

# AIGORA: Repositorios abiertos y colaborativos para aprender a programar

## AIGORA: Open and collaborative repositories to learn Programming

Raquel Cedazo León<sup>1</sup>, Óscar Perpiñán Lamigueiro<sup>1</sup>, Julia Uruel Sanz<sup>1</sup>, Miguel Pérez Mateo<sup>1</sup>  
 raquel.cedazo@upm.es, oscar.perpinan@upm.es, juliauru2@gmail.com, miguel.perez.mateo@alumnos.upm.es

<sup>1</sup>Departamento de Ingeniería Eléctrica,  
 Electrónica, Automática y Física Aplicada  
 Escuela Técnica Superior de Ingeniería y Diseño  
 Industrial  
 Madrid, España

**Resumen-** Este artículo presenta la iniciativa desarrollada bajo el proyecto de innovación educativa AIGORA (Aprendizaje de Informática con GitHub Organizado en Repositorios Abiertos) para aprender a programar. El artículo explica cómo el uso de repositorios de software colaborativos y abiertos a través de la plataforma GitHub permite fomentar el aprendizaje entre estudiantes y el autoaprendizaje, facilitando además el desarrollo del trabajo por equipos y la evaluación de grupos numerosos. Además, se exponen los resultados alcanzados durante el primer año de su implantación en el curso 2018/2019 con estudiantes de los Grados de diferentes especialidades del área industrial en la Escuela Técnica Industrial de Ingeniería y Diseño Industrial.

**Palabras clave:** *aprendizaje colaborativo, aprendizaje entre pares, aprendizaje orientado a proyectos, autoaprendizaje, competencias transversales, grupos numerosos de estudiantes, trabajo en grupo, uso de las TIC*

**Abstract-** This article presents the initiative developed under the educational innovation project AIGORA (Computer Learning with GitHub Organized in Open Repositories) to learn how to program. The article explains how the use of collaborative and open software repositories through the GitHub platform allows to promote learning among students and self-learning, facilitating the development of team work and the evaluation of large groups. In addition, the results achieved during the first year of its implementation in the 2018/2019 academic year with students of the Degrees of different specialties of the industrial area in the Escuela Técnica Industrial de Ingeniería y Diseño Industrial are exposed.

**Keywords:** *collaborative learning, peer learning, project based learning, self-learning, transversal skills, large groups of students, group work, use of ICT*

### 1. INTRODUCCIÓN

La asignatura “Informática” es una materia de carácter básico y transversal ubicada en el segundo semestre del primer curso de los grados impartidos en la Escuela Técnica Industrial

de Ingeniería y Diseño Industrial e incluye entre sus competencias, tal y como refleja la guía de la asignatura<sup>1</sup>, la adquisición de “conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería”, “la capacidad para diseñar, desarrollar, implementar, gestionar y mejorar productos, sistemas y procesos en los distintos ámbitos industriales”, la aplicación de “los conocimientos adquiridos para identificar, formular y resolver problemas en contextos amplios, siendo capaces de integrarlos trabajando en equipos multidisciplinares”, la incorporación de “las TIC y las tecnologías y herramientas de la Ingeniería Industrial en sus actividades profesionales”.

Como propuesta innovadora, este curso 2018/2019 se ha implantado el uso de la plataforma GitHub para el desarrollo de la asignatura y enseñar a los alumnos a utilizar un sistema de control de versiones a la vez que aprenden programación, ya que es una herramienta esencial para mantener y compartir código. GitHub es la plataforma de alojamiento de código fuente más importante a nivel mundial. Emplea el sistema de control de versiones git y ofrece una amplia variedad de funcionalidades para el alojamiento y revisión del código, el trabajo colaborativo y la publicación de páginas web asociadas al repositorio de código, entre otras cosas.

### 2. CONTEXTO

La asignatura “Informática” se desarrolla en base a tres pilares:

- Lecciones magistrales para exponer los conceptos principales de la programación. Estas lecciones se apoyan en diapositivas y en material bibliográfico.
- Espacios de trabajo en clase con apoyo del profesorado para la resolución de problemas en los que se aplica lo expuesto en las lecciones magistrales.

