

PSP\Tarea4PSP\src\tarea4pspsolucion\GestionPeticiones.java

```
package tarea4pspsolucion;

import java.util.logging.Logger;
import java.util.Random;
import java.util.UUID;
import java.util.concurrent.ConcurrentHashMap;
import java.util.logging.Level;
import java.util.regex.*;

public class GestionPeticiones {

    private final ConcurrentHashMap<String, SesionJuego> sesiones;
    private final Logger logger;
    private final String emailRegex = "^([\\w.-]+@[\\w.-]+\\.?[a-zA-Z]{2,})$";
    private final String passwordRegex = "^(?=.*[A-Za-z])(?=.*\\d)[A-Za-z\\d]{6,}$";

    public GestionPeticiones(ConcurrentHashMap<String, SesionJuego> sesiones, Logger logger) {
        this.sesiones = sesiones;
        this.logger = logger;
    }

    public String manejarLogin(String cuerpo, String sessionId) {
        int codigo = 200;
        boolean login = false;
        String respuestaHTML = "<p>Introduce email/contraseña para realizar login o registro</p>";

        try {
            if (!cuerpo.isEmpty()) {
                String email = (cuerpo.split("&")[0]).split("=")[1];
                String contrasena = (cuerpo.split("&")[1]).split("=")[1];

                email = email.replace("%40", "@"); // Reemplazamos el valor codificado por un arroba
                if (GestionUsuarios.comprobarLogin(email, contrasena)) {
                    sessionId = UUID.randomUUID().toString();
                }
            }
        }
    }
}
```

```
        sesiones.put(sessionId, new SesionJuego());
        login = true;
    } else {
        respuestaHTML = "<p>Email o contraseña incorrectos. Intente de nuevo</p>";
    }
}
} catch (Exception e) {
    respuestaHTML = "<p>Error procesando tu petición. Intenta de nuevo. </p>";
    codigo = 400;
    //e.printStackTrace(); // Muestra errores en la consola.
}
return construirRespuesta(codigo, (login ? Paginas.html_index : Paginas.html_LoginRegistro(respuestaHTML)),
    sessionId);
}

public String manejarRegistro(String cuerpo, String sessionId) {
    int codigo = 200;
    String respuestaHTML = "<p>Introduce email/contraseña para realizar login o registro</p>";

    try {
        if (!cuerpo.isEmpty()) {
            String email = (cuerpo.split("&")[0]).split("=")[1];
            String contrasena = (cuerpo.split("&")[1]).split("=")[1];

            email = email.replace("%40", "@"); // Reemplazamos el valor codificado por un arroba

            if (Pattern.matches(emailRegex, email) && Pattern.matches(passwordRegex, contrasena)) {
                if (GestionUsuarios.realizarRegistro(email, contrasena)) {
                    respuestaHTML = "<p>Registro realizado correctamente. Inicie sesión.</p>";
                } else {
                    respuestaHTML = "<p>Error en el registro. Intente de nuevo.</p>";
                }
            } else {
                respuestaHTML = "<p>Error en el formato del email o la contraseña.</p>";
            }
        }
    } catch (Exception e) {
```

```
    respuestaHTML = "<p>Error procesando tu petición. Intenta de nuevo. </p>";
    codigo = 400;
    //e.printStackTrace(); // Muestra errores en la consola.
}

return construirRespuesta(codigo, Paginas.html_LoginRegistro(respuestaHTML), sessionId);
}

public String manejarAdivina(String cuerpo, SesionJuego sesion, String sessionId) {
    int codigo = 200;
    String respuestaHTML = "<p>Introduce un número para empezar el juego.</p>";
    String pagina;
    if (sesion != null) {
        if (sesion.numeroSecreto == 0) {
            sesion.numeroSecreto = new Random().nextInt(100) + 1; // Genera un número aleatorio al iniciar el juego.
            sesion.intentosAdivina = 0; // Resetea los intentos.
        }
        try {
            if (!cuerpo.isEmpty()) {
                System.out.println("Cuerpo Adivina: " + cuerpo);
                int numeroUsuario = Integer.parseInt(cuerpo.split("=")[1]);
                sesion.intentosAdivina++;

                if (numeroUsuario == sesion.numeroSecreto) {
                    respuestaHTML = "<p>Â¡Felicidades! Has acertado el número " + sesion.numeroSecreto + " en "
                        + sesion.intentosAdivina + " intentos.</p>";
                    sesion.numeroSecreto = 0; // Reinicia el juego.
