

contenedor I4H, abre otra consola PowerShell y ejecuta lo siguiente:

```
docker run --net=host --ipc=host --uts=host --pid=host -it \
--security-opt=seccomp=unconfined --privileged --rm -v /:/host alpine /bin/sh
```

Con esto descargareis una imagen de Linux Alpine (si no lo tenéis ya), arrancaréis un contenedor ¡ligado a tu máquina mobyLinux! y entraréis en la shell de Alpine. Una vez allí, al ejecutar un simple `ls -l`, veréis que hay un directorio `/host`, y ¡¡jet voilà!!!, dentro de él habrá un `./home/user1` donde tendréis todo vuestro contenido que queréis mantener.

Por supuesto existe la posibilidad de ubicar el `/durable` en un directorio de vuestro Windows. Si ejecutamos lo mismo pero cambiando el path (para que no haya conflicto de nombre y puertos debemos parar antes el otro contenedor con `docker stop my-iris`):

```
docker run --name my-iris -d --publish 9091:51773 --publish 9092:52773 \
--volume c:/Temp/Data:/durable \
--env ISCDATADIRECTORY=/durable/iris store/intersystems/iris-community:2019.4.0.379.0
```

Comprobaremos que, si existe en nuestro Windows el directorio `c:/Temp/Data`, se habrán copiado archivos y BDs de sistema de I4H en ese directorio y, por tanto, serán accesibles desde Windows. ¡Precaución! El problema que nos podemos encontrar aquí es con los permisos de los directorios y ficheros. Es un problema común cuando se habla de persistencia y contenedores Linux en Windows, no sólo de I4H, y viene derivado del hecho de que la tecnología de contenedores viene básicamente del mundo Linux, de ahí que se necesite la máquina virtual mobyLinux, y de que no hay una equivalencia exacta entre el sistema de permisos de Windows y el de Linux...

De hecho, si volvéis a entrar en la shell de Alpine, veréis que también tenéis acceso a vuestro host Windows desde mobyLinux. Ejecutad lo siguiente: `cd /host/hostmnt/c/Temp/Data/iris` y estaréis, desde Alpine->MobyLinux, viendo el contenido de `c:/Temp/Data/iris`. Si ahí ejecutáis un `ls -l`, veréis que todos los directorios pertenecen a root y están en el grupo root. Si ahora ejecutáis en la shell de Alpine: `cd /host/home/user1/iris` y después `ls -l`, veréis que aparecen usuarios y grupos (no definidos) con uid 52773 (irisuser) y gid 51773 (irisowner) respectivamente. Este tema de los permisos puede darnos conflictos cuando usamos imágenes con SO base Linux con Docker for Windows... ya digo, no es exclusivo de I4H y tiene que ver con la no equivalencia entre usuarios/permisos en Linux y en Windows. Conviene tenerlo en cuenta para evitar quebraderos de cabeza.

IMPORTANTE: Docker resetea el directorio `/home` de la máquina virtual mobyLinux cada vez que se reinicia, por lo que si rearrancamos el ordenador o simplemente paramos e iniciamos Docker, veremos que todo el contenido de nuestro `/home/user1` ha desaparecido. Para preservarlo, puedes almacenarlo en el mismo sitio en que él almacena los volúmenes: `/var/lib/docker/volumes`; que, si entramos con nuestra "shell mágica", estaría en `/host/var/lib/docker/volumes`.

Así, podríamos arrancar el contenedor de este modo:

```
docker run --name my-iris -d --publish 9091:51773 --publish 9092:52773 --volume \
/var/lib/docker/volumes/user1:/durable --env ISCDATADIRECTORY=/durable/iris \
store/intersystems/iris-community:2019.4.0.379.0
```

y así nos aseguraríamos de que nuestra información persistente sigue ahí, no sólo aunque destruyamos nuestro contenedor, sino incluso tras reiniciar Docker for Windows (ojo, si reinstalamos Docker ya sí que perderíamos todo).

Bueno, espero que os haya servido.

¡¡Disfrutad!!

[#Consejos y trucos](#) [#Contenedorización](#) [#Docker](#) [#InterSystems IRIS](#) [#InterSystems IRIS for Health](#)

URL de
fuente: <https://es.community.intersystems.com/post/pistas-para-entender-y-lidiar-con-la-persistencia-en-docker-windows>