# Generador de Código

Mayo 2023

Patricia Tovar Quintano
DESARROLLO DE APLICACIONES WEB

# Contenido

# Memoria Descriptiva

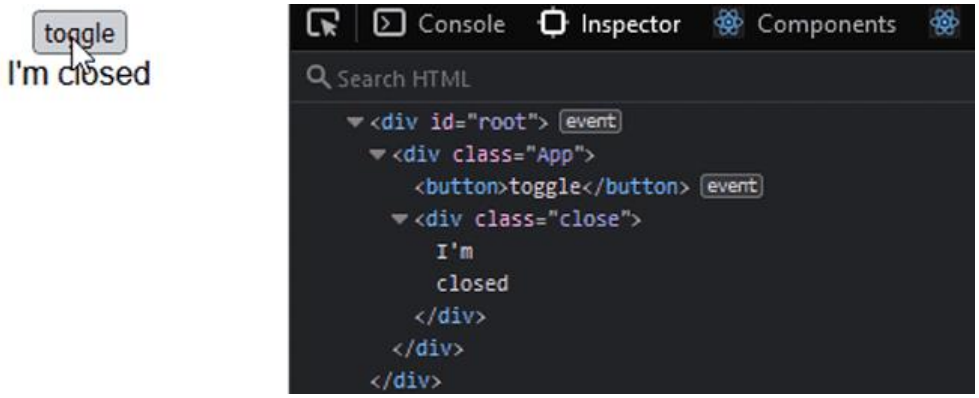
# 1.- Memoria descriptiva.

## 1.1.- Introducción.

### 1.1.1.- Objeto del proyecto:

El objetivo es desarrollar una aplicación Web de un Generador de Código que convierta HTML en JavaScript(JS) principalmente, permitiendo de esta forma generar DOM (ver en la página 33) de forma dinámica y eficiente, de esta forma evitando errores y facilitando dicha tarea.

Pasar HTML a JS es una tarea tediosa al generar DOM mediante eventos que se generan de forma asíncrona (ver en la página 33) al flujo del código, por ejemplo, hay un botón y *div* con texto, al hacer clic en un botón se modifiquen atributos, para darle unos estilos, y textos en el HTML.




Cogiendo la idea de generadores de código, como los de generar *Forms*, Botones, *GRID* que hay en internet, no hay ningún Generador de DOM, por lo que se vio una oportunidad.

*“The objective is to develop a Web application of a Code Generator that mainly converts HTML into JavaScript(JS), thus allowing to generate DOM dynamically and efficiently, thus avoiding errors and facilitating said task.*

*Passing HTML to JS is a tedious task when generating the DOM through events that are generated asynchronously to the flow of the code, for example, there is a button and a div with text, when a button is clicked attributes are modified, to give it some styles, and texts in the HTML.*

*Taking the idea of code generators, such as the ones to generate Forms, Buttons, GRID that are on the internet, there is no DOM Generator, so an opportunity was seen..”*

### 1.1.2.- Lenguajes empleados

Los lenguajes empleados para el desarrollo del proyecto


- JavaScript
- HTML
- CSS

Las IDE[1] utilizados

- Visual Studio Code (VSCode)

### 1.1.3.- Distribución

Las principales características del hosting necesarias son

- Hosting Free
- 5 MB de Almacenamiento
- Certificado SSL

Por lo que se ha elegido Netlify, que ofrece todo esto y además tiene un despliegue fácil y rápido, ya que permite conectar un repositorio GIT y cada vez que se realice algún cambio en el repositorio, Netlify lo compilara y desplegara automáticamente.

Además, Netlify utiliza una red global de entrega de Contenido (CDN[2]) permitiendo de esta forma la escalabilidad de forma automática para poder manejar un mayor tráfico.

### 1.1.4.- Requisitos de los clientes:

Los requisitos que deben cumplir los clientes es contar con acceso a Internet, un navegador, tener conocimientos de HTML y lo mínimo de JavaScript.

El navegador debe ser mínimamente actualizado, si no es así, la página puede llegar a presentar algunas deficiencias, de todas formas, las tecnologías aplicadas se han basado en la compatibilidad con estándares Web y para ello se ha ido comprobando dicha compatibilidad en la página Can I Use.

[1] Entorno de Desarrollo Integrado
[2] Content Delivery Network

### 1.1.5.- Licenciamiento:

El sitio Web no tiene ninguna Licencia, ya que todas las librerías y los lenguajes son de código abierto, lo que quiere decir que no tienen restricciones.

El software de desarrollo usado es de Código Abierto, esto no supone ningún tipo de obligación ni limitación.

La Aplicación no tiene recursos multimedia, ni tipografías especiales como para tener derecho de autor, lo único que hay es el logo y es de una página de libre Regalía (Shutterstock) así como los iconos utilizados, también son de libre distribución.

Como no se trata ningún tipo de dato y lo único que se llega a almacenar, es de forma local, el tema Claro/Oscuro y si el formulario ha sido enviado o no, no se necesita ningún derecho de autor, ya que un tercero no puede hacer cosas como subir alguna foto… Ni tampoco se necesita nada de protección de datos, ya que la información almacenada no es una información que pueda identificar directamente o indirectamente a una persona física como puede ser un correo, datos biométricos, datos de salud…

## 1.2.- Recursos.

### 1.2.1.- Hardware:

Para el desarrollo, implementación y distribución de la página web solo se ha necesitado un ordenador con unos requisitos mínimos como:

- Procesador (CPU), determina la capacidad para procesar las solicitudes entrantes y ejecutar las operaciones requeridas. Cuanto más potente sea el procesador mayor carga de trabajo podrá manejar y la respuesta será más rápida, el dispositivo con el que se ha desarrollado cuenta con un procesador i5
- Memoria (RAM), La memoria RAM es importante para mantener en ejecución las aplicaciones y procesos necesarios. Cuanta más memoria RAM tenga el equipo, mayor será la capacidad para manejar múltiples solicitudes y mantener un rendimiento óptimo. el dispositivo con el que se ha desarrollado cuenta con una RAM de 8GB.
- Almacenamiento (ROM), se necesita espacio de Almacenamiento, tampoco mucho ya que los ficheros no llegan a ocupar mucho, pero se ha separado en dos directorios el proyecto, por un lado, están todos los ficheros de desarrollo y por otro lado se ha separado los ficheros de producción, ya que para la

producción hay ficheros que no son necesarios como los de configuración, que estén indentados[3]…

### 1.2.2.- Software:

Para el desarrollo, implementación y distribución de mi página Web el Software que he necesitado es:

- node.js → Se ha necesitado implementar para poder usar tanto CodeMirror como Sass.
  Para instalar node.js es necesario acceder a la página web e instalarlo
- CodeMirror → está librería se ha usado para generar un editor de código en el navegador.
  Para instalar ver en la página 34
- Sass → Preprocesador de CSS, este lenguaje de secuencia de comandos permite interpretar y compilar en hojas de estilos.
  Para instalar ver en la página 38
- GitHub → Se usa tanto para el control de versiones como para poder hacer hosting al servidor.

### 1.2.3.- Humanos:

Las horas necesarias para cada fase han sido:

- Diseño de la Página (*Front End*) → 40h (incluye aprender e implementar *Sass* y *CodeMirror*)
- Generación de Código (*Back End*) → 70h
- Control de Errores → 2h
- Hosting Página Web → 8h
- Generación Documentación → 24h
- Pan de Negocio → 12 horas (incluye generar la base de datos y contactar con los programadores para hacer el estudio)

[3] Indentación es un anglicismo (de la palabra inglesa indentation) de uso común en informática; no es un término reconocido por la Real Academia Española. La Real Academia recomienda utilizar «sangrado». Este término significa mover hacia la derecha de forma tabulada el código y de esta forma distinguir visualmente la estructura de lenguaje.

### 1.2.4.- Previsión económica del coste del proyecto.

Esto es una previsión para un proyecto nuevo, con todos los gastos que tendría una persona al desarrollar el “Generador de DOM”, se va a reducir los costes al mínimo a 0€.

| Inversiones Anuales | Desde | Precio Medio |
|---|---|---|
| Equipo | 0€ | 500€ |
| Licencias a largo Plazo | 0€ | 200€ |
| Diseñador (Diseño y Contenido) | 0€ | 1000€ |

| Gastos Anuales | Desde | Precio Medio |
|---|---|---|
| Licencias a Corto Plazo | 0€ | 20€ |
| Servicio de Hosting | 0€ | 288€ |
| Registro de dominio | 0€ | 50€ |

## 1.3.- Descripción de la aplicación.

### 1.3.1.- Funcionamiento general:

El Programa espera a que el usuario introduzca código HTML.

En este modo presentación, se ha añadido unos ejemplos, para facilitar la prueba de código, al lado de este se ha configurado un botón para poder borrar los editores, que aparecerá tanto cuando se seleccione un ejemplo, como cuando se escriba código en el editor de texto de la izquierda.

El sistema genera el código una vez que detecte HTML mostrando el código JavaScript.

Una vez generado el código en JavaScript, en la esquina superior del editor derecho, aparecerá el icono que permitirá copiar el código en el Portapapeles como se ve en la imagen.

```
1  const CP = document.createElement('p');
2      CP.textContent('hola');
3  padre.appendChild(CP);
```

Además, si modificas el tema (Claro/Oscuro), actualizas la página y se queda guardado el tema elegido.

## 1.3.2.- Arquitectura:

### *1.3.2.1.- Diseño de las bases de datos.*

Según la estructura de la página web no se ha considerado necesario almacenar datos, ya que al manejar el texto introducido se convierte en HTML, y reconoce los atributos, elementos… y no se necesita validar nada.

Se ha contemplado generar Usuarios, pero con la funcionalidad de esta versión no es necesario implementar gestión de usuarios, en un futuro se podría implementar un histórico, y en ese caso si tiene sentido implementar Usuarios para llevar este registro.

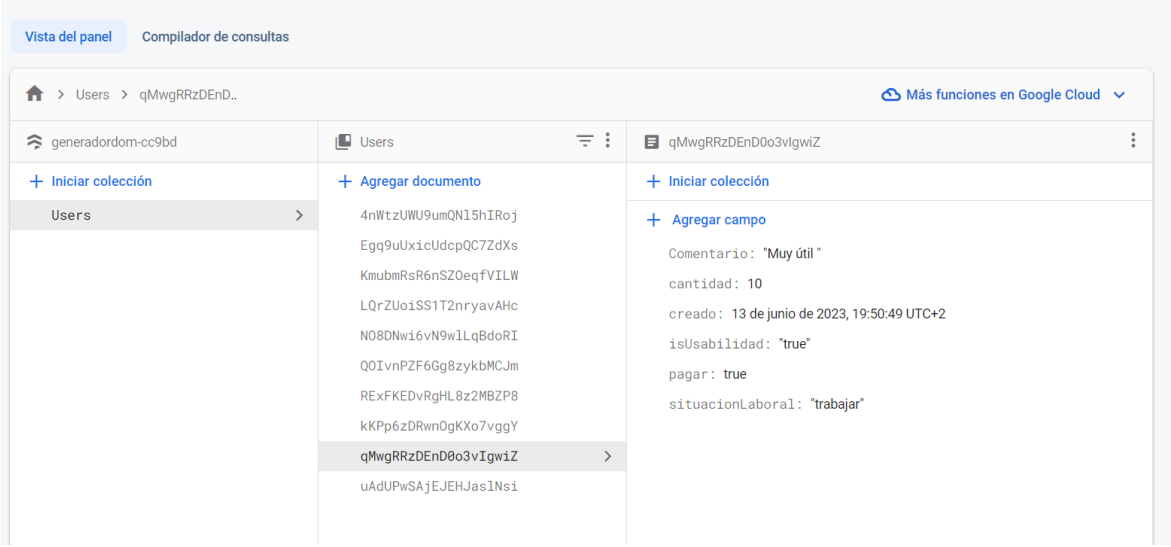
Para tratar con alguna base de datos, se implemento un estudio de mercado en esta versión de presentación para tener un *feedback* con algunos programadores que va a ser nuestro target, dichas respuestas se almacenarán en una base de datos con formato JSON. Contando con la siguiente estructura.

```
{
  "Users": {
    "user_id_1": {
      "creado": "fecha",
      "situacionLaboral": "situacionActual",
      "isUsabilidad": "usabilidad",
      "Comentario": "comentarios",
      "pagar": "precio",
      "cantidad": "cantidad"
    },
    "user_id_2": {
      "creado": "fecha",
      "situacionLaboral": "situacionActual",
      "isUsabilidad": "usabilidad",
      "Comentario": "comentarios",
      "pagar": "precio",
      "cantidad": "cantidad"
    }
  }
}
```

Los “*user_id_num*” se generan de forma automática.

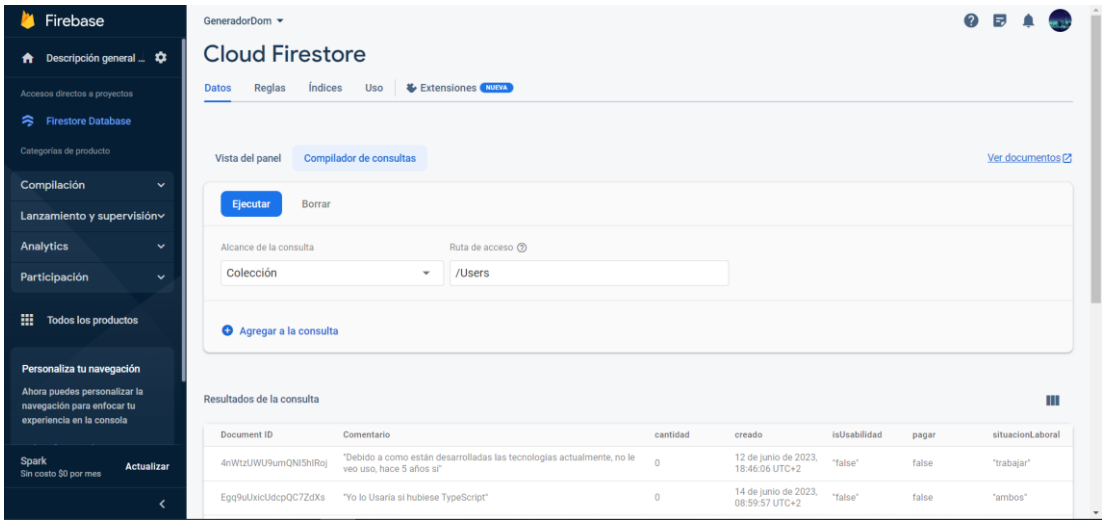
De esta forma se generará un formulario con la situación laboral, su usabilidad, si estaría dispuesto a pagar y que rango junto un comentario.

Trabajamos con Firebase ver más en Anexo/Manual/5.2.2.- Firebase, esto nos permite simplificar el proceso de Almacenamiento, Firebase genera Colecciones que en un SQL seria las entidades, luego en las colecciones están los documentos y estos documentos ya tienen los diferentes campos




Hay dos formas de ver los datos:

Firebase permite ver los datos en el compilador de consultas




Para poder ver los datos sin necesidad de hacerse usuario se ha generado un documento, ya que para disfrutar del plan gratuito no te permite generar una unidad de trabajo.

El fichero guardado como viewDatabase.html se encuentra en la siguiente Ruta https://github.com/Pattov/generadorDom/tree/master/otros aunque a continuación se adjunta el código

```
<!DOCTYPE html>
<html>
<head>
  <title>Tabla de datos de Firestore</title>
     <script    src="https://www.gstatic.com/firebasejs/8.2.7/firebase-app.js"></script>
     <script    src="https://www.gstatic.com/firebasejs/8.2.7/firebase-firestore.js"></script>
  <style>
    #tabla-datos {
      width: 100%;
      border-collapse: collapse;
    }

    #tabla-datos th, #tabla-datos td {
      padding: 8px;
      text-align: left;
      border-bottom: 1px solid #ddd;
    }

    #tabla-datos th {
      background-color: #f2f2f2;
      font-weight: bold;
    }

    #tabla-datos tbody tr:nth-child(even) {
      background-color: #f9f9f9;
    }

    #tabla-datos tbody tr:hover {
      background-color: #eaeaea;
    }
  </style>

</head>
<body>
  <h1>Tabla de datos de Firestore</h1>

  <table id="tabla-datos">
    <thead>
      <tr>
        <th>Creado</th>
        <th>Situación Laboral</th>
        <th>Usabilidad</th>
```

```
        <th>Comentario</th>
        <th>Pagarias</th>
        <th>Cantidad</th>
      </tr>
    </thead>
    <tbody></tbody>
  </table>

  <script>
    var firebaseConfig = {
      // Configuración de GeneradorDom Firebase
      apiKey: "AIzaSyC0SvehFakjZE0FvwqBejgPuvXHQdU9G4k",
      authDomain: "generadordom-cc9bd.firebaseapp.com",
      projectId: "generadordom-cc9bd",
      storageBucket: "generadordom-cc9bd.appspot.com",
      messagingSenderId: "989754279675",
      appId: "1:989754279675:web:648b80e2b4b0d7aedb125e"
    };

    // Inicializa Firebase
    firebase.initializeApp(firebaseConfig);

    // Obtiene una referencia a la colección de Firestore que se desea
mostrar
    var db = firebase.firestore();
    var collectionRef = db.collection('Users');
    // Obtiene una referencia al elemento tbody de la tabla
    var tbody = document.querySelector('#tabla-datos tbody');

    // Obtiene los datos de Firestore y muestra en la tabla
    collectionRef.get().then(function(querySnapshot) {
      querySnapshot.forEach(function(doc) {
        // Obtiene los datos de cada documento
        var data = doc.data();
        // Crea una nueva fila en la tabla
        var row = document.createElement('tr');

        // Convierte el campo "creado" a una fecha legible
        var timestamp = new Date(data.creado.seconds * 1000);
        var fecha = timestamp.toLocaleDateString();
        var hora = timestamp.toLocaleTimeString();

        // Crea las celdas y asigna los valores de los datos
        var creadoCell = document.createElement('td');
        creadoCell.textContent = fecha + " " + hora;;
        row.appendChild(creadoCell);

        var situacionLaboralCell = document.createElement('td');
```

```
        situacionLaboralCell.textContent = data.situacionLaboral;
        row.appendChild(situacionLaboralCell);

        var usabilidadCell = document.createElement('td');
        usabilidadCell.textContent = data.isUsabilidad;
        row.appendChild(usabilidadCell);

        var comentarioCell = document.createElement('td');
        comentarioCell.textContent = data.Comentario;
        row.appendChild(comentarioCell);

        var precioCell = document.createElement('td');
        precioCell.textContent = data.pagar;
        row.appendChild(precioCell);

        var cantidadCell = document.createElement('td');
        cantidadCell.textContent = data.cantidad;
        row.appendChild(cantidadCell);
        // Agrega la fila a la tabla
        tbody.appendChild(row);
      });
    });
  </script>
</body>
</html>
```

Y se verá algo como esto

**Tabla de datos de Firestore**

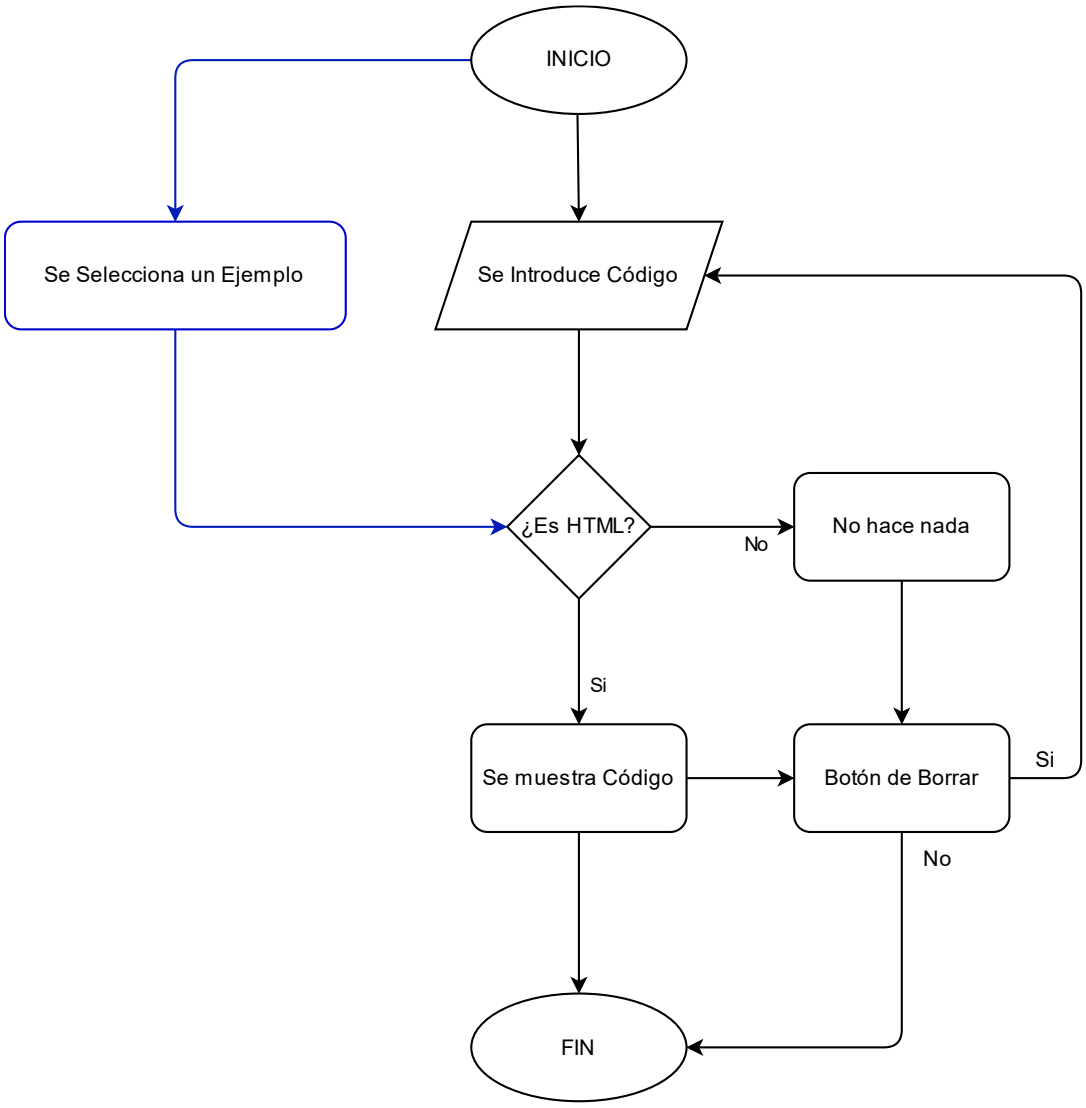
| Creado | Situación Laboral | Usabilidad | Comentario | Pagarias | Cantidad |
|---|---|---|---|---|---|
| 12/6/2023 18:46:06 | trabajar | false | Debido a como están desarrolladas las tecnologías actualmente, no le veo uso, hace 5 años si | false | 0 |
| 14/6/2023 8:59:57 | ambos | false | Yo lo Usaría si hubiese TypeScript | false | 0 |

### *1.3.2.2.- Arquitectura del sistema.*

Debido al tipo de Proyecto se ha generado un Diagrama de Flujos para entender el flujo del Programa.

La parte Azul hace referencia al proceso que se realiza en el modelo de presentación que se usara para presentar el proyecto.

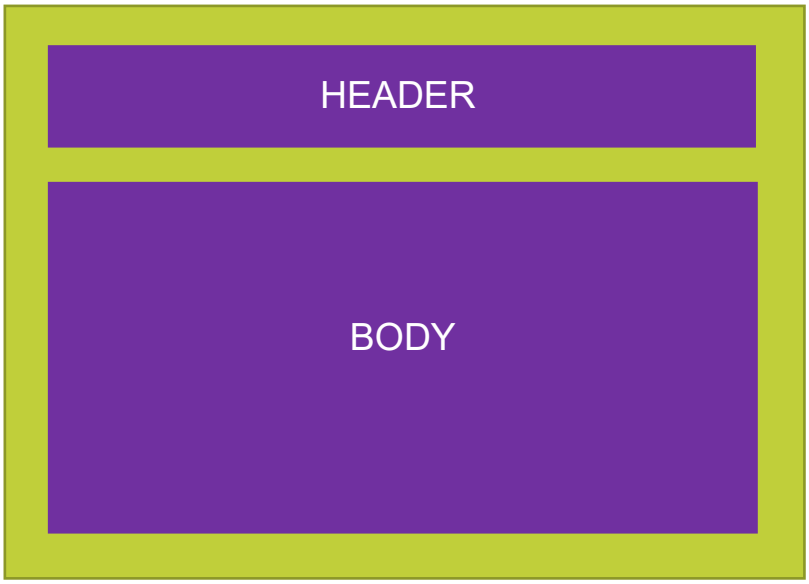
La parte Negra es el flujo normal del código.


INICIO
Se Selecciona un Ejemplo
Se Introduce Código
¿Es HTML?
No
No hace nada
Si
Se muestra Código
Botón de Borrar
Si
No
FIN

## 1.3.3.- Interfaz.

### *1.3.3.1.- Características generales:*

La interfaz cuanta de dos partes, el encabezado y el cuerpo




Partes del Header


Partes de Body

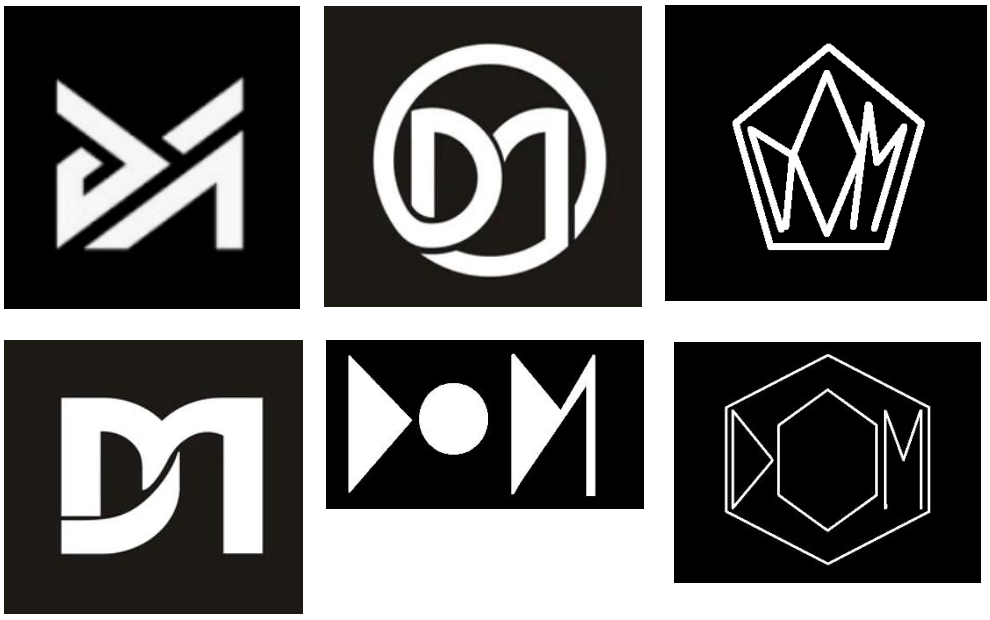
Paleta de colores

Se ha decidido hacer una paleta de elegante contando solo con el blanco, negro, gris para algún detalle y verde como único color

Exported from Coolors.co https://coolors.co/ffffff-d5d8d4-18f276-0c0c0d

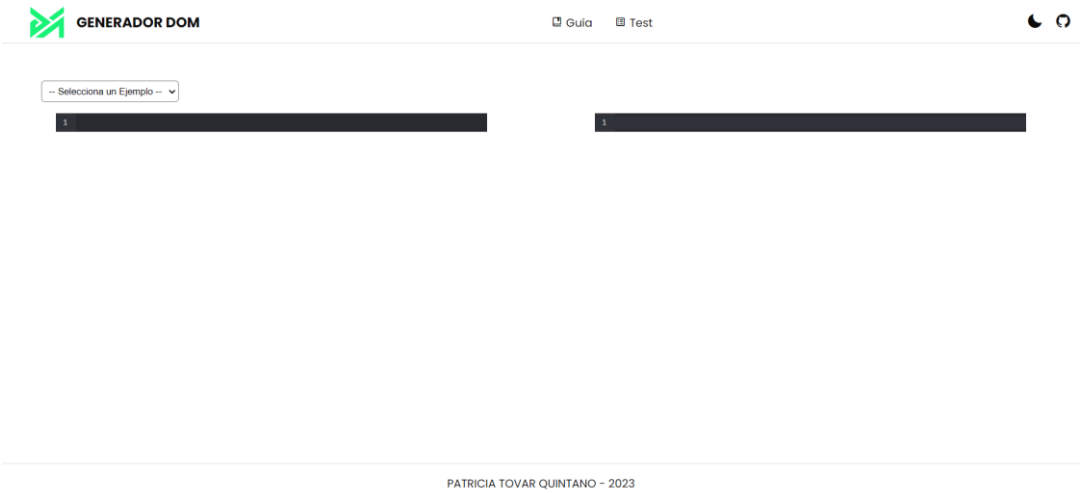
Icono

En el icono se buscaba algo sencillo y fácil, de acuerdo con la estética de la aplicación, partiendo de la base de querer jugar con la palabra DOM, ya sea solo letras o la palabra completa algunas ideas que se tuvieron en cuenta.



Al final la elegida fue la primera opción, figuras muy angulosas y queda bien con todo

El resultado final es este




Y con el código relleno se ve algo como esto


### *1.3.3.2.- Adaptación a dispositivos móviles:*

Se considera que no será necesario la adaptación a dispositivos móviles, ya que a la hora de desarrollar es más fácil en PC que en móvil, no se cuenta con que tenga usabilidad. De todas formas, la página se ha hecho responsive.

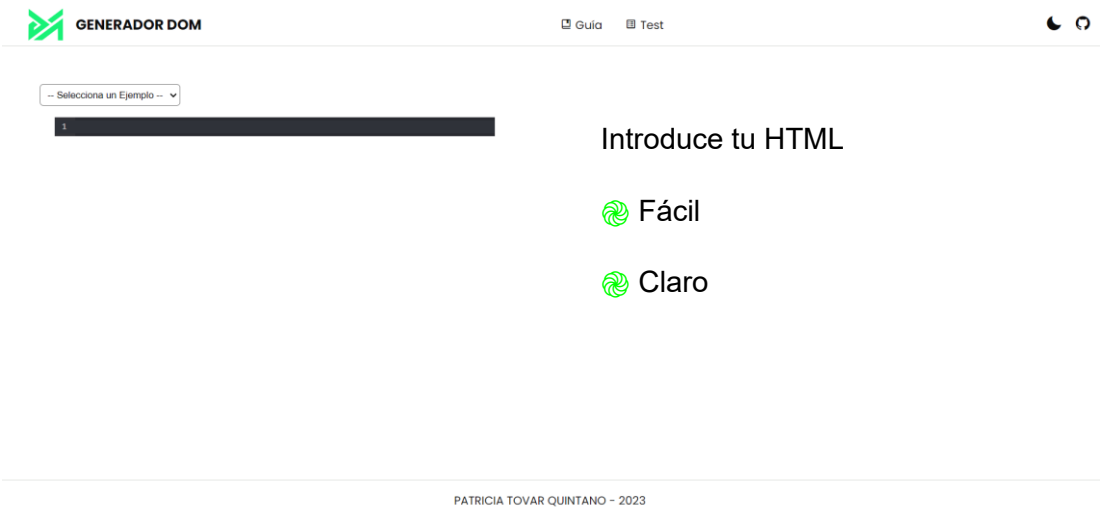
#### *1.3.3.3.- Usabilidad/accesibilidad:*

No se ha contado con que la página sea accesible para todas las personas, hubiese sido útil implementar algo de lectura o por ejemplo las guías tuviesen algún tipo de media para poder escucharlo, aunque sin quererlo si que hay un buen contraste entre el fondo y las letras.

En cuanto a la usabilidad la aplicación esta centrada en un target especializado por lo que no cualquier usuario puede entender la finalidad de generar este código. La navegación es intuitiva, ya que se ha considerado el uso normal de leer de izquierda a derecha por lo que se introduce código en la izquierda y aparece en la derecha. De todas formas, solo se permite escribir en la izquierda de esta forma evitando posibles errores.

El diseño es limpio e intuitivo, el tiempo de carga es rápido y toda la pagina en general es muy sencilla.

Como posibles mejoras se ha considerado que al iniciar la página la parte derecha en pantalla completa o con dimensiones superiores a 768px, se tape con un *div* que tape el panel de la derecha, permitiendo solo la vista en la izquierda, y en la zona derecha poniendo alguna instrucción, como se ve a continuación.




Y cuando se escriba o seleccione un ejemplo desaparezca.

## 1.4.- Autoevaluación y conclusiones.

### 1.4.1.- Valoración del trabajo y dificultades encontradas.

Para mi gusto me hubiese gustado contar con más tiempo, compaginar un trabajo, las practicas y me ha dejado muy poco tiempo para el proyecto, e ir muy apurada.

En cuanto dificultades encontradas muchas, los editores de código ha sido lo que mas me ha costado sin duda, la documentación esta de la versión anterior, y de la 6 hay muy poca con ejemplos prácticos, la generación del código en JS, hacer funciones recursivas, generar un objeto con los elementos y sus atributos de forma correcta.

Considero que ahora me desenvuelvo bien en la implementación de librerías, he tenido un primer contacto con node.js y la modularizacion, y este tipo de conocimientos me sirve para otros lenguajes como puede ser en .net con las dependencias.

### 1.4.2.- Valoración de la herramienta o aplicación desarrollada.

A pesar de las adversidades, he conseguido alcanzar la idea básica de lo que buscaba y sigo pensando que es un acierto ir por este tipo de aplicación al menos a nivel más básico de JS.

Además, viendo los resultados del estudio de mercado reafirma esta idea.

### 1.4.3.- Conclusiones finales:

Con el diseño he conseguido lo que buscaba, algo simple y elegante, los elementos que he generado en segundo plano me han dado algo de problemas.

Me he encontrado en el dilema de tener que descartar cosas que quería implementar e ideas que me han surgido, pero para un prototipo era necesario, sobre todo con mi principal factor que era el tiempo y las cosas que he implementado son cosas que suman a la usabilidad del proyecto como puede ser el botón de copiar.

Hay muchas cosas que en mi aplicación no eran necesarias y a la hora de generar la documentación he encontrado apartados en los que no he podido rellenar, como puede ser las Bases de datos, diagramas de casos de uso. Mi aplicación es “sencilla” en cuanto al funcionamiento general, pero es muy “abstracto” el concepto.

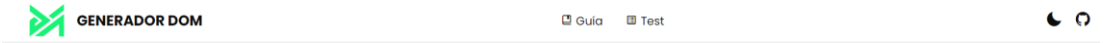
En definitiva, estoy muy contenta porque yo partía con un problema mientras cursaba 2º de DAW y me llena de satisfacción hacer esto, viendo el feedback que he tenido creo que no soy la única, aunque a nivel de una persona mas experimentada esto ya no es necesario, ya que se usan aplicaciones mas complejas que ya implementan este o servicios similares y también he podido aplicar pequeños matices que he aprendido en mis prácticas, como puede ser el despliegue de una aplicación.

# Manual de Usuario

# 2.- Manual de usuario.

Los elementos con los que se puede interactuar en la siguiente imagen aparecen numerados, a lo largo del manual aparecerán con un nombre y el numero de la imagen.


En la **parte superior** cuenta con




En la parte central hay:

- Enlace Guía [1] → abre un modal con todo lo necesario y posibles dudas que puedan surgir durante el uso de la aplicación
- Enlace Test [2] → abre el modal del formulario para el estudio de mercado. Esta opción solo está en la versión de presentación.

En la parte derecha cuenta con dos iconos

- El primer icono (Luna/Sol) [3] → Modifica el Tema Claro/Oscuro
- El segundo icono (GitHub) [4] → Enlace al repositorio

la **parte central** se divide en dos


En la parte izquierda hay tres elementos, aunque solo se vean dos

- Ejemplos [5] → Para el modo presentación tengo varios ejemplos en los que al seleccionarlo se genera código directamente en el editor HTML
- Icono Limpiar [6] → Al generar código ya sea con los ejemplos o al introducir texto en el HTML se hace visible el botón de borrar, este limpia el editor HTML (y por defecto el Editor JS)
- Editor de HTML[7] → Editor de Código de HTML (input)

En la parte derecha hay dos elementos, como en el ejemplo anterior se ve solo uno

- Icono Borrar [8] → Copia el código generado en el editor JS en el Portapapeles, solo aparece cuando hay código.
- Editor de JS [9] → Editor de Código de JS (output)

# Manual de Código

# 3.- Manual de código.

Para acceder al Manual estará todo en el repositorio del proyecto en la siguiente Ruta https://github.com/Pattov/generadorDom.git

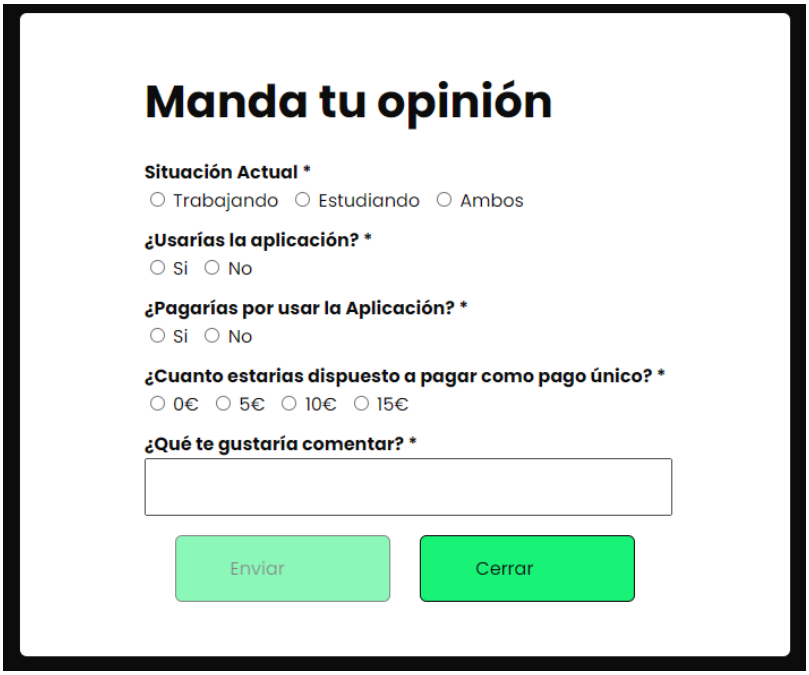
O en la página web https://generadordom.netlify.app/ en el icono superior de la derecha 

# Plan de Negocio

# 4.- Plan de negocio.

## 4.1.- Estudio de Mercado

Se realiza una encuesta a 10 personas dedicadas a la programación. En dicha encuesta se hacen preguntas para conocer los consumidores potenciales, unos posibles ingresos esperados, y posibles ventajas y desventajas del producto.




### 4.1.1.- Análisis Por Preguntas

#### *4.1.1.1.- Situación Actual.*

La finalidad de esta pregunta es conocer la utilidad que le puede dar y quiénes serían los consumidores potenciales, catalogándoles en estudiantes o trabajadores.

#### *4.1.1.2.- ¿Usarías la aplicación?*

Esta cuestión nos ayuda de forma directa a saber si le darían uso o no.

#### *4.1.1.3.- ¿Pagarías por usar la Aplicación?*

Al responder aquí se facilita una información muy útil sobre si se tendría una rentabilidad económica con ella o no.

### *4.1.1.4.- ¿Cuánto estarías dispuesto a pagar como pago único?*

Aquí se permite hacer un estudio tanto del precio al que valorar la aplicación, como de unos ingresos que se pueden esperar.

### *4.1.1.5.- ¿Qué te gustaría comentar?*

Espacio que pueden usar los encuestados para dar una opinión propia y más extensa que con las preguntas anteriores e incluir posibles mejoras. De esta pregunta se pueden analizar posibles ventajas y desventajas que no hayan sido analizadas anteriormente.

## 4.1.2.- Análisis de Resultados

En la siguiente imagen se muestran las respuestas de todas las personas encuestadas. Dichas respuestas son las que se analizan a continuación una a una.

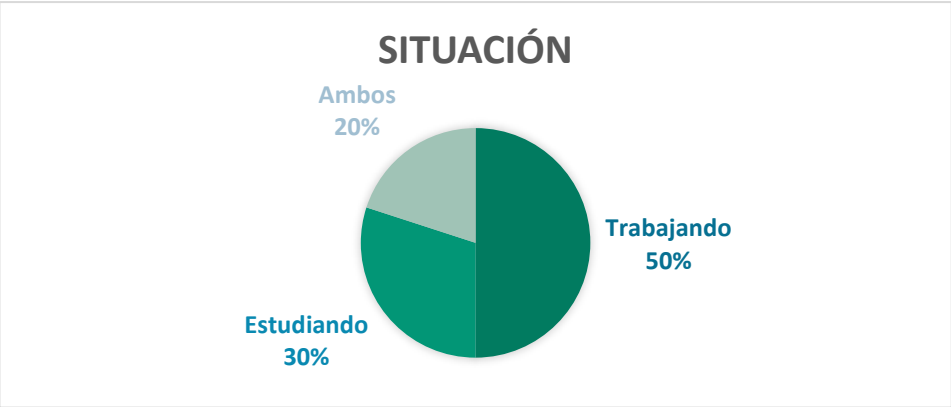
**Tabla de datos de Firestore**

| Creado | Situación Laboral | Usabilidad | Comentario | Pagarías | Cantidad |
|---|---|---|---|---|---|
| 12/6/2023 18:46:06 | trabajar | false | Debido a como están desarrolladas las tecnologías actualmente, no le veo uso, hace 5 años sí | false | 0 |
| 14/6/2023 8:59:57 | ambos | false | Yo lo Usaría si hubiese TypeScript | false | 0 |
| 12/6/2023 20:41:27 | trabajar | false | Yo uso Angular, que no usa JS, a lo mucho lo implementaría como una dependencia | false | 0 |
| 14/6/2023 21:05:42 | trabajar | false | Es una idea interesante, aunque queda lejos de ser cómodo y práctico. Quizá sería importante añadir algunas funcionalidades y opciones para la usabilidad. También procuraría cuidar el apartado visual. Por lo demás es una herramienta funcional que cumple con lo que promete. | false | 0 |
| 13/6/2023 21:55:06 | estudiar | true | Muy interesante! | true | 10 |
| 14/6/2023 14:44:05 | trabajar | true | Me parece una herramienta bastante útil, favorece el ahorro de costes en infraestructura. Hay alternativas fuertes en el mercado como puede ser React, pero sin lugar a duda, la idea es interesante. | true | 5 |
| 14/6/2023 22:17:59 | estudiar | true | Muy útil, pero añadiría jQuery. | false | 0 |
| 14/6/2023 22:12:58 | ambos | true | Aplicación muy útil para ayudarme tanto con el proyecto de clase como con el trabajo diario (Cypress). | true | 15 |
| 13/6/2023 19:50:49 | trabajar | true | Muy útil | true | 10 |
| 14/6/2023 11:36:50 | estudiar | true | Ojalá hubiese existido antes, lo he pasado mal en Desarrollo web en entorno cliente, en el proyecto me hubiese venido muy bien | true | 5 |

### *4.1.2.1.- Situación Actual. Posibles respuestas*

- Trabajando: 50%
- Estudiando: 30%
- Ambos: 20%

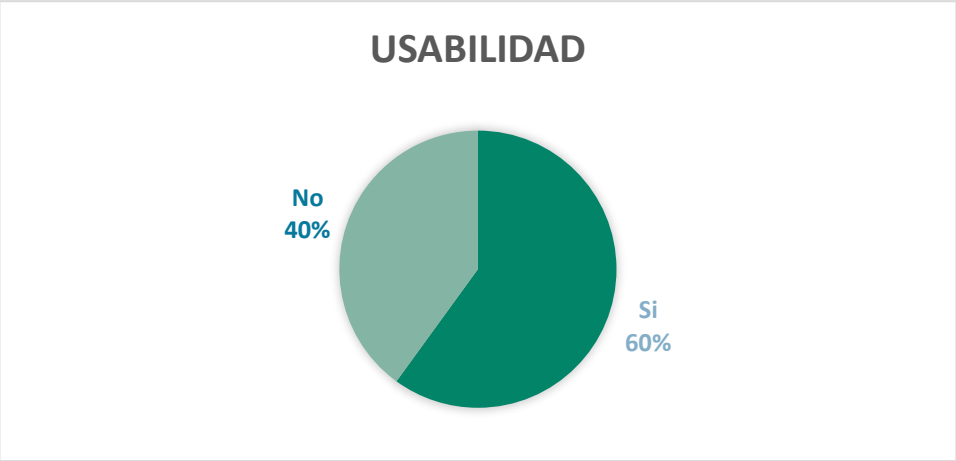
Con estos datos podemos conocer que la mayoría de las personas está trabajando.



#### *4.1.2.2.- ¿Usarías la aplicación? Posibles respuestas*

- Si: 60%
- No: 40%

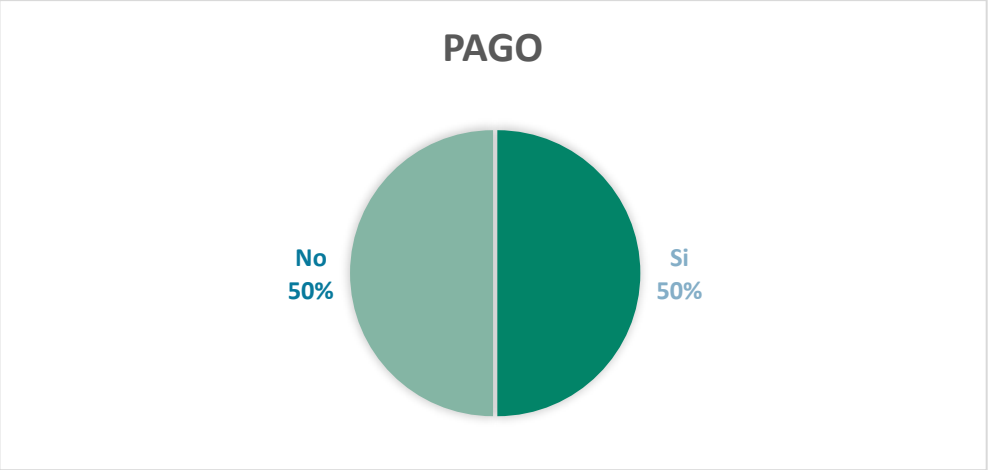
La mayoría de la gente utilizaría la aplicación.




#### *4.1.2.3.- ¿Pagarías por usar la Aplicación? Posibles respuestas*

- Si: 50%
- No: 50%

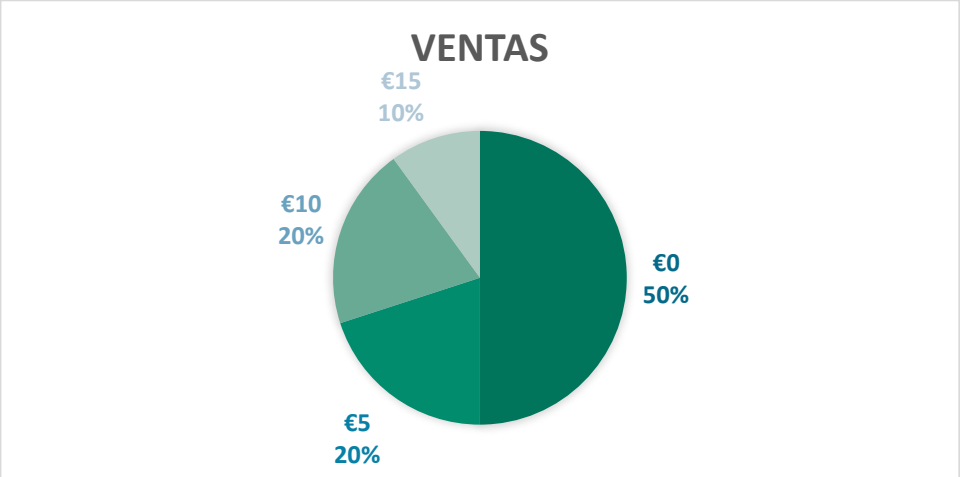
Al analizar los resultados de esta pregunta por separado se interpreta que la mitad estaría dispuesta a realizar un pago único por la aplicación.



### *4.1.2.4.- ¿Cuánto estarías dispuesto a pagar como pago único? Posibles respuestas*

- 0€: 50%
- 5€: 10%
- 10€: 20%
- 15€: 20%

Según las respuestas de los encuestados, y sabiendo en conjunto con la pregunta anterior que la mitad sí que pagaría por la aplicación. Se observa que la cantidad que estarían dispuesta a abonar es de 10€ aproximadamente.




### *4.1.2.5.- ¿Qué te gustaría comentar?*

Se recibieron comentarios como:

*"Debido a como están desarrolladas las tecnologías actualmente, no le veo uso, hace 5 años si"*

*"Yo lo Usaría si hubiese TypeScript"*

*"Yo uso Angular, que no usa JS, a lo mucho lo implementaría como una dependencia"*

*"Es una idea interesante, aunque queda lejos de ser cómodo y práctico. Quizá sería importante añadir algunas funcionalidades y opciones para la usabilidad. También procuraría cuidar el apartado visual. Por lo demás es una herramienta funcional que cumple con lo que promete."*

*"Me parece una herramienta bastante útil, favorece el ahorro de costes en infraestructura. Hay alternativas fuertes en el mercado como puede ser React, pero sin lugar a duda, la idea es interesante."*

*“Aplicación muy útil para ayudarme tanto con el proyecto de clase como con el trabajo diario (Cypress).”*

*“Ojalá hubiese existido antes, lo he pasado mal en Desarrollo web en entorno cliente, en el proyecto me hubiese venido muy bien”*

### 4.1.3.- Conclusión

Teniendo en cuenta todo lo mencionado anteriormente en el estudio de mercado, y haciendo un cruce de datos entre todas ellas podemos deducir que:

- La mayoría de la gente utilizaría la aplicación, dándose la circunstancia de que quienes más uso la darían serían estudiantes.
- Rentabilidad, se espera conseguir unos bajos ingresos ya que solo estarían dispuestos a pagar el 50%, y de estos la cantidad más seleccionada es de 10€ con un 20%.
- De los comentarios podemos sacar como mejoras a futuro implementar otros tipos de lenguajes como TypeScript, jQuery; o considerar la posibilidad de convertirlo en un framework. Como desventaja que ya hay alternativas en el mercado como React.

Conclusión final, con esta aplicación no se podría sacar una rentabilidad económica en un mercado abierto, se debería de orientar a estudiantes o empresas dedicadas a la formación de JavaScript.

# Anexo

# 5.- Anexo

## 5.1.- Conceptos

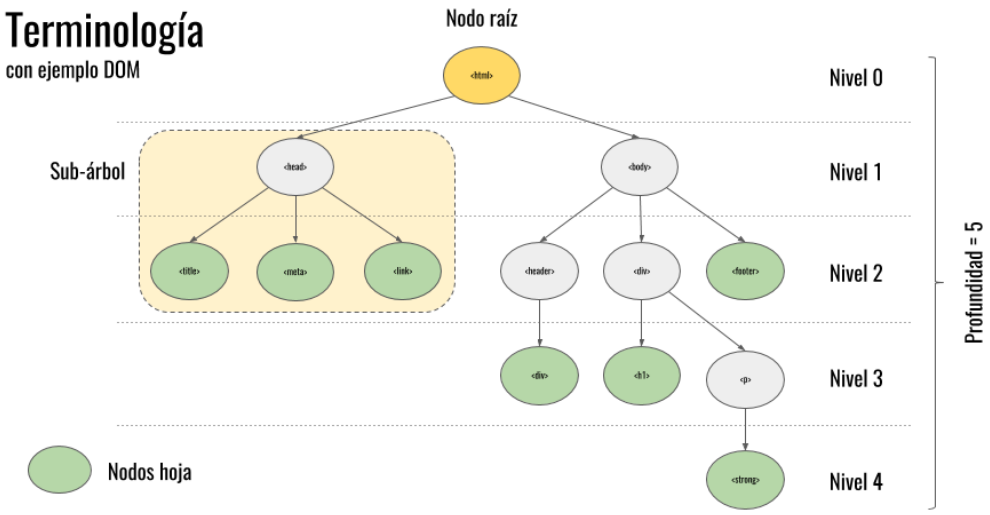
### 5.1.1.- Compilar Código

Es el proceso de traducir el código fuente escrito en un lenguaje de programación especifico a un código ejecutable para ser ejecutado por una maquina o entorno de ejecución.

### 5.1.2.- Generar Dom

Es el acrónimo de *Document Object Model* (Modelo de Objetos del Documento). Se refiere a una representación estructurada y jerárquica de un documento HTML, XHTML o XML. El DOM permite a los programas o scripts acceder y manipular de manera dinámica los elementos, atributos y contenido de un documento web.

El DOM organiza el documento en un árbol de nodos, tales como estos:




### 5.1.3.- LocalStorage

Es un mecanismo que permite almacenamiento persistente en el navegador web. Permite almacenar datos en forma de pares clave-valor de manera sencilla y sin fecha de caducidad.

Sin embargo una cookie es también un almacenamiento del navegador web, pero se diferencia en:

- Solo se pueden almacenar 4KB, mientras que el localStorage almacena hasta MB.
- Las Cookies tienen diferentes opciones de caducidad mientras localStorage no, incluso si cierras el navegador y reinicias el sistema.
- Las cookies se envían automáticamente al servidor con cada solicitud HTTP, mientras que un localStorage no lo envía automáticamente, esto significa que no se transmiten en cada solicitud.
- Las cookies están tanto en el lado del servidor como del lado del cliente, mientras que el localStorage solo en el lado del cliente.

### 5.1.4.- Programación Asíncrona

Este tipo de programación se enfoca en permitir que diferentes partes de una página web se cargué de manera independiente y no bloquear los procesos posteriores.

Al utilizar la programación asincrónica, una página web asincrónica puede cargar y actualizar su contenido de manera eficiente sin necesidad de recargar toda la página.

## 5.2.- Manuales

### 5.2.1.- CodeMirror

#### *5.2.1.1.- Como instalar CodeMirror*

CodeMirror se distribuye como una colección de módulos, para cargar dichas colecciones y que un navegador las pueda interpretar usaremos el administrador de Rollup.

Para instalarlo se usan los siguientes comandos en terminal

```
npm init -y

npm i codemirror @codemirror/lang-javascript

npm i rollup @rollup/plugin-node-resolve
```

Para compilar el código se usa

```
npm start
```

y se configura el Rollup.config.mjs

```
import { nodeResolve } from '@rollup/plugin-node-resolve';
import commonjs from '@rollup/plugin-commonjs';
```

```
import { babel } from '@rollup/plugin-babel';

export default {
  input: './js/editor.mjs', // Archivo de entrada de Rollup
  output: {
    file: './dist/js/editor.bundle.js', // Archivo de salida
    format: 'iife', // Formato de módulo deseado
  },
  plugins: [
    nodeResolve()
  ]
};
```

### *5.2.1.2.- Como darle un formato a un EditorView*

La librería que usamos es @uiw/codemirror-theme-xcode

En la terminal instalar esto

```
npm i @rollup/plugin-commonjs
npm i @rollup/plugin-babel
npm i @uiw/codemirror-theme-xcode
npm i @babel/runtime
```

el archivo rollup.config.mjs

```
import { nodeResolve } from '@rollup/plugin-node-resolve';
import commonjs from '@rollup/plugin-commonjs';
import { babel } from '@rollup/plugin-babel';

export default {
  input: './js/editor.mjs', // Archivo de entrada de Rollup
  output: {
    file: './dist/js/editor.bundle.js', // Archivo de salida
    format: 'iife', // Formato de módulo deseado
    external: ['moment'], // <-- suppresses the warning
    globals: {
      '@babel/runtime/helpers/extends': '_extends' // Nombre global
para el módulo externo
    }
  },
  plugins: [
    nodeResolve(),
    commonjs(),
    babel({
      babelHelpers: 'bundled', // Utiliza los helpers de babel
empaquetados
```

```
      exclude: 'node_modules/**' // Excluye la carpeta node_modules
    })
  ]
};
```

NOTA:

Para que no genere un error con @babel

Hay que poner el CDN en el HTML a mayores

```
<script
src="https://unpkg.com/@babel/runtime@7.15.4/helpers/extends.js"></scri
pt>
```

## 5.2.2.- Firebase

Firebase da servicio tanto de Base de datos como de Hosting incluso de formularios, se eligió ponerlo diferente porque Netlify permite ponerlo en dos diferentes.

### *5.2.2.1.- Crear Proyecto en Firestore Database*

1º Se genera Usuario + Un nuevo proyecto

2º Importante modificar las reglas


En este ejemplo este modo permitir escritura y lectura.

No olvidar dar al botón de Desarrollar y realizar pruebas.

Y en la zona de abajo esta para poder realizar pruebas con el API.

### *5.2.2.2.- Get/Set de Cloud Firestore*

Descripcion General>Configuracion del proyecto

En la zona de abajo esta agregar app, se enlaza con la pagina hosteada y te genera todos los parámetros para configurar

```
const FIREBASE_CONFIG = {
    apiKey: "API_KEY",
    authDomain: "PROJECT_ID.firebaseapp.com",
    projectId: "PROJECT_ID",
    storageBucket: "PROJECT_ID.appspot.com",
    messagingSenderId: "SENDER_ID",
    appId: "APP_ID"
  };
```

Nota: para usar el CDN de Firestore

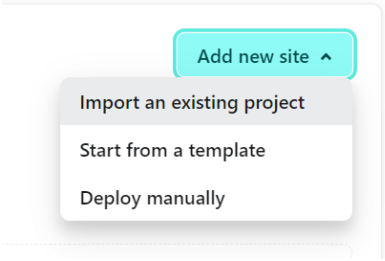
```
<script src="https://www.gstatic.com/firebasejs/8.6.8/firebase-app.js"></script>
<script src="https://www.gstatic.com/firebasejs/8.6.8/firebase-database.js"></script>
```

## 5.2.3.- Netlify

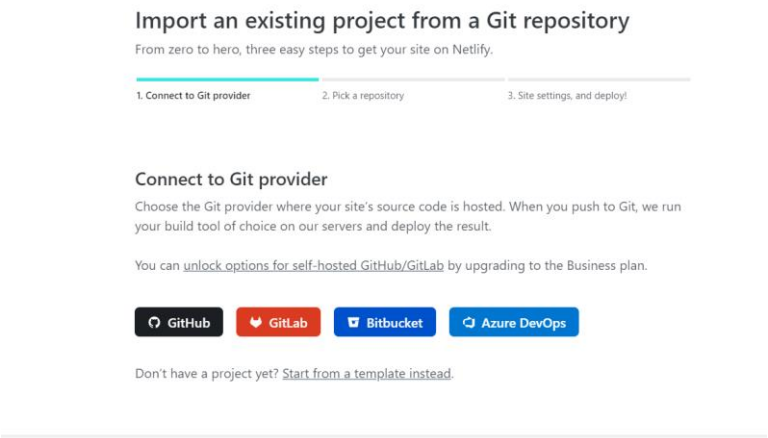
### *5.2.2.1.- Crear Proyecto en Netlify*

En este caso el Proyecto esta generado en GitHub

1º Crea un usuario y generar en Sites>Add new site un nuevo proyecto




2º Conectar a un Git y seleccionamos el repositorio




3º esto es lo importante

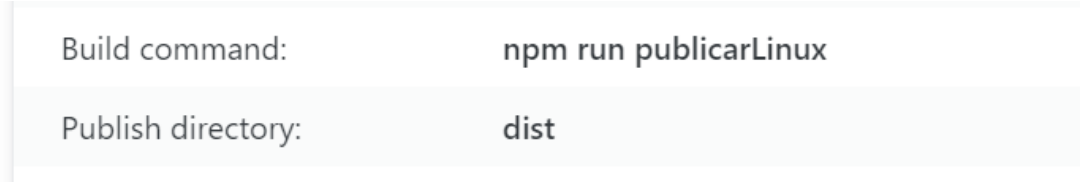
Ojo en:

Branch to deploy → Seleccionamos la Rama que se quiere publicar

Build command → Ponemos el comando para ejecutar, que se halla creado en package.json, hay que tener en cuenta que la mayoría de los servidores están montados en Linux.

Publish directory →El directorio desde donde ejecuta el hosting, una buena práctica es que sea en Dist para producción

Este es la configuración para el proyecto.




## 5.2.4.- Sass

### *5.2.4.1.- Instalación*

Para implementar sass, hay diferentes formas

- puede ser por consola con el comando

```
npm install -g sass
```

Y para compilar usaría

```
sass input.ccss output.css
```

- Pero con VSCode es más sencillo, hay que instalar dos extensiones
  - Live Server
  - Live Sass Compiler

Y para compilar se usa

```
ctrl+s
```

## 5.3.- Webgrafía

CodeMirror 6. (s.f.). *CodeMirror*. Obtenido de https://codemirror.net

*Colores*. (s.f.). Obtenido de Unidades de medida y colores CSS: http://dis.um.es/~lopezquesada/documentos/IES_1213/LMSGI/curso/css/css

11/Paginas/colores.html#:~:text=Los%20colores%20en%20CSS%20se,RGB%20num%C3%A9rico%20y%20RGB%20porcentual.

Daza Rincon, J. (12 de Mayo de 2022). *Funciones, Mixins y SASS!!* Obtenido de Linkedin: https://www.linkedin.com/pulse/funciones-mixins-y-sass-joshua-daza-rincon-?trk=public_profile_article_view

Fierro, E. (7 de Febrero de 2022). *Variables en SASS y SCSS | SASS CSS: Tutorial en español*. Obtenido de Eduardo Fierro: https://github.com/eduardofierropro/SASS-y-SCSS-Aumenta-tu-nivel-como-Frontend

Firebase. (14 de 11 de 2022). *Cloud Firestore*. Obtenido de Firebase: https://firebase.google.com/docs/firestore?hl=es-419

Kantor, I. (15 de Noviembre de 2022). *LocalStorage, sessionStorage*. Obtenido de Javascript.Info: https://es.javascript.info/localstorage

Manz. (s.f.). *¿Qué son los objetos?* Obtenido de Estructuras de datos tipo diccionario: https://lenguajejs.com/javascript/objetos/que-son/

MDN contributors. (21 de febrero de 2023). *Window.localStorage*. Obtenido de MDN: https://developer.mozilla.org/es/docs/Web/API/Window/localStorage

Montero, L. (16 de 04 de 2017). *Árboles (trees)*. Obtenido de Medium: https://medium.com/laboratoria-developers/%C3%A1rboles-trees-51783ba4ebe5

Rincon, W. (31 de diciembre de 2020). *Tutorial de Arreglos de Objetos en JavaScript - Cómo crear, actualizar y hacer un ciclo a través de los objetos usando los métodos de arreglos en JS*. Obtenido de freeCodeCamp: https://www.freecodecamp.org/espanol/news/arreglos-de-objetos-en-javascript-actualizar/

Segundo, G. A. (17 de septiembre de 2022). *Cómo Hacer Un Editor De Código Con Codemirror 6*. Obtenido de morioh: https://morioh.com/p/6f26d21b74b5