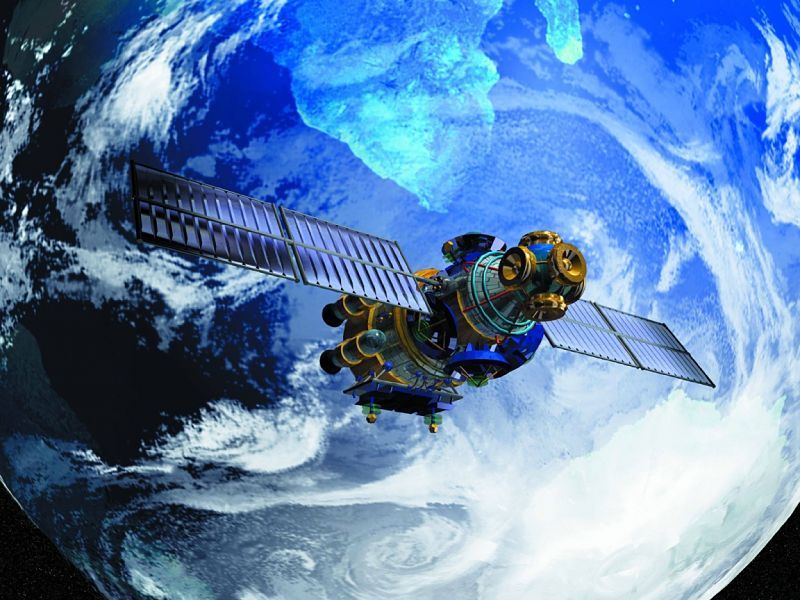
**PROYECTO MP4-5 UF3**



**CLOUDS OVER EUROPE**

**Natalia Soria**

**Natalia García**

INDEX

INTRODUCCIÓN……………………………………………..pag.2

DISEÑO……………………………………………………………pag.

DESARROLLO APLICACIÓN……………………………………..pag.

SUGERENCIAS DE MEJORAS………………………………….pag.

FUENTES…………………………………………………………pag.

INTRODUCCIÓN

DISEÑO

Nuestra idea como diseño fue implementar, mediante llamadas a diferentes apis, una página que calculara el clima en diferentes países de Europa. Quisimos que el usuario pueda elegir un país de los que se mostrara en pantalla y así saber en tiempo real su temperatura, precipitaciones, hora de amanecer, entre otras características.

Por lo tanto, en la primera página tendríamos tarjeras con imágenes de las banderas de los países de Europa, detrás sus ciudades y cada una lo llevaría a ver el clima.

Para la segunda página, a parte de los datos de clima, queríamos que, en base a sus estadísticas, cambiara el fondo dependiendo de los porcentajes en las precipitaciones, nubes, etc.

Al final de la segunda página pusimos una url, que te dirigiría a una tercera página que te muestra el estado del aire en dicho país.

DESARROLLO APLICACIÓN

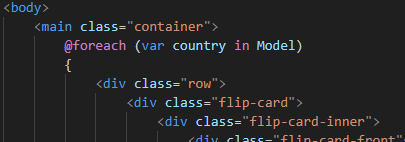
CSHTML

Para el diseño de la página web, creamos tres cshtml.

**“index.cshtml”**

En esta página principal, llamamos a las variables de “urlImagen”, “CountryName”, “capitalCountry” y “urlCityImage” desde la lista que hemos creado en el controller para los paises.

Declaramos un foreach al inicio para iterar sobre cada uno y que nos muestre los datos que pedimos.

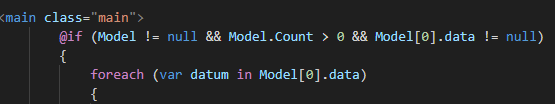


Utilizamos las “flip-card” para que se vean cada una de las banderas por el frente y las ciudades por detras. Las fuimos estilando de tal manera que queden posicionadas de cuatro en cuatro y fuimos añadiendo estilos de letras, medidas, colores (boton), entre otros.

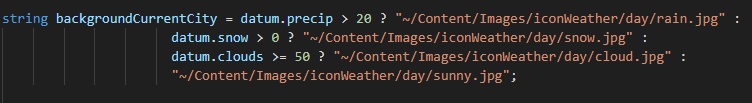
**“WeatherInformation”**

Para esta página tuvimos la idea de mostrar datos del clima del país seleccionado, y además que pueda cambiar el fondo, y las etiquetas en base a las estadísticas mostradas.

