**1era opción**

**Párrafo preliminar:**

"Deseo desarrollar una aplicación web y móvil interactiva que ponga a prueba los conocimientos de informática de los usuarios a través de adivinanzas. La interfaz consistirá en una tarjeta que muestre una pregunta de manera aleatoria. Al hacer clic en la tarjeta, esta rotará para revelar la respuesta correspondiente. Un botón adicional permitirá al usuario solicitar una nueva pregunta. Para garantizar una experiencia dinámica, las preguntas y respuestas se almacenarán en un archivo JavaScript y se cargarán aleatoriamente sin repeticiones. Una vez que se hayan agotado todas las preguntas, el ciclo se reiniciará."

Estructura básica HTML, CSS y JavaScript:

**Explicación:**

* **HTML:** Se crea una estructura básica con una tarjeta que se volteará para mostrar la respuesta.
* **CSS:** Se aplica un estilo para crear el efecto de voltear la tarjeta y centrar los elementos.
* **JavaScript:**
  + Se define un array con las preguntas y respuestas.
  + Se obtienen los elementos del DOM para manipularlos.
  + Se crea una función mostrar Pregunta para seleccionar una pregunta aleatoria y mostrarla.
  + Se crea una función siguiente Pregunta para avanzar a la siguiente pregunta.
  + Se agrega un event listener al botón "Siguiente" para llamar a la función siguiente Pregunta.

**Próximos pasos:**

1. **Completar el array:** Agrega más preguntas y respuestas a tu gusto.
2. **Implementar la lógica de aleatorización:** Utiliza un algoritmo de aleatorización para seleccionar preguntas sin repetir.
3. **Manejar el final del array:** Reinicia el índice cuando se llegue al final.
4. **Mejorar la interfaz:** Personaliza los estilos CSS para crear una interfaz más atractiva.
5. **Agregar funcionalidades adicionales:** Puedes implementar un sistema de puntuación, diferentes niveles de dificultad, etc.

**Consideraciones adicionales:**

* **Almacenamiento de datos:** Para aplicaciones más grandes, puedes considerar almacenar las preguntas y respuestas en un archivo JSON o en una base de datos.
* **Adaptabilidad a dispositivos móviles:** Asegúrate de que la aplicación se adapte correctamente a diferentes tamaños de pantalla.
* **Accesibilidad:** Diseña la aplicación teniendo en cuenta los principios de accesibilidad para que pueda ser utilizada por el mayor número de personas posible.

Con esta base, puedes comenzar a construir tu aplicación de adivinanzas y personalizarla según tus necesidades. ¡No dudes en preguntar si tienes alguna duda!

Fuentes y contenido relacionado

**2da opción**

**Párrafo preliminar:**

"Deseo desarrollar una aplicación web y móvil interactiva que ponga a prueba los conocimientos de informática de los usuarios a través de adivinanzas. La interfaz consistirá en una tarjeta que muestre una pregunta de manera aleatoria. Al hacer clic en la tarjeta, esta rotará para revelar la respuesta correspondiente. Un botón adicional permitirá al usuario solicitar una nueva pregunta. Para garantizar una experiencia dinámica, las preguntas y respuestas se almacenarán en un archivo JavaScript y se cargarán aleatoriamente sin repeticiones. Una vez que se hayan agotado todas las preguntas, el ciclo se reiniciará."

Estructura básica HTML, CSS y JavaScript:

**# Prompts para Describir Clasificar los proyectos.**

"Se presenta a continuación un proyecto de desarrollo web compuesto por los siguientes archivos: HTML [CÓDIGO HTML] y JavaScript [CÓDIGO JS]. Esta aplicación tiene como objetivo principal [**una aplicación web y móvil interactiva que ponga a prueba los conocimientos de informática de los usuarios a través de adivinanzas.**]. A partir de este código, realiza una evaluación exhaustiva considerando los siguientes 5 puntos:"

1.- Evaluación del proyecto:

"Califica este proyecto de desarrollo web en una escala de 1 a 5, siendo 1 básico y 5 avanzado, y justifica tu respuesta considerando las tecnologías utilizadas, la complejidad del proyecto y la calidad del código."

2.- Descripción del proyecto:

"Redacta una descripción concisa y persuasiva de este proyecto para un reclutador, destacando los desafíos superados, las tecnologías empleadas y el impacto que tuvo."

3.- Completar el archivo README:

"Completa el archivo README.md siguiendo esta estructura “LA PLANTILLA”, asegurándote de incluir una introducción clara, una descripción detallada de las funcionalidades, instrucciones de instalación y uso, y una sección de agradecimientos."

4.- Revisar el archivo .gitignore:

"Personaliza el archivo .gitignore para este proyecto específico, teniendo en cuenta las características del proyecto y las mejores prácticas de Git. Siguiendo esta plantilla “LA PLANTILLA”

5.- "Resume tus hallazgos clave sobre este proyecto, incluyendo tanto los aspectos positivos como las áreas de mejora. ¿Qué aprendizajes obtuviste durante la evaluación? ¿Qué recomendaciones harías para futuros desarrollos?"

