



# SPA to PWA

Single Page Application hasta Progressive Web Application

29 marzo 2023

## Descripción del Cloud Academy

Genera tú solución web con SPA (Single Page Application) y SWA (Azure Static Web Application) hasta despliegue como una Progressive Web App integrando herramientas de nivel empresarial sin ningún esfuerzo adicional. Tu solución web contará con ciclo de CI/CD, SSL, distribuido globalmente, seguro, routing, API y más.

## Agenda

HORARIO	ACTIVIDAD
10:00 a 10:20	Introducción
10:20 a 10:30	Single Page Application (SPA)
11:40 a 12:00	Break
12:00 a 12:10	Static Web Apps
13:00 a 14:00	Comida
14:00 a 15:30	Static Web Apps
15:30 a 16:30	Progressive Web apps
16:30 a 17:30	Conclusiones y cierre

# Contenido

---

Descripción del Cloud Academy .....	2
Agenda.....	2
1. Herramientas .....	5
Prerequisitos .....	<b>Error! Bookmark not defined.</b>
Instalación Node.js y NPM .....	5
Visual Studio Code.....	7
Extensiones para VS Code .....	7
Static Web Apps.....	7
Functions.....	8
Vue.js .....	10
Functions Core Tools .....	11
Git.....	14
Vue .....	15
2. Single Page Application (SPA) .....	17
¿Qué es Single Page Application? .....	17
Desplegar el código vía Vue .....	18
Ejecutar la aplicación.....	18
Exploración del SPA.....	26
Solo una página index.html.....	27
Carga de App .....	28
Carga de Route .....	29
Agregar una nueva Vista .....	30
Agregar la ruta a nueva vista.....	30
Prueba de la Vista.....	31
3. Integración Continua y Despliegue Continuo .....	32
Opción 1: Github .....	33
Opcion 2: Azure Dev Ops (ADO).....	34
4. Azure Static Web Apps (SWA).....	39
¿Qué es Azure Static Web Apps? .....	39

Despliegue de la primera SWA.....	41
Routing de SWA.....	47
Crear un API en SWA .....	49
Configurar carpeta API para el SWA.....	51
Agregar un elemento de prueba .....	52
Autenticación de SWA.....	53
Acceso de usuario en API .....	56
Roles .....	57
Integración con una Base de datos .....	59
Environments .....	72
Configurar Dominio .....	74
5.    Progressive Web App (PWA).....	78
¿Qué es Progressive Web App? .....	78
Actualización de SPA a PWA.....	79
Revisión de componentes de PWA .....	80
Despliegue con Builder.....	88

# 1. Herramientas

## Prerrequisitos

Las siguientes herramientas se requieren durante este Cloud Academy:

- Node.js
- Visual Studio Code
- Git
- Vue
- Function Core Tools

## Instalación Node.js y NPM

Descargar la versión de Node.js descargando de la URL [Node.js \(nodejs.org\)](https://nodejs.org)



### Downloads

Latest LTS Version: 18.15.0 (includes npm 9.5.0)

Download the Node.js source code or a pre-built installer for your platform, and start developing today.

LTS Recommended For Most Users	Current Latest Features
 Windows Installer node-v18.15.0-x64.msi	 macOS Installer node-v18.15.0.pkg
 Source Code node-v18.15.0.tar.gz	
<a href="#">Windows Installer (.msi)</a>	32-bit      64-bit
<a href="#">Windows Binary (.zip)</a>	32-bit      64-bit
<a href="#">macOS Installer (.pkg)</a>	64-bit / ARM64
<a href="#">macOS Binary (.tar.gz)</a>	64-bit
<a href="#">Linux Binaries (x64)</a>	64-bit
<a href="#">Linux Binaries (ARM)</a>	ARMv7      ARMv8
<a href="#">Source Code</a>	node-v18.15.0.tar.gz

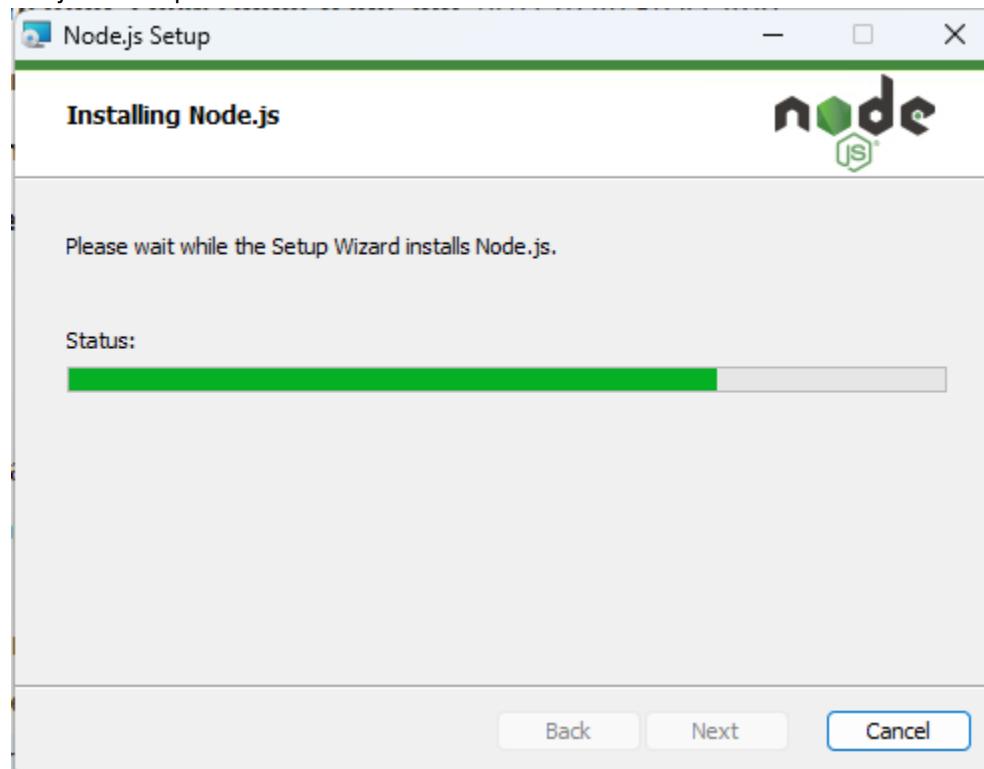
### Additional Platforms

Docker Image
<a href="#">Linux on Power LE Systems</a>
<a href="#">Linux on System z</a>
<a href="#">AIX on Power Systems</a>

Official Node.js Docker Image
64-bit
64-bit
64-bit

- Signed SHASUMS for release files ([How to verify](#))

Se ejecuta la aplicación de instalación:



Se valida la instalación por medio de ejecutar en una línea de comando (cmd) los siguientes comandos:

A screenshot of a Microsoft Windows Command Prompt window titled "Command Prompt". The window shows the command line interface with the following text:

```
node -v
npm -v
```

```
Microsoft Windows [Version 10.0.22621.1265]
(C) Microsoft Corporation. All rights reserved.

C:\Users\artmej>node -v
v18.15.0

C:\Users\artmej>npm -v
9.5.0

C:\Users\artmej>
```

## Visual Studio Code

VS Code se instalará desde la siguiente URL: <https://code.visualstudio.com/Download>

The screenshot shows the official Visual Studio Code download page at <https://code.visualstudio.com/Download>. At the top, there's a navigation bar with links to Docs, Updates, Blog, API, Extensions, FAQ, and Learn. A search bar and a 'Download' button are also present. Below the header, a message states 'Version 1.76 is now available! Read about the new features and fixes from February.' The main content area is titled 'Download Visual Studio Code' and includes the subtext 'Free and built on open source. Integrated Git, debugging and extensions.' Below this, there are three large download sections: one for Windows (Windows 8, 10, 11), one for Linux (Debian, Ubuntu, Red Hat, Fedora, SUSE), and one for Mac (macOS 10.11+). Each section contains a logo, a download button, and a table of available installers for different architectures (x64, x86, Arm64, Arm32).

Plataforma	Arquitectura	Instalador
Windows	x64	User Installer
	x86	System Installer
	Arm64	.zip
Linux	x64	.deb
	x86	.rpm
	Arm64	.tar.gz
Mac	x64	.zip
	x86	Intel chip
	Arm64	Apple silicon

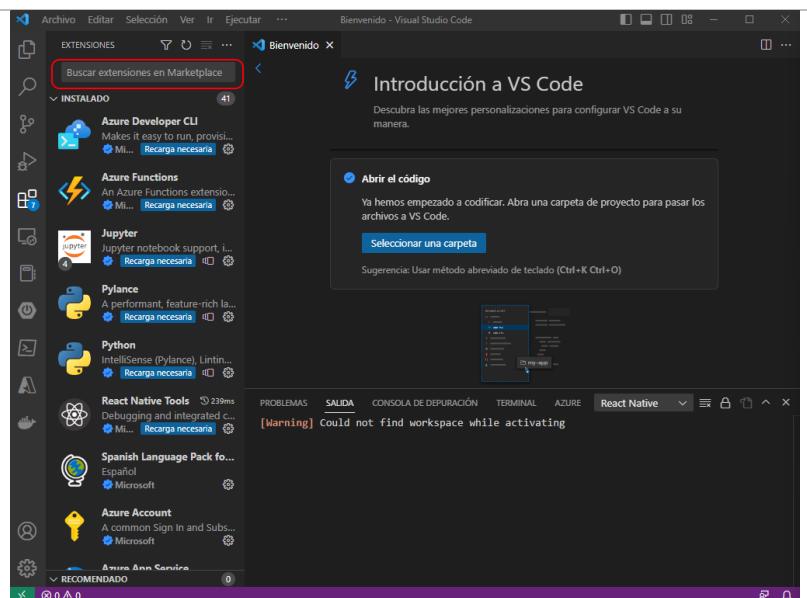
## Extensiones para VS Code

### Static Web Apps

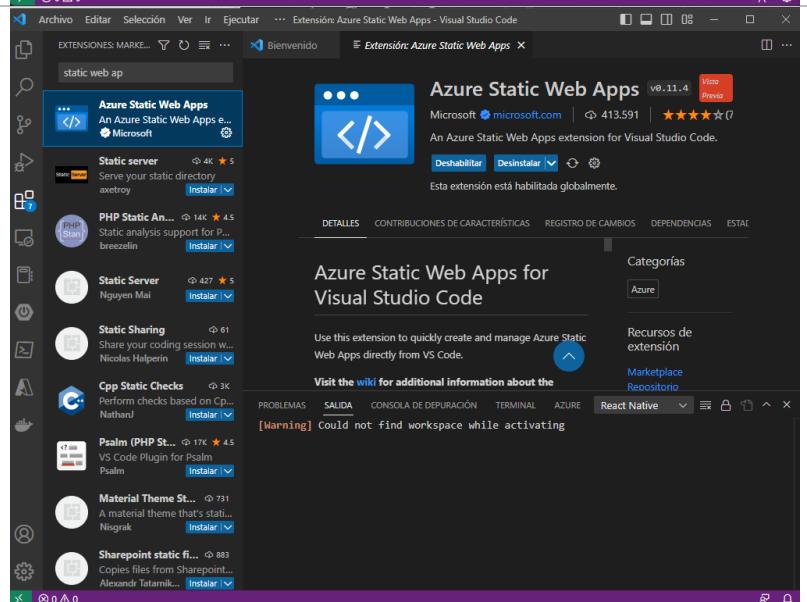
Ir la opción de Extensiones:

The screenshot shows the Visual Studio Code interface with the 'Extensions' view open. On the left, the Explorer sidebar shows a message 'NO HAY NINGUNA CARPETA ABIERTA'. In the center, the 'Bienvenido' (Welcome) panel is displayed, featuring an 'Introducción a VS Code' (Introduction to VS Code) section with a 'Abrir el código' (Open code) button. The bottom status bar indicates a warning: '[Warning] Could not find workspace while activating'. The bottom right corner of the status bar shows the page number '7'.

Ingresar en el espacio de búsqueda en la parte superior el texto “static web apps”.

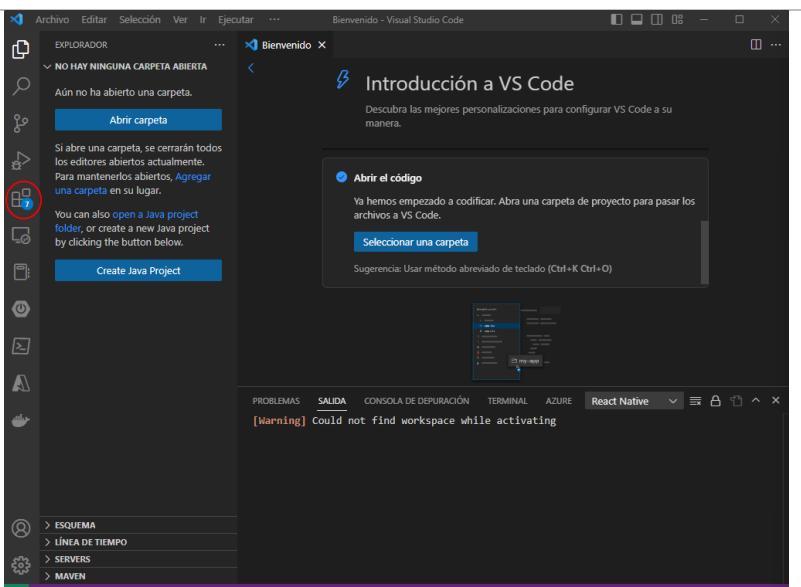


Se da opción de instalar a la siguiente extensión:

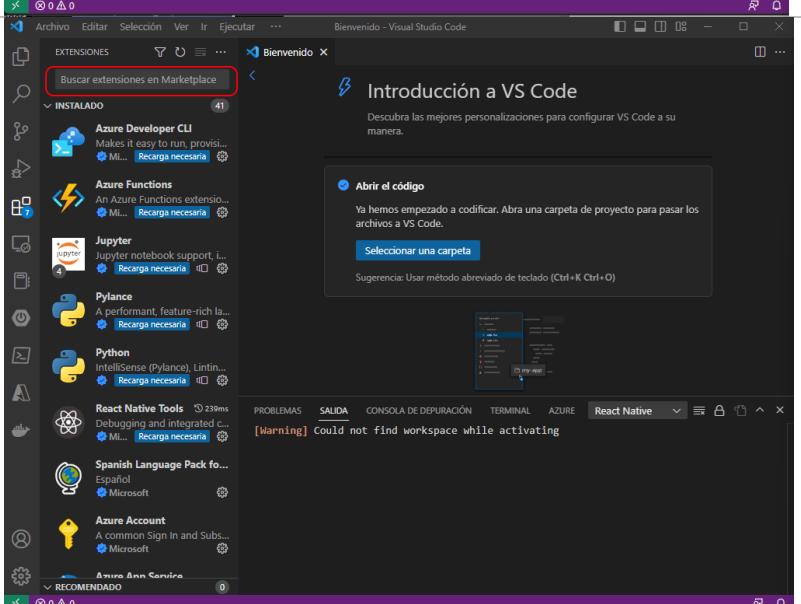


Functions

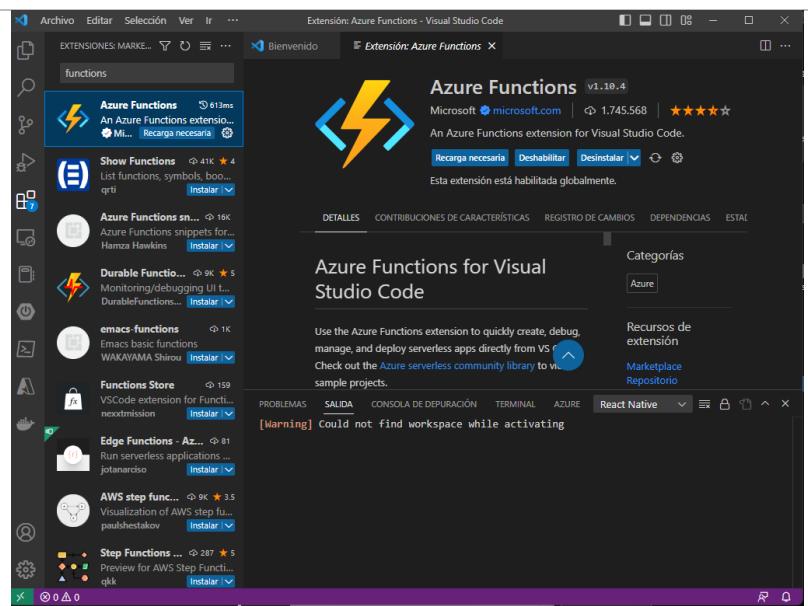
Ir la opción de Extensiones:



Ingresar en el espacio de búsqueda en la parte superior el texto “functions”.

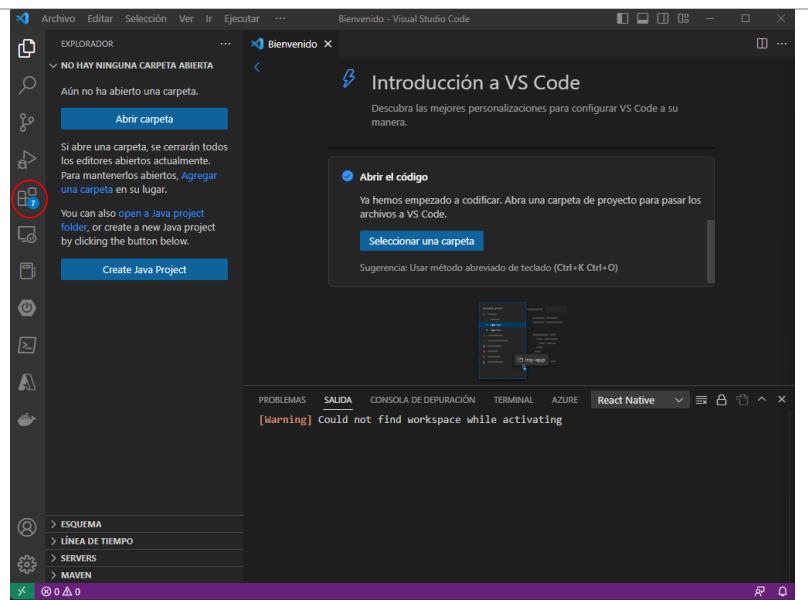


Se da opción de instalar a la siguiente extensión:

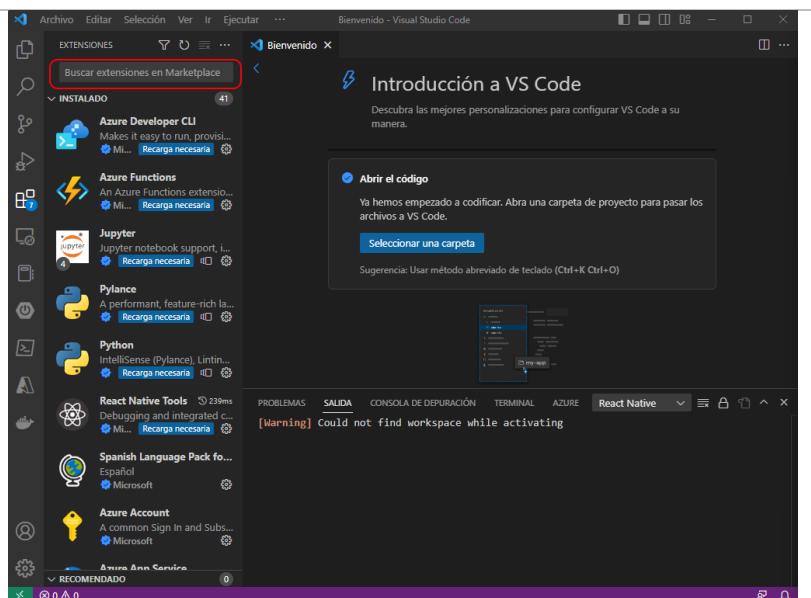


Vue.js

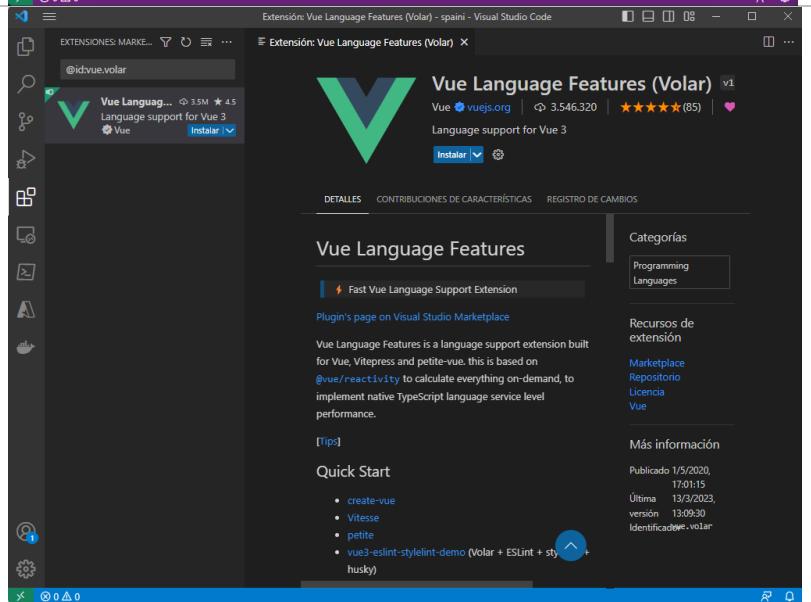
Ir la opción de Extensiones:



Ingresar en el espacio de búsqueda en la parte superior el texto "@id:vue.volar".



Se da opción de instalar a la siguiente extensión:



## Functions Core Tools

Son herramientas que ayudan en el uso de Azure functions en forma local. El primer paso es ir a la URL: <https://learn.microsoft.com/en-us/azure/azure-functions/functions-run-local>

Se descarga la opción de sistema operativo de la maquina local:

The screenshot shows a Microsoft Learn page titled "Work with Azure Functions Core Tools". The page content discusses the Azure Functions Core Tools, which allows development and testing of functions on a local machine. It includes sections on prerequisites, core tools versions, and how to install them. A sidebar on the left provides links for creating functions, connecting to services, debugging, and managing dependencies. The main content area features a "Windows" tab selected, providing instructions for installing the Core Tools on Windows using an MSI installer. It also lists options for macOS and Linux.

**Work with Azure Functions Core Tools**

Article • 02/14/2023 • 23 minutes to read • 55 contributors [Feedback](#)

In this article

- Prerequisites
- Core Tools versions
- Install the Azure Functions Core Tools
- Changing Core Tools versions
- Show 10 more

Azure Functions Core Tools lets you develop and test your functions on your local machine. **Azure Functions Core Tools** includes a version of the same runtime that powers Azure Functions runtime that you can run on your local development computer. It also provides commands to create functions, connect to Azure, and deploy function projects.

Starting with version 2.x, Core Tools runs on [Windows](#), [macOS](#), and [Linux](#).

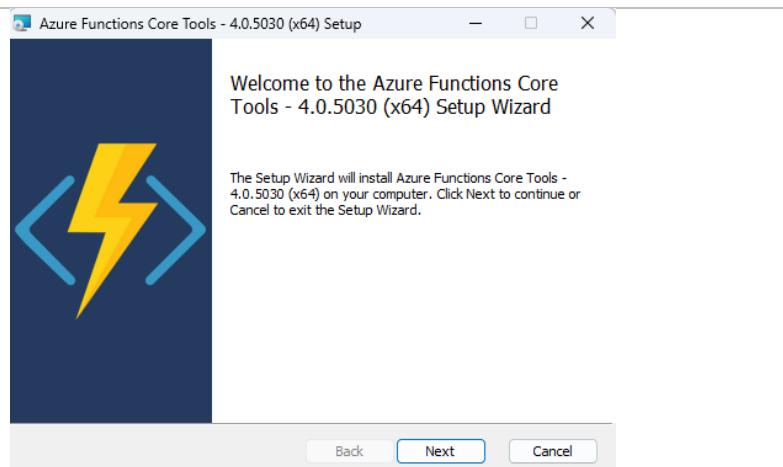
Windows [macOS](#) [Linux](#)

The following steps use a Windows installer (MSI) to install Core Tools v4.x. For more information about other package-based installers, see the [Core Tools readme](#).

