¿? ADIVINA QUIEN ¿?

Proyecto

Reporte de la elaboración paso a paso del juego ¿? ADIVINA QUIEN ¿?, proyecto para la clase Programación Orientada A Objetos.

¿? ADIVINA QUIEN ¿?

EL juego "¿?ADIVINA QUIEN¿? " , consta en adivinar un personaje mediante la realización de preguntas sobre sus características físicas. Conforme se van realizando las preguntas se van eliminando a los personajes que no cuentan con las características del personaje a encontrar, siempre y cuando la pregunta realizada sea asertiva a la descripción del personaje. De esta manera gana el jugador que haya eliminado a todos los personajes que no cumplen las características del personaje buscado, dejando únicamente en pie al mismo.
Este programa consta de dos formas de juego diferentes, la primera es "La básica" , esta forma de juego consiste en adivinar al personaje que piensa la computadora en 4 turnos. La segunda forma es "La master" que consiste en jugar contra la maquina sin limite de turnos ganas si adivinas primero el personaje y pierdes si la maquina adivina primero tu personaje.
Para la elaboración de este juego realice 13 interfaces gráficas, utilizando el lenguaje de programación java utilizando como compilador NetBeans. Consta de un menú principal que permite iniciar sesión, jugar en modo libre, registrarte, ver las estadísticas generales y salir. A continuación veremos cómo funcionan las 12 interfaces a las que podemos acceder a través del Menú principal.

Menú

Consta de 5 jButton y 2 JLabel y la variable estática libre que sirve para posteriormente registrar o no los datos del jugador.

Conceptos

Para la elaboración de este programa fue necesario la aplicación de los siguientes conceptos.

Clases y objetos: Una clase es una especie de estructura que agrupa datos y funciones. Un objeto es una variable cuyo tipo de dato es una clase.

Interfaz y encapsulamiento: Las variables de instancia se suelen definir como miembros privados para limitar su accesibilidad y así evitar que puedan ser manipuladas desde fuera de la clase. Uno de los objetos que buscamos cuando programamos clases es encapsular la complejidad que emerge de las operaciones asociadas a sus atributos.

Polimorfismo: Es la propiedad que tienen los objetos de reconocerse como miembros de una determinada clase. Ante la invocación de cualquiera de sus métodos siempre reaccionan como su propia clase lo define.

En su constructor después de initComponents();

Puse la siguiente sentencia

AudioClip sonido;

sonido = java.applet.Applet.newAudioClip(getClass().getResource("/adivinaq/Sonido/juego.wav"))
sonido.play();

Con la finalidad de que el juego tuviera una musica de fondo durante su ejecución.

Botón Jugar Modo Libre

Contenido:

```
libre=1;
MENU_3 M3=new MENU_3();
 M3.setVisible(true);
 this.setVisible(false);
```

Función:

Lo que hace es mandarnos a MENU_3 que es la interface previa que iniciar el juego.

Botón Iniciar Sesión

Contenido:

```
libre=0;
 IniciarC I=new IniciarC();
I.setVisible(true);
this.setVisible(false);
```

Función:

Lo que hace es mandarnos a IniciarC que es la interface que te permite Iniciar Sesion con tu usuario para así poder registrar los juegos, jugados, ganados o perdidos y tus puntos.

Botón Crear Usuario

Contenido:

```
libre=0;
 Rejistro r=new Rejistro();
 r.setVisible(true);
 this.setVisible(false);
```

Función:

Lo que hace es mandarnos a Rejistro que es la interface que te permite Crear tu usuario para así poder Iniciar Sesión y poder registrar los juegos, jugados, ganados o perdidos y tus puntos.

Conseptos

Herencia: Permite definir clases en función de otras ya existentes, en java todas las clases heredan de una clase base llamada object. Por esto, los métodos definidos en Object son comunes a todas las otras clases.

Clases abstractas: Una clase abstracta es una clase que tiene métodos que no pueden ser desarrollados por falta de información concreta. Estos métodos se llaman "métodos abstractos" y deben desarrollarse en las subclases, cuando está información este disponible.

