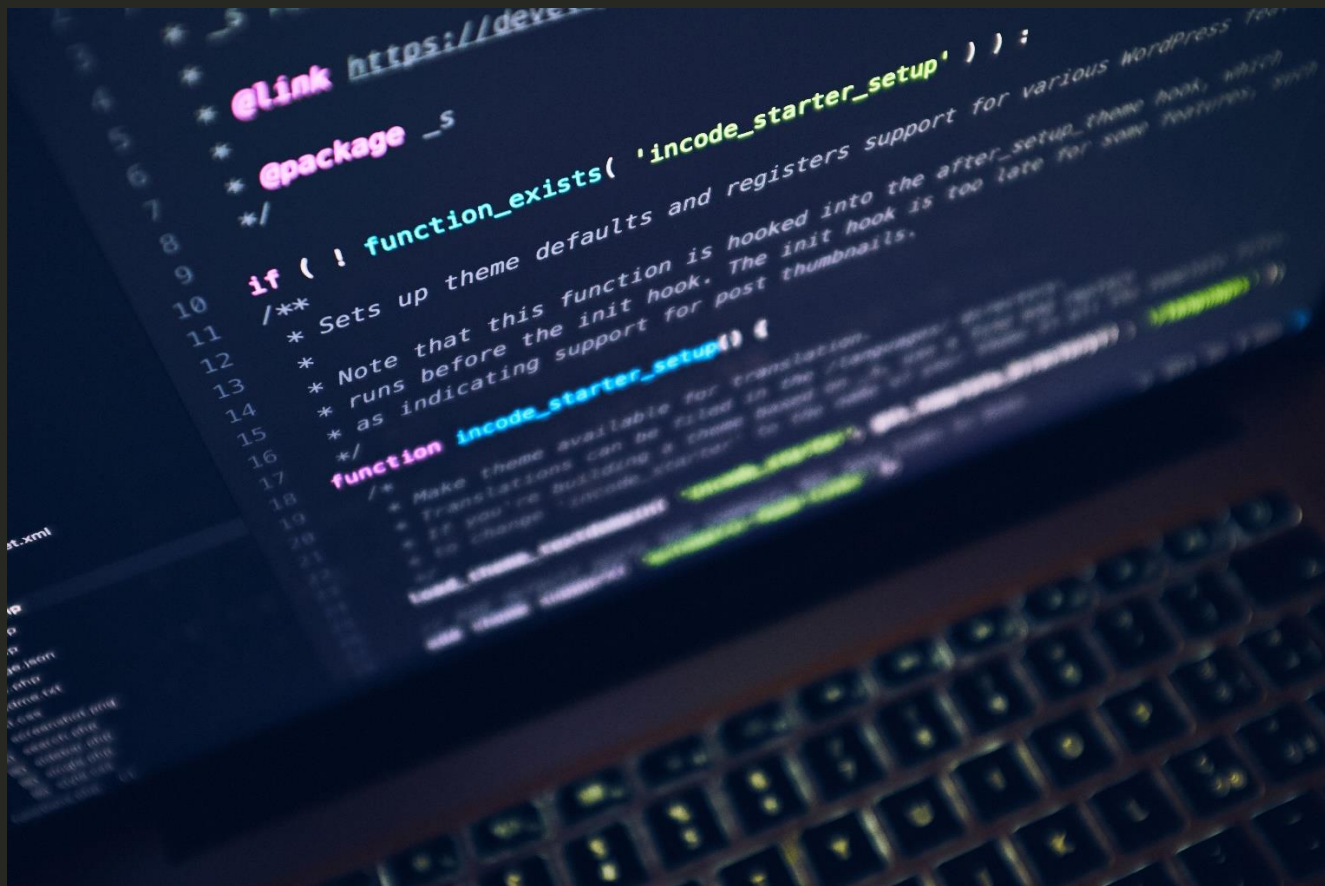


PROGRAMACIÓN DIDÁCTICA
CURSO 2022/23

ENTORNOS DE DESARROLLO

1º CFGS TÉCNICO EN DESARROLLO DE APLICACIONES WEB
Informática y Comunicaciones



ASPIRANTE

DAVID HORMIGO RAMÍREZ

DNI

77453948E

NÚMERO DE ASPIRANTE

29003350

CUERPO

PROFESORES DE ENSEÑANZA SECUNDARIA (590)

ESPECIALIDAD

INFORMATICA (107)

NÚMERO DE TRIBUNAL

7

El entorno de desarrollo es el conjunto de herramientas y recursos necesarios para desarrollar una aplicación.

El entorno de desarrollo se compone de:

- Herramientas de desarrollo: IDE, compilador, gestor de paquetes, etc.

- Recursos: librerías, frameworks, etc.

El entorno de desarrollo debe ser capaz de:

- Gestionar los recursos necesarios para el desarrollo.

- Ejecutar y depurar el código.

- Gestionar los cambios en el código.

- Gestionar los errores.

- Gestionar los recursos de red.

- Gestionar los recursos de almacenamiento.

- Gestionar los recursos de red.

- Gestionar los recursos de almacenamiento.

- Gestionar los recursos de red.

- Gestionar los recursos de almacenamiento.

- Gestionar los recursos de red.

- Gestionar los recursos de almacenamiento.

- Gestionar los recursos de red.

- Gestionar los recursos de almacenamiento.

- Gestionar los recursos de red.

- Gestionar los recursos de almacenamiento.

- Gestionar los recursos de red.

- Gestionar los recursos de almacenamiento.

- Gestionar los recursos de red.

- Gestionar los recursos de almacenamiento.

Contenido

Identificación y contexto	1
Identificación del módulo profesional	1
Unidades de competencia y cualificaciones profesionales	1
Marco legislativo completo	2
Contextualización del módulo dentro de su entorno productivo	3
Contextualización del Centro	3
Contextualización de los profesionales del desarrollo de aplicaciones web	3
Objetivos y Competencias	4
Objetivos generales del título	4
Competencias	6
Competencia general	6
Competencias profesionales, personales y sociales	6
Resultados de aprendizaje y criterios de evaluación	8
Unidades didácticas	10
Contenidos básicos del módulo	10
Contenidos Interdisciplinares	11
Contenidos transversales	12
Temporalización y secuenciación	13
Elementos organizados por unidades didácticas	14
Unidad Didáctica 1. Introducción al desarrollo de software	14
Unidad Didáctica 2. Instalación y uso de entornos de desarrollo integrado	16
Unidad Didáctica 3. Ciclo de vida del software. Metodologías	19
Unidad Didáctica 4. Defectos de software y depuración	21
Unidad Didáctica 5. Control de versiones, repositorios remoto y colaboración	23
Unidad Didáctica 6. Diseño orientado a objetos. Diagramas de comportamiento	25
Unidad Didáctica 7. Diseño orientado a objetos. Diagramas estructurales	27
Unidad Didáctica 8. Introducción a las pruebas de software	30
Unidad Didáctica 9. Diseño e implementación de pruebas unitarias	32
Unidad Didáctica 10. Diseño e implementación de pruebas de integración	34
Unidad Didáctica 11. Documentación de software	35
Unidad Didáctica 12. Código limpio y refactorización de código	36
Bloques de contenido	38
Resultados de aprendizaje, criterios de evaluación y unidades didácticas	38
Metodología	41

Metodología de aula.....	41
Agrupamientos del alumnado	41
Instrumentos de seguimiento	42
Tipos de instrumento de evaluación.....	42
Recursos didácticos y materiales.....	43
Material docente en el aula y equipos técnicos	43
Software específico	43
Usos de las TIC y herramientas de trabajo colaborativo	44
Organización de los espacios	44
Evaluación y calificación	44
Proceso de evaluación por resultados de aprendizaje.....	45
Momentos de evaluación	46
Ponderación de cada resultado de aprendizaje en la nota del módulo	46
Ponderación de los criterios de evaluación de cada resultado de aprendizaje	47
Ponderación de la relación entre resultados de aprendizaje, criterios de evaluación y actividades evaluables	49
Mecanismos de recuperación y mejora de la calificación	52
Autoevaluación.....	52
De la programación didáctica.....	52
De la labor docente	52
Evaluación por parte del alumnado	52
Evaluación por parte del propio docente.....	53
Atención a la diversidad	53
Atención a los alumnos con necesidades específicas de apoyo educativo ..	53
Atención a los diferentes ritmos de aprendizaje.....	54
Actividades complementarias y extraescolares.....	54
Bibliografía	54
Anexos	i
Anexo I. Indicadores de evaluación de la actividad docente.....	i
Anexo II. Formulario autoevaluación programación didáctica	ii

Índice de tablas

Tabla 1. Identificación del módulo profesional	1
Tabla 2. Temporalización y secuenciación UD	13
Tabla 3. Bloques conceptuales del módulo	38
Tabla 4. Relación RA, CE y UD	41
Tabla 5. Requisitos espacios del título	44
Tabla 6. ponderación RA en la nota del módulo	47
Tabla 7. Ponderación de los criterios de evaluación por cada RA.....	49
Tabla 8. Ponderación entre RA, CE y actividades evaluables	51

Identificación y contexto

Identificación del módulo profesional

El presente documento desarrolla la programación didáctica para el módulo profesional de Entornos de Desarrollo.

Familia Profesional	Informática y Comunicaciones
Nivel	Ciclo Formativo de Grado Superior
Título	Desarrollo de aplicaciones web
Nombre del módulo	Entornos de desarrollo
Código	0487
Equivalencia en créditos ECTS	6
Curso	Primero
Horas totales	96
Horas semanales	3
Referente en la Clasificación Internacional Normalizada de la educación	Nivel 5

Tabla 1. Identificación del módulo profesional

Unidades de competencia y cualificaciones profesionales

Según el Real Decreto 686/2010, el título tiene incluye las siguientes cualificaciones profesionales y unidades de competencia:

1. Cualificaciones profesionales completas:
Desarrollo de aplicaciones con tecnologías Web **IFC154_3** 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.
2. Cualificaciones profesionales incompletas:
 - a. Programación en lenguajes estructurados de aplicaciones de gestión **IFC155_3**.
UC0223_3 Configurar y explotar sistemas informáticos.
UC0226_3 Programar bases de datos relacionales.
 - b. Programación con lenguajes orientados a objetos y bases de datos relacionales **IFC 080_3**.
UC0223_3 Configurar y explotar sistemas informáticos.
UC0226_3 Programar bases de datos relacionales.

Aunque, según la orden de 16 de junio de 2011, el módulo profesional de Entornos de Desarrollo no está asociado directamente a cualificaciones profesionales este si contribuye a alcanzar las unidades de competencias **UC0491_3**, **UC0492_3** y **UC0493_3**, ya que posibilita el manejo efectivo de

entornos y herramientas para el desarrollo de software, la gestión de documentación y la implementación de pruebas de software

Marco legislativo completo

La elaboración de la programación del módulo profesional **de Entornos de Desarrollo**, perteneciente al Ciclo Formativo de Grado Superior de **Desarrollo de Aplicaciones Web**, se ha realizado de acuerdo con lo establecido en la normativa que se detalla a continuación:

A nivel **estatal**:

Ley Orgánica 3/2020, de 29 de diciembre, por la que se modifica la Ley Orgánica 2/2006, de 3 de mayo, de Educación.

Ley Orgánica 2/2006, de 3 de mayo, de Educación.

Ley Orgánica 5/2002, de 19 de junio, de las Cualificaciones y de la Formación Profesional, mediante la que se crea el Sistema Nacional de Cualificaciones Profesionales, cuyo instrumento fundamental es el Catálogo General de Cualificaciones Profesionales.

Real Decreto 1147/2011, de 29 de julio, por el que se establece la ordenación general de la formación profesional del sistema educativo (BOE 30/07/2011)

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.

Ley Orgánica 3/2022, de 31 de marzo, de ordenación e integración de la Formación Profesional (BOE 01-04-2022).

A nivel **autonómico**:

Ley 17/2007, de 10 de diciembre, de Educación de Andalucía (LEA).

Decreto 436/2008, de 2 de septiembre, por el que se establece la ordenación y las enseñanzas de la Formación Profesional inicial que forma parte del sistema educativo.

Orden de 29 de septiembre de 2010, por la que se regula la evaluación, certificación, acreditación y titulación académica del alumnado que cursa enseñanzas de formación profesional inicial que forma parte del sistema educativo en la Comunidad Autónoma de Andalucía.

Decreto 334/2009 de 22 de septiembre, por el que se regulan los centros integrados de formación profesional en la Comunidad Autónoma de Andalucía (BOJA 05-10-2009).

Decreto 382/2010 de 13 de octubre, por el que se modifica el Decreto 334/2009, de 22 de septiembre, por el que se regulan los centros integrados de Formación Profesional en la Comunidad Autónoma de Andalucía, para adecuarlo a la Directiva 2006/123/CE del Parlamento Europeo y del Consejo, de 12 de diciembre de 2006, relativa a los servicios en el mercado interior (BOJA 28-10-2010).

