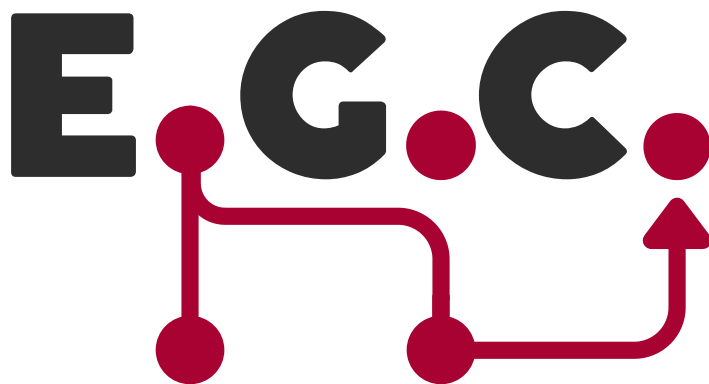




Grado en Ingeniería Informática – Ingeniería del Software

Evolución y Gestión de la Configuración



Escuela Técnica Superior de
Ingeniería Informática

Práctica 0 Preparación del entorno



1. ¿Qué es uvlhub.io?
 2. Documentación oficial
 3. Arquitectura de la aplicación
 4. Entornos
 5. ¿Qué necesitamos?
 6. Python: entornos virtuales
 7. Ejercicio práctico: configurar GitHub y forkear
 8. Ejercicio práctico: instalar uvlhub
 9. Ejercicio práctico: tutorial C.R.U.D.
- 

1. **¿Qué es uvlhub.io?**
 2. Documentación oficial
 3. Arquitectura de la aplicación
 4. Entornos
 5. ¿Qué necesitamos?
 6. Python: entornos virtuales
 7. Ejercicio práctico: configurar GitHub y forkear
 8. Ejercicio práctico: instalar uvlhub
 9. Ejercicio práctico: tutorial C.R.U.D.
- 

1. ¿Qué es uvlhub.io?

Dataset en ciencia e investigación



¡Muchos datasets!

1. ¿Qué es uvlhub.io?

Dataset en ciencia e investigación



- Datasets observacionales
- Datasets experimentales
- Datasets de encuestas
- Datasets genómicos
- Datasets de imágenes
- Datasets de series temporales
- Datasets geoespaciales
- Datasets de simulación
- ...

1. ¿Qué es uvlhub.io?

Dataset en ciencia e investigación



Son IMPORTANTES porque permiten...

- Base empírica
- Reproducibilidad
- Nuevos hallazgos
- Colaboración
- Transparencia
- Educación

1. ¿Qué es uvlhub.io?

El problema



**Repositorios
privados**



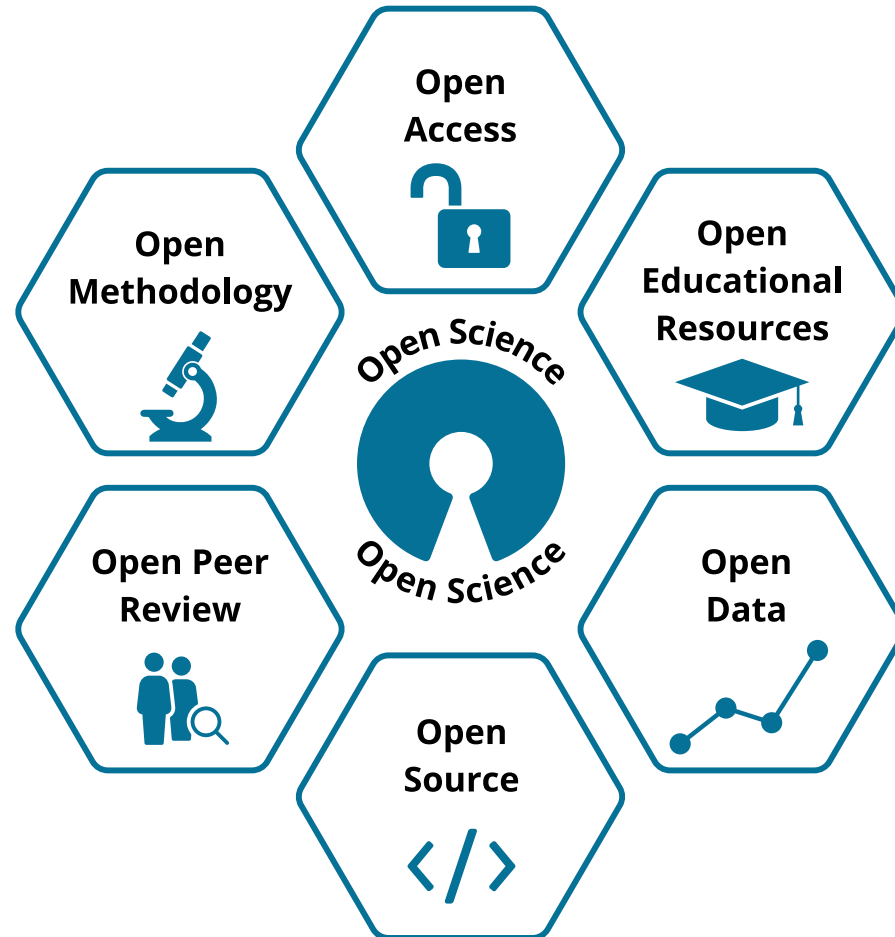
**Datasets perdidos,
corruptos,
incompletos...**



**¡Sistema
caído!**

1. ¿Qué es uvlhub.io?

