PROYECTO: TRIVIA DE CONOCIMIENTOS

IDEA: La idea de nuestra página web es que tenga una trivia donde el usuario pueda responder preguntas y retar su conocimiento acertando a las preguntas. La página tendrá diferentes preguntas acerca de entretenimiento que podrá responder sin ningún inconveniente.

* **Establecer funcionalidades**

**Básicas:**

* La página web debe permitir escoger una única respuesta a la pregunta. El usuario solo podrá elegir una opción de las disponibles para responder. Si elige la A no puede elegir la B al mismo tiempo.
* Al enviar la respuesta la página web debe indicarte si la respuesta es CORRECTA o INCORRECTA. En el caso de que la respuesta sea incorrecta mostrarte la POSIBLE RESPUESTA CORRECTA.
* Al enviar la respuesta correcta y mostrar resultado se mostrará un botón para continuar a otra pregunta. Al suceder eso el botón Enviar pregunta debe desaparecer y aparecer la otra pregunta.
* **Preguntas establecidas para el juego:**

<https://unilibrebog-my.sharepoint.com/:w:/g/personal/sarap-acunab_unilibre_edu_co/ES65Hh9ukZ1CkvbCcgK9d4kBm0NZXVY76lFec5RvCLJzbw?rtime=LF3QzOaf2kg>

* **Paso 1: Definir los diseños para el prototipo a desarrollar de la página web:**

Para esta fase utilizamos la página Balsamiq la cual es una herramienta que nos permite hacer mockups de algún proyecto, en este caso realizamos lo más básico que fue el prototipo de nuestra Trivia para el hito número 1.

Primer diseño:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Luego, se modificaron algunas cosas y al finalizar quedo el prototipo que planteamos de la siguiente manera: Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

* **Paso 2: Investigación**

Luego cada una de nosotras investigó sus propias dudas, en el caso de Melody, tener claro el concepto de aplicación web, para de ese modo tener una idea general de las tareas más importantes. Y en el caso de Sara investigó un poco acerca de Radio Button y cómo funciona en los formularios. También, realizó pruebas su funcionamiento y comportamiento en HTML.

Después de eso, ambas investigamos sobre metodología ágil, para así poder el día 27 de septiembre dividir las tareas de forma eficiente y coordinada.

* **Paso 3: Realizar el prototipo del diseño en usando HTML y CSS**

Para poder realizar el diseño más básico investigamos un poco acerca de HTML, viendo su sintaxis y funcionalidades de los elementos que creíamos que íbamos a necesitar y cada una en trabajo en equipo ir probando, así mismo se fueron agregando los estilos. En esta parte aun no se pudo lograr ubicar las opciones de respuesta en su lugar.

Lo más importante de esta parte fueron las pruebas y ver si funcionaba, borrar y volver a probar hasta llegar al estilo deseado, sobre todo en CSS, ya que íbamos aprendiendo en la marcha, compartiendo las ideas necesarias en conjunto.

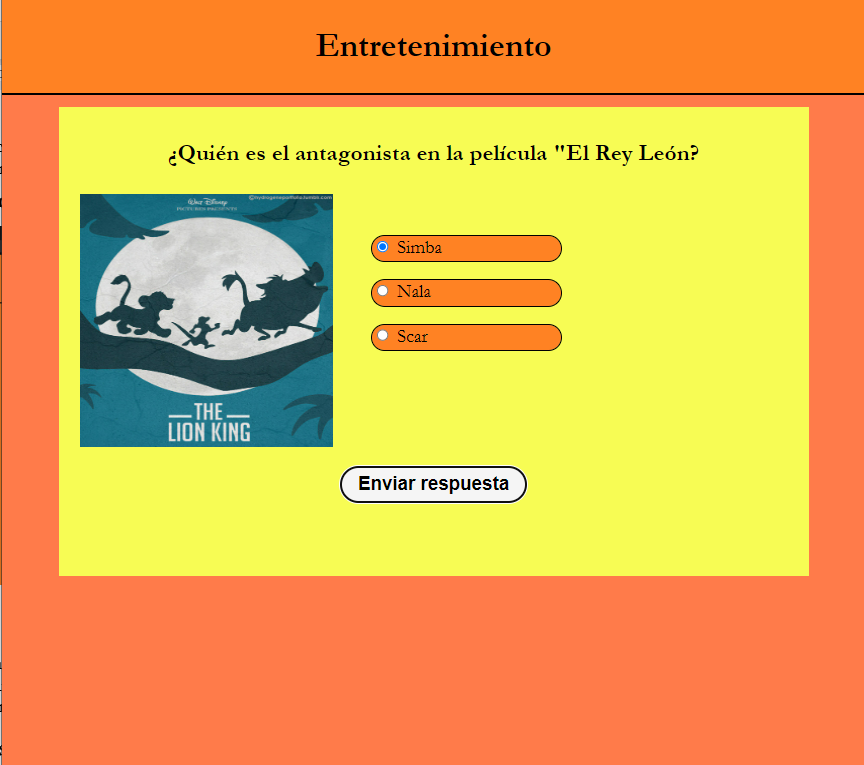
**Primer avance en HTML Y CSS**

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

* **Paso 4: Lograr estilo del prototipo**

Luego de muchas pruebas y errores se pudo lograr el diseño más básico que teníamos como objetivo ajustando cada elemento donde se había estipulado que estaría ubicado. Aquí, después de leer, investigar y ver videos se pudo lograr que las opciones de preguntas se ajustaran al lado de la imagen, para ello tuvimos que aprender sobre la forma de manipular los div y sus características por defecto.



