UNIT 1

JAVASCRIPT

Part 4 - Exercise

Client-side Web Development
2nd course – DAW
IES San Vicente 2025/2026
Arturo Bernal / Rosa Medina

Índice

Exercise JavaScript part 5 3
NPM and Vite 3
Showing a map 3

Exercise JavaScript part 5

In this exercise, you'll have to adapt the previous exercise to NPM, Vite and add a couple of features. **Don't include the node_modules directory** in the activity submission file.

NPM and Vite

- Create a Vite project (Vanilla JavaScript) and place the previous exercise code there (remove the automatically generated code).
  - I'm giving you index.html and add-property.html files again because of some minor changes explained later
- Install **tailwind** and **@tailwindcss/vite**
  - Importing the library from an external CDN is now disabled. It's now imported using @import tailwind (see HTML files).
  - Add the necessary configuration to the **vite.config.js** file so that tailwind can generate the necessary CSS:

```
...
import tailwindcss from '@tailwindcss/vite'

export default defineConfig({
  ...
  plugins: [
    tailwindcss(),
  ],
})
```

- Install Openlayers library (**ol**)
  - The file new_property.html imports its CSS and needs it. In the next section it's explained how to use it in this project.
- Create a new ESLint configuration (using **npm init**)
  - Select the "To check syntax and find problems" option
  - Add some rules like camelcase, curly, and eqeqeq. Those rules will give an error if not followed (https://eslint.org/docs/latest/rules/).
- Rename the script dev to start,
  - Create a test script to run eslint and execute it always before start.
- Add both index.html and new-property.html files to vite.config.js. When

running npm run build, both should be present in the dist/ folder.

Showing a map

In the **add-property.html** page, show a **map** inside the div#map element Initially geolocate the user and show his/her position (add a marker too).

I'm giving you a class (**MapService**) with methods to create a map (constructor) and creating a marker (createMarker). Use them.

There's also a class named MyGeolocation with an static method to get your geolocation coordinates {latitude, longitude} in a promise.

Each time the user selects a different town, center the map and move the marker to that town's coordinates. **Hint**: Save the towns in a global array when getting them from the server, and search the selected id to get latitude and longitude.