

# PORTAFOLIO DE EVIDENCIAS

AGOSTO -- DICIEMBRE

ESTRUCTURA DE DATOS

DOCENTE: ARTURO BUSTAMANTE  
LAZCANO

ALUMNO: JOSE DANI CALIXTRO DE JESUS

TERCER SEMESTRE



## Contenido

Evaluación Diagnóstica .....	2
Semana 1-.....	3
EN LA PRIMERA SEMANA NOS REGISTRAMOS AL CURSO DE HENRY .....	3
INSTALACIÓN DE HERRAMIENTAS .....	4
VISUAL STUDIO CODE .....	5
GIT .....	6
CUENTA DE GIT HUB.....	7
INSTALACIÓN DE HERRAMIENTAS COMPLETA .....	8
SEMANA 2 .....	9
USO DE COMANDOS BASICOS.....	9
Creación de repositorio y primer commit.....	10
PRIMEROS ESTILOS CON CSS .....	11
JUEGOS DEL CURSO .....	12
SEMANA 3 .....	13
TIENDA EN LINEA.....	13
Semana 4.....	15
HOMEWORKS CSS .....	15
SEMANA 5 .....	17
SEMANA 6 .....	20
CONTROL DE FLUJO (IF/ELSE) .....	20
EJERCICIOS.....	21

# Evaluación Diagnóstica

## Semana 1-

EN LA PRIMERA SEMANA NOS REGISTRAMOS AL CURSO DE HENRY

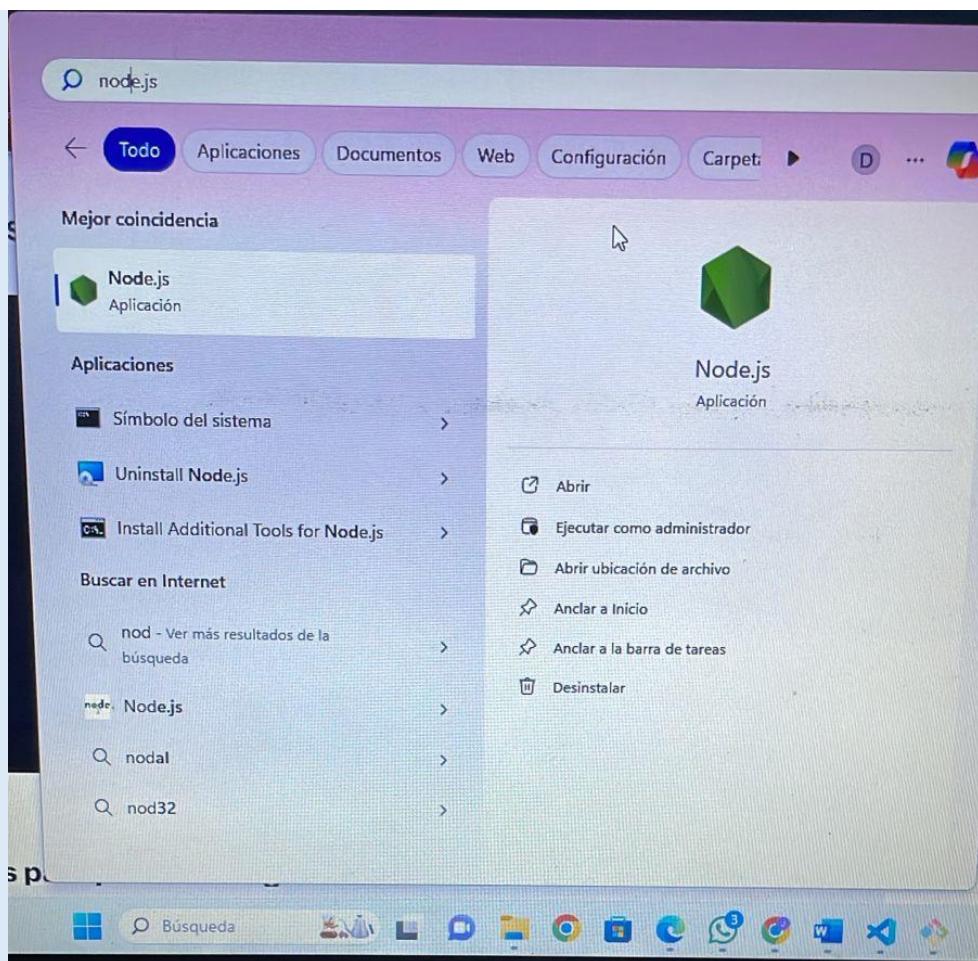
The image shows a screenshot of the Henry login page. On the left, there is a form for logging in with fields for 'Correo' and 'Contraseña', a 'Recuérdame' checkbox, and a 'Ingresar' button. Below the form, there is a link for new users: '¿No tienes una cuenta? [Regístrate](#)'. On the right, there is a large yellow promotional overlay. The overlay features the Davivienda logo (a house icon) and the text 'Súmate a grandes empresas como **Davivienda** y **MercadoLibre**'. Below this, there is a MercadoLibre logo (two hands shaking) and the text 'mercado libre'. A curved arrow points from the top right towards the Davivienda logo.

## INSTALACIÓN DE HERRAMIENTAS

- NODE JS

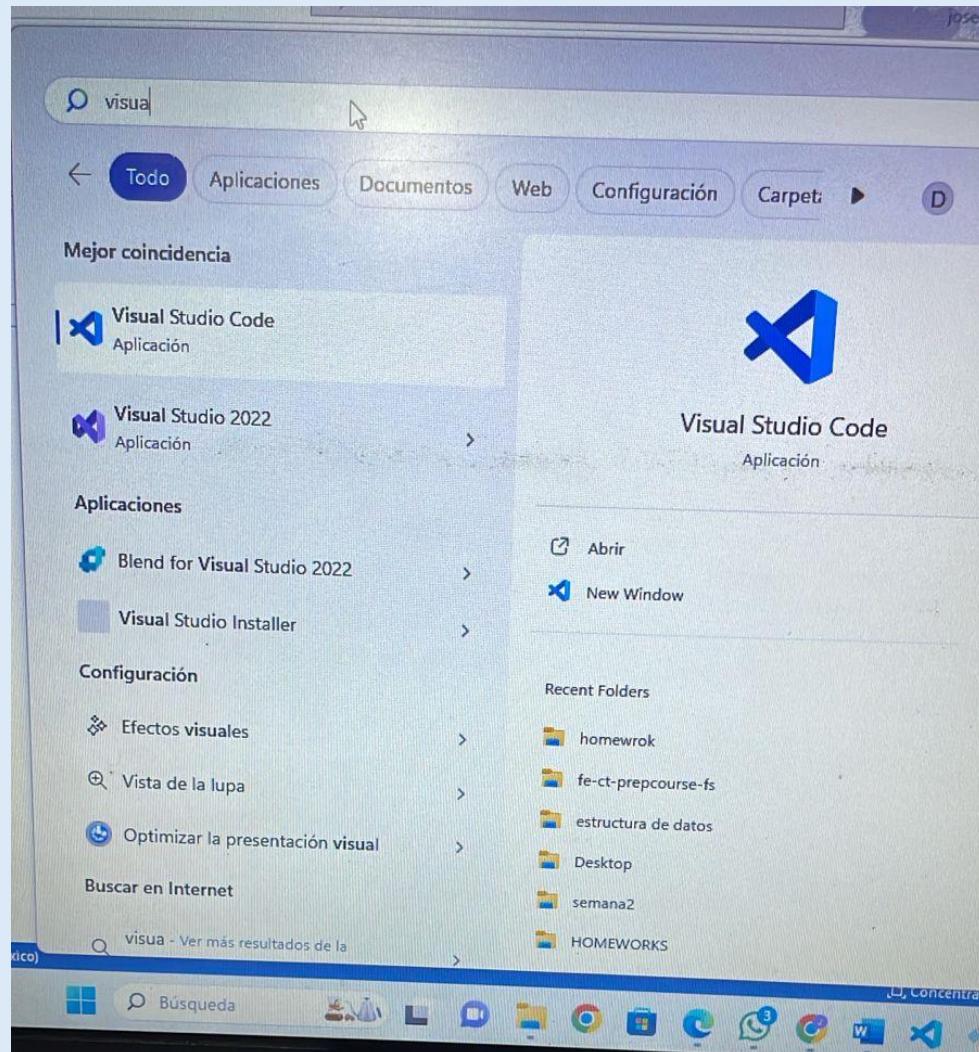
### ¿Qué es Node JS?