Orden de 16 de junio de 2011, por la que se desarrolla el currículo correspondiente al título de Técnico Superior en Desarrollo de Aplicaciones Web.

Contextualización del módulo dentro de su entorno productivo

Contextualización del Centro

En el desarrollo de la presente programación didáctica no sólo se ha tenido en cuenta las características particulares del centro y el entorno económico-social de la zona donde se sitúa.

El Centro Publico Integrado de Formación Profesional Alan Turing se encuentra ubicado en el Parque Tecnología de Andalucía (Málaga TechPark), uno de los principales polos de trabajo en el sector de las nuevas tecnologías en Andalucía. Esta ubicación marca las relaciones del centro con el tejido productivo de sus alrededores y establece sinergias importantísimas con empresas punteras en las nuevas tecnologías, influyendo notablemente en la planificación de las enseñanzas y la adecuación de los contenidos.

Como CPFIP, las enseñanzas impartidas en el centro son exclusivamente de Formación Profesional en la familia de informática y comunicaciones y en la familia de electricidad y electrónica. Dentro de la familia de informática y comunicaciones, se imparten el CFGM de Técnico en Sistemas Microinformáticos y Redes, el CFGS de Técnico Superior en Administración de Sistemas Informáticos en Red, Técnico Superior en Desarrollo de Aplicaciones Multiplataforma, Técnico Superior en Desarrollo de Aplicaciones Web, Curso de Especialización en Inteligencia Artificial y Big Data, Curso de Especialización en Ciberseguridad en Entornos de las Tecnologías de la Información y el Curso de Especialización en Desarrollo de videojuegos y realidad virtual. Esta amplia oferta de enseñanzas especializadas y su ubicación en pleno Málaga TechPark atrae alumnado de toda la geografía andaluza.

Dicho parque empresarial se encuentra ubicado en el distrito de Campanillas de la ciudad de Málaga. En este distrito podemos encontrar desde barrios muy desfavorecidos como son Los Asperones a otros acomodados como El Brillante. Esto, junto con las características intrínsecas del Málaga TechPark, tiene como consecuencia una ampliación en la diversidad socioeconómica del alumnado que recibe el centro.

Contextualización de los profesionales del desarrollo de aplicaciones web

Este módulo profesional contiene parte de la formación necesaria para desempeñar la función de desarrollador de aplicaciones.

La función de desarrollador de aplicaciones incluye aspectos como:

- La utilización de las herramientas software disponibles.
- La elaboración de documentación interna y técnica de la aplicación.
- La elaboración y ejecución de pruebas.
- La optimización de código.

Los avances en las tecnologías de la información y la comunicación han hecho que exista una demanda muy alta de profesionales del desarrollo de aplicaciones. Ya en 2017, la Comisión Europea (CBI, 2022) pronosticó un déficit de medio millón de profesionales del desarrollo de software para el año 2020. De esta manera, la empleabilidad de los egresados en el título es muy alta.

Sin embargo, la familia profesional de informática y comunicaciones es la tercera más demandada en Andalucía (FP, 2023) pero existe una importante brecha de género ya que solo el 17% del alumnado de estas familias son mujeres.

Dentro del ciclo formativo de grado superior en Desarrollo de Aplicaciones Web, el módulo de **Entornos de Desarrollo** aporta conocimientos y competencias necesarios para el uso efectivo de las herramientas específicas para el desarrollo de software, la construcción de software fiable y la gestión de documentación de software.

La evolución actual del desarrollo de software hace vital no solo que los profesionales sean competentes en el uso de las herramientas específicas de su práctica laboral, sino que sean capaces de desarrollar software fiable mediante el desarrollo de pruebas de software automáticas. Además, la actual tendencia laboral de equipos de desarrollo remotos y descentralizados hace que las habilidades de utilizar mecanismos eficaces de colaboración, el autoaprendizaje, generación de documentación y la de gestión de versiones hace que las competencias a las que contribuye el módulo sean esenciales para la práctica laboral.

Objetivos y Competencias

Objetivos generales del título

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 de este ciclo formativo 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.
- o) Verificar los componentes de software desarrollados, analizando las especificaciones, para completar el plan de pruebas.
- p) Utilizar herramientas específicas, cumpliendo los estándares establecidos, para elaborar y mantener la documentación de los procesos.
- q) Establecer procedimientos, verificando su funcionalidad, para desplegar y distribuir aplicaciones.
- r) Programar y realizar actividades para gestionar el mantenimiento de los recursos informáticos.
- s) 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.
- t) 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.
- u) 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.
- v) Desarrollar técnicas de liderazgo, motivación, supervisión y comunicación en contextos de trabajo en grupo para facilitar la organización y coordinación de equipos de trabajo.
- w) 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.
- bb) 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.

Según esta misma Orden, el módulo de Entornos de desarrollo contribuye a alcanzar los objetivos d, e, h, i y j.

Competencias

Competencia general

El Real Decreto 686/2010, de 20 de mayo, en su artículo 4 establece que 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.

El módulo profesional de Entornos de Desarrollo contribuye a alcanzar dicha competencia general a través de formar en el uso efectivo de herramientas desarrollo y la creación de software de calidad y fiable.

Competencias profesionales, personales y sociales

El Real Decreto 686/2010, de 20 de mayo, en su artículo 5 establece que 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.
- o) Desplegar y distribuir aplicaciones web en distintos ámbitos de implantación, verificando su comportamiento y realizando modificaciones.
- p) Gestionar y/o realizar el mantenimiento de los recursos de su área en función de las cargas de trabajo y el plan de mantenimiento.
- q) 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.
- r) 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.
- s) 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.
- t) 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.
- u) 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.
- v) 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.

- w) 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.

Según este mismo Real Decreto, el módulo de Entornos de Desarrollo contribuye a alcanzar las competencias profesionales, personales y sociales d, f, h, i y j.

Resultados de aprendizaje y criterios de evaluación

Según la Orden del 16 de julio de 2011, los resultados de aprendizaje del módulo de **Entornos de Desarrollo** son:

RA1. Reconoce los elementos y herramientas que intervienen en el desarrollo de un programa informático, analizando sus características y las fases en las que actúan hasta llegar a su puesta en funcionamiento.

Criterios de evaluación:

- a) Se ha reconocido la relación de los programas con los componentes del sistema informático, memoria, procesador, periféricos, entre otros.
- b) Se han clasificado los lenguajes de programación.
- c) Se han diferenciado los conceptos de código fuente, objeto y ejecutable.
- d) Se han reconocido las características de la generación de código intermedio para su ejecución en máquinas virtuales.
- e) Se ha evaluado la funcionalidad ofrecida por las herramientas utilizadas en programación.
- f) Se ha diferenciado el funcionamiento de los distintos tipos de traductores de lenguajes ante el código fuente de un programa.
- g) Se han identificado las fases de desarrollo de una aplicación informática.

RA2. Evalúa entornos integrados de desarrollo analizando sus características para editar código fuente y generar ejecutable.

Criterios de evaluación:

- a) Se han instalado entornos de desarrollo, propietarios y libres.
- b) Se han añadido y eliminado módulos en el entorno de desarrollo.
- c) Se ha personalizado y automatizado el entorno de desarrollo.
- d) Se ha configurado el sistema de actualización del entorno de desarrollo.
- e) Se han generado ejecutables a partir de código fuente de diferentes lenguajes en un mismo entorno de desarrollo.
- f) Se han generado ejecutables a partir de un mismo código fuente con varios entornos de desarrollo.
- g) Se han identificado las características comunes y específicas de diversos entornos de desarrollo.
- h) Se han identificado las funciones más usuales de las herramientas CASE para el desarrollo, prueba y documentación de código.

RA3. Verifica el funcionamiento de programas diseñando y realizando pruebas.

Criterios de evaluación:

- a) Se han identificado los diferentes tipos de pruebas.
- b) Se han definido casos de prueba.
- c) Se han identificado las herramientas de depuración y prueba de aplicaciones ofrecidas por el entorno de desarrollo.
- d) Se han utilizado herramientas de depuración para definir puntos de ruptura y seguimiento.
- e) Se han utilizado las herramientas de depuración para examinar y modificar el comportamiento de un programa en tiempo de ejecución.
- f) Se ha documentado el plan de pruebas.
- g) Se han efectuado pruebas unitarias de clases y funciones.
- h) Se han efectuado pruebas de integración, de sistema y de aceptación.
- i) Se han implementado pruebas automáticas.
- j) Se han documentado las incidencias detectadas.
- k) Se han aplicado normas de calidad a los procedimientos de desarrollo de software.
- l) Se han realizado medidas de calidad sobre el software desarrollado.

RA4. Optimiza código empleando las herramientas disponibles en el entorno de desarrollo.

Criterios de evaluación:

- a) Se han identificado los patrones de refactorización más usuales.
- b) Se han elaborado las pruebas asociadas a la refactorización.
- c) Se ha revisado el código fuente usando un analizador de código.
- d) Se han identificado las posibilidades de configuración de un analizador de código.
- e) Se han aplicado patrones de refactorización con las herramientas que proporciona el entorno de desarrollo.
- f) Se ha realizado el control de versiones integrado en el entorno de desarrollo.
- g) Se ha documentado el código fuente mediante comentarios.
- h) Se han utilizado herramientas del entorno de desarrollo para documentar los procesos, datos y eventos.
- i) Se han utilizado herramientas del entorno de desarrollo para documentar las clases.

RA5. Genera diagramas de clases valorando su importancia en el desarrollo de aplicaciones y empleando las herramientas disponibles en el entorno.

Criterios de evaluación:

- a) Se han identificado los conceptos básicos de la programación orientada a objetos.
- b) Se ha instalado el módulo del entorno integrado de desarrollo que permite la utilización de diagramas de clases.
- c) Se han identificado las herramientas para la elaboración de diagramas de clases.
- d) Se ha interpretado el significado de diagramas de clases.

- e) Se han trazado diagramas de clases a partir de las especificaciones de las mismas.
- f) Se ha generado código a partir de un diagrama de clases.
- g) Se ha generado un diagrama de clases mediante ingeniería inversa.

RA6. Genera diagramas de comportamiento valorando su importancia en el desarrollo de aplicaciones y empleando las herramientas disponibles en el entorno.

Criterios de evaluación:

- a) Se han identificado los distintos tipos de diagramas de comportamiento.
- b) Se ha reconocido el significado de los diagramas de casos de uso.
- c) Se han interpretado diagramas de interacción.
- d) Se han elaborado diagramas de interacción sencillos.
- e) Se han interpretado diagramas de estados.
- f) Se han planteado diagramas de estados sencillos.
- g) Se ha interpretado el significado de diagramas de actividades.
- h) Se han elaborado diagramas de actividades sencillos.

Unidades didácticas

Contenidos básicos del módulo

Según la **Orden del 16 de julio de 2011** los contenidos del módulo profesional de **Entornos de Desarrollo** son los siguientes:

1. Reconocimiento de elementos del desarrollo de software:
 - 1.1. Conceptos de programa informático y de aplicación informática.
 - 1.2. Concepto de lenguaje de programación.
 - 1.3. Tipos de lenguajes de programación.
 - 1.4. Características de los lenguajes más difundidos.
 - 1.5. Código fuente, código objeto y código ejecutable; máquinas virtuales.
 - 1.6. Proceso de obtención de código ejecutable a partir del código fuente; herramientas implicadas.
 - 1.6.1. Traductores de lenguajes.
 - 1.6.2. Depuradores.
 - 1.7. Fases del desarrollo de una aplicación, análisis, diseño, codificación, pruebas, documentación, explotación y mantenimiento, entre otras.
2. Evaluación de entornos integrados de desarrollo:
 - 2.1. Funciones de un entorno de desarrollo.
 - 2.2. Componentes de un entorno de desarrollo.
 - 2.3. Instalación de un entorno de desarrollo.
 - 2.4. Mecanismo de actualización de un entorno de desarrollo.
 - 2.5. Uso básico de un entorno de desarrollo.
 - 2.5.1. Edición de programas.
 - 2.5.2. Generación de ejecutables.
 - 2.6. Entornos de desarrollo libres y comerciales más usuales.
 - 2.7. Uso de herramientas CASE en el desarrollo de software.
3. Diseño y realización de pruebas:
 - 3.1. Tipos de pruebas, funcionales, estructurales, regresión.
 - 3.2. Procedimientos y casos de prueba.

