



MANUAL DE USUARIO

PROYECTO 2 TRANSPILADOR - MANUAL DE USUARIO





INTRODUCCION

Este es un manual de usuario que muestra como entender la pagina web de Transpilador USAC, mostrando como usarla y las distintas funciones que tiene.

PAGINA INICIAL

Al iniciar el proyecto se muestra la pagina principal donde se encuentran varias opciones.

Pensum USAC

[Home](#)[Reporte Error](#)[Archivos](#)[Manuales](#)

Editor de texto en C#

Analizar

Limpiar Editor

Salida Traducida a TypeScript

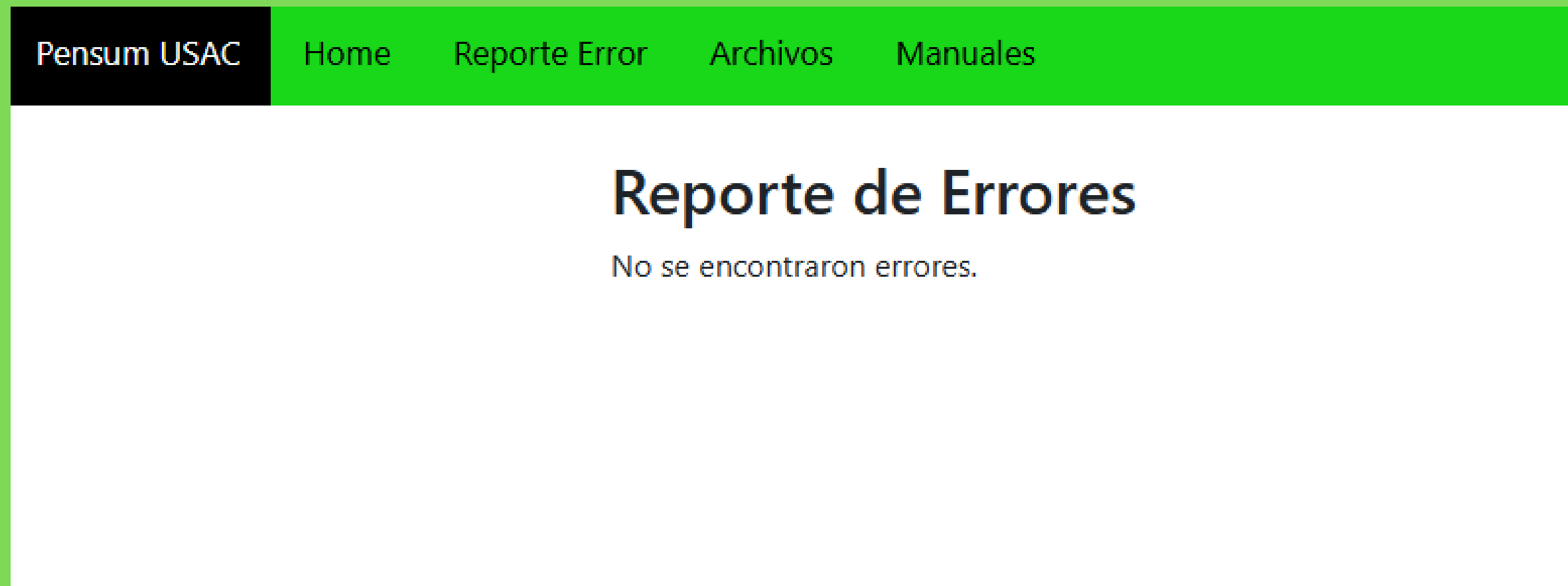
Tabla de Tokens

#	Fila	Columna	Lexema	Token
---	------	---------	--------	-------

Lo que esta mas a la vista son cosas como el menu de paginas que esta hasta arriba, luego un area para escribir codigo de C#, un area donde aparece el codigo de TypeScript traducido y dos botones uno de analizar y otro de limpiar editor.

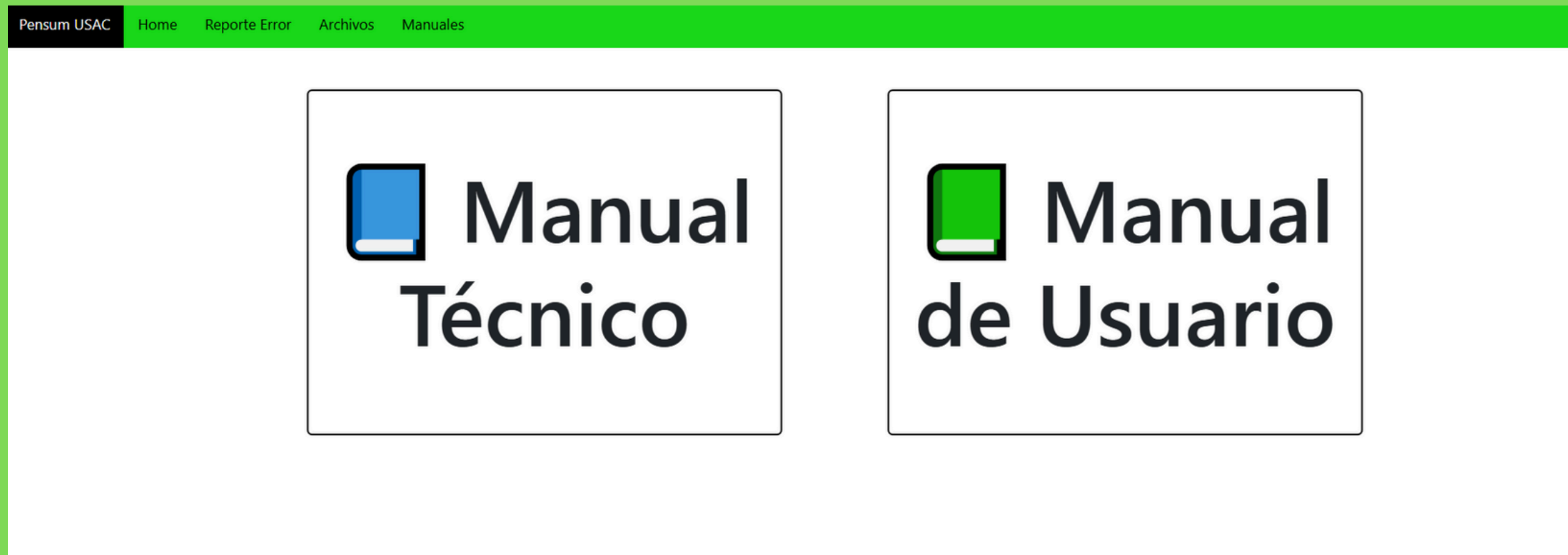
REPORTE ERROR

Al entrar a la pagina de Reporte de Error, nos mostrara una advertencia de que no hay reportes a menos que detecte errores en el editro de codigo.



MANUALES

Al entrar a la pagina de manuales hay dos opciones, una para descargar el manual tecnico y otro para descargar el manual de usuario.



ARCHIVOS

Al darle a la opcion de archivos nos abrira un navegador de archivos donde podemos buscar archivos con extencion .cs y usarlas en nuestro editor de texto.

Pensum USACHomeReporte ErrorArchivosManuales

Editor de texto en C#

Salida Traducida a Typo

Abrir

<< Escritorio >> USAC Cursos > Tercer Semestre > VACACIONES > Laboratorio de LFP >

Buscar en Laboratorio de LFP

Organizar Nueva carpeta

Matematica Inte ^

Proyecto 1

OneDrive

OneDrive - Persor

Este equipo

Descargas

Documentos

Escritorio

Imágenes

Música

Objetos 3D

Videos

Disco local (C:)

Red

Nombre ^

Fecha de modificación

Tipo

Tamaño

Elian GOAT

29/06/2025 21:58

Carpeta de archivos

Practica 1

13/06/2025 13:10

Carpeta de archivos

Programacion

29/06/2025 22:21

Carpeta de archivos

Proyecto 1

17/06/2025 15:56

Carpeta de archivos

Archivo1.cs

29/06/2025 23:33

Archivo de origen ...

2 KB

Archivo2.cs

29/06/2025 23:33

Archivo de origen ...

2 KB

Archivo3.cs

29/06/2025 23:33

Archivo de origen ...

3 KB

Archivo4.cs

29/06/2025 23:33

Archivo de origen ...

1 KB

Archivo5.cs

29/06/2025 23:33

Archivo de origen ...

8 KB

Archivo6.cs

29/06/2025 23:33

Archivo de origen ...

8 KB

Archivo7.cs

29/06/2025 23:33

Archivo de origen ...

8 KB

Nombre de archivo:

Archivo CS (*.cs)

Abrir

Cancelar

Tabla de Tokens

#	Fila	Columna	Lexema
---	------	---------	--------

EDITOR DE TEXTO

En la parte de editor de texto es en donde podemos colocar nuestro código en C#. Siguiendo siempre la estructura de ingresos de datos, acá también puede rellenarse automáticamente al abrir un archivo de extensión .cs.

Editor de texto en C#

Analizar

Limpiar Editor

ANALIZAR

Al presionar el boton de analizar cuando ya tenemos texto nos mostrara al lado derecho una salida de texto donde esta el codigo que enviamos pero traducido a TypeScript, una tabla de tokens con todos los tokens que encuentra en la entrada de texto de nuestro editor de texto detallandolos con su columan, fila, lexema y tipo de token.

