

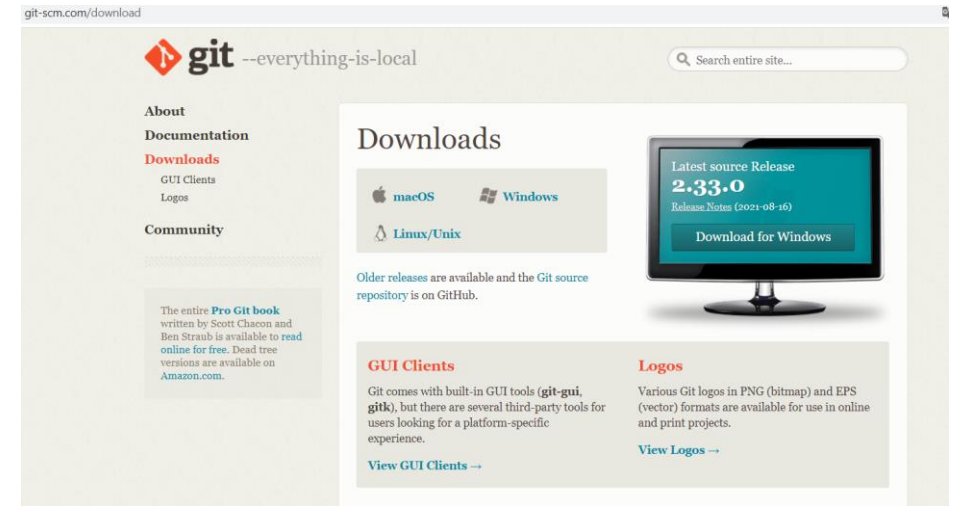
Buenas prácticas en proyectos de Ciencia de Datos

Índice

1. Malas prácticas en proyectos de Ciencia de datos
2. Herramientas para el control de versiones: git y github
3. Práctica Git y Github
4. Entornos de Python
5. Práctica: creación de un entorno de python

Material necesario para la clase

- Git: <https://git-scm.com/download/>



- Cuenta en Github: <https://github.com/>
- Jupyter notebook: <https://www.anaconda.com/products/individual>

Malas prácticas en proyectos

```
for i in people.data.users:
    response = client.api.statuses.user_timeline.get(screen_name=i.screen_name)
    print 'Got', len(response.data), 'tweets from', i.screen_name
    if len(response.data) != 0:
        ltdate = response.data[0]['created_at']
        ltdate2 = datetime.strptime(ltdate, '%a %b %d %H:%M:%S +0000 %Y')
        today = datetime.now()
        howlong = (today-ltdate2).days
        if howlong < daywindow:
            print i.screen_name, 'has tweeted in the past' , daywindow,
            totaltweets += len(response.data)
            for j in response.data:
                if j.entities.urls:
                    for k in j.entities.urls:
                        newurl = k['expanded_url']
                        urlset.add((newurl, j.user.screen_name))
        else:
            print i.screen_name, 'has not tweeted in the past', daywindow
```

Malas prácticas en proyectos

```
for i in people.data.users:
    response = client.api.statuses.user_timeline.get(screen_name=i.screen_name)
    print 'Got', len(response.data), 'tweets from', i.screen_name
    if len(response.data) != 0:
        ltdate = response.data[0]['created_at']
        ltdate2 = datetime.strptime(ltdate, '%a %b %d %H:%M:%S +0000 %Y')
        today = datetime.now()
        howlong = (today-ltdate2).days
        if howlong < daywindow:
            print i.screen_name, 'has tweeted in the past' , daywindow,
            totaltweets += len(response.data)
            for j in response.data:
                if j.entities.urls:
                    for k in j.entities.urls:
                        newurl = k['expanded_url']
                        urlset.add((newurl, j.user.screen_name))
        else:
            print i.screen_name, 'has not tweeted in the past' , daywindow
```



**We Have
To Go**

BACK →

Malas prácticas en proyectos


Escritorio > práctica

Nombre	Fecha de modifica...	Tipo
.ipynb_checkpoints	02/10/2021 18:43	Carpeta de archivos
preprocessing_data	02/10/2021 18:43	Python File
preprocessing_data_a	02/10/2021 18:43	Python File
preprocessing_data_aa_cambio	02/10/2021 18:43	Python File
preprocessing_data_aa_cambio_bueno_final	02/10/2021 18:43	Python File
preprocessing_data_aa_cambio_bueno	02/10/2021 18:43	Python File

Sin querer he enviado la versión errónea de la práctica...