<sup>1</sup> La guía docente de esta asignatura en el grado de Ingeniería Eléctrica está disponible en el siguiente enlace:

[https://www.upm.es/comun\\_gauss/publico/guias/2018-19/2S/GA\\_56IE\\_565000224\\_2S\\_2018-19.pdf](https://www.upm.es/comun_gauss/publico/guias/2018-19/2S/GA_56IE_565000224_2S_2018-19.pdf)

- Trabajo por grupos, consistente en un proyecto definido por cada equipo, y desarrollado durante un tiempo no inferior a un mes. El resultado de este proyecto es un programa informático que debe ser ejecutado y defendido en sesión pública.

Por una parte, con este proyecto se persigue la mejora de los resultados académicos de la asignatura mediante el uso de repositorios compartidos almacenando una amplia colección de problemas desarrollados por los propios estudiantes. Según este planteamiento, el aprendizaje de los estudiantes se lleva a cabo elaborando código y documentación de suficiente calidad como para superar la evaluación académica y la evaluación de sus pares, y mediante la revisión del código contribuido por otros estudiantes.

En la literatura existen trabajos similares para aprender programación en estudios universitarios. Angulo (2018) explica cómo han utilizado GitHub para la gestión de las prácticas de los estudiantes mediante la creación de un repositorio, envío de archivos a través del mecanismo drag-drop, uso del editor online y añadir colaboradores. Cuenta además que estos estudiantes han sido capaces de familiarizarse con la plataforma en sólo dos semanas. Case (2016) explica el proceso para introducir los sistemas de control de versiones de una forma incremental a los estudiantes. Todos coinciden en que la introducción adecuada de esta herramienta tan potente en una fase temprana del aprendizaje de la programación es una experiencia positiva y valiosa para su futuro.

Por otra parte, desde el punto de vista de la evaluación, el uso de GitHub permite trazar la evolución y el aprendizaje de cada estudiante a lo largo del curso, pudiendo asignar de forma automática una nota como parte de la evaluación continua. Esta característica es especialmente útil para **grupos numerosos**, en los que se hace difícil evaluar su trabajo diario y dar una retroalimentación de sus tareas, ya que, además, esta metodología fomenta y premia la **revisión por pares entre estudiantes**.

Debe resaltarse también que a través de este proyecto se facilita el aprendizaje de una herramienta que los estudiantes van a poder utilizar en asignaturas de cursos posteriores y en el desarrollo de los Trabajos Fin de Grado que traten con software. De hecho, hay otras asignaturas en las que de forma opcional u obligatoria ya se está empleando GitHub para el desarrollo de trabajos en equipo y prácticas. Además, el uso de GitHub proporciona un valor añadido en la asignatura de Informática por la conexión existente con el mundo profesional y con conocidos proyectos que mantienen el código a través de su plataforma. Es necesario señalar que GitHub cuenta con más de 30 millones de repositorios y se ha convertido en el estándar para desarrollar colaborativamente y compartir código. Por tanto, los estudiantes van a adquirir una **competencia transversal** desde el primer curso.

Finalmente, hay un beneficio indirecto y a más largo plazo que es el uso de esta herramienta para que los estudiantes comuniquen a terceros sus habilidades y destrezas profesionales, ya que la cuenta de GitHub puede convertirse en su portafolio profesional (Dabbish 2012, Marlow 2013). Numerosas empresas actualmente piden el perfil de GitHub

antes de contratar para poder ver el código del potencial empleado.

