

Manual Levantamiento de proyecto

Frontend (React) ISM



Contenido

Manual Levantamiento de proyecto	1
Frontend (React) ISM	1
Descarga del proyecto	2
Selección de versión y ramas.....	3
Ejecución del proyecto	4
Variables de entorno	5
Certificados SSL	6
Scripts de compilación.....	7
Ejecución en ambiente de desarrollo	8
Ejecución en ambiente de producción	9
Anexos	11
Instalación de Chocolatey.....	11
Instalación de Git.....	11
Instalación de NODEJS.....	13
Instalación de YARN.....	14
Instalación de MkCert.....	14

Para el levantamiento correcto del aplicativo es necesario contar con cierto software y dependencias. Todos los requerimientos de software son descritos en la sección de anexos anclada en la sección final de este documento.

El proyecto frontend de ISM se encuentra alojado en los repositorios de Gitlab, a continuación, se describen los pasos necesarios para poder descargar y compilar los binarios para los procesos de producción y desarrollo.

Esta guía se desarrolló utilizando el repositorio actualmente activo, la ubicación de este puede ser cambiada con el transcurso del tiempo. el mismo se encuentra en el siguiente enlace. <https://gitlab.com/darwinvinicio14/proyecto-ism>.

Descarga del proyecto

- Dirigirse a la página donde se encuentra alojado el repositorio del proyecto en Gitlab (Solicitar credenciales de acceso al repositorio).

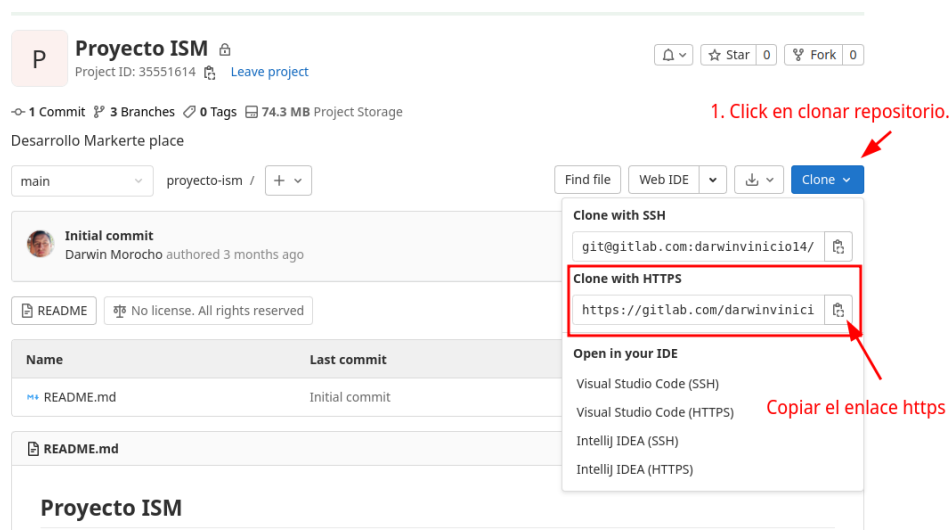


Fig. 1 Repositorio del Proyecto

Es necesario seleccionar la opción de clonar repositorio y posteriormente escoger la opción con el enlace https como se muestra en la figura 1.

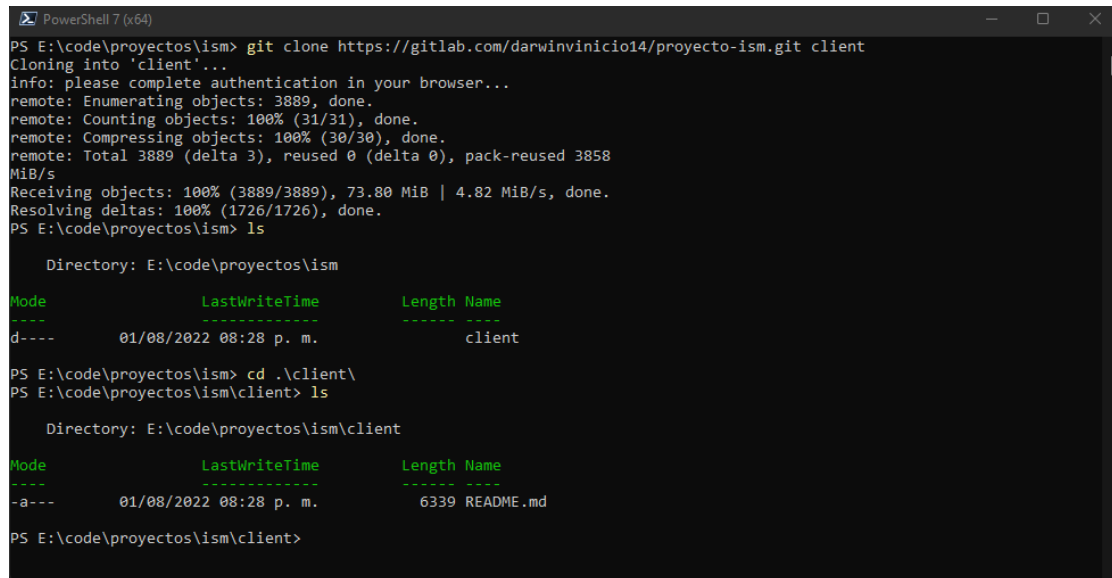
- En tu sistema operativo favorito abre una terminal (powershell, cmd o bash si usas un sistema operativo windows), y descarga el proyecto agregando la siguiente línea de comando.

```
git clone <url_repositorio> <nombre_carpeta_descarga>
```

Selección de versión y ramas

Una vez se halle descargado el código fuente del proyecto, ingresar a la carpeta seleccionada, solo se visualizará el archivo correspondiente al README.

