



Universidad de Oviedo

Facultad de Formación del Profesorado y Educación

**Máster en Formación del Profesorado de Educación
Secundaria Obligatoria, Bachillerato y Formación
Profesional**

**DISEÑO Y DESARROLLO DE VIDEOJUEGOS
APLICADO A LA ENSEÑANZA DE LA DISCIPLINA
INFORMÁTICA**

**VIDEOGAMES DESIGN AND DEVELOPEMENT
APPLIED TO THE TEACHING OF COMPUTER
SCIENCE**

TRABAJO FIN DE MÁSTER

Autor: Raúl Ángel Lago Castro

Tutor: María Ángeles Díaz Fondón

Especialidad: Informática

Curso académico: 2022-2023

Mayo 2023

Índice

Resumen.....	1
Abstract	1
Introducción	2
1 Reflexión: formación recibida y prácticas profesionales	3
1.1 Aprendizaje y Desarrollo de la Personalidad (ADP).....	3
1.2 Aprendizaje y Enseñanza en las especialidades de Informática y Tecnología (AEIT).....	3
1.3 Complementos a la Formación Disciplinar en las especialidades de Informática y Tecnología (CFDIT).....	4
1.4 Diseño y Desarrollo del Currículum (DDC).....	4
1.5 Innovación Docente e Iniciación a la Investigación Educativa (IDIIE) .	4
1.6 Lengua Inglesa para el Aula Bilingüe (LIAB).....	5
1.7 Procesos y Contextos Educativos (PCE)	5
1.8 Sociedad, Familia y Educación (SFE)	5
1.9 Tecnologías de la Información y la Comunicación (TIC)	5
1.10 Prácticum	6
1.11 Análisis del currículo oficial.....	6
2 Proyecto de innovación educativa.....	8
2.1 Introducción	8
2.2 Contextualización de la propuesta	9
2.3 Fundamentación teórica	9
2.3.1 Investigación previa	9
2.3.2 Relevancia a nivel social.....	11
2.3.3 Aclaraciones conceptuales	12
2.4 Análisis de necesidades	12
2.5 Instrumentos de recogida de información.....	14

2.6	Agentes involucrados.....	14
2.6.1	Equipo innovador.....	14
2.6.2	Participación, coordinación y toma de decisiones.....	15
2.7	Descripción e implementación de la propuesta de innovación.....	15
2.7.1	Implementación de la propuesta.....	16
2.7.2	Ejemplo extrapolado de intervención real.....	20
2.7.3	Materiales de apoyo y recursos necesarios.....	25
2.7.4	Evaluación de la implementación.....	25
2.7.5	Perspectivas de continuidad.....	26
2.8	Rendimiento de la propuesta.....	26
2.8.1	Objetivos de la innovación y resultados esperados.....	26
2.8.2	Objetivos a largo plazo y colaterales.....	27
2.8.3	Instrumento de evaluación de la propuesta de innovación.....	27
2.8.4	Análisis de los resultados.....	28
2.8.5	Puntos fuertes y puntos débiles de la innovación.....	29
2.8.6	Propuestas para mejorar la innovación.....	30
2.9	Reflexión personal sobre el proceso de innovación.....	30
3	Propuesta de programación docente.....	31
3.1	Introducción.....	31
3.2	Contexto.....	32
3.2.1	Legislación.....	32
3.2.2	Centro.....	37
3.2.3	Grupo clase.....	37
3.3	Desglose legislativo.....	38
3.3.1	Competencia general.....	38
3.3.2	Competencias profesionales, personales y sociales.....	38

3.3.3	Relación de cualificaciones y unidades de competencia del Catálogo Nacional de Cualificaciones Profesionales incluidas en el título	41
3.3.4	Objetivos generales	41
3.3.5	Contenidos.....	44
3.3.6	Resultados de aprendizaje y criterios de evaluación.....	47
3.4	Temporalización.....	52
3.5	Unidades de Trabajo	54
3.6	Metodología	61
3.7	Actividades	63
3.7.1	Superposición de trimestres	64
3.7.2	Ejemplo concreto de trimestre.....	65
3.8	Coordinación intradepartamental.....	75
3.9	Actividades complementarias/extraescolares	76
3.10	Atención a la diversidad	76
3.11	Evaluación	77
3.11.1	Evaluación inicial	77
3.11.2	Evaluación del aprendizaje.....	78
3.11.3	Instrumentos de evaluación	79
3.11.4	Estrategias de recuperación y evaluación extraordinaria	80
3.11.5	Evaluación de asignatura pendiente	81
3.12	Evaluación docente	81
	Conclusiones	82
	Referencias bibliográficas.....	83
	Investigación	83
	Legislación	85

Resumen

El presente documento constituye el Trabajo Fin de Máster de Raúl Ángel Lago Castro; en él se plasman todos los conocimientos adquiridos en el Máster Universitario en Formación del Profesorado de Educación Secundaria Obligatoria, Bachillerato y Formación Profesional en la especialidad de Informática.

El documento se estructura en tres partes diferenciadas: una reflexión sobre el propio máster, las prácticas del mismo y el currículo actual; una propuesta de innovación docente, cuyo objetivo es tomar un enfoque metodológico centrado en aplicar el diseño y desarrollo de videojuegos en la enseñanza de la disciplina informática; y el diseño de una programación docente que emplee la innovación para el primer curso del Ciclo Formativo de Grado Superior en Desarrollo de Aplicaciones Web en el Módulo de Programación.

Abstract

The present document constitutes Raúl Ángel Lago Castro's Master's Thesis, in which all the knowledge acquired in the Master's Degree in Teacher Training for Compulsory Secondary Education, Baccalaureate, and Vocational Training in the field of Computer Science is reflected.

The document is structured into three distinct parts: a reflection on the master's program itself, its practicum, and the current curriculum; a proposal for innovative teaching, aimed at adopting a methodological approach focused on the application of game design and development in computer science education; and the design of a didactic program that employs the before-mentioned innovation during the first year of the Advanced Specific Vocational Training in Web Application Development in the Programming Module.

Introducción

A modo de introducción, se abre este documento con la siguiente cita célebre, que es su día inspiró y sigue inspirando al autor de este documento mientras persigue un futuro en la docencia:

- “El profesor mediocre dice. El buen profesor explica. El profesor superior demuestra. El gran profesor inspira” (William A. Ward).

1 Reflexión: formación recibida y prácticas profesionales

La formación adquirida en el Máster ha sido indudablemente enriquecedora; son muchos los aspectos mejorables, pero también aquellos que han contribuido a formar al autor como docente y cuyos aprendizajes han acabado convergiendo en el prácticum. En este apartado se analizan los aspectos más destacados de la experiencia obtenida a través de las asignaturas cursadas, incluyendo el prácticum y, además, se incluye una reflexión sobre el currículum en sí.

1.1 Aprendizaje y Desarrollo de la Personalidad (ADP)

Esta asignatura es la joya de la corona del máster. No solo enseña conceptos importantísimos para entender el comportamiento del alumnado y conceptos pedagógicos fundamentales; sino que presenta y ejecuta un enfoque docente basado en el aprendizaje significativo (Ausubel, 1983). Es de las pocas asignaturas que predica con el ejemplo y practica lo que enseña.

1.2 Aprendizaje y Enseñanza en las especialidades de Informática y Tecnología (AEIT)

Esta asignatura es clave para entender la ley educativa, los elementos curriculares y demás conceptos burocráticos y formales pertinentes a la actividad docente. En el primer semestre hubiera sido extremadamente útil; en el segundo la alta carga de trabajo y el examen se superpone a otras clases, el prácticum y el TFM; haciendo que no se pueda aprovechar correctamente la asignatura y entorpeciendo una etapa ya suficientemente complicada del máster. Es prácticamente incompatible tener que ir a los IES a las 8:00, salir a las 14:00 y luego tener clase de 16:00 a 20:00.

1.3 Complementos a la Formación Disciplinar en las especialidades de Informática y Tecnología (CFDIT)

Esta asignatura debería ser dos, una para la especialidad de tecnología y otra para la especialidad de informática. El tener las dos simultáneamente no es intrínsecamente malo, pero la parte que le toca a tecnología ha sido completamente inútil para los informáticos; solo ha servido para quitar una cantidad ingente de tiempo tanto con las interminables lecciones magistrales, como con el trabajo semanal en el cuaderno de clase. Sin embargo, la parte de informática no solo ha sido mucho más amena, sino que ha dotado de una gran cantidad de herramientas, metodologías y valiosa experiencia delante de una clase; sin duda una aportación mucho más enriquecedora y útil para futuros docentes.

1.4 Diseño y Desarrollo del Currículum (DDC)

Al contrario que Aprendizaje y Enseñanza, esta asignatura debería ubicarse en el segundo semestre; tiene muy baja carga de trabajo y los conceptos que trata no son asimilables por alumnos y alumnas recién entrados en el máster. La asignatura se planteó como una reflexión sobre los nuevos conceptos de la nueva ley de educación, pero no se puede hacer una reflexión sobre conceptos que todavía no se entienden (ni los nuevos ni los antiguos).

1.5 Innovación Docente e Iniciación a la Investigación Educativa (IDIIE)

A esta asignatura le ocurre algo similar a Aprendizaje y Enseñanza, por situarse en el segundo semestre; además, a eso hay que añadirle su deficiente estructuración y lo insulso que es su contenido. El gran aporte que ofrece esta asignatura es un guion para la innovación y la obligación de entregarlo antes que el TFM, lo cual facilita el desarrollo del propio TFM.

La asignatura está simplemente mal planteada, debería ser un foro de debate en vez de la repetición incesante de unos conceptos manidos, redundantes y cansinos. Es un claro

ejemplo de una clase en la que es urgente innovar de verdad y no tratar de aplicar metodologías vacías que no aportan nada a los que serán los futuros docentes.

1.6 Lengua Inglesa para el Aula Bilingüe (LIAB)

A través de esta asignatura se ha tomado auténtica conciencia de lo que implica el Aprendizaje Integrado de Contenido en Lengua Extranjera (AICLE). Además, al tratar la lengua inglesa como una herramienta de aprendizaje, se ha obtenido una comprensión mucho más profunda de la misma; tanto a la hora de expresarse como a la hora de preparar clases en cualquier idioma, esta asignatura otorga un enfoque mucho más amplio y rico.

1.7 Procesos y Contextos Educativos (PCE)

Asignatura esencial para asentar todas las facetas fundamentales de la formación de un docente. Lo único mejorable es lo confuso que resulta tener tres docentes diferentes para una misma materia; por lo demás es la asignatura troncal que sustenta el máster y aporta un gran valor al mismo.

1.8 Sociedad, Familia y Educación (SFE)

Lo que más se extrae de la asignatura es un discurso coherente, empático y adaptado a las necesidades provocadas por la situación social actual. No solo aporta a nivel educativo, sino que provoca una reflexión más profunda que se puede llevar al día a día.

1.9 Tecnologías de la Información y la Comunicación (TIC)

La asignatura que más se ha disfrutado por el mero hecho del nuevo enfoque con el que se ha ofrecido cursarla a los estudiantes de la especialidad de informática. Con ese nuevo enfoque, se ha creado un producto tangible; un servicio a la asignatura y que, a su vez, permite a otras promociones escalarlo e iterar sobre él. No hay mejor ejemplo de innovación constructiva y eficaz en todo el máster.

1.10 Prácticum

Las prácticas en el IES han sido un golpe de realidad en muchos aspectos. Es un entorno mucho más hostil de lo esperado: con trifulcas internas entre el profesorado, colaboración interdepartamental e intradepartamental casi nula y enfoques metodológicos nefastos ejecutados por docentes incapaces de empatizar con su alumnado lo más mínimo. Por suerte hay excepciones en excelentes docentes que hacen lo que pueden por llevar al alumnado hacia delante.

Por otra parte, se ha podido constatar que el alumnado de FP es muy diferente al de la ESO. El ambiente es radicalmente opuesto, notándose la falta de interés del alumnado de FP frente a la falta de autonomía del alumnado de la ESO. Preparar clases para los primeros ha resultado un reto a la hora de estimar tiempos, pues el ritmo de trabajo es mucho más irregular de lo anticipado; a lo que hay que sumarle la diferencia de rendimiento que presentan dependiendo del docente que se encuentre en el aula, llegando incluso a adoptar actitudes y predisposiciones distintas entre clases consecutivas.

Por último, hay un componente que resulta altamente preocupante: los cargos administrativos en el IES. Todo docente entrevistado al respecto ha respondido lo mismo: odia su cargo extra (todo lo que no sea dar clase); le engañaron para tomarlo, no le dieron opción o le tocó; y espera poder renunciar, justificar una reestructuración o engañar a otra persona pronto para librarse del mismo. Esto evidencia un error de conceptualización de estos cargos, que no tienen que ver directamente con la actividad docente y que la administración no parece estar supervisando adecuadamente.

1.11 Análisis del currículo oficial

El currículo de los ciclos superiores de la disciplina informática data de 2009 y 2010; eso implica que no tiene en cuenta la cantidad de transformaciones que ha sufrido la disciplina informática en la última década. Mientras que en matemáticas las ecuaciones se hacen igual o en idiomas encontramos poco cambio (las lenguas extranjeras relevantes siguen siendo las mismas); en informática ni si quiera se ha mantenido un lenguaje de programación prevalente mucho tiempo (en su día lenguajes como COBOL, BASIC, Pascal, C, C++ o C# eran los más usados; hoy lo es Java y mañana probablemente lo sea

Python, si es que no lo es ya), mientras que el currículo está pensado para tecnologías cada vez más en desuso (como PHP).

Hay módulos pertenecientes a los anteriormente mencionados ciclos que simplemente están mal conceptualizados. En este documento hablaremos del Módulo de Programación que es común a los primeros años de dos de los ciclos de grado superior (Desarrollo de Aplicaciones Web y Desarrollo de Aplicaciones Multiplataforma); en el que el currículo, en lugar de centrarse en la programación (clave para el resto de módulos, parte troncal de la disciplina y extremadamente importante para garantizar que el ciclo sea profesionalizante) incluye contenidos y resultados de aprendizaje relacionados con bases de datos (algo pertinente a otros módulos y que resta un tiempo valiosísimo al módulo si no se opta por ignorar el currículo). En el apartado de “[Propuesta de programación docente](#)” se vuelve a reiterar sobre ello.

Por último, destacar la difícil labor a la que los docentes de la disciplina informática se tienen que enfrentar al entrar a dar clase en Formación Profesional; sin importar la formación previa del docente, puede acabar dando Programación, Bases de Datos, Redes, Ciberseguridad, Big Data, Inteligencia Artificial, Desarrollo de Videojuegos o Realidad Virtual; por mencionar algunas de las posibilidades. Además de esta gran casuística, el docente se ve obligado a actualizarse constantemente si quiere seguir el ritmo de las cada vez más volátiles tendencias en la disciplina.

2 Proyecto de innovación educativa

2.1 Introducción

La innovación educativa se refiere a la introducción de nuevos enfoques, métodos, recursos y tecnologías en el ámbito de la educación, con el objetivo de mejorar los procesos de enseñanza-aprendizaje. Se trata de una actividad dinámica y creativa que busca adaptar la educación a las necesidades y demandas cambiantes de los/las estudiantes y de la sociedad en general. A la hora de elaborar una programación docente (como se desarrolla en el apartado de “[Propuesta de programación docente](#)”), la importancia de la innovación educativa radica en su capacidad para fomentar la motivación y el interés de los/las estudiantes, promover un aprendizaje más activo y significativo, y desarrollar habilidades y competencias relevantes para el mundo actual. Al integrar enfoques innovadores en la programación docente, los/las educadores/as pueden generar un entorno de enseñanza más estimulante y efectivo; preparando así a los/las estudiantes para enfrentarse a los desafíos del siglo XXI y formando ciudadanos y ciudadanas críticos y creativos.

Esta propuesta es fruto de una situación particular a la que se debe hacer referencia a nivel personal. La intención de esta sección es aportar contexto sobre el pasado del autor (la persona redactando este documento), el cual le otorgó una perspectiva muy específica sobre el proceso innovador.

“En 2020 terminé mis estudios universitarios en Diseño y Desarrollo de Videojuegos; ese mismo año obtuve una plaza de docente para sustituir a uno de mis antiguos profesores que se retiraba. Sin haber recibido una formación docente formal, acabé al frente de 3 asignaturas de 2º y 3º de carrera durante dos años completos; lo cual, sin ser consciente de ello, me llevó a innovar. Mi intención era dar las clases como me hubiera gustado haberlas recibido, haciendo que, de la forma más amena posible, mis alumnos y alumnas aprendieran y se divirtieran”.

La propuesta en sí se titula “Metaludificación” (o “Metagamificación” para aquellos/as que prefieran usar el anglicismo), pues, si la ludificación consiste en llevar los juegos recreativos al contexto educativo, ésta trata de conseguir el mismo nivel de motivación en el proceso de crear los propios juegos. Podríamos decir que es una

metodología a través de la cual el alumnado crea videojuegos arraigados en sus intereses con los que consiguen afianzar los conceptos vistos en clase logrando un aprendizaje significativo incorporando un alto nivel de motivación intrínseca; es usar el desarrollo de videojuegos como motor de cambio en las asignaturas y módulos en los que la Programación es nuclear.

2.2 Contextualización de la propuesta

Para esta innovación en concreto, nos situamos en un centro del entorno urbano, en el primer curso de un Ciclo Formativo de Grado Superior en Desarrollo de Aplicaciones Web. El alumnado es principalmente masculino, con 22 hombres y 5 mujeres de entre 18 y 37 años. El módulo a tratar, Programación, se desarrolla en un aula equipada con equipos informáticos para todos los alumnos y alumnas, además del equipo del profesor, un proyector y pizarra.

Es necesario considerar que el alumnado actual presenta ciertas dificultades, vienen de dos años de irregularidades provocadas por la pandemia de 2020 y esto se evidencia en un nivel de partida muy heterogéneo que acentúa las diferencias existentes entre aquellos que vienen de Bachillerato, Grados Medios u otros estudios.

2.3 Fundamentación teórica

2.3.1 *Investigación previa*

Pese a haber gran abundancia de artículos e investigaciones sobre “gamificación” y el uso de videojuegos en la enseñanza, son escasos los documentos de esa índole en los que se trate la aplicación del propio desarrollo de videojuegos en un contexto educativo.