Los seis principios de la Ciencia Abierta



1. ¿Qué es uvlhub.io?

La solución



**repositorio de modelos de características en formato UVL
siguiendo principios de la ciencia abierta**



UNIVERSIDAD
DE MÁLAGA



universität
uulm

1. ¿Qué es uvlhub.io?



Repositorio oficial para la asignatura

github.com/EGCETSII/uvlhub

1. ¿Qué es uvlhub.io?
 2. **Documentación oficial**
 3. Arquitectura de la aplicación
 4. Entornos
 5. ¿Qué necesitamos?
 6. Python: entornos virtuales
 7. Ejercicio práctico: configurar GitHub y forkear
 8. Ejercicio práctico: instalar uvlhub
 9. Ejercicio práctico: tutorial C.R.U.D.
- 

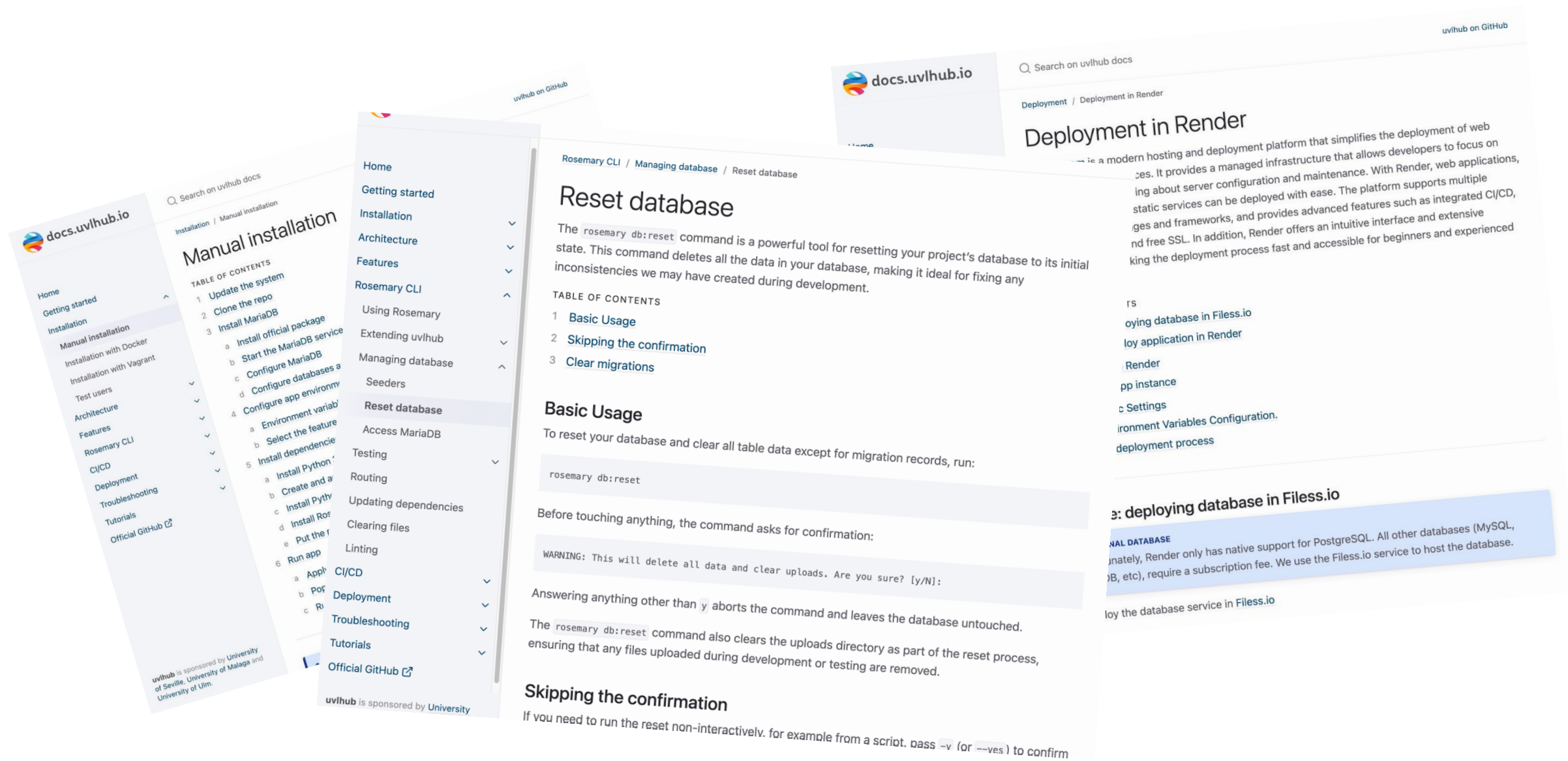
2. Documentación oficial



docs.uvhub.io

guías de instalación, guías de comandos,
migraciones, despliegue, módulos,
resolución de dudas...

