

## Dia 2: Crear imatges

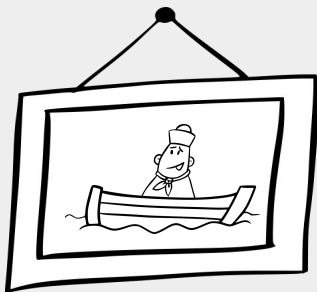
Xavier Sala Pujolar



Universitat  
de Girona

Gener 2021

# Imatges



Les Imatges són fitxers estatics que defineixen el sistema de fitxers i el comportament d'un contenidor concret.

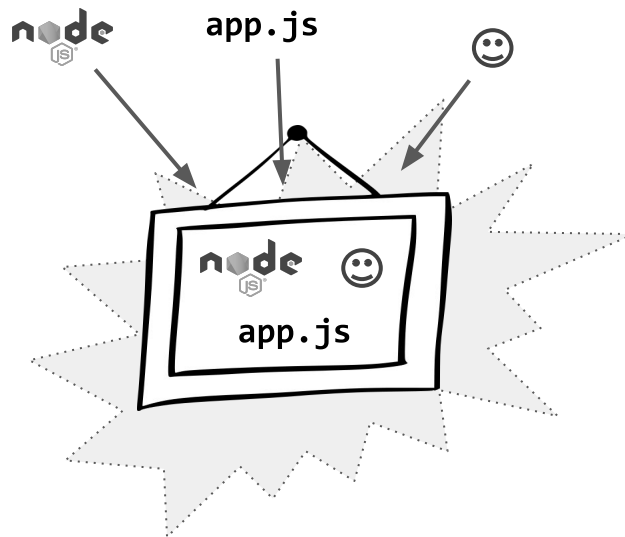
Es fan servir com a plantilles per crear contenidors

# Construir imatges

Les imatges es poden crear de zero o partint d'una imatge base.

El més habitual és crear-les

- des d'una imatge base
- des de contenidors

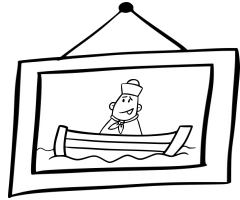




P1: Base de dades amb dades

# Creació a partir de contenidors

---

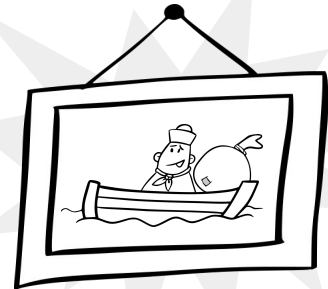


23456789

Hem afegit  
dades al  
contenedor

```
$ docker commit 23456789 marinerdata:v1
```

Copia l'estructura del  
contenedor



# Empaquetar dades amb SQLServer

---

```
$ docker run  
-e 'ACCEPT_EULA=Y'  
-e 'SA_PASSWORD=corona2020!'  
-p 1433:1433 -d  
mcr.microsoft.com/mssql/server:2017-latest
```



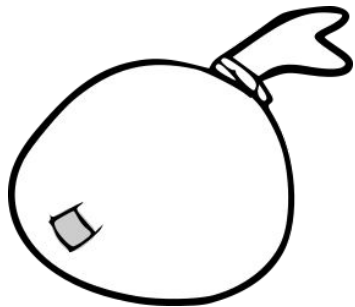
```
$ docker commit 23ba34f utrescu/data:latest
```