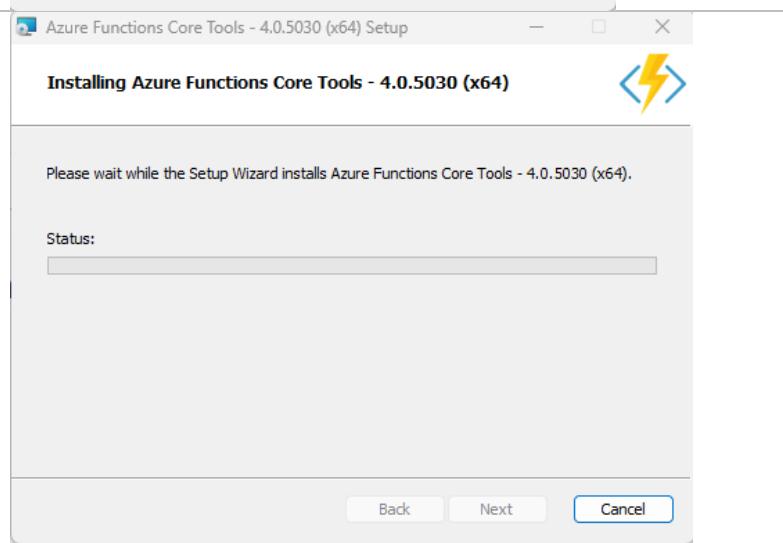
Download and run the Core Tools installer, based on your version of Windows:

- [v4.x - Windows 64-bit](#) (Recommended. [Visual Studio Code debugging](#) requires 64-bit.)
- [v4.x - Windows 32-bit](#)

Se ejecuta la solución de instalación:



Se continúa registrando los datos que se piden durante la instalación hasta concluir el proceso:

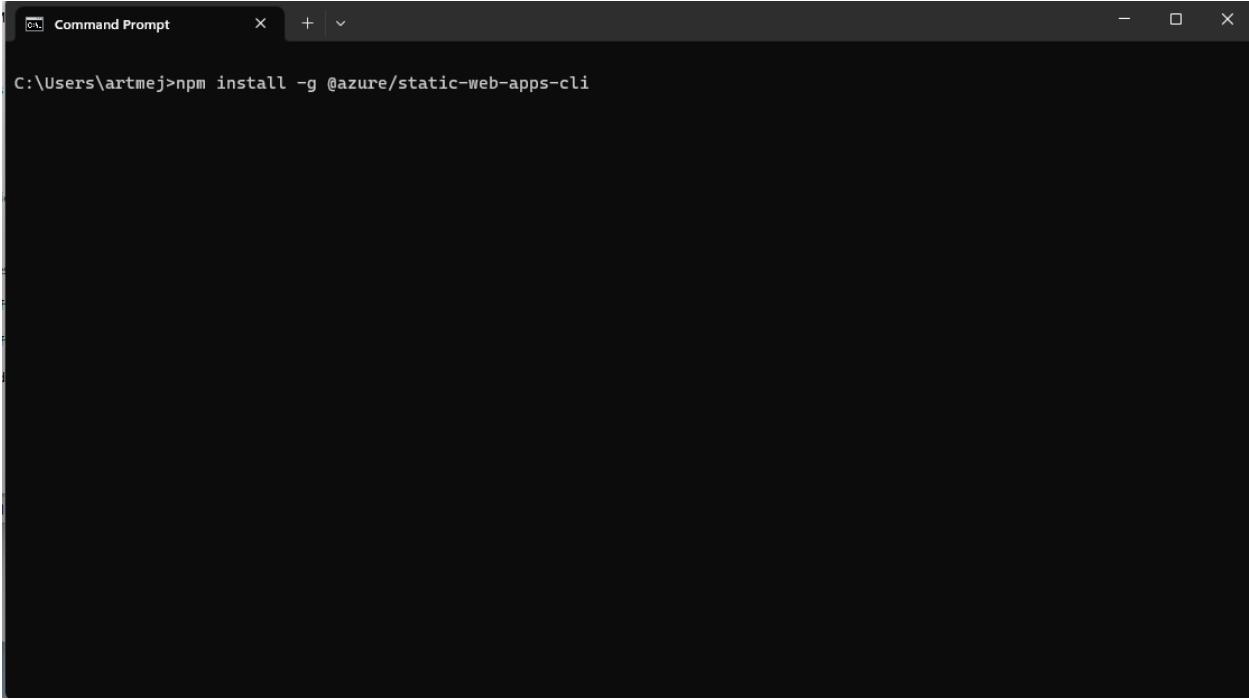


### Azure Static Web Apps CLI

Es una herramienta para el desarrollo local de Azure SWA. La información de esta herramienta se puede encontrar en la siguiente URL [GitHub - Azure/static-web-apps-cli: Azure Static Web Apps CLI](#).

La instalación se ejecuta la siguiente línea de comando basado en npm:

```
npm install -g @azure/static-web-apps-cli
```

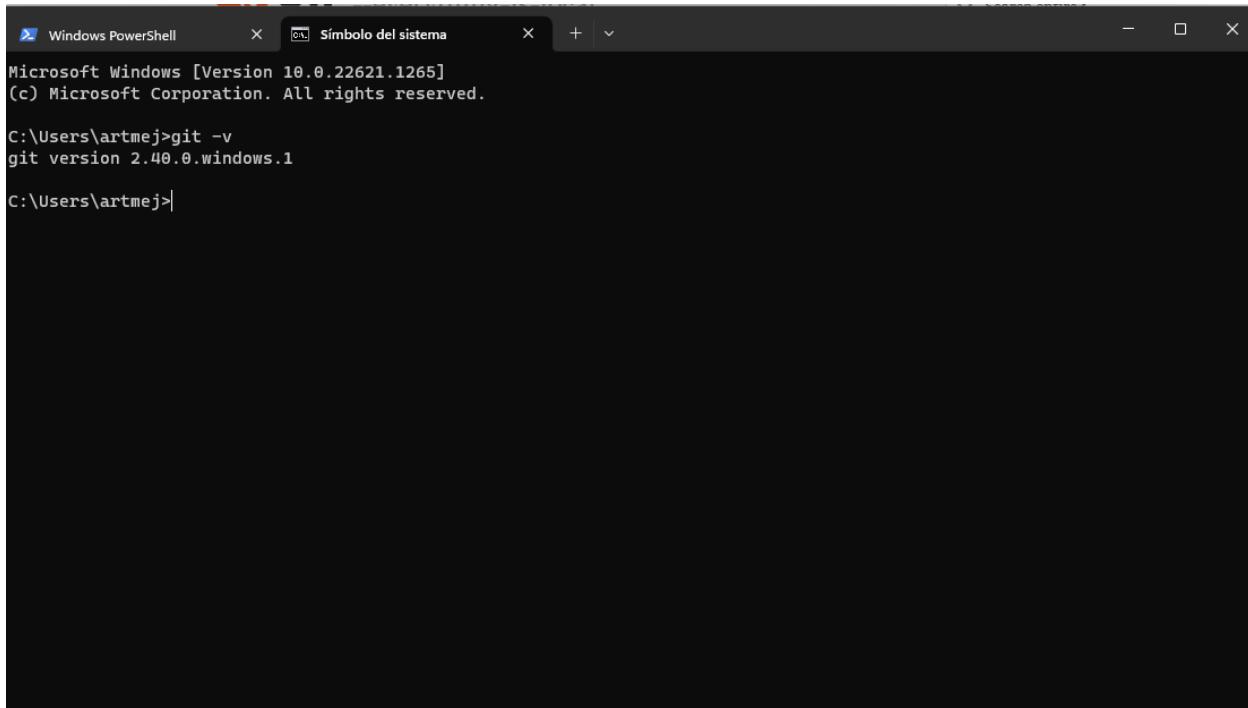


A screenshot of a Windows Command Prompt window titled "Command Prompt". The window shows the command "npm install -g @azure/static-web-apps-cli" being run at the prompt. The output of the command is completely blank, indicating that the package has been successfully installed without any visible logs or progress.

## Git

Git se usará para el manejo de repositorios. Se puede usar la guía publicada en [Install and set up Git - Azure DevOps | Microsoft Learn](#). Se puede validar que se tiene instalado o después de la instalación ejecutando el siguiente comando:

```
git -v
```



A screenshot of a Windows terminal window titled "Windows PowerShell". The window shows the following text:  
Microsoft Windows [Version 10.0.22621.1265]  
(c) Microsoft Corporation. All rights reserved.  
C:\Users\artmej>git -v  
git version 2.40.0.windows.1  
C:\Users\artmej>

Si se cuenta con una versión de git, se mostrará la información sobre la versión se tiene instalada.

## Vue

Vue es un framework de desarrollo con backgroud HTML. Se instala via Node Package Manager. Se ejecutarán dos comandos:

```
npm install vue
npm install -g @vue/cli
```

Mostrándose la terminal de la siguiente manera:

```

C:\Users\artmej>npm install vue
npm WARN deprecated sourcemap-codec@1.4.8: Please use @jridgewell/sourcemap-codec instead
added 21 packages in 3s

1 package is looking for funding
  run 'npm fund' for details

C:\Users\artmej>npm install -g @vue/cli
npm WARN deprecated source-map-url@0.4.1: See https://github.com/lydell/source-map-url#deprecated
npm WARN deprecated urix@0.1.0: Please see https://github.com/lydell/urix#deprecated
npm WARN deprecated resolve-url@0.2.1: https://github.com/lydell/resolve-url#deprecated
npm WARN deprecated source-map-resolve@0.5.3: See https://github.com/lydell/source-map-resolve#deprecated
npm WARN deprecated apollo-server-plugin-base@3.7.2: The 'apollo-server-plugin-base' package is part of Apollo Server v2 and v3, which are now deprecated (end-of-life October 22nd 2023). This package's functionality is now found in the '@apollo/server' package. See https://www.apollographql.com/docs/apollo-server/previous-versions/ for more details.
npm WARN deprecated apollo-server-errors@3.3.1: The 'apollo-server-errors' package is part of Apollo Server v2 and v3, which are now deprecated (end-of-life October 22nd 2023). This package's functionality is now found in the '@apollo/server' package. See https://www.apollographql.com/docs/apollo-server/previous-versions/ for more details.
npm WARN deprecated apollo-datasource@3.3.2: The 'apollo-datasource' package is part of Apollo Server v2 and v3, which are now deprecated (end-of-life October 22nd 2023). See https://www.apollographql.com/docs/apollo-server/previous-versions/ for more details.
npm WARN deprecated apollo-server-types@3.8.0: The 'apollo-server-types' package is part of Apollo Server v2 and v3, which are now deprecated (end-of-life October 22nd 2023). This package's functionality is now found in the '@apollo/server' package. See https://www.apollographql.com/docs/apollo-server/previous-versions/ for more details.
npm WARN deprecated apollo-reporting-protobuf@3.4.0: The 'apollo-reporting-protobuf' package is part of Apollo Server v2 and v3, which are now deprecated (end-of-life October 22nd 2023). This package's functionality is now found in the '@apollo/usage-reporting-protobuf' package. See https://www.apollographql.com/docs/apollo-server/previous-versions/ for more details.
npm WARN deprecated apollo-server-env@4.2.1: The 'apollo-server-env' package is part of Apollo Server v2 and v3, which are now deprecated (end-of-life October 22nd 2023). This package's functionality is now found in the '@apollo/utils.fetcher' package. See https://www.apollographql.com/docs/apollo-server/previous-versions/ for more details.
npm WARN deprecated apollo-server-express@3.12.0: The 'apollo-server-express' package is part of Apollo Server v2 and v3, which are now deprecated (end-of-life October 22nd 2023). This package's functionality is now found in the '@apollo/server' package. See https://www.apollographql.com/docs/apollo-server/previous-versions/ for more details.
npm WARN deprecated subscriptions-transport-ws@0.11.0: The 'subscriptions-transport-ws' package is no longer maintained. We recommend you use 'graphql-ws' instead. For help migrating Apollo software to 'graphql-ws', see https://www.apollographql.com/docs/apollo-server/data/subscriptions/#switching-from-subscriptions-transport-ws. For general help using 'graphql-ws', see https://github.com/enisdenjo/graphql-ws/blob/master/README.md
npm WARN deprecated apollo-server-core@3.12.0: The 'apollo-server-core' package is part of Apollo Server v2 and v3, which are now deprecated (end-of-life October 22nd 2023). This package's functionality is now found in the '@apollo/server' package. See https://www.apollographql.com/docs/apollo-server/previous-versions/ for more details.

added 855 packages in 34s

64 packages are looking for funding
  run 'npm fund' for details

C:\Users\artmej>

```

Se valida que vue se encuentre instalado y en que versión.

`vue -version`

```

C:\Users\artmej>vue --version
@vue/cli 5.0.8

C:\Users\artmej>

```

En caso de tener este mensaje:

```
PS C:\Users\artmej> vue
vue : File C:\Users\artmej\AppData\Roaming\npm\vue.ps1 cannot be loaded because running scripts is disabled on this
system. For more information, see about_Execution_Policies at https://go.microsoft.com/fwlink/?LinkID=135170.
At line:1 char:1
+ vue
+ ~~~
+ CategoryInfo          : SecurityError: () [], PSSecurityException
+ FullyQualifiedErrorId : UnauthorizedAccess
```

Se ejecuta el comando:

```
Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy RemoteSigned
```

En el clean up puede regresarse con el siguiente comando

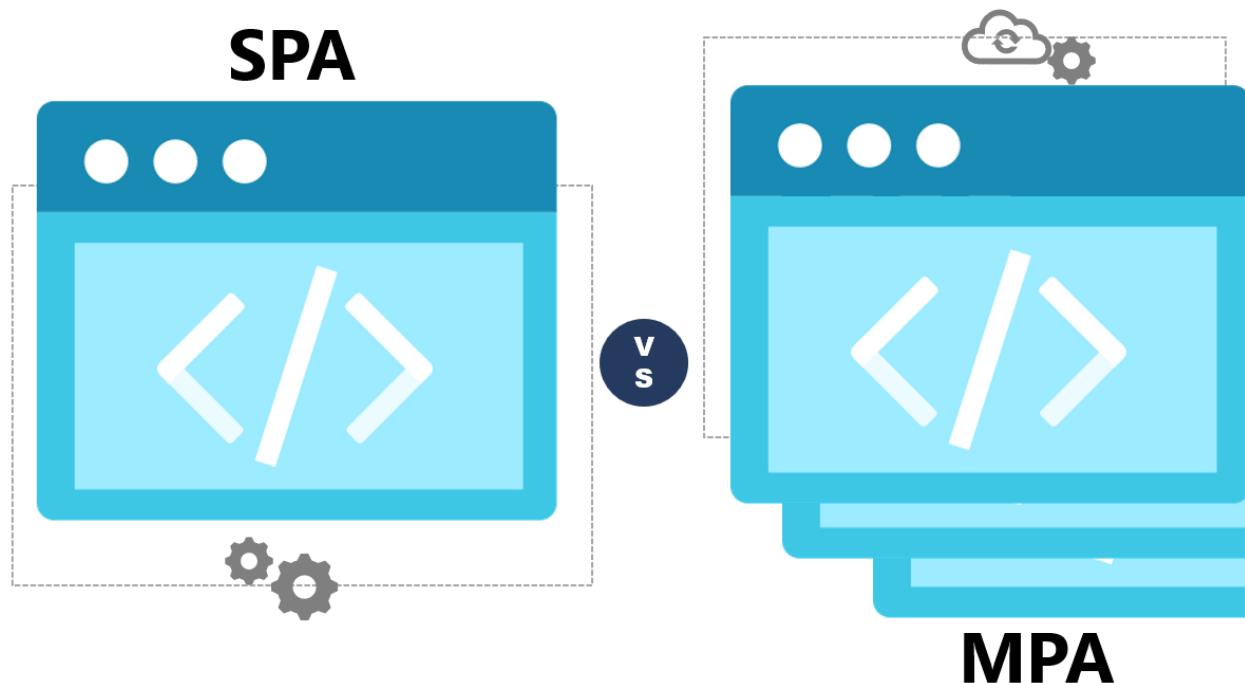
```
Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy Undefined
```

Se puede tener más información de esta instalación en: [Install Vue.js directly on Windows | Microsoft Learn](https://docs.microsoft.com/en-us/javascript/api/vue?view=vue-latest)

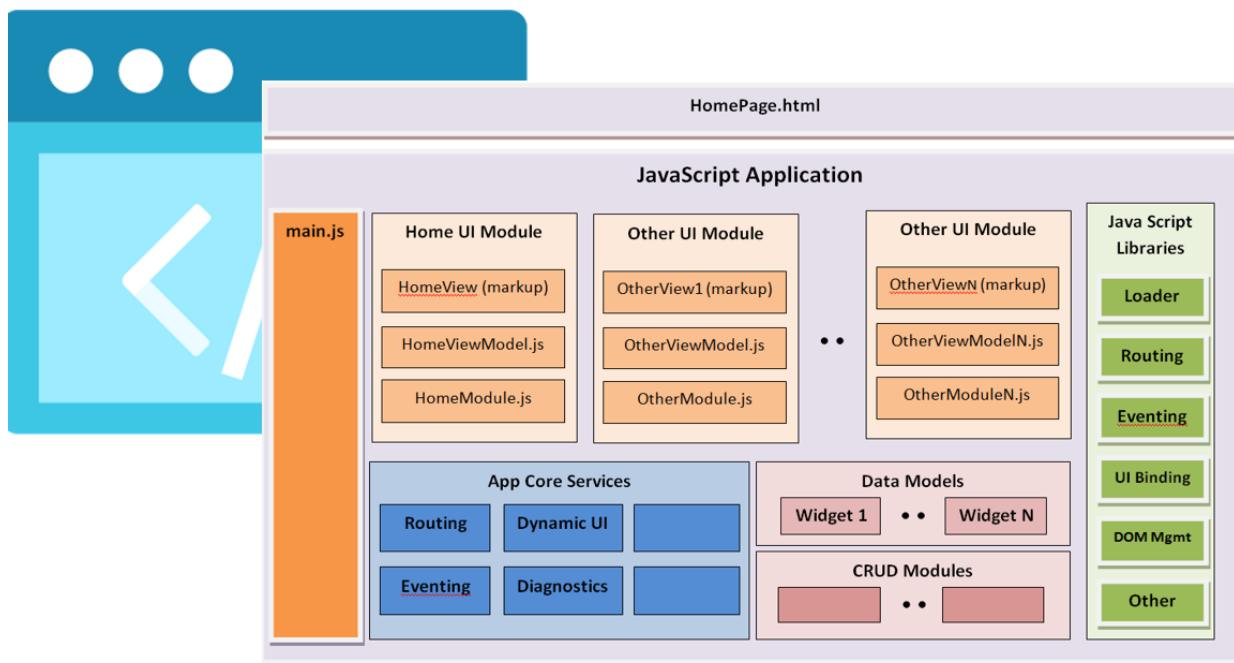
## 2. Single Page Application (SPA)

¿Qué es Single Page Application?

Es una arquitectura que mantiene el procesamiento de los elementos web en un solo documento HTML.



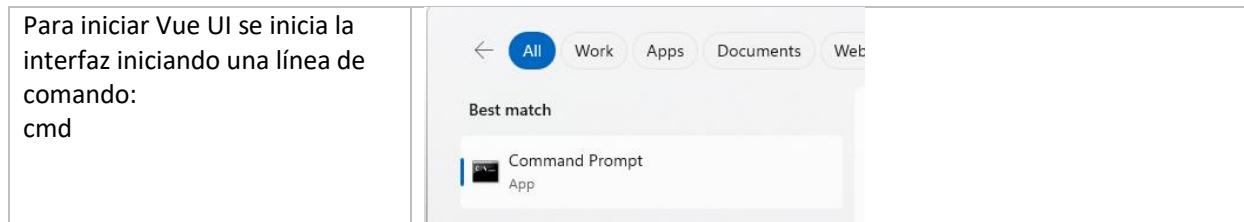
Esta es la arquitectura básica de un SPA



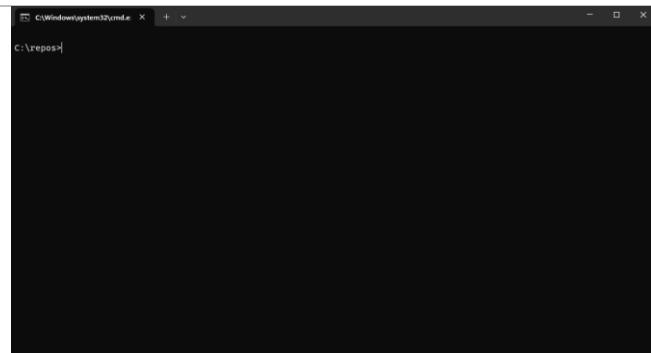
## Desplegar el código vía Vue

En esta sección se usará el Framework Vue para generar la primera SPA y se revisará las secciones con las que se cuenta esta aplicación base.

Vue puede ser generada por la línea de comando o por interfaz de Vue UI. En este caso se usará Vue UI que facilita visualmente la generación del proyecto.



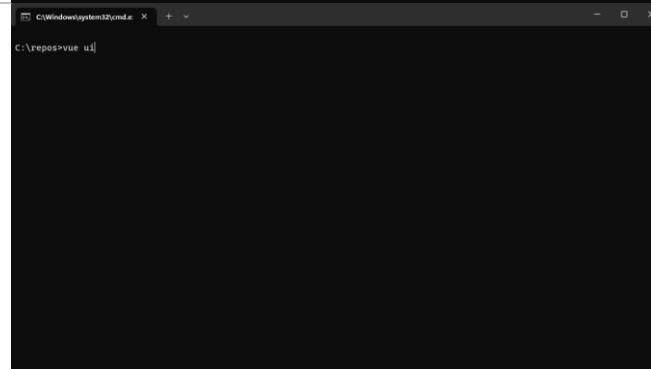
Al iniciar la línea de comando se recomienda ubicar en el directorio más cercano a donde se trabajará nuestro código. En mi caso se generó un folder “repos” en ruta local



```
C:\Windows\system32\cmd.exe + - x
C:\repos>
```

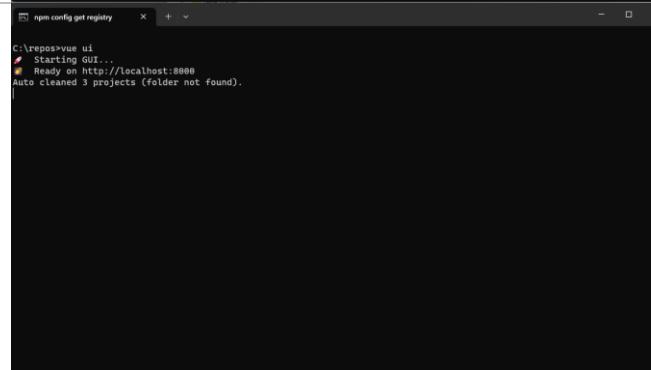
Ya en esta ubicación se iniciará el comando:

**vue ui**



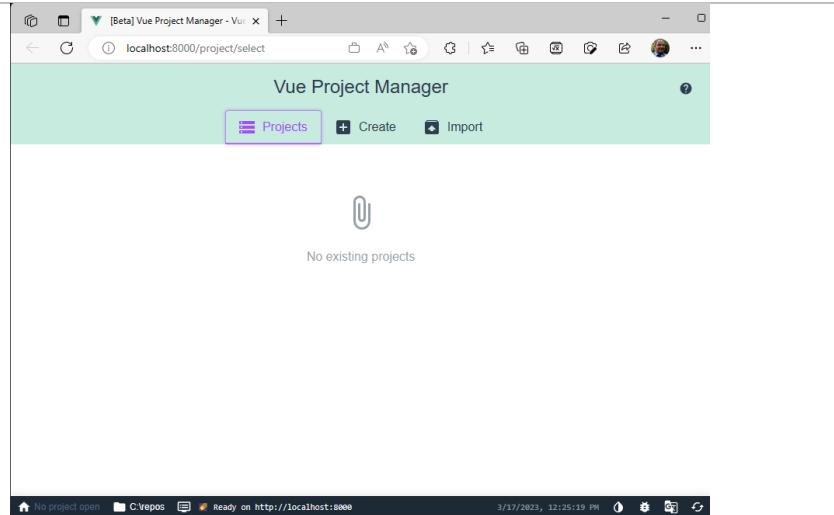
```
C:\Windows\system32\cmd.exe + - x
C:\repos>vue ui
```

Esto iniciará en un navegador la generación de proyecto

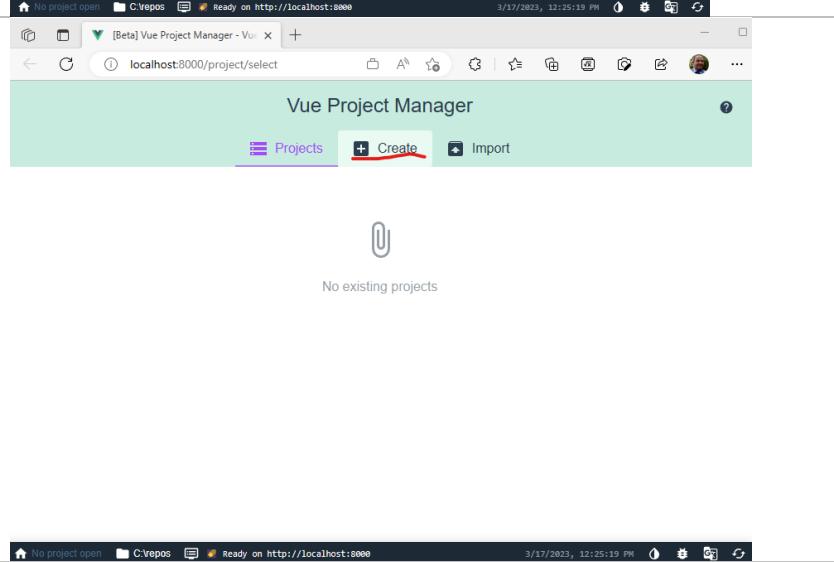


