

MANUAL DE INSTALACIÓN Y MANEJO PROYECTO BUBBLEBOX

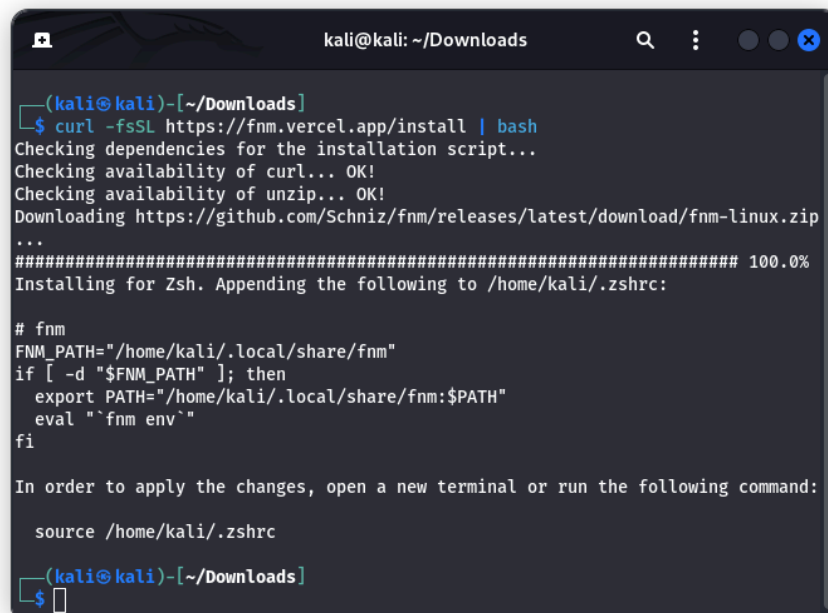
#Prerrequisitos:

- NodeJS, npm
- Git

INSTALACIÓN DE NODEJS CON FNM (FAST NODE MANAGER)

Comando #1: `sudo apt install curl`

Comando #2: `curl -fsSL https://fnm.vercel.app/install | bash`



```
(kali@kali)-[~/Downloads]
$ curl -fsSL https://fnm.vercel.app/install | bash
Checking dependencies for the installation script...
Checking availability of curl... OK!
Checking availability of unzip... OK!
Downloading https://github.com/Schniz/fnm/releases/latest/download/fnm-linux.zip
...
##### 100.0%
Installing for Zsh. Appending the following to /home/kali/.zshrc:

# fnm
FNM_PATH="/home/kali/.local/share/fnm"
if [ -d "$FNM_PATH" ]; then
    export PATH="/home/kali/.local/share/fnm:$PATH"
    eval "`fnm env`"
fi

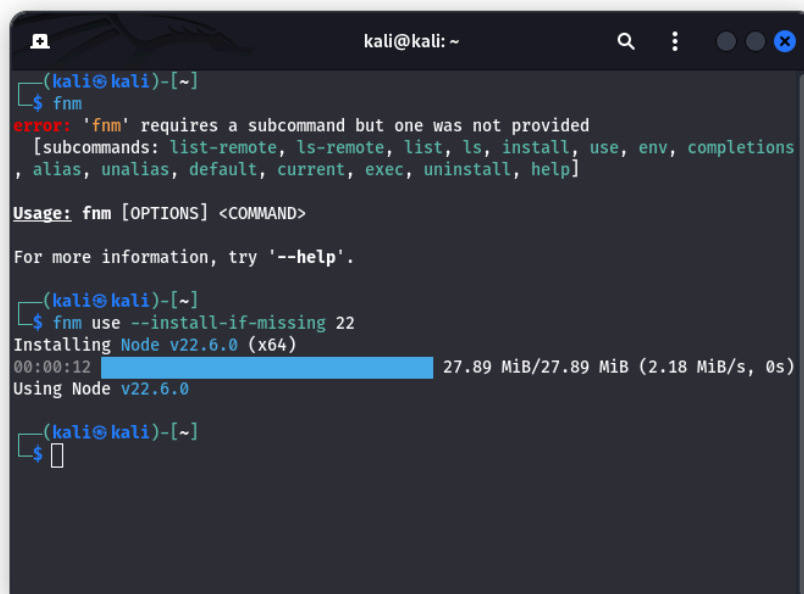
In order to apply the changes, open a new terminal or run the following command:

source /home/kali/.zshrc

(kali@kali)-[~/Downloads]
$
```

Comando #3: `source ~/.bashrc` (después de este comando cerrar la terminal y abrir otra)

