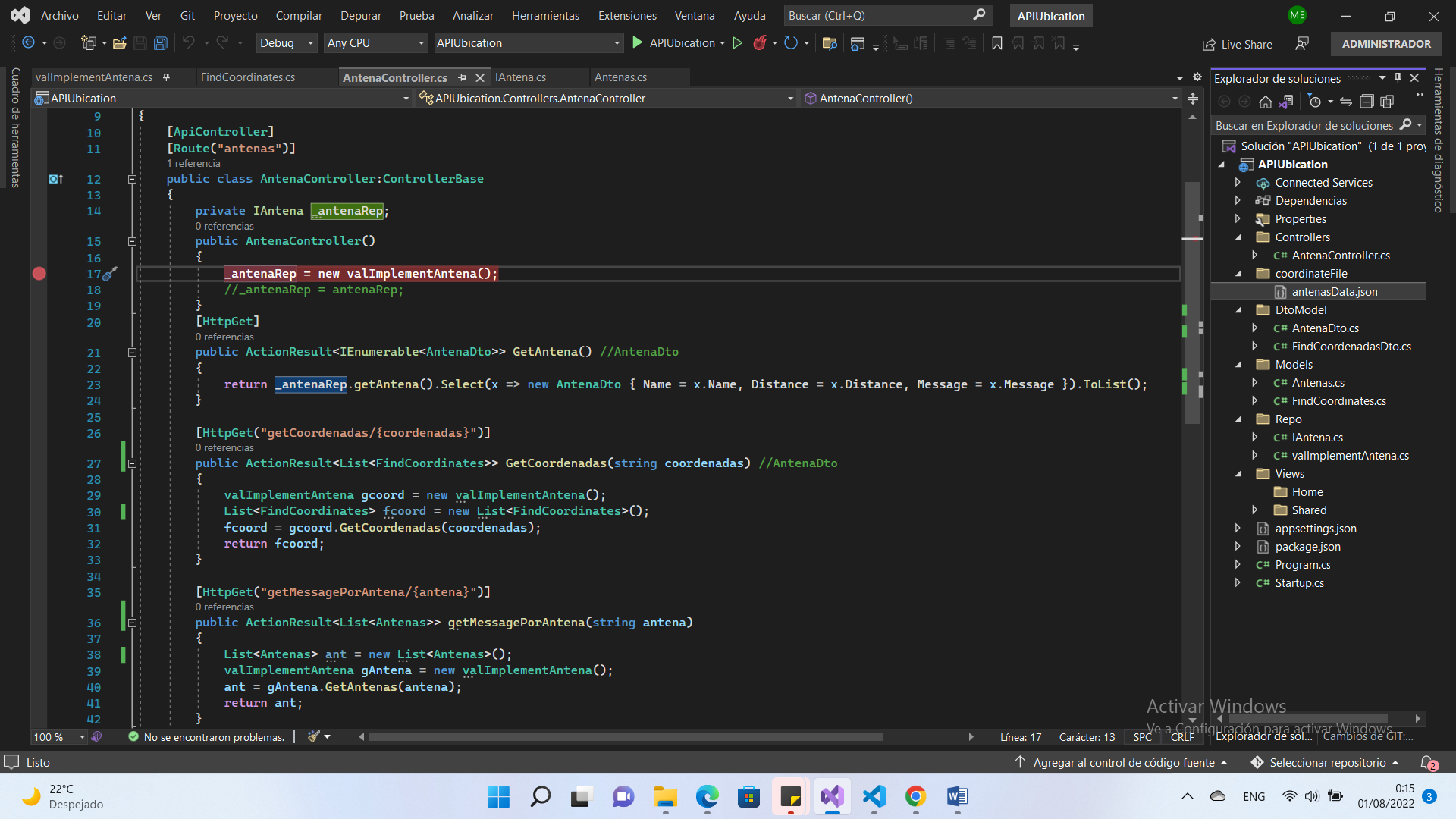
Autor: Ayerve E. Mario Empresa: Wonder Travel

**Documento Tecnico “Finding JP”**

1. **DESARROLLO SERVICIOS WEB / API REST (.NET CORE)**

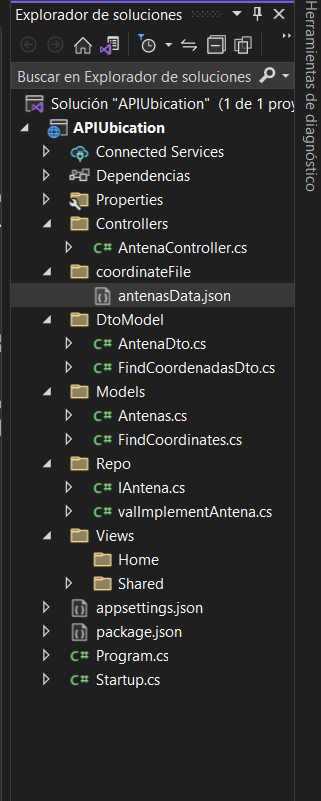
La herramienta utilizada para el desarrollo de ejercicio, es una herramienta de Microsoft conocida como c#, la cual nos provee la oportunidad de desarrollar servicios mediante .net Core, la cual es una herramienta de código abierto y se puede desarrollar proyectos de multiplataforma dirigidas a Windows, Linux y MacOS.



