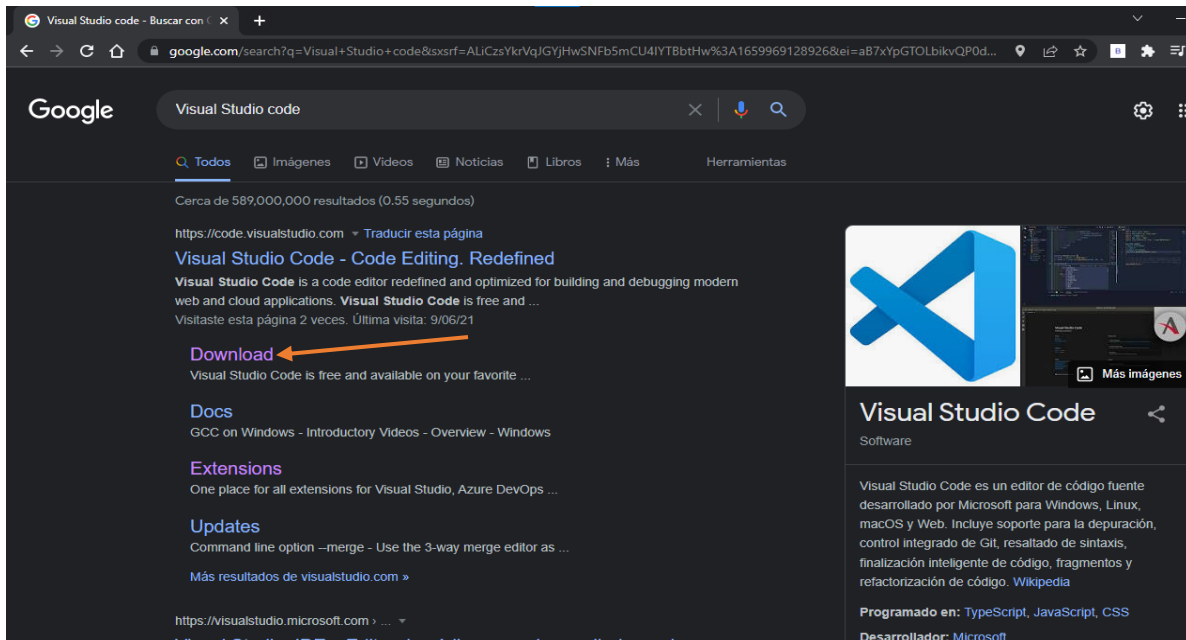
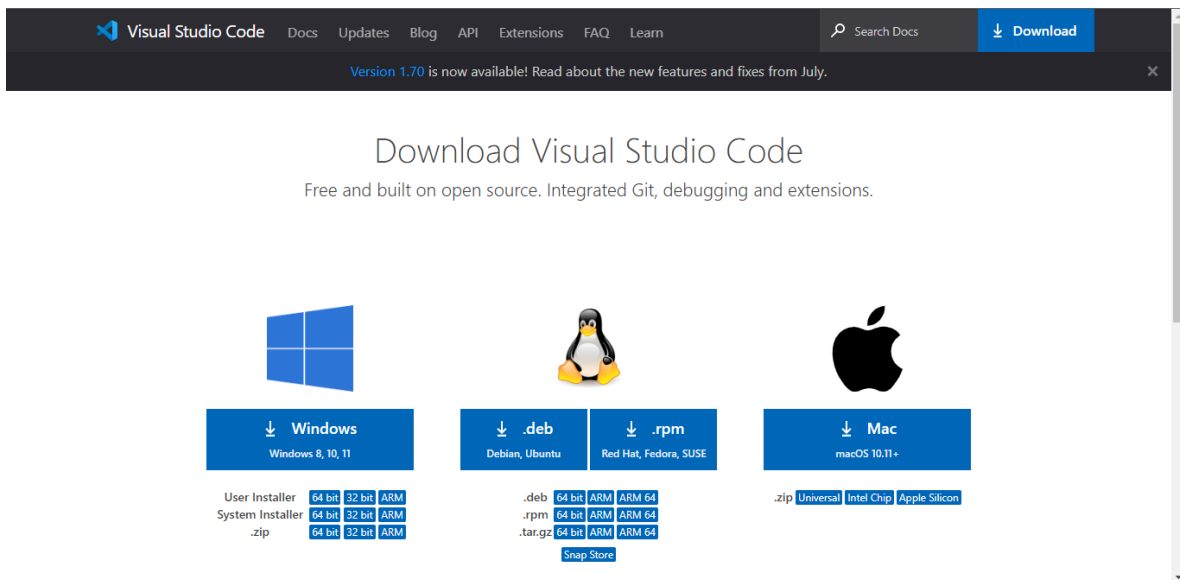


Proceso instalación Visual Studio Code

Paso 1: Escribimos en Google Visual Studio Code y seleccionamos donde dice “Download”.