En España, el documento que más se acerca a ese concepto es un Trabajo de Fin de Máster (TFM) del Máster de Formación del Profesorado de la Universidad de Cantabria titulado “*El Diseño de Videojuegos como Estrategia Pedagógica. Una Propuesta para Educación Secundaria*” y escrito por Yolanda Mojena Wilce (2018). En ese TFM se plantea que sean los docentes los que tomen el papel de desarrolladores de videojuegos,

creen videojuegos y los usen en sus clases; esto les daría una versatilidad enorme al permitir a los mismos crear herramientas personalizadas y adaptadas a las necesidades pedagógicas de un grupo en concreto sin casi importar la materia a tratar. A pesar de que ese enfoque es muy interesante, aparte de servir como fuente de referencia para algunas ideas, no termina de explorar las posibilidades que ofrece que sean los alumnos los que desarrollen los juegos.

Otros artículos de investigación extranjeros toman el mismo enfoque que Yolanda Mojena Wilce; como *“GAMED: digital educational game development methodology”* de Serdar Aslan y Osman Balci (2015); o *“Game object model version II: a theoretical framework for educational game development”* de Alen Amory (2007).

Por otro lado, están los artículos que tratan el desarrollo de videojuegos para estudios especializados en el mismo; siendo *“Game design and game-development education”* de Mohan Rajagopalan and David I. Schwartz (2005) su mayor exponente. Estos artículos son muy útiles dentro de su campo, pero tampoco abordan el usar el desarrollo de videojuegos como enfoque metodológico fuera de las aulas donde se estudia concretamente esa disciplina.

Por último, se encuentran los artículos que sí tratan el desarrollo de videojuegos en un contexto educativo:

- *“Game Development for Computer Science Education”* de Chris Johnson, Monica McGill, Durell Bouchard, Michael K. Bradshaw, Víctor A. Bucheli, Laurence D. Merkle, Michael James Scott, Z. Sweedyk, J. Ángel Velázquez-Iturbide, Zhiping Xiao y Ming Zhang (2016):
 - En este artículo se explora el valor de los videojuegos para enriquecer la educación en informática, facilitando varias condiciones que promueven el aprendizaje:
 - Motivación de los estudiantes.
 - Aprendizaje activo.
 - Adaptabilidad.
 - Colaboración.
 - Simulación.
 - Recalca lo fácil que es recolectar métricas de aprendizaje en ese contexto.

- “*Fundamental Computer Science Conceptual Understandings for High School Students Using Original Computer Game Design*” de Jeremy V. Ernst y Aaron C. Clark (2012):
 - En este artículo se explora el valor de los videojuegos en las disciplinas basadas en STEM y la cantidad de recursos disponibles para usarlos como metodología en el aula.
- “*Scalable game design and the development of a checklist for getting computational thinking into public schools*” de Alexander Repenning, David Webb y Andri Ioannidou (2010):
 - Este artículo contempla la dificultad de la adquisición del pensamiento computacional y cómo desarrollar una estrategia plausible a través del diseño de videojuegos.
- “*Exploring adolescent's STEM learning through scaffolded game design*” de Alex Games y Luke Kane (2011):
 - Este artículo es mucho más general e involucra los factores socio-culturales que pueden afectar a la adquisición de pensamiento computacional.

2.3.2 Relevancia a nivel social

Un aspecto que es importante destacar es la relevancia de los videojuegos en la sociedad actual. Según Newzoo (2022) los videojuegos han ingresado el año pasado más de 184 miles de millones de dólares a pesar de haber sufrido receso tras el gran crecimiento provocado por la pandemia mundial de 2020; durante el periodo de redacción del TFM de Yolanda Mojena Wilce, que se realizó en 2017, la industria estaba facturando cerca de 108 miles de millones según las mismas fuentes. Esto los sigue situando como una de las industrias de entretenimiento más prominentes de nuestra era y cuya relevancia justifica sobremanera que los currículos pertinentes a la disciplina informática los exploren, aunque sea colateralmente.

2.3.3 *Aclaraciones conceptuales*

Es imperativo tener claro el concepto de las metodologías ágiles y el marco de referencia Scrum. En contraste con metodologías de desarrollo más arcaicas que se enfocan hacia extensiva documentación, planificación y secuenciación; las metodologías ágiles priorizan la obtención temprana de software funcional, la retroalimentación continua y la iteración sobre un producto; resultan en un desarrollo muy adaptativo (Abrahamsson, Salo, Ronkainen, & Warsta, 2017). Por otro lado, dentro de las metodologías ágiles, el marco de referencia Scrum es muy sencillo de entender y aplicar; se basa en realizar incrementos (sprints) de un producto supervisado por personas con diferentes roles, creando una “pila” de trabajo de la que se van extrayendo tareas y puliéndolo cada vez más (Schwaber & Sutherland, 2020). Este enfoque le daría al docente una cantidad ingente de herramientas para adaptar el desarrollo de proyectos a las necesidades de un grupo clase, con un gran control y capacidad para analizar el desempeño de las tareas.

2.4 Análisis de necesidades

En las titulaciones de ciclos formativos de grado medio y superior relativas a la disciplina informática se encuentra un **alumnado extremadamente desmotivado** en las materias donde se enseña a programar, su único interés en esas materias son las salidas profesionales que posibilita. Esto se refleja en **bajas notas, escaso trabajo en casa** (según los docentes entrevistados, solo el 10% del alumnado realiza habitualmente alguna tarea fuera de clase) y gran desinterés **en general** (baja atención en clase, uso constante de dispositivos móviles para consumir contenido ajeno a la materia o jugar a juegos online; entre otros). Ante esta situación, el profesorado toma dos caminos:

- Bajar el nivel de exigencia para adaptarse al nivel de la clase e ir repescando a los que aun así suspendan a través de las sucesivas convocatorias y a lo largo del tiempo que sea necesario.
- Mantener el nivel de exigencia, asumiendo el fracaso general de la clase.

Durante la estancia en el centro, se ha podido constatar que el profesorado no se plantea en ningún momento un cambio metodológico real; en cambio, el miedo a nuevas herramientas que usan inteligencia artificial (como el ChatGTP) tiene a los docentes bloqueados y recurriendo a técnicas ineficaces para prevenir su uso; en vez de explorar las posibilidades que ofrece.

Para intentar constatar de manera más rigurosa estos hechos, se llevó a cabo una encuesta que pretendía sondear la opinión del alumnado. En el grupo clase que en el que se desarrolló la innovación se realizó una encuesta previa a la aplicación de la innovación a la que respondieron 22 de los/as 27 alumnos/as con los siguientes resultados:

- A la pregunta: “¿Consideras que la manera de enseñar el módulo de Programación es la adecuada?”
 - 7 personas respondieron “Sí” (31,8%).
 - 15 personas respondieron “No” (68,2%).
- A la pregunta: “¿Crees que tu rendimiento en el módulo de Programación es el adecuado?”
 - 12 personas respondieron “Sí” (54,5%).
 - 10 personas respondieron “No” (45,5%).
- A la pregunta: “¿Consideras que vas a aprobar el módulo de Programación?”
 - 6 personas respondieron “Sí” (27,3%).
 - 16 personas respondieron “No” (72,7%).

Estos resultados reflejan cierta disociación entre el rendimiento real de gran parte del alumnado en el módulo y el que perciben como el requerido, mientras que la otra parte es consciente de que no están esforzándose lo suficiente; tampoco están de acuerdo con la metodología empleada por el docente a cargo del módulo. En cuanto a los resultados académicos, el pesimismo reinante era justificado dado que de los 27 solo 5 (18,5%) aprobaron Programación el segundo trimestre.

El área de mejora es amplia, el mero hecho de asumir que la metodología actual no funciona o debería enfocarse de otra manera ya tendría un impacto positivo en el aula, pues se introduciría un elemento perturbador de la decadente rutina actual.

2.5 Instrumentos de recogida de información

Para fundamentar esta propuesta se han usado dos vías principales de recogida de información:

- Observación directa y entrevistas informales que analizan la información de manera cualitativa.
- Encuestas que analizan la información de manera cuantitativa.

Con las entrevistas informales se ha extraído información principalmente del equipo docente; este fue el primer contacto con la falta de motivación en el alumnado que luego se confirmó con la observación directa del mismo. Además, se obtuvo una perspectiva completa de las metodologías de los docentes que se desarrollaban hasta el momento; basadas en clases expositivas y una cantidad ingente de tiempo para realizar ejercicios en clase, que también se confirmó con la observación directa.

Con las encuestas se extrajo información directa del alumnado y con esta se analizarán los resultados de la innovación.

El autor de este documento fue el que recogió la información; hablando con el equipo docente, el alumnado y pidiendo a estos últimos que rellenaran las encuestas pertinentes (que no todos/as rellenaron).

2.6 Agentes involucrados

2.6.1 *Equipo innovador*

El equipo innovador a cargo de la implementación de esta innovación fue el autor de este documento y el tutor de prácticas que permitió el desarrollo de la innovación en su aula. No se contó con más apoyo ni con la participación de agentes externos, pues parte del espíritu de la innovación es que se pueda desarrollar con un solo docente.

2.6.2 Participación, coordinación y toma de decisiones

En la ejecución de la innovación se contó con la participación de los 27 alumnos y alumnas, aunque en la encuesta previa y posterior solo participaron 22 y 23 alumnos y alumnas respectivamente; por lo demás, el autor de la innovación fue el encargado de dirigir la clase y de la toma de decisiones, relegándose el tutor al papel de supervisor y espectador.

2.7 Descripción e implementación de la propuesta de innovación

Esta innovación es metodológica, tratándose de un proyecto que busca, sobre todo, paliar la gran falta de motivación e interés que presenta el alumnado de ciclos de grado superior en la disciplina informática. En apartados posteriores se exploran más profundamente los “[Objetivos de la innovación y resultados esperados](#)”, así como los “[Objetivos a largo plazo y colaterales](#)”.

La innovación se ha ejecutado en 16 sesiones (4 sesiones de 2h por semana, ocupando un mes del curso académico) durante la mitad del segundo trimestre en el módulo de Programación del Ciclo Formativo de Grado Superior de Formación Profesional en Desarrollo de Aplicaciones Web. Se trata de un proyecto plurimetodológico en el que se diseña y desarrolla un videojuego de temática libre en el que se aplican todos los conocimientos previos en un contexto diferenciado.

Las metodologías activas de enseñanza empleadas principalmente son las siguientes:

- Aprendizaje basado en proyectos.
 - Enfoque pedagógico que se centra en la realización de proyectos o tareas prácticas como base principal del aprendizaje de los estudiantes en lugar de centrarse únicamente en la adquisición de conocimientos teóricos. Este enfoque fomenta la participación activa de los estudiantes en la resolución de problemas y la aplicación de habilidades y conocimientos en contextos reales.
- Pensamiento de diseño.

- Enfoque que se utiliza para abordar problemas complejos y desarrollar soluciones innovadoras. Se basa en la forma en la que los diseñadores abordan los desafíos, centrándose en la comprensión profunda de las necesidades y deseos de las personas involucradas, y en la generación de ideas creativas y viables para resolver problemas.
- (Meta)gamificación.
 - Véase el apartado “[Introducción](#)” de este proyecto de innovación educativa.
- Aprendizaje cooperativo.
 - Enfoque educativo que se basa en la colaboración y el trabajo en equipo entre los estudiantes para lograr objetivos comunes de aprendizaje. A diferencia del aprendizaje individual, donde cada estudiante trabaja de manera independiente, en el aprendizaje cooperativo los estudiantes trabajan juntos en grupos pequeños, aprovechando sus habilidades y conocimientos individuales para lograr un objetivo compartido.

Durante el proyecto se usan los términos “sprint” para referirse a los incrementos de la metodología Scrum definida anteriormente y el término “feedback” para la retroalimentación; por tratarse de términos estándar en la industria de los videojuegos y los proyectos informáticos en general.

2.7.1 Implementación de la propuesta

El proyecto se distribuye en distintas fases repartidas en las 16 sesiones que ocupa. Como el módulo donde se realiza la innovación cuenta con 4 sesiones semanales de 2 horas, sería conveniente empezar el penúltimo día de una semana y seguir durante 4 semanas seguidas, reservando los dos primeros días y el último de cada semana para las sesiones “especiales” como se explica y desglosa a continuación:

- **Fase 1: Diseño del videojuego (Sesión 1).**
 - En esta primera sesión el docente explica el proyecto, pidiendo al alumnado que se distribuyan en grupos.

- El proyecto tiene unos requerimientos mínimos que todos los proyectos deben alcanzar. Al realizarse durante el segundo trimestre de este módulo en concreto los requerimientos de la entrega son:
 - Entregar un ejecutable del videojuego que pueda ejecutarse sin errores.
 - Usar estructuras de control.
 - Seguir la Metodología de la Programación Orientada a Objetos (no confundir con las metodologías activas de enseñanza):
 - Realizar un diseño de clases adecuado.
 - Usar clases abstractas y polimorfismo.
 - Usar estructuras de almacenamiento de una manera eficiente.
 - Comentar el código de manera funcional para el desarrollo.
 - Aplicar el feedback recibido de una manera significativa.
 - Documentar el proyecto.
 - Siguiendo la filosofía de las metodologías ágiles (no confundir con las metodologías activas de enseñanza) la documentación no debe ser extensa sino funcional, una foto de una servilleta se puede ilustrar una idea de manera más eficaz que una página de texto.
 - Se deberá plasmar el impacto que el feedback ha tenido en el entregable final.
- Se deja tiempo para que los distintos grupos lleguen a una idea (sería conveniente haber comentado previamente la realización de este proyecto para que ya se estén barajando opciones) que deberán exponer al resto de grupos y al docente en la siguiente sesión. El tiempo es suficiente para que no sea necesario trabajar en casa en la propia presentación.
- Es importante que la idea motive a todos los miembros del grupo, que esté arraigada en sus intereses. Como es obvio, hay grupos que se atascarán en esta fase, por lo que el docente debe estar atento para localizar dichos grupos, ver que cosas les interesan y proponerles ideas relacionadas con éstas. Es muy sencillo tomar muchos tipos de videojuegos sencillos y

adaptarlos a cualquier temática, aunque sea un programa de televisión o un deporte.

- **Fase 2: Presentación de la idea (Sesión 2).**
 - Ante toda la clase cada grupo expone su idea, explicando cual es el producto final que tratarán de desarrollar. El formato de dicha presentación se enfoca a “vender la idea”, como si los oyentes fueran un grupo de inversores del que dependiera sus empleos. Es lo que más se valorará de cara a calificar esta parte.
 - Se podrán emplear todo tipo de recursos: desde hacerlo a viva voz hasta realizar un montaje con música y vídeo.
 - Al final de cada presentación, el resto de grupos dan su impresión y critican la idea constructivamente.
 - Con esta presentación los grupos se comprometen a realizar ese proyecto. En realidad, durante el desarrollo surgen problemas y es normal que la idea inicial se altere, pero el alumnado debe tener en mente que se tienen que aproximar a ésta lo máximo la posible.
 - El docente vela porque los grupos menos ambiciosos revisen sus ideas y aquellos demasiado ambiciosos reduzcan sus expectativas, pero dentro de lo posible se intenta que la idea original sea respetada.
- **Fase 3: Sprint 1 (Sesiones 3 a 5).**
 - Al principio de cada semana los grupos eligen qué tareas realizar durante las tres primeras sesiones de entre todas las que determinen. Para ello deben reunirse cada día, siguiendo la metodología Scrum, y establecer sus prioridades.
 - El proyecto se realiza principalmente en clase, aunque los grupos más ambiciosos lo complementan con trabajo en casa.
 - El docente toma nota del trabajo diario, el cual los grupos deben plasmar en la documentación. La nota es individual y la observación de ese trabajo influirá en ella.
 - El docente también sirve de apoyo a los grupos, haciendo apartes para resolver las dudas más generales.

- Según esta distribución, los grupos cuentan con 3 sesiones (de 2 horas cada una) para preparar el entregable de cada semana (el fin del sprint). Ese entregable debe ser funcional (que se pueda ejecutar y probar), como dicta la metodología Scrum, pero no se tendrá en cuenta que estos sean versiones reducidas o muy poco pulidas.
- **Fase 4: Entregable, presentación y feedback 1 (Sesión 6).**
 - Al final de cada semana, durante la cuarta sesión tras haber pasado 3 sesiones desarrollando el proyecto, los grupos vuelven a presentar; exponiendo esta vez los avances realizados, los problemas solucionados y los problemas que aún están afrontando. Estas presentaciones se enfocan desde un punto de vista puramente de desarrollo, dejando a un lado el “vender la idea/producto”.
 - En las presentaciones se muestra el videojuego funcionando y el resto de grupos prueban el mismo ejecutable una vez termine la presentación para darles feedback; el cual anotan para enfocar el resto del desarrollo, fundamental de cara a la entrega final.
 - El resto de grupos tratan de aportar soluciones a los problemas de cada grupo y toman nota de las soluciones a los problemas resueltos.
 - El docente enviará su feedback de manera más formal y calificará tanto las presentaciones como la entrega.
- **Fase 5: Sprint 2 (Sesión 7 a 10).**
 - Igual que el sprint 1, pero contando con más feedback y recursos para seguir avanzando.
- **Fase 6: Entregable, presentación y feedback 2 (Sesión 11).**
 - Igual que la anterior presentación, pero se hará más hincapié en tener un prototipo avanzado.
- **Fase 7: Sprint 3 (Sesión 12 a 14).**
 - Igual que el sprint 1 y 2, pero contando con que será la última semana de desarrollo; se deben establecer las tareas finales y tener en consideración el tiempo necesario para solucionar errores y pulir el videojuego.
- **Fase 8: Entrega y presentación final (Sesión 15).**

- Esta vez la presentación es justo lo contrario, se deberá “vender” el videojuego a toda costa. Se presenta un tráiler y se juega al videojuego en directo; se podrá falsear todo: editar el tráiler, fingir jugar en directo, jugar a un ejecutable preparado para la presentación... lo que haga falta. Que la presentación “venta” será lo que más se valore de cara a la nota.
- Tras la presentación el resto de grupos juegan al juego, viendo si la presentación es fiel a la realidad.
- El ejecutable se entrega junto a la documentación, cerrándose así todo el proceso de desarrollo. Estos serán evaluados por el docente.
- **Fase 9: Defensa (Sesión 16).**
 - En esta última sesión, los grupos serán convocados de manera individual (un solo grupo simultáneamente).
 - El docente jugará a los videojuegos y revisará el código de los mismos mientras realiza preguntas a todos los miembros del grupo; haciendo hincapié en comprobar su conocimiento sobre el trabajo realizado por los demás miembros del grupo y llegando a poder pedir modificaciones en el código en los casos en los que el trabajo visto en clase no se corresponda con el nivel de calidad de la entrega.

Fuera de esta planificación, se pedirá feedback a los/as alumnos/as para mejorar futuras iteraciones de esta propuesta; siendo, en cierta manera, un proyecto a desarrollar en sí mismo.

Aunque se emplea el marco Scrum, como se comenta previamente varias veces, no se sigue a rajatabla; los roles no se llegan a asignar, siendo el docente el que tome los de mayor nivel y dando directrices al alumnado para que se centren en los sprints y realicen las reuniones que sean necesarias cuando sea necesario.