El proyecto solo descargará los archivos fuente que se encuentren en la rama principal (main), para el proyecto frontend de ISM, se establecieron 2 ramas para desarrollo, dejando la rama master como la rama de ejecución principal y los tags de versionamiento como se muestra en la figura 2.



```
PS E:\code\proyectos\ism> git clone https://gitlab.com/darwinvinicio14/proyecto-ism.git client
Cloning into 'client'...
info: please complete authentication in your browser...
remote: Enumerating objects: 3889, done.
remote: Counting objects: 100% (31/31), done.
remote: Compressing objects: 100% (30/30), done.
remote: Total 3889 (delta 3), reused 0 (delta 0), pack-reused 3858
MiB/s
Receiving objects: 100% (3889/3889), 73.80 MiB | 4.82 MiB/s, done.
Resolving deltas: 100% (1726/1726), done.
PS E:\code\proyectos\ism> ls

Directory: E:\code\proyectos\ism

Mode                LastWriteTime         Length Name
----                -
d-----           01/08/2022 08:28 p. m.         client

PS E:\code\proyectos\ism> cd .\client\
PS E:\code\proyectos\ism\client> ls

Directory: E:\code\proyectos\ism\client

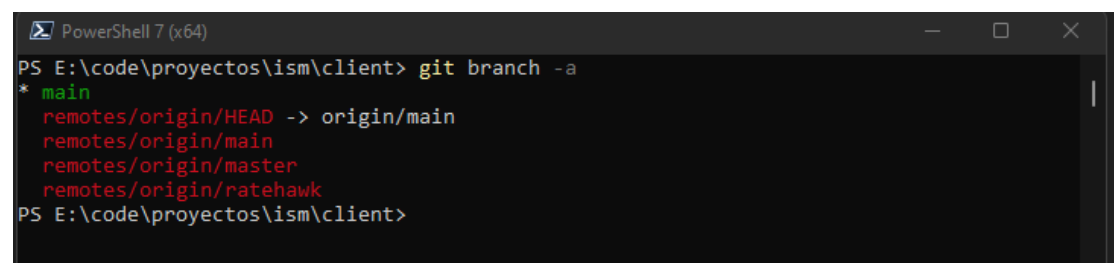
Mode                LastWriteTime         Length Name
----                -
-a---           01/08/2022 08:28 p. m.         6339 README.md

PS E:\code\proyectos\ism\client>
```

Fig. 2 Descarga de la rama main

Ingresando el siguiente comando en la consola de tu sistema operativo se puede listar las ramas activas en el repositorio para su posterior descarga, el puntero HEAD apuntará a la rama actual como se muestra en la figura 3.

git branch -a



```
PS E:\code\proyectos\ism\client> git branch -a
* main
remotes/origin/HEAD -> origin/main
remotes/origin/main
remotes/origin/master
remotes/origin/ratehawk
PS E:\code\proyectos\ism\client>
```

Fig. 3 Listado de ramas remotas

Seleccionar la descarga de la rama **master** con el siguiente comando.

git checkout -b master origin/master

```
PowerShell 7 (x64)
PS E:\code\proyectos\ism\client> git checkout -b master origin/master
Switched to a new branch 'master'
branch 'master' set up to track 'origin/master'.
```

Fig. 4 Descarga de rama master

Si se inspecciona una vez mas los archivos de la carpeta del proyecto, esta vez se puede visualizar la estructura de los archivos fuente para su construcción a binarios de producción o la continuación del desarrollo, como se muestra en la figura 5.

```
PowerShell 7 (x64)
PS E:\code\proyectos\ism\client> ls

Directory: E:\code\proyectos\ism\client

Mode                LastWriteTime         Length Name
----                -
d-----          01/08/2022 08:39 p. m.         .vscode
d-----          01/08/2022 08:39 p. m.         public
d-----          01/08/2022 08:39 p. m.         src
d-----          01/08/2022 08:39 p. m.         webpack
-a----          01/08/2022 08:39 p. m.           389 .babelrc
-a----          01/08/2022 08:39 p. m.        1029 .eslintrc.js
-a----          01/08/2022 08:39 p. m.           84 .gitignore
-a----          01/08/2022 08:39 p. m.        155 .prettierrc.js
-a----          01/08/2022 08:39 p. m.       3273 package.json
-a----          01/08/2022 08:39 p. m.        196 postcss.config.js
-a----          01/08/2022 08:39 p. m.        774 tailwind.config.js
-a----          01/08/2022 08:39 p. m.       7168 tsconfig.json
-a----          01/08/2022 08:39 p. m.     461571 yarn.lock

PS E:\code\proyectos\ism\client>
```

Fig. 5 Estructura y archivos Fuente del proyecto

Ejecución del proyecto

Abrir el proyecto en tu editor de código preferido, en este manual se hace uso de Visual Studio Code.

El proyecto fue construido utilizando webpack como empaquetador de módulos, TypeScript para establecer un tipado estricto, Babel para la compilación del código fuente y TailwindCSS conjuntamente con PostCSS para el tratamiento de los estilos.

Antes de iniciar con la ejecución del proyecto es necesario realizar la instalación de todas las bibliotecas y archivos de dependencia como se muestra en la figura 6. Estas dependencias deben irse actualizando con las versiones más recientes con el transcurso del tiempo, de esta manera se puede evitar vulnerabilidades en el software.

