

WeatherViewer - Documento de análisis y diseño

USER STORY

Como responsable de la unidad de información climática quiero un módulo web para consultar las condiciones climáticas actuales de una determinada ciudad, con la opción de ver el histórico de consultas previas.

Mockup de referencia:

Informe del clima

Selecione la Ciudad ☐ Incluir histórico

Ciudad: Buenos Aires, Argentina
Clima: 31,6° - Sensacion Termica: 33°

País	Ciudad	Clima	Sensación Térmica
Argentina	Buenos Aires	31.6°	33°
Argentina	Buenos Aires	28.2°	27°
Argentina	Buenos Aires	31°	32.5°
Argentina	Buenos Aires	29.7°	30°
Argentina	Buenos Aires	28.2°	27.6°
Argentina	Buenos Aires	27.6°	26.8°
Argentina	Buenos Aires	27°	26°

Criterios de Aceptación:

- Se requiere utilizar la API abajo definida
- Se debe poder configurar las ciudades que se podrán consultar.
- La información debe ser persistida en base de datos de forma de poder consultar los históricos de la ciudad configurada.
- El checkbox incluir histórico deberá incluir al clima actual los últimos X registros del historial de consultas, en caso que esté seleccionado.
- El botón consultar mostrará el estado del clima actual y la historia dependiendo del punto anterior.

HERRAMIENTAS/TECNOLOGÍA

- Para Back office: .NET Core API Rest
- Para la GUI MVC Razor
- DataBase: SQL Server

API: <http://openweathermap.org/current>

Ej:

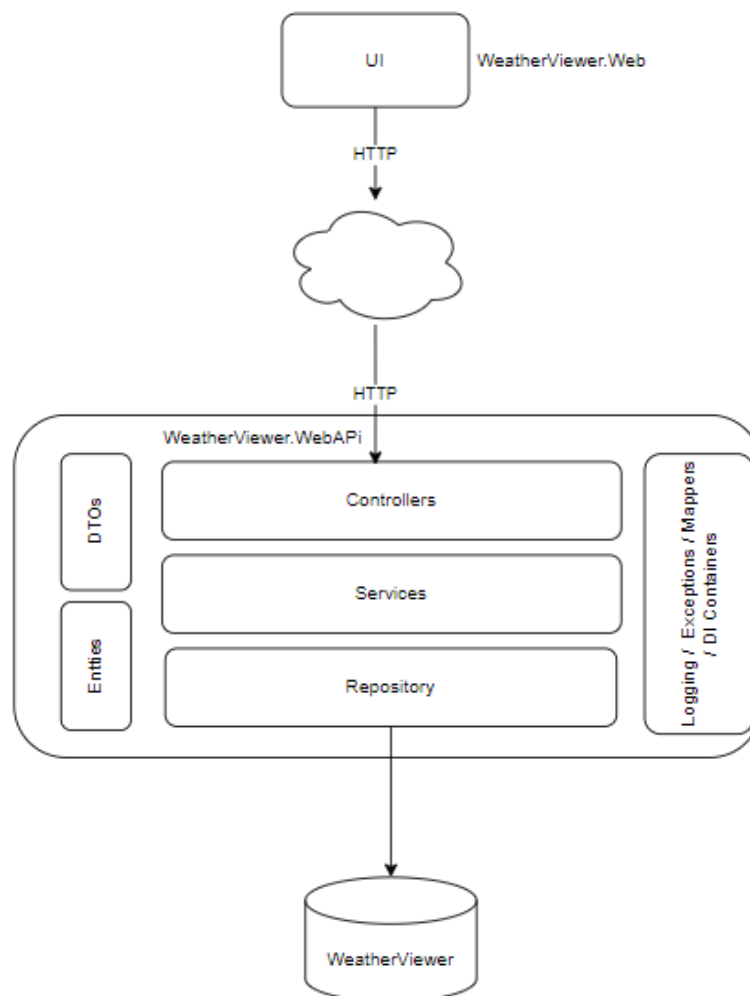
[http://api.openweathermap.org/data/2.5/weather?q=Buenos%20Aires.ar&appid=\[MyApiKey\]](http://api.openweathermap.org/data/2.5/weather?q=Buenos%20Aires.ar&appid=[MyApiKey])