Pensum USAC

[Home](#)[Reporte Error](#)[Archivos](#)[Manuales](#)

Editor de texto en C#

```
using System;

public class LFP_VJ2025 {

    static void Main(string[] args) {
        int edad;
        float altura = 1.72;
        string nombre = "Calificador1";
        bool mayor_edad = false;
        char genero;

        edad = 17;
        genero = 'H';

        Console.WriteLine("Bienvenido: " + nombre + " su altura es de: " + altura + " metros y tiene: " + edad + " años");
    }
}
```

Analizar

Limpiar Editor

Salida Traducida a TypeScript

```
let edad: number;
let altura: number = 1.72;
let nombre: string = "Calificador1";
let mayor_edad: boolean = false;
let genero: string;
edad = 17;
genero = 'H';
console.log("Bienvenido: " + nombre + " su altura es de: " + altura + " metros y tiene: " + edad + " años");
let nota_1: number, nota_2: number = 61, nota_3: number = 80, nota_4: number;
let nombre_1: string = "Cali2", nombre_2: string, nombre_3: string;
let isTree: boolean, isApple: boolean = false, isMountain: boolean;
let inicial: string = 'A', final: string;
let sueldo_base: number = 1250.25, sueldo_netto: number, igss: number, cobro_iva: number = 12.0;
```

Tabla de Tokens

#	Fila	Columna	Lexema	Token
1	1	1	using	30
2	1	7	System	31
3	1	13	;	7

LIMPIAR EDITOR

La opcion de limpiar editor lo que hace es limpiar tanto el editor de texto como la tabla de tokens, la salida de codigo y asi pueda usarse nuevamente.

Pensum USAC

[Home](#)[Reporte Error](#)[Archivos](#)[Manuales](#)

Editor de texto en C#

Analizar

Limpiar Editor

Salida Traducida a TypeScript

Tabla de Tokens

#	Fila	Columna	Lexema	Token
---	------	---------	--------	-------

ERRORES

Al detectar un error en el editor de texto este mostrara un reporte de que detecto errores y te llevara a la seccion de reporte error.

Editor de texto en C#

```
using System;

public class LFP_VJ2025 {

    static void Main(string[] args) {
        int a, b; @
        a = 12;
        string hola = "hola";

        if(13 > 12){ &
            int edad = 23;
            Console.WriteLine("Mi edad es:
        } else {
            edad = 20;
            string perro = "Lucho";
```

Analizar

Limpiar Editor

Salida Traducida a TypeScript

```
let a: number, b: number;
a = 12;
let hola: string = "hola";
if (13 > 12) {
    let edad: number = 23;
    console.log("Mi edad es: " + edad);

    = "Lucho";
    ad es: " + edad);
    n;
    = 'h';

    = 12.5;
    ;
```



Errores detectados

Se detectaron errores léxicos. Revisa el reporte.

Entendido

Tabla de Tokens

#	Fila	Columna	Lexema	Token
1	1	1	using	30
2	1	7	System	31

REPORTE ERROR

Ahora que el sistema si detecto un o varios errores entonces mostrara una lista con los errores que contiene el editor de texto al momento de analizarlo ya sean errores lexicos o sintacticos.

Pensum USAC

[Home](#)[Reporte Error](#)[Archivos](#)[Manuales](#)

Reporte de Errores

Errores Léxicos

No.	Fila	Columna	Lexema	Token
1	6	19	@	0
2	10	22	&	0
3	39	11	\$	0
4	39	13	%	0
5	47	11	?	0
6	72	22	#	0
7	72	23	\$	0

TRANSPILADOR

Al darle al boton de analizar, muestra como se tradujo el codigo de C# a TypeScript y tambien en la parte de abajo muestra la tabla de tokens que se detectan en el editor de texto en C#

Editor de texto en C#

```
using System;

public class LFP_VJ2025 {

    static void Main(string[] args) {

        for (int i = 0; i < 10; i++) {
            Console.WriteLine("i = " + i);
        }

        for (int i = 10; i >= 0; i--) {
            Console.WriteLine("i = " + i);
        }

        int k;
        int l;
```

Analizar

Limpiar Editor

Salida Traducida a TypeScript

```
for (let i: number = 0; i < 10; i++) {
    console.log("i = " + i);
}
for (let i: number = 10; i >= 0; i--) {
    console.log("i = " + i);
}
let k: number;
let l: number;
for (let i: number = 0; i < 10; i++) {
    for (l = 10; l >= 0; l--) {
        for (let j: number = 0; j < 5; j++) {
            for (k = 5; k > 0; k--) {
                console.log(i * j * k * l);
            }
            console.log(i * j * l);
        }
    }
}
```

TABLA DE TOKENS

Tabla de Tokens

#	Fila	Columna	Lexema	Token
1	1	1	using	30
2	1	7	System	31
3	1	13	;	7
4	3	1	public	32
5	3	8	class	33
6	3	14	LFP_VJ2025	23
7	3	25	{	1
8	5	5	static	34
9	5	12	void	35
10	5	17	Main	36

CONCLUSIÓN

- Esas son todas las funciones que tiene la pagina web diseñada para el usuario y su uso de ayuda para pasar de codigo de C# a TypeScript.
- Muestra tambien el funcionamiento de como se distribuyen los tokens de los textos agregados para mayor comprension.
- Muestra tambien el funcionamiento del analizador sintactico.

Nombre: Alberto Moisés Gerardo Lémus Alvarado

Carne: 202400999