El servicio desarrollado se estructura de acuerdo a los requerimientos solicitados, entre estos tenemos el desglose de los distintos llamados a los servicios.

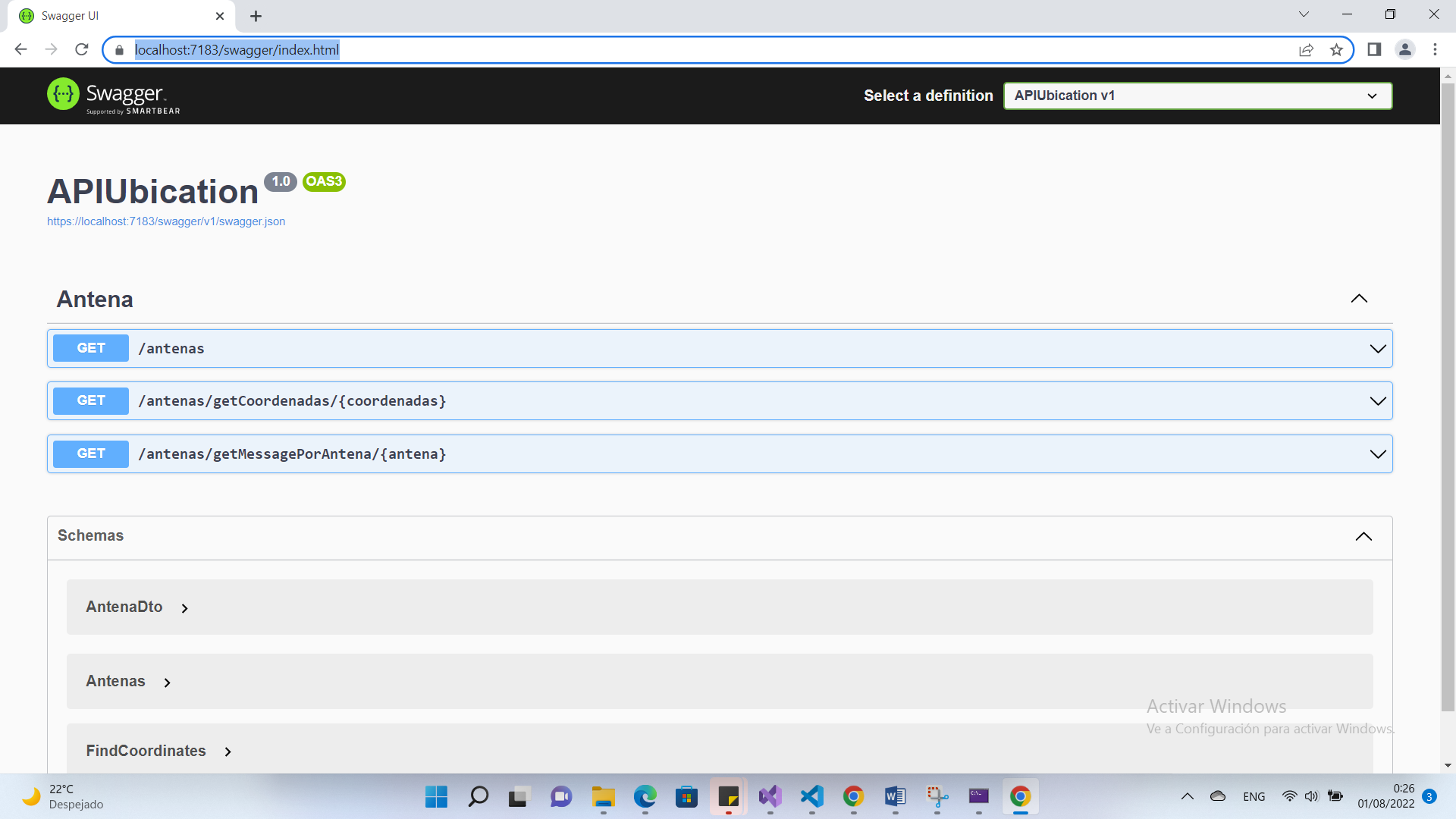
Los conocidos y comúnmente usados son: GET, POST, PUT Y DELETE.

Para este ejercicio, el solo utilizar el método GET es suficiente, debido a que solo obtendremos información, mas no modificaremos ni colocaremos, y no eliminaremos información alguna.