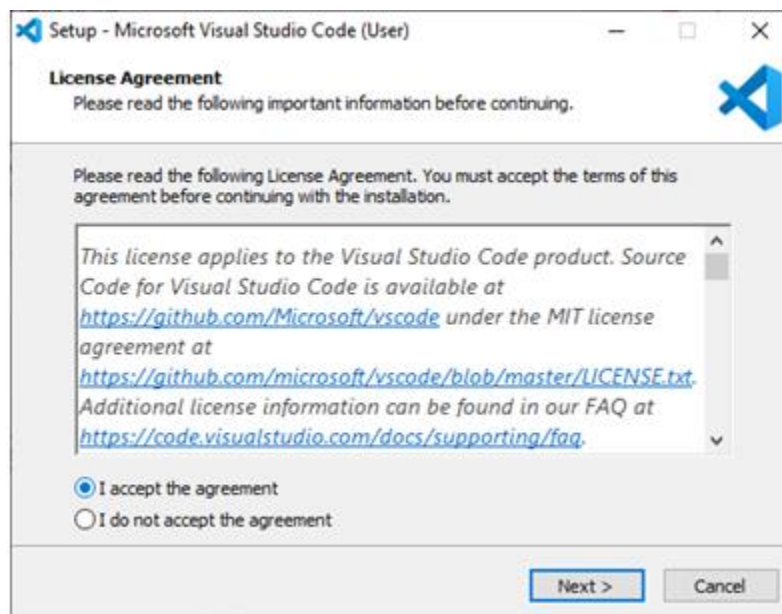
Paso 2: Seleccionamos el sistema operativo que tenemos y lo descargamos.



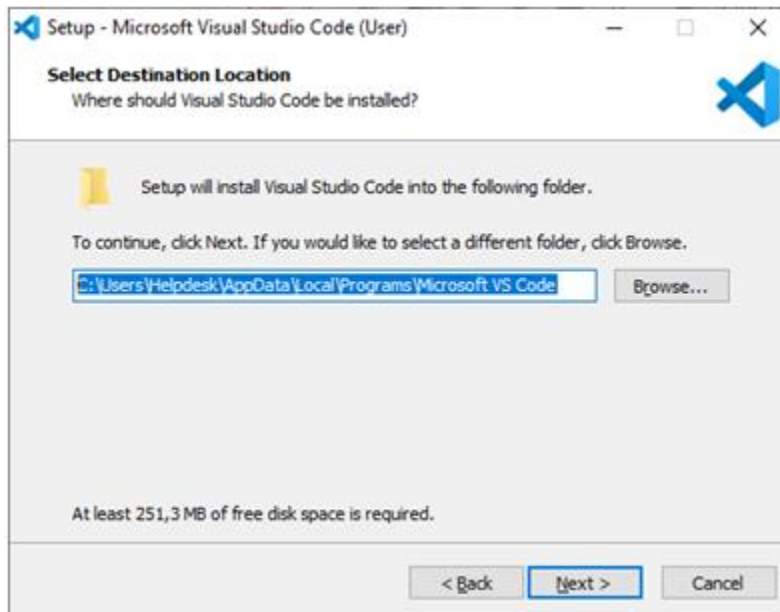
Paso 3: Al darle clic nos descargará un .exe, al cual le daremos clic encima.