Si posees el gestor de paquetes de npm. Ingresa el siguiente comando.

npm **install**

Si en cambio prefieres el gestor de paquetes de yarn (revisar anexos para su instalación).

yarn **install**

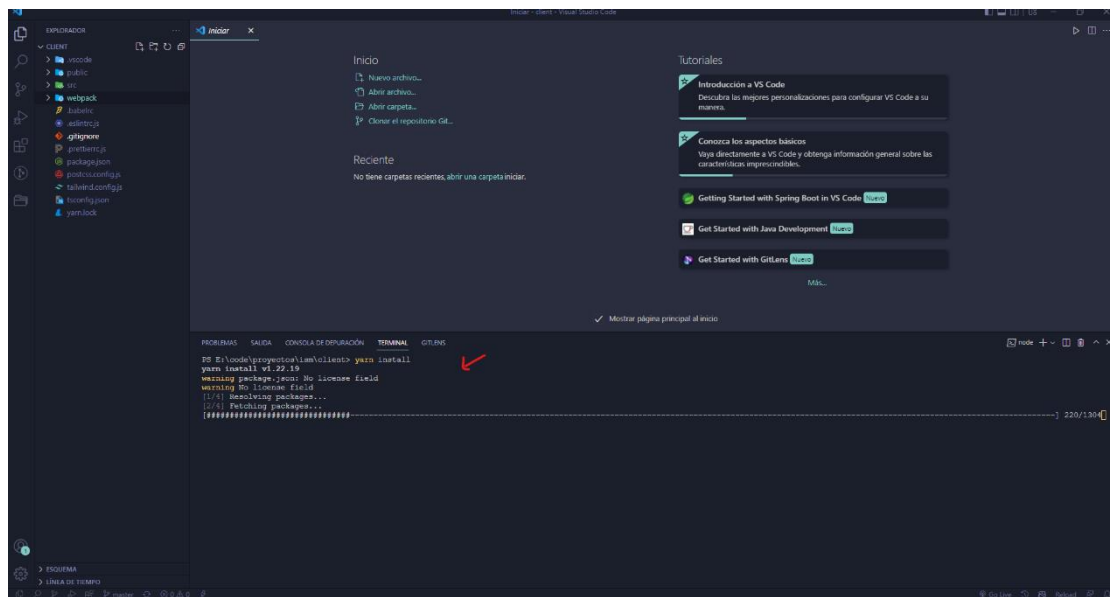


Fig. 6 Instalación de dependencias

Variables de entorno

La ejecución del proyecto necesita de la parametrización de las variables de entorno, estas variables se deben configurar en archivos independientes de acuerdo con el ambiente de ejecución.

En la raíz del proyecto crear un archivo de variables con el **nombre.env.<ambiente>** como se muestra en la figura 7.

El ambiente configurado únicamente puede ser development o production.

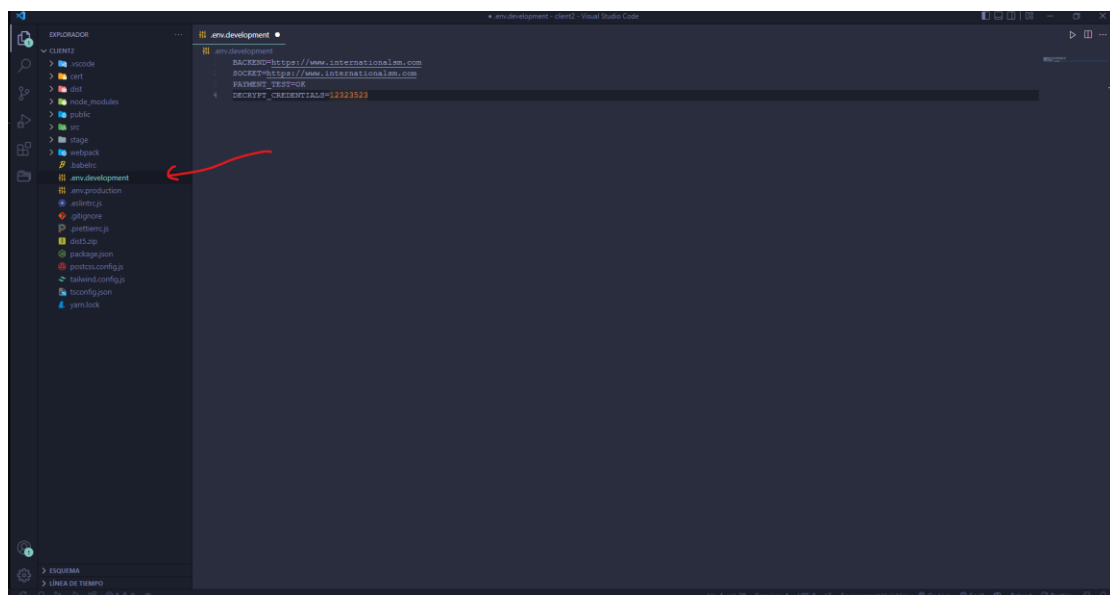


Fig. 7 Variables de entorno

En la tabla 1 se detalla la clave y valor de cada variable de entorno.

CLAVE	VALOR
BACKEND	URL remota del servidor donde se aloja el proyecto de backend
SOCKET	URL remota del servidor donde se aloja el proyecto de Chat.
PAYMENT	Si su valor es OK, permitirá hacer pagos con las credenciales de prueba para Datafast, se recomienda que su valor sea NONE para los archivos de entorno de producción.
DECRYPT_CREDENTIALS	Credenciales del token especificadas en el proyecto backend del Chat.

