1. **Setting up the sample project:**
2. Clone this repository : <https://github.com/axell9641/mfe-model>
3. Install the dependecies:

*npm i*

1. Run the project:

*ng serve -o*

1. If everything went well, you will be able to see this page:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

1. **Activate and Configure Module Federation**
2. Install the module federation plugin into the micro frontend by running the folowing command:

*ng add @angular-architects/module-federation@12.5.3 --project pokedex --port 5000*

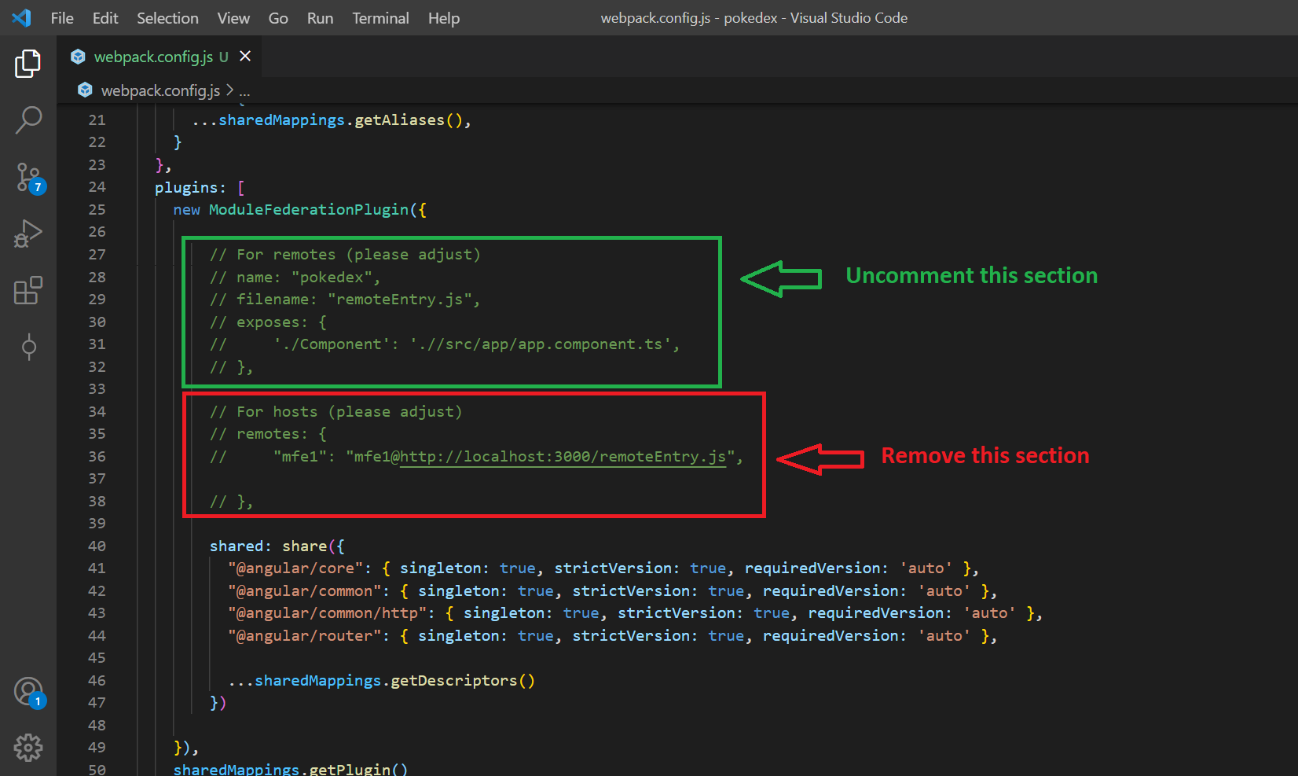
This activates module federation, assigns a port for ng serve (used in local development), and generates the skeleton of the module federation configuration (*webpack.config.js*)

**Note**: The value used as **project** should be the same configured in the package.json file

Texto

Descripción generada automáticamente

1. Switch into the project pokedex and open the generated configuration file : *webpack.config.js*. It contains the module federation configuration for pokedex. Adjust it as follows:



Complete the process by changing the value of the key “./Module” according to the path of the feature module you want to expose.

Captura de pantalla de un celular

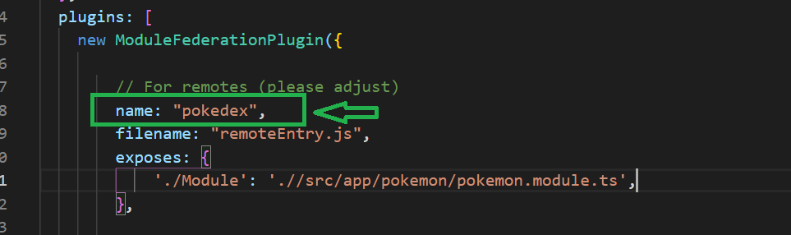
Descripción generada automáticamente

This exposes the feature module, PokemonModule, under the the path “./Module”

1. **Sharing metadata for the provider to load the recently configured MFE:**

Once configured the module federation plugin for the micro frontend, you need to share the following information so that the provider can load it.

1. The base url of your published MFE: Ex: <https://www.pokedex.com>
2. The remote name of the micro frontend: “pokedex”



1. The name of the feature module being exposed: “PokemonModule”