Malas prácticas en proyectos

Enviar	
Asunto	Cambio 3 práctica
Adjunto	 preprocessing data aa cambio.py

Te envío el cambio que he hecho en el código en la línea 3 introduciendo el for.



Herramientas para control de versiones: git y github

- El control de versiones ayuda a los desarrolladores llevar un registro y administrar cualquier cambio en el código de un proyecto de software
- En los sistemas de control de versiones existe un registro de todos los cambios realizados por todos los miembros que desarrollan código en el proyecto
- Se pueden unir diferentes versiones de códigos para que todo pueda fluir y se reduzca el margen de error



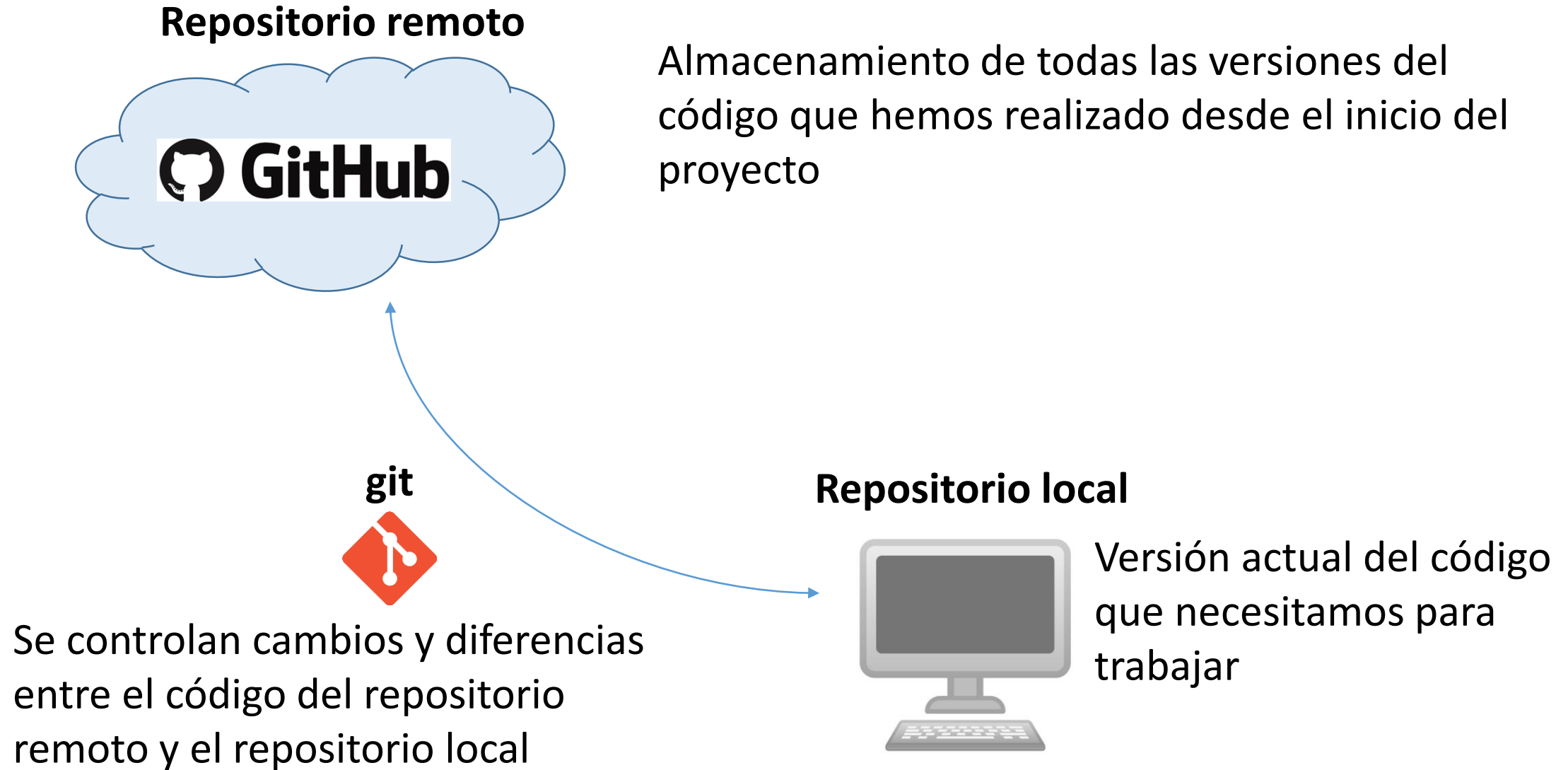
Sistema de control de versiones

+



- **Servicio en la nube** que ayuda a llevar un registro, almacenamiento y control de cualquier cambio del código desarrollado en un proyecto
- Gran comunidad de código abierto

Flujo de uso de git y github



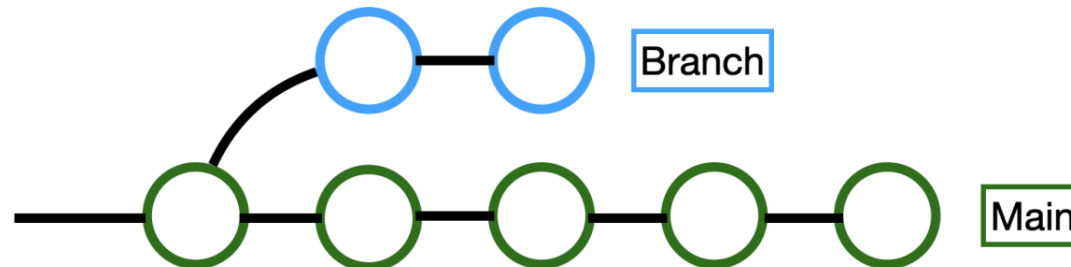
Vocabulario esencial de git

Repositorio

- Un repositorio es un almacenamiento virtual del proyecto
- Permite guardar versiones del código a las que puedes acceder cuando lo necesites