- 3.3. Pruebas de Código, cubrimiento, valores límite, clases de equivalencia.
- 3.4. Herramientas de depuración de código.
- 3.5. Planificación de Pruebas.
 - 3.5.1. Pruebas unitarias; herramientas.
 - 3.5.2. Pruebas de integración.
 - 3.5.3. Pruebas del sistema.
 - 3.5.4. Pruebas de aceptación.
 - 3.5.5. Automatización de pruebas.
- 3.6. Calidad del software.
 - 3.6.1. Normas y certificaciones.
 - 3.6.2. Medidas de calidad del software.
- 4. Optimización y documentación:
 - 4.1. Refactorización.
 - 4.1.1. Concepto y limitaciones.
 - 4.1.2. Patrones de refactorización más usuales.
 - 4.1.3. Refactorización y pruebas.
 - 4.1.4. Herramientas de ayuda a la refactorización.
 - 4.2. Control de versiones.
 - 4.2.1. Concepto y características.
 - 4.2.2. Tipos.
 - 4.2.3. Herramientas.
 - 4.2.4. Repositorio.
 - 4.3. Documentación.
 - 4.3.1. Uso de comentarios.
 - 4.3.2. Alternativas.
 - 4.3.3. Herramientas.
- 5. Elaboración de diagramas de clases:
 - 5.1. Notación de los diagramas de clases.
 - 5.1.1. Clases. Atributos, métodos y visibilidad.
 - 5.1.2. Objetos. Instanciación.
 - 5.1.3. Relaciones. Herencia, composición, agregación, asociación y uso.
 - 5.2. Herramientas para la elaboración de diagramas de clases. Instalación.
 - 5.3. Generación de código a partir de diagramas de clases.
 - 5.4. Generación de diagramas de clases a partir de código.
- 6. Elaboración de diagramas de comportamiento:
 - 6.1. Tipos. Campo de aplicación.
 - 6.2. Diagramas de casos de uso. Actores, escenario, relación de comunicación.
 - 6.3. Diagramas de interacción.
 - 6.3.1. Diagramas de secuencia. Línea de vida de un objeto, activación, envío de mensajes.
 - 6.3.2. Diagramas de colaboración. Objetos, mensajes.
 - 6.3.3. Diagramas de estados. Estados, eventos, señales, transiciones.
 - 6.3.4. Diagramas de actividades. Actividades, transiciones, decisiones y combinaciones.

Contenidos Interdisciplinares

El módulo profesional de Entornos de Desarrollo está íntimamente ligado al módulo profesional de Programación (0485) y es vital realizar una coordinación

para la consecución de objetivos y el desarrollo de competencias. Ambos módulos comparten ciertos contenidos como son:

- Entornos integrados de desarrollo
 - Definición y tipos. Entornos comerciales y de Software libre.
 - Instalación y descripción de entornos integrados de desarrollo.
 - Creación de proyectos. Estructura y componentes.
- Entornos de desarrollo para programación orientada a objetos.
 - Entornos específicos.
 - Plugins de integración en entornos genéricos.
- Depuración de programas.
 - El depurador como herramienta de control de errores.
 - Documentación de programas.
- Documentación interna, comentarios.
- Documentación externa, diagramas de clases, requisitos, guías, etc.

Es por ello por lo que ciertos contenidos de este módulo han sido secuenciados de tal manera que ambos módulos se benefician.

De esta manera las unidades didácticas 4 y 5 se han planificado para ser impartidas durante la primera evaluación. El principal motivo es que el alumnado se beneficia de un conocimiento del control de versiones para la entrega de actividades y las técnicas de depuración aportan una técnica primordial para la creación de software funcional.

A su vez, las unidades 6 y 7 serán trabajadas al principio de la segunda evaluación para reforzar los conceptos sobre programación orientadas a objetos.

Finalmente, los contenidos impartidos en el módulo de Lenguajes de Marcas y Sistemas de Información sobre el uso de HTML y CSS se han tenido en cuenta a la hora de planificar la unidad didáctica 11. Es esencial que el alumnado tenga conocimientos sobre selectores CSS para el desarrollo de la unidad.

Contenidos transversales

De acuerdo con los objetivos de la formación profesional inicial establecidos en el artículo 40 de la Ley Orgánica 2/2006, de 3 de mayo, de Educación, con los artículos 2 y 3 del Real Decreto 1147/2011, de 29 de julio, por el que se establece la ordenación general de la formación profesional del sistema educativo, y con el artículo 3.2 del Decreto 436/2008, de 2 de septiembre, por el que se establece la ordenación y las enseñanzas de la Formación Profesional inicial que forma parte del sistema educativo, las enseñanzas de formación profesional, además de tener por objeto conseguir que el alumnado adquiera las competencias profesionales, personales y sociales correspondientes, también tendrán por objeto la adquisición de otras competencias relacionadas con la **educación en valores éticos, personales, profesionales y sociales** que preparen a los alumnos y a las alumnas para asumir una vida responsable en una sociedad libre y democrática.

En la presente programación didáctica se tomarán en consideración como elementos transversales el aprendizaje y fortalecimiento de los siguientes valores:

1. Consolidar hábitos de disciplina, trabajo individual y en equipo, así como capacidades de autoaprendizaje
2. Utilizar las tecnologías de la información y la comunicación como fuente de aprendizaje y el uso de las lenguas extranjeras necesarias en el desarrollo de software
3. Entender la necesidad de estar en constante cambio y renovación de conocimientos para el ejercicio efectivo de una profesión
4. Valorar las ventajas e inconvenientes

Temporalización y secuenciación

Según el calendario escolar oficial, para el curso 2022/2023 se disponen de 102 horas lectivas en la localidad de Málaga para el módulo de Entornos de Desarrollo. En la Tabla 2 se puede ver la distribución horaria de cada unidad didáctica y su temporalización por evaluaciones.

Evaluación	Unidad didáctica	Horas
Primera	1. Introducción al desarrollo de software	6 horas
	2. Instalación y uso de entornos integrados de desarrollo	9 horas
	3. Ciclo de vida del software. Metodologías	3 horas
	4. Defectos de software y depuración	9 horas
	5. Control de versiones, repositorios remotos y colaboración	12 horas
		39 horas
Segunda	6. Diseño orientado a objetos. Diagramas de comportamiento	9 horas
	7. Diseño orientado a objetos. Diagramas estructurales	9 horas
	8. Introducción a las pruebas de software	6 horas
	9. Diseño e implementación de pruebas unitarias	15 horas
		39 horas
Tercera	10. Diseño e implementación de pruebas de integración	15 horas
	11. Documentación de software	3 horas
	12. Código limpio y refactorización	6 horas
		24 horas

Tabla 2. Temporalización y secuenciación UD

Elementos organizados por unidades didácticas

Unidad Didáctica 1. Introducción al desarrollo de software

1		Introducción al desarrollo de software	6 Horas
Contenidos	1.1. ¿Qué es el desarrollo del software? 1.2. Lenguajes de programación compilados e interpretados 1.2.1. Ejecutar un programa en Python 1.2.2. Proceso de compilación en Java 1.3. Instalar un compilador par Java 1.3.1 Instalar Java Development Toolkit 1.4. Proceso de compilación en Java.	Contenidos básicos	
		1.1-1.6.1	
		Contenidos Transversales	
		Autonomía en el trabajo. Uso de las TIC. Interés de ampliación. Visión crítica.	
Objetivos de aprendizaje			
1. Entender los conceptos básicos del proceso de desarrollo de software 2. Conocer las diferencias y características de los lenguajes de programación 3. Entender el proceso de compilación y sus particularidades en función del lenguaje 4. Reconocer los elementos implicados en el proceso de desarrollo de software			
Objetivos generales			
Competencias profesionales, personales y sociales			

Resultados de aprendizaje	RA1. Reconoce los elementos y herramientas que intervienen en el desarrollo de un programa informático, analizando sus características y las fases en las que actúan hasta llegar a su puesta en funcionamiento	Criterios de evaluación	a) Se ha reconocido la relación de los programas con los componentes del sistema informático, memoria, procesador, periféricos, entre otros. b) Se han clasificado los lenguajes de programación. c) Se han diferenciado los conceptos de código fuente, objeto y ejecutable. d) Se han reconocido las características de la generación de código intermedio para su ejecución en máquinas virtuales. f) Se ha diferenciado el funcionamiento de los distintos tipos de traductores de lenguajes ante el código fuente de un programa.	
Actividades				
Tipo	Actividad	Objetivos	Actores	Tiempo
Motv.	¿Cómo le decimos a un ordenador que hacer?	1,3,4	Alumnos-Profesor	½ h.
Ev. In.	Cuestionario de conocimientos previos	1,2,3,4	Alumnos-Profesor	½ h.
Des.	Impartición UD 1	1,2,3,4	Alumnos Profesor.	2 h.
Des.	Como realizar un mapa conceptual con Miro		Alumnos-Profesor	¼ h.
Des.	Clasificar lenguajes de programación	2	Alumnos	½ h
Des.	Crear y usar contenedores para desarrollo		Alumnos-Profesor	¼ h.
Des.	Ejecutar “Hola Mundo!” en Python	1,2,3,4	Alumnos-Profesor	½ h.
Des.	Compilar “Hola Mundo!” en Java	1,2,3,4	Alumnos	½ h.
Eva.	Prueba escrita UD 1	12,3,4	Alumnos	½ h
Eva.	Test de comprobación de los conocimientos.	1,2,3,4	Alumnos	½ h.
Activades evaluables				
#	Actividad	Instrumento		
1.1	Clasificar lenguajes de programación	Rubrica		
1.2	Ejecutar “Hola Mundo” en Python	Lista de control		

1.3	Compilar “Hola Mundo” en Java	Lista de control
1.4	Prueba escrita UD 1	Solucionario

Unidad Didáctica 2. Instalación y uso de entornos de desarrollo integrado

2		Introducción al desarrollo de software		9 Horas	
Contenidos		Contenidos básicos			
		2			
		Contenidos Transversales			
		Autonomía en el trabajo. Uso de las TIC. Interés de ampliación. Visión crítica.			
Objetivos de aprendizaje					
1. Seleccionar e Instalar Entornos de Desarrollo Integrados en función de las necesidades y características del lenguaje de programación 2. Configurar el entorno de desarrollo para a las necesidades del proceso de desarrollo 3. Identificar las herramientas esenciales accesibles desde un IDE 4. Generar código ejecutable desde el IDE					
Objetivos generales					
Competencias profesionales, personales y sociales					

Resultados de aprendizaje	RA2. Evalúa entornos integrados de desarrollo analizando sus características para editar código fuente y generar ejecutable.	Criterios de evaluación	a) Se ha reconocido la relación de los programas con los componentes del sistema informático, memoria, procesador, periféricos, entre otros. b) Se han clasificado los lenguajes de programación. c) Se han diferenciado los conceptos de código fuente, objeto y ejecutable. d) Se han reconocido las características de la generación de código intermedio para su ejecución en máquinas virtuales. f) Se ha diferenciado el funcionamiento de los distintos tipos de traductores de lenguajes ante el código fuente de un programa.		
Actividades					
Tipo	Actividad	Objetivos	Actores	Tiempo	
Mot.	¿Conoces las herramientas de trabajo de un programador?	1	Profesor, Alumnos	½ h	
	Impartición UD2		Profesor, Alumnos	2 h	
Des.	Búsqueda/indagación sobre IDEs			½ h	
Des.	Instalación Eclipse		Profesor	¼ h	
Des.	Instalación Visual Studio Code		Alumnos	½ h	
Des.	Instalación IntelliJ IDEA		Alumnos	½ h	
Des.	Adaptar el tema en IntelliJ IDEA		Alumnos, profesor	½ h	
Des.	¿Para qué sirven las extensiones y complementos en un IDE?		Alumnos, Profesor	½ h	
Des	Instalar extensiones para desarrollo Java en VS Code y Actualizar el IDE		Alumnos	½ h	
Des.	Compilar “Hola Mundo!” en Java desde IDEs		Alumnos	½ h	
Des	¿Puedes compilar “Hola Mundo!” en Go desde VS Code?		Alumnos	½ h	
Des	Prueba escrita UD 2		Alumnos	½ h	
R/A	Se adaptará a las necesidades del alumnado.	1,2,3,4	Alumno.	1 h	
Explicit. Eva.	Test de comprobación de los conocimientos.	1,2,3,4	Alumnos	½ h	

Activades evaluables		
#	Actividad	Instrumento
2.1	Instalación Visual Studio Code	Lista de control
2.2	Instalación IntelliJ IDEA	Lista de control
2.3	Adaptar el tema en IntelliJ IDEA	Lista de control
2.4	Instalar extensiones para desarrollo Java en VS Code y Actualizar el IDE	Lista de control
2.5	Compilar "Hola Mundo!" en Java desde IDEs	Lista de control
2.6	¿Puedes compilar "Hola Mundo!" en Go desde VS Code?	Lista de control
2.7	Prueba escrita UD 2	Solucionario