Excepciones:Las excepciones son un mecanismos de tratamiento de error a través del cual los métodos pueden finalizar abruptamente ante la ocurrencia de una situación anómala que imposibilite su normal de desarrollo

Botón Estadisticas

Contenido:

```
EstadisticaG EG=new EstadisticaG();
        EG.setVisible(true);
        this.setVisible(false);
```

Lo que hace es mandarnos a EstadisticaG que es la interface que te permite ver los juegos, jugados, ganados o perdidos y los puntos de todos los usuarios registrados.

Botón Salir

Contenido:

```
System.exit(0);
```

Función:

Lo que hace es finalizar el programa.

Registro

Esta interface cuenta con 3 JTextField que reciben la información solicitada al usuario y un JPasswordField que recibe la contraseña.

Botón Entrar

Contenido:

```
 nombre=this.name.getText();
      edad=this.ages.getText();
      correo=this.mail.getText();
      contrasenia=this.jPasswordField1.getText();

      juegosg=String.valueOf(juegosG);
      juegosp=String.valueOf(juegosP);
      juegosj=String.valueOf(juegosJ);
```

Funciones mas utilizadas

this.setLocationRelativeTo(null);

Esto fue para centrar todas las ventanas.

Para quitarles lo botones de minimizar, maximizar, cerrar. A cada JFrame le modifique la propiedad undecorated, excepto a las ventanas de los juegos.

Botón X en la mayoría de los casos sirve para regresar al menú o ventana anterior la única excepción fue el Menu2 ya que en esta ventana este botón sirve para finalizar el programa

Variables estáticas

Menu

static int libre;

Sirve para saber si es juego libre o no

IniciarC

static String usuario_adivina[] = new String[8] ;

Sirve para saber que inicio sesión

MENU_3

static int dificultad;

Sirve para saber qui tipo de juego se jugó

JUEGO Y Juego1

static int puntos,personaje;

static int personaje,personajec,puntos, puntosc;

Indican a Puntuacion y puntosp los resultados del juego

```
puntos=String.valueOf(Puntos);

try{
   File f=new File ("Registro.txt");
   try (FileWriter fw = new FileWriter(f,true);
         BufferedWriter bw = new BufferedWriter(fw)) {
      if (f.exists()){
         bw.newLine();
         bw.write(nombre+','+edad+','+correo+','+contrasenia+','+juegosg+','+juegosp+','+puntos+','+juegosj);
      }

      else{
         bw.write(nombre+','+edad+','+correo+','+contrasenia);
      }
      bw.close();
      fw.close();
   }

}
catch(IOException e){
   System.out.println(e);

}
IniciarC I=new IniciarC();
I.setVisible(true);
this.setVisible(false);
```

Función:

Lo que hace es almacenar en un archivo txt los datos del jugador y agrega los juegos jugados los juegos ganados los juegos perdidos y los puntos inicializándolos en 0. Al finalizar manda a llamar a la interface IniciarC para que el jugador inicie sesión y comience a acumular sus puntos.

IniciarC

Esta interface cuenta con 1 JTextField que recibe el nombre o seudónimo del usuario, un JPasswordField que recibe la contraseña y un JLabel que inica si la contraseña o usuario son incorrectos.

Tambien tiene una variable static String usuario_adivina[] = new String[8] ; que guarda la línea del usuario para poderlo identificar y saber quién inicio sesión para posterior mente modificar sus datos en caso de haber ganado o perdido en un juego y almacenar sus puntos.