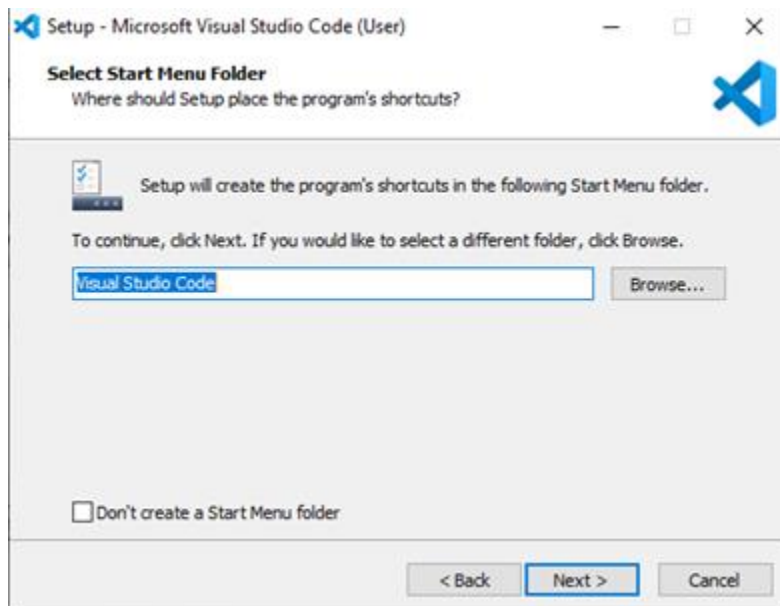
Paso 4: Lee y acepta el acuerdo de licencia. Haz clic en Next para continuar.



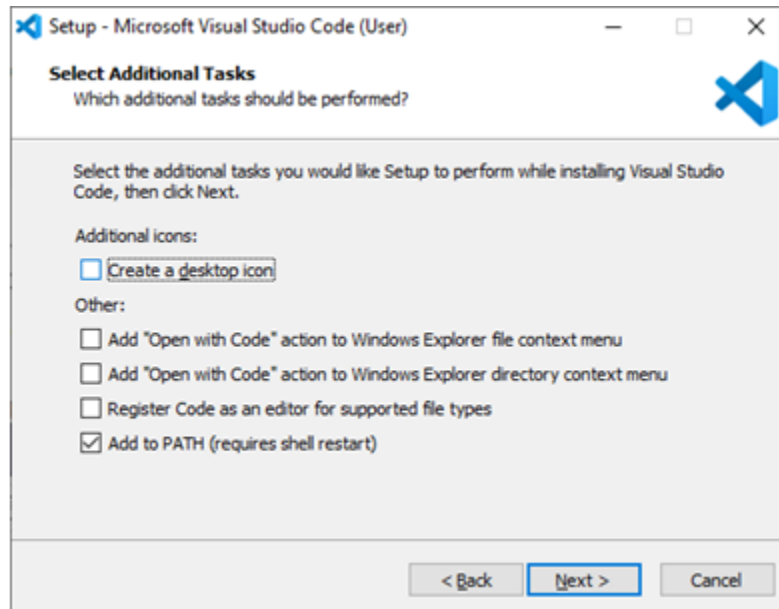
Paso 5: Puedes cambiar la ubicación de la carpeta de instalación o mantener la configuración predeterminada. Haz clic en Next para continuar.



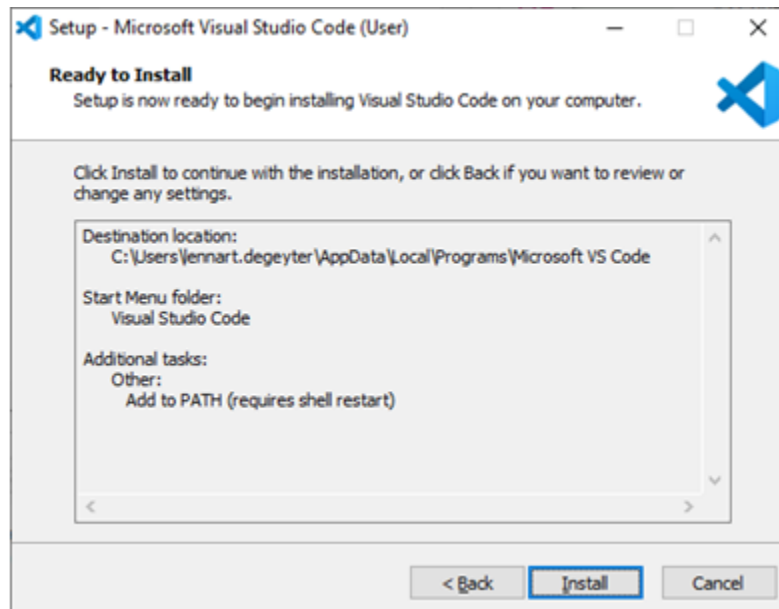
Paso 6: Elige si deseas cambiar el nombre de la carpeta de accesos directos en el menú Inicio o si no deseas instalar accesos directos en absoluto. Haz clic en Next.



