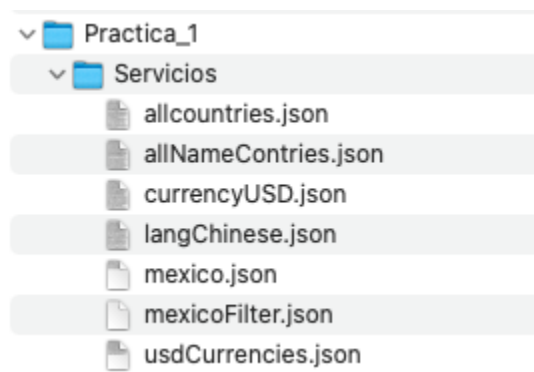


Practica 2 - Aprovisionamiento del entorno Git en Github

En este ejercicio crearemos un repositorio en **Github** (<https://github.com>) , y subiremos nuestras prácticas.

Objetivo 1: Organización del proyecto

Con los archivos generados en la práctica anterior, que fueron guardados vamos a ponerlos la ruta **Practica_1/Servicios**.



Objetivo 2: Generando un repositorio

NOTA: Es importante que en este punto ya tengan una cuenta en Github.

- Ir a <https://github.com/new>

Creemos un nuevo repositorio, con las propiedades:

1. **Repository Name:** uvm_frontend_modulo4
2. **Descripción:** MÓDULO 4: WEB APIS Y APLICACIONES ASÍNCRONA
3. **Repositorio:** Public
 - a. En la imagen de abajo aparece la opción de privado, pero debe de ser pública.
4. **Marcamos la opción:** Add a README file.
5. **Seleccionamos el template:** NODE para el .gitignore

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Required fields are marked with an asterisk (*).

Owner *

 lubelmont ▾

Repository name *

uvm_frontend_modulo4

✓ uvm_frontend_modulo4 is available.

Great repository names are short and memorable. Need inspiration? How about [probable-doodle](#) ?

Description (optional)

MODULO 4: WEB APIS Y APLICACIONES ASINCRONAS

☐



Public

Anyone on the internet can see this repository. You choose who can commit.

☒



Private

You choose who can see and commit to this repository.

Initialize this repository with:

☒

Add a README file

This is where you can write a long description for your project. [Learn more about READMEs](#).

Add .gitignore

.gitignore template: Node ▾

Choose which files not to track from a list of templates. [Learn more about ignoring files](#).

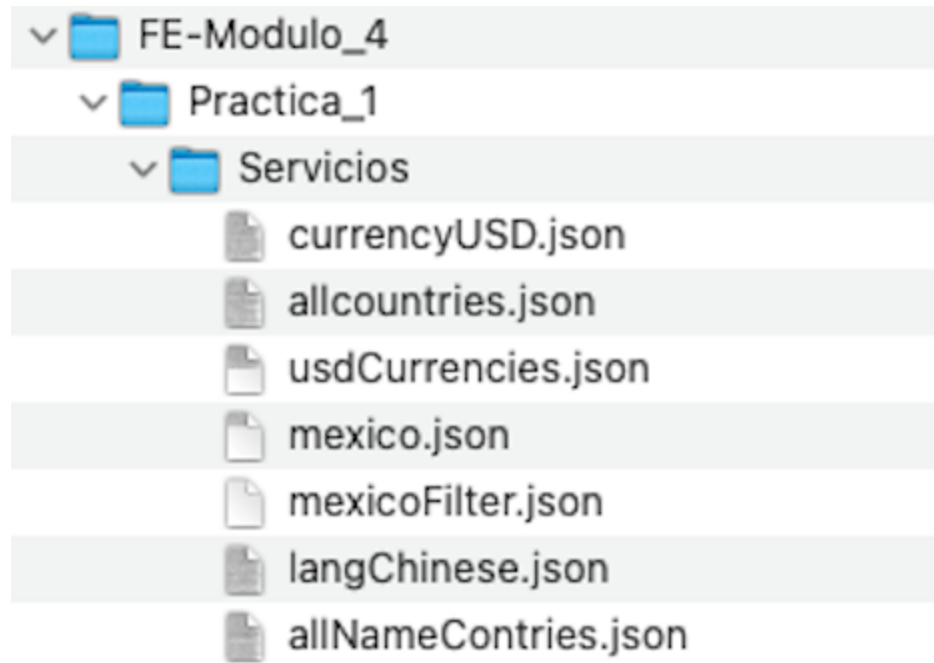
Objetivo 3: Hacer PUSH a nuestro repositorio.

Ahora con el repositorio creado y nuestros archivos listos, vamos a hacer “push” al repositorio.

Lo primero que debemos hacer es crear la conexión entre el repositorio en Github y nuestro respectivo repositorio.

NOTA: Para este punto ya deben de tener los permisos necesarios para poder escribir desde su repositorio a Github.

NOTA 2: Verificar que la estructura del proyecto tenga la estructura como en la imagen de abajo.



Pasos para sincronizar repositorios

1. Dentro de la carpeta raíz (FE-MODULO_4) iniciamos un nuevo repositorio Git

```
git init
```

2. Agregamos todos los cambios que tengamos en nuestro proyecto.

```
git add .
```

3. Creamos un nuevo Commit con nuestros cambios y agregamos un mensaje

```
git commit -m "Practica 1"
```

4. Creamos el vínculo entre repositorios. **IMPORTANTE** sustituir `<URL-del-repositorio>` por su repositorio,
 - a. ejemplo de URL git@github.com:lubelmont/uvm_frontend_modulo4.git

```
git remote add origin <URL-del-repositorio>
```

5. Hacemos el seguimiento de las ramas entre la local y la de github

```
git branch --set-upstream-to=origin/main main
```

6. Descargamos los archivos que están en el repositorio remoto

```
git pull
```

7. Cargamos los archivos que incluimos en el commit

```
git push
```