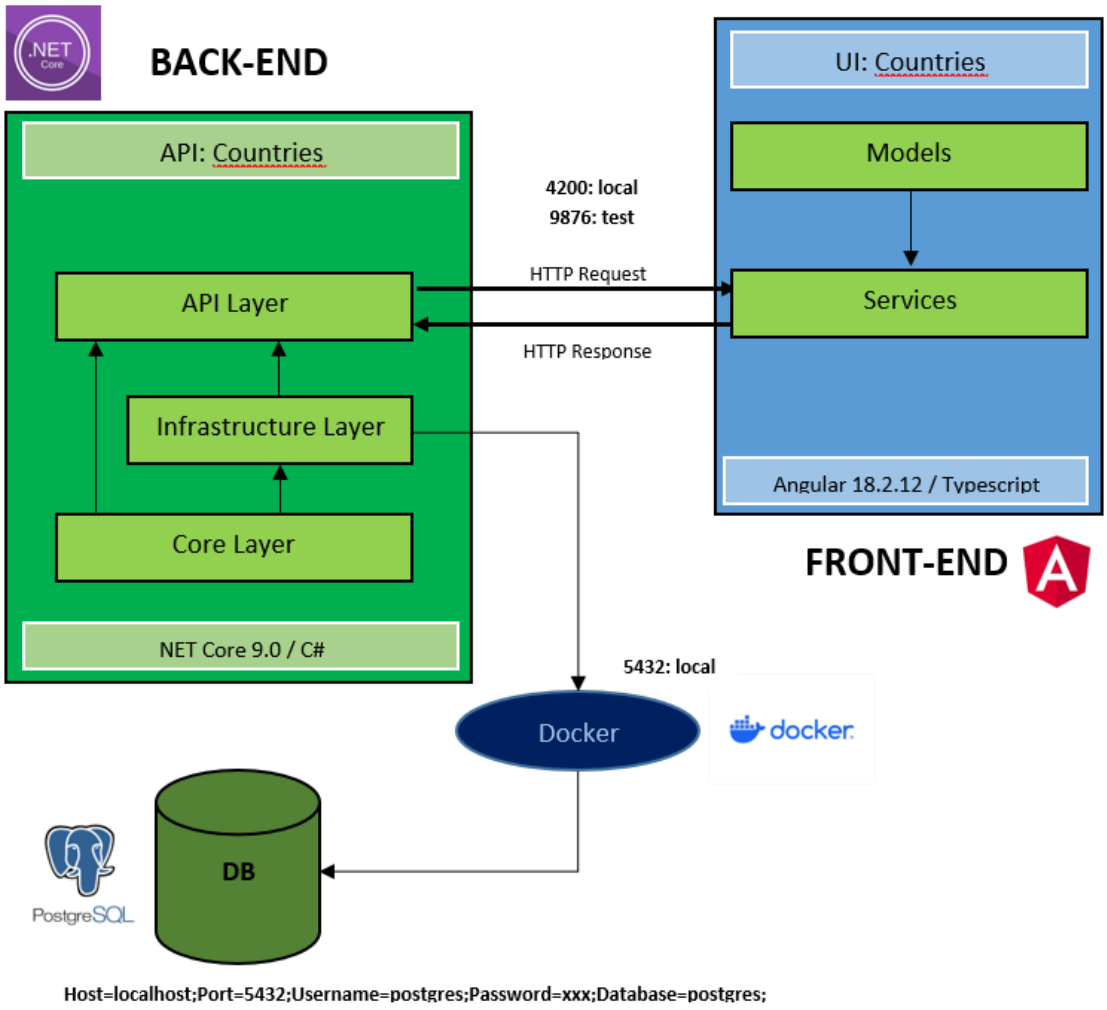
User Manual

Arquitecture:



BACKEND:

An ONION-type architecture is chosen for the backend, with the idea of separating responsibilities between layers. NET Core 9.0 is used. The following design patterns are used: Repository / UnitOfWork / BaseEntity.

Repository: Abstracts data access, providing an interface to perform operations without exposing implementation details.

