

Tutorial de como poner el razor, css y js:

Mi compañero Luis Ceballos, subió dos archivos a su Github de archivos comprimidos del razor y css, les enseñare a implementar también js, empezaremos primero con Corporativo y regiones ya que es el principal de todo.

Corporativo y regiones:

Bueno en este caso, yo tengo el de corporativo y regiones, pero para que mis demás compañeros puedan tenerlo aquí va lo siguiente:

Empezaremos haciendo algo sumamente rápido para que nada más puedan agarrar el archivo, copiarlo y pegarlo en donde corresponde, que es lo que les dire a continuacion:

PAGES:

Nuestro primer archivo pages será el siguiente:



La primera imagen es la principal de toda la página, nada más copian el archivo, lo pegan y ya queda, en caso de no entender, pueden arrastrar el archivo hacia la carpeta Pages, después harán lo mismo pero para regiones en este caso serán las siguientes imágenes:



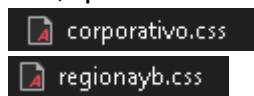
Esas se ponen ya que son las principales de la API que nos proporcionaron y las llevan directamente a las regiones que se muestran en la imagen.

CSS:

Lo siguiente será el css y cuales son y donde van a pegarlo:



Esa es la carpeta principal donde pegaremos los css a continuación se muestran los css, que usaremos:



Y eso es todo el CSS para corporativo.

JS:

En este caso por temas de privacidad el js se los dejare a continuación para que nada más puedan copiar y pegar pero primero les dire que carpeta hacer y dónde mandarlo a llamar.



En wwwroot haremos la siguiente carpeta y archivo:



Código:

```
var chartInstances = {};  
  
function destroyChart(chartId) {  
    if (chartInstances[chartId]) {  
        chartInstances[chartId].destroy();  
        delete chartInstances[chartId];  
    }  
}  
  
const baseOptions = {  
    responsive: true,  
    maintainAspectRatio: false,  
    plugins: {  
        legend: { display: false },  
        tooltip: { mode: "index", intersect: false }  
    }  
};  
  
window.chartInterop = {  
  
    createLineChart: (chartId, labels, datasets, yAxisDisplay) => {  
  
        destroyChart(chartId);  
        const ctx = document.getElementById(chartId);  
        if (!ctx) return;  
  
        datasets.forEach(d => d.tension = 0.5);  
  
        datasets[0].backgroundColor = "rgba(100, 181, 246, 0.4)";  
        datasets[0].borderColor = "#64B5F6";  
        datasets[0].fill = "start";  
        datasets[0].pointRadius = 0;  
    }  
};
```

```
if (datasets.length > 1) {  
    datasets[1].backgroundColor = "rgba(63, 81, 181, 0.3)";  
    datasets[1].borderColor = "#3F51B5";  
    datasets[1].fill = "start";  
    datasets[1].pointRadius = 0;  
}  
  
chartInstances[chartId] = new Chart(ctx, {  
    type: "line",  
    data: { labels: labels, datasets: datasets },  
    options: {  
        ...baseOptions,  
        scales: {  
            y: {  
                beginAtZero: false,  
                display: yAxisDisplay,  
                min: 0,  
                max: 60,  
                ticks: { stepSize: 10 },  
                grid: { color: "var(--border-light)" }  
            },  
            x: { grid: { display: false } }  
        }  
    }  
});  
},
```

```
createDoughnutChart: (chartId, labels, dataValues, colors) => {
    destroyChart(chartId);
    const ctx = document.getElementById(chartId);
    if (!ctx) return;

    chartInstances[chartId] = new Chart(ctx, {
        type: "doughnut",
        data: {
            labels,
            datasets: [
                {
                    data: dataValues,
                    backgroundColor: colors,
                    borderWidth: 0,
                    hoverOffset: 10
                }
            ]
        }
    });
}
```

```

    },
    options: {
      ...baseOptions,
      cutout: "70%",
    }
  });
},
};

createBarChart: (chartId, labels, datasets) => {
  destroyChart(chartId);
  const ctx = document.getElementById(chartId);
  if (!ctx) return;

  chartInstances[chartId] = new Chart(ctx, {
    type: "bar",
    data: { labels: labels, datasets: datasets },
    options: {
      ...baseOptions,
      scales: {
        y: {
          beginAtZero: true,
          grid: { color: "var(--border-light)" }
        },
        x: { grid: { display: false } }
      }
    }
  });
}
};

```

Ese es el código, para que se generen automáticamente las gráficas con efectos e interacciones.

