**Virtualización**

**Definición**: En Informática, la virtualización es la creación a través de software de una versión virtual de algún recurso tecnológico, como puede ser una plataforma de hardware, un sistema operativo, un dispositivo de almacenamiento o cualquier otro recurso de red.

Básicamente es simular un hardware o software para poder ejecutar un sistema operativo, aplicación o servicio que no estaba nativamente preparado para ello.

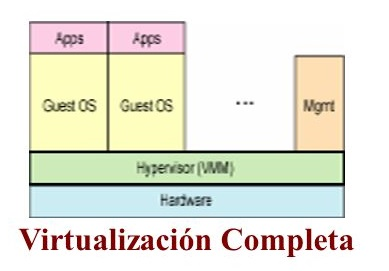
**Tipos de Virtualización**:

* De Plataforma: Consiste en simular una máquina real con todos sus componentes.
* Emulación:
* Completa:
* Paravirtualización:
* Nivel Sistema operativo:
* Nivel Aplicación:
* Nivel Kernel
* De Recursos: Agrupar o dividir recursos para que sean vistos de otra forma.

**Trabajo**

1. Definir y poner un ejemplo de cada tipo de virtualización:

* **De Plataforma:** Consiste en simular una máquina real con todos sus componentes.
  + Amazon Web Services
* **Emulación:** Un emulador es un programa de software que simula la funcionalidad de otro programa o un componente de hardware. Dado que implementa funcionalidad por software, proporciona una gran flexibilidad y la capacidad de recopilar información muy detallada acerca de la ejecución.
  + Game Boy Advance
* **Completa:** Esta es en donde la máquina virtual simula un hardware suficiente para permitir un sistema operativo huésped sin modificar (uno diseñado para la misma CPU) para ejecutar de forma aislada.
  + Oracle VM VirtualBox



* **Paravirtualización:** La paravirtualización es un método que permite que el software que se ejecuta en un sistema virtual omita la interfaz virtual y ejecute operaciones en el hardware real del sistema. En un sistema virtual estándar, el único programa que utiliza el hardware real del sistema es la interfaz virtual. El resto del software se ejecuta totalmente dentro del entorno virtual. Con la paravirtualización, hay formas en que el software incluido puede acceder a recursos reales en lugar de virtuales. Esto acelera ciertas funciones sin sacrificar la potencia informática.



* **Nivel Sistema operativo:** La **Virtualización** de SO mejora el rendimiento, gestión y eficiencia. En la base reside un sistema operativo **anfitrión** estándar, como en el caso de **Parallels Virtuozzo** que incluye **Windows** y un sistema con núcleo **Linux**. A continuación encontramos la capa de virtualización, con un sistema de archivos propietario y una capa de abstracción de servicio de *kernel* que garantiza el aislamiento y seguridad de los recursos entre distintos contenedores. La capa de virtualización hace que cada uno de los contenedores aparezca como servidor autónomo. Finalmente, el contenedor aloja la aplicación o carga de trabajo.

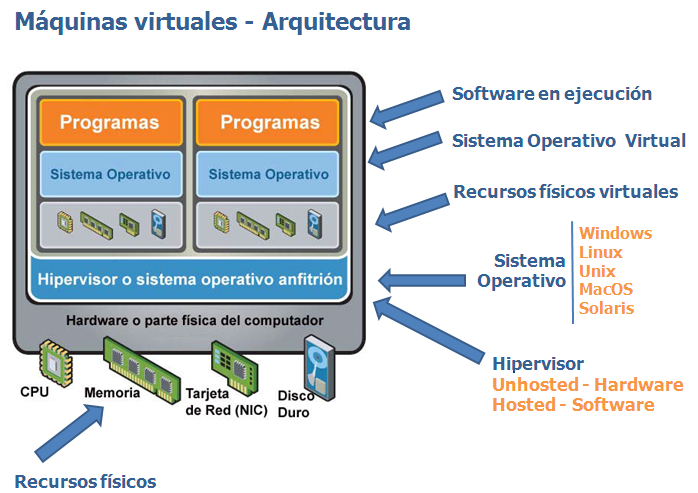


* **Nivel Aplicación:** La virtualización de las aplicaciones consiste en aislar la componente "lógica de la aplicación" del componente sistema operativo.

El objetivo de esta virtualización es conseguir que las aplicaciones puedan funcionar con independencia de las características concretas del entorno en que se ejecutan. De este modo se eliminan los problemas de compatibilidad entre aplicaciones, o entre componentes de aplicaciones, y otros errores típicos de la ejecución concurrente de aplicaciones.