Fue ideado como un **entorno de ejecución de JavaScript**, Node JS nos va a permitir ejecutar nuestro código, lo leerá y si algo anda mal, nos retornará lo que debemos corregir o en caso contrario, nos brindará el resultado de la ejecución de nuestro código JS.



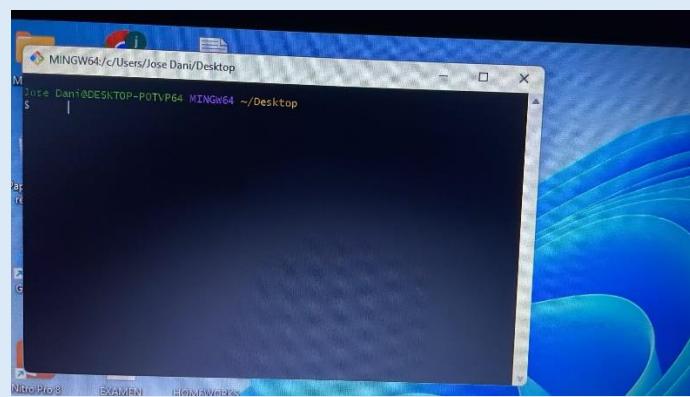
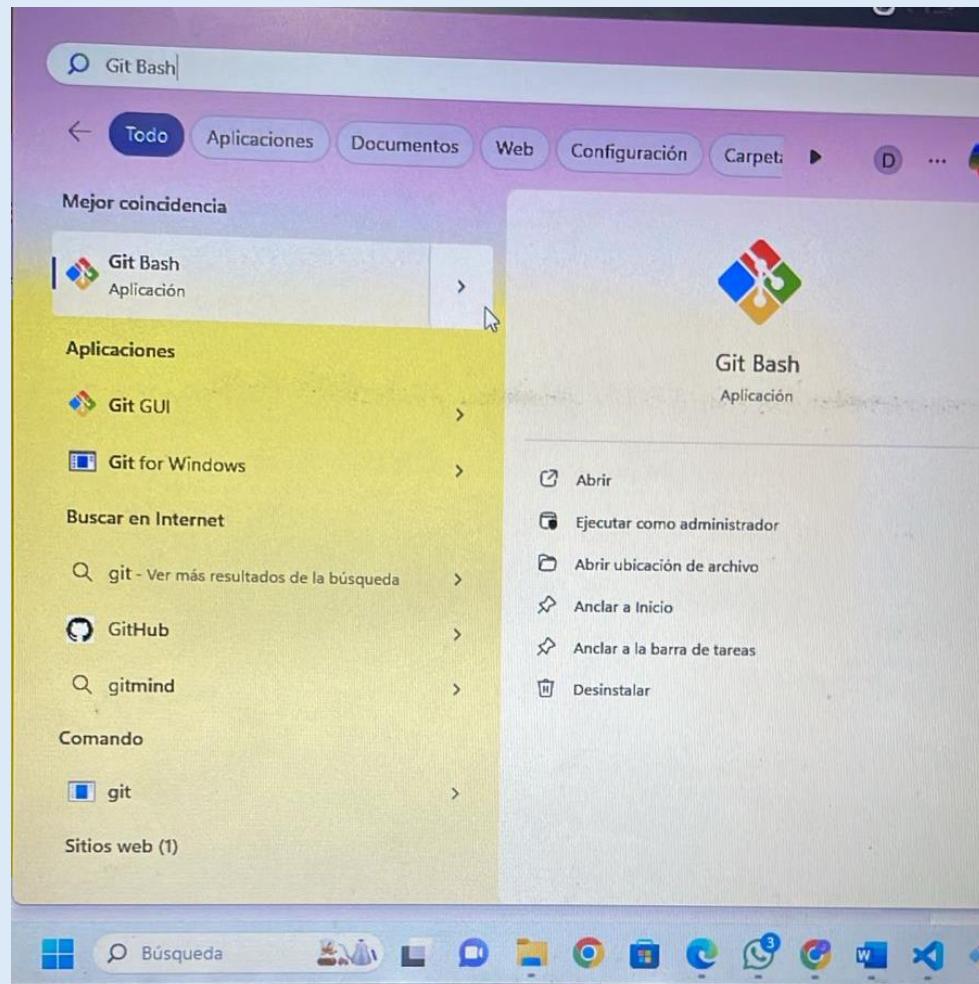
## VISUAL STUDIO CODE

Es un editor desarrollado por Microsoft. Tiene integrado el control de versiones mediante Git y GitHub para tener un seguimiento de tus proyectos. Brinda una cantidad de extensiones que facilitan el trabajo de un desarrollador.