Tabla 1 Descripción de variables de entorno

Certificados SSL

El proyecto se construyó utilizando protocolos de transferencia segura (https), si no se habilita los protocolos https, es posible que puedas tener problemas con las compras realizadas en la aplicación a través de Datafast, del mismo modo obtendrás problemas de conexión con los sockets del chat y las consultas al servidor de backend si este se encuentra alojado en un dominio con encriptación https.

Para alojar los certificados https para el dominio de localhost, crea una carpeta llamada cert en la raíz del proyecto.

Si está utilizando el software de mkcert para la generación de certificados entonces abra una terminal de comandos, ubíquese en la carpeta cert y ejecute la siguiente instrucción.

```
mkcert -install  
mkcert 0.0.0.0 localhost 127.0.0.1 ::1
```

esto generará dos archivos con la extensión **.pem**

renombré los archivos como se muestra a continuación.

```
0.0.0.0+3-key.pem -> key.pem  
0.0.0.0+3.pem -> cert.pem
```

Entonces debe tener 2 archivos, uno llamado key.pem y el otro llamado cert.pem

Nota. Los certificados emitidos por esta entidad son válidos única y exclusivamente para la máquina host que emitió dichos certificados y se utilizan únicamente para entornos de desarrollo. No es necesario instalar certificados de protocolo de transferencia seguro si únicamente se va a compilar el código fuente para procesos de producción.

Si no desea utilizar los certificados SSL en el desarrollo del frontend siga los siguientes pasos.

- Ubicarse en la raíz del proyecto
- Ingresar en la carpeta webpack
- Abrir el archivo webpack.dev.js (archivo de configuración para desarrollo)
- Eliminar el objeto https de la configuración devServer.

Utilizar como guía la figura 8 para la eliminación de configuración de protocolos https.

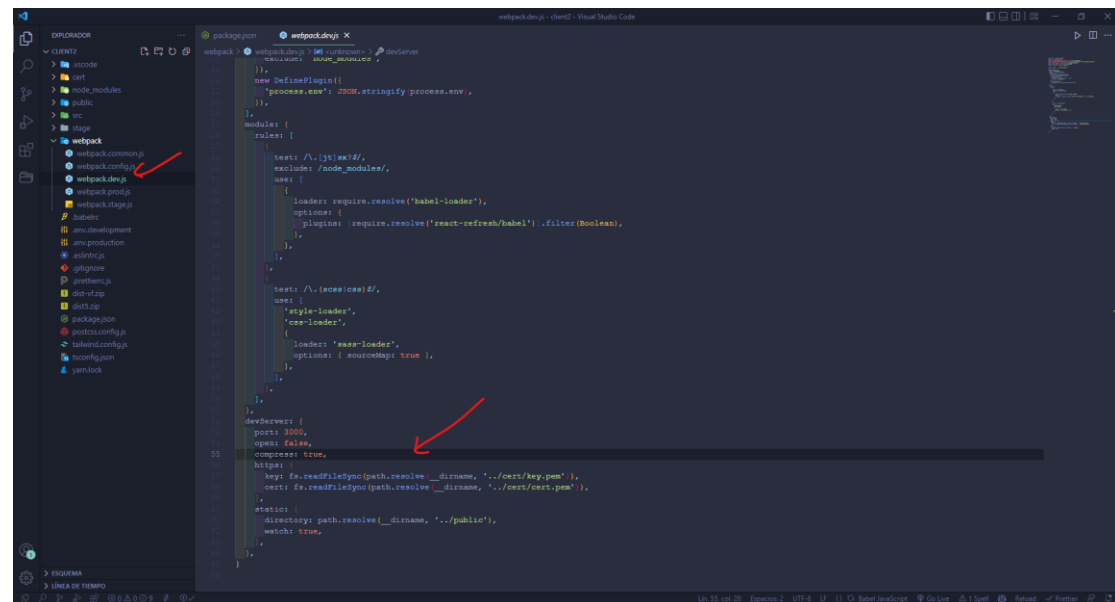


Fig. 8 Archivo de configuración webpack para desarrollo

Scripts de compilación

Una vez se hallen configuradas las variables de entorno, entonces se puede ejecutar distintas fases del proyecto, cada fase se halla configurado en un script listado en el archivo package.json ubicado en la raíz del proyecto como se muestra en la figura 9.

