

Programación Didáctica del módulo "Proyecto Intermodular"

CFGS Desarrollo de Aplicaciones Multiplataforma (DAM) - Comunitat Valenciana

▼ Programación Didáctica del módulo "Proyecto Intermodular"

▼ CFGS Desarrollo de Aplicaciones Multiplataforma (DAM) - Comunitat Valenciana

- ▼ 1. Marco Normativo y Justificación de la Programación
 - 1.1. Justificación y Contextualización
 - 1.2. Contextualización
 - 1.3. Alineación con los Objetivos de Desarrollo Sostenible (ODS)
- ▼ 2. Relación entre los estándares de competencia y los módulos del ciclo formativo
 - 2.1. Objetivos Generales del Proyecto (Metodología S.M.A.R.T.)
 - 2.2. Competencias Profesionales, Personales y Sociales
 - 2.3. Resultados de Aprendizaje (RA) del Módulo "Proyecto Intermodular"
 - 2.4. Matriz de Integración Curricular
- ▼ 3. Metodología del proceso de enseñanza-aprendizaje
 - 3.1. Metodología de Trabajo Ágil (SCRUM)
 - 3.2. Herramientas y Entorno de Trabajo Colaborativo
- ▼ 4. Esquema general y secuenciación de las unidades de programación
 - 4.1. Secuenciación de Actividades o Retos
 - 4.2. Cronograma General por Quincenas (Sprints)
- 5. Evaluación del aprendizaje
 - 5.1. Directrices
- ▼ 5.2. Elementos de la Evaluación
 - 5.2.1 Asistencia en modalidad presencial
 - 5.2.2 Instrumentos de Evaluación
- ▼ 5.3. Criterios de Calificación
 - 5.3.1. Calificación de la Evaluación Continua
 - 5.3.2. Calificación de la Evaluación Final Ordinaria
- ▼ 6. Recursos
 - 6.1. Recursos Materiales
 - 6.2 Uso de espacios y equipamientos
- 8. Medidas de Atención a la Diversidad
- 9. Criterios y procedimientos para la evaluación del desarrollo de la programación y de la práctica docente, así como los criterios de calificación
- 10. Actividades complementarias y extraescolares

1. Marco Normativo y Justificación de la Programación

1.1. Justificación y Contextualización

La presente programación didáctica se ha diseñado para el módulo profesional de "Proyecto Intermodular" del segundo curso del Ciclo Formativo de Grado Superior en Desarrollo de Aplicaciones Multiplataforma (DAM). Su concepción responde y se alinea estrictamente con las directrices de la [Ley Orgánica 3/2022, de 31 de marzo, de ordenación e integración de la Formación Profesional \(LFP\)](#) en sus **artículos del 40 al 45**, así como con su desarrollo normativo a través del [Real Decreto 659/2023](#) en su **artículo 96.3** y la actualización específica del título mediante los Reales Decretos [Real Decreto 405/2023](#) y [Real Decreto 500/2024](#).

Además, se tendrá en cuenta el **Artículo 5** del [Decreto 114/2025, de 29 de julio](#) por el que se establece el currículo de los ciclos formativos de Formación Profesional en la Comunitat Valenciana, asegurando la coherencia con el contexto educativo autonómico.

Este módulo representa una evolución pedagógica fundamental respecto al anterior "Proyecto de Fin de Ciclo". Abandona su carácter eminentemente sumativo y finalista para convertirse en una herramienta formativa **continua, transversal y simultánea** al resto de los módulos del curso. Actúa como el **eje vertebrador** y el **nexo de unión** que materializa en el aula los principios de la LFP, creando un escenario dinámico donde los conocimientos y destrezas de los distintos módulos se aplican, integran y evalúan en tiempo real. El proyecto deja de ser un escaparate final para transformarse en el laboratorio principal de aprendizaje, donde la evaluación es, por su propia naturaleza, integrada y continua.

