

El Juego del Ahorcado

1. [Introducción](#)
2. [Objetivos](#)
3. [Alcance del proyecto](#)
 - 3.1. [Funcionalidades específicas](#)
 - 3.2. [Limitaciones y exclusiones](#)
4. [Diseño y Arquitectura](#)
 - 4.1. [Diseño de la interfaz de usuario](#)
 - 4.2. [Estructura de la información](#)
 - 4.3. [Arquitectura técnica](#)
5. [Tecnologías Utilizadas](#)
6. [Herramientas de desarrollo](#)
7. [Plataformas de alojamiento](#)
8. [Desarrollo del proyecto](#)
 - 8.1. [Fases](#)
 - 8.1.1. [Fase 1: Juego básico](#)
 - 8.1.1.1. [Diseño de la interfaz](#)
 - 8.1.1.2. [Integración de estilos con CSS.](#)
 - 8.1.1.3. [Implementación de la lógica del juego en JavaScript](#)
 - 8.1.2. [Fase 2: Limitación por tiempo + Registro de estadísticas](#)
 - 8.1.2.1. [Cuenta atrás](#)
 - 8.1.2.2. [Estadísticas](#)
 - 8.1.3. [Fase 3: Juego temático](#)
 - 8.2. [Pruebas, correcciones y control de calidad](#)
 - 8.2.1. [Plan de pruebas](#)
 - 8.2.2. [Corrección de errores](#)
 - 8.2.3. [Procedimientos de control de calidad](#)
 - 8.3. [Finalización y documentación.](#)
 - 8.3.1. [Fecha de entrega](#)
 - 8.3.2. [Criterios de aceptación](#)
 - 8.3.3. [Documentación](#)
 - 8.3.3.1. [Diario de desarrollo](#)
 - 8.3.3.2. [Documentación técnica](#)
 - 8.3.3.3. [Instrucciones del juego](#)
9. [Evaluación del proyecto](#)
10. [Actualizaciones y sugerencias](#)
 - 10.1. [Selección del tiempo para la cuenta atrás](#)
 - 10.2. [Gestión de usuarios](#)
 - 10.3. [Records](#)

1. Introducción

El proyecto consiste en desarrollar el juego clásico "El Ahorcado" y está dirigido a alumnos del segundo curso del CFGS "Desarrollo de Aplicaciones Web".

Se busca aplicar y reforzar los conocimientos adquiridos en los primeros meses del módulo "Desarrollo Web en Entorno Cliente", centrándose en JavaScript ES6.

La creación del juego responde a la necesidad de ofrecer a los alumnos un proyecto desafiante que integre los conceptos enseñados hasta el momento. Proporciona un entorno práctico para la aplicación de las tecnologías web aprendidas.

Los alumnos consolidarán sus conocimientos teóricos mediante la aplicación práctica, fortaleciendo sus habilidades de desarrollo web. El juego resultante será una demostración tangible de su progreso y servirá como base para proyectos futuros.

2. Objetivos

- Practicar conceptos clave como:
 - Arrays
 - Objetos
 - Clases
 - LocalStorage
 - JSON
 - Manejo del tiempo (Date)
 - DOM
 - Eventos
- Desarrollar habilidades avanzadas en JavaScript ES6.
- Crear una interfaz atractiva y funcional utilizando HTML5 y CSS.
- Fomentar la lógica de programación y la resolución de problemas.
- Mejorar la comprensión de la interacción entre el código JavaScript y la interfaz de usuario.
- Completar el juego de "El Ahorcado" en los 8 días asignados.
- Proporcionar a los alumnos una base sólida para proyectos web más complejos.

3. Alcance del proyecto

3.1. Funcionalidades específicas

El juego incluirá las funcionalidades esenciales del juego de "El Ahorcado", con énfasis en la manipulación del DOM a través de JavaScript, de manera que:

- La interfaz de usuario resulte interactiva y atractiva.
- Se manipule el DOM para gestionar la lógica del juego.
- Se utilice el almacenamiento local para guardar el progreso del juego y otras características del mismo.

3.2. Limitaciones y exclusiones

Se limitará a los conceptos enseñados hasta la fecha de inicio del proyecto y a los que se expliquen durante el desarrollo del proyecto, excluyendo características avanzadas. Las inclusiones opcionales podrían ser la integración de fuentes importadas de Google Fonts para mejorar la presentación, incorporación de imágenes y videos.

- No se incluirán características avanzadas que no hayan sido enseñadas en el curso hasta la fecha de inicio del proyecto.
- No se abordarán temas fuera del alcance del módulo actual.
- Las inclusiones opcionales podrían ser la integración de fuentes importadas de Google Fonts para mejorar la presentación, incorporación de imágenes y videos.