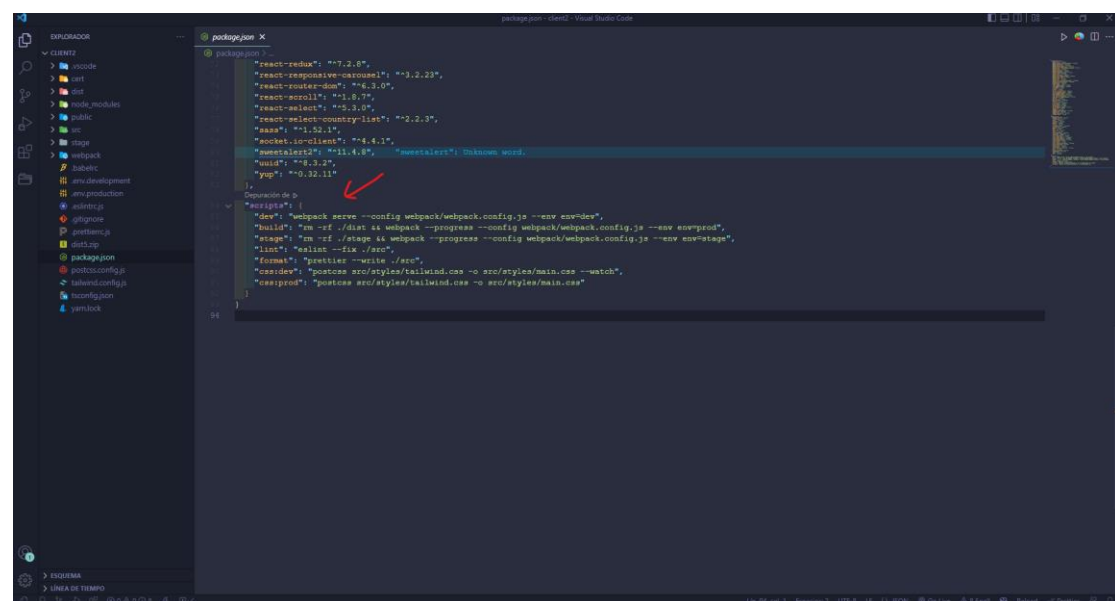


Fig. 9 Scripts de inicialización del Proyecto

En la tabla 2 se detalla el funcionamiento de cada script listado en el archivo package.json

SCRIPT	DESCRIPCIÓN
Dev	Inicia la ejecución del proyecto en modo desarrollo
Build	Realiza la compilación de los binarios para producción.
Stage	Ejecuta pruebas de rendimiento, peso de librerías y un análisis completo del bundle de producción. El informe detallado puede revisarse accediendo a http://127.0.0.1:8888 una vez se termine la ejecución del script.
Lint	Hace un barrido en todos los archivos incluidos en el source del aplicativo (src), analiza el código fuente, revisando la escritura correcta del mismo en base a las reglas parametrizadas en el archivo eslint.js ubicado en la raíz del proyecto.
Format	Formatea todos los archivos incluidos en el source(código fuente) siguiendo las reglas especificadas en el archivo .prettierrc.js ubicado en la raíz del proyecto.
Css:dev	Optimiza y vigila los archivos de css para el entorno de desarrollo a través del compilador de PostCSS.
Css:prod	Optimiza y minifica los archivos fuente del código css para el entorno de producción a través del compilador PostCSS.

Tabla 2 Descripción de Scripts de inicialización

Si se ha decidido utilizar el gestor de paquetes NPM en tu proyecto, la ejecución de cada script se realizará con el siguiente comando.

```
npm run <script>
```

Si se ha optado por la utilización de yarn como gestor de paquetes, la ejecución de cada script se realizará con el siguiente comando.

```
yarn <script>
```

Ejecución en ambiente de desarrollo

Una vez se tenga todo listo y configurado, es momento de iniciar el proyecto en un ambiente de desarrollo, para realizar dicha acción se debe abrir una terminal de comandos y ubicarse en la raíz del proyecto, luego, ejecutar el script dev como se indica a continuación.

```
yarn dev
```

La ejecución del script cargará todos los módulos necesarios para iniciar con el desarrollo del software como se muestra en la figura 10, por defecto el software estará escuchando el puerto 3000 si accedes a través del dominio local, dicho puerto puede ser parametrizado en el archivo de configuración de webpack de desarrollo, ver figura 8.

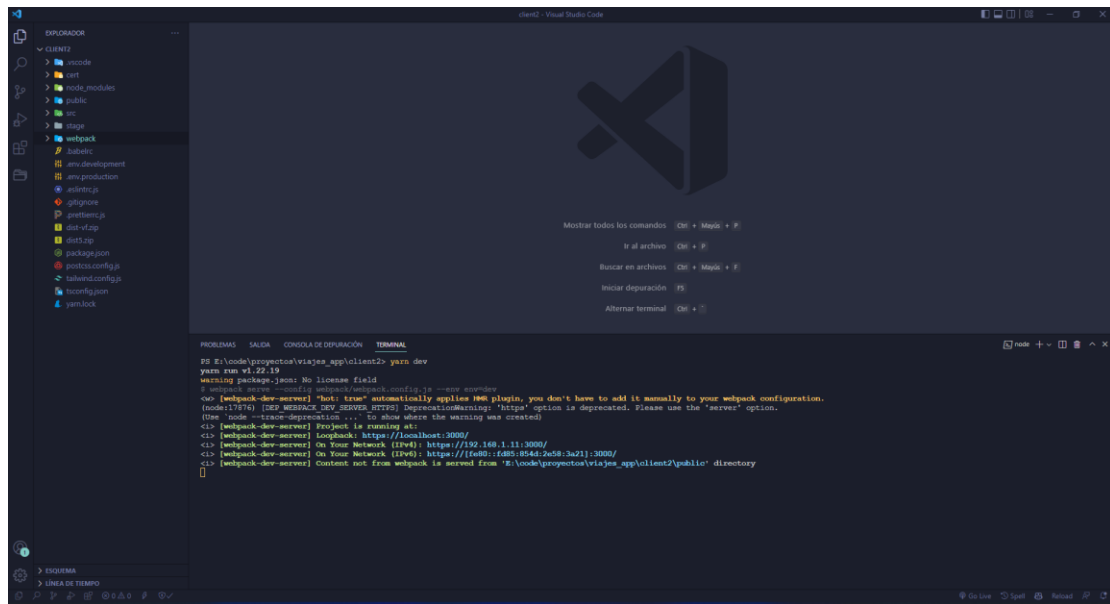


Fig. 10 Inicialización de entorno de Desarrollo

Si toda la configuración del proyecto ha salido bien, puedes acceder a él ingresando a través del siguiente enlace. <https://localhost:3000>. En Dicha ruta se mostrará el landing page del proyecto como se muestra en la figura 11.