Comando #4: `fnm use --install-if-missing 22`



```
(kali@kali)-[~]
$ fnm
error: 'fnm' requires a subcommand but one was not provided
[subcommands: list-remote, ls-remote, list, ls, install, use, env, completions,
alias, unalias, default, current, exec, uninstall, help]

Usage: fnm [OPTIONS] <COMMAND>

For more information, try '--help'.

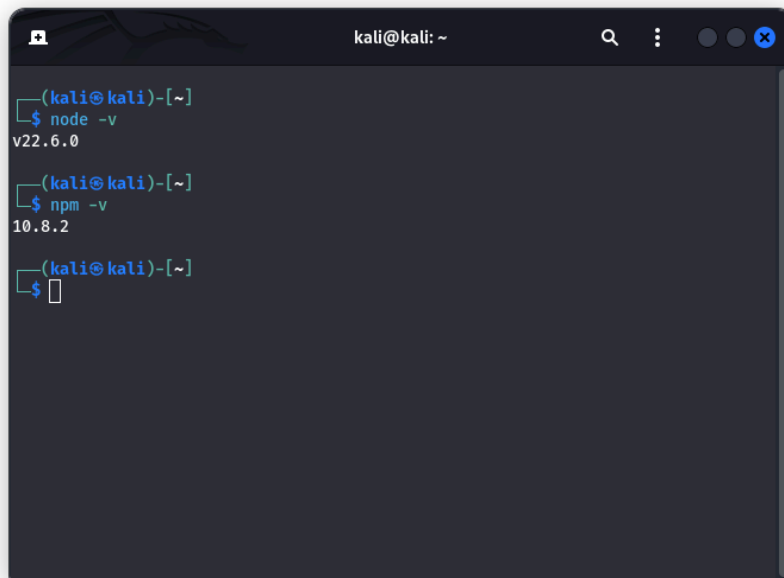
(kali@kali)-[~]
$ fnm use --install-if-missing 22
Installing Node v22.6.0 (x64)
00:00:12 27.89 MiB/27.89 MiB (2.18 MiB/s, 0s)
Using Node v22.6.0

(kali@kali)-[~]
$
```

Para verificar la instalación de NodeJS ejecutar los siguientes comandos, estos nos devuelven la versión instalada:

Comando #1: `node -v`

Comando #2: `npm -v`



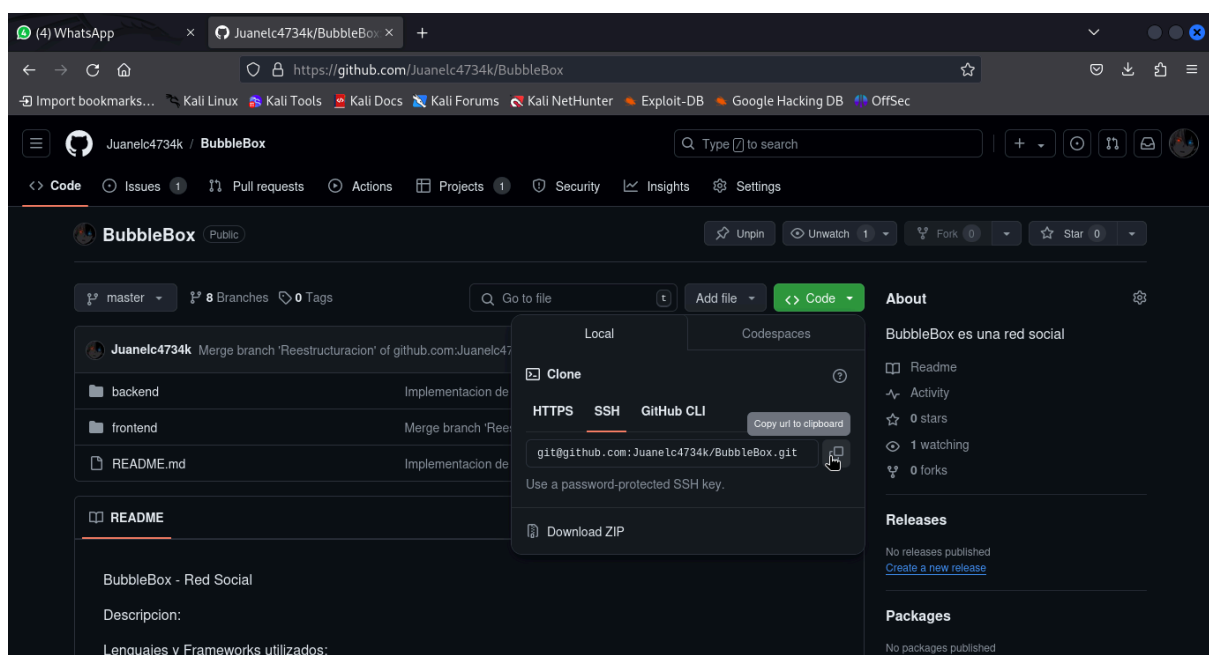
```
(kali㉿kali)-[~]  
$ node -v  
v22.6.0  
  
(kali㉿kali)-[~]  
$ npm -v  
10.8.2  
  
(kali㉿kali)-[~]  
$
```