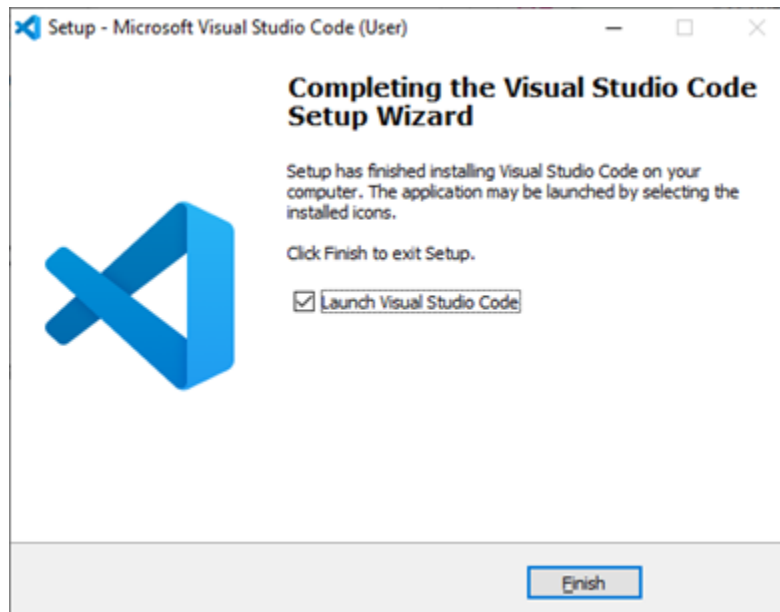
Paso 7: Selecciona las tareas adicionales, por ej. crear un icono en el escritorio o añadir opciones al menú contextual de Windows Explorer. Haz clic en Next.



Paso 8: Haz clic en Install para iniciar la instalación.

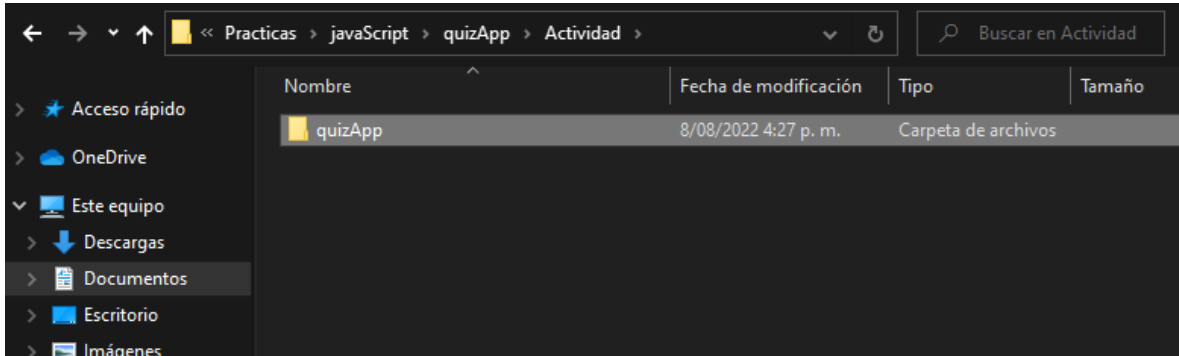


Paso 9: El programa está instalado y listo para usar. Haz clic en Finish para finalizar la instalación y lanzar el programa.

