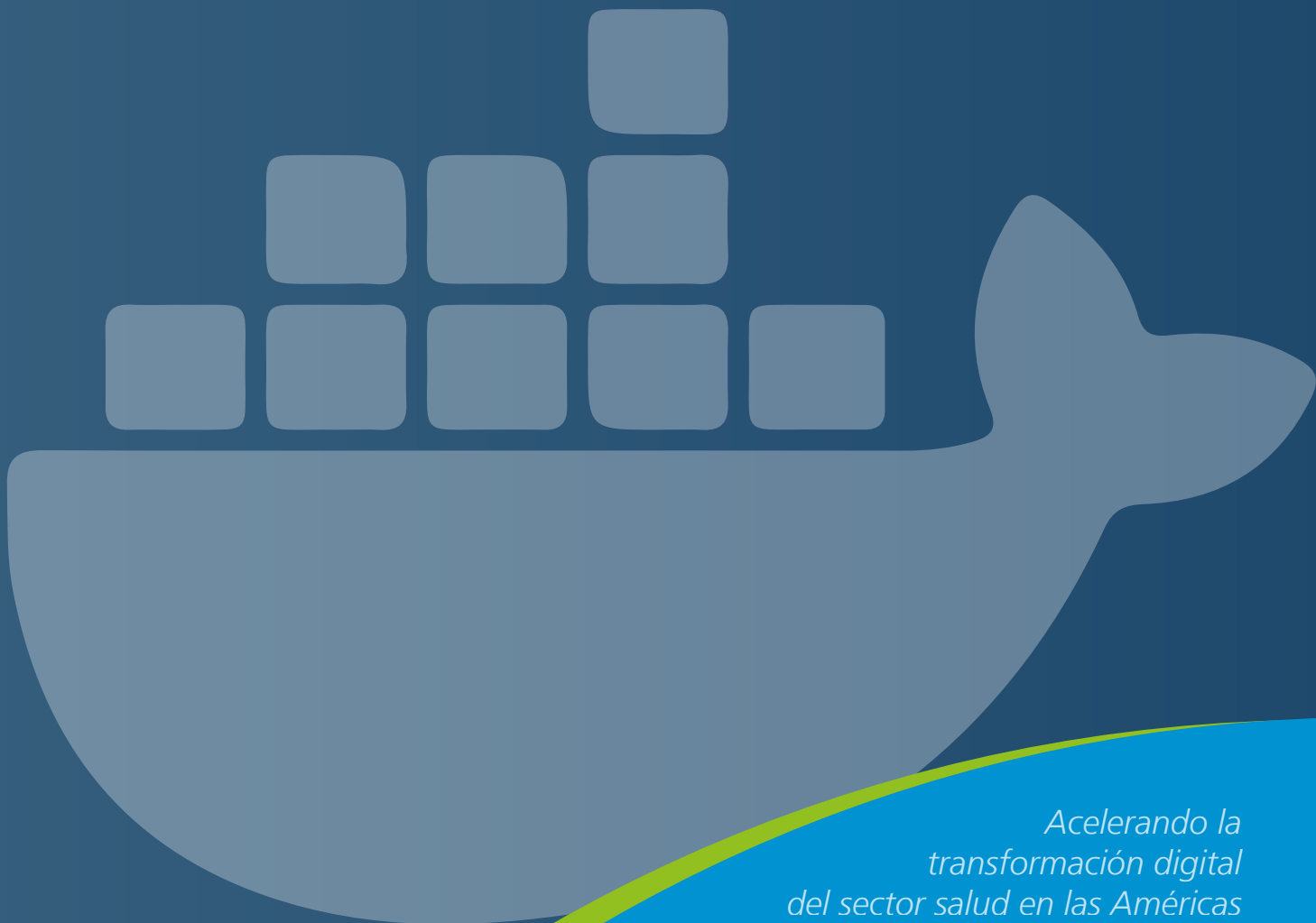




*Acelerando la
transformación digital
del sector salud en las Américas*

ALL-IN-ONE

Guía de Despliegue con Docker

OPS



Organización
Panamericana
de la Salud



Organización
Mundial de la Salud
Región de las Américas



ALL-IN-ONE

Guía de Despliegue con Docker

Versión 2.1

Washington, D.C., 2024

OPS



Organización
Panamericana
de la Salud



Organización
Mundial de la Salud
Región de las Américas





Contenidos

1. Descripción	3
2. Requerimientos mínimos	4
3. Proceso de despliegue	5
4. Despliegue de módulo de telesalud	7
4.1. Submódulo frontend	
4.2. Submódulo backend	
5. Despliegue de OpenEMR	10
6. Reverse Proxy	12
6.1. Configuración de red compartida entre contenedores	
6.2. Configuración de NPM	
7. Anexo: Actualización de los módulos	15
8. Documentación de referencia	16