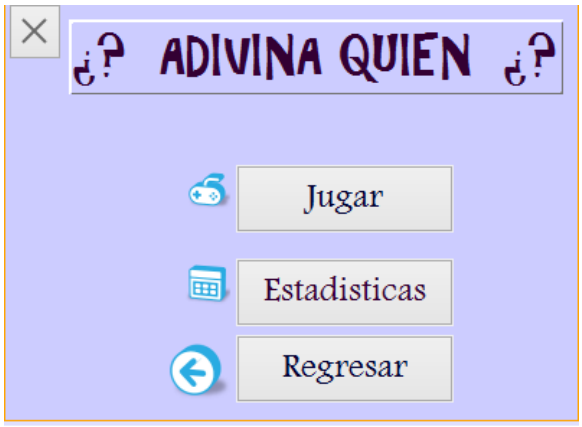
Botón Entrar

```
User=this.Usuario.getText();
      Pas=this.Contrasenia.getText();


       try{
          File f=new File("Registro.txt");
          FileReader fr=new FileReader(f);
          BufferedReader br=new BufferedReader(fr);
          if(f.exists()){
            String linea;
            while((linea=br.readLine())!=null){
               String jugador[]=linea.split(",");
               if((jugador[0].equals(User))&& (jugador[3].equals(Pas))){
                  for(int i=0;i<jugador.length;i++)
```

```
                usuario_adivina[i] = jugador[i];
              Menu2 M2=new Menu2();



              M2.setVisible(true);
              this.setVisible(false);
            }
          }
          ErrorU.setText("Usuario o Contraseña Incorrectos!!");

      }
      else{
          System.out.println("No hay jugadores registrados");

      }
      br.close();
      fr.close();
  }
  catch(Exception e){
      System.out.println(e);
      e.printStackTrace();

  }
```

Su función es leer y comparar si el usuario está registrado y que su contraseña esta correcta. Si es correcta manda abre la interface Menu2 que nos lleva al juego.

Menu2



Botón Jugar

Contenido:

```
libre=0;
        MENU_3 m3=new MENU_3();
         m3.setVisible(true);
         this.setVisible(false);
```

Función:

Lo que hace es mandarnos a MENU_3 que es la interface previa que iniciar el juego.

Botón Estadisticas

Contenido:

```
         Estadisticas E= new Estadisticas();
        E.setVisible(true);
        this.setVisible(false);
```

Función:

Lo que hace es mandarnos a Estadisticas que es la interface que le permite al jugador ver sus juegos ganados, perdidos, jugados y sus puntos.

Botón Regresar

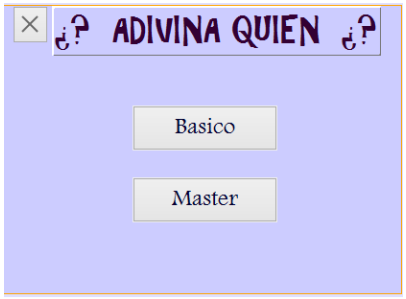
Contenido:

```
         Menu M=new Menu();
         M.setVisible(true);
         this.setVisible(false);
```

Función:

Lo que hace es mandarnos Menu que es la interface del menú principal.

MENU_3



Botón Basico

Contenido:

```
dificultad=1;
JUEGO J=new JUEGO();
J.setVisible(true);
this.setVisible(false);Función:
```

Lo que hace es mandarnos a JUEGO que es la interface del juego "más fácil" el cual solo tienes que adivinar el personaje de la máquina.

Botón Master

Contenido:

```
dificultad=2;
Juego1 J1=new  Juego1();
J1.setVisible(true);
this.setVisible(false);
```

Lo que hace es mandarnos a Juego1 que es la interface del juego "más difícil" en el cual juegas contra la máquina.

JUEGO

Botón Iniciar Partida

Contenido:

```
Random aleatorio = new Random(System.currentTimeMillis());
personaje = aleatorio.nextInt(16)+1;
for(int i=0;i<16;i++){
vector[i]=1;
 preguntas.setVisible(true);
Iniciar.setVisible(false);
```

Lo que hace es generar aleatoriamente el número del personaje que el usuario tiene que adivinar y al vector que contiene el mismo número de jugadores inicializa sus casillas en 1 y hace visible el panel de preguntas haciéndose invisible este botos..