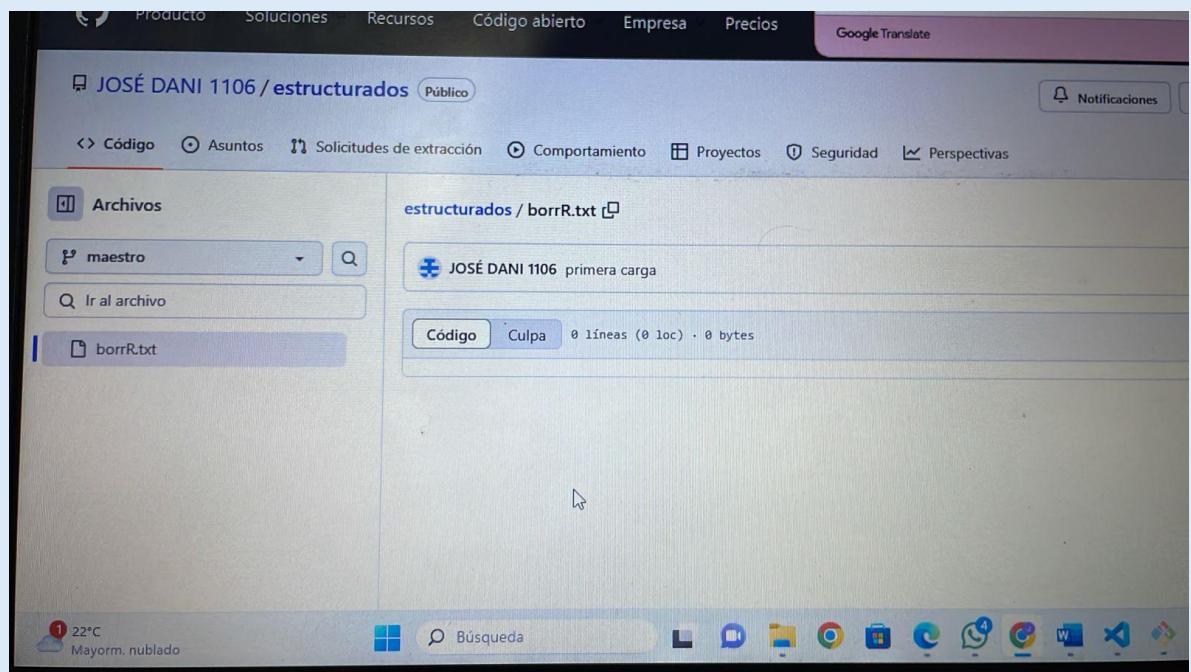


## GIT

**Git** es un sistema de **control de versiones de código**, es decir; un sistema que registra los cambios realizados en un archivo o conjunto de archivos a lo largo del tiempo, de modo que puedas recuperar versiones específicas más adelante.



## CUENTA DE GIT HUB



## INSTALACIÓN DE HERRAMIENTAS COMPLETA

The screenshot shows a computer monitor displaying a web-based course management system. At the top, there's a header with a profile icon for "JOSE DANI CALIXTRO DE..." and some other icons. Below the header, a yellow banner titled "Homework" contains the text: "El objetivo de esta primera homework, es verificar que tengas instaladas las herramientas/aplicaciones necesarias para avanzar con el Prep Course." A small button in the top right corner of this banner says "Pantalla completa". To the right of the banner, there's a sidebar with a "iSumate a la comunidad!" button and a message: "Enterate de las novedades y dudas." The main content area lists five tasks, each preceded by a checked checkbox:

- He instalado Node
- He instalado Slack
- He instalado Visual Studio Code (u otro editor de texto)
- He instalado Git
- He creado mi usuario en GitHub

At the bottom of the screen, the Windows taskbar is visible, showing the Start button, a search bar labeled "Búsqueda", and various pinned application icons. The system tray shows the date and weather ("21°C Nublado") and connectivity status ("ESP LAA").

# SEMANA 2

## USO DE COMANDOS BASICOS

### CLI

La **interfaz de línea de comandos o CLI**, en inglés, command line interfaces, que no es más que una **caja negra** que puede recibir comandos a través de la escritura por medio de un teclado conectado a la computadora, procesarlos y arrojar un resultado, ya sea positivo si el comando se leyó con éxito, o negativo si escribimos algo erróneo o la computadora no pudo procesar el comando escrito.

### Comando

Un **comando** es simplemente una **instrucción** que al ejecutarlo en la CLI realiza una **acción**, cuando se recibe el comando el sistema operativo ejecuta un proceso que puede leerlo, manipular datos y producir una salida.

### Comandos más importantes para el manejo de la terminal:

1.

**pwd**: imprimir el directorio en donde nos encontramos.

2.

**ls**: imprime archivos o carpetas dentro del cual estamos trabajando.

3.

**mkdir carpetaPrueba**: crea una carpeta llamada 'carpetaPrueba' en el directorio que estamos situados. Si queremos crear una carpeta que lleve espacios en su nombre se debe utilizar comillas. Por ej "PrepCourse Henry"

4.

**cd**: para ir hacia el directorio que contiene la carpeta o directorio en donde nos encontramos. También podemos usarla para movernos entre carpetas, por ejemplo al usar **cd carpetaPrueba** cambiamos de la carpeta actual hacia "carpetaPrueba"

5.

**touch archivo.txt** : crea un archivo con nombre "archivo.txt"

6. **rm archivo.txt** : elimina el archivo "archivo.txt"

7.

**rm -r carpetaPrueba**: elimina la carpeta de nombre "carpetaPrueba"

## Creación de repositorio y primer commit

A continuación de compartimos los comandos más usados con **git**:

1. **git init**: comando para inicializar un repositorio local.
2. **git add**: comando para que nuestro repositorio sepa de la existencia de un archivo o de sus últimos cambios, no almacena las actualizaciones de forma definitiva, únicamente las guarda en algo que conocemos como “Staging Area” (área de montaje o ensayo). Se puede armar como git add ArchivoEjemplo.js (solo agrega ese archivo) o git add (agrega todos los archivos modificados de la carpeta donde estas con el punto).
3. **git commit -m "mensaje"**: comando para almacenar definitivamente todos los cambios que por ahora viven en el staging área. En el mensaje ponemos el mensaje que explica los cambios commiteados.
4. **git push**: comando para mandar nuestros commits a un servidor remoto.
5. **git status**: ofrece una descripción del estado de los archivos.
6. **git pull**: sirve para recibir cambios de repositorio remoto a local.

## PRIMEROS ESTILOS CON CSS

The image shows a computer screen with two windows open. The top window is a code editor displaying an HTML file named 'index.html' and a CSS file named 'styles.css'. The bottom window is a web browser displaying the rendered HTML content.

**Code Editor (index.html):**

```
1 <!--Estructura de datos-->
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Practica 2 Estilos y HTML</title>
7     <!--Esta es la linea de codigo que conecta el html con mi css-->
8     <link rel="stylesheet" href="estilos.css" type="text/css">
9   </head>
10  <body class="fondo">
11    <h1>PRIMEROS ESTILOS CON CSS</h1>
12    <p class="miTexto">Esta es mi página web y le daré estilos.</p>
13    <p>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Culpa, fugiat similique placeat accusamus veniam repellendus totam ab ducimus amet ratione. Dolorum recusandae, mollitia quos doloremque libero fugiat velit temporibus ea?</p>
14    <p>Lorem, ipsum dolor 82827y282828 282 8y273 238u328u2 823u2828. Culpa, fugiat similique placeat accusamus veniam repellendus totam ab ducimus amet ratione. Dolorum recusandae, mollitia quos doloremque libero fugiat velit temporibus ea?</p>
15    <p>Lorem, ipsum dolor 82827y282828 282 8y273 238u328u2 823u2828. Culpa, fugiat similique placeat accusamus veniam repellendus totam ab ducimus amet ratione. Dolorum recusandae, mollitia quos doloremque libero fugiat velit temporibus ea?</p>
16    <p>Lorem, ipsum dolor 82827y282828 282 8y273 238u328u2 823u2828. Culpa, fugiat similique placeat accusamus veniam repellendus totam ab ducimus amet ratione. Dolorum recusandae, mollitia quos doloremque libero fugiat velit temporibus ea?</p>
17    <p>Lorem, ipsum dolor 82827y282828 282 8y273 238u328u2 823u2828. Culpa, fugiat similique placeat accusamus veniam repellendus totam ab ducimus amet ratione. Dolorum recusandae, mollitia quos doloremque libero fugiat velit temporibus ea?</p>
18    <p id="titulo">Este es mi titulo con un id</p>
19  </body>
20 </html>
```

