

# Aplicaciones empresariales en el aula: Experiencia educativa con GitHub Classroom

## ***Autores***

*Rodríguez Lera, Francisco Javier\*; Fernández Llamas, Camino; Gutiérrez Fernández, Alexis; González Santamarta, Miguel Ángel; Sánchez González, Lidia; Conde González, Miguel Ángel*

*\*Departamento de Ingenierías Mecánica, Informática y Aeroespacial*

## ***Nombre del Grupo de Innovación***

*ULE ROBLATIC. Robótica, Learning Analytics y TICS aplicadas a los procesos de enseñanza/aprendizaje*

## **RESUMEN**

El profesorado de asignaturas de programación necesita identificar de forma clara el resultado del aprendizaje bajo los criterios del EEES. Esto significa medir los conocimientos, competencias y habilidades adquiridas de forma que se consiga avalar la correcta inserción del alumnado en el mundo laboral. El uso de la herramienta de control de versiones Git, bien conocida en entornos empresariales, permitirá hacer el seguimiento y evaluación analítica del estudiante, mientras, al mismo tiempo, hace que se familiarice con herramientas y métricas que se utilizan en el mercado laboral. Esta propuesta muestra nuestra experiencia utilizando Git con GitHub, portal empresarial de gestión de repositorios, y con GitHub Classroom, portal de gestión de control de versiones para educadores, en diferentes asignaturas que trabajan la competencia de programación de ordenadores del grado en Ingeniería Informática de la Universidad de León.

**Líneas de actuación:** Aprendizaje basado en problemas, proyectos o retos; Aprendizaje Colaborativo; Innovación en Evaluación.

## **Introducción**

Preparar y capacitar a los egresados en el uso de herramientas actuales y bien establecidas, es clave en la formación actual y permite ofrecer orientación hacia la visión competencial de las materias relacionadas con el desarrollo de software. En este sentido destaca el uso de herramientas de control de versiones, que son aquellas que permiten visualizar y gestionar los cambios que se hacen en el código fuente de un proyecto software de forma individual o grupal a lo largo de su ciclo de vida. La aplicación más conocida en este ámbito es Git, acompañada de la herramienta web de gestión GitHub.

Los números de GitHub son llamativos. Su uso en empresas según los datos del propio GitHub en su informe “The State of the Octoverse” es masivo: más de 2,9 millones de organizaciones de más de 70 países utilizan repositorios privados y públicos. Estas organizaciones sumadas al uso por otros actores hacen un total de 44 millones de repositorios creados solamente contando entre el 1 de octubre de 2018 y el 30 de septiembre de 2019 (GitHub Inc, 2019).

Dado este contexto, se plantearía la siguiente pregunta de investigación: ¿El uso de herramientas empresariales es viable y fácilmente acomodable al paradigma educativo actual para garantizar la

adquisición del saber y del saber hacer necesarios para el mercado de trabajo? Para poder abordar esta pregunta se lleva a cabo un experimento a partir de un planteamiento, de aprendizaje basado en proyectos, en el que el trabajo, tanto individual como en equipo, va a realizarse mediante el uso de la herramienta GitHub Classroom (GitHub Inc, 2018).

*GitHub Classroom* es una aplicación web donde docentes y discentes interactúan durante un curso. La aplicación permite automatizar el proceso de crear y compartir repositorios con las tareas a realizar, así como controlar el acceso a los mismos y supervisar la entrega de tareas por los estudiantes. Los profesores pueden crear una lista de estudiantes para el aula y, a continuación, crear, asignar, revisar y calificar las tareas dentro del aula. Dicha herramienta nos permitirá realizar evaluaciones verticales y horizontales de los estudiantes en los cursos de programación. Entendiendo evaluación horizontal, como la realización de una evaluación continua del alumnado en la que se valore la progresión y el esfuerzo realizado a lo largo de las prácticas, identificando aquellos conceptos de estas que resultan más difíciles de asimilar; mientras que la evaluación vertical permite realizar seguimiento de problemas y proyectos entre cursos académicos y asignaturas.

Este artículo presenta primeramente la experiencia innovadora desarrollada. A continuación, describe cómo se ha realizado la experiencia y muestra y analiza los resultados obtenidos. Finalmente, el documento recoge las conclusiones alcanzadas en la experiencia propuesta.