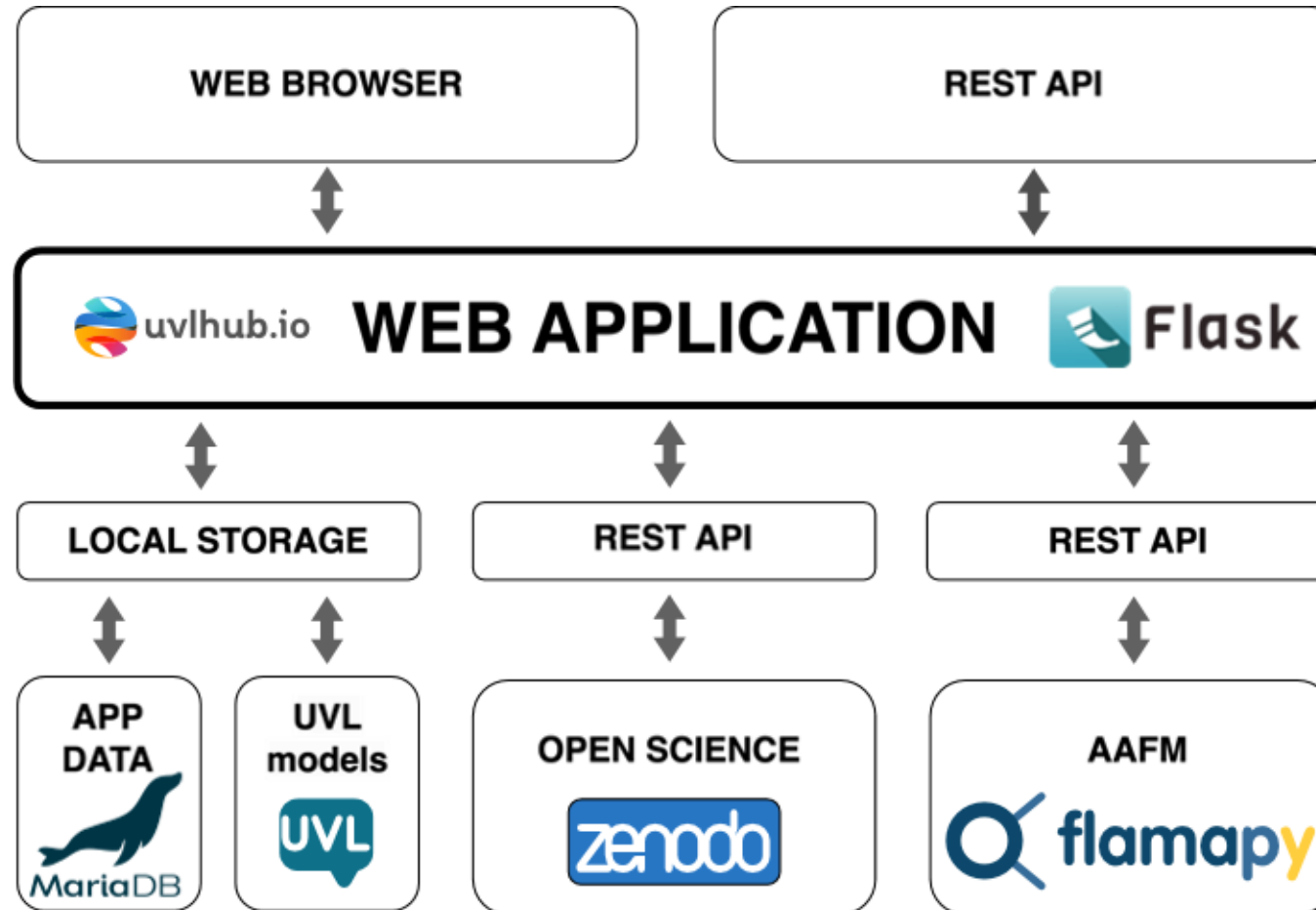
2. Documentación oficial



1. ¿Qué es uvlhub.io?
 2. Documentación oficial
 3. **Arquitectura de la aplicación**
 4. Entornos
 5. ¿Qué necesitamos?
 6. Python: entornos virtuales
 7. Ejercicio práctico: configurar GitHub y forkear
 8. Ejercicio práctico: instalar uvlhub
 9. Ejercicio práctico: tutorial C.R.U.D.
- 