**Code Editor (styles.css):**

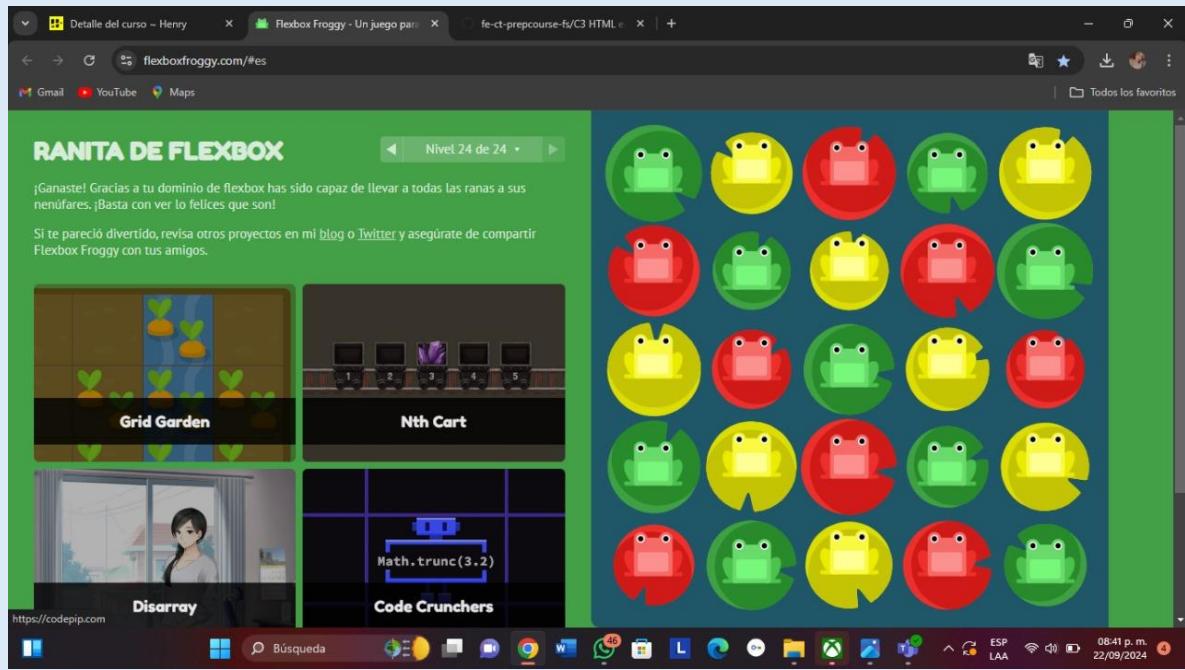
```
/* Estructura de datos */
```

**Browser Window:**

The browser window displays the rendered HTML content. The title is 'Practica 2 Estilos y HTML'. The page contains an H1 heading 'PRIMEROS ESTILOS CON CSS', several paragraphs of placeholder text ('Lorem ipsum'), and a paragraph with the ID 'titulo' containing the text 'Este es mi titulo con un id'.

At the bottom of the screen, the taskbar shows various open applications and system icons.

## JUEGOS DEL CURSO



# SEMANA 3

## TIENDA EN LINEA

The screenshot shows a code editor interface with two tabs open. The left tab displays a file tree for a project named 'Semana 3'. The right tab shows the content of 'style.css'.

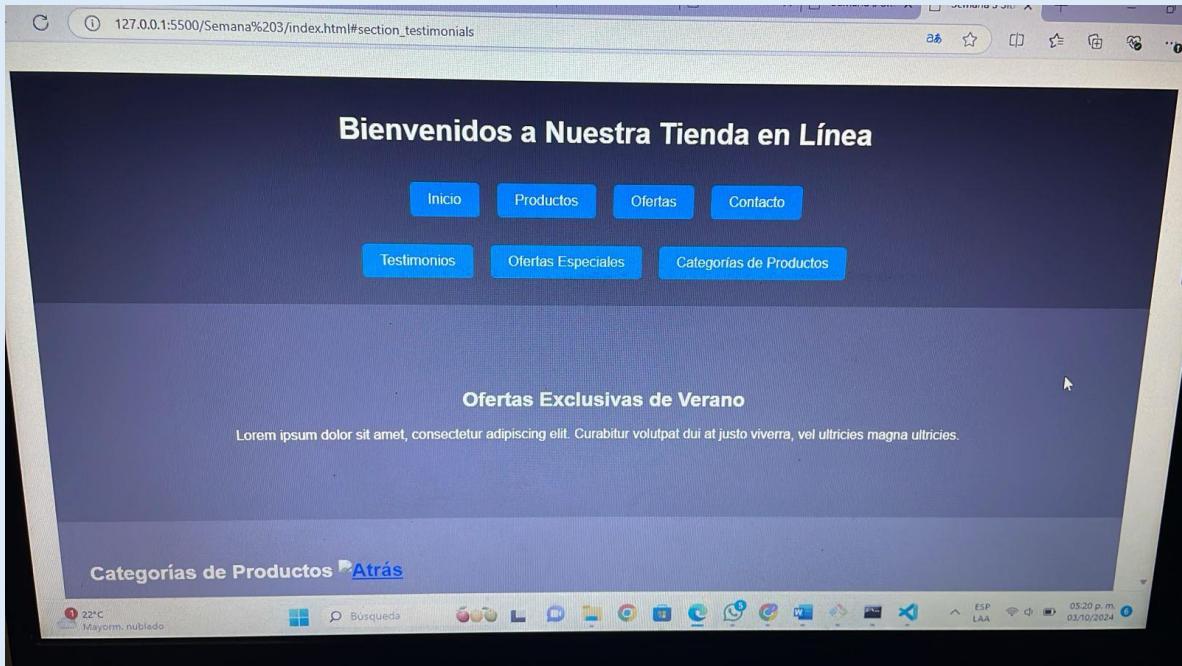
```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Semana 3 Sitio Web Personal</title>
7   <link rel="stylesheet" href="style.css">
8 </head>
9 <body>
10  <header>
11    <h1>Bienvenidos a Nuestra Tienda en Línea</h1>
12    <nav id="my_nav">
13      <div class="out">
14        <a href="#">Inicio</a></li>
15        <a href="#">productos</a></li>
16        <a href="#">ofertas</a></li>
17        <a href="#">contacto</a></li>
18      </div>
19      <div class="in">
20        <a href="#section_testimonials">Testimonios</a>
21        <a href="#section_offers">Ofertas Especiales</a>
22        <a href="#section_categories">Categorías de Productos</a>
23      </div>
24    </nav>
25  </header>
26
27  <section class="hero">
28    <h2>Ofertas Exclusivas de Verano</h2>
29    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur volutpat dui at justo viverra, vel...
30  </section>
31
32  <section class="s1" id="section_categories">
```

The status bar at the bottom indicates the following settings: Ln 1 Col 1, Spaces: 4, UTF-8, CRLF, HTML, and other system icons.

