

# **DOCUMENTACIÓN AHORCADO**

**PABLO MANUEL OTÓN CARIDAD**

**MANUEL GARCÍA REY**

**ANTÓN FARALDO MOSQUERA**

DOCUMENTACIÓN AHORCADO.....	1
1. Requisitos del Proyecto: Juego "Ahorcado".....	2
A. Capa de Seguridad y Transporte.....	2
B. Persistencia de Datos (Hibernate).....	2
C. Concurrencia y Lógica del Servidor.....	3
D. Interfaz y Cliente.....	3
4.....CONSIDERACIONES EN NUESTRO JUEGO	
	3

## 1. Requisitos del Proyecto: Juego "Ahorcado"

El proyecto consiste en el desarrollo de una solución **cliente/servidor** para el juego del "Ahorcado", integrando persistencia avanzada, seguridad en las comunicaciones y concurrencia.

### A. Capa de Seguridad y Transporte

- **Protocolo de Comunicación:** Implementación de sockets seguros utilizando SSL/TLS para la comunicación entre cliente y servidor.
- **Criptografía Híbrida:** Uso de seguridad en la capa de transporte para proteger el canal.

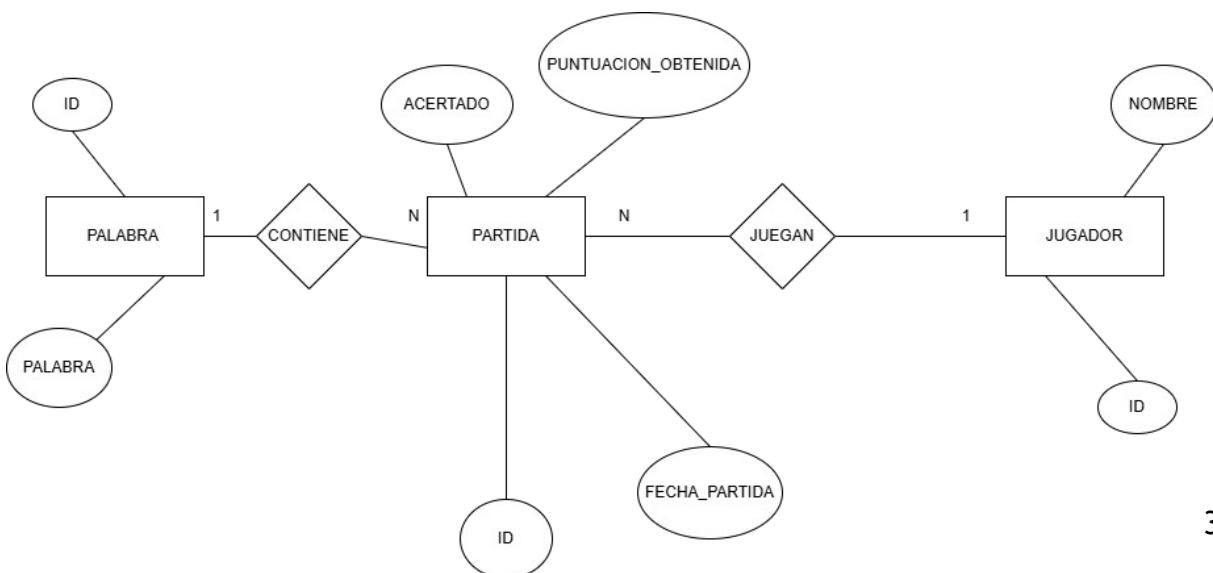
- **Gestión de Claves:** Generación y uso de un almacén de claves JKS (Java KeyStore) para la autenticación del servidor y el cifrado de la sesión.
- **Protección de Datos:** Aplicación de Criptografía Simétrica (AES) para cifrar campos específicos dentro de la base de datos.

## B. Persistencia de Datos (Hibernate)

- **Tecnología:** Uso de **Hibernate** para la gestión de la persistencia sobre la conexión segura.
- **Importación de Datos:** Carga inicial de 50 palabras relacionadas con la naturaleza desde un archivo **JSON** (palabras de entre 6 y 15 caracteres).
- **Entidades a Almacenar:**
  - **Palabras:** Identificador y texto de la palabra.
  - **Jugadores:** Identificador y nombre (con verificación de existencia/alta automática).
  - **Partidas/Puntuaciones:** Fecha y hora, resultado (acierto/fallo) y puntuación global.

## C. Conurrencia y Lógica del Servidor

- **Arquitectura Multihilo:** El servidor debe permitir la conexión simultánea de múltiples clientes.
- **Puerto de Escucha:** La aplicación servidor operará en el puerto **65000**.
- **Mecánica de Juego:**
  - Selección aleatoria de la palabra mediante su ID.
  - Gestión de turnos (en caso de 2 jugadores) y validación de caracteres.
  - Límite de intentos: La **mitad de la longitud** de la palabra elegida.
  - Cálculo de puntos: **1 punto** si la palabra tiene < 10 caracteres; **2 puntos** si tiene 10 o más.



El diagrama superior muestra la arquitectura de datos del sistema. Se destaca la relación de uno a muchos entre **Jugador** y **Partida**, permitiendo la persistencia del histórico de puntuaciones. La entidad **Palabra** actúa como repositorio central para la mecánica del juego, vinculándose lógicamente con cada sesión de partida generada.

## D. Interfaz y Cliente

- **Entorno Gráfico:** La parte cliente debe desarrollarse en un entorno de **ventanas** (Swing o JavaFX).
- **Funcionalidades del Cliente:**
  - Ventana de inicio para introducir el nombre.
  - Interfaz de juego para enviar letras y cancelar partida.
  - Consulta de puntuación global.
- **Feedback:** El servidor enviará mensajes a la consola, mientras que el cliente los recibirá en la interfaz gráfica.

## 4. Consideraciones

### A. Uso único de 2 usuarios

En nuestro caso para hacer más sencillo el inicio de sesión hemos decidido emplear solamente 2 usuarios, para limitar esto hemos creado un login en el que aparecen solo 2 botones, uno se corresponde con el Jugador 1 y otro con el Jugador 2.

De esta manera nos tenemos que evitar registrar un número innecesario de jugadores, ya que como mucho solo pueden jugar 2 a la vez.

### B. Uso de botones para el envío de información

Hemos decidido crear una pantalla de juego un tanto distinta a lo que se pedía, en vez de enviar las letras por cajas de texto hemos asociado cada letra a un botón, facilitando así el envío de información, y abriendo otras funcionalidades, como por ejemplo desactivar el botón si la letra ya ha sido usada.

### C. Encriptado de las palabras en la Base de Datos

Para una mayor seguridad hemos decidido no solo encriptar por ejemplo el nombre del Jugador, sino usar esta criptografía simétrica para “cubrir con otra capa” las palabras almacenadas en nuestro JSON, ya que en este caso son solamente palabras, pero en otro podrían ser datos más relevantes en el que sí que sería necesario emplear nuestro algoritmo de cifrado.

### D. Aspectos clave en el juego

- En nuestra manera de jugar hemos cambiado algunas cosas, cuando un jugador acierta una letra de la palabra, se le muestra al otro usuario también en su pantalla, pero en caso de que falle una letra, esta no se muestra como fallada, ayudando así a que el juego sea más igualado para ambos.
- En caso de que un usuario cancele la partida al otro se le desactivará el teclado, y se le mostrará un botón que no aparecía antes para volver al menú de selección de usuario, facilitando así el cierre de programa.
- El sistema irá avisando a los usuarios por la pantalla de si es su turno o no

lo es, además de decir si ha acertado la palabra o se ha quedado sin vidas.

