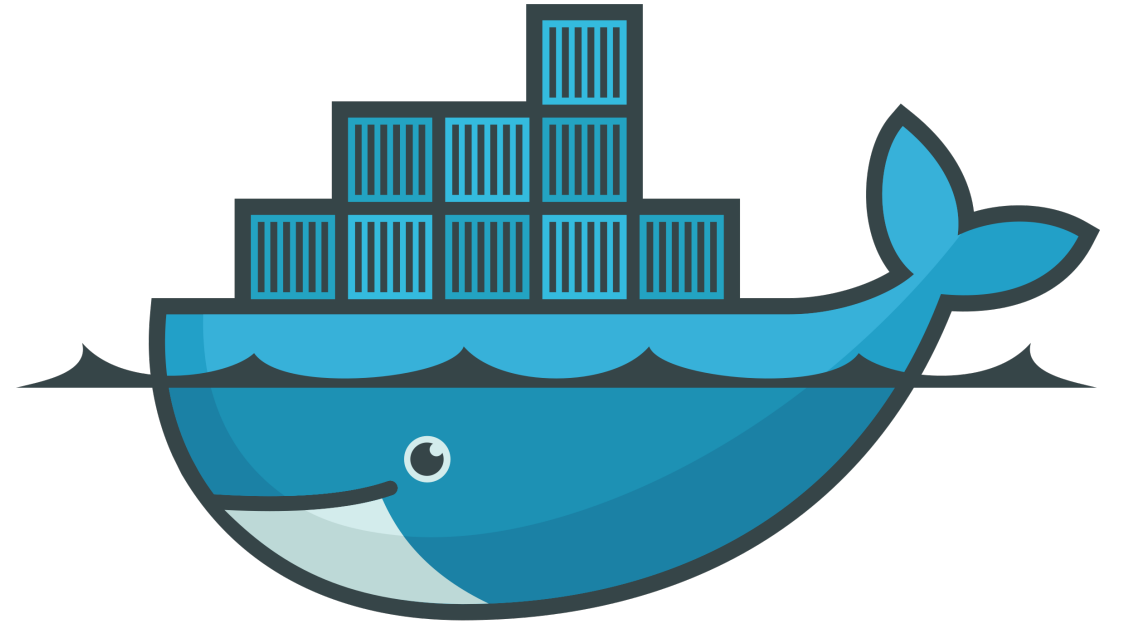


# Docker 101 – Módulo 1

GBM Tech Academy

# Agenda

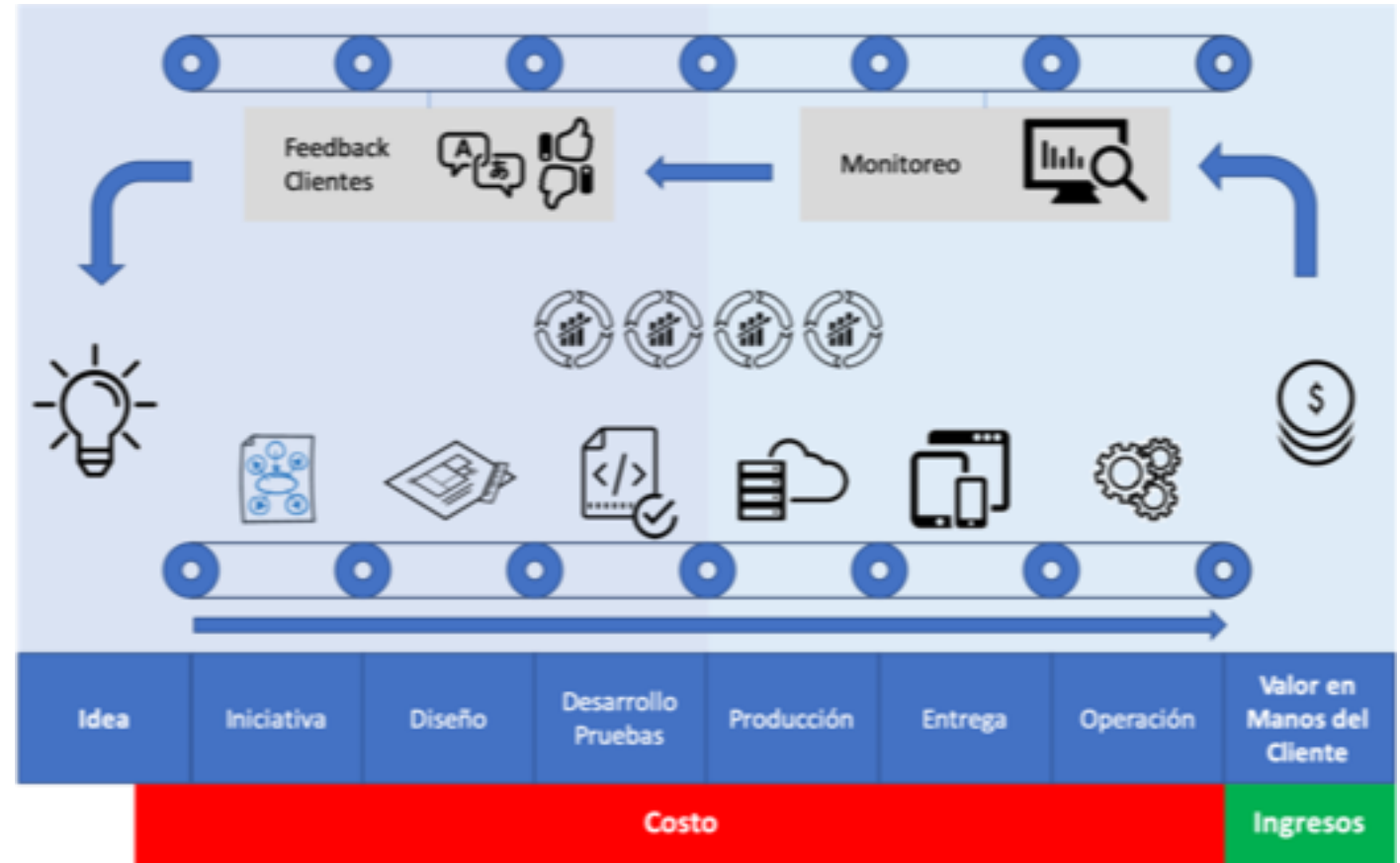
- ¿Qué es Docker y para qué se usa?
- Introducción a GitHub
- Instalación de Docker
- Dinámica del Curso
- Asignación



docker

# El reto de todo negocio

- System Thinking
- Feedback Loops
- Cultura de Experimentación y Mejora Continua



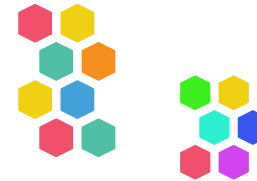
# Evolución del modelo de desarrollo y gestión

