

DATA SCIENCE TOOLKIT

Ejercicios Git, GitHub y
Markdown

Adolfo Sanz De Diego

1 Acerca de

1.1 Autor

- **Adolfo Sanz De Diego**
 - Blog: asanzdiego.blogspot.com.es
 - Correo: asanzdiego@gmail.com
 - GitHub: github.com/asanzdiego
 - Twitter: twitter.com/asanzdiego
 - LinkedIn: in/asanzdiego
 - SlideShare: slideshare.net/asanzdiego

1.2 Licencia

- **Este obra está bajo una licencia:**
 - Creative Commons Reconocimiento-CompartirIgual 3.0

1.3 Fuente

- Las slides y sus fuentes las podéis encontrar en:
 - <https://github.com/asanzdiego/curso-git-github-markdown-2015>

2 Ejercicios

2.1 Repositorio campusciff

1. Crear un repositorio en vuestro GitHub llamado **campusciff**.
2. Clonar vuestro repositorio en local.

2.2 README

1. Crear (si no lo habéis creado ya) en vuestro repositorio local un documento **README.md**.

*Notas: en este documento tendréis que ir poniendo los **comandos** que habéis tenido que utilizar durante todos los ejercicios y las **explicaciones y capturas de pantalla** que consideréis oportunas.*

2.3 Commit y push inicial

1. Añadir los comanddos utilizados hasta ahora y hacer un coomit inicial con el mensaje **commit inicial**.
2. Subir los cambios al repositorio remoto.

2.4 Ignorar archivos

1. Crear en el repositorio local un fichero llamado **privado.txt**.
2. Crear en el repositorio local una carpeta llamada **privada**.
3. Realizar los cambios oportunos para que tanto el archivo como la carpeta sean ignorados por git.

2.5 Crear el tag v0.1

1. Añadir fichero **1.txt** al repositorio local.
2. Crear un tag **v0.1**.
3. Subir los cambios al repositorio remoto.

2.6 Crear una rama remota v0.2

1. Crear una rama remota **v0.2**.
2. Posiciona tu carpeta de trabajo en esta rama.
3. Añadir un fichero **2.txt** en la rama **v0.2**.
4. Subir los cambios al repositorio remoto.

2.7 Merge directo

1. Posicionarse en la rama **master**.
2. Hacer un merge de la rama **v0.2** en la rama **master**.

2.8 Merge con conflicto

1. En la rama **master** poner **Hola** en el fichero **1.txt** y hacer commit.
2. Posicionarse en la rama **v0.2** y **Adios** en el fichero "1.txt" y hacer commit.
3. Posicionarse de nuevo en la rama **master** y hacer un merge con la rama **v0.2**
4. Listar las ramas con merge y las ramas sin merge.
5. Arreglar el conflicto anterior y hacer un commit.

2.9 Borrar rama

1. Crear un tag **v0.2**
2. Listar los distintos commits con sus ramas y sus tags.
3. Borrar la rama **v0.2**

2.10 Cuenta de GitHub

1. Poner una foto en vuestro perfil de GitHub.
2. Poner el doble factor de autenticación en vuestra cuenta de GitHub.
3. Añadir (si no lo habéis hecho ya) la clave pública que se corresponde a tu ordenador.

2.11 Uso social de GitHub

1. Preguntar los nombres de usuario de GitHub de tus compañeros de clase, búscalos, y sigueles.
2. Seguir los repositorios **campusciff** del resto de tus compañeros.
3. Añadir una estrella a los repositorios **campusciff** del resto de tus compañeros.

2.12 Crear una tabla

1. Crear una tabla de este estilo en el fichero **README.md** con la información de varios de tus compañeros de clase:

NOMBRE	GITHUB
Nombre del compañero 1	enlace a github 1
Nombre del compañero 2	enlace a github 1
Nombre del compañero 3	enlace a github 3

2.13 Colaboradores

1. Poner a github.com/asanzdiego como colaborador del repositorio **campusciff**

2.14 Crear una organización

1. Crear una organización llamada **campusciff-tunombredeusuariodegithub**

2.15 Crear equipos

1. Crear 2 equipos en la organización **campusciff-tunombredeusuariodegithub**, uno llamado **administradores** con más permisos y otro **colaboradores** con menos permisos.
2. Meter a github.com/asanzdiego y a 2 de vuestros compañeros de clase en el equipo **administradores**.
3. Meter a github.com/asanzdiego y a otros 2 de vuestros compañeros de clase en el equipo **colaboradores**.

2.16 Crear un index.html

1. Crear un index.html que se pueda ver como página web en la organización.

2.17 Crear Pull-requests

1. Hacer 2 forks de 2 repositorios **campusciff-tunombredeusuariodegithub.io** de 2 organizaciones de las que no seais ni administradores ni colaboradores.
2. Crearos una rama en cada fork.
3. En cada rama modificar el fichero **index.html** añadiendo vuestro nombre.
4. Con cada rama hacer un pull-request.

2.18 Gestionar Pull-requests

1. Aceptar los pull-request que lleguen a los repositorios de tu organización.