

ENTORNO VIRTUAL

Crear entorno virtual

```
python -m venv <myenvname>
```

Activar entorno virtual

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

Desactivar entorno virtual

```
deactivate
```

CONFIGURAR GITHUB

DaniTorto_03

Instalar y configurar

```
sudo apt install git
```

```
git config --global user.name "DanTorBar"
```

```
git config --global user.email dantorbar1@alum.us.es
```

```
git init
```

Generar par de claves SSH

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

Visualizar clave pública y copiar

```
cd ~/.ssh
```

```
cat id_rsa.pub
```

Guardar clave pública en GitHub

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

GIT

Crear rama

git branch <nombre_rama>

Saltar a la rama

git checkout <nombre_rama>

Crear y saltar a rama

git checkout -b <nombre_rama>

Subir rama a repositorio remoto

git push -u origin <nombre_rama>

-u = upstream = referencia de seguimiento, es la conexión entre la rama local y la rama remota

Listar ramas

git branch

Listar ramas remotas

git branch -r

Listar ramas remotas y locales

git branch -a

Eliminar rama local (si la rama ya ha sido fusionada con otra)

```
git branch -d <nombre_rama>
```

Eliminar rama local (forzar eliminación)

```
git branch -D <nombre_rama>
```

Eliminar rama remota

```
git branch origin --delete <nombre_rama>
```

Fusionar ramas (merge) (no altera el historial de las ramas)

```
git merge <nombre_rama_que_quieres_mergegear_a_en_la_que_estas>
```

Fusionar ramas (rebase) (altera el historial)

```
git rebase -i <nombre_rama_que_quieres_mergegear_a_en_la_que_estas> (ó HEAD~30)
```

Aplicar un commit de una rama a otra (sin fusionar)

```
git cherry-pick <commit_hash>
```

```
git cherry-pick <commit_hash1>...<commit_hash2>
```

Reseteo

git reset --soft : mueve la rama local al commit anterior (Staging y Working Tree sin cambios)

git reset --mixed (por defecto): mueve la rama local al commit anterior y quita los cambios del Staging (Working Tree sin cambios)

git reset --hard : mueve la rama local al commit anterior y borra los cambios tanto del Staging como del Working Tree

Ver histórico de commits

git log (--graph) --oneline (--all)

Deshacer un commit (crea uno nuevo deshaciendo el anterior)

git revert <commit_hash>

Deshacer un commit (igual, pero sin editar el mensaje)

git revert --no-edit <commit_hash>

Corregir último commit

git commit --amend -m "feat: Add testing file. Closes #<ID>"

Ver cambios no confirmados

git diff

Fusionar rama remota sin hacer rebase

git pull --no-rebase origin feature/practicandogit_a

Eliminar stash en la pila

git stash drop

Dejar de hacer seguimiento de un archivo

echo debug.log >> .gitignore

git rm --cached debug.log

Hacer seguimiento de un archivo perteneciente a un conjunto ignorado

cat .gitignore

*log

!debug.log

git add debug.log

Proceso normal de merge

- Me pongo en la rama sobre la que quiero mezclar los cambios
- Me aseguro de que la rama no está desactualizada
- Mezclar los cambios sobre la rama actual desde la rama origen (source)
- Resolver conflictos
- Hacer commit del merge
- Abortar el merge

git checkout main

git pull origin main

git merge feature_x

...

git commit -m "Resolve merge

conflicts"

git merge --abort

Ver las diferencias entre dos archivos / ramas / commits

git diff --> Cambios no añadidos al área de staging

git diff --staged --> Cambios añadidos al área de staging pero no confirmados

git diff main feature-branch --> Diferencias entre dos ramas

git diff commit1 commit2 --> Diferencias entre dos commits

git diff -w (si se quiere ignorar los espacios en blanco)

Crear un parche (patch)

git diff > my-changes.patch --> Cambios no confirmados en mi área de trabajo

git diff --staged > my-staged-changes.patch --> Cambios en el área de staging

git format-patch -1 <commit-hash> --> Patch de un commit específico

Aplicar un parche (patch)

git apply my-changes.patch

Crear un commit a partir de un parche (patch)

git am my-changes.patch

Revertir un parche (patch)

git apply -R my-changes.patch

MARIADB

Instalar MariaDB

```
sudo apt install mariadb-server -y
```

Iniciar MariaDB

```
sudo systemctl start mariadb
```

Configurar MariaDB

```
sudo mysql_secure_installation
```

- Enter current password for root (enter for none): (enter)
- Switch to unix_socket authentication [Y/n]: `y`
- Change the root password? [Y/n]: `y`
 - New password: `uvlhubdb_root_password`
 - Re-enter new password: `uvlhubdb_root_password`
- Remove anonymous users? [Y/n]: `y`
- Disallow root login remotely? [Y/n]: `y`
- Remove test database and access to it? [Y/n]: `y`
- Reload privilege tables now? [Y/n]: `y`