```
C:\Windows\system32\cmd.exe + - x
C:\repos>npm config get registry
C:\repos>vue ui
Starting GUI...
Ready on http://localhost:8080
Auto cleared 3 projects (folder not found).
```

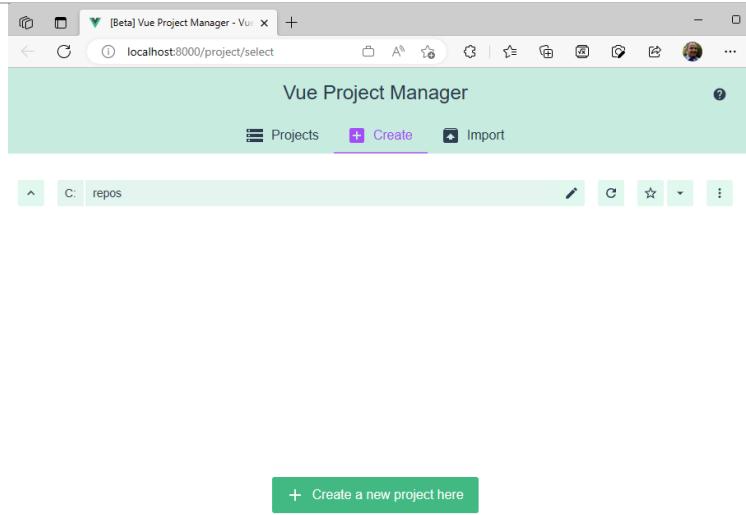
Y muestra la siguiente interfaz



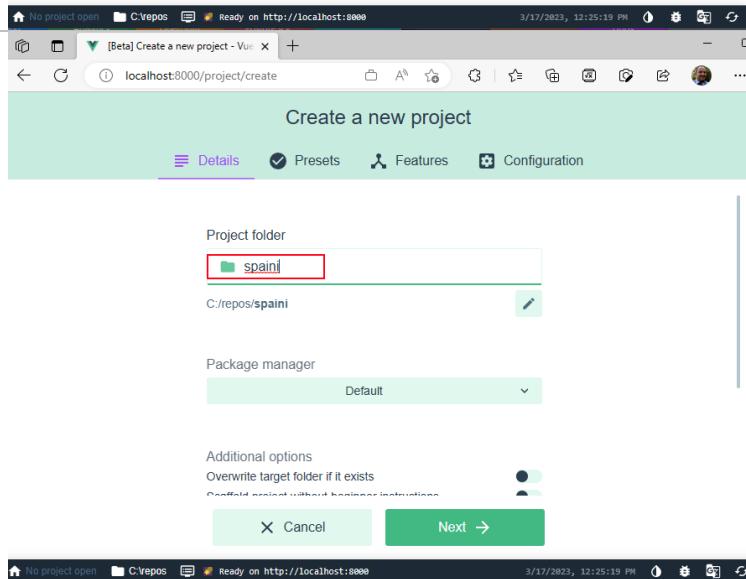
Se iniciará el generar el proyecto en la opción de Create/Crear



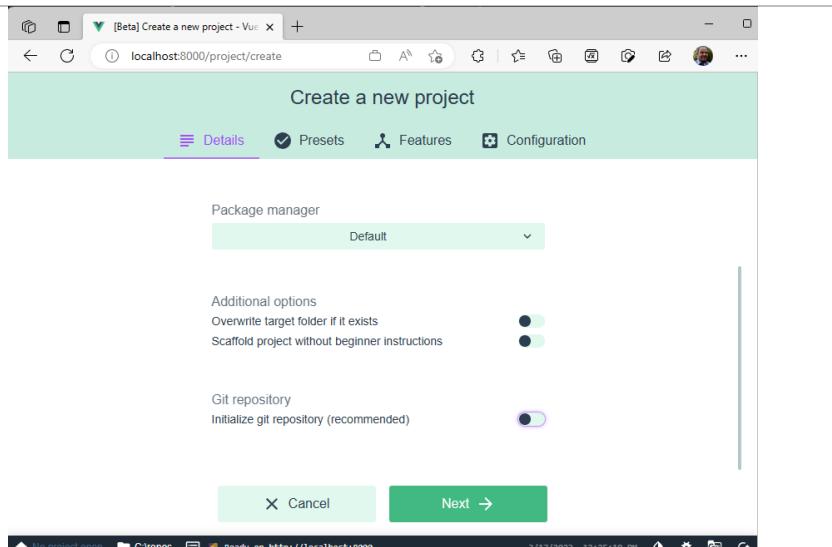
Se indica la ruta donde se genera el proyecto y se da clic en el botón inferior de “Create a new Project here”



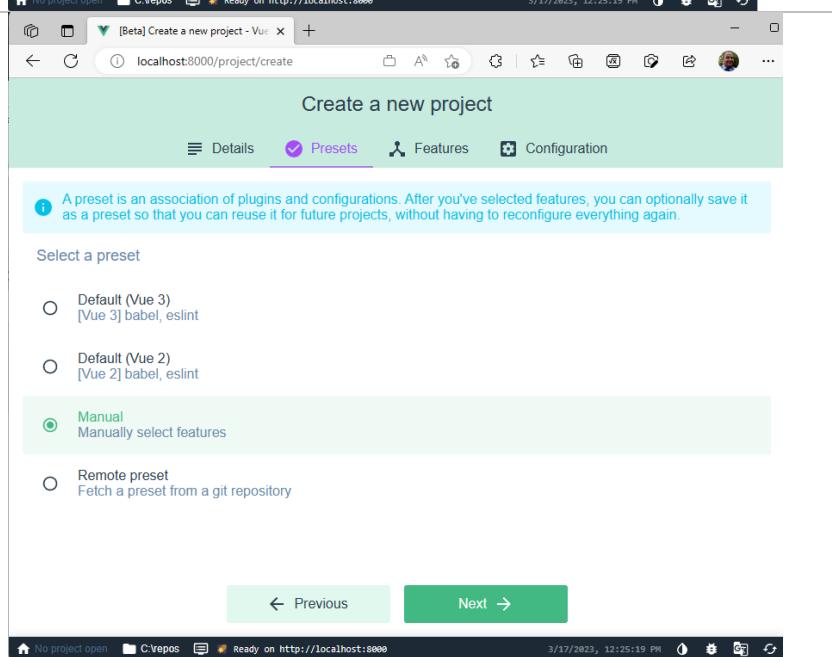
Se agrega el nombre del folder del proyecto



Se dejarán las siguientes opciones de la siguiente manera. El ligar al repositorio se realizará más adelante, se da Next en este paso.



En la siguiente pantalla se seleccionará “Manual” para seleccionar nuestras opciones y se dará clic a “Next”



En la siguiente pantalla se seleccionará “Router” y se da clic a “Next”.

Create a new project

Details Presets Features Configuration

You will be able to add features after the project is created by installing plugins.

Enable features

Babel  
Transpile modern JavaScript to older versions (for compatibility) More Info

TypeScript  
Add support for the TypeScript language More Info

Progressive Web App (PWA) Support  
Improve performances with features like Web manifest and Service workers More Info

Router  
Structure the app with dynamic pages More Info

Vuex  
Manage the app state with a centralized store More Info

CSS Pre-processors  
Add support for CSS pre-processors like Sass, Less or Stylus More Info

Linter / Formatter  
Check and enforce code quality with ESLint or Prettier More Info

Unit Testing  
Add a Unit Testing solution like Jest or Mocha More Info

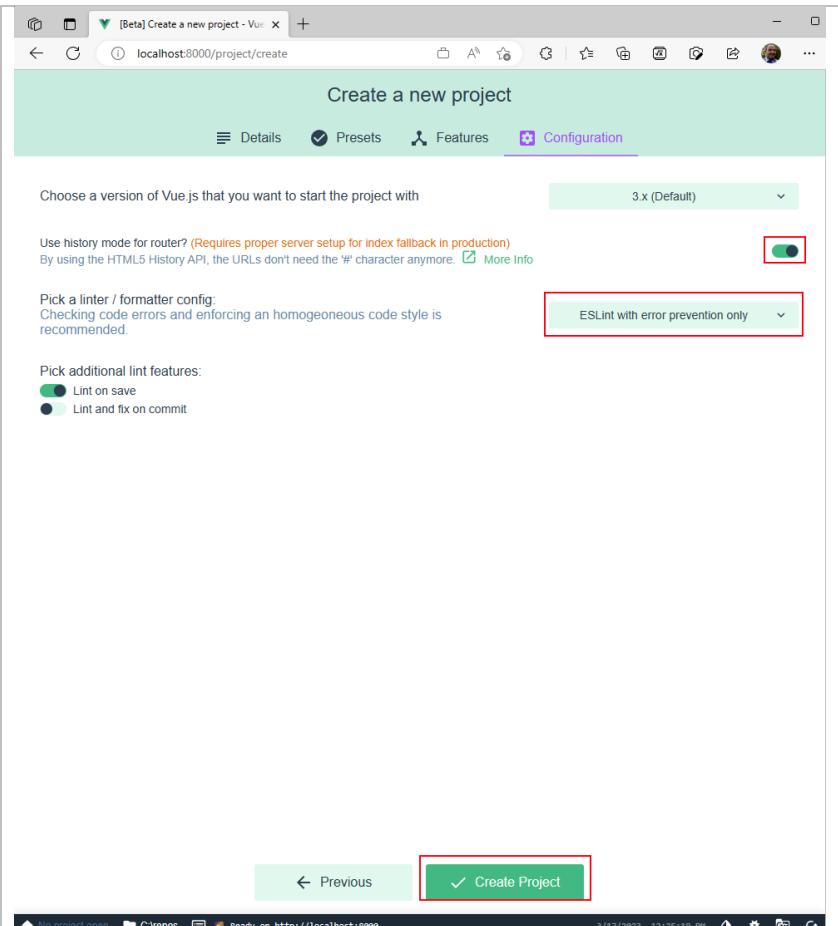
E2E Testing  
Add an End-to-End testing solution to the app like Cypress or Nightwatch More Info

Use config files  
Use specific configuration files (like '.babelrc') instead of using 'package.json'.

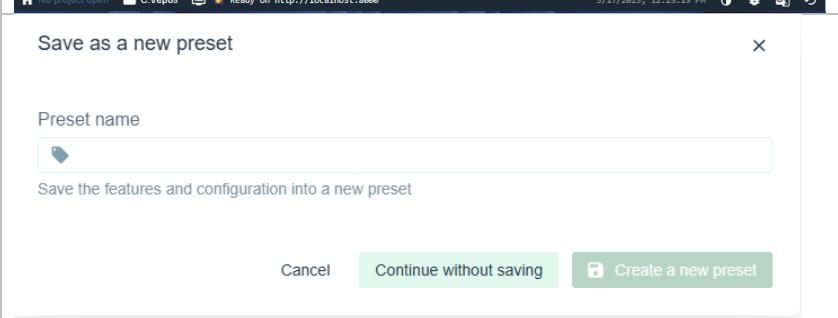
← Previous Next →

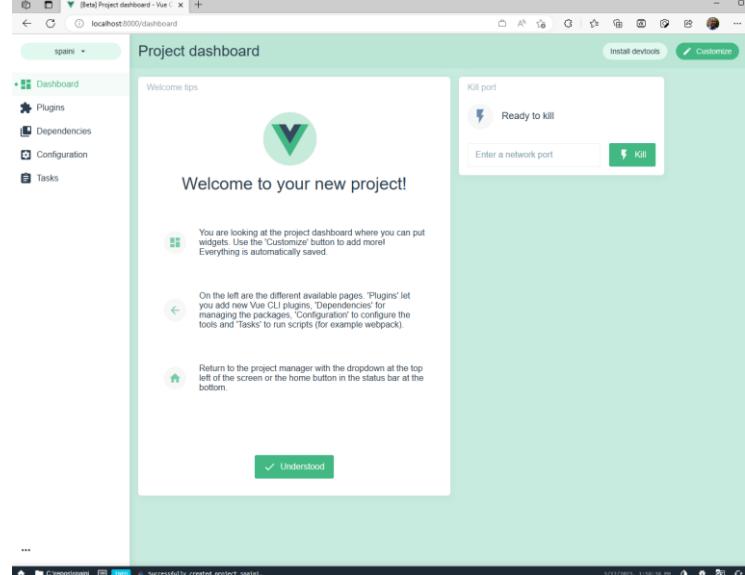
No project open C:\repos Ready on http://localhost:8000 3/17/2023, 12:25:19 PM

En la siguiente pantalla se habilita la opción de “history mode for router” y se selecciona la opción de Linter que se quiere manejar y se da clic en Create Project



En caso de querer generar un preset se puede asignar un nombre y guardar o se puede continuar sin guardarlo.



<p>Al seleccionar una opción se generará el proyecto.</p>	 <p>Creating project...</p>
<p>Al terminar se tendrá la página de configuración del proyecto en el cual se puede regresar en cualquier momento para configurar.</p>	
<p>Se puede regresar al cmd y cerrar con combinación de teclas "ctrl+c" y cerrando el navegador.</p>	 <pre>C:\repos&gt;vue ui 🚀 Starting GUI... ⚡ Ready on http://localhost:8000 C:\repos&gt; </pre>

## Ejecutar la aplicación

Posicionados en la ruta que generamos se ingresará al directorio donde está la aplicación.

En el caso de ejemplo se ingresa al directorio “spaini”

```
C:\repos>dir
Volume in drive C has no label.
Volume Serial Number is 08E6-895A

Directory of C:\repos

17/03/2023  01:57 p. m.    <DIR>
17/03/2023  01:58 p. m.    <DIR>          spaini
                           0 File(s)           0 bytes
                           2 Dir(s)  888,553,746,432 bytes free

C:\repos>cd spaini
C:\repos\spaini>
```

En este punto se ejecutará el comando, esto compilará la solución y nos permite revisar la aplicación localmente.

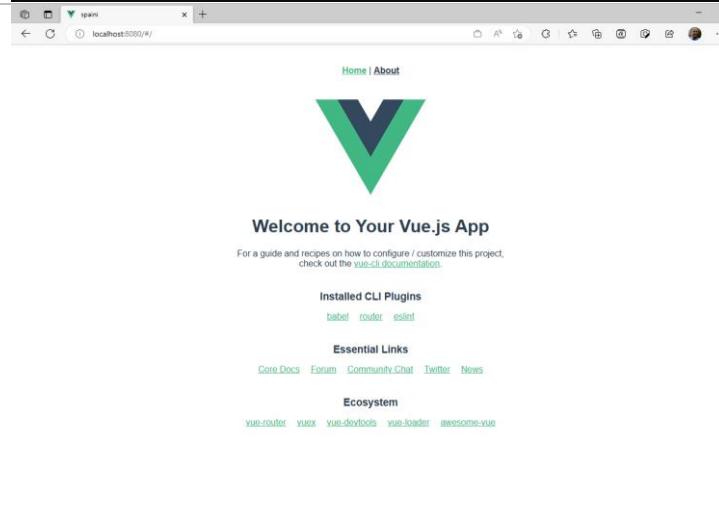
`npm run serve`

```
[DONE] Compiled successfully in 5191ms

App running at:
- Local:  http://localhost:8080/
- Network: http://192.168.1.75:8080/

Note that the development build is not optimized.
To create a production build, run npm run build.
```

Si todo es correcto, contaremos con una SPA con Vue que podemos consultar en el navegador.



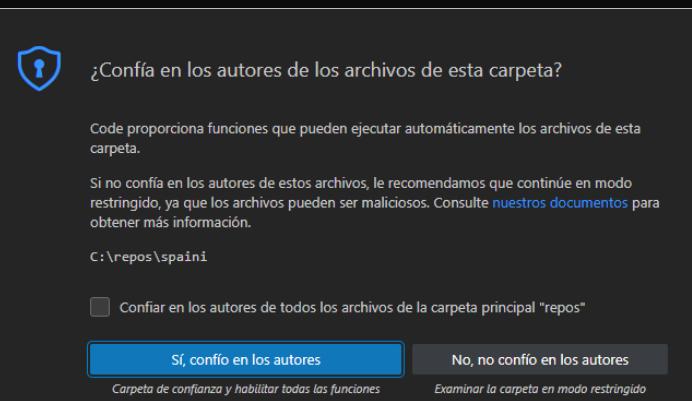
## Exploración del SPA

En este submódulo revisaremos los elementos de la SPA en vue, cómo funciona una SPA y el ruteo de los elementos.

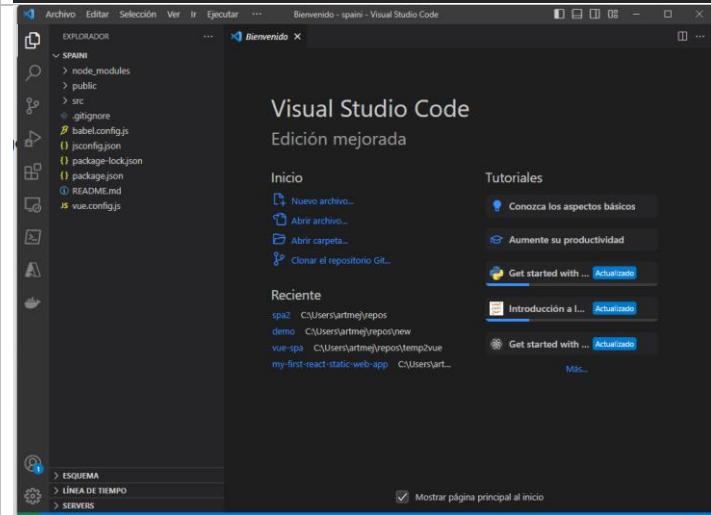
Para iniciar se ejecutará el comando desde línea de comando “code .” o abrir el folder en vs code.

```
C:\Windows\system32\cmd.exe
C:\repos\spaini>code .
```

Al abrir el Vs Code es posible que pregunte si se confía en los autores se ingresará la opción de “Sí, confío en los autores”



En VS Code se verá nuestro código de la siguiente forma.  
Se expande la aplicación de directorio base y se ingresa al directorio “public”. Dentro de este directorio *public* está un archivo llamado *index.html*. En *index.html* se pude ver una sección en un tag *div* llamada *app*. Aquí es donde todas nuestras vistas serán desplegadas.



## Solo una página index.html

Lo importante en esta página es validar que en la sección del objeto *html* *div* llamado *app* es donde se cargarán las vistas que se generen.

Vue es un framework que está basado en uso de *html* para ubicar las vistas. Por lo que durante la ejecución de la solución estará haciendo los cambios visuales en este *div* del contenido que queremos mostrar.

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,initial-scale=1.0" />
    <link rel="icon" href="<%= BASE_URL %>/favicon.ico" />
    <title>We're sorry but <%= htmlWebpackPlugin.options.title %> does not exist</title>
    </head>
    <body>
        <script>
            <div id="app"></div>
        </script>
        <!-- built files will be auto injected -->
    </body>
</html>

```

## Carga de App

Las vistas generadas en la SPA se cargan al objeto App como se explica a continuación:

Los objetos se tipo .vue se cargan jerárquicamente sobre el archivo *src/main.js*. Se puede notar la relación del objeto principal *App* y la carga/monta de *#app*.

Este objeto se relaciona directamente con nuestro objeto *#app* que es una Vue en *src/App.vue*.

```

import { createApp } from 'vue'
import App from './App.vue'
import router from './router'

createApp(App).use(router).mount('#app')

```

```

<template>
    <nav>
        <a href="#">Home</a>
        <a href="#">About</a>
    </nav>
    <router-view></router-view>
</template>

<style>
    #app {
        font-family: Avenir, Helvetica, Arial, sans-serif;
        -webkit-font-smoothing: antialiased;
        -moz-osx-font-smoothing: grayscale;
        text-align: center;
        color: #2c3e50;
    }
    nav {
        padding: 30px;
    }
    nav a {
        font-weight: bold;
        color: #2c3e50;
    }
    nav a.router-link-exact-active {
        color: #42b983;
    }
</style>

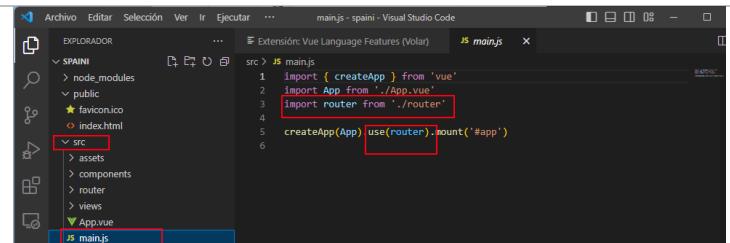
```

## Carga de Route

Route es un elemento relevante para el SPA. Esto debido a que nos va a ayudar a diferenciar el envío de rutas en el URL y poderlas interpretar como una vista en la SPA. En caso de que no se habilite al manejo de rutas el servidor que está entregando los archivos va a intentar localizar el objeto que se envía como parte de la *url* como un archivo, el cual no existe.

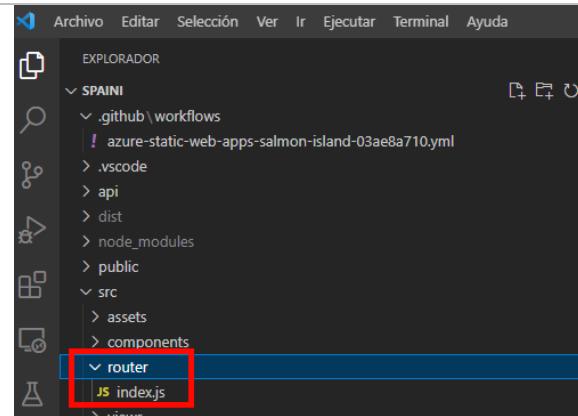
Cada objeto que se requiere manejar en la URL se requiere validar en el ruteo y relacionarlo con un objeto que es una vista.

Se puede ver que se relaciona el objeto de ruteo en la misma línea de la App y `#app`.



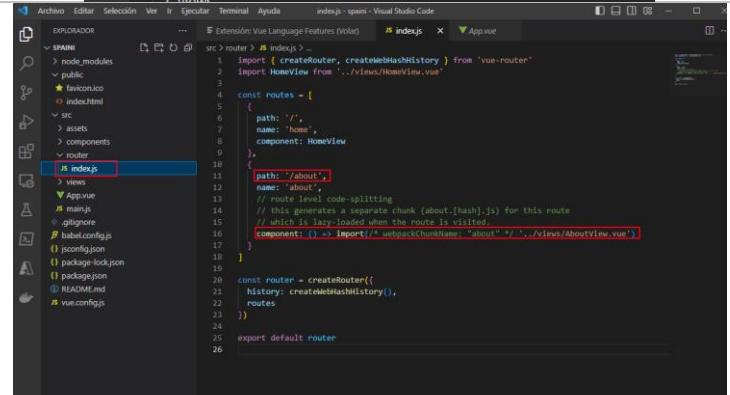
```
src > JS main.js
1 import { createApp } from 'vue'
2 import App from './App.vue'
3 import router from './router'
4
5 createApp(App).use(router).mount('#app')
```

El objeto de Router se encuentra dentro de `/src/router/index.js`. Este archivo configura el *path* que se va a usar en la *url*, un nombre y la vista que se cargará en el navegador.



```
src > JS index.js
1 import { createRouter, createWebHashHistory } from 'vue-router'
2 import HomeView from '../views/HomeView.vue'
3
4 const routes = [
5   {
6     path: '/',
7     name: 'home',
8     component: HomeView
9   },
10  {
11    path: '/about',
12    name: 'about',
13    // route level code-splitting
14    // this generates a separate chunk (about.[hash].js) for this route
15    // which is lazy-loaded when the route is visited.
16    component: () => import(/* webpackChunkName: 'about' */ '../views/AboutView.vue')
17  }
18]
19
20 const router = createRouter({
21   history: createWebHashHistory(),
22   routes
23 })
24
25 export default router
```

Un elemento que se debe revisar en este archivo es el manejo del identificador `"#"` en las *url* en la historia.



```
src > router > JS index.js
1 import { createRouter, createWebHashHistory } from 'vue-router'
2 import HomeView from '../views/HomeView.vue'
3
4 const routes = [
5   {
6     path: '/',
7     name: 'home',
8     component: HomeView
9   },
10  {
11    path: '/about',
12    name: 'about',
13    // route level code-splitting
14    // this generates a separate chunk (about.[hash].js) for this route
15    // which is lazy-loaded when the route is visited.
16    component: () => import(/* webpackChunkName: 'about' */ '../views/AboutView.vue')
17  }
18]
19
20 const router = createRouter({
21   history: createWebHashHistory(),
22   routes
23 })
24
25 export default router
```

Estos elementos se referencian en ligas en el objeto *App.vue* que permite enviar las vistas configuradas:

```

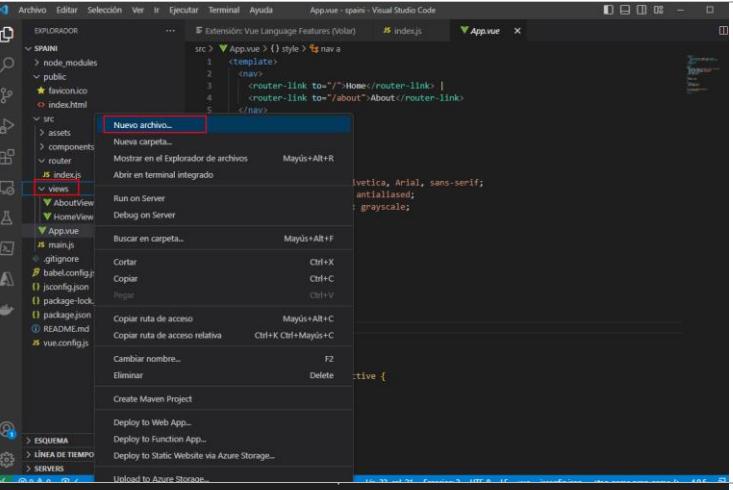
src > App.vue > {} style > nav a
1 <template>
2   <nav>
3     <a href="#"><router-link to="/">Home</router-link></a>
4     <a href="#"><router-link to="/about">About</router-link></a>
5   </nav>
6 </template>
7
8 <style>
9   #app {
10     font-family: Avenir, Helvetica, Arial, sans-serif;
11     -webkit-font-smoothing: antialiased;
12     -moz-osx-font-smoothing: grayscale;
13     text-align: center;
14     color: #2e3436;
15   }
16
17   nav {
18     padding: 30px;
19   }
20
21   nav a {
22     font-weight: bold;
23     color: #2e3436;
24   }
25
26   nav a.router-link-exact-active {
27     color: #42b983;
28   }
29
30 </style>
31

```

## Agregar una nueva Vista

En este módulo se agrega una nueva vista a la SPA.

Lo primero a realizar es agregar la sección de contacto en la carpeta de *views*. Se agregará el archivo *ContactoView.vue*. Se da clic derecho del mouse sobre el folder de vistas y se selecciona nuevo archivo.