1. Descripción

La plataforma All In One es una solución de registro clínico y de telesalud basado en componentes de software libre y de código abierto para ser compartido como Bien Público Digital por la OPS.

La solución es la combinación de la adaptación del sistema de registros clínicos electrónicos OpenEMR (www.open-emr.org) y la creación de un módulo de video consultas basado en la tecnología de código abierto de videoconferencias de Jitsi (www.jitsi.org) con interfaces de interoperabilidad entre ambas para la comunicación de la información generada en la video consulta.

La solución está diseñada como una plataforma con dos módulos que pueden ser implementados de manera independiente e integrados a los sistemas preexistentes en el país.

Esta guía tiene como propósito servir de orientación para el despliegue de los módulos de OpenEMR y Telesalud en un solo servidor. Sin embargo, es importante destacar que los módulos son independientes entre sí y pueden desplegarse en servidores separados según sea necesario. Esto permite que la arquitectura sea flexible, pudiendo adaptarse o escalarse según los requerimientos específicos de cada implementación. Se asume como precondition que se cuentan con conocimientos básicos de Docker y Docker Compose.

Más información: <https://docs.docker.com/>.



2. Requerimientos Mínimos

Los requerimientos mínimos son los siguientes:

- Un servidor o VPS con 2 o 4 núcleos, 2 GB de RAM.
 - Estos requerimientos son los mínimos indispensables para un entorno de pruebas, los requerimientos de hardware deben ser escalados dependiendo de la cantidad de usuarios, el factor más importante es la cantidad de teleconsultas concurrentes.
- Un dominio con 3 subdominios apuntando a la IP del servidor.
- Un servidor de correos con SMTP y POP3 o IMAP en caso de desplegar una mesa de ayuda.



3. Proceso de despliegue

En esta guía se toma como escenario un servidor basado en Ubuntu, el mismo ya tiene instalado previamente Docker, Docker Compose y git.

Al mismo tiempo se tomaron las medidas de seguridad pertinentes.

El usuario que se utilizará será *ubuntu*.

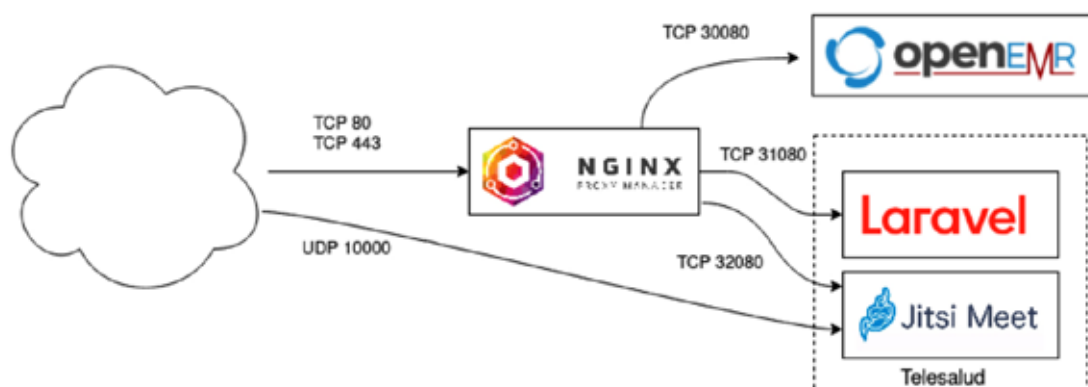
Los puertos abiertos en el servidor son los siguientes:

- TCP
 - 22 (ssh, en caso de ser necesario)
 - 80 (http)
 - 443 (https)
 - 4443 (Jitsi videobridge fallback, en caso de instalar módulo de telesalud)
- UDP
 - 10000 (Jitsi videobridge, en caso de instalar módulo de telesalud)

Debido a que múltiples módulos requieren de los puertos http y https, se utiliza un reverse proxy basado en Nginx Proxy Manager que enruta el tráfico en base al dominio solicitado.