Unit of Work: Coordinates multiple repository operations into a single transaction, ensuring consistency.

BaseEntity: involves creating a base class that contains properties common to all entities in the data model, such as a unique identifier (Id).

These patterns help to keep code clean, facilitate testing and improve maintainability.

FRONTEND:

On the front end side, models and services are used, with the idea of taking API responses and loading them into the Angular data model. Angular version 18.2.12 is used.

In this application I tried to use the concept of parent/child component.

parent component:

show-master-info

child components:

show-info-map
show-info-flag
show-info-country
show-info-coat-of-arms
show-info-anthems

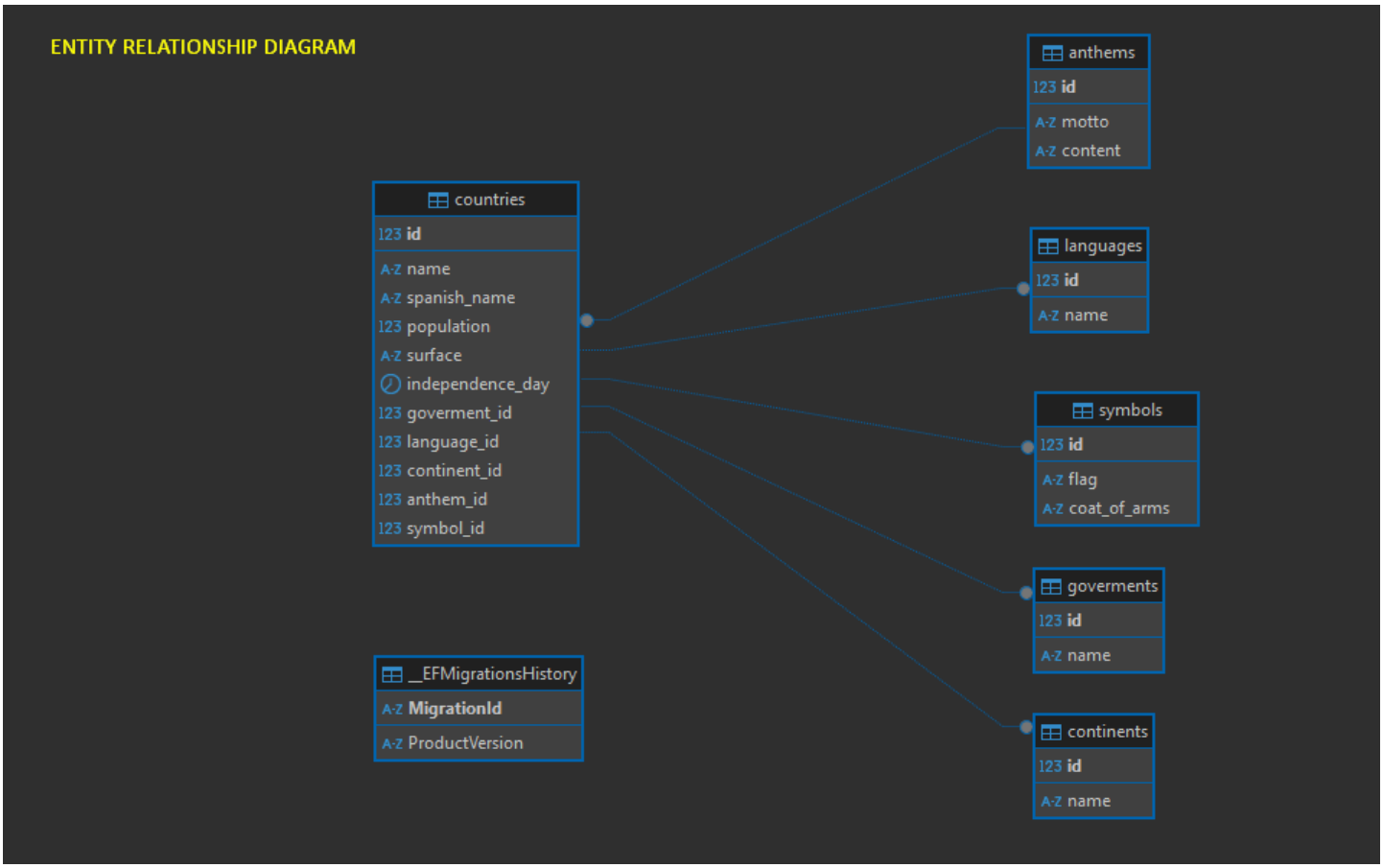
the idea is to practice this concept in a way to separate functionality and not to keep everything in one component. So that:

<table>
<tr><td rowspan="2">FATHER COMPONENT

show-master-info</td><td>show-info-map</td><td>show-info-flag
show-info-coat-of-arms</td></tr>
<tr><td>show-info-map</td><td>show-info-anthems</td></tr>
</table>

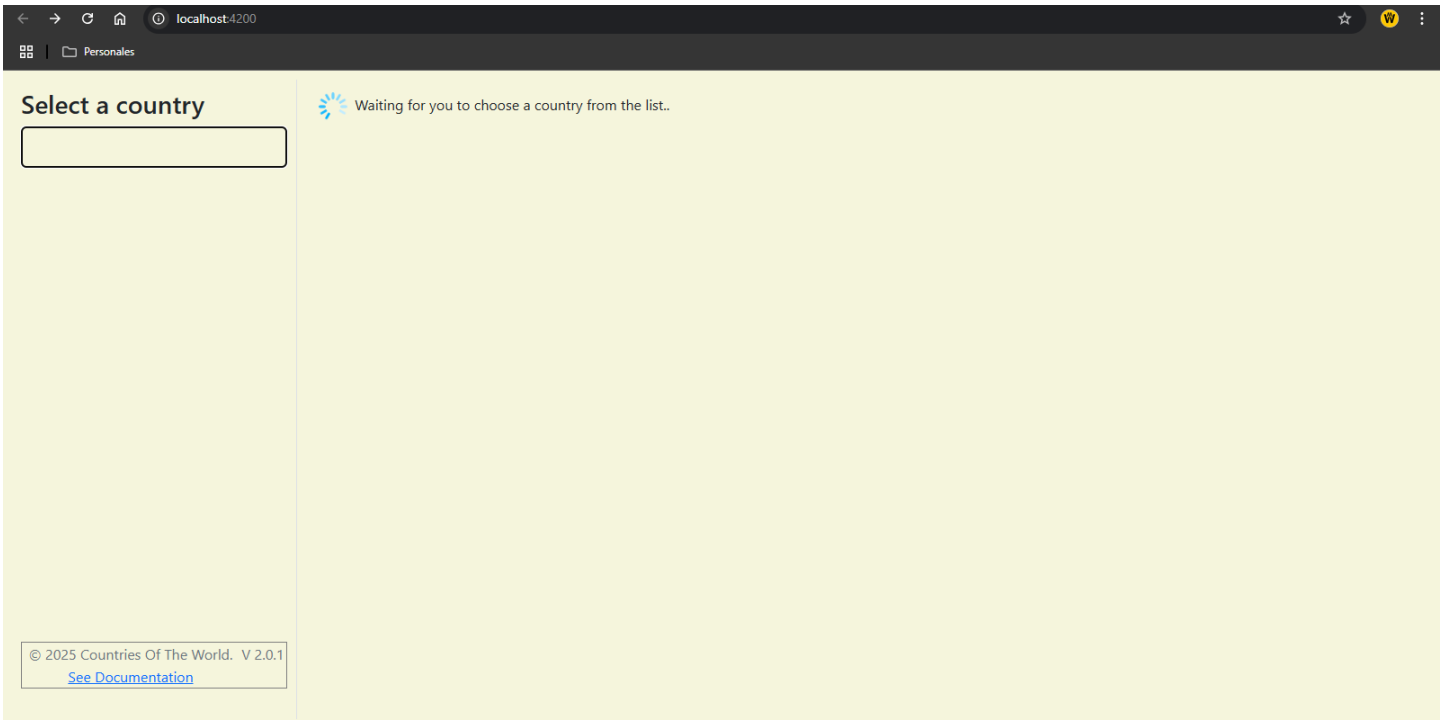
Database:

At database level PostgreSQL is used, to work with this database engine Docker is used.