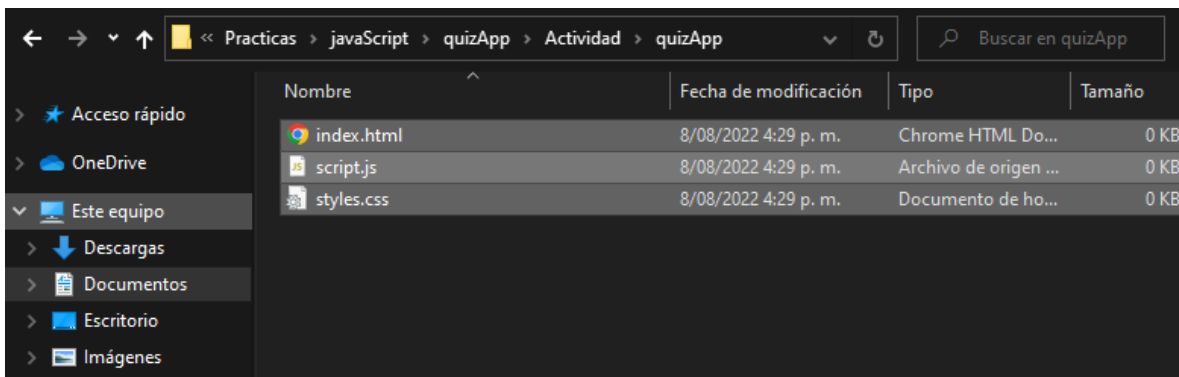


Creación de archivos

Creamos una carpeta raíz. En nuestro caso la llamaremos “quizApp”.

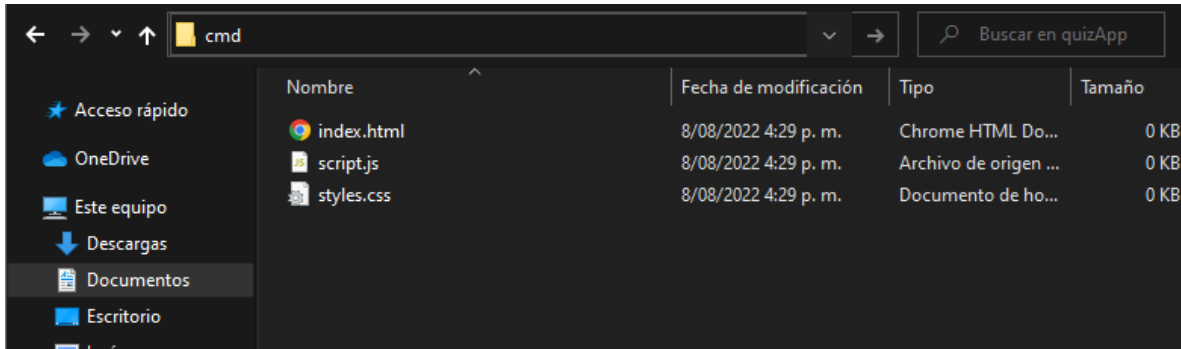


Dentro de esta, crearemos tres archivos. El primero llamado index.html, otro llamado styles.css y el último script.js (Es importante tener exactamente esa extensión en el archivo).

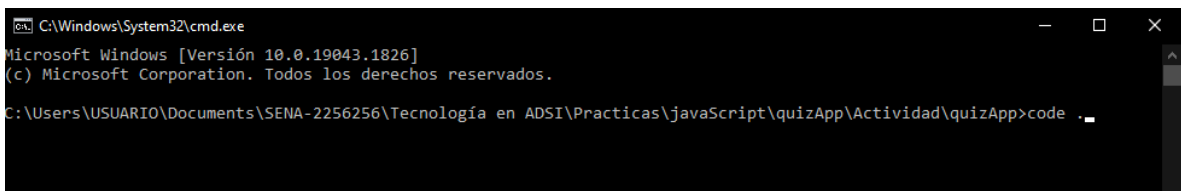


Project – Quiz App

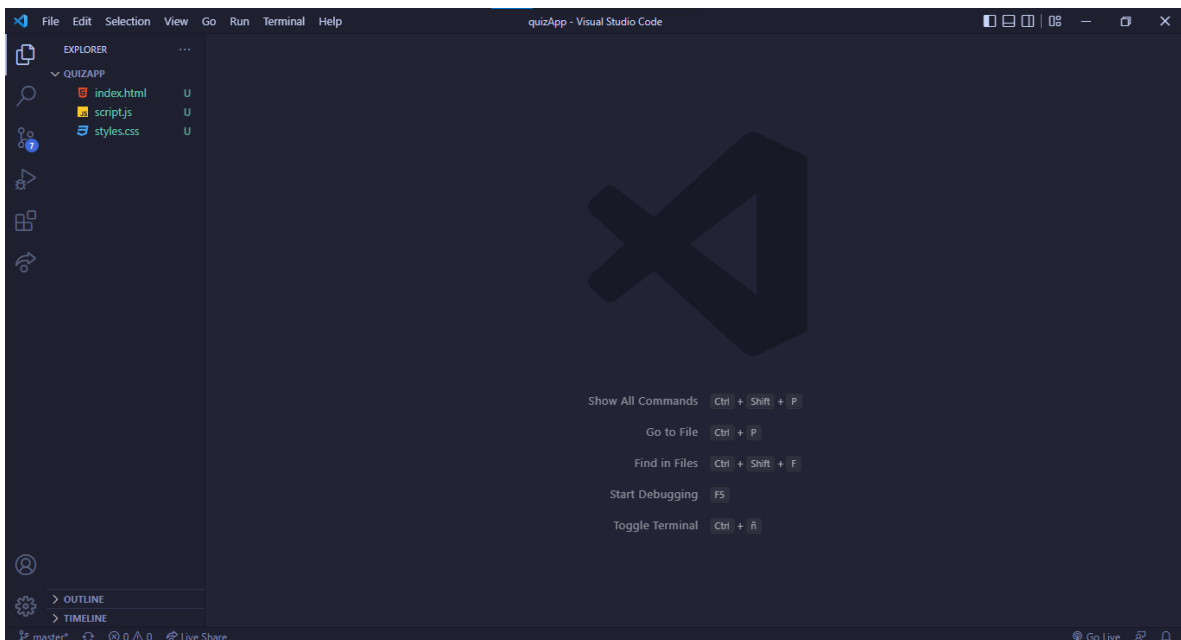
Para abrir el Visual Studio Code, haremos lo siguiente: En la barra superior, escribiremos cmd y le damos enter.