Cientes  
Pesados



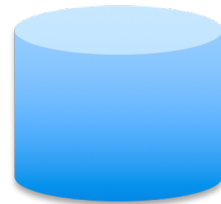
Cientes  
livianos

Stacks  
definidos



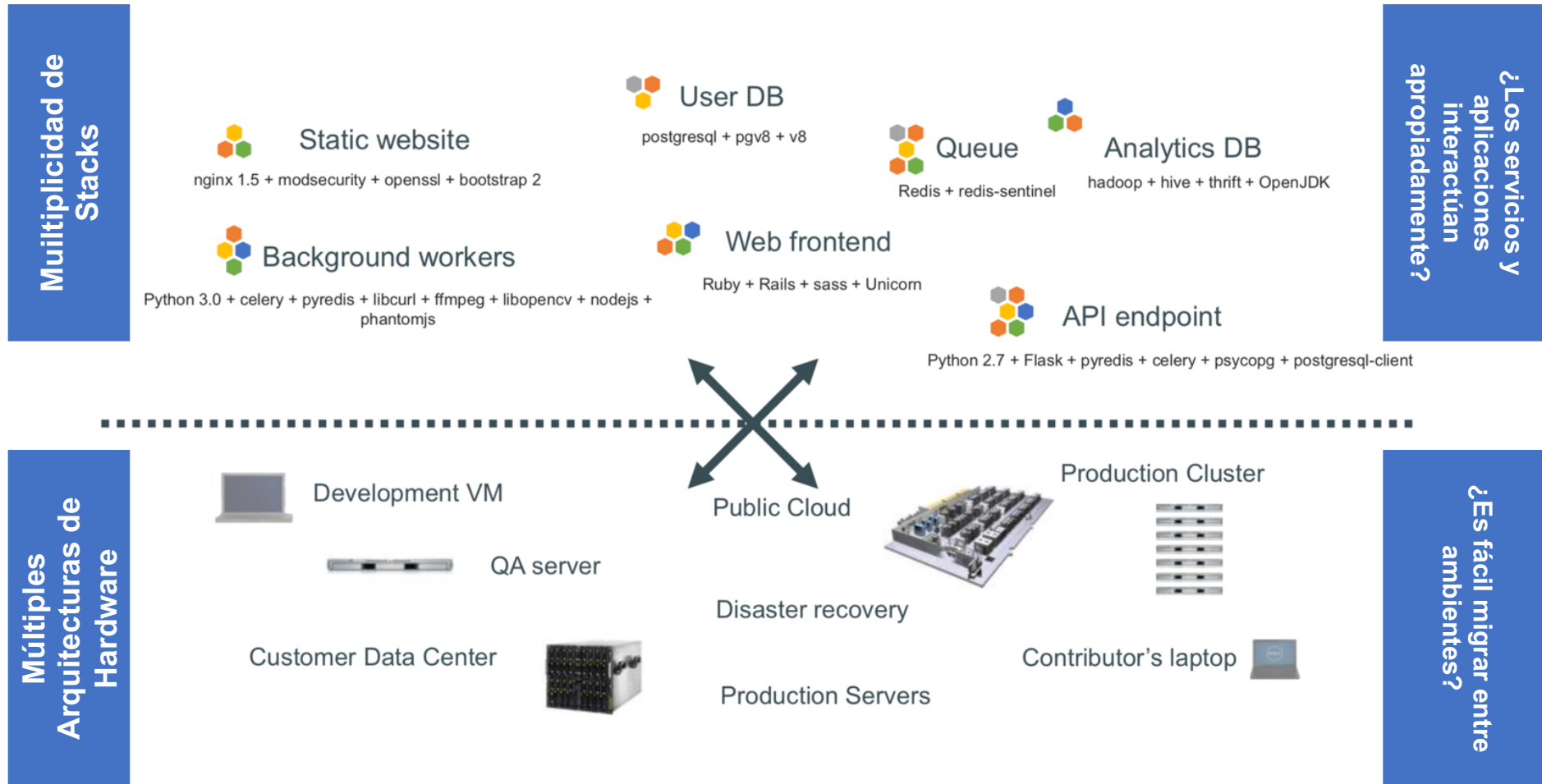
Ensamblados bajo  
demanda por  
desarrolladores  
utilizando los mejores  
componentes