The screenshot shows a code editor interface with two tabs open. The left tab displays a file tree for a project named 'Semana 3'. The right tab shows the content of 'style.css'.

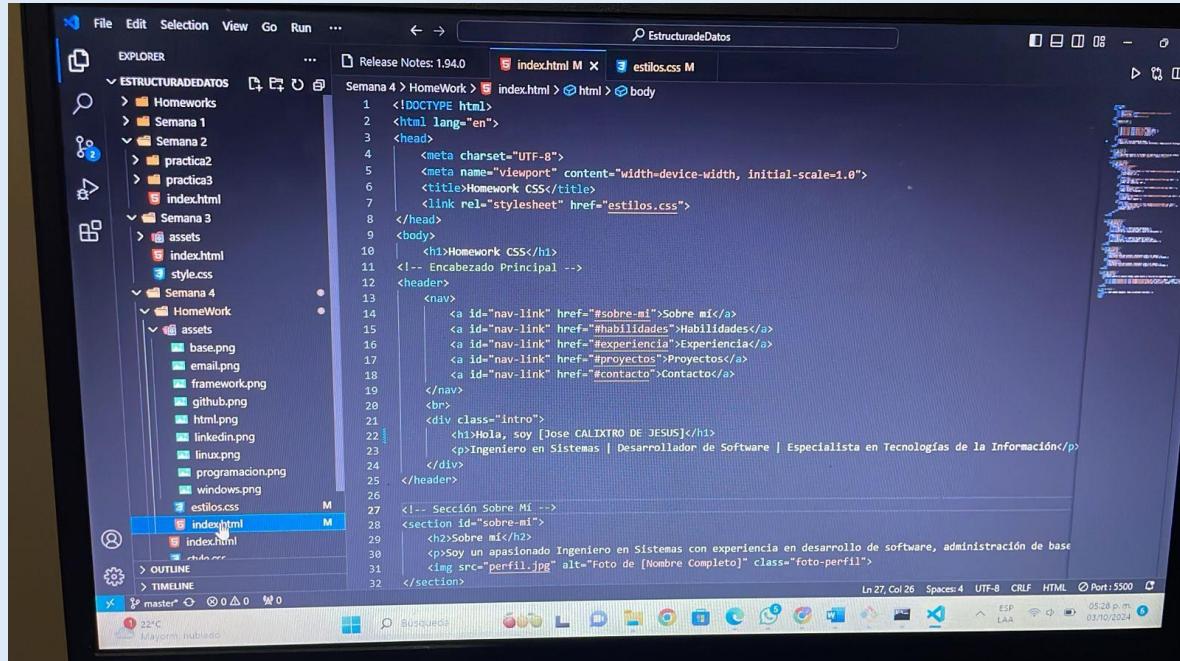
```
1 .my_back{
2   width: 25px;
3   height: 25px;
4 }
5
6 .my_back:hover{
7   width: 27px;
8   height: 27px;
9 }
10
11 body {
12   font-family: Arial, sans-serif;
13   margin: 0;
14   padding-left: 30px;
15   padding-right: 30px;
16   background-color: #white;
17 }
18
19 header {
20   margin-top: 30px;
21   background-color: #3c3c3c;
22   color: #white;
23   padding: 1rem 2rem;
24   text-align: center;
25 }
26
27 nav a {
28   color: #white;
29   margin-left: 10px;
30   margin-right: 10px;
31   text-decoration: none;
32   padding: 10px 20px;
```

The status bar at the bottom indicates the following settings: Release Notes: 1.94.0, style.css, and other system icons.



