

2021-09-08

Anton-Bravo, Adolfo

## Contents

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Contenidos</b>   | <b>1</b>  |
| 1.1      | Tres saberes implicados en Periodismo de Datos . . . . .      | 1         |
| 1.2      | Uso de herramientas o software libres . . . . .               | 2         |
| 1.3      | Git . . . . .   | 3         |
| 1.4      | Github . . . . .  | 4         |
| 1.5      | Datos . . . . .   | 5         |
| 1.6      | Markdown . . . . .  | 5         |
| 1.7      | Pero, ¿por qué tantas tecnologías y términos? . . . . .       | 6         |
| 1.7.1    | Programación literaria e investigación reproducible . . . . . | 6         |
| 1.7.2    | Content as Code . . . . .                                     | 7         |
| 1.7.3    | COPE . . . . .  | 7         |
| 1.8      | Manos a la obra: Github . . . . .                             | 7         |
| 1.8.1    | Editar . . . . .  | 8         |
| 1.8.2    | Crear archivo . . . . .                                       | 9         |
| <b>2</b> | <b>Para la próxima sesión...</b>                              | <b>9</b>  |
| 2.1      | Tener instalada el emulador de terminal . . . . .             | 9         |
| 2.2      | Cosas que nunca te dije . . . . .                             | 9         |
| <b>3</b> | <b>Pruebas</b>  | <b>10</b> |

## 1 Contenidos

### 1.1 Tres saberes implicados en Periodismo de Datos

A propósito de una pregunta sobre "¿quiénes somos, de dónde venimos y a dónde vamos?", recordamos los tres saberes implicados en periodismo de datos:

1. Periodismo
2. Visualización
3. Datos

Los tres son importantes pero ojo que la visualización va más allá de la visualización como producto final. En la etapa de visualizar también realizamos análisis, aplicamos técnicas estadísticas, programas informáticos que hagan que de un gran volumen de datos seamos capaces de sacar hipótesis o conclusiones.

## **1.2 Uso de herramientas o software libres**

Cuando se habla de herramientas libres, de software, se refiere al hecho de que tengan licencias de software libres y/o abiertas.

- Sin abordar una vieja polémica entre software libre vs software de código abierto, hablamos aquí de todas ellas como un conjunto.
- Tened en cuenta que el software, en España/Europa, tiene consideración de obra inmaterial y tiene los mismos derechos que el Copyright de las obras escritas.
- Cuando escribimos no hace falta señalar el Copyright, lo tenemos por ley por defecto, de manera predeterminada.
- Se pueden resumir los usos sobre una obra inmaterial en cuatro:
  1. Uso. Lectura de un libro o lectura de un software.
  2. Distribución. Dejar prestado el libro, texto o software.

3. Modificación. Poder adaptar el texto, libro o software.

4. Redistribución. Poder distribuir esa adaptación

- El software libre y las licencias Creative Commons (utilizadas en producción cultural) "juegan" con esos derechos del Copyright para cederlos.
- Por ejemplo, eldiario.es tiene una licencia Creative Commons BY-NC, es decir, se puede usar su contenido pero hay que citar su procedencia y autoría (BY) pero no se puede hacer esto de forma comercial (NC, Non Commercial).
- Hay otras licencias como las consideradas "AntiCopyright" de BSD ya que solo con mencionar la procedencia permitían la adaptación y redistribución total de las obras derivadas. Esto es lo que ocurrió con parte del sistema operativo MacOSX que está basado en OpenBSD (como siempre, en los resúmenes se pierden detalles importantes, en StackExchange hay un hilo sobre esto).
- Con el software libre no solo se pueden hacer las cosas que el software permita sino que, si avanzamos en el conocimiento del propio software, podemos participar de su creación.

### 1.3 Git

- Git se utiliza en proyectos de software. Es un programa para el trabajo colaborativo y distribuido.
- También se utiliza en proyectos de periodismo y visualización de datos.
- Hay una visión del periodismo de datos denominada "Content as Code" que encaja con este uso.