## **Experiencia innovadora**

### *Objetivos*

Teniendo en cuenta la pregunta de investigación planteada y la propuesta a llevar a cabo, el objetivo de la presente experiencia es:

*Formular criterios académicos, competenciales y de cualificación profesional que nos ayuden a responder a la demanda del mercado laboral europeo en cuanto al desarrollo software*

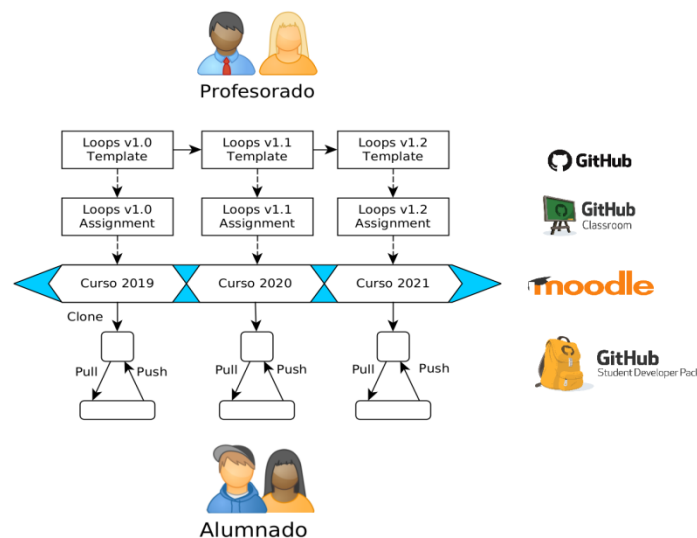
El enfoque metodológico adoptado en esta iniciativa de innovación docente es el de aprendizaje basado en proyectos, en concreto, en el contexto de la enseñanza de ingenierías y empleando la herramienta de colaboración y seguimiento GitHub Classroom, mencionada anteriormente. Respecto a la metodología empleada en el aula se aplica en 4 fases básicas:

1. **Inicio.** Se plantean problemas y proyectos a los estudiantes, se les facilitan herramientas para entender cómo abordarlos y para desarrollarlos/colaborar.
2. **Análisis del problema y diseño de la solución.** Donde los estudiantes profundizan en la definición de los objetivos de los proyectos, en aplicar los conocimientos necesarios para resolverlos y en plantear y planificar una posible definición.

3. **Desarrollo.** Resolución de los proyectos de forma individual o colaborativa de cara a obtener una o varias versiones posibles de la solución.
4. **Evaluación.** Presentación de los resultados y evaluación de los mismos.

Para facilitar especialmente las fases 2 y 3 para el estudiante y la fase 4 para el docente se ha empleado GitHub Classroom. Se trata de un servicio web proporcionado por GitHub a los educadores. Dicho servicio permite crear lo que se denominan “assignments” o tareas, que definen repositorios (espacio de trabajo digital para preservar y compartir el software desarrollado) para trabajos en el aula, tanto para aproximaciones individuales como grupales. Así, la tarea se le entrega al alumnado en forma de un repositorio que facilita la colaboración de los estudiantes.

El uso de GitHub Classroom facilita la interacción activa entre profesorado y alumnado pudiendo ofrecer comentarios oportunos y monitorizar el progreso síncrono o asíncrono de los estudiantes. La gestión e inclusión del alumnado se realiza de forma manual o automática desde los LMS habituales tales como Moodle (este caso) o Sakai. La organización de los proyectos a través de GitHub Classroom, nos permite: 1) dar soporte estructural al estudiante, lo que le permitirá la selección, organización y el establecimiento de relaciones de ciertos conceptos nuevos con otros anteriores. 2) Facilitar el trabajo en equipo y colaborativo mediante el uso de herramientas de desarrollador que les permite trabajar con roles y gestión de incidencias para solventar el proyecto como equipo. 3) Hacer seguimiento del trabajo individual o en equipo, evaluando lo que han hecho sus compañeros, interactuar, generar versiones de soluciones, etc. Además, el docente tendrá acceso a toda esa información que le facilitará la evaluación y la toma de decisiones.