La programación didáctica se basa en la normativa vigente que regula los Ciclos Formativos de Grado Superior en España y, específicamente, en la Comunidad Valenciana. La normativa más destacada es:

- [Ley Orgánica 3/2022](#), de 31 de marzo, de ordenación e integración de la Formación Profesional.
- [Real Decreto 659/2023](#), de 18 de julio, por el que se establece la ordenación del Sistema de Formación Profesional.
- [La ORDEN 8/2025](#) de 22 de abril (DOGV núm. 10098, 30-04-2025) por la que se regula la evaluación del alumnado de los ciclos formativos de Formación Profesional.
- La [RESOLUCIÓN de 17](#) de julio de 2025 (DOGV núm. 10155, 21-07-2025), por la que se dictan instrucciones sobre ordenación académica y de organización de los centros que impartan los grados D y E de Formación Profesional durante el curso 2025-2026 en la Comunidad Valenciana.
- Guía para la programación del módulo profesional de Proyecto Intermodular en la Comunitat Valenciana. [Punto 8 del documento](#)

1.2. Contextualización

En el **punto 6 del Anexo de la Resolución de 17 de julio de 2025 (DOGV núm. 10155, 21-07-2025)**

Se especifica que:

Esta programación pues, está diseñada para ser implementada en un grupo de segundo curso de DAM en un centro educativo de la Comunitat Valenciana. Se asume un grupo heterogéneo de estudiantes que **trabajarán en equipos de 4 o 5 miembros, fomentando la colaboración, la coordinación, la comunicación y la responsabilidad compartida.**

En este marco, el rol del profesorado evoluciona desde el de un instructor tradicional al de un **"facilitador"** La función principal del docente será guiar a los equipos, proveer de las herramientas metodológicas y técnicas necesarias, practicar la escucha activa y actuar como mentor **para ayudar a los alumnos a superar los retos y alcanzar sus objetivos de forma autónoma.**

1.3. Alineación con los Objetivos de Desarrollo Sostenible (ODS)

Un pilar fundamental del nuevo enfoque pedagógico, y un requisito explícito de esta programación, es el anclaje de todos los proyectos en un marco ético de impacto global: los **Objetivos de Desarrollo Sostenible (ODS)** de las Naciones Unidas Aunque en esta programación no se define un proyecto concreto, se establece como condición indispensable que cada equipo, en su fase de ideación, identifique un desafío social, medioambiental o económico real y alinee su propuesta de solución con uno o varios de los ODS.

Este requisito funciona como el principal mecanismo de filtrado y validación de las ideas de proyecto, integrando desde el inicio las competencias del módulo transversal **Sostenibilidad Aplicada al Sistema Productivo (SOP)** y expandiendo el Resultado de Aprendizaje PROY-RA1 ("Identifica necesidades del sector productivo") para abarcar no solo las demandas del mercado, sino también los grandes retos de la sociedad

2. Relación entre los estándares de competencia y los módulos del ciclo formativo

2.1. Objetivos Generales del Proyecto (Metodología S.M.A.R.T.)

Al inicio del curso, cada equipo deberá definir los objetivos específicos de su proyecto utilizando la metodología S.M.A.R.T. Como guía, los objetivos generales de este módulo son:

- **S - Específico (Specific):** Desarrollar una solución software multiplataforma completa que incluya una aplicación de escritorio (WPF con C# y arquitectura MVVM), una aplicación móvil (Android con Kotlin, Jetpack Compose y arquitectura MVI) y un servicio backend (API REST con Java y seguridad). El sistema estar gestionado mediante un flujo de trabajo profesional (Git, GitHub Projects, Markdown).
- **M - Medible (Measurable):** Implementar el 100% de los casos de uso mínimos definidos en la fase de diseño. Todos los artefactos del proyecto (código, documentación, diario de trabajo) deben estar actualizados semanalmente en el repositorio de GitHub.

- **A - Alcanzable (Achievable):** El alcance tecnológico es ambicioso pero factible en la temporalización asignada de **100 horas lectivas** (3h/semana durante 32-33 semanas) y aproximadamente **64 horas de trabajo autónomo** (2h/semana), utilizando las tecnologías del currículo de 2º de DAM.
- **R - Relevante (Relevant):** El proyecto exige la aplicación e integración práctica de las competencias clave de todos los módulos técnicos y transversales de 2º de DAM, simulando un entorno de desarrollo profesional real.
- **T - Delimitado en el Tiempo (Time-bound):** El proyecto se desarrollará íntegramente a lo largo de **16 quincenas (32 semanas)**, con una entrega final y presentación en la última quincena.

