:

- **usuario2**: El nombre del jugador 2.

static void main(String[] args)

Método principal que inicia la aplicación.

Parámetros:

- **args**: Los argumentos de línea de comandos.

Clase Logro

Paquete: `java.lang.Object`

Extiende: `Object`

Descripción

La clase **Logro** representa un objetivo que los jugadores pueden alcanzar al jugar contra la computadora.

Constructores

Logro()

Constructor por defecto para la clase **Logro**.

Logro(String nombre, String descripcion, int criterio, String tipo)

Constructor que inicializa un nuevo logro con un nombre, descripción, criterio y tipo.

Parámetros:

- **nombre**: El nombre del logro.

- **descripción**: La descripción del logro.
- **criterio**: El número requerido para alcanzar el logro.
- **tipo**: El tipo de logro.

Métodos

void cambiar_estado(boolean estado)

Cambia el estado del logro.

Parámetros:

- **estado**: El nuevo estado del logro.

Logro cargar_estado(String contenido, Logro act)

Separa el contenido de la cadena **contenido** para devolver un logro.

Parámetros:

- **contenido**: El contenido del logro en formato de texto.
- **act**: El logro que se está cargando.

Returns:

- El logro que contenía la cadena.

ArrayList<Logro> cargar_Logros(String direc_logros)

Carga los logros contenidos en el archivo de la dirección especificada.

Parámetros:

- **direc_logros**: La dirección del archivo que contiene los logros.

Returns:

- Una lista de logros.

String getDescripcion()

Obtiene la descripción del logro.

Returns:

- La descripción del logro.

`boolean getEstado()`

Devuelve el estado del logro, indicando si ha sido completado o no.

Returns:

- `true` si el logro ha sido alcanzado; `false` en caso contrario.

`String getNombre()`

Obtiene el nombre del logro.

Returns:

- El nombre del logro.

`int getRequerimiento()`

Obtiene el requerimiento necesario para alcanzar el logro.

Returns:

- El criterio del logro.

`String getTipo()`

Obtiene el tipo de logro.

Returns:

- El tipo de logro.

`void guardarLogros(ArrayList<Logro> logros, String direc_logros)`

Guarda la lista de logros en un archivo.

Parámetros:

- `logros`: La lista de logros que se guardarán.
- `direc_logros`: La dirección del archivo donde se guardarán los logros.

`ArrayList<Logro> lista_logros()`

Crea la lista de logros predeterminados.

Returns:

- Una lista de logros.

`void setDescripcion(String descripcion)`

Establece la descripción del logro.

Parámetros:

- `descripcion`: La nueva descripción del logro.

`void setEstado(int racha, int partida)`

Establece el estado del logro basándose en la racha o número de partidas.

Parámetros:

- `racha`: La racha actual del jugador.
- `partida`: El número de partidas jugadas por el jugador.

`void setNombre(String nombre)`

Establece el nombre del logro.

Parámetros:

- `nombre`: El nuevo nombre del logro.

`void setRequerimiento(int criterio)`

Establece el requerimiento necesario para alcanzar el logro.

Parámetros:

- `criterio`: El nuevo criterio del logro.

`void setTipo(String tipo)`

Establece el tipo de logro.

Parámetros:

- `tipo`: El nuevo tipo del logro.

Clase Logros

Paquete: `javafx.swing.JFrame`

Extiende: `JFrame`

Descripción

La clase `Logros` representa la ventana donde se muestran los logros alcanzados y aquellos que aún no han sido conseguidos.

Constructores

`Logros()`

Constructor que inicializa la ventana de logros.

Métodos

`void setDireccion(String direc)`

Establece la dirección del archivo de logros.

Parámetros:

- `direc`: La dirección a establecer.

`void setUsuario(String usuario)`

Establece el nombre del usuario.

Parámetros:

- `usuario`: El nombre del usuario a establecer.

`static void main(String[] args)`

Método principal que inicia la aplicación y muestra la ventana de logros.

Parámetros:

- `args`: Argumentos de línea de comandos.

```
void mostrar_logros(ArrayList<Logro> lista)
```

Muestra los logros del usuario en la interfaz.

Parámetros:

- `lista`: Lista de logros a mostrar.