Panel Preguntas

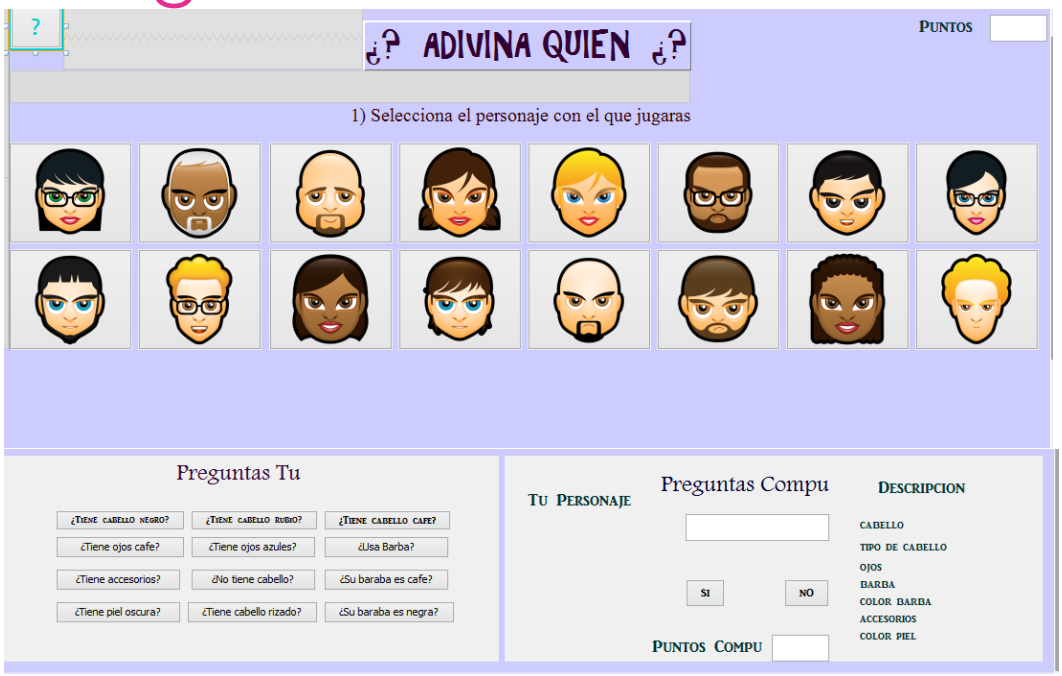
El panel consta de16 botones cada uno una pregunta escrita, cada pegunta describe a uno o más personajes. Cada botón cuenta con un switch case que recibe el número del personaje si el botón presionado concuerda con la descripción del personaje seleccionado, hace que desaparcan los personajes que no cuentan con las caracterices de este, también modifica al vector en las posiciones de los personajes eliminados poniéndolos en 0 y en ocasiones desaparece algunas preguntas. Cabe mencionar que cada personaje es un botón que no tiene funcionalidad al hacerle click solo desaparece según las circunstancias. En caso de no ser la pregunta correcta manda a un JLabel un mensaje diciendo que característica no es correcta. Al finalizar el switch case se evalúa a través de un id si el número de jugadas es menor a 4 si es menor a 4 , recorre el arreglo para ver si

queda un solo1, si es así manda a llamar a la interface Puntos, si no continua el juego. Si el número de jugadas es mayor a 4 llama a la interface puntosp.

Botón ?

Manda a llamar Reglas r=new Reglas(); que es el instructivo del juego.

Juego1



Inicia el juego con los paneles de preguntas invisibles solo se ven los botones de los personajes, para que el usuario seleccione su personaje que será el que la maquina adivine.

Botones Personajes

Contenido:

```
            Menu9.setVisible(false);
        jPanel2.setVisible(true);
       Random aleatorio = new Random(System.currentTimeMillis());
    personaje = aleatorio.nextInt(16)+1;
      personajec=15;
      turno=1;
      Imagen.setIcon(new javax.swing.ImageIcon(getClass().getResource("/adivinaq/IMAGENES/1.5.png")));
      D1.setText("CAFE");
      D2.setText("RIZADO");
      D3.setText("CAFE");
      D4.setText("NO");
      D5.setText("NO");
      D6.setText("NO");
      D7.setText("SI");
```

```
for(int i=0;i<16;i++){
        vectorU [i]=1;
    }
```

Lo que haces generar aleatoriamente el número de personaje que el usuario tiene que adivinar y guarda en personajes el número de personaje que el usuario eligió también inicializa el arreglo vector en unos y hace visible el panel de preguntas con el cual el usuario podrá adivinar el personaje

Panel Preguntas

El panel de preguntas consta de 16 botones cada uno con una pregunta escrita y cada pregunta describe uno o más personajes para saber si el botón tiene la descripción del personaje que se busca cuenta con un switch case que recibe el número de personaje que Se generó aleatoriamente al iniciar el programa de esta manera hay 16 Case cada uno es un personaje si el botón presionado concuerda con la descripción del personaje los personajes que no concuerden con esa descripción se borraran los iconos de los personajes y sus posiciones en el vector serán igual a cero posteriormente se evalúa Cuántos unos que dan en el vector Sí sólo queda uno se le llama a la interfase juntos si no se vuelve invisible el panel 1 y se llama al método JuegoC volviéndose visible el panel 2 en el cual se muestra la imagen del personaje seleccionado por el usuario la pregunta que realiza la máquina la descripción del personaje y dos botones uno que dice sí y el otro dice no.

