

UNIVERSIDAD NACIONAL AUTÓNOMA DE NICARAGUA

UNAN – LEÓN

FACULTAD DE CIENCIAS Y TECNOLOGÍA  
INGENIERÍA EN SISTEMAS DE LA INFORMACIÓN



AÑO LECTIVO: 2025

SEMESTRE: II

Tarea 2

Componente Curricular: **PROGRAMACION  
ORIENTADA A LA WEB II**

Grupo: GP1

Profesor(a): Lic. Juan Carlos Leyton Briones

Autor:

1. Francisco José Jarquín Briceño.

León, Nicaragua, 1 de agosto de 2025.

*“¡A la Libertad por la Universidad!”*

## Introducción

Esta aplicación es una aplicación para probar y experimentar el uso de Modelos, y la implementación de una API, consumiendo datos de ella.

## Aplicación

La aplicación es una página web que te permite visualizar los datos del clima actual de las cabeceras departamentales de Nicaragua.

Esta aplicación consta de todos los elementos por defecto al crear un nuevo proyecto usando el modelo MVC con el comando `dotnet new mvc -o NombreDeProyecto`, a esto se le ha implementado la API proporcionada por [meteoblue](#), la cual proporciona una API para recolectar información variada y diversa del clima.

Aparte de eso, durante el desarrollo de la aplicación se han agregado los siguientes archivos:

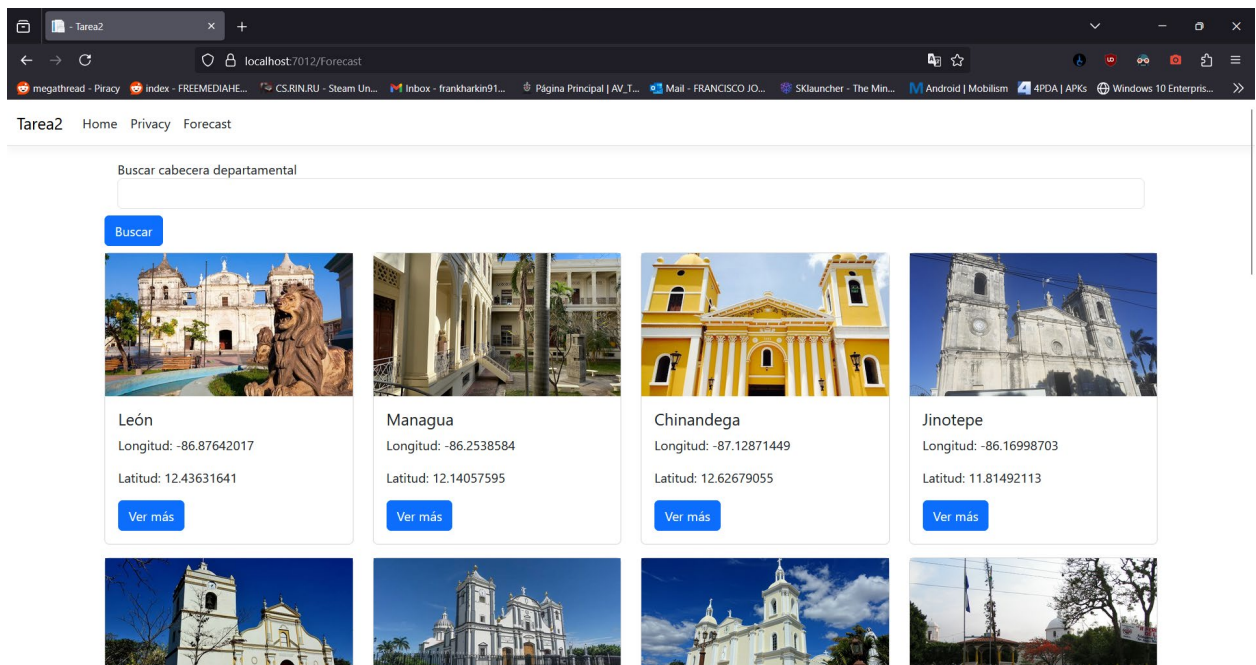
### **ForecastController.cs**

Es un controlador que posee los siguientes elementos:

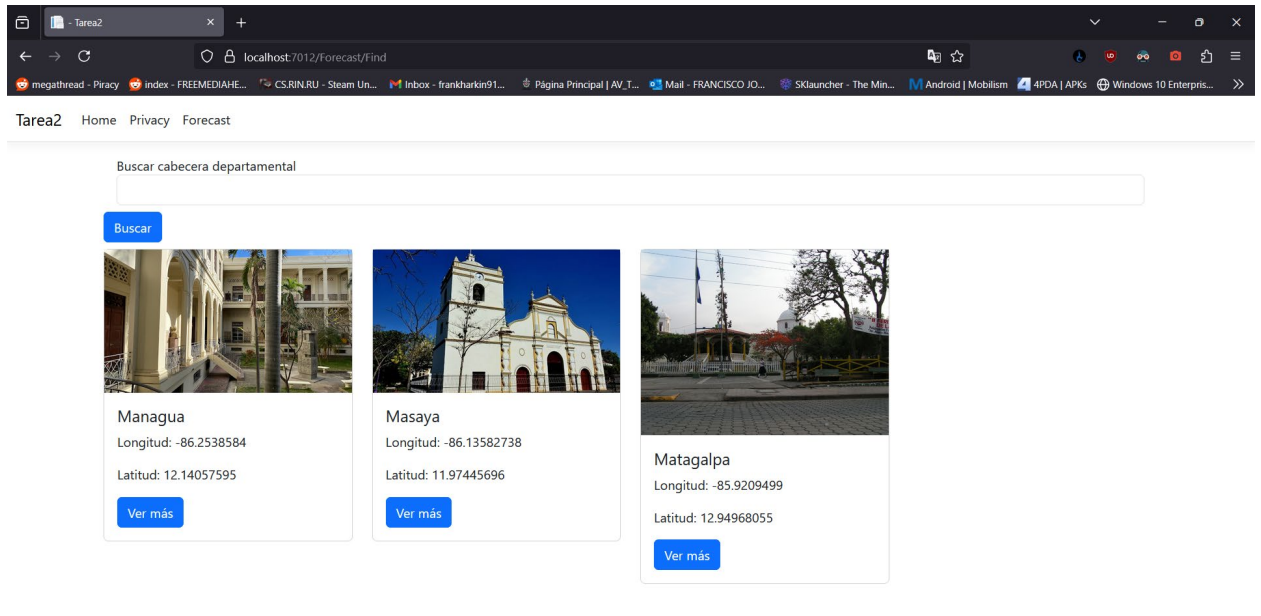
- `currentForecast`: Es una variable que contiene el pronóstico actual cuyos datos nos brinda la API de meteoblue. Es del tipo `CurrentForecast`.
- `cabeceras`: Es una lista de las cabeceras departamentales de Nicaragua, cuyos datos vienen de un archivo JSON con datos relevantes de ellas. Es del tipo `List<Cabecera>`.
- `FetchData(url)`: Es un método que hace el llamado a la API y asigna los datos a la variable `currentForecast`. Debido a la naturaleza del llamado a la API es un método asíncrono que retorna un `Task`, del cual se debe esperar que finalice para poder continuar la ejecución de la aplicación.
- `SignUrl(query)`: Es un método que firma el URL usado para llamar a la API con un secreto compartido. Retorna el nuevo URL a usar para el llamado a la API.
- `InitializeCurrentForecast(jsonObj)`: Es un método que inicializa una instancia de `CurrentForecast` con los datos proporcionados por la API a través de `jsonObj`, el cual es el contenido del archivo JSON en un objeto del tipo

JsonObject. Retorna una instancia de CurrentForecast con los datos del llamado a la API.

- ForecastController(): Constructor del controlador en cuestión, durante este momento se lee el archivo ".../Models/Cabeceras.json" y se asignan esos datos a la lista de cabeceras.
- Index(): Es el método para invocar a la vista ".../Forecast/Index.cshtml", a esta vista se le pasa la lista de cabeceras para poder mostrar información de estas. Retorna la vista "Index" con la lista de cabeceras.