Unidad Didáctica 3. Ciclo de vida del software. Metodologías

3		Ciclo de vida del software. Metodologías	3 Horas
Contenidos	3.1. De la artesanía a la ingeniería en el desarrollo de software 3.2. Ciclo de vida. Fases comunes del desarrollo de software 3.3. Metodologías clásicas 3.3.1 Ciclo de vida en cascada 3.3.2 Ciclo de vida en espiral 3.4. Metodologías ágiles 3.4.1. Crítica a las metodologías clásicas. El manifiesto ágil 3.4.2. Scrum 3.4.3. Kanban 3.4.4. Desarrollo guiado por pruebas	Contenidos básicos	
		1.7	
		Contenidos Transversales	
		Autonomía en el trabajo. Uso de las TIC. Interés de ampliación. Visión crítica.	
Objetivos de aprendizaje			
1. Valorar la importancia de la implantación de metodologías de desarrollo para la creación de software 2. Describir las fases del ciclo de vida del software 3. Resumir las principales características de las metodologías de desarrollo 4. Analizar y comparar metodologías de desarrollo de software			
Objetivos generales			
Competencias profesionales, personales y sociales			

Resultados de aprendizaje	RA1. Reconoce los elementos y herramientas que intervienen en el desarrollo de un programa informático, analizando sus características y las fases en las que actúan hasta llegar a su puesta en funcionamiento	Criterios de evaluación			
	g) Se han identificado las fases de desarrollo de una aplicación informática				
Actividades					
Tipo	Actividad	Objetivos	Actores	Tiempo	
Motv.	Charla-Debate ¿Conoces las tareas diarias de un programador?	1	Alumno, profesor	½ h	
Des.	Impartición UD 3	1,2,3,4	Alumno, profesor	1 h	
Des.	Crear Workspace en Trello	2,4	Alumno	½ h	
Des.	Prueba escrita UD3	1,2,3,4	Alumno	½ h	
R/A	Se adaptará a las necesidades del alumnado	1,2,3,4	Alumno	½ h	
Activades evaluables					
#	Actividad	Instrumento			
2.1	Crear Workspace en Trello	Rúbrica			
2.2	Prueba escrita UD 3	Solucionario			

Unidad Didáctica 4. Defectos de software y depuración

4		Defectos de software y depuración		9 Horas	
Contenidos	4.1. El ciclo de vida de los fallos de software 4.2. El concepto de “bug” 4.3. Tipos de errores de software 4.4. Errores en tiempo de compilación 4.4.1. Uso de analizadores estáticos para el análisis de errores de compilación 4.5. Errores en tiempo de ejecución 4.5.1. Excepciones 5. Uso del depurador en un IDE 5.1 Puntos de ruptura, edición de variables y control del flujo			Contenidos básicos	
				1.6.2, 3.4	
				Contenidos Transversales	
				Autonomía en el trabajo. Uso de las TIC. Interés de ampliación. Visión crítica.	
Objetivos de aprendizaje					
1. Analizar la naturaleza falible del desarrollo de software 2. Diferenciar entre errores de tiempo de compilación y errores de ejecución 3. Diagnosticar el origen de errores en tiempo de compilación mediante el uso de las herramientas de un IDE 4. Utilizar el depurador de software como medida de análisis y corrección de errores					
Objetivos generales					
Competencias profesionales, personales y sociales					
Resultados de aprendizaje	RA3. Verifica el funcionamiento de programas diseñando y realizando pruebas	CE	c) Se han identificado las herramientas de depuración y prueba de aplicaciones ofrecidas por el entorno de desarrollo. d) Se han utilizado herramientas de depuración para definir puntos de ruptura y seguimiento. e) Se han utilizado las herramientas de depuración para examinar y modificar el comportamiento de un programa en tiempo de ejecución.		
	RA4. Optimiza código empleando las herramientas disponibles en el entorno de desarrollo	CE	c) Se ha revisado el código fuente usando un analizador de código. d) Se han identificado las posibilidades de configuración de un analizador de código.		
Actividades					

Tipo	Actividad	Objetivos	Actores	Tiempo
Motv.	¿Se puede programar software libre de fallos?	1	Alumnos, profesor	½ h
Des.	Impartición UD4	1,2,3,4	Alumnos, profesor	2 h
Des.	Corrige y explica errores estáticos	1,2,3	Alumnos	½ h
Intr.	¿Puedes encontrar el bug?	1,2,4	Alumnos, Profesor	½ h
Des.	Corrige y explicar errores en tiempo de ejecución	1,2,4	Alumnos, Profesor	½ h
Des	Cuestionario UD4	1,2,3,4	Alumnos	½ h
R/A	Se adaptará a las necesidades del alumnado	1,2,3,4	Alumno	1 h
Eva.	Test de comprobación de los conocimientos.	1,2,3,4	Alumnos	½ h.
Activades evaluables				
#	Actividad	Instrumento(s)		
4.1	Corrige y explica errores estáticos	Lista de control		
4.2	Corrige y explica errores en tiempo de ejecución	Lista de control		
4.3	Cuestionario UD4	Solucionario		

Unidad Didáctica 5. Control de versiones, repositorios remoto y colaboración

5		UD5. Control de versiones, repositorios remoto y colaboración		12 Horas	
Contenidos	5.1. La importancia del control de versiones 5.2. Instalación de git 5.3. Uso básico de git como herramienta de líneas de comandos 5.3.1. Comandos básicos 5.4. Uso de git desde el IDE 5.5. El desarrollo cómo tarea colaborativa 5.6. Trabajar con repositorios remotos 5.6.1. Github 5.6.2. Comandos trabajo con remote 5.6.3 Colisiones 5.7 Trabajo con ramas 5.7.1. Comandos creación y cambiar ramas 5.7.2. Comandos fusión ramas 5.8. Flujos de trabajo más comunes 5.9. Estrategias para deshacer cambios 5.9.1. Opciones para deshacer cambios	Contenidos básicos			
		4.2			
		Contenidos Transversales			
		Autonomía en el trabajo. Uso de las TIC. Interés de ampliación. Visión crítica.			
Objetivos de aprendizaje					
1. Analizar las ventajas del uso de un sistema de control de versiones en el proceso de desarrollo 2. Ejecutar las principales tareas de control de versiones desde línea de comandos y desde un IDE 3. Usar repositorios de código para la implementación de software de manera colaborativa 4. Identificar diferentes flujos de trabajo a la hora de coordinar versiones de código					
Objetivos generales					
Competencias profesionales, personales y sociales					
Resultados de aprendizaje	RA4. Optimiza código empleando las herramientas disponibles en el entorno de desarrollo.		Criterios de evaluación	f) Se ha realizado el control de versiones integrado en el entorno de desarrollo.	

Actividades				
Tipo	Actividad	Objetivos	Actores	Tiempo
Motv.	¿Cómo gestionas las versiones de un trabajo?	1	Alumnos, Profesor	½ h
Des	Impartición UD5	1,2,3,4	Alumnos, Profesor	3 ½ h
Des	Instalar y configurar git	2	Alumnos	½ h
Des	Crear repositorio y primer commit desde la línea de comandos	2,3	Alumnos	½ h
Des	Trabajar en el repositorio desde el IDE	2,3	Alumnos	½ h
Des	Subir el código a un repositorio de github	2,3	Alumnos	½ h
Des	Clonar el repositorio de un compañero	2,3,4	Alumnos	1 h
Des	Crear y sincronizar ramas	2,3,4	Alumnos	1 h
Des	Deshacer el ultimo commit	1,2,3,4	Alumnos	½ h
Des	Cuestionario UD5	1,2,3,4	Alumnos	½ h
R/A	Se adaptará a las necesidades del alumnado	1,2,3,4	Alumno	2 ½ h
Eva.	Test de comprobación de los conocimientos.	1,2,3,4	Alumnos	½ h.
Activades evaluables				
#	Actividad	Instrumento		
5.1	Instalar y configurar git	Lista de control		
5.2	Crear repositorio y primer commit desde la línea de comandos	Lista de control		
5.3	Trabajar en el repositorio desde el IDE	Lista de control		
5.4	Subir el código a un repositorio de github	Lista de control		
5.5	Clonar el repositorio de un compañero	Lista de control		
5.6	Crear y sincronizar ramas	Rúbrica		
5.7	Deshacer el ultimo commit	Lista de control		
5.8	Prueba escrita UD5	Solucionario		

Unidad Didáctica 6. Diseño orientado a objetos. Diagramas de comportamiento

6		Diseño orientado a objetos. Diagramas de comportamiento		12 Horas	
Contenidos	<div>1. Definición elementos dinámicos del sistema</div> <div>2. Tipos de diagramas de comportamiento</div> <div>3. Diagramas de casos de uso</div> <div>3.1. Utilidad de los casos de uso</div> <div>3.2 Elementos de un diagrama de caso de uso</div> <div>5. Diagramas de actividad</div> <div>4.1 Utilidad de los diagramas de actividad</div> <div>4.2 Elementos de un diagrama de actividad</div> <div>4. Diagramas de secuencia</div> <div>4.1 Utilidad de los diagramas de secuencia</div> <div>4.2 Elementos de un diagrama de secuencia</div> <div>5. Diagramas de estado</div> <div>5.1 Utilidad de los diagramas de estado</div> <div>5.2 Elemento de un diagrama de estado</div> <div>6. Herramientas para la elaboración de diagramas de comportamiento</div>	Contenidos básicos			
		6			
		Contenidos Transversales			
		Autonomía en el trabajo. Uso de las TIC. Interés de ampliación. Visión crítica.			
Objetivos de aprendizaje					
<div>1. Identificar la utilidad de representar el comportamiento del sistema mediante diagramas</div> <div>2. Interpretar los principales diagramas de comportamiento para obtener información sobre el sistema</div> <div>3. Elaborar diagramas de comportamiento a partir de especificaciones de software</div> <div>4. Seleccionar herramientas para el desarrollo de diagramas de comportamiento</div>					
Objetivos generales					
Competencias profesionales, personales y sociales					

Resultados de aprendizaje	RA6. Genera diagramas de comportamiento valorando su importancia en el desarrollo de aplicaciones y empleando las herramientas disponibles en el entorno.	Criterios de evaluación	a) Se han identificado los distintos tipos de diagramas de comportamiento. b) Se ha reconocido el significado de los diagramas de casos de uso. c) Se han interpretado diagramas de interacción. d) Se han elaborado diagramas de interacción sencillos. e) Se han interpretado diagramas de estados. f) Se han planteado diagramas de estados sencillos. g) Se ha interpretado el significado de diagramas de actividades. h) Se han elaborado diagramas de actividades sencillos.	
Actividades				
Tipo	Actividad	Objetivos	Actores	Tiempo
Activades evaluables				
#	Actividad	Instrumento		
6.1	¿Qué podemos conocer a partir de los casos de uso?	Entrevista		
6.2	Interpretando un diagrama de secuencia	Rúbrica		
6.3	Crear un diagrama de secuencia para el proyecto integrador	Rúbrica		
6.4	Interpretando un diagrama de actividad para el proyecto integrador	Rúbrica		
6.5	Completar el diagrama de actividad	Rúbrica		
6.6	Interpretando un diagrama de estados para el proyecto	Rúbrica		
6.7	Completar el diagrama de actividad para el proyecto	Rúbrica		
6.8	Cuestionario UD 6	Solucionario		

Unidad Didáctica 7. Diseño orientado a objetos. Diagramas estructurales

7		Diseño orientado a objetos. Diagramas estructurales		9 Horas	
Contenidos	7.1. Conceptos y ventajas del diseño orientado a objetos 7.2. Diagramas de clases. 7.3. Elementos de un diagrama de clases: clases, atributos, métodos 7.4. Relaciones en un diagrama de clases: herencia, asociación, composición y agregación 7.5. Elaboración de un diagrama de clases a partir de requisitos 7.6. Elaboración de diagramas de clases a partir de código 7.7. Escribir código a partir de un diagrama de clases 7.8. Herramientas para la realización de diagramas de clases	Contenidos básicos			
		5			
		Contenidos Transversales			
		Autonomía en el trabajo. Uso de las TIC. Interés de ampliación. Visión crítica.			
Objetivos de aprendizaje					
1. Conceptualizar el modelado de software orientado a objetos 2. Diseñar software para lenguajes orientados a objetos siguiendo buenas practicas de ingeniería de software 3. Manejar herramientas específicas para el tratamiento de diagramas estructurales 4. Interpretar diagramas estructurales como especificación de software					
Objetivos generales					
Competencias profesionales, personales y sociales					

Resultados de aprendizaje	RA5. Genera diagramas de clases valorando su importancia en el desarrollo de aplicaciones y empleando las herramientas disponibles en el entorno.	Criterios de evaluación	a) Se han identificado los conceptos básicos de la programación orientada a objetos.	
			b) Se ha instalado el módulo del entorno integrado de desarrollo que permite la utilización de diagramas de clases.	
			c) Se han identificado las herramientas para la elaboración de diagramas de clases.	
			d) Se ha interpretado el significado de diagramas de clases.	
			e) Se han trazado diagramas de clases a partir de las especificaciones de las mismas.	
f) Se ha generado código a partir de un diagrama de clases.				
g) Se ha generado un diagrama de clases mediante ingeniería inversa.				
Actividades				
Tipo	Actividad	Objetivos	Actores	Tiempo
Activades evaluables				
#	Actividad	Instrumento		
7.1	Crear diagrama básico de clases con diagrams.net	Rúbrica		
7.2	Crear diagrama de clases con UMLet	Rúbrica		
7.3	Crear un diagrama de clases a partir de código	Rúbrica		
7.4	Generar diagrama de clases proyecto integrador	Rúbrica		
7.5	Cuestionario UD 7	Solucionario		