P2: Compartir la imatge

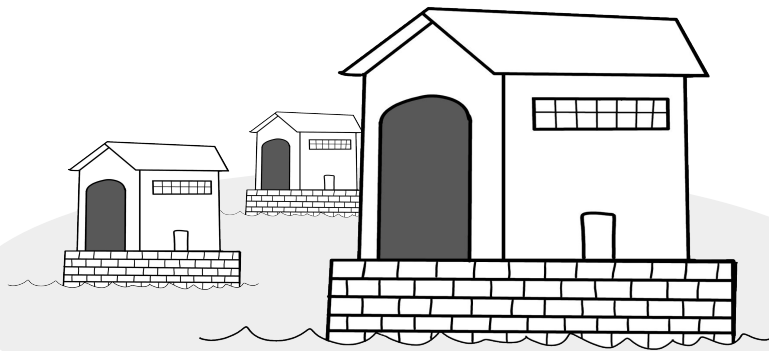
# Compartir la imatge

---



Paquets

Públics o privats



Registres d'imatges



# Compartir la imatge: paquets

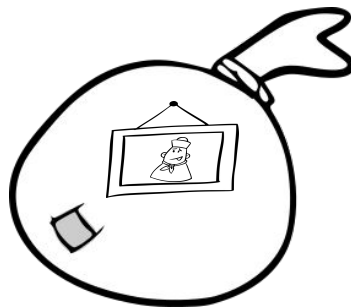
---

Podem passar imatges a un  
altre sistema de forma  
manual



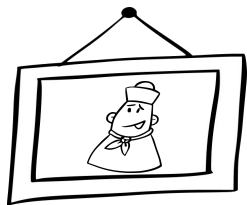
```
$ docker save utrescu-mariner > mariner.tar
```

```
$ docker load -i mariner.tar
```



# Compartir la imatge: Registres

---



Per compartir una imatge l'enviem a un repositori

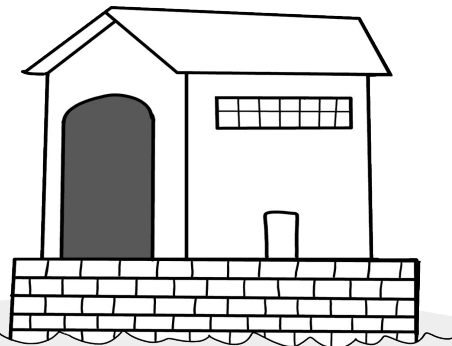
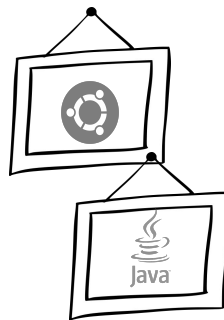


```
$ docker login  
$ docker push utrescu/Mariner
```