Rama

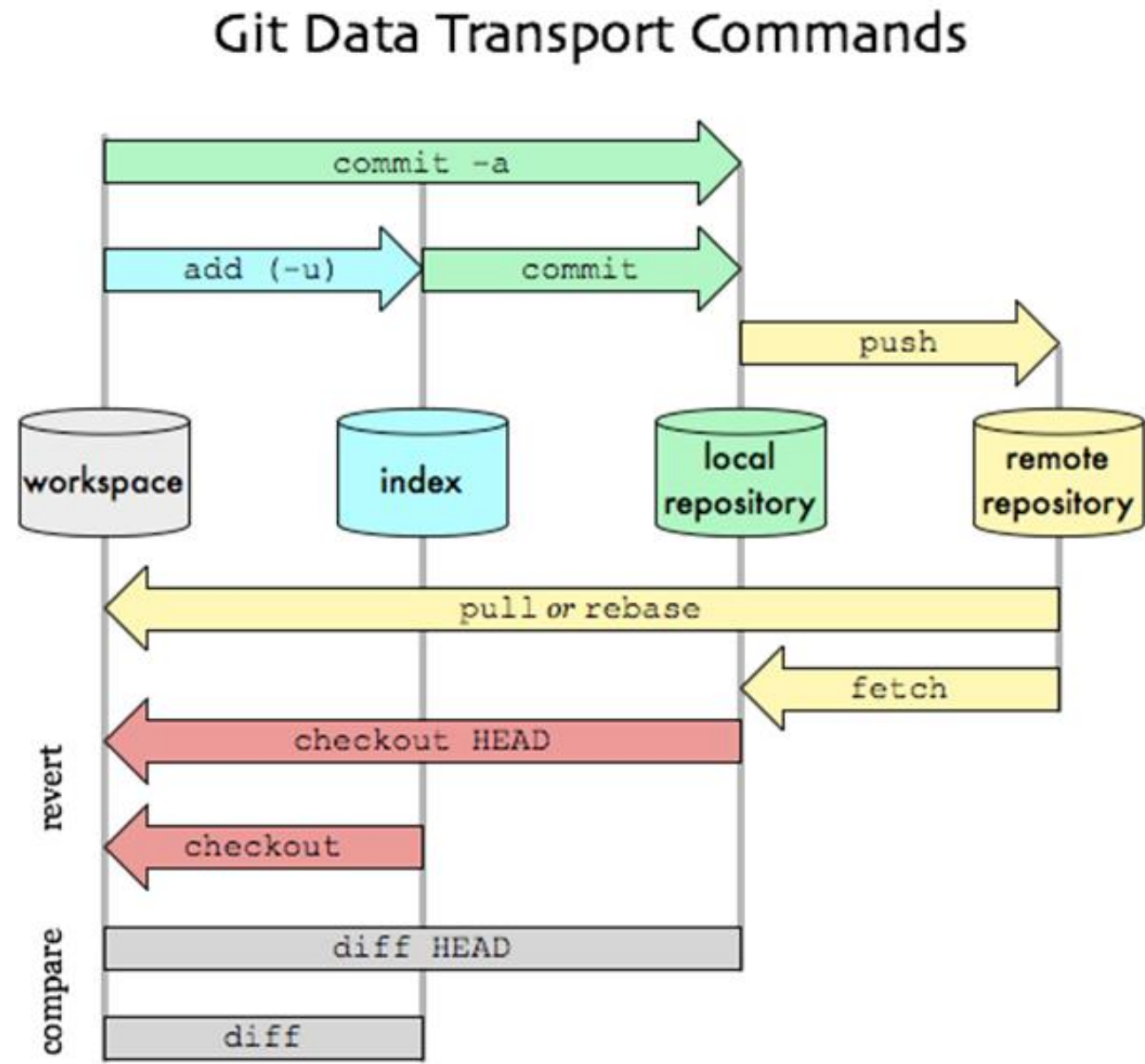
- Una rama es un entorno de desarrollo independiente donde se genera un nuevo área de trabajo (o versión del repositorio)
- En la nueva rama, se pueden realizar pruebas sin afectar al entorno original



PRÁCTICA

1. Creación de un repositorio en github
2. Clonar el repositorio desde la consola de git al ordenador local
3. Rellenar el README con el contenido que va a tener el proyecto
4. Subir al repositorio remoto de github los cambios
5. Ir subiendo las prácticas a github

Flujo entre repositorios git



Entornos de Python

- Un entorno de Python es una herramienta para mantener espacio separado para un proyecto con sus dependencias y librerías
- Este entorno es específico para el proyecto particular y no interfiere con las dependencias de otros proyectos.

Proyecto1: entorno_a

Pandas==2.1.1
Numpy==1.8.1
Tensorflow==2.0

Proyecto2: entorno_b

Pandas==2.1.1
Numpy==1.8.7
BeutifuSoup==3.2
Nltk==0.0.2

Proyecto3: entorno_c

Pandas==2.1.1
Numpy==1.8.1
Tensorflow==2.1.2

PRÁCTICA

1. Creación de un entorno de python
2. Instalar 5 librerías
3. Instalar un kernel asociado al entorno en conda
4. Ver en un notebook de jupyter las librerías instaladas
5. Realizar pip freeze > requirements.txt