INSTALACIÓN DEL PROYECTO DE GITHUB

Si NO se tiene instalado git ejecutar lo siguiente:

Comando: `sudo apt install git`

Si ya se tiene instalado ejecutar los siguientes comandos (esto va desde la descarga del repositorio hasta la configuración para ejecutar el proyecto):



Antes de este comando desplazarse a una carpeta como por ejemplo documentos

Comando #1: `cd Documents`

Comando #2: `git clone git@github.com:Juanelc4734k/BubbleBox.git`

Comando #3: `cd BubbleBox`

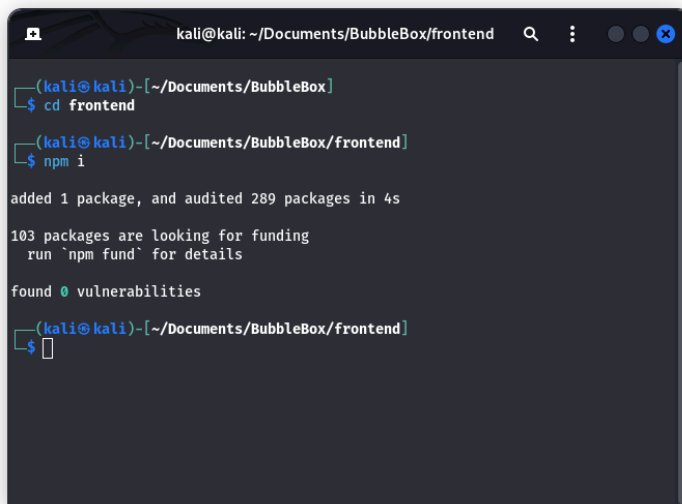
Ya estaría instalado el repositorio los comandos siguientes es para instalar las dependencias tanto del backend como del frontend (se debe saber que para el frontend solo es necesario el backend para ingresar al Home y páginas)

Nota: No tocar código del backend

FRONTEND:

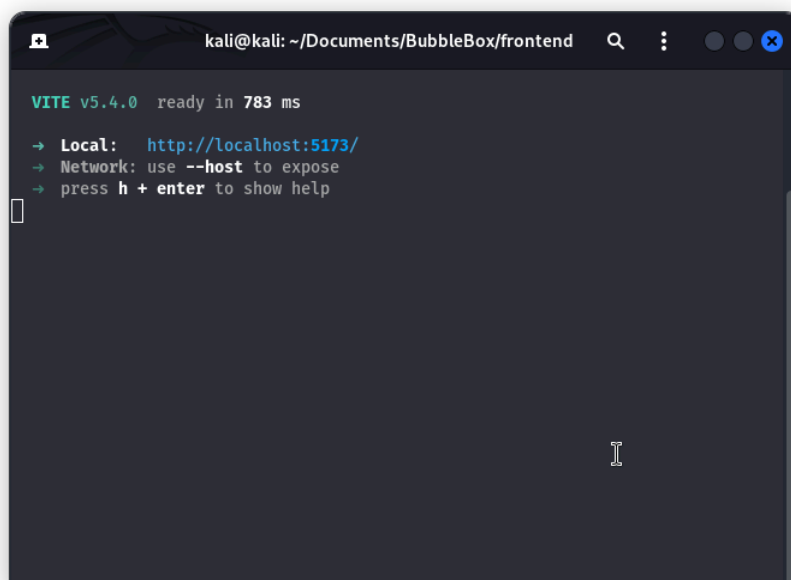
Comando #1: `cd frontend`

Comando #2: `npm i` (en caso de algún error en este comando ejecutar (`sudo npm i`))



```
kali@kali: ~/Documents/BubbleBox/frontend
(kali@kali)-[~/Documents/BubbleBox]
$ cd frontend
(kali@kali)-[~/Documents/BubbleBox/frontend]
$ npm i
added 1 package, and audited 289 packages in 4s
103 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
(kali@kali)-[~/Documents/BubbleBox/frontend]
$
```

