

Repositorio GitHub con objetos de aprendizaje para grados de ingeniería industrial en la asignatura de programación

GitHub repository with learning objects for industrial engineering degrees in programming

Isabel María del Águila Cano¹, José Rafael García Lázaro¹, Clara Marcela Miranda Sarmiento¹, Rafael Guirado Clavijo¹

imaguila@ual.es, jrgarcia@ual.es, cmiranda@ual.es, rguirado@ual.es

¹Departamento de Informática
Universidad de Almería
Almería, España

Resumen- En este trabajo mostramos cómo se está desarrollando un proyecto de innovación docente cuyo objetivo es la elaboración de objetos de aprendizaje para la asignatura de programación en los estudios de Ingeniería Industrial de la Universidad de Almería. Estos objetos de aprendizaje persiguen un doble objetivo, por un lado, mostrar ejemplos sencillos y significativos de los principales conceptos de programación y, por otro lado, representan una batería de problemas extrapolables a cualquier titulación de carácter técnico. Los problemas se presentan dentro de un repositorio de GitHub, por lo que se pueden descargar y/o duplicar/bifurcar directamente, además de tener una página web disponible sobre la misma plataforma que está construida para guiar al estudiante en el acceso a los problemas, convirtiéndose en un recurso docente adicional.

Palabras clave: *Objetos de aprendizaje, ingeniería industrial, código compartido*

Abstract- In this work we show how a teaching innovation project is being developed whose objective is the elaboration of learning objects for the programming subject in Industrial Engineering studies at the University of Almería. These learning objects seek, on the one hand, to show simple and meaningful examples of the main concepts of programming and, on the other, are a battery of problems that can be extrapolated to any degree of a technical nature. The problems are presented within a GitHub repository, so that it is directly downloadable and/or forked, in addition to having a web page available directly on the same platform that is built to guide the learner in accessing the problems. Becoming an additional learning resource.

Keywords: *learning objects, Industrial engineering, shared code*

1. INTRODUCCIÓN

El pensamiento computacional como base de la programación se entiende como el medio para resolver problemas, diseñar sistemas y comprender el comportamiento humano, haciendo uso de los conceptos fundamentales de la informática, (Wing, 2006), ya está presente en la mayoría de los currícula de educación primaria y secundaria, llegando incluso a enseñarse el llamado pensamiento computacional desenchufado (Caeli y Yadav, 2020). No obstante, las

habilidades para entender el software y escribir nuevos programas no han sido alcanzadas por todos los estudiantes que entran en el mundo universitario.

Como instituciones que preparan a los futuros profesionales, las universidades se enfrentan a la tarea de cómo armonizar la educación de los estudiantes que presentan un conocimiento avanzado con respeto al desarrollo de programas con la educación de los estudiantes neófitos en el uso de lenguajes de programación (Skalka, 2021). Por otra parte, cuando el estudiante se enfrenta a la programación en entornos reales con herramientas y lenguajes profesionales, el aprendizaje va más allá del simple pensamiento computacional, haciéndose necesario disponer de recursos y actividades que guíen a los estudiantes en el aprendizaje de la programación, dentro del entorno de plataformas compartidas sobre las que actualmente se desarrollan la mayor parte las nuevas aplicaciones software.

Programar no es ni más ni menos que construir programas entendiéndolo por programa de computadora una secuencia de instrucciones que una computadora puede interpretar y ejecutar, y que resuelve un problema específico de procesamiento automático de información. La Association for Computing Machinery (ACM) y la International Electric and Electronics Engineers organization (IEEE) definen el término programación de computadoras (a menudo abreviado como programación) como la colección completa de actividades que rodean la descripción, el desarrollo y la implementación efectiva de soluciones algorítmicas a problemas bien especificados (Topi, 2010). Programar no significa solamente “codificar” en un lenguaje de programación particular, sino también “diseñar” el algoritmo que resuelve el problema que ha de ser “comprendido y analizado”.