### 3. DESCRIPCIÓN

El proyecto AIGORA plantea los siguientes objetivos concretos:

- Crear un repositorio colaborativo, de libre acceso, de programas planteados y resueltos por los propios alumnos para el aprendizaje de la asignatura.
- Aprendizaje de un lenguaje de programación mediante el uso de herramientas y técnicas empleadas habitualmente en entornos profesionales.
- Desarrollar la capacidad de mejorar el código a partir de las revisiones de pares o de un maestro.
- Desarrollar la capacidad de contribuir y revisar el código desarrollado por otros.
- Aprendizaje del trabajo por objetivos, con gestión del tiempo individual y del equipo para desarrollar ideas y alcanzar objetivos con restricciones temporales.
- Desarrollar la capacidad de trabajo en equipo.
- Desarrollar el hábito de documentar el código, ya sea de manera individual o en equipo.
- Gamificar las clases mediante el uso de “insignias” premiando a aquellos alumnos que hagan más contribuciones en los repositorios.
- Publicar y difundir el trabajo realizado en páginas web.

Las fases del proyecto han sido las siguientes:

1. Sesión introductoria a GitHub para los estudiantes en la primera semana de curso.
2. Configuración de cuentas individuales y de equipos en GitHub. Esta fase ha sido asistida por los dos becarios del proyecto, fuera del horario de clase, siendo un apoyo fundamental para los estudiantes.
3. Publicación del material docente (diapositivas, enunciados de problemas y códigos de ejemplo) en GitHub por parte del profesorado.
4. Resolución de problemas con código alojado en GitHub Classroom, usando las cuentas individuales.
5. Planteamiento inicial del proyecto informático a desarrollar por cada grupo, alojando el texto descriptivo en GitHub con los objetivos del programa.
6. Desarrollo incremental del proyecto por grupos, alojando el código en el repositorio del equipo. El profesorado revisa y comenta este código de forma gradual, y el equipo atiende a los requerimientos y da las contestaciones oportunas mediante los *issues* de los repositorios.
7. Cierre del proyecto. El resultado de cada trabajo consiste en un programa en el repositorio asociado y

una Wiki explicativa con toda la documentación del proyecto: diagrama de flujo, explicación de las funcionalidades y pantallazos del programa en ejecución.

- Finalmente, se ha elaborado una página web alojada en el mismo GitHub (<https://aigora.github.io>) que agrupa y enlaza todos los repositorios de los programas desarrollados por los estudiantes.

#### 4. RESULTADOS

El producto resultante del proyecto es una colección de repositorios públicos en GitHub bajo la organización AIGORA, con toda la colección de ejercicios resueltos y proyectos informáticos desarrollados por los estudiantes. Todo ello se ha publicado en una página web creada para el proyecto <https://aigora.github.io/> (véase Fig. 1) donde se listan y enlazan, componiendo el portafolio de trabajos de curso. En total, en la primera convocatoria durante el curso académico 2018/2019, se han desarrollado 150 proyectos por un total de 317 estudiantes, distribuidos en grupos de 2 y 3 estudiantes. Los estudiantes han realizado programas completos en C para dar soluciones informáticas en diferentes contextos: desde un sistema para controlar las horas de los trabajadores de una empresa, una aplicación para gestionar el aparcamiento de un determinado negocio, una colección de juegos de mesa, hasta un traductor de código Morse, entre otros muchos.

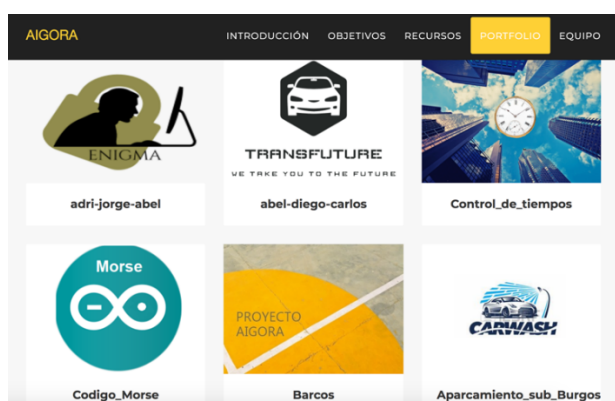


Figura 1. Página web del proyecto AIGORA con el portafolio de todos los trabajos de curso de los estudiantes (<https://aigora.github.io>)

La manera de medir los resultados del proyecto ha sido a través de la propia API que ofrece GitHub, por la cual se han podido extraer diferentes métricas que analizan las contribuciones de los estudiantes a lo largo del curso. El resultado del análisis se encuentra en el repositorio del proyecto (<https://github.com/aigora/api>), donde se pueden encontrar todas las gráficas extraídas, así como el código en lenguaje R con las que se han producido. En este artículo hemos incluido las gráficas más significativas.

