



# TRIVIAL DE JAVASCRIPT



**TOMAS CANO NIETO**  
**DESARROLLO WEB EN ENTORNO CLIENTE**  
**2ºDAW**

# INDICE:

1. Explicación del Trivial de manera Visual.....	(3 a 4)
2. Explicación de la Estructura del Código.....	(5 a 7)
3. Explicación de los ficheros CSS.....	(8 a 10)
4. Explicación de los ficheros HTML.....	(11 a 12)
5. Explicación del fichero JavaScript.....	(13 a 15)
6. Recursos Utilizados en el Proyecto.....	(16 a 17)
7. Resumen del Trivial.....	(18 a 19)
8. Bibliografía/Webgrafía .....	(20)
9. Imágenes del Código (Figuras).....	(21 a 30)

# FIGURAS:

1. FIG. 1: Menú de Inicio.....	(21)
2. FIG. 2: Interfaz de Juego.....	(22)
3. FIG. 3: Selección de Respuesta.....	(23)
4. FIG. 4: Retroalimentación de Respuestas.....	(24)
5. FIG. 5: Pantalla de Resultados Finales.....	(25)
6. FIG. 6: Código HTML (index.html).....	(26)
7. FIG. 7: Código HTML (trivial.html).....	(27)
8. FIG. 8: Código JavaScript (script.js).....	(28)
9. FIG. 9: Código CSS (style_menu.css).....	(29)
10. FIG. 10: Código CSS (style_trivial.css).....	(30)
11. FIG. 11: Código CSS (Style_jugar.css).....	(31)

# ACRÓNIMOS:

1. Listado de Acrónimos.....	(32)
------------------------------	------

# Explicación del Trivial de manera Visual

El Trivial JavaScript es un juego de preguntas y respuestas que evalúa los conocimientos de los usuarios en temas específicos de JavaScript. En el juego, el usuario debe responder una serie de 20 preguntas, cada una con tres opciones posibles, de las cuales solo una es correcta.

## Flujo Visual del Juego:

### 1. Menú de Inicio:

- Al iniciar el juego, el usuario se encuentra en la pantalla de menú, que contiene tres opciones principales:
  - **Jugar:** Comienza el trivial, llevando al usuario a la primera pregunta.
  - **Cómo Jugar:** Dirige a una página con las instrucciones y reglas del juego.
  - **Salir del Juego:** Muestra una confirmación para salir del juego.
- **Interfaz Visual:** El menú presenta un diseño atractivo con botones estilizados que resaltan al pasar el ratón por encima. Los botones se han diseñado para facilitar la navegación.

### 2. Pantalla de Juego:

- **Estructura de Preguntas y Respuestas:** Cada pregunta se muestra en la parte superior, seguida de las tres opciones de respuesta.
- **Selección de Respuestas:** Las opciones están dispuestas como botones de opción que permiten una selección única por pregunta.
- **Corrección y Navegación:**
  - Al hacer clic en "Corregir Respuesta", el juego evalúa la respuesta seleccionada.
  - Un mensaje informa al usuario si la respuesta es correcta o incorrecta. Si es incorrecta, se muestra también la respuesta correcta para retroalimentación.
  - Después de un breve retardo, el juego avanza automáticamente a la siguiente pregunta.
- **Visualización Dinámica:** El diseño visual asegura que las preguntas y opciones se presenten claramente, con espacio suficiente para facilitar la lectura y selección.

### 3. Pantalla de Resultados Finales:

- Al finalizar el juego, el usuario es dirigido a una pantalla de resumen de resultados, donde se muestran:
  - El número total de respuestas correctas e incorrectas.
  - Un botón para regresar al menú principal.
- **Presentación Final:** Los resultados están diseñados de manera destacada para resaltar el rendimiento del usuario.

### 4. Visualización del Juego

En la sección **Imágenes del Código (Figuras)** de la documentación se incluirán capturas de pantalla de cada fase del juego para una comprensión visual completa. Estas capturas mostrarán:

- El **menú de inicio** y sus opciones.
- La **interfaz de preguntas** con las opciones de respuesta, incluyendo un ejemplo de respuesta correcta e incorrecta.
- La **pantalla de resultados finales** con el conteo de respuestas correctas e incorrectas.

Este diseño visual facilita la experiencia de usuario, ya que permite navegar intuitivamente por las diferentes secciones del trivial mientras aprende y evalúa sus conocimientos en JavaScript.

# Explicación de la Estructura del Código

El código del Trivial JavaScript se ha organizado en varios archivos que cumplen roles específicos. Esta organización permite mantener una estructura ordenada, separando la lógica de JavaScript, la estructura HTML y el estilo CSS. Cada archivo cumple una función específica:

## a) Estructura General de Archivos

1. **index.html**: Página principal, contiene el menú de inicio con opciones para navegar al trivial, ver las instrucciones o salir del juego.
2. **trivial.html**: Contiene la interfaz del juego, mostrando cada pregunta y sus opciones de respuesta.
3. **como\_jugar.html**: Proporciona al usuario las instrucciones para jugar, explicando el sistema de puntuación y los objetivos del juego.
4. **script.js**: Archivo JavaScript que controla la lógica del trivial, desde mostrar las preguntas hasta evaluar las respuestas y calcular los resultados finales.
5. **Archivos CSS** (style\_menu.css, style\_trivial.css, style\_jugar.css): Archivos que controlan la apariencia de cada página del proyecto.

## b) Explicación Detallada de la Lógica de JavaScript (script.js)

El archivo script.js contiene las siguientes secciones clave:

### 1. Variables Globales:

- `numero_pregunta`, `preguntas_correctas`, `preguntas_incorrectas`: Controlan el estado del juego, rastrean la pregunta actual y el conteo de respuestas correctas e incorrectas.
- `texto_pregunta`, `texto_numero_pregunta`, `texto_opcion`, `boton_enviar`, `resultado_respuesta`, `salir_juego`: Referencias a los elementos del DOM (Document Object Model) en `trivial.html`, permitiendo la manipulación del contenido de la página desde el script.

### 2. Funciones Principales:

- **`mostrar_pregunta()`**:
  - Carga la pregunta actual desde el array `preguntas_array` y la muestra en la página.
  - Actualiza el número de la pregunta y reinicia las opciones para que estén desmarcadas al cambiar de pregunta.
- **Verificación de Respuestas**  
(`boton_enviar.addEventListener("click")`):
  - Verifica la respuesta seleccionada por el usuario:
    - Si es correcta, incrementa el contador de respuestas correctas y muestra un mensaje de éxito.
    - Si es incorrecta, muestra un mensaje indicando cuál era la respuesta correcta.
  - Implementa un retardo de 5 segundos para mostrar el mensaje antes de cargar la siguiente pregunta.
- **`siguiente_pregunta()`**:
  - Incrementa el índice de la pregunta (`numero_pregunta`) y, si quedan preguntas, llama a `mostrar_pregunta()`; de lo contrario, llama a `terminar_juego()` para finalizar el trivial.
- **`terminar_juego()`**:
  - Muestra una pantalla final con el conteo de respuestas correctas e incorrectas, utilizando `document.write()` para reemplazar el contenido de la página.