En este trabajo se presenta un proyecto de innovación que busca en el ámbito de los estudios técnicos, en especial los grados de Ingeniería Industrial, poner a disposición de estudiantes y profesores un conjunto de problemas a utilizar como objetos de aprendizaje de pequeño contenido. En especial para la asignatura de Programación, que se encuentra integrada

en el módulo de formación básica y que se imparte en el primer curso de las titulaciones de grado de la rama industrial en la Universidad de Almería, y que además de las competencias específicas, se le han asignado las competencias transversales de trabajo en equipo, conocimientos básicos de la profesión, y capacidad para resolver problemas.

Un objeto de aprendizaje es un contenido digital reutilizable, simple e independiente compuesto por objetivos, actividades y autoevaluación. Las tendencias actuales son las de reducir al mínimo estos objetos de aprendizaje para hablar de micro aprendizaje, que no es más que la organización planificada de breves experiencias de aprendizaje diseñadas para cumplir un objetivo concreto de aprendizaje, y que pueden o no insertarse en un itinerario de formación más extenso (Hung y Friesen, 2007). Estas experiencias siempre deben ayudar a los estudiantes a practicar lo que están aprendiendo y puesto que nuestra disciplina, la programación, es eminentemente práctica, hemos planteado el despliegue de una batería de problemas, cuyo resultado es un programa de computadora, desplegado sobre la plataforma colaborativa GitHub (GitHub); ésta es una de las principales plataformas para crear proyectos abiertos de herramientas y aplicaciones, y se caracteriza sobre todo por sus funciones colaborativas que ayudan a que todos puedan aportar su granito de arena para mejorar el código. Además desde su compra por Microsoft en 2018, su expansión a nivel mundial es un hecho y también el incremento en número de funcionalidades y nuevas capacidades.

Existen diversos repositorios desplegados en GitHub de programación (Repositorios recomendados por GitHub para aprender a programar, 2021), que contienen múltiples recursos didácticos y material de consulta de diferentes temas y usando diferentes lenguajes de programación, con ejemplos de aplicación en ámbitos muy variados. Por ejemplo, dentro del contexto de la ingeniería industrial, encontramos el repositorio abierto y colaborativo para aprender a programar desarrollado por el proyecto AIGORA, que incluye los programas planteados y resueltos por los propios alumnos en el ámbito de los grados de ingeniería industrial de la Universidad Politécnica de Madrid (AIGORA: Repositorios abiertos y colaborativos para aprender a programar, 2021) y además facilita la evaluación de los estudiantes en grupos numerosos fomentando el trabajo diario y la revisión por pares (Cedazo, 2019).

Sin embargo, nosotros proponemos un único repositorio GitHub que está mucho más especializado en problemas de programación de pequeño tamaño en el ámbito de la ingeniería industrial (mecánica, eléctrica, electrónica y química industrial), y que aporta también como novedad la utilización de estos problemas como objetos de micro aprendizaje, los cuales creemos que son de especial relevancia para motivar a los estudiantes de estas titulaciones que se inician en la disciplina de la programación.

Partiendo del repositorio de objetos de aprendizaje, los estudiantes desarrollarán sus propias soluciones a los problemas duplicando/bifurcando (fork) el trabajo inicial y generando contenidos que podrán incorporarse (merge) al proyecto de partida. De esta forma los equipos de estudiantes, además de adquirir competencias de trabajo en equipo, serán los creadores y consumidores de contenidos realizados por el resto de sus compañeros en términos de distintas soluciones sobre los mismos problemas.

2. CONTEXTO

En las dos últimas décadas hemos sido testigos y actores de importantes cambios en el modelo de enseñanza universitaria español. Dichos cambios comenzaron con la Declaración de Bolonia del 19 de junio de 1999 (Marquand, 1999) donde se generó el germen que ha dado lugar al Espacio Europeo de Educación Superior (EEES).