**Figura 1:** Planteamiento de la propuesta

La Figura 1 ilustra el planteamiento en el que se realizan actividades similares durante varios cursos académicos empleando GitHub, *GitHub Classroom* y Moodle. La propuesta se articula del siguiente modo: primeramente, el profesorado genera la plantilla de la tarea que tendrá que resolver el

estudiante (en la figura, el ejercicio Loops, dedicado a bucles). A continuación, se crea la tarea en *GitHub Classroom*, incluyendo los estudiantes que pueden hacer la tarea y la visibilidad (público/privado) que se le quiere otorgar a dicho repositorio. Como resultado se obtiene un enlace para compartir. Posteriormente, se publica este enlace en Moodle para que el alumnado pueda acceder a la tarea. Una vez acceden, se crea una copia de la tarea para cada uno de ellos y a partir de ahí ya pueden interactuar con dicho repositorio con los mecanismos habituales. Por último, resaltar que, en nuestro caso, cada repositorio tiene visibilidad restringida a privada y los estudiantes no pueden ver el trabajo de sus compañeros. Esta aproximación nos permite realizar un seguimiento de: 1) las tareas propuestas cada año (revisión horizontal profesorado), 2) la tutorización profesor-alumnado (revisión vertical de las prácticas) y 3) el comportamiento del alumnado cada año (revisión horizontal alumnado).

### *Descripción de la experiencia*

Si bien el uso de GitHub en el ámbito docente por parte del grupo dio comienzo en 2015 siguiendo corrientes que empezaban a estimularse en el ámbito de innovación docente (Kelleher, J., 2014) el curso de inicio del que parte este proyecto con *GitHub Classroom* comenzó en el curso 2018-2019 manteniéndose hasta la actualidad.

Las propuestas de actuación que se hacen en este plan intentan: a) dar respuesta a las necesidades educativas del estudiante mediante esta estrategia de enseñanza, b) favorecer la aplicación de las metodologías digitales disponibles en el aula, y c) posibilitar la familiarización del discente con herramientas técnicas utilizadas en las empresas:

- Incrementar y extender el uso de ciertas herramientas transversales al aprendizaje de programación, principalmente aquellas relacionadas con la integración continua.
- Desarrollar nuevas metodologías y contextos innovadores de enseñanza-aprendizaje.
- Impulsar la formación en entornos virtuales, potenciando el uso de diferentes canales de comunicación entre estudiantes y profesores con el objetivo de desarrollar la dimensión asíncrona.
- Enfatizar la dimensión tutorial en el modelo de enseñanza-aprendizaje centrado en el estudiante remoto.
- Dar soporte al profesorado para la preparación de materiales y su uso a través de la red.
- Impulsar publicaciones que refuercen los procesos de enseñanza-aprendizaje.

Las fases que articulan este proyecto se dirigen hacia uno de los principios de enseñanza-aprendizaje de acuerdo con el EEES, favorecer la participación activa del estudiante. Esto es, que mediante el uso de *GitHub Classroom* se favorezca el aprendizaje autónomo, facilite el apoyo tutorial (alumno-profesor) e impulsen el aprendizaje colaborativo.

Esto implica ir más allá del currículo adaptándonos a los modelos de formación centrados en el trabajo, eso sí, no solo de los estudiantes sino también del profesorado participante en la innovación docente. A continuación, se describen las fases, personal involucrado y las actividades realizadas:

**Fase 0 (F0):** En esta fase se realiza la Evaluación Experta en la que se detallan las competencias a desarrollar y las estrategias organizativas de los involucrados. El personal involucrado en este caso es un conjunto de Profesorado Experto que comparte premisas con el Profesor Coordinador. Período: año 18 (M0), posteriores (M2, M7)

Actividades:

- Pre-evaluación de disciplinas de acorde a la idoneidad de la metodología y la herramienta
- Post-evaluación de la experiencia

**Fase 1 (F1):** En esta etapa se realiza la definición y análisis de las estrategias de enseñanza. El personal participante es el Profesor Coordinador con el profesorado participante. Período: año 18 (M0), posteriores (M2, M7)

- Definición de los tipos de proyectos y las funcionalidades de *GitHub Classroom* con profesores expertos de diferentes áreas.
- Sesión informativa acerca del uso general de la plataforma a profesores implicados.
- Análisis de aplicación de la metodología por el profesorado participante.
- Sesión de análisis y discusión.