Clase Menu

Paquete: `javax.swing.JFrame`

Extiende: `JFrame`

Descripción

La clase `Menu` representa la ventana principal del juego. Permite al usuario acceder a diferentes modos de juego, historial, logros y salir del juego.

Constructores

`Menu()`

Constructor que inicializa la ventana del menú.

Métodos

```
void setDirec_Historial(String direc_Historial)
```

Establece la dirección del archivo de historial.

Parámetros:

- `direc_Historial`: La dirección a establecer.

```
void setDirec_Logros(String direc_Logros)
```

Establece la dirección del archivo de logros.

Parámetros:

- `direc_Logros`: La dirección a establecer.

`void setUsuario(String usuario)`

Establece el nombre del usuario.

Parámetros:

- `usuario`: El nombre del usuario a establecer.

`static void main(String[] args)`

Método principal que inicia la aplicación y muestra la ventana del menú.

Parámetros:

- `args`: Argumentos de línea de comandos.

Clase Partida

Paquete: `java.lang.Object`

Extiende: `Object`

Descripción

La clase `Partida` representa una partida del juego de piedra, papel o tijeras.

Constructores

`Partida()`

Constructor por defecto para la clase `Partida`.

`Partida(int num, String estado)`

Constructor que inicializa una nueva partida.

Parámetros:

- `num`: El número de la partida.

- **estado**: El resultado de la partida.

Métodos

void setFinal_Partida(int pc, int jugador)

Establece el resultado final de la partida en función de la jugada de los jugadores.

Parámetros:

- **pc**: El valor de la jugada de la computadora.
- **jugador**: El valor de la jugada del jugador.

String getFinal_Partida()

Obtiene el estado final de la partida.

Returns:

- El resultado de la partida.

void setNum(int num)

Establece el número de la partida.

Parámetros:

- **num**: El nuevo número de la partida.

int getNum()

Obtiene el número de la partida.

Returns:

- El número de la partida.

void guardarHistorial(Partida act, String direc_historial)

Guarda el historial de la partida en un archivo.

Parámetros:

- **act**: La partida que se quiere guardar.
- **direc_historial**: La dirección del archivo donde se guarda el historial.

Clase Resultado_PC

Paquete: javax.swing.JFrame

Extiende: JFrame

Descripción

La clase `Resultado_PC` representa la ventana de resultados después de una partida contra la computadora. Muestra las jugadas realizadas, el resultado de la partida y permite al usuario volver a jugar o terminar el juego y regresar al menú.

Constructores

`Resultado_PC()`

Constructor que inicializa la ventana de resultados.

Métodos

`void setRacha(int racha)`

Establece la racha de victorias.

Parámetros:

- `racha`: La racha a establecer.

`void setNum_partida(int num_partida)`

Establece el número de la partida actual.

Parámetros:

- `num_partida`: El número de partida a establecer.

`void setJugada_PC(int jugada)`

Establece la jugada realizada por la computadora.

Parámetros:

- `jugada`: La jugada a establecer.

```
void setJugada_Usuario(int jugada)
```

Establece la jugada realizada por el usuario.

Parámetros:

- `jugada`: La jugada a establecer.

```
void setUsuario(String usuario)
```

Establece el nombre del usuario.

Parámetros:

- `usuario`: El nombre del usuario a establecer.

```
void setFinal_Partida(String final_partida)
```

Establece el resultado final de la partida.

Parámetros:

- `final_partida`: El resultado a establecer.

```
void setDirec_Logros(String direc_logros)
```

Establece la dirección del archivo de logros.

Parámetros:

- `direc_logros`: La dirección a establecer.

```
void setDirec_Historial(String direc_historial)
```

Establece la dirección del archivo de historial.

Parámetros:

- `direc_historial`: La dirección a establecer.

```
void cargar_datos()
```

Carga los datos de la partida en la interfaz gráfica.

```
void obtener_logro(ArrayList<Logro> lista, int racha, int partida)
```

Obtiene y muestra los logros obtenidos basados en la racha de victorias y la partida actual.

Parámetros:

- **lista**: Lista de logros.
- **racha**: La racha actual del jugador.
- **partida**: El número de la partida.

static void main(String[] args)

Método principal que inicia la aplicación y muestra la ventana de resultados.