Configurar bases de datos y usuarios

sudo mysql -u root -p

```
CREATE DATABASE uvlhubdb;
```

```
CREATE DATABASE uvlhubdb_test;
```

```
CREATE USER 'uvlhubdb_user'@'localhost' IDENTIFIED BY 'uvlhubdb_password';
```

```
GRANT ALL PRIVILEGES ON uvlhubdb.* TO 'uvlhubdb_user'@'localhost';
```

```
GRANT ALL PRIVILEGES ON uvlhubdb_test.* TO 'uvlhubdb_user'@'localhost';
```

```
FLUSH PRIVILEGES;
```

```
EXIT;
```

ROSEMARY

Instalar Rosemary

pip install -e ./

Migrar base de datos

rosemary db:migrate

Poblar base de datos

rosemary db:seed

Crear módulo

rosemary make:module notepad

Listar módulos

rosemary module:list

Listar rutas de un módulo

rosemary route:list notepad

Testing con Rosemary

Ejecutar tests unitarios

rosemary test / rosemary test <module_name> / rosemary test -k <expression>

Ejecutar tests de cobertura

rosemary coverage / rosemary coverage <module_name> / rosemary coverage -html

Ejecutar tests de carga (Locust)

rosemary locust / rosemary locust <module_name>

rosemary locust:stop

Ejecutar tests de interfaz (Selenium)

rosemary selenium / rosemary selenium <module_name>

PIP

Instalar un paquete

pip install <paquete>

Instalar una versión específica de un paquete

pip install <paquete>==<versión>

Instalar un paquete desde un archivo local

pip install ./ruta/del/paquete

Instalar múltiples paquetes desde un archivo

pip install -r requirements.txt

Actualizar un paquete

pip install --upgrade <paquete>

Desinstalar un paquete

pip uninstall <paquete>

Desinstalar múltiples paquetes desde un archivo

pip uninstall -r requirements.txt

Listar paquetes instalados

pip list

Mostrar detalles de un paquete instalado

pip show <paquete>

Generar un archivo requirements.txt con los paquetes instalados

pip freeze > requirements.txt

Buscar un paquete en PyPI

pip search <palabra_clave>

Verificar dependencias conflictivas

pip check

Instalar un paquete sin usar la caché

pip install --no-cache-dir <paquete>

Actualizar pip

pip install --upgrade pip

Mostrar ayuda general de pip

pip help

Mostrar ayuda de un comando específico

pip help <comando>

Limpiar la caché de pip

pip cache purge

Instalar un paquete con dependencias opcionales

pip install <paquete>[extra]

Instalar un paquete desde un repositorio privado

pip install --extra-index-url <url_del_repositorio> <paquete>

Listar dependencias desactualizadas

pip list --outdated

Comprobar si está instalado un paquete

pip show <nombre_paquete>

FLASK

Crear una migración

```
flask db migrate -m "create_notepad_model"
```

Aplicar migraciones

```
flask db upgrade
```

Correr aplicación

```
flask run --host=0.0.0.0 --reload --debug
```

Testing

Tests de Python (pytest)

pytest

Cobertura de tests (pytest)

pytest --cov=app tests/

Este comando ejecutará todas las pruebas en la carpeta tests/ y generará un informe de cobertura que mostrará qué porcentaje del código en el archivo app.py fue ejecutado durante las pruebas.

Informe de cobertura en HTML (pytest)

pytest --cov=app --cov-report=html tests/

Para ver el informe: *xdg-open htmlcov/index.html*

Ejecución de tests con Selenium (Selenium)

pytest -s tests/test_selenium.py

Ejecución de tests Locust (Locust)

locust -f locustfile.py

Docker

Ejecutar contenedor

docker run <nombre-contenedor> / docker run -it <nom-cont> bash

docker run -td ubuntu bash

-it: interactivo + terminal

-td: terminal + detached (para que se ejecute en segundo plano)

bash: comando que se ejecuta dentro del contenedor

Ejecutar contenedor en segundo plano

docker exec -ti id_contenedor bash

Ver las imágenes que hay en el sistema

docker images

Descargar imagen

docker pull <nombre-imagen>

Listar contenedores activos

docker ps

Listar todos los contenedores

docker ps -a

Eliminar contenedor / es

`docker rm id_contenedor`

`docker rm $(docker ps -aq)` --> elimina todos los contenedores inactivos

Detener contenedor

`docker stop nombre_contenedor`

Listar imagenes

`docker images`

Eliminar imagen

`docker rmi id_imagen`

Crear contenedor NGINX con archivos HTML personalizados

`docker run -it --rm -d -p 8080:80 --name web nginx`

`--rm`: Elimina el contenedor cuando se detiene

`-p 8080:80`: mapea el puerto 80 del contenedor al puerto 8080 de tu máquina local