En la estructura podemos notar que nos encontramos de la siguiente manera:

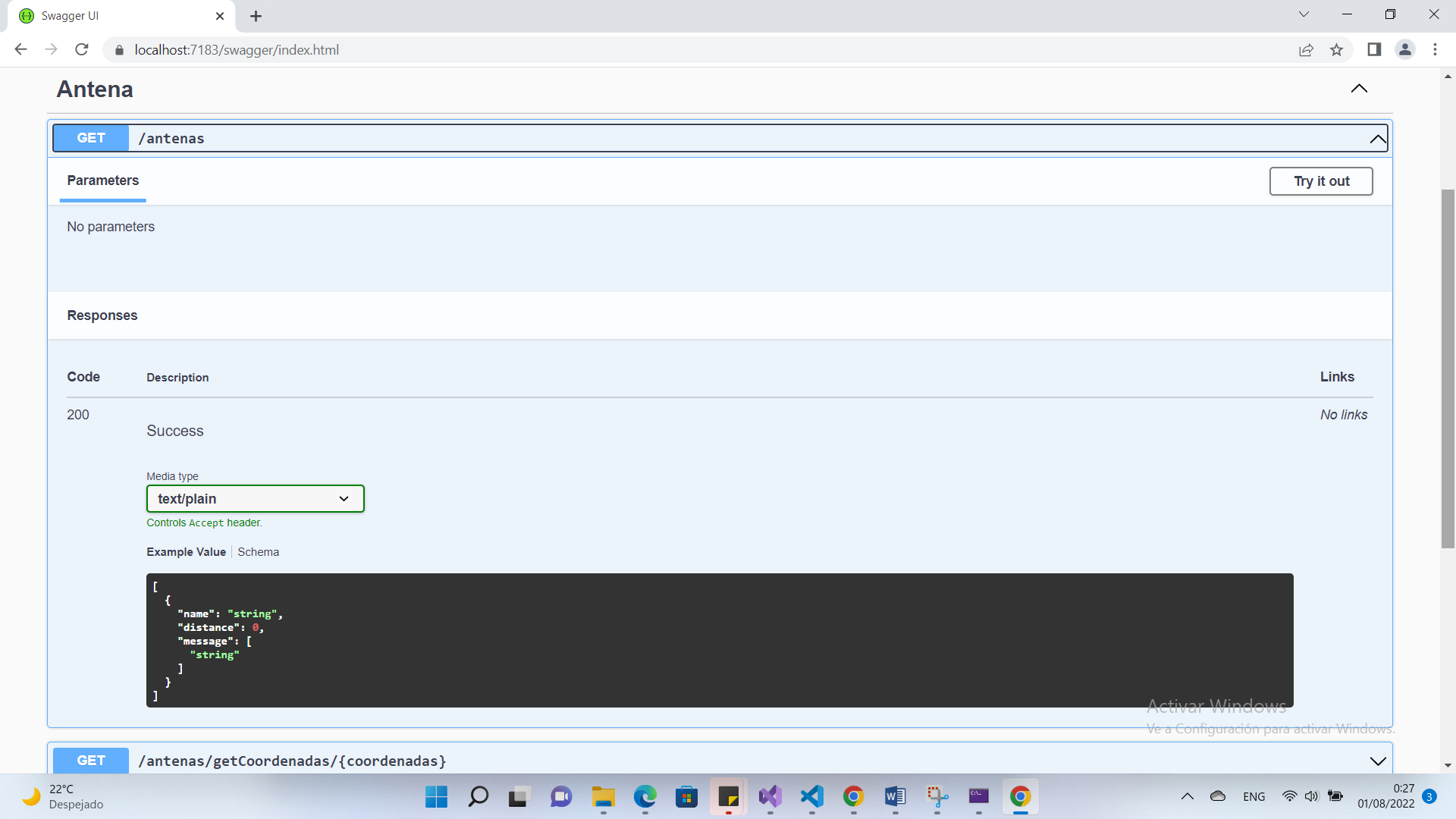
* Controllers: Se especificará los métodos de las llamadas a ejecutar, mas no se tendrá ninguna lógica de desarrollo en esta clase.
* CoordinateFile: En esta carpeta se encuentra el archivo a validar en la prueba, es decir un archivo JSON en donde se encuentra la información que se debe analizar para obtener el resultado final.
* DtoModel: Los archivos en esta carpeta sirven como un mapeo de información para los modelos del proyecto, un canal para obtener la información que se obtendrá desde la capa más baja (llamadas a servidor o traer información de BD), para este caso validar la información del archivo JSON.
* Models: Los modelos son archivos en donde encontramos los modelos de información que se obtendrán de acorde al proceso. Con el fin de definir la información que se va a consultar, insertar, modificar o eliminar.
* Repo: En la carpeta Repo tenemos una interfaz y una clase validadora en donde encontraremos el proceso que se realizó para obtener la información que posteriormente se mostrara al cliente.

