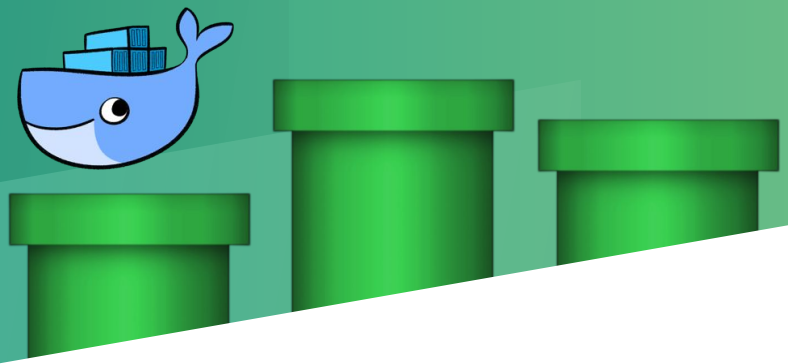
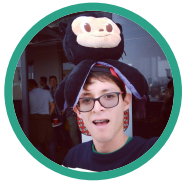
Crea un pipeline con Docker y Luigi





Rodolfo Ferro
https://rodolfoferro.xyz

¡Hola!

Soy Rodo Ferro.

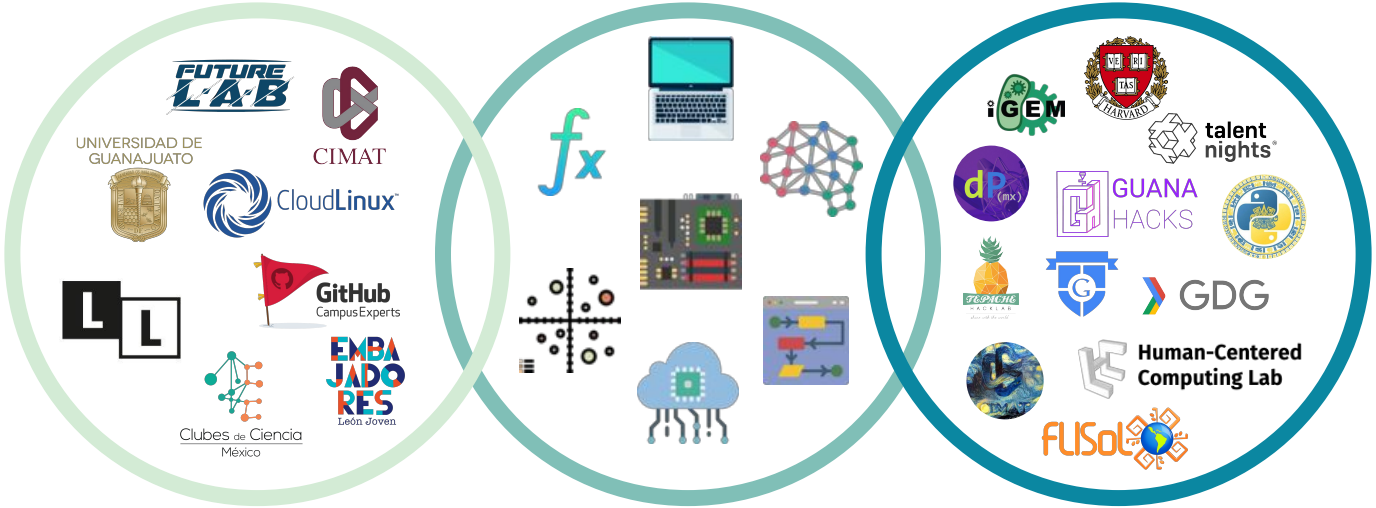
Computer-mathemagician $f[\text{💻}]$

Encuéntrame:

@FerroRodolfo

@rodo_ferro

Sobre mí:


FUTURE L·A·B
UNIVERSIDAD DE GUANAJUATO
CIMAT
CloudLinux
GitHub
Campus Experts
EMBAJADORES
León Joven
Clubes de Ciencia
México
fx
iGEM
talent nights
dP (mx)
GUANA HACKS
GDG
Human-Centered Computing Lab
FLISoL

Objetivos:

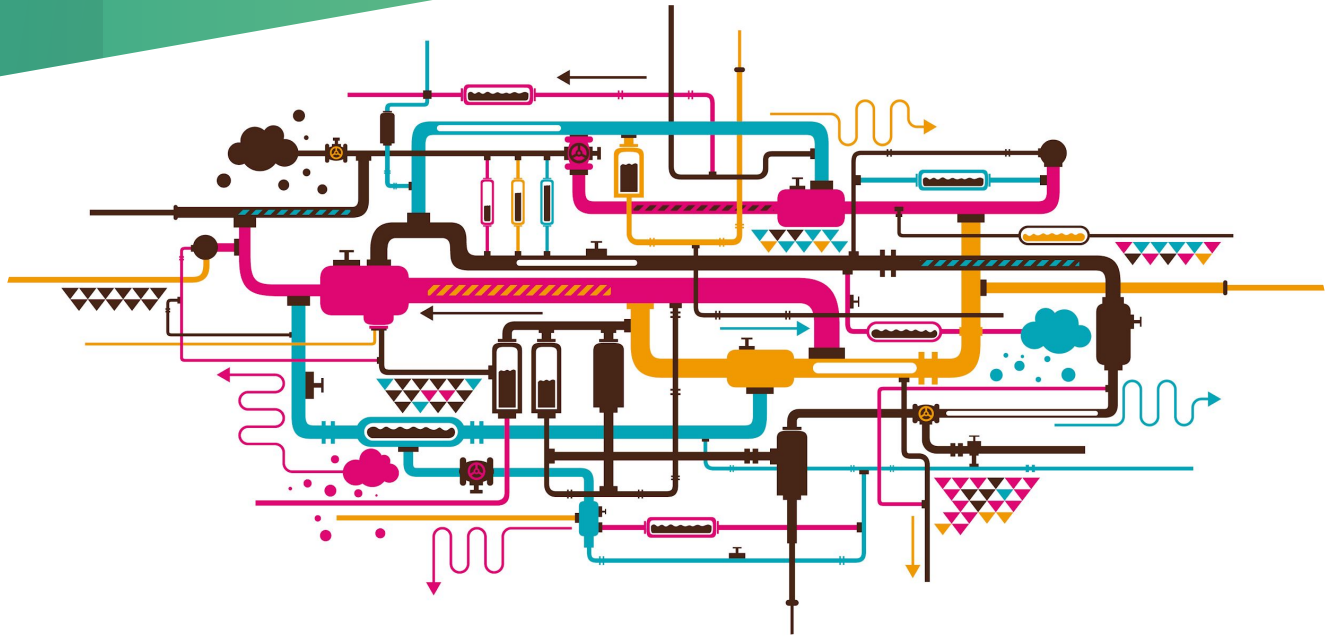
- ¿Por qué armar un *Pipeline*?
- Panorama general de la charla
- Contenedorización con Docker
- Pipeline básico con Luigi

Charla → Código

1.
¿Por qué armar un *Pipeline?*

Eficientemos trabajo

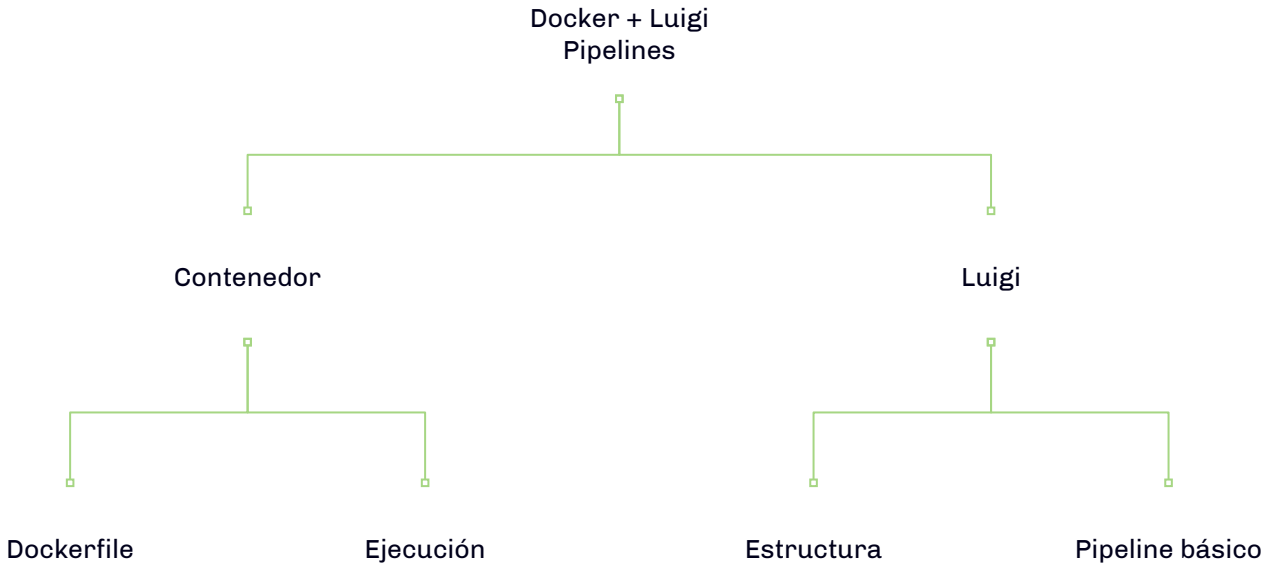
Pipeline



2. Panorama general de la charla

¿De qué va todo esto?