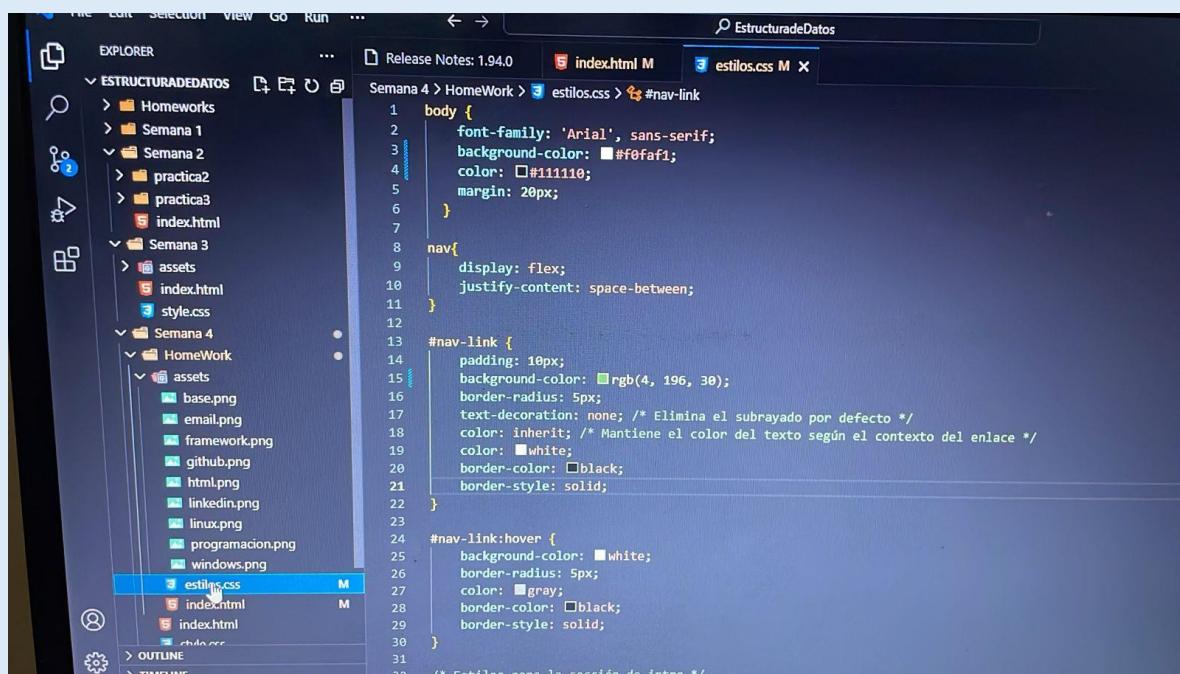
# Semana 4

## HOMEWORKS CSS



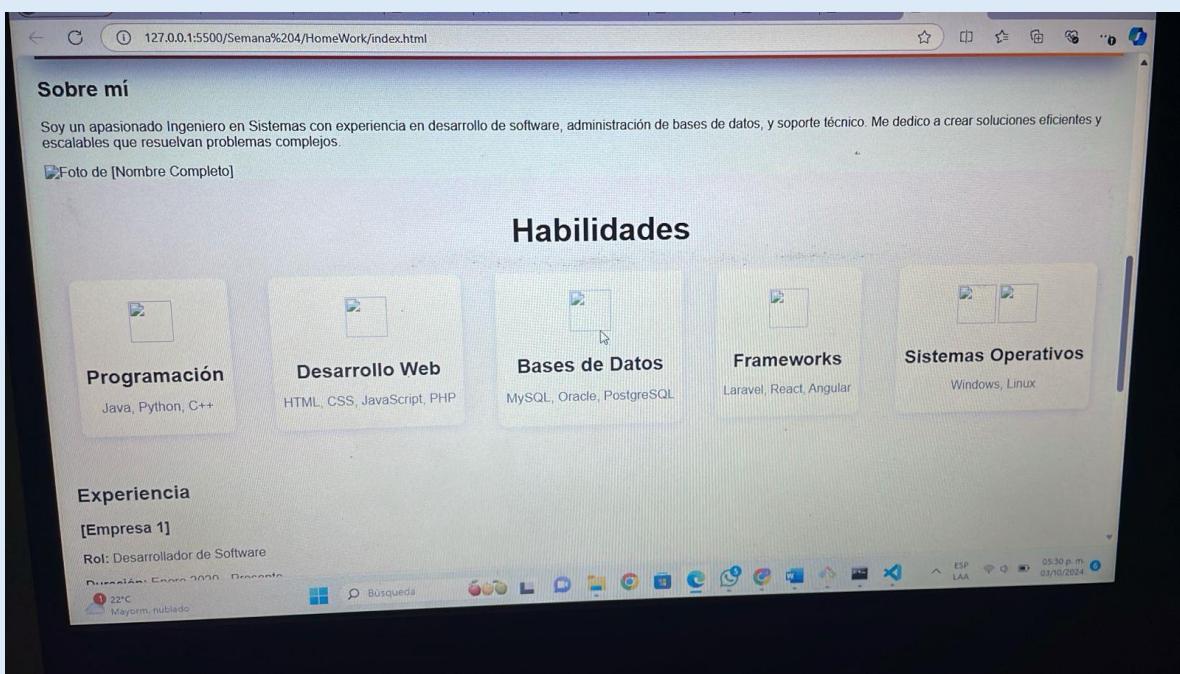
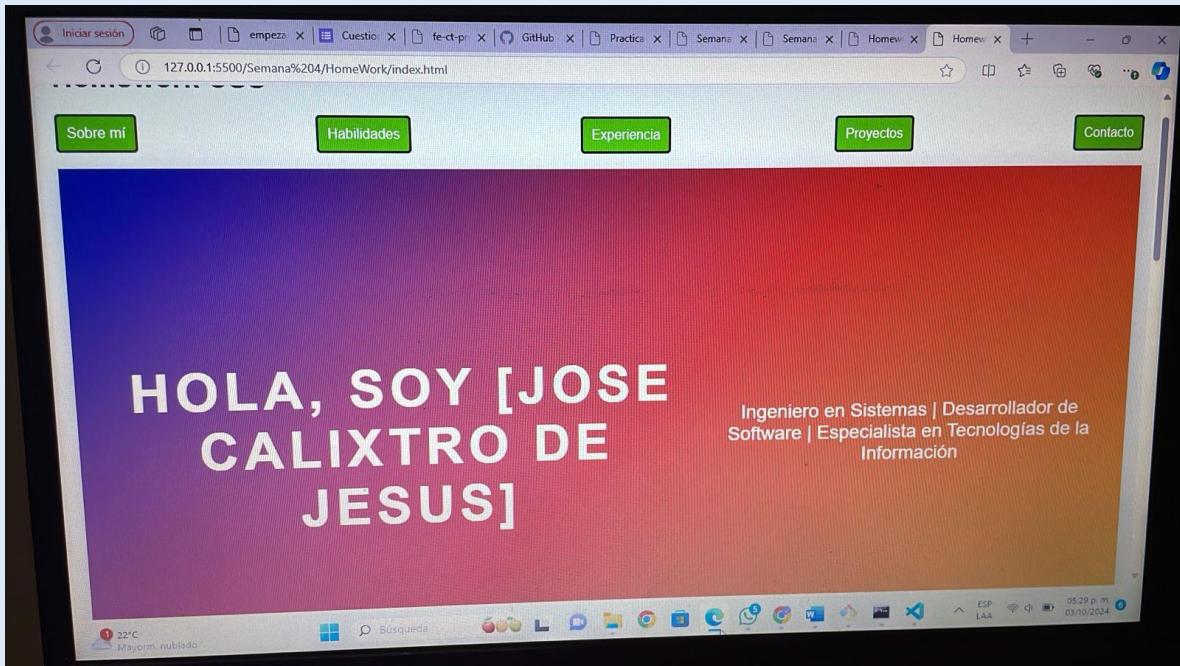
The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows a tree view of the project structure under "ESTRUCTURADEDATOS". The "Semana 4" folder contains "Homework", "practica2", "practica3", and "Semana 3". "Semana 3" contains "assets" (base.png, email.png, framework.png, github.png, html.png, linkedin.png, linux.png, programacion.png, windows.png), "index.html", and "estilos.css".
- Editor:** The main editor area displays the content of "index.html". The code includes an HTML5 doctype, meta tags for charset and viewport, a title "Homework CSS", and a link to "estilos.css". It features a header with a navigation menu containing links to "Sobre mí", "Habilidades", "Experiencia", "Proyectos", and "Contacto". Below the header is an "intro" section with a welcome message and a brief bio.
- Status Bar:** Shows the file path "Semana 4 > Homework > index.html", the file name "index.html M", the file type "HTML", the line number "Ln 27, Col 26", the character count "Spaces: 4", the encoding "UTF-8", the line separator "CRLF", the file extension ".HTML", and the port "Port: 5500".