2.7.2 Ejemplo extrapolado de intervención real

A continuación, se recoge uno de los proyectos que se realizaron siguiendo esta innovación. Es un ejemplo muy aislado que solo abarca una Unidad de Trabajo que se le

dio al docente: Estructuras de almacenamiento. Se establece la Unidad de Trabajo de la siguiente manera:

Unidad de Trabajo 5 – Estructuras de almacenamiento		
Temporalización:		
Segundo trimestre	28 horas	16 sesiones
Resultados de aprendizaje:		
<p>RA3. Escribe y depura código, analizando y utilizando las estructuras de control del lenguaje.</p> <p>RA4. Desarrolla programas organizados en clases analizando y aplicando los principios de la programación orientada a objetos.</p>		
Contenidos:		
<ul style="list-style-type: none"> – Vectores y sus operaciones. – Matrices y sus operaciones. – Cadenas de caracteres. – Características de las colecciones de objetos. – Estructuras dinámicas: listas. – Operaciones con listas lineales: inserción, búsqueda, recorrido, borrado. 		
Actividades de enseñanza-aprendizaje:		
<ul style="list-style-type: none"> • Escribir programas que utilicen arrays. • Reconocer las librerías de clases relacionadas con tipos de datos avanzados. • Reconocer las características y ventajas de cada una de las colecciones de datos disponibles. • Utilizar listas para almacenar y procesar información. • Utilizar iteradores para recorrer los elementos de las listas 		

Siguiendo lo establecido en la Unidad de Trabajo anterior, el proyecto realizado por un grupo de alumnos/as avanzó por las fases de la innovación establecidas de la siguiente manera:

- **Fase 1: Diseño del videojuego (Sesión 1).**
 - Este grupo en concreto se formó rápidamente, pero tuvo alguna dificultad para entender lo que el docente planteaba; acostumbrados a una metodología muy diferente, esperaban un enunciado muy concreto y el concepto de diseñar ellos un programa que se adaptara a unos requerimientos les resultaba extraño.
 - Los requerimientos mínimos del proyecto fueron los siguientes:
 - Entregar un ejecutable del videojuego que pueda ejecutarse sin errores.
 - Usar vectores, matrices y/o cadenas de caracteres.

- Implementar al menos una lista que forme parte del loop central del juego y se recorra al menos una vez.
- Seguir la Metodología de la Programación Orientada a Objetos:
 - Realizar un diseño de clases adecuado.
 - Usar clases abstractas y polimorfismo.
- Usar estructuras de almacenamiento de una manera eficiente.
- Comentar el código de manera funcional para el desarrollo.
- Aplicar el feedback recibido de una manera significativa.
- Documentar el proyecto.
 - Siguiendo la filosofía de las metodologías ágiles la documentación no debe ser extensa sino funcional, una foto de una servilleta se puede ilustrar una idea de manera más eficaz que una página de texto.
 - Se deberá plasmar el impacto que el feedback ha tenido en el entregable final.
- El grupo que estamos tratando estaba barajando una idea muy abstracta, querían hacer un juego de construcción como *Minecraft*, que se viera en la consola con una vista 2D lateral y en la que el personaje se desplazara al estilo de los juegos de *Super Mario Bros*; el docente, les ayudó a llegar a una idea menos ambiciosa, un juego con vista cenital en el que se pudiera “minar” hacia abajo por distintas capas, al estilo del juego *Dwarf Fortress* (cuya versión original se ejecutaba en la consola nativa de Windows y sus elementos se representaban solo con código ASCII).
- Tras aterrizar la idea, se pusieron a trabajar en la presentación y el docente les recomendó investigar el juego *Dwarf Fortress* para tener más referencias.
- **Fase 2: Presentación de la idea (Sesión 2).**
 - Ese fin de semana el grupo preparó una presentación y el día de la sesión escenificando un pequeño teatro en el que se realizaban ellos mismos preguntas sobre el juego (imitando una parodia de un creador de contenido online sobre las presentaciones del E3, un festival anual de videojuegos en el que se presentan los títulos que saldrán durante los próximos años).

- El feedback recibido de los compañeros fue bueno y la idea se transmitió correctamente; pero el docente les sugirió más ajustes, pues seguía siendo una idea demasiado ambiciosa.
- **Fase 3: Sprint 1 (Sesiones 3 a 5).**
 - Para el primer sprint, el grupo estableció que lo mejor sería generar un array tridimensional y rellenarlo de “piedras” y “minerales” que luego el personaje pudiera “minar” y guardar en la lista que representaría su inventario.
 - Las reuniones se solventaban rápidamente y las clases las dedicaban a programar e intercambiar ideas y problemas entre ellos/as y los grupos cercanos.
 - La semana fue bien y con algunas indicaciones del profesor crearon una jerarquía de clases adecuada y consiguieron llenar el array tridimensional según los criterios establecidos.
- **Fase 4: Entregable, presentación y feedback 1 (Sesión 6).**
 - El entregable fue simple, pero cumplía con los criterios. El array se mostraba de manera no gráfica (simplemente líneas de texto con el nombre del “material” en cada “casilla”), pero se generaba aleatoriamente cada vez, pudiendo establecer el porcentaje de aparición de cada “material”.
 - La presentación fue un poco accidentada, dado que la persona que tomó la voz cantante no entendía algunos de los problemas que habían superado. Se notaba que no habían preparado nada y estaban improvisando. Sin embargo, la explicación del juego y su ejecución fue correcta.
 - El resto de grupos aportaron algo de feedback, sobre todo ideas sobre como representar el juego de manera gráfica.
 - El docente envió feedback más formal, recomendando un curso de acción para la siguiente semana.
- **Fase 5: Sprint 2 (Sesión 7 a 10).**
 - Para este sprint, el grupo estableció prioritario representar la primera capa o nivel del array de una manera gráfica y poder mover un “personaje” por ella.

- El desarrollo no fue muy fructífero y dedicaron la semana a probar y desechar enfoques.
- Al final del sprint solo la representación gráfica funcionaba correctamente.
- **Fase 6: Entregable, presentación y feedback 2 (Sesión 11).**
 - En el entregable solo se añadió la representación gráfica del primer nivel; no incluyeron el intento de implementar el personaje porque provocaba errores.
 - La presentación siguió la misma línea; sin preparar, pero mostrando lo que llevaban hecho.
 - El feedback de los otros grupos fue muy positivo, los caracteres ASCII elegidos gustaron y los elementos que limitaban el “terreno” se veían claros.
 - En el feedback el docente se les recomendó dejar esa representación gráfica como la final y desechar el modelo por niveles del array tridimensional y dedicar el último sprint a conseguir que el personaje “minara” y “guardara en el inventario minerales” desde un array bidimensional. Con un botón para “crear objetos” si se tenía la cantidad suficiente de ciertos “materiales”.
- **Fase 7: Sprint 3 (Sesión 12 a 14).**
 - Con los nuevos objetivos dados por el docente desarrollaron esa versión más simple del videojuego.
- **Fase 8: Entrega y presentación final (Sesión 15).**
 - El entregable consistió en un videojuego de minar una capa de tierra y crear objetos, con el objetivo de crear la mayor cantidad de objetos posible (la puntuación del juego). Es un juego diferente del que tenían en mente, orientado a construir una fortaleza subterránea, pero que llegaba a los mínimos establecidos en la rúbrica.
 - El tráiler del juego fue muy espectacular, con muchos efectos de edición y postproceso que estaban incluidos en un programa de edición de vídeo para móvil. No falsearon nada, aunque mientras jugaban exageraron todo lo que hacían como si fuera lo mejor que hubieran probado en su vida.

- Los otros grupos dieron feedback algo negativo (sin ser destructivo), pues consideraban que el videojuego era muy simple y carente de un objetivo real.
- La documentación no fue muy extensa, pero quedaba muy claro el trabajo realizado por cada miembro del grupo y se correspondía con lo observado en clase por el docente.
- **Fase 9: Defensa (Sesión 16).**
 - La defensa de ese grupo fue muy corta, todos sus miembros respondieron a las preguntas del docente satisfactoriamente y realizaron las modificaciones pertinentes correctamente. Durante la defensa repitieron varias veces que no estaban orgullosos del resultado final y que ojalá hubieran tenido más tiempo para realizar la idea original del videojuego.
- **Nota final: 6**
 - El grupo llegó a los mínimos, pero falló a la hora de realizar ampliaciones, incluir contenido extra o pulir el resultado final. Las presentaciones de las sesiones 6 y 11 les penalizaron, obteniendo solo la mitad del punto al que estas equivalían en la nota.

2.7.3 Materiales de apoyo y recursos necesarios

Para realizar esta innovación no es necesario ningún material ni recurso especial; un aula de informática al uso (equipos y proyector) y que el docente tenga un equipo mínimamente potente y con espacio para almacenar los proyectos de los/as alumnos/as.

2.7.4 Evaluación de la implementación

La implementación se realizó sin incidentes y todos los grupos pudieron entregar su proyecto. Al haber el autor realizado versiones de esta innovación con anterioridad en el ámbito universitario, ya había pulido su técnica lo suficiente para adaptarla al alumnado de un grado superior sin percances.

El principal problema, desde el punto de vista del autor, fue el entorno de desarrollo empleado; al estar los alumnos habituados a usar Eclipse (el entorno principal de Java) y

no haber usado otra cosa que la consola proporcionada por este, las opciones de soporte gráfico han resultado muy limitadas. Sin embargo, esto ha hecho que el alumnado haya tenido que ser mucho más creativo (cosa fundamental en esta propuesta) a la hora de usar caracteres ASCII para representar cosas tan complejas como personajes y escenarios.

En el apartado de “[Instrumento de evaluación de la propuesta de innovación](#)” se trata la evaluación de los resultados fruto de la encuesta posterior a la innovación.

2.7.5 Perspectivas de continuidad

Esta propuesta tiene el potencial de adaptarse a cualquier asignatura que tenga una parte importante de programación, incluso en los módulos más centrados en desarrollo web o asignaturas de secundaria. En el futuro, crear un guía de implementación de esta propuesta sería lo adecuado para hacer que otros docentes que no tengan un trasfondo en el mundo del desarrollo de videojuegos la puedan aplicar sin dificultad.

2.8 Rendimiento de la propuesta

2.8.1 Objetivos de la innovación y resultados esperados

Los objetivos y los resultados esperados de la innovación son los siguientes:

- Lograr que emerja una motivación intrínseca en los/as alumnos/as que provoque un nivel de esfuerzo superior de cara a la realización del proyecto.
- Conseguir que el alumnado adquiera las competencias de una manera amena, eficiente y duradera.
- Cambiar la concepción y predisposición el alumnado en las asignaturas y módulos de programación.
- Fomentar la creatividad, el pensamiento de diseño y el pensamiento computacional.

2.8.2 *Objetivos a largo plazo y colaterales*

Aparte de los objetivos de la innovación y los resultados esperados; se establecen una serie de objetivos a largo plazo y colaterales, que la innovación fomenta y podrían ser muy beneficiosos para el futuro laboral del alumnado:

- Volver el aprendizaje de la programación una actividad creativa que lleve a los/las alumnos/as a encontrar sus propias soluciones, no a seguir una serie de pasos marcados.
- Acostumbrar al alumnado a hablar en público y a saber “venderse”. Estas habilidades no solo aumentarán su éxito en las entrevistas de trabajo, sino en el desempeño de sus profesiones; una idea solo es tan buena como la persona sea capaz de venderla.

2.8.3 *Instrumento de evaluación de la propuesta de innovación*

El instrumento para evaluar esta propuesta es la encuesta, como se incluye a continuación. Antes de ejecutar la propuesta se realizó una encuesta previa como ya se plasmó en el apartado de “[Análisis de necesidades](#)”. De manera posterior se realizó otra encuesta, a la que respondieron 23 de los/las 27 alumnos/as, para ver si se han alcanzado los objetivos establecidos; los resultados fueron los siguientes:

- A la pregunta: “¿Consideras que el proyecto es una manera de enseñar el módulo de Programación que mejora la manera en la que se enseña normalmente?”
 - 21 personas respondieron “Sí” (91,3%).
 - 2 personas respondieron “No” (8,7%).
- A la pregunta: “¿Crees que tu rendimiento durante el proyecto ha sido el adecuado?”
 - 18 personas respondieron “Sí” (78,3%).
 - 5 personas respondieron “No” (21,7%).
- A la pregunta: “¿Consideras que vas a sacar buena nota en el proyecto?”
 - 16 personas respondieron “Sí” (69,6%).
 - 7 personas respondieron “No” (30,4%).

A estas preguntas se les adjuntó un campo extra para que los/as alumnos/as escribieran comentarios adicionales, aunque estos no aportaron información significativa a lo ya recogido en las preguntas.

2.8.4 Análisis de los resultados

A lo recogido en la encuesta hay que añadirle las notas finales de los proyectos contando con que eran 9 grupos (uno de 5 personas, otro de 4, 5 de 3, otro de 2 y 1 de una sola persona):

- De los 9 proyectos, solo 2 no llegaron a los requisitos mínimos; suponiendo el suspenso de 3 de las 4 personas de uno de los grupos, ya que una de ellas se esforzó muy por encima de lo que se esforzaron sus compañeros/as de grupo; y el suspenso de las 2 personas del otro grupo.
- Los 7 grupos restantes aprobaron, con 3 de los grupos sacando la máxima nota (el grupo de 5 personas y dos de los grupos de 3).

Es muy significativo que los resultados hayan sido prácticamente inversos a los resultados del segundo trimestre (ya que este proyecto solo contaba dentro de las “tareas de clase” y la mayor parte de la nota se establecía con un examen); 4 suspensos y 23 aprobados (85,2% de aprobados) frente a los 22 y 5 (18,5% de aprobados) bajo la metodología del tutor.

Al comparar las respuestas de la primera y segunda encuesta la tendencia está clara, los alumnos se encuentran mucho más motivados; esto reafirma lo observado durante el transcurso de las clases, donde la mayoría del alumnado mantenía un nivel de trabajo alto e incluso trabajo voluntario en casa.

Tomando estos resultados juntos (las notas y las encuestas), es seguro afirmar que se han alcanzado los objetivos:

- Se ha logrado que emerja una motivación intrínseca en los/as alumnos/as que provoque un nivel de esfuerzo superior de cara a la realización del proyecto.

- Se ha conseguido que el alumnado adquiriera las competencias de una manera amena y eficiente; no se ha observado el tiempo suficiente para saber si va a ser duradera.
- Se ha cambiado la concepción y predisposición el alumnado durante la intervención en el módulo de programación.
- Se fomentó la creatividad, el pensamiento de diseño y el pensamiento computacional; la creatividad incluso por encima de lo esperado, como se comentó previamente.

En conclusión, la innovación ha sido un éxito y está lista para seguir aplicándose y mejorándose de cara al futuro.

2.8.5 Puntos fuertes y puntos débiles de la innovación

El punto más fuerte de la innovación es lo fiable que resulta en el contexto al que va dirigida; resulta muy fácil que el alumnado de los ciclos formativos y secundaria sientan un deseo inherente hacia el diseño de juegos, es algo que los seres humanos hacemos desde pequeños sin darnos cuenta y que estos perfiles retienen en mayor o menor medida. Otro punto fuerte son los pocos recursos necesarios para realizarla, hasta el autor se sorprendió con el buen resultado que ha dado desarrollar videojuegos para la consola de Eclipse (básicamente mostrar texto crudo). Además, bien ejecutada esta propuesta libera de mucho trabajo al docente, pues resulta obvio que cuantos más proyectos realice el alumnado, más autónomo se volverá (aunque esto no se ha podido comprobar fehacientemente).

Un punto débil es precisamente lo concreta que es. Se pueden encontrar puntos en los que volverla pluriparadigmática, colaborar con otras asignaturas/módulos y emplearla en situaciones de aprendizaje con la filosofía del aprendizaje servicio; pero ésta solo puede ser ejecutada por alumnado del perfil informático. Otro punto débil es que un profesor que no tenga formación en diseño y desarrollo de videojuegos o metodologías ágiles, puede fallar a la hora de ejecutar la innovación; creando así las horas de trabajo en clase en balde que tanto parecen caracterizar el panorama actual.

Sin embargo, el punto más débil de esta innovación es poder secuenciarla en el trimestre de manera adecuada; si se sitúa al final del trimestre (la opción que se tomó para ejecutar esta innovación), el proyecto coincide con el periodo de exámenes de otros módulos; si se sitúa al principio, habría que obviar el primer trimestre y refrescar los contenidos vistos en el trimestre anterior; si se sitúa a la mitad, corta el ritmo de clase y no se afianzan correctamente los conceptos de cara al proyecto. Ninguna opción es perfecta, pero sin duda ubicar el proyecto al final del trimestre es la opción más lógica y que mejor se adapta a la estructura actual de la innovación.

2.8.6 Propuestas para mejorar la innovación

Como propuesta de mejora, es importante encontrar un modo de trabajo, programa o técnica que permita a un docente no formado en el diseño y desarrollo de videojuegos y/o metodologías ágiles aplicar esta innovación de manera eficaz. Incluso este documento probablemente no sea material suficiente sobre el que trabajar, ya que los problemas que surgen al aplicarla son muy específicos de dicha disciplina y metodologías.

2.9 Reflexión personal sobre el proceso de innovación

Como ya se comentó al principio, la innovación para el autor ha sido algo accidental e inconsciente durante mucho tiempo; el hecho de haber realizado un esfuerzo consciente por investigar, preparar los materiales, recoger información, preocuparse por aplicar una estrategia adaptada a un grupo clase y observar los efectos de lo preparado a lo largo del tiempo ha sido una experiencia muy enriquecedora.

3 Propuesta de programación docente

3.1 Introducción

La programación docente es un documento elaborado por los/las educadores/as que establece los objetivos, contenidos, metodología, evaluación y recursos necesarios para llevar a cabo el proceso de enseñanza-aprendizaje en un determinado contexto educativo. Se trata de un instrumento fundamental que guía la labor del docente y proporciona una estructura organizada para el desarrollo de las actividades educativas. Según lo establecido en la **Ley Orgánica 2/2006, de 3 de mayo, de Educación**, la programación docente es un componente esencial en el ámbito educativo, ya que garantiza la coherencia y calidad de la enseñanza. Permite planificar y secuenciar los contenidos de acuerdo con los objetivos educativos establecidos, adaptar la metodología y los recursos a las características y necesidades de los estudiantes, y evaluar de manera adecuada los logros alcanzados. Además, la programación docente asegura la equidad y la igualdad de oportunidades educativas al establecer criterios claros y transparentes para la planificación y desarrollo de las actividades. En resumen, la programación docente desempeña un papel crucial en el proceso enseñanza-aprendizaje al proporcionar una guía clara y coherente para los/as educadores/as y al garantizar la calidad y equidad en la educación.