Este comando, nos abrirá una terminal, ahí dentro, solo tendremos que copiar “code .” y posteriormente darle enter, esto nos abrirá el Visual Studio Code.



Esto nos abrirá como anteriormente dije, el Visual Studio Code.



Ingresamos a index.html y codificamos lo básico: html, head y body.

```
index.html U X
index.html > html > body
1  <!DOCTYPE html>
2  <html lang="en">
3      <head>
4          <meta charset="UTF-8">
5          <meta http-equiv="X-UA-Compatible" content="IE=edge">
6          <meta name="viewport" content="width=device-width, initial-scale=1.0">
7          <title>Document</title>
8      </head>
9      <body>
10
11
12  </body>
</html>
```

Importamos la fuente que vayamos a utilizar y linkeamos el css y el javascript.

```
1  <!DOCTYPE html>
2  <html lang="en">
3      <head>
4          <meta charset="UTF-8">
5          <meta http-equiv="X-UA-Compatible" content="IE=edge">
6          <meta name="viewport" content="width=device-width, initial-scale=1.0">
7          <title>Quiz App</title>
8          <!-- Importamos la fuente que vamos a utilizar -->
9          <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;500&display=swap" rel="stylesheet">
10         <!-- Linkeamos el css -->
11         <link rel="stylesheet" href="styles.css">
12         <!-- Linkeamos el js -->
13         <script defer src="script.js"></script>
14     </head>
15     <body>
16
17
18 </body>
</html>
```

Dentro de la etiqueta “body” creamos un contenedor, donde estará toda la información, de igual manera, creamos un contenedor para las preguntas y respuestas, y luego otro contenedor para los botones.


```

15     <body>
16         <!-- Contenedor principal -->
17         <div class="container">
18             <!-- Contenedor de preguntas -->
19             <div id="question-container" class="hide">
20                 <!-- Contenedor para las preguntas -->
21                 <div id="question">Question</div>
22                 <!-- Contenedor para las respuestas -->
23                 <div id="answer-buttons" class="btn-grid">
24                     <button class="btn">Answer 1</button>
25                     <button class="btn">Answer 2</button>
26                     <button class="btn">Answer 3</button>
27                     <button class="btn">Answer 4</button>
28                 </div>
29             </div>
30             <!-- Contenedor de botones -->
31             <div class="controls">
32                 <button id="start-btn" class="start-btn btn">Start</button>
33                 <button id="next-btn" class="next-btn btn hide">Next</button>
34             </div>
35         </div>
36     </body>