Unidad Didáctica 8. Introducción a las pruebas de software

8		Introducción a las pruebas de software		9 Horas	
Contenidos	8.1 El problema de la verificación de software 8.2 Niveles de prueba 8.2.1. Pruebas unitarias 8.2.2. Pruebas de integración 8.2.3. Pruebas de sistema 8.2.4. Pruebas de aceptación 8.2.5. Un enfoque práctico de pruebas: la pirámide del test 8.3. Tipos de prueba 8.3.1. Pruebas de caja blanca 8.3.2. Pruebas de caja negra 8.3.3. Pruebas de caja gris 8.4. Técnicas de prueba 8.4.1. Valores limite 8.4.2. Cubrimiento 8.4.3. Clases de equivalencia 8.5. Proceso y planificación de pruebas 8.6. Procedimientos y casos de prueba 8.7. Métricas de la calidad de software			Contenidos básicos	
				3	
				Contenidos Transversales	
				Autonomía en el trabajo. Uso de las TIC. Interés de ampliación. Visión crítica.	
Objetivos de aprendizaje					
1. Descubrir las limitaciones de la computación y su efecto en la verificación del software 2. Identificar los diferentes niveles tipos, técnicas y niveles de prueba 3. Analizar la utilidad de la planificación en la implementación de pruebas 4. Describir medidas de calidad de software					
Objetivos generales					
Competencias profesionales, personales y sociales					
Resultados de aprendizaje	RA4. Verifica el funcionamiento de programas diseñando y realizando pruebas		Criterios de evaluación	a) Se han identificado los diferentes tipos de pruebas. b) Se han definido casos de prueba. f) Se ha documentado el plan de pruebas. k) Se han aplicado normas de calidad a los procedimientos de desarrollo de software. l) Se han realizado medidas de calidad sobre el software desarrollado.	

Actividades				
Tipo	Actividad	Objetivos	Actores	Tiempo
Activades evaluables				
#	Actividad	Instrumento		

Unidad Didáctica 9. Diseño e implementación de pruebas unitarias

9		Diseño e implementación de pruebas unitarias		15 Horas	
Contenidos	9.1. Herramientas para el desarrollo de pruebas unitarias. Junit 5 9.1.1 Incorporar JUnit5 a un proyecto Java 9.2. Organización de clases de prueba 9.3. Ejecución de pruebas unitarias desde el IDE 9.4. Generación de código de prueba desde el IDE 9.5. Desarrollo de pruebas con JUnit 9.5.1 Anotaciones 9.5.2 Aserciones 9.6. Diseño de pruebas unitarias: el patrón AAA 9.7. Pruebas unitarias con dependencias 9.7.1 Mocking y Test-Doubles 9.7.2 Incorporar Mockito a un proyecto Java 9.7.3 Creación de mocks de clase y stubs de métodos 9.7.4 Verificando interacciones con Mockito	Contenidos básicos		3.5.1, 3.5.5	
		Contenidos Transversales			
		Autonomía en el trabajo. Uso de las TIC. Interés de ampliación. Visión crítica.			
Objetivos de aprendizaje					
1. Conceptualizar la filosofía e importancia de las pruebas unitarias en la calidad del software 2. Implementar y ejecutar pruebas automáticas unitarias 3. Estructurar las pruebas unitarias de una manera práctica y metódica 4. Identificar la necesidad del uso de mock para la realización de pruebas unitarias en código dependiente					
Objetivos generales					
Competencias profesionales, personales y sociales					
Resultados de aprendizaje	RA3. Verifica el funcionamiento de programas diseñando y realizando pruebas	Criterios de evaluación	g) Se han efectuado pruebas unitarias de clases y funciones. i) Se han implementado pruebas automáticas. j) Se han documentado las incidencias detectadas.		
Actividades					
Tipo	Actividad	Objetivos	Actores	Tiempo	

Motv.	¿Son preferibles las pruebas automáticas?	1	Alumnos, Profesor	½ h.
Des	Ejecutar y analizar pruebas desde el IDE	1,2	Alumnos, Profesor	½ h.
Des.				
Activades evaluables				
#	Actividad	Instrumento		

Unidad Didáctica 10. Diseño e implementación de pruebas de integración

10		Diseño e implementación de pruebas de integración			15 Horas	
Contenidos	10.1. Herramientas para el desarrollo de pruebas de integración 10.2. Descarga e instalación de Selenium WebDriver Incorporar Selenium WebDriver a un proyecto Java 10.3. Elementos Web. Identificar y trabajar con objetos en el DOM 10.4. Interacciones con el navegador 10.5 Implementación de pruebas automáticas de integración con WebDriver y JUnit 10.6. Ejecutar pruebas de integración desde el IDE				Contenidos básicos	
					3.5.2, 3.5.5	
					Contenidos Transversales	
					Autonomía en el trabajo. Uso de las TIC. Interés de ampliación. Visión crítica.	
Objetivos de aprendizaje						
1. Reconocer la importancia de la documentación en el proceso de desarrollo de software 2. Analizar las posibilidades de capturar requisitos de software de manera textual 3. Documentar código internamente mediante el uso de comentarios 4. Generar documentación a partir de anotaciones de código						
Objetivos generales						
Competencias profesionales, personales y sociales						
Resultados de aprendizaje	RA3. Verifica el funcionamiento de programas diseñando y realizando pruebas		Criterios de evaluación	h) Se han efectuado pruebas de integración, de sistema y de aceptación. i) Se han implementado pruebas automáticas. j) Se han documentado las incidencias detectadas.		
Actividades						
Tipo	Actividad			Objetivos	Actores	Tiempo

Activades evaluables				
#	Actividad	Instrumento		

Unidad Didáctica 11. Documentación de software

11		Documentación de software	3 Horas
Contenidos	11.1. El papel de la documentación en el desarrollo 11.2. Uso de comentarios para la documentación 11.3. Generación de documentación a partir de código: JavaDoc 11.3.1. Anotaciones para clases 11.3.2. Anotaciones para métodos 11.4. Herramientas externas de documentación 11.4.1 Notion 11.4.2 Read The Docs 11.4.3 README.MD en GitHub	Contenidos básicos	
		4.3	
		Contenidos Transversales	
		Autonomía en el trabajo. Uso de las TIC. Interés de ampliación. Visión crítica.	
Objetivos de aprendizaje			
1. Reconocer la importancia de la documentación en el proceso de desarrollo de software 2. Analizar las posibilidades de capturar requisitos de software de manera textual 3. Documentar código internamente mediante el uso de comentarios 4. Generar documentación a partir de anotaciones de código			
Objetivos generales			
Competencias profesionales, personales y sociales			

Resultados de aprendizaje	RA4. Optimiza código empleando las herramientas disponibles en el entorno de desarrollo	Criterios de evaluación	g) Se ha documentado el código fuente mediante comentarios. h) Se han utilizado herramientas del entorno de desarrollo para documentar los procesos, datos y eventos. i) Se han utilizado herramientas del entorno de desarrollo para documentar las clases.		
Actividades					
Tipo	Actividad	Objetivos	Actores	Tiempo	
Motv.	¿Son necesarios los comentarios de código?	1,2	Alumnos, profesor	¼ h	
Des.	Impartición UD 11	1,2,3,4	Alumnos, profesor	1 h	
Des.	Comenta tu proyecto integrador	1,2,3,4	Alumnos	1 ½ h	
Des	Publica el JavaDoc de tu proyecto en GitHub pages	1,2,3,4	Alumnos, profesor	¼ h	
Activades evaluables					
#	Actividad	Instrumento			
11.1	Comenta tu proyecto integrador	Rúbrica			
11.2	Publica el JavaDoc de tu proyecto en GitHub pages	Lista de control			

Unidad Didáctica 12. Código limpio y refactorización de código

12	Código limpio y refactorización de código	6 Horas
Contenidos	12.1. El Código limpio 12.1.1. Los cinco principios básicos de la programación orientada a objetos. SOLID 12.2. Patrones de diseño. 12.2.1. Patrones de diseño comunes	Contenidos básicos
		4.1
		Contenidos Transversales

	12.2.2. El patrón Modelo-Vista-Controlador 12.3. Anti-patrones de diseño 12.3.1 Los anti-patrones de diseño más habituales 12.3.2 El anti-patrón del código espagueti 12.4. Refactorización 12.4.1. Code-Smells 12.4.2 Técnicas de refactorización 12.5. Herramientas de refactorización en IDE			Autonomía en el trabajo. Uso de las TIC. Interés de ampliación. Visión crítica.
Objetivos de aprendizaje				
1. Analizar la importancia del código limpio en el proceso de desarrollo de software 2. Reconocer los principios de las buenas prácticas de desarrollo 3. Aplicar patrones estructurales básicos 4. Implementar refactorizaciones sencillas de código con ayuda del IDE				
Objetivos generales				
Competencias profesionales, personales y sociales				
Resultados de aprendizaje	RA4. Optimiza código empleando las herramientas disponibles en el entorno de desarrollo	Criterios de evaluación	g) Se ha documentado el código fuente mediante comentarios. h) Se han utilizado herramientas del entorno de desarrollo para documentar los procesos, datos y eventos. i) Se han utilizado herramientas del entorno de desarrollo para documentar las clases.	
Actividades				
Tipo	Actividad	Objetivos	Actores	Tiempo
Activades evaluables				

#	Actividad	Instrumento

Bloques de contenido

Las unidades didácticas están divididas en bloques conceptuales como se puede ver en la Tabla 3. Es importante señalar que estos bloques no se corresponden a las evaluaciones del curso y que obedecen a una agrupación de contenidos afines.

Bloque	Unidades didácticas
El desarrollo de software y sus herramientas	UD 1. Introducción al desarrollo de Software UD 2. Instalación y uso de entornos de desarrollo integrado UD 3. Ciclo de vida del software. Metodologías UD 5. Control de versiones, repositorios remotos y colaboración
Documentación y requisitos de software	UD 6. Diseño orientado a objetos. Diagramas de comportamiento UD 10. Diseño orientado a objetos. Diagramas estructurales UD 11. Documentación de software
Calidad, pruebas y refactorización	UD 4. Defectos de software y depuración UD 8. Introducción a las pruebas de software UD 9. Diseño e implementación de pruebas unitarias UD 9. Diseño e implementación de pruebas de integración UD 12. Código limpio y refactorización de código

Tabla 3. Bloques conceptuales del módulo

Resultados de aprendizaje, criterios de evaluación y unidades didácticas

La Tabla 4 muestra la relación entre las unidades didácticas programadas, los criterios de evaluación y los resultados de aprendizaje del módulo profesional.

UD	#	Resultados de aprendizaje	Criterios de evaluación
UD1	RA1	Reconoce los elementos y herramientas que intervienen en el desarrollo de un programa informático, analizando sus características y las fases en las que actúan hasta	a) Se ha reconocido la relación de los programas con los componentes del sistema informático, memoria, procesador, periféricos, entre otros. b) Se han clasificado los lenguajes de programación. c) Se han diferenciado los conceptos de código fuente, objeto y ejecutable.

		llegar a su puesta en funcionamiento.	d) Se han reconocido las características de la generación de código intermedio para su ejecución en máquinas virtuales. f) Se ha diferenciado el funcionamiento de los distintos tipos de traductores de lenguajes ante el código fuente de un programa.
UD2	RA2	Evalúa entornos integrados de desarrollo analizando sus características para editar código fuente y generar ejecutable.	a) Se han instalado entornos de desarrollo, propietarios y libres. b) Se han añadido y eliminado módulos en el entorno de desarrollo. c) Se ha personalizado y automatizado el entorno de desarrollo. d) Se ha configurado el sistema de actualización del entorno de desarrollo. e) Se han generado ejecutables a partir de código fuente de diferentes lenguajes en un mismo entorno de desarrollo. f) Se han generado ejecutables a partir de un mismo código fuente con varios entornos de desarrollo. g) Se han identificado las características comunes y específicas de diversos entornos de desarrollo. h) Se han identificado las funciones más usuales de las herramientas CASE para el desarrollo, prueba y documentación de código.
UD3	RA1	Reconoce los elementos y herramientas que intervienen en el desarrollo de un programa informático, analizando sus características y las fases en las que actúan hasta llegar a su puesta en funcionamiento.	g) Se han identificado las fases de desarrollo de una aplicación informática.
UD4	RA3	Verifica el funcionamiento de programas diseñando y realizando pruebas.	c) Se han identificado las herramientas de depuración y prueba de aplicaciones ofrecidas por el entorno de desarrollo. d) Se han utilizado herramientas de depuración para definir puntos de ruptura y seguimiento. e) Se han utilizado las herramientas de depuración para examinar y modificar el comportamiento de un programa en tiempo de ejecución.
	RA4	Optimiza código empleando las herramientas disponibles en el entorno de desarrollo	c) Se ha revisado el código fuente usando un analizador de código. d) Se han identificado las posibilidades de configuración de un analizador de código.