- Incluye un botón para que el usuario vuelva al menú principal.

### 3. Control de Eventos:

- Se utilizan eventos de clic en los botones para manejar la selección de respuestas y la navegación de preguntas, asegurando una experiencia interactiva para el usuario.

## c) Descripción de los Archivos HTML

Cada archivo HTML contiene el contenido estructural para las diferentes secciones del juego, en combinación con el archivo script.js:

- **index.html:**
  - Incluye un menú inicial con botones para acceder al trivial, ver las instrucciones y salir.
- **trivial.html:**
  - Estructura de la interfaz del juego, con espacios para mostrar cada pregunta y opciones de respuesta.
- **como\_jugar.html:**
  - Proporciona instrucciones detalladas sobre el sistema de juego y puntaje.

## d) Interacción entre HTML y JavaScript

El archivo script.js interactúa con los archivos HTML utilizando selecciones de DOM para modificar dinámicamente el contenido y los elementos de cada página, basándose en la selección de respuestas y el avance del usuario.

Con esta organización estructurada, el código es fácil de seguir y permite realizar mejoras o ajustes a cada componente (lógica, estructura o diseño) sin afectar el resto del proyecto.

# Explicación de los ficheros CSS

El proyecto cuenta con varios archivos CSS para estilizar cada sección del trivial. Esta estructura de estilos garantiza que cada parte del juego mantenga un diseño consistente, atractivo y fácil de navegar. A continuación, se explica la funcionalidad y los elementos principales de cada archivo CSS:

## A) Archivo `style_menu.css`

Este archivo estiliza la página de inicio (`index.html`), que incluye el menú principal con las opciones de **Jugar**, **Cómo Jugar** y **Salir del Juego**.

- **Estilos Generales del Cuerpo (body):**
  - Se aplica una fuente consistente (Roboto), un fondo en degradado que pasa de azul a púrpura, y alineación vertical y horizontal para centrar el contenido del menú en la pantalla.
  - Propiedades como `height: 100vh` y `display: flex` aseguran que el contenido se mantenga centrado independientemente del tamaño de la pantalla.
- **Estilos de #menu:**
  - Este contenedor principal incluye propiedades de fondo, borde y animación. Tiene un fondo semitransparente (`rgba(255, 255, 255, 0.1)`) y un borde redondeado para darle un aspecto moderno.
  - La animación `slideDown` añade un efecto de deslizamiento suave al cargar el menú.
- **Botones del Menú (.menu-btn):**
  - Los botones del menú están diseñados para resaltar al usuario. Tienen un fondo naranja (`#f39c12`) que cambia a un tono más oscuro al pasar el ratón, con un efecto de sombra y un ligero movimiento hacia arriba.
  - Propiedades de estilo como `text-transform: uppercase` y `letter-spacing: 1px` mejoran la legibilidad y el diseño visual.



## B) Archivo `style_trivial.css`

Este archivo CSS aplica estilos a la página principal del trivial (`trivial.html`), que muestra cada pregunta y sus opciones de respuesta.

- **Contenedor de la Pregunta (`#menu`):**
  - Los estilos aplicados al contenedor principal de preguntas son similares a los del menú, pero incluyen un diseño específico para que el usuario enfoque su atención en la pregunta y las opciones de respuesta.
  - Este contenedor también usa `box-shadow` y `border-radius` para mantener la apariencia visual consistente con el menú de inicio.
- **Opciones de Respuesta (`.options` y `label`):**
  - Las opciones de respuesta están organizadas verticalmente y son fáciles de seleccionar. Cada `<label>` muestra un margen superior de 10px y una fuente de tamaño adecuado para mejorar la legibilidad.
  - Cada botón de opción (`input[type="radio"]`) incluye un margen a la derecha para que el texto quede correctamente alineado y sea fácil de leer.
- **Resultados de Respuestas Correctas e Incorrectas:**
  - Se utiliza `#resultado-respuesta` para mostrar retroalimentación al usuario, indicando si la respuesta fue correcta o incorrecta.
  - Además, en la pantalla final de resultados, los contadores de respuestas correctas (`.correct-count`) e incorrectas (`.incorrect-count`) están estilizados en verde y rojo, respectivamente, para resaltar visualmente los resultados finales del usuario.

### C) Archivo `style_jugar.css`

Este archivo da estilo a la página de instrucciones (como `_jugar.html`), donde se explica el sistema de juego y las reglas de puntuación.

- **Estilos Generales (#menu):**
  - Similar al menú principal, esta sección utiliza un contenedor centrado y con borde redondeado para que el diseño sea consistente.
  - El contenedor tiene un fondo semitransparente y utiliza la animación `slideIn` para que la experiencia de usuario sea fluida y profesional.
- **Puntuación y Objetivos:**
  - Los estilos de `#menu` incluyen una fuente estándar y colores que mejoran la legibilidad. Los textos de puntuación (`.points`) están resaltados para que el usuario entienda fácilmente el sistema de puntos.
- **Botón de Regreso al Menú:**
  - El botón "Volver al menú" tiene el mismo estilo que los botones en la página de inicio, asegurando una experiencia de usuario coherente.

### Resumen de los Estilos CSS

Cada archivo CSS asegura que cada página del proyecto mantenga un estilo consistente y visualmente atractivo. Al aplicar estilos específicos a cada sección, los usuarios pueden navegar fácilmente entre las distintas páginas, mejorando la experiencia de usuario. Además, la coherencia en colores y tipos de letra asegura que el trivial tenga una apariencia profesional y moderna.

# Explicación de los ficheros HTML

El proyecto del Trivial JavaScript está compuesto por varios archivos HTML que sirven como la base estructural de la aplicación. Cada archivo HTML está diseñado para cumplir funciones específicas y contribuir a la experiencia del usuario. A continuación, se detallan los archivos HTML presentes en el proyecto:

## A) [index.html](#)

Este es el archivo principal del proyecto y actúa como el punto de entrada para el usuario. Contiene el menú de inicio que ofrece las siguientes opciones:

- **Jugar:** Al hacer clic en esta opción, el usuario es redirigido a la página del juego ([trivial.html](#)), donde comenzará a responder las preguntas.
- **Cómo Jugar:** Dirige al usuario a la página de instrucciones ([como\\_jugar.html](#)), que explica las reglas y el sistema de puntuación del juego.
- **Salir del Juego:** Esta opción permite al usuario cerrar la aplicación, mostrando una confirmación para evitar salidas accidentales.

La estructura del menú incluye un diseño atractivo que destaca las opciones disponibles y facilita la navegación.