```

Vamos al archivo css y comenzamos a darle estilos a nuestra página.

```

1  /* Estilos para toda la pagina como tal */
2  *, ::before, ::after {
3      box-sizing: border-box;
4      font-family: "Poppins", sans-serif;
5  }
6
7  /* Definimos las formas que puede tomar la variable --hue */
8  :root {
9      --hue-neutral: 200;
10     --hue-wrong: 0;
11     --hue-correct: 145;
12 }
13
14 /* Estilos para el cuerpo de la pagina */
15 body {
16     --hue: var(--hue-neutral);
17     padding: 0;
18     margin: 0;
19     display: flex;
20     width: 100vw;
21     height: 100vh;
22     justify-content: center;
23     align-items: center;
24     background-color: hsl(var(--hue), 100%, 20%);
25 }
26
27 /* Estilos para cuando el body tenga como clase correct */
28 body.correct {
29     --hue: var(--hue-correct);
30 }
31
32 /* Estilos para cuando el body tenga como clase wrong */
33 body.wrong {
34     --hue: var(--hue-wrong);
35 }
36
37 /* Estilos para el contenedor */
38 .container {
39     width: 800px;
40     max-width: 80%;
41     background-color: white;
42     border-radius: 5px;
43     padding: 10px;
44     box-shadow: 0 0 10px 2px;
45 }

```

```

47 /* Estilos para el contenedor de respuestas */
48 .btn-grid {
49     display: grid;
50     grid-template-columns: repeat(2, auto);
51     gap: 10px;
52     margin: 20px 0;
53 }
54
55 /* Estilos para los botones */
56 .btn {
57     --hue: var(--hue-neutral);
58     border: 1px solid hsl(var(--hue), 100%, 30%);
59     background-color: hsl(var(--hue), 100%, 50%);
60     border-radius: 5px;
61     padding: 5px 10px;
62     color: white;
63     outline: none;
64 }
65
66 /* Estilo para cuando se le pase el mouse
67 por encima al boton (Para eso es el hover) */
68 .btn:hover {
69     border-color: black;
70 }
71
72 /* Estilos para cuando el boton tenga como clase correct */
73 .btn.correct {
74     --hue: var(--hue-correct);
75     color: black;
76 }
77
78 /* Estilos para cuando el boton tenga como clase wrong */
79 .btn.wrong {
80     --hue: var(--hue-wrong);
81 }
82
83 /* Estilos para el los botones de next y start */
84 .start-btn, .next-btn {
85     font-size: 1.5rem;
86     font-weight: bold;
87     padding: 10px 20px;
88 }
89
90 /* Estilos para el contenedor de botones */
91 .controls {
92     display: flex;
93     justify-content: center;
94     align-items: center;
95 }

```

```

97 /* Estilos para la clase hide */
98 .hide {
99     display: none;
100 }

```

Ahora vamos al javascript (js) y comenzamos trayendo el botón de start, el de next, el contenedor de preguntas, las preguntas y los botones de respuestas.

```
1  /* Creamos constantes o variables, y en cada una almacenamos un elemento
2  traído por el id */
3  const startButton = document.getElementById('start-btn');
4  const nextButton = document.getElementById('next-btn');
5  const questionContainerElements = document.getElementById('question-container')
6  const questionElement = document.getElementById('question');
7  const answerButtonsElement = document.getElementById('answer-buttons');
8
9  let shuffledQuestions, currentQuestionIndex;
```

Ahora vamos a escuchar el evento (Click) que se realice en el botón de start, y correrá la función startGame. De igual manera escuchamos el evento del botón next y realizamos acciones.

```
11  /* Escuchamos el evento del boton de start, que debe ser click */
12  startButton.addEventListener('click', startGame);
13  /* Escuchamos el evento del boton de next, que debe ser click
14  y definimos la accion que realizara*/
15  nextButton.addEventListener('click', () => {
16    currentQuestionIndex++;
17    setNextQuestion();
18  });
```

Ahora comencemos con las funciones, empezaremos realizando la función para iniciar el juego. Y la siguiente función será para definir la siguiente pregunta.

```
20 // Funcion para iniciar el juego
21 function startGame() {
22   // Ocultamos el boton de start
23   startButton.classList.add('hide');
24   shuffledQuestions = questions.sort(() => Math.random() - .5);
25   currentQuestionIndex = 0;
26   // Y mostramos el contenedor de preguntas
27   questionContainerElements.classList.remove('hide');
28   setNextQuestion();
29 }
30
31 // Funcion para la siguiente pregunta
32 function setNextQuestion() {
33   // Reseteamos las preguntas
34   resetState();
35   // Y mostramos otra pregunta
36   showQuestion(shuffledQuestions[currentQuestionIndex]);
37 }
```