Se agregara un nuevo template a mostrar.

```

<template>
<div>
  <h1>Gracias por contactarnos</h1>
</div>
</template>

```

## Agregar la ruta a nueva vista

Para agregar la ruta modificaremos dos archivos el *App.vue* y *router/index.js*.

Primero modificaremos el archivo `index.js`. Se agregará a la colección de rutas nuestra vista y referencias:

```

    // router > JS index.js
    import { createRouter, createWebHashHistory } from 'vue-router'
    import HomeView from '../views/HomeView.vue'

    const routes = [
      {
        path: '/',
        name: 'home',
        component: HomeView
      },
      {
        path: '/about',
        name: 'about',
        // route level code-splitting
        // this generates a separate chunk (about.[hash].js) for this route
        // which is lazy-loaded when the route is visited,
        component: () => import(`../views/AboutView.vue`)
      },
      {
        path: '/contacto',
        name: 'contacto',
        component: () => import('../views/ContactoView.vue')
      }
    ]

    const router = createRouter({
      history: createWebHashHistory(),
      routes
    })

    export default router
  
```

Se agrega al menú de navegación en `App.vue` la ruta a nuestra vista. Para esto se agrega el `router-link` en `App.vue`.

```

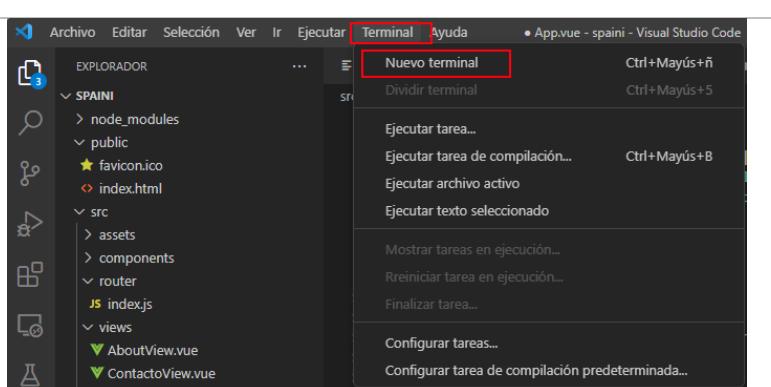
<template>
  <nav>
    <a href="#">router-link</a>
    <a href="#">router-link to="/about">About</a>
    <a href="#">router-link to="/contacto">Contacto</a>
  </nav>
  <router-view></router-view>
</template>
<style>
  #app {
    font-family: Avenir, Helvetica, Arial, sans-serif;
    -webkit-font-smoothing: antialiased;
    -moz-osx-font-smoothing: grayscale;
    text-align: center;
    color: #2c3e50;
  }
  nav {
    padding: 30px;
  }
  nav a {
    font-weight: bold;
    color: #2c3e50;
  }
  nav a.router-link-exact-active {
    color: #4db6ac;
  }
</style>

```

## Prueba de la Vista

Usaremos para iniciar nuestra ejecución la terminal en VS Code.

Se puede iniciar la terminal por medio de menú superior en Terminal o por shortcut (`[us]Ctrl+`` o `[es]Ctrl+Shift+ñ`).



En nuestra terminal ejecutamos el comando

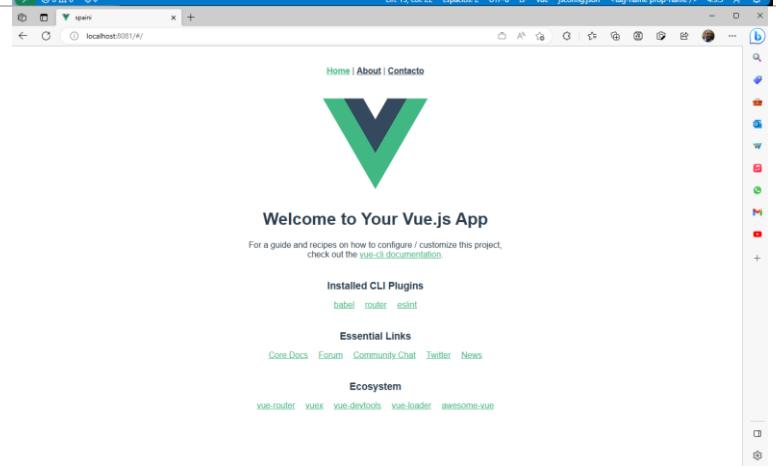
**npm run serve**

```
src \ App.vue
  <template>
    <nav>
      <router-link to="/">Home</router-link> | 
      <router-link to="/about">About</router-link> | 
      <router-link to="/contacto">Contacto</router-link>
    </nav>
    <router-view>
    </router-view>
  </template>
  <style>
    @import url('https://fonts.googleapis.com/css2?family=Avenir+Next:ital,wght@0,400;1,700&display=swap');
    font-family: Avenir, Helvetica, Arial, sans-serif;
    -webkit-font-smoothing: antialiased;
    -moz-osx-font-smoothing: grayscale;
    text-align: center;
    color: #2c3e50;
  </style>
  <nav>
    padding: 30px;
  </nav>
```

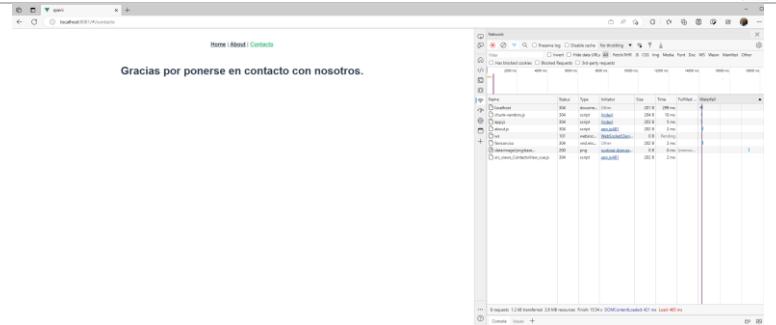
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL AZURE

```
PS C:\repos\spain> npm run serve
spain@0.1.0 serve
vue-cli-service serve
[INFO] Starting development server...
[108] building (0/0 modules)
```

Ingresamos a la dirección que nos da en la línea de comando en un navegador.



Aquí se puede probar la navegación y por medio de devtools validar las llamadas que se están realizando.



### 3. Integración Continua y Despliegue Continuo

El código que se generó se puede integrar a Github o Azure Devops. En caso de que se cuente con una cuenta de Github se puede usar la opción 1. En caso contrario se puede optar por generar un Azure DevOps y continuar la carga del repositorio con la opción 2.

## Opción 1: Github

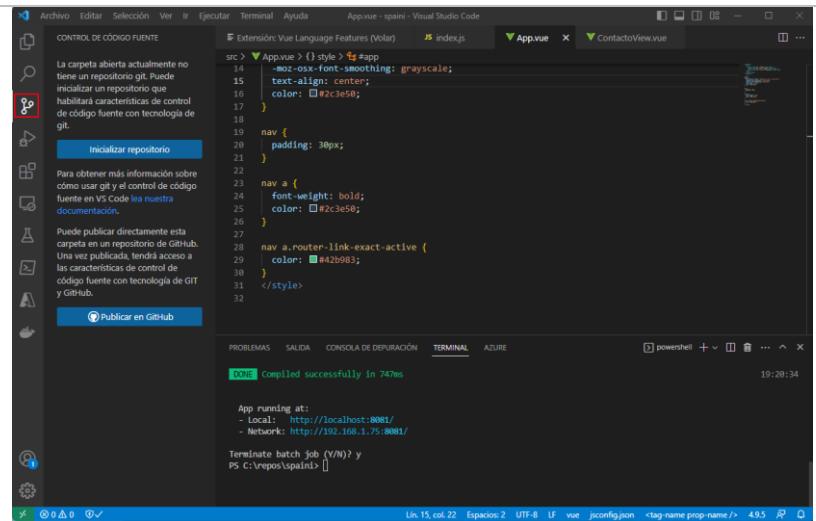
Se selecciona la opción de "control de código fuente en la barra lateral de VS Code.

En caso de requerir conectar la cuenta a los comandos de git. Se requiere ejecutar:

```
git config --global  
user.email  
"hello@example.com"
```

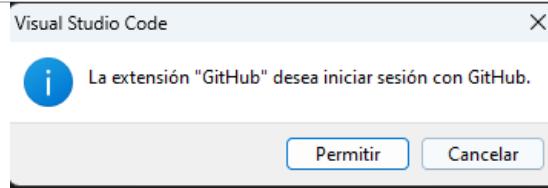
```
git config --global  
username "hello"
```

Se da clic en la *Iniciar repositorio*

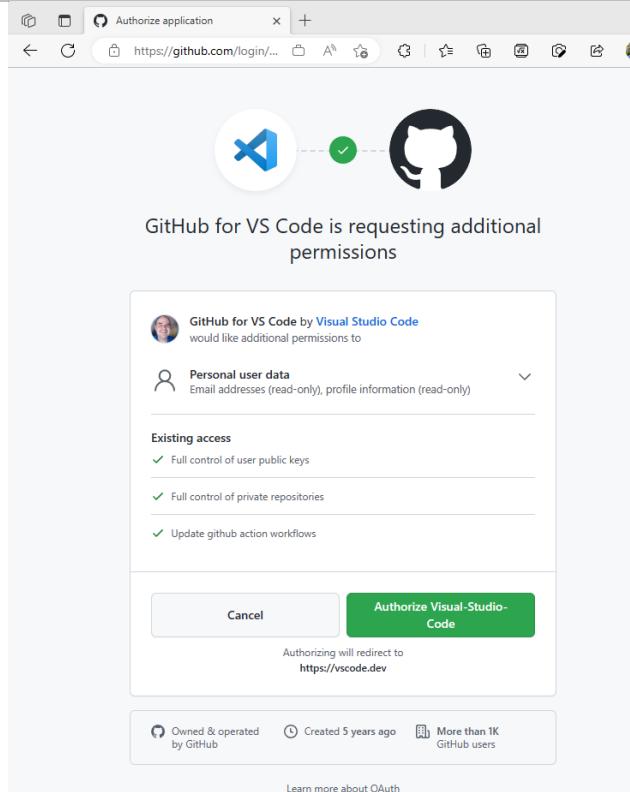


The screenshot shows the Visual Studio Code interface. On the left, there's a sidebar titled 'CONTROL DE CÓDIGO FUENTE' with a red box highlighting the 'Iniciar repositorio' button. The main area is a code editor with some CSS code. The bottom right shows a terminal window with the message 'Compiled successfully in 747ms'.

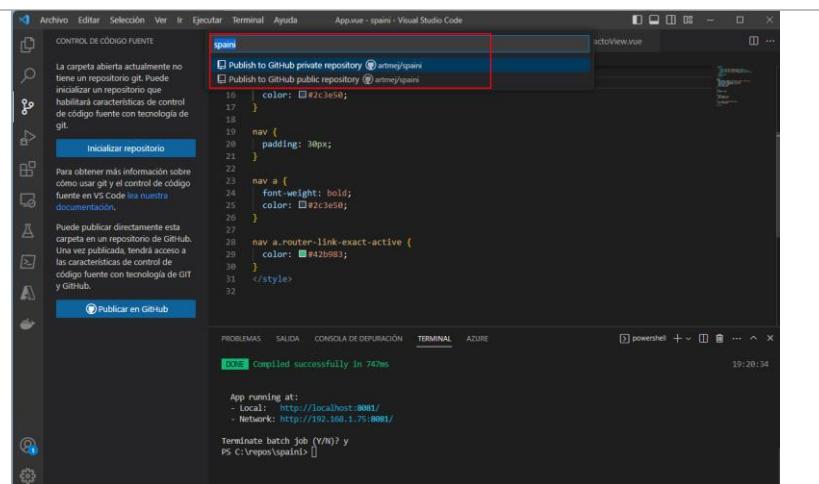
Nos pide iniciar sesión en la cuenta de GitHub.



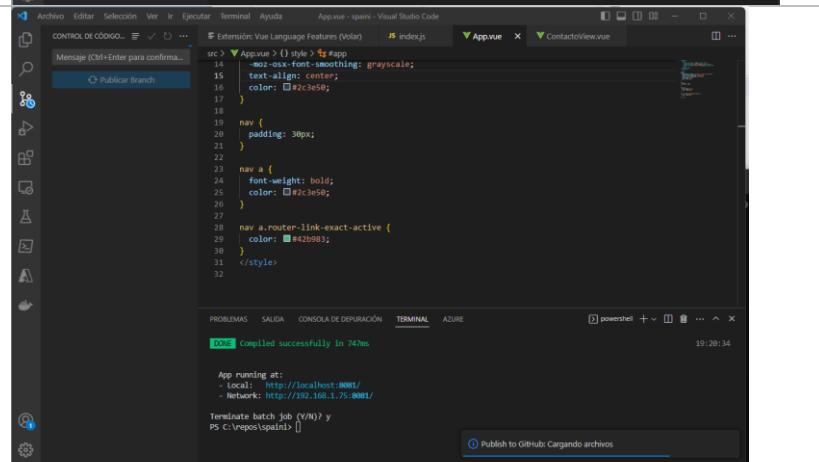
Se abrirá un navegador y pedirá la autorización de acceso a recursos. Se da clic en "Autorizar Visual Studio Code"



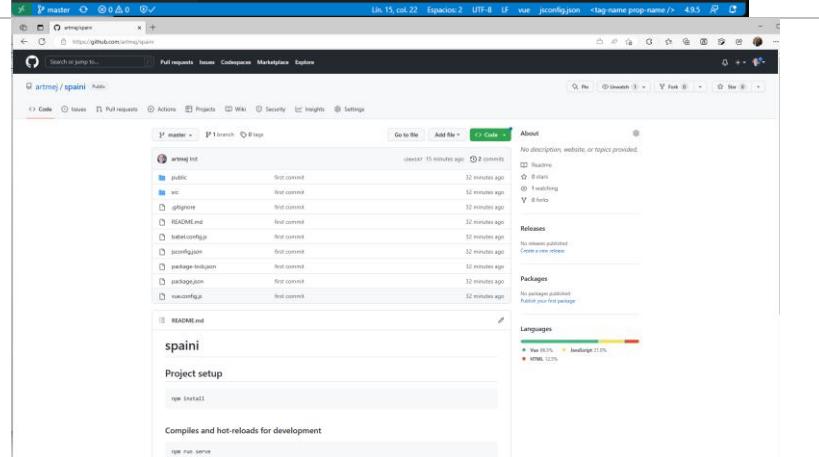
Al terminar la autenticación, Se regresa al VS Code y pregunta el tipo de repositorio se va a usar, público o privado y también pregunta el nombre del repositorio.



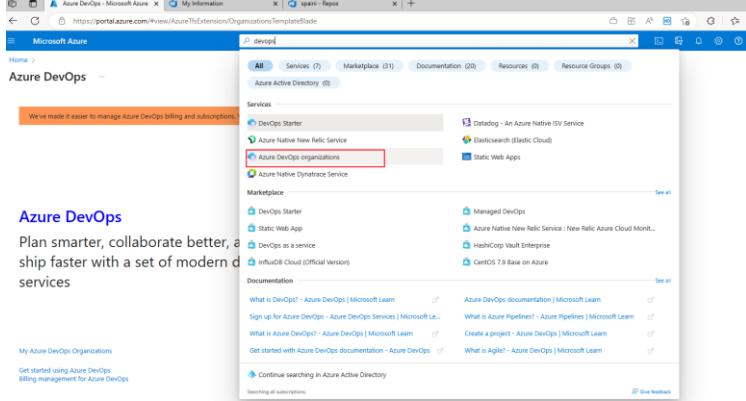
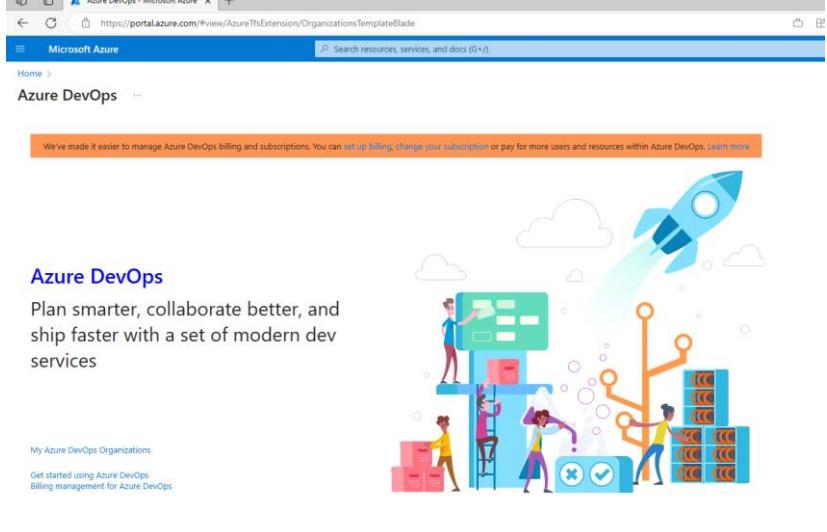
Se inicia la publicación en GitHub.



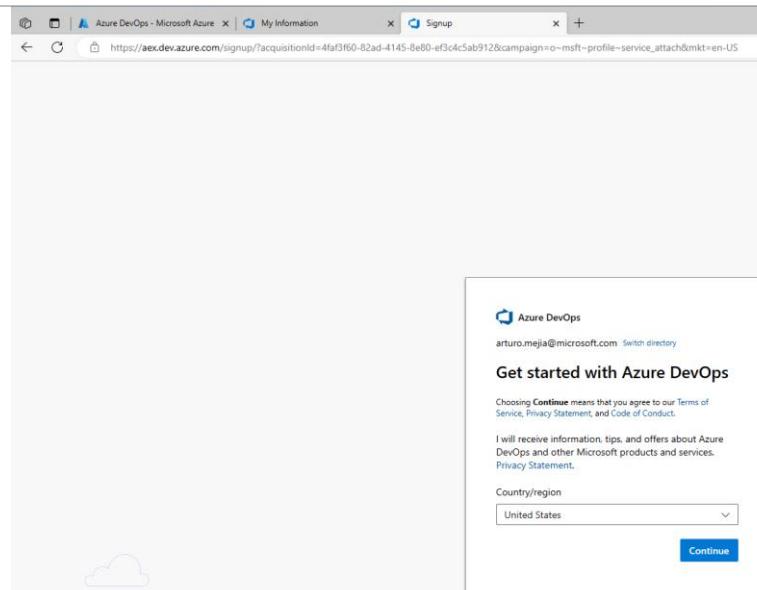
Al finalizar podemos validar en GitHub que el código está listo para los siguientes pasos.



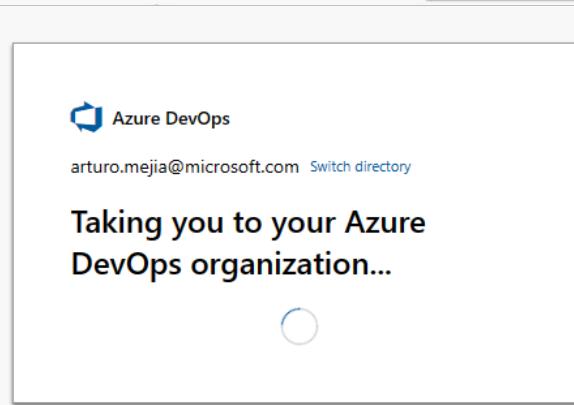
## Opcion 2: Azure Dev Ops (ADO)

<p><b>Generar Azure DevOps desde el portal de Azure. Se inicia en el portal de Azure y se busca la opción Devops.</b></p>	
<p>Ingresando en esta sección se muestra la siguiente página y se selecciona "My Azure DevOps Organizations".</p>	

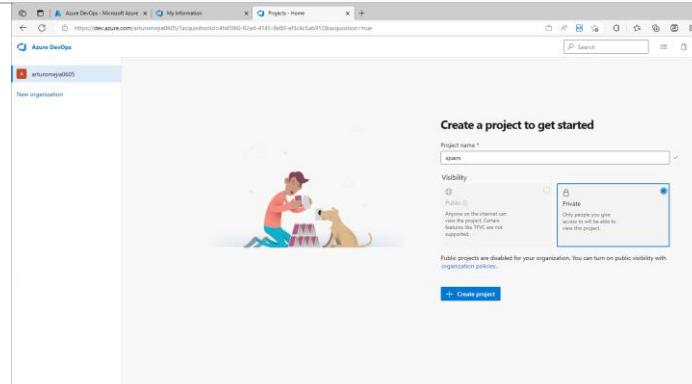
Si no se cuenta con una organización se puede generar para iniciar.



Se generará una organización en la cual se puede trabajar.



Al contar con una Organización se inicia un proyecto. Se ingresa un nombre y se da clic en Create Project.



Dentro del proyecto se va a la opción de Repos y se copia la segunda opción.

A screenshot of the Azure DevOps repository page for the 'spaini' repository. The URL is https://dev.azure.com/arturomejia0605/\_git/spaini. The left sidebar shows options like Overview, Boards, File, Commits, Pushes, Branches, Tags, Pull requests, Pipelines, Test Plans, and Artifacts. The main area has a message: 'spaini is empty. Add some code!'. Below it is a 'Clone to your computer' section with 'HTTPS' and 'SSH' buttons, and a link to the repository's URL. A red box highlights the 'Push an existing repository from command line' section, which contains 'HTTP' and 'SSH' buttons, and a link to 'https://arturomejia0605@dev.azure.com/arturomejia0605/spaini/\_git/spaini'. There are also sections for 'Import a repository' and 'Initialize P main branch with a README or .gitignore'.

Copiamos nuestro texto para hace *push* en la terminal de VS Code.

A screenshot of the Visual Studio Code interface. The left sidebar shows the file structure of the 'SPAADO' project, including 'node\_modules', 'public', 'src', '.gitignore', 'babel.config.js', 'jsconfig.json', 'package-lock.json', 'package.json', 'README.md', and 'vue.config.js'. The main area has sections for 'Inicio' (New file..., Open file..., Open folder..., Clone repository Git...), 'Recente' (fdol, Cyrepos, spaini, C:\repos, spaini2, C:\repos, spa2, C:\Users\artmej\repos, demo, C:\Users\artmej\repos\new), and 'Tutoriales' (Conozca los aspectos básicos, Aumente su productividad, Get Started with Azure in VS..., Get started with Python dev..., Introducción a los cuadernos...). At the bottom, there is a terminal window with the following content:

```
PS C:\repos\spaado> git remote add origin https://arturomejia0605@dev.azure.com/arturomejia0605/spaini/_git/spaini
PS C:\repos\spaado> git push -u origin --all
```

Pedirá autenticación a la cuenta de Azure.

```
PS C:\repos\spaa...> git remote add origin https://arturomejia0605@dev.azure.com/arturomejia0605/_git/spaini
PS C:\repos\spaa...> git push -u origin --all
No refs in common and none specified; doing nothing.
Perhaps you should specify a branch.
Everything up-to-date
PS C:\repos\spaa...> git remote add origin https://arturomejia0605@dev.azure.com/arturomejia0605/_git/spaini
error: remote origin already exists.
PS C:\repos\spaa...> git push -u origin --all
No refs in common and none specified; doing nothing.
Perhaps you should specify a branch.
Everything up-to-date
PS C:\repos\spaa...>
```

Se realiza el sync a nuestro repositorio.

```
PS C:\repos\spaa...> git remote add origin https://arturomejia0605@dev.azure.com/arturomejia0605/_git/spaini
PS C:\repos\spaa...> git push -u origin --all
No refs in common and none specified; doing nothing.
Perhaps you should specify a branch.
Everything up-to-date
PS C:\repos\spaa...> git remote add origin https://arturomejia0605@dev.azure.com/arturomejia0605/_git/spaini
error: remote origin already exists.
PS C:\repos\spaa...> git push -u origin --all
```

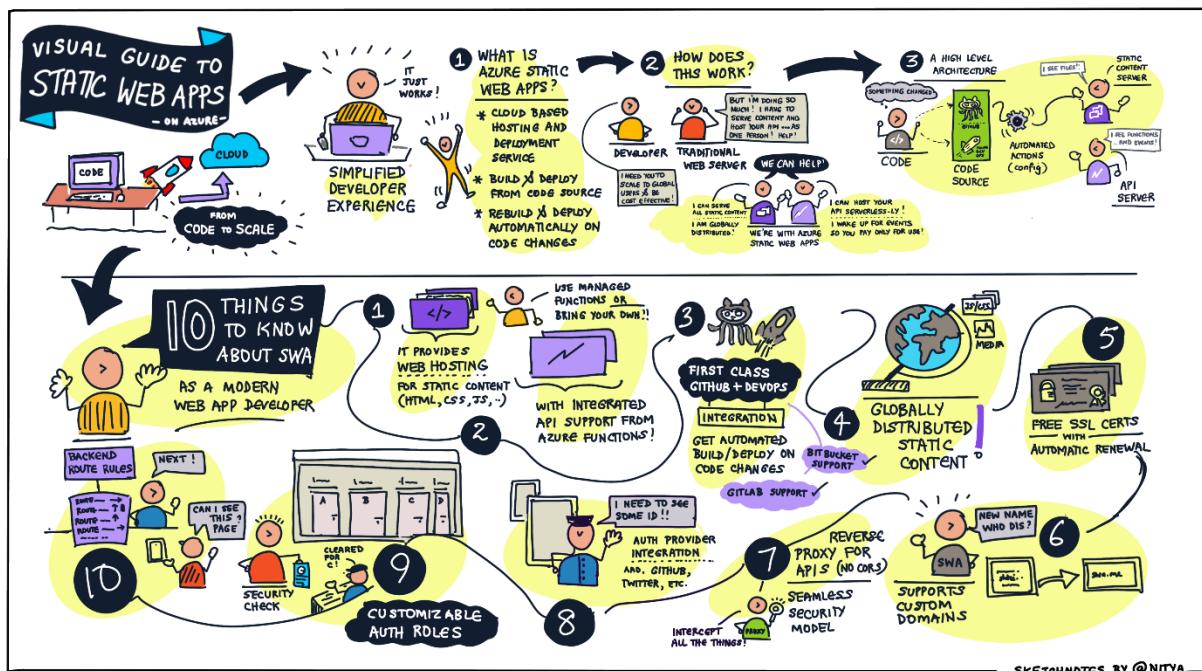
Tras de verificar que se generó el Branch y se envió el código podemos validar en el Repos de ADO.