La mayoría de universidades españolas, que ya disponían de programas de innovación docente, multiplicaron sus actividades como consecuencia de la necesidad de incorporar el nuevo paradigma que impulsa un cambio en las metodologías docentes centrado en el aprendizaje del estudiante a lo largo de la vida (Foce, 2020). Se ha de otorgar el protagonismo del aprendizaje al estudiante, situándolo en el centro del proceso y repensando la carga de trabajo en términos de actividad del mismo, midiéndose ésta en una nueva unidad denominada créditos ECTS (European Credit and Accumulation Transfer System).

El trabajo total de un estudiante a tiempo completo se fijó en 60 créditos ECTS para un año académico, con 40 semanas al año y 40 horas de trabajo a la semana. Un crédito ECTS corresponde a 25 horas de tiempo total de trabajo de un estudiante (asistencia a clases teóricas y prácticas, tareas autónomas, formación de estudios y exámenes). Esta nueva unidad de medida implica un nuevo modelo educativo basado en el trabajo del estudiante y no en las horas de clase. De esta forma, todos los títulos universitarios oficiales se han adaptado y son más homogéneos con titulaciones similares en el resto de países de la Unión Europea, por lo que ahora la movilidad de los estudiantes y la integración laboral son más fáciles.

La Ingeniería Industrial ha sido la rama de la Ingeniería más demandada en España, tanto para los estudiantes como para el mercado laboral. Desde su creación en el siglo XIX, estos estudios universitarios integran la ingeniería mecánica, eléctrica, electrónica y química en un solo título. Las competencias específicas a adquirir para los estudiantes de la rama industrial están reguladas en España por la normativa CIN / 351/2009 (BOE, 2009) y se dividen en tres módulos: competencias formativas básicas, competencias comunes a la rama industrial y competencias específicas a la tecnología.

La asignatura de programación se ha planificado en la Universidad de Almería en el primer curso y nos encontramos de partida con dos situaciones contrapuestas: por un lado, hay estudiantes con amplios conocimientos previos pero que han de enfrentarse a entornos profesionales (y no solo académicos) que le son desconocidos, y por otro lado, estudiantes que no conocen ni la materia ni los entornos de programación y que suelen percibirla como muy difícil porque es muy diferente de las demás materias básicas que si les resultan más familiares.

El mundo tecnológico en el que nos movemos cada día se encuentra en un constante desarrollo y esto además de hacer que el futuro de los programadores sea muy prometedor, conlleva que la programación sea un valor añadido a las competencias de los profesionales de cualquier ámbito tecnológico. Tal como lo dijo Mark Zuckerberg “La programación te permite crear algo totalmente nuevo de la nada”. Además nos permite explotar las habilidades creativas y ejercitar el cerebro al máximo, pensando de manera lógica.

La asignatura “Programación” se sitúa dentro del área “Fundamentos del Desarrollo de Software” en los Currícula de

Computación de ACM e IEEE (Topi, 2010), los cuales recomiendan la fluidez en al menos un lenguaje de programación, junto con habilidades y conceptos que son esenciales para la práctica de la programación, incluyendo claramente el diseño y el análisis. El lenguaje de programación elegido habitualmente en los estudios de Ingeniería Industrial en las universidades españolas es el lenguaje de programación C, por ser un lenguaje muy estandarizado y ampliamente utilizado.

El equipo docente de programación, basándose en la experiencia de 16 años impartiendo esta asignatura, tanto en los estudios precedentes de Ingeniería Técnica Industrial (Mecánica), donde se desarrollaron diversas experiencias piloto, como en la implantación definitiva de las titulaciones de grado en Ingeniería Mecánica, Eléctrica, Electrónica Industrial y Química Industrial hace ya 11 años, ha decidido extender los trabajos previos en la mejora docente basados en el aprendizaje colaborativo y el aprendizaje basado en problemas (García, 2015) para incorporar esos problemas como elementos de aprendizaje dentro de la plataforma GitHub, en concreto dentro del repositorio: ([ProblemasProgramacion, 2021](#)). Cada problema debe ser resuelto generando un programa codificado en un lenguaje de programación concreto, en nuestro caso C.