La Figura 2 recoge el número de commits por equipo de trabajo, y la Fig. 3 recoge el número de commits por estudiante. Ambos indicadores reflejan cómo los estudiantes han utilizado

la herramienta a lo largo del curso. En la Figura 2 se observa cómo la mayoría de grupos han realizado muy pocos commits para subir todo su trabajo, lo que indica que únicamente han utilizado GitHub para subir la versión final del programa. En la Figura 3 se ve también que la mayoría de los estudiantes han realizado pocos commits.

A largo plazo, durante los próximos cursos, se pretende analizar el comportamiento de los estudiantes, como el número de abandonos, la evolución del número de ejercicios resueltos, el número de líneas de código por estudiante/grupos, etc.

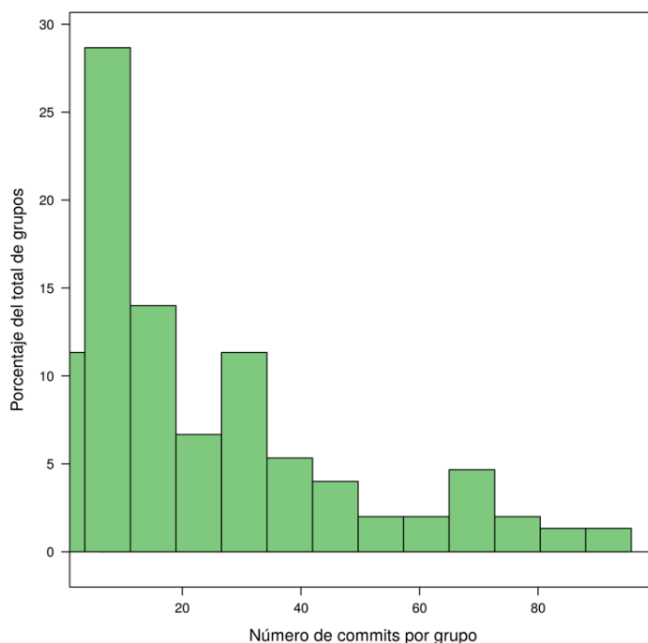


Figura 2. Estadísticas: número de commits por grupo de trabajo

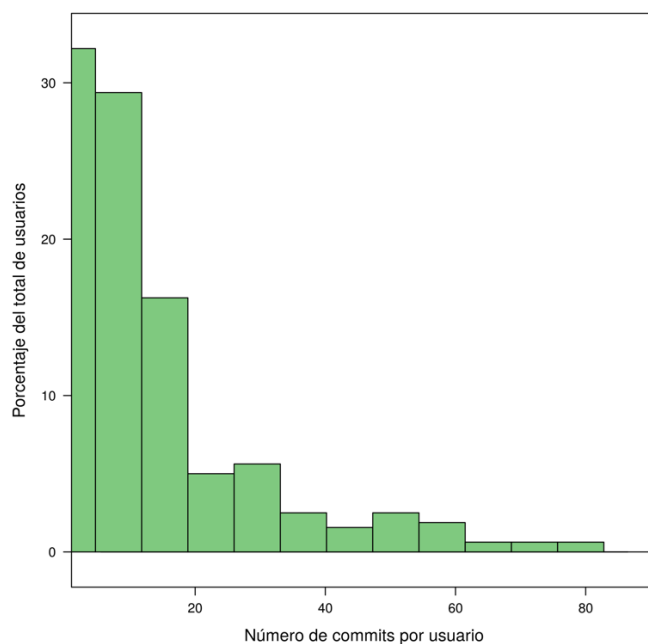


Figura 3. Estadísticas: número de commits por estudiante

Una vez terminado el curso se ha realizado una encuesta telemática y anónima para conocer la satisfacción de los estudiantes sobre lo que les ha parecido la experiencia y la utilidad de GitHub en la asignatura. La encuesta, como puede verse en la Tabla 1, consta de 4 preguntas de valoración en una escala de 1 (nada de acuerdo) a 10 (totalmente de acuerdo) y una pregunta abierta para enviar sugerencias y comentarios. En total se han recogido 72 respuestas.