The screenshot shows the Azure DevOps interface for the repository 'spaini'. The left sidebar lists 'Overview', 'Boards', 'Repos', 'Files', 'Commits', 'Pushes', 'Branches', 'Tags', 'Pull requests', 'Pipelines', 'Test Plans', and 'Artifacts'. The 'Repos' section is selected. The main area shows the repository structure: '.gitignore', 'JS babel.config.js', 'jsonconfig.json', 'package-lock.json', 'package.json', 'MI README.md', and 'JS vue.config.js'. Below each file is its name, last change, and the commit ID '6b3fbc43 init Arturo Mejia'. A search bar at the top right says 'Search'.

## 4. Azure Static Web Apps (SWA)

### ¿Qué es Azure Static Web Apps?

Los Static Web Apps es una solución PaaS que se pensó para simplificar al desarrollador el despliegue de soluciones web.

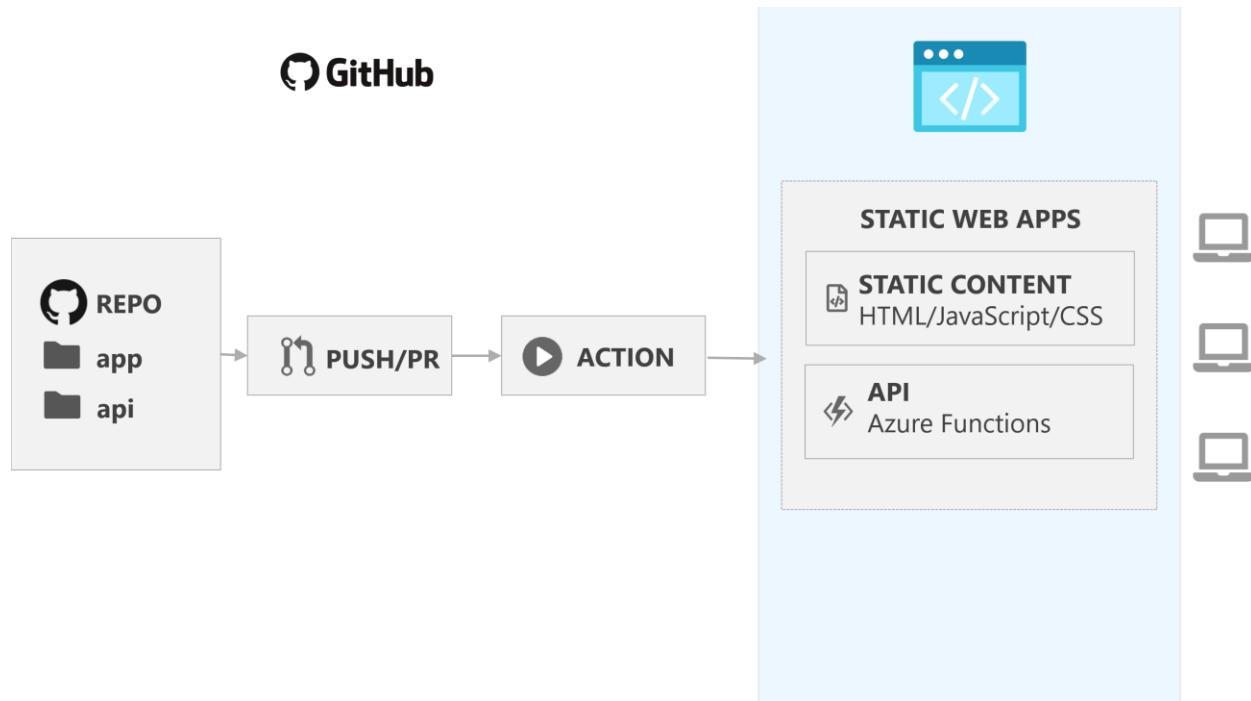


Los Azure Static Web Apps permiten el despliegue y host de aplicaciones integrando elementos de autenticación, autorización, API, Data-API y CI/CD desde su primera implementación.

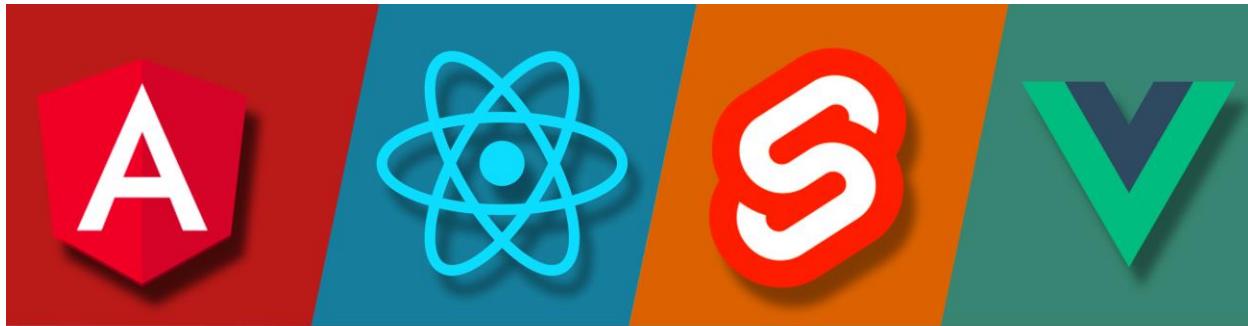
# build and host static web apps authentication and authorization APIs powered by Azure Functions CI/CD



El funcionamiento de un Azure SWA se tiene a partir de un repo hasta el host y distribución segura tipo CDN.

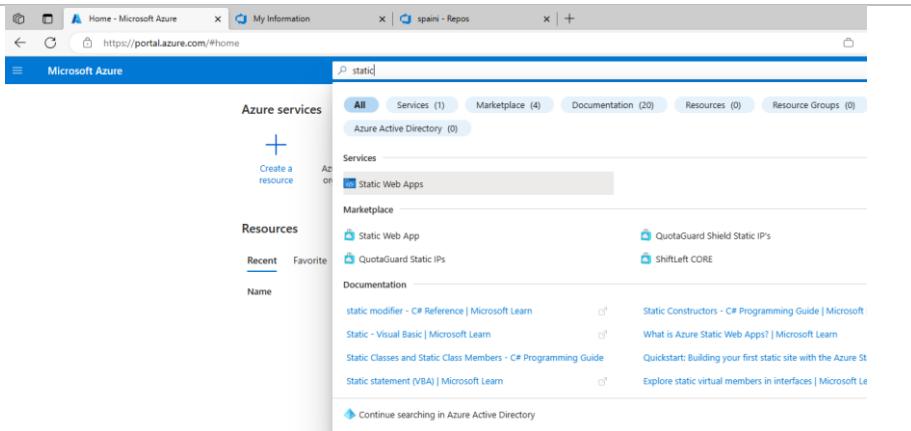


Las soluciones desplegadas soportan el uso de los frameworks como son Angular, Svelte, React, Vue y Blazor. Así como despliegue vía Gatsby como generador de sitios estáticos.



## Despliegue de la primera SWA

En este módulo desplegaremos nuestra SPA en un Azure SWA y se irán integrando los elementos de la plataforma.

Dentro de Azure buscamos la opción de "Static Web apps" y la seleccionamos.	
---	--

**Se da clic en "Crear"**

No static web apps to display

Build modern web applications that automatically publish to the web as your code changes. Deploy your front end applications built with popular frameworks and libraries like React, Vue, Next.js, Nuxt, Blazor and others.

Create static web app

Learn more ↗

Give feedback

**En la siguiente página llenamos el formulario con las siguientes opciones:**

<b>Subscripción</b>	<Selección según el combo>
<b>Resource Group</b>	Generar un nuevo Resource Group ej. RG-CA-SPA-001
<b>Name</b>	Nombre para nuestro recurso de SWA ej. SPA-CA-APP-001
<b>Plan type</b>	Free
<b>Region</b>	Se selecciona una región ej. Central US
<b>Github Account</b>	Se selecciona la ubicación del repositorio

**Create Static Web App - Microsoft Azure**

Basics Tags Review + create

App Service Static Web Apps is a streamlined, highly efficient solution to take your static app from source code to global high availability. Pre-rendered content is distributed globally with no web servers required. [Learn more ↗](#)

**Project Details**

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \*  Azure Pass - Sponsorship  Resource Group \*  (New) RG-CA-SPA-001 [Create new](#)

**Static Web App details**

Name \*  SWA-CA-APP-001

**Hosting plan**

The hosting plan dictates your bandwidth, custom domain, storage, and other available features. [Compare plans](#)

Plan type  Free: For hobby or personal projects  Standard: For general purpose production apps

**Azure Functions and staging details**

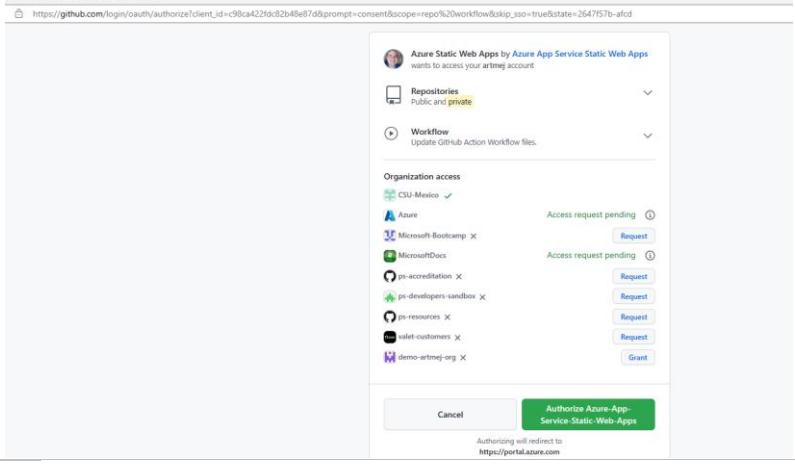
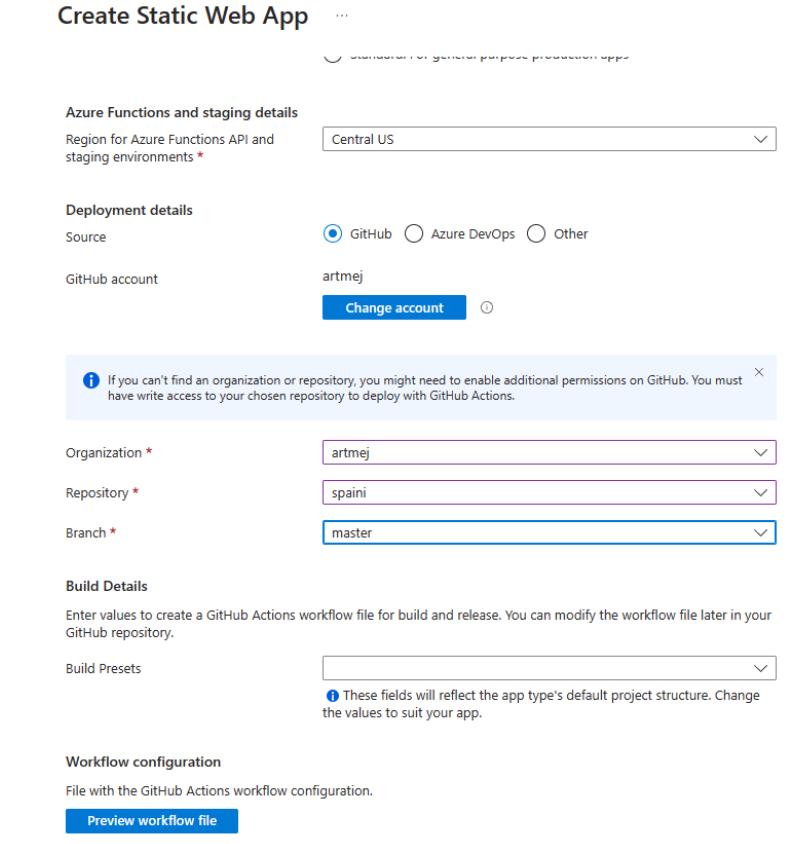
Region for Azure Functions API and staging environments \*  Central US

**Deployment detail**

Source  GitHub  Azure DevOps  Other

GitHub account [Sign in with GitHub](#)

[Review + create](#) < Previous Next : Tags >

<p>Se da clic en el botón de <b>Sign in</b>. En mi caso será <b>Github</b>. Y se pedirá autorización para ingresar a los recursos.</p>	
<p>Al ingresar nos pide los datos del repositorio y Branch estaremos trabajando.</p>	

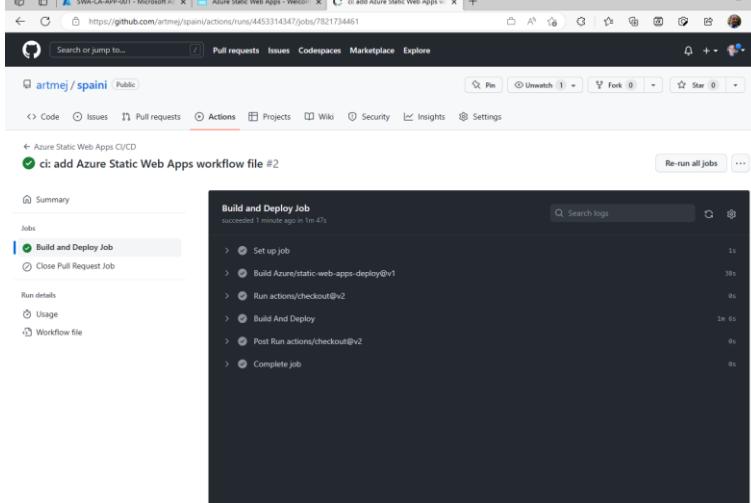
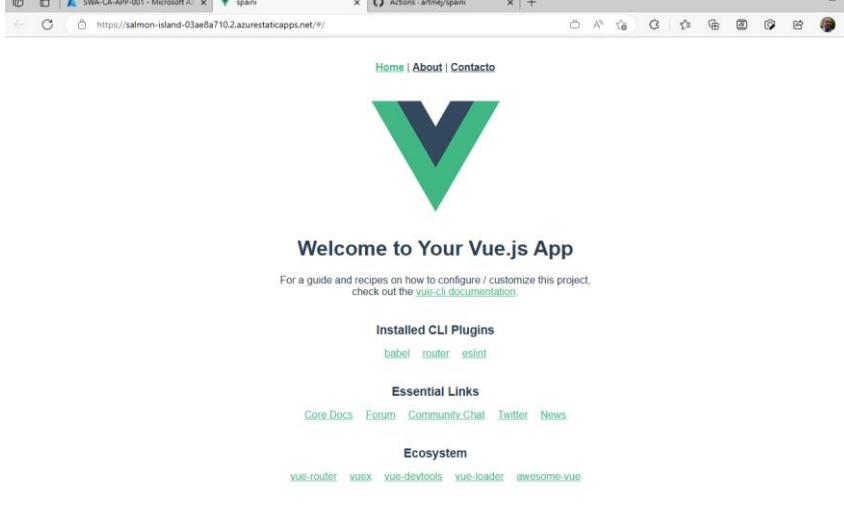
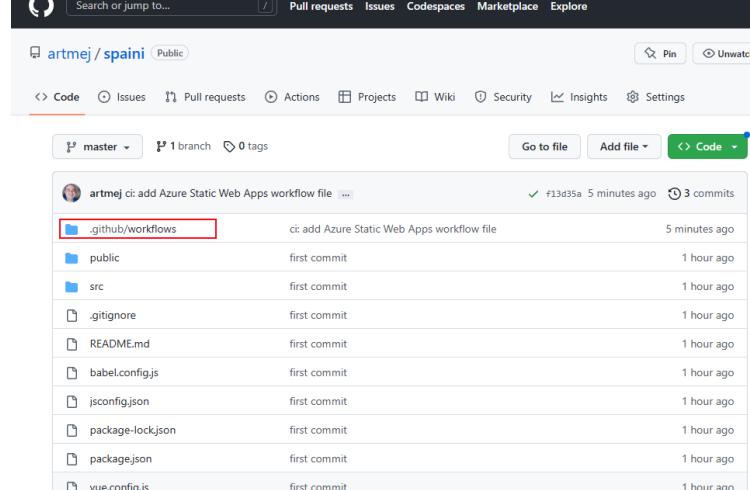
Damos clic a **Review + create** y validará nuestras opciones. Y por último damos **create**.

The screenshot shows the 'Create Static Web App' page in the Azure portal. The top navigation bar includes 'Home > Static Web Apps > Create Static Web App'. Below the navigation, there are tabs: 'Basics', 'Tags', and 'Review + create', with 'Review + create' being the active tab. A 'Summary' section shows the selected template: 'Static Web App by Microsoft'. The 'Details' section lists the configuration: Subscription (d414141c-a82b-4d11-bb47-87781ea4e609), Resource Group (RG-CA-SPA-001), Name (SWA-CA-APP-001), Region (centralus), SKU (Free), Repository (https://github.com/artmej/spaini), Branch (master), App location (/), API location (/), and Output location (/). At the bottom of the page are buttons for 'Create', '< Previous', 'Next >', and 'Download a template for automation'.

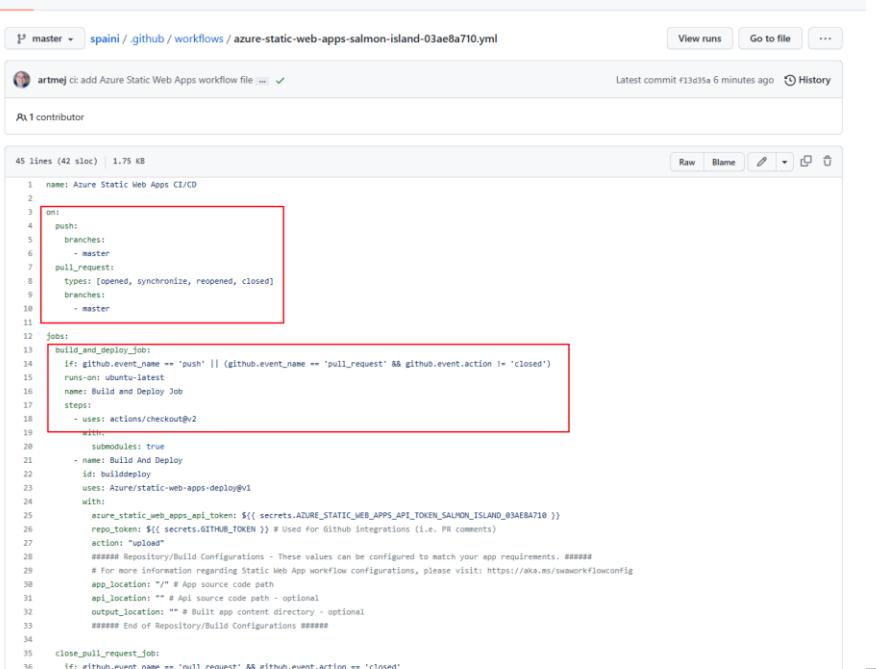
En unos momentos tendremos nuestro recurso en generado con nuestro código.

The screenshot shows the 'Microsoft.Web-StaticApp-Portal-57813cbb-9a71 | Overview' page in the Azure portal. The top navigation bar includes 'Home > Microsoft.Web-StaticApp-Portal-57813cbb-9a71 | Overview'. The page displays deployment details: Deployment name (Microsoft.Web-StaticApp-Portal-57813cbb-9a71), Start time (3/17/2023, 8:55:29 PM), Subscription (Azure Pass - Sponsorship), Correlation ID (7681988d-88d2-467b-8d45-036627bb87), and Resource group (RG-CA-SPA-001). A message indicates 'Your deployment is complete' with a green checkmark. Below this, sections for 'Deployment details' and 'Next steps' are shown, along with a 'Go to resource' button. At the bottom, there are links for 'Give feedback' and 'Tell us about your experience with deployment'.

<p>Se da clic a ir al recurso.</p>	
<p>Se da clic a la URL del recurso y se valida que contamos con url.</p>	
<p>Pero el código aún se está ejecutando en el Despliegue Continuo. Se puede ir a nuestro repositorio y revisar las acciones realizadas para el despliegue.</p>	

<p>Se ingresa al workflow que se esta ejecutamos y revisamos el build y despliegue.</p>	
<p>En este caso se ve exitoso el <i>Action</i> y si recargamos la página ya se verá nuestro contenido.</p>	
<p>Regresando a nuestro Repositorio se puede ver la carpeta <code>.github/workflows</code>. En donde puede revisar cuando se ejecuta nuestro <i>build</i> y que acciones genera.</p>	

Aquí se puede ver que en una acción en branch *master* con un *pull request* se genera esta acción



```

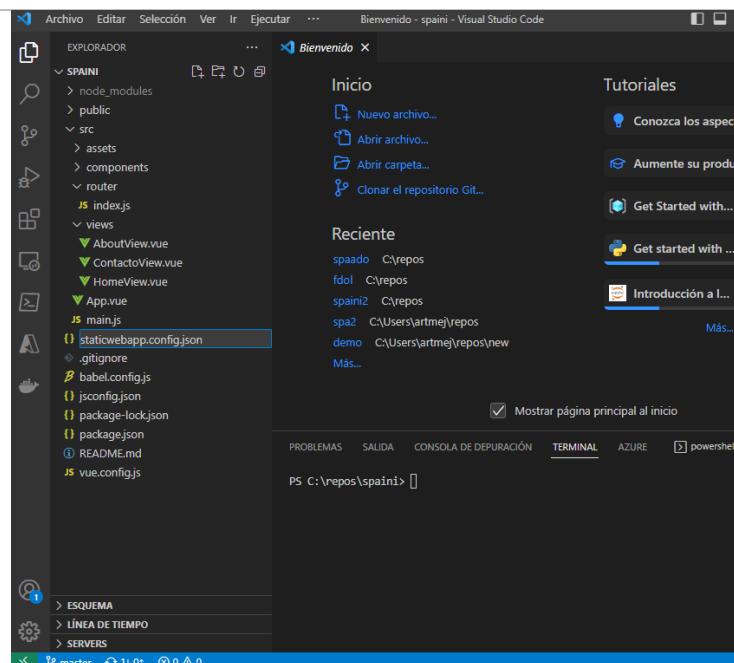
1 name: Azure Static Web Apps CI/CD
2
3 on:
4   push:
5     branches:
6       - master
7   pull_request:
8     types: [opened, synchronize, reopened, closed]
9     branches:
10      - master
11
12 jobs:
13   build_and_deploy_job:
14     if: github.event_name == 'push' || (github.event_name == 'pull_request' & github.event.action != 'closed')
15     runs-on: ubuntu-latest
16     name: Build and Deploy Job
17     steps:
18       - uses: actions/checkout@v2
19         with:
20           submodules: true
21       - name: Build And Deploy
22         id: buildAndDeploy
23         uses: Azure/static-web-apps-deploy@v1
24         with:
25           azure_static_web_apps_api_token: ${{ secrets.AZURE_STATIC_WEB_APPS_API_TOKEN_SALMON_ISLAND_03ae8a710 }}
26           repo_token: ${{ secrets.GITHUB_TOKEN }} # Used for Github Integrations (i.e. PR comments)
27           action: "upload"
28           ##### Repository/Build Configurations - These values can be configured to match your app requirements. #####
29           # For more information regarding Static Web App workflow configurations, please visit: https://aka.ms/swworkflowconfig
30           app_location: /* # App source code path - optional
31           api_location: /* # Api source code path - optional
32           output_location: /* # Built app content directory - optional
33           ##### End of Repository/Build Configurations #####
34
35   close_pull_request_job:
36     if: github.event_name == 'pull_request' & github.event.action == 'closed'

```

## Routing de SWA

Se agregará una configuración de *routing* adicional en el SWA.

Este enrutamiento permite el control en el SWA de rutas opcionales o alias. Para esto se agrega un nuevo archivo al proyecto con nombre *staticwebapp.config.json*



En este archivo se encuentran dos secciones. La primera permite definir una regla de reserva (fallback) y también sus excepciones. La segunda permite el reescribir alguna ruta a otra. Si se quiere más información ver la siguiente URL

[Configuración de Azure Static Web Apps](#)

```
{
    "navigationFallback": {
        "rewrite": "/index.html",
        "exclude": ["*.{css,scss,js,png,gif,ico,jpg,svg}"]
    },
    "routes": [
        {
            "route": "/data",
            "redirect": "/contact",
            "statusCode": 301
        }
    ]
}
```

```

{
    "navigationFallback": {
        "rewrite": "/index.html",
        "exclude": ["*.{css,scss,js,png,gif,ico,jpg,svg}"]
    },
    "routes": [
        {
            "route": "/data",
            "redirect": "/contact",
            "statusCode": 301
        }
    ]
}

```

Se empuja el cambio a nuestro repositorio de Integración continua.