2.2. Competencias Profesionales, Personales y Sociales

A través del desarrollo del proyecto, se trabajarán de forma integrada las competencias definidas en el perfil del título 1:

- **Competencia General:** Desarrollar, implantar, documentar y mantener aplicaciones informáticas multiplataforma, utilizando tecnologías y entornos de desarrollo específicos, garantizando el acceso a los datos de forma segura y cumpliendo los criterios de usabilidad y calidad exigidos.
- **Competencias Transversales:** Se potenciarán de forma explícita las "4C" (Colaboración, Comunicación, Pensamiento Crítico y Creatividad), así como la competencia emprendedora, la digital y la socioemocional, fundamentales en el entorno laboral actual.

2.3. Resultados de Aprendizaje (RA) del Módulo "Proyecto Intermodular"

- **RA1:** Identifica necesidades del sector productivo, relacionándolas con proyectos tipo que puedan satisfacerlas.
- **RA2:** Diseña proyectos relacionados con las competencias expresadas en el título, desarrollando explícitamente las fases que lo componen.
- **RA3:** Planifica la ejecución del proyecto, determinando el plan de intervención y la documentación asociada.
- **RA4:** Define los procedimientos para el seguimiento y control en la ejecución del proyecto, justificando la selección de variables e instrumentos empleados.

2.4. Matriz de Integración Curricular

Esta matriz evidencia cómo cada reto del proyecto permite trabajar y evaluar Resultados de Aprendizaje (RA) de múltiples módulos de forma simultánea.

Actividad / Criterio de Evaluación	PI	AD	DI	PSP	PMDM	SGE	DAS	SOP	IPE II
Definición del proyecto, ODS y casos de uso	RA1, RA2						RA1	RA2, RA4	RA3, RA4

Actividad / Criterio de Evaluación	PI	AD	DI	PSP	PMDM	SGE	DAS	SOP	IPE II
Diseño del modelo (Java) y diagrama (PlantUML)	RA2	RA3	RA6				RA4		
Transpilación de clases a C# y Kotlin con IA	RA3, RA4				RA1		RA4		
Prototipado de UI/UX (Figma, Material 3, XAML)	RA2, RA4		RA1, RA2, RA4		RA1, RA2		RA4		RA3
Implementación de API REST (ORM, JPA, BD)	RA3, RA4	RA2, RA3, RA6		RA3, RA4		RA3	RA3	RA5	RA2
Implementación de seguridad (Sesiones, JWT)	RA3, RA4	RA2		RA5			RA5		
Desarrollo de frontend de escritorio (WPF, MVVM, C#)	RA3, RA4		RA1, RA3, RA8			RA4			RA2
Desarrollo de frontend móvil (Android, Compose, MVI, Kotlin)	RA3, RA4		RA1, RA4, RA8		RA1, RA2, RA3				RA2
Integración de frontends con la API REST	RA3, RA4	RA6	RA8	RA3	RA2				
(Opcional) Despliegue en la nube con CI/CD	RA3		RA7	RA4			RA3, RA6	RA5	
Gestión ágil del proyecto	RA3, RA4								RA2, RA3

Actividad / Criterio de Evaluación	PI	AD	DI	PSP	PMDM	SGE	DAS	SOP	IPE II
(SCRUM, GitHub Projects)									
Elaboración de memoria final y presentación	RA4		RA6						RA1, RA5

3. Metodología del proceso de enseñanza-aprendizaje

3.1. Metodología de Trabajo Ágil (SCRUM)

El proyecto se **planteará por retos**, siguiendo el marco de trabajo ágil **SCRUM**, por ser el estándar de facto en la industria del desarrollo de software y por su adecuación a los principios pedagógicos de flexibilidad, entrega iterativa y mejora continua.