La programación docente que aquí se presenta se sitúa en el Ciclo Formativo de Grado Superior de Desarrollo de Aplicaciones Web en su módulo de Programación, con las siguientes características:

- Ciclo Formativo de Grado Superior de Desarrollo de Aplicaciones Web:
 - Código: IFC-303LOE.
 - Clasificación Internacional Normalizada de la Educación:
 - Clasificación: CINE-5b.
 - Familia profesional: Informática y Comunicaciones.
 - Duración: 2000 horas.
 - Título: Técnico Superior en Desarrollo de Aplicaciones Web.
- Módulo de Programación:
 - Código: 0485.

- Duración: 224 horas (7 horas semanales).
- Modalidad y Régimen: Presencial.
- Créditos ECTS: 14.
- Secuenciación: Primer curso.
- Temporalización: Anual.

3.2 Contexto

3.2.1 *Legislación*

El **Real Decreto 686/2010**, de 20 de mayo, por el que se establece el título de **Técnico Superior en Desarrollo de Aplicaciones Web** y se fijan sus enseñanzas mínimas, establece las dependencias legislativas a las que el ciclo está sujeto; citando textualmente:

La Ley Orgánica 2/2006, de 3 de mayo, de Educación, dispone en el artículo 39.6 que el Gobierno, previa consulta a las Comunidades Autónomas, establecerá las titulaciones correspondientes a los estudios de formación profesional, así como los aspectos básicos del currículo de cada una de ellas.

La Ley Orgánica 5/2002, de 19 de junio, de las Cualificaciones y de la Formación Profesional, establece en el artículo 10.1 que la Administración General del Estado, de conformidad con lo dispuesto en el artículo 149.1.30.^a y 7.^a de la Constitución y previa consulta al Consejo General de la Formación Profesional, determinará los títulos y los certificados de profesionalidad, que constituirán las ofertas de formación profesional referidas al Catálogo Nacional de Cualificaciones Profesionales.

El Real Decreto 1538/2006, de 15 de diciembre, ha establecido la ordenación general de la formación profesional del sistema educativo, y define en el artículo 6 la estructura de los títulos de formación profesional tomando como base el Catálogo Nacional de Cualificaciones Profesionales, las directrices fijadas por la Unión Europea y otros aspectos de interés social.

Por otra parte, del mismo modo, concreta en el artículo 7 el perfil profesional de dichos títulos, que incluirá la competencia general, las competencias profesionales, personales y sociales, las cualificaciones y, en su caso, las unidades de competencia del Catálogo Nacional de Cualificaciones Profesionales incluidas en los títulos, de modo que cada título incorporará, al menos, una cualificación profesional completa, con el fin de lograr que, en efecto, los títulos de formación profesional respondan a las necesidades demandadas por el sistema productivo y a los valores personales y sociales para ejercer una ciudadanía democrática.

Este marco normativo hace necesario que ahora el Gobierno, previa consulta a las Comunidades Autónomas, establezca cada uno de los títulos que formarán el Catálogo de títulos de la formación profesional del sistema educativo, sus enseñanzas mínimas y aquellos otros aspectos de la ordenación académica que, sin perjuicio de las competencias atribuidas a las Administraciones educativas en esta materia, constituyan los aspectos básicos del currículo que aseguren una formación común y garanticen la validez de los títulos, en cumplimiento con lo dispuesto en el artículo 6.2 de la Ley Orgánica 2/2006, de 3 de mayo, de Educación.

A estos efectos, procede determinar para cada título su identificación, su perfil profesional, el entorno profesional, la prospectiva del título en el sector o sectores, las enseñanzas del ciclo formativo, la correspondencia de los módulos profesionales con las unidades de competencia para su acreditación, convalidación o exención y los parámetros básicos de contexto formativo para cada módulo profesional (espacios, equipamientos necesarios, las titulaciones y especialidades del profesorado y sus equivalencias a efectos de docencia), previa consulta a las Comunidades Autónomas, según lo previsto en el artículo 95 de la Ley Orgánica 2/2006, de 3 de mayo, de Educación.

Asimismo, en cada título también se determinarán los accesos a otros estudios y, en su caso, las modalidades y materias de bachillerato que facilitan la conexión con el ciclo formativo de grado superior, las convalidaciones, exenciones y equivalencias y la información sobre los requisitos necesarios según la legislación vigente para el ejercicio profesional cuando proceda.

Con el fin de facilitar el reconocimiento de créditos entre los títulos de técnico superior y las enseñanzas conducentes a títulos universitarios y viceversa, en los ciclos formativos de grado superior se establecerá la equivalencia de cada módulo profesional con créditos europeos, ECTS, tal y como se definen en el Real Decreto 1125/2003, de 5 de septiembre, por el que se establece el sistema europeo de créditos y el sistema de calificaciones en las titulaciones universitarias de carácter oficial y validez en todo el territorio nacional.

Así, el presente real decreto conforme a lo previsto en el Real Decreto 1538/2006, de 15 de diciembre, establece y regula, en los aspectos y elementos básicos antes indicados, el título de formación profesional del sistema educativo de Técnico Superior en Desarrollo de Aplicaciones Web.

Se ha recurrido a una norma reglamentaria para establecer bases estatales conforme con el Tribunal Constitucional, que admite «excepcionalmente», las bases puedan establecerse mediante normas reglamentarias en determinados supuestos como ocurre en el presente caso, cuando «resulta complemento indispensable para asegurar el mínimo común denominador establecido en las normas legales básicas» (así, entre otras, en las SSTC 25/1983, 32/1983, y 48/1988). (p. 1-2)

Por otro lado, Orden **EDU/2887/2010**, de 2 de noviembre, por la que se establece el currículo del **ciclo formativo de Grado Superior** correspondiente al título de **Técnico Superior en Desarrollo de Aplicaciones Web**, concreta los ítems curriculares generales pertinentes a cada módulo; enmarcando así los grandes bloques de contenido a desarrollar.

Finalmente, a nivel regional, el **Decreto 184/2012**, de 8 de agosto, por el que se establece el currículo del ciclo formativo de Grado Superior de Formación Profesional en Desarrollo de Aplicaciones Web en el **Principado de Asturias**, concentra lo establecido en documentos legislativos de orden superior y desglosa todos los ítems curriculares de los módulos profesionales pertinentes; estableciendo así los resultados de aprendizaje, criterios de evaluación y, de nuevo, contenidos de los mismos.

Cabe destacar la relevancia de los siguientes documentos los cuales, sin ser centrales al ciclo o módulo a tratar, tienen cierta influencia en el desarrollo natural de los mismos en las aulas o en el estado actual de la educación en España:

- Real Decreto 275/2007, de 23 de febrero, por el que se crea el Observatorio Estatal de la Convivencia Escolar, modificado por el Real Decreto 3/2018, de 12 de enero.
- Real Decreto 694/2007, de 1 de junio, por el que se regula el Consejo Escolar del Estado.
- Ley 4/2019, de 7 de marzo, de mejora de las condiciones para el desempeño de la docencia y la enseñanza en el ámbito de la educación no universitaria.
- Real Decreto 984/2021, de 16 de noviembre, por el que se regulan la evaluación y la promoción en la Educación Primaria, así como la evaluación, la promoción y la titulación en la Educación Secundaria Obligatoria, el Bachillerato y la Formación Profesional.
- Ley Orgánica 5/2002, de 19 de junio, de las Cualificaciones y de la Formación Profesional.
- Ley 2/2011, de 4 de marzo, de Economía Sostenible.
- Real Decreto 1128/2003, de 5 de septiembre, por el que se regula el Catálogo Nacional de Cualificaciones Profesionales. Modificado posteriormente mediante el Real Decreto 1416/2005, de 25 de noviembre.
- Ley Orgánica 4/2011, de 11 de marzo, complementaria de la Ley de Economía Sostenible, por la que se modifican las Leyes Orgánicas 5/2002, de 19 de junio,

de las Cualificaciones y de la Formación Profesional, 2/2006, de 3 de mayo, de Educación, y 6/1985, de 1 de julio, del Poder Judicial.

- Ley Orgánica 3/2022, de 31 de marzo, de ordenación e integración de la Formación Profesional.
- Real Decreto 1529/2012, de 8 de noviembre, por el que se desarrolla el contrato para la formación y el aprendizaje y se establecen las bases de la formación profesional dual.
- Orden ESS/2518/2013, de 26 de diciembre, por la que se regulan los aspectos formativos del contrato para la formación y el aprendizaje, en desarrollo del Real Decreto 1529/2012, de 8 de noviembre, por el que se desarrolla el contrato para la formación y el aprendizaje y se establecen las bases de la formación profesional dual.
- Real Decreto 1224/2009, de 17 de julio, de reconocimiento de las competencias profesionales adquiridas por experiencia laboral.
- Orden PRE/910/2011, de 12 de abril, por la que se crea la Comisión Interministerial para el seguimiento y evaluación del procedimiento de reconocimiento de las competencias profesionales adquiridas por experiencia laboral.
- Real Decreto 581/2017, de 9 de junio, por el que se incorpora al ordenamiento jurídico español la Directiva 2013/55/UE del Parlamento Europeo y del Consejo, de 20 de noviembre de 2013, por la que se modifica la Directiva 2005/36/CE relativa al reconocimiento de cualificaciones profesionales y el Reglamento (UE) n.º 1024/2012 relativo a la cooperación administrativa a través del Sistema de Información del Mercado Interior (Reglamento IMI).
- Real Decreto 472/2021, de 29 de junio, por el que se incorpora al ordenamiento jurídico español la Directiva (UE) 2018/958, del Parlamento Europeo y del Consejo, de 28 de junio de 2018, relativa al test de proporcionalidad antes de adoptar nuevas regulaciones de profesiones.
- Orden EFP/1010/2021, de 20 de septiembre, por la que se establece el procedimiento permanente para la evaluación y acreditación de competencias profesionales, adquiridas por la experiencia laboral o vías no formales de

formación, en las Ciudades de Ceuta y Melilla, y se formalizan las bases para su desarrollo.

- Real Decreto 1085/2020, de 9 de diciembre, por el que se establecen convalidaciones de módulos profesionales de los títulos de Formación Profesional del sistema educativo español y las medidas para su aplicación, y se modifica el Real Decreto 1147/2011, de 29 de julio, por el que se establece la ordenación general de la formación profesional del sistema educativo.
- Orden EFP/1210/2021, de 2 de noviembre, por la que se establece la equivalencia, a los efectos de acceso a enseñanzas de formación profesional, de determinados estudios y títulos anteriores al actual sistema educativo.
- Real Decreto 1558/2005, de 23 de diciembre, por el que se regulan los requisitos básicos de los Centros integrados de formación profesional.
- Real Decreto 229/2008, de 15 de febrero, por el que se regulan los Centros de Referencia Nacional en el ámbito de la formación profesional.
- Resolución de 1 de diciembre de 2021, de la Consejería de Educación, por la que se aprueban instrucciones sobre la evaluación y la promoción en la Educación Primaria, así como la evaluación, la promoción y la titulación en la Educación Secundaria Obligatoria, el Bachillerato y la Formación Profesional.
- Resolución de 5 de abril 2022, de la Consejería de Educación, por la que se aprueba el Calendario Escolar para el curso 2022-2023 y las instrucciones necesarias para su aplicación.

Otro documento relevante, pero aún no implementado formalmente en el sistema educativo, es la nueva **Ley Orgánica 3/2022, de 31 de marzo, de ordenación e integración de la Formación Profesional**. Esta ley cambia el panorama de la formación profesional, enfocándola a un modelo centrado en la importancia de la formación en las empresas, la colaboración con las mismas y complementación de los aprendizajes obtenidos en el entorno laboral. Cabe la duda de si será posible su aplicación en el futuro (planeada para el curso académico 2023/2024); la cual, como mínimo, será dificultosa; de momento, el documento de referencia sigue siendo la Ley Orgánica 5/2002, de 19 de junio, de las Cualificaciones y de la Formación Profesional.

3.2.2 *Centro*

El centro para el que esta programación está pensada es un IES del entorno urbano de una ciudad mediana. Este centro tiene particularidades organizativas que dificultan la labor docente; los espacios están distribuidos de manera disgregada, pero alojando a una gran cantidad de alumnos y alumnas (más de 1300) y docentes (más de 130). A pesar de su ubicación, el alumnado es heterogéneo, tanto en sus características y expectativas personales como en sus ámbitos de procedencia y entornos.

Por suerte, las aulas donde se desarrolla el Ciclo Formativo de Grado Superior de Desarrollo de Aplicaciones Web están localizadas en el mismo pasillo y gozan de cierto aislamiento e independencia dentro del centro. Las aulas son suficientes para el desarrollo del ciclo, incluyendo recursos como:

- Equipos informáticos (ordenadores).
- Proyector.
- Espacio amplio.

Pese a que el departamento de Informática y Comunicaciones es pequeño en comparación con la mayoría del resto de departamentos, sus 15 integrantes son incapaces de coordinarse lo más mínimo.

3.2.3 *Grupo clase*

El grupo clase a tratar tiene ciertas particularidades, viene de dos años de irregularidades provocadas por la pandemia de 2020 y esto se evidencia en que el alumnado del mismo cuenta con un nivel de partida muy heterogéneo, que acentúa las diferencias existentes entre aquellos que vienen de Bachillerato, Grados Medios u otros estudios. El rendimiento durante las clases es muy bajo y aún menor en casa; notándose un clima de desmotivación y dejadez generalizado.

3.3 Desglose legislativo

A continuación, se tratarán los elementos recogidos en la legislación que tienen importancia dentro de la programación docente que aquí se desarrolla.

3.3.1 *Competencia general*

Según el Real Decreto 686/2010, de 20 de mayo, por el que se establece el título de Técnico Superior en Desarrollo de Aplicaciones Web y se fijan sus enseñanzas mínimas:

- La competencia general de este título consiste en desarrollar, implantar, y mantener aplicaciones web, con independencia del modelo empleado y utilizando tecnologías específicas, garantizando el acceso a los datos de forma segura y cumpliendo los criterios de accesibilidad, usabilidad y calidad exigidas en los estándares establecidos.

Esta competencia general se complementa en el Decreto 184/2012, de 8 de agosto, por el que se establece el currículo del ciclo formativo de Grado Superior de Formación Profesional en Desarrollo de Aplicaciones Web en el Principado de Asturias con los siguientes objetivos generales:

- a) Conocer el sector informático de Asturias.
- b) Aplicar la lengua extranjera para el uso profesional.

3.3.2 *Competencias profesionales, personales y sociales*

Según el Real Decreto 686/2010, de 20 de mayo, por el que se establece el título de Técnico Superior en Desarrollo de Aplicaciones Web y se fijan sus enseñanzas mínimas, las competencias profesionales, personales y sociales de este título son las 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.

- b) Aplicar técnicas y procedimientos relacionados con la seguridad en sistemas, servicios y aplicaciones, cumpliendo el plan de seguridad.
- c) Gestionar servidores de aplicaciones adaptando su configuración en cada caso para permitir el despliegue de aplicaciones web.
- d) Gestionar bases de datos, interpretando su diseño lógico y verificando integridad, consistencia, seguridad y accesibilidad de los datos.
- e) Desarrollar aplicaciones web con acceso a bases de datos utilizando lenguajes, objetos de acceso y herramientas de mapeo adecuados a las especificaciones.
- f) Integrar contenidos en la lógica de una aplicación web, desarrollando componentes de acceso a datos adecuados a las especificaciones.
- g) Desarrollar interfaces en aplicaciones web de acuerdo con un manual de estilo, utilizando lenguajes de marcas y estándares web.
- h) Desarrollar componentes multimedia para su integración en aplicaciones web, empleando herramientas específicas y siguiendo las especificaciones establecidas.
- i) Integrar componentes multimedia en el interface de una aplicación web, realizando el análisis de interactividad, accesibilidad y usabilidad de la aplicación.
- j) Desarrollar e integrar componentes software en el entorno del servidor web, empleando herramientas y lenguajes específicos, para cumplir las especificaciones de la aplicación.
- k) Desarrollar servicios para integrar sus funciones en otras aplicaciones web, asegurando su funcionalidad.
- l) Integrar servicios y contenidos distribuidos en aplicaciones web, asegurando su funcionalidad.
- m) Completar planes de pruebas verificando el funcionamiento de los componentes software desarrollados, según las especificaciones.
- n) Elaborar y mantener la documentación de los procesos de desarrollo, utilizando herramientas de generación de documentación y control de versiones.
- ñ) Desplegar y distribuir aplicaciones web en distintos ámbitos de implantación, verificando su comportamiento y realizando modificaciones.
- o) Gestionar y/o realizar el mantenimiento de los recursos de su área en función de las cargas de trabajo y el plan de mantenimiento.

- p) Adaptarse a las nuevas situaciones laborales, manteniendo actualizados los conocimientos científicos, técnicos y tecnológicos relativos a su entorno profesional, gestionando su formación y los recursos existentes en el aprendizaje a lo largo de la vida y utilizando las tecnologías de la información y la comunicación.
- q) Resolver situaciones, problemas o contingencias con iniciativa y autonomía en el ámbito de su competencia, con creatividad, innovación y espíritu de mejora en el trabajo personal y en el de los miembros del equipo.
- r) Organizar y coordinar equipos de trabajo, supervisando el desarrollo del mismo, con responsabilidad, manteniendo relaciones fluidas y asumiendo el liderazgo, así como, aportando soluciones a los conflictos grupales que se presentan.
- s) Comunicarse con sus iguales, superiores, clientes y personas bajo su responsabilidad utilizando vías eficaces de comunicación, transmitiendo la información o conocimientos adecuados, y respetando la autonomía y competencia de las personas que intervienen en el ámbito de su trabajo.
- t) Generar entornos seguros en el desarrollo de su trabajo y el de su equipo, supervisando y aplicando los procedimientos de prevención de riesgos laborales y ambientales de acuerdo con lo establecido por la normativa y los objetivos de la empresa.
- u) Supervisar y aplicar procedimientos de gestión de calidad, de accesibilidad universal y de diseño para todos, en las actividades profesionales incluidas en los procesos de producción o prestación de servicios.
- v) Realizar la gestión básica para la creación y funcionamiento de una pequeña empresa y tener iniciativa en su actividad profesional con sentido de la responsabilidad social.
- x) Ejercer sus derechos y cumplir con las obligaciones derivadas de su actividad profesional, de acuerdo con lo establecido en la legislación vigente, participando activamente en la vida económica, social y cultural.

3.3.3 *Relación de cualificaciones y unidades de competencia del Catálogo Nacional de Cualificaciones Profesionales incluidas en el título*

- Cualificaciones profesionales completas:
 - Desarrollo de aplicaciones con tecnologías Web IFC154_3 (Real Decreto 1087/2005, de 16 de septiembre), que comprende las siguientes unidades de competencia:
 - UC0491_3 Desarrollar elementos software en el entorno cliente.
 - UC0492_3 Desarrollar elementos software en el entorno servidor.
 - UC0493_3 Implementar, verificar y documentar aplicaciones web en entornos internet, intranet y extranet.
- Cualificaciones profesionales incompletas:
 - Programación en lenguajes estructurados de aplicaciones de gestión IFC155_3 (Real Decreto 1087/2005, de 16 de septiembre), que comprende las siguientes unidades de competencia:
 - UC0223_3 Configurar y explotar sistemas informáticos.
 - UC0226_3 Programar bases de datos relacionales.
 - Programación con lenguajes orientados a objetos y bases de datos relacionales IFC 080_3 (Real Decreto. 295/2004, de 20 de febrero), que comprende las siguientes unidades de competencia:
 - UC0223_3 Configurar y explotar sistemas informáticos.
 - UC0226_3 Programar bases de datos relacionales.

3.3.4 *Objetivos generales*

Según el Real Decreto 686/2010, de 20 de mayo, por el que se establece el título de Técnico Superior en Desarrollo de Aplicaciones Web y se fijan sus enseñanzas mínimas, los **objetivos generales del título** son los siguientes:

- a) Ajustar la configuración lógica analizando las necesidades y criterios establecidos para configurar y explotar sistemas informáticos.
- b) Identificar las necesidades de seguridad verificando el plan preestablecido para aplicar técnicas y procedimientos relacionados.

- c) Instalar módulos analizando su estructura y funcionalidad para gestionar servidores de aplicaciones.
- d) Ajustar parámetros analizando la configuración para gestionar servidores de aplicaciones.
- e) Interpretar el diseño lógico, verificando los parámetros establecidos para gestionar bases de datos.
- f) Seleccionar lenguajes, objetos y herramientas, interpretando las especificaciones para desarrollar aplicaciones web con acceso a bases de datos.
- g) Utilizar lenguajes, objetos y herramientas, interpretando las especificaciones para desarrollar aplicaciones web con acceso a bases de datos.
- h) Generar componentes de acceso a datos, cumpliendo las especificaciones, para integrar contenidos en la lógica de una aplicación web.
- i) Utilizar lenguajes de marcas y estándares web, asumiendo el manual de estilo, para desarrollar interfaces en aplicaciones web
- j) Emplear herramientas y lenguajes específicos, siguiendo las especificaciones, para desarrollar componentes multimedia.
- k) Evaluar la interactividad, accesibilidad y usabilidad de un interfaz, verificando los criterios preestablecidos, para Integrar componentes multimedia en el interfaz de una aplicación.
- l) Utilizar herramientas y lenguajes específicos, cumpliendo las especificaciones, para desarrollar e integrar componentes software en el entorno del servidor web.
- m) Emplear herramientas específicas, integrando la funcionalidad entre aplicaciones, para desarrollar servicios empleables en aplicaciones web.
- n) Evaluar servicios distribuidos ya desarrollados, verificando sus prestaciones y funcionalidad, para integrar servicios distribuidos en una aplicación web.
- ñ) Verificar los componentes de software desarrollados, analizando las especificaciones, para completar el plan de pruebas.
- o) Utilizar herramientas específicas, cumpliendo los estándares establecidos, para elaborar y mantener la documentación de los procesos.
- p) Establecer procedimientos, verificando su funcionalidad, para desplegar y distribuir aplicaciones.

- q) Programar y realizar actividades para gestionar el mantenimiento de los recursos informáticos.
- r) Analizar y utilizar los recursos y oportunidades de aprendizaje relacionadas con la evolución científica, tecnológica y organizativa del sector y las tecnologías de la información y la comunicación, para mantener el espíritu de actualización y adaptarse a nuevas situaciones laborales y personales.
- s) Desarrollar la creatividad y el espíritu de innovación para responder a los retos que se presentan en los procesos y organización de trabajo y de la vida personal.
- t) Tomar decisiones de forma fundamentada analizando las variables implicadas, integrando saberes de distinto ámbito y aceptando los riesgos y la posibilidad de equivocación en las mismas, para afrontar y resolver distintas situaciones, problemas o contingencias.
- u) Desarrollar técnicas de liderazgo, motivación, supervisión y comunicación en contextos de trabajo en grupo para facilitar la organización y coordinación de equipos de trabajo.
- v) Aplicar estrategias y técnicas de comunicación adaptándose a los contenidos que se van a transmitir, la finalidad y a las características de los receptores, para asegurar la eficacia en los procesos de comunicación.
- x) Evaluar situaciones de prevención de riesgos laborales y de protección ambiental, proponiendo y aplicando medidas de prevención personales y colectivas, de acuerdo a la normativa aplicable en los procesos del trabajo, para garantizar entornos seguros.
- y) Identificar y proponer las acciones profesionales necesarias para dar respuesta a la accesibilidad universal y al diseño para todos.
- z) Identificar y aplicar parámetros de calidad en los trabajos y actividades realizados en el proceso de aprendizaje para valorar la cultura de la evaluación y de la calidad y ser capaces de supervisar y mejorar procedimientos de gestión de calidad.
- aa) Utilizar procedimientos relacionados con la cultura emprendedora, empresarial y de iniciativa profesional, para realizar la gestión básica de una pequeña empresa o emprender un trabajo.

- ab) Reconocer sus derechos y deberes como agente activo en la sociedad, teniendo en cuenta el marco legal que regula las condiciones sociales y laborales para participar como ciudadano democrático.

De entre esos objetivos generales, **el Módulo de Programación contribuye a alcanzar los siguientes:**

- e) Interpretar el diseño lógico, verificando los parámetros establecidos para gestionar bases de datos.
- j) Emplear herramientas y lenguajes específicos, siguiendo las especificaciones, para desarrollar componentes multimedia.
- ñ) Verificar los componentes de software desarrollados, analizando las especificaciones, para completar el plan de pruebas.
- o) Utilizar herramientas específicas, cumpliendo los estándares establecidos, para elaborar y mantener la documentación de los procesos.
- p) Establecer procedimientos, verificando su funcionalidad, para desplegar y distribuir aplicaciones.
- q) Programar y realizar actividades para gestionar el mantenimiento de los recursos informáticos.
- r) Analizar y utilizar los recursos y oportunidades de aprendizaje relacionadas con la evolución científica, tecnológica y organizativa del sector y las tecnologías de la información y la comunicación, para mantener el espíritu de actualización y adaptarse a nuevas situaciones laborales y personales.

3.3.5 *Contenidos*

Según el **Real Decreto 686/2010**, de 20 de mayo, por el que se establece el título de Técnico Superior en Desarrollo de Aplicaciones Web y se fijan sus enseñanzas mínimas, los **contenidos a desarrollar en el Módulo** de Programación son los siguientes:

1. Identificación de los elementos de un programa informático:
 - Estructura y bloques fundamentales.

- Variables.
 - Tipos de datos.
 - Literales.
 - Constantes.
 - Operadores y expresiones.
 - Conversiones de tipo.
 - Comentarios.
2. Utilización de objetos:
- Características de los objetos.
 - «Instanciación» de objetos.
 - Utilización de métodos.
 - Utilización de propiedades.
 - Utilización de métodos estáticos.
 - Constructores.
 - Destrucción de objetos y liberación de memoria.
3. Uso de estructuras de control:
- Estructuras de selección.
 - Estructuras de repetición.
 - Estructuras de salto.
 - Control de excepciones.
4. Desarrollo de clases:
- Concepto de clase.
 - Estructura y miembros de una clase.
 - Creación de atributos.
 - Creación de métodos.
 - Creación de constructores.
 - Utilización de clases y objetos.
 - Utilización de clases heredadas.
5. Lectura y escritura de información:
- Tipos de flujos. Flujos de bytes y de caracteres.
 - Clases relativas a flujos.
 - Utilización de flujos.

- Entrada desde teclado.
 - Salida a pantalla.
 - Ficheros de datos. Registros.
 - Apertura y cierre de ficheros. Modos de acceso.
 - Escritura y lectura de información en ficheros.
 - Utilización de los sistemas de ficheros.
 - Creación y eliminación de ficheros y directorios.
 - Interfaces.
 - Concepto de evento.
 - Creación de controladores de eventos.
6. Aplicación de las estructuras de almacenamiento:
- Estructuras.
 - Creación de arrays.
 - Arrays multidimensionales.
 - Cadenas de caracteres.
 - Listas.
7. Utilización avanzada de clases:
- Composición de clases.
 - Herencia.
 - Superclases y subclases.
 - Clases y métodos abstractos y finales.
 - Sobreescritura de métodos.
 - Constructores y herencia.
8. Mantenimiento de la persistencia de los objetos:
- Bases de datos orientadas a objetos.
 - Características de las bases de datos orientadas a objetos.
 - Instalación del gestor de bases de datos.
 - Creación de bases de datos.
 - Mecanismos de consulta.
 - El lenguaje de consultas: sintaxis, expresiones y operadores.
 - Recuperación, modificación y borrado de información.
 - Tipos de datos objeto; atributos y métodos.

- Tipos de datos colección.
9. Gestión de bases de datos relacionales:
- Establecimiento de conexiones.
 - Recuperación de información.
 - Manipulación de la información.
 - Ejecución de consultas sobre la base de datos

El **Decreto 184/2012**, de 8 de agosto, por el que se establece el currículo del ciclo formativo de Grado Superior de Formación Profesional en Desarrollo de Aplicaciones Web en el Principado de Asturias, también recoge los contenidos a desarrollar en el Módulo de Programación; pero estos son los mismos que los del Real Decreto.

3.3.6 Resultados de aprendizaje y criterios de evaluación

En el Real Decreto 686/2010, de 20 de mayo, por el que se establece el título de Técnico Superior en Desarrollo de Aplicaciones Web y se fijan sus enseñanzas mínimas, se recogen los **resultados de aprendizaje y sus criterios de evaluación de desempeño asociados al módulo** como se exponen a continuación:

Resultados de aprendizaje	Criterios de evaluación
RA1. Reconoce la estructura de un programa informático, identificando y relacionando los elementos propios del lenguaje de programación utilizado.	<ul style="list-style-type: none"> 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.

	<ul style="list-style-type: none"> h) Se ha comprobado el funcionamiento de las conversiones de tipo explícitas e implícitas. i) Se han introducido comentarios en el código.
<p>RA2. Escribe y prueba programas sencillos, reconociendo y aplicando los fundamentos de la programación orientada a objetos.</p>	<ul style="list-style-type: none"> 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.
<p>RA3. Escribe y depura código, analizando y utilizando las estructuras de control del lenguaje.</p>	<ul style="list-style-type: none"> a) Se ha escrito y probado código que haga uso de estructuras de selección. b) Se han utilizado estructuras de repetición. c) Se han reconocido las posibilidades de las sentencias de salto. d) Se ha escrito código utilizando control de excepciones. e) Se han creado programas ejecutables utilizando diferentes estructuras de control. f) Se han probado y depurado los programas. g) Se ha comentado y documentado el código.
<p>RA4. Desarrolla programas organizados en clases analizando y aplicando los principios de la programación orientada a objetos.</p>	<ul style="list-style-type: none"> a) Se ha reconocido la sintaxis, estructura y componentes típicos de una clase. b) Se han definido clases. c) Se han definido propiedades y métodos.

	<ul style="list-style-type: none"> d) Se han utilizado constructores y destructores. e) Se han desarrollado programas que instancien y utilicen objetos de las clases creadas anteriormente. f) Se han utilizado mecanismos para controlar la visibilidad de las clases y de sus miembros. g) Se han definido y utilizado clases heredadas. h) Se han creado y utilizado métodos estáticos. i) Se han definido y utilizado interfaces. j) Se han creado y utilizado conjuntos y librerías de clases.
<p>RA5. Realiza operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases.</p>	<ul style="list-style-type: none"> 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 implementado técnicas para envío y almacenamiento de objetos en ficheros. g) Se han utilizado las herramientas del entorno de desarrollo para crear interfaces gráficos de usuario simples. h) Se han programado controladores de eventos. i) Se han escrito programas que utilicen interfaces gráficos para la entrada y salida de información.
<p>RA6. Escribe programas que manipulen información, seleccionando y utilizando tipos avanzados de datos.</p>	<ul style="list-style-type: none"> a) Se han escrito programas que utilicen arrays.

	<ul style="list-style-type: none"> 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. h) Se han identificado las clases relacionadas con el tratamiento de documentos XML. i) Se han realizado programas que realicen manipulaciones sobre documentos XML.
<p>RA7. Desarrolla programas, aplicando características avanzadas de los lenguajes orientados a objetos y del entorno de programación.</p>	<ul style="list-style-type: none"> 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 sobrescriban 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.
<p>RA8. Utiliza bases de datos orientadas a objetos, analizando sus características y aplicando técnicas para mantener la persistencia de la información.</p>	<ul style="list-style-type: none"> 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

	<p>mediante lenguajes orientados a objetos.</p> <ul style="list-style-type: none"> 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.
<p>RA9. Gestiona información almacenada en bases de datos relacionales manteniendo la integridad y la consistencia de los datos.</p>	<ul style="list-style-type: none"> 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 utilizado diferentes asistentes del entorno de desarrollo. e) Se han identificado las características y métodos de acceso a sistemas gestores de bases de datos relacionales. f) Se han programado conexiones con bases de datos. g) Se ha escrito código para almacenar información en bases de datos. h) Se han utilizado diferentes asistentes del entorno de desarrollo.

El Decreto 184/2012, de 8 de agosto, por el que se establece el currículo del ciclo formativo de Grado Superior de Formación Profesional en Desarrollo de Aplicaciones Web en el Principado de Asturias, también recoge los resultados de aprendizaje y sus criterios de evaluación de desempeño asociados del Módulo de Programación; pero estos son los mismos que los del Real Decreto.

3.4 Temporalización

Teniendo en cuenta los elementos curriculares recogidos en el apartado anterior y contando con el contexto definido anteriormente, se establecen una serie de Unidades de Trabajo de acuerdo con los resultados de aprendizaje fruto de la mismas. Cada Unidad de Trabajo tiene asignado un número de horas determinado hasta cubrirse las 224 horas (7 horas semanales) establecidos para el Módulo de Programación, pero, dentro de ese límite, las horas de cada Unidad de Trabajo pueden variar durante el transcurso del curso.

Trimestre			Horas	Unidades de Trabajo (UT)	Resultados de Aprendizaje									
1º	2º	3º			1	2	3	4	5	6	7	8	9	
			17	UT1 – Conceptos básicos de programación.										
			37	UT2 – Estructura de código para programas de consola.										
			30	UT3 – Introducción a la Programación Orientada a Objetos (POO).										
			35	UT4 – Estructuras de almacenamiento.										
			50	UT5 – Metodología avanzada de la Programación Orientada a Objetos (POO).										
			25	UT6 – Introducción a bases de datos.										
			30	UT7 – Uso avanzado de bases de datos.										

El curso comienza con una aproximación a los conceptos básicos de la programación y a lo largo de los dos primeros trimestres se va construyendo sobre esos conocimientos para dotar al alumnado de las herramientas y pensamiento computacional necesario para defenderse en el resto de asignaturas. En el último trimestre se tratan las

bases de datos y su uso de una forma más aislada, orientando el aprendizaje de cara a poder desenvolverse en las asignaturas de segundo curso que requieren de bases de datos.

La asignación concreta de horas se ha establecido por las siguientes razones, de acuerdo con la cantidad de contenidos y dificultad de los mismos:

- La UT1 es la más teórica y asienta las bases para el resto del curso, es la unidad con menos horas pues los contenidos son sencillos y se reitera sobre los mismos en el resto de Unidades de Trabajo.
- Las UT2, UT3 y UT4 tienen una asignación de horas similar, pues los conceptos requieren de un mayor detenimiento y el tiempo necesario para que estos se asienten.
- La UT5 es la unidad con el mayor número de horas asignadas por su complejidad y por tratarse de la culminación de todo lo visto en las Unidades de Trabajo anteriores. El tiempo dedicado podría ser mayor, pero las Unidades de Trabajo restantes requieren que éste se limite.
- Finalmente, las UT6 y UT7 son un tercio completamente diferente y que no tiene mucho sentido dentro del módulo (como ya se comentó en la reflexión sobre el currículo en el apartado de “[Análisis del currículo oficial](#)”); pese a ello, la complejidad de los contenidos que se tratan en esas Unidades de Trabajo, y al ser conceptos totalmente nuevos para el alumnado, es inevitable dedicarle un número de horas considerable. Las horas establecidas se consideran que son las mínimas para desarrollar esas Unidades de Trabajo de manera satisfactoria y, evidentemente, con ello se limita el número de horas que se pueden dedicar a las otras Unidades de Trabajo (mucho más nucleares al módulo y troncales al ciclo).

En el apartado de “[Superposición de trimestres](#)” se explica la superposición de dos trimestres en una misma Unidad de Trabajo, de acuerdo a la planificación concreta de tareas.

3.5 Unidades de Trabajo

A continuación, se desglosan todas las Unidades de Trabajo a un nivel de abstracción menor; un resumen de todo lo que cada Unidad de Trabajo trata de conseguir y la relación con los contenidos a desarrollar. Las sesiones se reparten semanalmente en 3 sesiones de 2 horas y 1 de 1 hora, con lo que algunas unidades terminarán en la misma sesión en la que empiece otra.

Unidad de Trabajo 1 – Conceptos básicos de programación		
Temporalización:		
Primer trimestre	17 horas	10 sesiones
Resultados de aprendizaje:		
RA1. Reconoce la estructura de un programa informático, identificando y relacionando los elementos propios del lenguaje de programación utilizado.		
Contenidos:		
1. Identificación de los elementos de un programa informático: <ul style="list-style-type: none"> – Estructura y bloques fundamentales. – Variables. – Tipos de datos. – Literales. – Constantes. – Operadores y expresiones. – Conversiones de tipo. – Comentarios. 		
Criterios de evaluación:		
<ul style="list-style-type: none"> • Se han identificado los bloques que componen la estructura de un programa informático. • Se han creado proyectos de desarrollo de aplicaciones. • Se han utilizado entornos integrados de desarrollo. • Se han identificado los distintos tipos de variables y la utilidad específica de cada uno. • Se ha modificado el código de un programa para crear y utilizar variables. • Se han creado y utilizado constantes y literales. • Se han clasificado, reconocido y utilizado en expresiones los operadores del lenguaje. • Se ha comprobado el funcionamiento de las conversiones de tipo explícitas e implícitas. • Se han introducido comentarios en el código. 		