El siguiente diagrama muestra gráficamente la estructura de la solución desplegada.

Figura 1. Diagrama All in one



El primer paso es crear el directorio que alojará todos los archivos.

Se recomienda que no sea un directorio personal a menos que solo un usuario se encargue del mantenimiento de la solución.

En esta guía se utilizará el directorio */opt/ops*.

Se procede a crearlos y asignar los permisos necesarios.

En esta guía se usa el grupo *ubuntu*, pero se puede crear otro grupo para tener mejor control de los permisos.

```
cd /opt
sudo mkdir ops
sudo chown -R ubuntu: ubuntu ops
cd ops
```



4. Despliegue de módulo de telesalud

El Módulo de Telesalud se compone de dos submódulos:

- Submódulo de backend, basado en Jitsi, que provee el servicio de videoconferencia.
- Submódulo de frontend basado en Laravel, que provee seguridad, control y características extras al módulo de telesalud.

4.1. Submódulo frontend

En primer lugar, se clona el repositorio del módulo en un nuevo directorio llamado *telesalud*.

```
git clone https://github.com/ciips-code/ciips-telesalud.git telesalud
cd /opt/ops
cd telesalud
```

Luego, se hace una copia del archivo

```
env.example llamada .env
cp .env.example .env
```

También se debe hacer una copia del archivo

```
docker-compose.prod.yml a docker-compose.yml
cp docker-compose.prod.yml docker-compose.yml
```

Con un editor de texto, se edita el archivo *.env*

nano .env

Dentro de este archivo se encuentran comentadas las líneas que se deben personalizar, detalladas a continuación.

- *WEB_LISTEN_PORT*: El puerto que escuchará las peticiones HTTP, de acuerdo al diagrama, el puerto será el 31080.
- *WEB_HTTPS_LISTEN_PORT*: Puerto que escuchará HTTPS, en este despliegue no será utilizado, sin embargo por convención se debe utilizar el 31443.
- *DB_LISTEN_PORT*: Puerto para conectarse con la base de datos Mysql del módulo, por seguridad debe estar protegido del exterior y luego puede omitirse. Por convención se debe utilizar el 31006.
- *APP_URL*: Url principal del módulo, en esta guía se utilizará el dominio *portaldepacientes.com* como dominio principal, y este submódulo utilizará el subdominio *vc*. De manera que el valor del campo en esta guía será *https://vc.portaldepacientes.com*.

- LANGUAGE: Idioma del módulo, los valores aceptados son *es* y *en*.
- TIMEZONE: Zona horaria, se pueden consultar los valores posibles en <https://www.php.net/manual/en/timezones.php>
- NOTIFICATION_URL: Una URL para el endpoint de OpenEMR que recibirá las notificaciones sobre cambios en teleconsultas. Para esta guía será https://emr.portal-depacientes.com/telehealth/controllers/C_TSalud_Vc.php?action=vcNotify
- NOTIFICATION_TOKEN: En caso de que el endpoint requiera seguridad, se establece el token.
- DB_DATABASE: Nombre del esquema de base de datos, modificarlo es opcional.
- DB_USERNAME: Nombre de usuario de base de datos que utilizará el submódulo, modificarlo es opcional.
- DB_PASSWORD: Contraseña de base de datos, utilice una clave fuerte. **NOTA: evitar usar espacios o los símbolos numeral o barra invertida (# o \) debido a que estos rompen el esquema del archivo .env**
- JITSY_PROVIDER: El submódulo de frontend tiene la posibilidad de trabajar con el submódulo de backend hosteado localmente o bien con la instancia pública de Jitsi, por cuestiones de seguridad esta última opción no es recomendada. Los valores posibles son *jitsi* para utilizar la instancia pública o *self* para utilizar una instancia hosteada por usted.
- JITSY_BASE_URL: URL base del submódulo de backend, para esta guía se utilizará <https://vcbknd.portaldepacientes.com/>
- JITSY_APP_ID: Identificador de la integración con Jitsi, debe coincidir con el valor configurado en JWT_APP_ID en la configuración de Jitsi. Para esta guía se utilizará *telesalud*.
- JITSY_APP_SECRET: Token secreto de la integración con Jitsi, debe coincidir con el valor configurado en JWT_APP_SECRET en la configuración de Jitsi. Para esta guía se utilizará *OafDjrVt8r*. Es recomendable utilizar una clave fuerte. **NOTA: evitar usar espacios o los símbolos numeral o barra invertida (# o \) debido a que estos rompen el esquema del archivo .env**

