

# Escritura colaborativa en la nube sobre clones de plantilla

Arturo Jaime<sup>1</sup>, José Miguel Blanco<sup>2</sup>, César Domínguez<sup>1</sup>, Ana Sánchez<sup>2</sup>

<sup>1</sup>Dto. de Matemáticas y Computación, Universidad de La Rioja, Logroño. <sup>2</sup>Dto. de Lenguajes y Sistemas Informáticos, Universidad del País Vasco/Euskal Herriko Unibertsitatea, Donostia-San Sebastián.  
arturo.jaime@unirioja.es, josemiguel.blanco@ehu.es, cesar.dominguez@unirioja.es,  
ana.sanchez@ehu.es

## Resumen

En los estudios de ingeniería informática es habitual la escritura colaborativa de memorias como colofón a los trabajos prácticos. Esta actividad exige algo más que utilizar un buen editor en la nube, ya que son bien conocidas las dificultades de los estudiantes al redactar, tanto individualmente como en equipo. Las plantillas de documentos se utilizan mucho, tanto en la empresa como en los trabajos prácticos universitarios. Estas facilitan la comprensión del alcance y tamaño de la tarea a realizar. Además, aportan un andamiaje muy interesante que ayuda a estructurar documentos complejos haciendo que los resultados sean más comprensibles. Por tanto, facilitan la escritura individual y colaborativa. Las plantillas se pueden complementar con ejemplos destacables realizados por estudiantes o revisados por profesores. Para promover una escritura colaborativa real aportamos a nuestros equipos de estudiantes clones, o copias, de plantillas en la nube. Si necesitamos manejar muchos clones de forma manual la tarea asociada nos puede hacer desistir. En el caso de los documentos de Google Drive se pueden programar funciones sencillas en Google Apps Script, de forma que se automaticen las tareas de gestión de los clones. Además, gracias al historial de versiones, podemos conocer cuánto y cuándo han contribuido los escritores de un documento.

## Abstract

In computer science studies, it is common to write collaborative reports as the culmination of practical works. This activity requires more than just using a good editor in the cloud, as students' difficulties in writing, both individually and in teams, are well known. Document templates are widely used, both in the industry and in university practical works. They make it easier to understand the scope and size of the task to be performed. In addition, they provide a very interesting scaffolding that helps to structure complex

documents, making the results more comprehensible. Thus, they facilitate individual and collaborative writing. The templates can be complemented with relevant examples made by students or reviewed by teachers. To promote real collaborative writing we provide clones, or copies, of templates in the cloud to our student teams. If we need to handle many clones manually the associated task may discourage us. In the case of Google Drive documents, simple functions can be programmed in Google Apps Script to automate clone management tasks. In addition, thanks to the version history, we can know how much and when the writers of a document have contributed.

## Palabras clave

Escritura colaborativa, andamiaje, plantillas, google apps script.

## 1. La escritura colaborativa

Algunos autores destacan que la escritura colaborativa consigue resultados de mayor calidad, aumenta la productividad y mejora las relaciones interpersonales [7, 9, 11].

Diferentes experiencias de escritura colaborativa identifican las dificultades que tienen los estudiantes al redactar, organizar contenidos y sintetizar su trabajo [3, 8, 13]. Estos problemas no se consiguen superar con una actividad docente aislada, aunque puede contribuir a mejorar la capacidad de comunicación escrita. Dice Joe Miró que aprender algo escribiendo sobre ello exige buena dosis de paciencia y trabajo, no solo del alumno, sino también del profesor [8]. También destaca que uno sólo escribe bien de lo que sabe, y pudo observar la tendencia en muchos estudiantes a circunvalar problemas centrándose en los puntos más simples de los temas abordados.

La escritura colaborativa exige de los equipos algo más que fragmentar la redacción entre los miembros y concatenar lo que entrega cada uno. En este sentido,

algunos estudios observan que los estudiantes son reacios a cuestionar o simplemente reflexionar sobre las ideas de sus compañeros [7]. La tecnología no resuelve todas las dificultades y es importante conocer y usar un modelo útil de trabajo colaborativo.

Una propuesta habitual para la escritura colaborativa es el uso de una plataforma en la nube como Google Drive o Microsoft OneDrive [9, 12, 13]. Con la ayuda de gráficas automáticas, que reflejan la participación de cada escritor durante el proceso de redacción, Yim *et al.* distinguen cuatro estilos de escritura colaborativa [12]. En el estilo *escritor principal* el documento lo escribe una persona con aportaciones secundarias de otros. En el estilo *divide y vencerás* cada escritor elabora su propio trozo de documento, sin retocar lo escrito por los demás. El estilo de *revisión cooperativa* se parece al anterior pero los escritores modifican lo escrito por otros, sobre todo en la fase final. Por último, en el estilo de *trabajo simultáneo*, se escriben frases conjuntas, se elaboran trozos de forma simultánea y se retoca el trabajo de los demás, sobre todo al principio y al final. Los estilos más habituales son la *revisión cooperativa* y el *escritor principal* y la mejor calidad se obtiene con el *divide y vencerás* [12]. A pesar de su interés, en el presente trabajo descartamos la realización de un análisis del estado del arte de la escritura colaborativa.