**Fase 2 (F2):** el Profesorado Participante define la Planificación y Diseño de las prácticas que realizará el estudiante adecuándolo a su materia. Períodos: M2 y M7

- Estructuración del uso de la herramienta durante el semestre. Se han propuesto dos alternativas a los ejercicios:
  - TipoA - pasivo: se utiliza únicamente como herramienta de control de versiones, tickets y trabajo colaborativo.
  - TipoB - activo: se utilizan las herramientas de Integración Continua disponibles en la aplicación permitiendo la automatización de las pruebas de evaluación.
- Planificación y/o actualización de las prácticas a través de la herramienta:
  - Ejercicio Simple Individual (ESI): ejercicio semanal independiente, donde el profesor plantea un ejercicio para cada práctica y estudiante, uno por repositorio.
  - Ejercicio Incremental Individual (EII): repositorio único para el estudiante para todo el cuatrimestre. El estudiante incorpora cada práctica en periodos definidos.
  - Ejercicio Grupal (EG): varios estudiantes contribuyen con su trabajo en el repositorio
- Aprovisionamiento respecto al número de estudiantes involucrados en la materia.
- Gestión de las aulas (denominadas Classroom):

- Classroom para el curso académico (CCA): implica crear aulas nuevas cada curso e incorporar nuevamente los ejercicios.
- Classroom única por Asignatura (CUA):
  - CUA1: Crear ejercicio y añadir estudiantes a dicho ejercicio.
  - CUA2: Mantener ejercicio y actualizar el listado de estudiantes asociados.

**Fase 3 (F3):** el Profesorado Participante realiza la Implantación y explotación del material docente basado en Git y mediante *GitHub Classroom* para que el alumnado realice su aprendizaje. Períodos: M9-M1 (primer cuatrimestre) y M3-M6 (segundo cuatrimestre).

- Agregar los links en el CMS correspondiente (Moodle).
- Informar al estudiante en la sesión inicial del uso de la plataforma.
- Potenciar en el aula el uso de las herramientas a lo largo del curso:
  - como herramienta optativa (no influye en la evaluación)
  - como herramienta obligatoria (influye en la evaluación)
- Recibir las impresiones de su uso y resaltar en el retorno los conocimientos adquiridos.
- Seguimiento del uso de los repositorios.

**Fase 4 (F4):** Finalmente se realiza la Valoración de resultados. Donde el profesor participante traslada al Profesor Coordinador su experiencia. Período: M2 y M7

- Análisis del profesor participante
  - Seguimiento vertical: Revisión de cómo ha funcionado la práctica y actualización, en caso necesario, para el siguiente curso. Este proceso dependerá de forma directa de la investigación teórica del profesorado, de la titulación y del fenómeno educativo del contexto actual, p. ej. inclusión de un nuevo lenguaje de programación o adaptación a situación anómala temporal (confinamiento).
  - Seguimiento horizontal: Revisión de la carga de trabajo y del proceso de aprendizaje de los estudiantes al realizar la práctica: consolidaciones de cambios en el código (*commits*), análisis de resultados, interacción entre miembro del equipo, etc.
- Análisis del profesor coordinador
- Informe final del curso académico

**Asignaturas participantes en enseñanza reglada:**

- Profesorado de Grado en Ingeniería en Informática: Asignaturas implicadas durante los cursos: Programación 1 (18-19, 19-20, 20-21, 21-22), Estructura de computadores (20-21, 21-22), Aplicaciones Web (18-19, 19-20, 21-22)
- Profesorado de Máster en Ingeniería en informática: Computación de altas prestaciones (20-21, 21-22)

- Profesorado de Máster en Investigación en Ciberseguridad: Diseño y programación seguras (20-21, 21-22)

La Figura 2 muestra la distribución de cada una de esas fases a lo largo de los cursos académicos en los que se ha empleado.

	enero	febrero			marzo	abril	mayo	junio	julio			agosto	septiembre	octubre	noviembre	diciembre
<b>2018</b>								F0, F1	F2				F3	F3	F3	F3
<b>2019</b>	F3	F4	F2	F0, F1	F3	F3	F3	F3	F4	F2	F0, F1		F3	F3	F3	F3
<b>2020</b>	F3	F4	F2	F0, F1	F3	F3	F3	F3	F4	F2	F0, F1		F3	F3	F3	F3
<b>2021</b>	F3	F4	F2	F0, F1	F3	F3	F3	F3	F4	F2	F0, F1		F3	F3	F3	F3
<b>2022</b>	F3	F4	F2	F0, F1	F3	F3	F3									
Reuniones de seguimiento						Informe Final de Curso			Curso Confinamiento/Pandemia				Vacaciones			

**Figura 2.** Cronograma de tareas seguido en el proyecto.

## Resultados alcanzados

Los resultados alcanzados en este proyecto se resumen cuantitativamente en:

- Repositorios creados: 9.600.
- Estudiantes involucrados: más de 500 personas y 241 equipos.
- Profesorado involucrado: alrededor de 15 personas.