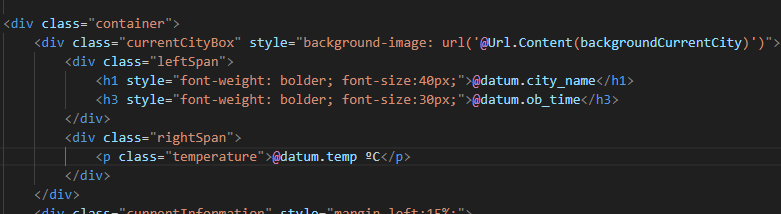
Dentro del main empezamos declarando un if, para el caso de haber nulos, y un foreach para iterar sobre los datos que vayamos pidiendo al model del tiempo.



Creamos una variable para que nos muestre una imagen si el dato pedido cumple con dicha condición.

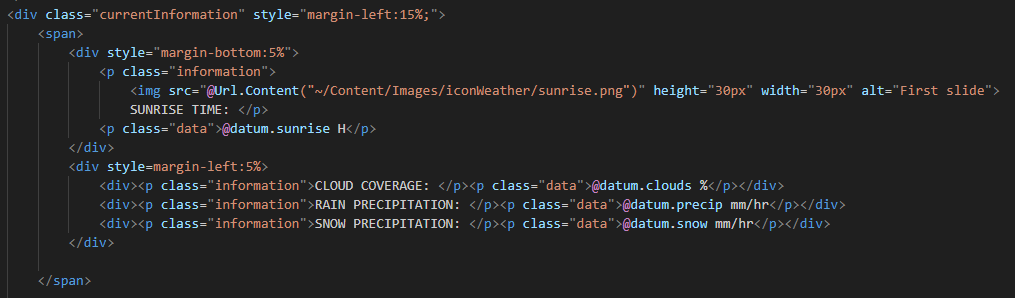


Empezamos realizando una clase “container” para contener todos los datos en un mismo bloque. la clase “currentCityBox”, sería como un navbar dentro del container, en él tiene la temperatura, ciudad, fecha y hora actual (aunque está atrasado por el envío de la api) del país. Llamamos a los datos requeridos y los posicionamos segun como queramos. Ademas utilizamos la variable anterior para mostrar la imagen en el currentcityBox.

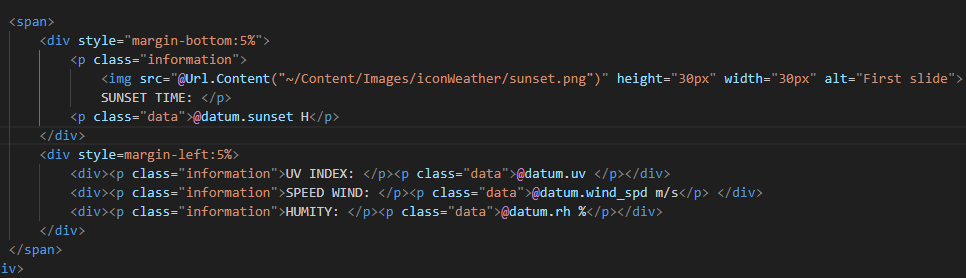


Para el siguiente sector de información, dividimos en dos secciones los datos para mostrarse en formato de columna, la primera sección la posicionamos a la izquierda y la otra a la derecha.

La primera seccion es la de “Sunrise time” con sus respectivos llamados a los datos de model y fuimos acomodando las medidas según se iba requiriendo.



La segunda seccion es “suset time”. Se encuentra estructurada como la anterior, con la diferencia de que está posicionada a la derecha.



Al final del container creamos un apartado que muestra la cantidad de contaminación en el aire y además nos dirige a otra página donde se muestran los datos del estado del aire. “Quality Air <datum.aqui> ICA”.

Realizamos condiciones con if para que, en caso de ser muy alto, sea considerado malo y en caso de ser bajo sea muy bueno.

Por ejemplo, en el caso de tener + de 100 sería considerado malo.



Dentro del if, declaramos una clase, y un cambio de fondo de alrededor de “Quality Air” para dicha condición.



A su vez declaramos dos clases, lo estilamos y hacemos referencia a la url a la que se navegara cuando se haga click en el enlace. Realizamos una llamada a la función especificada y desde ahí se genera la url a la página “AqiAirInformation”. También se pasan los valores de latitud y longitud y se los asigna a dos variables.