## 2. Las plantillas de documentos

Una plantilla es un esquema que sirve de guía para redactar un documento de calidad, proporciona secciones obligatorias y opcionales que completar e incluye comentarios intercalados que explican el objetivo y los contenidos a incluir [3]. Un estudiante de gestión de proyectos escribió una lección aprendida sobre las ventajas de las plantillas de proyecto [5]. Destacaba que las plantillas agilizan la elaboración de documentos, recordando los puntos a incluir, ayudando a concentrarse en lo importante y aportando uniformidad, que facilita las posteriores búsquedas. Además, su uso permite incorporar en sus sucesivas versiones mejoras relacionadas con el método de trabajo.

Las plantillas de documentos proporcionan cierto andamiaje (*scaffolding*) a los estudiantes que les facilita la comprensión del alcance y tamaño de la tarea. Además les ayuda a organizar y explicar mejor las tareas. El andamiaje combina el aprendizaje mediante ejemplos con el aprendizaje mediante la práctica [3, 7]. Las plantillas ayudan a estructurar documentos extensos o complejos, como una memoria de trabajo fin de grado, haciendo el resultado más comprensible. La estructura común

facilita la revisión de documentos escritos por otros compañeros y mejora las interacciones y la comunicación en el equipo, constituyendo una base común de entendimiento [1].

Las plantillas se pueden complementar con ejemplos ilustrativos. En el caso de trabajos fin de grado (TFG) pueden ser secciones destacables extraídas de memorias de otros estudiantes, como unos antecedentes o unas conclusiones bien escritas.

También se derivan algunos inconvenientes del uso de plantillas [3]. Muchos estudiantes evitan leer los comentarios de la plantilla y, cuando lo hacen, son reacios a preguntar por lo que no entienden. Como ocurre con los libros de texto, aunque la plantilla esté muy bien escrita y comentada, es necesario que los profesores expliquen a sus estudiantes cómo utilizarla [3]. Incluso con buenas plantillas habrá estudiantes con dificultades para realizar su trabajo de redacción. Por otra parte, cuando se escribe un documento en equipo, los estudiantes tienden a pensar que sus compañeros han entendido lo mismo que ellos y no ven la necesidad de discutir sobre los contenidos a elaborar y su ubicación más razonable [3].

## 3. La memoria del TFG

Diferentes autores parecen asociar al tutor de TFG un papel de mero verificador de la calidad de la memoria sin jugar ningún papel activo durante el proceso de redacción [6, 10]. Sin embargo, como decíamos anteriormente, muchos estudiantes tienen dificultades para escribir y organizar una memoria de cierto tamaño [3, 13]. Pensamos que puede tener sentido ver a la redacción de las memorias de TFG como un caso de escritura colaborativa donde el estudiante hace el rol de *escritor principal* (según los estilos identificados por Yim *et al.* [12]) y el tutor aporta comentarios, hace sugerencias y resuelve dudas. En los casos en los que el estudiante muestra muchas dificultades podría llegar a retocar algunas frases o incluso a escribir algún párrafo.

Este proceso de redacción supervisada se puede beneficiar del uso de editores en la nube. La memoria se puede ir redactando en un documento textual compartido entre estudiante y tutor. El tutor puede aportar un documento de partida con el formato y los estilos aceptados en la facultad o escuela (portada, fuentes, colores, etc.). Además, este documento puede incluir una plantilla con secciones obligatorias y optativas y con comentarios sobre los contenidos y estructura interna de cada una. También es interesante que la plantilla explique el orden de redacción más razonable, por ejemplo, se puede sugerir comenzar con los antecedentes y dejar para el final las

conclusiones, introducción y resumen. Esta plantilla puede ayudar al estudiante a superar bloqueos.

Lo ideal sería utilizar una plantilla consensuada entre el cuadro de profesores que participan en los tribunales evaluadores. Sin embargo, no es sencillo conseguir estos consensos. Cada uno suele tener su propio punto de vista sobre qué secciones interesan o qué contenidos son esenciales. También hay muchos tipos de proyecto que exigen su propia estructura. No es lo mismo un proyecto de desarrollo de una aplicación web, que uno de testeado de una aplicación, o de depuración de una fuente de datos (en inteligencia de negocio), por poner algunos ejemplos. Seguramente sería más fácil ponernos de acuerdo respecto a un estilo de redacción que sea claro, directo, correcto y breve.