**Evaluación del Proyecto**

**1. Evaluación del proyecto (Calificación: 3/5)**

El proyecto tiene características sólidas para un nivel intermedio, como el uso de **HTML5, CSS3, JavaScript** y la integración de **SweetAlert2** para mejorar la experiencia del usuario. Sin embargo:

* **Puntos fuertes**:
  + Se utiliza un cronómetro funcional que incrementa la interactividad.
  + Uso de una API Fetch para cargar preguntas dinámicamente.
  + Diseño modular en el uso de funciones.
  + SweetAlert mejora la experiencia visual de los mensajes.
* **Áreas de mejora**:
  + El manejo de estilos es básico; podría beneficiarse del uso de librerías como Bootstrap o Tailwind CSS.
  + La lógica del cronómetro podría optimizarse para evitar redundancias en el código.
  + No se utilizan herramientas modernas como frameworks (React/Vue) o control de estado para manejar la lógica de preguntas.
  + No hay validación de errores robusta al cargar preguntas.

Por estas razones, se considera un proyecto **intermedio (3/5)**, con margen de mejora hacia un nivel más avanzado.

**2. Descripción del proyecto**

Este proyecto interactivo desafía a los usuarios a probar sus conocimientos de informática mediante un emocionante juego de preguntas y respuestas. Entre los desafíos superados destacan la implementación de un cronómetro funcional, la carga dinámica de preguntas desde un archivo JSON y el manejo de mensajes estilizados con SweetAlert2. Desarrollado con **HTML5, CSS3 y JavaScript**, este proyecto demuestra habilidades en diseño de interfaces y lógica de programación, ofreciendo una experiencia divertida y educativa.

**3. Archivo README.md**

# \*\*Adivinanzas de Informática\*\*

[Enlace al proyecto](URL\_DEL\_PROYECTO)

![Imagen del proyecto](URL\_DE\_LA\_IMAGEN)

## \*\*Descripción\*\*

Este proyecto es un juego interactivo diseñado para probar los conocimientos de informática de los usuarios mediante preguntas dinámicas y un sistema de puntuación. Inspirado en la gamificación educativa, combina aprendizaje con diversión al permitir a los jugadores medir su progreso en tiempo real y competir contra ellos mismos.

## \*\*Características\*\*

- \*\*Juego interactivo\*\*: Responde preguntas de opción múltiple en una interfaz amigable y accesible.

- \*\*Cronómetro funcional\*\*: Mide el tiempo mientras juegas para añadir emoción y urgencia.

- \*\*Puntuación dinámica\*\*: Calcula automáticamente las respuestas correctas y muestra los resultados.

- \*\*Alertas estilizadas\*\*: SweetAlert2 mejora la experiencia del usuario con mensajes personalizados.

## \*\*Tecnologías Utilizadas\*\*

- \*\*HTML5\*\*: Estructura semántica del proyecto y diseño accesible.

- \*\*CSS3\*\*: Personalización de estilos mediante Flexbox y animaciones básicas.

- \*\*JavaScript\*\*: Implementación de la lógica del juego, cronómetro y Fetch API para la carga dinámica de preguntas.

- \*\*SweetAlert2\*\*: Creación de mensajes visualmente atractivos y dinámicos.

## \*\*Instalación\*\*

1. Clona el repositorio:

```bash

git clone <https://github.com/usuario/nombre_del_proyecto.git>

2. Abre el archivo index.html en tu navegador web favorito para ejecutar el proyecto.

**## Cómo Contribuir**

1. Haz un fork del repositorio.
2. Crea una nueva rama:

bash

Copiar código

git checkout -b nombre-de-la-rama

1. Realiza tus cambios y envía un pull request.

**## Licencia**

Este proyecto está bajo la licencia MIT.

**## Autor**

Este proyecto fue desarrollado por **Jairo Cárdenas**.

* **Correo**: [jairo.cardenas.dev@gmail.com](mailto:jairo.cardenas.dev@gmail.com)
* **GitHub**: [@jairo51067](https://github.com/jairo51067)

---

**4. Archivo .gitignore**

# Archivos de configuración local

config.local

.env.local

settings.json

# Archivos de prueba generados

\*.test.js

# Logs de npm o yarn

npm-debug.log\*

yarn-debug.log\*

# Dependencias de Node.js

node\_modules/

# Archivos de distribución

dist/

build/

# Archivos temporales y logs

\*.tmp

\*.log

\*.swp

# Configuración de editores

.vscode/

.idea/

**# Documentos generados automáticamente**

\*.docx

\*.pdf

\*.pptx

# Caché

cache/

# Configuración de Webpack

webpack.config.js

# Bases de datos locales

db/

**5. Hallazgos clave y recomendaciones**

**Aspectos positivos**:

* El juego es funcional, atractivo y educativo, con una estructura modular.
* La integración de SweetAlert2 demuestra atención a la experiencia del usuario.
* El cronómetro añade dinamismo y compromiso.

**Áreas de mejora**:

1. **Estilos avanzados**: Incorporar librerías como Tailwind o Bootstrap.
2. **Gestión de estados**: Implementar Redux o contextos para mejorar la lógica.
3. **Pruebas unitarias**: Añadir pruebas unitarias para funciones clave.
4. **Escalabilidad**: Usar frameworks como React para manejar componentes dinámicos.

**Aprendizajes**: Este proyecto muestra cómo la programación básica puede evolucionar hacia un producto interactivo utilizando tecnologías simples pero efectivas. La integración de herramientas visuales como SweetAlert es un ejemplo claro de cómo mejorar la experiencia del usuario.

**Recomendación**: Para futuros proyectos, trabajar en modularidad del código y considerar tecnologías como TypeScript o frameworks modernos.