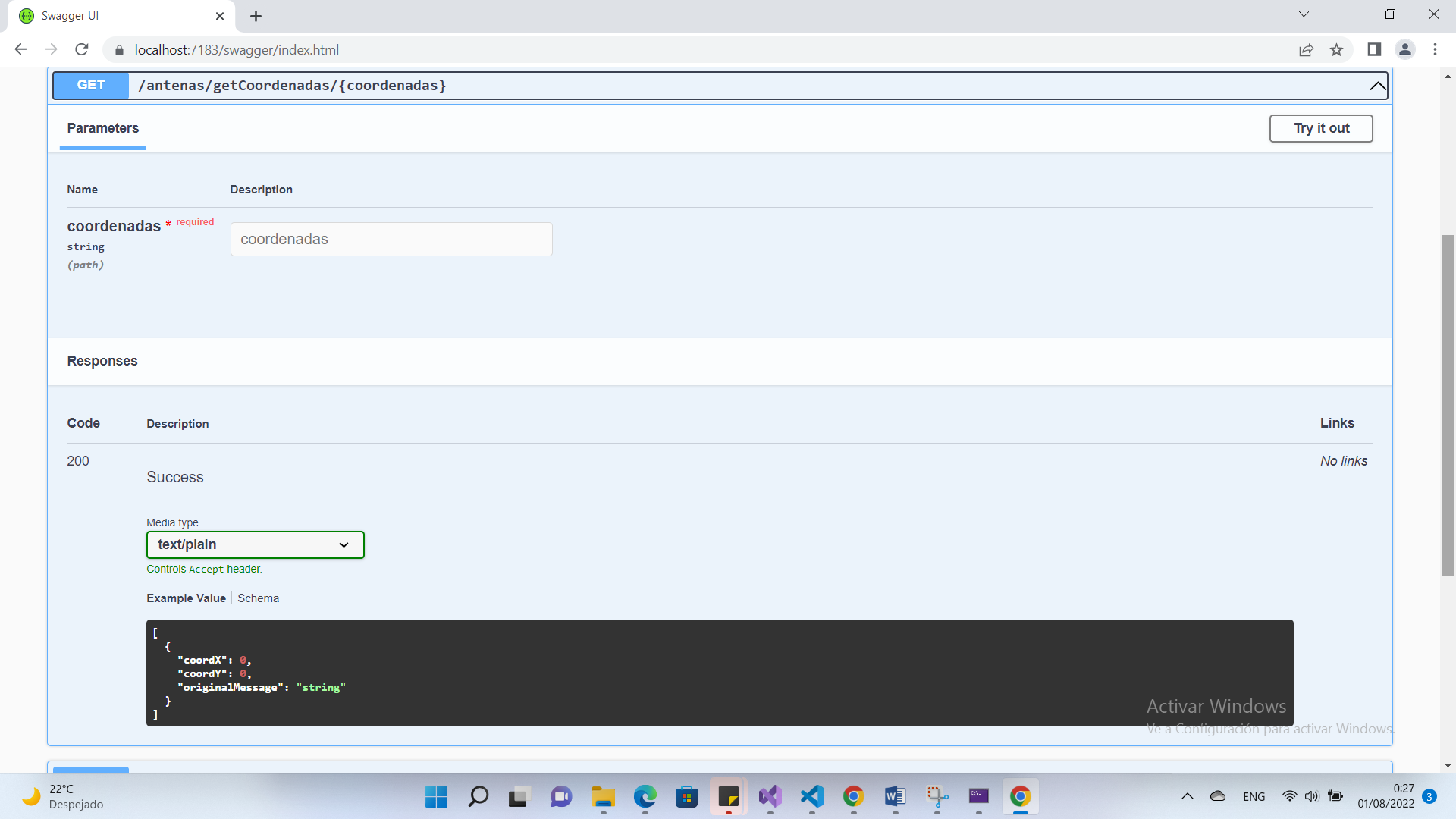


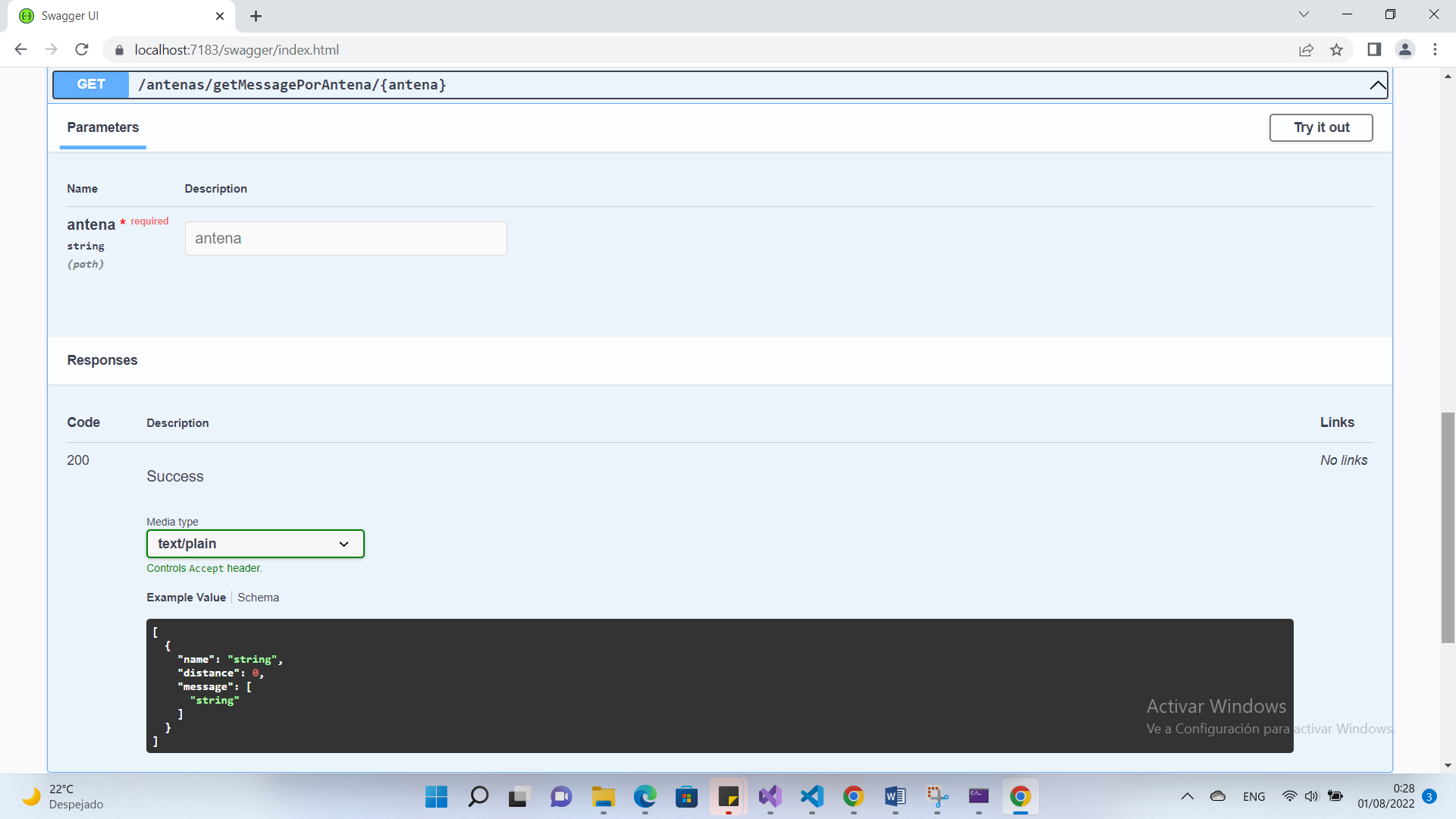
En la imagen anterior podemos ver como se encuentran los servicios una vez que se ejecute el proyecto.

Esta pantalla indica que los procesos se encuentran operativos y listos para utilizarse.

Cada opción cuenta con un ejemplo de que requiere cada servicio y como se debe enviar la información.