Ya hemos explicado cómo puede ayudar a los estudiantes el andamiaje proporcionado por una plantilla al realizar esta tarea. Nosotros diseñamos hace dos cursos una plantilla sencilla para la elaboración de memorias y la hemos utilizado con unos pocos TFG con buenos resultados. Disponemos de algunas memorias recomendables, realizadas por estudiantes, que complementan la plantilla. Nuestra idea es seguir seleccionando memorias o secciones de memoria que ilustren nuevos tipos de proyecto.

El documento donde se escribe la memoria del TFG es inicialmente propiedad del profesor, quien comienza aportando un clon o copia de la versión con la plantilla. El estudiante adquiere el rol de escritor. Esta situación se puede reorientar y buscar la configuración que parezca más razonable. Trabajar de esta forma aporta algunas ventajas. Tutor y estudiante tienen siempre acceso a la última versión y a la rama de versiones que se generan de manera automática. Al consultar una versión se puede saber quién introdujo nuevas líneas o cambios, como se observa en la Figura 1. Cuando alguno hace comentarios el otro los recibe por email y los ve en el documento.

#### 4. Documentación en equipo

La realización de trabajos o pequeños proyectos es una práctica muy frecuente en las asignaturas de los grados de informática [3, 4, 6, 10]. Es habitual que los estudiantes entreguen una memoria breve donde se describan, por ejemplo, los análisis, diseños, implementaciones o problemas superados. Además, en muchas ocasiones, estas actividades se desarrollan en equipo, generalmente debido al elevado número de estudiantes que cursan la asignatura. Por tanto, es deseable que la memoria se redacte de forma colaborativa entre los miembros de cada equipo.

Una de estas asignaturas es la que introduce los conceptos de dirección de proyectos, en las que

habitualmente se realizan miniproyectos donde aplicar el ciclo de vida de un proyecto informático [4] y se tocan algunas de las áreas básicas del PMI. Dos de los autores de este artículo impartimos la asignatura sobre esta materia en nuestros respectivos centros universitarios. En cada miniproyecto los estudiantes elaboran al menos dos documentos, uno contiene la planificación y en el otro se va recopilando información de seguimiento y control.

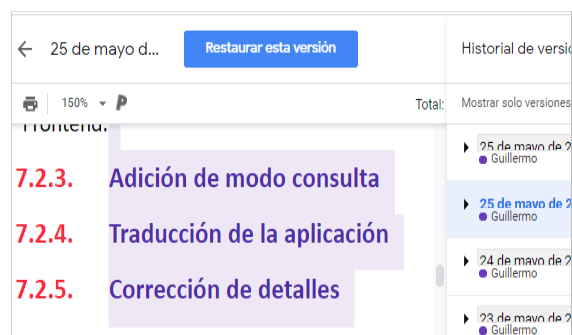


Figura 1: Sistema de versiones de Google Drive. Los colores marcan qué autor introdujo el cambio en una determinada versión del documento.

Hemos ido variando el modo de recopilar esta documentación. Los primeros cursos la pedíamos en papel para evitar su descarga desde una plataforma y la retroalimentación la escribíamos a mano sobre el entregable, que después mostrábamos a los interesados. Para hacernos una idea de las dimensiones de esta tarea de revisión, vamos a poner un ejemplo. En una de nuestras asignaturas del curso 19-20 participaron en las prácticas 39 estudiantes. Organizamos para ellos una serie de cuatro miniproyectos. El primero fue individual, el segundo lo realizaron distribuidos en catorce equipos, el tercero en doce equipos y el último en seis equipos. Cada miniproyecto individual o en equipo generó dos documentos. Esto da 142 documentos, de los cuales 64 se redactaron en equipo de forma colaborativa. Las páginas de los documentos varían. Los del primer proyecto tenían entre dos y tres páginas, mientras que los del último proyecto tenían entre seis y nueve. Además, si deseamos revisar versiones intermedias, todavía se generan más páginas. Esto último tiene sentido cuando queremos comprobar si se está haciendo seguimiento del miniproyecto. No sería raro que la tarea de seguimiento se redujera a redactar el documento final sin que haya habido ninguna actividad de seguimiento real. En conclusión, tendríamos demasiadas páginas para revisar impresas en papel, con lo que ello conlleva. Así que pasados dos cursos comenzamos a pedir los documentos en

