



Conectarse a un servidor que tiene firewall

Prof. Antonio López Jaimes

1^{er} escenario

Conexión a un puerto no accesible directamente de un servidor



Puerto
22



labred.cua.uam.mx

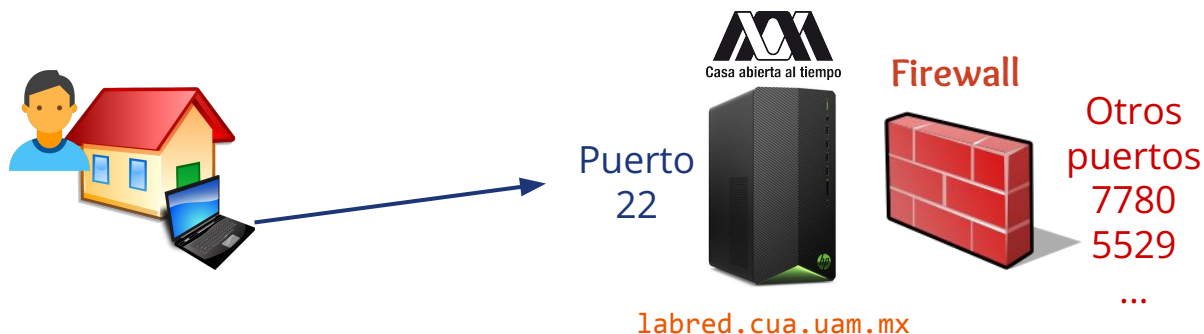
Otros
puertos
7780
5529

...

Descripción del escenario

Queremos tener nuestro **proceso servidor** en una máquina remota y el **cliente en nuestra casa**, y tenemos el siguiente escenario.

- Tenemos acceso directo a un servidor remoto (tiene **IP pública**).
 - PERO, por seguridad, **solamente tiene abierto el puerto 22** para conexiones por SSH.
- Nuestra computadora podría tener **IP pública** o **local**.



Resumen



Desde nuestra PC de casa creamos el túnel al puerto destino de la máquina remota:

```
casa> ssh -L 7780:localhost:7780 menganito@148.206.180.144
```

Desde nuestra PC de casa usamos nuestro programa cliente mediante:

```
casa> ./cliente 127.0.0.1 7780
```

Noten que usamos la IP local **127.0.0.1**, pero como tendimos el túnel, la solicitud llegará al servidor remoto **148.206.180.144**.

**Explicación de los pasos
anteriores...**

Paso 1: establecer el túnel

Desde nuestra PC de casa creamos el túnel al puerto destino de máquina remota:

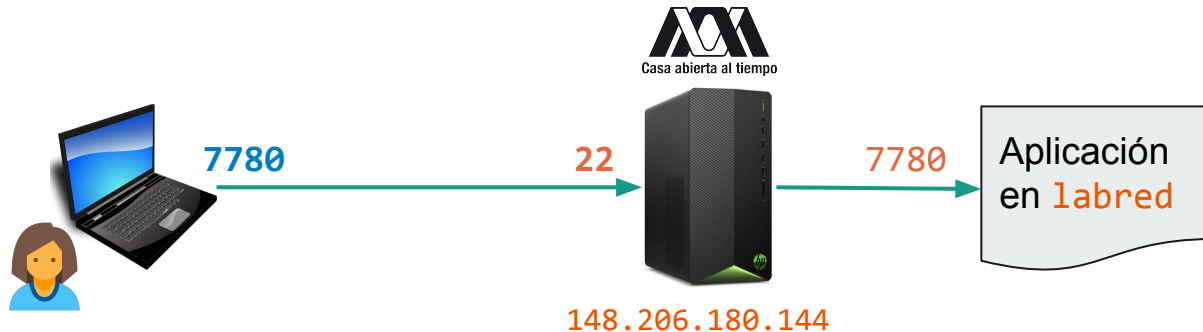
```
casa> ssh -L 7780:localhost:7780 antonio@148.206.180.144
```

Máquina local

Máquina remota



Lo anterior hará que toda petición a nuestro puerto 7780 se redirija al puerto 7780 pero de la labred.