Assumptions:

- Que el usuario ya está logueado al sistema y los endpoints provistos por la web api son públicos.
- Que la lista de ciudades son aquellas con más de 500.000 habitantes.
- El Front mostrará por defecto ciudades de Argentina (país por defecto)
- Que la vista mostrará páginas de 10 registros históricos, máximo 1 mes (configurable en appsettings.json)

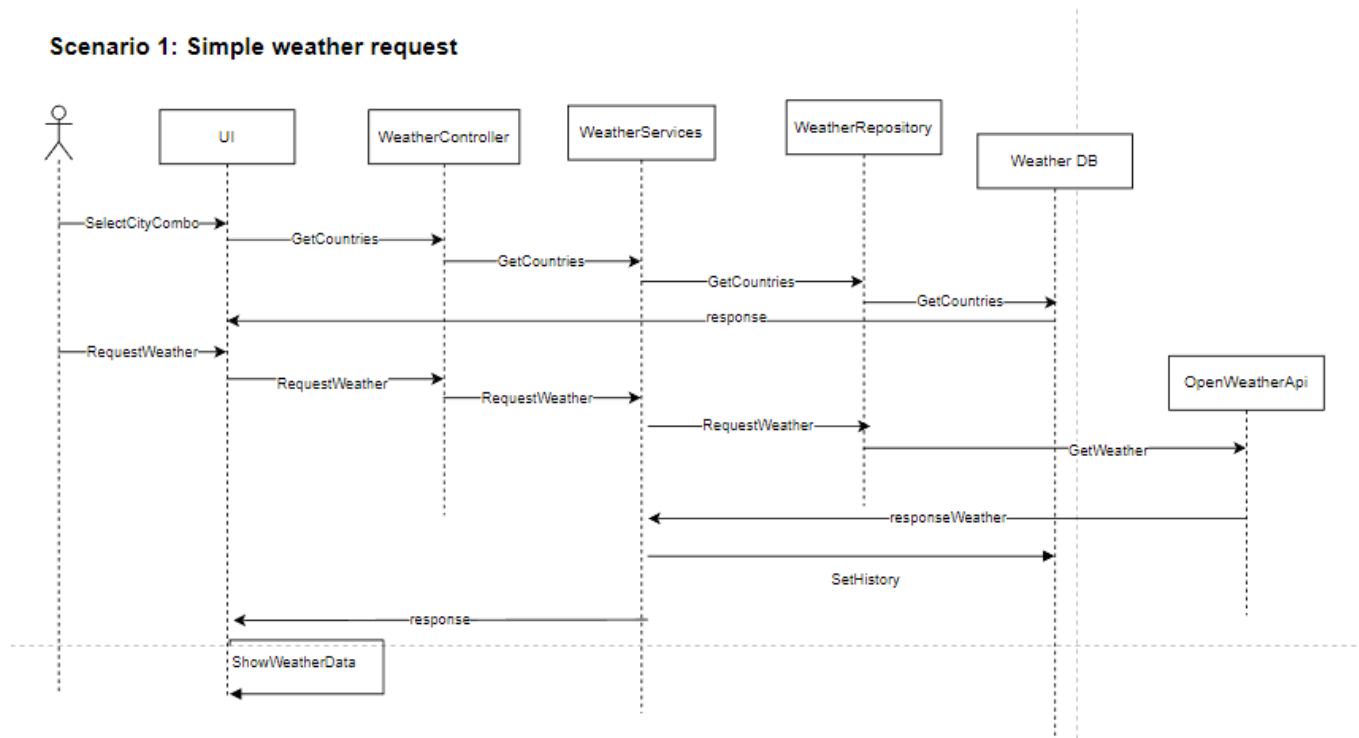
ARQUITECTURA

Vista General

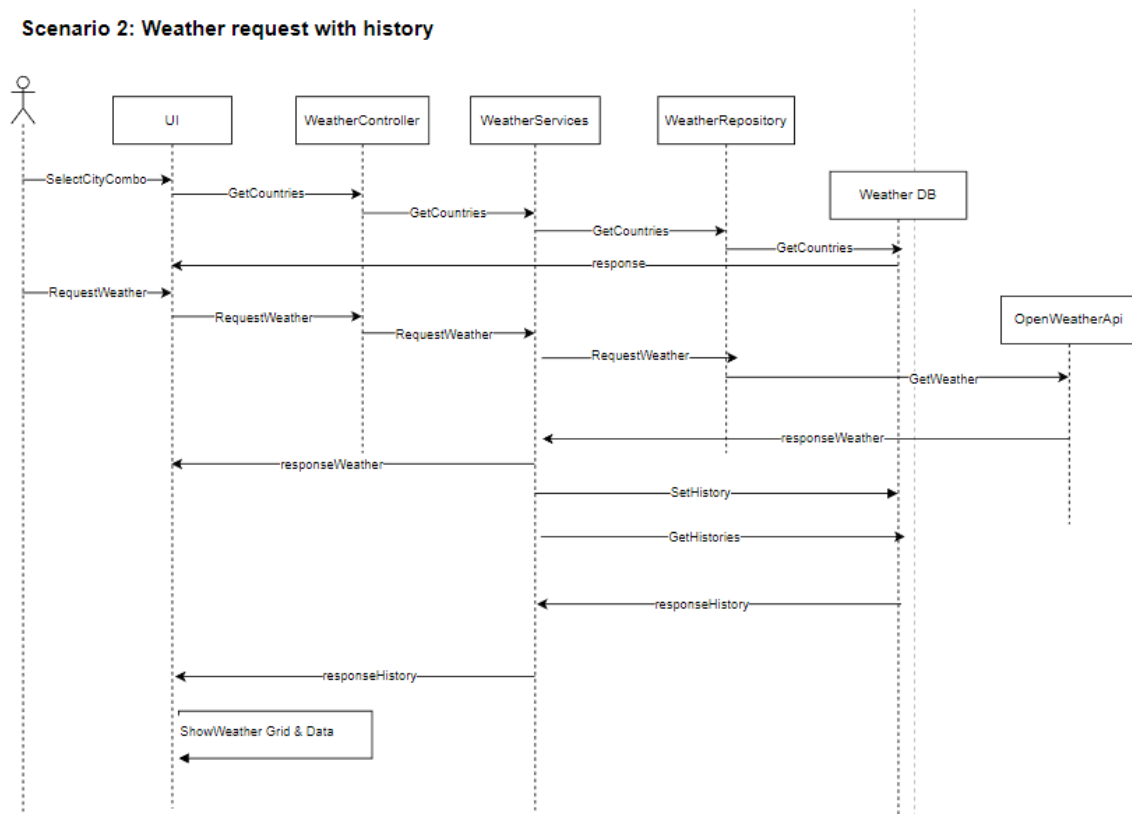


Vista dinámica

Scenario 1: Simple weather request



Scenario 2: Weather request with history



PATRONES A UTILIZAR:

Repository: Para separar la lógica de acceso a datos con la de los controladores. Se agrega una capa de servicios para agregar lógica de común a todas las operaciones , como registro de históricos.

Separation of concepts: Aplicar código limpio en métodos y clase (una clase -> una funcionalidad, un método -> una funcionalidad)

Dependency Injection: Utilizada en forma nativa al inyectar el DbContext y el IMapper y en forma custom al agregar objetos de acceso a servicios, repositorios y mapas custom (en caso de ser necesario)

Async / Await Calls: Los controladores expondrán endpoints asincrónicas para beneficiar al sistema de la característica de async/await que ofrece .Net. Esto beneficia la performance del sistema.

ACCESO A DATOS

Se utilizará Entity Framework Core 7 con el approach de Code First

Modelo Entidad Relación

El MER contiene 3 entidades:

- **WeatherHistory** contendrá todas las consultas hechas a la web api de open weather.
- Country: Listado de paises con su nombre