- **Sprints:** Cada quincena lectiva constituirá un **Sprint** de dos semanas de duración. El proyecto se completará en un total de **16 sprints**.
- **Ceremonias SCRUM:** Se llevarán a cabo de **forma flexible** las ceremonias clave:
 - **Sprint Planning:** Al inicio de cada quincena, para seleccionar las tareas del Product Backlog que se abordarán.
 - **Daily Stand-up:** Al comienzo de cada sesión de trabajo en el aula, para sincronizar al equipo. Máximo **5 minutos**.
 - **Sprint Review:** Al final de cada quincena, para demostrar el incremento de producto funcional.
 - **Sprint Retrospective:** Tras la Review, para reflexionar sobre el proceso y proponer mejoras para el siguiente sprint.
- **Roles:** El profesor actuará como **Product Owner**, definiendo los requisitos generales y el marco de trabajo. Los equipos de alumnos constituirán el **Development Team**. Para fomentar el liderazgo, el rol de **Scrum Master** podrá ser rotativo entre los miembros del equipo en cada sprint.

3.2. Herramientas y Entorno de Trabajo Colaborativo

- **GitHub:** Será la plataforma central para el control de versiones del código fuente. Se exigirá un uso correcto de un flujo de trabajo de ramas (ej. GitFlow simplificado), la realización de **Pull Requests** para la integración de código y la revisión de código entre compañeros como parte del proceso de calidad.
- **GitHub Projects:** Se utilizará para la gestión del **Product Backlog** y los **Sprint Backlogs**. Cada equipo mantendrá un tablero Kanban para visualizar el estado de las tareas (To Do, In Progress, Done), proporcionando transparencia sobre el progreso del proyecto.
- **Documentación en Markdown:** Toda la documentación del proyecto (definición, análisis, diseño, manuales, memoria) se generará en formato **Markdown** y se alojará en el propio repositorio de

GitHub. Se recomienda el uso de extensiones como **Markdown Preview Enhanced** para la inclusión de diagramas (PlantUML), fórmulas y otros elementos enriquecidos.

- **Diario de Trabajo Individual:** Cada **alumno** deberá mantener un fichero a modo de diario de trabajo denominado **NOMBRE_APELLIDO.md** en la raíz del repositorio de proyecto. En este diario, se registrará diariamente las tareas realizadas, el tiempo dedicado, los problemas encontrados, las soluciones aplicadas y una reflexión sobre su aprendizaje.

Ejemplo:

Fecha	Horas	Tareas Realizadas	Problemas Encontrados	Soluciones Aplicadas	Reflexión
29/09	2	Haciendo Pull Request GitHub	Definir ramas de origen y destino	Preguntando al profesor	Aprendí que para hacer un pull request mi rama local debe estar subida aGitHub

Este documento es un instrumento fundamental para la evaluación individual y **solo se tendrá en cuenta si se ha llevado al día.**

4. Esquema general y secuenciación de las unidades de programación

4.1. Secuenciación de Actividades o Retos

La secuencia sigue un **flujo lógico de retos** que abarca desde la formación inicial en herramientas y metodologías, pasando por el diseño e implementación de las distintas capas del sistema, hasta la integración final y la entrega del proyecto.

- **Actividad 1: Presentación:** Presentación del módulo y presentación del grupo utilizando dinámicas de grupo para fomentar la cohesión y el trabajo en equipo.
- **Actividad 2: Formación de equipos** Formación de equipos con técnica **H.A.D.A.** (Hacedor, Analista, Divergente, Armonizador).
- **Actividad 3: Taller de Git y GitHub** Instalación VSCode, Creación de la organización y primer repositorio en GitHub. Definición de flujos de trabajo con ramas y Pull Requests.
- **Actividad 4: Definición del proyecto y ODS.** Cada equipo selecciona un desafío social y define su proyecto alineado con uno o varios ODS a partir de una plantilla usando una LLM (Gemini, ChatGPT, Perplexity, DeepSeek. etc.). Se crea un repositorio en la organización y en un fichero **PROYECTO.md** se documentan los casos de uso y los objetivos S.M.A.R.T. (Specific, Measurable, Achievable, Relevant, Time-bound).
- **Actividad 5: Gestión Ágil con SCRUM** Repaso ceremonias SCRUM y configuración de GitHub Projects para la gestión del Product Backlog y los Sprint Backlogs. Definen un Backlog inicial y planifican el primer sprint.
- **Actividad 6: Diseño del Modelo de Dominio (Sprint 1)** Sprint Planning 1 y uso de IA generativa para ayudar en el diseño del modelo de dominio. Diseño del modelo de dominio en **C#** y creación del diagrama de clases con **PlantUML**. Documentación en **DISEÑO.md**.
- **Actividad 7: Implementación del Modelo de Dominio (Sprint 2)** Sprint Planning 2 y transpilación de las clases Java a **Java** y **Kotlin** usando IA generativa. Creación de Mocks de datos e imágenes para pruebas con IA.
- **Actividad 8: Prototipado UI/UX (Sprint 3)** Sprint Planning 3 y prototipado de UI/UX para Android (Material 3) y Escritorio (XAML) con **Figma/Penpot** y herramientas de IA como Google Stitch AI o Lovable.
- **Actividad 9: Frontend Escritorio (Sprint 4)** Sprint Planning 4. Implementación de la UI de la aplicación de admin con **WPF**.
- **Actividad 10: Frontend Escritorio (Sprint 5)** Sprint Planning 5. Definición arquitectura **MVVM** vinculando VM a los datos mock.
- **Actividad 11: Frontend Móvil (Sprint 6)** Sprint Planning 6. Implementación de la UI de la aplicación de cliente con **Android (Jetpack Compose)**.
- **Actividad 12: Frontend Móvil (Sprint 7)** Sprint Planning 7. Definición arquitectura **MVI** vinculando estados a los datos mock.
- **Actividad 13: Backend - API REST (Sprint 8)** Sprint Planning 8. Implementación de API REST con algún framework de Java. Definición de entidades y sus relaciones con a partir del modelo y

generación base de datos siguiendo el esquema Entity-First o viceversa utilizando algún ORM (JPA/Hibernate o JPA/Eclipselink).

- **Actividad 14: Backend - API REST (Sprint 9)** Sprint Planning 9. Implementación de la lógica de negocio en los servicios de la API.
- **Actividad 15: Backend - Seguridad (Sprint 10)** Sprint Planning 10. Implementación de autenticación y autorización con **Sesiones** o **JWT**. Gestión de usuarios y claves cifradas en la BD y protección de endpoints por roles.
- **Actividad 16: Backend - Seguridad (Sprint 11)** Sprint Planning 11. Documentación de la API y sustitución de datos mock por el consumo real de la API REST en las aplicaciones de Escritorio (C#)
- **Actividad 17: Integración Full-Stack (Sprint 12)** Sprint Planning 12. Sustitución de datos mock por el consumo real de la API REST en la aplicación de Android (Kotlin).
- **Actividad 18: Avanzado y Entrega Final (Sprint 13)** Sprint Planning 13. Elaboración de la memoria final, vídeo y preparación de la presentación.

4.2. Cronograma General por Quincenas (Sprints)

Sprint	Semanas	Fechas	Objetivo Principal del Sprint	Entregables Clave
	1-2	08 sep 19 sep	Fundación y Metodología	Creación de equipos con técnicas H.A.D.A.
	3-4	22 sep 03 oct	Git y GitHub y Organización Definición del Proyecto y ODS	Documento de casos de uso y objetivos S.M.A.R.T. (MD).
	5-6	06 oct 17 oct	GitHub Project y Scrum	Documentación SCRUM en MD utilizando GitHub Projects.
1	7-8	20 oct 31 oct	IA y Modelo de Dominio (C#)	Diagrama de clases (PlantUML). Clases del modelo en C#.
2	9-10	03 nov 14 nov	Transpilación y Mocks con IA	Refinamientos, Clases en Kotlin y Java, ficheros de mocks de datos.
3	11-12 (1º Ev)	17 nov 05 dic	Prototipado UI/UX	Prototipos de UI/UX en Figma/Penpot.
4	13-14	08 dic 19 dic	Frontend Escritorio (WPF+MVVM)	Aplicación WPF con vistas y ViewModels, usando datos mock.
5	15-16	07 ene 16 ene	Frontend Escritorio (Funcional)	Lógica de negocio y navegación implementada en la app de escritorio.