UD5	RA4	Optimiza código empleando las herramientas disponibles en el entorno de desarrollo	f) Se ha realizado el control de versiones integrado en el entorno de desarrollo.
UD6	RA6	Genera diagramas de comportamiento valorando su importancia en el desarrollo de aplicaciones y empleando las herramientas disponibles en el entorno	a) Se han identificado los distintos tipos de diagramas de comportamiento. b) Se ha reconocido el significado de los diagramas de casos de uso. c) Se han interpretado diagramas de interacción. d) Se han elaborado diagramas de interacción sencillos. e) Se han interpretado diagramas de estados. f) Se han planteado diagramas de estados sencillos. g) Se ha interpretado el significado de diagramas de actividades. h) Se han elaborado diagramas de actividades sencillos.
UD7	RA5	Genera diagramas de clases valorando su importancia en el desarrollo de aplicaciones y empleando las herramientas disponibles en el entorno	a) Se han identificado los conceptos básicos de la programación orientada a objetos. b) Se ha instalado el módulo del entorno integrado de desarrollo que permite la utilización de diagramas de clases. c) Se han identificado las herramientas para la elaboración de diagramas de clases. d) Se ha interpretado el significado de diagramas de clases. e) Se han trazado diagramas de clases a partir de las especificaciones de las mismas. f) Se ha generado código a partir de un diagrama de clases. g) Se ha generado un diagrama de clases mediante ingeniería inversa.
UD8	RA3	Verifica el funcionamiento de programas diseñando y realizando pruebas.	a) Se han identificado los diferentes tipos de pruebas. b) Se han definido casos de prueba. f) Se ha documentado el plan de pruebas. k) Se han aplicado normas de calidad a los procedimientos de desarrollo de software. l) Se han realizado medidas de calidad sobre el software desarrollado.
UD9	RA3	Verifica el funcionamiento de programas diseñando y realizando pruebas.	g) Se han efectuado pruebas unitarias de clases y funciones. i) Se han implementado pruebas automáticas. j) Se han documentado las incidencias detectadas.
UD10	RA3	Verifica el funcionamiento	h) Se han efectuado pruebas de

		de programas diseñando y realizando pruebas.	integración, de sistema y de aceptación. i) Se han implementado pruebas automáticas. j) Se han documentado las incidencias detectadas.
UD11	RA4	Optimiza código empleando las herramientas disponibles en el entorno de desarrollo	g) Se ha documentado el código fuente mediante comentarios. h) Se han utilizado herramientas del entorno de desarrollo para documentar los procesos, datos y eventos. i) Se han utilizado herramientas del entorno de desarrollo para documentar las clases.
UD12	RA4	Optimiza código empleando las herramientas disponibles en el entorno de desarrollo	a) Se han identificado los patrones de refactorización más usuales. b) Se han elaborado las pruebas asociadas a la refactorización. e) Se han aplicado patrones de refactorización con las herramientas que proporciona el entorno de desarrollo.

Tabla 4. Relación RA, CE y UD

Metodología

Como se recoge en la Orden de 16 de junio de 2011, las líneas de actuación en el proceso de enseñanza-aprendizaje que permite alcanzar los objetivos del módulo versarán sobre:

- La interpretación de documentación técnica.
- La instalación, configuración y personalización de diversos entornos de desarrollo.
- La utilización de distintos entornos de desarrollo para la edición y prueba de aplicaciones.
- La utilización de herramientas de depuración, optimización y documentación de aplicaciones.
- La generación de diagramas técnicos.
- La elaboración de documentación interna de la aplicación.

Según el **artículo 8.6 del Real Decreto 1147/2011, de 29 de julio**, por el que se establece la ordenación general de la formación profesional del sistema educativo:

“La metodología didáctica de las enseñanzas de formación profesional integrará los aspectos científicos, tecnológicos y organizativos que en cada caso correspondan, con el fin de que el alumnado adquiera una visión global de los procesos productivos propios de la actividad profesional correspondiente.”

Metodología de aula

Agrupamientos del alumnado

Los agrupamientos del alumnado obedecerán a criterios pedagógicos

Instrumentos de seguimiento

Todas las actividades entregables del curso se registrarán en la plataforma de enseñanza a distancia Moodle Centros

Cuaderno de clase

Anecdotario

Portafolio

Tipos de instrumento de evaluación

Solucionario con criterios de corrección de pruebas

Un solucionario con criterios de corrección es un documento que establece los criterios para corregir pruebas. En otras palabras, define lo que se espera que la persona "sepa" sobre un determinado aspecto que se presenta en la prueba. Si estas pruebas se utilizan para asignar una calificación, también se determina el valor que cada parte de la prueba aportará a la calificación final.

Listas de control

Una lista de control es un conjunto de criterios o elementos que se deben verificar para determinar su cumplimiento. Estas listas son herramientas útiles para guiar observaciones durante la ejecución de actividades prácticas, exposiciones y otros contextos similares. Permiten comprobar si se cumplen ciertos pasos, técnicas, procedimientos o normas establecidas, así como evaluar actitudes específicas.

El diseño de una lista de control puede variar según el caso particular, pero generalmente incluye los criterios a evaluar, un espacio para indicar si se cumplen o no, y a veces, un espacio adicional para realizar observaciones. En ocasiones, algunos de estos criterios pueden requerir ser transformados en indicadores más útiles para su valoración efectiva.

Escalas de registro, escalas de valoración y rúbricas

Las escalas de registro, las escalas de valoración y las rúbricas comparten un fundamento similar al de las listas de control. Consisten en un listado de variables o criterios que se observan, pero a diferencia de las listas de control, estas herramientas permiten registrar la calidad de lo observado al proporcionar más de una categoría para el registro.

Las escalas de registro pueden ser tanto cuantitativas como cualitativas. En las escalas cuantitativas se asignan valores numéricos (como 0, 1, 2, 3, etc.) que permiten el tratamiento y análisis de los datos obtenidos. Por otro lado, las escalas cualitativas utilizan categorías descriptivas (como bien, regular o mal; totalmente de acuerdo, de acuerdo, en desacuerdo, etc.) para valorar la calidad de lo observado.

En el caso de las rúbricas, estas describen de manera concisa los aspectos o criterios incluidos en relación al nivel de logro esperado. Permiten evaluar y proporcionar retroalimentación detallada sobre el desempeño en cada uno de los aspectos evaluados. Las rúbricas suelen incluir niveles de logro o

categorías descriptivas que indican el grado de cumplimiento o competencia en cada criterio evaluado.

Recursos didácticos y materiales

Para la realización de las actividades asociadas al módulo se precisa una dotación material de equipos y materiales. Esta dotación podrá verse modificada durante el curso en función de las necesidades.

Aunque hay que tener en cuenta la importancia de contar con un equipamiento propio y adecuado al centro para el alumnado, hay que atender a aquellos alumnos en riesgo de exclusión digital que no cuenten con equipos informativos propios para la continuación de las actividades fuera del horario lectivo. Para ello se referirán dichos alumnos a la coordinación TDE para hacer uso del servicio de préstamo de dispositivos digitales.

Material docente en el aula y equipos técnicos

Todas las sesiones formativas se llevarán a cabo en un aula de programación, que según el **Anexo IV de la Orden de 16 de junio de 2011** debe contar con el siguiente equipamiento:

- Ordenadores instalados en red y con acceso a Internet.
- Conexión a la red Internet que permita configurar y redireccionar todos los parámetros y servicios de red.
- Medios de proyección.
- Impresora láser.
- Sistemas Operativos.
- Software de control remoto.
- Servidores de Ficheros, Web, Bases de datos y Aplicaciones.
- Software de creación y edición de máquinas virtuales.
- Herramientas de clonación de equipos.
- Cortafuegos, detectores de intrusos, aplicaciones de Internet, entre otras.
- Gestores de Bases de Datos. Servidores y clientes.
- Entornos de desarrollo, compiladores e intérpretes, analizadores de código fuente, empaquetadores, generadores de ayudas, entre otros.
- Dispositivos de interconexión de redes.

Software específico

Se prevé que el siguiente software será necesario para la realización de las actividades del curso y será instalado en los equipos del alumnado como del profesor:

1. Visual Studio Code
2. Eclipse IDE
3. Docker
4. IntelliJ Idea
5. Apache Maven
6. Java Development Toolkit
7. Selenium WebDriver

Usos de las TIC y herramientas de trabajo colaborativo

Dada la naturaleza del ciclo y del del módulo de Entornos de Desarrollo el uso de las TIC es parte fundamental de los procesos de enseñanza-aprendizaje. Se plantea el uso de Moodle Centros para la publicación de materiales, actividades y la gestión de aula virtual. Esta plataforma servirá como punto de encuentro entre profesores y alumnado. Se irán colgando los diferentes materiales propuestos en el aula, como documentos, enlaces a recursos y tareas. Será en esta plataforma donde se realizará la entrega de las actividades escritas del alumnado, quedando un registro de la fecha de entrega y pudiendo el alumno acceder al resultado de evaluación y la calificación de cada una de ellas. Así mismo se crearán actividades en la Moodle asociadas a las prácticas de taller, para que el alumno pueda ver su evaluación y calificación en las mismas. La Moodle nos aporta además un foro de interacción entre los alumnos y el profesor, pudiéndose crear hilos de debate e intercambio de información entre los alumnos, así como creando un canal de comunicación oficial entre el alumno y el profesor.

El trabajo del desarrollador es un trabajo eminente colaborativo por lo que durante el curso se fomentará el uso de herramientas donde poder trabajar de manera colaborativa:

- Diagrams.net para la realización de diagramas colaborativamente
- Trello para la planificación de tareas por parte de equipos
- Notion para la generación de documentación de manera colaborativa
- Miro para la realización de mapas mentales y lluvias de ideas
- Github para tareas de desarrollo colaborativo de software

Organización de los espacios

Así mismo, el **Anexo IV de la Orden de 16 de junio de 2011** también establece que los requisitos de espacios mostrados en la Tabla 5.

Espacio Formativo	Superficie m ² 30 alumnos	Superficie m ² 20 alumnos
Aula de programación	60	40
Aula de desarrollo web	60	40
Aula polivalente	60	40

Tabla 5. Requisitos espacios del título

En el aula, debido a la naturaleza del módulo de Entornos de Desarrollo gran parte de la actividad se desarrollará con ayuda de un PC en un aula de programación que tiene asignado cada ciclo formativo. Hay que tener en cuenta, que el uso de equipos informáticos puede convertirse en una distracción para el alumnado por lo que deberá hacer hincapié en la necesidad de aprovechar el tiempo en el aula y la obligación de usar los equipos del centro solo en las actividades propuestas.

Evaluación y calificación

Sobre la evaluación podemos ver en el artículo 51 del Real Decreto 1147/2011 lo siguiente:

1. La evaluación del aprendizaje del alumnado de las enseñanzas de formación profesional se realizará por módulos profesionales. Los procesos de evaluación se adecuarán a las adaptaciones metodológicas de las que haya podido ser objeto el alumnado con discapacidad y se garantizará su accesibilidad a las pruebas de evaluación.

2. En todo caso, la evaluación se realizará tomando como referencia los objetivos, expresados en resultados de aprendizaje, y los criterios de evaluación de cada uno de los módulos profesionales, así como los objetivos generales del ciclo formativo o curso de especialización.

Según la orden de 29 de septiembre de 2010, por la que se regula la evaluación, certificación, acreditación y titulación académica del alumnado que cursa enseñanzas de formación profesional inicial que forma parte del sistema educativo en la Comunidad Autónoma de Andalucía, en su artículo 2 dice:

1. La evaluación de los aprendizajes del alumnado que cursa ciclos formativos será continua y se realizará por módulos profesionales.
2. La aplicación del proceso de evaluación continua del alumnado requerirá, en la modalidad presencial, su asistencia regular a clase y su participación en las actividades programadas para los distintos módulos profesionales del ciclo formativo.
4. La evaluación del alumnado será realizada por el profesorado que imparta cada módulo profesional del ciclo formativo, de acuerdo con los resultados de aprendizaje, los criterios de evaluación y contenidos de cada módulo profesional, así como las competencias y objetivos generales del ciclo formativo asociados a los mismos. [...]