Infraestructura  
Monolítica

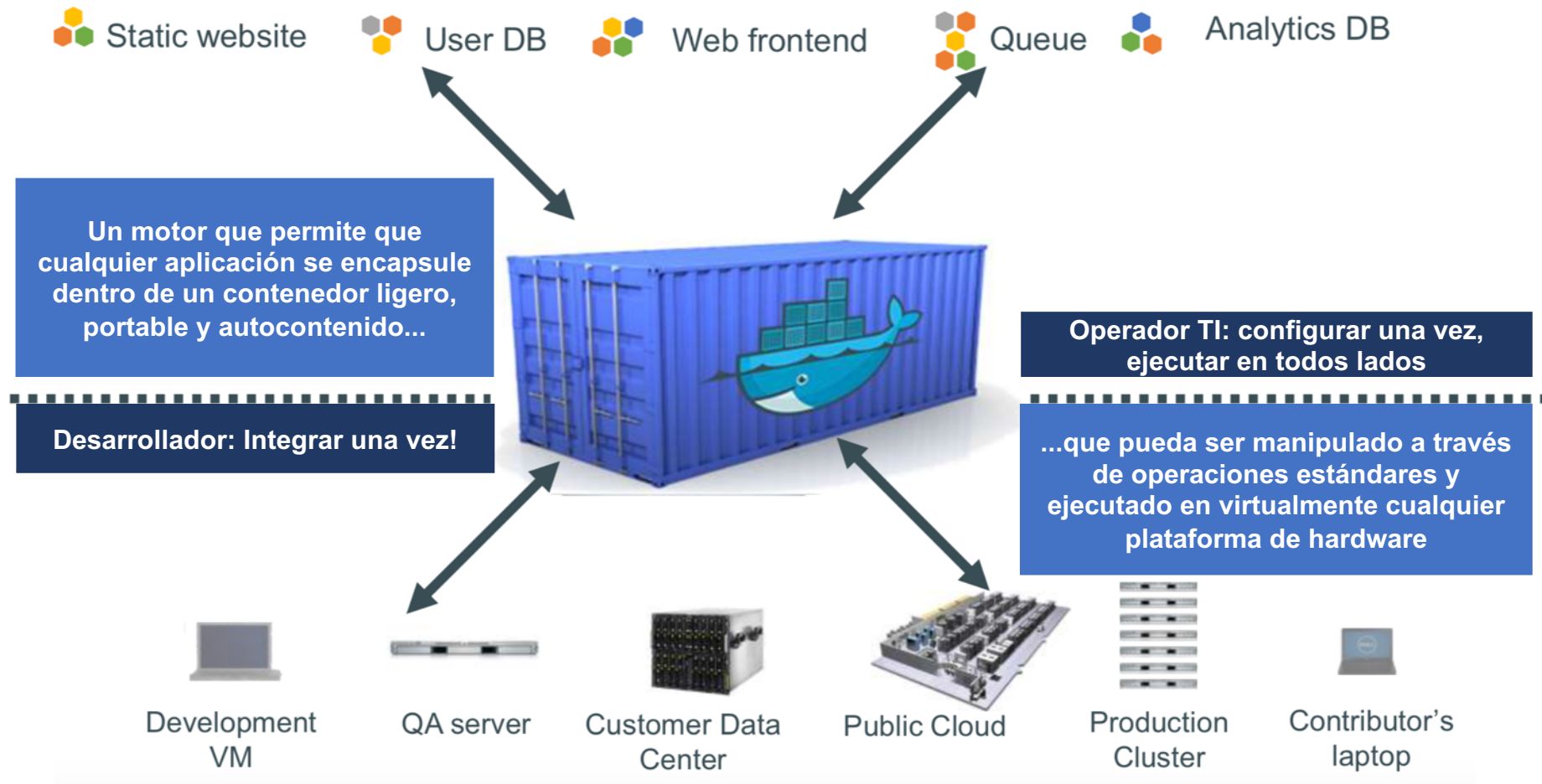


Ejecutándose en  
ambientes  
heterogéneos

# Pesadilla de Compatibilidad

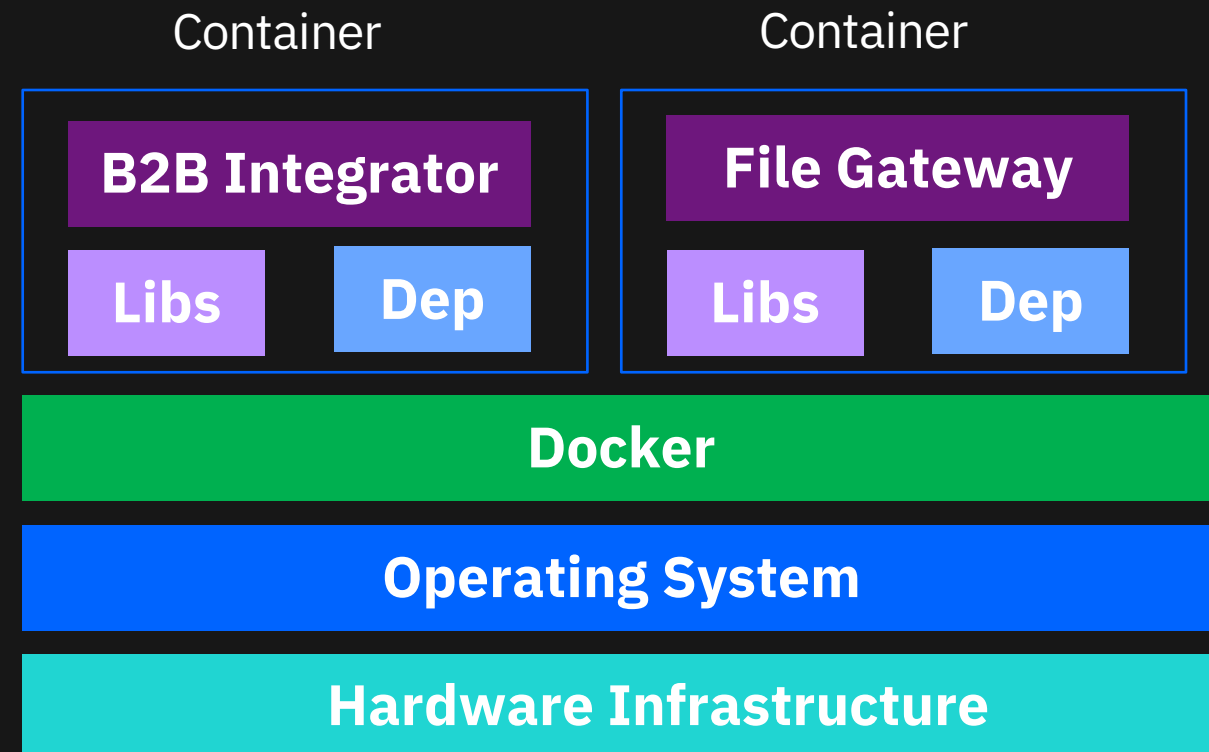
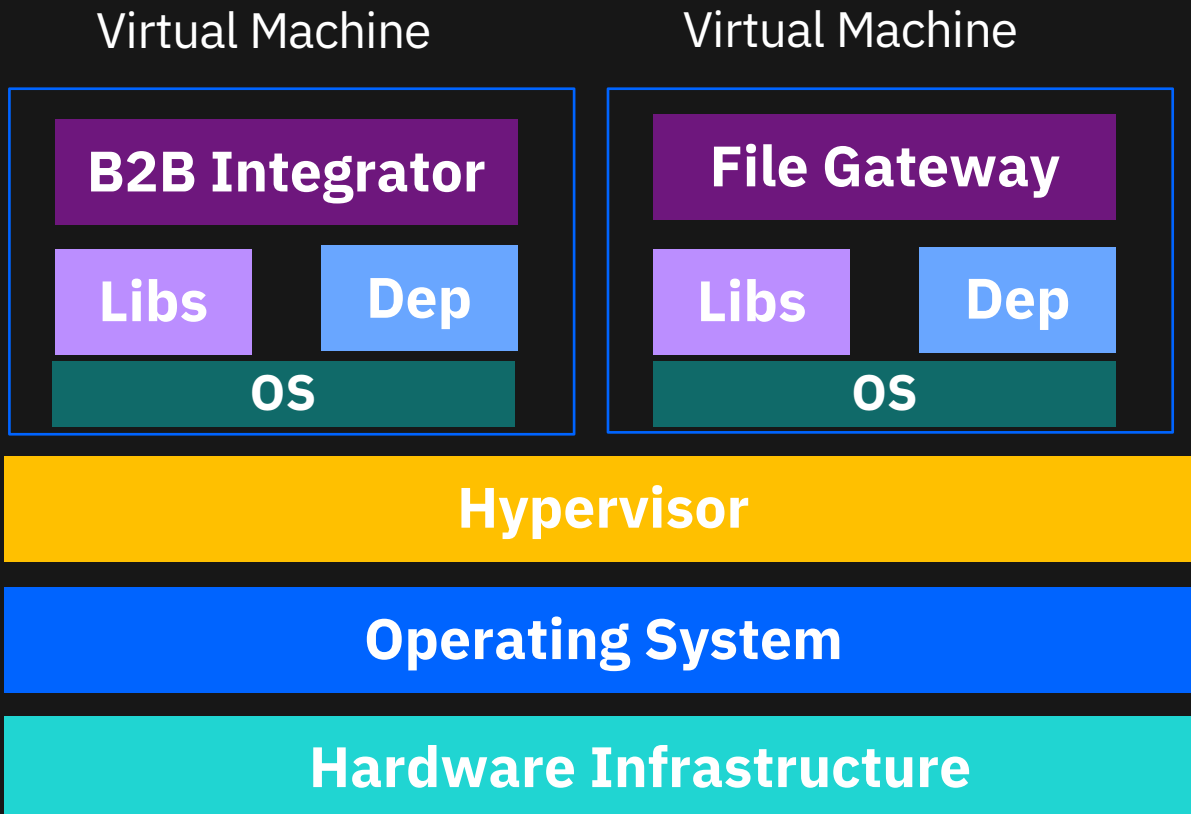
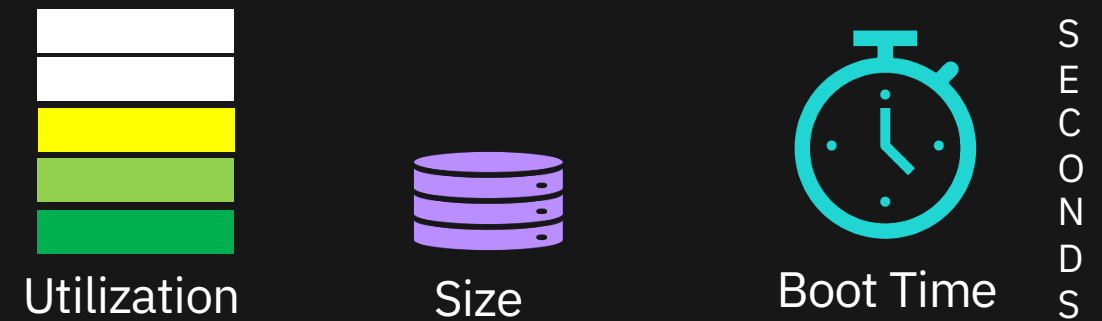
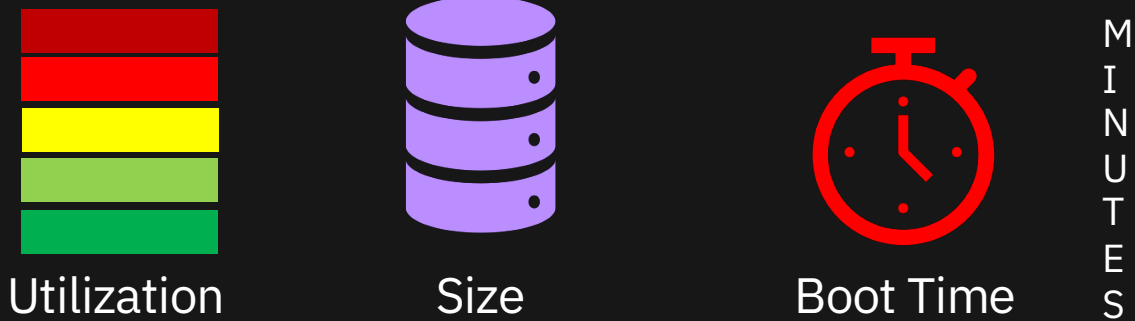