`--name web`: asigna “web” como nombre del contenedor

`nginx`: utiliza la imagen nginx para crear el contenedor

Iniciar un contenedor con el volumen montado

`docker run -it --rm -d -p 8080:80 --name web -v ~/site-content:/usr/share/nginx/html nginx`

`-v ~/site-content:/usr/share/nginx/html`: crea un **bind mount** que mapea el directorio `/site-content` de tu máquina local al directorio `/usr/share/nginx/html`

Crear volumen persistente

docker volume create <nombre-volumen>

Iniciar contenedor con volumen persistente

*docker run -it --rm -d -p 8080:80 --name web -v -<nombre-volumen>:/usr/share/nginx/html
nginx*

Ejecutar Docker compose

docker compose -f docker/docker-compose.dev.yml up -d

-f docker/docker-compose.dev.yml: leerá el fichero correspondiente

Detener contenedores con Docker compose

docker compose -f docker/docker-compose.dev.yml down

docker compose -f docker/docker-compose.dev.yml down -v

-v: Elimina los volúmenes del contenedor

Reconstruir / Hacer re-build de un contenedor Docker compose

Si se modifica un archivo docker-compose.*.yml es necesario ejecutar este comando

docker compose -f docker/docker-compose.dev.yml up -d --build

Vagrant

Instalar vagrant

```
sudo apt install vagrant ansible virtualbox
```

--- ó ---

```
wget -O - https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o  
/usr/share/keyrings/hashicorp-archive-keyring.gpg  
echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg]  
https://apt.releases.hashicorp.com $(lsb_release -cs) main" | sudo tee  
/etc/apt/sources.list.d/hashicorp.list  
sudo apt update && sudo apt install vagrant ansible virtualbox
```

Iniciar un entorno

```
vagrant init <nombre-entorno/box>
```

Levantar una máquina virtual

```
vagrant up / vagrant up --provision
```

Acceder a una máquina virtual

```
vagrant ssh
```

Ver el estado de las máquinas virtuales

```
vagrant global-status
```

Ver el estado de una

Apagar máquina virtual

vagrant halt / *vagrant halt -f* --> *-f*: apagado forzado

Recargar máquina virtual

vagrant reload / *vagrant reload --provision*

Destruir / eliminar una máquina

vagrant destroy

Listar boxes / imágenes

vagrant box list

Eliminar boxes / imágenes

vagrant box remove <nombre-box>

Configurar la redirección de puertos

config.vm.network "forwarded_port", guest: 80, host: 8080

OTROS

Crear archivo vacío · Actualizar fecha y hora de acceso

```
touch <nombre_archivo>
```

Listar los procesos que corren en un puerto

```
sudo lsof -i:<numero-puerto>
```

Detener un proceso

```
sudo kill -9 <id-proceso>
```

Problemas con las rutas / problemas con el módulo app

```
PYTHONPATH=/home/daniel-tortorici/Desktop/practicas/flask_testing_project <comndo>
```

```
PYTHONPATH=. <comando>
```

Error THIRD_PARTY_NOTICES

OSError: [Errno 8] Exec format error:

```
'/home/user/.wdm/drivers/chromedriver/linux64/130.0.6723.58/chromedriver-linux64/THIRD_PARTY_NOTICES.chromedriver'
```

```
pip uninstall webdriver-manager
```

```
rm ~/.wdm/ -r
```

```
pip install webdriver-manager
```

Ejecutar archivo .sh

```
sh nombre_del_archivo.sh
```

bash nombre_del_archivo.sh

Problemas con Selenium

https://uses0-my.sharepoint.com/:b:/g/personal/dantorbar1_alum_us_es/EcOHg_T--0JAKGw6enCjgF0BqzcfoW1zW7yTZnb2zlsCJw?e=Z78d0n

Workflows

Integración continua (CI)