Con esta propuesta hemos realizado un seguimiento horizontal de las prácticas en las diferentes asignaturas a nivel de profesorado (actualizando erratas o incluyendo sugerencias de los estudiantes) y se ha podido analizar cuál es el nivel de entrada del alumnado. Por otro lado, hemos obtenido gran cantidad de datos para su análisis vertical (qué estudiantes están más involucrados, cómo se comportan los miembros de un equipo, etc.).

Aunque la plataforma permite trabajar con infinitos repositorios y ejercicios en su uso pasivo (sin corrección automática), solamente permite 3000 acciones automáticas de GitHub en su uso activo. De este modo, si se usa en esta modalidad hay que ser metódico y cuidadoso para evitar alcanzar el número máximo disponible. Por ejemplo, en nuestra experiencia se ha llegado a alcanzar más de 2000 correcciones en menos de dos semanas en una sola asignatura con dos prácticas semanales.

## Conclusiones y valoración de la experiencia

La metodología presentada en este trabajo permitió a los profesores realizar una evaluación del impacto de la herramienta de control de versiones GitHub sobre los estudiantes. Dada la integración entre las herramientas del alumnado, podríamos decir que, con una estrategia de enseñanza adecuada, es posible desplegar en el aula otro tipo de herramientas empresariales. Además, la aproximación ha permitido al profesorado hacer un seguimiento de la evolución del material y del alumnado en diferentes asignaturas de la titulación del Grado en Ingeniería Informática de la

Universidad de León, quedando su evaluación para trabajo futuro. Del mismo modo, las aproximaciones específicas como la evaluación automática externa de los repositorios (Guerrero-Higueras ÁM, 2020), y el trabajo con *Pull Request*, que define un mecanismo formal de solicitud de incorporación y consolidación de cambios a los repositorios donde los desarrolladores notifican a otros que han terminado una función y donde se fuerza al desarrollador a realiza la incorporación de cambios mediante su revisión, continúan en paralelo y pueden suponer un punto de partida para mejorar el aprendizaje colaborativo de los estudiantes.

## Agradecimientos

Agradecemos a los miembros del grupo ROBLATIC su colaboración en esta experiencia y a la Escuela de Formación el soporte económico para la misma.

## Referencias bibliográficas

- Guerrero-Higueras, A. M., Fernández Llamas, C., Sánchez González, L. Gutiérrez Fernández, A., Esteban Costales, G., & Conde González, M. A. (2020). Academic success assessment through version control systems. *Applied Sciences*, *10*(4), 1492.
- Fiksel, J., Jager, L. R., Hardin, J. S., & Taub, M. A. (2019). Using GitHub classroom to teach statistics. *Journal of Statistics Education*, *27*(2), 110-119.
- Nelson, M. A., & Ponciano, L. (2021, May). Experiences and insights from using GitHub Classroom to support Project-Based Courses. In *2021 Third International Workshop on Software Engineering Education for the Next Generation (SEENG)* (pp. 31-35). IEEE.
- Anbaroğlu, B. (2021). A collaborative GIS programming course using GitHub Classroom. *Transactions in GIS*, *25*(6), 3132-3158.
- Angulo, M. A., & Aktunc, O. (2019, April). Using GitHub as a teaching tool for programming courses. In *2018 Gulf Southwest Section Conference*.
- GitHub Inc, The State of the Octoverse, At School, (2019). Disponible online [11/01/2023]: URL:<https://octoverse.github.com/2019/#at-school>
- Github Inc, GitHub Classroom, (2018), Disponible online [11/01/2023]: URL: <https://classroom.github.com>
- Sprint, G., & Conci, J. (2019). Mining GitHub classroom commit behavior in elective and introductory computer science courses. *The Journal of Computing Sciences in Colleges*, *35*(1).
- Kelleher, J. (2014, January). Employing git in the classroom. In *2014 World Congress on Computer Applications and Information Systems (WCCAIS)* (pp. 1-4). IEEE.