

# Git y Docker: Herramientas necesarias en el desarrollo de software actual

# Presentación

**ANDROMEDA**  
latam



# Propósito de esta charla

¿Ya utilizan o conocen GIT?

Sistema distribuido de control de versiones de archivos

## Flujo de trabajo con los branches

- master o main
- development
- release
- feature
- hotfix

# GIT

## Comandos esenciales

- `git init`
- `git add`
- `git commit`
- `git branch`
- `git merge`

# GIT

## Comando para GitHub

- `git clone`
- `git remote`
- `git push`
- `git pull`



# GIT

## Documentación

- Web Oficial: <https://git-scm.com/docs>
- Desde la terminal:
  - `git help`
  - `git <command> --help`
  - *git pull --help*

# GIT

## Clientes

- Por línea de comandos: [\*https://git-scm.com/doc\*](https://git-scm.com/doc)
- GitKraken: [\*https://www.gitkraken.com\*](https://www.gitkraken.com)
- SourceTree: [\*https://www.sourcetreeapp.com\*](https://www.sourcetreeapp.com)
- Otros: [\*https://git-scm.com/downloads/guis\*](https://git-scm.com/downloads/guis)
- IDE: *VSCode, Eclipse, Idea*

## Conclusión

Nos ofrece seguridad de poder realizar cambios, investigando alternativas, que luego fácilmente podemos integrar o descartar.

# Docker

Es un empaquetador de software ligero.

O lo que también podemos llamar,  
una maquina virtual ligera.

# Docker

## Conceptos fundamentales

- Imágenes (Images)
- Contenedores (Containers)

# Docker

## Imágenes

Son las plantillas base, que contienen el entorno necesario para lo que van a ejecutar.

# Docker

## Imágenes

- Se construyen en capas
- Son inmutables
- Se pueden reutilizar para nuevas imágenes
- No se ejecutan, solo ocupan espacio en disco

# Docker

## Contenedores

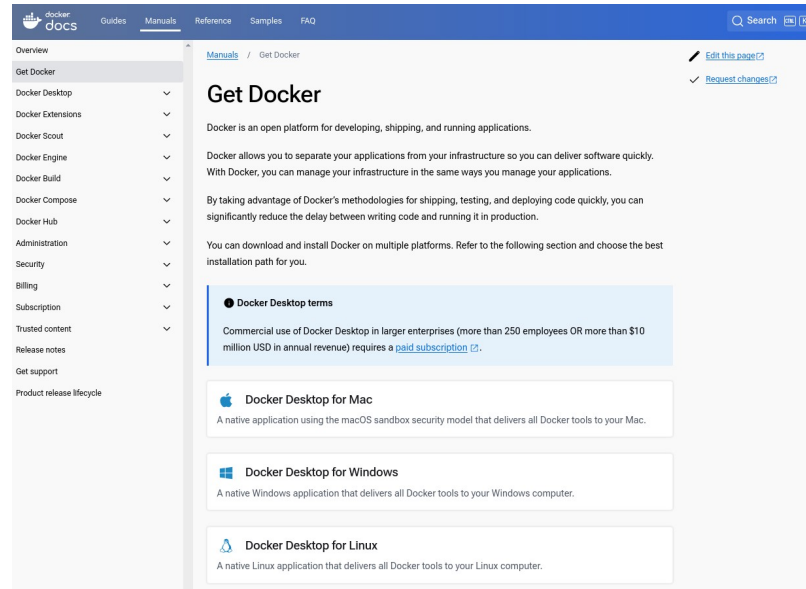
- Se crean a partir de una imagen, más una configuración inicial de lanzamiento
- Son instancias ejecutables una imagen



# Docker

## Instalación

<https://docs.docker.com/get-docker>



# Docker

## Gestión de Imágenes - Comandos esenciales

- `docker image ls`
- `docker image rm`
- `docker image build`

# Docker

## Gestión de Imágenes - Comandos esenciales

- `docker image ls` → `docker images`
- `docker image rm` → `docker rmi`
- `docker image build` → `docker build`

# Docker

## Gestión de Contenedores - Comandos esenciales

- `docker container ls`
- `docker container rm`
- `docker container run`
- `docker container start`
- `docker container stop`

# Docker

## Gestión de Contenedores - Comandos esenciales

- `docker container ls` → `docker ps`
- `docker container rm` → `docker rm`
- `docker container run` → `docker run`
- `docker container start` → `docker start`
- `docker container stop` → `docker stop`

# Docker

## Otros comandos adicionales

- `docker container cp` → `docker cp`
- `docker container attach` → `docker attach`
- `docker container exec` → `docker exec`
- `docker container logs` → `docker logs`
- `docker container commit` → `docker commit`
- `docker image push` → `docker push`
- `docker image pull` → `docker pull`

# Docker

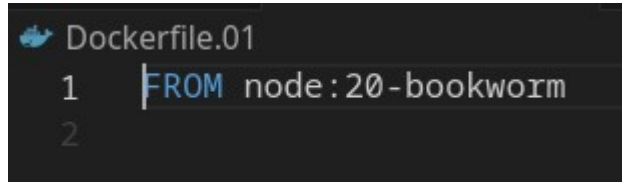
## Documentación

- Web Oficial: <https://docs.docker.com/reference/>
- Desde la terminal:
  - `docker help`
  - `docker <command> --help`
  - *`docker container start --help`*