Tabla 1. Encuesta de satisfacción realizada a los alumnos al final de la asignatura de Informática

| Pregunta   | Respuesta   |
|--|---|
| 1. ¿Crees que GitHub es útil para ayudarte en la asignatura?   | Grado de satisfacción (valor numérico del 1 al 10.) |
| 2. ¿Crees que te será útil GitHub para tu futuro?              | Grado de satisfacción (valor numérico del 1 al 10.) |
| 3. ¿Nos recomiendas seguir utilizando GitHub en la asignatura? | Grado de satisfacción (valor numérico del 1 al 10.) |
| 4. En general, valora tu experiencia con GitHub                | Grado de satisfacción (valor numérico del 1 al 10.) |
| 5. Si tienes alguna sugerencia o comentario indícalo.          | Campo de texto abierto para responder.              |

Los resultados se muestran en la Figura 4. En la pregunta 1 el 49% de los estudiantes da una puntuación mayor o igual a 5. En la pregunta 2 este porcentaje aumenta hasta el 56%, se sitúa en el 51% en la pregunta 3, y en el 56% en la pregunta 4. La apreciación general es que los estudiantes no han llegado a valorar bien la herramienta porque, entre otras cosas, lo ven desconectado con la programación y con la materia en sí.

El análisis de estos resultados y la observación del quehacer cotidiano de los estudiantes por los profesores, nos permiten afirmar que este proyecto ha cumplido parcialmente con los objetivos planteados inicialmente.

Se ha puesto en marcha un repositorio colaborativo de libre acceso y un amplio número de estudiantes han contribuido a poblarlo de ejercicios. Sin embargo, la revisión entre pares se ha llevado a cabo de forma bastante reducida. En general, los estudiantes se han limitado a contribuir con su código sin utilizar la herramienta para interactuar con los demás. En este sentido, el objetivo de gamificación a través de insignias para distinguir a estudiantes excelentes no se ha llevado a cabo.

El proyecto ha contribuido a enriquecer el proceso de aprendizaje con herramientas y técnicas empleadas habitualmente en entornos profesionales. Sin embargo, un número elevado de estudiantes no han alcanzado a valorar adecuadamente sus funcionalidades y su potencialidad a medio y largo plazo.

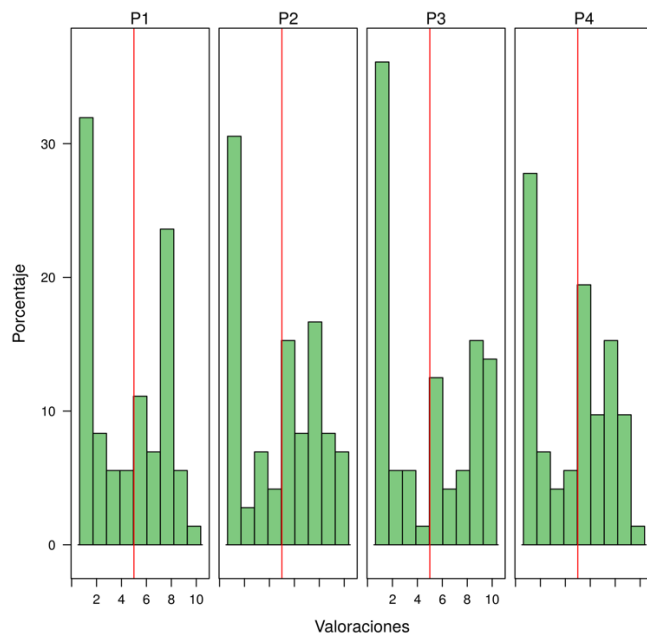


Figura 4. Resultado de las encuestas realizadas a los estudiantes sobre el uso de GitHub en la asignatura de Informática. Curso 2018/2019.