* **Nivel Kernel:** Este tipo de virtualización convierte el Kernel de Linux en el Hypervisor , permitiendo ejecutar máquinas virtuales con independencia del sistema operativo sin modificar, que se almacena en una imagen de disco. Cada máquina virtual tiene su propio hardware virtualizado: tarjetas de red, discos duros, gráfica, etc. Aunque también se puede hacer passthrough de dispositivos.
* **De Recursos:** Agrupar o dividir recursos para que sean vistos de otra forma.

2. Definir las siguientes palabras/conceptos:



• **Máquina virtual:** es un software que emula un ordenador. Es como tener un ordenador dentro de otro pero funcionando de forma virtual, simulada.

Máquina real → Sistema anfitrión, host

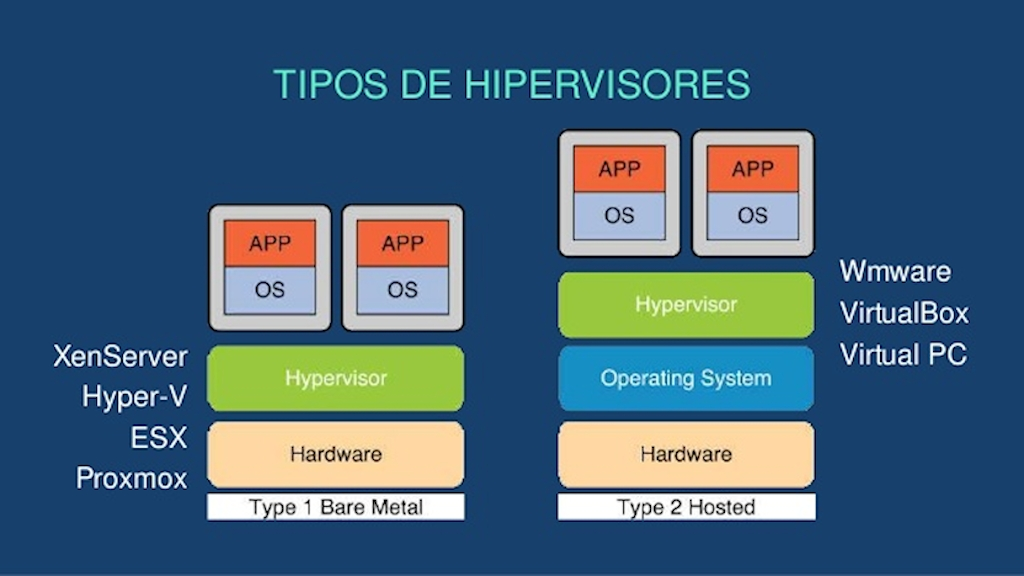
Máquina virtual → Huésped, guest

En el huésped se puede instalar cualquier S.O.

• **Anfitrión:** mi maquina real

**• Hipervisor:** capa de abstracción entre el hardware real de la máquina física y la máquina virtual (con su hardware y su software virtualizado). Es común para todas las MVs

* **Tipo 1:** El hypervisor o monitor corre directamente sobre nuestro hardware y nos permite crear máquinas virtuales. Desaparece la necesidad de contar con un sistema operativo anfitrión, solo tendremos sistemas huéspedes, y el anfitrión será directamente nuestro hipervisor.
* **Tipo 2:** el hipervisor se monta sobre el sistema operativo anfitrión, y este monitor crea los sistemas operativos invitados.



• Cloud: Virtualización en sistemas remotos, con todas la ventajas del cloud computing (Azure, AWS, GoogleCloud..)



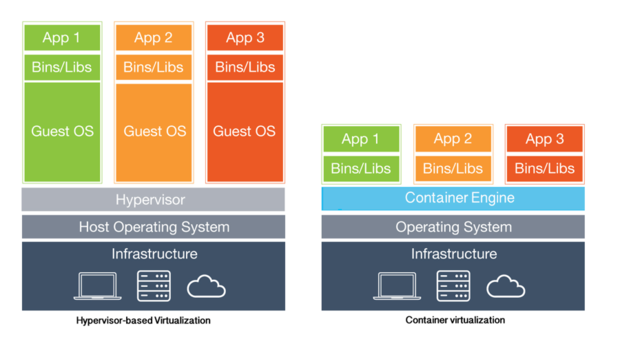
**• IaaS (Infraestructura como servicio):** Lo que se ofrece en la nube son capacidades de cómputo (máquinas) y almacenamiento, que sustituyen a los servidores físicos tradicionales de las empresas. Amazon Elastic Compute Cloud (EC2) y Simple Storage Service (S3).

**• PaaS (Plataforma como servicio):** Lo que se ofrece en la nube es la posibilidad de desarrollar o desplegar aplicaciones. Con este esquema cambia la forma tradicional de desarrollar software con los recursos propios. Google App Engine, Windows Azure o Red Hat OpenShift.

• **SaaS (Software como servicio):** Lo que se ofrece en la nube es software. No es necesario que el usuario instale ningún programa en su equipo, simplemente accede a un recurso web remoto y utiliza el software allí instalado y configurado. Google Apps, Salesforce, Microsoft Office 365, etc. son algunos de los SaaS más utilizados y conocidos.

**• Contenedor:** son el siguiente paso en la evolución de la virtualización de sistemas operativos, su objetivo principal es OPTIMIZAR el uso de los recursos de la máquina anfitrión(host). Se puede entender como una virtualización a nivel de sistema operativo. Se evita la sobrecarga asociada con tener a cada huésped ejecutando un sistema operativo completamente instalado. Una desventaja de la virtualización basada en contenedores, sin embargo, es que cada invitado debe utilizar el mismo sistema operativo que utiliza el host, además de tener un menor nivel de aislamiento.

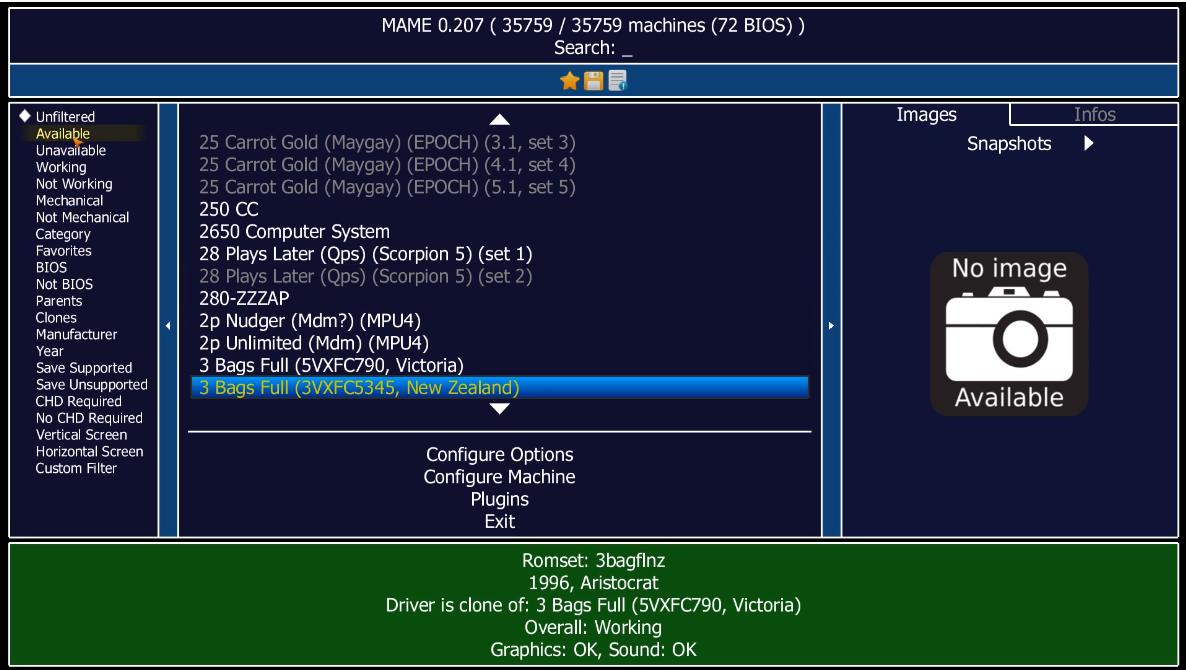
Docker.

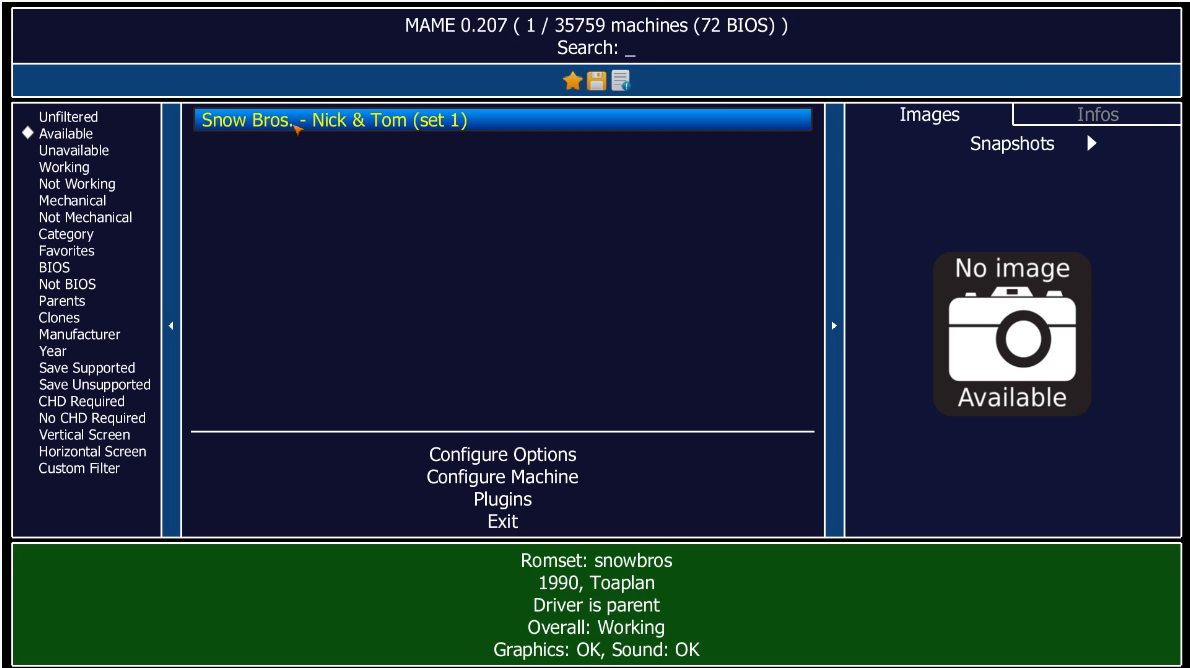


3. Encuentra, instala y prueba un programa de emulación/virtualización para los siguientes casos:

• Máquina recreativa: Mame:

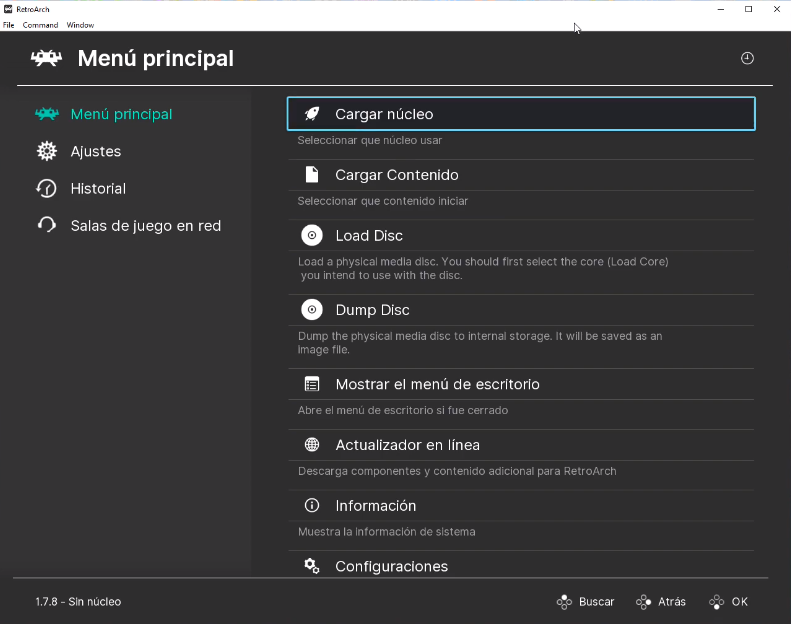
Ejecutamos el archivo de instalación se crean todas las carpetas y el ejecutable, y descargamos una ROM para probar algún juego, y la metemos en la carpeta roms

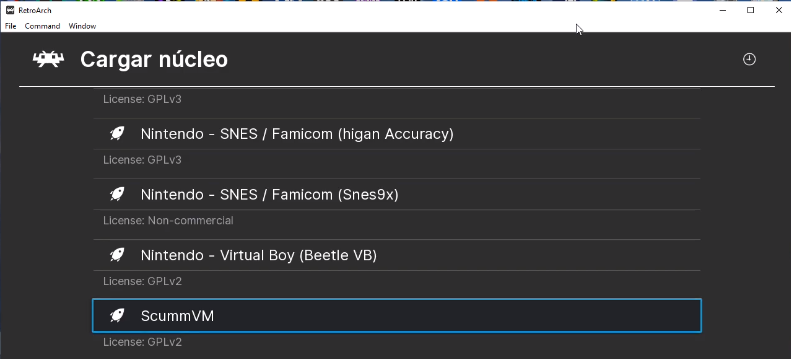




<https://tecnowindows.com/mejores-emuladores-de-juegos-de-recreativa-para-windows/>

• Juegos MS-DOS scumm

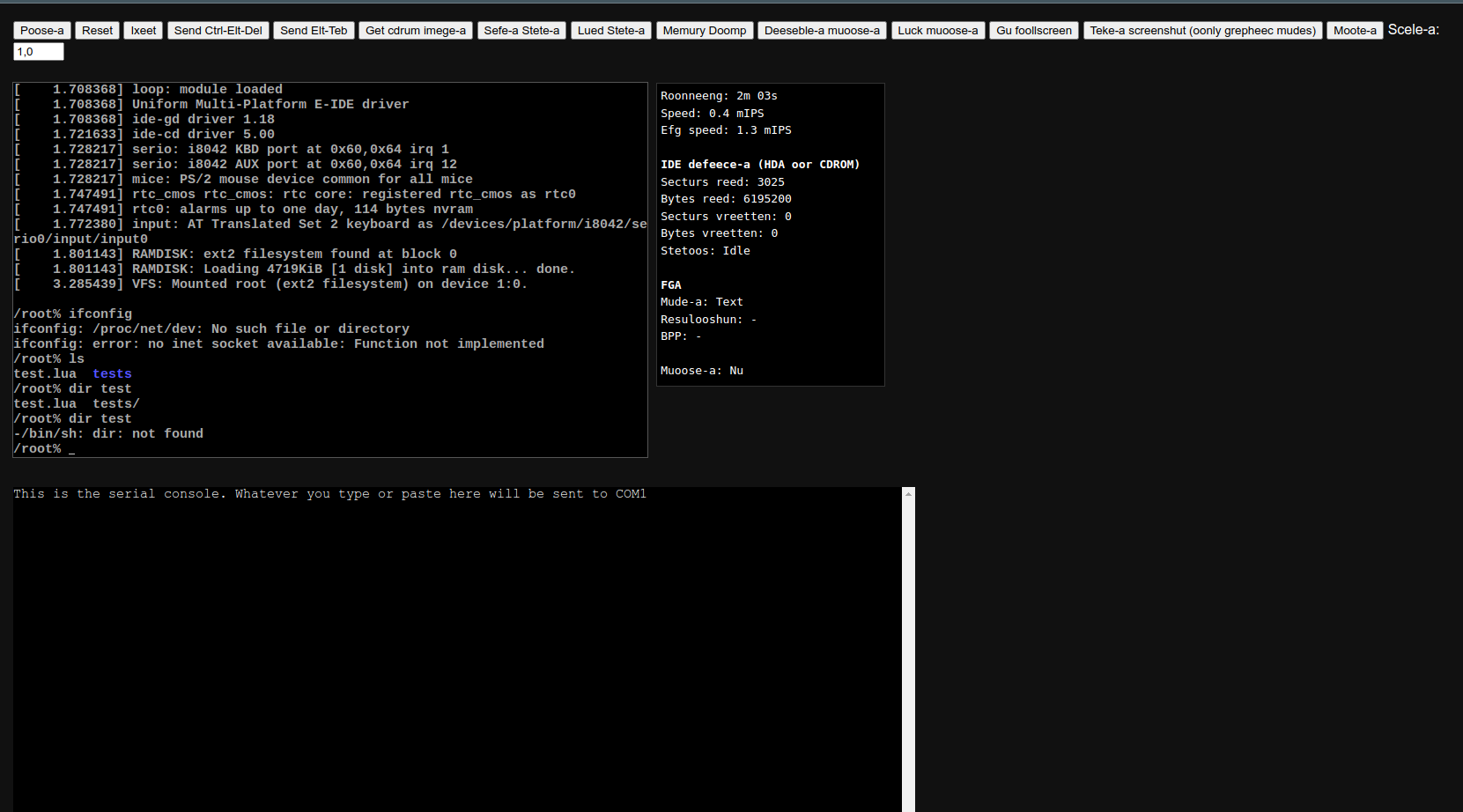




Después de cargar el núcleo que solo se hace una vez en retroarch, hay distintas configuraciones en scumm, utilizar la deseada. Y agregas juegos a la carpeta.



• Terminal de Linux: <https://copy.sh/v86/?profile=linux26>



• Sistema Android: descargamos e instalamos