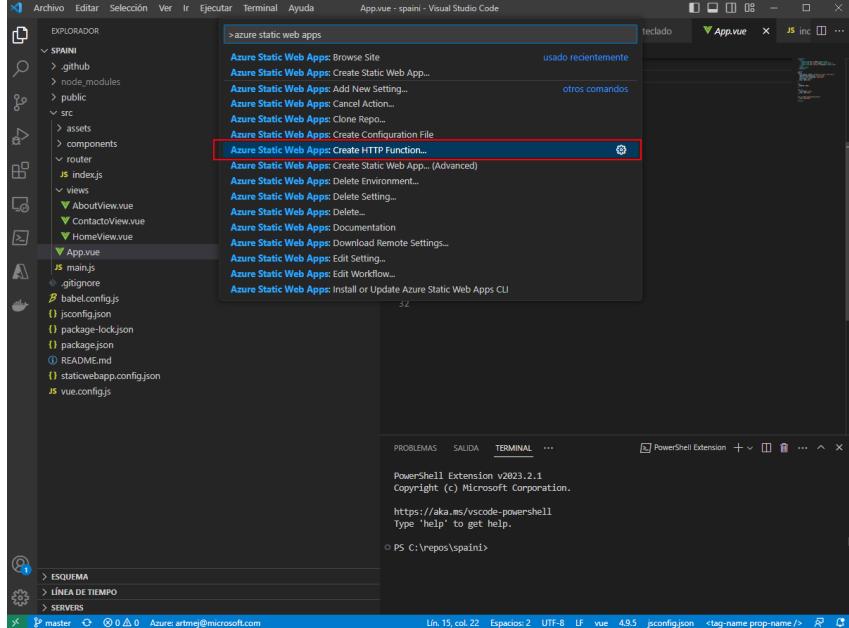
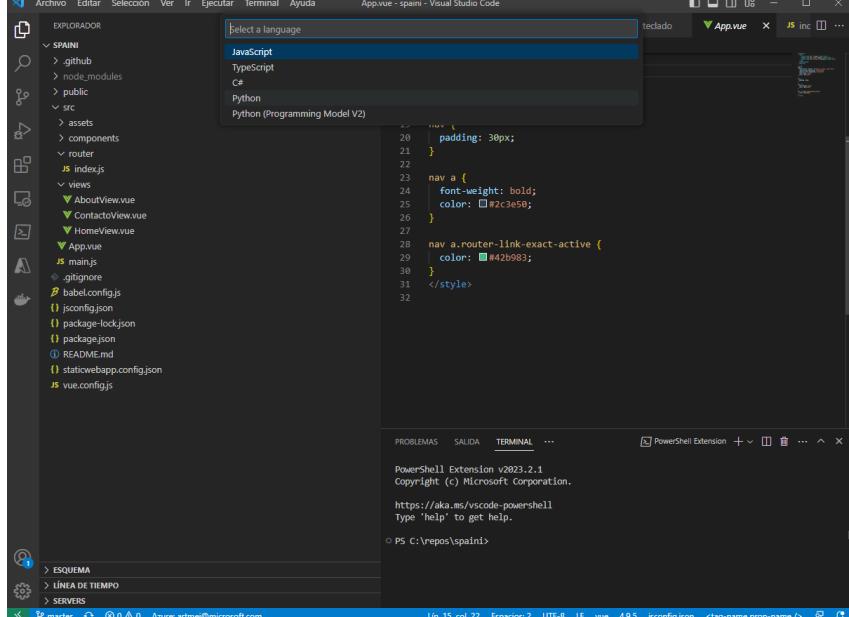
```

{
    "navigationFallback": {
        "rewrite": "/index.html",
        "exclude": ["*.{css,scss,js,png,gif,ico,jpg,svg}"]
    },
    "routes": [
        {
            "route": "/data",
            "redirect": "/contact",
            "statusCode": 301
        }
    ]
}

```

## Crear un API en SWA

En VS Code usaremos la paleta de comandos. Esta se encuentra en el menú ver o Ctrl+Shift+P.

<p>Se buscan los comandos de SWA con el texto “azure static web apps” y se selecciona la opción de Create HTTP Function como se muestra a continuación.</p>	
<p>Se selecciona el lenguaje a usar, en este caso se selecciona Javascript</p>	

Se seleccion el modelo V3.

The screenshot shows the Visual Studio Code interface. In the top right, a modal window titled "Select a JavaScript programming model" is open, with "Model V3" highlighted. The background shows a file tree for a "SPAINI" project containing files like index.js, App.vue, and main.js. The bottom status bar indicates the path "C:\repos\spaini".

Se asigna un nombre en este caso pondremos SPA API.

The screenshot shows the Visual Studio Code interface again. A modal window titled "Create new HTTP trigger (2/4)" is open, prompting for a function name. The input field contains "SPA API". The background shows the same "SPAINI" project structure. The bottom status bar indicates the path "C:\repos\spaini".

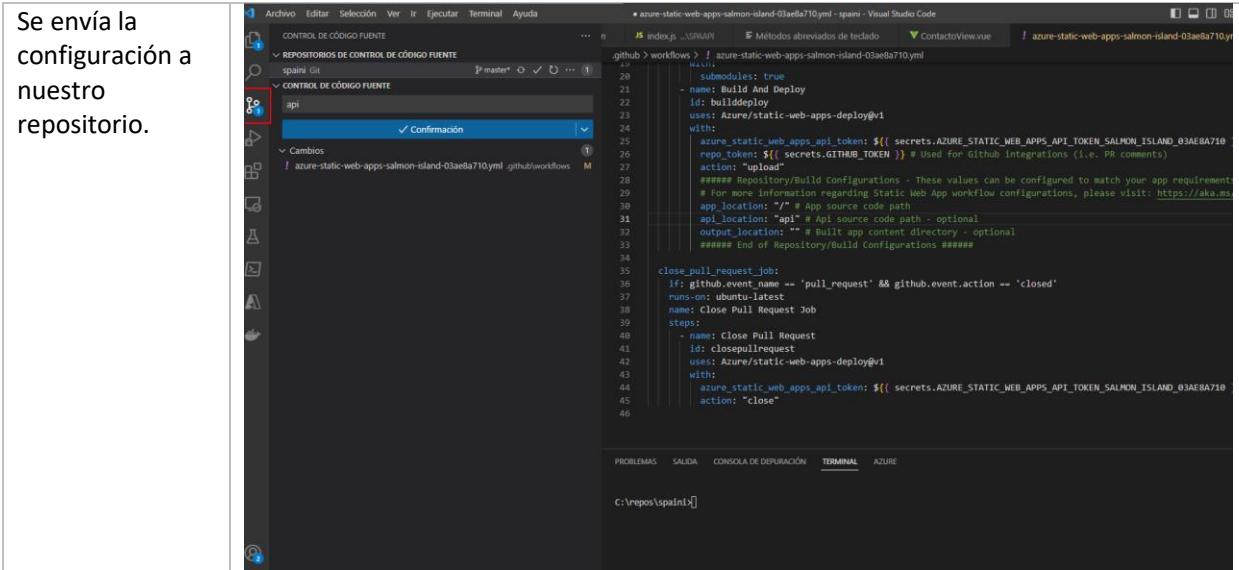
Abrimos el archivo que se generó en la carpeta `api/SPA API` llamado `index.js` y reescribimos el código con lo siguiente:

```
module.exports = 
async function
(context, req) {
    context.res.json(
{
    text: "Hello
from the API"
});
}
```

## Configurar carpeta API para el SWA

Ya que contamos con un API generado requerimos indicarle al Action en que carpeta se encuentran nuestros API.

Se selecciona el archivo `.yml` en la carpeta `github\workflows` y se busca la sección `jobs/steps` el atributo `api_locations` se pone el valor `api`



## Agregar un elemento de prueba

Se agrega una llamada de prueba en la vista de contacto para validar que nuestro API funciona correctamente. Agregaremos en el archivo de vista la siguiente sección de script:

```
<script>
export default {
  name: "App",
  data() {
    return {
      message: ""
    };
  },
  async mounted() {
    const { text } = await (await fetch("/api/info")).json();
    this.message = text;
  }
};
</script>
```

Viéndose de la siguiente manera dentro de la vista:

The screenshot shows the Visual Studio Code interface with the following details:

- Project Explorer:** Shows the project structure with files like .github\workflows, .vscode, api, SPAPI, index.json, sample.dat, .funcignore, host.json, local.settings.json, package-lock.json, package.json, dist, node\_modules, public, src, assets, logo.png, components (HelloWorld.vue), router, views (AboutView.vue, ContactoView.vue, HomeView.vue), App.vue, main.js, .gitignore, babel.config.js, jsonconfig.json, and package-lock.json.
- Code Editor:** Displays the 'ContactoView.vue' component code. The template section contains a single-line message. The script section uses Fetch API to get info from the '/api/info' endpoint and sets it to the 'message' variable.
- Terminal:** Shows the command 'C:\repos\spaini>'.
- Status Bar:** Shows the file is 'master\*' and the user is 'Azure: armej@microsoft.com'.

Se agrega a function.json un alias en forma opcional como un atributo "route"

```
{
  "bindings": [
    {
      "authLevel": "anonymous",
      "type": "httpTrigger",
      "direction": "in",
      "name": "req",
      "methods": [
        "get",
        "post"
      ],
      "route": "info"
    },
    {
      "type": "http",
      "direction": "out",
      "name": "res"
    }
  ]
}
```

## Autenticación de SWA

Azure SWA permite una integración fácil con los siguientes:

### Identity provider

Azure Active Directory

GitHub

Twitter

### Login route

/.auth/login/aad

/.auth/login/github

/.auth/login/twitter

Esto quiere decir que al agregar una petición al servicio nos genera una integración a la autenticación del usuario. Este es un ejemplo de cómo realizarlo:

```
<a href="/.auth/login/github">Login</a>
```

En nuestra aplicación se agregará el *login* y *logout* para ver la integración. Se realizará por el *login* de *github* pero se puede usar cualquiera de los proveedores integrados. Para esta acción se modificará el archivo *App.vue* para agregar a la navegación las siguientes secciones:

Sección en nav después de contacto.	<pre>        &lt;a v-if="!userInfo"         :href="`/.auth/login/github?post_login_redirect_uri=\${redirect}`"&gt; Login &lt;/a&gt;       &lt;template v-if="userInfo"&gt;         &lt;a v-if="userInfo" :href="`/.auth/logout`"&gt; Logout       &lt;/a&gt;           {{ userInfo.userDetails }}       &lt;/template&gt;</pre>
Sección de Script.	<pre>&lt;script&gt; export default {   data() {     return {       userInfo: {         type: Object,         default() { },       },       redirect: window.location.pathname,     };   },   methods: {     async getUserInfo() {       try {         const response = await fetch("/.auth/me");         const payload = await response.json();         const { clientPrincipal } = payload;         return clientPrincipal;       }       catch (error) {         console.error("No profile could be found");         return undefined;       }     }   } &lt;/script&gt;</pre>

```

        }
    },
},
async created() {
    this.userInfo = await this.getUserInfo();
}
};

</script>

```

Tras agregar estas secciones se puede ver el archivo de la siguiente forma:

```

<template>
  <nav>
    <router-link to="/">Home</router-link>
    <router-link to="/about">About</router-link>
    <router-link to="/contact">Contact</router-link>
    <a v-if="userInfo" href="#" @click="githubLogout">Logout</a>
  </template>

<script>
export default {
  data() {
    return {
      userInfo: {
        type: Object,
        default: () => {}
      },
      redirect: window.location.pathname,
      clientPrincipal: null
    };
  },
  methods: {
    async getUserInfo() {
      try {
        const response = await fetch('/auth/me');
        const payload = await response.json();
        const { clientPrincipal } = payload;
        return clientPrincipal;
      } catch (error) {
        console.error('No profile could be found');
        return undefined;
      }
    },
    async created() {
      this.userInfo = await this.getUserInfo();
    }
  }
};
</script>

```

Para demostración en el código se agregará en `staticwebapp.config.js` en la configuración el uso de las rutas solo a usuarios autenticados. Se asegura nuestra página de contacto las siguientes rutas

```

,
{
  "route": "/api/*",
  "allowedRoles": ["authenticated"]
},
{
  "route": "/contacto",
  "allowedRoles": ["authenticated"]
}

```

Viéndose el json de la siguiente manera:

Viéndose el json de la siguiente manera:

```
staticwebapp.config.json
1  "navigationFallback": {
2    "rewrite": "/index.html",
3    "exclude": ["*.css, *.scss, *.js, *.png, *.gif, *.ico, *.jpg, *.svg"]
4  },
5  "routes": [
6    {
7      "route": "/data",
8      "redirect": "/contacto",
9      "statusCode": 301
10   },
11   {
12     "route": "/logueo",
13     "redirect": "../../auth/login/github"
14   },
15   {
16     "route": "/logueo",
17     "redirect": "/auth/logout"
18   },
19   {
20     "route": "/api/*",
21     "allowedRoles": ["authenticated"]
22   },
23   {
24     "route": "/contacto",
25     "allowedRoles": ["authenticated"]
26   }
27 ],
28 }
29
30 }
```

## Acceso de usuario en API

También se puede el usuario autenticado y enviarlo a nuestra API. Cambiaremos el servicio api/index.js el siguiente código:

```
module.exports = async function (context, req) {
    const header = req.headers['x-ms-client-principal'];
    const encoded = Buffer.from(header, 'base64');
    const decoded = encoded.toString('ascii');

    context.res.json({
        text: JSON.parse(decoded)
    });
};
```

El resultado es el siguiente documento:

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer (Left):** Shows the project structure for "staticwebapp.spain". It includes:
  - SPA API folder with files: .github\workflows, .vscode, api, function.json, index.js, sample.dat, .ignore, host.json, localSettings.json, package-lock.json, package.json, README.md, staticwebapp.config.json, swa-cl-config.json, and vue.config.js.
  - src folder with assets (logo.png), components (HelloWorld.vue), router, views (AboutView.vue, ContactoView.vue, HomeView.vue), App.vue, main.js, .ignore, babel.config.js, jsonconfig.json, package-lock.json, package.json, and README.md.
- Code Editor (Center):** Displays the content of index.js under the SPA API folder. The code is as follows:
 

```

1 module.exports = async function (context, req) {
2   const header = req.headers['x-ms-client-principal'];
3   const encoded = Buffer.from(header, 'base64');
4   const decoded = encoded.toString('ascii');
5
6   context.res.json({
7     text: JSON.parse(decoded)
8   });
9 };
      
```
- Terminal (Bottom):** Shows the command line interface with the current directory as C:\repos\spain\index.js. It also shows environment variables like powershell and cmd.

Y como resultado de la llamada se puede ver lo que se puede decodificar en el lado de API.

The screenshot shows a browser window with the following details:

- Address Bar:** https://salmon-island-03ae8a7102.azurestaticapps.net/contacto
- Content Area:** Displays the page "Gracias por ponerse en contacto con nosotros." Below it, a JSON object is shown in a red-bordered box:
 

```
{"identityProvider": "github", "userId": "aa28268a65e845dead93b6d1bca7a198", "userDetails": "artmej", "userRoles": ["anonymous", "authenticated"]}
```

## Roles

Azure SWA puede realizar asignación de roles por invitación a usuarios en específico. Esto se realiza en Azure en la sección de *Role Assginment* y se realiza una *invitación*.

Microsoft Azure

Home > SWA-CA-APP-001

**SWA-CA-APP-001 | Role management**

Static Web App

Search  Invite Refresh Delete

Overview Access control (IAM) Tags Diagnose and solve problems Settings Configuration Application Insights Custom domains APIs Database connection (preview) Environments **Role management** Identity Enterprise-grade edge Hosting Plan Private endpoints Locks Automation

User Identity ↑ Authentication Provider ↑ Role ↑

No results.

Nos presenta un formulario donde se indica el proveedor de autenticación, usuario, id del rol y tiempo de expiración de invitación. Asignando este usuario con un nuevo role al final

Microsoft Azure

Home > SWA-CA-APP-001

**SWA-CA-APP-001 | Role management**

Static Web App

Search  Invite Refresh Delete

Overview Access control (IAM) Tags Diagnose and solve problems Settings Configuration Application Insights Custom domains APIs Database connection (preview) Environments **Role management** Identity Enterprise-grade edge Hosting Plan Private endpoints Locks Automation

User Identity ↑ Authentication Provider ↑ Role ↑

No results.

Create invitation link

Invitation links grant specific users access to your domain. You can specify when the links expire.

Authentication provider \* GitHub

User handle \* artmej

Domain \* salmon-island-03ae8a710.2.azurestaticapp...

Role \* adminSWA

Invitation expiration (in hours) \* 1

Generate Close

Se acepta la invitación por parte del usuario y se puede ver como se ve el nuevo rol en la página de prueba.

## Gracias por ponerse en contacto con nosotros.

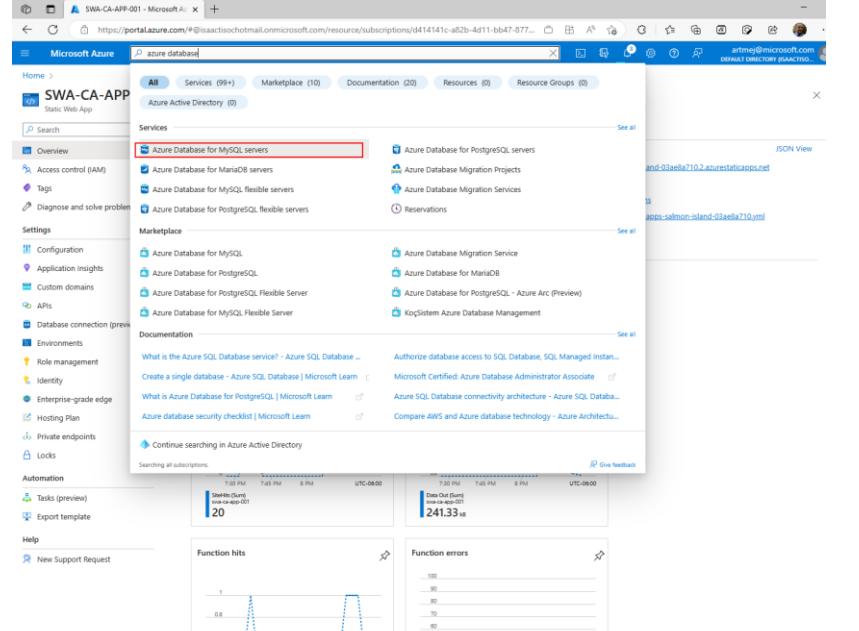
```
{"identityProvider": "github", "userId": "aa28268a65e845dead93b6d1bca7a198", "userDetails": "artmej", "userRoles": ["adminswa", "anonymous", "authenticated"] }
```

## Integración con una Base de datos

Static Web Apps tiene una nueva funcionalidad en preview que permite generar un servicio rest y/o graphql automáticamente. Esto facilita el acceso y configuración a las bases de datos.

### Generar base de datos

El proceso de acceso empieza contando con una base de datos, se genera una base tipo MySQL flex



The screenshot shows the Microsoft Azure portal interface. The user has searched for 'azure database'. In the search results, 'Azure Database for MySQL servers' is highlighted with a red box. Other options listed include 'Azure Database for PostgreSQL servers', 'Azure Database Migration Projects', 'Azure Database Migration Services', and 'Azure Database for PostgreSQL Flexible servers'. On the left sidebar, there are various navigation links such as Overview, Access control (IAM), Tags, Diagnose and solve problems, Settings, Configuration, Application insights, Custom domains, APIs, Database connection (preview), Environments, Role management, Identity, Enterprise-grade edge, Hosting Plan, Private endpoints, Locks, Automation, Tasks (preview), Export template, Help, and New Support Request. At the bottom of the page, there are two charts: 'Function hits' and 'Function errors', both showing data over time from 7:30 PM to 8:00 PM UTC-08:00.

Se pide el generar una nueva base mysql del menú.

The screenshot shows the Azure portal interface for managing MySQL servers. The title bar reads "Azure Database for MySQL servers". The main content area displays a message: "No Azure Database for MySQL servers to display. Try changing or clearing your filters." Below this message is a "Create Azure Database for MySQL server" button. At the top of the page, there are several filter options: "Save the current columns, sorting, filtering and summary as a view and access your saved views here.", "resource group equals all", "Location equals all", and "Add filter". There are also buttons for "Create", "Manage view", "Refresh", "Export to CSV", "Open query", and "Assign tags". The bottom of the page includes sorting and grouping controls: "Name ↑", "Resource type ↑", "Status ↑", "High availability ↑", "Resource group ↑", and "Location ↑".

Se selecciona una base de datos tipo flex.

The screenshot shows the Azure portal interface for creating a new MySQL server. The title bar reads "Select Azure Database for MySQL deployment option". The main content area displays two options under "How do you plan to use the service?": "Flexible server" and "Wordpress + MySQL Flexible server". Each option has a "Create" button and a "Learn More" link. A note at the top of the page states: "Azure Database for MySQL - Single Server is scheduled for retirement by September 16, 2024. Learn More".

Se genera una base para Desarrollo, al final del formulario se pide un usuario y contraseña, hay que copiarlos pues más adelante se usarán

The screenshot shows the Microsoft Azure portal interface for creating a MySQL flexible server. The process is divided into several steps:

- Basics:** Substitution: Azure Pass - Sponsorship; Resource group: RG-CA-SPA-001.
- Project details:** Server name: mysql-ca-spa-001; Region: East US; MySQL version: 5.7; Workload type: For development or hobby projects; Compute + storage: Burstable, B1ms (1 vCore, 2 GiB RAM, 20 GiB storage, 360 IOPS); Geo-redundancy: Disabled.
- High availability:** No preference selected.
- Authentication:** MySQL authentication only selected; Admin username: mydemouser; Password: [redacted].
- Estimated costs:**

Compute Sku	MXN
Standard_B1ms (1 vCore)	261.85

Storage	MXN
Storage selected 20 GiB (MXN 2.43 per GiB)	48.54

Backup Retention	MXN
Backup retention is billed based on additional storage used for retaining backups.	310.39

En la sección de networking es importante permitir los accesos los servicios de Azure.

Se continua con “Review + Create” y después “Create”.

**Networking**

Allow public access from any Azure service within Azure to this server

Firewall rule name	Start IP address	End IP address
ClientIP_Address_2023-3-24_20-28-32	189.186.189.189	189.186.255.255

**Estimated costs**

Compute SKU	MXN
Standard_B1ms (1 vCore)	261.85/month
Storage	MXN 48.54/month
Storage selected 20 GiB (MXN 2.43 per GiB)	20 x 2.43
Backup Retention	
Bandwidth	

**Estimated total** MXN 310.39/month

Al terminar se agrega una nueva base de datos llamada *personasdb*

**Databases**

+ Add

Name	Character set	Collation	Schema type
information_schema	utf8	utf8_general_ci	System
mysql	latin1	latin1_swedish_ci	System
performance_schema	utf8	utf8_general_ci	System
sys	utf8	utf8_general_ci	System

Se salva la nueva base de datos.

The screenshot shows the Microsoft Azure portal interface for creating a database. The URL is https://portal.azure.com/#/resource/subscriptions/d414141c-a82b-4d1... . The left sidebar shows the 'Databases' section selected under 'Settings'. The main area is titled 'Create Database' with fields for 'Name' (set to 'personasdb'), 'Character set' (set to 'utf8'), and 'Collation' (set to 'utf8\_general\_ci'). A note at the top states: 'You can create, view and deleting MySQL databases on this server. Note mysql.sys.information\_schema.performance\_schema. You can connect to this database using MySQL client tools.' A preview table shows existing system databases: information\_schema, mysql, performance\_schema, and sys.

Se abre un Azure cli para agregar un par de registros.

The screenshot shows the Microsoft Azure portal interface with the Azure CLI integrated. The URL is https://portal.azure.com/#/resource/subscriptions/d414141c-a82b-4d1... . The left sidebar shows the 'Databases' section selected under 'Settings'. The main area displays the 'Databases' list, where 'personasdb' is now listed as a 'User' schema. Below the list is a PowerShell terminal window showing the command 'Requesting a Cloud Shell. succeeded.' and 'Connecting terminal...'. The PowerShell prompt is PS C:\>.

Se conecta a la base de datos con el siguiente comando:

```
mysql -h
<<base>>.mysql.database.azure.com -u
<<usuario>>-p
```

Sustituyendo los parámetros de base y usuario.