- En cualquier caso, tiene muchas ventajas. Los únicos inconvenientes pueden venir de no conocer bien cómo funciona.
- Permite ramas de los proyectos que luego pueden integrarse.
- Permite volver a momentos concretos de los proyectos.
- Nos acostumbra a trabajar comentando lo que hacemos.
- Nos habituamos a un software cliente/servidor.
- Permite más de un servidor.
- Lo crea Linus Torvalds, el mismo creador del kernel Linux (de los sistemas operativos GNU/Linux) precisamente para el desarrollo del kernel, de esta parte fundamental del software de un sistema operativo.

## 1.4 Github

- Dado que git nace de los proyectos de software conviene saber qué tres piezas suelen tener estos proyectos, aunque a veces esté todo resumido en el propio archivo del software:
  1. El archivo Readme, en texto plano, donde se explica de qué va el software, cómo surgió, quién/es están detrás de él, cómo se usa, etc.
  2. La licencia del software.
  3. El propio software
- Mis notas de Github las podéis encontrar aquí.

## 1.5 Datos

- Volviendo la mirada sobre los datos, recordad que cuando hablamos de datos no nos referimos solo a unos datos estructurados o a un conjunto de datos.
- Veremos datos de todo tipo y también tipos de datos.
- Pero pensad que se tratan de registros electrónicos, donde puede haber datos estructurados pero también no estructurados, el texto y/o la web como datos, la música, las imágenes, los vídeos...
- E incluso nuestro comportamiento, tanto con las aplicaciones como con los dispositivos electrónicos.
- Se pregunta si en una página se puede seguir cuando pinchas en un enlace y cuando no... se puede seguir todo, es decir, se registra todo nuestro comportamiento, se puede reproducir, se hacen pruebas constantes con lo que hacemos, con nuestras decisiones, nuestros estímulos... técnicas A/B, etc.

## 1.6 Markdown

- En el mundo de los datos, el tamaño importa.
- La Web funciona con HTML, actualmente la versión 5: HTML5
- HTML está muy bien pero Dan Gruber pensó que tenía un inconveniente: era difícil de leer el texto de un código HTML.
- Por eso inventó Markdown. Markdown es dos cosas a la vez:
  1. Una sintaxis simple. Veréis que es muy, muy simple.

## 2. Un "parseador" o conversor de esa sintaxis en HTML

- Así, en Github escribimos Markdown y Github lo muestra como HTML, pero si vemos el archivo fuente sigue siendo Markdown. Tiene truco, claro. Si le das a "ver código fuente" lo ves en HTML pero ese HTML no es el que modificas para que se vea un HTML u otro sino que lo que modificas es Markdown y luego se hace la conversión de nuevo.
- Markdown es tan simple que hay sitios donde no llega. Por eso desde el inicio se permite en Markdown usar también lenguaje HTML si lo necesitamos.
- También hay versiones de Markdown como Markdown BlackFriday o RMarkdown en R.
- Hay muchas guías de Markdown. Lo mejor es probar y aprender.
- Mi guía la podéis encontrar en github.

### 1.7 Pero, ¿por qué tantas tecnologías y términos?

- Veamos, volvamos al principio. Estamos en Periodismo de Datos. Sí, podríamos tener una pequeña tabla, ir a un servicio de terceros y hacer un gráfico. Fácil. De hecho, lo haremos.
- Parémonos un momento en eso que dijimos del *Computer Assisted Reporting*, periodismo asistido por ordenador. Sí, esto es fundamental, debemos saber manejar un ordenador.

#### 1.7.1 Programación literaria e investigación reproducible

- Ya en programación Donald Knuth, inventor del fundamental  $\text{\LaTeX}$ , pieza clave en la investigación reproducible, lo hizo pensando en el

paradigma de la "programación literaria".

- Es decir, como vimos con HTML, todo lenguaje informático tiene "comentarios" sobre el código.
- Estos comentarios son útiles para recordarme por dónde voy, por qué he hecho tal cosa si tal otra falló, lo que quiero hacer, etc.
- La programación literaria propone hacerlo "al revés". Es decir, escribir lo que quiero hacer y los trozos que sean código identificarlos como tal.