Application operation

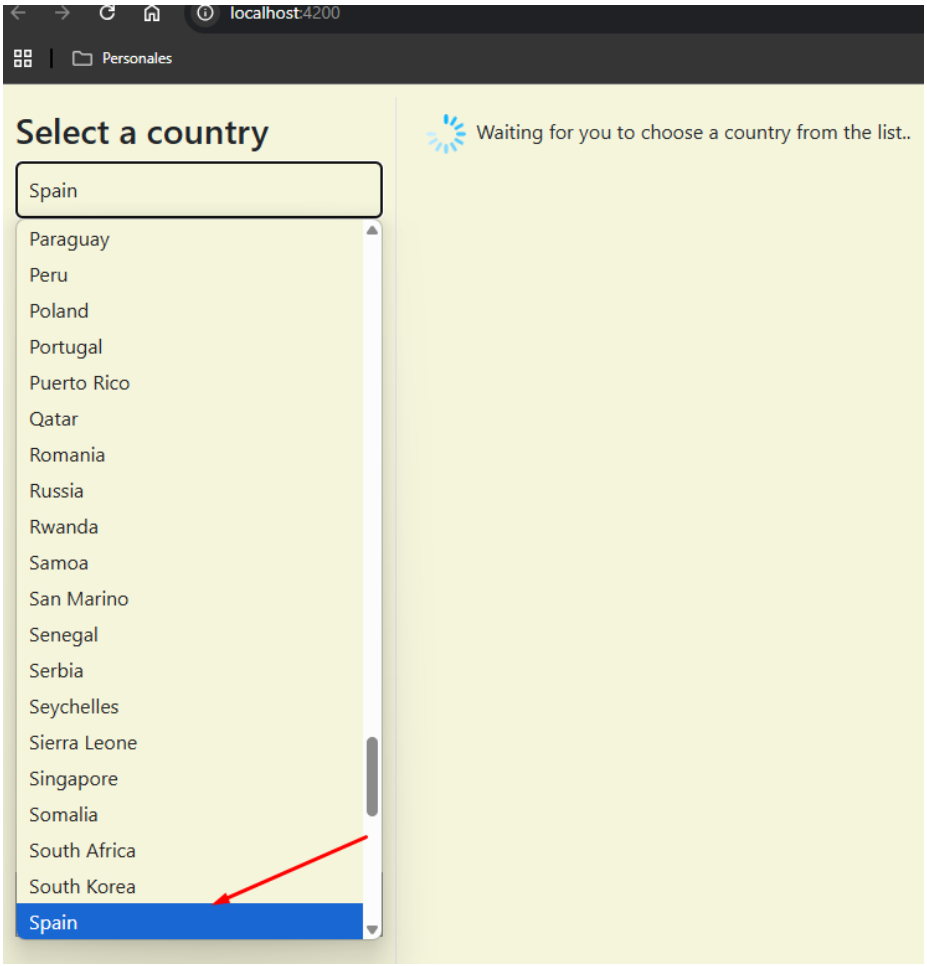
If everything is configured ok, the api is up and the Docker image is running, we would see like this:



If the api fails or Docker is not up, the program will not be able to load the country select, and a message will be displayed.

Selecting a country

Select a country from the list: e.g. Spain



The data of the country will be shown: general information (population, surface, independence, continent, language, form of government), then you will see its symbols (flag, coat of arms). Continent, language, form of government, then you will see its symbols (flag, coat of arms), a map with the region where it belongs and finally the possibility to listen to the national anthem played by the US Navy Band.

This is all, for the moment, the idea is that each selection brings data from each country and you have to use Angular hooks to be able to update the data: ngOnChanges, ngOnInit, ngOnDestroy