The screenshot shows the Microsoft Azure portal interface for managing a MySQL flexible server named 'msql-ca-spa-001'. The portal provides a detailed overview of the server's configuration, including its subscription, resource group, status, and location. Below the portal, a PowerShell window is open, displaying the command used to connect to the MySQL database via the Azure portal, followed by the MySQL prompt and some initial commands.

```
F:\> mysql -h msql-ca-spa-001.mysql.database.azure.com -u mydemouser -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 24
Server version: 5.7.41-log MySQL Community Server (GPL)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE TABLE PersonasTabla (
    Id INT AUTO_INCREMENT NOT NULL,
    Nombre VARCHAR(25) NULL,
    PRIMARY KEY (Id)
);

mysql> INSERT INTO PersonasTabla (Nombre)
VALUES ('Leonel');

mysql> INSERT INTO PersonasTabla (Nombre)
VALUES ('Arturo');
```

Se selecciona la base de datos usando el comando “use personasdb”. Se genera la tabla de trabajo y un par de registros.

```
CREATE TABLE PersonasTabla (
    Id INT AUTO_INCREMENT NOT NULL,
    Nombre VARCHAR(25) NULL,
    PRIMARY KEY (Id)
);

INSERT INTO PersonasTabla (Nombre)
VALUES ('Leonel');

INSERT INTO PersonasTabla (Nombre)
VALUES ('Arturo');
```

Teniendo una terminal con una vista similar

```
PowerShell | ⌂ ? ⌂ {} ⌂
Your MySQL connection id is 28
Server version: 5.7.41-log MySQL Community Server (GPL)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input

mysql> use personasdb
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> INSERT INTO PersonasTable (Nombre)
      -> VALUES ('Leonel');
Query OK, 1 row affected (0.05 sec)

mysql>
mysql> INSERT INTO PersonasTable (Nombre)
      -> VALUES ('Arturo');
Query OK, 1 row affected (0.04 sec)

mysql>
```

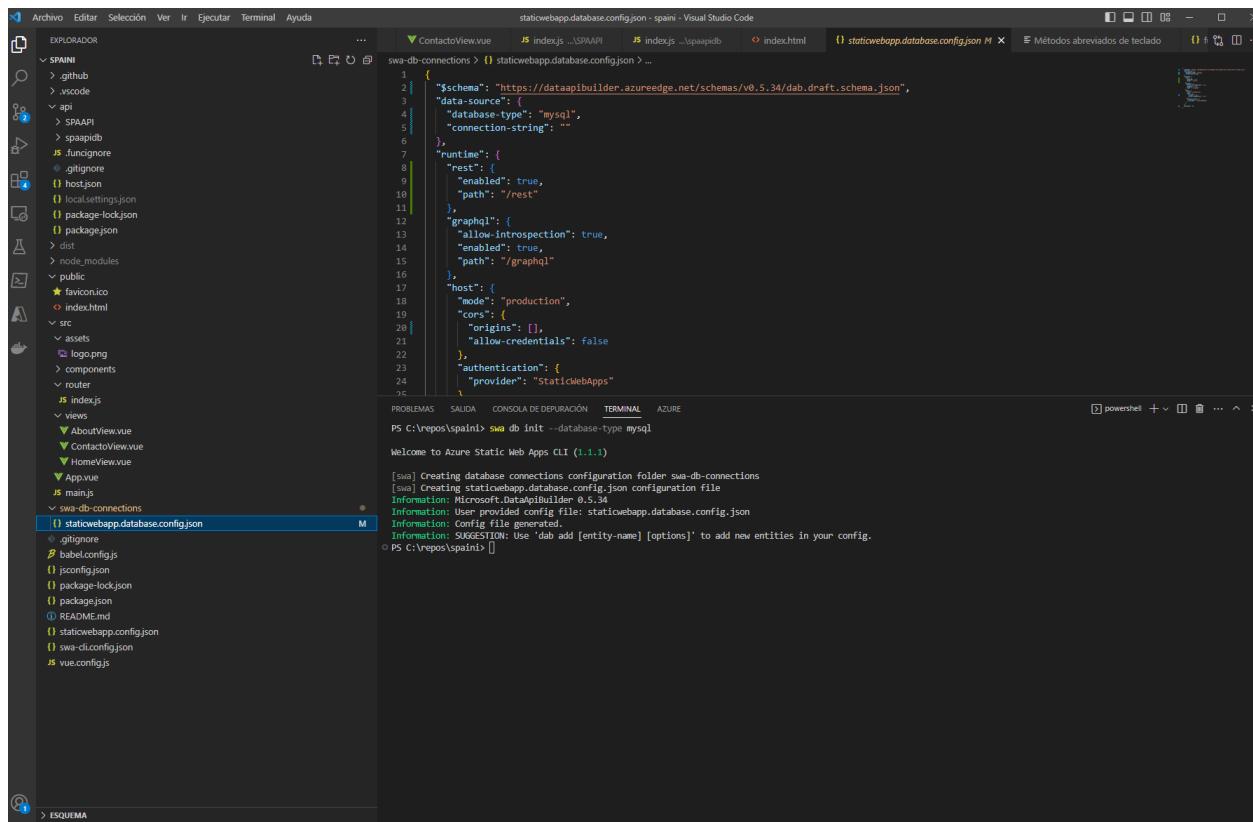
Se da "quit" y se puede cerrar el az cli.

Configuración del código para acceso a la base

El siguiente paso es configurar al código. El primer paso es ejecutar el siguiente comando que nos genera el template de configuración del data-api.

```
swa db init --database-type mysql
```

Se puede observar que se generó un nuevo archivo en archivo *swa-db-connections/satitcwebapp.database.config.json* donde tendremos todas las configuraciones.



Se agregará la siguiente configuración al archivo:

```
{
  "$schema": "https://dataapibuilder.azureedge.net/schemas/latest/dab.draft.schema.json",
  "data-source": {
    "database-type": "mysql",
    "options": {
      "set-session-context": false
    },
    "connection-string": "@env('DATABASE_CONNECTION_STRING')"
  },
  "runtime": {
    "rest": {
      "enabled": true,
      "path": "/rest"
    },
    "graphql": {
      "allow-introspection": true,
      "path": "/graphql"
    }
  }
}
```

```
        "enabled": true,
        "path": "/graphql"
    },
    "host": {
        "mode": "production",
        "cors": {
            "origins": ["http://localhost:4280"],
            "allow-credentials": false
        },
        "authentication": {
            "provider": "StaticWebApps"
        }
    }
},
"entities": {
    "Persona": {
        "source": "PersonasTabla",
        "permissions": [
            {
                "actions": ["*"],
                "role": "anonymous"
            }
        ]
    }
}
}
```

Quedando de la siguiente forma:

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer (Left):** Shows the project structure with files like .github, .vscode, SPAPI, api, SPAPIdb, .funcignore, host.json, localSettings.json, package-lock.json, package.json, dist, node\_modules, public, favicon.ico, index.html, src, assets, logo.png, components, router, index.js, views, AboutView.vue, ContactView.vue, HomeView.vue, App.vue, main.js, swa-db-connections, staticwebapp.database.config.json, .gitignore, babel.config.js, jsonconfig.json, package-lock.json, package.json, README.md, staticwebapp.config.json, swa-clc.config.json, and vue.config.js.
- Code Editor (Right):** Displays the content of staticwebapp.database.config.json. The code defines a database connection named "swa-db-connections" with the following properties:
  - \$schema:** "https://dataapibuilder.azureedge.net/schemas/latest/dab.draft.schema.json"
  - data-source:** { "database-type": "mysql", "options": { "set-session-context": false } }
  - connection-string:** "@env('DATABASE\_CONNECTION\_STRING')"
  - runtime:** { "rest": { "enabled": true, "path": "/rest" } }, "graphql": { "allow-introspection": true, "enabled": true, "path": "/graphql" }, "host": { "mode": "production", "cors": { "origins": ["http://localhost:4280"], "allow-credentials": false } }, "authentication": { "provider": "StaticWebApps" } }
  - entities:** { "Personal": { "source": "PersonasTabla", "permissions": [ { "actions": ["\*"], "role": "anonymous" } ] } }

Agregaremos un botón en Nuestra vista views/Contactoview.vue:

```
<button id="list" onclick="list()">List</button>
```

Teniendo esta vista

' is highlighted with a red box."/>

```

<template>
  <div class="contacto">
    <h1>Gracias por ponerte en contacto con nosotros.</h1>
    {{ message }}
    <button id="list" onclick="list()">>List</button>
  </div>
</template>
<script>
export default {
  name: "App",
  data() {
    return {
      message: ""
    };
  },
  async mounted() {
    const { text } = await await fetch("/api/info").json();
    this.message = text;
  }
}</script>

```

Y agregaremos un Script a public/index.html

```

<script>
async function list() {

const query = `

  personas {
    items {
      Id
      Nombre
    }
  }
`;

const endpoint = "/data-api/graphql";
const response = await fetch(endpoint, {
  method: "POST",
  headers: { "Content-Type": "application/json" },
  body: JSON.stringify({ query: query })
});`;
```

```

const result = await response.json();
console.table(result.data.personas.items);
}
</script>

```

Quedando de la siguiente manera

The screenshot shows the Visual Studio Code interface with the file `index.html` open. The code editor displays the following content:

```

<!DOCTYPE html>
<html lang="">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE-edge">
<meta name="viewport" content="width=device-width,initial-scale=1.0">
<link rel="icon" href="<%= BASE_URL %>/favicon.ico">
<title><%= htmlWebpackPlugin.options.title %></title>
</head>
<body>
<noscript>
| <strong>We're sorry but <%= htmlWebpackPlugin.options.title %> doesn't work properly without JavaScript enabled. Please enable<br>
| <strong>JavaScript</strong> in your browser.</strong>
</noscript>
<div id="app"></div>
<!-- built files will be auto injected -->
<script>
async function list() {
  const query = `

    personas {
      items {
        Id
        Nombre
      }
    }
  `;

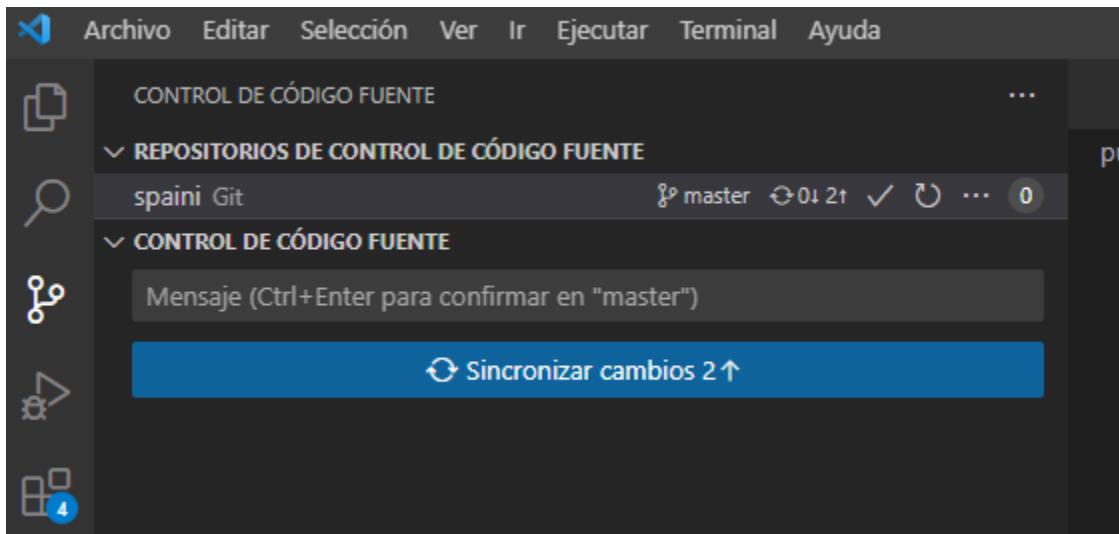
  const endpoint = "/data-api/graphql";
  const response = await fetch(endpoint, {
    method: "POST",
    headers: { "Content-Type": "application/json" },
    body: JSON.stringify({ query: query })
  });
  const result = await response.json();
  console.table(result.data.personas.items);
}

</script>
</body>
</html>

```

A GitHub commit dialog is visible in the center of the screen, prompting for confirmation to push changes to the "master" branch. The commit message is "Confirmación".

Enviamos nuestros cambios al repositorio.



Entramos al portal y ligamos nuestra base de datos:

Microsoft Azure | Search resources, services, and docs (G+)

Home > SWA-CA-APP-001

## SWA-CA-APP-001 | Database connection (preview)

Static Web App

Search Refresh Send us your feedback

Azure Static Web Apps allows user-provided database resources for your Static Web App. [Learn more](#)

**Production**

Every environment can at most have one linked database

Environment name ↑	Database type ↑	Database resource name ↑	Action ↑
Production		msq...	<a href="#">Link existing da...</a>

**Preview deployments**

Open pull requests against the linked repository or configure your workflow to trigger on branch code changes to deploy to a preview environment.

Environment name ↑	Database type ↑	Database resource name ↑	Action ↑

Se da clic en *link existing database* y llenamos el formulario:

Microsoft Azure | Search resources, services, and docs (G+)

Home > SWA-CA-APP-001

## SWA-CA-APP-001 | Database connection (preview)

Static Web App

Search Refresh Send us your feedback

Azure Static Web Apps allows user-provided database resources for your

**Production**

Every environment can at most have one linked database

Environment name ↑	Database type ↑
Production	

**Preview deployments**

Open pull requests against the linked repository or configure your workflow

Environment name ↑	Database type ↑

**Link existing database**

Environment name: Production

Database type \*: Azure Database for MySQL (flex)

Subscription \*: Azure Pass - Sponsorship

Resource group \*: rg-ca-spa-001

Resource name \*: msql-ca-spa-001

Database name \*: personasdb

Authentication type \*: Connection String

Username \*: mydemouser

Password \*: \*\*\*\*\*

You must complete the following steps manually to ensure a successful database connection:

- Your database's network must be configured to allow access.
- Your database connection configuration has been added to your repository.

[Learn more](#)

I acknowledge the steps needed to enable this static web app to connect to my database.

**Link** **Cancel**

Entramos a nuestra SPA para dar clic en el nuevo botón y observamos en nuestras herramientas del navegador que se hizo la consulta, esta prueba usa graphql:

The screenshot shows a browser window with two tabs: 'spaini' and 'https://salmon-island-03ae8a710.2.azurestaticapps.net/contacto'. The main content area displays a success message: 'Gracias por ponerte en contacto con nosotros.' Below it is a JSON object representing user details. To the right, the developer tools' 'Console' tab is open, showing the results of a GraphQL query. The results are displayed in a table with columns 'Index', 'Id', and 'Nombre'. There are two rows: one for 'Leonel' (Id: 1) and another for 'Arturo' (Id: 2).

Index	Id	Nombre
0	1	'Leonel'
1	2	'Arturo'

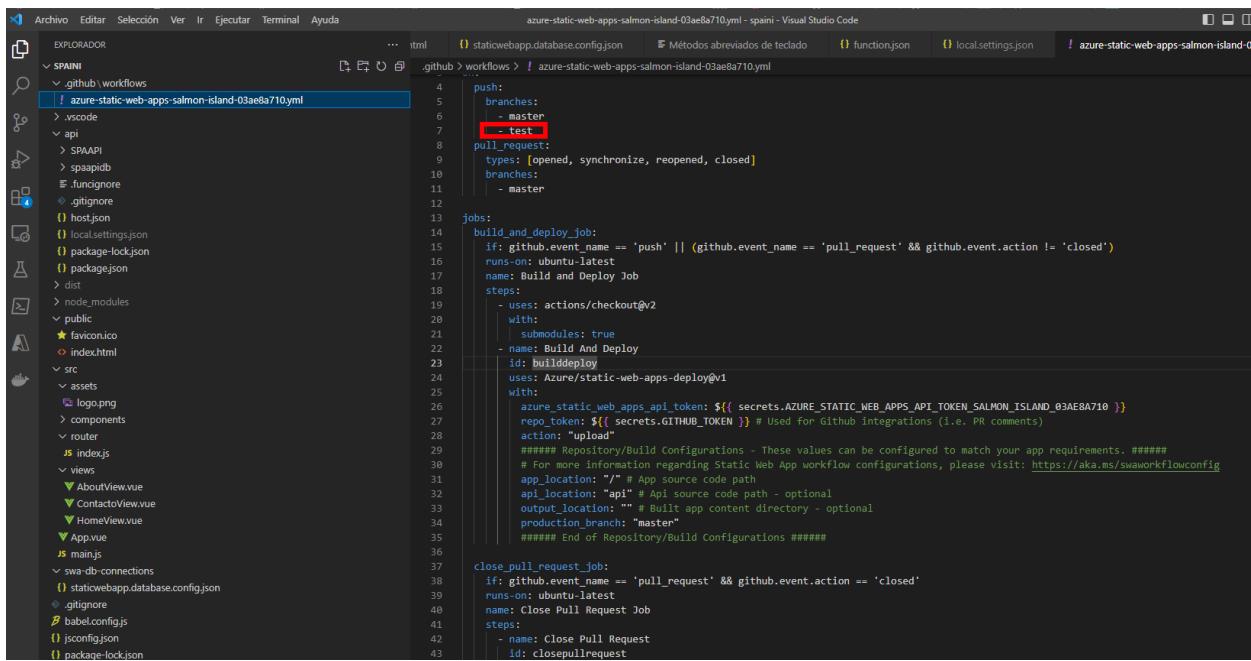
Pero se puede hacer la prueba mediante llamada de Rest:

The screenshot shows a browser window with two tabs: 'spaini' and 'https://salmon-island-03ae8a710.2.azurestaticapps.net//data-api/rest/Persona'. The main content area shows the JSON response from the API call. The response is an array containing two objects, each representing a person with an Id and a Name.

```
{"value": [{"Id": 1, "Nombre": "Leonel"}, {"Id": 2, "Nombre": "Arturo"}]}
```

## Environments

En SWA podemos tener diferentes ambientes de prueba, se pueden configurar por pull request, Branch o named. En este caso agregamos uno tipo Branch de nombre test. Se envía el cambio del archivo de acciones de git a nuestro repositorio. También se tiene que indicar cual es el production\_branch, en este caso master.



```

version: '2.1'

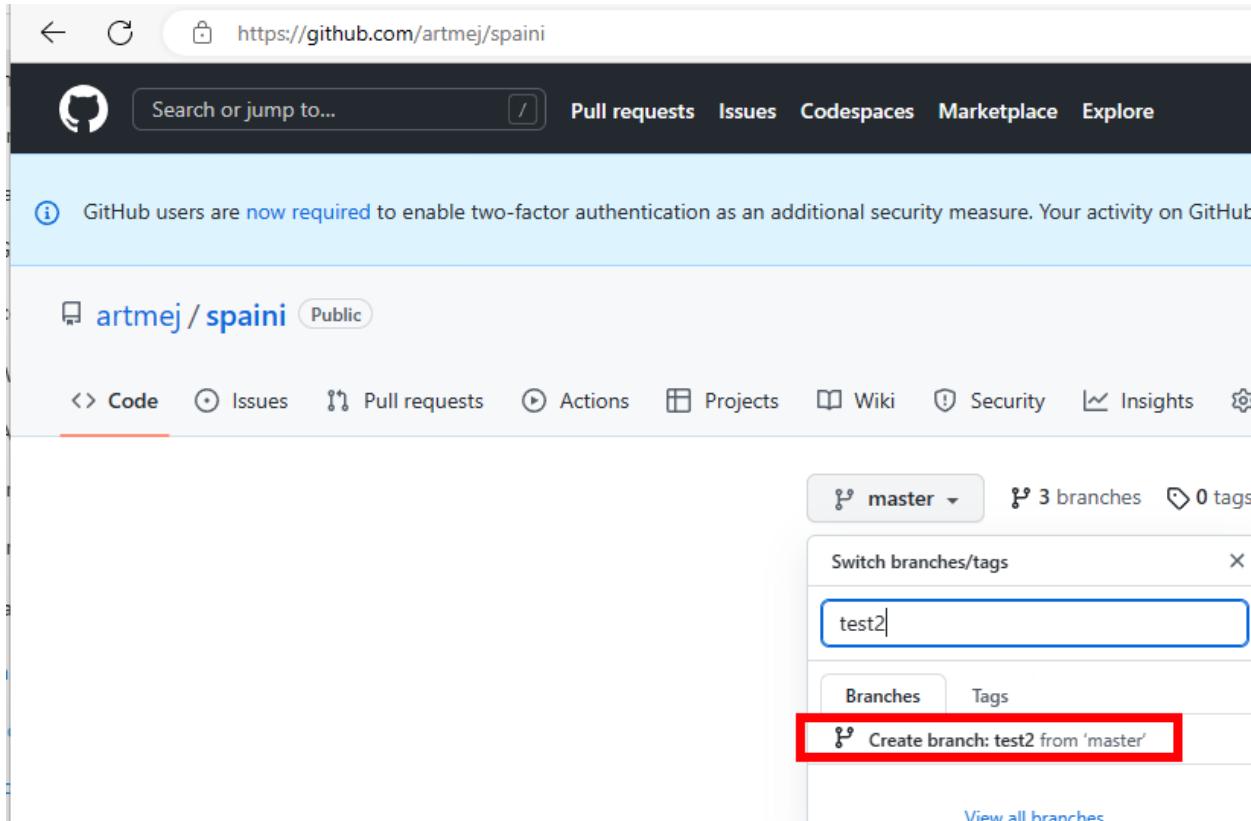
branches:
  - master
  - test

pull_request:
  types: [opened, synchronize, reopened, closed]
  branches:
    - master

jobs:
  build_and_deploy_job:
    if: github.event_name == 'push' || (github.event_name == 'pull_request' & github.event.action != 'closed')
    runs-on: ubuntu-latest
    name: Build and Deploy Job
    steps:
      - uses: actions/checkout@v2
        with:
          submodules: true
      - name: Build And Deploy
        id: buildAndDeploy
        uses: Azure/static-web-apps-deploy@v1
        with:
          azure_static_web_apps_api_token: $(secrets.AZURE_STATIC_WEB_APPS_API_TOKEN_SALMON_ISLAND_03AE8A710)
          repo_token: ${secrets.GITHUB_TOKEN} # Used for Github integrations (i.e. PR comments)
          action: "upload"
          ###### Repository/Build Configurations - These values can be configured to match your app requirements. ######
          # For more information regarding Static Web App workflow configurations, please visit: https://aka.ms/swaWorkflowConfig
          app_location: "/" # App source code path
          api_location: "api" # Api source code path - optional
          output_location: "" # Built app content directory - optional
          production_branch: "master"
          ##### End of Repository/Build Configurations #####
      - name: Close Pull Request Job
        if: github.event_name == 'pull_request' & github.event.action == 'closed'
        runs-on: ubuntu-latest
        name: Close Pull Request
        steps:
          - name: Close Pull Request
            id: closepullrequest

```

Se genera este branch en nuestro repositorio y se tendra un nuevo ambiente.



artmej / spaini Public

Code Issues Pull requests Actions Projects Wiki Security Insights

master 3 branches 0 tags

Switch branches/tags

test2

Create branch: test2 from 'master'

View all branches

Se puede verificar en la opción environments y se puede ver el nuevo branch que se puede validar independiente de nuestro main branch.

The screenshot shows the Azure portal interface for a Static Web App named "SWA-CA-APP-001". On the left, there's a sidebar with various settings like Configuration, Application Insights, and Custom domains. The "Environments" option is selected. The main area displays two environments: "Production" (branch master) and "test" (branch test). Both environments are shown as "Ready".

Name	Branch	Last Update Time	Status	Browse
Production	master	March 24, 2023 at 9:21:04 PM ...	Ready	<a href="#">Browse</a>
test	test	March 24, 2023 at 9:21:37 PM...	Ready	<a href="#">Browse</a>

Si naveguemos a la url vemos que tiene el prefijo de nuestro Branch.

The screenshot shows a browser window with the URL <https://salmon-island-03ae0b.10-test.azurwebsites.net>. The page content includes a large green and blue 'V' logo, the text "Welcome to Your Vue.js App", and a link to "vue-cli documentation". Below the logo, it lists "Installed CLI Plugins" (babel, router, eslint) and "Essential Links" (Core Docs, Forum, Community Chat, Twitter, News).

## Configurar Dominio

Si se cuenta con un dominio se puede registrar. Azure puede llegar a ayudar a adquirir dominios en App services Domain. No todas las cuentas pueden adquirir un Dominio como son el caso del Cloud Academy. La configuración es sencilla. Se entra a la opción de custom domains y se agrega add.

Se pide agregar un registro:

Agregamos el registro:

**Registros DNS**

Los registros de DNS definen cómo se comporta tu dominio, como por ejemplo, mostrar el contenido de tu sitio web y entregar tu correo electrónico.

**Registros nuevos**

Los registros CNAME son un tipo de subdominio, o alias, que señala a otro nombre de dominio.

Tipo*	Nombre*	Valor*	TTL
CNAME	www	salmon-island-03ae8a710.2.azurestaticapps.net	1 Hora

**Añadir más registros** **Guardar** **Cancelar**

Y después de un tiempo se puede regresar a hacer el registro y verificarlo.

Microsoft Azure

Home > RG-CA-SPA-001 > SWA-CA-APP-001 | Custom domains

Add custom domain

Enter domain: www.arturomejia.com.mx

Validate domain ownership:

Type	Host	Value	Status
CNAME	www	salmon-island-03ae8a710.2.azurestaticapps.net	Validated

Se puede validar nuestro nuevo dominio agregado:

Microsoft Azure Search resources, services, and docs (G+)

Home > RG-CA-SPA-001 > SWA-CA-APP-001

**SWA-CA-APP-001 | Custom domains** ...

Static Web App

Search Add Refresh Delete Set default

Overview Access control (IAM) Tags Diagnose and solve problems

Settings Configuration Application Insights

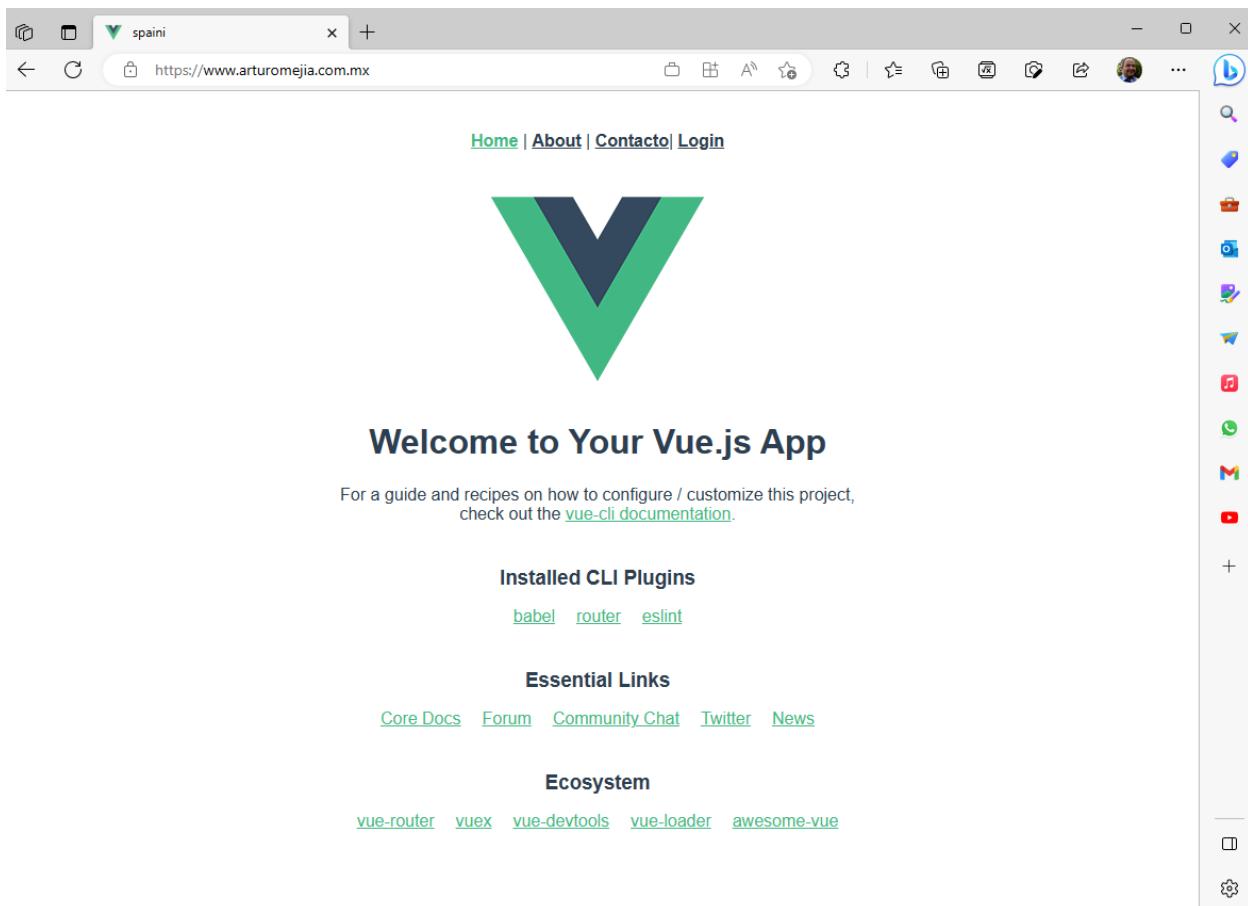
Custom domains APIs Database connection (preview) Environments Role management Identity Enterprise-grade edge Hosting Plan Private endpoints Locks

Map custom domains to this Static Web App. Free SSL/TLS certificates are automatically provided for custom domains. [Learn more](#)

Filter by name...

