

## Configuración del entorno frontend paso a paso

### 1. Descargar e Instalar Git Bash

- Descarga e instala Git Bash desde <https://www.git-scm.com/downloads>

### 2. Crear una Carpeta para el Proyecto

### 3. Instalar NVM (Node Version Manager)

- Descarga e instala NVM (Node Version Manager) desde <https://github.com/coreybutler/nvm-windows/releases/download/1.1.12/nvm-setup.exe>

### 4. Instalar Node.js Versión 20

- Ejecuta el siguiente comando en Powershell:

```
nvm install 20
```

- Ítem de lista desordenada Verifica que la instalación fue exitosa verificando su versión con:

```
nvm ls
```

```
* Si estás utilizando otra versión de Node.js, ejecuta:  
nvm use 20
```

### 5. Clonar el Proyecto con Git Clone

- Utiliza el siguiente comando para clonar el proyecto desde GitHub:
- git clone <https://github.com/simongilr/appVersionOne.git>

### 6. Crear Rama de desarrollo

- Para crear su rama de desarrollo ejecute en gitbash:

```
git checkout -b hotfix/nombre-de-la-rama
```

### 7. Instalar Dependencias del Proyecto

- En Git Bash, navega hasta el directorio del proyecto y ejecuta:

```
npm install
```

### 8. Iniciar y Compilar el Proyecto

- Para iniciar el proyecto, ejecuta:

```
npm run start
```

- Para generar el traspilado, utiliza:

```
npm run build
```

- Se puede especificar el ambiente deseado agregando un sufijo, por ejemplo:

npm run start:qa

```
npm run build:qa
```

## 9. Generar Componentes Angular

- Al generar un componente usando `ng g c ruta-del-componente/nombre-del-componente`, ten en cuenta que los componentes standalone no se están utilizando.
- Elimina las líneas de imports y standalone.
- En su lugar, crea un módulo usando:

ng g m /ruta-al-componente-generado-anteriormente

```
* 0 ejecuta el comando siguiente para crear el componente sin standalone:  
ng g c /ruta-del-componente/nombre-del-componente --standalone false
```

## 10. Configurar Módulos de Angular

1. Dentro del módulo creado, agrega el componente en "imports" y "exports" para su uso dentro del proyecto.
2. Para hacer uso del componente se debe de exportar el módulo

11. Componentes de PrimeNG - Antes de hacer uso de los componentes de PrimeNG se debe de seguir sus pasos de instalación los cuales puedes encontrar en: <https://primeng.org/installation> - Una vez instala PrimeNG se puede hacer uso de los componentes de la sección "Componentes" de su página web, sección FEATURES. - Copie y pegue el componente que desea usar y acomode a sus necesidades - En la sección de API puede encontrar todas las propiedades del componente que necesite

Ejemplo: se desea hacer uso de un calendario de PrimeNG; en este caso se consigue el código:

```
<ng-container *ngIf="field.type === 'calendar'">
```

```
    <div class="flex-auto">  
      <label for="buttondisplay" class="font-bold block mb-2"> Button  
    </label>  
      <p-calendar [(ngModel)]="user[field.model]" dateFormat="yy-mm-dd"  
[utc]="true" firstDayOfWeek="0"  
      [(ngModel)]="date1"  
      [showIcon]="true"  
      inputId="buttondisplay"  
      [showOnFocus]="false" />  
    </div>
```

```
</ng-container>
```

Este segmento hace parte de un formulario, pero se debe enlazar a la API formterceros.ts para enlazar el componente con el backend

```
{ id: 'fechadecreacion', name: 'fechadecreacion', label: 'Fecha de  
creación', type: 'calendar', placeholder: 'fechadecreacion', model:
```

```
'fechadecreacion', maxlength: 80, required: false },
```

De esta forma “fechadecreacion” es tipo “calendar” lo que lo enlaza al componente

**12. Configurar hook git en el proyecto** Es necesario que el hook locales tengan permiso en la raiz del proyecto

- `chmod +x /hooks/pre-push`

Es necesario que git apunte al directorio hock

- `git config core.hooksPath hooks`

[←Regresar](#)

From:  
<http://wiki.adacsc.co/> - Wiki

Permanent link:  
<http://wiki.adacsc.co/doku.php?id=ada:howto:sicoferp:factory:new-migracion-sicoferp:front:primeros&rev=1719435948>

Last update: **2024/06/26 21:05**