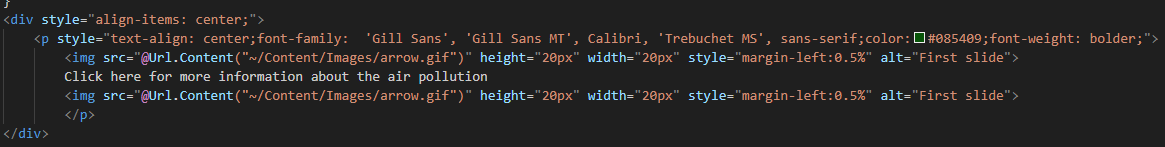


Por último, dependiendo de si es malo o bueno, se agrega un icono diferente a la derecha del texto.



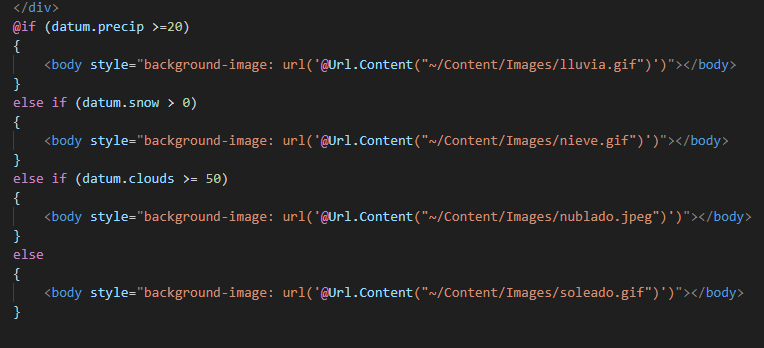
Realizamos la misma accion en los demas elif. Para cada condición cambiara la imagen, la cantidad y el icono.

Debajo del bloque de “Quality AIR” quisimos que se sepa que es una url, entonces pusimos “Click here for more information about the air pollution” para que al clickear sepan que hay más información en otra página.



En esta página tuvimos la idea de que, dependiendo el clima de cada pais, se cambie el fondo (soleado, nublado, lluvia, nieve).

Realizamos una condición para cada caso y lo aplicamos dentro del “body”.



SUGERENCIAS DE MEJORAS

Como sugerencia de mejora para el proyecto, podríamos realizar más apartados, donde el usuario tenga posibilidades de ingresar a más datos relacionados con los paises, clima, etc.

Podriamos reorganizar la página del inicio de tal forma que no se vean solo las tarjetas, sino otros sectores.

Estaria bueno ademas, que se pueda redireccionar hacia atrás desde el “Clouds over Europe” y no solo al inicio.

FUENTES

API’S USADAS EN LA APLICACIÓN:

* [GeoNames](https://www.geonames.org/)
* [Weatherbit | API Documentation](https://www.weatherbit.io/api)

WEBS CONSULTADAS:

* [Convert JSON to C# Classes Online - Json2CSharp Toolkit](https://json2csharp.com/)
* [Bootstrap en Español · La biblioteca HTML, CSS y JS más popular del mundo. (esdocu.com)](https://getbootstrap.esdocu.com/)
* [CSS: Textos con color degradado | Front.id](https://front.id/es/articles/css-textos-con-color-degradado)
* [Bordes de color para perfilar letras: text-stroke y una alternativa | Oloblogger](https://www.oloblogger.com/2014/06/borde-texto-stroke-shadow.html)
* [Flip Cards o Tarjetas Giratorias con CSS3 – Josetxu.com](https://josetxu.com/flip-cards-o-tarjetas-giratorias-con-css3/)
* [Llamar a una API web desde un cliente .NET (C#): ASP.NET 4.x | Microsoft Learn](https://learn.microsoft.com/es-es/aspnet/web-api/overview/advanced/calling-a-web-api-from-a-net-client)
* [List of European capitals by countries (countries-ofthe-world.com)](https://www.countries-ofthe-world.com/capitals-of-europe.html)
* <https://www.w3schools.com/html/html_layout.asp>
* <https://josetxu.com/flip-cards-o-tarjetas-giratorias-con-css3/>
* <https://learn.microsoft.com/es-es/aspnet/mvc/overview/older-versions/mvc-music-store/mvc-music-store-part-3>
* <https://learn.microsoft.com/es-es/aspnet/core/mvc/views/razor?view=aspnetcore-8.0>
* <https://www.byronvargas.com/web/como-centrar-una-card-en-css/>