3. DESCRIPCIÓN

Las diversas metodologías de desarrollo de objetos de aprendizaje definen diversas etapas que tienen por objetivo servir de referencia para el establecimiento de los requisitos necesarios en el diseño y construcción de Objetos de Aprendizaje (Morales y Diez-Martínez, 2020). La mayoría de éstas consideran como criterio “el equilibrio” en cuanto a la integración de aspectos pedagógicos y tecnológicos con los que se busca tener: en la parte pedagógica, una competencia en el estudiantes, y en la parte tecnológica, las herramientas adecuadas que faciliten la construcción y uso del objetos de aprendizaje (Silva, 2013). En nuestra experiencia se han definido cinco etapas: Planificación, Análisis, Desarrollo, Validación e Implantación, de las que se ha ejecutado casi al completo las tres primeras, puesto que será en el segundo semestre del año que viene cuando se pueda realizar la implantación definitiva del trabajo hecho.

Se han fijado las actuaciones a realizar incluyendo la temporización, durante la etapa de Planificación. Puesto que el proyecto de innovación se desarrolla en dos cursos académicos, el estudio de las necesidades, las decisiones de carácter tecnológico y los primeros prototipos se desarrollarán durante el primer año, completando los desarrollos durante el primer semestre del segundo año, para finalizar la implantación y validación en el segundo semestre del segundo año, durante el periodo en el que se cursa la asignatura.

Durante la etapa de Análisis, una de las decisiones clave para este trabajo ha sido la definición de la arquitectura en la que se desplegarán los objetos de aprendizaje. Como necesidades básicas se identifica en primer lugar que el despliegue ha de ser público y que las plataformas de control de versiones que son básicas en el desarrollo de aplicaciones software deben ser el soporte para las soluciones a ofrecer a los estudiantes. Es decir, el código solución a los problemas propuestos debe estar en un sistema de control de versiones abierto para desarrolladores de software, y no solo para los estudiantes.

En este sentido se optó por GitHub (GitHub) que es un repositorio online gratuito que permite gestionar proyectos y controlar versiones de código. Es muy utilizado por desarrolladores de programas para almacenar sus trabajos dando así la oportunidad a millones de personas de todo el mundo de cooperar en ellos. Se podría hablar de GitHub como la red social pensada para desarrolladores, siendo este repositorio uno de los más usados a nivel mundial, esto unido la característica llamada “GitHub Pages”, la cual hace fácil y gratuito que se pueda crear un sitio web de varias páginas que queda hospedado como “.github.io”, en nuestro caso (del Aguila, 2021), nos llevó a seleccionar esta plataforma como la base para el despliegue de los objetos de aprendizaje.

Otro aspecto importante es la necesidad de utilizar un lenguaje para completar la página web ofrecida a los estudiantes. Si bien se podría haber optado por el uso de lenguajes con más opciones de diseño que mejorarían el resultado final, la opción elegida ha sido utilizar Markdown (Gruber, 2002) que es un lenguaje de marcado que facilita la aplicación de formato a un texto empleando una serie de caracteres de una forma especial. Este lenguaje fue pensado para elaborar textos, cuyo destino iba a ser la web, con más rapidez y sencillez que si estuviésemos empleando directamente HTML. Y si bien este suele ser el mejor uso que podemos darle, también podemos emplearlo para cualquier tipo de texto, independientemente de cual vaya a ser su destino, lo que permite reutilizar esos textos en la elaboración de otros materiales de aprendizaje, como las notas de los profesores o los propios estudiantes.

Finalizadas las etapas de planificación y análisis, actualmente estamos realizando las tareas de desarrollo de los contenidos, la figura 1 muestra el aspecto de una de las páginas ya publicadas. Se han utilizado iconos simples y representativos de cada uno de los lenguajes de programación utilizados. Todos los problemas están en C y pseudocódigo, aunque algunos de ellos también se han incluido las implementaciones en Java, C++, Python y lenguaje ensamblador.