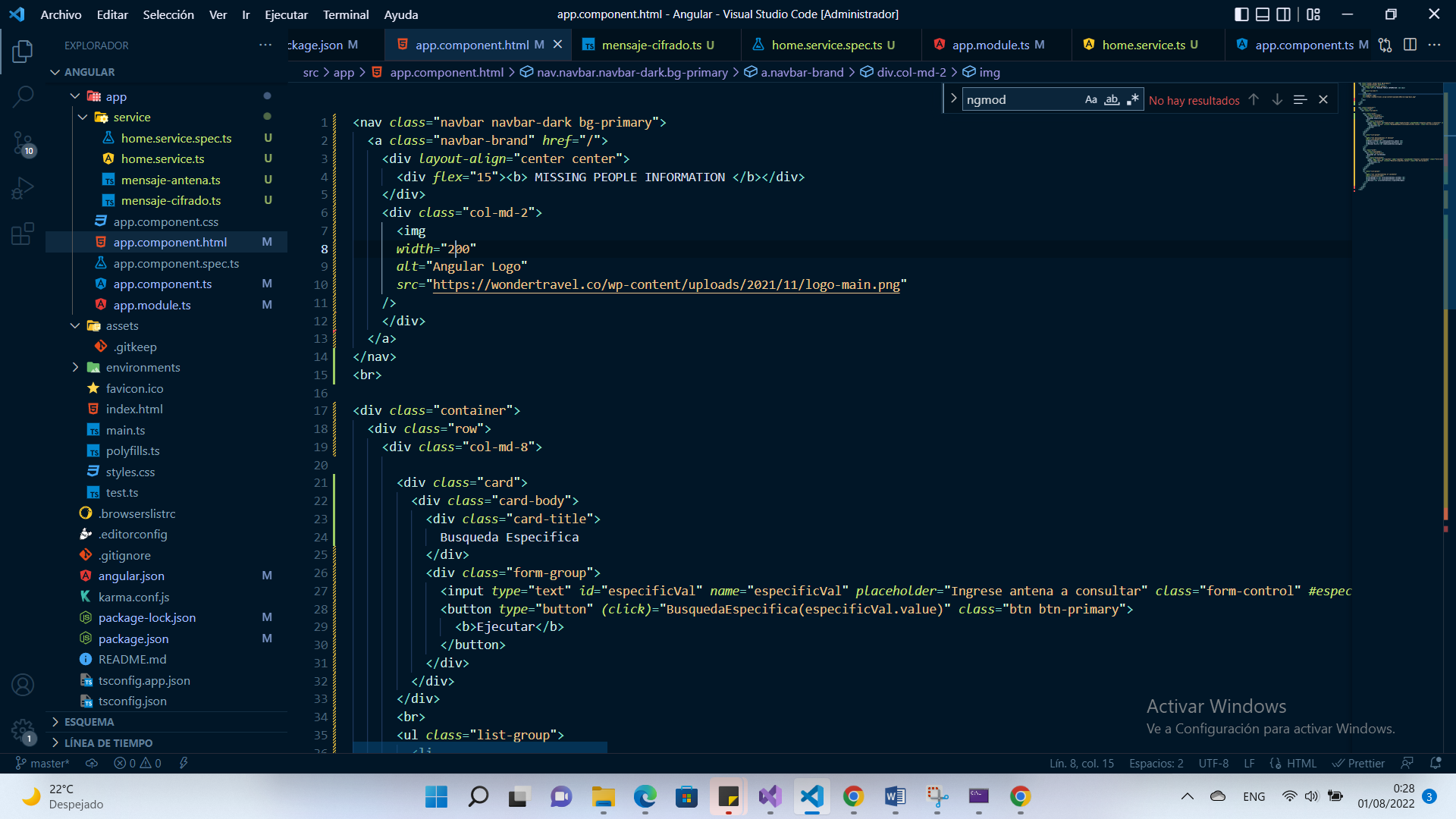






1. **DESARROLLO FRONT END / BACK END (ANGULARJS / TYPESCRIPT)**

La herramienta utlizada para desarrollar el front end y presentar la información que se valido se llama AngularJS, la cual es una multi-plataforma que utiliza typescript, una herramienta que tiene una escalabilidad muy alta debido a los componentes que podemos integrar con esta.



