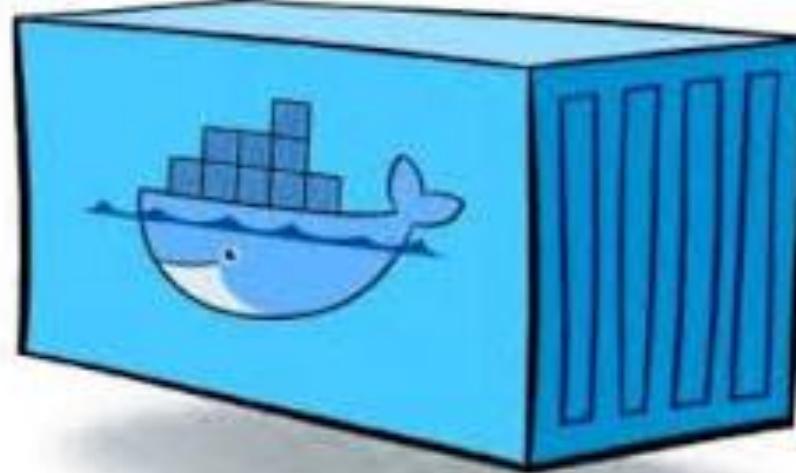


Contenedores

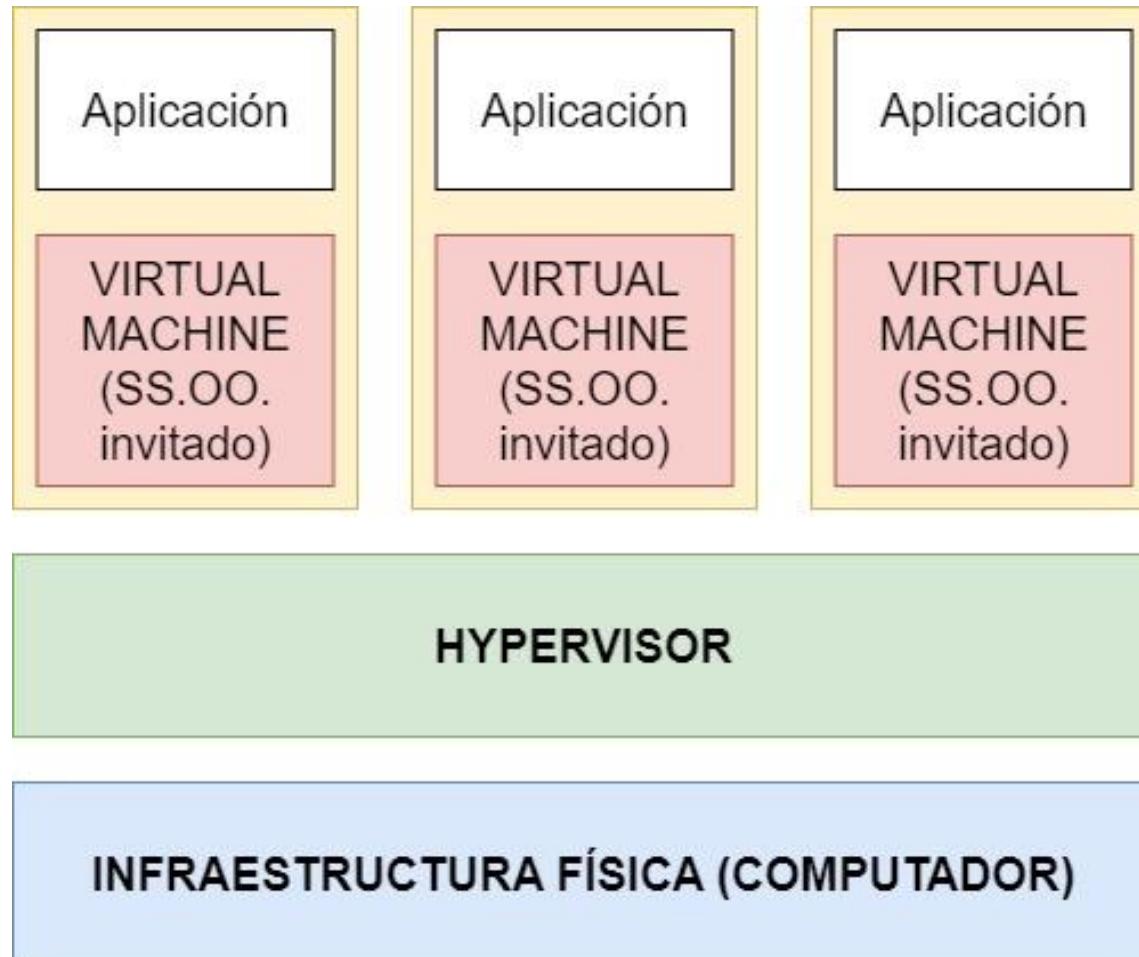


Qué no es un contenedor

Qué no es un contenedor

- **No es una máquina virtual**
- Las máquinas virtuales sirven para una cosa, y los contenedores para otra
- La base sobre la que ambos se sustentan es diferente. Veámoslas:

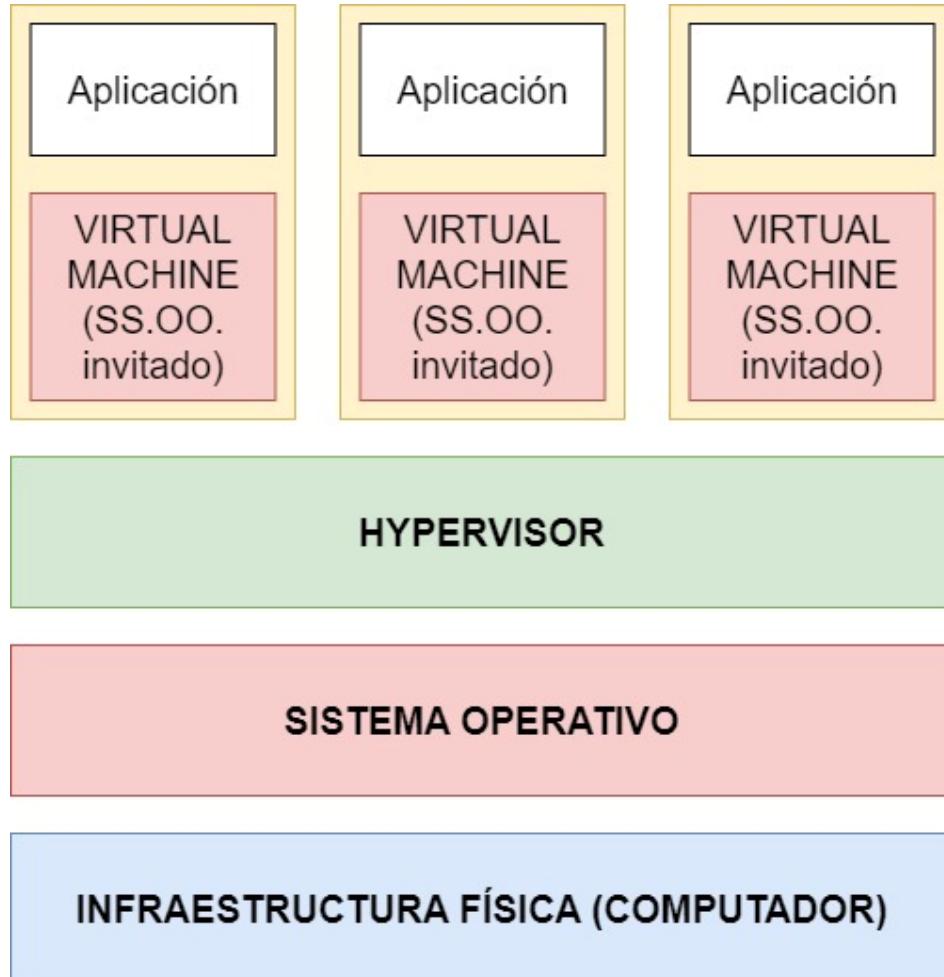
Máquina virtual (de tipo 1: Bare Metal)