Metodo JuegoC

El método juegos e consta de un switch Case que recibe la variable personaje de esta manera cada caso es un personaje y dentro de cada personaje existen cuatro posibles preguntas cada una de estas preguntas describen al personaje de manera correcta y en algunas ocasiones hace preguntas incorrectas, Qué conforma el jugador va indicando si son correctas o incorrectas las preguntas van cambiando forzando al jugador a adivinar al personaje en menos de cuatro turnos así el juego tiene mayor dificultad.

Botón Si

Al presionar el botón sí si la pregunta es correcta la computadora obtiene 10 puntos y la variable pregunta vale 2 cabe mencionar que la variable pregunta Al iniciar el juego vale 1 se activa el panel 2 dándole el turno al usuario en caso de que el usuario ingresé si, sin que su personaje tenga esa característica el programa lo contara como trampa y sus puntos se convertirán en 0

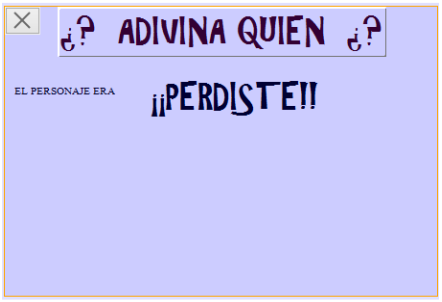
Botón No

Al presionar el botón sí si la pregunta es correcta la computadora obtiene 10 puntos y la variable pregunta vale 2 cabe mencionar que la variable pregunta Al iniciar el juego vale 1 se activa el panel 2 dándole el turno al usuario en caso de que el usuario ingresé si, sin que su personaje tenga esa característica el programa lo contara como trampa y sus puntos se convertirán en 0

Botón ?

Manda a llamar reglas2 r=new reglas2(); que es el instructivo del juego.