Nota: por omisión el servidor SSH escucha en el **puerto 22**

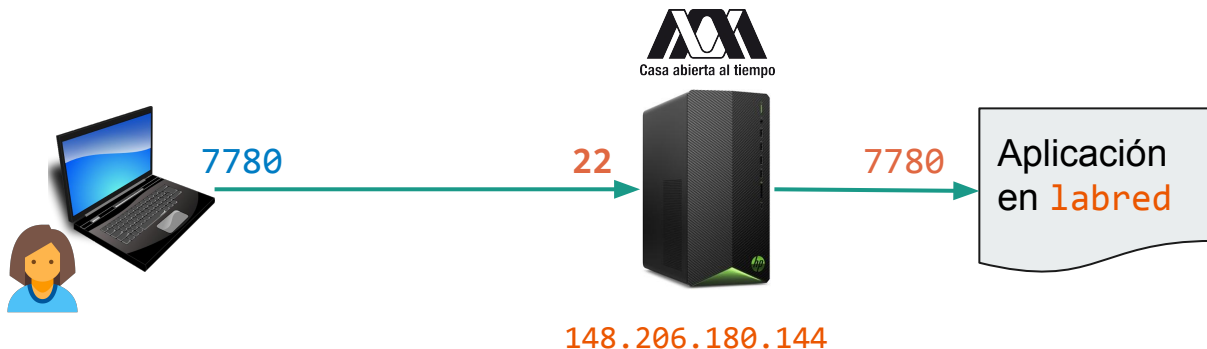
Paso 2: usar nuestro cliente con un pequeño cambio

Desde nuestra PC de casa usamos nuestro programa cliente mediante:

```
casa> ./cliente localhost 7780
```



El nombre *localhost* se refiere a nuestra PC, pero ya con el túnel establecido, la petición viajará hasta *labred*, y una vez allí se hará la petición a la aplicación del *puerto 7780*.



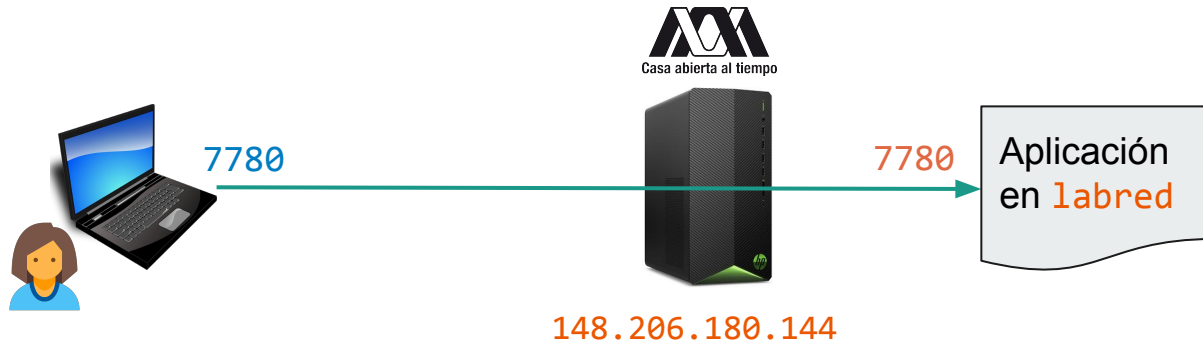
Paso 2: usar nuestro cliente con un pequeño cambio

Desde nuestra PC de casa usamos nuestro programa cliente mediante:

```
casa> ./cliente localhost 7780
```



El nombre *localhost* se refiere a nuestra PC, pero ya con el túnel establecido, la petición viajará hasta *labred*, y una vez allí se hará la petición a la aplicación del *puerto 7780*.



2.º escenario

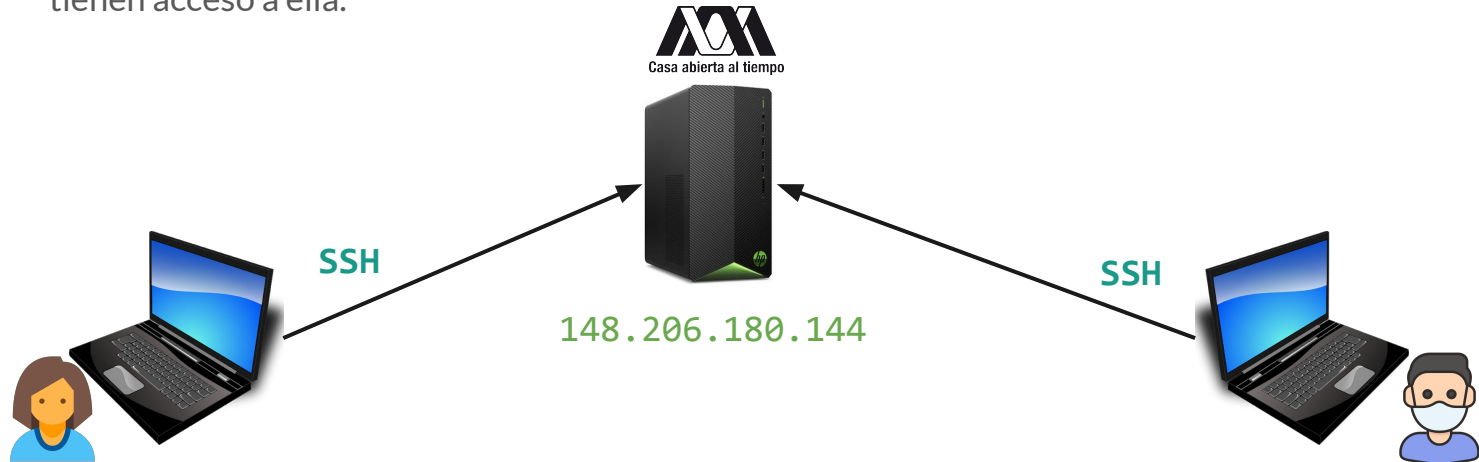
Mi PC en casa como servidor



Descripción del escenario

Queremos conectar **dos computadoras remotas de casa** para ejecutar de un lado un **proceso servidor** y en la otra los **procesos cliente**. Tenemos los siguientes elementos:

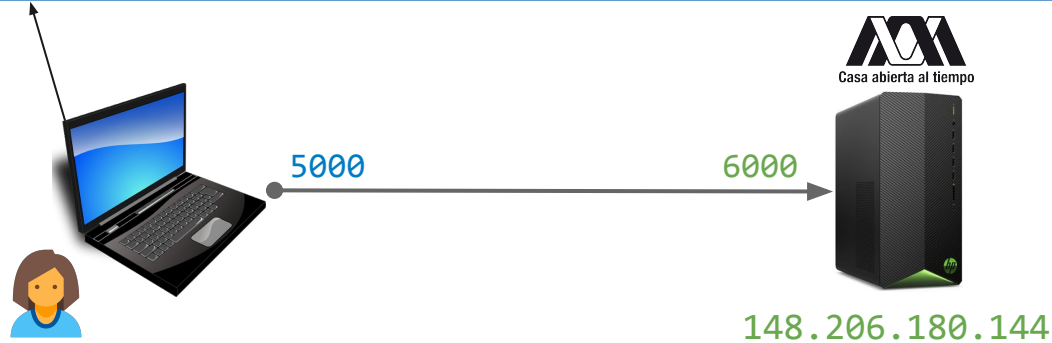
- Ambas computadoras solo tienen **IP local** (Infinitum, Totalplay, Izzi, etc.).
- **Tenemos acceso a una tercer computadora con IP pública**. Es decir, ambas computadoras de casa tienen acceso a ella.



Paso 1: el cliente crea un túnel hacia Ryoma (148.206.180.144)

La computadora de casa que **será el cliente** debe crear un túnel hacia la computadora accesible **Ryoma**:

```
Ada@casa> ssh -L 5000:localhost:6000 menganito@148.206.180.144
```



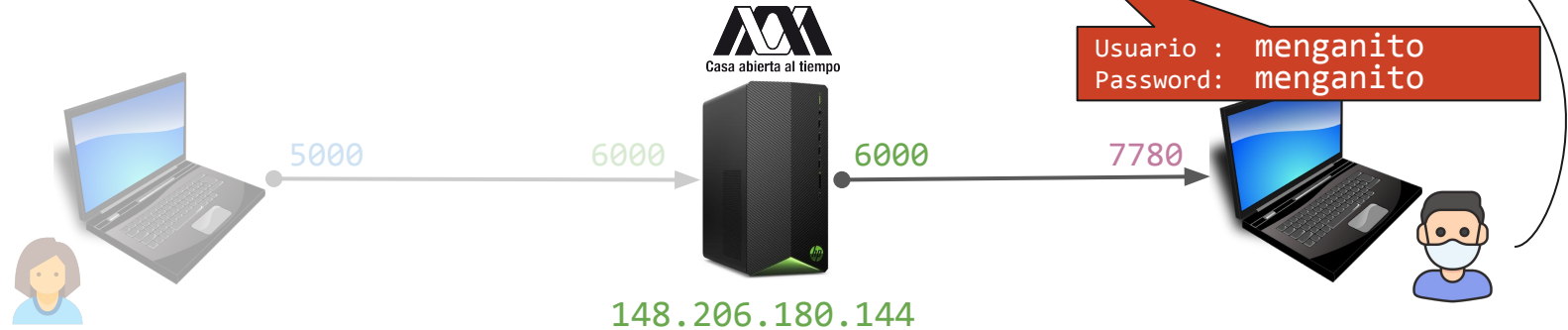
Usuario : menganito
Password: menganito

Lo anterior hará que toda petición a la computadora de **Ada** a su puerto **5000** se redireccione al **puerto 6000** pero de la **Ryoma**.

Paso 2: el **servidor** crea un túnel **inverso** hacia **Ryoma**

Para crear el túnel **inverso**, la computadora **que será el servidor** debe ejecutar la siguiente instrucción:

```
Babbage@casa> ssh -R 6000:localhost:7780 menganito@148.206.180.144
```



Lo anterior hará que toda petición a **Ryoma** a su puerto **6000** se redirija al **puerto 7780** de la computadora de **casa de Babbage**.

Paso 3: ejecutar nuestra aplicación usando el túnel

Babbage debe ejecutar en su computadora el programa servidor de manera normal:

```
Babbage@casa> ./servidor
```

Por su parte, **Ada** debe ejecutar el cliente en su computadora de la siguiente manera:

```
Ada@casa> ./cliente 127.0.0.1 5000
```