                } else if (sesion.intentosAdivina >= 10) {
                    respuestaHTML = "<p>No has acertado en 10 intentos. El número era " + sesion.numeroSecreto
                        + ". Pulsa <a href='/adivina'>aquí</a> para reiniciar el juego.</p>";
                    sesion.numeroSecreto = 0; // Reinicia el juego.
                } else {
                    respuestaHTML = "<p>Intento " + sesion.intentosAdivina + ": El número es "
                        + (numeroUsuario < sesion.numeroSecreto ? "mayor" : "menor") + "</p>";
                }
            }
        } catch (Exception e) {
```

```
    respuestaHTML = "<p>Error procesando tu número. Intenta de nuevo. </p>";
    codigo = 400;
    //e.printStackTrace(); // Muestra errores en la consola.
    logger.log(Level.SEVERE, "Error juego Adivina en la línea " + e.getStackTrace()[3].getLineNumber()
        + ": El valor introducido no es correcto. Valor recibido:" + cuerpo.split("=")[1]);
}
pagina = construirRespuesta(codigo, Paginas.generarHtmlAdivina(respuestaHTML), sessionId);
} else {
    respuestaHTML = "<p>Debes tener cuenta para jugar. Introduce email/contraseña para realizar login o registro</p>";
    pagina = construirRespuesta(codigo, Paginas.html_LoginRegistro(respuestaHTML), sessionId);
}

return pagina;

}
```

```
public String manejarDatos(String cuerpo, SesionJuego sesion, String sessionId) {
    int codigo = 200;
    String respuestaHTML = "<p>Pulsa el botón para lanzar los dados. </p> ";
    String pagina;

    if (sesion != null) {
        try {
            if (!cuerpo.isEmpty()) {
                Random random = new Random();
                int dadoUsuario = Integer.parseInt(cuerpo.split("=")[1]);
                int dadoServidor = random.nextInt(6) + 1;

                if (dadoUsuario > dadoServidor) {
                    sesion.marcadorUsuarioDados++;
                    sesion.rondaDados++;
                    respuestaHTML = "<p>Ronda " + sesion.rondaDados + " .Ganaste esta ronda. Tu dado: "
                        + dadoUsuario + " - Dado del servidor: " + dadoServidor + " </p> ";
                } else if (dadoUsuario < dadoServidor) {
                    sesion.marcadorServidorDados++;
                    sesion.rondaDados++;
                    respuestaHTML = "<p>Ronda " + sesion.rondaDados + " .Perdiste esta ronda. Tu dado: "
```

```

        + dadoUsuario + " - Dado del servidor: " + dadoServidor + "  </p>    ";
    } else {
        respuestaHTML = "<p>Ronda " + sesion.rondaDados + " .Empate en esta ronda. Ambos sacaron: "
            + dadoUsuario + "  </p>    ";
    }

    if (sesion.rondaDados == 5) {
        if (sesion.marcadorUsuarioDados > sesion.marcadorServidorDados) {
            respuestaHTML += "<p>Â¡Ganaste el juego! Marcador final: " + sesion.marcadorUsuarioDados
                + " - " + sesion.marcadorServidorDados
                + ". Vuelve a pulsar el botón para jugar de nuevo.</p>    ";
        } else {
            respuestaHTML += "<p>Perdiste el juego. Marcador final: " + sesion.marcadorUsuarioDados
                + " - " + sesion.marcadorServidorDados
                + ". Vuelve a pulsar el botón para jugar de nuevo.</p>    ";
        }
        sesion.marcadorUsuarioDados = 0;
        sesion.marcadorServidorDados = 0;
        sesion.rondaDados = 0;
    }
}

} catch (Exception e) {
    respuestaHTML = "<p>Error procesando tu elección. Intenta de nuevo.</p>";
    codigo = 400;
    //e.printStackTrace(); // Muestra errores en la consola.
    logger.log(Level.SEVERE, "Error juego Dados en la línea " + e.getStackTrace()[3].getLineNumber()
        + ": El valor introducido no es correcto. Valor recibido:" + cuerpo.split("=")[1]);
}

pagina = construirRespuesta(200, Paginas.generarHtmlDados(respuestaHTML), sessionId);
} else {
    respuestaHTML = "<p>Debes tener cuenta para jugar. Introduce email/contraseña para realizar login o registro</p>";
    pagina = construirRespuesta(codigo, Paginas.html_LoginRegistro(respuestaHTML), sessionId);
}

return pagina;
}

```

```

public String manejarPPT(String cuerpo, SesionJuego sesion, String sessionId) {

```

```
int codigo = 200;
String[] opciones = {"Piedra", "Papel", "Tijeras"};
String respuestaHTML = "<p>Pulsa un botón para jugar.</p> ";
String pagina;

if (sesion != null) {
    try {
        if (!cuerpo.isEmpty()) {
            String eleccionUsuario = cuerpo.split("=")[1];
            String eleccionServidor = opciones[new Random().nextInt(3)];

            if (eleccionUsuario.equals(eleccionServidor)) {
                respuestaHTML = "<p>Ronda " + sesion.rondaPPT + " .Empate en esta ronda. Ambos eligieron: "
                    + eleccionServidor + "</p> ";
            } else if ((eleccionUsuario.equals("Piedra") && eleccionServidor.equals("Tijeras"))
                || (eleccionUsuario.equals("Papel") && eleccionServidor.equals("Piedra"))
                || (eleccionUsuario.equals("Tijeras") && eleccionServidor.equals("Papel"))) {
                sesion.marcadorUsuarioPPT++;
                sesion.rondaPPT++;
                respuestaHTML = "<p>Ronda " + sesion.rondaPPT + " .Ganaste esta ronda. Elegiste: "
                    + eleccionUsuario + " - El servidor eligió: " + eleccionServidor + " </p> ";
            } else {
                sesion.marcadorServidorPPT++;
                sesion.rondaPPT++;
                respuestaHTML = "<p>Ronda " + sesion.rondaPPT + " .Perdiste esta ronda. Elegiste: "
                    + eleccionUsuario + " - El servidor eligió: " + eleccionServidor + " </p> ";
            }
        }

        if (sesion.rondaPPT == 5) {
            if (sesion.marcadorUsuarioPPT > sesion.marcadorServidorPPT) {
                respuestaHTML += "<p>Â¡Ganaste el juego! Marcador final: " + sesion.marcadorUsuarioPPT
                    + " - " + sesion.marcadorServidorPPT
                    + ". Vuelve a pulsar un botón para jugar de nuevo.</p> ";
            } else {
                respuestaHTML += "<p>Perdiste el juego. Marcador final: " + sesion.marcadorUsuarioPPT
                    + " - " + sesion.marcadorServidorPPT
                    + ". Vuelve a pulsar un botón para jugar de nuevo.</p> ";
            }
        }
    }
}
```

```

        }
        sesion.marcadorUsuarioPPT = 0;
        sesion.marcadorServidorPPT = 0;
        sesion.rondaPPT = 0;
    }
}
} catch (Exception e) {
    respuestaHTML = "<p>Error procesando tu elección. Intenta de nuevo.</p> ";
    codigo = 400;
    //e.printStackTrace(); // Muestra errores en la consola.
    logger.log(Level.SEVERE, "Error juego PPT en la línea " + e.getStackTrace()[0].getLineNumber()
        + ": El valor introducido no es correcto. Valor recibido:");
}
pagina = construirRespuesta(codigo, Paginas.generarHtmlPpt(respuestaHTML), sessionId);
} else {
    respuestaHTML = "<p>Debes tener cuenta para jugar. Introduce email/contraseña para realizar login o registro</p>";
    pagina = construirRespuesta(codigo, Paginas.html_LoginRegistro(respuestaHTML), sessionId);
}

return pagina;
}

public String manejarLogout(String sessionId) {
    int codigo = 200;
    String respuestaHTML = "<p>Sesión cerrada correctamente</p>";

    sesiones.remove(sessionId);
    sessionId = " ; MaxAge=0;";

    return construirRespuesta(codigo, Paginas.html_LoginRegistro(respuestaHTML), sessionId);
}

public String construirRespuesta(int codigo, String contenido, String sessionId) {
    return (codigo == 200 ? "HTTP/1.1 200 OK" : "HTTP/1.1 400 Bad Request") + "\n" // Línea inicial
        + "Content-Type: text/html; charset=UTF-8" + "\n" // Metadatos
        + "Content-Length: " + contenido.length() + "\n"
        + "Set-Cookie: sessionId=" + sessionId + "; Path=/;\n"

```

```
+ "Content-Type: text/html; charset=UTF-8\n"  
+ "\n" // Línea vacía  
+ contenido; // Cuerpo  
}  
}
```