# Docker

## Creación de una Imagen

- Dockerfile



```
Dockerfile.01
1 FROM node:20-bookworm
2
```

- Documentación:

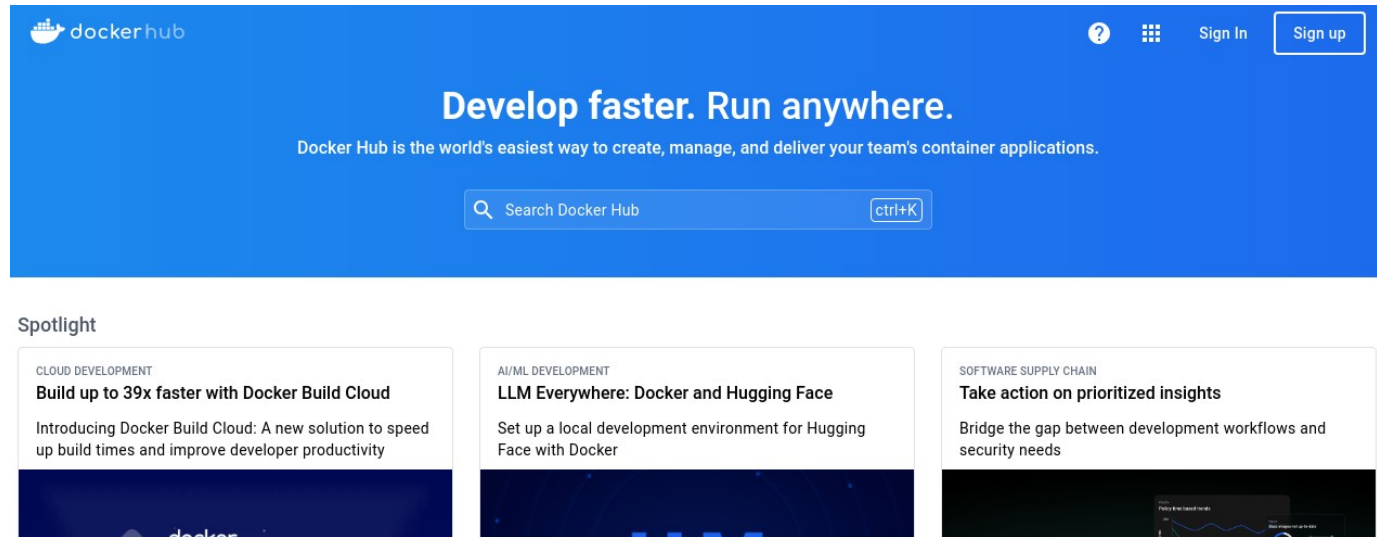
<https://docs.docker.com/reference/dockerfile/>



# Docker

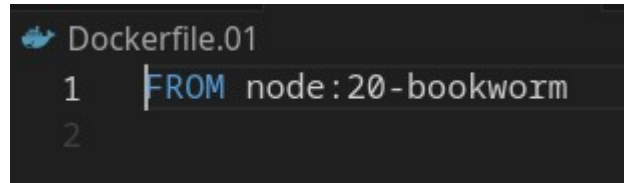
## Docker Hub

<https://hub.docker.com>



# Docker

## Creación de una Imagen

A screenshot of a text editor showing a Dockerfile named 'Dockerfile.01'. The file contains two lines: '1 FROM node:20-bookworm' and '2'. The word 'FROM' is highlighted in blue.

```
Dockerfile.01
1 FROM node:20-bookworm
2
```

```
docker build -t ejemplo01 \
  --force-rm \
  -f Dockerfile.01 .
```

## Ejemplo Práctico

<https://github.com/danieldaf/docker-scripts>

# Propuesta de Práctica

Crear un contenedor de docker  
que sirva como entorno de desarrollo  
para un proyecto sobre la tecnología  
que les sea de interes

<https://github.com/danieldaf/docker-scripts>

# Propuesta de Práctica

## Pasos a seguir

- Instalar docker
- Definir la tecnología (angular, react, java, c)
- Buscar en DockerHub una imagen de referencia
- Escribir el Dockerfile para armar la imagen

# Propuesta de Práctica

## Pasos a seguir

Les propongo que suban a GitHub los archivos de creación de su proyecto, con un proyecto de ejemplo de la tecnología

mail de contacto: *danieldaf2000@gmail.com*

# Propuesta de Práctica

## Imágenes de Base

- Node: *[https://hub.docker.com/\\_/node](https://hub.docker.com/_/node)*
- Python: *[https://hub.docker.com/\\_/python](https://hub.docker.com/_/python)*
- Java: *[https://hub.docker.com/\\_/debian](https://hub.docker.com/_/debian)*  
*[https://hub.docker.com/\\_/ubuntu](https://hub.docker.com/_/ubuntu)*

Muchas Gracias