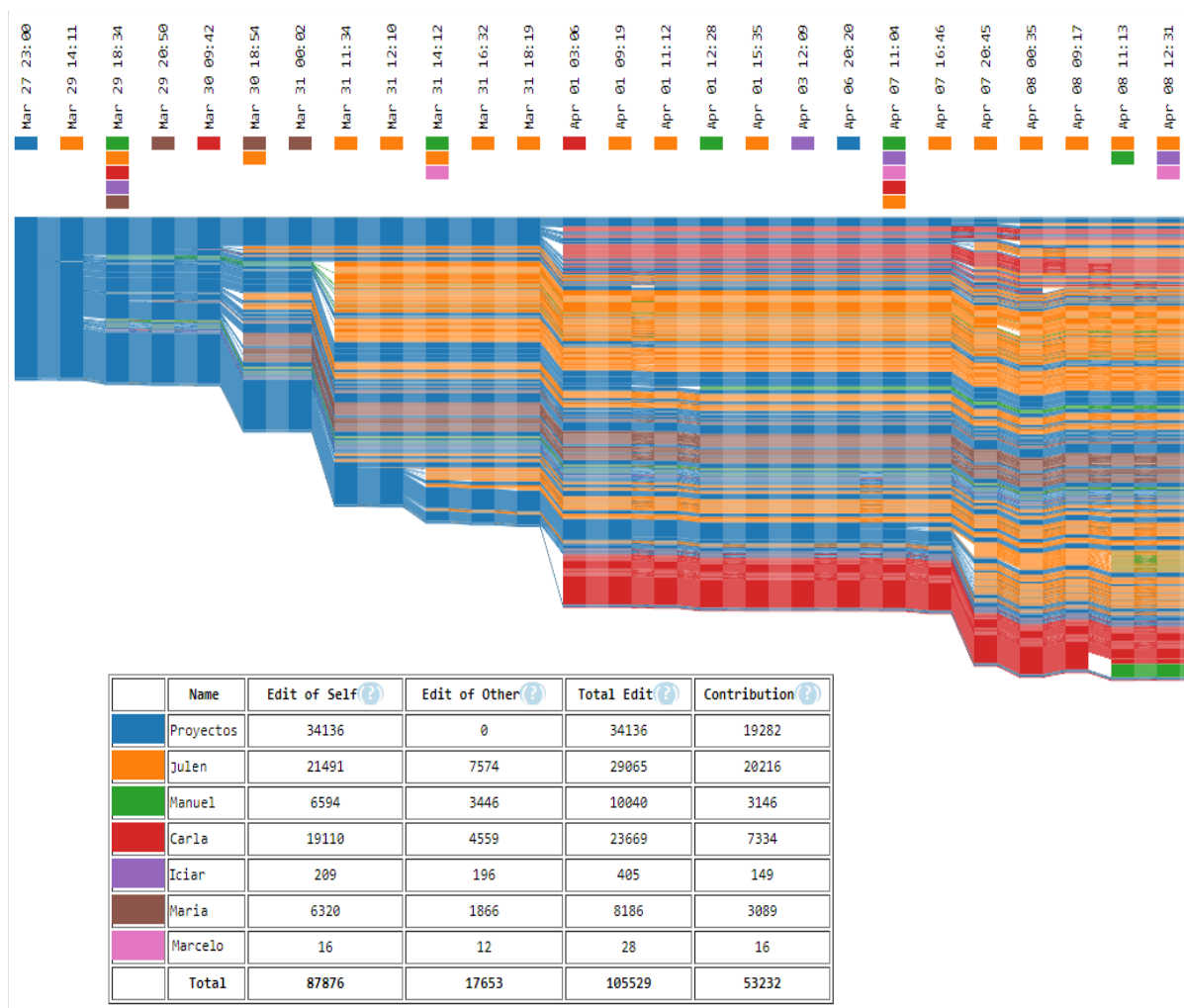


Figura 2: Gráfica docuViz de una escritura colaborativa de tipo “trabajo simultáneo” por un equipo de siete personas. Este estilo suele incluir bastantes modificaciones entre compañeros y sesiones simultáneas.

formato PDF y a buscar formas cómodas de corregirlos en pantalla.

Durante los primeros cursos dábamos unas pautas de los contenidos esperados en los documentos. Por ejemplo, para el documento de seguimiento y control del cuarto proyecto pedíamos aproximadamente lo siguiente: *contendrá las tareas realizadas, cuándo se hicieron y el tiempo dedicado. Incluirá un apartado con las actividades realizadas por cada miembro del equipo. También se comparará lo que se ha hecho con lo planificado, explicando las razones de las desviaciones significativas. Esto último se hará tanto para el conjunto del equipo como para cada uno de sus miembros. Se incluirán los aspectos oportunos en relación al seguimiento de comunicaciones, calidad, riesgos, cambios y adquisiciones.* Los documentos

recogidos incluían algunos de estos puntos. Pero lo sorprendente es que muchos equipos, incluyendo aquellos formados por buenos estudiantes, “olvidasen” partes de lo solicitado, por breve que parezca la descripción anterior. Esto nos llevó a facilitarles, a partir del curso 14-15, una versión muy simple de plantilla para cada documento, que incluía básicamente las secciones obligatorias. Con el paso de los cursos, y a la vista de las entregas que corregimos, hemos ido mejorando estas plantillas, que actualmente cuentan con comentarios, tablas y figuras personalizables, textos introductorios genéricos, etc.