Parámetros:

- **args**: Argumentos de línea de comandos.

Clase Resultado_Torneo

Paquete: javax.swing.JFrame

Extiende: JFrame

Descripción

La clase **Resultado_Torneo** representa la ventana de resultados de un torneo entre dos usuarios. Muestra las jugadas realizadas y el resultado de la partida, además de permitir a los usuarios volver a jugar o ver el historial.

Constructores

Resultado_Torneo()

Constructor que inicializa la ventana de resultados del torneo.

Métodos

void setResultado(String res)

Establece el resultado de la partida.

Parámetros:

- `res`: Resultado a establecer.

`void setJugada_Usuario1(int jugada_usuario1)`

Establece la jugada realizada por el primer usuario.

Parámetros:

- `jugada_usuario1`: Jugada a establecer.

`void setJugada_Usuario2(int jugada_usuario2)`

Establece la jugada realizada por el segundo usuario.

Parámetros:

- `jugada_usuario2`: Jugada a establecer.

`void setDirec_Historial(String direc_historial)`

Establece la dirección del archivo de historial.

Parámetros:

- `direc_historial`: Dirección a establecer.

`void setRacha(int racha)`

Establece la racha de victorias.

Parámetros:

- `racha`: Racha a establecer.

`void setNum_partida(int num_partida)`

Establece el número de la partida actual.

Parámetros:

- `num_partida`: Número de partida a establecer.

`void setUsuario1(String usuario1)`

Establece el nombre del primer usuario.

Parámetros:

- `usuario1`: Nombre del usuario a establecer.

`void setUsuario2(String usuario2)`

Establece el nombre del segundo usuario.

Parámetros:

- `usuario2`: Nombre del usuario a establecer.

`void cargar_datos()`

Carga los datos de las jugadas y el resultado en la interfaz gráfica.

`static void main(String[] args)`

Método principal que inicia la aplicación y muestra la ventana de resultados del torneo.

Parámetros:

- `args`: Argumentos de línea de comandos.

Clase Usuario

Paquete: `java.lang.Object`

Extiende: `Object`

Descripción

La clase `Usuario` representa a un usuario del juego. Cada usuario tiene un nombre y una contraseña, y se pueden realizar operaciones relacionadas con la verificación y el registro de usuarios.

Constructores

`Usuario(String nombre, String contrasena)`

Constructor que inicializa un nuevo usuario.

Parámetros:

- **nombre**: El nombre del usuario.
- **contrasena**: La contraseña del usuario.

Métodos

void setNombre(String nombre)

Establece el nombre del usuario.

Parámetros:

- **nombre**: El nuevo nombre del usuario.

void setContrasena(String contrasena)

Establece la contraseña del usuario.

Parámetros:

- **contrasena**: La nueva contraseña del usuario.

String getNombre()

Obtiene el nombre del usuario.

Returns:

- El nombre del usuario.

String getContrasena()

Obtiene la contraseña del usuario.

Returns:

- La contraseña del usuario.

boolean Usuario_Existe(File archivo, String nombre)

Verifica si un usuario existe.

Parámetros:

- **archivo**: El archivo que contiene los usuarios.
- **nombre**: El nombre del usuario a verificar.

Returns:

- **true** si el usuario existe, **false** en caso contrario.

boolean Confirmar_Contrasena(File archivo, String nombre, String contrasena)

Confirma si la contraseña corresponde al usuario.

Parámetros:

- **archivo**: El archivo que contiene los usuarios registrados.
- **nombre**: El nombre del usuario.
- **contrasena**: La contraseña a verificar.

Returns:

- **true** si la contraseña es correcta, **false** en caso contrario.

void Registrar_Usuario(File archivo, String nombre, String contrasena, String direccionPrincipal)

Registra un nuevo usuario y crea sus archivos y carpetas.

Parámetros:

- **archivo**: El archivo donde se guardará el nuevo usuario.
- **nombre**: El nombre del nuevo usuario.
- **contrasena**: La contraseña del nuevo usuario.
- **direccionPrincipal**: La dirección principal donde se crearán los archivos del usuario.