Flujo de la charla



3. Contenedorización con Docker

Docker salvando el mundo

Dockerfile

```
# We will use Ubuntu for our image:
FROM ubuntu:latest

# Updating Ubuntu packages:
RUN apt-get update && yes|apt-get upgrade && \
    apt-get install -y wget && \
    apt-get install -y sudo && \
    apt-get install -y vim && \
    apt-get install -y nano && \
    apt-get install -y htop

# Set locales for encoding:
RUN apt-get update && apt-get install -y locales && rm -rf /var/lib/apt/lists/* \
    && localedef -i en_US -c -f UTF-8 -A /usr/share/locale/locale.alias en_US.UTF-8
ENV LANG en_US.utf8

# Add working directory:
WORKDIR /docker-luigi-pipes

# Copy files into our container:
COPY . /docker-luigi-pipes
```

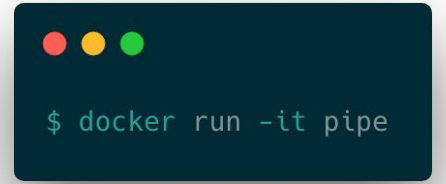
Dockerfile

```
# Install Anaconda:
RUN wget https://repo.continuum.io/archive/Anaconda3-5.0.1-Linux-x86_64.sh && \
    bash Anaconda3-5.0.1-Linux-x86_64.sh -b && \
    rm Anaconda3-5.0.1-Linux-x86_64.sh

# Set path to conda:
ENV PATH /root/anaconda3/bin:$PATH

# Update Anaconda packages:
RUN conda update conda && \
    conda update anaconda && \
    conda update --all

# Install conda packages:
RUN conda install -c conda-forge luigi scipy numpy pillow tesseract && \
    conda install -c menpo opencv && \
    conda install -c jim-hart pytesseract
```

A correr...

Construimos el contenedor:

Lo corremos:



4. Pipeline básico con Luigi

¿Luigi es mejor que Mario?

About Luigi

Luigi is a Python (2.7, 3.6, 3.7 tested) package that helps you build complex pipelines of batch jobs. It handles dependency resolution, workflow management, visualization, handling failures, command line integration, and much more.

Estructura

```python
import luigi


class MyTask(luigi.Task):
    filename = luigi.Parameter(default='../assets/test.png')

    def requires(self):
        # Returns a list of requirements (other tasks):
        return []

    def output(self):
        # Returns the output of the task into a Target object:
        return luigi.LocalTarget('my_output_file.txt')

    def run(self):
        # Runs main action of task, outputs results:
        with self.output().open('w') as out:
            out.write(new_name)
```

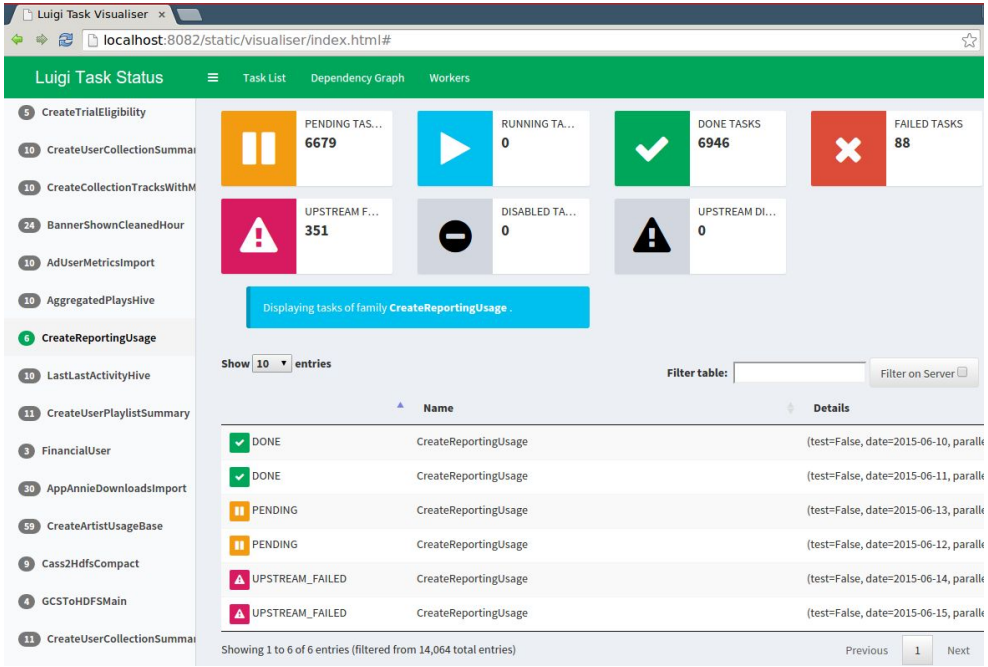
A correr...

Ejecutamos la tarea:

Por hacer:

- Extender tareas
- Ejecutar visualizador
- ¿Hacer **PR** al repo?
- **...**

Visualizador



¡Gracias!

¿Preguntas?

Rodolfo Ferro:

@FerroRodolfo

@rodo_ferro

https://rodolfoferroperez.xyz