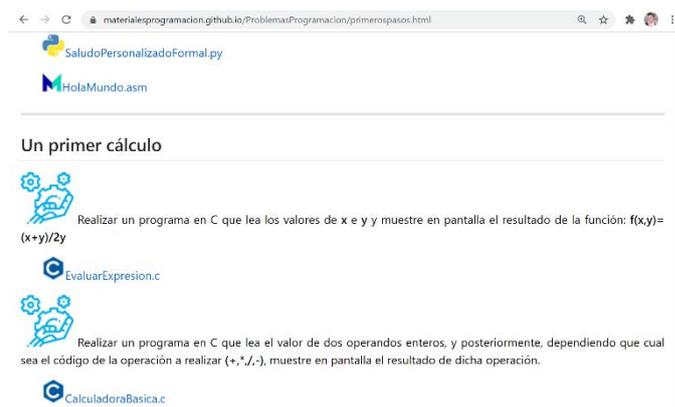


Figura 1: Página GitHub

La validación e implantación definitiva están previstas para el segundo semestre del próximo curso académico, cuando de nuevo se curse la asignatura. El planteamiento general será que los estudiantes dupliquen/bifurquen el repositorio y lo amplíen con sus propias aportaciones a soluciones alternativas a los problemas, generando nuevos contenidos compartidos. Además puesto que GitHub facilita el trabajo colaborativo, se partirá de

grupos de trabajo que colectivamente construyen conocimiento. Esta forma de trabajo grupal también desarrolla habilidades mixtas (aprendizaje y desarrollo personal y social), donde cada miembro del equipo es responsable tanto de su aprendizaje como del aprendizaje del resto de los miembros del grupo de trabajo (Felder y Brent, 2001)

Las tareas a resolver por el grupo están estructuradas de modo que los miembros del grupo se necesitan mutuamente para completarlas (interdependencia positiva). Estas tareas se distribuyen entre los miembros del grupo, y cada estudiantes tiene que ser responsable de su trabajo y también del trabajo del resto de sus compañeros, para que pueda responder al profesor de todo el trabajo realizado (García, 2015). Con estos equipos de trabajo cooperativo, estamos promoviendo las explicaciones y discusiones dentro del grupo, y la resolución de conflictos de manera constructiva. El propósito de esta estrategia educativa es que los estudiantes desarrollen habilidades interpersonales y grupales, como conocerse y confiar en los demás o comunicarse correctamente. Una vez finalizadas las soluciones, es importante un proceso de reflexión dentro del grupo para identificar qué ha sido útil y qué aspectos se podrían mejorar.

Los repositorios generados por cada uno de los grupos de trabajo podrán fusionarse en el repositorio raíz empleando peticiones, de forma que los estudiantes además de generar nuevos objetos de aprendizaje, podrán practicar sobre una herramienta del mundo real para el desarrollo colaborativo de software como es GitHub, para la gestión de versiones y generación de ramas en proyectos.

4. RESULTADOS

Los problemas se han estructurado siguiendo la clasificación de macro-aprendizaje, meso-aprendizaje y micro-aprendizaje, donde en este último nivel están los problemas. A continuación, muestra esta estructuración pero se han incluido sólo parte de los problemas puesto que son muy numerosos, para mayor detalle consultar el repositorio directamente.

- **Primeros pasos**

- Hola Mundo
- Calculadora básica

- **Programación estructurada**

- Secuencial
 - Calcular la masa de una bola de hierro
 - Calcular la aceleración angular
 - Calcular el momento de fuerza
- Selectiva, control condicional
 - Conversión del voltaje del sensor en temperatura
 - Conversión de coordenadas cartesianas a coordenadas polares
 - Calcular operaciones básicas de vectores
 - Resolver una ecuación de segundo grado
 - Resolver sistema lineal de 3 ecuaciones con 3 incógnitas
 - Calcular intersección de dos segmentos en el plano
- Iteración
 - Calcular la potencia entera de un número flotante
 - Calcular el factorial de un número natural
 - Calcular la suma de una serie numérica