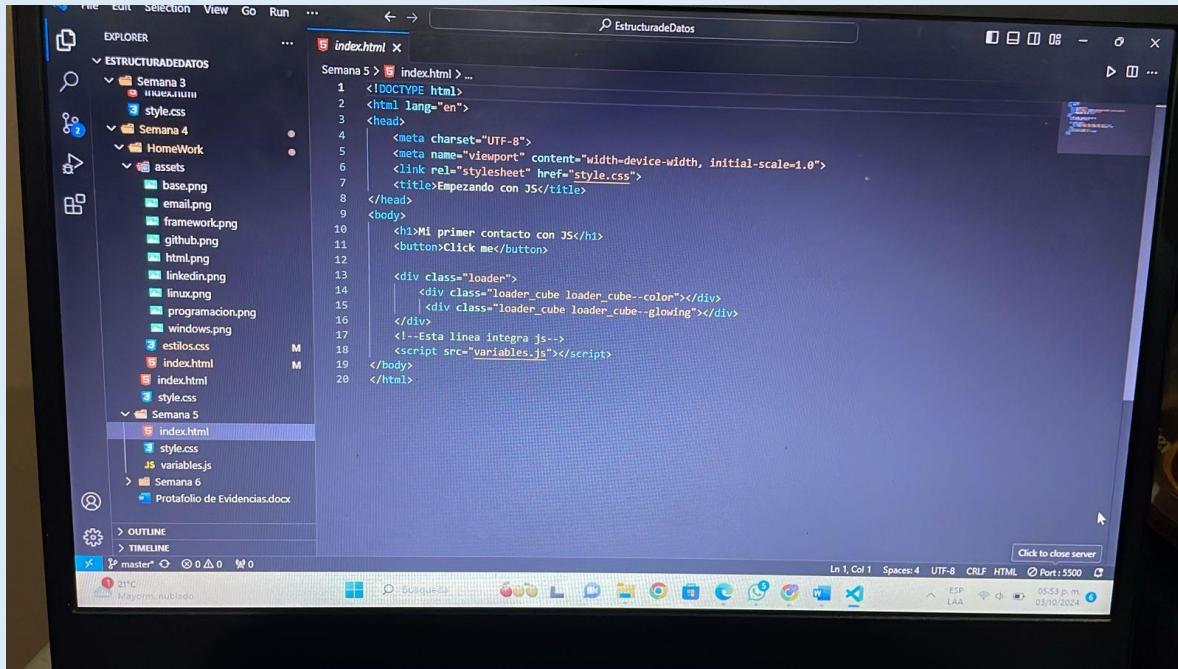


The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the same project structure as the previous screenshot, with "estilos.css" selected in the "Semana 4" folder.
- Editor:** The main editor area displays the content of "estilos.css". The code defines styles for the "body" and "nav" elements, and specifically styles for the "nav-link" class, including padding, background color, border-radius, text-decoration, color, and border properties.
- Status Bar:** Shows the file path "Semana 4 > Homework > estilos.css", the file name "estilos.css M", the file type "CSS", and the line number "Ln 31, Col 26".



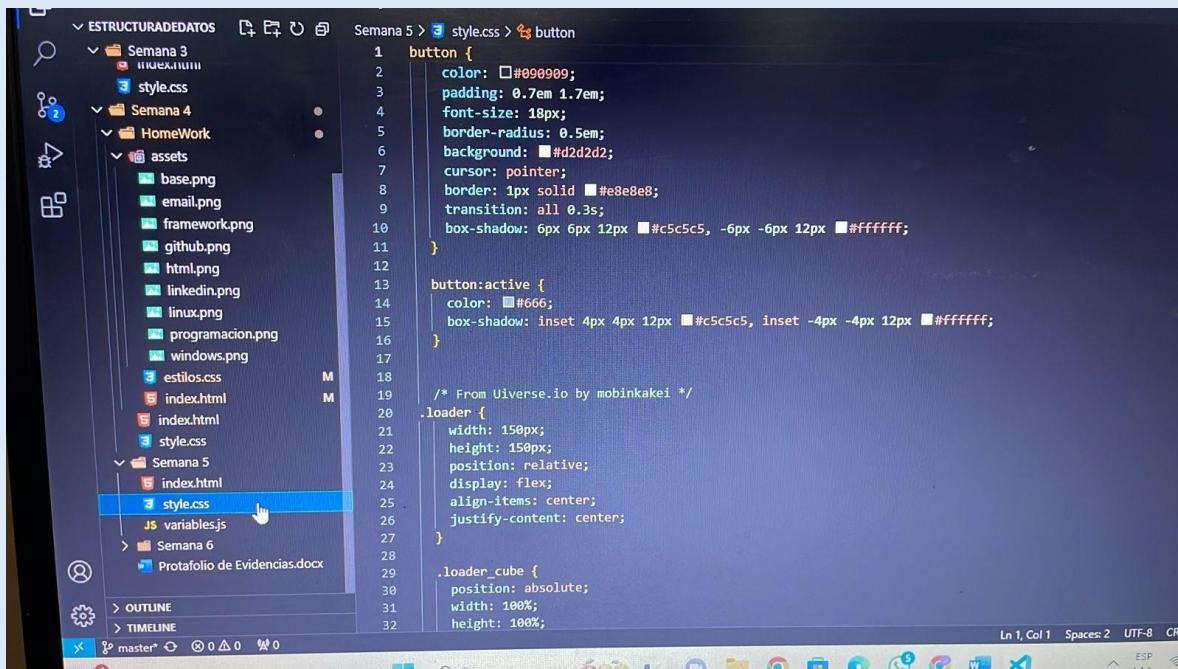
# SEMANA 5



The screenshot shows a code editor window with the file `index.html` open. The code contains HTML, CSS, and JavaScript. The CSS part includes styles for a button and a loader. The JavaScript part includes a script to change the background color of the loader cube.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="style.css">
    <title>Empezando con JS</title>
</head>
<body>
    <h1>Mi primer contacto con JS</h1>
    <button>Click me</button>

    <div class="loader">
        <div class="loader_cube loader_cube--color"></div>
        <div class="loader_cube loader_cube--glowing"></div>
    </div>
    <!-- Esta linea integra js-->
    <script src="variables.js"></script>
</body>
</html>
```



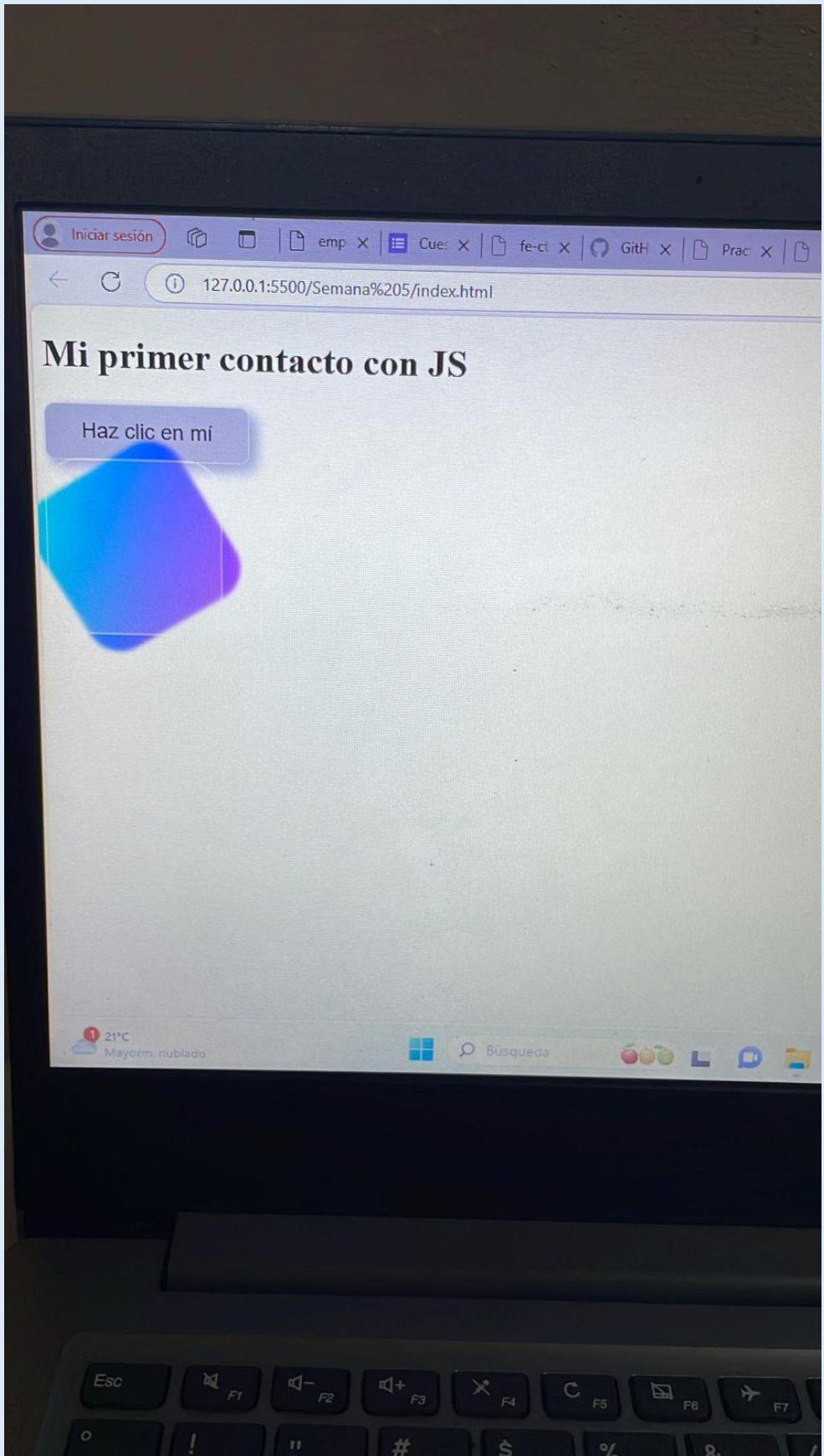
The screenshot shows a code editor window with the file `style.css` open. The code defines styles for a button and a loader component. The loader is styled with a 3D cube effect using CSS transforms and gradients.