Puntuacion y puntosp




```
initComponents();
this.setLocationRelativeTo(null);
int j= MENU_3. dificultad;
int l= Menu .libre;


String varClave = IniciarC.usuario_adivina[0];
jg=IniciarC.usuario_adivina[4];
jj=IniciarC.usuario_adivina[7];
p=IniciarC.usuario_adivina[6];
if(j==1){
   PER=JUEGO.personaje;
   Pun = JUEGO.puntos;
}
else
   PER=Juego1.personaje;
   Pun=Juego1.puntos;

switch (PER){
   case 1:
      icon.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/adivinaq/IMAGENES/1.1.png")));
      break;
      case 2:
      icon.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/adivinaq/IMAGENES/1.10.png")));
      break;
      case 3:
```

```
            icon.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/adivinaq/IMAGENES/1.11.png")));
            break;
            case 4:
            icon.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/adivinaq/IMAGENES/1.3.png")));
            break;
            case 5:
            icon.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/adivinaq/IMAGENES/1.4.png")));
            break;
            case 6:
            icon.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/adivinaq/IMAGENES/1.9.png")));
            break;
            case 7:
            icon.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/adivinaq/IMAGENES/1.16.png")));
            break;
            case 8:
            icon.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/adivinaq/IMAGENES/2.1.png")));
            break;
            case 9:
            icon.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/adivinaq/IMAGENES/1.14.png")));
            break;
            case 10:
            icon.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/adivinaq/IMAGENES/1.13.png")));
            break;
            case 11:
            icon.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/adivinaq/IMAGENES/1.6.png")));
            break;
            case 12:
            icon.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/adivinaq/IMAGENES/1.15.png")));
            break;
```

```
            case 13:
              icon.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/adivinaq/IMAGENES/1.12.png")));
             break;
            case 14:
              icon.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/adivinaq/IMAGENES/1.19.png")));
             break;
             case 15:
              icon.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/adivinaq/IMAGENES/1.5.png")));
             break;
             case 16:
              icon.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/adivinaq/IMAGENES/1.7.png")));
             break;

        }


        System.out.println("l= "+l);
        if(l==0){
           switch (j){
              case 1:
                 pp=String.valueOf(Pun);
                 puntuacion.setText("Puntuacion: "+pp);
                 nombre1.setText(varClave);
           try{
              File f=new File ("Registro.txt");
              FileReader fr=new FileReader(f);
              BufferedReader br=new BufferedReader(fr);


              StringBuffer strbf = new StringBuffer();

              if (f.exists()){
                 String linea;
```

```
    while((linea=br.readLine())!=null){

        String jugador[]=linea.split(",");

        if(jugador[0].equals(varClave)){

            Puntos=Double.parseDouble(p);
            JuegosG=Double.parseDouble(jg);
            JuegosJ=Double.parseDouble(jj);
            Puntos=Puntos + Pun;
            JuegosG=JuegosG+1;
            JuegosJ=JuegosJ+1;
            jj=String.valueOf(JuegosJ);
            jg=String.valueOf(JuegosG);
            p=String.valueOf(Puntos);
            strbf.append(jugador[0]+","+jugador[1]+","+jugador[2]+","+
                    jugador[3]+","+jg+","+jugador[5]+","+ p+","+jj+"\n");

        }else{

        strbf.append(linea+"\n");



        }
    }
    br.close();
    fr.close();

    FileWriter fw = new FileWriter(f,false);
BufferedWriter bw = new BufferedWriter(fw);
PrintWriter pw=new PrintWriter(bw);
pw.println(strbf.toString());
    bw.close();
    fw.close();
```

```
    }
}
catch(IOException e){
    System.out.println(e);

}
        break;
    case 2:
pp=String.valueOf(Pun);
puntuacion.setText("Puntuacion: "+pp);
nombre1.setText(varClave);
try{
    File f=new File ("Registro.txt");
    FileReader fr=new FileReader(f);
    BufferedReader br=new BufferedReader(fr);


    StringBuffer strbf = new StringBuffer();

    if (f.exists()){
        String linea;
        while((linea=br.readLine())!=null){

            String jugador[]=linea.split(",");

            if(jugador[0].equals(varClave)){

                Puntos=Double.parseDouble(p);
                JuegosG=Double.parseDouble(jg);
                JuegosJ=Double.parseDouble(jj);
                Puntos=Puntos + Pun;
                JuegosG=JuegosG+1;
                JuegosJ=JuegosJ+1;
                jj=String.valueOf(JuegosJ);
                jg=String.valueOf(JuegosG);
                p=String.valueOf(Puntos);
```

```
                strbf.append(jugador[0]+","+jugador[1]+","+jugador[2]+","+
                        jugador[3]+","+jg+","+jugador[5]+","+ p+","+jj+"\n");

            }else{

            strbf.append(linea+"\n");


            }
          }
          br.close();
          fr.close();

          FileWriter fw = new FileWriter(f,false);
       BufferedWriter bw = new BufferedWriter(fw);
       PrintWriter pw=new PrintWriter(bw);
       pw.println(strbf.toString());
          bw.close();
          fw.close();


       }
    }
    catch(IOException e){
       System.out.println(e);

    }
          break;
    }

}
else{
nombre1.setText("Invitado");
ppp=String.valueOf(Pun);
puntuacion.setText("Puntuacion: "+ppp);
```

```
}
```

En ambas interfaces si el juego es libre solo se imprime la puntuación obtenida si no envía el nombre del usuario, su puntuación y se lee el archivo se localiza al usuario que inicio sesión se guarda en strbf.append(jugador[0]+","+jugador[1]+","+jugador[2]+","+

```
jugador[3]+","+jg+","+jugador[5]+","+ p+","+jj+"\n");
```

lo que se modificaría del archivo y se reescribe en el archivo. Lo único cambia en Puntuación y puntosp es que en uno de guardan JuegoG y en el otro JuegosP.