AngularJS es una herramienta cuyo objetivo es desarrollar páginas web, de una forma rápida, con estilos únicos y adaptada a cambios de diseño.

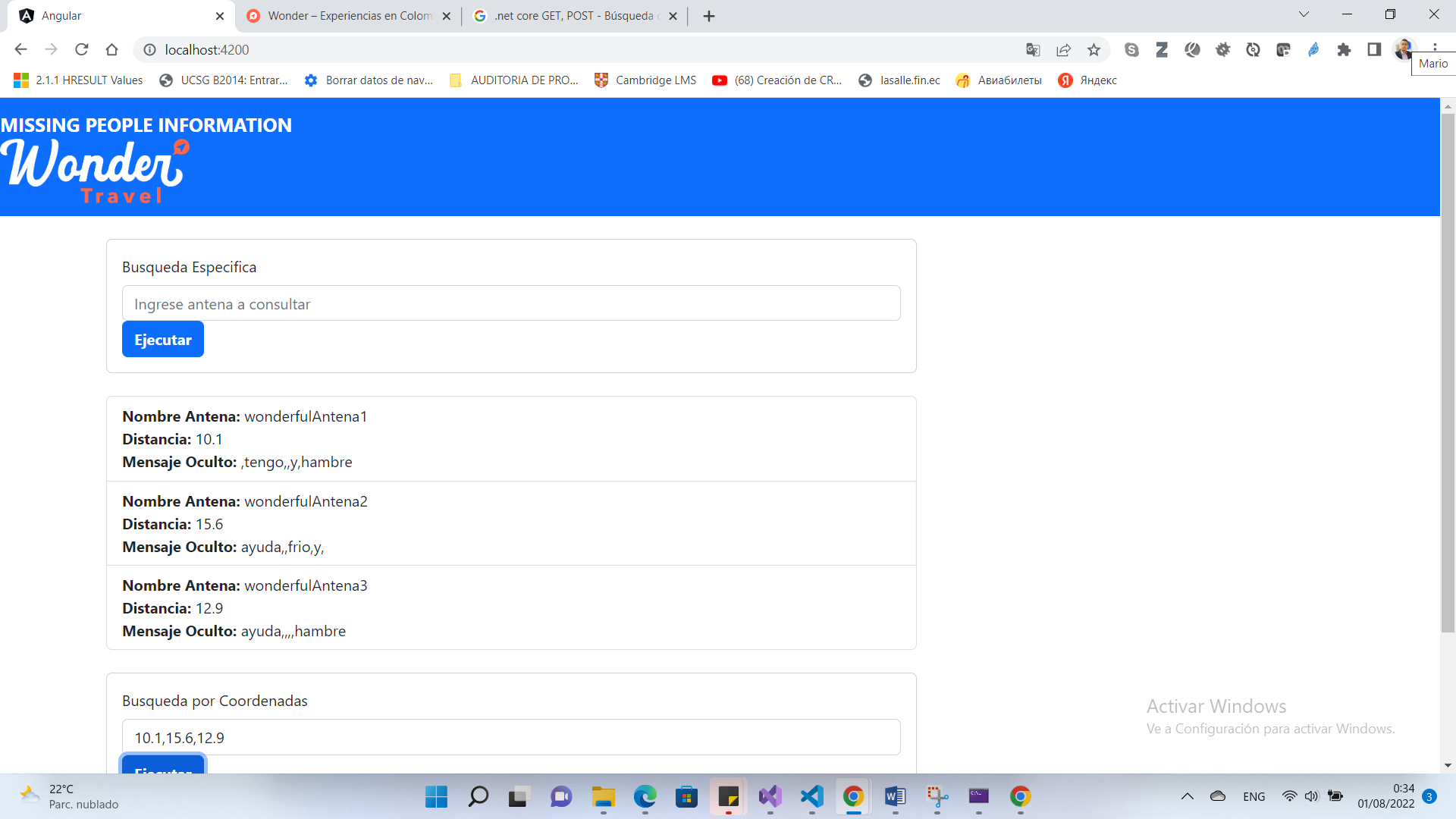
Su estructura pareciera que es una herramienta difícil o complicada de usar, pero realmente nos ahorramos código y tiempo para crear diseños web de gran alcance.