COMO PONER CORPORATIVO EN EL SIDEBAR:

```
<div class="@NavMenuCssClass nav-scrollable" @onclick="ToggleNavMenu">
    <nav class="nav flex-column">
        <div class="nav-item px-3">
            <NavLink class="nav-link" href="/dashboard">
                <span class="bi bi-list-nested-nav-menu" aria-hidden="true"></span> Corporativo
            </NavLink>
        </div>
```

CÓMO HACER REFERENCIA AL CSS Y JS:

```
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Corporativo_Net</title>
    <base href="/" />
    <link rel="stylesheet" href="lib/bootstrap/dist/css/bootstrap.min.css" />
    <link rel="stylesheet" href="css/app.css" />
    <link rel="stylesheet" href="css/output.css" />
    <link rel="stylesheet" href="css/corporativo.css" />
    <link rel="stylesheet" href="css/Operador.css" />
    <link rel="stylesheet" href="css/regionayb.css" />
    <link rel="stylesheet" href="css/gestiondeclientes.css" />
    <link rel="stylesheet" href="css/gestiondeproductos.css" />
    <link rel="stylesheet" href="css/registrodeventas.css" />
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css" />

    <link rel="icon" type="image/png" href="favicon.png" />
    <link href="Corporativo_Net.styles.css" rel="stylesheet" />
</head>
```

En este caso sería únicamente los css que están en corporativo y regiones para hacer referencia en el caso de js haremos lo siguiente:

```
<body>
    <div id="app">
        <svg class="loading-progress">
            <circle r="40%" cx="50%" cy="50%" />
            <circle r="40%" cx="50%" cy="50%" />
        </svg>
        <div class="loading-progress-text"></div>
    </div>

    <div id="blazor-error-ui">
        An unhandled error has occurred.
        <a href"." class="reload">Reload</a>
        <span class="dismiss">X</span>
    </div>
    <script src="_framework/blazor.webassembly.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/chart.js@4.4.0/dist/chart.umd.min.js"></script>
    <script src="js/chart-interop.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/chart.js@4.4.1/dist/chart.umd.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

</body>
```

Sería hasta el último div antes del Body, dejare a continuacion los links que se encuentran ya que son frameworks que funcionan para la gráfica:

```
<script  
src="https://cdn.jsdelivr.net/npm/chart.js@4.4.0/dist/chart.umd.min.js"></script>  
<script src="js/chart-interop.js"></script>  
<script  
src="https://cdn.jsdelivr.net/npm/chart.js@4.4.1/dist/chart.umd.min.js"></script>  
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
```

REGIONES:

Ahora la siguiente parte será lo de regiones, como sabremos lo de corporativo ya está pero, si regiones quiere tener únicamente en su proyecto haremos lo siguiente:

PAGES:



Dentro de esta carpeta meteremos el archivo correspondiente a su región ya sea A o B (recuerden que se puede copiar el archivo y pegarlo para mas rapido, o arrastrar el archivo a la carpeta):



Y eso sería todo para las Pages.

CSS:

Lo siguiente que haremos es el css donde ponerlo y cómo implementarlo



Tendríamos que meterlo a esa carpeta el archivo de la siguiente imagen:



Y sería todo nada más haremos referencia en Index.html y al nav menu en este caso:

```
<link rel="stylesheet" href="css/regionayb.css" />
```

Ese sería para tu index.html para nav menu sería lo siguiente:

```
<div class="nav-item px-3">
  <NavLink class="nav-link" href="/region-a">
    <span class="bi bi-list-nested-nav-menu" aria-hidden="true"></span> Region A
  </NavLink>
</div>
```

Y como último paso, tendrías que hacer lo mismo que en corporativo de implementar el JS y listo.

SUCURSALES:

Y como último serían los de sucursales, aquí están los pasos para que puedan hacer lo siguiente de implementar el razor y el css:

PAGES:

Para esto tendrían que hacer lo siguiente:



Se meten aqui y pondrán los siguientes archivos mostrados (recuerden que pueden copiar y pegar el archivo en esa carpeta o arrastrar el archivo a esa carpeta):

```
Operador.razor
GestionDeClientes.razor
GestionDeProductos.razor
RegistroDeVentas.razor
```

CSS:

En este caso harán lo siguiente:

```
wwwroot
  css
  Operador.css
  gestiondeclientes.css
  gestiondeproductos.css
  registrodeventas.css
```

En este caso tendrías que meter todos los archivos que se muestran en la imagen y hacer referencia a su index.html y a su nav menu a continuación se muestra como:

```
<link rel="stylesheet" href="css/gestiondeclientes.css" />
<link rel="stylesheet" href="css/gestiondeproductos.css" />
<link rel="stylesheet" href="css/registroeventas.css" />
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css" />
<link rel="stylesheet" href="css/Operador.css" />
```

Y para su nav menu sería lo siguiente:

```
<div class="nav-item px-3">
  <NavLink class="nav-link" href="/operador">
    <span class="bi bi-list-nested-nav-menu" aria-hidden="true"></span> Operador
  </NavLink>
</div>
```

Y listo por fin quedaría lo de las sucursales, en este caso, no tienen que implementar el js, ya que solo les tocó esto.