Proceso de evaluación por resultados de aprendizaje

La evaluación del módulo se llevará a cabo de forma continua, centrándose en las competencias profesionales, personales y sociales mencionadas previamente en el plan de estudios, y estará en línea con los resultados de aprendizaje y los criterios de evaluación establecidos por la legislación vigente.

Se concibe la evaluación como un proceso continuo, sistemático y personalizado, en el que se pueden y deben emplear diferentes técnicas, evitando que se reduzca a acciones aisladas o que se confunda con la mera calificación final. La calificación consistirá en expresar, mediante un código preestablecido, la conclusión alcanzada luego de este proceso de evaluación.

La evaluación tiene como objetivos los siguientes fines:

- Ser útil para el alumno, brindándole una comprensión clara de su progreso en relación con sus posibilidades.
- Detectar las áreas de insuficiencia y analizar sus causas, con el propósito de planificar refuerzos específicos y adecuados.
- Identificar los recursos que permitan superar las deficiencias encontradas.
- Guiar al profesor en todo momento en la elaboración de su planificación didáctica y en el proceso de enseñanza-aprendizaje.

Momentos de evaluación

En la evaluación continua, podemos identificar tres fases o momentos distintos y complementarios: la evaluación inicial, la evaluación formativa o procesual, y la evaluación sumativa o final.

La **evaluación inicial** se lleva a cabo al comienzo del proceso de enseñanza-aprendizaje, tanto al iniciar el módulo como cada una de las unidades didácticas. Su propósito es obtener información previa sobre las características individuales de los estudiantes, como sus conocimientos sobre la materia, habilidades, actitudes, entre otros, con el fin de adaptar los contenidos a sus necesidades. Para realizar esta evaluación inicial, se emplearán cuestionarios al inicio del curso. Posteriormente, al iniciar cada unidad didáctica, se utilizarán diversas actividades, como lluvia de ideas, cuestionarios, entre otros, para recopilar información adicional.

La **evaluación formativa o procesual** se llevará a cabo durante todo el proceso de formación, con el objetivo de determinar hasta qué punto se están logrando los objetivos establecidos al inicio de la unidad didáctica, así como diagnosticar las deficiencias en el aprendizaje de los estudiantes y permitir al profesorado replantear las estrategias utilizadas.

- Para medir estos avances y dificultades, se recopilarán datos relacionados con:
- Los resultados de pruebas objetivas teóricas y/o prácticas que abarquen los contenidos tratados en la unidad.
- Las actividades y tareas realizadas por los estudiantes durante el desarrollo de la unidad.
- El grado de participación y colaboración del alumno tanto con el profesor como con sus compañeros.
- Las actitudes hacia el proceso de enseñanza-aprendizaje, así como aquellas relacionadas con el respeto, la tolerancia, entre otras.
- La responsabilidad mostrada en el trabajo.

Esta evaluación también nos permitirá reflexionar sobre la idoneidad del proceso que estamos siguiendo y realizar modificaciones en la programación, en caso necesario. En la sección de Autoevaluación se detalla el proceso de revisión de la presente programación.

La **evaluación sumativa o final** tiene como objetivo principal determinar lo que se ha aprendido y en qué medida se ha logrado. Esta etapa se basa en los datos recopilados durante la evaluación formativa o procesual y se lleva a cabo de manera parcial al final de cada trimestre, y de manera global en la evaluación ordinaria.

Ponderación de cada resultado de aprendizaje en la nota del módulo

Los seis resultados de aprendizaje del módulo han sido ponderados en su contribución a la nota del módulo como se detalla en la Tabla 6.

#	Resultado de aprendizaje	%
RA1	Reconoce los elementos y herramientas que intervienen en el desarrollo de un programa informático, analizando sus características y las fases en las que actúan hasta llegar a su	10%

	puesta en funcionamiento	
RA2	Evalúa entornos integrados de desarrollo analizando sus características para editar código fuente y generar ejecutable	10%
RA3	Verifica el funcionamiento de programas diseñando y realizando pruebas	35%
RA4	Optimiza código empleando las herramientas disponibles en el entorno de desarrollo	25%
RA5	Genera diagramas de clases valorando su importancia en el desarrollo de aplicaciones y empleando las herramientas disponibles en el entorno	10%
RA6	Genera diagramas de comportamiento valorando su importancia en el desarrollo de aplicaciones y empleando las herramientas disponibles en el entorno	10%

Tabla 6. ponderación RA en la nota del módulo

Ponderación de los criterios de evaluación de cada resultado de aprendizaje

La Tabla 7 muestra la ponderación que tiene cada criterio de evaluación en el resultado de aprendizaje. El primer porcentaje mostrado es la aportación del criterio de evaluación al resultado de aprendizaje, mientras que el segundo refleja su contribución al total del módulo profesional.

RA	Criterio de evaluación	%RA	%Tot
RA1 10%	a) Se ha reconocido la relación de los programas con los componentes del sistema informático, memoria, procesador, periféricos, entre otros	15%	1,5%
	b) Se han clasificado los lenguajes de programación	15%	1,5%
	c) Se han diferenciado los conceptos de código fuente, objeto y ejecutable	10%	1%
	d) Se han reconocido las características de la generación de código intermedio para su ejecución en máquinas virtuales	15%	1,5%
	e) Se ha evaluado la funcionalidad ofrecida por las herramientas utilizadas en programación	15%	1,5%
	f) Se ha diferenciado el funcionamiento de los distintos tipos de traductores de lenguajes ante el código fuente de un programa	10%	1%
	g) Se han identificado las fases de desarrollo de una aplicación informática.	20%	2%
RA2 10%	a) Se han instalado entornos de desarrollo, propietarios y libres.	20%	2%
	b) Se han añadido y eliminado módulos en el entorno de desarrollo.	15%	1,5%
	c) Se ha personalizado y automatizado el entorno de desarrollo	15%	1,5%
	d) Se ha configurado el sistema de actualización del entorno de desarrollo.	10%	1%
	e) Se han generado ejecutables a partir de código fuente de diferentes lenguajes en un mismo entorno de	10%	1%

	desarrollo.		
	f) Se han generado ejecutables a partir de un mismo código fuente con varios entornos de desarrollo.	10%	1%
	g) Se han identificado las características comunes y específicas de diversos entornos de desarrollo.	10%	1%
	h) Se han identificado las funciones más usuales de las herramientas CASE para el desarrollo, prueba y documentación de código.	10%	1%
RA3 35%	a) Se han identificado los diferentes tipos de pruebas.	3%	1%
	b) Se han definido casos de prueba.	3%	1%
	c) Se han identificado las herramientas de depuración y prueba de aplicaciones ofrecidas por el entorno de desarrollo.	3%	1%
	d) Se han utilizado herramientas de depuración para definir puntos de ruptura y seguimiento.	3%	1%
	e) Se han utilizado las herramientas de depuración para examinar y modificar el comportamiento de un programa en tiempo de ejecución.	3%	1%
	f) Se ha documentado el plan de pruebas.	3%	1%
	h) Se han efectuado pruebas unitarias de clases y funciones.	26%	9%
	g) Se han efectuado pruebas de integración, de sistema y de aceptación.	20%	7%
	i) Se han implementado pruebas automáticas.	29%	10%
	j) Se han documentado las incidencias detectadas.	3%	1%
RA4 25%	k) Se han aplicado normas de calidad a los procedimientos de desarrollo de software.	3%	1%
	l) Se han realizado medidas de calidad sobre el software desarrollado.	3%	1%
	a) Se han identificado los patrones de refactorización más usuales.	8%	2%
	b) Se han elaborado las pruebas asociadas a la refactorización.	8%	2%
	c) Se ha revisado el código fuente usando un analizador de código.	8%	2%
	d) Se han identificado las posibilidades de configuración de un analizador de código.	4%	1%
	e) Se han aplicado patrones de refactorización con las herramientas que proporciona el entorno de desarrollo.	12%	3%
	f) Se ha realizado el control de versiones integrado en el entorno de desarrollo.	40%	10%
	g) Se ha documentado el código fuente mediante comentarios.	4%	1%
	h) Se han utilizado herramientas del entorno de desarrollo para documentar los procesos, datos y eventos.	8%	2%
	i) Se han utilizado herramientas del entorno de desarrollo para documentar las clases.	8%	2%

RA5 10%	a) Se han identificado los conceptos básicos de la programación orientada a objetos.	10%	1%
	b) Se ha instalado el módulo del entorno integrado de desarrollo que permite la utilización de diagramas de clases.	10%	1%
	c) Se han identificado las herramientas para la elaboración de diagramas de clases.	10%	1%
	d) Se ha interpretado el significado de diagramas de clases.	20%	2%
	e) Se han trazado diagramas de clases a partir de las especificaciones de las mismas.	20%	2%
	f) Se ha generado código a partir de un diagrama de clases.	10%	1%
	g) Se ha generado un diagrama de clases mediante ingeniería inversa.	20%	2%
RA6 10%	a) Se han identificado los distintos tipos de diagramas de comportamiento.	12,5%	1,25%
	b) Se ha reconocido el significado de los diagramas de casos de uso.	12,5%	1,25%
	c) Se han interpretado diagramas de interacción.	12,5%	1,25%
	d) Se han elaborado diagramas de interacción sencillos.	12,5%	1,25%
	e) Se han interpretado diagramas de estados.	12,5%	1,25%
	f) Se han planteado diagramas de estados sencillos.	12,5%	1,25%
	g) Se ha interpretado el significado de diagramas de actividades.	12,5%	1,25%
	h) Se han elaborado diagramas de actividades sencillos.	12,5%	1,25%

Tabla 7. Ponderación de los criterios de evaluación por cada RA

Ponderación de la relación entre resultados de aprendizaje, criterios de evaluación y actividades evaluables

RA	Criterio de Evaluación	Act.	%
RA1	a) Se ha reconocido la relación de los programas con los componentes del sistema informático, memoria, procesador, periféricos, entre otros	1.4	100%
	b) Se han clasificado los lenguajes de programación	1.1	70%
		1.4	30%
	c) Se han diferenciado los conceptos de código fuente, objeto y ejecutable	1.2	20%
		1.3	20%
		1.4	60%
	d) Se han reconocido las características de la generación de código intermedio para su ejecución en máquinas virtuales	1.3	80%
		1.4	20%
	e) Se ha evaluado la funcionalidad ofrecida por las herramientas utilizadas en programación	1.2	20%
		1.3	20%
		1.4	60%
	f) Se ha diferenciado el funcionamiento de los distintos	1.2	20%
		1.3	20%

	tipos de traductores de lenguajes ante el código fuente de un programa	1.4	60%
	g) Se han identificado las fases de desarrollo de una aplicación informática.	3.1	50%
		3.2	50%
RA2	a) Se han instalado entornos de desarrollo, propietarios y libres.	2.1	50%
		2.2	50%
	b) Se han añadido y eliminado módulos en el entorno de desarrollo.	2.4	100%
	c) Se ha personalizado y automatizado el entorno de desarrollo	2.3	100%
	d) Se ha configurado el sistema de actualización del entorno de desarrollo.	2.4	100%
	e) Se han generado ejecutables a partir de código fuente de diferentes lenguajes en un mismo entorno de desarrollo.	2.6	100%
	f) Se han generado ejecutables a partir de un mismo código fuente con varios entornos de desarrollo.	2.5	100%
	g) Se han identificado las características comunes y específicas de diversos entornos de desarrollo.	2.1	30%
		2.2	30%
		2.7	40%
	h) Se han identificado las funciones más usuales de las herramientas CASE para el desarrollo, prueba y documentación de código.	2.4	50%
		2.7	50%
RA3	a) Se han identificado los diferentes tipos de pruebas.		
	b) Se han definido casos de prueba.		
	c) Se han identificado las herramientas de depuración y prueba de aplicaciones ofrecidas por el entorno de desarrollo.	4.2	25%
		4.3	25%
	d) Se han utilizado herramientas de depuración para definir puntos de ruptura y seguimiento.	4.2	100%
	e) Se han utilizado las herramientas de depuración para examinar y modificar el comportamiento de un programa en tiempo de ejecución.	4.2	100%
	f) Se ha documentado el plan de pruebas.		
	h) Se han efectuado pruebas unitarias de clases y funciones.		
	g) Se han efectuado pruebas de integración, de sistema y de aceptación.		
	i) Se han implementado pruebas automáticas.		
	j) Se han documentado las incidencias detectadas.		
	k) Se han aplicado normas de calidad a los procedimientos de desarrollo de software.		
	l) Se han realizado medidas de calidad sobre el software desarrollado.		
RA4	a) Se han identificado los patrones de refactorización más usuales.		
	b) Se han elaborado las pruebas asociadas a la refactorización.		
	c) Se ha revisado el código fuente usando un analizador	4.1	100%

	de código.		
	d) Se han identificado las posibilidades de configuración de un analizador de código.	4.1	50%
		4.3	50%
	e) Se han aplicado patrones de refactorización con las herramientas que proporciona el entorno de desarrollo.		
	f) Se ha realizado el control de versiones integrado en el entorno de desarrollo.	5.1	10%
		5.2	10%
		5.3	10%
		5.4	10%
		5.5	10%
		5.6	10%
		5.7	20%
		5.8	20%
	g) Se ha documentado el código fuente mediante comentarios.	11.1	100%
	h) Se han utilizado herramientas del entorno de desarrollo para documentar los procesos, datos y eventos.	11.1	90%
		11.2	10%
	i) Se han utilizado herramientas del entorno de desarrollo para documentar las clases.	11.1	90%
		11.2	10%
RA5	a) Se han identificado los conceptos básicos de la programación orientada a objetos.	7.5	100%
	b) Se ha instalado el módulo del entorno integrado de desarrollo que permite la utilización de diagramas de clases.	7.2	100%
	c) Se han identificado las herramientas para la elaboración de diagramas de clases.	7.1	50%
		7.2	50%
	d) Se ha interpretado el significado de diagramas de clases.	7.3	100%
	e) Se han trazado diagramas de clases a partir de las especificaciones de las mismas.	7.1	15%
		7.2	15%
		7.4	70%
	f) Se ha generado código a partir de un diagrama de clases.	7.3	100%
	g) Se ha generado un diagrama de clases mediante ingeniería inversa.	7.3	100%
RA6	a) Se han identificado los distintos tipos de diagramas de comportamiento.	6.8	100%
	b) Se ha reconocido el significado de los diagramas de casos de uso.	6.1	100%
	c) Se han interpretado diagramas de interacción.	6.2	100%
	d) Se han elaborado diagramas de interacción sencillos.	6.3	100%
	e) Se han interpretado diagramas de estados.	6.4	100%
	f) Se han planteado diagramas de estados sencillos.	6.5	100%
	g) Se ha interpretado el significado de diagramas de actividades.	6.6	100%
	h) Se han elaborado diagramas de actividades sencillos.	6.7	100%

Tabla 8. Ponderación entre RA, CE y actividades evaluables

Mecanismos de recuperación y mejora de la calificación

Dado que la evaluación es un proceso continuo, se hará uso de los instrumentos de seguimiento establecidos en la presente programación didáctica para la detección de posibles deficiencias del alumnado a la hora de alcanzar algún resultado de aprendizaje. De esta manera, se podrán establecer la realización de actividades complementarias de refuerzo para apoyar a los alumnos en los aspectos que presenten dificultades. Para ello, las actividades entregables por el alumnado se corregirán en el menor tiempo posible con la finalidad de servir como instrumento de seguimiento del desempeño del alumnado.

Sin embargo, cuando el alumno o alumna no logre la superación de las deficiencias y fallos detectados y haya obtenido una valoración insuficiente en cualquiera de los conceptos evaluados, se establecerán unas actividades específicas de recuperación. Estas actividades estarán en función de los contenidos a reforzar o recuperar y las capacidades implicadas. Estas actividades pueden consistir en: resolución de cuestionarios, análisis y resolución de casos prácticos, trabajos individuales, realización de exposiciones, pruebas escritas, etc.

De la misma manera, y teniendo en cuenta los diferentes ritmos de aprendizaje que podemos tener en el aula, podemos tener alumnos y alumnas que quieran mejorar la calificación obtenida. El profesor o profesora debe incentivar este tipo de iniciativas y potenciarlas, como forma de mostrar interés por el aprendizaje. Para ello se establecerán actividades de mejora de la calificación que podrán ser del mismo tipo que las anteriores, pero con cierto grado de profundidad, según las capacidades del alumnado.

Estos mecanismos de recuperación y mejora de la calificación se planificarán para ser realizado tras la tercera evaluación y antes del fin del periodo lectivo.

Autoevaluación

De la programación didáctica

Durante el final de cada periodo de evaluación se realizará una autoevaluación de la programación didáctica a carga de los docentes encargados del modulo de entornos de desarrollo. Para ello se rellenará el formulario descrito en el Anexo II. Formulario autoevaluación programación didáctica con diferentes indicadores de logro referenciados a los aspectos importantes de la programación que queremos evaluar.

De la labor docente

Para realizar la evaluación de la práctica docente y del proceso de enseñanza-aprendizaje se realizarán las siguientes actuaciones durante el curso escolar:

Evaluación por parte del alumnado

Durante el final de cada unidad didáctica se distribuirán al alumnado mediante la plataforma Moodle unos formularios anónimos donde puedan valorar la actividad del profesor, la metodología, la adecuación de los contenidos o los

instrumentos de evaluación. Así mismo, estos formularios darán la posibilidad de realizar aportaciones a modo de observaciones para mejorar y adecuar el proceso de resultado aprendizaje

Evaluación por parte del propio docente

Además, al final de cada trimestre el docente también rellenará un formulario como el descrito en Anexo I. Indicadores de evaluación de la actividad docente con los indicadores de consecución de diferentes aspectos importantes de la práctica docente y su nivel de consecución:

1. Metodología docente y adecuación del contenido
2. Adecuación de la evaluación
3. Atención a la diversidad
4. Fomento de la empleabilidad del alumnado
5. Desarrollo de competencias transversales

Atención a la diversidad

Para poder abordar la diversidad que siempre está presente en el aula, derivada de las diferencias individuales en intereses, motivación, nivel de desarrollo y capacidad intelectual, se implementarán estrategias para atender las necesidades de todos los estudiantes.

Se propondrán actividades de ampliación para aquellos alumnos que avanzan más rápidamente, brindándoles desafíos adicionales que estimulen su crecimiento. Al mismo tiempo, se ofrecerán actividades de refuerzo para aquellos estudiantes que encuentren dificultades, proporcionándoles el apoyo necesario para superar sus desafíos de aprendizaje.

Además de considerar las diferencias en el ritmo de aprendizaje, también se tendrán en cuenta las necesidades específicas de apoyo educativo (NEAE) que algunos estudiantes puedan tener. La programación se mantendrá abierta y flexible para adaptarse a nuevos enfoques y campos de actuación que sean necesarios.

Es importante destacar que, en la formación profesional, no se implementarán adaptaciones curriculares significativas, sino que se realizarán ajustes para facilitar el acceso al currículo. Esto puede incluir aspectos como la distribución del espacio, la disposición del aula, los recursos didácticos y la agrupación de estudiantes.

En todos los casos, se llevará a cabo un seguimiento continuo a través de la observación del profesor y los resultados de las evaluaciones para asegurar que las medidas implementadas estén cumpliendo con los objetivos establecidos y apoyando el progreso de los estudiantes.

Atención a los alumnos con necesidades específicas de apoyo educativo

Según la Ley de Educación de Andalucía, se considera alumnado con necesidades específicas de apoyo educativo:

- Aquel que presenta necesidades educativas especiales debidas a diferentes grados y tipos de capacidades personales de orden físico, psíquico, cognitivo o sensorial.
- El que, por proceder de otros países o por cualquier otro motivo, se incorpore de forma tardía al sistema educativo.
- El alumnado que precise de acciones de carácter compensatorio.
- Al que presenta altas capacidades intelectuales.

Atención a los diferentes ritmos de aprendizaje

Actividades complementarias y extraescolares

Como actividad extraescolar se asistirá durante el día 17 de diciembre de 2022 al BiznagaFest que se celebra en las instalaciones de la Escuela Técnica Superior de Ingeniería Informática de la UMA y cuenta con numerosos talleres sobre el desarrollo de software. Se coordinará con el resto del equipo educativo para la coordinación de las charlas y talleres más adecuados para el alumnado.

Se plante la actividad complementaria que durante el segundo trimestre del curso se realice un taller sobre calidad y pruebas automáticas de software en la empresa de desarrollo Freepik. Esta startup malagueña tiene un fuerte componente tecnológico y colabora habitualmente con centros públicos que imparten enseñanzas de Formación Profesional en la provincia de Málaga. Dado que la finalidad de esta actividad es la de resaltar la importancia de alcanzar los resultados de aprendizaje del módulo profesional, la posibilidad de poder descubrir que ciertas competencias a las que contribuye el módulo son muy demandadas por empresas del sector productivo relacionado con el ciclo de Técnico Superior en Desarrollo de Aplicaciones Web.

Bibliografía

Atlassian. (s.f.). *Getting Git Right*. Obtenido de Bitbucket: <https://www.atlassian.com/git/tutorials>

Bechtold, S. (s.f.). *JUnit5*. Obtenido de <https://junit.org/junit5/>

Casado Iglesias, C. (2012). *Entornos de Desarrollo*. Paracuellos del Jarama: RA-MA.

CBI. (07 de febrero de 2022). *The European market potential for software development services*. Obtenido de <https://www.cbi.eu/market-information/outsourcing-itobpo/software-development-services/market-potential#:~:text=Software%20developer%20shortage,to%20a%20shortage%20of%20500%2C000%20>

Dangit, Git!?! (s.f.). Obtenido de <https://dangitgit.com/es>

Dudler, R. (01 de 06 de 2023). *git - la guía sencilla*. Obtenido de <https://rogerdudler.github.io/git-guide/index.es.html>

- Faber, S. (s.f.). *Mockito*. Obtenido de <https://javadoc.io/doc/org.mockito/mockito-core/latest/org/mockito/Mockito.html>
- Foundation, E. (s.f.). *Eclipse Documentation*. Obtenido de <https://www.eclipse.org/documentation/>
- FP, O. (21 de 06 de 2023). *CaixaBank Dualiza*. Obtenido de <https://www.observatoriofp.com/indicadores-destacados/andalucia>
- Jetbrains. (s.f.). *IntelliJ IDEA*. Obtenido de <https://www.jetbrains.com/help/idea/getting-started.html>
- Martin, R. C. (2008). *Clean code: a handbook of agile software craftsmanship*. Upper Saddle River, NJ: Prentice Hall.
- Microsoft. (s.f.). *Visual Studio Code Docs*. Obtenido de <https://code.visualstudio.com/docs>
- Oracle. (02 de 2004). *Javadoc Tool*. Obtenido de <https://www.oracle.com/es/technical-resources/articles/java/javadoc-tool.html>
- Pressman, R. S. (2010). *Ingenieria del software. Un enfoque práctico*. Ciudad de México: Mc Graw Hill.
- Selenium. (7 de 12 de 2021). *WebDriver documentation*. Obtenido de <https://www.selenium.dev/documentation/webdriver/>
- Sourcemaking. (s.f.). *Refactoring*. Obtenido de <https://sourcemaking.com/refactoring>
- Sourcemaking. (s.f.). *UML*. Obtenido de <https://sourcemaking.com/uml>

Anexos

Anexo I. Indicadores de evaluación de la actividad docente

Indicador de logro	Puntuación de 0 a 10	Observaciones
Diseño estrategias y programo actividades variadas y creativas que se ajustan de manera precisa a los objetivos didácticos planteados, abordando diferentes tipos de contenidos y adaptándolas de manera efectiva a las características individuales de los alumnos		
Utilizo métodos de evaluación formativa y sumativa de manera equilibrada, diseñando instrumentos de evaluación adecuados que permiten medir de forma precisa el logro de los objetivos didácticos establecidos, considerando tanto el progreso individual de los estudiantes como el desempeño grupal		
Implemento estrategias de enseñanza diferenciadas que atienden a la diversidad de los alumnos, adaptando los contenidos y las actividades de acuerdo con las necesidades, estilos de aprendizaje y ritmos de cada estudiante, garantizando así un entorno inclusivo y propicio para el aprendizaje de todos		
Los instrumentos de seguimiento han permitido conocer el progreso de los alumnos		
La programación se ha realizado en coordinación con el resto del equipo educativo		
Las medidas de atención a la diversidad han sido suficientes y ajustadas a las necesidades del alumnado		
Fomento el desarrollo de habilidades socioemocionales y competencias transversales en mis alumnos, integrando en las estrategias y actividades didácticas aspectos como el trabajo en equipo, la resolución de problemas, la comunicación efectiva y la reflexión crítica, con el objetivo de prepararlos de manera integral para enfrentar los desafíos del siglo XXI		
Selecciono y estructuro los contenidos del curso de manera que estén alineados con las demandas y tendencias actuales del mercado laboral, integrando conceptos, habilidades y conocimientos que son relevantes y necesarios para que los estudiantes puedan enfrentar con éxito los desafíos y requerimientos del entorno profesional		

Anexo II. Formulario autoevaluación programación didáctica

Indicador de logro	Puntuación de 0 a 10	Observaciones
La selección de contenidos se ajusta a las necesidades e intereses del alumnado		
La temporalización y secuenciación de actividades ha sido ajustada		
Los criterios de evaluación y calificación son claros y conocidos por los alumnos.		
Los instrumentos de seguimiento han permitido conocer el progreso de los alumnos		
La programación se ha realizado en coordinación con el resto del equipo educativo		
Las medidas de atención a la diversidad han sido suficientes y ajustadas a las necesidades del alumnado		