Angular en forma general trabajo solo con:

* Componentes
* Modelos
* Vistar HTML
* TypeScript

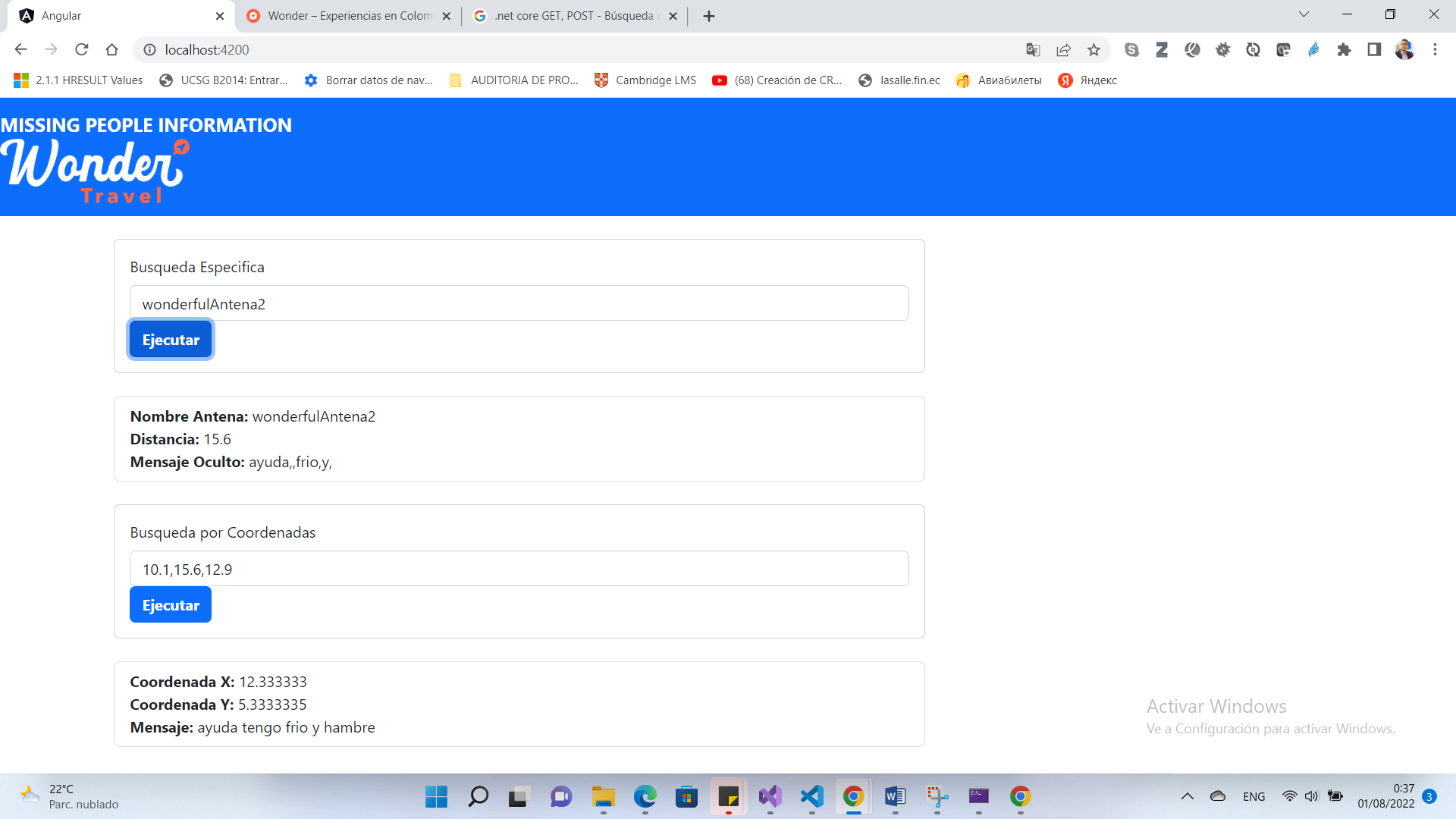
De igual forma se pueden integrar:

* Servicios
* Interfaces
* Otros diseños.
* Conexiones a distintas DB.