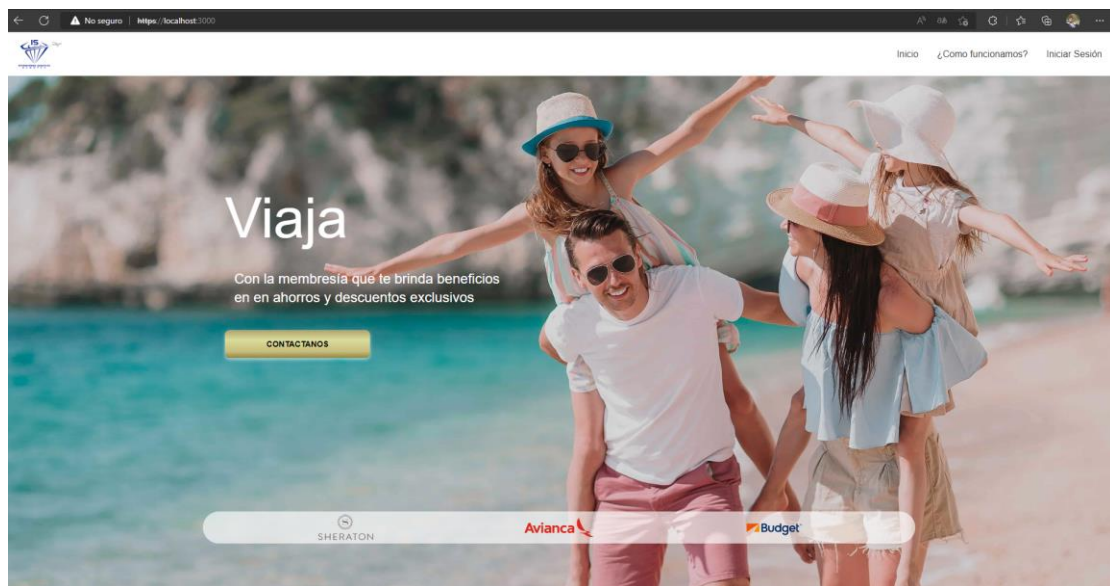


Fig. 11 Landing Page del aplicativo.

Ejecución en ambiente de producción

Una vez se tenga el efecto deseado en el desarrollo del software, lo siguiente es compilar el script para un entorno de producción, para realizar dicha acción se debe abrir una terminal de comandos y ubicarse en la raíz del proyecto, luego, ejecutar el script build como se indica a continuación.

`yarn build`

Esto generará una carpeta llamada dist en la raíz del proyecto como se muestra en la figura 12, esta carpeta contiene todos los bundles y chunks necesarios para que el proyecto pueda ser ejecutado en un servidor independiente o en el mismo proyecto donde se encuentra alojado el backend.

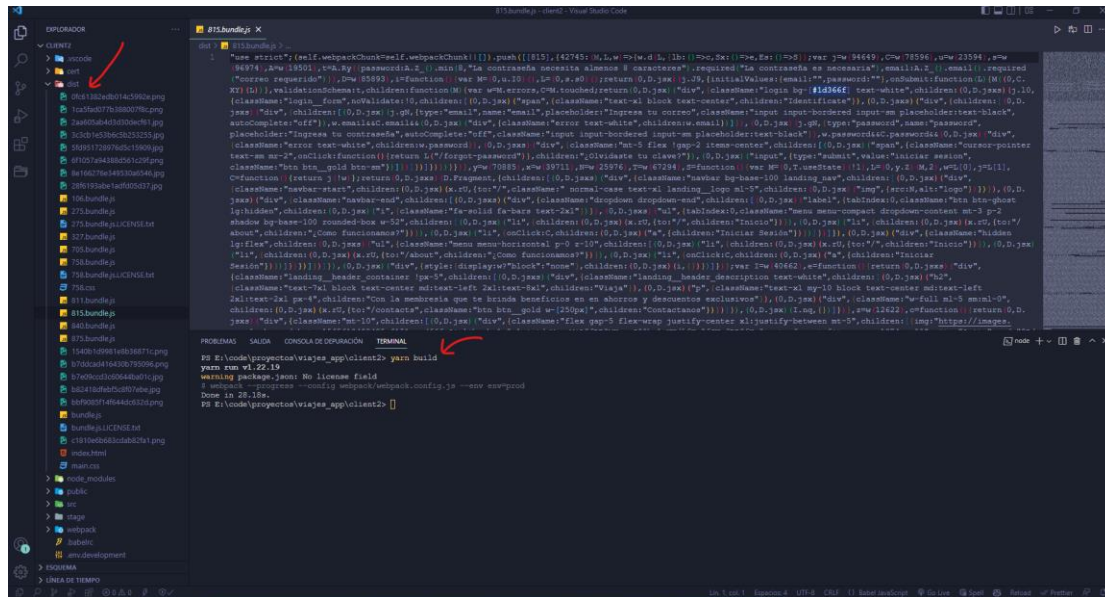


Fig. 12 Binarios compilados para producción

Nota. No olvidar establecer las variables de entorno para la compilación de producción.

Anexos

La instalación del software y dependencias listadas a continuación se realizó en un sistema operativo Windows como consecuencia de estar catalogado como el sistema operativo mas usado actualmente.