## B) [trivial.html](#)

Este archivo contiene la interfaz del juego, donde se muestran las preguntas y las opciones de respuesta. La estructura de esta página incluye:

- **Sección de Pregunta:** Presenta la pregunta actual en la parte superior de la pantalla.
- **Opciones de Respuesta:** Las respuestas se muestran como botones de opción, permitiendo al usuario seleccionar una respuesta única.
- **Botón de Corregir Respuesta:** Al hacer clic en este botón, el juego evalúa la respuesta seleccionada y proporciona retroalimentación instantánea.

Esta página está diseñada para ser intuitiva y mantener la atención del usuario en las preguntas.

### C) como\_jugar.html

Este archivo proporciona instrucciones detalladas sobre cómo jugar al Trivial JavaScript. Incluye:

- **Sistema de Puntuación:** Explicaciones sobre cómo se otorgan los puntos y qué objetivos tiene el juego.
- **Reglas del Juego:** Detalles sobre la dinámica de juego, tiempo para responder y cómo finalizar el juego.

La estructura de esta página se enfoca en ser informativa y clara, asegurando que los usuarios comprendan completamente cómo participar en el trivial.

### D) Interacción entre HTML y JavaScript

Los archivos HTML están estrechamente relacionados con el archivo script.js, que maneja la lógica del juego. A través de selecciones de DOM, el JavaScript interactúa con los elementos HTML para actualizar dinámicamente el contenido, permitiendo una experiencia de juego fluida y responsive. Por ejemplo:

- La función `mostrar_pregunta()` actualiza el contenido de la página con la pregunta y las opciones actuales.
- Los eventos de clic en los botones permiten al usuario seleccionar respuestas y avanzar a la siguiente pregunta.

Esta interactividad es clave para mantener el juego dinámico y entretenido.

Con esta organización y funcionalidad, los archivos HTML proporcionan una estructura sólida y accesible para la experiencia del usuario en el Trivial JavaScript.

# Explicación del fichero JavaScript

El archivo script.js es fundamental en el proyecto del Trivial JavaScript, ya que contiene toda la lógica necesaria para el funcionamiento del juego. Su organización y estructura permiten gestionar de manera eficiente el flujo del juego, la interacción del usuario y la evaluación de respuestas. A continuación, se detallan las secciones clave del archivo y su funcionalidad:

## A) Variables Globales

El archivo comienza con la declaración de variables globales que controlan el estado del juego y permiten la interacción con los elementos del DOM:

- **Control de Preguntas:**
  - `numero_pregunta`: Indica la pregunta actual que se está mostrando.
  - `preguntas_correctas`: Contador de las respuestas correctas proporcionadas por el usuario.
  - `preguntas_incorrectas`: Contador de las respuestas incorrectas.
- **Elementos del DOM:**
  - `texto_pregunta`, `texto_numero_pregunta`, `texto_opcion`, `boton_enviar`, `resultado_respuesta`, `salir_juego`: Referencias a los elementos HTML que se utilizan para mostrar las preguntas, opciones, y resultados. Esto permite manipular el contenido visualmente desde el JavaScript.

## B) Funciones Principales

El archivo incluye varias funciones esenciales que controlan el flujo del juego:

### 1. **mostrar\_pregunta():**

- Esta función carga y muestra la pregunta actual desde el array `preguntas_array`.
- Actualiza el texto de la pregunta y reinicia las opciones de respuesta, asegurando que ninguna opción esté seleccionada al cambiar de pregunta.
- También imprime en consola información útil sobre la pregunta actual, facilitando el proceso de depuración.

### 2. **Comprobación de Respuestas:**

- **Evento de clic en boton\_enviar:** Al hacer clic en este botón, se verifica si el usuario ha seleccionado una opción:
  - Si no hay opción seleccionada, se muestra una alerta solicitando al usuario que elija una respuesta.
  - Si hay una opción seleccionada, se compara la respuesta del usuario con la respuesta correcta. Dependiendo del resultado, se actualiza el conteo de respuestas correctas o incorrectas y se muestra un mensaje de retroalimentación al usuario.
  - La función implementa un retardo de 5 segundos antes de avanzar a la siguiente pregunta, lo que permite al usuario ver el resultado antes de continuar.

### 3. **siguiente\_pregunta():**

- Incrementa el índice de `numero_pregunta` para cargar la siguiente pregunta.
- Si quedan preguntas por mostrar, llama a `mostrar_pregunta()`. Si no quedan preguntas, llama a `terminar_juego()`.

### 4. **terminar\_juego():**

- Esta función se encarga de finalizar el juego, mostrando una pantalla de resultados.
- Utiliza `document.write()` para reemplazar el contenido de la página con un resumen de las respuestas correctas e incorrectas, junto con un botón que permite regresar al menú principal.

### **C) Control de Eventos**

El archivo script.js utiliza eventos para hacer que el juego sea interactivo. Principalmente, se gestiona el evento de clic en el botón de enviar respuesta, que activa la lógica de verificación y transición entre preguntas. Este enfoque asegura que la experiencia del usuario sea fluida y atractiva.

### **D) Interacción con el DOM**

A través del uso de métodos de selección y manipulación del DOM, el JavaScript interactúa con los elementos HTML para actualizar dinámicamente la interfaz de usuario. Esto incluye mostrar preguntas, opciones de respuesta, y resultados, proporcionando una experiencia de juego intuitiva y responsiva.

# Recursos Utilizados en el Proyecto

El desarrollo del Trivial JavaScript ha involucrado una variedad de recursos que abarcan desde bibliotecas y herramientas de programación hasta referencias educativas. Estos recursos han sido fundamentales para garantizar la calidad y la funcionalidad del juego. A continuación, se detallan los recursos utilizados en el proyecto:

## A) Herramientas de Desarrollo

### 1. Editores de Código:

- **Visual Studio Code:** Se utilizó como editor principal para escribir y organizar el código JavaScript, HTML y CSS. Sus características, como resaltado de sintaxis, autocompletado y terminal integrado, facilitaron el desarrollo.
- **Sublime Text:** También se empleó como un editor alternativo para tareas rápidas y edición ligera.

### 2. Navegadores Web:

- **Google Chrome:** Utilizado para probar el juego, verificar la compatibilidad y depurar el código mediante las herramientas de desarrollo que ofrece.
- **Mozilla Firefox:** Se utilizó para comprobar el comportamiento del juego en diferentes navegadores y asegurar la compatibilidad cross-browser.

## B) Recursos Educativos

### 1. Documentación de JavaScript:

- **MDN Web Docs (Mozilla Developer Network):** Proporcionó información valiosa sobre funciones de JavaScript, manejo del DOM, eventos y buenas prácticas en programación.
- **W3Schools:** Ofreció tutoriales y ejemplos que ayudaron a comprender conceptos básicos y avanzados del lenguaje.

### 2. Tutoriales en Línea:

- Varios tutoriales en línea y videos de plataformas como YouTube facilitaron la comprensión de técnicas específicas, como el manejo de eventos y la manipulación del DOM.