Sprint	Semanas	Fechas	Objetivo Principal del Sprint	Entregables Clave
6	17-18	19 ene 30 ene	Frontend Móvil (Android+MVI)	App Android con pantallas y estados (Compose), usando datos mock.
7	19-20	02 feb 13 feb	Frontend Móvil (Funcional)	Lógica de negocio y navegación implementada en la app móvil.
8	(2º Ev) 21-22	23 feb 06 mar	Backend: API REST (CRUD)	Definición de las entidades en el ORM y la BD, con las operaciones CRUD necesarias.
9	23-24	09 mar 20 mar	Backend: API REST (Lógica)	Lógica de negocio implementada en los servicios de la API.
FE	FE	FE	FE	FE
10	25-26	23 mar 24 abr	Backend: Seguridad y autorización	API con endpoints de login y registro más endpoints protegidos por roles.
11	27-28	27 abr 08 may	Documentación API e Integración Full-Stack Escritorio	Documentar la API con Swagger y conectar la App de Escritorio al API Rest de forma segura.
12	29-30	11 may 22 may	Integración Full-Stack Android	Conectar la App de Android a la API REST de forma segura.
13	31-32	25 may 05 jun	Avanzado y Entrega Final	(Opcional) CI/CD. Memoria final, vídeo y presentación.

5. Evaluación del aprendizaje

5.1. Directrices

Como con cualquier otra metodología, tenemos que evaluar la consecución de los resultados de aprendizaje y sus criterios de evaluación asociados, además de las competencias transversales que determinemos. El éxito de la evaluación dependerá en parte de la capacidad de los docentes para saber desgranar al máximo los RA y CE y saber asociar a cada uno de ellos el procedimiento de evaluación o evidencia que nos indique si el alumno ha sido capaz de conseguirlos.

La evaluación debe ser continua y formativa para servir al alumnado a aprender de sus propios errores a medida que avanza en el desarrollo del proceso de enseñanzaaprendizaje.

La evaluación debe tener en cuenta si se va a evaluar el trabajo individual o de grupo. Además, deben indicarse los criterios de calificación y la ponderación de cada una de las tareas realizadas sobre la nota final.

También es conveniente explicar desde el principio los derechos y las obligaciones de las partes respecto

a la evaluación y las posibles reclamaciones.

La manera de poder desarrollar la evaluación del Proyecto intermodular está detallada en el apartado 4 de la guía del PCCF, y más concretamente en el apartado 4.4.1. sobre evaluación parcialmente intermodular.

5.2. Elementos de la Evaluación

La evaluación del módulo será **continua, formativa e integradora**, combinando la valoración del producto final con el seguimiento del proceso y las contribuciones individuales.

5.2.1 Asistencia en modalidad presencial

La asistencia a clase y el trabajo diario tanto en clase como en casa, es requisito fundamental para llevar a cabo el proceso de **evaluación continua**, por lo que, aplicando la legislación vigente a tal efecto, **la no asistencia al 85% o más de las horas asignadas al módulo o la entrega de menos del 50% de las prácticas propuestas** por el profesor, conlleva la pérdida de este derecho.

El incumplimiento de dicho requisito supondrá la pérdida del derecho a la evaluación continua en el módulo, manteniendo el alumno su derecho a ser evaluado en la evaluación final ordinaria mediante la entrega de los trabajos no presentados.

Por otra parte, la inasistencia injustificada (o con justificación improcedente) al 15 por ciento o más de las horas totales correspondientes a los módulos en los que se haya matriculado el alumno podrá conllevar la anulación automática de la matrícula. También podrá ser motivo de anulación de matrícula la inasistencia injustificada durante 10 días consecutivos.

Se considera pues que la **asistencia a clase** es una base fundamental para la evaluación, ya que es muy importante el trabajo diario del alumno para plantear, resolver y justificar los **ejercicios individuales o colectivos propuestos** por el profesor con ayuda de las explicaciones, los apuntes, la información disponible en Internet y material de apoyo.