- Hypervisor

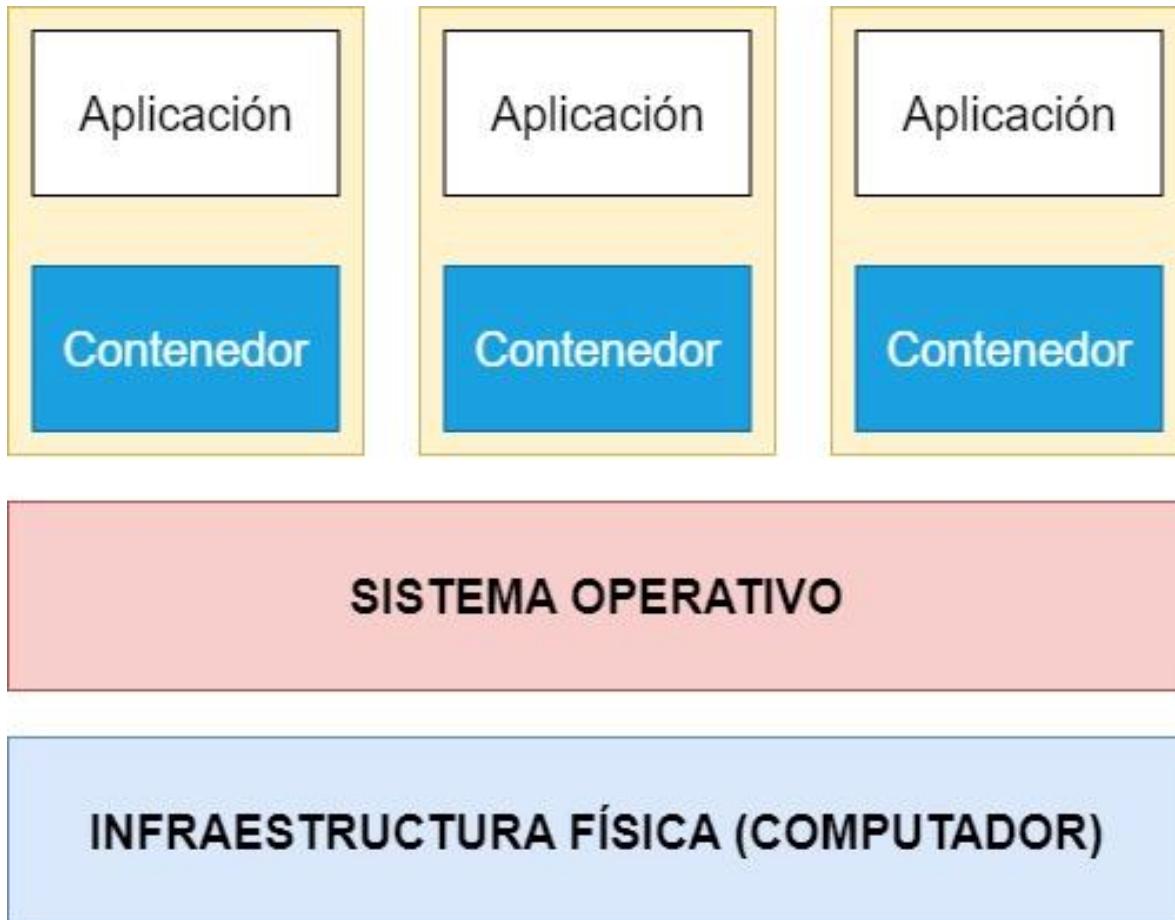
- Software que se ejecuta directamente sobre el hardware (sin necesidad de sistema operativo)
- Obtiene recursos de la computadora (CPU, memoria, almacenamiento, etc.) y los distribuye entre las máquinas virtuales
- Ejemplos: Hyper-V Server (Microsoft), XenServer (Citrix), Oracle VM Server (Oracle)

Máquina virtual (de tipo 2: Hosted)