- Calcular la resistencia equivalente de un conjunto de resistencias en paralelo
- Calcular máximo común divisor
- Iteraciones anidadas
 - Triángulo de Tartaglia
 - Aristas de un paralelepípedo
- **Diseño modular**
 - Descomposición modular
 - Calcular la suma de una serie de Taylor
 - Calcular la posición del centro de masa de un polígono
 - Conversión de coordenadas cartesianas en coordenadas esféricas
 - Calcular el momento de inercia de un cilindro
 - Calcular una función de pulso
 - Calcular una cicloide
 - Recursividad
 - Calcular el factorial
 - Calcular el producto con el método de campesino ruso
 - Calcular número de caminos independientes
 - Funciones como parámetros
 - Calcular el área bajo una curva mediante la regla trapezoidal
 - Calcular el cero de una función mediante el teorema de Bolzano
 - Calcular la longitud de un tramo de una curva paramétrica alabeada
- **Estructuras de datos**
 - Arrays de una dimensión
 - Derivación de polinomios
 - Calcular la media y mediana
 - Calcular una ecuación diferencial
 - Arrays multidimensionales
 - Multiplicación de matrices
 - Procesado de imágenes
 - Registros
 - Transformaciones geométricas en el plano
 - Análisis de un sistema de partículas
 - Gestión de una lista de nuevos materiales.
 - Gestionar las notas de estudiantes en una asignatura.
 - Manejo de un conjunto de piezas planas.
 - Algoritmos de búsqueda y ordenación
 - Cálculo de la mediana y cuartiles de un conjunto de datos
- **Archivos**
 - Modelo de regresión lineal de un conjunto de datos y validación de hipótesis del modelo.
 - Fusión de archivos para la generación de facturas de consumo eléctrico

Tal como se indicó en el apartado anterior aún se están desarrollando los objetos de aprendizaje, pero ya están disponibles los niveles que suponen los elementos básicos de la programación estructurada. Cabe destacar, la figura 2, que aún sin estar totalmente desplegado, ya hay 14 usuarios de GitHub que se han interesado en el proyecto definiendo bifurcaciones sobre el mismo y de los que durante el resto del trabajo esperamos recibir retroalimentación y sugerencias de mejoras debido al interés que ya han mostrado.

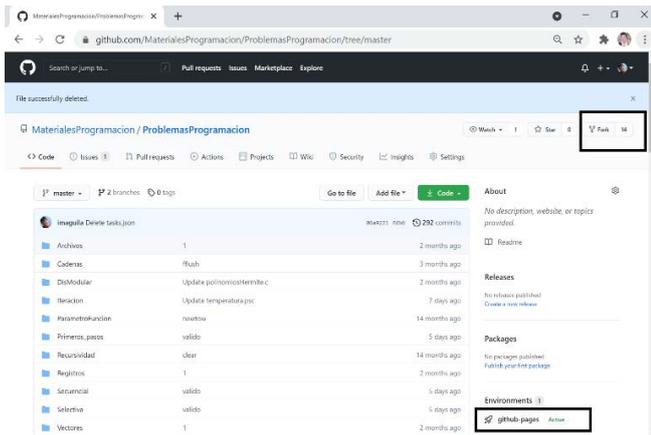


Figura 2: Repositorio GitHub con 14 Forks

5. CONCLUSIONES

En este trabajo se muestra una experiencia donde, con relativamente poco esfuerzo, se pueden generar contenidos atractivos en el ámbito de la programación para estudiantes universitarios. Se han desplegado sobre la plataforma GitHub un conjunto de problemas del dominio de la Ingeniería Industrial que se pueden utilizar como objetos de aprendizaje y de trabajo colaborativo en asignaturas de programación de estudios tecnológicos.

Si bien aún queda trabajo por hacer, sobre todo en el apartado de la evaluación de la experiencia, es positivo el interés ya mostrado por los usuarios de GitHub que no estando relacionados directamente con la asignatura objeto de la experiencia, ya están interesados en los resultados.

