

Juego del Ahorcado – Proyecto JSP 2025

1. Elementos que se utilizan para la lógica

El proyecto se organiza en varias capas y tecnologías:

- Archivos JSP (Vista):
 - index.jsp: Página principal con opciones de navegación.
 - registro.jsp: Formulario para registrar usuarios.
 - ahorcado.jsp: Interfaz del juego del ahorcado.
- JavaScript (Interactividad del cliente):
 - MyScript.js: Maneja la lógica del juego en el navegador (mostrar palabra, controlar botones, temporizador, imágenes, pistas, etc.).
- Clases Java (Controladores y Modelos):
 - Conexion.java: Gestiona la conexión con la base de datos MySQL.
 - Controlador.java: Servlet principal que dirige el flujo entre vistas y lógica.
 - PalabraMP.java: Sirve como punto de acceso para obtener palabras (devuelve datos en JSON para el frontend).
 - Registrar.java: Controla el registro de nuevos usuarios.
 - Validar.java: Maneja la validación de credenciales al iniciar sesión.
 - Palabras.java: Clase modelo que representa la entidad Palabra.
 - PalabrasDAO.java: Acceso a datos para la tabla Palabra (CRUD).
 - Usuarios.java: Clase modelo para la entidad Usuario.
 - UsuariosDAO.java: Acceso a datos para la tabla Usuario (CRUD).
- Base de Datos (MySQL):
 - DB_Ahorcado.sql: Script para crear la base de datos, tablas (Usuario, Palabra) y procedimientos almacenados.

2. Imports utilizados en Java

- java.sql: Para manejar la conexión con la base de datos (Connection, PreparedStatement, ResultSet).
- javax.servlet y javax.servlet.http: Para el manejo de servlets y peticiones HTTP.
- java.util: Listas y colecciones para el manejo de usuarios y palabras.
- com.google.gson: Para convertir objetos Java a JSON y enviarlos al frontend.

3. Lógica del programa

3.1. Modelo de datos (bean y DAO)

- Usuarios: Clase que define los atributos de un usuario (codigoUsuario, nombre, apellido, correo, pass) con getters y setters.
- UsuariosDAO: Implementa los métodos CRUD para usuarios, utilizando procedimientos almacenados.
- Palabras: Clase que define los atributos de una palabra (codigoPalabra, palabra, pista1, pista2, pista3).
- PalabrasDAO: Implementa los métodos CRUD para palabras (insertar, eliminar, actualizar, buscar).

3.2. Controladores (Servlets)

- Controlador.java: Centraliza la navegación del sistema. Mediante parámetros que dirigen hacia las JSP utilizados.
- Registrar.java: Procesa el formulario de registro de registro.jsp. Inserta el usuario en la base de datos usando UsuariosDAO.
- Validar.java: Verifica si el correo y contraseña existen en la base de datos. Si son válidas las credenciales nos lleva al menú principal, si no son correctas tira un error.
- PalabraMP.java: Devuelve una lista de palabras con sus pistas en formato JSON para que el JS (MyScript.js) las use en el juego.

3.3. Lógica del juego

En MyScript.js:

- Carga las palabras mediante fetch("PalabraMP").
- Escoge una palabra secreta al azar.
- Muestra guiones bajos y las pistas.
- Controla errores (máximo. 6) y cambia imágenes del ahorcado segun los errores que tenga hasta llegar a 6 donde ya ahorca al muñequito.
- Maneja botones: Iniciar, Reiniciar, Pausar, Salir.
- Temporizador de 10 minutos con cuenta regresiva.
- Modal de victoria y mensajes de derrota.

3.4. Base de Datos (DB_Ahorcado.sql)

- Tabla Usuario: Guarda datos de jugadores registrados.
- Tabla Palabra: Guarda palabras y 3 pistas de las palabras.
- Procedimientos almacenados que use:
 - sp_AgregarUsuario, sp_VerUsuarios, sp_BuscarUsuario, sp_EliminarUsuario, sp_ActualizarUsuario.
 - sp_AgregarPalabra, sp_VerPalabras, sp_BuscarPalabra, sp_EliminarPalabra, sp_ActualizarPalabra.

4. Flujo del programa

1. El usuario entra a index.jsp.
2. Puede registrarse (registro.jsp) Registrar.java guarda datos en DB_Ahorcado.
3. Puede iniciar sesión (Validar.java).
4. Una vez validado, entra al menú y selecciona el juego (ahorcado.jsp).
5. MyScript.js carga las palabras desde PalabraMP.java.
6. El usuario juega al ahorcado: intenta adivinar antes de perder por intentos gastados o tiempo vencido.

5. Mapeo

El mapeo entre las tablas y los objetos Java se realiza manualmente mediante las clases DAO:

- Cada registro de la tabla Usuario se transforma en un objeto de la clase Usuarios.
- Cada registro de la tabla Palabra se transforma en un objeto de la clase Palabras.

Esto se hace recorriendo el ResultSet y asignando valores a los atributos del bean correspondiente. De esta manera se logra un mapeo manual.