Cal obrir un compte per compartir



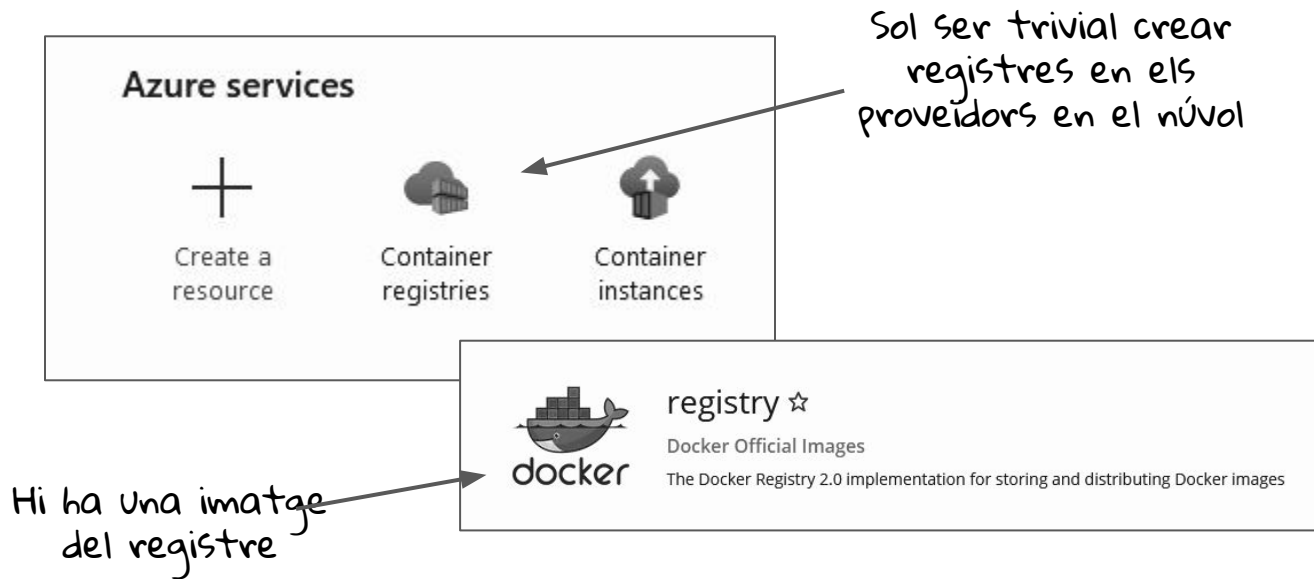
Docker Hub



Repositoris d'imatges

# Registres

---





P3: Empaquetar un programa de forma automatizable

# Creació amb fitxers Dockerfile

---

El gran problema de crear imatges a partir del contenidor és la part “manual”



# Dockerfile

---

Lista  
d'instruccions per  
crear la imatge

## Dockerfile

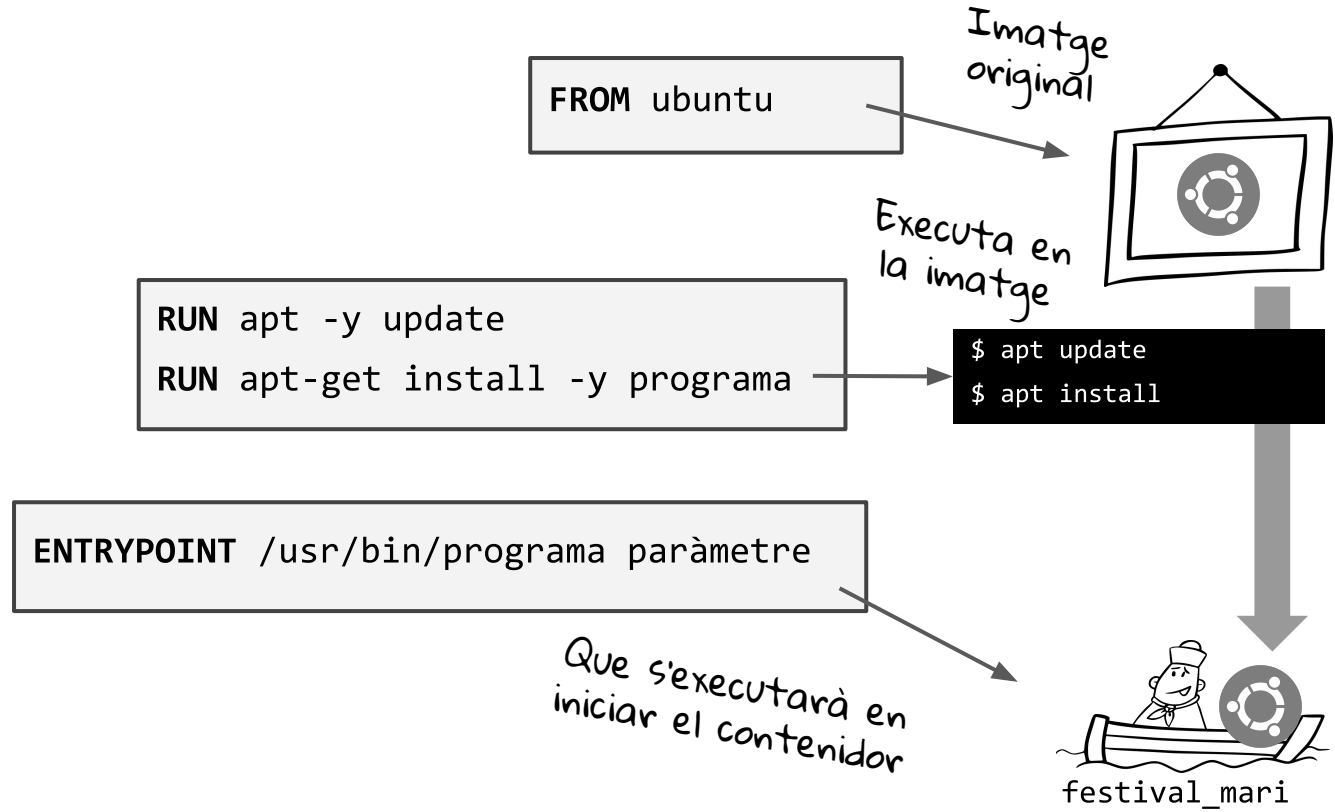
```
FROM ubuntu
```

```
RUN apt -y update
```

```
RUN apt-get install -y programa
```