En el próximo curso está previsto implantar el modelo de aprendizaje al completo utilizando los problemas como objeto de aprendizaje e implicando a los estudiantes en la duplicación/bifurcación del repositorio para la generación de sus propias soluciones que se podrán incorporar a la finalización del semestre.

AGRADECIMIENTOS

Este trabajo ha sido financiado por la Universidad de Almería, en concreto a través del proyecto “programación” (21_22_2_13C), dentro de la convocatoria para la Creación de Grupos Docentes de Innovación y Buenas Prácticas y para la Creación de Materiales Didácticos de la UAL.

REFERENCIAS

AIGORA: Repositorios abiertos y colaborativos para aprender a programar. (2021, 14 de julio) <https://aigora.github.io/>

BOE, Boletín Oficial del Estado (2009) 20 February 2009. Ministerio de la Presidencia. Gobierno de España.

Repositorios recomendados por GitHub para aprender a programar. (2021, 14 de julio) <https://blogthinkbig.com/repositorios-github-aprender-a-programar>.

Caeli, E. N., & Yadav, A. (2020). Unplugged approaches to computational thinking: A historical perspective. *TechTrends*, 64(1), 29-36.

Cedazo, R., Perpiñán, Ó. Uruel, J., y Pérez, M. (2019). AIGORA: Repositorios abiertos y colaborativos para aprender a programar *Proceeding of V Congreso Internacional sobre Aprendizaje, Innovación y Cooperación, CINAIC-2019* pp 368-372.

del Águila, I., García, J., Miranda, C. y Guirado, R. (2021, 09 de Julio). *Problemas de Programación*. <https://materialesprogramacion.github.io/ProblemasProgramacion/>

Felder, R. M., y Brent, R. (2001). Effective strategies for cooperative learning. *Journal of Cooperation & Collaboration in College Teaching*, 10(2), 69-75.

García, J., Moreno, J., del Águila, I. y Guirado, R. (2015). Teaching computer programming in the degrees of industrial engineering with collaborative and problem-based learning, *EDULEARN15 Proceedings*, pp. 7623-7633.

GitHub. *Plataforma de desarrollo colaborativo*. <https://github.com/>

Gruber, J. (2002). *Lenguaje Markdown*. Daring Fireball. <https://daringfireball.net/projects/markdown/>

Marquand, J., & Scott, P. (2018). The Bologna Declaration of 19 June 1999. In *Democrats, Authoritarians and the Bologna Process*. Emerald Publishing Limited.

Hug, T., y Friesen, N. (2007). Outline of a microlearning agenda. *Didactics of Microlearning. Concepts, Discourses and Examples*, 15-31.

Morales, R. A., y Diez-Martinez, E. (2020). Revisión de metodologías para diseñar Objetos de Aprendizaje OA: un apoyo para docentes. *Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología*, (26), e4-e4.

Poce Fatou, J. A. (2020). Panorama de la Innovación Docente en la Universidad Española. Perspectivas desde la Universidad de Cádiz. *RESED. Revista de Estudios Socioeducativos*, 8, 179-191.

Problemas de Programación. (2021, 09 de Julio). <https://github.com/MaterialesProgramacion/ProblemasProgramacion>

Silva Sprock, A., Ponce Gallegos, J. C., y Hernández Bieliukas, Y. (2013). Estado del Arte de las Metodologías para el Desarrollo de Objetos de Aprendizaje. *Conferencias LACLO*, 4(1).

Skalka, J., Drlik, M., Benko, L., Kapusta, J., Rodríguez del Pino, J. C., Smyrnova-Trybulska, E., Anna Stolinska, Peter Svec, y Turcinek, P. (2021). Conceptual Framework for Programming Skills Development Based on Microlearning and Automated Source Code Evaluation in Virtual Learning Environment. *Sustainability*, 13(6), 3293.

Topi, H., Kaiser, K. M., Sipior, J. C., Valacich, J. S., Nunamaker Jr, J. F., de Vreede, G. J., & Wright, R. (2010). *Curriculum guidelines for undergraduate degree programs in information systems*. ACM.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33