INSTALACIÓN DE TAILWIND:

Iremos volando en esta parte, para la instalación haremos lo siguiente:

Le aplastaran al simbolo de windows y buscarán Powershell (lo vas a ejecutar como administrador) podrá el siguiente comando para que les de permisos:

Set-ExecutionPolicy RemoteSigned -Scope CurrentUser

Una vez dentro y ejecutado el comando, les saldrá los siguiente:

Cambio de directiva de ejecución

La directiva de ejecución te ayuda a protegerte de scripts en los que no confías. Si cambias dicha directiva, podrías exponerte a los riesgos de seguridad descritos en el tema de la Ayuda about_Execution_Policies en:

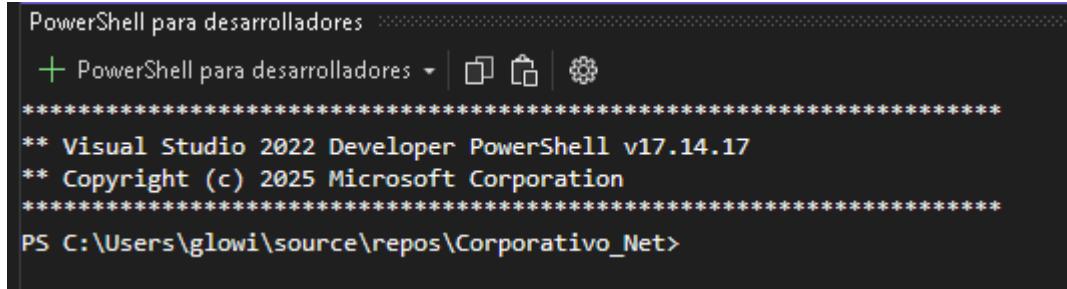
<https://go.microsoft.com/fwlink/?LinkID=135170>.

¿Quieres cambiar la directiva de ejecución?

[S] Sí [O] Sí a todo [N] No [T] No a todo [U] Suspender [?] Ayuda (el valor predeterminado es "N"):

Le aplastarán que “si” o “yes”, una vez hecho eso, cerrarán powershell y se irán a Visual Studio 2022, harán lo siguiente:

Abrirán la terminal de Visual Studio 2022:



```
PowerShell para desarrolladores
+ PowerShell para desarrolladores - | ⌂ ⌂ | ⚙
=====
** Visual Studio 2022 Developer PowerShell v17.14.17
** Copyright (c) 2025 Microsoft Corporation
=====
PS C:\Users\glowi\source\repos\Corporativo_Net>
```

Y pondrán los siguientes comandos:

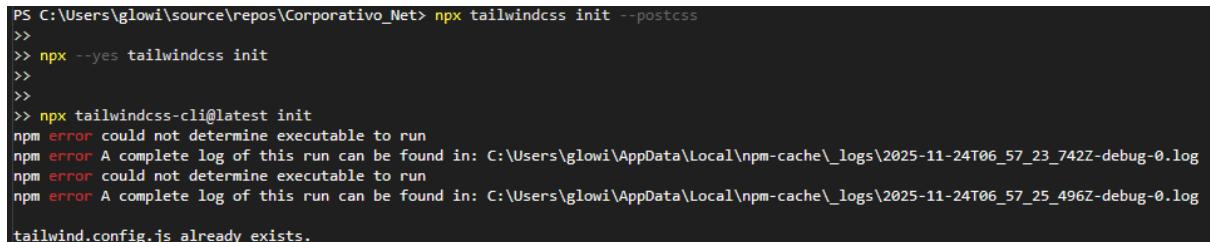
Cabe aclarar algo antes, cuando los pongan les saldrán dos fallidos o algo así, no pasa nada, si está bien el comando y se creará

npx tailwindcss init --postcss

npx --yes tailwindcss init

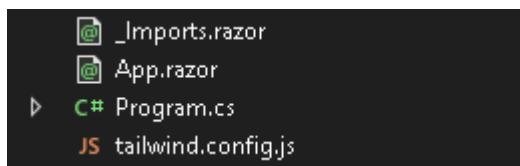
npx tailwindcss-cli@latest init

Una vez hecho eso, les aparecerá lo siguiente:



```
PS C:\Users\glowi\source\repos\Corporativo_Net> npx tailwindcss init --postcss
>
>> npx --yes tailwindcss init
>
>
>> npx tailwindcss-cli@latest init
npm error could not determine executable to run
npm error A complete log of this run can be found in: C:\Users\glowi\AppData\Local\npm-cache\_logs\2025-11-24T06_57_23_742Z-debug-0.log
npm error could not determine executable to run
npm error A complete log of this run can be found in: C:\Users\glowi\AppData\Local\npm-cache\_logs\2025-11-24T06_57_25_496Z-debug-0.log
tailwind.config.js already exists.
```

En mi caso ya lo tengo pero a ustedes se les va a crear y saldrá este nuevo archivo fuera de las carpetas:



Ya dependerá aquí si modifican su Tailwind, lo siguiente que haremos será esto:

Dentro de su wwwroot y css



Haremos un nuevo archivo, el cual será output.css:



Y copiarán el código que tiene su **app.css** y lo pegarán en **output** ya que lo reemplazamos con unas cosas de tailwind:

A screenshot of a code editor showing the contents of "app.css". The code consists of four numbered lines: 1. @tailwind base; 2. @tailwind components; 3. @tailwind utilities; 4. An empty line below the third line.

App.css

```
@tailwind base;  
@tailwind components;  
@tailwind utilities;
```

Y listo, hemos acabado, aquí está el manual de todo y si tienen dudas pueden preguntarme, hasta la próxima.