Comando #3: `npm run dev` (Si hay algún error ejecutar (`sudo npm run dev`))



```
kali@kali: ~/Documents/BubbleBox/frontend
VITE v5.4.0 ready in 783 ms
→ Local: http://localhost:5173/
→ Network: use --host to expose
→ press h + enter to show help

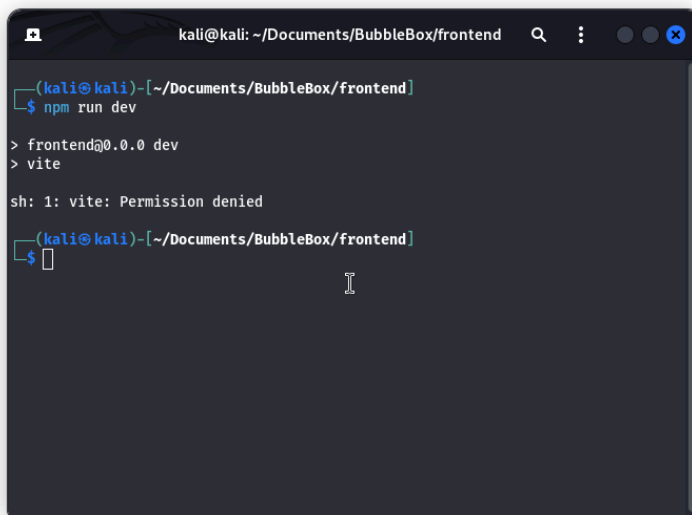
```

Errores comunes:

Error en el comando `npm run dev`: Esto se debe a que no se tiene instalado vite como dependencia global, para solucionar ejecutar:

Comando: `npm i vite -g`

Ya con esto ejecutan de nuevo (`npm run dev`) y debería funcionar



```
kali@kali: ~/Documents/BubbleBox/frontend
(kali@kali)~[~/Documents/BubbleBox/frontend]
$ npm run dev
> frontend@0.0.0 dev
> vite
sh: 1: vite: Permission denied
(kali@kali)~[~/Documents/BubbleBox/frontend]
$
```

BACKEND:

Para configurar el backend deben volver a la carpeta(directorio) principal es decir a BubbleBox deben salir de la carpeta frontend, después de esto ejecutar los siguientes comandos:

Comando #1: `cd backend`

Configurar el gateway:

Comando #1: `cd gateway`

Comando #2: `npm i`

```
kali@kali: ~/Documents/BubbleBox/backend/gateway
(kali@kali)~/Documents/BubbleBox/backend$ cd gateway
(kali@kali)~/Documents/BubbleBox/backend/gateway$ npm i
up to date, audited 106 packages in 2s
18 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
(kali@kali)~/Documents/BubbleBox/backend/gateway$
```

Comando: cd ..

Configurar servicios:

Comando #1: cd services

Comando #2: npm i

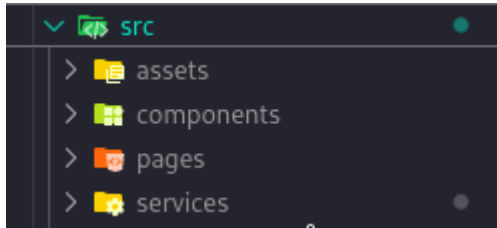
```
(kali@kali)~/Documents/BubbleBox/backend/gateway$ cd ..
(kali@kali)~/Documents/BubbleBox/backend$ cd services
(kali@kali)~/Documents/BubbleBox/backend/services$ npm i
up to date, audited 123 packages in 2s
17 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
(kali@kali)~/Documents/BubbleBox/backend/services$
```

Con esto ya estaría configurado para iniciar el backend se debe hacer desde la carpeta scripts y ejecutar el archivo startAllServices.js esto iniciara todos los servicios (se puede iniciar con la extension code runner o desde la terminal con el comando node startAllServices.js)

También se debe iniciar el gateway que seria el archivo index.js que esta en la carpeta gateway/src/tests (ejecutar con node index.js)

GUIA PARA LAS DE FRONTEND XD

Dentro de la carpeta frontend tenemos varias carpetas en la que se deben crear los componentes es la carpeta src dentro de esta carpeta estan distintas carpetas las cuales son:



assets: Esta carpeta es para imágenes, y los estilos css

components: Aquí se almacenan organizadamente los componentes que son reutilizables en la interfaz del usuario;

pages: Contiene los componentes que representan las diferentes páginas de la aplicación

services: Contiene los endpoints a los que la aplicación va a realizar las solicitudes

En la misma carpeta de src existen dos archivos:

App.jsx: Aquí se define la estructura de la aplicación, incluyendo las rutas y la integración de cada página;

main.jsx: Renderiza a la App.jsx sin este archivo no se podría renderizar o mostrar las páginas definidas en el archivo App.jsx

Líneas de código para importar el css

```
import 'nombre_del_archivo';
```

COMANDOS GIT

Configuración de nombre y email:

```
git config --global user.name "Tu Nombre"
git config --global user.email "tuemail@example.com"
```

Subir cambios a un repo mediante otra rama:

Crear la rama de tu cambio:

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

Realizar Cambios y Confirmar (Commit)

```
git add .  
git commit -m "Descripción de los cambios realizados"
```

Subir la Rama al Repositorio Remoto:

```
git push origin nombre-de-la-rama
```

Actualizar tu Rama Local

```
git checkout master  
git pull origin master
```