3. Arquitectura de la aplicación

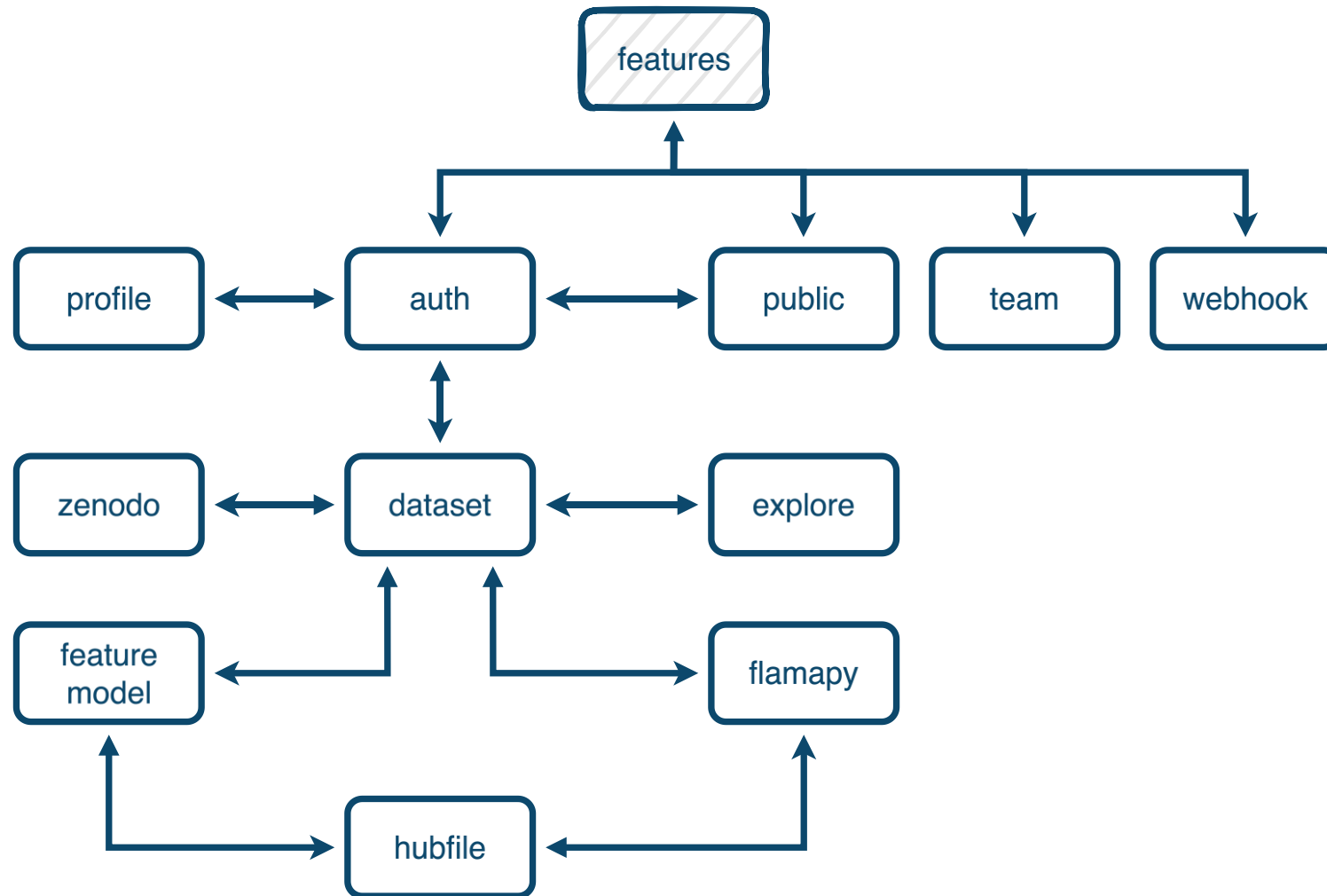
Vistazo general



docs.uvlhub.io/architecture/overview

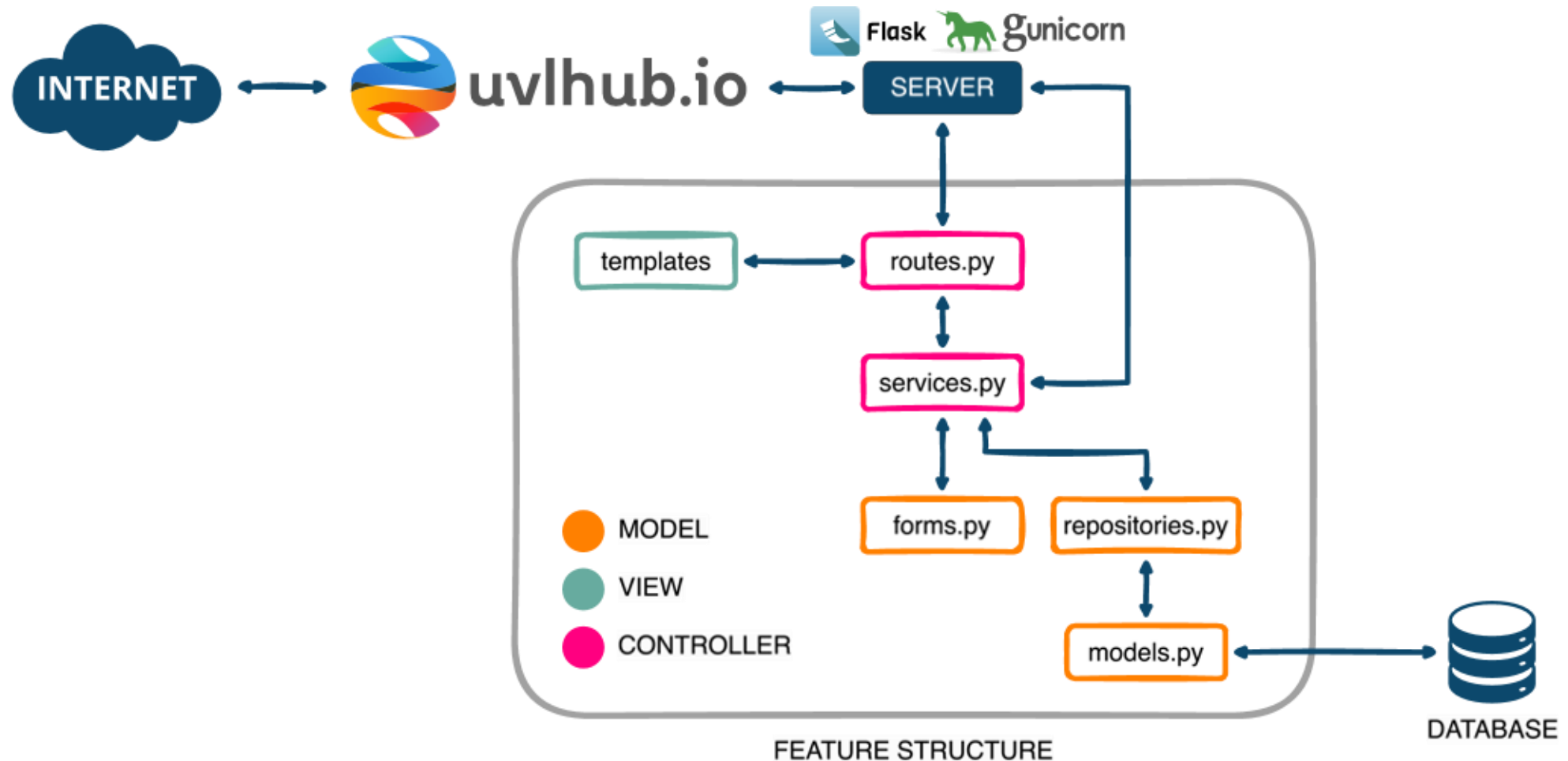
3. Arquitectura de la aplicación

Features



3. Arquitectura de la aplicación

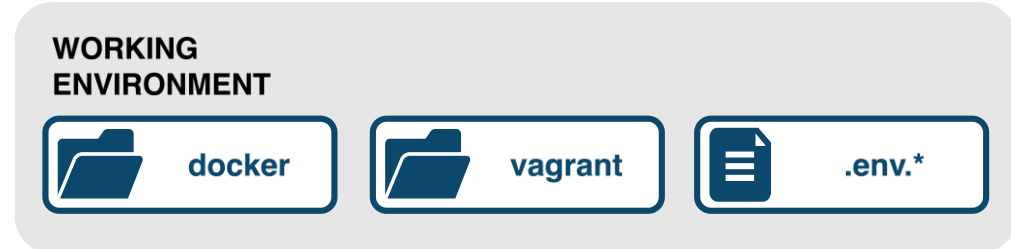
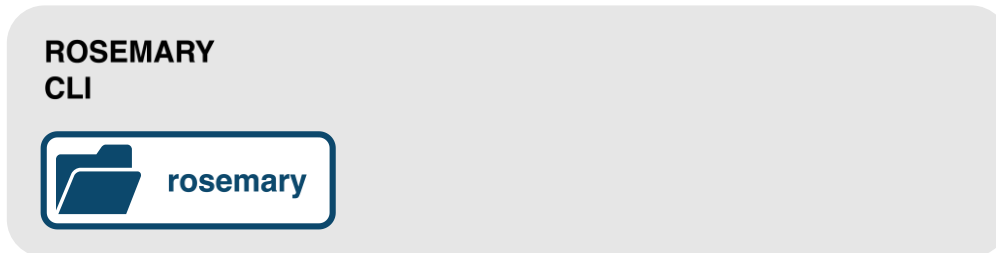
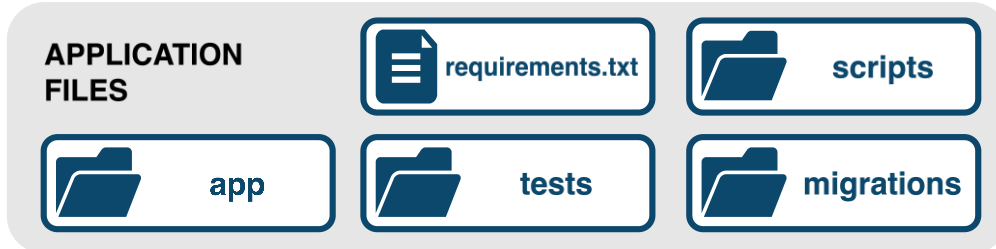
Petición HTTP



docs.uvlhub.io/architecture/http_request

3. Arquitectura de la aplicación

Estructura del proyecto



docs.uvlhub.io/architecture/project_structure

3. Arquitectura de la aplicación

Rosemary CLI



Rosemary es una herramienta CLI (*Command Line Interface*) pensada para facilitar la gestión de proyectos y las tareas de desarrollo