Hay, al menos, dos puntos positivos derivados de esta organización. El primero es que con las plantillas nadie se “olvida” de presentar algo que le estemos

pidiendo y que las entregas contienen generalmente la información que pedimos. El segundo es que la corrección es mucho más fácil, ya que los documentos que revisamos, correspondientes a una misma entrega, siguen una estructura común. Además, los equipos plantean dudas (sobre todo a última hora) respecto a los contenidos de alguna sección, a pesar de los comentarios de la plantilla, en lugar de saltárselo, como hacían antes.

Por otra parte, las plantillas eran inicialmente documentos ODT o DOCX. Esto llevaba a que los documentos en elaboración no se compartieran en el equipo hasta conseguir una versión muy avanzada o definitiva. Tampoco es algo que nos extrañase, ya que nosotros mismos hacíamos cosas parecidas al escribir artículos de investigación. Pero, a partir del curso 17-18, pensamos que debíamos dar un impulso definitivo a la escritura colaborativa y compartimos clones de las plantillas exclusivamente como documentos textuales de Google Drive. Además, obligamos a los equipos a redactar sobre los propios clones hasta llegar a la versión final. Elegimos esta plataforma porque es accesible, tiene muchas herramientas interesantes y sobre todo porque dispone de un lenguaje de *script* que nos permite desarrollar programas para manipular documentos. Entendemos que cualquier otra plataforma de características similares serviría para conseguir idénticos propósitos.

Algunos trabajos [12] utilizan trazas automáticas, como la que se muestra en la Figura 2, que representan la colaboración entre los escritores de un documento. Esta traza ha sido generada por una extensión de Google Chrome denominada docuViz [11]. El gráfico distingue a los escritores por colores. Los ejemplos de las Figuras 2 y 3 corresponden a los tipos de colaboración “trabajo simultáneo” y “escritor principal” en dos equipos de siete personas cada uno. Como se puede observar en los ejemplos, varios miembros no hacen ninguna contribución. En ambos casos, el primer autor (de color azul) representa al profesor, ya que es quien ha elaborado la plantilla inicial. En las cuadrículas inferiores se asocia a cada escritor el color que lo representa. Cada barra vertical oscura de la gráfica corresponde al documento en un punto temporal, es decir a una versión. En la parte de arriba aparece la fecha y hora de la versión y el color de los escritores que participaron en ella. Cada fragmento de barra del mismo color corresponde a un grupo de líneas consecutivas escritas por la misma persona. La altura de los fragmentos representa el número de líneas. Dentro de una barra vertical se respeta el orden de las líneas en el documento, las de arriba representan las primeras líneas y las de abajo las últimas. Los colores se reparten de esa forma

porque están representando en qué parte del documento ha participado cada autor. Las líneas que unen las barras, están conectando a las líneas que se mantienen sin eliminar del documento y así se pueden apreciar las inserciones y borrados entre versiones del documento. El recuadro inferior contiene información de tipo estadístico. Para cada escritor figura el número de líneas que editó, modificó de otros escritores y el total de líneas en las que contribuyó.

La gráfica que presenta docuViz es muy interesante para comprobar quién ha contribuido, en qué medida, cuándo lo ha hecho y en qué zona aproximada del documento. Pero, al utilizar la herramienta hemos podido comprobar que necesita bastante memoria, que es algo lenta y que si no dispone de suficiente memoria se bloquea en la fase de carga de datos.

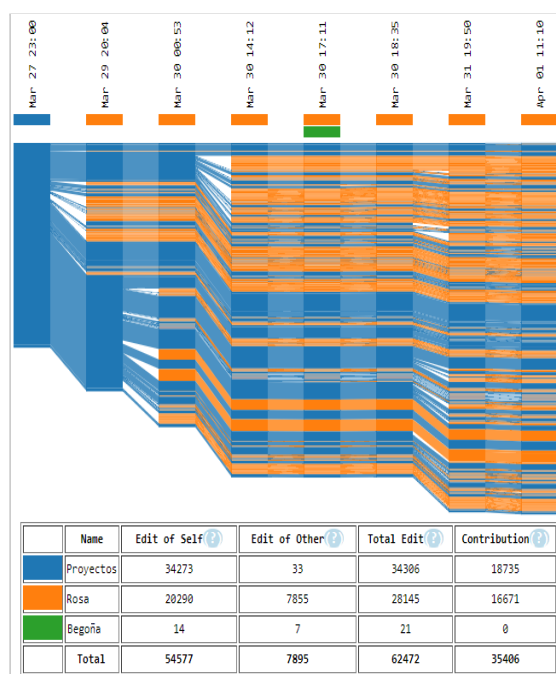


Figura 3: Escritura colaborativa de tipo “escritor principal” en un equipo de siete personas.