## C) Recursos Gráficos y Diseño

### 1. Imágenes y Gráficos:

- Se utilizaron recursos gráficos de plataformas como **Unsplash** y **Pexels** para obtener imágenes de fondo y otros elementos visuales que mejoran la estética del juego.
- **Figma**: Herramienta utilizada para diseñar el layout inicial y la interfaz gráfica del juego, permitiendo realizar prototipos y ajustes visuales.

## D) Bibliotecas y Frameworks

### 1. Librerías de Animaciones:

- **Animate.css**: Se consideró para aplicar animaciones suaves al menú y a las transiciones entre preguntas, aunque se implementaron animaciones personalizadas a través de CSS.

## E) Otros Recursos

### 1. GitHub:

- Se utilizó para el control de versiones del proyecto, permitiendo gestionar cambios en el código y colaborar con otros desarrolladores si es necesario.

### 2. Stack Overflow:

- Este foro fue un recurso invaluable para resolver dudas específicas y encontrar soluciones a problemas técnicos durante el desarrollo.

# Resumen del Trivial

El Trivial JavaScript es un juego interactivo diseñado para evaluar los conocimientos de los usuarios sobre JavaScript a través de un formato de preguntas y respuestas. Este proyecto combina la lógica del juego, una interfaz amigable y un diseño atractivo, proporcionando una experiencia educativa y entretenida. A continuación, se presentan los elementos clave del juego:

## A) Objetivo del Juego

El objetivo principal del Trivial es responder correctamente a una serie de 20 preguntas relacionadas con JavaScript. Los jugadores deben seleccionar la respuesta correcta entre tres opciones proporcionadas para cada pregunta. El juego evalúa no solo el conocimiento del usuario, sino también su capacidad para tomar decisiones rápidas.

## B) Estructura del Juego

### 1. Menú Principal:

- El juego inicia en un menú que ofrece opciones para jugar, ver las instrucciones y salir. Este menú está diseñado para ser intuitivo, permitiendo a los jugadores navegar fácilmente por las diferentes secciones del juego.

### 2. Preguntas y Respuestas:

- Cada pregunta se presenta con tres opciones. El usuario puede seleccionar solo una respuesta por pregunta, lo que añade un componente de desafío. Las preguntas cubren diversos aspectos de JavaScript, incluyendo conceptos básicos, funciones avanzadas y mejores prácticas.

### 3. Corrección de Respuestas:

- Después de que el jugador selecciona una respuesta, el juego evalúa si es correcta o incorrecta. Se proporciona retroalimentación instantánea, y si la respuesta es incorrecta, se muestra la respuesta correcta. Esto ayuda a los jugadores a aprender mientras juegan.

### 4. Resultados Finales:

- Al finalizar las preguntas, se muestra un resumen del rendimiento del jugador, destacando el número de respuestas correctas e incorrectas. Esta pantalla final

proporciona una visión clara de los logros del usuario y un botón para regresar al menú principal.

### C) Interactividad y Diseño

- El juego está diseñado para ser interactivo y atractivo, utilizando animaciones suaves y estilos visuales consistentes que mejoran la experiencia del usuario. La combinación de un diseño limpio y una estructura lógica facilita la navegación y mantiene el interés del jugador.

### D) Tecnologías Utilizadas

El Trivial está desarrollado utilizando HTML, CSS y JavaScript. Esta combinación permite un control total sobre la presentación y la lógica del juego, asegurando que sea accesible desde la mayoría de los navegadores web. Los archivos CSS se encargan de la estética, mientras que JavaScript maneja la lógica del juego y la interactividad

### Conclusión

El Trivial JavaScript es una herramienta efectiva y entretenida para aprender y evaluar conocimientos sobre JavaScript de una manera dinámica. A través de preguntas interactivas, los usuarios pueden explorar conceptos clave del lenguaje, lo que hace que el aprendizaje sea más atractivo y accesible.

La estructura del juego, que combina un diseño visual atractivo con una interfaz intuitiva, permite a los estudiantes navegar por el contenido de manera fluida. La retroalimentación inmediata sobre las respuestas seleccionadas ayuda a identificar áreas de mejora, consolidando el aprendizaje en tiempo real.

Además, el Trivial JavaScript motiva a los usuarios a seguir aprendiendo al ofrecer recompensas por respuestas correctas y un seguimiento visible del progreso. Este enfoque no solo hace que la experiencia de aprendizaje sea más amena, sino que también ayuda a construir confianza en las habilidades adquiridas.

En resumen, el Trivial JavaScript va más allá de ser un simple juego; es una plataforma educativa que une diversión y aprendizaje de forma efectiva. Su diseño interactivo fomenta un ambiente en el que los usuarios pueden desarrollar sus habilidades en programación, despertando su interés por continuar explorando el mundo de JavaScript.

# Bibliografía/Webgrafía

A continuación, se detallan las fuentes y recursos utilizados en la elaboración del Trivial JavaScript, que han sido fundamentales para el desarrollo del proyecto y la implementación de sus características:

## 1. W3Schools:

- [W3Schools](#) es una plataforma de aprendizaje en línea que ofrece tutoriales y documentación sobre tecnologías web, incluyendo HTML, CSS y JavaScript. Ha sido una fuente valiosa para comprender la sintaxis y las mejores prácticas en el desarrollo web.

## 2. Visual Studio Code:

- [Visual Studio Code](#) es un editor de código fuente que facilita el desarrollo de aplicaciones web. Se ha utilizado para escribir, depurar y organizar el código del proyecto, ofreciendo herramientas de autocompletado y gestión de versiones.

## 3. ChatGPT:

- [ChatGPT](#) ha sido utilizado como asistente para obtener sugerencias sobre la implementación de estilos CSS. Su capacidad para proporcionar ejemplos y aclarar dudas ha sido de gran ayuda durante el proceso de desarrollo.

## 4. MDN Web Docs:

- [MDN Web Docs](#) es una referencia completa para desarrolladores web. Aunque no se ha mencionado explícitamente, es recomendable como fuente de información adicional sobre HTML, CSS y JavaScript, proporcionando guías y ejemplos de uso.