Ahora, haremos una función para mostrar las preguntas y mostrar las respuestas, cada respuesta tendrá un botón. Y creamos otra función para resetear las clases del body y removemos los botones de respuestas.

```
39 // Funcion para mostrar la pregunta
40 ✓ function showQuestion(question) {
41     // Se muestra el texto (La pregunta)
42     questionElement.innerText = question.question;
43     // Por cada respuesta que definamos abajo, se creará un boton
44     ✓ question.answers.forEach(answer => {
45         const button = document.createElement('button');
46         button.innerText = answer.text;
47         button.classList.add('btn');
48         // Evaluamos si la respuesta es correcta
49     ✓ if (answer.correct) {
50         |     button.dataset.correct = answer.correct;
51         |     }
52         // Escuchamos el evento (Que respuesta se selecciono)
53         button.addEventListener('click', selectAnswer);
54         answerButtonsElement.appendChild(button);
55     })
56 }
57
58 | // Funcion para resetear las preguntas
59 ✓ function resetState() {
60     // Limpiamos las clases (correct, wrong) de todo el body
61     clearStatusClass(document.body);
62     // Ocultamos el boton de next
63     nextButton.classList.add('hide');
64     ✓ while (answerButtonsElement.firstChild) {
65         |     answerButtonsElement.removeChild(answerButtonsElement.firstChild);
66         |     }
67 }
```

Creamos una función para ver si la respuesta que seleccione el usuario es la correcta o no.

```

69 // Funcion para la respuesta correcta
70 function selectAnswer(e) {
71 // Creamos dos constantes una para traer el boton
72 // y la otra para definir el correcto
73 const selectedButton = e.target;
74 const correct = selectedButton.dataset.correct;
75 // Ponemos el body de color verde
76 setStatusClass(document.body, correct);
77 // Creamos una instancia de array a partir de un objeto iterable
78 Array.from(answerButtonsElement.children).forEach(button => {
79     setStatusClass(button, button.dataset.correct);
80 })
81 // Mientras que las preguntas barajadas sea menor que las preguntas
82 // Se mostrara el boton de next
83 if (shuffledQuestions.length > currentQuestionIndex + 1) {
84     nextButton.classList.remove('hide');
85 } else {
86     // Si lo anterior no se cumple, cambiamos el nombre del boton a restart
87     startButton.innerText = `Restart`;
88     startButton.classList.remove('hide');
89 }
90 }

```

Crearemos las dos últimas clases, la primera será para definir la clase correct, y wrong, para poder luego ponerle los colores respectivos. Y la última función nos servirá para eliminar todas las clases correct y wrong, y así eliminamos y reseteamos todos los colores.

```

92 // Funcion para definir la clase correct
93 function setStatusClass(element, correct) {
94     clearStatusClass(element);
95     // Si es correct, añadimos la clase correct al elemento
96     if (correct) {
97         element.classList.add('correct');
98     } else {
99         // Si no es correct, añadimos la clase wrong al elemento
100         element.classList.add('wrong');
101     }
102 }
103
104 // Funcion para eliminar las clases (correct, wrong)
105 function clearStatusClass(element) {
106     element.classList.remove('correct');
107     element.classList.remove('wrong');
108 }

```

Ahora, definiremos las preguntas y respuestas (Podremos poner cuantas preguntas y respuestas queramos, obviamente siguiendo el mismo patrón).

```
110 // Estas son las preguntas
111 const questions = [
112   {
113     // Se define la pregunta
114     question: 'Cuanto es 2 + 2?',
115     // Se definen las respuestas
116     answers: [
117       { text: '4', correct: true },
118       { text: '22', correct: false }
119     ]
120   },
121   {
122     question: 'Cual es un lenguaje para IA?',
123     answers: [
124       { text: 'CSS', correct: false },
125       { text: 'HTML', correct: false },
126       { text: 'PYTHON', correct: true },
127       { text: 'REACT.JS', correct: false }
128     ]
129   },
130   {
131     question: 'Cual lenguaje es mas facil?',
132     answers: [
133       { text: 'HTML', correct: true },
134       { text: 'JAVASCRIPT', correct: true },
135       { text: 'PHP', correct: true },
136       { text: 'PYTHON', correct: true }
137     ]
138   },
139   {
140     question: 'Cuanto es 4 * 3',
141     answers: [
142       { text: '13', correct: false },
143       { text: '18', correct: false },
144       { text: '12', correct: true },
145       { text: '7', correct: false }
146     ]
147   }
148 ]
```

Hemos terminado el desarrollo del aplicativo, y se debería ver de la siguiente manera:

