

# TRABAJO FIN DE MÁSTER



**UCAM**

UNIVERSIDAD CATÓLICA  
DE MURCIA

## FACULTAD DE EDUCACIÓN

*Máster Universitario en Formación del Profesorado  
Especialidad en Matemáticas e Informática*

MICROAPRENDIZAJE Y GAMIFICACIÓN EN LA  
ENSEÑANZA DE PROGRAMACIÓN INFORMÁTICA

*Autor:*

*Francisco José Pastor Álvarez*

<https://www.youtube.com/watch?v=8G9V2mwzFQ4>

*Directora:*

*Dra. Dña. María Magdalena Cantabella Sabater*

*Murcia, mayo de 2023*









# TRABAJO FIN DE MÁSTER



**UCAM**

UNIVERSIDAD CATÓLICA  
DE MURCIA

## FACULTAD DE EDUCACIÓN

*Máster Universitario en Formación del Profesorado  
Especialidad en Matemáticas e Informática*

### MICROAPRENDIZAJE Y GAMIFICACIÓN APLICADO A LA PROGRAMACIÓN INFORMÁTICA

*Autor:*

*Francisco José Pastor Álvarez*

*Directora:*

*Dra. Dña. María Magdalena Cantabella Sabater*

*Murcia, mayo de 2023*

## **Agradecimientos**

Agradezco este proyecto a los compañeros con los que compartí la experiencia de práctica docente en el Instituto Superior de Formación Profesional San Antonio de Murcia, los cuales fueron clave en la inspiración para este trabajo.

## ÍNDICE

<b>1. JUSTIFICACIÓN</b> .....	<b>11</b>
<b>2. MARCO TEÓRICO</b> .....	<b>15</b>
2.1. Área de trabajo.....	15
2.2. Necesidad de mejora en el proceso de aprendizaje .....	17
2.3. Metodología y estrategias de aprendizaje.....	20
<b>3. OBJETIVOS</b> .....	<b>23</b>
3.1. Objetivo General .....	23
3.2. Objetivos Específicos .....	23
<b>4. METODOLOGÍA</b> .....	<b>25</b>
4.1. Contenidos .....	26
4.1.1. Microaprendizajes.....	26
4.1.2. Tareas .....	27
4.1.3. Microevaluaciones .....	29
4.1.4. Sistema de gamificación.....	30
4.2. Contenidos curriculares.....	33
4.3. Temporalización y actividades .....	33
4.4. Recursos .....	40
4.5. Atención a la diversidad .....	41
4.6. Evaluación de las actividades .....	42
4.6.1. Evaluación del primer objetivo específico.....	43
4.6.2. Evaluación del segundo objetivo específico .....	45
4.6.3. Evaluación del tercer objetivo específico.....	47
<b>5. EVALUACIÓN</b> .....	<b>51</b>
5.1. Rúbricas de evaluación.....	51
<b>6. REFLEXIÓN Y VALORACIÓN FINAL</b> .....	<b>55</b>
<b>7. REFERENCIAS</b> .....	<b>59</b>

<b>8. ANEXOS</b> .....	<b>61</b>
8.1. Anexo I. Ejemplo de microevaluación .....	61
8.2. Anexo II. Sistema de gamificación .....	61



## 1. JUSTIFICACIÓN

Nos encontramos en un momento de transformación del proceso industrial de la humanidad. Lo que conocemos como Industria 4.0 está impactando de manera directa en la denominada Cuarta Revolución Industrial. Según un estudio publicado por McKinsey Global Institute (Manyika et al., 2017), dentro de este contexto se prevé que para el año 2030 entre 75 y 375 millones de trabajadores se verán en la necesidad de cambiar su profesión y aprender nuevas habilidades. Este es solo uno de los grandes retos a los que nos enfrentamos la sociedad contemporánea en esta nueva era digital.

Ante este escenario, cada vez son más demandados los perfiles profesionales relacionados con informática, lo que a su vez promueve la necesidad de aumentar la oferta de formación de este tipo de trabajador. Esa demanda se debe, entre otros factores, al comprobado aumento que tiene para la productividad de las empresas incluir la digitalización en sus procesos internos, realidad visible desde hace varios años (Berumen & Ibarra, 2008). Todo esto incide en un aumento del alumnado que opta por este tipo de formación, atraído en muchos casos por la optimista perspectiva laboral que ofrece. Dentro de estas formaciones, independientemente de su especialidad, se observa una habilidad básica y transversal a todas ellas: la programación informática. Destaquemos que, en el caso de España, a día de hoy no existe una asignatura de Programación como tal en la educación básica, por lo que la gran mayoría de estudiantes se enfrentan por primera vez a ella al cursarla en un ciclo especializado.

El objetivo de este proyecto se va a centrar en la mejora del proceso de enseñanza-aprendizaje de los alumnos de la asignatura de Programación que cursen la misma por primera vez, con independencia de que la estén cursando en un ciclo de formación profesional o un grado universitario. En mi experiencia educativa, tanto como alumno de Ingeniería como durante mis prácticas como profesor en un Ciclo Formativo de Grado Superior en Desarrollo de Aplicaciones Web, he observado como la gran mayoría de los estudiantes de primera matriculación no tienen ninguna experiencia con la programación. Solo unos pocos han realizado previamente alguna tarea formativa relacionada,

apoyándose en contenidos encontrados en la red, habiendo cursado alguna materia optativa disponible en algunos centros educativos o habiendo realizado alguna formación extraescolar en academias especializadas.

A continuación, pasamos a describir algunas de las dificultades que se observan en un aula durante el aprendizaje de la programación, y a las cuales pretenderá dar respuesta el contenido de este proyecto. Como he comentado previamente, estas problemáticas se sustentan principalmente en mi observación como alumno en prácticas de este Máster y, hace algunos años, como estudiante de Ingeniería Informática. No obstante, en mi experiencia laboral, también he podido observar carencias en profesionales de la informática las cuales posiblemente también estén relacionadas con esta dificultad formativa.

Es visible que para muchos alumnos el alcanzar el nivel de razonamiento lógico necesario para la programación se vuelve un obstáculo. Este tipo de pensamiento se desarrolla en base a la práctica y son pocas las asignaturas en la educación básica que trabajan la misma. Una de ellas serían las matemáticas, pero ocurre que un alumno puede alcanzar un nivel de desempeño requerido para superar esta materia en base a una comprensión instrumental (saber que hacer para superar un problema, aplicar reglas o fórmulas sin comprender la razón de su aplicación, etc.) en lugar de una comprensión relacional (saber la razón por la que algo funciona) (Greaney, 2015). Esto hace que no siempre esta asignatura ayude a desarrollar en el alumno esta habilidad cognitiva. En el caso de la programación, tenemos el hecho de que al alumno medio que cursa por primera vez esta materia se le debe trasladar la mentalidad de trabajar la misma en base a la repetición y la práctica de ejercicios, lo cual requiere una disciplina y rutina de trabajo que puede diferir de un enfoque memorístico abundante en las asignaturas comunes de la enseñanza básica.

Como desencadénate de la problemática anterior, podemos detectar que esta situación también deriva en una disminución de la motivación del alumnado, provocada principalmente por la percepción de dificultad de esta asignatura. Será otro de los objetivos de este proyecto incentivar esta motivación en el alumno (principalmente su motivación intrínseca) aplicando técnicas

contrastadas a través de diversos estudios y que abordaremos en el marco teórico. Adicionalmente, plantearemos otra serie de actividades dirigidas a la concienciación sobre la importancia de dominar la herramienta principal de trabajo dentro de la programación (entorno de desarrollo) con lo que perseguiremos un doble objetivo: por un lado la propia concienciación de esta necesidad sumada a una mejora en el desempeño de un entorno en concreto y por otro la incorporación de un nuevo factor motivador dentro del aprendizaje de la materia.

Para finalizar con los objetivos perseguidos en el proyecto, detectamos que la evaluación de la asignatura de programación puede ser abordada desde otra perspectiva. Tradicionalmente, la evaluación de la programación tiene diferentes enfoques posibles que hipotéticamente podríamos agrupar en evaluación práctica y evaluación teórica. Nos encontramos con que, ya se opte por una opción u otra, las evaluaciones suelen tener como denominador común el presentar varios contenidos desarrollados durante un periodo del curso en una única prueba, fomentando el temor y la ansiedad por parte del alumnado a realizar la misma. Debido a la naturaleza secuencial y progresiva de los contenidos en programación, plantearemos otra manera de abordar dichas pruebas.

En este punto, esbozaremos la propuesta de intervención en base a las diferentes problemáticas que hemos expuesto en este apartado. Comentábamos anteriormente que el alumnado objetivo de este proyecto eran estudiantes que cursaran la asignatura de Programación por primera vez. Vamos a acotar este perfil a un alumno de Formación Profesional de Grado Superior Técnico de Desarrollo de Aplicaciones Web con el objetivo de enmarcar legalmente el proyecto. Las técnicas de aprendizaje y metodologías desarrolladas se abordarán durante todo el curso académico, siendo por tanto esta la duración total de aplicación del proyecto.

Comenzamos describiendo en que va a consistir la propuesta de solución para la problemática inicial que hemos descrito: trabajar el razonamiento lógico. Para trabajar este punto, se va a introducir el concepto de microlearning o microaprendizaje. Este se caracteriza por el desglose de los contenidos de la

asignatura en lecciones de pequeño tamaño, con lo que pretendemos conseguir que los alumnos trabajen los mismos con mayor asiduidad y fomenten su desempeño con este tipo de razonamiento. El modo de presentación de estas pequeñas lecciones también será diseñado en base a un formato que facilite su consumo por parte del alumnado.

Una segunda dificultad que abordaremos en el proyecto consiste en introducir una metodología de evaluación diferente a la tradicional. Para esto se quiere aplicar la técnica de microevaluación que consistiría, de manera genérica, en realizar pequeñas evaluaciones de corta duración durante todo el curso que nos den una visión del nivel de desempeño que tiene el alumno en un contenido concreto. Estas microevaluaciones irán directamente relacionadas con los microaprendizajes previamente citados, pudiendo ser evaluados uno o varios de los contenidos presentes en estas lecciones en función de su grado de complejidad y al nivel de desempeño que ya se suponga que deba haber alcanzado un alumno en un periodo determinado del curso.

Finalmente, se van a desarrollar dinámicas de gamificación para conseguir los dos últimos objetivos citados en este apartado. La gamificación irá principalmente enfocada para trabajar la motivación del estudiante como objetivo principal y a desarrollar la concienciación de la necesidad de dominio de su entorno de desarrollo como objetivo secundario. Estas dinámicas de gamificación van estrechamente relacionadas con los procesos de microevaluación, siendo la puntuación obtenida en estas últimas el principal factor que aplica a la valoración en el sistema diseñado.