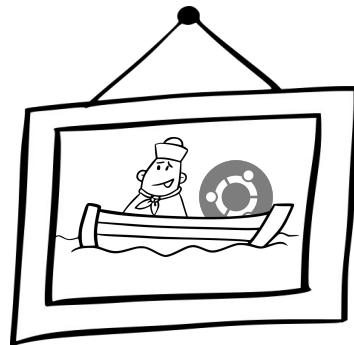
```
ENTRYPOINT ["programa","paràmetre"]
```





# Construir la imatge

---



```
$ docker build -t utrescu/nom .
```

Etiqueta de la  
imatge

On és el  
Dockerfile?



# Variables d'entorn

---

Les variables  
d'entorn  
permeten enviar  
dades dinàmiques

Importants en  
desplegaments  
en el núvol

## Dockerfile

```
FROM ubuntu

RUN apt -y update
RUN apt-get install -y cowsay
ENV TEXT "Hola"

ENTRYPOINT /usr/games/cowsay $TEXT
```

Es poden  
sobreescriure en  
crear el  
contenedor

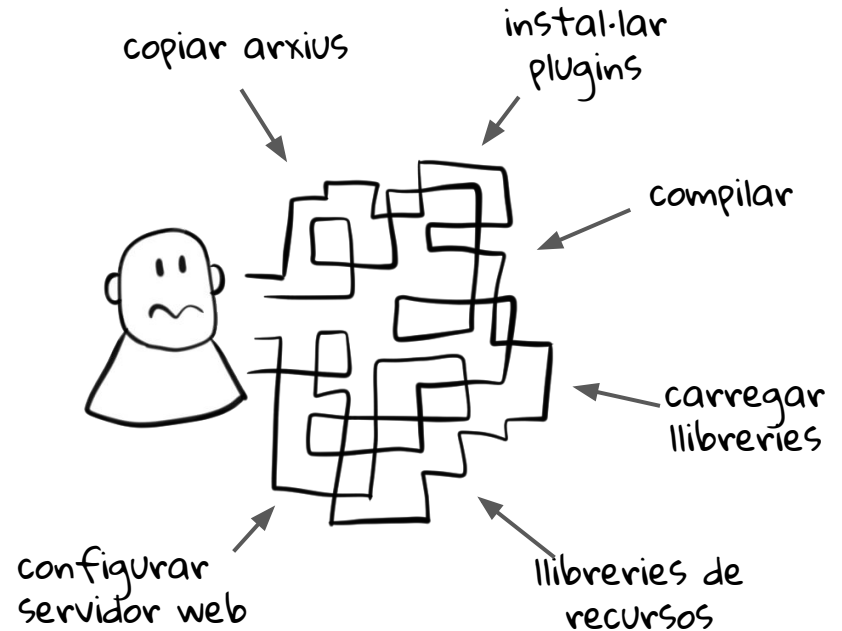


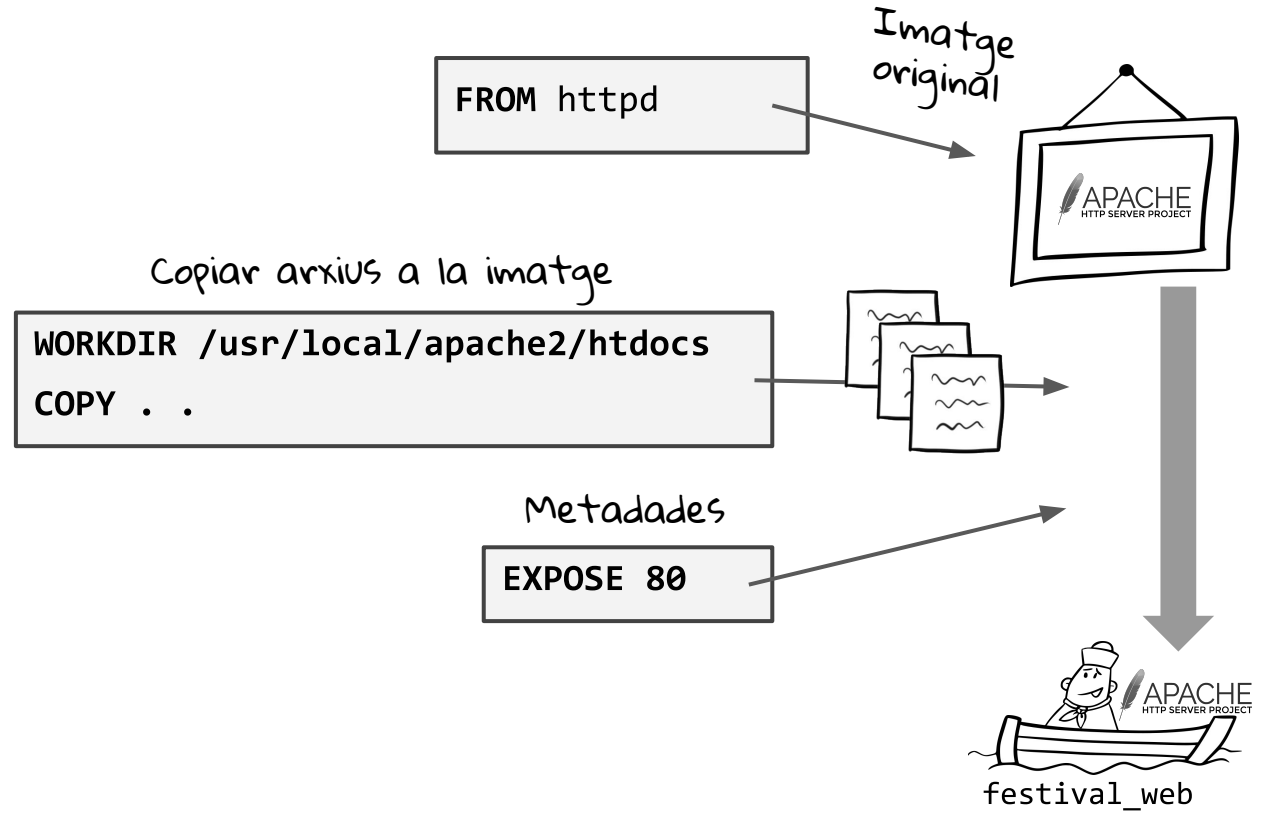
P4: Quina web més maca