## Agradecimientos

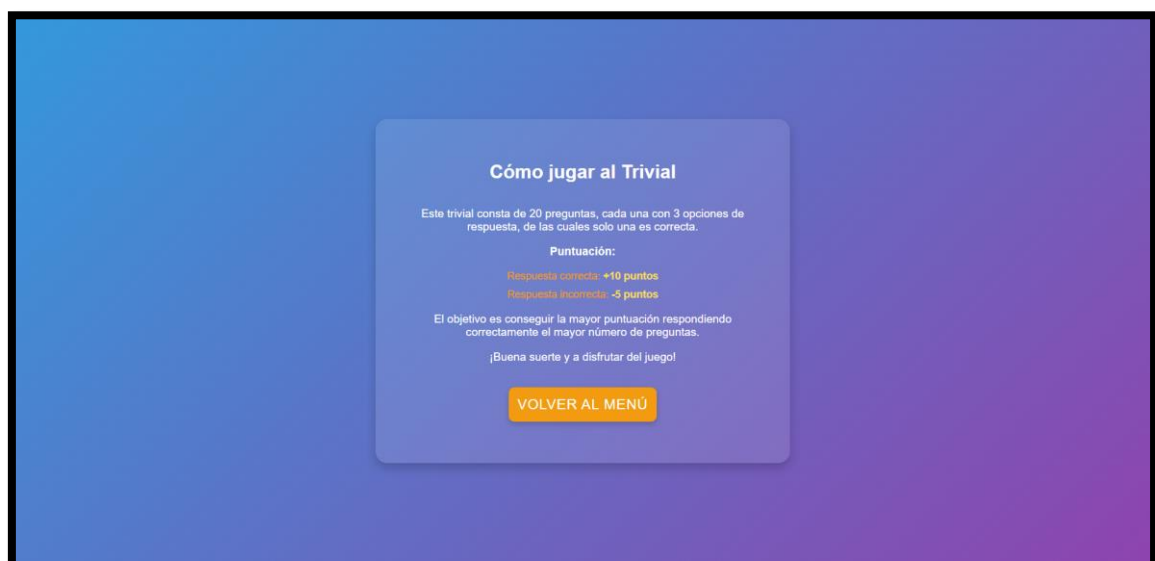
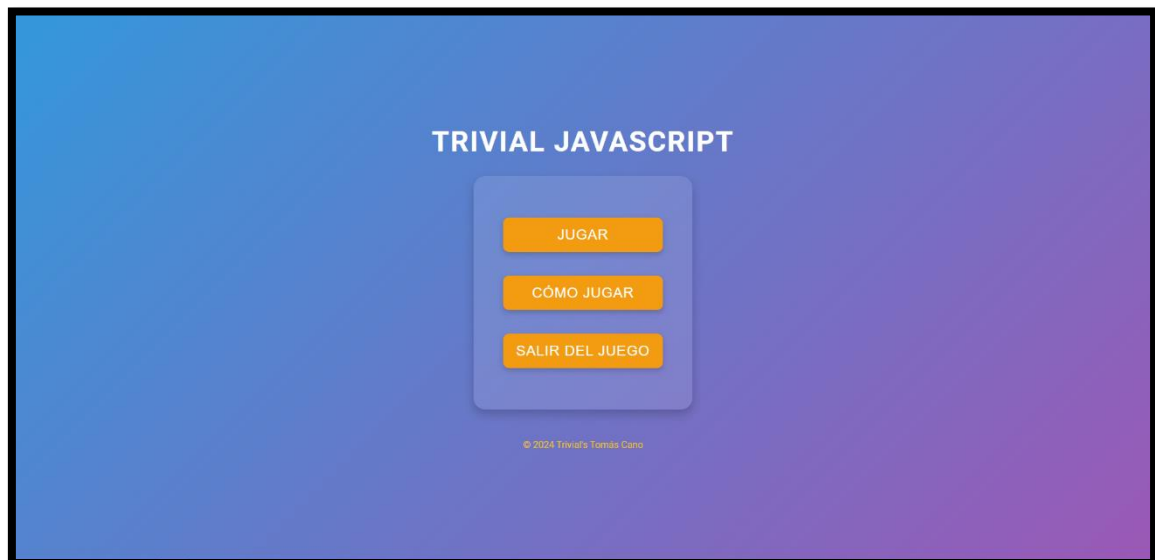
Agradezco a los recursos y herramientas mencionados que han hecho posible la creación de este proyecto, facilitando el aprendizaje y la implementación de conceptos de programación web.

# Imágenes del Código (Figuras)

A continuación, se enumeran las figuras que se deben capturar para documentar visualmente el Trivial JavaScript. Estas imágenes no solo ilustrarán la apariencia del juego, sino que también ayudarán a comprender mejor la estructura y el funcionamiento de cada componente del proyecto. Capturar estas imágenes permitirá a los usuarios y desarrolladores tener una referencia visual clara sobre la interfaz y las interacciones del juego.

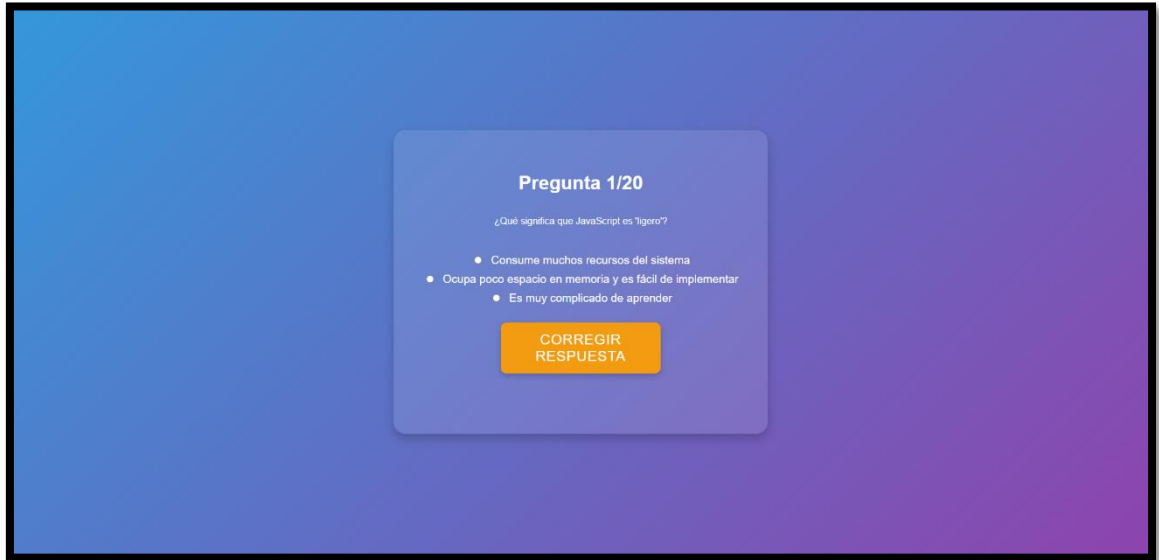
## FIG. 1: Menú de Inicio

- Captura de pantalla del menú principal (index.html) con las opciones de Jugar, Cómo Jugar y Salir del Juego.