# Contenedores Docker



# Virtual Machines vs Containers

IBM



# Demo Contenedores

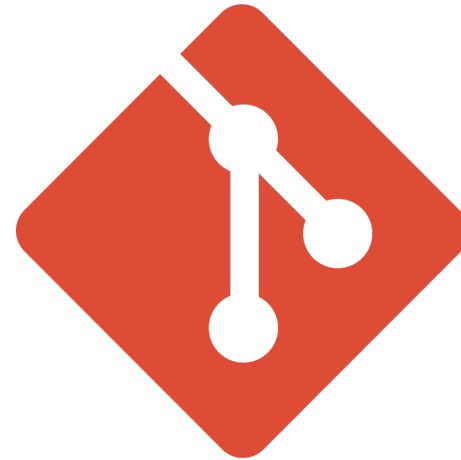
```
docker run -p "27017:27017" mongo
```



# Introducción a Git y Github

# Version Control System (VCS) para hacer seguimiento a cambios

- Control de Versiones Distribuida
- Coordinación de múltiples desarrolladores
- ¿Quién hizo qué y cuándo?
- Revertir cambios
- Repositorios Locales y Remotos



**git**

# Arquitectura Conceptual Git

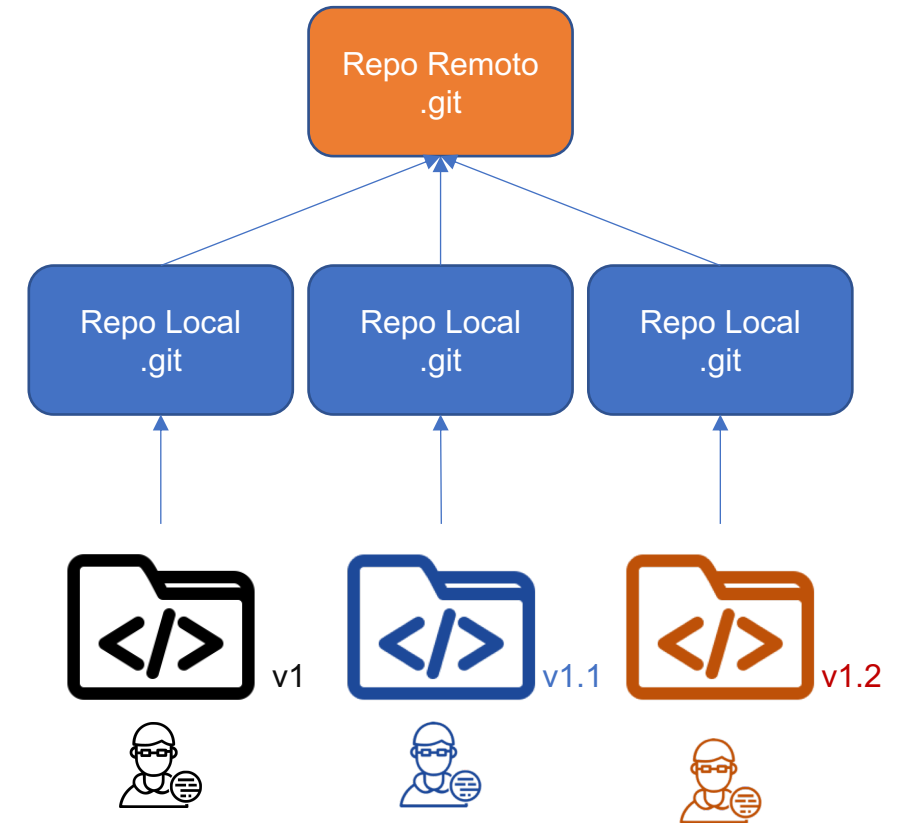
- Version de Código

3aa387b639e0e696e5d6001d78c957ce6729b3d4

- Carpeta Local

- Repositorio Local

- Repositorio Remoto



# Clone, Pull, Commit & Push

Comando	
> git init	Inicializa un repositorio en una carpeta
> git add .	Añade archivos al area stage del control de código
> git commit -m "First commit"	Hace "commit" de los archivos agregados y modificados
> git remote add origin <i>remote repository URL</i>	Agrega un repositorio remote al repositorio local
> git push -u origin master	Sube los archivos "committed" a un repositorio remoto
> git status	Muestra el estatus de la carpeta de código
> git clone <i>remote repository URL</i>	Descarga un repositorio remoto en un nuevo repositorio local