Unidad de Trabajo 2 – Estructura de código para programas de consola		
Temporalización:		
Primer trimestre	37 horas	21 sesiones
Resultados de aprendizaje:		

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

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

Contenidos:

3. Uso de estructuras de control:

- Estructuras de selección.
- Estructuras de repetición.
- Estructuras de salto.
- Control de excepciones.

5. Lectura y escritura de información:

- Tipos de flujos. Flujos de bytes y de caracteres.
- Clases relativas a flujos.
- Utilización de flujos.
- Entrada desde teclado.
- Salida a pantalla.
- Ficheros de datos. Registros.
- Apertura y cierre de ficheros. Modos de acceso.
- Escritura y lectura de información en ficheros.
- Utilización de los sistemas de ficheros.
- Creación y eliminación de ficheros y directorios.
- Interfaces.
- Concepto de evento.
- Creación de controladores de eventos.

Criterios de evaluación:

- Se ha escrito y probado código que haga uso de estructuras de selección.
- Se han utilizado estructuras de repetición.
- Se han reconocido las posibilidades de las sentencias de salto.
- Se ha escrito código utilizando control de excepciones.
- Se han creado programas ejecutables utilizando diferentes estructuras de control.
- Se han probado y depurado los programas.
- Se ha comentado y documentado el código.
- Se ha utilizado la consola para realizar operaciones de entrada y salida de información.
- Se han aplicado formatos en la visualización de la información.
- Se han reconocido las posibilidades de entrada / salida del lenguaje y las librerías asociadas.
- Se han utilizado ficheros para almacenar y recuperar información.
- Se han creado programas que utilicen diversos métodos de acceso al contenido de los ficheros.
- Se han implementado técnicas para envío y almacenamiento de objetos en ficheros
- Se han utilizado las herramientas del entorno de desarrollo para crear interfaces gráficos de usuario simples.
- Se han programado controladores de eventos.

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

Unidad de Trabajo 3 – Introducción a la Programación Orientada a Objetos (POO)

Temporalización:

Primer y segundo trimestre	30 horas	17 sesiones
----------------------------	----------	-------------

Resultados de aprendizaje:

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

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

Contenidos:

2. Utilización de objetos:

- Características de los objetos.
- «Instanciación» de objetos.
- Utilización de métodos.
- Utilización de propiedades.
- Utilización de métodos estáticos.
- Constructores.
- Destrucción de objetos y liberación de memoria.

4. Desarrollo de clases:

- Concepto de clase.
- Estructura y miembros de una clase.
- Creación de atributos.
- Creación de métodos.
- Creación de constructores.
- Utilización de clases y objetos.
- Utilización de clases heredadas.

Criterios de evaluación:

- Se han identificado los fundamentos de la programación orientada a objetos.
- Se han escrito programas simples.
- Se han instanciado objetos a partir de clases predefinidas.
- Se han utilizado métodos y propiedades de los objetos.
- Se han escrito llamadas a métodos estáticos.
- Se han utilizado parámetros en la llamada a métodos.
- Se han incorporado y utilizado librerías de objetos.
- Se han utilizado constructores.
- Se ha utilizado el entorno integrado de desarrollo en la creación y compilación de programas simples.
- Se ha reconocido la sintaxis, estructura y componentes típicos de una clase.
- Se han definido clases.
- Se han definido propiedades y métodos.
- Se han utilizado constructores y destructores.
- Se han desarrollado programas que instancien y utilicen objetos de las clases creadas anteriormente.

- Se han utilizado mecanismos para controlar la visibilidad de las clases y de sus miembros.
- Se han definido y utilizado clases heredadas.
- Se han creado y utilizado métodos estáticos.
- Se han definido y utilizado interfaces.
- Se han creado y utilizado conjuntos y librerías de clases.

Unidad de Trabajo 4 – Estructuras de almacenamiento		
Temporalización:		
Segundo trimestre	35 horas	20 sesiones
Resultados de aprendizaje:		
<p>RA3. Escribe y depura código, analizando y utilizando las estructuras de control del lenguaje.</p> <p>RA6. Escribe programas que manipulen información, seleccionando y utilizando tipos avanzados de datos.</p>		
Contenidos:		
<p>3. Uso de estructuras de control:</p> <ul style="list-style-type: none"> – Estructuras de selección. – Estructuras de repetición. – Estructuras de salto. – Control de excepciones. <p>6. Aplicación de las estructuras de almacenamiento:</p> <ul style="list-style-type: none"> – Estructuras. – Creación de arrays. – Arrays multidimensionales. – Cadenas. 		
Criterios de evaluación:		
<ul style="list-style-type: none"> • Se ha escrito y probado código que haga uso de estructuras de selección. • Se han utilizado estructuras de repetición. • Se han reconocido las posibilidades de las sentencias de salto. • Se ha escrito código utilizando control de excepciones. • Se han creado programas ejecutables utilizando diferentes estructuras de control. • Se han probado y depurado los programas. • Se ha comentado y documentado el código. • Se han escrito programas que utilicen arrays. • Se han reconocido las librerías de clases relacionadas con tipos de datos avanzados. • Se han utilizado listas para almacenar y procesar información. • Se han utilizado iteradores para recorrer los elementos de las listas. • Se han reconocido las características y ventajas de cada una de las colecciones de datos disponibles. • Se han creado clases y métodos genéricos. • Se han utilizado expresiones regulares en la búsqueda de patrones en cadenas de texto. 		

- Se han identificado las clases relacionadas con el tratamiento de documentos XML.
- Se han realizado programas que realicen manipulaciones sobre documentos XML.

Unidad de Trabajo 5 – Metodología avanzada de la Programación Orientada a Objetos (POO)

Temporalización:

Segundo y tercer trimestre	50 horas	29 sesiones
----------------------------	----------	-------------

Resultados de aprendizaje:

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

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

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

Contenidos:

3. Uso de estructuras de control:

- Estructuras de selección.
- Estructuras de repetición.
- Estructuras de salto.
- Control de excepciones.

5. Lectura y escritura de información:

- Tipos de flujos. Flujos de bytes y de caracteres.
- Clases relativas a flujos.
- Utilización de flujos.
- Entrada desde teclado.
- Salida a pantalla.
- Ficheros de datos. Registros.
- Apertura y cierre de ficheros. Modos de acceso.
- Escritura y lectura de información en ficheros.
- Utilización de los sistemas de ficheros.
- Creación y eliminación de ficheros y directorios.
- Interfaces.
- Concepto de evento.
- Creación de controladores de eventos.

7. Utilización avanzada de clases:

- Composición de clases.
- Herencia.
- Superclases y subclases.
- Clases y métodos abstractos y finales.
- Sobreescritura de métodos.
- Constructores y herencia.

Criterios de evaluación:

- Se ha escrito y probado código que haga uso de estructuras de selección.
- Se han utilizado estructuras de repetición.

- Se han reconocido las posibilidades de las sentencias de salto.
- Se ha escrito código utilizando control de excepciones.
- Se han creado programas ejecutables utilizando diferentes estructuras de control.
- Se han probado y depurado los programas.
- Se ha comentado y documentado el código.
- Se ha utilizado la consola para realizar operaciones de entrada y salida de información.
- Se han aplicado formatos en la visualización de la información.
- Se han reconocido las posibilidades de entrada / salida del lenguaje y las librerías asociadas.
- Se han utilizado ficheros para almacenar y recuperar información.
- Se han creado programas que utilicen diversos métodos de acceso al contenido de los ficheros.
- Se han implementado técnicas para envío y almacenamiento de objetos en ficheros.
- Se han utilizado las herramientas del entorno de desarrollo para crear interfaces gráficos de usuario simples.
- Se han programado controladores de eventos.
- Se han escrito programas que utilicen interfaces gráficos para la entrada y salida de información.
- Se han identificado los conceptos de herencia, superclase y subclase.
- Se han utilizado modificadores para bloquear y forzar la herencia de clases y métodos.
- Se ha reconocido la incidencia de los constructores en la herencia.
- Se han creado clases heredadas que sobrescriban la implementación de métodos de la superclase.
- Se han diseñado y aplicado jerarquías de clases.
- Se han probado y depurado las jerarquías de clases.
- Se han realizado programas que implementen y utilicen jerarquías de clases.
- Se ha comentado y documentado el código.

Unidad de Trabajo 6 – Introducción a bases de datos		
Temporalización:		
Tercer trimestre	25 horas	14 sesiones
Resultados de aprendizaje:		
RA8. Utiliza bases de datos orientadas a objetos, analizando sus características y aplicando técnicas para mantener la persistencia de la información.		
Contenidos:		
8. Mantenimiento de la persistencia de los objetos: <ul style="list-style-type: none"> – Bases de datos orientadas a objetos. – Características de las bases de datos orientadas a objetos. – Instalación del gestor de bases de datos. – Creación de bases de datos. – Mecanismos de consulta. 		

<ul style="list-style-type: none"> – El lenguaje de consultas: sintaxis, expresiones y operadores. – Recuperación, modificación y borrado de información. – Tipos de datos objeto; atributos y métodos. – Tipos de datos colección.
Criterios de evaluación:
<ul style="list-style-type: none"> • Se han identificado las características de las bases de datos orientadas a objetos. • Se ha analizado su aplicación en el desarrollo de aplicaciones mediante lenguajes orientados a objetos. • Se han instalado sistemas gestores de bases de datos orientados a objetos. • Se han clasificado y analizado los distintos métodos soportados por los sistemas gestores para la gestión de la información almacenada. • Se han creado bases de datos y las estructuras necesarias para el almacenamiento de objetos. • Se han programado aplicaciones que almacenen objetos en las bases de datos creadas. • Se han realizado programas para recuperar, actualizar y eliminar objetos de las bases de datos. • Se han realizado programas para almacenar y gestionar tipos de datos estructurados, compuestos y relacionados.

Unidad de Trabajo 7 – Uso avanzado de bases de datos		
Temporalización:		
Tercer trimestre	30 horas	17 sesiones
Resultados de aprendizaje:		
RA9. Gestiona información almacenada en bases de datos relacionales manteniendo la integridad y la consistencia de los datos.		
Contenidos:		
9. Gestión de bases de datos relacionales: <ul style="list-style-type: none"> – Establecimiento de conexiones. – Recuperación de información. – Manipulación de la información. – Ejecución de consultas sobre la base de datos. 		
Criterios de evaluación:		
<ul style="list-style-type: none"> • Se han identificado las características y métodos de acceso a sistemas gestores de bases de datos relacionales. • Se han programado conexiones con bases de datos. • Se ha escrito código para almacenar información en bases de datos. • Se han utilizado diferentes asistentes del entorno de desarrollo. • Se han identificado las características y métodos de acceso a sistemas gestores de bases de datos relacionales. • Se han programado conexiones con bases de datos. • Se ha escrito código para almacenar información en bases de datos. • Se han utilizado diferentes asistentes del entorno de desarrollo. 		

3.6 Metodología

La metodología a emplear durante el proceso de enseñanza aprendizaje no es una metodología única, sino un enfoque plurimetodológico que trata de enfatizar los componentes motivacionales intrínsecos existentes en el diseño de actividades, fomentar la motivación y captar la atención del alumnado.

Las metodologías activas en las que se basa esta programación son las siguientes:

- Aprendizaje basado en proyectos.
 - Enfoque pedagógico que se centra en la realización de proyectos o tareas prácticas como base principal del aprendizaje de los estudiantes en lugar de centrarse únicamente en la adquisición de conocimientos teóricos. Este enfoque fomenta la participación activa de los estudiantes en la resolución de problemas y la aplicación de habilidades y conocimientos en contextos reales.
- Pensamiento de diseño.
 - Enfoque que se utiliza para abordar problemas complejos y desarrollar soluciones innovadoras. Se basa en la forma en la que los diseñadores abordan los desafíos, centrándose en la comprensión profunda de las necesidades y deseos de las personas involucradas, y en la generación de ideas creativas y viables para resolver problemas.
- (Meta)gamificación.
 - Véase el apartado “[Introducción](#)” del proyecto de innovación educativa.
- Aprendizaje cooperativo.
 - Enfoque educativo que se basa en la colaboración y el trabajo en equipo entre los estudiantes para lograr objetivos comunes de aprendizaje. A diferencia del aprendizaje individual, donde cada estudiante trabaja de manera independiente, en el aprendizaje cooperativo los estudiantes trabajan juntos en grupos pequeños, aprovechando sus habilidades y conocimientos individuales para lograr un objetivo compartido.

Pese a tener estas metodologías en mente, no se trata de aplicarlas durante todas las sesiones; sino de aplicarlas en los momentos clave para lograr un aprendizaje significativo y duradero. Siguiendo lo expuesto en la “Teoría del Aprendizaje Significativo” (Ausubel, 1983), un aprendizaje significativo no tiene que ser por descubrimiento, puede darse perfectamente por recepción, siempre y cuando se tenga en cuenta la estructura cognitiva del/de la educando/a y se vincule con aprendizajes y/o conocimientos previos del/de la mismo/a; de esta manera, complementar clases guiadas con ejercicios y un proyecto final motivante en el que aplicar esos conocimientos y destrezas adquiridos en un contexto diferenciado (crear algo tangible), permite alcanzar los objetivos de esta programación.

También se aplica el DUA (Diseño Universal de Aprendizaje), según lo expuesto en “Diseño Universal para el Aprendizaje: un modelo teórico-práctico para una educación inclusiva de calidad” (Alba Pastor, 2019); es necesario activar ciertas zonas del cerebro del alumnado a través de tareas que suponen demandas cognitivas. Alba Pastor organiza estas zonas en tres grupos de redes neuronales (afectivas, de reconocimiento y estratégicas); siendo activar dichas redes el referente de esta programación, especialmente a la hora de diseñar y realizar los proyectos.

Se da la opción al alumnado de realizar el proyecto de manera individual, esto es debido a que la libertad de realizar un proyecto que motive a cada uno/a de los/las alumnos/as no es posible si forzamos a aquellos desparejados a unirse al proyecto de otros grupos. En el proyecto el aprendizaje cooperativo se da siempre: durante el trabajo en clase al compartir ideas y problemas con otros grupos, durante las presentaciones y, sobre todo, al recibir feedback y consejos de otros grupos que es parte obligatoria de la metodología. El otro extremo también se contempla, grupos muy grandes (dentro de lo razonable) recibirán un nivel de exigencia superior al del resto de grupos, aunque se tiene en cuenta las dificultades organizativas que un grupo de gran tamaño (más de 5 personas) puede suponer.

Colateralmente, este enfoque favorece que ningún grupo sienta la necesidad de proteger su trabajo de copia. Cada proyecto es diferente y el ambiente que se busca es de compartir el conocimiento y ayudar a resolver problemas; algo que no es posible con un enunciado cerrado en el que el producto final “correcto” es el mismo para todo el alumnado.

3.7 Actividades

Todas las Unidades de Trabajo de la programación se desarrollan en bloques de clases guiadas acompañadas de ejercicios a realizar en clase o como tarea para casa. Las tareas siguen la misma filosofía que el proyecto expuesto en la innovación, están arraigadas en los intereses de los/las alumnos/as y promueven la motivación de los/las mismos/as; llegando incluso a servir como prototipo o inspiración para la realización del proyecto posteriormente. Estas tareas no exigen una ejecución perfecta ni se evalúa la calidad de las mismas; con entregar o mostrar al docente que la actividad ha sido realizada es suficiente. Lo que se busca con estas tareas es ver que los conceptos clave explicados en las clases guiadas han sido absorbidos correctamente por el alumnado; si un/a alumno/a no entrega, entrega en blanco o entrega la tarea mal hecha, es una oportunidad para que el/la docente explore las dificultades que éste/a esté experimentando y/o prepare clases y/o materiales de repaso si la situación abarca a una cantidad significativa del alumnado. No todas las clases guiadas tienen una tarea asociada, pudiendo éstas abarcar los conceptos vistos en varias clases guiadas.

Esos bloques **de clases guiadas acompañadas de ejercicios** son complementados por la **realización de un proyecto** en el que se aplique todo lo visto hasta el momento y que ocupa **un mes de cada trimestre** (tiempo extraído de manera equitativa de cada una de las Unidades de Trabajo que abarca). Esta es la oportunidad de **ejecutar la innovación** que se plantea en el apartado de “[Proyecto de innovación educativa](#)”.

De esta manera y siguiendo el enfoque metodológico, organizativo y conceptual de la programación; la estructura de un trimestre sería la siguiente:

- **UT₁**
 - Clase/s guiada/s 1.
 - Tarea 1.
 - Clase/s guiada/s N.
 - Tarea N.
- **UT₂**
 - Clase/s guiada/s 1.
 - Tarea 1.

- Clase/s guiada/s N.
 - Tarea N.
- **UT₃**
 - Clase/s guiada/s 1.
 - Tarea 1.
 - Clase/s guiada/s N.
 - Tarea N.
- **Proyecto (UT₁, UT₂ y UT₃)**
 - Desarrollar la innovación para que el alumnado aplique todos los contenidos, destrezas y actitudes adquiridos en un contexto diferenciado para crear un producto tangible arraigado en sus intereses y con un alto componente motivacional.

Las clases guiadas no son meras aportaciones teóricas que el alumnado deba absorber, son intercambios entre el docente y el alumnado en los que el docente les da píldoras teóricas que los/as alumnos/as desarrollan inmediatamente de manera práctica; con reiteraciones finales del docente para afianzar los conocimientos y destrezas que acaban de ejecutar.

Las tareas se realizan principalmente en casa, a menos que sobre tiempo de clase para ello. La idea es que, el hecho de que una clase avance más rápido de lo normal, puede significar dos cosas: que todo se ha entendido y el tiempo extra les quitará trabajo innecesario en casa; o que las dudas no han aflorado y la tarea es el momento perfecto para que afloren.

3.7.1 Superposición de trimestres

Según el apartado de “[Temporalización](#)” hay dos Unidades de Trabajo en las que dos trimestres se superponen, esto es intencional pues se busca tener cierta flexibilidad a la hora de dar por finalizada una Unidad de Trabajo. Podría darse el caso en el que no haya dado tiempo a tratar ciertos contenidos, pero llegue el último mes del trimestre y se empiece a trabajar en el proyecto; lejos de olvidarnos de esa parte de la Unidad de Trabajo,

al llegar el siguiente trimestre se realizarán las clases guiadas y tareas restantes de la misma y se pedirá al alumnado que de manera retroactiva los aplique en sus proyectos ya finalizados. Esta situación, como la innovación base, también se ha probado en un contexto real y el resultado aparente es provocar en el alumnado la comprensión de la reducción de tiempo o complejidad que hubiera supuesto haber aplicado dichos conceptos durante la realización del proyecto (siendo las frases “Claro, si hubiera sabido esto antes no lo hubiera hecho así” y “Ya decía yo que tenía que existir una manera más fácil de hacerlo” sorprendentemente comunes en esta fase). Sigue siendo un aprendizaje significativo y añade el valor de visitar aprendizajes pasados para consolidar aprendizajes futuros; sin duda un tema de interés para próximas investigaciones.

3.7.2 Ejemplo concreto de trimestre

Para exponer lo que esta programación trata de conseguir, a continuación, se incluye el diseño de lo que sería el primer trimestre del Ciclo Formativo de Grado Superior de Desarrollo de Aplicaciones Web en el módulo de Programación. Este desglose abarca las tres primeras Unidades de Trabajo de la programación y en él se desarrollarán ejemplificaciones de las tareas concretas a realizar por el alumnado.

Unidad de Trabajo 1 – Conceptos básicos de programación		
Temporalización:		
Primer trimestre	17 horas (8 dedicadas al proyecto)	10 sesiones (4 sesiones dedicadas al proyecto)
Resultados de aprendizaje:		
RA1. Reconoce la estructura de un programa informático, identificando y relacionando los elementos propios del lenguaje de programación utilizado.		
Contenidos:		
1. Identificación de los elementos de un programa informático: <ul style="list-style-type: none"> – Estructura y bloques fundamentales. – Variables. – Tipos de datos. – Literales. – Constantes. – Operadores y expresiones. – Conversiones de tipo. – Comentarios. 		
Criterios de evaluación:		
<ul style="list-style-type: none"> • Se han identificado los bloques que componen la estructura de un programa informático. 		

- Se han creado proyectos de desarrollo de aplicaciones.
- Se han utilizado entornos integrados de desarrollo.
- Se han identificado los distintos tipos de variables y la utilidad específica de cada uno.
- Se ha modificado el código de un programa para crear y utilizar variables.
- Se han creado y utilizado constantes y literales.
- Se han clasificado, reconocido y utilizado en expresiones los operadores del lenguaje.
- Se ha comprobado el funcionamiento de las conversiones de tipo explícitas e implícitas.
- Se han introducido comentarios en el código.

Actividades a realizar:

- **Clases guiadas (Sesiones 1 a 2)**
 - Tratar los elementos básicos de un programa informático; también se dará contexto sobre la programación en general y las habilidades de un/a buen programador/a.
- **Clases guiadas (Sesión 3)**
 - Primer contacto con los entornos integrados de desarrollo que se van a usar durante el curso. Así mismo, se crearán los primeros proyectos de desarrollo.
- **Tarea**
 - Realizar programas sencillos usando los elementos básicos de un programa informático en un proyecto dentro del IDE (Entorno Integrado de Desarrollo según las siglas en inglés).
 - Ejemplo: ¡Hola Mundo!
- **Clases guiadas (Sesión 5)**
 - Tratar elementos más avanzados de un programa informático, haciendo hincapié en la conversión de tipos.
- **Clases guiadas (Sesión 6)**
 - Tipos de comentarios en el código y su correcto uso. Introducción a la documentación de código.
- **Tarea**
 - Realizar programas informáticos usando elementos más avanzados de los mismos, con conversión de tipos y los comentarios pertinentes.
 - Ejemplo: Programa que encripta y desencripta de mensajes convirtiendo de caracteres a números y viceversa.

Metodología:

- Clases guiadas.
- Aprendizaje/trabajo individual.
- Tutorización/seguimiento.

Unidad de Trabajo 2 – Estructura de código para programas de consola

Temporalización:

Primer trimestre	37 horas (14 dedicadas al proyecto)	21 sesiones (7 sesiones dedicadas al proyecto)
------------------	-------------------------------------	--

Resultados de aprendizaje:
RA3. Escribe y depura código, analizando y utilizando las estructuras de control del lenguaje.
RA5. Realiza operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases.
Contenidos:
3. Uso de estructuras de control: <ul style="list-style-type: none">– Estructuras de selección.– Estructuras de repetición.– Estructuras de salto.– Control de excepciones.
5. Lectura y escritura de información: <ul style="list-style-type: none">– Tipos de flujos. Flujos de bytes y de caracteres.– Clases relativas a flujos.– Utilización de flujos.– Entrada desde teclado.– Salida a pantalla.– Ficheros de datos. Registros.– Apertura y cierre de ficheros. Modos de acceso.– Escritura y lectura de información en ficheros.– Utilización de los sistemas de ficheros.– Creación y eliminación de ficheros y directorios.– Interfaces.– Concepto de evento.– Creación de controladores de eventos.
Criterios de evaluación:
<ul style="list-style-type: none">• Se ha escrito y probado código que haga uso de estructuras de selección.• Se han utilizado estructuras de repetición.• Se han reconocido las posibilidades de las sentencias de salto.• Se ha escrito código utilizando control de excepciones.• Se han creado programas ejecutables utilizando diferentes estructuras de control.• Se han probado y depurado los programas.• Se ha comentado y documentado el código.• Se ha utilizado la consola para realizar operaciones de entrada y salida de información.• Se han aplicado formatos en la visualización de la información.• Se han reconocido las posibilidades de entrada / salida del lenguaje y las librerías asociadas.• Se han utilizado ficheros para almacenar y recuperar información.• Se han creado programas que utilicen diversos métodos de acceso al contenido de los ficheros.• Se han implementado técnicas para envío y almacenamiento de objetos en ficheros• Se han utilizado las herramientas del entorno de desarrollo para crear interfaces gráficos de usuario simples.• Se han programado controladores de eventos.

<ul style="list-style-type: none"> • Se han escrito programas que utilicen interfaces gráficas para la entrada y salida de información.
<p>Actividades a realizar:</p> <ul style="list-style-type: none"> • Clases guiadas (Sesiones 1 a 6) <ul style="list-style-type: none"> ○ Explicación extensiva sobre estructuras de control y ejemplos guiados creando aplicaciones de consola. • Tarea <ul style="list-style-type: none"> ○ Aplicación de consola de cierta complejidad que use estructuras de control. ○ Ejemplo: Programa estilo libros de “elige tu propia aventura”. • Clases guiadas (Sesiones 7 a 10) <ul style="list-style-type: none"> ○ Depuración de código. • Clases guiadas (Sesiones 11 a 14) <ul style="list-style-type: none"> ○ Reiterar sobre estructuras de control, su uso y más ejemplos guiados. • Tarea <ul style="list-style-type: none"> ○ Aplicación de consola de mayor complejidad que use estructuras de control. ○ Ejemplo: Tres en raya.
<p>Metodología:</p> <ul style="list-style-type: none"> • Clases guiadas. • Aprendizaje/trabajo individual. • Tutorización/seguimiento.

Unidad de Trabajo 3 – Introducción a la Programación Orientada a Objetos (POO)		
Temporalización:		
Primer y segundo trimestre	30 horas (10 dedicadas al proyecto)	17 sesiones (5 sesiones dedicadas al proyecto)
Resultados de aprendizaje:		
<p>RA2. Escribe y prueba programas sencillos, reconociendo y aplicando los fundamentos de la programación orientada a objetos.</p> <p>RA4. Desarrolla programas organizados en clases analizando y aplicando los principios de la programación orientada a objetos.</p>		
Contenidos:		
<p>2. Utilización de objetos:</p> <ul style="list-style-type: none"> – Características de los objetos. – «Instanciación» de objetos. – Utilización de métodos. – Utilización de propiedades. – Utilización de métodos estáticos. – Constructores. – Destrucción de objetos y liberación de memoria. <p>4. Desarrollo de clases:</p> <ul style="list-style-type: none"> – Concepto de clase. – Estructura y miembros de una clase. 		

- Creación de atributos.
- Creación de métodos.
- Creación de constructores.
- Utilización de clases y objetos.
- Utilización de clases heredadas.

Criterios de evaluación:

- Se han identificado los fundamentos de la programación orientada a objetos.
- Se han escrito programas simples.
- Se han instanciado objetos a partir de clases predefinidas.
- Se han utilizado métodos y propiedades de los objetos.
- Se han escrito llamadas a métodos estáticos.
- Se han utilizado parámetros en la llamada a métodos.
- Se han incorporado y utilizado librerías de objetos.
- Se han utilizado constructores.
- Se ha utilizado el entorno integrado de desarrollo en la creación y compilación de programas simples.
- Se ha reconocido la sintaxis, estructura y componentes típicos de una clase.
- Se han definido clases.
- Se han definido propiedades y métodos.
- Se han utilizado constructores y destructores.
- Se han desarrollado programas que instancien y utilicen objetos de las clases creadas anteriormente.
- Se han utilizado mecanismos para controlar la visibilidad de las clases y de sus miembros.
- Se han definido y utilizado clases heredadas.
- Se han creado y utilizado métodos estáticos.
- Se han definido y utilizado interfaces.
- Se han creado y utilizado conjuntos y librerías de clases.

Actividades a realizar: 12

- **Clases guiadas** (Sesiones 1 y 2)
 - Conceptos básicos de la Programación Orientada a Objetos.
- **Clases guiadas** (Sesiones 3 y 4)
 - Explicación extensiva y aplicación de conceptos básicos de la Programación Orientada a Objetos en ejemplos guiados más elaborados.
- **Tarea**
 - Programa sencillo que use los conceptos básicos de la Programación Orientada a Objetos.
 - Ejemplo: Arena de combate (dos personajes atacándose y ganando y perdiendo vida).
- **Clases guiadas** (Sesiones 5 a 8)
 - Diseño de programas siguiendo la Metodología de la Programación Orientada a Objetos, haciendo hincapié en estructurar las mentes del alumnado para entender un programa a varios niveles de abstracción.
- **Clases guiadas** (Sesiones 9 a 12)

<ul style="list-style-type: none"> ○ Diseño paso a paso de un programa siguiendo los criterios de la Metodología de la Programación Orientada a Objetos y ejemplos guiados. ● Tarea <ul style="list-style-type: none"> ○ Diseñar y crear un programa de cierta complejidad siguiendo la Metodología de la Programación Orientada a Objetos. ○ Ejemplo: Combate Pokémon.
Metodología:
<ul style="list-style-type: none"> ● Clases guiadas. ● Aprendizaje/trabajo individual. ● Tutorización/seguimiento. ● Pensamiento de diseño.

Proyecto (UT1, UT2 y UT3)		
Temporalización:		
Primer y segundo trimestre	28 horas (8, 14 y 10 de la UT1, la UT2 y la UT3 respectivamente)	16 sesiones (4, 7 y 5 de la UT1, la UT2 y la UT3 respectivamente)
Resultados de aprendizaje:		
<p>RA1. Reconoce la estructura de un programa informático, identificando y relacionando los elementos propios del lenguaje de programación utilizado.</p> <p>RA2. Escribe y prueba programas sencillos, reconociendo y aplicando los fundamentos de la programación orientada a objetos.</p> <p>RA3. Escribe y depura código, analizando y utilizando las estructuras de control del lenguaje.</p> <p>RA4. Desarrolla programas organizados en clases analizando y aplicando los principios de la programación orientada a objetos.</p> <p>RA5. Realiza operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases.</p>		
Contenidos:		
<ol style="list-style-type: none"> 1. Identificación de los elementos de un programa informático: <ul style="list-style-type: none"> – Estructura y bloques fundamentales. – Variables. – Tipos de datos. – Literales. – Constantes. – Operadores y expresiones. – Conversiones de tipo. – Comentarios. 2. Utilización de objetos: <ul style="list-style-type: none"> – Características de los objetos. – «Instanciación» de objetos. – Utilización de métodos. – Utilización de propiedades. – Utilización de métodos estáticos. – Constructores. 		

- Destrucción de objetos y liberación de memoria.
- 3. Uso de estructuras de control:
 - Estructuras de selección.
 - Estructuras de repetición.
 - Estructuras de salto.
 - Control de excepciones.
- 4. Desarrollo de clases:
 - Concepto de clase.
 - Estructura y miembros de una clase.
 - Creación de atributos.
 - Creación de métodos.
 - Creación de constructores.
 - Utilización de clases y objetos.
 - Utilización de clases heredadas.
- 5. Lectura y escritura de información:
 - Tipos de flujos. Flujos de bytes y de caracteres.
 - Clases relativas a flujos.
 - Utilización de flujos.
 - Entrada desde teclado.
 - Salida a pantalla.
 - Ficheros de datos. Registros.
 - Apertura y cierre de ficheros. Modos de acceso.
 - Escritura y lectura de información en ficheros.
 - Utilización de los sistemas de ficheros.
 - Creación y eliminación de ficheros y directorios.
 - Interfaces.
 - Concepto de evento.
 - Creación de controladores de eventos.

Criterios de evaluación:

- Se han identificado los bloques que componen la estructura de un programa informático.
- Se han creado proyectos de desarrollo de aplicaciones.
- Se han utilizado entornos integrados de desarrollo.
- Se han identificado los distintos tipos de variables y la utilidad específica de cada uno.
- Se ha modificado el código de un programa para crear y utilizar variables.
- Se han creado y utilizado constantes y literales.
- Se han clasificado, reconocido y utilizado en expresiones los operadores del lenguaje.
- Se ha comprobado el funcionamiento de las conversiones de tipo explícitas e implícitas.
- Se han introducido comentarios en el código.
- Se ha escrito y probado código que haga uso de estructuras de selección.
- Se han utilizado estructuras de repetición.
- Se han reconocido las posibilidades de las sentencias de salto.
- Se ha escrito código utilizando control de excepciones.

- Se han creado programas ejecutables utilizando diferentes estructuras de control.
- Se han probado y depurado los programas.
- Se ha comentado y documentado el código.
- Se ha utilizado la consola para realizar operaciones de entrada y salida de información.
- Se han aplicado formatos en la visualización de la información.
- Se han reconocido las posibilidades de entrada / salida del lenguaje y las librerías asociadas.
- Se han utilizado ficheros para almacenar y recuperar información.
- Se han creado programas que utilicen diversos métodos de acceso al contenido de los ficheros.
- Se han implementado técnicas para envío y almacenamiento de objetos en ficheros
- Se han utilizado las herramientas del entorno de desarrollo para crear interfaces gráficos de usuario simples.
- Se han programado controladores de eventos.
- Se han escrito programas que utilicen interfaces gráficos para la entrada y salida de información.
- Se han identificado los fundamentos de la programación orientada a objetos.
- Se han escrito programas simples.
- Se han instanciado objetos a partir de clases predefinidas.
- Se han utilizado métodos y propiedades de los objetos.
- Se han escrito llamadas a métodos estáticos.
- Se han utilizado parámetros en la llamada a métodos.
- Se han incorporado y utilizado librerías de objetos.
- Se han utilizado constructores.
- Se ha utilizado el entorno integrado de desarrollo en la creación y compilación de programas simples.
- Se ha reconocido la sintaxis, estructura y componentes típicos de una clase.
- Se han definido clases.
- Se han definido propiedades y métodos.
- Se han utilizado constructores y destructores.
- Se han desarrollado programas que instancien y utilicen objetos de las clases creadas anteriormente.
- Se han utilizado mecanismos para controlar la visibilidad de las clases y de sus miembros.
- Se han definido y utilizado clases heredadas.
- Se han creado y utilizado métodos estáticos.
- Se han definido y utilizado interfaces.
- Se han creado y utilizado conjuntos y librerías de clases.

Actividades a realizar:

- **Fase 1: Diseño del videojuego (Sesión 1)**

- En esta primera sesión el docente explica el proyecto, pidiendo al alumnado que se distribuyan en grupos (de manera opcional, siempre se puede realizar individualmente).
- El proyecto tiene unos requerimientos mínimos que todos los proyectos deben alcanzar:
 - Entregar un ejecutable del videojuego que pueda ejecutarse sin errores.
 - Usar estructuras de control.
 - Seguir la Metodología de la Programación Orientada a Objetos:
 - Realizar un diseño de clases adecuado.
 - Usar herencia.
 - Comentar el código de manera funcional para el desarrollo.
 - Aplicar el feedback recibido de una manera significativa.
 - Documentar el proyecto.
 - Siguiendo la filosofía de las metodologías ágiles (no confundir con las metodologías activas de enseñanza) la documentación no debe ser extensa sino funcional, una foto de una servilleta se puede ilustrar una idea de manera más eficaz que una página de texto.
 - Se deberá plasmar el impacto que el feedback ha tenido en el entregable final.
- Se deja tiempo para que los distintos grupos lleguen a una idea (sería conveniente haber comentado previamente la realización de este proyecto para que ya se estén barajando opciones) que deberán exponer al resto de grupos y al docente en la siguiente sesión. El tiempo es suficiente para que no sea necesario trabajar en casa en la propia presentación.
- Es importante que la idea motive a todos los miembros del grupo, que esté arraigada en sus intereses. Como es obvio, hay grupos que se atascarán en esta fase, por lo que el docente debe estar atento para localizar dichos grupos, ver que cosas les interesan y proponerles ideas relacionadas con éstas. Es muy sencillo tomar muchos tipos de videojuegos sencillos y adaptarlos a cualquier temática, aunque sea un programa de televisión o un deporte.
- **Fase 2: Presentación de la idea (Sesión 2)**
 - Ante toda la clase cada grupo expone su idea, explicando cual es el producto final que tratarán de desarrollar. El formato de dicha presentación se enfoca a “vender la idea”, como si los oyentes fueran un grupo de inversores del que dependiera sus empleos. Es lo que más se valorará de cara a calificar esta parte.
 - Se podrán emplear todo tipo de recursos: desde hacerlo a viva voz hasta realizar un montaje con música y vídeo.
 - Al final de cada presentación, el resto de grupos dan su impresión y critican la idea constructivamente.
 - Con esta presentación los grupos se comprometen a realizar ese proyecto. En realidad, durante el desarrollo surgen problemas y es normal que la idea inicial se altere, pero el alumnado debe tener en mente que se tienen que aproximar a ésta lo máximo la posible.