5.2.2 Instrumentos de Evaluación

1. Las actividades y retos grupales guiados iniciales, se evaluarán en la plataforma Aules **de forma individual**, mediante la calificación de **No Hecho** / **Hecho** según el **entregable especificado** en cada caso y la participación del miembro del equipo al que se está evaluando.
2. En algún caso donde haya muchos conceptos o definiciones como en la teoría de SCRUM, se podrá utilizar un **cuestionario tipo test individualizado** en la plataforma Aules para evaluar la comprensión de dichos conceptos.
3. A partir de cierto momento el **diario de trabajo individual** será un componente clave para evaluar la contribución y el aprendizaje de cada alumno. Se revisará cada quincena para asegurar que se está utilizando correctamente.
4. Cada Sprint culminará con una **Sprint Review** donde el equipo presentará el incremento de producto desarrollado. Esta presentación será evaluada. Por el profesor y, en algunos casos, por los compañeros, utilizando las siguientes rúbricas:

Rúbrica evaluación de cada Sprint Review (Grupal):

Criterio de Evaluación	Insuficiente	Suficiente	Excelente
Funcionalidad Incremento	No cumple Producto Mínimo Viable (MVP).	Cumple el MVP con errores o funcionalidades incompletas	Cumple el MVP y algunas funcionalidades avanzadas sin errores.
Han realizado los eventos de Scrum	No han realizado eventos SCRUM.	Han realizado algunos eventos SCRUM, pero de forma incompleta o incorrecta.	Han realizado todos los eventos SCRUM correctamente.
Han seguido el Workflow de Scrum usando GitHub Projects	No han utilizado GitHub Projects.	Han utilizado GitHub Projects, pero de forma incompleta o incorrecta.	Han utilizado GitHub Projects correctamente para gestionar el sprint con errores leves.

Rúbrica evaluación de cada Sprint Review (Individual):

Criterio de Evaluación	Insuficiente	Suficiente	Excelente
Participación en el grupo	No ha asistido con regularidad a las sesiones de trabajo en equipo.	Ha asistido a la mayoría de las sesiones, pero con poca participación activa.	Ha asistido a todas las sesiones y ha participado activamente en discusiones, tareas y eventos
Registro en Diario de Trabajo	El diario está vacío, incompleto o las entradas son superficiales.	El diario se actualiza, pero con poca regularidad. Las descripciones son breves y poco reflexivas.	El diario se actualiza casi diariamente con detalle, explicando tareas, problemas, soluciones y reflexiones profundas.
Participación en el Incremento	No ha contribuido al desarrollo del incremento de producto o al workflow de GitHub.	Ha contribuido de forma limitada al desarrollo del incremento y al workflow de GitHub.	Ha contribuido significativamente al desarrollo del incremento y ha seguido correctamente el workflow de GitHub.
Defensa Individual	No ha participado en la presentación del sprint o no ha	Ha participado en la presentación, pero	Ha participado activamente en la presentación,

Criterio de Evaluación	Insuficiente	Suficiente	Excelente
	defendido su trabajo.	con poca preparación o claridad.	defendiendo su trabajo con claridad y confianza.

5.3. Criterios de Calificación

La calificación final del módulo se calculará ponderando diferentes bloques para equilibrar la valoración grupal e individual. El número de actividades y retos evaluados grupalmente e individualmente dependerá del ritmo de trabajo de cada equipo y del número de actividades/retos entregados hasta el momento de la evaluación.

5.3.1. Calificación de la Evaluación Continua

La **calificación de la evaluación continua y los resultados de aprendizaje** se realizará en cada una de las tres evaluaciones parciales, y la nota final del módulo será la suma ponderada de las tres evaluaciones parciales tal y como se indica en la siguiente tabla:

Bloque de Evaluación	Ponderación	Criterios de Evaluación	RA del Módulo Relacionados
1ª Evaluación	35%	60% media simple de las calificaciones individuales en aules. 40% media simple de las calificaciones grupales en aules	RA1 100% RA2 100%
2ª Evaluación	35%	60% media simple de las calificaciones individuales en aules. 40% media simple de las calificaciones grupales en aules	RA3 60% RA4 60%
3ª Evaluación	30%	40% media simple de las calificaciones individuales en aules. 60% media simple de las calificaciones grupales en aules.	RA3 40% RA4 40%