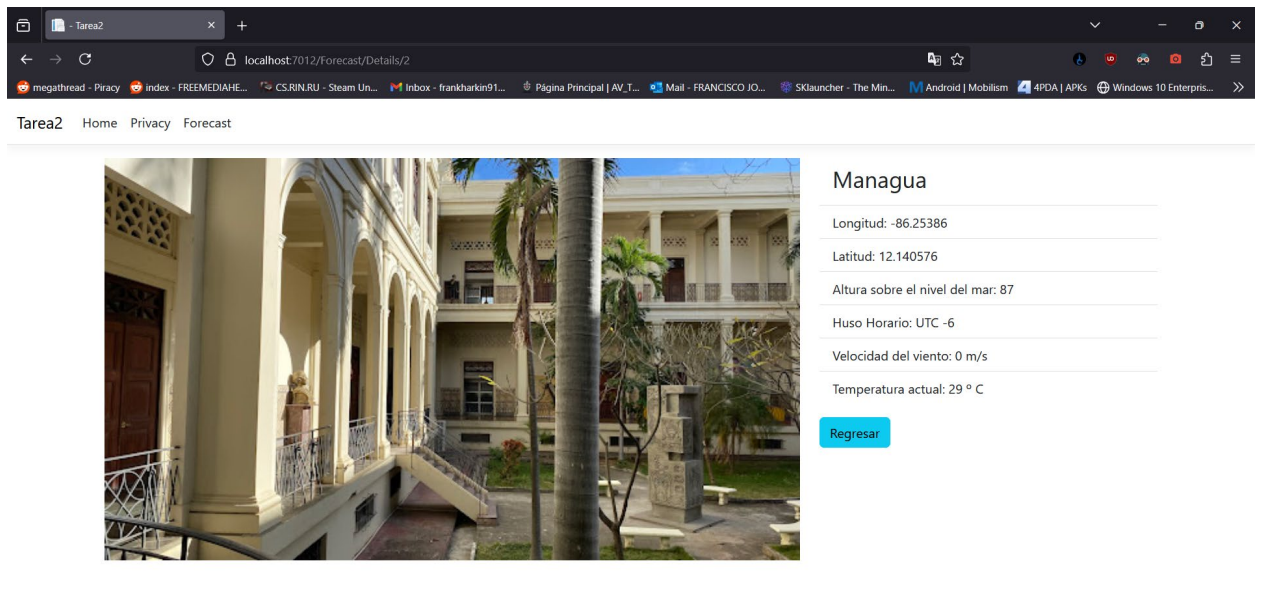


- Find(cabecera): Es un método que busca en la lista de cabeceras cualquier cabecera que contenga el texto pasado por su argumento y los asigna a una nueva lista de resultados. Retorna la vista "Index" con una lista de resultados.



### Resultados al buscar “Ma”

- **Details(ID):** Es un método que invoca a la vista “.../Forecast/Details.cshtml”, mostrando los detalles de la cabecera departamental seleccionada proporcionados por la API. Retorna la vista “Details” con los detalles de la cabecera, o con detalles vacíos de no ser encontrada.



Models/Cabecera.cs

Es una clase que contiene los datos de cada cabecera departamental.

Models/Cabeceras.json

Es un archivo JSON con todas las cabeceras departamentales y datos relevantes, en especial la latitud y longitud de estas.

Ejemplo de una cabecera:

```
{
  "ID": 1,
  "Nombre": "León",
  "Longitud": -86.87642017,
  "Latitud": 12.43631641,
  "Imagen": "/img/leon.jpeg"
}
```

Models/CurrentForecast.cs

Es una clase que contiene la información del pronóstico actual del tiempo proporcionados por la API, esta información está dividida en 3: Metadata, Units, y DataCurrent, los cuales son clases en sí.

Ejemplo de los datos retornados al llamado de la API, los cuales son guardados por CurrentForecast:

```
{
  "metadata": {
    "modelrun_updatetime_utc": "2025-09-01 02:35",
    "name": "",
    "height": 87,
    "timezone_abbreviation": "CST",
    "latitude": 12.140576,
    "modelrun_utc": "2025-09-01 02:35",
    "longitude": -86.25386,
    "utc_timeoffset": -6.0,
    "generation_time_ms": 20.491005
  },
  "units": {
```

```

        "temperature": "C",
        "time": "YYYY-MM-DD hh:mm",
        "windspeed": "ms-1"
    },
    "data_current": {
        "time": "2025-08-31 20:00",
        "isobserveddata": 1,
        "metarid": "METAR_MNMG",
        "isdaylight": 0,
        "windspeed": 2.057778,
        "zenithangle": 134.22475,
        "pictocode_detailed": 4,
        "pictocode": 2,
        "temperature": 29.0
    }
}

```

#### Models/DataCurrent.cs

Es una clase que contiene la información actual del tiempo como tal, incluyendo la temperatura y velocidad del viento.

#### Models/Metadata.cs

Es una clase que contiene los metadatos del llamado a la API, incluyendo el tiempo del llamado, latitud y longitud, entre otras cosas.

#### Models/Units.cs

Es una clase que contiene las unidades de medida utilizadas, por ejemplo, grados Celsius o Fahrenheit.

#### Forecast/Details.cshtml

Es la vista con los detalles de la cabecera departamental, muestra una imagen representativa de la cabecera en cuestión, el nombre de esta cabecera, y datos proporcionados por la API, como la longitud, latitud, temperatura, velocidad del viento, altitud sobre el nivel del mar, etc.

Forecast/Index.cshtml

Es la vista que nos da un listado de todas las cabeceras departamentales, permitiéndonos ver una imagen representativa de ellos, su nombre, y longitud y latitud respectiva. También nos permite buscar entre las cabeceras departamentales y ver más detalles de estos.

wwwroot/img

Contiene las imágenes representativas de cada cabecera departamental.