4. Diseño y Arquitectura

4.1. Diseño de la interfaz de usuario

Se utilizará HTML5 y CSS para crear una interfaz intuitiva y atractiva, priorizando la experiencia del usuario.

4.2. Estructura de la información

El juego incluirá elementos visuales y de texto para guiar al jugador, junto con un diseño claro y legible.

4.3. Arquitectura técnica

JavaScript ES6 llevará el peso del proyecto, manipulando el DOM para gestionar la lógica del juego. Se enfocará en una arquitectura modular y fácilmente mantenible.

5. Tecnologías Utilizadas

JavaScript ES6, HTML5, CSS: Estas tecnologías serán la base del desarrollo, con un énfasis especial en las capacidades de JavaScript para interactuar con el DOM.

6. Herramientas de desarrollo

Se recomiendan herramientas de desarrollo estándar, como editores de código ([Visual Studio Code](#)) y navegadores web ([Chrome](#), [FireFox](#)).

7. Plataformas de alojamiento

El proyecto estará visible en [GitHub](#) y publicado en [Netlify](#).

8. Desarrollo del proyecto

8.1. Fases

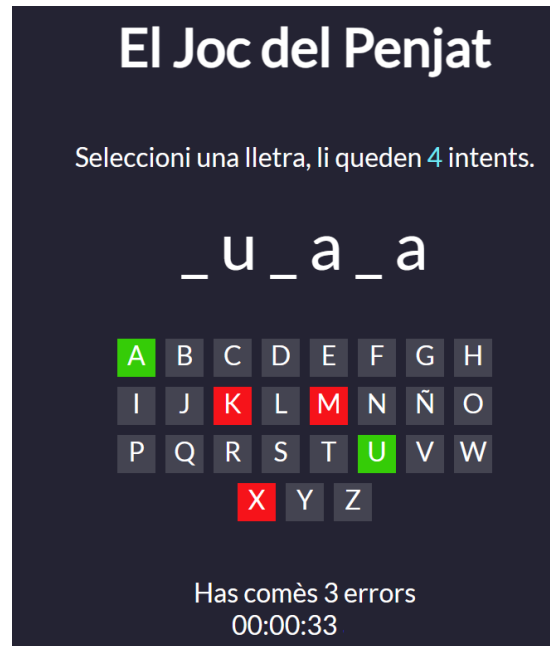
El desarrollo del juego consta de tres fases, siendo funcional desde la primera.

8.1.1. Fase 1: Juego básico

En esta fase el juego ha de resultar plenamente funcional.

- El programa planteará una palabra al usuario, mostrando tantos guiones como letras la formen.
- Se mostrará un indicador con los intentos restantes para encontrar la palabra que se irá actualizando según los errores cometidos.
- Se mostrará el abecedario. Cada letra estará dentro de un recuadro de color diferente al color de fondo de la página.
- El fondo de las letras del abecedario cambiará de color cuando se haga clic sobre ella:
 - Color verde si la letra pertenece a la palabra secreta.
 - Color rojo si la letra no está en la palabra secreta.
- Se mostrará un indicador de los errores cometidos.
- Se mostrará un cronómetro que se pondrá en marcha después del primer clic sobre una letra y se detendrá al finalizar la partida.

Ejemplo:



8.1.1.1. Diseño de la interfaz

Crear una estructura de HTML, con los elementos (con o sin identificadores) y las clases necesarias para poder interactuar con ellos desde JavaScript, que contenga las siguientes zonas/elementos:

- Título del juego
- Intentos restantes
- Palabra oculta
- Letras del abecedario
- Errores cometidos
- Cronómetro

8.1.1.2. Integración de estilos con CSS

Generar una hoja de estilos de manera que:

- El juego tenga un aspecto atractivo e intuitivo.
- JavaScript pueda utilizar estilos definidos en la hoja para darle funcionalidad al juego.
- Opcionalmente, importe alguna fuente de [Google Fonts](#).

8.1.1.3. Implementación de la lógica del juego en JavaScript

Desarrollar un código limpio, eficiente, funcional y comentado. Ha de seguir todo lo aprendido hasta ahora en el curso y manejar los conceptos descritos en el [punto 2 \(Objetivos\)](#).

8.1.2. Fase 2: Limitación por tiempo + Registro de estadísticas

8.1.2.1. Cuenta atrás

En esta fase se añadirá una cuenta atrás, justo debajo del cronómetro, que indique un tiempo límite para seleccionar una letra. Pasado ese tiempo se rebajará en una unidad el número de intentos restantes. Si esto ocurre o el usuario selecciona una letra, la cuenta atrás se reiniciará.

8.1.2.2. Estadísticas

Al finalizar una partida se guardará, si se ha acertado la palabra, utilizando el LocalStorage, un registro que indique la palabra con la que se estaba jugando, el número de errores cometidos y el tiempo total empleado.