## 5. Clones en Google Drive

Clonar plantillas elaboradas por profesores y asignarlas a equipos, o a estudiantes que escribirán su memoria del trabajo de fin de grado, nos pareció una buena idea para impulsar la escritura colaborativa sobre una estructura de documento común. Pero, recordemos que en el ejemplo del curso 19-20 necesitamos crear 142 clones. Cada clon habrá que compartirlo, por ejemplo, con el director del equipo. Hacerlo de forma manual, como explicábamos para el caso de los TFG, es una tarea larga y tediosa. Después, se necesita abrir cada uno para corregirlo,

quizá descargar un PDF, etc. Afortunadamente, en Google Drive contamos con un lenguaje de *scripting* denominado Google Apps Script (GAS) [2]. Se trata de una API que se invoca desde código JavaScript y permite realizar de forma automatizada, por ejemplo, tareas de edición en documentos.

Para automatizar la gestión de nuestros clones codificamos cuatro *funciones*. Los datos necesarios para ejecutar las funciones los almacenamos en hojas de cálculo. La Figura 4 muestra la información para clonar un documento (en formato JSON) situada en una celda de una hoja de cálculo. Hay tres identificadores de google, “userList” es el de una libreta de cálculo con información de los equipos, “template” el identificador del documento textual a clonar y “folder” el de la carpeta donde quedarán los clones. Además contiene el nombre de la pestaña y columna (“userListTab” y “userListCol”) donde está la lista de usuarios con quienes se compartirán inicialmente los clones. La Figura 5 muestra la hoja de cálculo con la información de los equipos y, para cada equipo, el usuario de google con quien se compartirá el clon (semioculto por privacidad).

```
{
  "userList": "13X6aBPw34puKGIZyCdNn62vt4sxcKA",
  "userListTab": "P2",
  "userListCol": "E",
  "template": "1Zd5b5aXUUuhj2qztbIMGBnE3WpVcrmkUo",
  "folder": "10NBUJo-mWJFIDUppO0mxmFDH1Npw9",
  "clonesPrefix": "PI20-E220-",
  "cloneShare": "google"
}
```

Figura 4: JSON para gestión de clones.

A	B	C	D	E	
Código	Mod	IdSecreto	google user_id	E220	E230
Puppis	Ca	730J	@gmail.com	1r545GeMGyOTHK1	
Orion	Ci	054B	@gmail.co	14CBpts5ACN_Z6Nr	
Monoceros	Do	073Z	@gmail.co	12jAje6rR0c818IKGT	
Cetus	G	404W	@gmail.com	1ENiuJ2TgwiRpIRBRK	
Pyxis	GM	984Q	er76@gmail	1gaA426lcVerjUfyx	
UrsaMajor	F	428V	4@gmail.cc	1IMagLqG95Lte8gW	
Lepus	In	800R	3@gmail.cc	11-srUFUo0610_jxr-t	
Columba	Mik	746M	1@gmail.co	1G3j_xAGquRc92VC	
Triangulum	Mu	190B	mail.com	1u9jBDbkcaFOmYD	

Figura 5: Información de los equipos e identificadores de los clones que se van generando

La primera de las funcionalidades es *generar clones*. La función crea un clon del documento que contiene la plantilla para cada equipo de la Figura 5.

Los usuarios de google con los que se comparte el clon son los directores de equipo, quienes lo compartirán después con sus compañeros. Los clones se crean en la carpeta “folder” del profesor. El nombre se construye con el prefijo “clonesPrefix” de la Figura 4 y el nombre de equipo de la Figura 5. Se asigna al director el rol de editor. En la hoja de cálculo de la Figura 5 se almacenan los identificadores google de los clones que se van generando (columna E). En la Figura 6 vemos cómo al seleccionar un documento y pulsar el icono de información (letra i en un círculo), Drive muestra los iconos de los usuarios que lo comparten. Al pasar el cursor por encima vemos el rol de cada uno (“lector” para el usuario de la M).

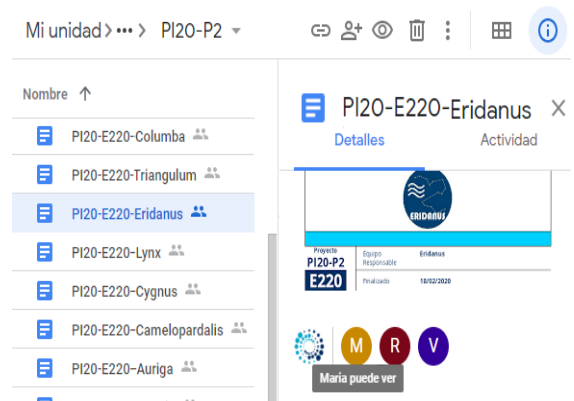


Figura 6: Usuarios que comparten un documento.

```
// Ejecutar Depuración cloneC

// Crea un clon de la plantilla
// json.template en la carpeta
// json.folder. El nombre del clon
// comienza con json.clonesPrefix y se
// completa con el parámetro team. La
// función devuelve el fichero clonado.
//
function cloneOne(team, json) {
  var folder =
    DriveApp
      .getFolderById(json.folder);
  var file =
    DriveApp
      .getFileById(json.template)
      .makeCopy(json.clonesPrefix+team,
                folder);
  return file;
}
```