### 1.7.2 Content as Code

- Tanto Github como Markdown como otras tecnologías están alineadas con algo que se ha denominado "el contenido como código".
- En el mundo del periodismo de datos se utiliza mucho Python con Jupyter; R con RStudio; Javascript con ObservableHQ; etc. Los veremos en otro momento.
- El uso de sintaxis simple nos permite reutilizar el texto, nuestro código, y también insertar código.

### 1.7.3 COPE

- En la radio pública de EE.UU. desarrollaron la estrategia *COPE: Create Once, Publish Everywhere* para reutilizar contenidos de las distintas emisoras.

## 1.8 Manos a la obra: Github

- Entramos en Github

- Creamos un repositorio nuevo
- Lo llamamos, por ejemplo, "uc3m-periodismo-datos". Es importante en este momento que, si no lo hacíais hasta ahora, empecéis a no usar espacios en blanco para los nombres de archivo y directorios/carpetas :boom:
- Sí que inicializamos el repositorio con un archivo que se llamará "README.md". El ".md" significa que tiene sintaxis Markdown.
- Finalizamos el proceso.
- Ahora tenemos un archivo README.md que contiene una sola línea "# uc3m-periodismo-datos". Esto es porque Github ha tomado el nombre del repositorio y lo ha convertido en el título o H1 de HTML en este documento.

### 1.8.1 Editar

Vamos al lápiz, pinchamos y editamos. Escribimos:

```
language=markdown,label=,caption=,captionpos=b,numbers=none Periodismo de Datos en UC3M
Notas sobre Periodismo de Datos en UC3M
Qué es el periodismo de datos - Periodismo - Visualización - Datos
HTTP Es una API que tiene cuatro tareas posibles : 1.POST2.GET3.DELETE4.PUT
```

- Veis que en el editor hay una pestaña a la derecha para previsualizar el texto en HTML.
- Si estamos de acuerdo, "guardamos".
- En Github, como es software git lo que hay detrás, no se "guarda" el documento sino que se hace un "commit", una explicación simple o extensa de lo que hemos hecho, por si tuviéramos que volver a este punto. Ahora sí, "comitemos".



- Si queremos volver a editar para cambiar algo, repetimos la operación.

### 1.8.2 Crear archivo

- Pinchamos en crear archivo y lo nombramos.
- Importante:
  - No usar tildes ni espacios en blanco ni caracteres que no sean ASCII, es decir, tampoco las eñes.
  - Si se trata de un archivo markdown tendrá una extensión "md", como el README, es decir, por ejemplo, "nuevo-archivo.md"

## 2 Para la próxima sesión...

### 2.1 Tener instalada el emulador de terminal

- En Windows hay que descargarse Cygwin
- En MacOSX hay que descargarse y activar XCode. En MacPorts hay una guía pero seguro que hay otras.

### 2.2 Cosas que nunca te dije

- Arquitectura cliente/servidor
- Periodismo de bases de datos
- Mostrar D3js.
- Crear una cuenta en Datawrapper.

- Explicar JSON como formato de datos.
- Explicar formatos de datos.
- Licencias de datos
- Bibliografía
- Comunidades
- Ejemplos
- Taxonomías de visualización
- Periodismo de datos como ciencia de datos.
- La pirámide invertida del periodismo de datos.

### **3 Pruebas**

- Qué saberes están implicados en periodismo de datos. Razona la respuesta.
- Cuál es la materia del periodismo de datos. Razona la respuesta.
- Ahora que ya tenemos un primer repositorio en Github y que sabéis crear archivos, os propongo que escribáis un comentario crítico sobre una o varias visualizaciones de datos y/o infografías.

– No se exige un número mínimo de palabras ni máximo.

- Puede ser de una o de varias.
- Lo interesante no es que me enseñéis la que pensáis que puede ser para mí la mejor visualización o infografía sino que escribáis libremente, sin prejuicios.
- Importante, repito: este texto tiene que estar en formato Mark-down en un documento creado en el mismo repositorio de Github que habéis creado.
- Si la/s visualizaciones o infografía/s tienen URL, podéis añadirlas.
- Cualquier duda la podemos resolver también por el foro.