Al finalizar una partida se deberá comprobar si ya existe un registro para la palabra jugada. Si es así, se comparará tanto el tiempo empleado como el número de errores y se actualizará si se han mejorado uno, otro o ambos.

8.1.3. Fase 3: Juego temático

Deberán generarse conjuntos temáticos de palabras. Estos conjuntos deberán almacenarse en un fichero JSON e incorporarse al juego al cargar la página. El usuario dispondrá de un elemento "Select" para poder escoger sobre qué tema quiere que sean las palabras a descubrir.

Para ello, se implementará un servidor JSON que lea el fichero.

8.2. Pruebas, correcciones y control de calidad

8.2.1. Plan de pruebas

Para cada fase deberán realizarse:

- Pruebas de funcionalidades clave del juego.
- Verificación de la interacción correcta entre JavaScript, HTML y CSS.
- Pruebas de compatibilidad con diferentes navegadores.

8.2.2. Corrección de errores

El programa no puede ejecutarse con errores así que durante su desarrollo y al dar por finalizada cada fase deberán depurarse todos aquellos errores que se detecten, ya sean funcionales o de ejecución.

Se recomienda el uso de las herramientas de desarrollador del navegador.

8.2.3. Procedimientos de control de calidad

- Revisión de código entre pares por parte del alumno
- Revisiones periódicas del código in situ o via GitHub por parte del profesor.

8.3. Finalización y documentación

Para considerar la finalización del proyecto, en cada una de sus fases se deben tener en cuenta los siguientes hitos:

- a. Diseño aprobado.
- b. Lógica del juego implementada.
- c. Interfaz y estilos integrados.
- d. Pruebas exitosas.

8.3.1. Fecha de entrega

14 de diciembre de 2023.

8.3.2. Criterios de aceptación

El alumno indicará en qué fase entrega el proyecto y en base a ello se valorarán los siguientes aspectos:

- a. El juego funciona sin errores.
- b. La interfaz es atractiva y fácil de usar.
- c. Cumplimiento de los estándares de codificación.
- d. Cumplimiento de los requisitos expuestos en este documento.

8.3.3. Documentación

Durante el desarrollo del proyecto deberá generarse la siguiente documentación:

8.3.3.1. Diario de desarrollo

Al finalizar cada sesión de clase (o en cualquier momento durante la misma) el alumno dejará por escrito, en un documento compartido en GitHub, un breve resumen de lo desarrollado en dicha sesión.

8.3.3.2. Documentación técnica

Describir, brevemente, los siguientes puntos del proyecto:

- **Descripción general del proyecto:** Incluye una descripción general del proyecto, su propósito y objetivos.
- **Funcionalidad:** Describe la funcionalidad del proyecto, incluyendo las características principales y las funcionalidades específicas.
- **Estructura:** Describe la estructura del proyecto, incluyendo la organización de los archivos HTML, CSS y JavaScript.
- **Requisitos del sistema:** Incluye los requisitos del sistema necesarios para ejecutar el proyecto, como el navegador web y la versión de JavaScript.
- **Instalación y configuración:** Proporciona instrucciones detalladas sobre cómo instalar y configurar el proyecto.
- **Uso del proyecto:** Incluir aquí el punto 8.3.3.3.
- **Actualización:** Incluir aquí el punto X.
- **Referencias:** Incluye una lista de referencias utilizadas en el proyecto, como tutoriales, documentación y recursos en línea.

8.3.3.3. Instrucciones del juego

En este documento se explica al usuario cómo se juega a nuestra aplicación de “El Ahorcado”, cuál es la limitación por tiempo, los intentos que dispone, etc.

9. Evaluación del proyecto

Para la evaluación del proyecto se tendrá en cuenta:

- El [punto 8.3.2.](#)
- Fase y nivel de fase alcanzado.
- Puntualidad en el entrega.
- Seriedad de la propuesta entregada.
- Calidad de la documentación.
- Organización del proyecto en GitHub.
- Aportaciones extra al proyecto que supongan una mejora del juego y/o de la experiencia de usuario.

10. Actualizaciones y sugerencias

En el caso que el alumno acabe el proyecto y disponga de tiempo, este proyecto podría incorporar las mejoras siguientes.

10.1. Selección del tiempo para la cuenta atrás

Incorporar la posibilidad de modificar la dificultad del juego mediante la selección del tiempo de respuesta en la cuenta atrás.

Se ha de tener en cuenta que esto modificaría la manera de guardar registros en el LocalStorage.

10.2. Gestión de usuarios

Añadir una gestión de usuarios, de manera que puedan darse de alta y baja.

Escoger si guardarlos a nivel de LocalStorage o de JSON.

Al iniciar la aplicación debería introducir el nombre de usuario y guardar los registros del juego de manera nominal.

10.3. Records

Crear una pantalla donde ver los records por palabra.