Figura 7: Función simple con Google Apps Script



La segunda funcionalidad es *descargar instantáneas en PDF* del contenido de los clones de un determinado documento (el plan E220 en el ejemplo de las Figuras 4 y 5). La tercera funcionalidad es el *cambio de rol desde escritor a solo lector* (y viceversa) a todos los usuarios que tienen compartido cualquiera de los clones de un determinado documento. La última funcionalidad consiste en *insertar texto común en los clones* de un documento (tomando el texto de una celda de una hoja de cálculo) con retroalimentación general para todos los equipos. La retroalimentación específica para un documento se hace escribiendo comentarios o texto destacado directamente con el editor.

La Figura 7 muestra un ejemplo de una función sencilla que crea un clon (con `.makeCopy`) en la carpeta `json.folder` del documento con identificador `google json.template`. El nombre que recibirá el clon es la concatenación del prefijo `json.clonesPrefix` y el nombre del equipo, pasado como parámetro.

Además de las funcionalidades explicadas, nos gustaría disponer en el futuro de información de seguimiento sobre la participación de los componentes sin recurrir a soluciones externas y ligadas a un navegador concreto como `docuViz`.

El código GAS también tiene limitaciones que conviene tener presentes. Una de ellas es el tiempo máximo de ejecución, que no puede superar los seis minutos. En algunas ocasiones la generación de clones supera este tiempo y nos obliga a realizarla en varias fases. Es posible superar este problema, por ejemplo, mediante la definición de un disparador temporal (activador, en nomenclatura GAS) que genere en cada activación un grupo pequeño de clones y se desactive al crear todos ellos.

## 6. Conclusiones

Numerosos autores identifican las dificultades de los estudiantes, tanto al redactar como al comprender los contenidos solicitados, en las memorias de sus trabajos universitarios. Una forma de aliviar estos problemas es construir plantillas que guíen a los estudiantes en el alcance, contenido y longitud de los escritos. Nuestra experiencia de nueve años revisando documentos en asignaturas de dirección de proyectos (planificación, seguimiento y control...) nos permite afirmar que la incorporación y revisión de plantillas ha conllevado una mejora sustancial de la calidad de las entregas. Poco a poco hemos conseguido que sean más completas, más claras de entender y que contengan lo esperado. Creemos que esta percepción se confirmará en un estudio de tipo cuantitativo a realizar como trabajo futuro.

Además, muchos de los trabajos del ámbito de los grados de ingeniería informática se realizan en equipo. Hace ya tiempo están disponibles buenas herramientas para la escritura colaborativa en la nube. Aún así, animar a utilizarlas en nuestras asignaturas no fue suficiente y sólo se empezaron a usar cuando empezamos a obligarlo desde el curso 17-18. Esto no penalizó el nivel de satisfacción de nuestros estudiantes con la asignatura y, por el contrario, les parece una solución lógica.

Pero, si impulsamos la escritura colaborativa en la nube, lo coherente parece utilizar el mismo medio para canalizar las entregas. Y una vez en este medio, también vemos la posibilidad de hacer algún tipo de seguimiento de lo que ha escrito cada estudiante. Esta es una oportunidad única, que no la ofrecen las entregas finales firmadas por todos los miembros del equipo. Algunas herramientas muestran gráficamente el grado de participación de los estudiantes y cuándo contribuyeron. Nuestra propuesta para conseguirlo es que los profesores también formemos parte del equipo de escritores de cada documento. Para ello necesitamos compartir todos los documentos con los equipos. Nuestra oportunidad de conseguirlo es ofrecer la versión inicial con una plantilla elaborada por nosotros y mejorada curso a curso.

Cuando el uso de editores en la nube implica pocos documentos, como es el caso de las memorias de TFG, compartir clones de una plantilla es una tarea sencilla. Sin embargo, cuando se maneja y revisa un número elevado de documentos la gestión manual de clones exige demasiado tiempo. En el contexto de Google Drive contamos con el lenguaje de *scripting* Google Apps Script que permite automatizar procesos como la generación de clones, la descarga de instantáneas en PDF, el cambio de rol a los estudiantes que comparten los clones o la inserción de texto en los clones para aportar retroalimentación. La programación de *scripts* sencillos, como los mencionados aquí, está a nuestro alcance y facilitan enormemente la gestión de los clones.