- El docente vela porque los grupos menos ambiciosos revisen sus ideas y aquellos demasiado ambiciosos reduzcan sus expectativas, pero dentro de lo posible se intenta que la idea original sea respetada.
- **Fase 3: Sprint 1 (Sesiones 3 a 5)**
 - Al principio de cada semana los grupos eligen qué tareas realizar durante las tres primeras sesiones de entre todas las que determinen. Para ello deben reunirse cada día, siguiendo la metodología Scrum, y establecer sus prioridades.
 - El proyecto se realiza principalmente en clase, aunque los grupos más ambiciosos lo complementan con trabajo en casa.
 - El docente toma nota del trabajo diario, el cual los grupos deben plasmar en la documentación. La nota es individual y la observación de ese trabajo influirá en ella.
 - El docente también sirve de apoyo a los grupos, haciendo apartes para resolver las dudas más generales.
 - Según esta distribución, los grupos cuentan con 3 sesiones (de 2 horas cada una) para preparar el entregable de cada semana (el fin del sprint). Ese entregable debe ser funcional (que se pueda ejecutar y probar), como dicta la metodología Scrum, pero no se tendrá en cuenta que estos sean versiones reducidas o muy poco pulidas.
- **Fase 4: Entregable, presentación y feedback 1 (Sesión 6)**
 - Al final de cada semana, durante la cuarta sesión tras haber pasado 3 sesiones desarrollando el proyecto, los grupos vuelven a presentar; exponiendo esta vez los avances realizados, los problemas solucionados y los problemas que aún están afrontando. Estas presentaciones se enfocan desde un punto de vista puramente de desarrollo, dejando a un lado el “vender la idea/producto”.
 - En las presentaciones se muestra el videojuego funcionando y el resto de grupos prueban el mismo ejecutable una vez termine la presentación para darles feedback; el cual anotan para enfocar el resto del desarrollo, fundamental de cara a la entrega final.
 - El resto de grupos tratan de aportar soluciones a los problemas de cada grupo y toman nota de las soluciones a los problemas resueltos.
 - El docente enviará su feedback de manera más formal y calificará tanto las presentaciones como la entrega.
- **Fase 5: Sprint 2 (Sesión 7 a 10)**
 - Igual que el sprint 1, pero contando con más feedback y recursos para seguir avanzando.
- **Fase 6: Entregable, presentación y feedback 2 (Sesión 11)**
 - Igual que la anterior presentación, pero se hará más hincapié en tener un prototipo avanzado.
- **Fase 7: Sprint 3 (Sesión 12 a 14)**
 - Igual que el sprint 1 y 2, pero contando con que será la última semana de desarrollo; se deben establecer las tareas finales y tener en consideración el tiempo necesario para solucionar errores y pulir el videojuego.
- **Fase 8: Entrega y presentación final (Sesión 15)**

- Esta vez la presentación es justo lo contrario, se deberá “vender” el videojuego a toda costa. Se presenta un tráiler y se juega al videojuego en directo; se podrá falsear todo: editar el tráiler, fingir jugar en directo, jugar a un ejecutable preparado para la presentación... lo que haga falta. Que la presentación “venta” será lo que más se valore de cara a la nota.
- Tras la presentación el resto de grupos juegan al juego, viendo si la presentación es fiel a la realidad.
- El ejecutable se entrega junto a la documentación, cerrándose así todo el proceso de desarrollo. Estos serán evaluados por el docente.
- **Fase 9: Defensa (Sesión 16)**
 - En esta última sesión, los grupos serán convocados de manera individual (un solo grupo simultáneamente).
 - El docente jugará a los videojuegos y revisará el código de los mismos mientras realiza preguntas a todos los miembros del grupo; haciendo hincapié en comprobar su conocimiento sobre el trabajo realizado por los demás miembros del grupo y llegando a poder pedir modificaciones en el código en los casos en los que el trabajo visto en clase no se corresponda con el nivel de calidad de la entrega.

Metodología:

- Aprendizaje cooperativo/individual.
- Gamificación.
- Pensamiento de diseño.
- Aprendizaje cooperativo/individual.

3.8 Coordinación intradepartamental

Esta programación da pie a grandes oportunidades para colaborar dentro del departamento de Informática y Comunicaciones. Los proyectos pueden ocurrir simultáneamente en varios módulos a la vez; siendo una sola entrega para varios módulos y permitiendo que contenidos transversales se traten desde varios puntos de vista para producir una sola aplicación/videojuego.

Este módulo en concreto, el de Programación, es tan nuclear al ciclo que todos los módulos de primero son susceptibles de una coordinación/colaboración fructífera:

- Sistemas informáticos.
- Bases de datos.
- Lenguajes de marcas y sistemas de gestión de información.
- Entornos de desarrollo.

Pese al gran potencial que supondría una colaboración absoluta dentro del departamento, en la que todos estos módulos se relacionen unos con otros y se transvasen conocimientos y aprendizajes de manera activa, en la realidad resulta muy difícil de conseguir. Habría que contar con el apoyo del departamento y la voluntad de esos docentes para realizar más trabajo del que ya están realizando; la experiencia en dichos departamentos demuestra que hasta para las cosas más simples (discutir la situación de un alumno u homogeneizar respuestas para la atención a la diversidad) la colaboración es nula. Por todo ello, esta programación no requiere de la existencia de coordinación intradepartamental y simplemente la considera una vía interesante que explorar, pero ardua de ejecutar.

3.9 Actividades complementarias/extraescolares

El módulo de Programación se presta a organizar salidas muy orientadas al entorno laboral de empresas de desarrollo de software al uso; como propuesta, esta programación plantea organizar salidas a empresas de desarrollo de videojuegos asturianas, tanto de “serious games” (empresas que hacen juegos como servicio social y educativos, como CuiCui) como de videojuegos de entretenimiento (como Milkstone Studios). Exponer a los alumnos a maneras de producir, diseñar y desarrollar diferentes a las tradicionales, puede ser muy enriquecedor y abrir sus horizontes a puestos de trabajo más allá del desarrollador front, backend o full stack al uso.

3.10 Atención a la diversidad

El alumnado es cada vez más diverso y es necesario tener en mente medidas que atiendan a dicha diversidad. Dadas las características del perfil de los/las alumnos/as que ingresan en el Ciclo Formativo de Grado Superior de Desarrollo de Aplicaciones Web, se ha observado que es necesario tener en cuenta a personas con problemas de visión y neurodivergentes. Siguiendo las directrices DUA (Diseño Universal de Aprendizaje) (Alba Pastor, 2019), se diseñan las clases ofreciendo la mayor cantidad de medios de

representación que los recursos disponibles permiten; de esta manera se seguirán las siguientes medidas:

- Ofrecer toda la teoría tanto en dispositivas como en texto, disponible para todo el alumnado.
- Verbalizar los procedimientos prácticos que se realizan durante las clases.
- Retransmitir la pantalla del profesor tanto en el proyector como a través de una aplicación remota que además grabe las clases para futura consulta por parte del alumnado.
- Facilitar buenas condiciones de visibilidad y luminosidad; así como que todo el contenido de la asignatura se presente en una gama de colores que favorezcan el contraste y la legibilidad.

3.11 Evaluación

La evaluación, según esta programación, es formativa, continua, global y sumativa; siendo la adquisición de competencias (alcanzar los resultados de aprendizaje) radical a la hora de valorar el nivel de desempeño del alumnado en el módulo.

3.11.1 Evaluación inicial

Al situarse el módulo de Programación durante el primer curso del Ciclo Formativo de Grado Superior de Desarrollo de Aplicaciones Web, no se presupone ningún conocimiento en el alumnado. Es posible que los/las alumnos/as cuenten con experiencia programando adquirida en la educación secundaria obligatoria o incluso en el entorno laboral; pero no se puede tomar un nivel de partida tan irregular como base sólida para afianzar nuevos aprendizajes. Para esta programación, se ha optado por dar el módulo desde cero; se imparte desde los conceptos más básicos y se realiza una progresión homogénea controlada y supervisada por el docente.

3.11.2 Evaluación del aprendizaje

La evaluación de cada **trimestre** se realiza de la misma manera, usando instrumentos y técnicas que permitan establecer la nota final siguiendo los objetivos de la programación. No es necesario aprobar cada parte de manera separada, excepto el proyecto y la defensa (si se suspende uno se suspende el otro). Se establece así:

- **Participación y colaboración: 10%**
 - Una participación en clase positiva, apoyo a otros/as compañeros/as, comentarios constructivos y buena disposición en general.
 - El instrumento de evaluación será una **guía de observación**, en la que se anotará las intervenciones positivas y negativas relevantes de cada alumno/a.
- **Entrega de tareas: 30%**
 - La correcta entrega de todas las tareas cuenta lo mismo de cara a este porcentaje; solo la correcta entrega (que se haya intentado alcanzar el objetivo de la tarea).
 - El número de tareas puede variar dependiendo de las características y/o rendimiento del grupo clase.
 - El instrumento de evaluación será una **lista de cotejo**, en la que se establecerá si se ha entregado o no una tarea correctamente.
- **Proyecto: 50%**
 - Entrega del proyecto, las presentaciones, el trabajo diario observado, la documentación y el ejecutable correspondiente.
 - El instrumento de evaluación será una **rúbrica** con los requisitos mínimos y de ampliación recogidos, de manera que el alumno sepa perfectamente a que nota aspira de manera objetiva. Se reparte la nota de la siguiente manera:
 - 25% entrega del proyecto (del IDE correspondiente) y ejecutable.
 - 10% presentación inicial y final.
 - 10% presentaciones de desarrollo.
 - 5% trabajo observado durante los sprints.
- **Defensa del Proyecto: 10%**

- Los grupos (o individuo) serán convocados de manera individual (un solo grupo simultáneamente).
- El docente jugará a los videojuegos y revisará el código de los mismos mientras realiza preguntas a todos los miembros del grupo; haciendo hincapié en comprobar su conocimiento sobre el trabajo realizado por los demás miembros del grupo y llegando a poder pedir modificaciones en el código en los casos en los que el trabajo visto en clase no se corresponda con el nivel de calidad de la entrega.
- También es la oportunidad para comprobar si algún miembro del grupo ha realizado más o menos trabajo que el resto y comparar lo observado durante el desarrollo con la documentación y la capacidad de cada alumno/a para responder preguntas y realizar modificaciones.
- Es posible que algunos miembros del grupo suspendan esta parte y otros no.
- El instrumento de evaluación será una **escala de valoración**, en la que los ítems valorarán el nivel de participación en el proyecto, conocimiento del mismo y capacidad de modificarlo.

La **nota final del módulo** (en el caso de necesitarse) es la **media aritmética** entre los tres trimestres, siendo en este caso necesario aprobarlos todos para tener el módulo aprobado:

- Nota final del primer trimestre: **33,33%**
- Nota final del segundo trimestre: **33,33%**
- Nota final del tercer trimestre: **33,33%**

3.11.3 Instrumentos de evaluación

Como se puede ver en el apartado anterior, los instrumentos de evaluación son la guía de observación, la lista de cotejo, la rúbrica y la escala de valoración. En la mayoría es fácil de inferir el uso que se le da (por ejemplo, la lista de cotejo establece si una tarea

está entregada o no y cada entrega representar un porcentaje equitativo de la nota), pero es necesario explicar el uso de la rúbrica en más profundidad.

Con la rúbrica se evalúa la entrega del proyecto y en ella se establecen los requisitos mínimos para el aprobado, así como los criterios para conseguir más nota. Los mínimos para el aprobado serán ítems muy generales (usar un diseño de clases correcto, que el juego ejecute, etc.) que en total sumarán 5 puntos y que todo proyecto deberá alcanzar para poder aprobar. Los requisitos de ampliación serán más específicos (usar polimorfismo a través de una interfaz, crear una clase estática que se involucre en el loop principal del juego, etc.) y cada uno equivaldrá a una puntuación extra por encima del 5 hasta llegar a 10 puntos al completarlos todos. Es por esta rúbrica que, aunque el alumnado cuenta con gran libertad para hacer muchos tipos de juegos diferentes, el docente se asegura de que están trabajando las Unidades de Trabajo diseñadas; además de ser esta rúbrica el punto de referencia para guiar a los grupos durante las fases de desarrollo (recortando contenido en los juegos más ambiciosos para que completen primero los ítems de la rúbrica y añadiendo complejidad a aquellos juegos que no lleguen a estos mínimos).

3.11.4 Estrategias de recuperación y evaluación extraordinaria

Las tareas, una vez se supera el plazo de entrega no se pueden recuperar y la nota de participación y colaboración es fija una vez establecida; los proyectos y sus defensas en cambio, se pueden recuperar con un examen en convocatoria extraordinaria. La nota obtenida en dicho examen de recuperación equivaldrá a la nota del proyecto y su defensa; será tipo test y tratará desde preguntas teóricas hasta preguntas sobre extractos de código.

Este examen sigue el espíritu del resto de la programación al otorgar al alumnado de un medio alternativo con el que ser evaluados. Pese a ello, si el resto de partes están suspensas con un 0, por pura matemática, solo un 8,33 haría la media suficiente para resultar en aprobado; con ello se fomenta la asistencia, participación, realización de tareas y colaboración en clase incluso para aquellos que prefieran no realizar los proyectos y enfrentarse al examen.

3.11.5 Evaluación de asignatura pendiente

Todo alumno/a que promocione con este módulo pendiente tendrá la oportunidad de entregar y defender los proyectos que tenga suspensos o enfrentarse a los exámenes de recuperación equivalentes (una de las dos cosas). El llegar al aprobado (en todos los proyectos y defensas o en todos los exámenes) supondrá un 5 en el módulo; independientemente de sus notas previas.

3.12 Evaluación docente

Para evaluar el proceso enseñanza-aprendizaje, recoger retroalimentación y mejorar el desempeño en la labor docente, al final del tercer trimestre se le pedirá al alumnado que rellene encuestas y aporte comentarios de manera anónima; reflejando así su punto de vista sobre el módulo y permitiendo adaptar y mejorar el mismo de cara al próximo curso.

Conclusiones

A modo de conclusión, destacar el amplio camino a recorrer en el que este Trabajo de Fin de Máster solo ha dado los primeros pasos. La innovación podría beneficiarse y transformarse con futuras investigaciones; nutrirse de trabajos como el de Fernando Sánchez Cenella, “*Aumento de tiempo de estudio en alumnos mediante la utilización de videojuegos y gamificación*” (2021); y realizar un estudio más profundo de los autores ya mencionados en el apartado de “[Fundamentación teórica](#)”. La programación docente requiere de más trabajo de campo y uso real en las aulas para explorar más a fondo sus fortalezas y debilidades. El trabajo no ha acabado y el trabajo real solo acaba de empezar.

Muchas gracias, querido lector/a, por su tiempo. Espero que la lectura de este documento haya por lo menos suscitado su interés, alentado nuevas ideas y, en el mejor de los casos, le haya inspirado.

Referencias bibliográficas

Investigación

- Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2017). Agile software development methods: Review and analysis. *arXiv preprint arXiv:1709.08439*.
- Alba Pastor, C. (2019). Diseño Universal para el Aprendizaje: un modelo teórico-práctico para una educación inclusiva de calidad. *Participación educativa*.
- Amory, A. (2007). Game object model version II: a theoretical framework for educational game development. *Educational Technology Research and Development*, 55, 51-77.
- Aslan, S., & Balci, O. (2015). GAMED: digital educational game development methodology. *Simulation*, 91(4), 307-319.
- Ausubel, D. (1983). Teoría del aprendizaje significativo. *Fascículos de CEIF*, 1(1-10), 1-10.
- Ernst, J. V., & Clark, A. C. (2012). Fundamental computer science conceptual understandings for high school students using original computer game design.
- Fernández, A., Hombreiro, T. J., Fernández, M. et al. (2022). *Programaciones del Departamento de Informática y Comunicaciones*. Web del IES Doctor Fleming en educastur. Recuperado el 12 de enero de 2023 de https://alojaweb.educastur.es/web/iesdoctorfleming/centro/departamentos/formacion_profesional/informatica_comunicaciones
- Games, A., & Kane, L. (2011, June). Exploring adolescent's STEM learning through scaffolded game design. In *Proceedings of the 6th international conference on foundations of digital games* (pp. 1-8).
- Hombreiro, T. J. et al. (2022). *Documentación Institucional*. Web del IES Doctor Fleming en educastur. Recuperado el 12 de enero de 2023 de https://alojaweb.educastur.es/web/iesdoctorfleming/centro/documentacion_institucional;jsessionid=C809921AC88BDB2C80C944CE06878CCA

Johnson, C., McGill, M., Bouchard, D., Bradshaw, M. K., Bucheli, V. A., Merkle, L. D., ... & Zhang, M. (2016). Game development for computer science education. In *Proceedings of the 2016 ITiCSE Working Group Reports* (pp. 23-44).

McGill, M., Johnson, C., Atlas, J., Bouchard, D., Merkle, L. D., Messom, C., ... & Scott, M. J. (2017, June). Game Development for Computer Science Education. In *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education* (pp. 385-385).

Mojeda Wilce, Y. (2018). El diseño de videojuegos como estrategia pedagógica. Una propuesta para Educación Secundaria.

Newzoo (2022). *2022 Global Games Market*.

<https://www.newzoo.com/globalgamesreport>

Rajagopalan, M., & Schwartz, D. I. (2005, June). Game design and game-development education. In *Phi Kappa Phi Forum* (Vol. 85, No. 2, pp. 29-33). Honor Society of Phi Kappa Phi.

Repenning, A., Webb, D., & Ioannidou, A. (2010, March). Scalable game design and the development of a checklist for getting computational thinking into public schools. In *Proceedings of the 41st ACM technical symposium on Computer science education* (pp. 265-269).

Sánchez Canella, F. (2021). Aumento de tiempo de estudio en alumnos mediante la utilización de videojuegos y gamificación.

Schwaber, K., & Sutherland, J. (2020). La guía definitiva de Scrum: las reglas del juego.

Legislación

Decreto 184/2012, de 8 de agosto, por el que se establece el currículo del ciclo formativo de Grado Superior de Formación Profesional en Desarrollo de Aplicaciones Web. «BOPA» núm. 190, de 16 de agosto de 2012.

<https://sede.asturias.es/bopa/disposiciones/repositorio/LEGISLACION38/66/11/001U004WK20001.pdf>

Ley Orgánica 5/2002, de 19 de junio, de las Cualificaciones y de la Formación Profesional. «BOE» núm. 147, de 20 de junio de 2002.

<https://www.boe.es/eli/es/lo/2002/06/19/5/con>

Ley Orgánica 3/2022, de 31 de marzo, de ordenación e integración de la Formación Profesional. «BOE» núm. 78, de 01 de abril de 2022.

<https://www.boe.es/eli/es/lo/2022/03/31/3/con>

Orden EDU/2887/2010, de 2 de noviembre, por la que se establece el currículo del ciclo formativo de Grado Superior correspondiente al título de Técnico Superior en Desarrollo de Aplicaciones Web. «BOE» núm. 273, de 11 de noviembre de 2010.

<https://www.boe.es/eli/es/o/2010/11/02/edu2887>

Real Decreto 686/2010, de 20 de mayo, por el que se establece el título de Técnico Superior en Desarrollo de Aplicaciones Web y se fijan sus enseñanzas mínimas. «BOE» núm. 143, de 12 de junio de 2010.

<https://www.boe.es/eli/es/rd/2010/05/20/686>

Real Decreto 1147/2011, de 29 de julio, por el que se establece la ordenación general de la formación profesional del sistema educativo. «BOE» núm. 182, de 30 de julio de 2011. <https://www.boe.es/eli/es/rd/2011/07/29/1147>