Instalación de Chocolatey

Chocolatey es un administrador de paquetes de software para Windows, si desarrollas en sistemas operativos MacOS (puedes instalar Homebrew o similar), si desarrollas en linux, debes consultar el gestor de paquetes utilizado por tu distribución.

La instalación de chocolatey únicamente debe realizarse si posees un sistema operativo Windows, este software no corresponde a una dependencia obligatoria, sin embargo, facilitara mucho el trabajo de instalación de todas las dependencias aquí listadas. Si deseas saltarte la instalación de este software puedes realizar la instalación tradicional del software requerido.

Para instalar el software de chocolatey, sigue la guía de instalación descrita en la pagina oficial en el siguiente enlace. [Chocolatey Software | Installing Chocolatey](#)

Instalación de Git.

- Si utilizas choco en tu sistema operativo Windows, únicamente abre una terminal de powershell y digita el siguiente comando.

Choco install git

- Alternativamente, si no deseas utilizar chocolatey, dirigirse a la página principal de Git [Git \(git-scm.com\)](#) y descargar los archivos ejecutables para tu sistema operativo como se muestra en la Figura 13.

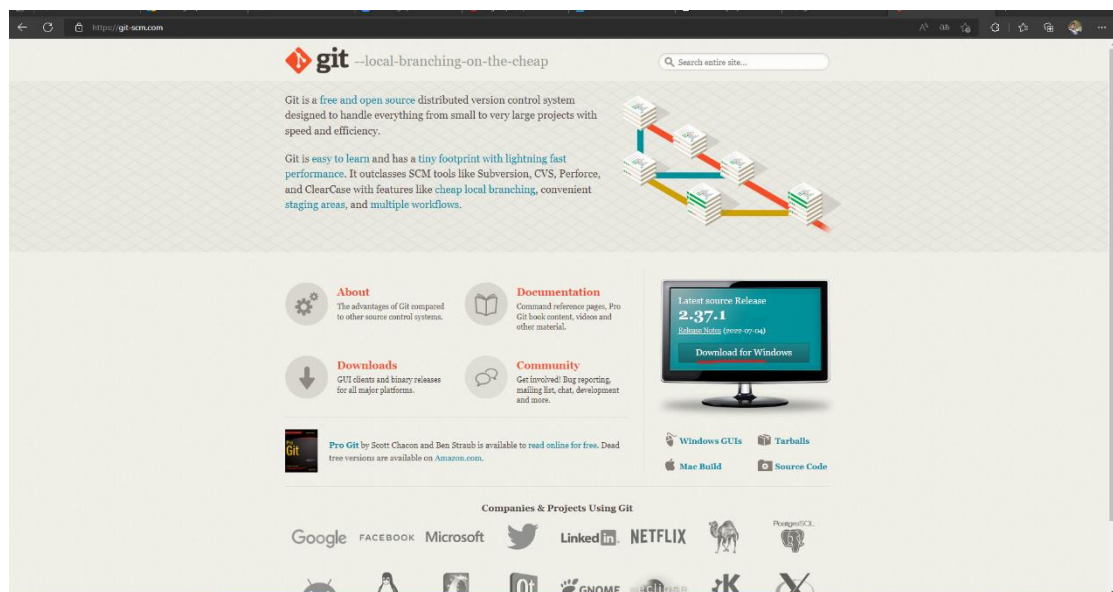


Fig. 13 Página oficial para la descarga de Git

- Abrir el archivo ejecutable, seleccionar la ruta de instalación y elegir las opciones Git bash y Git Gui. Como se muestra en la figura 14.

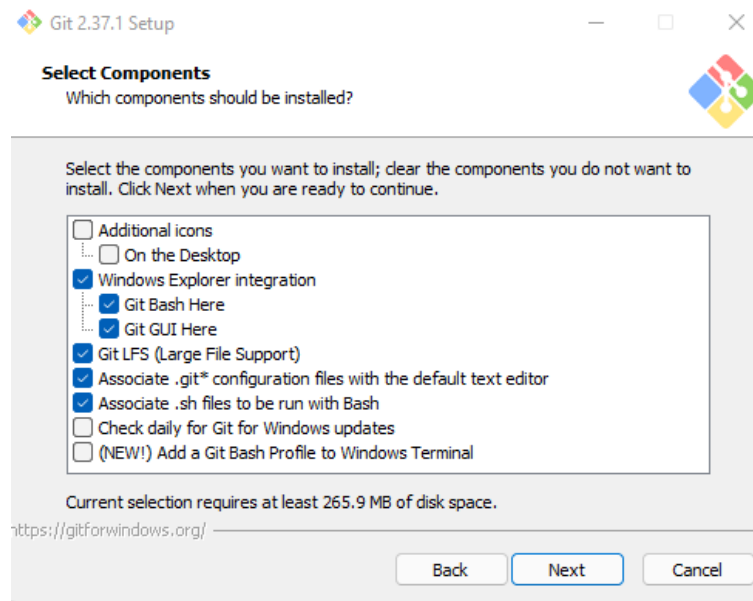


Fig. 14 Proceso de instalación de Git

- Hacer clic en siguiente, dejando los parámetros por defecto para todas las opciones posteriores y esperar que la instalación termine.
- Cuando la instalación termine, abre una terminal de powershell o cmd, y ejecuta el siguiente comando. `git --version`. Esta instrucción deberá imprimir en la consola la versión instalada de Git como se muestra en la figura 15, si en cambio se obtiene una salida diferente, asegúrate de que el proceso de instalación se haya completado exitosamente.