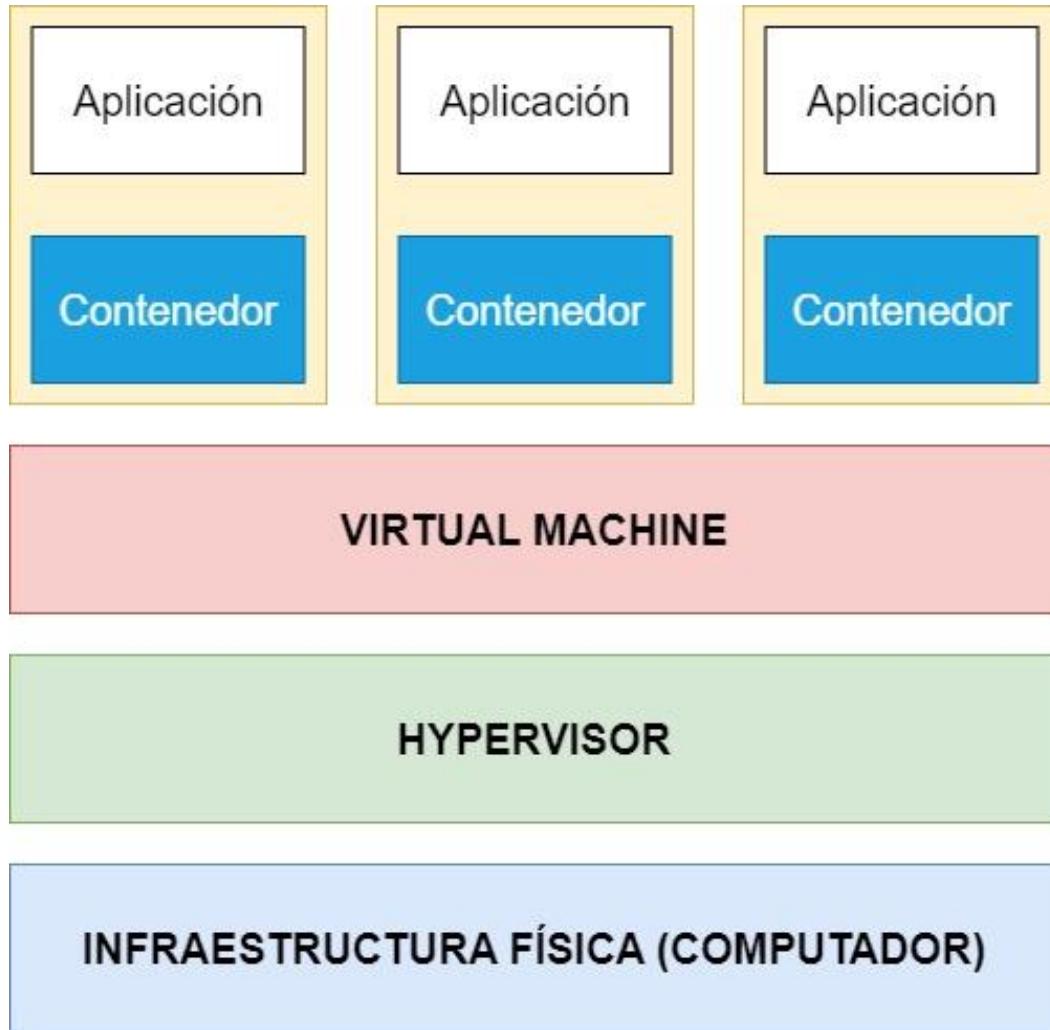
- **Sistema operativo**
 - Es el que llamamos **host** o **anfitrión**
- **Hypervisor**
 - No es como el de Bare Metal
 - Software que corre sobre un sistema operativo anfitrión
 - Depende del sistema operativo para gestionar el hardware sobre el que se ejecuta
 - Ejemplos: VirtualBox, VMWare

Contenedor (habitual en un PC de escritorio)



- **No hay necesidad de un hypervisor**
- Ejemplos: Docker, Podman, Singularity

Contenedor (habitual en la nube)