# Crear imatges més complexes

---

Preparar l'entorn  
on desplegar no  
sempre compilar





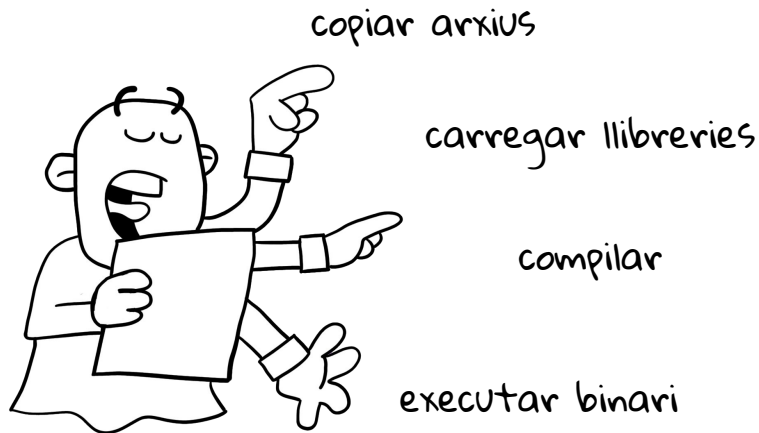


P5: Compila per mi

# Empaquetar un programa des de codi font

---

Es pot  
automatitzar el  
procés des de codi  
font



```
FROM mcr.microsoft.com/dotnet/sdk:5.0
```

Imatge original

Copiar codi font a la imatge

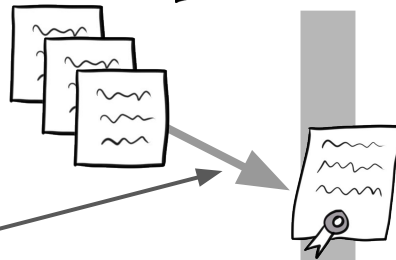
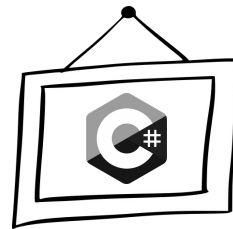
```
WORKDIR /app  
COPY . .
```

compilar

```
RUN dotnet build
```

Executar

```
ENTRYPOINT ["dotnet", "programa.dll"]
```





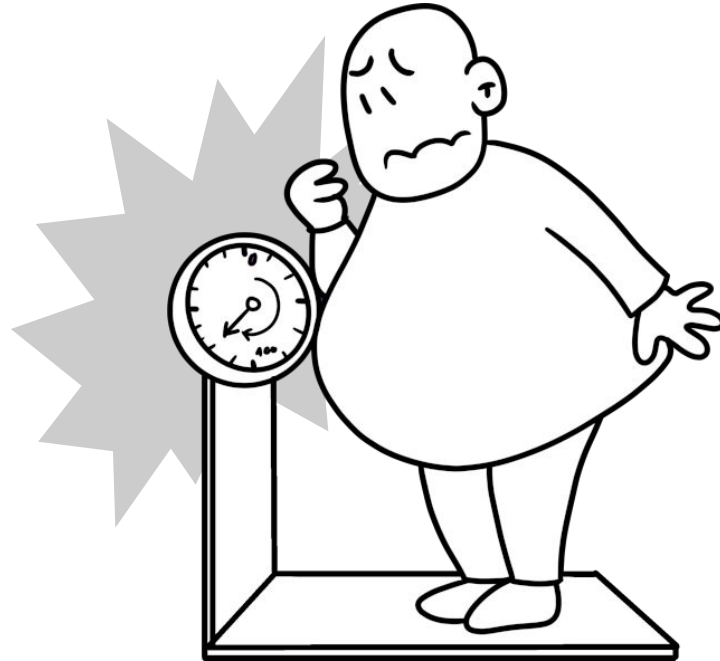
P6: Imatges a dieta



# Aprimar imatges

---

A mesura que s'hi  
afegeixen coses  
les imatges es fan  
més grans



# Construccions amb etapes

---

Un contenidor  
per construir  
l'executable

## Dockerfile

Nom per fer-hi  
referència

```
FROM mcr.microsoft.com/dotnet/sdk:5.0 as BUILD
COPY . .
RUN dotnet publish -c Release -o out

FROM mcr.microsoft.com/dotnet/runtime:5.0
COPY --from=BUILD /out .
ENTRYPOINT ["dotnet", "programa.dll"]
```

Contenidor de  
desplegament



En vull més!

# Més feina?

- Feu un programa, en el llenguatge que vulgueu, que compti les paraules d'un text i empaqueteu-lo en un contenidor.
  - El text a avaluar li ha d'arribar en una variable d'entorn