Editar el archivo *docker-compose.yml*. Cambiar la contraseña de root de MySQL en la línea 30.

Si bien en este despliegue no se utilizará https internamente, se debe crear un certificado y clave para evitar errores en el submódulo. Para ello se debe ejecutar el siguiente comando, se le harán algunas preguntas, las cuales pueden ser omitidas presionando enter.

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout docker-config/cert.key -out docker-config/cert.crt
```

Se debe ejecutar el submódulo con el comando `docker compose up -d`. Este proceso tardará unos minutos la primera vez que se ejecuta.

Una vez que se haya ejecutado, se deben ejecutar una serie de comandos. Estos comandos deben ser ejecutados solamente la primera vez que se levanta el submódulo:

1. **`docker compose exec app php artisan key:generate`** Para generar una clave única.
2. **`docker compose exec app php artisan migrate`** Para crear la estructura de base de datos.
3. **`docker compose exec app php artisan token:issue`** Para generar un token de acceso a la API, el valor resultante debe ser configurado en el archivo `.env` de OpenEMR, por lo que debe guardarlo para más adelante.

4.2. Submódulo backend

Este submódulo se encuentra en el directorio `jitsi-docker`
`cd jitsi-docker`

Se debe hacer una copia del archivo `.env.example` llamada `.env`
`cp .env.example .env`

Editar este archivo personalizando los siguientes parámetros:

- **HTTP_PORT**: Puerto que escucha las peticiones http. Siguiendo la guía será el 32080.
- **HTTPS_PORT**: Puerto que escucha las peticiones https. No se usa en este despliegue pero por convención se configura el 32443.
- **TZ**: Zona horaria, se configura de igual manera que el submódulo frontend.
- **PUBLIC_URL**: URL del submódulo de backend. Para esta guía será `https://vcbknd.portaldepacientes.com`
- **JWT_APP_ID**: Identificador de la integración con el frontend, debe coincidir con el configurado previamente.
- **JWT_APP_SECRET**: Clave secreta de la integración con el frontend, debe coincidir con el configurado previamente.

Luego se debe correr el comando `./gen-passwords.sh` para generar claves fuertes de los servicios de Jitsi.

Por último se debe ejecutar **`docker compose up -d`**. Al igual que con el frontend, el primer arranque demora unos minutos.



5. Despliegue de OpenEMR

Como primer paso, se debe clonar el repositorio de openemr-telesalud en su rama *develop*. Los siguientes comandos clonan el repositorio en un directorio nuevo llamado *openemr* y se posiciona dentro del mismo.

```
cd /opt/ops
git clone https://github.com/ciips-code/openemr-telesalud.git openemr
cd openemr
```

En el directorio *ciips/docker* se encuentran los archivos necesarios para el despliegue del módulo de OpenEMR, por lo que se debe ubicar en dicho directorio