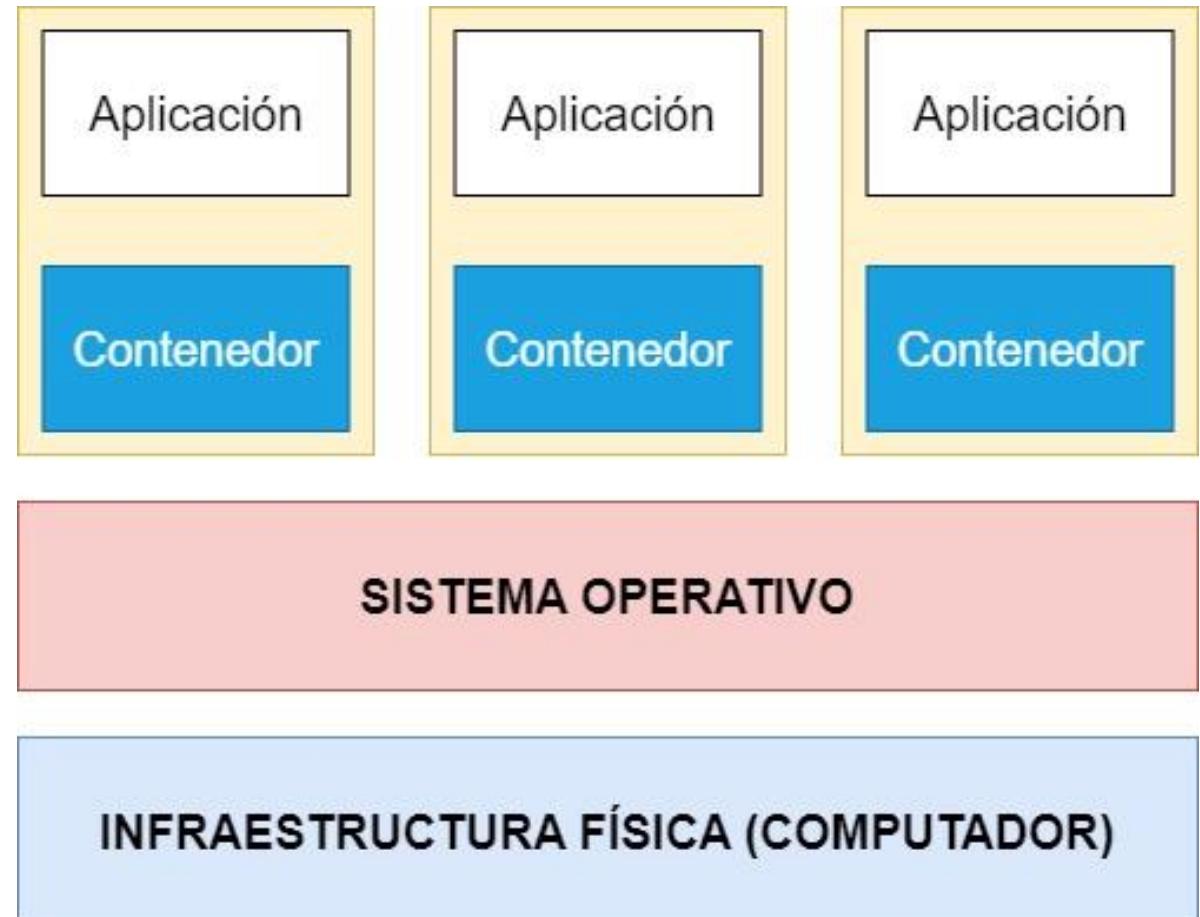


- Una empresa trabaja en la nube de este modo:
 - En la **nube**, las máquinas físicas están ubicadas en un lugar remoto
 - En la **empresa**, el desarrollador *levanta* (sobre esa máquina física remota) una máquina para un cliente: esa máquina es virtual
 - Dentro de esa máquina corren los contenedores

Qué es un contenedor

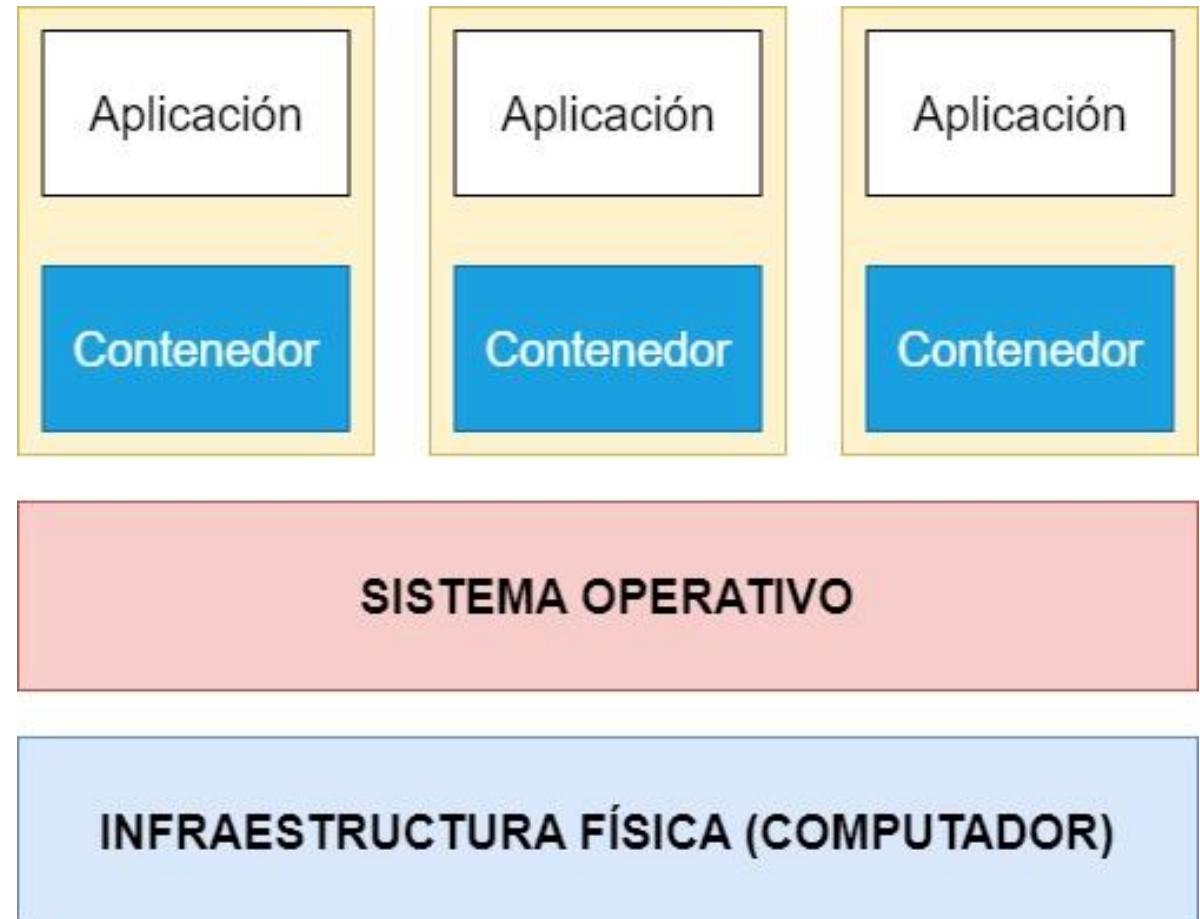
Contenedor

- Es un software independiente que encapsula una aplicación y todas sus dependencias en un entorno aislado
- El sistema operativo (Linux) otorga recursos al contenedor (extrayéndolos del hardware)



Contenedor

- Las aplicaciones no se ejecutan sobre un sistema operativo, sino sobre un contenedor
 - De este modo, la aplicación es independiente del sistema operativo
- Pero el contenedor sí que accede al kernel y a los recursos del sistema operativo en el que corre



Qué es un contenedor

- El objetivo de un contenedor es crear bloques de software que puedan:
 - encajarse en sistemas operativos
 - y ejecutarse sin más (sin instalaciones ni configuraciones)
- Estos bloques de software contienen aplicaciones, dependencias, archivos de configuración, todo preparado para funcionar de inmediato
- Estos bloques de software se llaman **contenedores**
- Estos bloques de software están **aislados**: tienen todo lo que necesitan para funcionar, y no saben nada del mundo exterior
- Los bloques de software pueden verse, simbólicamente, como cajas

Qué es un contenedor

- **Ejemplo 1:**
 - Creamos un **contenedor**, en el que instalamos:
 - Un Linux mínimo (como Alpine)
 - Una base de datos ligera (como MySQL)
 - Configuración de seguridad (usuarios, iptables, archivos de configuración)
 - Llevamos ese **contenedor** a otro ordenador
 - Y basta con ejecutar ese contenedor para tener en ejecución un Linux con MySQL, configurado para seguridad
 - ¡Ojo! No tenemos un Linux con su escritorio y entorno gráfico; lo tenemos por línea de comandos

Qué es un contenedor

- **Ejemplo 2:**
 - Creamos un **contenedor**, en el que instalamos Jupyter Notebook (un software para ciencia de datos):
 - Jupyter Notebook necesita bibliotecas como *numpy*, *pandas*, *matplotlib*, *scikit-learn*
 - Estas son las **dependencias** de Jupyter Notebook
 - Así que instalamos las dependencias
 - Instalamos *numpy*
 - Instalamos *pandas*
 - Instalamos *matplotlib*
 - Instalamos *scikit-learn*
 - Llevamos ese **contenedor** a otro ordenador
 - Y basta con ejecutar el contenedor para tener Jupyter Notebook funcionando

Qué es un contenedor

- Previamente se dijo que un contenedor está aislado del mundo exterior:
 - del resto de contenedores
 - de la máquina (real o virtual) en la que se ejecuta
- No obstante, un contenedor ofrece mecanismos para que un contenedor **se pueda comunicar con otros** contenedores y con la máquina en la que se ejecuta

Qué es un contenedor

- Hay que distinguir el “contenedor” del “sistema en el que se ejecuta”
 - **Contenedor:** la ejecución del propio contenedor
 - **Sistema en el que se ejecuta** (anfitrión o host): ordenador (físico o virtual) con un sistema operativo en el que se ha instalado el contenedor
- Los contenedores no salen de la nada. ¿Quién los prepara?
 - Terceras personas: algunos contenedores preexisten y basta con usarlos
 - Nosotros mismos: haremos el trabajo necesario para crearlos, pero solo se hará una vez