Como conclusión, este proyecto pretende involucrar al alumno en una metodología de trabajo innovadora y que pretende ser más efectiva para conseguir mejorar el proceso de aprendizaje de programación informática. Las técnicas y metodologías que se han expuesto pueden aplicarse de manera independiente en esta y otras especialidades formativas, pero su unión dentro de una misma propuesta dentro de la asignatura de Programación se considera que puede potenciar su eficacia. En el siguiente apartado nos enfocaremos en exponer los fundamentos teóricos que darán soporte a este proyecto.

## 2. MARCO TEÓRICO

Comenzamos este apartado estableciendo el contexto legal de la asignatura en la que vamos a enmarcar nuestro proyecto. En el *Real Decreto 686/2010, de 20 de mayo, por el que se establece el título de Técnico Superior en Desarrollo de Aplicaciones Web y se fijan sus enseñanzas mínimas* podemos encontrar los objetivos y competencias generales que esta titulación persigue alcanzar en el alumno. En el *Anexo I* de dicho ordenamiento, más concretamente en la página 50027, podemos observar dentro del módulo profesional de Programación el desglose tanto de los resultados de aprendizaje como de los criterios de evaluación a considerar. En primer lugar, es destacable que el peso de este módulo es superior al resto (14 ECTS, 135 horas), exceptuando únicamente la formación en los centros de trabajo. Esto verifica la importancia que se le debe dar a la habilidad de programar dentro de la formación.

En este documento también tenemos desglosados los contenidos básicos que serán la base a partir de los cuales se diseñarán los microaprendizajes desarrollados dentro de la metodología que se elaborará en este proyecto. Partimos de que el citado Real Decreto establece los contenidos básicos a nivel estatal, siendo posible su detalle dentro de cada Decreto en las distintas comunidades autónomas. Vemos suficiente con abordar los contenidos a nivel estatal por su homogeneidad entre regiones y por tener incluidos los conocimientos importantes necesarios para la formación en la asignatura. Respecto a los criterios de evaluación, también podemos detectar una correspondencia válida entre los mismos y el diseño de las microevaluaciones que se irán desarrollando durante el curso.

### 2.1. Área de trabajo

En los inicios de la década de los 80, desde el Departamento de Psicología de la Universidad de California el autor Richard E. Mayer publicó un artículo científico en el cual se analizaba como incrementar el aprendizaje de la programación informática en alumnos que se enfrentaban a este conocimiento por primera vez (Mayer, 1981). Destacamos de esta publicación la diferenciación entre el aprendizaje basado en la memoria y el entendimiento puro (*Rote*

*Learning vs Understanding*). Estos conceptos los hemos citado anteriormente como comprensión instrumental y comprensión relacional. Se hace visible en el estudio la necesidad de conseguir una comprensión relacional para hacer efectivo el aprendizaje, aspecto que, como hemos comentado en nuestra justificación, es crucial para el buen desempeño en esta asignatura.

Unos años más tarde, se publicó otro interesante artículo en este caso desde la Universidad de Dayton, que ponía el foco en el proceso de transformación de un programador novel a un programador experto (Winslow, 1996). Desde una perspectiva psicológica, esta publicación concluye y hace énfasis en que para alcanzar esta meta la clave es práctica, práctica y más práctica, comenzando por tareas sencillas y evolucionando poco a poco a tareas más complejas. También establece una cadena de tres pasos recomendados para ir avanzando material, siendo estos el aprender un concepto nuevo de programación cada vez, aprender habilidades de diseño combinando a la vez viejas y nuevas estrategias y por último aplicar este nuevo conocimiento en un problema real. Vemos una relación directa de encajar el primero de los pasos de la cadena a través de microaprendizajes que se ocupen de enseñar este nuevo concepto.

Continuando ya en nuestro siglo, encontramos nuevos estudios que ahondan en los conceptos ya comentados. Es interesante el artículo *Learning and Teaching Programming: A Review and Discussion* (Robins et al., 2003), al abordar este tema desde el punto de vista de la docencia. Varios son los conceptos a destacar en esta publicación. Inicialmente, establece una diferenciación entre el diferentes tipos de estudiantes noveles en función de sus capacidades cognitivas y habilidades previas en la programación. Con independencia de esta base, se destaca como factor influyente la predisposición y motivación previa de cada alumno, la cual hace más o menos efectivo el aprendizaje de la materia. Uno de las conclusiones de dicha publicación es precisamente la importancia de poner el foco en mejorar esta motivación con la finalidad de comprometer al estudiante en su proceso de aprendizaje.

Otro de los conceptos clave que tenemos que definir en nuestra propuesta es el pensamiento lógico. En el libro *Brain power: Learn to improve your thinking*

*skills* (Albrecht, 2009) el autor define las bases del pensamiento lógico como la capacidad de pensar secuencialmente. Este proceso implica el tomar las ideas, hechos y conclusiones importantes involucradas en un problema y ser capaz de secuenciarlas de manera encadenada para darles un significado de manera conjunta. Básicamente, pensar de manera lógica es pensar cómo actuar paso a paso. En la publicación *Algorithmic and Logical Thinking Development: base of programming skills* (Milková & Hulková, 2013), se realiza la importancia de desarrollar este pensamiento lógico en relación con las habilidades de la programación, y establece varios juegos y tareas que pretenden ser de inspiración para el educador en esta materia.

Nos encontramos con artículos más actuales que también van esta línea. China es uno de los países que comienza a experimentar con la impartición de asignaturas introductorias a la programación en la escuela básica, enfocados no a la habilidad de programar en sí, si no al desarrollo de lo que llaman en un estudio pensamiento computacional (Wei et al., 2021). En esta publicación vemos como se hace palpable que el desarrollo de este pensamiento computacional involucra un amplio rango de procesos intelectuales, en los que destaca el propio pensamiento lógico, así como la selección de algoritmos y pensamiento sistemático. Se subraya el hecho de que este tipo de pensamiento se convierte a día de hoy en una competencia básica para el estudiante, a la altura de leer, escribir o el cálculo aritmético.

Llegados a este punto, ya podemos extraer algunos aspectos importantes necesarios para mejorar el proceso de aprendizaje de los alumnos de nuestra materia. Por un lado, la importancia de la práctica para la mejora de las habilidades unida a un desarrollo del pensamiento lógico. Por otro, la motivación que puede verse en este contexto como el motor que hace funcionar de manera efectiva los procesos necesarios para el aprendizaje significativo.

## **2.2. Necesidad de mejora en el proceso de aprendizaje**

En el año 2007, se publicó un estudio en Dinamarca que analizaba las tasas de fallo en estudiantes de primer año en ciencias de computación (Bennedsen & Caspersen, 2007). La base de dicho estudio consistió en una serie

de encuestas a lo largo del mundo cuyo objetivo era tener una visión real del porcentaje de estudiantes que matriculados en la asignatura de introducción a la programación fallaban en la superación del curso. Se concluyó con una tasa media de fallo del 33%. Se sugiere que este valor se justifica por diferentes causas, todas relacionados con diferentes características del curso de enseñanza en cuestión tales como el número de alumnos o el paradigma de programación utilizado en la enseñanza. Lo interesante de este estudio es que, 12 años más tarde, se replicó por los mismos profesionales (Bennedsen & Caspersen, 2019) utilizando tres veces más encuestados. En este caso se observó una ligera mejora, una tasa del 28% de fallo, justificando que la metodología de enseñanza ha podido ser pulida a lo largo de estos años en búsqueda de conseguir una mejora en el proceso. Aunque no se destaca este porcentaje como demasiado alto, se considera un valor muy mejorable y por tanto este último estudio concluye que la enseñanza introductoria a la programación es, a día de hoy, uno de los grandes retos de la educación resaltando la necesidad de que la comunidad educativa debería hacer un gran esfuerzo en desarrollar métodos de aprendizaje más efectivos para esta materia.

También se han publicado varios artículos centrados en la motivación del alumno de programación. Se visualiza una serie de carencias en este sentido, siendo una de ellas el hecho de que el estudiante no disfruta la experiencia de aprender a programar (Jenkins, 2001). Si el alumnado no disfruta, difícilmente se comprometerá con las actividades necesarias en este proceso. También se ha analizado el hecho de que mientras los estudiantes con una motivación extrínseca y predisposición positiva antes de cursar programación tienen posibilidades de alcanzar buenas calificaciones y habilidades técnicas, son los estudiantes movidos por una motivación intrínseca los que muestran una actitud y comportamiento más favorable para alcanzar mejores resultados (Zainal et al., 2012).

En España, desde la Universitat Oberta de Catalunya, tenemos un estudio sobre esta problemática mucho más completo y reciente (Marco-Galindo et al., 2021). Centrándose de nuevo en una asignatura introductoria de la programación, y apoyado en varios estudios, se desglosan los factores que propician la no superación del curso. El primero de ellos vuelve a ser la falta de

motivación y el hecho de que el estudiante que no busca información adicional ni intenta superar los retos que se les presentan en la asignatura. Otro es la convención del alumnado sobre la dificultad de aprender a programar. Muy destacable es la detección de que el alumno utiliza técnicas inapropiadas para la programación, citando explícitamente la memorización. Aquí se observa de nuevo una problemática comentada anteriormente, y es la necesidad de desarrollar una comprensión relacional basada en el uso del pensamiento lógico en oposición a una comprensión instrumental basado en intentar aplicar conocimientos desestructurados y memorizados sin comprender el funcionamiento del todo. También aparece en este estudio un punto referente al entorno de programación, el cual es detectado por parte del alumnado como difícil de instalar y utilizar.

Poniendo en orden toda la documentación expuesta hasta ahora, tenemos ya visible una línea de trabajo. Se detecta una vía de mejora en hacer más atractivo el proceso de aprendizaje en la asignatura de Programación. En los siguientes puntos nos centraremos en la metodología utilizada para tal efecto. En todo el proceso de enseñanza, se deberá tener presente una didáctica basada en el desarrollo del pensamiento lógico evitando en todo momento que el alumno caiga en prácticas ineficaces en este campo de aprendizaje, como el abuso de la memorización.

Antes de finalizar esta sección, citemos otro de los puntos que vamos a abordar en metodología, la evaluación. Se ha estudiado el sentimiento de ansiedad que presentan los estudiantes ante los exámenes, así como intervenciones psicopedagógicas para su tratamiento (Bausela Herreras, 2005). Una de las razones principales por las que se produce este fenómeno es la planificación temporal por parte del alumno. Son comunes en este sentido los malos hábitos de estudio, como la intensificación del mismo los días o día previo a una prueba evaluativa. En lo que respecta a la programación, este tipo de hábito de trabajar la materia horas previas al examen tiene un impacto aún menor en el resultado que en otras asignaturas, siendo otra de las líneas de actuación de este proyecto el abordar un enfoque diferente.

Otro estudio más reciente se enfoca en los diferentes factores que afectan al estrés del estudiante de secundaria en sus últimos años de formación (Wuthrich et al., 2020). En relación con las evaluaciones, se puede observar un incremento del nivel de estrés en las semanas previa al periodo de exámenes y una disminución del mismo en los cinco días posteriores a este periodo. Es decir, tenemos una relación directa entre este nivel de estrés y el periodo evaluativo cuando estamos calendarizando dicho periodo en un momento concreto del curso. La eliminación de estas barreras temporales con el diseño de microevaluaciones repartidas a lo largo del curso pueden ayudar a normalizar la evaluación y a disminuir el nivel de estrés del alumno.

### 2.3. Metodología y estrategias de aprendizaje

En el año 2019 se publicó el libro *The Pragmatic Programmer: Your Journey to Mastery* (Thomas & Hunt, 2019). Este era una edición especial 20 aniversario de otro libro publicado en 1999 por los mismos autores y basado en los mismos principios. Básicamente, en esta publicación nos encontramos con una recopilación de consejos y buenas prácticas para mejorar en el desempeño de la programación. Uno de los conceptos presentes en el libro sirve de base para una de las claves que vamos a desarrollar en nuestra metodología, los microaprendizajes. En este libro, el concepto en cuestión es lo que se denomina *code katas*, o katas de programación. La definición de este concepto, que proviene en su origen de las artes marciales japonesas, es la presentación de ejercicios de programación cuya finalidad es ser realizados repetidamente para mejorar esta habilidad. Es decir, se centra en la importancia de practicar con ejercicios reales como estrategia inicial de aprendizaje.

Una de las características de estos katas de programación es su naturaleza incremental. En cuanto a extensión y complejidad, se comienza desde un nivel básico con problemas sencillos que se van enrevesando conforme el alumno va superando los mismos. Es crucial por tanto ir desarrollando estas tareas más sencillas para ir avanzando. Nuestra metodología seguirá esta misma filosofía de actuación ya que los microaprendizajes que van a ser diseñados tendrán como objetivo principal el aprender las piezas básicas que, junto a la práctica de las mismas, irán

desarrollando en el alumno un nivel de desempeño en programación cada vez mayor. Por tanto, nuestra metodología se basará en el consumo de estos microaprendizajes como la presentación principal de contenidos para posteriormente trabajarlos mediante ejercicios prácticos de complejidad incremental con la finalidad de ir desarrollando el pensamiento lógico que mejorará la eficacia en el proceso de aprendizaje.

Centrándonos en el propio microaprendizaje, podemos definirlo como una manera de presentación de contenido en cantidades breves y con una extensión limitada, ya sea varias veces a la semana o diariamente (Reyes, 2018). Es un concepto totalmente vinculado a la educación y el aprendizaje. Erróneamente se considera que hay una vinculación directa entre la presentación de este contenido y su consumo en dispositivos móviles, pero esto no siempre es así. Existen varias maneras de ofrecer este tipo de lecciones, cuestión que profundizaremos en el apartado de metodología de este proyecto. Estas características han hecho que esta estrategia sea muy efectiva en ciertos entornos, siendo una tendencia en aquellos en los que no es posible una dedicación al aprendizaje en horario completo, como ocurre en ciertos entornos laborales (Leong et al., 2021).

Llegados a este punto, vamos a desarrollar los fundamentos teóricos de otro de los pilares de nuestro proyecto, la gamificación. En el año 2014, el autor Ferran Teixes publicó el libro *Gamificación: fundamentos y aplicaciones* (Teixes, 2014). Una definición de esta metodología que podemos encontrar en el libro es la de una estrategia de aplicación de recursos relacionados con juegos tales como elementos, diseño y dinámicas en contextos no lúdicos con la finalidad de actuar sobre la motivación del individuo y conseguir modificar su comportamiento. La primera parte del libro ahonda en los principales fundamentos psicológicos que sustentan esta metodología para finalizar con algunas ideas de aplicación en diversos ámbitos, incluida la educación.

Existen otras publicaciones más centradas en el propio diseño de las estrategias de gamificación. En Estados Unidos, fue publicado un interesante artículo que desglosaba un total de ocho posibles diseños de juego aplicados a estrategias de gamificación en el ámbito de la educación (Nah et al., 2014). El

uso de estas estrategias están documentadas en base a otras publicaciones científicas que avalan su efectividad. En nuestro proyecto, pretendemos combinar dos de estas dinámicas en el diseño de nuestro sistema: barras de progreso y niveles o etapas. En este apartado profundizaremos en la metodología pero, podemos adelantar en este punto, que nuestro objetivo será que el sistema de progresión sea multidimensional, no lineal, al considerar efectivo que el alumno tenga diferentes líneas posibles de mejora que puedan ser desarrolladas de manera independiente con un objetivo común.

Los autores Patrick Buckley y Elaine Doyle publicaron hace pocos años un completo artículo en el cual se analizaba y desglosaba el impacto de la gamificación sobre la motivación de estudiante (Buckley & Doyle, 2016). Como hemos comentado en nuestra justificación, la aplicación de la gamificación persigue incentivar la motivación intrínseca del estudiante. Este estudio corrobora esta efectividad detectando una correlación positiva entre la participación en el proceso de enseñanza basado en esta metodología y la existencia de este tipo de motivación. En definitiva, este artículo hace visible el potencial que tiene la gamificación en el aprendizaje del estudiante además de destacar la figura del profesor como pieza clave en la conducción y el aprendizaje del proceso metodológico.

Nos encontramos ya en un punto dónde podemos esbozar la línea de trabajo que vamos a desarrollar en la sección dedicada a la metodología. Para empezar, nos encontramos con la necesidad de diseñar una serie de microaprendizajes, concebidos en base a los contenidos mínimos requeridos en la asignatura de Programación y que tendrán dos características: presentación que facilite su consumo y su poca extensión. Estos microaprendizajes serán trabajados en clase mediante tareas que irán incrementando sucesivamente su complejidad y que irán enfocadas al desarrollo del pensamiento lógico del estudiante. Por último, estos contenidos serán evaluados mediante microevaluaciones secuenciados a lo largo del curso para ir teniendo conocimiento del nivel de adquisición de contenido logrado por el alumnado. Tareas y microevaluaciones estarán desarrolladas bajo una metodología basada en gamificación con la finalidad de conseguir implicar y aumentar la motivación del estudiante intensificando la eficacia del proceso de aprendizaje.

### 3. OBJETIVOS

En los apartados anteriores hemos realizado un análisis exhaustivo de las necesidades detectadas en el proceso de enseñanza de la asignatura de Programación, así como justificar las estrategias que queremos desarrollar. Será esta sección donde formalicemos los objetivos que pretenden dar solución a dichas problemáticas.

#### 3.1. Objetivo General

El objetivo general que persigue este proyecto viene definido a continuación:

“Aumentar la eficacia en el proceso de aprendizaje de la asignatura de programación mediante la aplicación de microaprendizaje y gamificación”.

#### 3.2. Objetivos Específicos

Los objetivos específicos que persiguen la consecución del objetivo general van enfocados siempre al desempeño del alumnado. Son los siguientes:

1. Reforzar el razonamiento lógico en base a microaprendizajes.
2. Mejorar la motivación en el estudio de la asignatura de programación a través de un sistema de gamificación.
3. Favorecer la percepción positiva del proceso de evaluación a través de las microevaluaciones.



#### 4. METODOLOGÍA

En este apartado se va a desarrollar en detalle este proyecto educativo. Partimos de tres objetivos específicos que pretendemos alcanzar para la consecución de un objetivo general. El primero de estos objetivos, reforzar el pensamiento lógico, será perseguido a través de las tareas que vamos a describir. El segundo de los objetivos, la mejora de la motivación, se pretende alcanzar en base a las estrategias de gamificación que engloban toda la metodología planteada. El tercer objetivo, favorecer la percepción sobre el proceso evaluativo, se pretende que sea alcanzado a través de la naturaleza de la evaluación planteada para esas actividades, las microevaluaciones. En cada apartado específico ahondaremos en cada una de estas metas y analizaremos como cada uno de estos objetivos concretos tendrá una relación directa con los demás.

Se ha considerado como punto fuerte de innovación en este proyecto la introducción de una evaluación continua basada en microevaluaciones calendarizadas a lo largo del curso. Tras la búsqueda de estudios o artículos relacionados con este tipo de metodología evaluativa no se han encontrado precedentes en esta línea. La integración de este sistema con una metodología basada en estrategias de gamificación supone un plus innovador a la par que retroalimenta la consecución del segundo de nuestros objetivos específicos, la mejora de la motivación. Es decir, se pretende que un alumno se sienta más motivado en la propia realización de la evaluación considerando que ese resultado va a suponer una mejora en su puntuación dentro del sistema.

Como hemos mencionado anteriormente, vamos a desarrollar nuestro proyecto dentro de la asignatura de Programación del Ciclo Formativo de Grado Superior en Desarrollo de Aplicaciones Web. En base al currículo actual de este módulo, plantearemos desarrollar unos contenidos que tendrán una correspondencia con los contenidos básicos definidos en el *Real Decreto 686/2010, de 20 de mayo, por el que se establece el título de Técnico Superior en Desarrollo de Aplicaciones Web*. Aunque nuestro proyecto pretende desarrollar todos estos contenidos a lo largo del curso, en este documento

concretaremos solo una parte de los mismos por no exceder innecesariamente la extensión del mismo.

#### 4.1. Contenidos

Vamos a concretar en cuatro los contenidos principales que se incluyen en nuestro proyecto. Tres de ellos han sido ya presentados en apartados anteriores y son microaprendizajes, microevaluaciones y el diseño de un sistema de gamificación. Respecto a los microaprendizajes, estos derivarán en el cuarto contenido a desarrollar, las tareas a elaborar en clase en base a estas lecciones. En este apartado vamos a describir en más detalle las características de estos contenidos y a relacionar su utilidad con la consecución de los objetivos específicos del proyecto.

##### 4.1.1. Microaprendizajes

Suponen una piedra angular en la metodología planteada. Se definen como lecciones de contenidos prácticos de programación que cumplirán al menos tres características. La primera, no deben enseñar más de un concepto nuevo cada vez. Se partirá por tanto de lecciones que desarrollen los contenidos más básicos y se irá evolucionando a lecciones que, dando por conocidos estos contenidos base, desarrollarán conceptos más avanzados. La segunda de las características es su presentación audiovisual. Dichas lecciones deberán ser un vídeo cuyo formato dependerá del medio en el que vaya a ser consumido, siendo el estándar un formato horizontal apropiado para verse en un monitor de ordenador. La tercera y última de las características será su duración, está no deberá superar los 30 segundos. Habrá casos concretos en que algún contenido pueda superar este límite, pero deberá evitarse en la manera de lo posible.

Un ejemplo de microaprendizaje sería un vídeo editado por el profesor en el cual grabe la pantalla de su ordenador mientras, en el entorno de desarrollo utilizado en la asignatura y con una clase de programación orientada a objetos ya creada, explique una estructura simple donde se indique lo que es el nombre de dicha clase, sus atributos, sus métodos y la identificación de llaves como elementos agrupadores de algunos de estos elementos. En la siguiente tabla vemos este ejemplo de manera más esquemática.

Tabla 1. *Ejemplo de microaprendizaje*

<b>Duración</b>	<ul style="list-style-type: none"> <li>• 30 segundos</li> </ul>
<b>Formato</b>	<ul style="list-style-type: none"> <li>• Vídeo grabado horizontalmente</li> </ul>
<b>Contenido</b>	<ul style="list-style-type: none"> <li>• Pantalla del profesor donde comparte el entorno de desarrollo con un código fuente de apoyo ya creado</li> </ul>
<b>Contenido curricular</b>	<ul style="list-style-type: none"> <li>• Se explican los conceptos de una clase en programación orientado a objetos</li> </ul>

#### 4.1.2. Tareas

Apoyándose en los microaprendizajes como base para el diseño de actividades, estas se corresponderán con tres tipos de tareas que el alumno tendrá que realizar en clase. El primer tipo será el de tareas sencillas que tendrán una dificultad similar al del propio microaprendizaje. Inspiradas en los katas de programación que hemos citado en nuestro marco teórico, estas tareas serán diferentes ejemplos del mismo contenido impartido en el microaprendizaje y su finalidad principal será el de ser una técnica de entrenamiento para desarrollar el pensamiento lógico y mejorar en la habilidad de programar. Esto implica que sean las responsables de alcanzar el primero de nuestros objetivos específicos (OE 1). Será labor del profesor el diseño de dichas tareas para, ejemplificando con diferentes variaciones y siendo fiel al contenido que se pretende inculcar, sirvan de herramienta al alumno para alcanzar el dominio de dicha lección.

Un segundo tipo de tareas corresponderá a ejercicios prácticos más elaborados que integren los contenidos básicos trabajados en el primer tipo de tarea. Su finalidad será la misma, trabajarlos de manera continua con el fin de ir progresando en la asignatura. Por supuesto, estas tareas tendrán una complejidad y extensión mayor a las primeras, siendo un reto posterior tras la superación de las tareas de tipo uno. Su realización por tanto se verá incentivada

en el sistema de gamificación, como detallaremos en el apartado dedicado al mismo.

Tendremos un tercer tipo de tareas que corresponderá a contenidos transversales dentro de la programación. Estos contenidos se enfocarán en mejorar el desempeño del alumno en lo que al manejo de su entorno de desarrollo se refiere. Los contenidos en estas tareas no tendrán una correspondencia con los contenidos curriculares, pero si tendrán un peso en el sistema de gamificación. Su finalidad por tanto es doble ya que por un lado su superación será un incentivo más en la motivación del alumno y por otro se aumentará la productividad como programador al mejorar el dominio que tendrá sobre una herramienta de trabajo concreta. Destacamos que este tipo de tareas serán de una dificultad menor que las anteriores, siendo básicamente su aprendizaje memorístico el único factor necesario.

Un ejemplo de tarea de tipo uno, relacionado con el ejemplo del microaprendizaje anterior, sería el pedir al alumno que cree una clase *Gato* en *Java* que tenga los atributos *raza* y *color*, ambos de tipo *String*, además de un método llamado *maullar()*, el cual no recibirá argumentos de entrada ni retornará ningún valor y en cuyo cuerpo imprimirá por pantalla el texto "*miau!!*". Esta misma tarea puede servir de base para una tarea de tipo dos que, además de pedir al alumno que realice la creación de esta clase, la utilice en una estructura de datos tipo lista para añadir diferentes objetos de tipo gato a misma y luego buscar uno que tenga un color concreto.

A continuación se expone una tabla con la ejemplificación de tres tareas de cada uno de los tipos:

Tabla 2. *Ejemplo de tareas*

Tipo	Descripción
1	<ul style="list-style-type: none"> <li>• Crear clase <i>Ave</i>.</li> <li>• Debe tener los atributos <i>tipo</i> y <i>color</i> de tipo <i>String</i>.</li> <li>• Crear un método <i>descripción()</i> en esta clase. Deberá imprimir por pantalla el texto "<i>El (ave) es de color (color)</i>".</li> </ul>

2	<ul style="list-style-type: none"> <li>• Crear clase <i>Alumno</i>.</li> <li>• Debe tener los atributos <i>edad</i> y <i>nombre</i> de tipo <i>Integer</i> y <i>String</i> respectivamente.</li> <li>• Crear un método <i>presentarse()</i> en esta clase. Deberá imprimir por pantalla el mensaje “<i>Me llamo (nombre) y tengo (edad) años</i>”.</li> <li>• Crear una estructura de datos para almacenar una lista de alumnos.</li> <li>• Crear dos objetos de tipo alumno y almacenarlos en dicha estructura.</li> <li>• Crear un método que recorra dicha lista e imprima los alumnos (con todos sus atributos) por pantalla.</li> </ul>
3	<ul style="list-style-type: none"> <li>• Realizar una búsqueda de fichero en el entorno de desarrollo mediante el uso de un atajo de teclado.</li> </ul>

#### 4.1.3. Microevaluaciones

Las microevaluaciones son otro de los aspectos relevantes del proyecto. Se trata de pequeños exámenes que el alumno realizará en un tiempo reducido, aproximadamente de 15 minutos, y cuyo contenido será una variación de las tareas que hemos descrito en el apartado anterior. La modalidad de examen será la de una o más tareas prácticas con una dificultad o complejidad similar a las que se han trabajado en clase. Su combinación dentro de una misma prueba dependerá de la dificultad de estas tareas. Por ejemplo, será posible combinar dos tareas sencillas de tipo uno con una de tipo tres, o realizar una prueba con una única tarea de tipo dos. Otra característica destacable de las evaluaciones es que no serán realizadas a la misma vez por toda la clase. Se organizará su realización en pequeños grupos dentro de un periodo definido como veremos en nuestra temporalización. Este tipo de metodología de evaluación irá enfocada a la consecución de nuestro tercer objetivo específico (OE 3).

Un ejemplo de microevaluación viene incluido al final de este documento (ver Anexo I). Básicamente, nos encontramos con una combinación de dos

tareas: una de tipo uno y otra de tipo tres. La tarea de tipo tres debe poder hacerse si el alumno ha visualizado y trabajado el microaprendizaje correspondiente. En este caso concreto, este podría ser un vídeo grabado por el profesor donde se enseñe como formatear el código fuente en el entorno de desarrollo utilizado en la asignatura a través de un atajo de teclado.

Para finalizar este apartado, se hace hincapié de nuevo que este tipo de pruebas serían el punto principal de innovación en este proyecto. La naturaleza innovadora se considera que viene dada por estas tres características que pasamos a enumerar:

1. Duración reducida con respecto a pruebas evaluativas más tradicionales.
2. Se reparten a lo largo del curso en las diferentes unidades didácticas como las ejemplificadas en el apartado 4.3 dedicado a la temporalización.
3. Las microevaluaciones, al estar a su vez formadas por diferentes tareas, pueden combinarse de manera que distintos alumnos puedan realizar pruebas diferentes en un momento concreto aunque deban haber realizado las mismas tareas a lo largo de una unidad didáctica concreta.

#### *4.1.4. Sistema de gamificación*

Será el eje central alrededor del cual orbitarán el resto de elementos de nuestro proyecto. Antes de nada, centrémonos en su estructura. Como hemos citado en el marco teórico, tendremos un diseño de gamificación en dos dimensiones. Por un lado, tendremos diferentes niveles o etapas y por otro tendremos una barra de progreso asociada a cada uno de estos. Cada uno de los niveles irá destinado a puntuar el desempeño del alumno en un tipo de trabajo concreto, siendo el peso de cada nivel diferente y más o menos relevante para la superación de la asignatura. Es importante que las puntuaciones de los alumnos sean visibles dentro del aula, con el fin de introducir el factor competitivo propio de las dinámicas de gamificación. La puntuación del alumno comenzará a cero al inicio de curso, e irá aumentando durante el transcurso del mismo a

través de las diferentes evaluaciones. La consecución del segundo de nuestros objetivos específicos va unida directamente con este sistema (OE 2).

Tendremos tres tipos de niveles (ver Anexo II). El nivel uno será el correspondiente a tareas de tipo uno y dos. El alumno recibirá puntuación por cada tarea que complete y entregue al profesor. Tendrá una puntuación de un punto cada tarea de tipo uno que el alumno complete, y una puntuación que pueda variar entre dos y cinco puntos por cada tarea de tipo dos, en función de la complejidad que le considera a dicha tarea. Es decir, en este primer nivel se pretende incentivar el trabajo práctico del alumno. La barra de progreso asociada a este nivel tendrá una puntuación máxima conocida en función de la totalidad de tareas de estos tipos que el profesor tenga planificado en el curso académico. En este nivel vemos la relación entre nuestros objetivos específicos uno y dos: el alumno se verá motivado a realizar tareas para progresar en el sistema, influyendo esto en el desarrollo de su pensamiento lógico.

El nivel dos corresponde a la valoración de las tareas de tipo tres. Como hemos citado anteriormente, la naturaleza de estas tareas es la de evaluar el conocimiento que tiene el alumno sobre el entorno del desarrollo. Son por tanto tareas que se podrían considerar como adicionales o extras para que el alumno pueda conseguir un plus en el sistema de puntuación a la vez que obtiene un beneficio valioso para su desempeño en la asignatura. Al contrario que en el primer nivel, en el cual las tareas puntúan en el momento que son entregadas, en este segundo nivel las tareas de este tipo solo son evaluadas durante las microevaluaciones, siendo su superación en ese momento la que aumentará la puntuación.

El último y tercer nivel será el correspondiente a la evaluación real de los contenidos. Durante las microevaluaciones, el alumno intentará superar una o varias tareas las cuales llevarán asociadas una puntuación en función de su complejidad. La consecución de cada ejercicio aumentará por tanto la barra de progreso de este nivel. Recordemos que en este punto se evaluarían solo las tareas de tipo uno y dos que tengamos dentro de la prueba evaluativa, las cuales tienen una correspondencia directa con los contenidos curriculares de la asignatura. Por ejemplo, si en una microevaluación ponemos dos tareas de tipo

uno y otra de tipo tres, la superación de las dos primeras aumentaría la barra de progreso de este nivel tres pero la consecución de la última aumentaría el del nivel dos. Este tercer nivel será el más influyente de cara a la evaluación final del alumno.

Por ejemplo, un alumno realiza durante una semana un total de cinco tareas de clase. De todas estas, tres eran tareas de tipo uno y dos de ellas eran de tipo dos. De estas dos tareas de tipo dos, una tenía una puntuación de dos al ser relativamente simple y otra una puntuación de cuatro, algo más compleja. Este alumno habrá por tanto sumado una puntuación de nueve puntos en su nivel uno al superar las cinco tareas (de puntuaciones uno, uno, uno, dos y cuatro).

Este mismo alumno del ejemplo anterior, ahora se presenta a la microevaluación que hemos propuesto en este documento (ver Anexo I). Se le asigna un total de cinco puntos al ejercicio uno de esta prueba, y tres al ejercicio dos. La superación de ambos ejercicios supondrá un incremento de la puntuación del alumno en tres puntos en su nivel dos (el ejercicio dos era una tarea de tipo tres, secundaria) y de cinco puntos en su nivel tres (el ejercicio uno era una tarea relacionada con los contenidos obligatorios, principal).

En la siguiente tabla mostramos de manera más esquemática el valor que aportan las diferentes tareas al sistema de gamificación:

Tabla 3. *Puntuación en función del tipo de tarea*

<b>Tipo de tarea</b>	<b>Lugar donde se realiza</b>	<b>Puntuación</b>	<b>Nivel al que aplica</b>
1	<ul style="list-style-type: none"> <li>Clase o casa</li> </ul>	1	1
1	<ul style="list-style-type: none"> <li>Microevaluación</li> </ul>	5	3
2	<ul style="list-style-type: none"> <li>Clase o casa</li> </ul>	2 a 5	1
2	<ul style="list-style-type: none"> <li>Microevaluación</li> </ul>	6 a 10	3

3	• Microevaluación	3	2
---	-------------------	---	---

#### 4.2. Contenidos curriculares

Respecto a los contenidos curriculares que abordaremos, vamos a optar por enmarcar el proyecto dentro de una hipotética unidad en la que se trabaje el concepto de clases dentro de un lenguaje de programación orientado a objetos. Teniendo como base el Real Decreto que hemos citado anteriormente, y enfocándonos en el apartado cuarto de los contenidos básicos de la asignatura de Programación, vamos a elaborar en la siguiente sección una línea de trabajo que abarque los siguientes cinco contenidos:

- Concepto de clase
- Estructura y miembros de una clase
- Creación de atributos
- Creación de métodos
- Creación de constructores

La suposición de que estos contenidos forman parte de una unidad didáctica concreta será un aspecto importante dado que la línea de trabajo que expondremos en la siguiente sección se mantendrá entre las diferentes unidades.

#### 4.3. Temporalización y actividades

En este apartado, vamos a comenzar describiendo la estructura que tendrá nuestra metodología para poner en relación todos los contenidos presentados anteriormente. Tras esto, tendremos una visión de la temporalización para cada uno de los contenidos curriculares que se han planteado, lo que facilitará la calendarización a mayor escala (todo el curso o una evaluación completa), incluyendo todos los contenidos que tengamos organizados dentro de una unidad didáctica específica, como los enumerados

en la sección anterior. Cada una de las unidades se desarrollará en varias sesiones, siendo una sesión una clase presencial de la asignatura con una duración de 55 minutos. Se da por supuesto que al inicio de curso habrá una sesión exclusiva dedicada a la explicación de la metodología de trabajo al alumnado.

Dentro de una sesión podremos realizar tres tipos de trabajos en el aula. El primero será la presentación del contenido apoyándonos en un microaprendizaje, el cual estará accesible posteriormente para que el alumno pueda repasar la lección. Se acompañará la exposición de este microaprendizaje con una breve clase magistral que aclare dudas sobre el mismo o amplíe lo que vea necesario. El segundo tipo de trabajo será la presentación de la tarea o tareas que el alumno realizará en clase y su posterior realización. El papel del profesor en este punto será el de resolución de dudas durante el trabajo de su alumnado. Paralelamente a la realización de las tareas en el aula, podrán establecerse un periodo de microevaluaciones donde un grupo reducido de cinco alumnos podrá realizar una prueba mientras el resto realiza las tareas. El objetivo de que el grupo sea reducido es que el profesor pueda estar dedicado a vigilar la realización del examen, con el fin de comprobar la aplicación de los conocimientos evaluados.

Tras esta introducción se presenta una tabla donde tengamos visibles los cuatro tipos de actividades que se van a trabajar en las sesiones así como el objetivo específico que se trabaja en ellas (en el caso que corresponda):

Tabla 4. Tipo de actividades y contenidos específicos que trabajan

<b>Actividad</b>	<b>Objetivo específico</b>
1. Explicación de los conceptos apoyado en el microaprendizaje	-
2. Presentación y explicación de tareas relacionadas con los contenidos	-
3. Trabajo de las tareas en clase	1 y 2

4. Realización de microevaluaciones en clase	3
--	---

Con el fin de clarificar todo el proceso que acabamos de describir, vamos a realizar una temporalización esquemática en el que se aborden los cinco contenidos curriculares que hemos citado anteriormente más otros tres contenidos transversales respecto al manejo del entorno de desarrollo. Desglosaremos las actividades realizadas en cada sesión, con un total de ocho sesiones, además de indicar entre paréntesis el tiempo dedicado a cada una y los objetivos específicos trabajados, utilizando la nomenclatura OE N siendo N el número del mismo. Vamos a suponer también que trabajaremos sobre una clase con 30 alumnos.

Se enmarca esta temporalización dentro de una unidad didáctica concreta, cuyos contenidos han sido desglosados en el apartado 4.2. En nuestra planificación anual, por tanto, se seguirá un esquema idéntico al aquí descrito pero replicándose por cada una de las unidades didácticas que queramos tener organizadas en el curso. A continuación de esta temporalización de ocho sesiones, incluiremos una segunda tabla para concretar las actividades que haremos en una de ellas que tomaremos como ejemplo. Cabe destacar que los recursos necesarios para estas actividades son todos los que están desglosados en el apartado 4.4.

Tabla 5. *Temporalización de unidad didáctica de ejemplo*

Sesión	Actividades
1 (55 minutos)	<ol style="list-style-type: none"> <li>1. Explicación del concepto de clase. (20 minutos)</li> <li>2. Explicación del contenido transversal 1 – Atajo de teclado para formatear un documento. (2 minutos)</li> <li>3. Presentación de una tarea de tipo uno para clase. (5 minutos)</li> </ol>

	<p>4. Dejar a los alumnos trabajar con la tarea. (28 minutos, OE 1, OE 2)</p>
<p>2 (55 minutos)</p>	<p>1. Explicación de estructura y miembros de una clase. (20 minutos)</p> <p>2. Presentación de una tarea de tipo uno para clase. (5 minutos)</p> <p>3. Dejar a los alumnos trabajar con la tarea. (30 minutos, OE 1, OE 2)</p>
<p>3 (55 minutos)</p>	<p>1. Explicación de atributos. (15 minutos)</p> <p>2. Explicación del contenido transversal 2 – Atajo de teclado para seleccionar una línea completa en el entorno . (2 minutos)</p> <p>3. Presentación de una tarea de tipo uno para clase. (5 minutos)</p> <p>4. Dejar a los alumnos trabajar con la tarea. (33 minutos, OE 1, OE 2)</p>
<p>4 (55 minutos)</p>	<p>1. Explicación de métodos. (25 minutos)</p> <p>2. Presentación de una tarea de tipo dos para clase. (5 minutos)</p> <p>3. Dejar a los alumnos trabajar con la tarea. (25 minutos, OE 1, OE 2)</p>
<p>5 (55 minutos)</p>	<p>1. Explicación de constructores. (20 minutos)</p> <p>2. Presentación de una tarea de tipo dos para clase. (5 minutos)</p> <p>3. Dejar a los alumnos trabajar con la tarea. (30 minutos, OE 1, OE 2)</p>

<p>6 (55 minutos)</p>	<ol style="list-style-type: none"> <li>1. Explicación del contenido transversal 3 – Atajo de teclado para navegar a una línea determinada. (2 minutos)</li> <li>2. Presentación de una tarea de tipo dos para clase. (5 minutos)</li> <li>3. Dejar a los alumnos trabajar con la tarea. (48 minutos, OE 1, OE 2)</li> </ol>
<p>7 (55 minutos)</p>	<ol style="list-style-type: none"> <li>1. Presentación de una tarea de tipo dos para clase. (5 minutos)</li> <li>2. Dejar a los alumnos trabajar con la tarea. (50 minutos)</li> <li>3. Paralelamente comienza el periodo de microevaluaciones. Los alumnos serán llamados en grupos de cinco para que cada uno realice su prueba con una duración máxima de 15 minutos. Es decir, en el tiempo asignado podrán realizar las microevaluaciones 15 alumnos, tres turnos de cinco. Dejaremos cinco minutos de margen para la rotación de los tres grupos. Mientras esos cinco alumnos realizan la microevaluación, el resto sigue trabajando en la tarea. (45 minutos, OE 3)</li> </ol>
<p>8 (55 minutos)</p>	<ol style="list-style-type: none"> <li>1. Este sesión tendrá una estructura similar a la anterior, con lo que completaremos el proceso de evaluación de la clase, la evaluación de los restantes 15 alumnos.</li> </ol>

Ahora pasamos a detallar la quinta sesión de la temporalización recién desarrollada. Partimos de que se va a trabajar un contenido curricular concreto, los constructores de objetos en programación, y procedemos a detallar de qué manera se va a impartir dicho contenido a través de la ejemplificación de las actividades que forman parte del trabajo a realizar en el aula.

Tabla 6. Actividades de la quinta sesión de la temporalización de ejemplo

Tipo	Minutos	Descripción
Explicación	20	Se visualizan dos microaprendizajes. El primero consiste en un vídeo donde se observa una grabación de un entorno de desarrollo mientras se crea un constructor sencillo en <i>Java</i> . El segundo consiste en la realización de un constructor algo más avanzado, que reciba dos argumentos de entrada. Tras estas visualizaciones, el profesor profundiza en el concepto y ejemplifica más, esta vez compartiendo su código a través del proyector.
Tarea de clase	5	Se presenta una tarea donde los alumnos tienen que crear una clase <i>Java</i> llamada <i>Persona</i> que tenga dos constructores, uno sin argumentos y el otro con el argumento <i>nombre</i> . Además, esta clase tendrá un atributo llamado <i>nombre</i> de tipo <i>String</i> y un método <i>correr()</i> que recibirá el argumentos de tipo entero <i>pasos</i> y que imprimirá por pantalla el texto: " <i>Corriendo</i> " + <i>pasos</i> + " <i> pasos</i> ".
Trabajo en clase	30	Los alumnos trabajarán en clase la tarea y de este modo estaremos trabajando los objetivos

		específicos uno y dos durante este periodo.
--	--	---

Lo que acabamos de desarrollar es el trabajo programado para el alumno dentro del aula. Puede darse la situación de que este no finalice su tarea en el tiempo asignado, por lo que se dará la oportunidad de entregarla desde casa durante la siguiente semana. Si no se entrega en ese periodo, no se podrá entregar en la convocatoria vigente. Con esto se quiere incentivar el trabajo continuado del alumno, evitando que deje contenidos por trabajar. Existe una posibilidad de que el alumno, al realizar la tarea en casa, la copie o la haga con ayuda externa. Se deberá inculcar desde el minuto uno la mentalidad práctica de la asignatura, y hacer hincapié en que de nada servirá la copia cuando en la microevaluación el alumno se enfrentará solo a la tarea con la vigilancia del profesor.

También existe la posibilidad de que un alumno no se presente a una microevaluación. Si justifica su ausencia podrá recuperar la misma en algún periodo de realización de tareas posterior. Las microevaluaciones deberán abordar todos los contenidos, por lo que, aunque en la temporalización anterior se hayan concentrado en las dos sesiones finales, se da la posibilidad de sesiones futuras realizar las pruebas en periodos dentro de las sesiones dedicados a trabajar tareas. También es posible que en sesiones similares (como las anteriores sesiones 7 y 8) se evalúen contenidos diferentes a diferentes alumnos, aunque finalmente todos tengan que evaluarse de todos los contenidos.

Dentro de nuestra metodología también será posible la realización de tareas extraordinarias a realizar en casa. Estas irán enfocadas a mejorar la puntuación del alumno en el sistema de gamificación, por tanto supondrá un plus para aquellos perfiles que lleven bien la asignatura y quieran progresar tanto en habilidades como en el sistema de gamificación. De este modo estaríamos fomentando la consecución de nuestro segundo objetivo específico, la mejora de la motivación.

#### 4.4. Recursos

A continuación resumimos en una tabla los recursos necesarios para la realización de nuestro proyecto teniendo en cuenta un aula de 30 alumnos. En principio, vamos a necesitar disponer de un espacio físico donde impartir la clase pero además será necesario tener un entorno de almacenamiento online (alojamiento a modo de servidor multimedia para poder visualizar los vídeos, similar a *YouTube*) donde alojemos el material audiovisual que corresponderá con los microaprendizajes. Esta necesidad se justifica por el hecho de que estas lecciones están diseñadas con la mente puesta en que el alumno repase los contenidos en pequeños huecos que tenga, bien antes de una clase, en casa o durante la realización de las tareas, por lo que será necesario que pueda revisarlas en cualquier parte y con cualquier dispositivo con acceso a internet.

Idealmente, y en caso de aumentar los recursos de este proyecto, sería una mejora importante el desarrollo de una aplicación móvil que tuviera integrados tanto los microaprendizajes como un perfil del alumno donde se observe su puntuación. Esta aplicación tendría métricas asociadas al uso de la misma por parte del profesor, como el número de visualizaciones que tiene un microaprendizaje por parte de un alumno, con lo que este tendría más información acerca del trabajo extra del mismo en la asignatura. En esta propuesta inicial de recursos necesarios no incluiremos esta posibilidad pero se considera destacable el hecho de que este proyecto es susceptible a una evolución tecnológica.

Tabla 7. *Recursos necesarios para desarrollar el proyecto*

<b>Tipo</b>	<b>Elemento</b>	<b>Descripción</b>
Espacial	<ul style="list-style-type: none"> <li>• 1 aula</li> </ul>	Aula de informática con conexión a internet.
Humano	<ul style="list-style-type: none"> <li>• 1 profesor</li> </ul>	Profesor que guíe en el proceso del proyecto, explique su funcionamiento y sea participe en la evaluación y

		resolución de dudas del alumno.
Materiales	<ul style="list-style-type: none"> <li>• 30 ordenadores</li> <li>• 1 proyector o pizarra digital</li> </ul>	<p>Ordenadores con el software necesario para trabajar la asignatura en función del lenguaje y el entorno de desarrollo escogidos.</p> <p>Proyector o pizarra digital que permita el profesor compartir su pantalla.</p>
Entorno de almacenamiento online	<ul style="list-style-type: none"> <li>• Plataforma interna, Youtube o similar</li> </ul>	Alojamiento accesible desde el exterior a través de internet en el cual poder alojar los microaprendizajes para su consumo por parte del alumno.

#### 4.5. Atención a la diversidad

A modo esquemático se describen las medidas diseñadas para la adaptación del proyecto a alumnos con necesidades específicas de apoyo educativo:

Tabla 8. *Relación de medidas para atender la diversidad del alumnado*

Tipología de alumnos	Medidas
Con necesidades educativas especiales	<ul style="list-style-type: none"> <li>• Adaptación de los microaprendizajes: mayor extensión, adaptación de formato, etc.</li> <li>• Adaptación del equipamiento informático o espacio de trabajo</li> <li>• Refuerzo individual (fuera o dentro del aula)</li> </ul>

Con dificultades específicas de aprendizaje	<ul style="list-style-type: none"> <li>• Creación de grupo de apoyo durante las actividades de realización de tareas con el fin de reforzar conocimientos</li> <li>• Ampliación de tiempo en las microevaluaciones</li> <li>• Permitir la realización de tareas de clase en grupos de dos alumnos</li> </ul>
Con altas capacidades intelectuales	<ul style="list-style-type: none"> <li>• Ampliación de tareas extra de mayor complejidad con una puntuación especial asignada</li> <li>• Enriquecimiento de contenidos con conceptos más avanzados y no exigibles a nivel curricular</li> </ul>
Con incorporación tardía al sistema educativo español	<ul style="list-style-type: none"> <li>• Apoyo individual</li> <li>• Adaptación en el diseño de las tareas para adaptarse a las condiciones y circunstancias del alumno</li> </ul>
Con condiciones personales que supongan una desventaja educativa	<ul style="list-style-type: none"> <li>• Priorizar el apoyo individual</li> <li>• Agrupamientos flexibles dentro del aula</li> </ul>

#### 4.6. Evaluación de las actividades

En este apartado vamos a enfocarnos en describir el método de evaluación de nuestras actividades a través de los objetivos específicos. La sección estará estructurada en dos partes. Por un lado, tenemos esta primera introducción donde van a ser expuestos los criterios de evaluación de manera general asociados a los diferentes objetivos específicos y por otro tendremos una sección dedicada a analizar la evaluación de cada uno de los mismos.

Se presenta a continuación una tabla resumen donde vamos a realizar una primera aproximación a la evaluación de cada objetivo específico, asociando el instrumento de evaluación elegido así como identificar la tipología de evaluación a la que corresponde dicho instrumento. También se observa el objeto de evaluación, siendo este el alumnado en todos los casos al ser el agente destinatario del proyecto.

Tabla 9. *Resumen de la evaluación de los objetivos específicos*

<b>Objetivo específico</b>	<b>Agente objeto de evaluación</b>	<b>Instrumento de evaluación</b>	<b>Tipología de evaluación</b>
OE1 (razonamiento lógico)	Alumnado	Test especializado	Cuantitativa
OE2 (motivación)	Alumnado	Registro de trabajo y asistencia sistemático	Cuantitativa
OE3 (percepción positiva de la evaluación)	Alumnado	Entrevista personalizada	Cualitativa

#### 4.6.1. *Evaluación del primer objetivo específico*

El primer objetivo específico pretende mejorar el razonamiento lógico del alumno, considerado clave para la eficiencia en la asignatura de programación. En todo caso, el desarrollo de este tipo de pensamiento es independiente de la propia habilidad de programar por lo que para su evaluación vamos a utilizar un instrumento que no tiene que ver con la misma. Se trata de los test de razonamiento lógico dedicados. Este tipo de test utiliza diferentes tipos de preguntas y esquemas, como los famosos diagramas de Venn, para evaluar el nivel de habilidad en este campo.

Será por tanto necesario evaluar este primer objetivo específico en diferentes momentos del curso para detectar si hemos conseguido una mejora en este aspecto. Se plantea una evaluación inicial mediante un test, seguida de una evaluación a mitad de curso y finalizando con una evaluación al final del mismo. Al ser este tipo de evaluación cuantitativa, podemos tener una calificación concreta con la nota del test para así poder comparar si ha habido mejora respecto al test anterior. Es importante que las tres pruebas planteadas tengan preguntas diferentes, con el fin de evitar la contestación por conocimiento de respuesta correcta. La siguiente tabla resumen establece la estrategia de evaluación de este primer objetivo específico.

Tabla 10. *Estrategia de evaluación del primer objetivo específico*

<b>Momento de evaluación</b>	<b>Tipo de prueba</b>	<b>Criterio de superación</b>
Inicio de curso	Test de evaluación de razonamiento lógico tipo uno	Esta primera prueba no evalúa, nos plantea un punto de partida
Mitad de curso	Test de evaluación de razonamiento lógico tipo dos	Se supondrá superada si obtenemos una calificación mejorando al test anterior
Final de curso	Test de evaluación de razonamiento lógico tipo tres	Se supondrá superada si obtenemos una calificación mejorando al test anterior

Por cada alumno, consideraremos una superación total del objetivo (100%) si las pruebas dos y tres mejoran los resultados de las pruebas anteriores. La superación será parcial (50%) si solo una de las pruebas mejora los resultados y se considerará no superado (0%) si los resultados no mejoran o

empeoran (con la excepción que pueden no mejorar si la puntuación es máxima, con lo que se considera superado). La puntuación total será la suma de los porcentaje de superación de los alumnos dividido por el número de alumnos. Una puntuación mayor del 50% se considerará objetivo superado. Ejemplificamos esta rúbrica de manera simplificada para cuatro alumnos en la siguiente tabla.

Tabla 11. *Ejemplo de evaluación del primer objetivo específico*

<b>Alumno</b>	<b>Nota prueba 1</b>	<b>Nota prueba 2</b>	<b>Nota prueba 3</b>	<b>Porcentaje de consecución</b>
Uno	3/10	4/10	5/10	100%
Dos	3/10	3/10	4/10	50%
Tres	4/10	3/10	3/10	0%
Cuatro	7/10	8/10	10/10	100%

Los alumnos uno y cuatro han mejorado siempre la nota con respecto a la prueba anterior, por lo que se considera que han superado totalmente el objetivo específico. Tres no mejora en ningún caso, incluso empeora, por lo que no se considera superado. Dos lo supera solo en la última prueba, con lo que se considera parcialmente superado. La evaluación global del objetivo sería  $(100+50+0+100)/4 = 62,5\%$ , es decir, se consideraría superado.

#### 4.6.2. *Evaluación del segundo objetivo específico*

En este objetivo específico pretendemos mejorar la motivación del alumno con el sistema de gamificación desarrollado en apartados anteriores. Se va a trabajar en base a la suposición de que un alumno motivado tendrá una asistencia a clase mayor así como una gran participación en las tareas de la asignatura. Se trabajará la consecución del objetivo en base a un registro tanto de asistencia como de trabajo de las tareas y plantearemos unos criterios de consecución similares a los del primer objetivo específico.

Tabla 12. *Estrategia de evaluación del segundo objetivo específico*

Momento de evaluación	Tipo de registro	Criterio de superación
Final de la primera evaluación	<ul style="list-style-type: none"> <li>• Porcentaje de asistencia</li> <li>• Porcentaje de tareas finalizadas</li> </ul>	Este primer registro no evalúa, nos plantea un punto de partida
Final de la segunda evaluación	<ul style="list-style-type: none"> <li>• Porcentaje de asistencia</li> <li>• Porcentaje de tareas finalizadas</li> </ul>	Se supone superado si mejoramos alguno de los porcentajes con respecto a la evaluación anterior
Final de la tercera evaluación	<ul style="list-style-type: none"> <li>• Porcentaje de asistencia</li> <li>• Porcentaje de tareas finalizadas</li> </ul>	Se supone superado si mejoramos alguno de los porcentajes con respecto a la evaluación anterior

Como en el caso anterior, se clarifica el modo de evaluar ejemplificando en un caso concreto. Observemos de nuevo una posible evaluación de este objetivo para cuatro alumnos.

Tabla 13. *Ejemplo de evaluación del segundo objetivo específico*

Alumno	Registro evaluación 1	Registro evaluación 2	Registro evaluación 3	Porcentaje de consecución
Uno	<ul style="list-style-type: none"> <li>• 80%</li> <li>• 70%</li> </ul>	<ul style="list-style-type: none"> <li>• 70%</li> <li>• 70%</li> </ul>	<ul style="list-style-type: none"> <li>• 80%</li> <li>• 70%</li> </ul>	50%

Dos	<ul style="list-style-type: none"> <li>• 80%</li> <li>• 70%</li> </ul>	<ul style="list-style-type: none"> <li>• 80%</li> <li>• 75%</li> </ul>	<ul style="list-style-type: none"> <li>• 90%</li> <li>• 90%</li> </ul>	100%
Tres	<ul style="list-style-type: none"> <li>• 60%</li> <li>• 20%</li> </ul>	<ul style="list-style-type: none"> <li>• 60%</li> <li>• 20%</li> </ul>	<ul style="list-style-type: none"> <li>• 60%</li> <li>• 40%</li> </ul>	50%
Cuatro	<ul style="list-style-type: none"> <li>• 40%</li> <li>• 50%</li> </ul>	<ul style="list-style-type: none"> <li>• 40%</li> <li>• 30%</li> </ul>	<ul style="list-style-type: none"> <li>• 40%</li> <li>• 30%</li> </ul>	0%

Se observa que el alumno uno no mejora ningún registro en la segunda evaluación, pero si mejora levemente su asistencia en la última. Por tanto se considera un que ha conseguido parcialmente el objetivo. En el caso del alumno dos, mejora el porcentaje de tareas realizadas en la segunda evaluación, pero en la tercera mejora tanto en tareas como en asistencia, con lo que el objetivo se da por superado totalmente. El alumno tres solo mejora en tareas en la última evaluación, con lo que también se consigue parcialmente el objetivo. Por último, el alumno cuatro no mejora ningún registro (incluso empeora en tareas en la segunda evaluación) con lo que el objetivo no se da por superado. En este caso tendríamos una consecución del objetivo de  $(50+100+50+0)/4 = 50\%$ , con lo que lo damos por superado por la mínima.

#### 4.6.3. Evaluación del tercer objetivo específico

En este tercer objetivo hemos planteado un instrumento de evaluación cualitativo, y por tanto de diferente tipología que el resto. No nos basamos en datos concretos si no en la percepción que tiene el profesor sobre la consecución o no del objetivo. Por tanto, se va a simplificar la evaluación de este punto con respecto a los anteriores, basándonos únicamente en entrevistas personalizadas con cada alumno al inicio y al fin del curso donde establezcamos una serie de preguntas abiertas y dirigidas a conocer las impresiones que tiene el estudiante sobre el proceso de evaluación tradicional en contraposición al planteado en el proyecto. Podemos observar en la siguiente tabla la estrategia de evaluación de este objetivo.

Tabla 14. Estrategia de evaluación del tercer objetivo específico

<b>Momento de evaluación</b>	<b>Tipo de prueba</b>	<b>Criterio de superación</b>
Inicio de curso	Entrevista dirigida a conocer los sentimientos del alumno de cara a la realización de una evaluación tradicional	Esta primera prueba no evalúa, nos plantea un punto de partida
Final de curso	Entrevista dirigida a conocer las impresiones del alumno sobre las microevaluaciones y su percepción sobre las mismas	Se considera superado si el alumno muestra una preferencia por el sistema de microevaluaciones planteado en el proyecto respecto al tradicional

Con el fin de establecer una correspondencia entre este método de evaluación y la asignación de un porcentaje de superación para poder ser evaluado dentro de la rúbrica final, se va a simplificar con un único criterio: se considera que se ha cumplido el objetivo (100%) para un alumno si el profesor detecta en la entrevista de final de curso que el alumno prefiere el método de microevaluaciones respecto a los exámenes tradicionales. No se considera superado (0%) si el alumno no muestra esta preferencia o muestre indiferencia por uno u otro. Planteamos una ejemplificación similar a las anteriores para cerrar este apartado.

Tabla 15. *Ejemplo de evaluación del tercer objetivo específico*

<b>Alumno</b>	<b>Percepción en la entrevista final de curso</b>	<b>Porcentaje de consecución</b>

Uno	El alumno prefiera el método planteado en el proyecto	100%
Dos	El alumno ve grandes ventajas y se siente más cómodo con el método planteado en el proyecto	100%
Tres	Al alumno le da igual un método u otro, se siente cómodo con ambos	0%
Cuatro	El alumno prefiere el método tradicional	0%

En este ejemplo tenemos un cumplimiento parcial del tercer objetivo al establecerse en un 50% =  $(100+100+0+0)/4$ .



## 5. EVALUACIÓN

Finalmente nos encontramos con la necesidad de evaluar nuestro objetivo global, y con este nuestro proyecto al completo. El método que se ha escogido para la realización de esta evaluación corresponde con el desarrollo de rúbricas para cada objetivo específico, definiendo criterios que tienen una correspondencia con alguna de las cualidades de las actividades que desarrollan dichos objetivos. Cada criterio tendrá asociado un peso definido como porcentaje en el total del cómputo a lo largo de los tres objetivos específicos del proyecto. Se indicará este porcentaje entre paréntesis dentro de la columna donde se defina cada uno de los criterios.

### 5.1. Rúbricas de evaluación

Nos disponemos a definir matrices de evaluación o rúbricas para cada objetivo específico, persiguiendo su evaluación por separado. Comenzaremos por el objetivo específico uno, relativo a la mejora del razonamiento lógico y con dos elementos que participan en su consecución: tareas y microaprendizajes.

Tabla 16. *Rúbrica para el OE1*

<b>Criterio</b>	<b>Excelente</b>	<b>Bien</b>	<b>Regular</b>	<b>Mal</b>
Las tareas están bien diseñadas (20%)	Las tareas tienen originales y variados ejemplos y no caen en las tareas tipo habituales	Las tareas son similares a las que encontramos en otra literatura sobre la enseñanza de esta materia	Las tareas son pobres y no consiguen motivar al alumnado	Las tareas son confusas y erróneas, provocando la desmotivación total del alumnado
Elaboración de los microaprendizajes (20%)	Están muy bien diseñados y bien editados y producidos. La edición tiene un	Están bien diseñados y producidos. No tenemos opening ni ending, pero se	La grabación es muy mejorable. En ocasiones cuesta ver el código y no	Grabación deficiente. La mayoría de vídeos son inaudibles además de que es

	opening y un ending.	observa bien el contenido.	se escucha con nitidez.	imposible seguir el hilo de la lección.
--	----------------------	----------------------------	-------------------------	---

Continuamos con el objetivo específico dos, el cual persigue aumentar la motivación a través del sistema de gamificación definido.

Tabla 17. Rúbrica para el OE2

<b>Criterio</b>	<b>Excelente</b>	<b>Bien</b>	<b>Regular</b>	<b>Mal</b>
Correspondencia de cada nivel con la tarea que evalúa (15%)	La correspondencia es muy clara y se identifica sin esfuerzo.	La correspondencia es clara aunque en ocasiones pueda surgir alguna duda puntual a que nivel evalúa una tarea en concreto.	La correspondencia es confusa y en ocasiones no está claro si una tarea evalúa a un nivel u otro.	La correspondencia es deficiente y en casi la totalidad de casos es imposible saber que estamos evaluando.
Diseño de la gráfica asociada a los niveles (10%)	La gráfica es original y ayuda a que de un simple vistazo tengamos claro que puntuación tiene un alumno concreto.	La gráfica es clásica pero deja claro que puntuación tiene un alumno al observarla.	La gráfica elegida es confusa o compleja y requiere prestar atención al familiarizarse con ella para poder saber que puntuación tiene un alumno.	El diseño de la gráfica es muy malo. Requiere una explicación del creador para entender como poder extraer de la misma la puntuación.

Por último evaluaremos el objetivo específico tres. Aquí nos centramos en el diseño de las microevaluaciones, las cuales están enfocadas a aumentar la predisposición del alumno hacia la prueba evaluativa.

Tabla 18. Rúbrica para el OE3

<b>Criterio</b>	<b>Excelente</b>	<b>Bien</b>	<b>Regular</b>	<b>Mal</b>
Diseño del enunciado de las microevaluaciones (20%)	El diseño es original. Los enunciados se salen de lo común y son percibidos con positividad por el alumnado.	El diseño es habitual pero bien planteado. Los enunciados son los típicos pero consiguen el objetivo de evaluar los contenidos a los que están destinados.	Diseño mejorable. Crea confusión en el alumno que en ocasiones necesita que se le aclare alguno de los apartados.	Diseño deficiente. Los alumnos no consiguen ver con claridad lo que requiere el enunciado con lo que parte del tiempo destinado a la evaluación lo pierden en aclarar lo que tienen que hacer.
Extensión de las microevaluaciones (15%)	La extensión se ajusta perfectamente a la prueba en concreto, habiendo una correspondencia a las tareas realizadas en clase.	La extensión se ajusta más o menos al tiempo destinado. En ocasiones puede que algunos ejercicios se alarguen o queden cortos.	La extensión no se ajusta siempre a los contenidos. Tenemos bastantes casos en los que el ejercicio se extiende más de lo necesario y no se ajusta el tiempo asignado a la prueba.	La extensión no coincide con las tareas de clase correspondientes a la prueba en concreto, alargándose o quedándose cortas. Las evaluaciones resultantes no son por tanto clarificadoras.



## 6. REFLEXIÓN Y VALORACIÓN FINAL

En esta reflexión final vamos a comenzar repasando las características destacables del proyecto. Para comenzar, el objetivo general que se persigue se considera de vital importancia, puesto que se trata de ayudar a la formación del alumno en lo que es sin duda la piedra angular de una profesión de futuro: la programación para un desarrollador informático. Para la consecución de tal meta, este trabajo se ha basado en dos pilares fundamentales de reconocida eficiencia como son el desarrollo del pensamiento lógico y la mejora de motivación. Es también importante destacar como virtud del proyecto que el desarrollo de pensamiento lógico se considera una aptitud transversal que puede ayudar al alumno en otras áreas diferentes a la programación.

Otro aspecto importante del proyecto es su facilidad de implementación. Como hemos visto en el apartado de recursos, no estamos hablando de una propuesta especialmente tediosa o difícil de llevar a cabo. Los recursos humanos y técnicos necesarios están totalmente cubiertos en cualquier aula de impartición de programación y esto supone un valor añadido. Su puesta en práctica por tanto puede ser inmediata, siempre y cuando se tenga en cuenta la limitación que supone que la temporalización del proyecto requiere una aplicación desde el inicio al final del curso, cuando puede haber otras propuestas que abarquen el problema en una ventana temporal más reducida.

Se podría considerar otra característica destacable la gran variedad de perfiles que pueden beneficiarse de obtener habilidades de programación a través de esta vía. Esta cualidad se consigue a través del diseño flexible del proyecto. Si bien en el apartado metodológico se ha planteado una propuesta concreta a través de un curso presencial tradicional, podemos adaptar fácilmente este diseño para módulos online u otras tipologías de carácter más flexible. La presencialidad del profesor no se considera una necesidad obligatoria en este caso.

En este apartado volvemos a poner de manifiesto el tercer pilar del proyecto y en el que reside su principal carácter innovador: las microevaluaciones. Se considera esta tipología evaluativa como una novedad en cuanto a diseño y aplicación dentro de la rama de la programación informática,

siendo su eficiencia solamente reseñable si, en una posible puesta en práctica del proyecto, se supera la evaluación propuesta en el quinto apartado de este trabajo, la evaluación.

Las vías futuras que pueden surgir a partir de este proyecto son variadas gracias al enfoque tecnológico planteado, el cual le confiere a este trabajo un gran margen de evolución. Recordemos que la propuesta de gamificación y el alojamiento de los microaprendizajes han sido gestionados de manera independiente. Una posible versión mejorada del trabajo podría sustentarse en el desarrollo de una plataforma integradora de estos dos componentes. Esta plataforma podría tener varios enfoques, entre los que podríamos destacar el desarrollo una aplicación móvil para que el alumno pueda visualizar los microaprendizajes en cualquier momento y lugar con mayor facilidad así como revisar su puntuación y tareas pendientes. Por otro lado podríamos tener una plataforma web, a modo de LMS, que también integrara todos los componentes.

Para ir finalizando con esta reflexión final, se puede citar una última característica importante y es la posible adaptabilidad del diseño del proyecto a otros ámbitos que requieran una capacidad de razonamiento similar. Por ejemplo en las matemáticas. Esto podría conseguirse adaptando únicamente tareas y microaprendizajes, puesto que el sistema de gamificación e incluso las microevaluaciones podrían ser totalmente compatibles conceptualmente con una nueva área de estudio. También podrían ser objeto de aplicación, y más fácilmente adaptables, otras áreas de estudio de la informática como sistemas informáticos, algoritmia o estructuras de datos avanzada.

Por último, destaco el valor que me ha aportado la realización de este proyecto a nivel de desempeño docente. Sin duda, la investigación y la innovación educativa son aspectos cruciales, y algunas veces olvidados, en el desarrollo de un profesor. No solo en la mejora del proceso de enseñanza y aprendizaje, si no también en el ámbito puramente interno del profesional de la enseñanza que, a través de la aplicación de trabajos de esta naturaleza, consigue impulsar su motivación interna como profesional además de engrandecer su figura con el resto de la comunidad educativa. Tras la realización

de este proyecto, tengo una visión clara de la importancia del desarrollo y aplicación de ideas como la que se ha elaborado en este trabajo.



## 7. REFERENCIAS

- Albrecht, K. (2009). *Brain power: Learn to improve your thinking skills*. Simon and Schuster.
- Bausela Herreras, E. (2005). Ansiedad ante los exámenes: evaluación e intervención psicopedagógica. *Educere*, 9(31), 553–557.
- Bennedsen, J., & Caspersen, M. E. (2007). Failure rates in introductory programming. *AcM SIGcSE Bulletin*, 39(2), 32–36.
- Bennedsen, J., & Caspersen, M. E. (2019). Failure rates in introductory programming: 12 years later. *ACM Inroads*, 10(2), 30–36.
- Berumen, S. A., & Ibarra, K. A. (2008). *Evolución y desarrollo de las TIC en la economía del conocimiento*. Ecobook.
- Buckley, P., & Doyle, E. (2016). Gamification and student motivation. *Interactive Learning Environments*, 24(6), 1162–1175.
- Greaney, K. (2015). Skills-based learning within a constructivist curriculum: The case for rote learning to automaticity the multiplication facts for maths and alphabetic skills for reading. *Journal of Education*, 14(2), 1–8.
- Jenkins, T. (2001). The motivation of students of programming. *Proceedings of the 6th Annual Conference on Innovation and Technology in Computer Science Education*, 53–56.
- Leong, K., Sung, A., Au, D., & Blanchard, C. (2021). A review of the trend of microlearning. *Journal of Work-Applied Management*, 13(1), 88–102.
- Manyika, J., Lund, S., Chui, M., Bughin, J., Woetzel, J., Batra, P., Ko, R., & Sanghvi, S. (2017). Jobs lost, jobs gained: Workforce transitions in a time of automation. *McKinsey Global Institute*, 150.
- Marco-Galindo, M.-J., Minguillón, J., García-Solórzano, D., & Sancho-Vinuesa, T. (2021). *¿ Quién tropieza dos veces con la misma piedra en una asignatura inicial de programación?*

- Mayer, R. E. (1981). The psychology of how novices learn computer programming. *ACM Computing Surveys (CSUR)*, 13(1), 121–141.
- Milková, E., & Hulková, A. (2013). Algorithmic and logical thinking development: Base of programming skills. *WSEAS Transactions on Computers*, 12(2), 41–51.
- Nah, F. F.-H., Zeng, Q., Telaprolu, V. R., Ayyappa, A. P., & Eschenbrenner, B. (2014). Gamification of education: a review of literature. *HCI in Business: First International Conference, HCIB 2014, Held as Part of HCI International 2014, Heraklion, Crete, Greece, June 22-27, 2014. Proceedings 1*, 401–409.
- Reyes, C. B. (2018). Un sistema de Wooc para la actualización docente. *Revista de La Facultad de Ciencias Económicas*, 20, 75–87.
- Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education*, 13(2), 137–172.
- Teixes, F. (2014). Gamificación: fundamentos y aplicaciones. *Gamificación*, 1–136.
- Thomas, D., & Hunt, A. (2019). *The pragmatic programmer*. Addison-Wesley Professional.
- Wei, X., Lin, L., Meng, N., Tan, W., & Kong, S.-C. (2021). The effectiveness of partial pair programming on elementary school students' computational thinking skills and self-efficacy. *Computers & Education*, 160, 104023.
- Winslow, L. E. (1996). Programming pedagogy—a psychological overview. *ACM Sigcse Bulletin*, 28(3), 17–22.
- Wuthrich, V. M., Jagiello, T., & Azzi, V. (2020). Academic stress in the final years of school: A systematic literature review. *Child Psychiatry & Human Development*, 51, 986–1015.
- Zainal, N. F. A., Shahrani, S., Yatim, N. F. M., Abd Rahman, R., Rahmat, M., & Latih, R. (2012). Students' perception and motivation towards programming. *Procedia-Social and Behavioral Sciences*, 59, 277–286.

## 8. ANEXOS

### 8.1. Anexo I. Ejemplo de microevaluación

- 1) Ejercicio 1 (5 puntos de nivel 3). Realiza un programa que implemente una clase Estudiante con los siguientes atributos: curso y edad. Implementa también un método estudiar() que no devuelva ningún tipo de dato pero que reciba un argumento materia para que el cuerpo de la función pueda sacar por pantalla el texto: "Estoy estudiando " + materia.
- 2) Ejercicio transversal (3 puntos de nivel 2). Demuestra que sabes formatear el texto a través de teclado.

### 8.2. Anexo II. Sistema de gamificación