```
cd ciips/docker
```

Copiar el archivo

```
.env.example a .env y docker-compose.prod.yml a docker-compose.yml
cp .env.example .env
cp docker-compose.prod.yml docker-compose.yml
```

Editar el archivo *.env* personalizando los siguientes parámetros:

- COMPOSE_PROJECT_NAME: Nombre base para el proyecto dentro de docker.
- MYSQL_ROOT_PASSWORD: Contraseña para el usuario *root* en el contenedor MySQL, utilice una contraseña extra fuerte. **NOTA: evitar usar espacios o los símbolos numeral o barra invertida (# o \) debido a que estos rompen el esquema del archivo .env**
- MYSQL_PORT: Puerto donde escuchará el contenedor de MySQL, por razones de seguridad este puerto debe estar protegido del exterior. Para esta guía se utilizará el puerto 30006.
- MYSQL_DATABASE: Nombre del esquema de base de datos donde se almacenarán los datos del módulo.
- MYSQL_USER: Nombre de usuario de MySQL que utilizará el módulo.
- MYSQL_PASSWORD: Contraseña para el usuario de MySQL del módulo. Elija una contraseña fuerte. **NOTA: por problemas en el script de inicialización, evitar usar caracteres especiales en la contraseña**

- TELEHEALTH_BASE_URL: URL base del submódulo frontend del módulo de telesalud. Para esta guía se utilizará *<https://vc.portaldepacientes.com>*
- TELEHEALTH_PORT: Puerto de escucha del módulo de telesalud. Para esta guía se utiliza 443.
- TELEHEALTH_API_TOKEN: Token obtenido previamente luego de ejecutar ***docker compose exec app php artisan token:issue*** en el submódulo frontend.
- OPENEMR_PORT: Puerto donde OpenEMR escuchará peticiones http. Para esta guía se utilizará el puerto 30080.

Ejecutar el módulo con el comando ***docker compose up -d***



6. Reverse Proxy

6.1. Configuración de red compartida entre contenedores

En esta guía se utilizará Nginx Proxy Manager (en adelante, NPM) como solución de reverse proxy.

En caso de utilizar otra solución, se debe tener en cuenta que el backend del módulo de telesalud requiere soporte de Websockets, por lo que este tipo de tráfico también debe ser redirigido.

Se desplegará NPM en un directorio llamado *proxy* en el directorio principal:

```
cd /opt/ops
mkdir proxy
cd proxy
```

Dentro de este directorio, se crea un archivo *docker-compose.yml* con los contenidos del siguiente enlace <https://pastebin.com/LbhRq22V> (debido a la naturaleza de YAML, no es práctico copiar el contenido desde este documento). Si se desea cambiar el puerto de administración web de NPM, se puede personalizar la línea 10, por defecto el puerto es el 81. Si se desea personalizar, se debe cambiar el primer número de la línea. Por ejemplo, si se desea utilizar el puerto 3081, la línea se edita de esta manera: - '3081:81'.

Se debe ejecutar el comando *docker compose up -d* y esperar.

6.2. Configuración de NPM

Para configurar NPM se debe abrir un navegador y dirigirse a la IP o dominio del servidor y el puerto elegido, para esta guía es el 81. Es decir que para esta guía la URL resultante es <http://portaldepacientes.com:81/>. Nótese que se utiliza http y no https, esto es para evitar problemas de certificados en la primera etapa.

En la pantalla de inicio de sesión se deben introducir el usuario y contraseña por defecto:

- **Usuario:** admin@example.com
- **Contraseña:** changeme

Una vez que se ingresa, se solicitará cambiar el usuario y contraseña, elija una contraseña fuerte.

Luego de cambiar los datos, se ingresará a la pantalla principal de NPM. Para configurar el acceso a los módulos se debe ir al menú **Hosts**, opción **Proxy Hosts**. En la siguiente pantalla estarán listados los módulos, en un principio este listado estará vacío. Se debe agregar un nuevo *Host* presionando el botón **Add Proxy Host**. Esto abre un popup donde se debe configurar una serie de parámetros para cada servicio:

- Pestaña Details
 - *Domain names*: En este campo se configura el dominio para el servicio, por ejemplo ***vc.portaldepacientes.com***
 - *Forward Hostname/IP*: Se debe especificar el nombre del contenedor que responde las peticiones http de cada módulo. Para esta guía de ejemplo se lista más abajo cada nombre.
 - *Forward Port*: El puerto a redireccionar, siempre se ingresa 80
 - *Block common exploits*: activar siempre
 - *Websockets support*: activar para ***telesalud-backend***
- Pestaña SSL
 - *SSL Certificate*: Seleccionar opción ***Request a new SSL certificate***. Esto hará que se solicite un certificado de Let's encrypt automáticamente
 - *Force SSL*: Activar para que siempre se use https
 - *HTTP/2 Support*: Activar para que se activen mejores características.

Luego se presiona en Save y se deben esperar unos segundos. Se puede navegar hacia el nuevo dominio para comprobar que la configuración fue correcta.

A continuación se detalla la configuración de los tres servicios que se deben crear con las configuraciones que deben ir de acuerdo a los valores ingresados en esta guía:

- OpenEMR
 - Pestaña *Details*
 - ▷ *Domain names*: ***emr.portaldepacientes.com y/o www.portaldepacientes.com***
 - ▷ *Forward Hostname/IP*: ***openemr-web***
 - ▷ *Forward Port*: 80
 - ▷ *Block common exploits*: Activar
 - Pestaña *SSL*
 - ▷ *SSL Certificate*: Request a new SSL certificate
 - ▷ *Force SSL*: Activar
 - ▷ *HTTP/2 Support*: Activar

- Telesalud frontend
 - Pestaña *Details*
 - ▷ *Domain names:* **vc.portaldepacientes.com**
 - ▷ *Forward Hostname/IP:* **telesalud-frontend**
 - ▷ *Forward Port:* 80
 - ▷ *Block common exploits:* Activar
 - Pestaña *SSL*
 - ▷ *SSL Certificate:* **Request a new SSL certificate**
 - ▷ *Force SSL:* Activar
 - ▷ *HTTP/2 Support:* Activar
- Telesalud backend
 - Pestaña *Details*
 - ▷ *Domain names:* **vcbknd.portaldepacientes.com**
 - ▷ *Forward Hostname/IP:* **telesalud-backend**
 - ▷ *Forward Port:* 80
 - ▷ *Block common exploits:* Activar
 - ▷ *Websockets Support:* Activar
 - Pestaña *SSL*
 - ▷ *SSL Certificate:* **Request a new SSL certificate**
 - ▷ *Force SSL:* Activar
 - ▷ *HTTP/2 Support:* Activar

Con estas acciones se da por finalizado el despliegue de los módulos y está listo para funcionar.

Para comprobar su instalación debe navegar hacia la URL definida para el módulo OpenEMR.

Para esta guía sería <https://portaldepacientes.com>.

Si todo salió bien debería verse la pantalla de inicio de sesión de OpenEMR. Las credenciales de ingreso por defecto son las siguientes:

- **Nombre de usuario:** admin
- **Contraseña:** AdminOps2023**

Debido a que estas credenciales son estándares para todas las instalaciones realizadas con docker se recomienda cambiarlas para proteger la instalación.





7. Anexo: Actualización de los módulos

En caso de requerir actualizar los módulos se deben seguir los siguientes pasos:

1. Módulo de telesalud
 - a. Submódulo frontend
 - i. `cd /opt/ops/telesalud`
 - ii. `git pull`
 - iii. Examinar el archivo `docker compose.prod.yml` en busca de cambios y actualizar el archivo `docker compose.yml` en caso de ser necesario
 - iv. `docker compose build --no-cache`
 - v. `docker compose up -d`
 - b. Submódulo backend
 - i. `cd /opt/ops/telesalud/jitsi-docker`
 - ii. Examinar el archivo `.env.example` en busca de cambios y actualizar el archivo `.env` en caso de ser necesario
 - iii. `docker compose up -d`
2. Módulo OpenEMR
 - a. `cd /opt/ops/openemr`
 - b. `git pull`
 - c. Examinar el archivo `Readme.md` para encontrar instrucciones extra de actualización en el log de cambios
 - d. `cd ciips/docker`
 - e. `docker compose build --no-cache`
 - f. `docker compose up -d`



8. Documentación de Referencia

Recursos oficiales Docker

<https://docs.docker.com/>

Documentación oficial de Laravel

<https://laravel.com/docs/9.x>

Documentación oficial de Jitsi

<https://laravel.com/docs/9.xhttps://jitsi.github.io/handbook/docs/intro/>

Repositorio del Módulo de Registro Clínico Electrónico

<https://github.com/ciips-code/openemr-telesalud>

Repositorio de la Plataforma

<https://github.com/ciips-code/>

<https://github.com/ciips-code/ciips-telesalud>

Documentación oficial de OpenEMR

https://www.open-emr.org/wiki/index.php/OpenEMR_Wiki_Home_Page



OPS

www.paho.org



Organización
Panamericana
de la Salud



Organización
Mundial de la Salud
Región de las Américas