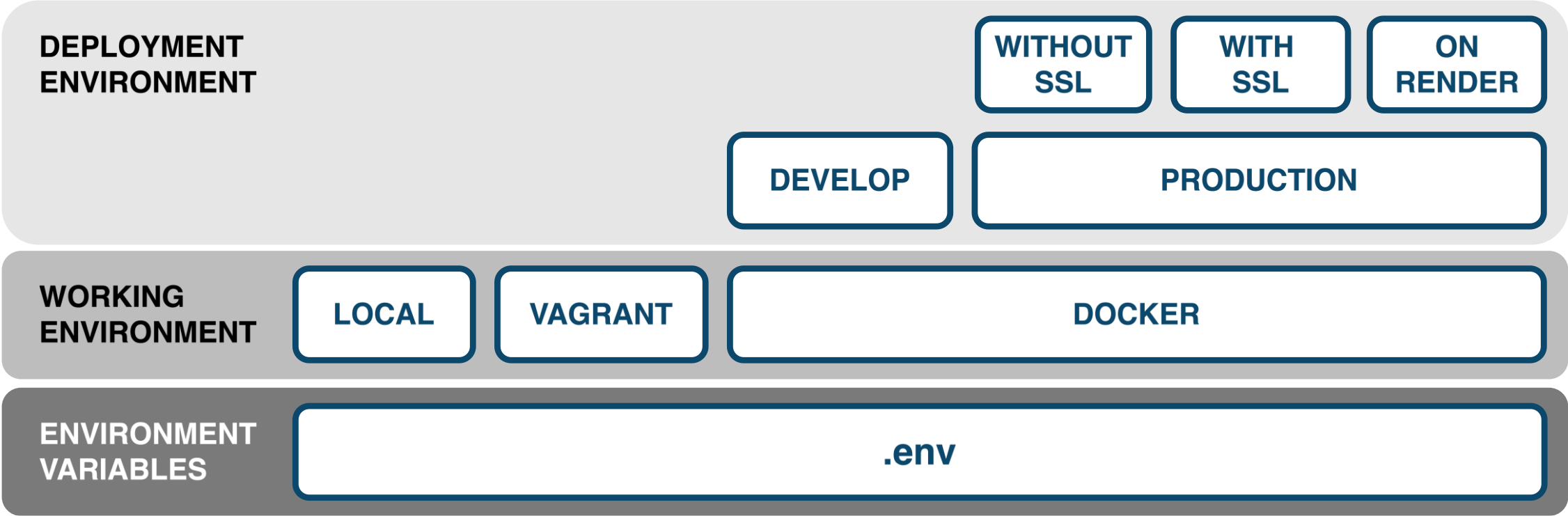


docs.uvlhub.io/rosemary

1. ¿Qué es uvlhub.io?
 2. Documentación oficial
 3. Arquitectura de la aplicación
 - 4. Entornos**
 5. ¿Qué necesitamos?
 6. Python: entornos virtuales
 7. Ejercicio práctico: configurar GitHub y forkear
 8. Ejercicio práctico: instalar uvlhub
 9. Ejercicio práctico: tutorial C.R.U.D.
- 

4. Entornos

Entorno de despliegue VS. entorno de trabajo



4. Entornos

Variables de entorno

```
FLASK_APP_NAME="UVLHUB.IO(dev)"  
FLASK_ENV=development  
DOMAIN=localhost:5000  
MARIADB_PORT=3306  
MARIADB_HOSTNAME=localhost  
MARIADB_DATABASE=uvlhubdb  
MARIADB_TEST_DATABASE=uvlhubdb_test  
MARIADB_USER=uvlhubdb_user  
MARIADB_PASSWORD=uvlhubdb_password  
MARIADB_ROOT_PASSWORD=uvlhubdb_root_password  
WORKING_DIR=""
```

Archivo .env (local)

```
FLASK_APP_NAME="UVLHUB.IO(dev)"  
FLASK_ENV=development  
DOMAIN=localhost  
MARIADB_PORT=3306  
MARIADB_HOSTNAME=db  
MARIADB_DATABASE=uvlhubdb  
MARIADB_TEST_DATABASE=uvlhubdb_test  
MARIADB_USER=uvlhubdb_user  
MARIADB_PASSWORD=uvlhubdb_password  
MARIADB_ROOT_PASSWORD=uvlhubdb_root_password  
WORKING_DIR=/workspace/
```

Archivo .env (docker)

4. Entornos

Variables de entorno

```
FLASK_APP_NAME="UVLHUB.IO(dev)"
FLASK_ENV=development
DOMAIN=localhost:5000
MARIADB_PORT=3306
MARIADB_HOSTNAME=localhost
MARIADB_DATABASE=uvlhubdb
MARIADB_TEST_DATABASE=uvlhubdb_test
MARIADB_USER=uvlhubdb_user
MARIADB_PASSWORD=uvlhubdb_password
MARIADB_ROOT_PASSWORD=uvlhubdb_root_password
WORKING_DIR=""
```