```
button {
    color: #000000;
    padding: 0.7em 1.7em;
    font-size: 18px;
    border-radius: 0.5em;
    background: #d2d2d2;
    cursor: pointer;
    border: 1px solid #e8e8e8;
    transition: all 0.3s;
    box-shadow: 6px 6px 12px #c5c5c5, -6px -6px 12px #ffffff;
}

button:active {
    color: #666;
    box-shadow: inset 4px 4px 12px #c5c5c5, inset -4px -4px 12px #ffffff;
}

/* From Uiverse.io by mobinkakei */
.loader {
    width: 150px;
    height: 150px;
    position: relative;
    display: flex;
    align-items: center;
    justify-content: center;
}

.loader_cube {
    position: absolute;
    width: 100%;
    height: 100%;
```



The screenshot shows a code editor interface with the following details:

- Explorer View:** Shows a tree structure of files and folders. The 'variables.js' file is selected in the list.
- Code Editor:** Displays the content of the 'variables.js' file. The code is as follows:

```
1 //Ejercicio de Variables en js Declaración de variables con var
2 //Variables de tipo String no hay diferencia si uso comilla simple o doble
3 // este es el error var nombre : "Felipe";
4 //esta es la solución
5 var nombre = "Felipe";
6 var ciudad = 'Ajalpan';
7 //Variables de tipo número
8 var entero = 90;
9 var negativo = -120;
10 var decimal = 19.67;
11 //Variables de tipo Booleano
12 var verdad = true;
13 var mentira = false;
14 //Variable de tipo Undefined
15 var noseeltipo;
16 //Variable Null
17 var nula = null;
18
19 //Operadores matematicos en js
20 //variables contenedoras
21 var num1, num2;
22 //Suma
23 num1 = 50;
24 console.log(num1 + 5);
25 console.log(num1 + 5);
26 //Resta
27 num2 = 20;
28 console.log(num1-num2);
29 //División
30 console.log(256/num1);
31 console.log(num1/num2);
32 //Modulo
```

The status bar at the bottom indicates the following information: Ln 1, Col 1, Spaces: 4, UTF-8, CRLF, JavaScript, Port: 5500, 05:54 p.m., 03/10/2024.

# SEMANA 6

## CONTROL DE FLUJO (IF/ELSE)

En esta lección abordaremos la forma de direccionar nuestro código a través de los condicionales (If/else) son estructuras fundamentales para la resolución de problemas.

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface. The left sidebar displays a file tree under the 'EXPLORER' tab, showing a folder structure for 'ESTRUCTURADEDATOS' containing several 'Semana' folders (3, 4, 5, 6) and a 'HomeWork' folder. Inside 'Semana 6', there is a 'ejercicios' folder which contains the file 'condicionales.js'. The main editor area shows the following code:

```
function viaje(destino){
    if(destino === "Brasil"){
        console.log("Gire a la Derecha");
    } else if(destino === "Argentina"){
        console.log("Gire a la Izquierda");
    } else {
        console.log("Nos Perdimos");
    }
}

viaje("BRASIL");

function puedeManejar(edad){
    if(edad >= 18){
        console.log("Puede Manejar");
    } else{
        console.log("No Puede Manejar");
    }
}

puedeManejar(8);
```

The code defines two functions: 'viaje' and 'puedeManejar'. The 'viaje' function takes a destination as an argument and prints a turn instruction based on the destination. The 'puedeManejar' function takes an age as an argument and prints whether the person can drive or not. The code then calls 'puedeManejar' with the argument 8. The bottom right corner of the screen shows the Windows taskbar with various icons.

## EJERCICIOS

The screenshot shows a code editor interface with a terminal window open. The terminal window is titled 'fe-ct-precourse-fs [Administrator]' and displays the following content:

```
C5 Fundamentos Javascript > ejercicios > JS_36.js > numeroRandom
1 // Recuerda que debes utilizar el objeto global "Math".
2
3 function numeroRandom() {
4     // La función numeroRandom debe generar un número al azar entre 0 y 1 y retornarlo.
5     // Tu código:
6     return Math.random();
7 }
8
9 module.exports = numeroRandom;
10
```

Below the code, the terminal shows test results:

```
Test Suites: 6 failed, 36 passed, 44 total
Tests:    27 failed, 63 passed, 90 total
Schemas:  0 total
Time:    4.93 s
Run all test suites matching /S/.
```

The terminal also shows the command: PS C:\estructura de Datos\lomerWords\fe-ct-precourse-fs>

The left sidebar shows a file tree for 'FE-CT-PRECOOURSE-FS' containing various JavaScript files (JS 20.js, JS 21.js, JS 22.js, JS 23.js, JS 24.js, JS 25.js, JS 26.js, JS 27.js, JS 28.js, JS 29.js, JS 30.js, JS 31.js, JS 32.js, JS 33.js, JS 34.js, JS 35.js, JS 36.js) and other project files like .gitignore, package-lock.json, package.json, and README.md.