## Agradecimientos

Este trabajo ha sido parcialmente subvencionado por el Ministerio de Economía, Industria y Competitividad, proyecto EDU2016-79838-P.

## Referencias

- [1] Heidi J. C. Ellis, Gregory W Hislop. Structuring Software Engineering Learning within Open Source Software Participation. In *Proceedings of the 2014 conference on Innovation & technology in computer science*

- education, ITiCSE '14*, pp. 326–326, Uppsala, Sweden, Junio 2014.
- [2] Google team. Google Apps Script. Reference. 6 de octubre de 2020. Disponible en <https://developers.google.com/apps-script/reference>
- [3] Gregory W. Hislop, Heidi J. C. Ellis. Using Scaffolding to Improve Written Communication of Software Engineering Students. En *Proceedings of the 2009 Sixth International Conference on Information Technology: New Generations, ITNG '09*, pp. 707–712, Las Vegas, USA, April, 2009.
- [4] Arturo Jaime, José Miguel Blanco, César Domínguez, Ana Sánchez, Jónathan Heras, Imanol Usandizaga. Spiral and Project-Based Learning with Peer Assessment in a Computer Science Project Management Course. *Journal of Science Education and Technology*, 25:439–449, 2016.
- [5] David Lacalle-Castillo. Plantillas de documentos: ventajas e inconvenientes. Project Knowledge 14, Junio 2019. Disponible en <http://projectknowledge14.blogspot.com/2019/06/plantillas-de-documentos-ventajas-e.html>.
- [6] Ángeles López, Lledó Museros, María José Aramburu Cabo, Pedro García Sevilla. Experiencia de implantación de la asignatura de Proyecto de Final de Grado en Ingeniería Informática dentro de un contexto empresarial real. En *Actas de las XXIII Jornadas sobre Enseñanza Universitaria de la Informática, Jenui 2017*, pp. 23–30, Cáceres, julio 2017.
- [7] Rose M Marra, Linsey Steege, Chia-Lin Tsai, Nai-En Tang. Beyond “group work”: an integrated approach to support collaboration in engineering education. *International Journal of STEM Education*, 3(1):1–15, 2016.
- [8] Joe Miró Julià. Aprendizaje a través de la escritura: experiencias. En *Actas de las VI Jornadas de Enseñanza Universitaria de Informática, Jenui 2000*, pp. 451–456, Alcalá de Henares, septiembre 2000.
- [9] Lucía Ubilla Rosales, Lilian Gómez Álvarez, Katia Sáez Carrillo. Escritura colaborativa de textos argumentativos en inglés usando Google Drive. *Estudios Pedagógicos*, 43(1):331–348, 2017.
- [10] Fermín Sánchez Carracedo, Jaume Moral Ros, David López Álvarez. Mis estudiantes acaban el TFG (todos) en el tiempo previsto (casi todos). En *Actas de las XXI Jornadas de la Enseñanza Universitaria de la Informática, jenui 2015*, pp. 42–49, Andorra La Vella, julio 2105.
- [11] Dakuo Wang, Judith S. Olson, Jingwen Zhang, Trung Nguyen, Gary M. Olson. DocuViz: Visualizing Collaborative Writing. En *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, CHI 2015*, pp. 1865–1874, April 18 - 23 2015, Seoul, Republic of Korea, Abril 2015.
- [12] Soobin Yim, Dakuo Wang, Judith Olson, Viet Vu, Mark Warschauer. Synchronous writing in the classroom: Undergraduates’ collaborative practices and their impact on text quality, quantity, and style. En *Proceedings of the Conference on Computer Supported Cooperative Work, CSCW'17*, pp. 468–479, Portland Oregon USA, Febrero 2017.
- [13] Marta Zorrilla. Desarrollo de las habilidades de escritura y trabajo colaborativo utilizando herramientas de la nube. *Actas de las XLX Jornadas de Enseñanza Universitaria de Informática, Jenui 2013*, pp. 3–10, Castellón de la Plana, julio 2013.



**JEN 2021**  
VALENCIA  
01-08 JULIO

**JEN 2021**  
VALENCIA  
01-08 JULIO

# Actas de las XXVII Jornadas sobre la Enseñanza Universitaria de la Informática

# Actas de las XXVII Jornadas sobre la Enseñanza Universitaria de la Informática

Emitiendo  
desde Valencia  
7/8 de Julio  
de 2021

Emitiendo desde Valencia  
7/8 de Julio de 2021

ISSN: 2531-0607

#### Editores:

Carlos Catalán Cantero  
Francisco Grimaldo Moreno

#### Revisores de formato:

Agustín Cernuda del Río  
Antonio Pérez Poch

#### Organiza:

Centro de Estudios e Innovación  
**ETSE-UV**  
Escuela Técnica Superior de Ingeniería  
Universidad de Valencia

UNIVERSITAT  
ID VALÈNCIA



#### Colabora:



#### Financia:

