

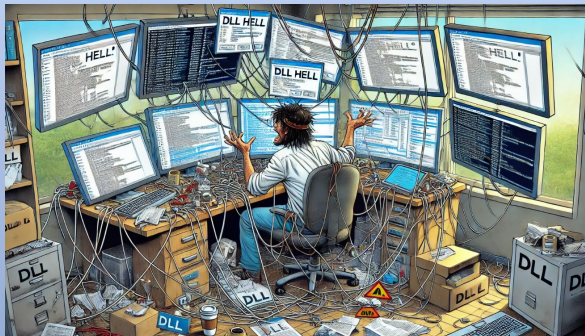
Lección 1:

Entendiendo la contenerización de aplicaciones

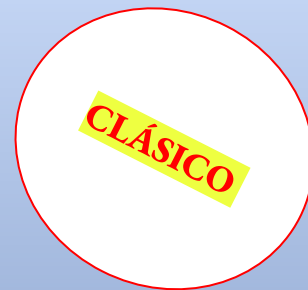
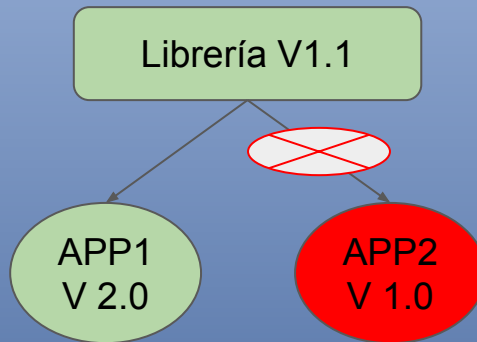
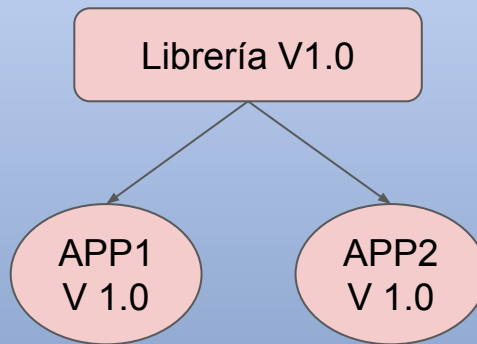
Retrocedamos en el tiempo



Falta de aislamiento ...



Actualización



Falta de uniformidad



Diferente para cada aplicación

- Instalación
- Actualización
- Eliminación

Difíciles de escalar y mantener



CLÁSICO

Trabajas en un centro de datos

- Hay que instalar la aplicación en 500 servidores?
- Que pasa si falla en alguno?
- Mantenimiento?

LA SOLUCIÓN....



Nace en el entorno **linux**,

Progresivamente se fueron inventando "cosas" para :

- **aislar** la aplicación y así evitar que el estado del sistema en el que se instala afecte a la aplicación.
- permitir que cualquier aplicación se **instale y configure de la misma manera**
- **Facilitar la gestión y despliegue** de cientos de aplicaciones e instancias

Y se llama.....

El contenedor:



Un contenedor informático es una unidad de software que empaqueta el código y todas sus dependencias necesarias para que una aplicación se ejecute de manera consistente en cualquier entorno.

Utiliza recursos del sistema operativo anfitrión de manera aislada, proporcionando una forma eficiente y portable de desplegar aplicaciones.

Comparativa con la virtualización de servidores



Es lo mismo?

Ni remotamente parecido, porque la MV quiere simular ser una máquina real .. y hereda los problemas que hemos comentado antes .. además de eso:

- recursos retenidos para la máquina virtual
- el contenedor es muchísimo más rápido arrancando y parando
- Cuando se ejecután muchos contenedores en la misma máquina, se aprovechan mejor los recursos de la máquina



Prestaciones de los contenedores



Hay sobrecarga respecto a la misma aplicación sin contenedor?

SI

Pero es insignificante.

Quando es conveniente usar contenedores?

Probablemente ... SIEMPRE....

- Porque nos ofrece “GRATIS”:

Aislamiento, uniformidad, aprovechamiento óptimo de recursos, y escalabilidad

Cuando no difícil encontrar estas situaciones, pero por dar algún ejemplo:

- Cuando el contenedor introduzca una “sobrecarga” de trabajo...por ejemplo en fases muy tempranas de desarrollo de aplicaciones.
- Entornos monolíticos sin planes de escalabilidad y con requerimientos de seguridad muy, muy, muy altos.