Archivo .env (local)

```
FLASK_APP_NAME="UVLHUB.IO(dev)"
FLASK_ENV=development
DOMAIN=localhost
MARIADB_PORT=3306
MARIADB_HOSTNAME=db
MARIADB_DATABASE=uvlhubdb
MARIADB_TEST_DATABASE=uvlhubdb_test
MARIADB_USER=uvlhubdb_user
MARIADB_PASSWORD=uvlhubdb_password
MARIADB_ROOT_PASSWORD=uvlhubdb_root_password
WORKING_DIR=/workspace/
```

Archivo .env (docker)

1. ¿Qué es uvlhub.io?
 2. Documentación oficial
 3. Arquitectura de la aplicación
 4. Entornos
 5. ¿Qué necesitamos?
 6. Python: entornos virtuales
 7. Ejercicio práctico: configurar GitHub y forkear
 8. Ejercicio práctico: instalar uvlhub
 9. Ejercicio práctico: tutorial C.R.U.D.
- 

5. ¿Qué necesitamos?

REQUISITOS **HARDWARE**



- **Intel Core i5 de 10ª** generación
o **AMD Ryzen 5** de la serie 4000
- **8 GB** de RAM DDR4
- SSD o NVMe de **256 GB**
- **100 GB de espacio en disco**
(partición en Ubuntu o MV)

REQUISITOS **SOFTWARE**



- **Ubuntu 24.04 LTS** o superior
- **Python 3.13**
- Docker, Docker compose
- Virtualbox, Vagrant, Ansible
- Visual Studio Code
- Git

5. ¿Qué necesitamos?

Nuestro IDE: Visual Studio Code



EXTENSIONES ÚTILES

- Docker
- Python
- Debugpy
- Python indent
- Flake8
- ESLint
- PyPi Assistant
- Flask Snippets

5. ¿Qué necesitamos?

Instalar Python 3.13

Añadir repositorio

```
sudo apt install -y software-properties-common  
sudo add-apt-repository -y ppa:deadsnakes/ppa
```

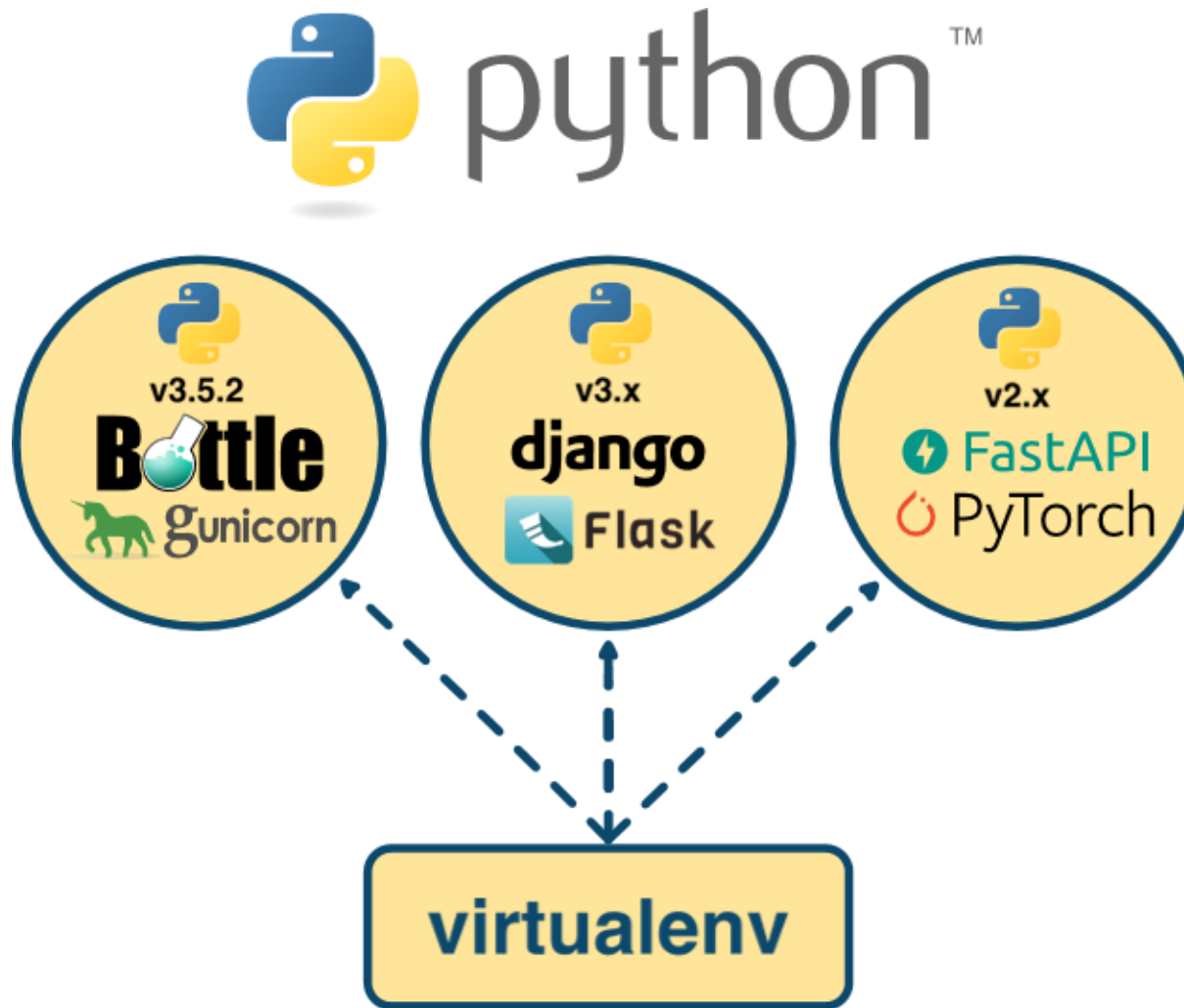
Actualizar e instalar

```
sudo apt update -y  
sudo apt install -y python3.13 python3.13-venv
```