5.3.2. Calificación de la Evaluación Final Ordinaria

6. Recursos

6.1. Recursos Materiales

- **Aula de Informática:** Equipamiento con ordenadores, software de desarrollo (Visual Studio Code, NetBeans, Android Studio, Git, etc), acceso a internet y pizarra digital.
- **Plataforma GitHub:** Para control de versiones, gestión de proyectos SCRUM y documentación.
- **Herramientas de Prototipado:** Figma o Penpot para diseño UI/UX.
- **Herramientas de IA Generativa:** Acceso a modelos como ChatGPT, Gemini, Kimi, Claude, Stitch, etc., para apoyo en diseño y codificación.
- **Documentación y Recursos Online:** Acceso a documentación oficial de las tecnologías utilizadas, tutoriales y foros de desarrollo.
- **Material Didáctico:** Guías, plantillas y ejemplos proporcionados por el profesor.
- **Recursos Audiovisuales:** Vídeos y tutoriales en línea para complementar el aprendizaje.
- **Espacios de Colaboración:** Plataformas Microsoft Teams.
- **Recursos de Evaluación:** Plantillas de rúbricas, cuestionarios, etc.
- **Soporte Técnico:** Profesores de los módulos involucrados para resolver problemas con hardware o software durante el desarrollo del proyecto.

6.2 Uso de espacios y equipamientos

El desarrollo del módulo se llevará a cabo principalmente en el aula de informática del centro, equipada con los recursos materiales mencionados anteriormente. Se fomentará el uso de espacios colaborativos dentro del aula para facilitar el trabajo en equipo y la comunicación entre los miembros de cada grupo. Además, **se podrá reservar el Aula Innova del centro** para actividades específicas como presentaciones o sesiones de brainstorming.

8. Medidas de Atención a la Diversidad

- **Medidas de Ampliación:** Para el alumnado con mayor capacidad se propondrán retos adicionales como:
 - Profundizar en las tareas opcionales (CI/CD).
 - Despliegue en la nube con servicios como AWS, Azure o Google Cloud.
 - Diseño de interfaces más complejas o accesibles.
 - Investigar y proponer arquitecturas o tecnologías alternativas.
 - Asumir un rol de mentoría técnica dentro del equipo.
- **Medidas de Refuerzo:** Para el alumnado que encuentre dificultades se ofrecerán apoyos como:
 - **Plantillas de código** y configuraciones base más detalladas.
 - Talleres específicos sobre las tecnologías más complejas.
 - Guías paso a paso para las tareas críticas.
 - Priorización del trabajo en el Producto Mínimo Viable (MVP).

- **Flexibilidad Metodológica:** La propia naturaleza de SCRUM **permite adaptar la carga de trabajo** de cada sprint a la capacidad real del equipo. La formación de grupos mediante la técnica H.A.D.A. busca crear equipos equilibrados.

9. Criterios y procedimientos para la evaluación del desarrollo de la programación y de la práctica docente, así como los criterios de calificación

La mejora continua es fundamental. Por ello, se establecerá un proceso para evaluar y refinar esta programación y la práctica docente asociada.

- **Indicadores de Evaluación:**
 - Tasa de éxito del alumnado en la consecución de los RA.
 - Calidad y complejidad de los proyectos finales.
 - Feedback recogido en las retrospectivas de cada sprint.
 - Análisis de las dificultades más comunes a través de los diarios de trabajo.
- **Instrumentos de Evaluación:**
 - Encuestas anónimas al alumnado al finalizar el módulo.
 - Diario de sesiones del profesor, registrando incidencias y áreas de mejora.
 - Reuniones de coordinación con el resto del profesorado del equipo docente de 2º de DAM.
- **Proceso de Mejora:** Los datos recogidos se analizarán al final del curso académico para identificar fortalezas y debilidades. Las conclusiones servirán para realizar ajustes en la secuenciación, proponer nuevas herramientas o refinar los instrumentos de evaluación para el siguiente curso, asegurando que la programación se mantenga actualizada y sea lo más efectiva posible.

10. Actividades complementarias y extraescolares

El presente módulo **no contempla actividades complementarias o extraescolares específicas**, dado su enfoque práctico y la carga de trabajo integrada en el horario lectivo. Sin embargo, se fomentará la participación en eventos relacionados con el desarrollo de software, como hackathons, Campeonatos de FP Skills de la Comunitat Valenciana, meetups tecnológicos o conferencias, siempre que sean compatibles con el calendario académico y las responsabilidades del alumnado.