



Universidad  
Internacional  
de Andalucía

## TÍTULO

**DESARROLLO DE VIDEOJUEGOS Y FORTALECIMIENTO DE  
HABILIDADES Y HERRAMIENTAS CLAVE**  
UNA ESTRATEGIA INNOVADORA EN LA ENSEÑANZA DEL MÓDULO  
DE PROGRAMACIÓN EN EL CICLO SUPERIOR DE DESARROLLO DE  
APLICACIONES MULTIPLATAFORMA

## AUTOR

**José Alberto Gandullo Ávila**

	<b>Esta edición electrónica ha sido realizada en 2024</b>
Tutora	Dra. D <sup>a</sup> . María Belén Prados Suárez
Institución	Universidad Internacional de Andalucía <i>Máster Universitario en Profesorado de Enseñanza Secundaria</i>
Curso	<i>Obligatoria, Bachillerato, Formación Profesional y Enseñanzas de Idiomas</i> <i>(2022/23)</i>
©	María Álvarez Perales
©	De esta edición: Universidad Internacional de Andalucía
Fecha documento	2023



Universidad  
Internacional  
de Andalucía



**Atribución-NoComercial-SinDerivadas  
4.0 Internacional (CC BY-NC-ND 4.0)**

Para más información:

<https://creativecommons.org/licenses/by-nc-nd/4.0/deed.es>

<https://creativecommons.org/licenses/by-nc-nd/4.0/deed.en>



MAES - UNIA

CURSO 2022/23

## TRABAJO FIN DE MÁSTER

**Desarrollo de videojuegos y fortalecimiento de habilidades y herramientas clave: Una estrategia innovadora en la enseñanza del Módulo de Programación en el ciclo superior de Desarrollo de Aplicaciones Multiplataforma.**

Alumno: JOSÉ ALBERTO GANDULLO ÁVILA  
Tutora: MARÍA BELÉN PRADOS SUÁREZ

Junio 2023

# Índice

<b>Resumen.....</b>	<b>3</b>
<b>1. Introducción y justificación.....</b>	<b>4</b>
1.1 Introducción.....	4
1.2. Justificación.....	5
<b>2. Estado del arte.....</b>	<b>10</b>
<b>3. Objetivos.....</b>	<b>22</b>
<b>4. Propuesta pedagógica.....</b>	<b>24</b>
4.1. Contexto: Marco normativo.....	24
4.1.1 Ámbito de aplicación.....	24
4.1.2 Competencias.....	24
4.1.3 Contenidos básicos.....	24
4.1.4 Resultados de Aprendizaje.....	25
4.2 Descripción.....	28
4.3 Cronograma e inserción en la PD/dinámica del centro.....	29
4.4. Actividades y su descripción.....	31
4.5. Recursos y materiales.....	44
4.6. Atención a la diversidad.....	47
<b>5. Evaluación:.....</b>	<b>49</b>
5.1. De la asignatura.....	49
5.2. De la propuesta de innovación.....	50
<b>7. Conclusiones y líneas futuras.....</b>	<b>52</b>
<b>8.Referencias.....</b>	<b>54</b>
<b>Anexo I: Información del Módulo de Programación de DAM extraída del BOJA.....</b>	<b>61</b>
I.I Competencias.....	61
I.II Contenidos básicos.....	62
I.III Resultados de Aprendizaje.....	65

# Resumen

Esta propuesta de innovación educativa tiene como objetivo integrar el desarrollo de videojuegos como una estrategia adicional de enseñanza en el módulo de programación del ciclo superior de Desarrollo de Aplicaciones Multiplataforma (DAM). Además a través de una serie de actividades prácticas, se busca fomentar el aprendizaje activo, promover el trabajo en equipo, potenciar habilidades de gestión del tiempo y familiarizar a los estudiantes con herramientas y metodologías relevantes para el sector. Con esta propuesta, se busca mejorar la experiencia de aprendizaje de los estudiantes, preparándolos de manera integral para su futura carrera en el campo del desarrollo de aplicaciones multiplataforma.

# 1. Introducción y justificación.

A continuación se hará una breve introducción seguida de la justificación del trabajo desarrollado en este documento.

## 1.1 Introducción

El desarrollo de aplicaciones multiplataforma es un campo en constante evolución que requiere de habilidades técnicas sólidas, así como competencias y herramientas clave para el sector. En el ciclo superior de Desarrollo de Aplicaciones Multiplataforma (DAM), es fundamental proporcionar a los estudiantes una formación completa que los prepare para los desafíos del mundo laboral. En este sentido, el uso de estrategias innovadoras de enseñanza se vuelve esencial para garantizar un aprendizaje efectivo y motivador.

La presente propuesta de innovación educativa tiene como objetivo principal introducir el desarrollo de videojuegos como una estrategia adicional de enseñanza en el módulo de programación. Los videojuegos, además de ser una herramienta atractiva y motivadora para los estudiantes, permiten abordar de manera práctica los conceptos y fundamentos de la programación, al mismo tiempo que fortalecen habilidades transversales, como el trabajo en equipo, la gestión del tiempo y el uso de herramientas colaborativas, de este modo los alumnos irán desarrollando partes de un videojuego sencillo conforme van aprendiendo conceptos de programación.

A lo largo de esta propuesta, se presentarán una serie de actividades diseñadas para promover el aprendizaje activo y práctico de los estudiantes. Desde la introducción a las soft skills de trabajo en equipo y gestión del tiempo, hasta el desarrollo de scripts en Unity y la participación en una Game Jam, se busca brindar a los estudiantes una experiencia enriquecedora y realista del proceso de desarrollo de videojuegos.

Además, se fomentará la utilización de herramientas y metodologías relevantes para el sector, como el control de versiones con Git y la metodología Scrum. Estos elementos son esenciales para la colaboración efectiva en equipos de desarrollo y para la gestión eficiente de proyectos.

## 1.2. Justificación

Cada vez hay más estudios que indican que el uso de ordenadores en clase puede reducir el rendimiento académico de los alumnos, sobre todo si no hay detrás un buen plan de uso que saque provecho de ellos.(BARREDO, 2017)

Por lo que pude escuchar durante las prácticas y de mis compañeros del MAES las distracciones y en general la falta de atención de los alumnos en el aula son sin duda unos de los problemas que más enfatizan los profesores actualmente a la hora de enfrentarse a las clases.

En concreto las aulas dotadas con ordenadores para los alumnos, estos están expuestos en mucho mayor medida a distracciones que en un aula sin ellos. Los ciclos de formación profesional de la familia de Informática suelen estar dotados con un ordenador por puesto así que suelen sufrir de este foco de distracciones.

De acuerdo al Instituto Nacional de Tecnologías Educativas y Formación del Profesorado de España (INTEF, s. f.), el empleo de tecnología en la enseñanza fomenta un aprendizaje activo, simplifica la abstracción, permite la adaptación al ritmo de aprendizaje de cada estudiante y promueve la independencia en el proceso de aprendizaje.

Un documento elaborado por la Universidad de Castilla La Mancha (España) señala que los profesores pueden valerse de la computadora de distintas maneras: como asistente de la enseñanza, pero también como herramienta que estimula la creatividad y los procesos de pensamiento y como elemento motivador, pues puede hacer más ameno el proceso de aprendizaje.

No obstante, es importante señalar que, sin guía adecuada, las computadoras pueden tener el efecto opuesto: en lugar de servir como herramientas para mejorar la educación, se pueden convertir en un factor de distracción para los estudiantes.(GrupoRPP, 2016)

Otro problema más generalizado e igual o más distractor es el hecho de que todos los alumnos llevan su teléfono móvil al aula, y ya hay estudios que indican que estos reducen la concentración de los alumnos incluso apagados.(PECO, 2019)

Según este otro estudio (Paredes et al., 2018) “En el aula el 86.4% de los estudiantes hacen uso del celular sin fines académicos; el 77.9% reconoce la dependencia al uso del celular, el 49.8% se distrae a veces por el sonido del celular.”

Al igual que con los ordenadores los dispositivos móviles pueden también usarse en beneficio del proceso de aprendizaje pero siempre con planificación y supervisión.(Juárez et al., 2015)

A modo de resumen podríamos citar como las principales ventajas e inconvenientes del uso de la tecnología en el aula las siguientes:

### Figura 1

Principales ventajas e inconvenientes del uso de la tecnología en el aula (Roberto, 2014)

↑ A Favor	↓ En Contra
<ul style="list-style-type: none"><li>• Acceso a más información, actividades, vídeos, etc</li><li>• Aprender se hace más divertido y más ameno. Alumnos más motivados.</li><li>• Facilidad para un aprender de forma personalizada. Cada alumno aprende a su ritmo.</li><li>• Favorece la autonomía personal y el desarrollo de la creatividad del alumno.</li></ul>	<ul style="list-style-type: none"><li>• Falta de infraestructura técnica, tanto eléctrica como de conexión a Internet.</li><li>• Distracción para los alumnos. Necesidad de supervisión por parte del profesor.</li><li>• Dificultades técnicas. Un fallo en la tablet o el portátil impiden seguir la clase o hacer los deberes.</li><li>• Demasiadas horas mirando una pantalla puede provocar fatiga visual.</li></ul>

El otro problema que comentábamos al inicio era la falta de atención de los alumnos y a su vez solucionar este problema también podría ser una solución, al menos en parte, para disminuir las distracciones.

En este otro estudio (Maquilón Sánchez & Hernández Pina, 2011) una de las conclusiones es que debido a inadecuadas motivaciones, seis de cada diez estudiantes de los ciclos formativos emplean predominantemente un enfoque de aprendizaje superficial, lo cual repercute muy negativamente en sus resultados académicos.

Si no se aborda el problema de la falta de atención, esto puede tener un impacto negativo en el aprendizaje de los estudiantes y en última instancia en el índice de inserción laboral ya que podrían no alcanzar el nivel requerido por las empresas.



La falta de atención e interés durante las clases de programación puede hacer que los estudiantes abandonen la asignatura o tengan un bajo rendimiento académico. Esto puede llevar a un aumento de las tasas de abandono escolar, lo que puede afectar negativamente la trayectoria educativa y laboral de los estudiantes.

Aunque en los últimos años la tasa de abandono temprano se ha ido reduciendo España sigue estando por encima de la media Europea en cuanto a abandono temprano de la educación-formación de personas de 18-24 años.(INE. Instituto Nacional de Estadística, s. f.)

En cuanto a los datos en Andalucía, se sitúa con 21,8% como la tercera comunidad con la tasa de abandono escolar más alta según EPA(Press, 2021)

En mi opinión si los estudiantes no están interesados en la materia, pueden no esforzarse lo suficiente y obtener calificaciones más bajas o incluso abandonar los estudios.

Según este estudio sobre el abandono en FP (Cerdà-Navarro et al., 2020), los resultados sugieren que las perspectivas profesionales son muy importantes para vincular al estudiante con la profesión estudiada, pero es necesario que éstas se complementen con implicación en las tareas escolares y buenas relaciones con el profesorado para reducir el riesgo de abandono.

En el proceso de aprendizaje, la atención es un factor fundamental y especialmente importante como primer paso antes de iniciar cualquier enseñanza. Si hay una falta de atención, dificultades para mantenerla o enfocarla, pueden surgir otros problemas que a menudo se asocian con un bajo rendimiento, problemas de comprensión, un desarrollo inadecuado de las habilidades esperadas y conductas que dificultan el progreso del grupo en su conjunto. (Gómez et al., s. f.)

En resumen, la falta de atención e interés tanto en la enseñanza de programación como en otras puede tener un impacto negativo en las tasas de abandono y fracaso escolar, las notas medias de la asignatura y el índice de inserción laboral de los estudiantes. Por lo tanto, es importante abordar este problema para mejorar la calidad de la educación y preparar adecuadamente a los estudiantes para el futuro.

En mi propuesta pretendo que los estudiantes tengan un aprendizaje útil trabajando toda una gama de habilidades tanto blandas (soft skills) como duras (hard or tech skills) relacionadas con el desarrollo de software que son muy solicitadas en el mercado laboral, en este sentido he elegido enseñar a programar en el marco del desarrollo de videojuegos abordándolo de modo que se trabajen habilidades duras y blandas, además espero que la enseñanza sea más atractiva para los alumnos ya que suelen estar bastante interesados en los videojuegos.

Según la Asociación Española de Videojuegos (AEVI), hace años que la industria de los videojuegos ha superado con creces la facturación del sector audiovisual y musical combinados. Aunque se sumaran la facturación del cine, con 587,5 millones anuales, y la de la música grabada, con 237,2 millones, no alcanzarían a los ingresos generados por los videojuegos. De esta forma, el sector de los videojuegos se establece como el principal referente en cuanto a ocio y entretenimiento en nuestro país.

Además de estos buenos resultados, hay que sumar el aumento del empleo en el sector en España, que ha sido del 16,5 por ciento, y se esperaba que hasta el año 2121 se mantuviera un ritmo positivo del 18,2 por ciento. También hay buenas perspectivas gracias al crecimiento de la inversión extranjera, que experimentó un aumento del 4 por ciento el pasado año, al incremento del 10 por ciento de los ingresos procedentes de los mercados internacionales y al aumento de los estudios de desarrollo internacionales en nuestro país. (Blanco, 2020)

Con relación a los videojuegos y la educación, en el Libro Blanco de la Prospectiva TIC: Proyecto 2020 se sugiere que, en la actualidad, es necesario “acortar la ‘distancia’ entre la industria y el sistema académico, la cual actualmente es muy grande . . . , lo cual redundará en potenciar las capacidades en ambos”

Según los autores (Conde & García, s. f.; Daza Pérez & Pardo, 2016; Galli et al., 2016; Mena, s. f.; Moreno Cano, 2019; Narváez-Díaz & López-Martínez, 2021; Palm, 2010; Wilce & Talledo, s. f.; Zabala et al., 2013) algunas de las expectativas que cabe esperar son:

- Al enseñar programación mediante el desarrollo de videojuegos, se espera que los estudiantes estén más atentos e interesados en la materia, lo que puede aumentar su compromiso y dedicación a la asignatura.
- Al desarrollar videojuegos, los estudiantes tendrán la oportunidad de aprender habilidades técnicas y prácticas que son útiles en el campo de la tecnología y las aplicaciones. Esto puede prepararlos mejor para el mundo laboral y aumentar sus oportunidades de empleo.
- Al aprender y aplicar habilidades técnicas y prácticas obteniendo resultados visibles y funcionales los estudiantes pueden aumentar su autoconfianza y autoestima, lo que puede mejorar su rendimiento académico y laboral.
- Si los estudiantes están más atentos y comprometidos en la asignatura, es probable que mejoren su rendimiento académico, lo que puede llevar a un aumento en las notas medias de la asignatura y disminuir las tasas de abandono y fracaso escolar.

- Al adquirir habilidades técnicas y prácticas útiles en el mundo laboral, los estudiantes pueden estar mejor preparados para el futuro y tener más oportunidades de empleo. Por ejemplo, pueden aprender a desarrollar software de entretenimiento o educativo, y utilizar técnicas de programación que son comunes en la industria del software
- Además, el desarrollo de videojuegos requiere una combinación de habilidades técnicas y creativas, lo que puede ayudar a los estudiantes a desarrollar una mentalidad más innovadora y a ser más creativos en su enfoque para resolver problemas.

## 2. Estado del arte.

La presente investigación se enfoca en la enseñanza de la asignatura de programación en un ciclo superior de formación profesional, abordando no solo las habilidades técnicas requeridas para programar, sino también algunas de las habilidades blandas más demandadas en el mercado laboral para los desarrolladores de software. Para lograr esto, se utilizará un enfoque basado en proyectos que permitirá a los estudiantes aplicar lo aprendido en un contexto más real y concreto. Además, se utilizará el desarrollo de videojuegos como hilo conductor del proceso de enseñanza-aprendizaje, aprovechando la experiencia previa del investigador en este campo.

De esta manera los diferentes aspectos a tratar son los siguientes:

- Aprendizaje útil y relevante de cara al mercado laboral.
  - Hard Skills
  - Soft skills
- Metodología ABP, sus beneficios y uso
- Beneficios de los videojuegos en la educación

### **Aprendizaje útil y relevante de cara al mercado laboral.**

Uno de los pilares de este proyecto consiste en intentar hacer más atractivo el módulo de programación ofreciendo a los alumnos un aprendizaje más práctico y relevante de cara al mercado laboral.

En esta sección se intentarán describir las habilidades demandadas en el mercado laboral que se pueden enseñar en el aula para proporcionar este aprendizaje útil y relevante que haga que los estudiantes tengan una ventaja de empleabilidad, todo esto en el contexto de la programación ya que este es el módulo que abarca este trabajo.

Una habilidad se refiere a la capacidad de realizar una tarea con destreza y eficacia. En otras palabras, se trata de la capacidad de aplicar el conocimiento de manera efectiva y con facilidad durante la ejecución o desempeño de una tarea. La palabra "habilidad" tiene varios significados y sinónimos comunes, tales como capacidad, competencia, destreza, aptitud, talento, entre otros.

Según el Foro Económico Mundial entre las habilidades más demandadas en 2025 se encuentran tanto habilidades duras como el uso y diseño de tecnología y la programación como habilidades blandas como la resiliencia o la flexibilidad:

**Figura 2**

Las 10 mejores habilidades para 2025 (World Economic Forum, s. f.)



En general, existen dos categorías principales de habilidades: las habilidades blandas y las habilidades duras. En mi opinión tanto las soft skills (habilidades blandas) como las hard skills (habilidades duras) son importantes para cualquier trabajo, pero en el campo del desarrollo de software, ambas son esenciales para tener éxito.

Según (Birt, s. f.) mientras que las habilidades duras son necesarias para realizar tareas técnicas en un trabajo de manera exitosa, las habilidades blandas son necesarias para crear un ambiente de trabajo positivo y funcional. Por esta razón, los empleadores a menudo buscan individuos que

posean probadas habilidades duras y blandas. Algunos empleadores pueden preferir seleccionar candidatos que tengan un conjunto de habilidades blandas más fuerte que habilidades duras, ya que las habilidades blandas son a veces más difíciles de desarrollar.

Como síntesis de lo que dicen los artículos encontrados y mi propia experiencia sobre las habilidades duras y blandas es que dependiendo del puesto de trabajo al que se apunte existe un mínimo nivel de conocimiento sobre las tech skills que hay que tener para que un empleador empiece a considerar a un candidato para su contratación, superado ese mínimo nivel de conocimiento de las habilidades duras el empleador suele basar más su decisión última de contratación o no en las soft skills y no tanto en las tech skills ya que estas últimas se considera que pueden ser desarrolladas con más facilidad a posteriori.(Hidayati et al., 2020), (SEKHAR, 2019), (Sampaio et al., 2021)

Pasemos ahora a hablar concretamente de las denominadas habilidades duras o hard skills que en el caso de la programación se les suele llamar tech skills, éstas son competencias adquiridas a través del estudio o la experiencia que permiten realizar una labor o profesión. Las hard skills, reflejan el conocimiento técnico de un profesional y le ayudan a desempeñar con éxito un trabajo.

Dependiendo de la fuente se pueden encontrar diversas listas de estas tech skills para programadores pero haciendo síntesis y factor común de varios de ellos (Hidayati et al., 2020), (Thompson, 2020), (Kai, 2022), (UniversityOfDenver, 2020) podemos resumirlas en las siguientes:

- **Lenguajes de programación**

El código se escribe en lenguajes de programación como Java, C# o Python. No se espera que los programadores conozcan todos los lenguajes de programación existentes, pero sería beneficioso ser competente en al menos dos lenguajes para aumentar las oportunidades de empleo y carrera. La mayoría de los programadores generalmente aprenden los fundamentos de la programación en cualquier lenguaje importante, después deciden sobre un área de especialización y aprenden el lenguaje de programación que es más apropiado para ese campo.

- **Estructuras de datos y algoritmos**

Cada programador debe aprender estructuras de datos y algoritmos, ya que son la base para construir programas eficientes y resolver problemas de manera óptima. Los algoritmos son conjuntos de instrucciones para solucionar un problema, y las estructuras de datos contienen información que será utilizada por los algoritmos. El conocimiento de

algoritmos se aplica en situaciones como ordenamiento, búsqueda y optimización del código. Es esencial para mejorar la forma de construir programas y solucionar problemas.

- **Bases de datos**

Es imprescindible para los programadores saber cómo interactuar con las bases de datos, ya que la mayor parte de la programación se relaciona con la manipulación de datos. En muchas ofertas de trabajo, la experiencia en bases de datos es un requisito. Sin conocimientos de bases de datos, se pierden muchas oportunidades laborales. Aunque se sea principiante, es necesario tener conocimientos básicos de cómo crear, eliminar, actualizar e insertar registros. También es importante conocer la seguridad de las bases de datos.

- **Control de versiones**

Para ser un programador efectivo, se debe saber cómo usar un sistema de control de versiones. Esto ayuda a controlar y administrar los cambios en los programas. Si se comete un error, se puede volver a una versión anterior del código. Trabajar en un proyecto sin usar control de versiones es arriesgado, y puede ser muy frustrante volver al código original para arreglar errores. Es importante trabajar con versiones y mantener copias de seguridad.

- **Depuración y Pruebas**

Cada programador debe tener conocimientos básicos de depuración y pruebas de software. Esto ayuda a verificar que el código funciona correctamente. Los programadores suelen realizar pruebas unitarias para asegurarse de que el código funciona en diferentes situaciones. Aunque la mayoría de las empresas contratan a testers, es importante realizar pruebas antes de entregar el código para que sea probado por otros.

- **Entornos de desarrollo integrado (IDE)**

Las IDE (Integrated Development Environments) combinan una variedad de herramientas de desarrollo a través de una interfaz gráfica de usuario (GUI) única. Son como un banco de trabajo para los programadores, donde todas las herramientas que necesitan están disponibles y listas para usar, como un banco de trabajo con sierra, taladro, clavos y martillo si estuvieras planeando construir una casita para pájaros.

- **Scrum o algún otro marco de trabajo para desarrollo ágil**

Scrum es un marco de trabajo para el desarrollo ágil de software que se ha extendido a otras industrias. Se trata de un proceso en el que se aplican regularmente un conjunto de buenas prácticas para trabajar en equipo de manera colaborativa y obtener el mejor resultado posible en los proyectos. El uso de estas metodologías de trabajo suelen venir acompañadas del uso de software de gestión de proyectos.

La mayoría de estas habilidades duras ya están incluidas de por sí en el currículo del módulo de Programación a impartir (Lenguajes de programación, IDE, Bases de Datos) pero en este proyecto intentaré también incluir al menos una introducción al resto de estas habilidades (Control de versiones, Depuración y Pruebas y desarrollo ágil) ya que según la lista anterior son las que los empleadores quieren.

Ahora que ya tenemos más claras las habilidades duras más imprescindibles para los programadores vamos a hacer una revisión de qué son las habilidades blandas y por qué los empleadores les dan tanta importancia.

Se ha escrito mucho sobre las soft skills, sintetizando las definiciones de algunos autores se podría decir que las habilidades blandas son habilidades no técnicas que permiten a las personas manejar su propia conducta y relacionarse de manera positiva con los demás en un entorno laboral. Estas habilidades suelen estar relacionadas con la capacidad de comunicación y las relaciones interpersonales, y a menudo se desarrollan a través de la experiencia personal y la reflexión.

Existen muchísimas clasificaciones de Soft Skills según su autor, por ejemplo para (García-Bullé, 2019) las más importantes serían la inteligencia emocional, la creatividad, la percepción, la comunicación, el trabajo en equipo y la negociación

Según (SEKHAR, 2019) las principales soft skills serían:

- Comunicación verbal y escrita.
- Aprendizaje autodirigido
- Pensamiento creativo
- Resolución de problemas
- Resolución de conflictos.
- Gestión del tiempo
- Liderazgo
- Actitud positiva
- Sentido del humor
- Trabajo en equipo

Y (Alex, 2009) enumera en su libro hasta 60 soft skills entre las que se pueden encontrar la cortesía, honestidad, adaptabilidad, sentido común y muchas otras.

Para este trabajo me centraré en las habilidades blandas descritas por Henry y Sampaio (Henry, 2021), (Sampaio et al., 2021), ya que enumeran las que para el mercado laboral se describen como las más importantes en el contexto de la programación.



**Figura 3**

Soft Skills para programadores. Elaboración propia.



- **Comunicación clara.**

Es crucial que las empresas tengan la capacidad de comunicar mensajes transparentes y sinceros internamente. Para lograrlo, es esencial utilizar la comunicación asertiva como herramienta de diálogo en los equipos, lo que permite tomar decisiones y realizar tareas de manera eficiente. Esto implica priorizar mensajes breves y relevantes, transmitidos con sinceridad y respeto.

- **Trabajo en equipo.**

A menudo se cree que los desarrolladores son personas introvertidas que trabajan de manera aislada. Sin embargo, este estereotipo es erróneo. De hecho, los programadores necesitan aprender a trabajar con otros, ya sea con otros desarrolladores o con miembros de diferentes equipos dentro de la empresa. Es una habilidad fundamental que todos deberían practicar para lograr equipos de trabajo más colaborativos y productivos.

- **Adaptación al cambio.**

En un entorno tecnológico en constante evolución, la velocidad es crucial para adaptarse a las novedades del sector y a los cambios en el negocio. Los programadores junior deben aprender a manejar entornos en constante mutación, incluyendo cambios en proyectos, rotaciones de equipos, nuevos lenguajes de programación, objetivos cambiantes y dinámicas nuevas. La capacidad de convivir con la incertidumbre y ser flexible es esencial para evitar la frustración y tener éxito en este campo.

- **Gestión del tiempo.**

En un mundo de tecnología cambiante y trabajo remoto, es importante que los desarrolladores, incluso los junior, desarrollen habilidades de gestión del tiempo efectivas. Además de entrenar sus habilidades técnicas, es esencial que aprendan a planificar adecuadamente el tiempo disponible para cada proyecto, definir prioridades y urgencias, establecer plazos y llevar un registro de todo ello. Demostrar un buen criterio para organizar tareas puede convertir a los desarrolladores en colaboradores imprescindibles.

- **Responsabilidad.**

La responsabilidad es un rasgo fundamental que demuestra profesionalidad en cualquier ámbito laboral, incluyendo el de programación. En última instancia, la responsabilidad determina la confianza que puedes transmitir a tus líderes y otros colaboradores. Cumplir con las tareas asignadas, ser organizado y puntual son atributos que sientan las bases para establecer relaciones laborales duraderas y efectivas con los demás.

- **Resolución de problemas.**

En el mundo de la tecnología, el trabajo se centra en la resolución de problemas. Los desarrolladores deben desarrollar esta habilidad y adoptarla como perspectiva en cada proyecto en el que trabajen, ya sea para agregar valor a sus equipos o a la empresa en su conjunto. Resolver problemas implica una actitud proactiva, así como planificación y creatividad para abordar los desafíos que surjan en el camino.

- **Habilidades interpersonales.**

Los desarrolladores, como seres sociales que trabajan en equipo para solucionar problemas, pueden mejorar su desempeño en cualquier proyecto mediante el entrenamiento de habilidades blandas. La escucha activa, el respeto, la tolerancia y, sobre todo, la empatía son habilidades clave que pueden hacer que la experiencia de trabajo sea más agradable y exitosa. La empatía es especialmente importante, ya que implica la capacidad de entender las necesidades de los demás y responder adecuadamente a ellas, lo que puede marcar la diferencia en el éxito del proyecto.

De cara a la enseñanza del módulo de programación intentaré centrarme en trabajar el Trabajo en Equipo y la Gestión del Tiempo. El trabajo en equipo es algo con lo que prácticamente todos los alumnos se encontrarán al llegar a la empresa y además con el trabajo en equipo se trabajarán casi automáticamente otras soft skills como la comunicación y el Liderazgo. La gestión del tiempo les proporcionará el plus de productividad que tanto buscan las empresas.

Según Lencioni et al. (Lencioni, 2002), algunas de las dimensiones que tiene la habilidad blanda del trabajo en equipo son:

- Comunicación
- Roles
- Confianza
- Cooperación
- Liderazgo
- Resolución de Conflicto
- Flexibilidad

Según Allen, Covey y Morgenstern (Allen & Fallows, 2015; Covey, 2015; Morgenstern, 2001), algunas de las dimensiones de la Gestión del tiempo como habilidad blanda son:

- Creación de Tareas
- Priorización
- Planificación
- Delegación
- Eliminación de Distracciones
- Automatización

Las habilidades blandas son difíciles de aprender en de forma tradicional, algunos de los métodos son los siguientes:

**Figura 4**

Traducción de Metodologías para desarrollar soft skills (Caeiro-Rodríguez et al., 2021)

<b>Expositoria</b>	<b>Guiada</b>	<b>Activa</b>
Lectura	Discusión, Debate	Lluvia de ideas
Seminario	Taller	Juego de roles
Conferencia	Caso de estudio	Juego de negocios
Demostración	Proyecto de trabajo	Visitas, Jornadas
	Simulación	Prácticas en el exterior
	Tutoría	Coaching

En mi propuesta educativa he elegido la metodología de Aprendizaje Basado en Proyectos ya dentro de un ABP se pueden incluir algunas actividades como las de la figura anterior y creo reúne las características necesarias para poder trabajar tanto las habilidades duras y blandas que he comentado anteriormente. Además hay ejemplos de uso de ABP con similares propósitos por ejemplo el de Villagra que desarrolla de una metodología ABP para el itinerario Creación y Entretenimiento Digital del Cuarto Curso del Grado en Ingeniería Multimedia (Villagrà-Arnedo et al., 2018).

### **Metodología ABP, sus beneficios y uso**

El aprendizaje basado en proyectos (ABP) es una metodología de enseñanza que se enfoca en el aprendizaje a través de la realización de proyectos prácticos e interdisciplinarios que involucran a los estudiantes en la solución de problemas reales. Esta metodología se basa en la premisa de que el aprendizaje es más efectivo cuando se aplica a situaciones y problemas del mundo real.

Otras metodologías educativas que comparten algunos principios con el ABP incluyen el aprendizaje basado en servicio (ABS), el aprendizaje basado en competencias (ABC) y el aprendizaje basado en problemas (ABPro). En el ABS, los estudiantes aplican sus conocimientos y habilidades en proyectos que benefician a la comunidad o a organizaciones sin fines de lucro. El ABC se enfoca en desarrollar habilidades y competencias prácticas en lugar de simplemente transmitir información. Por su parte, el ABPro utiliza problemas complejos y realistas para motivar a los estudiantes a aprender y a desarrollar habilidades de resolución de problemas.

Estas metodologías enfocadas en el aprendizaje activo y la resolución de problemas han demostrado ser efectivas para involucrar a los estudiantes en su propio proceso de aprendizaje y fomentar el desarrollo de habilidades y competencias relevantes para el mundo laboral actual. El ABP, en particular, ha ganado popularidad en los últimos años debido a su capacidad para preparar a los estudiantes para enfrentar desafíos del mundo real y desarrollar habilidades valiosas en el mercado laboral.

Existen numerosos estudios que sostienen que el Aprendizaje Basado en Proyectos (ABP) impulsa habilidades clave como el **trabajo en equipo, el aprendizaje autónomo, la gestión del tiempo, la realización de proyectos y la capacidad de comunicación oral y escrita.**(Agudo-Martínez, 2012; Aznar et al., s. f.). Como ya hemos mencionado anteriormente estas son muchas de las soft skills que son tan importantes de cara al mercado laboral así que

ABP parece una buena metodología adecuada para trabajar muchas de las soft skills mencionadas ya.

El método de aprendizaje consiste en llevar a cabo un proyecto de cierta envergadura en grupo, el cual ha sido previamente analizado por el profesor para asegurarse de que el alumno tiene los recursos necesarios para resolverlo y desarrollar las destrezas requeridas. El proyecto comienza con una pregunta generadora que requiere el ejercicio del pensamiento crítico para su resolución. Esto ayuda a modelar el pensamiento crítico y brinda apoyo para que el estudiante aprenda a realizar las tareas cognitivas que caracterizan el pensamiento crítico.(Mergendoller, s. f.)

Por tanto el Aprendizaje Basado en Proyectos (ABP) es una técnica de enseñanza en la que el estudiante adquiere los conocimientos de una materia a través de la realización de un proyecto o la resolución de un problema, que ha sido diseñado y planteado de manera adecuada por el profesor. Un proyecto se considera bien diseñado si requiere que el estudiante adquiera los conocimientos y habilidades específicas que el profesor desea transmitir para poder completarlo con éxito.

El Aprendizaje Basado en Proyectos (ABP) es una metodología que se basa en problemas y que comenzó a utilizarse en la arquitectura y la ingeniería en el siglo XIX. Fue introducida en el mundo académico en los años 60 por las universidades estadounidenses y llegó a Europa unos años después.

Esta metodología supone un cambio de paradigma en la educación tradicional, ya que los alumnos se convierten en protagonistas de su propio aprendizaje y crean un resultado en base a un problema planteado. El profesor cambia su papel en el aula y se convierte en un organizador y coordinador de la actividad.

Además, el ABP hace más atractivo el aprendizaje para el estudiante, lo que se traduce en un mejor desempeño académico y una mayor perseverancia en el aprendizaje.  
(Gallego & Llorens, 2007), (Garrigós & Valero, 2012), (Valero & García, 2011), (Valero, 2012).

Todo esto hace que el enfoque práctico del ABP sea especialmente importante en la Formación Profesional.

Los proyectos de aprendizaje se llevan a cabo de diversas maneras en la actualidad, pero en general constan de varias fases comunes según estas fuentes (González, 2021; Larmer & Mergendoller, s. f.)

- En primer lugar, surge una necesidad de conocer algo y el grupo se enfrenta a una situación problemática.

- Luego, los estudiantes se plantean qué saben y qué necesitan saber.
- Después, se realiza una investigación en grupos heterogéneos y se muestra el resultado a la clase.
- Durante la elaboración del producto, los estudiantes aprenden y se crea un mapa conceptual de lo investigado.
- Finalmente, se realiza una evaluación y se plantean nuevas cuestiones, lo que puede dar lugar a un nuevo proyecto. En resumen, el proyecto de aprendizaje es un proceso iterativo y una espiral de aprendizaje que puede no tener fin.

### **Desarrollo de videojuegos para la enseñanza de programación.**

Como opción propia para mi propuesta de ABP y debido a mi expertise en la temática he elegido el desarrollo de videojuegos como forma de hacer más atractiva la materia a enseñar, Módulo de Programación del primer curso de los ciclos de Desarrollo de Aplicaciones Multiplataforma y Desarrollo de Aplicaciones Web.

Autores como Graves (Graves, 2015) dicen que la creación de videojuegos implica el uso de múltiples habilidades y conocimientos, desde la lógica y la programación hasta el diseño gráfico y la narrativa. Además, los videojuegos son una forma de entretenimiento muy popular y atractiva para los jóvenes, lo que los convierte en una herramienta educativa muy atractiva para ellos.

Al utilizar los videojuegos como proyecto educativo, los estudiantes aprenden a través de la práctica y la experimentación. Pueden aplicar sus conocimientos de programación en un contexto real y crear sus propias historias, personajes y escenarios. Al mismo tiempo, aprenden a solucionar problemas y a trabajar en equipo, ya que el desarrollo de un videojuego requiere la colaboración de múltiples disciplinas.

Además, el desarrollo de videojuegos como proyecto educativo puede ser personalizado para cada estudiante, permitiéndoles avanzar a su propio ritmo y en función de sus intereses. Los estudiantes pueden elegir el tipo de juego que desean crear y personalizarlo según sus propios gustos y preferencias. (Sevilla et al., 2012)

La idea de enseñar computación mediante el desarrollo de videojuegos no es nueva, y está bastante documentada, por ejemplo tanto Villagrà-Arnedo (Villagrà-Arnedo et al., 2018) como Watson & Fang (Watson & Fang, 2012) han usado la metodología de Aprendizaje Basado en Proyecto y el desarrollo de Videojuegos en sus aulas pero en el caso de Watson & Fang el desarrollo de videojuegos era un fin en sí mismo mientras que para Villagrà los videojuegos eran

secundarios e incluso opcionales pudiendo ser sustituidos por otro tipo de aplicaciones si así lo preferían los alumnos (cosa que según los autores no solía pasar).

Algunos autores como Douglas (Douglas, 2018), Chiazzese (Chiazzese et al., 2018) o el mismo Villagrà (Villagrà-Arnedo et al., 2018) se apoyan en el desarrollo de videojuegos como una introducción a la tecnología y la computación en general, abarcando muchos aspectos diferentes de la tecnología sin profundizar demasiado en ninguno de ellos mientras otros como Pantoja (Pantoja, 2017), Doerschuk (Doerschuk et al., 2013), Graves (Graves, 2015) o (Muratet et al., 2009) se centran en enseñar a programar y usan el desarrollo de videojuegos como hilo conductor. Este último acercamiento es el que más se aproxima a mi caso.

Los lenguajes de programación usados para enseñar programación de estos autores es variada y van desde lenguajes visuales con cajas como Scratch, pasando por lenguajes orientados a objetos como *c#* y llegando a lenguajes a más bajo nivel como C. Uno de ellos (Graves, 2015) incluso hace uso de un motor de videojuegos comercial como es Unity, el mismo que usaré yo en mi propuesta ya que es uno de los motores de videojuegos más usados en el mundo para el desarrollo a nivel profesional y además usa como lenguaje de programación *C#*, un lenguaje moderno y orientado a objetos que también se usa en muchas otras tecnologías y se presta muy bien poder aprender con él los conceptos más importantes relacionados con la programación.

Con respecto a la enseñanza del desarrollo de software en el aula, autores como Graves (Graves, 2015) dicen que la creación de videojuegos implica el uso de múltiples habilidades y conocimientos, desde la lógica y la programación hasta el diseño gráfico y la narrativa. Además, los videojuegos son una forma de entretenimiento muy popular y atractiva para los jóvenes, lo que los convierte en una herramienta educativa muy atractiva para ellos.

Al utilizar los videojuegos como proyecto educativo, los estudiantes aprenden a través de la práctica y la experimentación. Pueden aplicar sus conocimientos de programación en un contexto real y crear sus propias historias, personajes y escenarios. Al mismo tiempo, aprenden a solucionar problemas y a trabajar en equipo, ya que el desarrollo de un videojuego requiere la colaboración de múltiples disciplinas.

Además, el desarrollo de videojuegos como proyecto educativo puede ser personalizado para cada estudiante, permitiéndoles avanzar a su propio ritmo y en función de sus intereses. Los estudiantes pueden elegir el tipo de juego que desean crear y personalizarlo según sus propios gustos y preferencias. (Sevilla et al., 2012)

## 3. Objetivos.

**Objetivo General (OG):** Desarrollar habilidades valoradas por empleadores para mejorar la inserción laboral de los estudiantes.

El objetivo final es formar a los estudiantes para que estén preparados para enfrentar los retos del mercado laboral en constante cambio y evolución en el ámbito de la programación y el desarrollo de software.

**Indicadores:**

- **(OG-I1)** Porcentaje de inserción laboral al terminar los estudios.
- **(OG-I2)** Porcentaje de abandono.
- **(OG-I3)** Calidad de los empleos obtenidos.
- **(OG-I4)** Satisfacción laboral de los graduados
- **(OG-I5)** Retroalimentación de los empleadores sobre las habilidades y competencias de los estudiantes.

**Objetivo Específico 1 (OE1):** Desarrollar habilidades de colaboración, comunicación efectiva y planificación de proyectos.

Este objetivo pretende trabajar soft skills valoradas en programadores por los empleadores, especialmente las relacionadas con trabajo en equipo y con gestión del tiempo.

**Indicadores:**

- **(OE1-I1)** Calificación promedio de los proyectos grupales.
- **(OE1-I2)** Nivel de participación activa en actividades de colaboración.
- **(OE1-I3)** Retroalimentación de compañeros y profesores sobre habilidades de colaboración y comunicación.
- **(OE1-I4)** Número de conflictos entre estudiantes resueltos por ellos mismos.
- **(OE1-I5)** Cumplimiento de hitos y objetivos del proyecto
- **(OE1-I6)** Número de tareas entregadas a tiempo.

**Objetivo Específico 2 (OE2):** Aplicar metodología SCRUM.

Este objetivo se centra en proporcionar a los estudiantes las habilidades y conocimientos necesarios para trabajar en proyectos utilizando la metodología Scrum. Al desarrollar esta



habilidad, los estudiantes estarán mejor preparados para trabajar en equipos ágiles y adaptarse eficientemente a los cambios en los proyectos.

**Indicadores:**

- (OE2-I1) Cumplimiento de los roles y responsabilidades de Scrum.
- (OE2-I2) Uso efectivo de las ceremonias de Scrum.
- (OE2-I3) Entrega iterativa y cumplimiento de objetivos de los sprints.
- (OE2-I4) Capacidad de los estudiantes para adaptarse a los cambios.

**Objetivo Específico 3 (OE3):** Manejar aplicaciones de control de versiones.

El objetivo de este objetivo específico es que los estudiantes adquieran habilidades en el uso de sistemas de control de versiones, como Git, para la gestión eficiente de cambios y el seguimiento de versiones en proyectos de desarrollo de software.

**Indicadores:**

- (OE3-I1) Utilización correcta del sistema de control de versiones.
- (OE3-I2) Colaboración efectiva en el trabajo con control de versiones.
- (OE3-I3) Mantenimiento de un historial completo y organizado de cambios.
- (OE3-I4) Resolución de conflictos de fusión.

**Objetivo Específico 4 (OE4):** Acercar el desarrollo de videojuegos como una salida profesional real a los estudiantes.

Este objetivo tiene como finalidad brindar a los estudiantes una comprensión sólida del desarrollo de videojuegos como una posible salida profesional que está actualmente en auge. Se busca fomentar su interés en esta área y proporcionarles las habilidades y el conocimiento necesarios para embarcarse en una carrera en el desarrollo de videojuegos.

**Indicadores:**

- (OE4-I1) Número de alumnos interesados en ejercer como desarrolladores de videojuegos
- (OE4-I2) Número de videojuegos completados
- (OE4-I3) Nivel de calidad de los videojuegos completados
- (OE4-I4) Participación en exposiciones o eventos de videojuegos
- (OE4-I5) Empleabilidad de los estudiantes en la industria del desarrollo de videojuegos

## 4. Propuesta pedagógica

En este apartado se desarrollará ya más en profundidad la propuesta de innovación y las partes que la componen.

### 4.1. Contexto: Marco normativo

Empezaremos con el marco normativo del módulo que le da contexto a la propuesta de innovación.

#### 4.1.1 Ámbito de aplicación

La propuesta pedagógica planteada está diseñada para impartirse junto con el módulo de **Programación** de los estudios de Técnico Superior de Aplicaciones Multiplataforma. Este módulo tiene una duración total de 256 horas. Equivalencia en créditos ECTS: 14. Código: 0485. Marco normativo: [BOE](#) y [BOJA](#)

#### 4.1.2 Competencias

En el Anexo I.I de este documento se pueden ver las competencias de este módulo tal y como aparecen en el BOJA, pero en resumen podría decirse que el módulo de Programación del Ciclo DAM tiene como objetivo desarrollar competencias en la configuración y explotación de sistemas informáticos, el desarrollo de aplicaciones multiplataforma con acceso a bases de datos, la implementación de sistemas completos de formularios e informes, la participación en el desarrollo de juegos y aplicaciones en el ámbito del entretenimiento y la educación, la creación de aplicaciones para dispositivos móviles, el establecimiento de relaciones profesionales y comunicación efectiva, y el fomento del espíritu de innovación y actualización en el entorno de trabajo.

#### 4.1.3 Contenidos básicos

Como se ha hecho anteriormente con las competencias en este caso los contenidos básicos del módulo extraídos del BOJA, se pueden encontrar en el Anexo I.II

Estos contenidos básicos abarcan una amplia gama de temas. En primer lugar, se exploran los elementos fundamentales de un programa informático, como la estructura y bloques, variables, tipos de datos, operadores y expresiones. También se estudian los entornos integrados de desarrollo, tanto comerciales como de software libre, y se aprende a instalar y utilizar estos entornos, así como a crear proyectos con sus respectivas estructuras y componentes.

Luego, se profundiza en la utilización de objetos, comprendiendo sus características, instanciación, métodos, propiedades y liberación de memoria. Además, se explora el uso de librerías de objetos y se abordan conceptos como constructores y la utilización de objetos en la programación orientada a objetos.

Las estructuras de control son otro aspecto fundamental, donde se estudian las estructuras de selección, repetición y salto, junto con el control de excepciones y la depuración de programas. También se destaca la importancia de la documentación interna y externa, incluyendo comentarios y diagramas de clases.

Se aborda el desarrollo de clases, su estructura, miembros, atributos, métodos y constructores. Además, se explora el concepto de herencia y se aprende a utilizar clases heredadas y librerías de clases.

La lectura y escritura de información se estudia a través de los tipos de flujos, entrada desde teclado, salida a pantalla y manipulación de ficheros y sistemas de ficheros. También se introducen los conceptos de XML y su aplicación en la creación y manipulación de documentos.

Otros temas incluyen la utilización avanzada de clases, la gestión de la persistencia de objetos en bases de datos orientadas a objetos, y la gestión de bases de datos relacionales, incluyendo la conexión, recuperación y manipulación de información mediante consultas.

#### 4.1.4 Resultados de Aprendizaje.

Enumerados a continuación se pueden ver los distintos Resultados de aprendizaje del Módulo de programación y en el Anexo I.III se puede consultar la misma lista pero con todos sus criterios de evaluación asociados.

- RA1: Reconocer la estructura de un programa informático, identificar y relacionar los elementos propios del lenguaje de programación utilizado.
- RA2: Escribir y probar programas sencillos, reconociendo y aplicando los fundamentos de la programación orientada a objetos.
- RA3: Escribir y depurar código, analizando y utilizando las estructuras de control del lenguaje.
- RA4: Desarrollar programas organizados en clases, analizando y aplicando los principios de la programación orientada a objetos.
- RA5: Realizar operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases.

- RA6: Escribir programas que manipulen información seleccionando y utilizando tipos avanzados de datos.
- RA7: Desarrollar programas aplicando características avanzadas de los lenguajes orientados a objetos y del entorno de programación.
- RA8: Utilizar bases de datos orientadas a objetos, analizando sus características y aplicando técnicas para mantener la persistencia de la información.
- RA9: Gestionar información almacenada en bases de datos relacionales manteniendo la integridad y consistencia de los datos.

### Figura 5

Planificación y Secuenciación Módulo de Programación DAM (elaboración propia). La correspondencia de Contenidos, Resultados de Aprendizaje y Criterios de Evaluación se puede consultar en el Anexo I.

UT	Duración	CONTENIDO	RA	CE
UT1. Introducción a la Programación	32 hrs.	CB1, CB3	RA1	a) b) c) d) e) f) g) h) i)
UT2. Funciones	32 hrs.	CB1	RA2	b) d) e) f) i)
UT3. Estructuras de datos	32 hrs.	CB2	RA3	a) b) c) d) e) f) g) h)
UT4. Programación Orientada a Objetos	32 hrs.	CB2, CB4	RA4	a) b) c) d) e) f) g) h) i) j) k)
UT5. Herencia, polimorfismo, interfaces	32 hrs.	CB7	RA7	a) b) c) d) e) f) g)
UT6. Entorno Gráfico	32 hrs.	CB4, CB5	RA5	a) b) c) d) e) f) g) h)
UT7. Flujos de entrada/salida y control de excepciones	32 hrs.	CB5, CB6	RA6	a) b) c) d) e) f) g) h) i)
UT8. Gestión de bases de datos. Bases de datos orientadas a objeto	32 hrs.	CB8, CB9, CB10	RA8, RA9	a) b) c) d) e) f) g) h) a) b) c) d) e) f) g)



## 4.2 Descripción.

Como introducción corta diría que mi propuesta educativa consiste en utilizar el desarrollo de videojuegos como una herramienta práctica y atractiva para enseñar programación a los alumnos. A través del uso de C# y Unity, los estudiantes adquirirán habilidades técnicas en programación y desarrollo de videojuegos, al tiempo que desarrollarán competencias profesionales esenciales como el trabajo en equipo, la gestión del tiempo y el control de versiones. Esta propuesta intenta dar a los estudiantes una base sólida para continuar su educación en campos relacionados con la tecnología y la programación y un acercamiento al mundo profesional del desarrollo de videojuegos.

Mi propuesta educativa tiene como objetivo utilizar el desarrollo de videojuegos para enseñar a los alumnos a programar, centrándonos en el lenguaje de programación C# y el motor de videojuegos Unity. Esta propuesta se basa en aprovechar el interés de los estudiantes por los videojuegos y, al mismo tiempo, desarrollar habilidades técnicas y competencias profesionales necesarias en la industria del desarrollo de software.

Además de las habilidades técnicas, esta propuesta también se centrará en el desarrollo de habilidades blandas, como el trabajo en equipo y la gestión del tiempo. Los estudiantes trabajarán en proyectos grupales que les permitirán colaborar, comunicarse, resolver problemas y practicar la planificación efectiva. También se hará hincapié en el control de versiones utilizando herramientas como Git, que son ampliamente utilizadas en la industria del desarrollo de software para gestionar el código fuente.

El lenguaje de programación principal será C# debido a su carácter moderno y su enfoque orientado a objetos, lo cual está alineado con el marco normativo actual. Los alumnos aprenderán los conceptos fundamentales de la programación utilizando C#, como variables, estructuras de control, bucles, funciones y clases. Además, se enfatizará en la aplicación de los principios de la programación orientada a objetos para el desarrollo de videojuegos.

Unity será el motor de videojuegos utilizado en las clases ya que es el motor profesional más utilizado en la industria de los videojuegos en la actualidad y en el que tengo mucha experiencia. Unity proporciona un entorno de desarrollo accesible y poderoso que permitirá a los alumnos crear sus propios juegos de manera efectiva y visualizar los resultados de su trabajo de manera inmediata. Además, Unity ofrece una amplia documentación y una comunidad activa, lo que facilitará el aprendizaje y el apoyo continuo para los estudiantes.


El plan de estudios se dividirá en módulos progresivos que cubrirán tanto los aspectos técnicos de la programación como las habilidades blandas necesarias en el desarrollo de software.

## 4.3 Cronograma e inserción en la PD/dinámica del centro.

Características de temporización del Módulo de Programación de DAM:

- Equivalencia en créditos ECTS: 14
- Horas totales: 256
- Sesiones por semana: 4
- Horas por sesión: 2

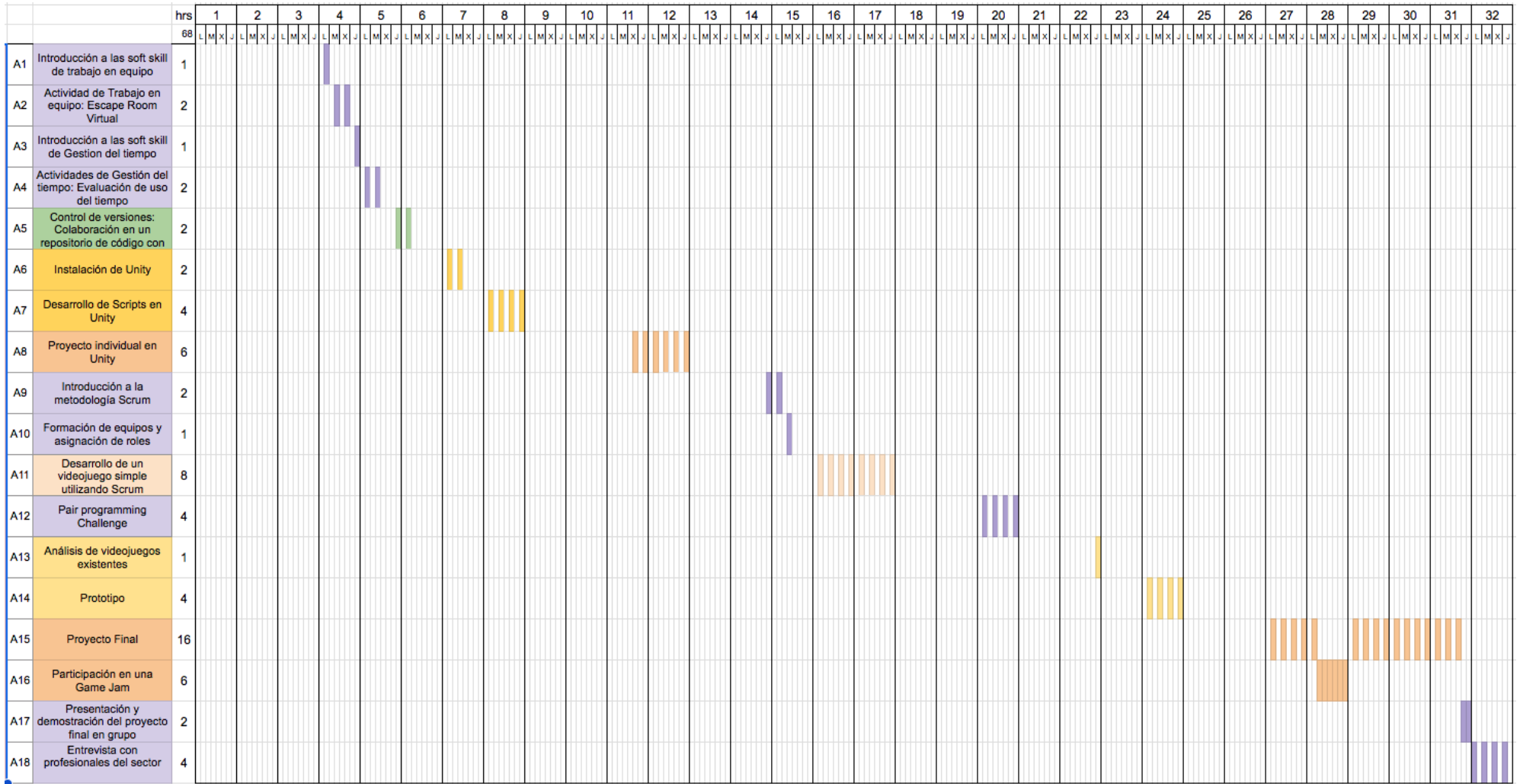
A continuación se presenta el cronograma de las actividades referentes a mi propuesta con su secuenciación y temporización.

Las actividades de mi propuesta tienen varios colores, soft skills en morado, hard skills en verde, videojuegos en amarillo y las que son una mezcla de lo anterior en naranja.  Cronogramas

Todas las actividades de la propuesta de innovación están divididas en sesiones de una hora dentro de la sesión de dos horas por día de la asignatura de programación, de este modo pretendo que las clases sean más amenas para los estudiantes. Hay un par de excepciones, las actividades de Participación en una Game Jam y la de Presentación de Proyecto Final si ocuparán las 2 horas completas por cada sesión.

**Figura 7**

Cronograma de las actividades referentes a mi propuesta con su secuenciación y temporización (elaboración propia).





## 4.4. Actividades y su descripción.

Lista de actividades referentes a mi propuesta y su duración en el aula aproximada:

- A1. Introducción a las soft skill de trabajo en equipo (1h)
- A2. **Actividad de Trabajo en equipo: Escape Room Virtual** (2h)
- A3. Introducción a las soft skill de Gestión del tiempo (1h)
- A4. **Actividades de Gestión del tiempo: Evaluación de uso del tiempo** (2h)
- A5. **Control de versiones: Colaboración en un repositorio de código con Git** (2h)
- A6. Instalación de Unity (2h)
- A7. Desarrollo de Scripts en Unity (4h)
- A8. Proyecto individual en Unity (6h)
- A9. Introducción a la metodología Scrum (2h)
- A10. Formación de equipos y asignación de roles (1h)
- A11. **Desarrollo de un videojuego simple utilizando Scrum** (8h)
- A12. **Pair programming Challenge** (4h)
- A13. Análisis de videojuegos existentes (1h)
- A14. Prototipo (4h)
- A15. Proyecto Final (16h)
- A16. **Participación en una Game Jam** (6h)
- A17. Presentación y demostración del proyecto final en grupo (2h)
- A18. **Entrevistas con Profesionales** (4h)

A continuación de describirán las actividades resaltadas en la lista anterior:

## **A2. Actividad de Trabajo en equipo: Escape Room Virtual**

### **- Preparación:**

Diseñamos una historia relacionada con la programación, como "El código secreto" o "La misión del hacker". Creamos una serie de desafíos basados en conceptos de programación, como resolver problemas algorítmicos, encontrar y corregir errores en código, descifrar códigos ocultos, etc.

### **- Desarrollo de la actividad:**

Dividimos a los estudiantes en equipos de 4 a 6 personas, asegurándonos de mezclar niveles de habilidad y conocimiento.

Explica la historia y las reglas del juego a todos los equipos.

Cada desafío debe requerir que los estudiantes colaboren y utilicen sus conocimientos técnicos.

A medida que los equipos avanzan en las estaciones, deben compartir información, trabajar en conjunto y combinar sus habilidades de programación para superar los desafíos.

El equipo que resuelva todos los desafíos y complete la misión en el menor tiempo posible será el ganador.

### **- Reflexión y discusión:**

Después de que todos los equipos hayan completado la actividad, se realiza una sesión de reflexión en grupo para que los estudiantes compartan sus experiencias y aprendizajes.

Se discutirá cómo se aplicaron las habilidades de trabajo en equipo durante la actividad, cómo se dividieron las tareas, cómo se comunicaron, cómo resolvieron los desafíos técnicos juntos, etc.

En este proceso debemos animar a los estudiantes a reflexionar sobre los obstáculos que enfrentaron y cómo los superaron trabajando en equipo destacando la importancia de la colaboración, la comunicación efectiva y el aprovechamiento de los conocimientos y habilidades individuales para lograr un objetivo común.

Para esta actividad debemos recordar adaptar los desafíos y la dificultad de la actividad según el nivel de los estudiantes y los conceptos de programación que estén estudiando. El objetivo es que trabajen juntos, utilicen sus habilidades técnicas y desarrollen la capacidad de trabajar en equipo de manera efectiva.

**Figura 8**Tabla sobre actividad **A2** Actividad de Trabajo en equipo: Escape Room Virtual

Objetivos	OG y OE1
Indicadores	OG-I4 OG-I5 OE1-I2 OE1-I2 OE1-I3 OE1-I4
Contenidos	CB1, CB3
RA y CE	RA1 a) b) c) d) e) f) g) h) i)
Evaluación (Peso)	Medición de Participación y colaboración activa (5%)
Calificación	Solo contribuye dentro del porcentaje reservado para comportamiento y actitud (5%)
Tipo de Actividad	Repaso, Refuerzo
Temporización	2h'
Recursos	Al menos un ordenador por grupo
Nivel de la pirámide de Bloom	Nivel 4: Analiza
Tipo de Agrupación	Grupos de 4 a 6 estudiantes por sorteo

#### A4. Actividades de Gestión del tiempo: Evaluación de uso del tiempo

Para esta actividad pediremos a los alumnos que registren y evalúen cómo utilizan su tiempo durante una semana.

Pueden llevar un diario de actividades o utilizar aplicaciones o herramientas en línea para hacer un seguimiento de su tiempo.

Al final de la semana abriremos un debate en clase para analizar sus registros y reflexionar sobre cómo pueden hacer ajustes para utilizar su tiempo de manera más productiva.

#### Figura 9

Tabla sobre actividad A4 Actividades de Gestión del tiempo: Evaluación de uso del tiempo

Objetivos	OG y OE1
Indicadores	OG-I2 OE1-I5 OE1-I6
Contenidos	- no aplica
RA y CE	- no aplica
Evaluación (Peso)	Que los estudiantes tomen conciencia del uso que hacen de su tiempo (5%)
Calificación	Solo contribuye dentro del porcentaje reservado para comportamiento y actitud (5%)
Tipo de Actividad	Repaso, Refuerzo
Temporización	2h'
Recursos	Diario
Nivel de la pirámide de Bloom	Nivel 5 Evalua
Tipo de Agrupación	Individual

## **A5. Control de versiones: Colaboración en un repositorio de código con Git**

En esta actividad los estudiantes trabajarán en equipos para colaborar en un proyecto de programación utilizando Git como sistema de control de versiones de este modo les permitirá a los estudiantes experimentar el flujo de trabajo de Git en un entorno colaborativo y adquirir habilidades prácticas en el uso de control de versiones. Además, aprenderán a resolver conflictos y a trabajar juntos para desarrollar un proyecto de programación.

Primero dividiremos a los estudiantes en equipos de 3 a 4 personas. Cada equipo será responsable de desarrollar una pequeña aplicación o proyecto de software de prueba, no hace falta que sea funcional.

Después cada equipo debe crear un repositorio Git en una plataforma como GitHub o GitLab. Explicaremos cómo se configura un repositorio y cómo se establece la colaboración entre los miembros del equipo.

En el siguiente paso cada miembro del equipo trabaja en una parte específica del proyecto de manera individual. Puede ser una función, una sección de código, etc. Los animaremos a utilizar ramas (branches) separadas para desarrollar sus funcionalidades y asegurarse de que cada miembro entienda cómo trabajar en su propia rama.

Después de que cada miembro haya completado su parte, deben combinar los cambios en la rama principal (por ejemplo, "main" o "master") utilizando el comando "merge" de Git. Esto implica resolver cualquier conflicto que pueda surgir durante la combinación de cambios.

Debemos introducir a los estudiantes en la resolución de conflictos de Git y explicarles cómo abordar y resolver cualquier conflicto que pueda surgir al combinar cambios. Podemos para ello crear intencionalmente algunos conflictos menores para que los estudiantes los resuelvan.

Si da tiempo también podemos introducir a los estudiantes los conceptos de tags y las revisiones de código en la plataforma.

**Figura 10**

Tabla sobre actividad **A5**. Control de versiones: Colaboración en un repositorio de código con Git

Objetivos	OG, OE1, OE3
Indicadores	OG-I1 OG-I3 OG-I5 OE1-I1 OE1-I5 OE3-I1 OE3-I2 OE3-I3 OE3-I4
Contenidos	CB1, CB3
RA y CE	RA1 a) b) c) d) e) f) g) h) i) RA2 b) d) e) f) i)
Evaluación (Peso)	Los estudiantes usan control de versiones para sus proyectos a nivel básico (5%)
Calificación	El uso de esta herramienta es transversal a la asignatura, cuenta para la clasificación de proyectos (10% de la UT correspondiente)
Tipo de Actividad	Desarrollo
Temporización	2h'
Recursos	Un ordenador con acceso a internet por alumno
Nivel de la pirámide de Bloom	Nivel 3 Aplica
Tipo de Agrupación	Grupos de 4 a 6 estudiantes por cercanía

## **A11. Desarrollo de un videojuego simple utilizando Scrum**

En esta actividad los alumnos trabajarán en equipos utilizando la metodología Scrum para desarrollar un videojuego simple, aplicando los conceptos y prácticas de Scrum durante todo el proceso. Esta actividad permitirá a los alumnos aplicar los principios y prácticas de Scrum en un contexto real de desarrollo de software. Aprenderán a gestionar un backlog, planificar sprints, trabajar de manera iterativa, recibir y aplicar retroalimentación, y presentar el producto final. Además, fomentará la colaboración en equipo, la comunicación efectiva y la adaptabilidad.

Pasos a seguir:

Formación de equipos: Dividimos a los alumnos en equipos de 4 a 6 personas. Cada equipo será responsable de desarrollar un videojuego simple.

Definición del backlog: Cada equipo debe crear un backlog de producto que contenga una lista de funcionalidades y características para el videojuego. Los alumnos pueden utilizar tarjetas o herramientas digitales para organizar y priorizar las tareas.

En este paso se le explicará a los equipos que trabajarán en sprints de una duración determinada (por ejemplo, una semana). Los equipos entonces deben seleccionar las tareas del backlog que se trabajarán en este Sprint y estimar el esfuerzo necesario para completarlas.

Sprint 1

Planificación: Los equipos deben realizar una reunión de planificación de sprint donde definan las tareas específicas para cada funcionalidad seleccionada. Pueden utilizar tableros Kanban o herramientas digitales para visualizar y asignar las tareas.

Desarrollo: Durante esta etapa, los equipos trabajan en el desarrollo del videojuego. Se debe fomentar el uso de las prácticas de Scrum, como las reuniones diarias de seguimiento (daily scrums), donde cada miembro comparte el progreso y los obstáculos que encuentra.

Revisión: Al finalizar el Sprint, los equipos realizan una reunión de revisión donde demuestran las funcionalidades desarrolladas y recopilan comentarios y sugerencias. Pueden recibir retroalimentación de otros equipos o incluso de invitados externos si es posible.

Retrospectiva: Antes de empezar un nuevo Sprint los equipos deben llevar a cabo una retrospectiva del Sprint anterior para reflexionar sobre el proceso y mejorar en futuros sprints. Discuten lo que funcionó bien, los desafíos encontrados y las oportunidades de mejora.

Siguientes sprints: Los equipos continúan con los sprints posteriores, seleccionando nuevas funcionalidades del backlog para cada sprint, planificando y desarrollando de manera iterativa.

Presentación final: Al finalizar todos los sprints, los equipos realizan una presentación final donde muestran el videojuego completo y explican el proceso de desarrollo utilizando Scrum. Pueden incluir una demostración del juego y discutir los desafíos y lecciones aprendidas durante todo el proyecto.

**Figura 11**

Tabla sobre actividad **A11**. Desarrollo de un videojuego simple utilizando Scrum

Objetivos	OG, OE1, OE2, OE4
Indicadores	OG-I1 OG-I3 OG-I4 OG-I5 OE1-I2 OE1-I5 OE1-I6 OE2-I1 OE2-I2 OE2-I3 OE2-I4 OE4-I1 OE4-I2 OE4-I3 OE4-I5
Contenidos	CB1, CB2, CB3
RA y CE	RA1 a) b) c) d) e) f) g) h) i) RA2 b) d) e) f) i) RA3 a) b) c) d) e) f) g) h)
Evaluación (Peso)	Los estudiantes siguen las pautas de scrum, entienden y respetan los procesos (5%)
Calificación	Este proyecto será parte de la calificación de la materia vista hasta el momento (10% de la UT correspondiente)
Tipo de Actividad	Desarrollo
Temporización	8h'
Recursos	Un ordenador con acceso a internet por alumno
Nivel de la pirámide de Bloom	Aplica, Desarrolla, Analiza
Tipo de Agrupación	Grupos de 4 a 6 estudiantes



## A.12 Pair Programming Challenge

En esta actividad los estudiantes trabajarán en parejas para completar un desafío de programación utilizando el enfoque de pair programming. Esta actividad permitirá a los estudiantes experimentar de primera mano los beneficios de trabajar en equipo utilizando el “pair programming”. Trabajarán la comunicación efectiva, el compartir ideas, el revisar código y la mejora de calidad de su trabajo a través de la colaboración.

Pasos a seguir:

**Formación de parejas:** Se organiza a los estudiantes en parejas. Se puede hacer al azar o intentar emparejar a estudiantes con diferentes niveles de habilidad para fomentar el aprendizaje mutuo.

**Selección del desafío:** Se elige un desafío de programación apropiado para el nivel de los estudiantes y el lenguaje de programación que estén estudiando. Puede ser un problema de algoritmos, un pequeño proyecto o cualquier otro desafío que requiera la escritura de código.

**Roles de programador y observador:** Explicamos a los estudiantes que uno de ellos actuará como el "programador" y el otro como el "observador". El programador será responsable de escribir el código, mientras que el observador se enfocará en revisar y brindar comentarios.

**Rotación de roles:** Se establece un tiempo determinado (por ejemplo, 20 minutos) para que los estudiantes desempeñen sus roles. Después de ese tiempo, se deben rotar los roles, de modo que el observador se convierta en el programador y viceversa.

**Comunicación constante:** Debe animarse a los estudiantes a mantener una comunicación constante y clara durante el proceso. Deben discutir y planificar juntos antes de comenzar a escribir código. Debemos fomentar que el observador haga preguntas, ofrezca sugerencias y proponga mejoras mientras el programador escribe el código.

**Refactorización y revisión conjunta:** Después de que ambos alumnos hayan tenido la oportunidad de ser programadores, se debe dedicar tiempo a la revisión conjunta del código. Entonces debemos animar a los estudiantes a refactorizar el código para mejorarlo y corregir cualquier error o problema identificado durante la revisión.

**Discusión y aprendizaje:** Finalizamos la actividad con una discusión grupal donde los estudiantes compartan sus experiencias. Preguntaremos qué aprendieron durante la actividad, cómo se sintieron al trabajar en pareja y qué beneficios encuentran en el enfoque del pair programming.

**Figura 12**Tabla sobre actividad **A.12** Pair Programming Challenge

Objetivos	OG, OE1
Indicadores	OG-I2 OE1-I1 OE1-I2 OE1-I3 OE1-I4
Contenidos	CB2 CB4 CB7
RA y CE	RA4 a) b) c) d) e) f) g) h) i) j) k) RA7 a) b) c) d) e) f) g)
Evaluación (Peso)	Evaluar si esta técnica se convierte en una herramienta más para el resto de actividades y su futuro profesional (5%)
Calificación	Esta tarea será parte de la calificación de la materia vista hasta el momento (10% de la UT correspondiente)
Tipo de Actividad	Desarrollo
Temporización	4h'
Recursos	Un ordenador con acceso a internet por cada 2 alumnos
Nivel de la pirámide de Bloom	Nivel 3 Aplica Nivel 4 Analiza
Tipo de Agrupación	En parejas

## A.16 Participación en una Game Jam

En esta actividad los alumnos trabajarán en equipos participando en una Game Jam, un evento de desarrollo de juegos de corta duración, donde pondrán en práctica sus habilidades de programación y creatividad. Esta actividad permite a los alumnos experimentar el desarrollo acelerado de un juego, trabajar en equipo, tomar decisiones rápidas y aplicar sus habilidades de programación y creatividad en un entorno desafiante y divertido.

Hay mucha variedad de Game Jams pero en general las más usuales duran 2 o 3 días en horario continuo según las posibilidades de los participantes, en el caso de esta actividad solo dispondremos de las 6 horas lectivas correspondientes a 3 sesiones de clase seguidas, pero se plantearía con el centro la posibilidad de que los alumnos que quieran se queden en el aula fuera de su horario para continuar con la actividad si es posible, otra posibilidad es que los alumnos continuarán la actividad desde casa en modalidad virtual.

Pasos a seguir:

- Formación de equipos
- Planificación y diseño respecto al tema de la game jam
- Desarrollo del juego
- Colaboración y comunicación
- Pruebas y ajustes
- Presentación final
- Reflexión y aprendizaje

### Figura 13

Tabla sobre actividad A.16 Participación en una Game Jam

Objetivos	OG, OE1, OE4
Indicadores	OG-I2 OE1-I1 OE1-I2 OE1-I3 OE1-I4 OE4-I1 OE4-I2 OE4-I3 OE4-I4 OE4-I5
Contenidos	CB2 CB4 CB7
RA y CE	RA4 a) b) c) d) e) f) g) h) i) j) k) RA7 a) b) c) d) e) f) g)
Evaluación (Peso)	Evaluar si los alumnos participan de buen agrado y con interés en la actividad (5%)
Calificación	Esta tarea será parte de la calificación de la materia vista hasta el momento (10%)

	de la UT correspondiente)
Tipo de Actividad	Trabajo en equipo y desarrollo
Temporización	6h'
Recursos	Un ordenador con acceso a internet por cada alumno
Nivel de la pirámide de Bloom	Aplica, Desarrolla, Analiza
Tipo de Agrupación	En grupos de 4 a 6 alumnos

## A.18 Entrevista con profesionales del sector

En esta actividad los alumnos entrevistarán a profesionales que trabajan como desarrolladores de software para obtener información sobre el día a día, los desafíos y consejos útiles relacionados con la programación, de este modo tendrán la oportunidad de interactuar directamente con profesionales de la industria y obtener información valiosa sobre la vida laboral como desarrollador de software. Esto les permitirá aclarar dudas, conocer diferentes perspectivas y obtener consejos prácticos para su futura carrera en programación.

Pasos a seguir:

- Identificación e invitación a profesionales.
- Preparación de preguntas por parte del profesor, por los alumnos e incluso por los profesionales invitados si lo desean.
- Realización de las entrevistas y posteriormente dar tiempo a los profesionales a expresar cualquier punto que crean que no se ha cubierto durante la entrevista.
- Intercambio de conclusiones, reflexión y análisis.

**Figura 14**

Tabla sobre actividad **A4** Actividades de Gestión del tiempo: Evaluación de uso del tiempo

Objetivos	OG y OE1
Indicadores	OG-I2 OG-I4 OE1-I2 OE4-I1, OE4-I5
Contenidos	- no aplica
RA y CE	- no aplica
Evaluación (Peso)	Evaluar si al final de la actividad los estudiantes tienen más claro como es el trabajo real de un programador (5%)
Calificación	Solo contribuye dentro del porcentaje reservado para comportamiento y actitud (5%)
Tipo de Actividad	Entrevista
Temporización	4h'
Recursos	Profesionales a entrevistar
Nivel de la pirámide de Bloom	Comprender
Tipo de Agrupación	Toda la clase como grupo

## 4.5. Recursos y materiales.

A continuación se describen posibles recursos y materiales necesarios para cada una de las actividades de la propuesta.

### A1.Introducción a las soft skills de trabajo en equipo:

- a. Presentación en PowerPoint o Prezi sobre las soft skills de trabajo en equipo.
- b. Ejemplos y casos de estudio que demuestren la importancia del trabajo en equipo en el desarrollo de software.
- c. Videos motivacionales que destaquen los beneficios del trabajo en equipo.

### A2.Actividad de Trabajo en equipo: Escape Room Virtual:

- a. Plataforma o herramienta para crear y ejecutar un escape room virtual.
- b. Puzzles y desafíos diseñados para fomentar la colaboración y la comunicación en equipo.

### A3.Introducción a las soft skills de Gestión del tiempo:

- a. Presentación en PowerPoint o Prezi sobre las soft skills de gestión del tiempo.
- b. Técnicas y consejos para mejorar la gestión del tiempo en un proyecto de desarrollo de software.

### A4.Actividades de Gestión del tiempo: Evaluación de uso del tiempo:

- a. Hojas de trabajo o cuestionarios para que los estudiantes evalúen y analicen cómo utilizan su tiempo.
- b. Ejercicios prácticos para planificar y organizar tareas de manera eficiente.

### A5.Control de versiones: Colaboración en un repositorio de código con Git:

- a. Acceso a una plataforma de control de versiones como GitHub o GitLab.
- b. Tutoriales o guías sobre cómo utilizar Git y realizar operaciones básicas como clonar repositorios, hacer commits y fusionar ramas.

### A6.Instalación de Unity:

- a. Instrucciones detalladas para instalar Unity en diferentes sistemas operativos.
- b. Recursos en línea, como videos o tutoriales, que guíen a los estudiantes a través del proceso de instalación.

### A7.Desarrollo de Scripts en Unity:

- a. Ejemplos de scripts simples en Unity que los estudiantes puedan estudiar y modificar.

- b. Ejercicios prácticos para que los estudiantes desarrollen sus propios scripts y comprendan los conceptos básicos de programación en Unity.
- A8. Proyecto individual en Unity:
- a. Descripción detallada del proyecto, incluyendo requisitos y objetivos.
  - b. Recursos adicionales específicos para el proyecto, como modelos 3D o assets para utilizar en Unity.
- A9. Introducción a la metodología Scrum:
- a. Presentación en PowerPoint o Prezi que explique los principios y roles de Scrum.
  - b. Ejemplos de proyectos exitosos desarrollados con Scrum.
- A10. Formación de equipos y asignación de roles:
- a. Dinámicas de formación de equipos para que los estudiantes se agrupen en equipos de desarrollo.
  - b. Roles definidos para cada miembro del equipo, como Scrum Master, Product Owner y Desarrollador.
- A11. Desarrollo de un videojuego simple utilizando Scrum:
- a. Plantilla o guía para la creación y gestión de un proyecto Scrum específico para el desarrollo de un videojuego simple.
  - b. Tablero Kanban o herramienta de gestión de proyectos en línea para realizar el seguimiento del progreso del proyecto.
- A12. Pair programming Challenge:
- a. Problemas o desafíos de programación específicos que los estudiantes puedan resolver utilizando la técnica de pair programming.
  - b. Entornos de desarrollo integrados (IDE) configurados para facilitar el pair programming.
- A13. Análisis de videojuegos existentes:
- a. Acceso a una variedad de videojuegos existentes para que los estudiantes los analicen y estudien.
  - b. Cuestionarios o pautas para el análisis de los videojuegos, enfocándose en aspectos como la jugabilidad, el diseño de niveles y la narrativa.
- A14. Prototipo:
- a. Herramientas de prototipado, como Adobe XD o Figma, para que los estudiantes creen prototipos interactivos de sus ideas de videojuegos.

- b. Retroalimentación y revisión de los prototipos por parte de los compañeros de clase.
- A15. Proyecto Final:
- a. Descripción detallada del proyecto final, incluyendo objetivos, requisitos y criterios de evaluación.
  - b. Recursos adicionales necesarios para la implementación del proyecto final en Unity.
- A16. Participación en una Game Jam:
- a. Información sobre las Game Jams disponibles, tanto presenciales como en línea.
  - b. Recursos para el desarrollo rápido de videojuegos, como assets preexistentes o plantillas de proyectos.
- A17. Presentación y demostración del proyecto final en grupo:
- a. Software de presentación, como PowerPoint o Google Slides, para que los equipos presenten su proyecto final.
  - b. Espacio adecuado y equipos audiovisuales para las demostraciones en vivo de los proyectos.
- A18. Entrevistas con Profesionales:
- a. Contactos o conexiones con profesionales de la industria del desarrollo de videojuegos.
  - b. Preguntas y guías para las entrevistas con profesionales, enfocadas en sus experiencias y consejos para los estudiantes.



## 4.6. Atención a la diversidad.

En mi opinión un plan de atención a la diversidad debería ser flexible y adaptable, considerando las necesidades cambiantes de los estudiantes a lo largo del proceso educativo. También creo que es fundamental trabajar en estrecha colaboración con el resto de docentes, el equipo directivo y los profesionales especializados para asegurar una implementación efectiva y exitosa del plan.

A este respecto algunos de los puntos a seguir para mi propuesta serían los siguientes:

- Identificación de necesidades y perfiles de los estudiantes mediante la realización de una evaluación inicial en relación con sus habilidades, conocimientos previos, estilos de aprendizaje y posibles dificultades. Esto ayudará a adaptar el plan de atención a la diversidad de manera efectiva.
- Adaptación de las actividades y materiales para garantizar que sean accesibles y adecuadas para todos los estudiantes. Se pueden considerar diferentes niveles de dificultad, formatos alternativos, apoyos visuales o auditivos, y recursos adicionales que puedan facilitar la participación y el aprendizaje de los estudiantes con necesidades específicas.
- Implementar estrategias de diferenciación instruccional para atender las diferentes capacidades y ritmos de aprendizaje de los estudiantes. Se podrían ofrecer opciones de tareas, actividades y proyectos que permitan a los estudiantes abordar los objetivos de aprendizaje de manera flexible y adaptada a sus necesidades individuales.
- Proporcionar apoyo lo más individualizado posible a los estudiantes que lo requieran. Esto puede incluir tutorías, orientación académica y emocional, retroalimentación personalizada e incluso sesiones de refuerzo específicas. Se intentará mantener una comunicación abierta y regular con los estudiantes para brindarles el apoyo necesario a lo largo del proceso de aprendizaje.
- En caso de contar con recursos disponibles se colaborará con profesionales especializados, como pedagogos, psicólogos educativos o terapeutas, para brindar apoyo adicional a los estudiantes con necesidades específicas. Se intentará establecer un trabajo conjunto para diseñar estrategias y adaptaciones personalizadas que favorezcan su participación y progreso.

- Se intentará adaptar los criterios y los métodos de evaluación para permitir que los estudiantes demuestren su aprendizaje de manera acorde a sus características y necesidades individuales. Consideraremos diferentes formas de evaluación, como proyectos alternativos, evaluaciones orales o evidencias de desempeño práctico, asegurando que sean justas y equitativas para todos los estudiantes.
- Se intentará promover un ambiente inclusivo y de colaboración entre todos los estudiantes. Haremos hincapié en la sensibilización y el respeto hacia la diversidad, fomentando la participación activa, la empatía y la valoración de las habilidades y perspectivas de cada estudiante.
- A través de la figura del tutor intentaremos mantener una comunicación abierta y fluida con las familias de los estudiantes para informarles sobre el plan de atención a la diversidad y que así se involucren en el proceso educativo y recabar su retroalimentación y colaboración.

# 5. Evaluación:

A continuación se hará un análisis de la evaluación en dos ámbitos diferentes, primero se hablará de cómo impactan las actividades de la propuesta en la nota del alumno y después sobre cómo valorar si la propuesta ha sido exitosa o no.

## 5.1. De la asignatura

Con respecto a la evaluación oficial del módulo de programación, al analizar cómo impactaría la propuesta de innovación en la nota del estudiante desde el punto de vista de cada una de las actividades planificadas que se han enumerado anteriormente podríamos clasificarlas en varios tipos según su impacto en la nota:

- **Actividades que no tienen ningún impacto en la nota.** En esta clasificación solo está la actividad que consiste en Entrevistar a profesionales del sector, esta actividad está más relacionada con la orientación profesional de los alumnos y no apoya los contenidos del módulo ni directa ni indirectamente.
- **Actividades que tienen un impacto indirecto.** Hay otras actividades que aunque no tiene una clasificación asociada a ella (una nota) son fundamentales para el desarrollo de otras actividades, y por lo tanto el haber entendido o ejecutado correctamente la actividad será ventajoso para poder obtener una mejor nota en actividades que si la tienen, en este grupo podrían estar las actividades de introducción a las soft skills o hard skills como las actividades de introducción al trabajo en equipo o de introducción al control de versiones. Estas actividades no tendrán una calificación asociada pero son base para otras actividades que sí la tendrán.
- **Actividades que tienen un impacto directo.** Estas son las actividades que aunque son específicas de la propuesta de innovación tienen una nota propia asociada al módulo de programación ya que lo que hacen es trabajar de una forma diferente el contenido de la misma junto con otras habilidades o herramientas. Por ejemplo en la actividad de desarrollo de un pequeño videojuego con Scrum estamos trabajando en el desarrollo de videojuegos y con la herramienta Scrum también propia de la innovación pero para el proyecto en sí es necesario aplicar los conocimientos de programación alcanzados hasta ese momento en el módulo por lo que este proyecto tendrá una nota asignada que impactará en la nota final. Este mismo es el caso de muchas otras actividades como la actividad de Pair Programming, la actividad de Participación en una Game Jam, el proyecto Final etc...

## 5.2. De la propuesta de innovación

El éxito de la propuesta de innovación en el módulo de programación del ciclo de DAM puede evaluarse en base a diversos factores. Aquí se presentan algunos aspectos que se podrían considerar como indicadores de éxito:

- **Adquisición de habilidades y conocimientos:** El éxito se reflejaría en el grado en que los estudiantes han desarrollado y mejorado sus habilidades técnicas, así como su comprensión de los conceptos fundamentales de programación y desarrollo de videojuegos. Esto implicaría que los estudiantes hayan adquirido competencias sólidas en el uso de herramientas y tecnologías relevantes, como Git, Unity y Scrum.
- **Desarrollo de soft skills:** En este aspecto podríamos considerar como éxito el crecimiento y la mejora de las soft skills abordadas en la propuesta, como el trabajo en equipo (colaboración y la comunicación efectiva) y la gestión del tiempo. Si los estudiantes demuestran un mayor nivel de competencia en estas áreas y aplican estas habilidades en sus proyectos y actividades, estamos ante un indicador positivo.
- **Calidad de los proyectos desarrollados:** El éxito en este caso se refleja en la creación y desarrollo de proyectos de alta calidad y en el cumplimiento de los objetivos establecidos. Podemos evaluar la complejidad, la funcionalidad, la creatividad y la presentación general de los proyectos. Si los estudiantes logran desarrollar videojuegos que demuestran un buen dominio técnico y una ejecución sólida, se considera un indicador de éxito.
- **Participación y colaboración activa:** En este aspecto el éxito se relacionaría con el grado de participación activa de los estudiantes en las actividades propuestas, como la Game Jam, el pair programming y las entrevistas con profesionales. Si los estudiantes se involucran de manera entusiasta, muestran compromiso y colaboran efectivamente con sus compañeros y profesionales de la industria, se considera un indicador positivo.
- **Retroalimentación positiva de los estudiantes:** Directamente podemos preguntar a los estudiantes. En este caso el éxito se reflejaría en la retroalimentación positiva y satisfacción de los estudiantes con la propuesta de innovación. Si los estudiantes manifiestan que han disfrutado y encontrado útiles las actividades, y que han logrado un mayor aprendizaje y crecimiento personal, esto indica un resultado exitoso.

- **Resultados académicos:** El éxito también debe evaluarse mediante el seguimiento de los resultados académicos de los estudiantes, como por ejemplo las calificaciones obtenidas en las actividades relacionadas con la propuesta. Si los estudiantes demuestran un mejor desempeño académico en comparación con periodos anteriores o con sus compañeros de otros grupos, esto puede considerarse un indicador de éxito.
- **Salidas Profesionales:** Respecto al mercado laboral sería un éxito que la inserción laboral de los alumnos egresados aumentara a causa de los conocimientos y habilidades desarrolladas, y también se podría considerar un éxito que los alumnos se planteen el desarrollo de videojuegos como una salida profesional apetecible y al alcance de sus manos.

## 7. Conclusiones y líneas futuras.

El problema identificado es la necesidad de mejorar la enseñanza en el Módulo de Programación del ciclo superior de Desarrollo de Aplicaciones Multiplataforma (DAM). Se ha observado que los enfoques tradicionales pueden resultar tediosos para los alumnos y no fomentan el desarrollo de habilidades clave demandadas en el sector por el mercado laboral.

La propuesta busca abordar este problema mediante la incorporación del desarrollo de videojuegos como una estrategia adicional de enseñanza. Se propone utilizar el desarrollo de videojuegos como un contexto de aprendizaje práctico y motivador, que involucre a los estudiantes en la aplicación de conceptos de programación y en el desarrollo de habilidades clave como el trabajo en equipo, la gestión del tiempo y el uso de herramientas y metodologías modernas. Las actividades diseñadas, como el escape room virtual, el uso de Git, la metodología Scrum y las entrevistas a profesionales, permiten a los estudiantes adquirir conocimientos técnicos, soft skills y experiencia práctica relevante para el sector.

Esta propuesta ofrece varias ventajas. Primero, pretende incrementar la atención y el compromiso de los estudiantes al trabajar en un contexto que a muchos les apasiona, como el desarrollo de videojuegos. Además, se promueve el desarrollo de habilidades técnicas y soft skills, lo que mejora la empleabilidad de los estudiantes y los prepara para el entorno laboral actual. Además, el enfoque práctico y colaborativo de las actividades fomenta el trabajo en equipo y la comunicación efectiva. Por último, se adapta a las demandas del sector, brindando a los estudiantes las herramientas y habilidades necesarias para tener éxito en la industria del desarrollo de aplicaciones multiplataforma.

Algunas posibles líneas futuras para esta propuesta educativa incluyen la exploración de tecnologías emergentes, la colaboración con la industria y la actualización constante de herramientas y tecnologías para mantener la relevancia y la calidad del programa educativo. Más en detalles podríamos enumerar las siguientes posibles líneas futuras:

- **Profundización en tecnologías emergentes:** Se pueden explorar tecnologías emergentes en el campo de los videojuegos, como la realidad virtual, la realidad aumentada o la inteligencia artificial. Esto permitirá a los estudiantes adquirir conocimientos y habilidades avanzadas en áreas de gran demanda y relevancia.
- **Colaboración con la industria del videojuego:** Se podrían establecer colaboraciones con empresas y profesionales del sector del desarrollo de videojuegos. Esto brindaría a los

estudiantes la oportunidad de trabajar en proyectos reales, recibir retroalimentación de expertos y establecer contactos profesionales que faciliten su inserción en la industria.

- Actualización constante de herramientas y tecnologías: Dado que el campo del desarrollo de videojuegos evoluciona rápidamente, es importante mantenerse actualizado con las últimas herramientas y tecnologías. Se podría considerar la integración de nuevas plataformas, motores de juego y lenguajes de programación populares para mantener la relevancia y preparar a los estudiantes para las tendencias actuales y futuras del sector.
- Expansión a otros módulos del ciclo de DAM: Se puede considerar la posibilidad de expandir la propuesta de innovación a otros módulos del ciclo de Desarrollo de Aplicaciones Multiplataforma. Esto permitirá integrar el desarrollo de videojuegos como una estrategia adicional de enseñanza en todo el programa educativo, brindando a los estudiantes una experiencia de aprendizaje coherente y enriquecedora.

## 8. Referencias

- Agudo-Martínez, M. (2012). *La metodología del ABP en un proyecto de Innovación Docente* (pp. 809-814.). <https://doi.org/10.13140/RG.2.1.2381.5521>
- Alex, K. (2009). *Soft Skills* (3.<sup>a</sup> ed., Vol. 1-1). S. Chand. [www.schandpublishing.com](http://www.schandpublishing.com)
- Allen, D., & Fallows, J. (2015). *Getting Things Done: The Art of Stress-Free Productivity*.
- Aznar, F., Pujol, M., Sempere, M., & Rizo, R. (s. f.). *Adquisición de competencias mediante Aprendizaje Basado en Proyectos como metodología docente: Valoración del alumnado*.
- BARREDO, Á. (2017, mayo 6). *Los ordenadores en clase reducen el rendimiento académico de los alumnos*. La Vanguardia. <https://www.lavanguardia.com/tecnologia/20170507/422311429662/ordenadores-educacion-clase-rendimiento-escolar.html>
- Birt, J. (s. f.). *Hard Skills vs. Soft Skills: What's the Difference?* | *Indeed.com*. Indeed Career Guide. Recuperado 2 de mayo de 2023, de <https://www.indeed.com/career-advice/resumes-cover-letters/hard-skills-vs-soft-skills>
- Blanco, B. M. (2020, febrero 18). *El «boom» del videojuego: Una industria que factura más que el cine y la música juntos*. The Conversation. <http://theconversation.com/el-boom-del-videojuego-una-industria-que-factura-mas-que-el-cine-y-la-musica-juntos-131871>
- Caeiro-Rodríguez, M., Manso-Vázquez, M., Mikic-Fonte, F. A., Llamas-Nistal, M., Fernández-Iglesias, M. J., Tsalapatas, H., Heidmann, O., De Carvalho, C. V., Jesmin, T., Terasmaa, J., & Sørensen, L. T. (2021). Teaching Soft Skills in Engineering Education: An European Perspective. *IEEE Access*, 9, 29222-29242.



<https://doi.org/10.1109/ACCESS.2021.3059516>

Cerdà-Navarro, A., Sureda-García, I., & Salvà-Mut, F. (2020). Intención de abandono y abandono durante el primer curso de Formación Profesional de Grado Medio: Un análisis tomando como referencia el concepto de implicación del estudiante («student engagement»). *Estudios sobre Educación*, 39, 33-57.

<https://doi.org/10.15581/004.39.33-57>

Chiazzese, G., Fulantelli, G., Pipitone, V., & Taibi, D. (2018). Engaging Primary School Children in Computational Thinking: Designing and Developing Videogames. *Education in the Knowledge Society (EKS)*, 19(2), Article 2.

<https://doi.org/10.14201/eks20181926381>

Conde, M. E., & García, E. (s. f.). Hací tu videojuego. Aprender a programar creando.

*Compendio de investigaciones sobre videojuegos serios*, 17.

Covey, S. R. (2015). *Los 7 hábitos de la gente altamente efectiva. Ed. revisada y actualizada: La revolución ética en la vida cotidiana y en la empresa* (J. Piatigorsky & F. M. Arribas, Trads.).

Daza Pérez, E. P., & Pardo, F. (2016). *Construir videojuegos con Scratch para fortalecer habilidades de pensamiento creativo. Aproximaciones a partir de una experiencia en el contexto rural* (pp. 20-32).

Doerschuk, P., Juarez, V., Liu, J., Vincent, D., Doss, K., & Mann, J. (2013). Introducing programming concepts through video game creation. *2013 IEEE Frontiers in Education Conference (FIE)*, 523-529. <https://doi.org/10.1109/FIE.2013.6684879>

Douglas, B. D. (2018). *Using Video Game Development as a Stepping Stone into the World of Technology* [Ashland University].

[https://etd.ohiolink.edu/apexprod/rws\\_olink/r/1501/10?clear=10&p10\\_accession\\_num=auhonors1525247444242345](https://etd.ohiolink.edu/apexprod/rws_olink/r/1501/10?clear=10&p10_accession_num=auhonors1525247444242345)

Galli, M. G., Colla, M. de la P., & Corsi, D. P. (2016). Desarrollo y Reutilización de Videojuegos como Estrategia Interdisciplinar de Enseñanza y Aprendizaje en la Educación Superior. *Compendio de investigaciones sobre videojuegos serios*.

García-Bullé, S. (2019, junio 24). *Población y efectividad en el aula, ¿cuántos alumnos son demasiados?* Observatorio / Instituto para el Futuro de la Educación.

<https://observatorio.tec.mx/edu-news/grupos-grandes-efectividad-ensenanza/>

Gómez, C. G., Aparicio, M. M., Carrasco, F. F., Sempere, M. L., Fernández, J. M. G., Soria, I. N., Juan, M. V., & Maciá, G. (s. f.). *Consecuencias conductuales derivadas del proceso atencional en el aula de aprendizaje: Propuestas de detección*.

González, F. (2021). Aprendizaje Basado en Proyectos en Formación Profesional: La aplicación de las nuevas tecnologías a la investigación de mercados en los ciclos de comercio y marketing. *Ensayos: Revista de la Facultad de Educación de Albacete*, 36(1), 105-121.

Graves, D. (2015). Promoting Computer Science with Video Games: Teaching Object-Oriented Programming through Unity. *Computer Engineering*.

<https://digitalcommons.calpoly.edu/cpesp/159>

GrupoRPP. (2016, agosto 10). *¿Es la computadora una herramienta o una distracción?* rpp.pe.

<https://rpp.pe/vital/vivir-bien/es-la-computadora-una-herramienta-o-una-distraccion-noticia-986154>

Henry. (2021, julio 13). *7 habilidades blandas para ser un gran programador*.

<https://blog.soyhenry.com/>.

<https://blog.soyhenry.com/7-habilidades-blandas-que-pueden-convertir-a-un-junior-en-un>

-gran-programador/

Hidayati, A., Budiardjo, E. K., & Purwandari, B. (2020). Hard and Soft Skills for Scrum Global Software Development Teams. *Proceedings of the 3rd International Conference on Software Engineering and Information Management*, 110-114.

<https://doi.org/10.1145/3378936.3378966>

INE. Instituto Nacional de Estadística. (s. f.). *Abandono temprano de la educación-formación de personas de 18 a 24 años. España y UE-28(12543)*. INE. Recuperado 22 de abril de 2023, de <https://www.ine.es/jaxiT3/Tabla.htm?t=12543>

INTEF. (s. f.). ¿Qué es el Aula del futuro? *El Aula del Futuro*. Recuperado 6 de mayo de 2023, de <https://auladelfuturo.intef.es/que-es-el-aula-del-futuro/>

Juárez, A. M. P., Flores, D. J. V., & Calderón, Y. I. P. (2015). Los medios distractores en el aula de clase. *Universidad y Ciencia*, 8(13), Article 13.

<https://doi.org/10.5377/uyv.v8i13.4538>

Kai. (2022, mayo 11). 29 essential hard and soft skills for Java developers. *TestGorilla*.

<https://www.testgorilla.com/blog/skills-java-developers/>

Larmer, J., & Mergendoller, J. (s. f.). *Gold Standard PBL: Essential Project Design Elements*.

Recuperado 2 de mayo de 2023, de

[https://my.pblworks.org/resource/document/gold\\_standard\\_pbl\\_essential\\_project\\_design\\_elements](https://my.pblworks.org/resource/document/gold_standard_pbl_essential_project_design_elements)

Lencioni, P. M. (2002). *The Five Dysfunctions of a Team: A Leadership Fable*.

Maquilón Sánchez, J. J., & Hernández Pina, F. (2011). Influencia de la motivación en el rendimiento académico de los estudiantes de formación profesional. *Revista electrónica interuniversitaria de formación del profesorado*, 14(1), 81-100.

- Mena, D. V. V. (s. f.). *EL DESARROLLO DE VIDEOJUEGOS EDUCATIVOS EN FORMACIÓN INICIAL, FAVORECE LA PRÁCTICA DOCENTE DE LOS FUTUROS DOCENTES.*
- Mergendoller, J. (s. f.). *Does project based learning teach critical thinking?*
- Moreno Cano, S. (2019). *Desarrollo de videojuegos en Unity para educación* [BachelorThesis].  
<https://repositorio.uam.es/handle/10486/688946>
- Morgenstern, J. (2001). *Organice su tiempo desde dentro hacia afuera* (A. S. J. Traducciones, Trad.).
- Muratet, M., Torguet, P., Jessel, J.-P., & Viallet, F. (2009). Towards a Serious Game to Help Students Learn Computer Programming. *International Journal of Computer Games Technology, 2009*, e470590. <https://doi.org/10.1155/2009/470590>
- Narváez-Díaz, L. E., & López-Martínez, R. (2021). Creación de Videojuegos como Estrategia Educativa en Algoritmia. *Revista Docentes 2.0, 12*(1), 22-30.  
<https://doi.org/10.37843/rted.v1i1.219>
- Palm, F. (2010). *LA PROGRAMACIÓN DE VIDEOJUEGOS CON PYTHON COMO ESTRATEGIA PARA UNA ALFABETIZACIÓN DIGITAL INTEGRAL.*  
<https://doi.org/10.13140/2.1.1187.6167>
- Pantoja, M. (2017). Designing a New Video Game App as an aid for Introduction to Programming classes that use C Programming Language. *Proceedings of the 3rd International Conference on Higher Education Advances, 10-16.*  
<https://doi.org/10.4995/HEAD17.2017.4606>
- Paredes, N. Y. P., Hurtado, S. N., & Malpica, D. M. R. (2018). Uso del teléfono celular como distractor del proceso enseñanza – aprendizaje. *Enfermería Investiga, 3*(4), Article 4.
- PECO, R. (2019, enero 23). *Los teléfonos en las aulas reducen la concentración incluso*

- apagados*. La Vanguardia.  
<https://www.lavanguardia.com/tecnologia/20190123/454264649476/telefonos-movil-whatsapp-colegio-prohibir-concentracion.html>
- Press, E. (2021, enero 29). *Andalucía se sitúa con 21,8% como la tercera comunidad con la tasa de abandono escolar más alta, según la EPA*. Europa Press.  
<https://www.europapress.es/andalucia/noticia-andalucia-situa-218-tercera-comunidad-tasa-abandono-escolar-mas-alta-epa-20210129183624.html>
- Roberto, C. (2014, noviembre 13). Portátiles y tablets en clase, ¿distracción o más involucración? *Blog de Lenovo*.  
<https://www.bloglenovo.es/portatiles-tablets-en-clase-mas-involucion-o-mas-distraccion/>
- Sampaio, S. C. B., Bastos, T. A., & Marinho, M. L. M. (2021). Soft skills for newborn software engineers in agile teams. *International Journal of Agile Systems and Management*, 14(1), 27-52. <https://doi.org/10.1504/IJASM.2021.114904>
- SEKHAR, G. R. (2019). TEACHING SOFT SKILLS: A NECESSITY IN MODERN ERA. *Research Journal of English Language and Literature (RJELAL)*.
- Sevilla, C., Santini, S., Haya, P. A., Rodríguez, P., & Sacha, G. M. (2012). Interdisciplinary design of videogames: A highly motivating method of learning. *2012 International Symposium on Computers in Education (SIIE)*, 1-6.
- Thompson, E. (2020). *5 hard skills every professional programmer should have*.  
tutorialride.com.  
<https://www.tutorialride.com/skills-for-software-programmers/5-hard-skills-every-professional-programmer-should-have.htm>

UniversityOfDenver. (2020, mayo 20). 18 Skills All Programmers Need to Have | University of Denver Coding Boot Camp. *University of Denver Boot Camps*.

<https://bootcamp.du.edu/blog/programming-skills/>

Villagr -Arnedo, C.-J., Gallego-Dur n, F. J., Llorens Largo, F., Lozano, M. A., Molina-Carmona, R., Mora Liz n, F. J., Sempere-Tortosa, M., I esta, J. M., Ponce de Le n Amador, P. J., & Garcia, G. J. (2018). *Desarrollo de una metodolog a ABP para el itinerario Creaci n y Entretenimiento Digital del Cuarto Curso del Grado en Ingenier a Multimedia (4002)*.

Universidad de Alicante. Instituto de Ciencias de la Educaci n.

<http://rua.ua.es/dspace/handle/10045/88673>

Watson, W. R., & Fang, J. (2012). PBL as a Framework for Implementing Video Games in the Classroom. *International Journal of Game-Based Learning (IJGBL)*, 2(1), 77-89.

<https://doi.org/10.4018/ijgbl.2012010105>

Wilce, Y. M., & Talledo, I. S. (s. f.). *El Dise o de Videojuegos como Estrategia Pedag gica. Una Propuesta para Educaci n Secundaria*.

World Economic Forum. (s. f.). *The Future of Jobs Report 2020*. World Economic Forum.

Recuperado 1 de mayo de 2023, de

<https://www.weforum.org/reports/the-future-of-jobs-report-2020/in-full/infographics-e4e69e4de7/>

Zabala, G., Mor n, R., & Blanco, S. (2013). *Una propuesta de ense anza de programaci n en escuela media mediante el desarrollo de videojuegos con Etoys*. XI Simposio sobre la Sociedad de la Informaci n (SSI) - JAIIO 42 (2013).

<http://sedici.unlp.edu.ar/handle/10915/94688>

# Anexo I: Información del Módulo de Programación de DAM extraída del BOJA

A continuación se exponen las Competencias, los Contenidos y los Resultados de Aprendizajes con sus Criterios de evaluación extraídos del marco normativo: [BOE](#) y [BOJA](#)

## I.I Competencias

La formación del módulo contribuye a alcanzar las competencias profesionales, personales y sociales de este título que se relacionan a continuación:

- a) Configurar y explotar sistemas informáticos, adaptando la configuración lógica del sistema según las necesidades de uso y los criterios establecidos.
- e) Desarrollar aplicaciones multiplataforma con acceso a bases de datos utilizando lenguajes, librerías y herramientas adecuados a las especificaciones.
- f) Desarrollar aplicaciones implementando un sistema completo de formularios e informes que permitan gestionar de forma integral la información almacenada.
- i) Participar en el desarrollo de juegos y aplicaciones en el ámbito del entretenimiento y la educación empleando técnicas, motores y entornos de desarrollo específicos.
- j) Desarrollar aplicaciones para teléfonos, PDA y otros dispositivos móviles empleando técnicas y entornos de desarrollo específicos.
- t) Establecer vías eficaces de relación profesional y comunicación con sus superiores, compañeros y subordinados, respetando la autonomía y competencias de las distintas personas.
- w) Mantener el espíritu de innovación y actualización en el ámbito de su trabajo para adaptarse a los cambios tecnológicos y organizativos de su entorno profesional.

## I.II Contenidos básicos

- CB1. Identificación de los elementos de un programa informático:
- a. - Estructura y bloques fundamentales.
  - b. - Variables.
  - c. - Tipos de datos.
  - d. - Literales.
  - e. - Constantes.
  - f. - Operadores y expresiones.
  - g. - Conversiones de tipo.
  - h. - Comentarios.
  - i. - Entornos integrados de desarrollo.
    - i. • Definición y tipos. Entornos comerciales y de Software libre.
    - ii. • Instalación y descripción de entornos integrados de desarrollo.
    - iii. • Creación de proyectos. Estructura y componentes.
- CB2. Utilización de objetos:
- a. - Características de los objetos.
  - b. - Instanciación de objetos.
  - c. - Utilización de métodos.
  - d. - Utilización de propiedades.
  - e. - Utilización de métodos estáticos.
  - f. - Librerías de objetos. Inclusión y uso.
  - g. - Constructores.
  - h. - Destrucción de objetos y liberación de memoria.
  - i. - Entornos de desarrollo para programación orientada a
  - j. Objetos.
    - i. • Entornos específicos.
    - ii. • Plugins de integración en entornos genéricos.
- CB3. Uso de estructuras de control:
- a. - Estructuras de selección.
  - b. - Estructuras de repetición.
  - c. - Estructuras de salto.
  - d. - Control de excepciones.
  - e. - Depuración de programas.
  - f. - El depurador como herramienta de control de errores.
  - g. - Documentación de programas.
    - i. • Documentación interna, comentarios.
    - ii. • Documentación externa, diagramas de clases, requisitos, guías, etc.
- CB4. Desarrollo de clases:
- a. - Concepto de clase y objeto.
  - b. - Estructura y miembros de una clase. Diagramas de clase.



- c. - Creación de atributos.
  - d. - Creación de métodos.
  - e. - Creación de constructores.
  - f. - Utilización de clases y objetos.
  - g. - Concepto de herencia. Tipos. Utilización de clases heredadas.
  - h. - Librerías de clases. Creación. Inclusión y uso de la interface.
- CB5. Lectura y escritura de información:
- a. - Tipos de flujos. Flujos de bytes y de caracteres.
  - b. - Clases relativas a flujos.
  - c. - Utilización de flujos.
  - d. - Entrada desde teclado.
  - e. - Salida a pantalla.
  - f. - Ficheros de datos. Registros.
  - g. - Apertura y cierre de ficheros. Modos de acceso.
  - h. - Escritura y lectura de información en ficheros.
  - i. - Utilización de los sistemas de ficheros.
  - j. - Creación y eliminación de ficheros y directorios.
  - k. - Interfaces.
  - l. - Concepto de evento.
  - m. - Creación de controladores de eventos.
- CB6. Aplicación de las estructuras de almacenamiento:
- a. - Estructuras. Definición y uso.
  - b. - Concepto de Array. Tipos. Creación de arrays. Recorrido y búsquedas en un array.
  - c. - Arrays multidimensionales.
  - d. - Cadenas de caracteres. Uso de las cadenas. Recorrido y manipulación. Uso de expresiones regulares en cadenas de texto.
  - e. - Concepto de Lista. Tipos. Operaciones.
  - f. - Aplicación del estándar XML.
  - g. - Concepto de XML Estructura de un documento XML.
  - h. - Especificación de documentos. DTD y XSD.
  - i. - Clases para la creación y manipulación de documentos XML.
- CB7. Utilización avanzada de clases:
- a. - Composición de clases.
  - b. - Herencia.
  - c. - Superclases y subclases.
  - d. - Clases y métodos abstractos y finales.
  - e. - Sobreescritura de métodos.
  - f. - Constructores y herencia.

- CB8. Mantenimiento de la persistencia de los objetos:
- a. - Bases de datos orientadas a objetos.
  - b. - Características de las bases de datos orientadas a objetos.
  - c. - Instalación del gestor de bases de datos.
  - d. - Creación de bases de datos.
  - e. - Mecanismos de consulta.
  - f. - El lenguaje de consultas, sintaxis, expresiones, operadores.
  - g. - Recuperación, modificación y borrado de información.
  - h. - Tipos de datos objeto; atributos y métodos.
  - i. - Tipos de datos colección.
- CB9. Gestión de bases de datos relacionales:
- a. - Conexión con bases de datos relacionales. Características, tipos y métodos de acceso.
  - b. - Establecimiento de conexiones. Componentes de acceso a datos.
  - c. - Recuperación de información. Selección de registros.
- CB10. Uso de parámetros.
- a. - Manipulación de la información. Altas, bajas y modificaciones.
  - b. - Ejecución de consultas sobre la base de datos

### I.III Resultados de Aprendizaje.

**RA1.** Reconoce la estructura de un programa informático, identificando y relacionando los elementos propios del lenguaje de programación utilizado.

Criterios de evaluación:

- a) Se han identificado los bloques que componen la estructura de un programa informático.
- b) Se han creado proyectos de desarrollo de aplicaciones.
- c) Se han utilizado entornos integrados de desarrollo.
- d) Se han identificado los distintos tipos de variables y la utilidad específica de cada uno.
- e) Se ha modificado el código de un programa para crear y utilizar variables.
- f) Se han creado y utilizado constantes y literales.
- g) Se han clasificado, reconocido y utilizado en expresiones los operadores del lenguaje.
- h) Se ha comprobado el funcionamiento de las conversiones de tipos explícitas e implícitas.
- i) Se han introducido comentarios en el código.

**RA2.** Escribe y prueba programas sencillos, reconociendo y aplicando los fundamentos de la programación orientada a objetos.

Criterios de evaluación:

- a) Se han identificado los fundamentos de la programación orientada a objetos.
- b) Se han escrito programas simples.
- c) Se han instanciado objetos a partir de clases predefinidas.
- d) Se han utilizado métodos y propiedades de los objetos.
- e) Se han escrito llamadas a métodos estáticos.
- f) Se han utilizado parámetros en la llamada a métodos.
- g) Se han incorporado y utilizado librerías de objetos.
- h) Se han utilizado constructores.
- i) Se ha utilizado el entorno integrado de desarrollo en la creación y compilación de programas simples.

**RA3.** Escribe y depura código, analizando y utilizando las estructuras de control del lenguaje.

Criterios de evaluación:

- a) Se ha escrito y probado código que haga uso de estructuras de selección.
- b) Se han utilizado estructuras de repetición.
- c) Se han utilizado estructuras de repetición.
- d) Se han reconocido las posibilidades de las sentencias de salto.

- e) Se ha escrito código utilizando control de excepciones.
- f) Se han creado programas ejecutables utilizando diferentes estructuras de control.
- g) Se han probado y depurado los programas.
- h) Se ha comentado y documentado el código.

**RA4.** Desarrolla programas organizados en clases analizando y aplicando los principios de la programación orientada a objetos.

Criterios de evaluación:

- a) Se ha reconocido la sintaxis, estructura y componentes típicos de una clase. b) Se han definido clases.
- c) Se han definido clases.
- d) Se han definido propiedades y métodos.
- e) Se han creado constructores.
- f) Se han desarrollado programas que instancian y utilizan objetos de las clases creadas anteriormente.
- g) Se han utilizado mecanismos para controlar la visibilidad de las clases y de sus miembros.
- h) Se han definido y utilizado clases heredadas.
- i) Se han creado y utilizado métodos estáticos.
- j) Se han definido y utilizado interfaces.
- k) Se han creado y utilizado conjuntos y librerías de clases.

**RA5.** Realiza operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases.

Criterios de evaluación:

- a) Se ha utilizado la consola para realizar operaciones de entrada y salida de información.
- b) Se han aplicado formatos en la visualización de la información.
- c) Se han reconocido las posibilidades de entrada/salida del lenguaje y las librerías asociadas.
- d) Se han utilizado ficheros para almacenar y recuperar información.
- e) Se han creado programas que utilicen diversos métodos de acceso al contenido de los ficheros.
- f) Se han utilizado las herramientas del entorno de desarrollo para crear interfaces gráficos de usuario simples.
- g) Se han programado controladores de eventos.

h) Se han escrito programas que utilicen interfaces gráficas para la entrada y salida de información.

**RA6.** Escribe programas que manipulen información seleccionando y utilizando tipos avanzados de datos.

Criterios de evaluación:

- a) Se han escrito programas que utilicen arrays
- b) Se han reconocido las librerías de clases relacionadas con tipos de datos avanzados.
- c) Se han utilizado listas para almacenar y procesar información.
- d) Se han utilizado iteradores para recorrer los elementos de las listas.
- e) Se han reconocido las características y ventajas de cada una de las colecciones de datos disponibles.
- f) Se han creado clases y métodos genéricos.
- g) Se han utilizado expresiones regulares en la búsqueda de patrones en cadenas de texto. Sevilla, 21 de julio 2011 BOJA núm. 142 Página núm. 169
- h) Se han identificado las clases relacionadas con el tratamiento de documentos XML.
- i) Se han realizado programas que realicen manipulaciones sobre documentos XML.

**RA7.** Desarrolla programas aplicando características avanzadas de los lenguajes orientados a objetos y del entorno de programación.

Criterios de evaluación:

- a) Se han identificado los conceptos de herencia, superclase y subclase.
- b) Se han utilizado modificadores para bloquear y forzar la herencia de clases y métodos.
- c) Se ha reconocido la incidencia de los constructores en la herencia.
- d) Se han creado clases heredadas que sobrescriben la implementación de métodos de la superclase.
- e) Se han diseñado y aplicado jerarquías de clases.
- f) Se han probado y depurado las jerarquías de clases.
- g) Se han realizado programas que implementen y utilicen jerarquías de clases. h) Se ha comentado y documentado el código.

**RA8.** Utiliza bases de datos orientadas a objetos, analizando sus características y aplicando técnicas para mantener la persistencia de la información.

Criterios de evaluación:

- a) Se han identificado las características de las bases de datos orientadas a objetos.
- b) Se ha analizado su aplicación en el desarrollo de aplicaciones mediante lenguajes orientados a objetos.
- c) Se han instalado sistemas gestores de bases de datos orientados a objetos.
- d) Se han clasificado y analizado los distintos métodos soportados por los sistemas gestores para la gestión de la información almacenada.
- e) Se han creado bases de datos y las estructuras necesarias para el almacenamiento de objetos.
- f) Se han programado aplicaciones que almacenen objetos en las bases de datos creadas.
- g) Se han realizado programas para recuperar, actualizar y eliminar objetos de las bases de datos.
- h) Se han realizado programas para almacenar y gestionar tipos de datos estructurados, compuestos y relacionados.

**RA9.** Gestiona información almacenada en bases de datos relacionales manteniendo la integridad y consistencia de los datos.

Criterios de evaluación:

- a) Se han identificado las características y métodos de acceso a sistemas gestores de bases de datos relacionales.
- b) Se han programado conexiones con bases de datos.
- c) Se ha escrito código para almacenar información en bases de datos.
- d) Se han creado programas para recuperar y mostrar información almacenada en bases de datos.
- e) Se han efectuado borrados y modificaciones sobre la información almacenada.
- f) Se han creado aplicaciones que ejecuten consultas sobre bases de datos.
- g) Se han creado aplicaciones para posibilitar la gestión de información presente en bases de datos relacionales.