1. ¿Qué es uvlhub.io?
 2. Documentación oficial
 3. Arquitectura de la aplicación
 4. Entornos
 5. ¿Qué necesitamos?
 - 6. Python: entornos virtuales**
 7. Ejercicio práctico: configurar GitHub y forkear
 8. Ejercicio práctico: instalar uvlhub
 9. Ejercicio práctico: tutorial C.R.U.D.
- 

6. Python: entornos virtuales

¿Qué son?



6. Python: entornos virtuales

Comandos básicos

Crear entorno virtual

```
python3.13 -m venv <myenvname>
```

Activar entorno virtual

```
source <myenvname>/bin/activate
```

Desactivar entorno virtual

```
deactivate
```

1. ¿Qué es uvlhub.io?
 2. Documentación oficial
 3. Arquitectura de la aplicación
 4. Entornos
 5. ¿Qué necesitamos?
 6. Python: entornos virtuales
 - 7. Ejercicio práctico: configurar GitHub y forkear**
 8. Ejercicio práctico: instalar uvlhub
 9. Ejercicio práctico: tutorial C.R.U.D.
- 

7. Ejercicio práctico: configurar GitHub y forkear

Git: instalar y configurar

```
sudo apt install git  
git config --global user.name "<tu nombre y apellidos entrecomillados>"  
git config --global user.email <tu email>
```

Git: generar par de claves SSH

```
ssh-keygen -t rsa -b 4096
```

7. Ejercicio práctico: configurar GitHub y forkear

Git: visualizar clave pública y copiar

```
cd ~/.ssh  
cat id_rsa.pub
```

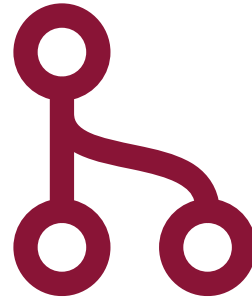
Git: guardar clave pública en GitHub

GitHub -> Settings -> SSH and GPG keys -> New SSH key -> Pegar clave pública (id_rsa.pub)

7. Ejercicio práctico: configurar GitHub y forkear



github.com/EGCETSII/uvlhub



FORK!



github.com/<tuusuario>/uvlhub_practicas



1. ¿Qué es uvlhub.io?
 2. Documentación oficial
 3. Arquitectura de la aplicación
 4. Entornos
 5. ¿Qué necesitamos?
 6. Python: entornos virtuales
 7. Ejercicio práctico: configurar GitHub y forkear
 - 8. Ejercicio práctico: instalar uvlhub**
 9. Ejercicio práctico: tutorial C.R.U.D.
- 

8. Ejercicio práctico: instalar uvlhub



docs.uvlhub.io/installation/manual_installation

Docs / Installation / Manual installation

1. ¿Qué es uvlhub.io?
 2. Documentación oficial
 3. Arquitectura de la aplicación
 4. Entornos
 5. ¿Qué necesitamos?
 6. Python: entornos virtuales
 7. Ejercicio práctico: configurar GitHub y forkear
 8. Ejercicio práctico: instalar uvlhub
 9. **Ejercicio práctico: tutorial C.R.U.D.**
- 

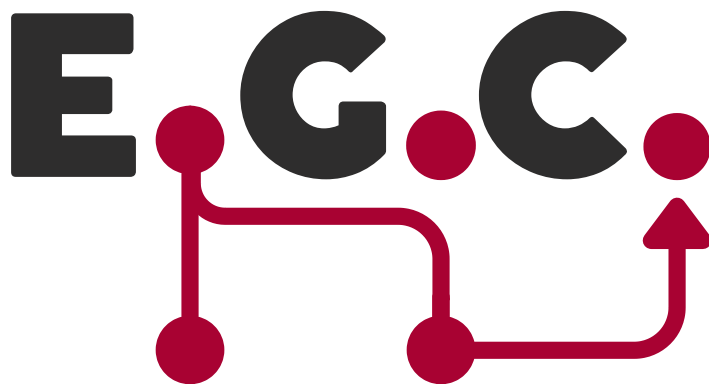
9. Ejercicio práctico: tutorial C.R.U.D.



docs.uvlhub.io/tutorials/crud_tutorial



Docs / Tutorials / C.R.U.D. tutorial



Grado en Ingeniería Informática – Ingeniería del Software

Evolución y Gestión de la Configuración



Escuela Técnica Superior de
Ingeniería Informática

¡Gracias!

“En una jerarquía, todo empleado tiende a ascender hasta su nivel más alto de incompetencia.”

- Lawrence J. Peter