CALONGE BRIEGA, CLARA

LOPEZ ROCHER, ADOLFO

MURILLO ZARZA, RAFAEL

PROYECTO SISTEMAS WEB 1

PROYECTO AHORCADO

INDICE

[1 DOCUMENTACIÓN 2](#_Toc502682050)

[1.1 RESUMEN DEL PROPOSITO DEL PROYECTO 2](#_Toc502682051)

[1.2 LISTA DE FUNCIONALIDADES DEL PROYECTO 2](#_Toc502682052)

[1.3 DOCUEMTO DE DISEÑO, DECISIONES DEL PROYECTO. 3](#_Toc502682053)

[1.4 PUESTA EN MARCHA DEL PROYECTO 4](#_Toc502682054)

[2 DIAGRAMAS 4](#_Toc502682055)

[2.1 UML 4](#_Toc502682056)

[2.2 E-R 4](#_Toc502682057)

# DOCUMENTACIÓN

## RESUMEN DEL PROPOSITO DEL PROYECTO

El proyecto consiste en llevar a cabo el *“El juego del Ahorcado”* como aplicación web, el ahorcado clásico (también llamado colgado) es un juego de adivinanzas de lápiz y papel para dos o más jugadores. Un jugador piensa en una palabra, frase u oración y el otro trata de adivinarla según lo que sugiere por letras. En nuestro caso lo hemos modificado, la maquina o el juego propone una palabra y el usuario o jugador debe adivinarla, se presenta el alfabeto (Español) en botones y el jugador debe ir pulsando según crea cual es la letra que estará en la palabra, también se da la opción de responder directamente a través de una caja de texto. El jugador tiene 6 vidas o intentos para adivinar la palabra, estas van restando si el jugador dice una letra o palabra incorrecta. La puntuación son las vidas que ha conseguido mantener intactas. También se presentará una media de éxito de palabras adivinadas por el jugador.

Las principales ventajas de usar una aplicación web frente a una aplicación simple o de escritorio son:

* Portabilidad, se pueden ejecutar desde cualquier ordenador con acceso a internet.
* Aplicaciones muy ligeras, por lo que el usuario no necesita un ordenador muy potente.
* Facilidad de actualizar y mantener.
* Su funcionalidad es independiente del SO.
* No hay incompatibilidad entre versiones, todos trabajan en la misma.
* Los usuarios pueden participar en la elaboración de los contenidos.

## LISTA DE FUNCIONALIDADES DEL PROYECTO

Debería ser capaz de:

* 1. Crear un usuario e insertarlo en la base de datos.
  2. Iniciar Sesión, comprobar si existe y dar una respuesta correcta si no es así.
  3. En el Inicio del juego dar diversas opciones (Empezar a jugar, ver tablero…)
  4. Dar la opción de ver *“Tablero de Campeones”* donde se mostrará la puntuación ordenada de los mejores jugadores.
  5. Ver un tutorial simple de cómo se juega y las distintas opciones que hay dentro del juego.
  6. La opción de *“Empezar a jugar”* donde te llevará a otra página donde se mostrarán:
     1. La palabra oculta por este signo: ‘\_’ , aparecerán el mismo número de signos que de letras tenga la palabra.
     2. Tabla con 27 botones, los cuales al pulsar deberán cambiar de color y bloquearse. Si la letra se encuentra en la palabra este botón se pondrá en verde y bloqueado, si es incorrecto se pondrá de color rojo y bloqueado.
     3. *Input text* donde se puede introducir un texto, si el texto coincide con la palabra se ganará inmediatamente independientemente del número de letras que falte por adivinar (Si se ha ganado el botón de enviar otra palabra se bloquea), si la palabra no es correcta se resta una vida.
     4. Una imagen que indique el número de vidas o intentos que le quedan al jugador.
  7. Independientemente se gane o pierda una partida aparecerá un botón que dará la opción de “siguiente palabra”, la palabra ya ha cambiado independientemente del jugador lo pulse o no.
  8. Al terminar la partida se almacenará y actualizará la nueva puntuación, al igual de un registro de si se ha perdido o ganado.
  9. En todo momento aparecerá un menú en la parte de arriba que facilitará en desplazamiento, exceptuando en las pagina de Tutorial, Iniciar Sesión y Crear Usuario.

## DOCUEMTO DE DISEÑO, DECISIONES DEL PROYECTO.

Estos fueron los pasos que fuimos siguiendo a lo largo del proyecto:

* Primer objetivo contar número de intentos con un único botón y saber si la letra está en la palabra o no.
* Insertar un *input text* con un botón de enviar y comprobar que la letra está en la palabra.
* Ir mostrando la palabra según la letra sea o no correcta.
* Condición de ganar o perder.
* Insertar una tabla de botones del alfabeto español, solo envían la letra, ni se bloquean ni cambian de color.
* Página de inicio de sesión empleando un fichero de texto como registro, se debe insertar a mano con 3 columnas Nombre, Contraseña y Puntuación.
* Mostrar una puntuación simple.

En este punto realizamos la primera presentación del proyecto.

Cambios propuestos o sugeridos, usar una base de datos, añadir página de crear usuario, tutorial, forma de contabilizar algún tipo de puntuación (en mente usar número de letras).

* Problema y solución del envió de la letra ‘Ñ’.
* Creación de la base de datos, con dos tablas:
  + Usuarios
  + Palabras

Evolución de las tablas:

1. Usuarios:
   1. idUsuarios, Nombre, Password, Puntuación, idPalabra. (En el proyecto se usará como primary key el Nombre)
   2. idUsuarios, Nombre, Password, Puntuación, PartidasG, PartidasP, Media, idPalabra.
   3. idUsuarios, Nombre, Password, Puntuación, PartidasG, PartidasP, Media, idPalabra, ListaAciertos, ListaFallos.
2. Palabras:
   1. idPalabra, palabra, tema.

* Añadir las siguientes opciones de Inicio, *“Añadir jugador”* (Opción en mente para dos jugadores), *“Tutorial”* breve información de cómo se juega, *“Tablero de Campeones”* muestra la lista de jugadores sin ordenar.
* Añadimos CSS.
* Añadimos imágenes en la aplicación web, tanto decorativas como vidas.
* Añadir mensajes de error si el usuario no es correcto o se ha creado mal.
* Cambio radical del código, la funcionalidad dividirla de la vista.

Se realiza la segunda presentación del proyecto, cambios que se piden o se sugieren, quitar código de los jsp, guardar partida, dos jugadores, centrarse en los mínimos.

* Quitar código de los jsp, no es posible quitarlo en Tutorial.jsp.
* Revisión de mínimos, añadir Cookie, Listener…
* Lograr guardar partida del usuario.
* Cambio en 2 jugadores, la decisión es que el usuario colabore añadiendo palabras.

## PUESTA EN MARCHA DEL PROYECTO

# DIAGRAMAS

## UML

## E-R