En la fase de trabajo por equipos, en términos generales, el proyecto ha contribuido al aprendizaje del trabajo por objetivos y a desarrollar la capacidad de trabajo en equipo. Sin embargo, hemos detectado que un amplio número de equipos y estudiantes no han sido capaces de organizar adecuadamente el reparto de tareas, ni gestionar correctamente el tiempo para atender a los objetivos de calidad.

En esta misma fase, el proyecto ha contribuido a desarrollar hábitos de documentación de código de manera progresiva, mediante la elaboración de ficheros README públicos como portada del trabajo, la inclusión de comentarios anidados en del código (requeridos para facilitar la evaluación por parte del profesor), y mediante la elaboración de documentación en modo tutorial del trabajo a través de wikis públicas.

## 5. CONCLUSIONES

El análisis de los resultados cuantitativos y de la observación del comportamiento de los estudiantes permiten afirmar un cumplimiento suficiente de los objetivos del proyecto, expresado a través de dos puntos clave:

- Los estudiantes han sido conscientes de que su trabajo en todo momento era público y eso ha provocado que hayan sido cuidadosos con el código y la documentación.
- Los repositorios colaborativos de los estudiantes han servido para que los profesores puedan comentar el código desarrollado y solucionar errores durante las clases. Esto es de gran ayuda en clases con gran número de alumnos en los que no se puede dar un feedback personalizado a cada estudiante.

Sin embargo, algunas de las conclusiones negativas y que se pretenden mejorar en el futuro, son las siguientes:

- Los estudiantes no han sido conscientes de que todas sus contribuciones quedan registradas y sirven para medir su evaluación durante el curso. Creemos que esto ha provocado que los alumnos no hayan hecho un gran esfuerzo en ir enviando sus contribuciones poco a poco, sino que la mayoría de ellos se han limitado a hacer una entrega al final del curso.
- No todos los estudiantes han sido capaces de ver la potencia y la utilidad que tienen los sistemas de control de versiones como GitHub, por lo que se cree que es fundamental que se dedique más tiempo en clase a explicarlo.
- Ha habido un número considerable de estudiantes que han aprovechado muy bien esta oportunidad, habiendo seguido correctamente el flujo de trabajo de git, incluso utilizando funcionalidades avanzadas como ramas y resolución de conflictos.

En general, la experiencia ha sido buena para saber los puntos fuertes y débiles del uso de GitHub en la asignatura. Como conclusión, GitHub ofrece muchas posibilidades para mantener software y documentarlo, por lo que la metodología presentada en este artículo puede ser fácilmente reproducible en cualquier disciplina académica que se relacione con el software.

## AGRADECIMIENTOS

*Este trabajo ha sido parcialmente financiado por el Proyecto de Innovación Educativa de la Universidad Politécnica de Madrid: "AIGORA: Aprendizaje de Informática con GitHub Organizado en Repositorios Abiertos".*

## REFERENCIAS

- Angulo, M. A., & Aktunc, O. (2018). Using GitHub as a Teaching Tool for Programming Courses.
- Case, D. M., Eloe, N. W., & Leopold, J. L. (2016). Scaffolding Version Control into the Computer Science Curriculum. In Proceedings of the 2016 International Workshop on Distance Education Technology (in conjunction with the 22nd International Conference on Distributed Multimedia Systems (DMS'16)).
- Dabbish, L., Stuart, C., Tsay, J., & Herbsleb, J. (2012, February). Social coding in GitHub: transparency and collaboration in an open software repository. In Proceedings of the ACM 2012 conference on computer supported cooperative work (pp. 1277-1286). ACM.
- Marlow, J., Dabbish, L., & Herbsleb, J. (2013, February). Impression formation in online peer production: activity traces and personal profiles in GitHub. In Proceedings of the 2013 conference on Computer supported cooperative work (pp. 117-128). ACM