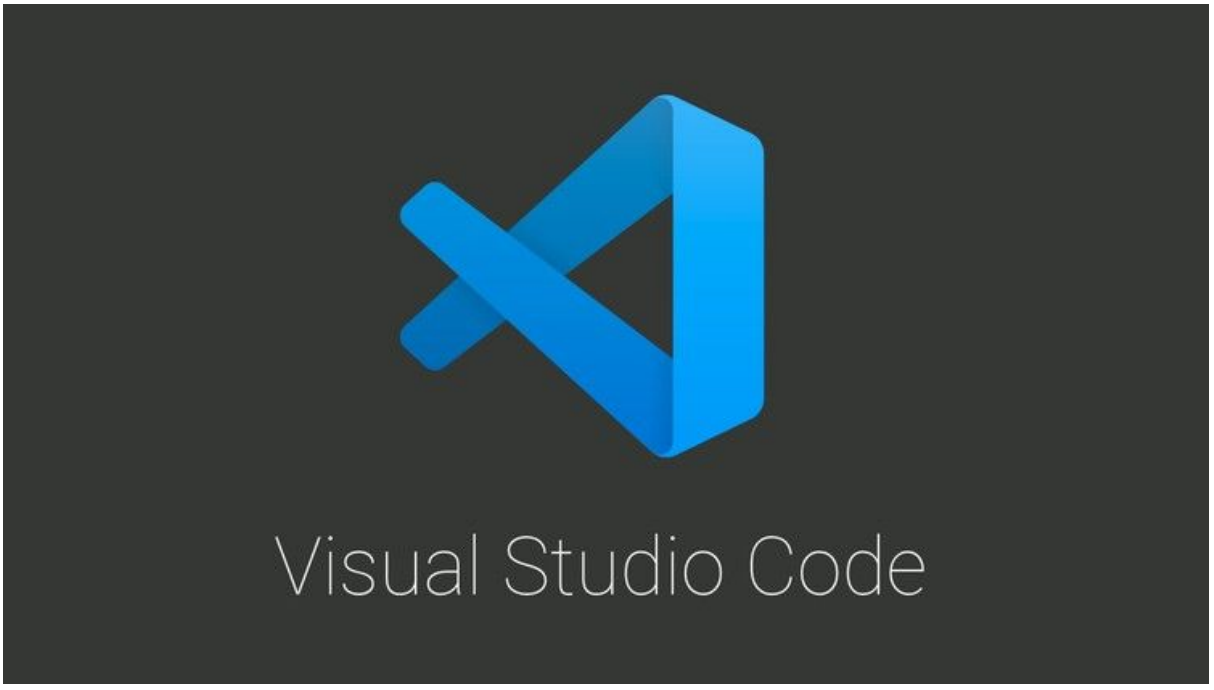
Manual Técnico

Entorno de desarrollo

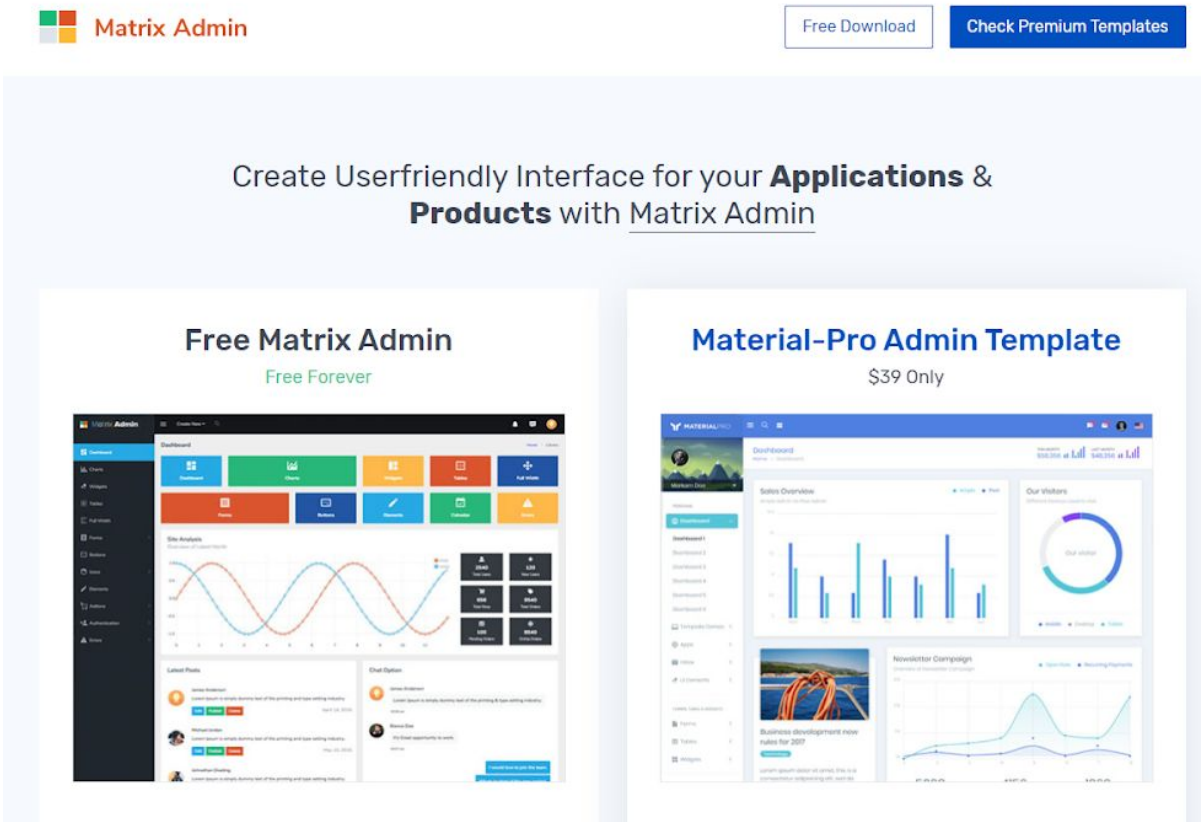
Lenguaje de desarrollo: javascript

Ide utilizado: visual studio code



Diseño Web

Para el desarrollo del diseño de la página web se hizo uso de una plantilla html que es la siguiente y se encuentra en el sitio
https://www.matrixadmin.wrappixel.com/



Se realizó la edición removiendo muchas páginas y componentes los cuales no servirán para nuestro sitio. en el home se dejan los divs los cuales generarán más orden en nuestro sitio, los componentes utilizados son los siguientes.
La barra lateral izquierda contendra las opciones de Archivo junto con el área de reportes para ello se hace uso de las siguientes etiquetas.

```
<!--sidebar-menu-->

<div id="sidebar"> <a href="#" class="visible-phone"><i class="icon icon-fullscreen"></i>Full width</a>
  <ul>
    <li class="active"><a href="index.html"><i class="icon icon-home"></i> <span>Dashboard</span></a> </li>
    <li class="submenu"> <a href="#"><i class="icon icon-list"></i> <span>Archivo</span> <span class="label label-important">3</span></a>
      <ul>
        <li><a>Open</a></li>
        <li><a>Save</a></li>
        <li><a>Save As</a></li>
      </ul>
    </li>
    <li class="submenu"> <a href="#"><i class="icon icon-file"></i> <span>Reportes</span> <span class="label label-important">5</span></a>
      <ul>
        <li><a type="button" onclick="javascript:create_report_html();">Html</a></li>
        <li><a type="button" onclick="javascript:create_report_json();">Json</a></li>
        <li><a type="button" onclick="javascript:create_report_errores();">Errores</a></li>
        <li><a type="button" onclick="javascript:create_report_Translation();">Python</a></li>
      </ul>
    </li>
  </ul>
</div>
```

para el área en el cual podemos añadir pestañas con cuadros de texto para añadir más pestañas se hace uso de un div con un estilo widget-box el cual tendra un titlo que contiene el nombre de está área luego tenemos un div al cual si le damos un identificador el cual es reqs ya que este va a ser al cual se le agregaran más cuadros de texto con sus respectivos botones de analize download y save.

```
<!--Texts Box Analize-->
<div class="widget-box">
  <div class="widget-title"> <span class="icon"> <i class="icon-list"></i> </span>
    <h5>Full Width</h5>
  </div>
  <div id="reqs" class="widget-content">
  </div>
</div>
<!--End Text Box Analize-->
```

luego tenemos una sección en el cual tendremos un div que contiene un título Python indicando que está será el área donde será añadida la traducción generada contiene también un textbox que es el área donde será agregada la traducción ya realizada al código c# indicado, para las traducciones del html json es la misma implementación solo cambia el nombre del textbox para poder indicar a qué cuadro de texto será enviada las respectivas traducciones.

```
<!--Texts Box Python-->
<div class="widget-box">
  <div class="widget-title"> <span class="icon"> <i class="icon-list"></i> </span>
    <h5>Python</h5>
  </div>
  <div class="widget-content">
    <textarea class="w3-input w3-border" id="python" value="1" style="width:97%" readonly ></textarea>
  </div>
</div>
<!--End Text Box Python-->

<!--Texts Box Html-->
<div class="widget-box">
  <div class="widget-title"> <span class="icon"> <i class="icon-list"></i> </span>
    <h5>Html</h5>
  </div>
  <div class="widget-content">
    <textarea class="w3-input w3-border" id="html" value="1" style="width:97%" readonly ></textarea>
  </div>
</div>
<!--End Text Box Html-->

<!--Texts Box Json-->
<div class="widget-box">
  <div class="widget-title"> <span class="icon"> <i class="icon-list"></i> </span>
    <h5>Json</h5>
  </div>
  <div class="widget-content">
    <textarea class="w3-input w3-border" id="json" value="1" style="width:97%" readonly ></textarea>
  </div>
</div>
<!--End Text Box Json-->
```

Luego definimos una tabla la cual será donde agregaremos los tokens que fueron reconocido durante el análisis al código c# para esto tenemos la etiqueta de table para crear la tabla la única variante es que al body de la table se le asigna un id para poder hacer inner de nuevas filas para la tabla.

```
<!--Table Tokens-->
<div class="widget-box">
  <div class="widget-title"> <span class="icon"> <i class="icon-th"></i> </span>
    <h5>Tokens Table</h5>
  </div>
  <div class="widget-content nopadding">
    <table class="table table-bordered table-striped">
      <thead>
        <tr>
          <th>Lexeme</th>
          <th>Column</th>
          <th>row(s)</th>
        </tr>
      </thead>
      <tbody id="content_table">
      </tbody>
    </table>
  </div>
</div>
<!--End Table Tokens-->
```

Para poder abrir un archivo tenemos una última sección en el sitio la cual contiene un inpute de type file el cual nos indicara que archivo quiere cargar el usuario.

```
<!--Load File-->
<div class="widget-box">
  <div class="widget-title"> <span class="icon"> <i class="icon-list"></i> </span>
    <h5>Select File to Load</h5>
  </div>
  <div class="widget-content">
    <input type="file" id="fileInput"/>
  </div>
</div>
<!--End Load FIle-->
```

En la última sección hacemos el llamado de todos los scripts que haremos uso en nuestro sitio el cual nos servira para darle funcionalidad al mismo, cabe resaltar que solo un archivo javascript contiene toda la lógica de análisis y abrir analizar archivos.el cual tiene por nombre analizador.

```
<script src="js/analizador.js"></script>
<script src="js/jquery.min.js"></script>
<script src="js/jquery.ui.custom.js"></script>
<script src="js/bootstrap.min.js"></script>
<script src="js/matrix.js"></script>
</body>
</html>
```

Javascript con la lógica para el análisis, y opciones de Archivo

Como bien mencionamos anteriormente este archivo se llama analizador.js el cual contiene lo siguiente:

Abrir, Crear nueva pestaña.

para esto se hace uso del id del div el cual contendría a todos los textbox los cuales representan a una nueva pestaña en la cual se puede escribir el código c#, esto se realiza por medio del inner html el cual incrusta etiquetas html en ejecución del sitio. para ello agregamos primero un textarea le damos tamaño le colocamos un identificador que es reqt + un número incremental el cual aumentará cada vez que se quiera agregar una nueva pestaña luego tenemos 3 botones uno de analize el cual obtiene el texto del textbox anteriormente creado y manda a llamar a nuestra función lexer analize. luego tenemos el botón de descargar el cual obtiene todo el código c# del textbox definido primeramente y luego manda a llamar al método download el cual es el encargado de generar la descarga con el código c#. Por último tenemos el botón remove que lo que hace es eliminar todos los elementos que fueron definidos anteriormente, también eliminando a sí mismo.
Por último agregamos al div los elementos anteriormente mencionados con la etiqueta appendchild.

```
//create textbox
var input = document.createElement('textarea');
input.type = "text";
input.setAttribute("class", "w3-input w3-border");
input.setAttribute('id', 'reqt' + reqs_id);
input.setAttribute('value', reqs_id);
input.setAttribute('style', "width:97%");
var reqs = document.getElementById("reqs");

//create analize button
var analize = document.createElement('button');
analize.setAttribute('id', 'reqsa' + reqs_id);
analize.setAttribute("type", "button");
analize.setAttribute("class", "btn btn-info");
analize.setAttribute('style', "width:32%");
analize.onclick = function() {
  var texto = document.getElementById('reqt' + reqs_id).value;
  lexer_analize(texto);
};
analize.innerHTML = "analize";
```

```
//create download
var descargar = document.createElement('button');
descargar.setAttribute('id', 'reqsd' + reqs_id);
descargar.setAttribute("type", "button");
descargar.setAttribute("class", "btn btn-success");
descargar.setAttribute('style', "width:32%");
descargar.onclick = function() {
  var texto = document.getElementById('reqt' + reqs_id).value;
  download("Entrada"+ reqs_id +".txt", texto);
};
descargar.innerHTML = "Download";

//create remove button
var remove = document.createElement('button');
remove.setAttribute('id', 'reqsr' + reqs_id);
remove.setAttribute("type", "button");
remove.setAttribute("class", "btn btn-danger");
remove.setAttribute('style', "width:33%");
remove.onclick = function() {
reqs.removeChild(input);
reqs.removeChild(analize);
reqs.removeChild(descargar);
reqs.removeChild(remove);
reqs_id--;
};
```

```
63
64    //append elements
65    reqs.appendChild(input);
66    reqs.appendChild(analize);
67    reqs.appendChild(descargar);
68    reqs.appendChild(remove);
69    return 'reqt' + reqs_id;
70  }
```

Luego tenemos la función la cual hace uso del fileinput que se definió en el front end de la página, este luego de abrir el archivo se procede a leer con filereader luego de obtener todo el texto del documento se procede a agregar una nueva pestaña y agregar el texto obtenido

```
}

document.getElementById('fileInput').addEventListener('change', function selectedFileChanged() {
    if (this.files.length === 0) {
      console.log('No file selected.');
      return;
    }

    const reader = new FileReader();
    reader.onload = function fileReadCompleted() {
      // when the reader is done, the content is in reader.result.
      //console.log(reader.result);
      var text_box = add();
      document.getElementById(text_box).value = reader.result;
    };
    reader.readAsText(this.files[0]);
  });
```

luego tenemos las dos siguientes funciones que son para crear una descarga con el texto en el cuadro de diálogo que corresponde al botón seleccionado.

tenemos la función que agrega una fila a la tabla de tokens definida en el front end del sitio.

```
  function download(filename, text) {
    var element = document.createElement('a');
    element.setAttribute('href', 'data:text/plain;charset=utf-8,' + encodeURIComponent(text));
    element.setAttribute('download', filename);

    element.style.display = 'none';
    document.body.appendChild(element);

    element.click();

    document.body.removeChild(element);
  }

function add_to_table(lexeme, column , row){

    var fila="<tr><td>"+lexeme+"</td><td>"+column+"</td><td>"+row+"</td></tr>";

    var btn = document.createElement("TR");
    btn.innerHTML=fila;
    document.getElementById("content_table").appendChild(btn);
}
```

Luego tenemos un área en el cual declaramos un conjunto de variables globales auxiliares para el análisis de léxico y sintáctico.una de las cuales son arreglos de objetos para almacenar a objetos de tipo error, u variables que sirven de control de acceso o simplemente arreglos que sirven de auxiliares para poder verificar si un lexema es una palabra reservada.

```
var errores_lexicos = [];
var errores_sintacticos = [];
var list = [];
var tabs = [];
var type_data = ["int", "double", "char", "bool", "string"];
var traduccion = "";
var aux_traduccion = "";
var panic_mode_var = false;
var method_main = false;
var method_s = false;
var method_main_f = false;
var oper_relationals = ["<", ">", "<=", ">=", "==", "!="];
var traduccion_without_do = "";
var bucle = [];
var is_method = false;
var html_code = "";
var etiquetas_html = [];
var json_string = "";
```

Luego tenemos la función lexer_analize el cual contiene toda la lógica para el análisis léxico, el cual va a generar tokens y guardarlos en un arreglo llamado lista junto a darles valores iniciales a las variables definidas anteriormente, luego de haber creado objetos de tipo tokens se agregan a la tabla la cual contendrá todos los tokens reconocidos durante el análisis .

```
// analisis lexico
function lexer_analize(entry)
{
    var row = 0;
    var column = 0;
    var size_entry = entry.length;
    var letters = "azAZ";
    var num_a = letters.charCodeAt(0);
    var num_z = letters.charCodeAt(1);
    var num_A = letters.charCodeAt(2);
    var num_Z = letters.charCodeAt(3);
    var bool_id = false;
    var numero = false;
    var commentary;
    method_main_f = false;;
```

Luego de haber realizado el análisis y haber guardado los tokens se llama al método parser el cual contiene las definiciones de cómo pueden venir los tokens dados por el análisis

léxico para luego de termina dicho análisis sintáctico agregar el resultado a nuestro text box python con consiguiente también pasar a realizar el análisis del html encontrado en el código c#

```
//analisis sintactico
function parser()
{
    //select method
    //verificar asignacion o funcion
    for(var d = 0 ; d < list.length ; d++)
    {
        if(verificate_type(list[0].lexeme) && panic_mode_var == false)
        {
            if(list[2].lexeme == "=" || list[2].lexeme == ";" || list[2].lexeme == ",")
            {
                analice_declaration();
                if(panic_mode_var)
                {
                    aux_traduccion = "";
                    continue;
                }
                //print_word(aux_traduccion);
                for(var i = 0 ; i < tabs.length ; i++)
                {
                    aux_traduccion = add_tabs(aux_traduccion);
                }
                concat_traduccion(aux_traduccion);
                aux_traduccion = "";
                document.getElementById("traducido").value = traduccion;
            }else if(list[2].lexeme == "(" && method_s == false)
            {
                //Se define una funcion
```

cabe aclarar que se hace bastante uso de la recursividad ya que los métodos definidos para poder traducir sentencia y traducción son llamados cada vez que se encuentra el inicio de cada función, esto en los tokens proporcionados del análisis léxico.

```
//definir while
function sent_while()
{

    list.shift();
    if(list[0].lexeme == "(")
    {
        list.shift();
        // definir condiciones
        concat_traduccion_aux("\r\nwhile ");
        verificate_conditions();
    }else
    {
        errores_sintacticos.push({lexeme: list[0].lexeme, row_l: list[0].row_l, column_l: list[0].column_l, descripcion: "Se esperaba un '(' se encontro:" + list[0].lexema});
        panic_mode_var = true;
        return;
    }
}

//console write
function console_write()
{
    list.shift();
    if(list[0].lexeme != "(")
    {
        errores_sintacticos.push({lexeme: list[0].lexeme, row_l: list[0].row_l, column_l: list[0].column_l, descripcion: "Se esperaba un '(' se encontro:" + list[0].lexema});
        panic_mode_var = true;
            return;
    }else
    {
        list.shift();

        var description = "\r\nprint(";
```