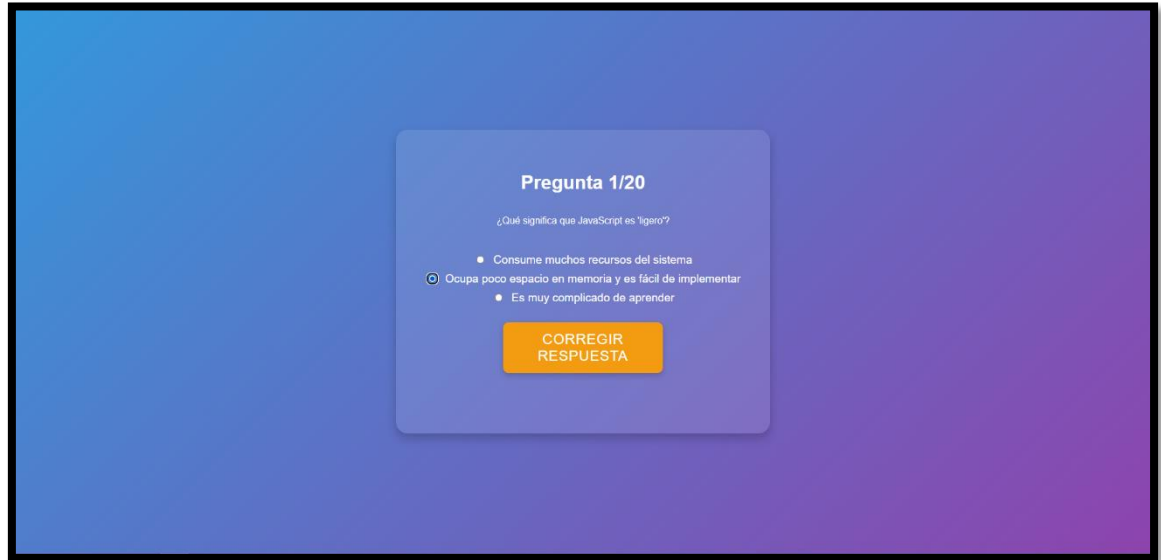
**FIG. 2: Interfaz de Juego**

- Captura de la pantalla de juego (trivial.html) mostrando una pregunta y sus opciones de respuesta.



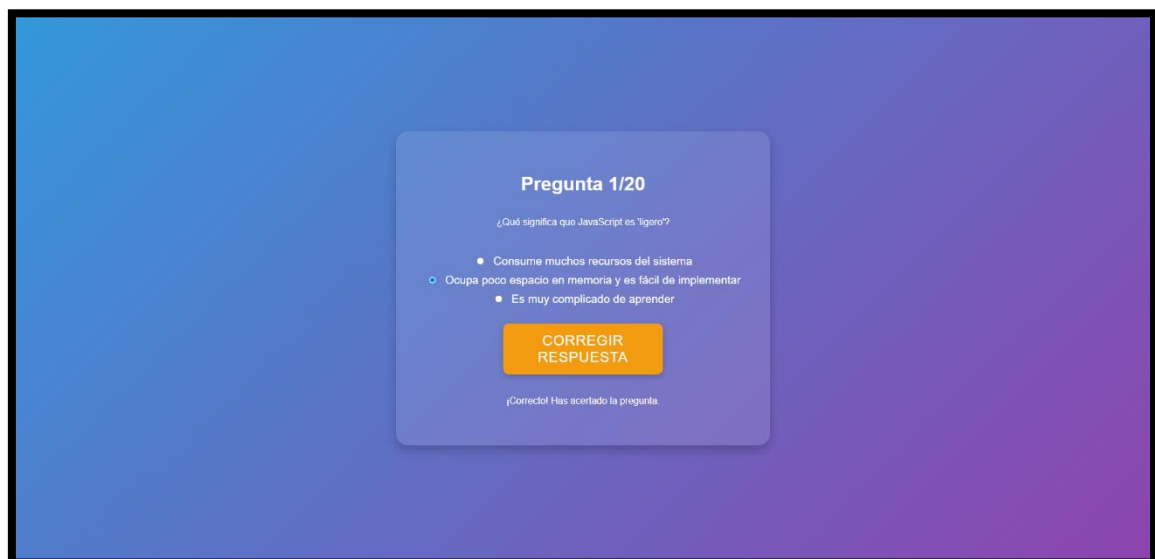
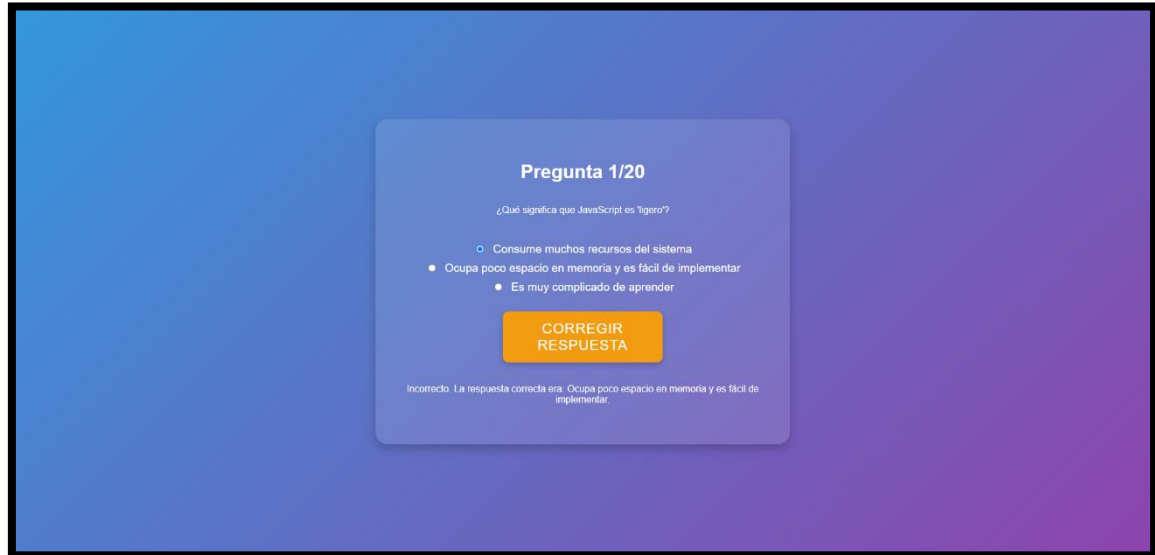
### FIG. 3: Selección de Respuesta

- Imagen que ilustra cómo se selecciona una respuesta en el juego, con el botón de opción marcado.



## FIG. 4: Retroalimentación de Respuestas

- Captura de pantalla que muestra el mensaje de respuesta correcta o incorrecta después de enviar una respuesta.





**FIG. 5: Pantalla de Resultados Finales**

- Imagen de la pantalla que se muestra al finalizar el juego, mostrando el conteo de respuestas correctas e incorrectas.

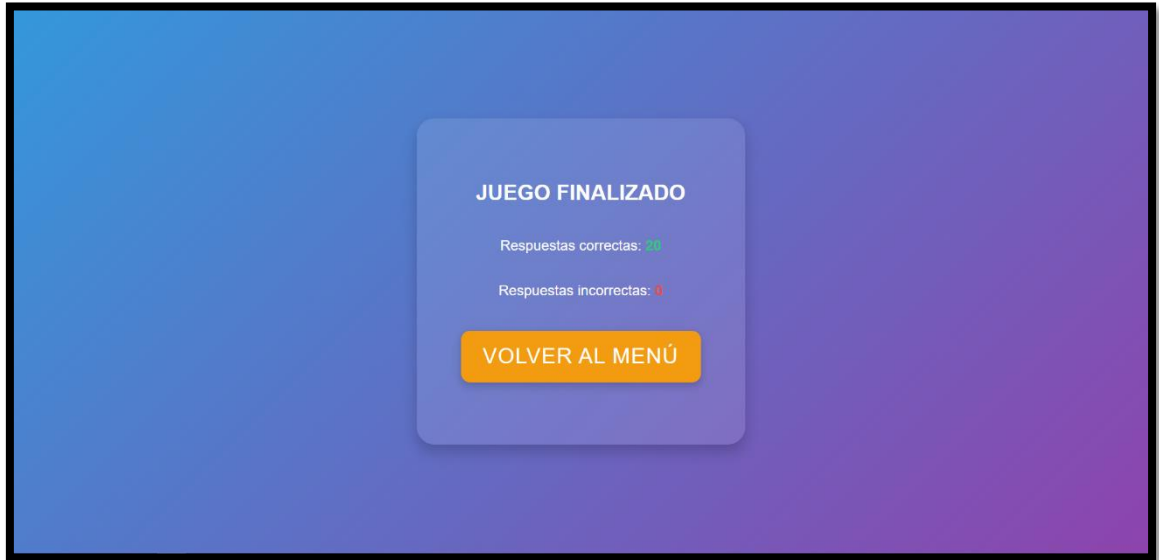


FIG. 6: Código HTML (index.html)

- Captura del código de la estructura HTML del menú de inicio.

```
index.html x
Proyectos > Proyecto_8_noviembre > index.html > ...
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <meta http-equiv="X-UA-Compatible" content="ie=edge">
7      <title>Trivial JavaScript</title>
8      <link href="https://fonts.googleapis.com/css2?family=Roboto:wght@400;700&display=swap" rel="stylesheet">
9      <link rel="stylesheet" href="./estilos/style_menu.css">
10 </head>
11 <body>
12     <header>
13         <h1>Trivial JavaScript</h1>
14     </header>
15
16     <!-- Menú de Inicio -->
17     <section id="menu">
18         <a href="./ficheros/trivial.html">
19             <button class="menu-btn" id="jugar">Jugar</button>
20         </a>
21         <a href="./ficheros/como_jugar.html">
22             <button id="how-to-play-btn" class="menu-btn">Cómo jugar</button>
23         </a>
24         <button class="menu-btn" id="boton-salir">Salir del Juego</button>
25     </section>
26
27     <footer>
28         <p>&copy; 2024 Trivial's Tomás Cano</p>
29     </footer>
30
31     <script>
32         /*Lo que hace es cuando le das al boton de salir del juego,
33         te sale un "confirm" para aceptar o rechazar*/
34         document.getElementById('boton-salir').addEventListener('click', function () {
35             let confirmar_alerta = confirm('¿Estás seguro de que quieres salir del juego?');
36             if (confirmar_alerta) {
37                 window.close();
38             } else {
39
40             }
41         });
42     </script>
43
44 </body>
45 </html>
46
```

FIG. 7: Código HTML (trivial.html)

- Imagen del código de la interfaz del juego.

```
trivial.html X
Proyectos > Proyecto_8_noviembre > ficheros > trivial.html > html > head > meta
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Trivial - Preguntas</title>
7      <link rel="stylesheet" href="../estilos/style_trivial.css">
8  </head>
9  <body>
10 <div id="menu">
11     <h1>Pregunta <span id="numero-pregunta">1</span>/20</h1>
12     <p id="texto-pregunta">Aquí irá la pregunta</p>
13
14     <div class="options">
15         <label>
16             <input type="radio" name="option" value="0">
17             <span id="opcion-0">Opción 1</span>
18         </label>
19         <label>
20             <input type="radio" name="option" value="1">
21             <span id="opcion-1">Opción 2</span>
22         </label>
23         <label>
24             <input type="radio" name="option" value="2">
25             <span id="opcion-2">Opción 3</span>
26         </label>
27     </div>
28
29     <button id="boton-enviar" class="menu-btn">Corregir Respuesta</button>
30
31     <p id="resultado-respuesta"></p>
32
33     <a href="../index.html" id="salir-juego" style="display: none;">
34         <button class="menu-btn">Volver al menú</button>
35     </a>
36 </div>
37
38 <script src="../script/script.js"></script>
39 </body>
40 </html>
```

FIG. 8: Código JavaScript (script.js)

- Captura del archivo script.js que contiene la lógica del juego, resaltando funciones clave como mostrar\_pregunta(), siguiente\_pregunta(), y terminar\_juego().

```
JS script.js x
Proyectos > Proyecto_8_noviembre > script > JS script.js > ...
204 /* Función para mostrar la pregunta actual (Va recorriendo las preguntas de "preguntas_array"
205 y va incrementado tanto el numero de pregunta y la pregunta, a funcion de que va corrigiendo
206 las preguntas del trivial y finalmente "resultado_respuesta" se va borrando cada vez que corrige)*/
207 function mostrar_pregunta() {
208     const current = preguntas_array[numero_pregunta];
209     texto_pregunta.textContent = current.question;
210     texto_numero_pregunta.textContent = numero_pregunta + 1;
211     console.clear();
212     console.log("Pregunta " + numero_pregunta + " : " + texto_pregunta.textContent)
213
214     for (let i = 0; i < 3; i++) {
215         texto_opcion[i].textContent = current.options[i];
216         texto_opcion[i].previousElementSibling.checked = false; // Desmarcar las opciones
217         console.log("Respuesta " + i + " : " + texto_opcion[i].textContent);
218     }
219     resultado_respuesta.textContent = "";
220 }
221
```

```
JS script.js x
Proyectos > Proyecto_8_noviembre > script > JS script.js > terminiar_juego
253 /* Función para cargar la siguiente pregunta*/
254 function siguiente_pregunta() {
255     numero_pregunta++;
256     if (numero_pregunta < preguntas_array.length) {
257         mostrar_pregunta();
258     } else {
259         terminar_juego();
260     }
261 }

```

```
JS script.js x
Proyectos > Proyecto_8_noviembre > script > JS script.js > ...
263 /* Función para finalizar el juego y mostrar los resultados*/
264 function terminar_juego() {
265     document.write(`
266     <html>
267     <head>
268         <link rel="stylesheet" href="../estilos/style_trivial.css">
269     </head>
270     <body>
271         <div id="menu" class="end-game-container">
272             <h2>JUEGO FINALIZADO</h2>
273             <p>Respuestas correctas: <span class="correct-count">${preguntas_correctas}</span></p>
274             <p>Respuestas incorrectas: <span class="incorrect-count">${preguntas_incorrectas}</span></p>
275             <a href="../index.html">
276                 <button class="menu-btn">Volver al menú</button>
277             </a>
278         </div>
279     </body>
280     </html>
281 `);
282 }
283
284 /* Cargar la primera pregunta al iniciar*/
285 mostrar_pregunta();