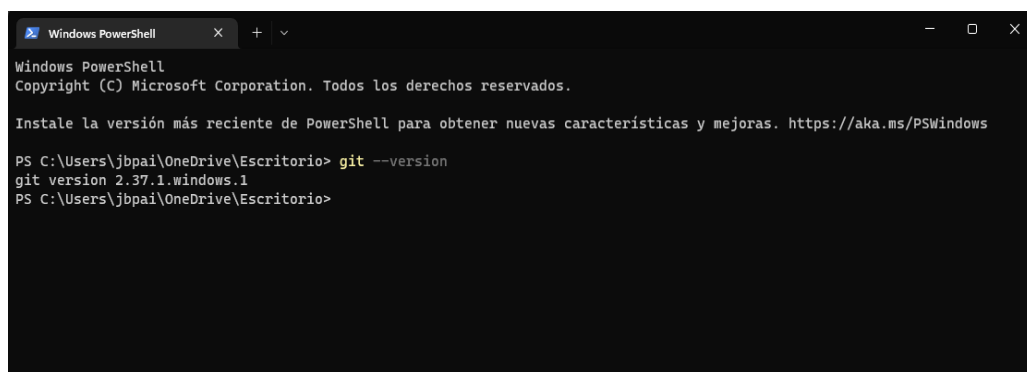


Fig. 15 Comprobación de version GIT

Configuración de GIT

Para utilizar el sistema de control de versiones de git, es necesario que se configuren ciertos parámetros si lo que deseas es subir archivos a cualquier repositorio, si únicamente necesitas realizar la descarga de repositorios puedes obviar este paso.

Las configuraciones básicas consisten en establecer el nombre de usuario y el correo electrónico que identifique a la persona que creara los commits del proyecto. No es necesario establecer permisos de administrador o sudo para la parametrización de estos comandos.

- Nombre de usuario
`git config --global user.name = "nombre de usuario"`
- Correo electrónico
`git config --global user.email=correo_electronico@example.com`
- Configuración de editor (Opcional)
`git config --global core.editor code`

Una vez que hayas configurado los parámetros básicos puedes revisar que todo este bien listando los parámetros de configuración con el siguiente comando.

- `git config --global --list`

Para establecer mas configuraciones visitar la documentación en la página oficial de git.

Instalación de NODEJS

El proyecto se desarrollo utilizando la versión estable de Nodejs en su versión (v16.16.0 LTS), sin embargo, no presenta problemas de compactibilidad para las versiones más actuales (v18.7.0).

- Si posees choco en tu sistema únicamente ingresa el siguiente comando.

`choco install nodejs`
- Si no deseas usar chocolatey, para la descarga e instalación de Nodejs debes ingresar en su página oficial [Node.js \(nodejs.org\)](https://nodejs.org) y seleccionar los ficheros ejecutables para tu sistema operativo como se muestra en la figura 16.



Fig. 16 Página oficial para la descarga de NodeJS

- Abrir el ejecutable, especificar la ruta de instalación y hacer clic en siguiente, dejando los parámetros por defecto para todas las opciones posteriores hasta que la instalación termine.
- Por defecto el instalador de Nodejs también instala por defecto el ejecutable de NPM (Node Package Manager), sin embargo, en algunos sistemas operativos este puede no ser el caso, abre tu consola o terminal y verifica que la instalación de Nodejs y NPM se ha realizado correctamente como se muestra en la figura 17.



```
PowerShell 7 (x64)
PowerShell 7.2.4
Copyright (c) Microsoft Corporation.
https://aka.ms/powershell
Type 'help' to get help.

PS C:\Users\jbpai> node --version
v18.7.0
PS C:\Users\jbpai> npm --version
8.15.0
PS C:\Users\jbpai>
```

Fig. 17 Comprobación de version NodeJS y NPM

Instalación de YARN

La instalación de esta dependencia es opcional, YARN es gestor de paquetes al igual que NPM, por lo tanto, si ya posees una instalación de NPM en tu sistema puedes reemplazar el comando 'yarn' por 'npm' en cualquier parte del ciclo de desarrollo, no obstante, si presentas problemas de lentitud en las descargas de npm, es recomendable usar el gestor yarn en su lugar.

- Para la instalación del gestor de paquetes yarn es necesario que cuentes con una versión de Nodejs superior a la 16.10.
- Para descargar el gestor de paquetes puedes ingresar a su pagina oficial y descargar los ejecutables de acuerdo a las especificaciones de tu sistema operativo, sin embargo si posees el gestor de paquetes npm instalado en tu sistema puedes realizar la instalación de Yarn con el siguiente comando con permisos de administrador o sudo si usas mac o linux.

```
npm install --global yarn
```

- Si tu sistema operativo es Windows y estas acostumbrado a la instalación de paquetes a través de un sistema gestor de paquetes como chocolatey puedes instalar el gestor yarn a través del siguiente comando

```
Choco install yarn
```

- Para mas opciones de instalación, visita la guía o documentación oficial, publicada en su página oficial, puedes acceder a ella a través de este enlace. [Getting Started | Yarn \(yarnpkg.com\)](https://yarnpkg.com)

Instalación de MkCert

Mkcert es un software que proporciona certificados validos para la entidad que los emite con el único objetivo de desarrollar software en un ambiente de navegación segura. La instalación de mkcert no es obligatoria, puede reemplazarse con un software similar para la emisión de certificados o ejecutar el proyecto sin la necesidad de especificar los certificados.

La instalación del software mkcert puede realizarse a través de chocolatey a través del siguiente comando.

```
choco install mkcert
```

Para mas opciones de instalación visitar la documentación oficial en la página del fabricante.
mkcert.org