Name	Status	Type	Documentation
salmon-island-03ae8a7102.azurestaticapps.net	Validated	Auto-generated	
www.arturomejia.com.mx	Validated	Custom domain	

Y validar que se resuelve la SPA con el dominio nuevo y con ssl.



## 5. Progressive Web App (PWA)

¿Qué es Progressive Web App?

El último paso es usar nuestro SPA para iniciararlo como una PWA. El PWA busca los siguientes elementos:



Similar a  
app



Independiente de la  
conectividad



Progresivo



Se requiere contar con elementos mínimos para un PWA.

## Elementos Mínimos



HTTPS\*



Web App Manifest



Service Worker

\*HTTPS not required when using localhost domain

### Actualización de SPA a PWA

Agregamos el módulo de pwa con el siguiente comando en una terminal.

```
vue add pwa
```

Se vera de la siguiente forma:

```
PS C:\repos\spaini> vue add pwa
WARN There are uncommitted changes in the current repository, it's recommended to commit or stash them first.
? Still proceed? Yes

📦  Installing @vue/cli-plugin-pwa...

up to date, audited 1075 packages in 1s

142 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
✓ Successfully installed plugin: @vue/cli-plugin-pwa

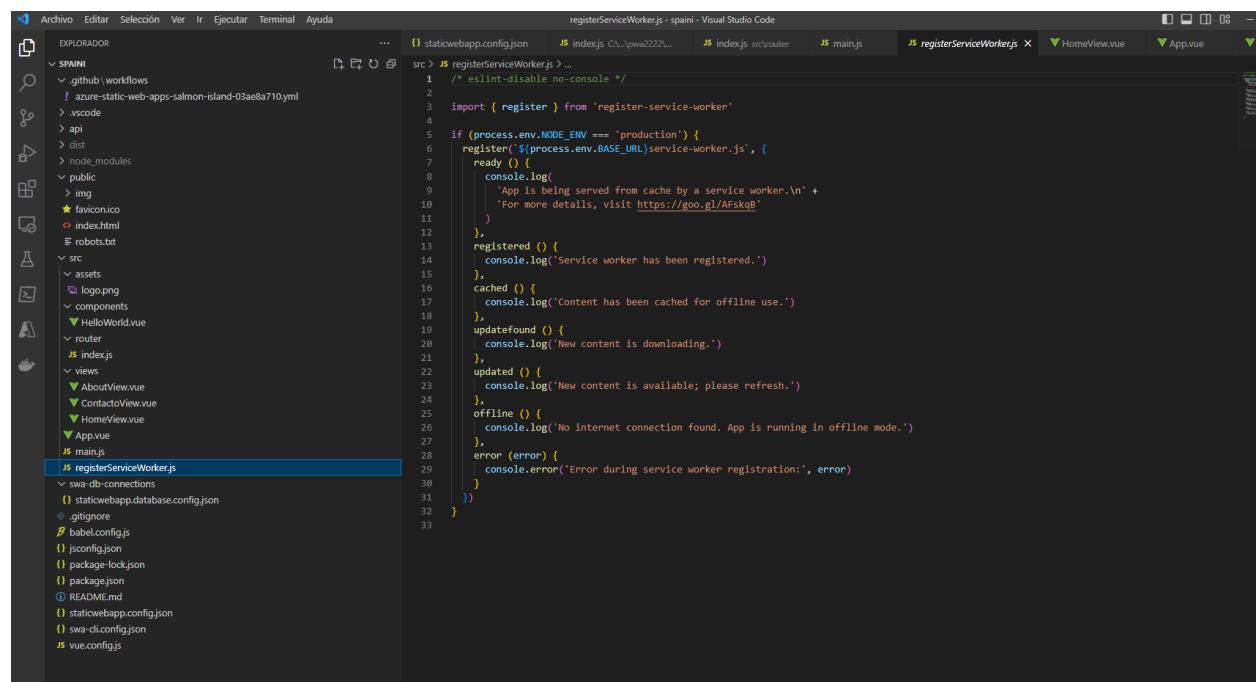
🚀 Invoking generator for @vue/cli-plugin-pwa...
🕒 Running completion hooks...

✓ Successfully invoked generator for plugin: @vue/cli-plugin-pwa
PS C:\repos\spaini>
```

## Revisión de componentes de PWA

### Service Worker

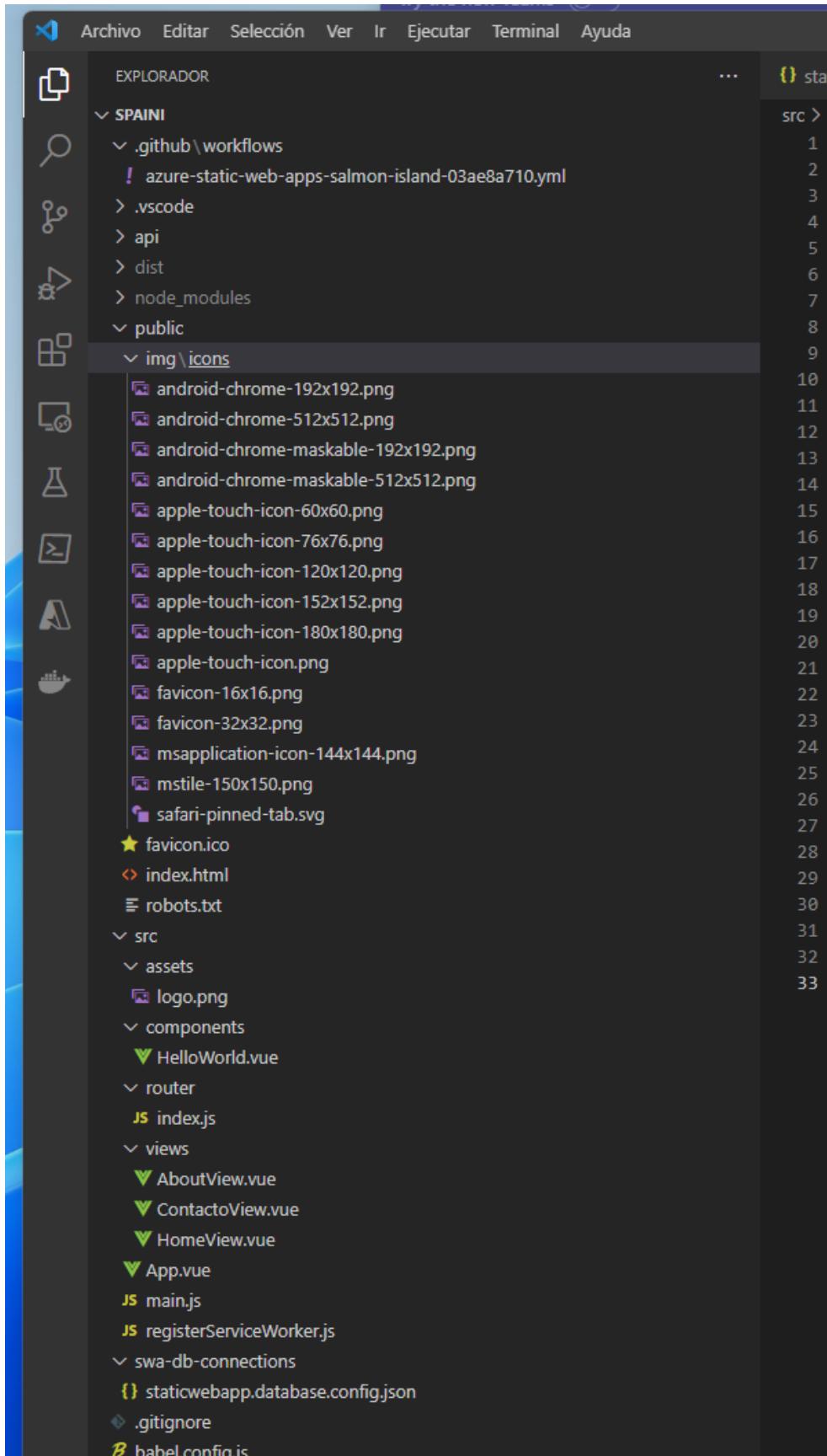
Se agrego un nuevo archivo en src/registerServiceWorker.js donde se puede observar el ciclo de vida del Service worker y realizar acciones.



```
src > JS registerServiceWorker.js - spaini - Visual Studio Code
staticwebapp.config.json JS index.js C:\...\pwa2222\... JS index.js src\router JS main.js JS registerServiceWorker.js ✘ HomeView.vue App.vue
src > JS registerServiceWorker.js ...
1  /* eslint-disable no-console */
2
3  import { register } from 'register-service-worker'
4
5  if (process.env.NODE_ENV === 'production') {
6    register(`${process.env.BASE_URL}service-worker.js`, {
7      ready () {
8        console.log(
9          'App is being served from cache by a service worker.\n' +
10            'For more details, visit https://goo.gl/AFskqB'
11        )
12      },
13      registered () {
14        console.log('Service worker has been registered.')
15      },
16      cached () {
17        console.log('Content has been cached for offline use.')
18      },
19      updatefound () {
20        console.log('New content is downloading.')
21      },
22      updated () {
23        console.log('New content is available; please refresh.')
24      },
25      offline () {
26        console.log('No internet connection found. App is running in offline mode.')
27      },
28      error (error) {
29        console.error(`Error during service worker registration: ${error}`)
30      }
31    })
32  }
33 }
```

## Manifest

El archivo *manifest* se genera hasta que se compila y se encontrará en la carpeta *dist*. Pero los elementos requeridos fueron agregados en la carpeta *public*. Si se quiere cambiar algún elemento del *manifest* file se puede copiar el compilado de *dist* a la carpeta de *public* del código y hacer las adecuaciones.



Para generar el archivo manifest se usará la terminal y se correrá el siguiente comando:

```
npm run build
```

Teniendo una vista como se muestra a continuación.

```
● PS C:\repos\spaini> npm run build
● > spaini@0.1.0 build
  > vue-cli-service build

All browser targets in the browserslist configuration have supported ES module.
Therefore we don't build two separate bundles for differential loading.

: Building for production...

[DONE] Compiled successfully in 20967ms

  File          Size          Gzipped
dist\js\chunk-vendors.c1b7958f.js  99.48 KiB    36.71 KiB
dist\js\app.3371d4df.js            16.91 KiB    10.16 KiB
dist\workbox-db5fc017.js          14.11 KiB    4.93 KiB
dist\service-worker.js             1.24 KiB     0.77 KiB
dist\js\190.cbce0fcd.js           0.65 KiB     0.44 KiB
dist\js\about.33fd4861.js          0.39 KiB     0.29 KiB
dist\css\app.9bdcf330.css          0.42 KiB     0.26 KiB

Images and other types of assets omitted.
Build at: 2023-03-25T13:43:43.918Z - Hash: 51372390c641f253 - Time: 20967ms

[DONE] Build complete. The dist directory is ready to be deployed.
[INFO] Check out deployment instructions at https://cli.vuejs.org/guide/deployment.html

○ PS C:\repos\spaini>
```

Al terminar se puede ver en el archivo *dist/manifest.json*

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** On the left, it shows a tree view of the project structure. The current file selected is `manifest.json`. Other files visible include `staticwebapp.config.json`, `index.js`, `index.js.srcRouter`, `main.js`, `manifest.json`, `HomeView.vue`, `App.vue`, and `ContactoView.vue`.
- Code Editor:** On the right, the code for `manifest.json` is displayed. The content is as follows:

```
manifest.json - spaini - Visual Studio Code
manifest.json
dist > manifest.json ...
  1  {"name": "DEMOSPA", "short_name": "DEMOSPA", "theme_color": "#4DBAB7", "icons": [{"src": "./img/icons/android-chrome-192x192.png", "sizes": "192x192"}, {"src": "./img/icons/android-chrome-512x512.png", "sizes": "512x512"}]}
```

Se copia a la carpeta `public` y se pueden realizar adecuaciones si es requerido.

```
public > () manifest.json > ...
```

```
1 [ "name": "DEMOSPA",
2   "short_name": "DEMOSPA",
3   "theme_color": "#4DBA87",
4   "icons": [
5     {
6       "src": "./img/icons/android-chrome-192x192.png",
7       "sizes": "192x192",
8       "type": "image/png"
9     },
10    {
11      "src": "./img/icons/android-chrome-512x512.png",
12      "sizes": "512x512",
13      "type": "image/png"
14    },
15    {
16      "src": "./img/icons/android-chrome-maskable-192x192.png",
17      "sizes": "192x192",
18      "type": "image/png",
19      "purpose": "maskable"
20    },
21    {
22      "src": "./img/icons/android-chrome-maskable-512x512.png",
23      "sizes": "512x512",
24      "type": "image/png",
25      "purpose": "maskable"
26    }
27  ],
28  "start_url": ".",
29  "display": "standalone",
30  "background_color": "#000000"
31]
32 ]
```

También se requiere generar una excepción de rutas para nuestros nuevos elementos. Esto se hace en dos puntos. El primer punto es en nuestro archivo *staticwebapp.config.json*. Agregando en el *exclude* la nueva ruta *img/icons*.

```

{
  "navigationFallback": {
    "rewrite": "/index.html",
    "exclude": ["*.css,*.scss,*.js,*.png,*.gif,*.ico,*.jpg,*.svg","img/icons/*.png,*.gif,*.ico,*.jpg,*.svg"] 
  },
  "routes": [
    {
      "route": "/login",
      "redirect": "./auth/login/github"
    },
    {
      "route": "/logout",
      "redirect": "./auth/logout"
    },
    {
      "route": "/api/*",
      "allowedRoles": ["authenticated"]
    },
    {
      "route": "/contacto",
      "allowedRoles": ["authenticated"]
    }
  ]
}

```

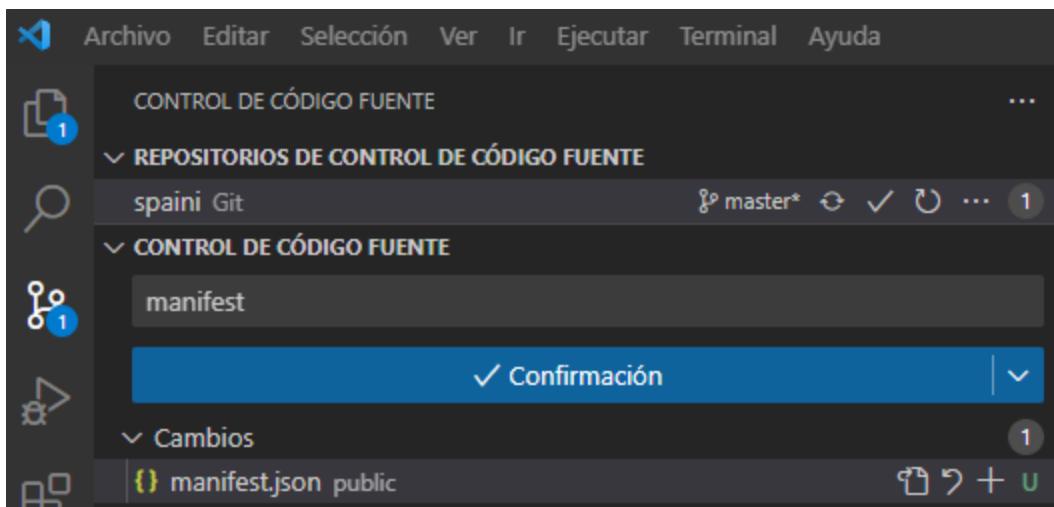
El segundo punto es en nuestro *router* de vue. Se agrega un *path* sin componente de la misma ruta.

```

const routes = [
  {
    path: '/',
    name: 'home',
    component: HomeView
  },
  {
    path: '/about',
    name: 'about',
    // route level code-splitting
    // this generates a separate chunk (about.[hash].js) for this route
    // which is lazy-loaded when the route is visited.
    component: () => import(/* webpackChunkName: "about" */ '../views/AboutView.vue')
  },
  {
    path: '/contacto',
    name: 'contacto',
    component: () => import('../views/ContactoView.vue')
  },
  {
    path: '/graph-api',
  },
  {
    path: '/img/icons',
  }
]

```

Se suben los cambios a nuestro repositorio.

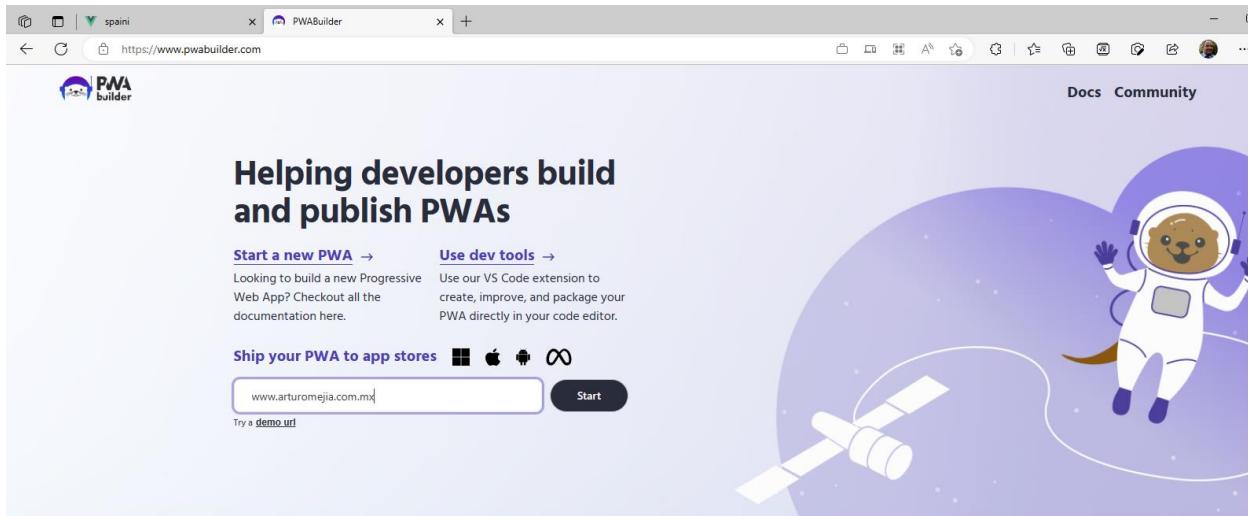


Al desplegarse se pueden ver algunos cambios en nuestra página. Lo primero es que puede instalarse como una aplicación y adicional se puede ver en las herramientas de desarrollo una sección donde se puede examinar tanto el manifest, como el Service worker.

The screenshot shows the Chrome DevTools interface with the 'Application' tab selected. On the left, the 'Manifest' section is highlighted with a red box. It lists the manifest file 'manifest.json' and other service workers. To the right, the 'App Manifest' panel shows the configuration for a PWA named 'DEMOSPA'. It includes fields for Name, Short name, Description, and Computed App Id. The 'Presentation' panel shows settings for Start URL, Theme color (#4DBA87), Background color (#000000), Orientation, and Display. The 'Protocol Handlers' panel allows defining protocol handlers in the manifest. The 'Icons' panel shows the current icon for the app. At the bottom, there are tabs for Console, Issues, and a plus sign.

## Despliegue con Builder

Contando con una PWA se puede usar la herramienta de [www.pwabuilder.com](https://www.pwabuilder.com) en donde usará nuestra *url* para validar nuestra aplicación y desplegarla en las diferentes plataformas. Ingresamos a la *url* y la pondremos en el espacio de texto y daremos *start*.



### Apps packaged

Después nos dará recomendaciones para que nuestra PWA cuente con un mejor comportamiento como aplicación en los dispositivos:

The screenshot shows the PWABuilder report card for the website [www.arturomejia.com.mx](https://www.arturomejia.com.mx). The main header includes the PWABuilder logo, a navigation bar with 'Docs' and 'Community', and a search bar. The report card itself has a light blue background.

- Manifest:** Shows a progress bar at 12/26. It includes a link to "Edit Your Manifest" and "Manifest Documentation".
- Action Items:** A list of 16 items with yellow warning icons:
  - Need at least one icon with purpose set to any
  - Add orientation to your manifest
  - Add screenshots to your manifest
  - Add shortcuts to your manifest
  - Add categories to your manifest
- Service Worker:** Shows a progress bar at 2/4.
- Security:** Shows a progress bar at 3/3. It states: "PWABuilder has done a basic analysis of your HTTPS setup and found no issues! Great job you have a perfect score!"

A "Share score" button is located in the top right corner of the report card area. Below the report card, there's a "Want to chat? Join us on Discord" button.

Al momento tenemos un PWA mínimo y se puede hacer pruebas en un dispositivo Windows mediante descargar el paquete de pruebas:

This screenshot shows the same PWABuilder report card as the previous one, but with a red box highlighting the "Download Test Package" button in the "Package For Stores" section. The rest of the interface is identical to the first screenshot.

Descargamos el archivo.



## Download Test Package

If you want to see what your app would look like in its current state, use the test package buttons below!



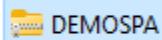
### Windows

[Download Test Package](#)

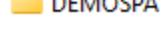
- PWAs can be indistinguishable from native apps on Windows
- PWAs are first class applications
- Collect 100% of revenue generated via third party commerce platforms
- 1B+ store enabled devices

El archivo se des compacta.

▼ Today



DEMOSPA



DEMOSPA

Se ejecuta el comando de PS *install*.

Name	Date modified	Type	Size
<b>▼ Today</b>			
DEMOSPA.classic	25/03/2023 08:03 a. m.	APPXBUNDLE File	972 KB
DEMOSPA	25/03/2023 08:03 a. m.	MSIXBUNDLE File	928 KB
DEMOSPA.sideload	25/03/2023 08:03 a. m.	MSIX File	926 KB
install	25/03/2023 08:03 a. m.	Script de Window...	2 KB
readme	25/03/2023 08:03 a. m.	Microsoft Edge H...	1 KB
utils	25/03/2023 08:03 a. m.	File folder	

```
Windows PowerShell
PS C:\repos\demospa>> dir

Directory: C:\repos\demospa

Mode                LastWriteTime         Length Name
----                ——————          ——— —
d-----        25/03/2023 08:06 a. m.            utils
-a----        25/03/2023 08:03 a. m.      995291 DEMOSPA.classic.appxbundle
-a----        25/03/2023 08:03 a. m.      949361 DEMOSPA.msixbundle
-a----        25/03/2023 08:03 a. m.      947629 DEMOSPA.sideload.msix
-a----        25/03/2023 08:03 a. m.       1549 install.ps1
-a----        25/03/2023 08:03 a. m.       184 readme.html

PS C:\repos\demospa>> ./install

Security warning
Run only scripts that you trust. While scripts from the internet can be useful, this script can potentially harm your
computer. If you trust this script, use the Unblock-File cmdlet to allow the script to run without this warning
message. Do you want to run C:\repos\demospa\install.ps1?
[D] Do not run [R] Run once [S] Suspend [?] Help (default is "D"): R
Installing DEMOSPA...
Installing: C:\repos\demospa\DEMOSPA.sideload.msix
Install complete
Launching DEMOSPA...
PS C:\repos\demospa>>
```

[Home](#) | [About](#) | [Contacto](#) | [Login](#)



## Welcome to Your Vue.js App

For a guide and recipes on how to configure / customize this project,  
check out the [vue-cli documentation](#).

### Installed CLI Plugins

[babel](#) [router](#) [eslint](#)

### Essential Links

[Core Docs](#) [Forum](#) [Community Chat](#) [Twitter](#) [News](#)

### Ecosystem

[vue-router](#) [vuex](#) [vue-devtools](#) [vue-loader](#) [awesome-vue](#)