```

FIG. 9: Código CSS (style\_menu.css)

- Imagen de los estilos aplicados al menú de inicio, destacando propiedades relevantes como fondo, botones, y animaciones.

```
# style_menu.css X
Proyectos > Proyecto_8_noviembre > estilos > # style_menu.css > ...
1  * {
2      box-sizing: border-box;
3      margin: 0;
4      padding: 0;
5  }
6
7  body {
8      font-family: 'Roboto', sans-serif;
9      background: linear-gradient(135deg, #3498db, #9b59b6);
10     display: flex;
11     flex-direction: column;
12     justify-content: center;
13     align-items: center;
14     min-height: 100vh;
15     overflow: hidden;
16 }
17
18 header {
19     text-align: center;
20     margin-bottom: 30px;
21 }
22
23 h1 {
24     font-size: 3rem;
25     color: #fff;
26     text-transform: uppercase;
27     letter-spacing: 2px;
28     animation: fadeIn 2s ease-in-out;
29 }
30
31 p {
32     font-size: 1.2rem;
33     color: #f1c40f;
34 }
35
36 #menu {
37     display: flex;
38     flex-direction: column;
39     align-items: center;
40     justify-content: center;
41     background: rgba(255, 255, 255, 0.1);
42     border-radius: 20px;
43     padding: 50px;
44     box-shadow: 0 8px 16px rgba(0, 0, 0, 0.2);
45     animation: slideIn 1.5s ease-out;
46 }
47
```

FIG. 10: Código CSS (style\_trivial.css)

- Captura de los estilos aplicados a la página de juego, mostrando cómo se estilizan las preguntas y las opciones de respuesta.

```
# style_trivial.css X
Proyectos > Proyecto_8_noviembre > estilos > # style_trivial.css > ...
1  /* Estilo general del cuerpo */
2  body {
3      font-family: Arial, sans-serif;
4      background: linear-gradient(135deg, #3498db, #8e44ad);
5      margin: 0;
6      height: 100vh;
7      display: flex;
8      justify-content: center;
9      align-items: center;
10     color: white;
11 }
12
13 /* Estilo del contenedor #menu */
14 #menu, .end-game-container {
15     display: flex;
16     flex-direction: column;
17     align-items: center;
18     justify-content: center;
19     background: rgba(255, 255, 255, 0.1);
20     border-radius: 20px;
21     padding: 50px;
22     box-shadow: 0 8px 16px rgba(0, 0, 0, 0.2);
23     animation: slideIn 1.5s ease-out;
24     max-width: 600px;
25     text-align: center;
26 }
27
28 /* Animación para el efecto de deslizamiento del menú */
29 @keyframes slideIn {
30     from {
31         opacity: 0;
32         transform: translateY(-50px);
33     }
34     to {
35         opacity: 1;
36         transform: translateY(0);
37     }
38 }
39
40 /* Opciones de respuesta */
41 .options {
42     margin-top: 20px;
43 }
```

**FIG. 11: Código CSS (style\_jugar.css)**

Captura de los estilos aplicados a la página de como jugar:

```
# style_jugar.css X
Proyectos > Proyecto_8_noviembre > estilos > # style_jugar.css > ...
1  /* Estilo general del cuerpo */
2  body {
3      font-family: Arial, sans-serif;
4      background: linear-gradient(135deg, #3498db, #8e44ad); /* Fondo con degradado */
5      margin: 0;
6      height: 100vh;
7      display: flex;
8      justify-content: center;
9      align-items: center;
10     color: white;
11 }
12
13 /* Estilo del contenedor #menu */
14 #menu {
15     display: flex;
16     flex-direction: column;
17     align-items: center;
18     justify-content: center;
19     background: rgba(255, 255, 255, 0.1); /* Fondo semi-transparente */
20     border-radius: 20px;
21     padding: 50px;
22     box-shadow: 0 8px 16px rgba(0, 0, 0, 0.2); /* Sombra para el contenedor */
23     animation: slideIn 1.5s ease-out; /* Animación para que el menú aparezca suavemente */
24     max-width: 600px; /* Limitar el ancho del contenedor */
25     text-align: center; /* Centrar el texto */
26 }
27
28 /* Animación para el efecto de deslizamiento del menú */
29 @keyframes slideIn {
30     from {
31         opacity: 0;
32         transform: translateY(-50px);
33     }
34     to {
35         opacity: 1;
36         transform: translateY(0);
37     }
38 }
39
40 /* Estilo de los párrafos y lista de la sección de contenido */
41 .content p, .content ul {
42     font-size: 1.2rem;
43     margin-bottom: 20px;
44 }
45
```

# Listado de Acrónimos

1. **HTML** (HyperText Markup Language) – Lenguaje de marcado utilizado para estructurar y presentar el contenido en la web. En este proyecto, se utiliza para crear las distintas páginas del trivial.
2. **CSS** (Cascading Style Sheets) – Hojas de estilo en cascada, usadas para estilizar y dar formato visual a los elementos de HTML. Cada sección del proyecto tiene un archivo CSS dedicado.
3. **DOM** (Document Object Model) – Modelo de objetos del documento, una interfaz de programación que permite manipular el contenido y la estructura del documento HTML de forma dinámica desde JavaScript.
3. **JS** (JavaScript) – Lenguaje de programación que permite agregar interactividad a las páginas web. Aquí se usa para controlar la lógica del trivial, como mostrar preguntas y evaluar respuestas.
4. **vh** (Viewport Height) – Unidad de medida en CSS que representa el 1% de la altura del viewport. Utilizada en el diseño del proyecto para asegurar que el contenido se ajuste correctamente a la pantalla del usuario.
5. **px** (Pixels) – Unidad de medida en CSS que representa un píxel en la pantalla del dispositivo. Utilizado para ajustar el tamaño de varios elementos visuales del trivial.
6. **API** (Application Programming Interface) – Interfaz de programación de aplicaciones. Aunque no se menciona directamente en el código, el DOM es un tipo de API que permite interactuar con los elementos de la página.
7. **RGB** (Red, Green, Blue) – Formato de color que utiliza combinaciones de los tres colores básicos para definir colores en CSS, como `rgba(255, 255, 255, 0.1)`.
8. **JSX** (JavaScript XML) – Aunque no se usa en este proyecto, es bueno conocerlo, ya que en otros proyectos JavaScript (React) se emplea JSX para integrar HTML con JavaScript de manera declarativa.
9. **UI/UX** (User Interface / User Experience) – No aparece específicamente, pero es relevante en diseño de interfaz y experiencia de usuario, principios que se aplicaron en el diseño del trivial para hacerlo claro e intuitivo.