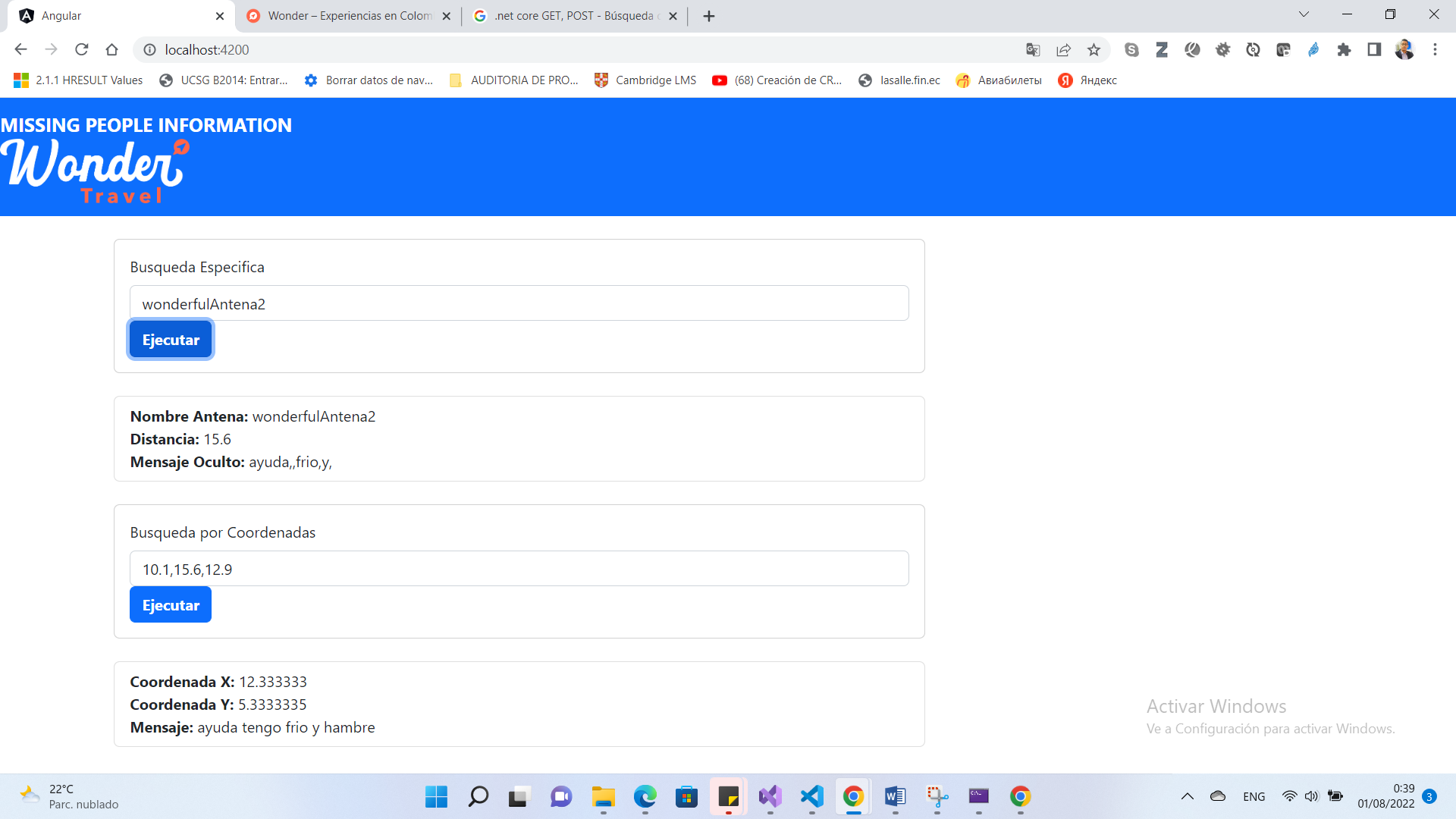


Como presentación final, tenemos que al momento de cargar la página podemos ver cómo nos muestra la información de “JP”, el cual se encuentra perdido, y los mensajes que cada Antena registra sobre su desaparición.

Se cuenta con una opción para realizar una búsqueda específica a la antena que se desea revisar. Una opción útil si en algún momento se tiene más de 100 o 1000 antenas con mensajes.



De igual manera, uno de los requerimientos es poder enviar las coordenadas/distancia de JP, para poder encontrar su ubicación y el mensaje oculto que contiene utilizando las antenas que registraron su desaparición.



1. **PANTALLAS DE DISENO Y PRUEBAS TECNICAS.**

Las pantallas se validaron de tal forma que exista un evento para cada escenario (correcto e incorrecto).

Las validaciones de los servicios se desarrollaron por medio del SWAGGER, de igual forma cada proceso cuenta con un try catch, con seteo por defecto, validaciones de NULL o En BLANCOS.

Existen diversas formas de obtener información, pero la más óptima es una estructura en donde se pueda validar por partes y resolver de la mejor manera.

Muchas Gracias.

Ayerve E. Mario