Conventional commits

<https://github.com/EGCETSII/uvlhub/blob/main/.github/workflows/commits.yml>

Lint8

<https://github.com/EGCETSII/uvlhub/blob/main/.github/workflows/lint.yml>

Testing

<https://github.com/EGCETSII/uvlhub/blob/main/.github/workflows/tests.yml>

Despliegue continuo (CD)

Despliegue en Webhook

https://github.com/EGCETSII/uvlhub/blob/main/.github/workflows/deployment_on_webhook.yml

Despliegue en Dockerhub

https://github.com/EGCETSII/uvlhub/blob/main/.github/workflows/deployment_on_dockerhub.yml

Despliegues propuestos en las prácticas

- Codacy:
 - 1.- Añadir un paso que verifique si hay nuevas versiones de las dependencias instaladas
 - 2.- Verificar la compatibilidad del código con las versiones 3.10, 3.11 y 3.12 de Python
 - 3.- Realizar un análisis de seguridad de las dependencias instaladas con la librería pip-audit

- Render:

- 1.- Permitir que se realice el despliegue solo si un pull request ha sido aprobado y mergeado
- 2.- Permitir que se realice el despliegue solo si todas las pruebas han pasado
- 3.- Realizar un despliegue solo cuando se crea un tag en GitHub

https://1984.lsi.us.es/wiki-egc/index.php/Workflows_soluci%C3%B3n_P2_2425

Workflow de releases automáticas

<https://uses0->

my.sharepoint.com/:b:/g/personal/dantorbar1_alum_us_es/EdZjlRNhrWJKiVlwJ5laEeoBzrjQ74bOFx12U_AQZgPVRg?e=O6OUBK

UVLHUB

Instalación local

https://docs.uvlhub.io/installation/manual_installation

GUÍA DE PRÁCTICAS

Práctica1

Rosemary, venv, entorno virtual, configurar git, instalar uvlhub, CRUD

https://1984.lsi.us.es/wiki-egc/images/egc/c/ca/EGC_2024-25_P1.pdf

Práctica 2

GitHub Actions, workflows, CI, CD, Codacy, Render

https://1984.lsi.us.es/wiki-egc/images/egc/2/2b/EGC_2024-25_P2.pdf

Práctica 3

Git, GitHub, commit, pr, issue, flujo de trabajo, ramas, hook

https://1984.lsi.us.es/wiki-egc/images/egc/c/c5/EGC_2024-25_P3.pdf

https://1984.lsi.us.es/wiki-egc/images/egc/0/0c/P3_usuario_A_%28soluciones%29.pdf

https://1984.lsi.us.es/wiki-egc/images/egc/c/c0/P3_usuario_B_%28soluciones%29.pdf

Práctica 4

Flask, testing, Locust, Selenium, pytest

https://1984.lsi.us.es/wiki-egc/images/egc/3/37/EGC_2024-25_P4.pdf

https://1984.lsi.us.es/wiki-egc/index.php/Ejemplos_sencillos_de_pruebas_para_el_m%C3%B3dulo_notepad_de_UV_LHUB

Práctica 5

Docker, imagenes, contenedores, volumenenes, servicios, compose

https://1984.lsi.us.es/wiki-egc/images/egc/1/12/EGC_2024-25_P5.pdf

Práctica 6

Vagrant, imagenes, aprovisionamiento, playbook, dotenv

https://1984.lsi.us.es/wiki-egc/images/egc/b/b9/EGC_2024-25_P6.pdf

https://1984.lsi.us.es/wiki-egc/index.php/Tutorial_Campo_de_entrenamiento_de_Vagrant

https://1984.lsi.us.es/wiki-egc/index.php/Tutorial_configurando_vagrant_para_una_aplicaci%C3%B3n

Tema 3 teoría

git, commits, tracked, untracked, sandbox, merge, ramas, diff, patch

<https://hdvirtual.us.es/discovirt/index.php/s/6y69azRFtQymsKm>

Potaje-hub

<https://github.com/potaje-hub/potaje-hub>

Serranito-hub

<https://github.com/EGC-Serranito/serranito-hub>

Tortilla-hub

<https://github.com/EGC-G2-tortilla/tortilla-hub>

Solución examen 18/19

<https://github.com/danhidsan/EGC-1819-danhidsan>

COMMITTS

Add a title

Please make sure to fill out all fields to help us resolve the issue faster.

Bug Description

Provide a clear and detailed description of the bug you encountered.

Steps to Reproduce

Provide the steps to reproduce this bug.

Expected Behavior

What did you expect to happen?

Priority

Select the priority level of this issue.

Additional Information

Do you have any additional relevant information (logs, screenshots, version, etc.)?