# Demo GitHub

# Dinámica del Curso

# Módulos del Curso

No.	Módulo	Duración	Fecha
1	Introducción	2 horas	Viernes 28 de Junio
2	Ejecución de contenedores Docker	4 horas	Jueves 04 de Julio
3	Creación de imágenes	4 horas	Viernes 05 de Julio
4	Aplicaciones en contenedores	4 horas	Jueves 11 de Julio
5	Arquitecturas multi-contenedores	4 horas	Viernes 12 de Julio
6	Integración continua con Docker	4 horas	Jueves 18 de Julio
7	Orquestación de Contenedores	4 horas	Viernes 19 de Julio

# Recursos Disponibles

Sitio en GitHub (<https://github.com/garciasebastian/docker-101>)

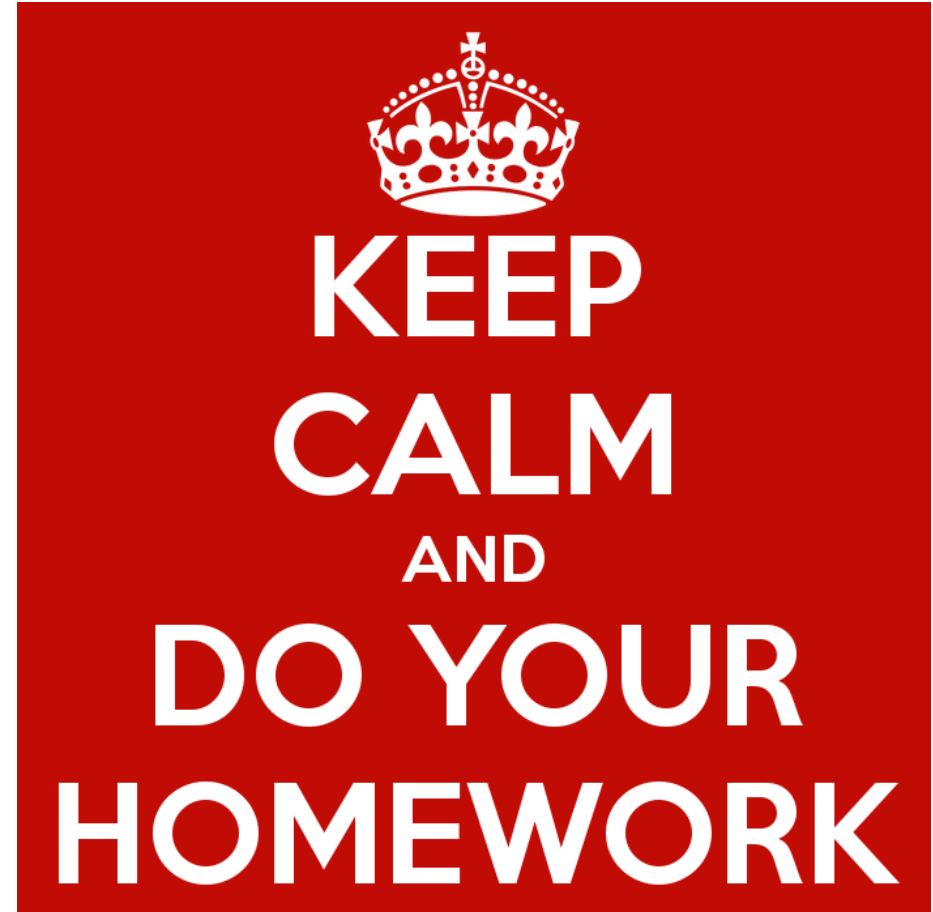
- Contenido del curso
- Pre-requisitos
- Presentaciones
- Asignaciones semanales
- Enlaces a recursos externos





# Asignaciones y Evaluación

Todos los viernes se enviará una asignación relacionada con el contenido visto durante la semana que debe completarse antes de la siguiente clase del jueves.



# Asignación Semana 1

- Crear usuario de DockerHub
- Instalar Docker Community Edition
- Crear usuario de GitHub
- Crear un repositorio en Github
- Clonar el repositorio localmente
- Ejecutar el comando de comprobación
- Sincronizar git
- Enviar por correo el url del repositorio y el SHA del commit