**Diseño Logrado HTML Y CSS**

* **Paso 5: Relacionarnos con Java Scripts y mejorando diseño**

Después de dejar lista la estructura en HTML y definir los bloques, clases e identificadores necesarios para manipularlos por medio de CSS, dejamos de lado un rato la interfaz de la vista 1 porque nos enfocamos en las funcionalidades básicas de nuestra página, para esto tuvimos que investigar sobre el control de flujo de nuestro código, para que pudiera manejar los datos provenientes de la interacción con nuestro formulario.

Luego de eso, descubrimos que necesitábamos una forma de identificar a nuestros elementos HTML para poder crear variables y funciones que cumplieran con las funcionalidades requeridas, de esa forma llegamos a la sección de DOM, tuvimos que aprender sobre la forma en la que DOM representa a nuestro documento HTML.

Hasta acá, la segunda vista aún no ha sido creada, por lo que nos proponemos enfocarnos en crearla para enlazarla en el botón de continuar. Tambien, realizamos pruebas que hicimos con las diferentes formas de acceder a los elementos DOM.

Manipulación de los elementos DOM para crear nuestro flujo condicional de sentencias: Con el objetivo de entender bien los condicionales.

Texto

Descripción generada automáticamente

**Paso 6: Terminar de estructurar la interfaz de la vista 1 y estilizar elementos**

En esta parte nos seguimos enfocando en la interfaz, notamos que era mejor en este caso usar la etiqueta <a> para hacer el botón para pasar a la siguiente vista, por lo que tuvimos que aprender a añadirle un estilo al enlace para darle la apariencia que necesitábamos, eso se logró después de varias pruebas con el Css. También hicimos algunas modificaciones simples de apariencia respecto a los mensajes que indican si la respuesta es correcta o no. Este paso nos ayudó a posteriormente crear la vista 2 de forma más fácil ya que, teníamos el diseño anterior como base. En esta parte con la reunión con el coach pudimos entender que como estábamos empezando era mejor duplicar lo que teníamos en la vista 1 y personalizar la pregunta en la vida 2 sin tener que crear elementos complejos para cambiar la pregunta.

Por ello la vista 1 quedo de esta forma:



**Paso 7: Agregar segunda vista y enlazarla con la primera**

Creamos el archivo correspondiente a la 2da vista, con un archivo CSS aparte, ya que nos propusimos investigar sobre las otras etiquetas usadas en vez de los div, para organizar mejor la 2da vista.

Tuvimos que investigar sobre ese tema, ya que surgieron otras necesidades, tal como descubrir cómo crear un botón con a con el enlace, y poder modificar su apariencia.

Luego tuvimos que seguir buscando sobre control de flujo, ya que notamos que nuestro código en Js solo era funcional para la primera vista, pero no para la 2da. En esta parte aprendimos un poco acerca de los Id y las Clases que se asignan en el HTML. Luego de tener 1 vista 1 y 2 como guía se realizó la tercera vista.

Demoramos un poco en hacer que los botones aparecieran en tal vista y se ocultaban en la vida requerida. Ejemplo: Botón continuar solo aparece cuando se selecciona enviar pregunta y así mismo una vez se envié la pregunta ese botón debe desaparecer. Esto nos llevo bastante tiempo.

**Realizando pruebas VISTA 1:**

* **Prueba 1:** Si no selecciona ninguna opción y le da enviar se le manda un mensaje de alerta para que sepa que debe seleccionar si o si una de las posibles respuestas.

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

* **Prueba 2:** Si elige alguna respuesta incorrecta se le debe mostrar “INCORRECTO” y cuál era la respuesta correcta y debe aparecer un botón para pasar a la siguiente pregunta.

Interfaz de usuario gráfica, Diagrama

Descripción generada automáticamente

* **Prueba 3:** Si selecciona la opción correcta se le debe mostrar “INCORRECTO” y debe aparecer un botón para pasar a la siguiente pregunta.

Interfaz de usuario gráfica, Diagrama

Descripción generada automáticamente

Así mismo también se realizaron las mismas pruebas para le pregunta 2 y 3. Y se implemento por medio de Js accediendo a que clase tiene mi id del tipo de pregunta para así conocer en que pregunta se encuentra mi página.

**Paso 8: Para finalizar**

En la parte final acordamos no añadir más funciones a nuestro código y coger el ultima día para acomodar nombres descriptivos a las variables para poder guiarnos mejor y también decidimos ponernos a revisar el código e ir entendiendo cosas que se nos dificulten y tenerlas claras.

El proyecto se subió a un repositorio en GitHub para guardar todos los cambios por si en el caso editamos algo se tenga una copia y también para poder compartirlo:

<https://github.com/Acuna21/Proyecto-Laboratoria>

Tambien trabajamos en CodeSandbox y este es el enlace: <https://codesandbox.io/s/sharp-voice-dpbbgq?file=/js/index.js>