PROGRAMACIÓN DIDÁCTICA F.P.

1.º DESARROLLO DE APLICACIONES WEB (DAW)

MÓDULO: PROGRAMACIÓN

ENCARNACIÓN PERÁLVAREZ AGUILERA

CURSO 2026/2027

ASPECTOS GENERALES

1. Contextualización y relación con el Plan de Centro

2. Marco Legal

3. Objetivos Generales del Ciclo Formativo

4. Organización, secuenciación y temporalización de los contenidos.

5. Fase de Formación en empresa

6. Principios Pedagógicos y Metodológicos

7. Evaluación

8. Atención a la diversidad

9.- Bibliografía

ÍNDICE

1. Contextualización y relación con el Plan de Centro **6**
1.1. Ubicación de la enseñanza 6
1.2. Cualificaciones profesionales de referencia y estándares de competencia 6
1.3. Descripción del centro, contexto socioeconómico y entorno productivo 6
1.4. Integración con los planes y proyectos del centro 7
2. Marco Legal **8**
2.1. Normativa estatal de formación profesional 8
2.2. Normativa autonómica de formación profesional 9
3. Objetivos Generales del Ciclo Formativo **10**
3.1. Objetivos generales 10
3.2. Competencias profesionales y para la empleabilidad del título 11
3.3. Resultados de aprendizaje y criterios de evaluación 12
4. Organización, secuenciación y temporalización de los contenidos. **14**
4.1. Contenidos 14
4.2. Secuenciación de los contenidos 14
4.3. Unidades didácticas 15
4.4 Interdisciplinariedad 32
4.5. Elementos transversales 33
5. Fase de Formación en Empresa (Dualización) 34
5.1. Marco General y Coordinación Centro-Empresa 34
5.2. Régimen y Distribución Horaria (Ley Orgánica 3/2022) 34
5.3. Concreción en el Módulo de Programación (1.º DAW) 34
6. Principios Pedagógicos y Metodológicos **35**
6.1. Orientaciones Metodológicas 35
6.2. Actividades de enseñanza-aprendizaje 35
6.3. Recursos y Materiales Didácticos 36
7. Evaluación **37**
7.1. Técnicas, Instrumentos y Recogida de Evidencias 37
7.2 Ponderación RA/CE, calificación y unidades didácticas. 37
7.3 Matriz de ponderación de Criterios de Evaluación y Unidades Didácticas. 50
7.4 Sistema de recuperación, subida de notas y evaluación Final 1ª y 2ª 58
7.4.1 Criterio de superación del módulo y RA imprescindibles. 58
7.4.2 Recuperación y mejora. 58
7.4.3 Evaluaciones parciales y finales 59
7.4.4 Pérdida del derecho a evaluación continua 59
7.4.5 Formación en empresa y evaluación 59
7.4.6 Insuficiencia de evidencias, conservación y garantías 59
8. Atención a la diversidad **60**
8.1. Marco General y Principios de la Respuesta Educativa 60
8.2. Medidas de Adaptación Curricular No Significativa 60

8.3. Diagnóstico de la Evaluación Inicial 61
9.- Bibliografía **61**

1. Contextualización y relación con el Plan de Centro

1.1. Ubicación de la enseñanza

El IES Jándula es un centro público situado en Andújar (Jaén) con una oferta educativa amplia y diversificada que integra Educación Secundaria Obligatoria, Bachillerato y Formación Profesional. La Formación Profesional ocupa un lugar relevante dentro de la oferta del centro y favorece su conexión con el entorno económico y productivo de Andújar y su comarca.

El Proyecto Educativo apuesta por una Formación Profesional conectada con el entorno productivo, la innovación, la orientación profesional, la formación en empresa u organismo equiparado, la empleabilidad y la internacionalización. La presente programación del módulo profesional programación, perteneciente al ciclo 1.º DAW, se desarrolla de acuerdo con estos principios y se adaptará a las características del alumnado, al perfil profesional y a la evolución del entorno productivo.

1.2. Cualificaciones profesionales de referencia y estándares de competencia

- **Cualificación de Referencia del Módulo:**
  - **IFC080_3:** *Programación con lenguajes orientados a objetos y bases de datos relacionales* (Real Decreto 295/2004).
- **Cualificación Global del Título DAW :**
  - **IFC154_3:** *Desarrollo de aplicaciones con tecnologías Web* (Real Decreto 1087/2005).

La acreditación del módulo profesional **0485 Programación** otorga la correspondencia directa con la siguiente Unidad de Competencia (UC):

- **Código de UC: UC0227_3**
- **Denominación:** *Desarrollar componentes software en lenguajes de programación orientados a objetos*.
- **Módulo Formativo Asociado en el CNCP: MF0227_3** (*Programación orientada a objetos*).

1.3. Descripción del centro, contexto socioeconómico y entorno productivo

El IES Jándula es un centro público de Educación Secundaria situado en Andújar (Jaén), con una oferta educativa amplia y diversificada que integra Educación Secundaria Obligatoria, Bachillerato y Formación Profesional. La Formación Profesional ocupa un lugar relevante dentro de la oferta del centro y favorece su conexión con el entorno económico y productivo de Andújar y su comarca.

El instituto escolariza alumnado de Andújar y de su zona de influencia, incluyendo alumnado procedente de núcleos rurales y zonas dispersas del término municipal, y se inserta en un entorno social, económico, cultural y

laboral heterogéneo. Esta realidad requiere una respuesta educativa basada en la equidad, la inclusión, la personalización del aprendizaje, la orientación, la convivencia positiva y la coordinación docente. La economía local se vincula, entre otros sectores, a los servicios, la agricultura, la actividad comercial, el transporte, la logística, la industria, el turismo de naturaleza, el patrimonio cultural y el tejido empresarial de la comarca. Esta realidad ofrece oportunidades relevantes para la orientación académica y profesional del alumnado, especialmente en Bachillerato y Formación Profesional, así como para la colaboración con empresas, instituciones y entidades del entorno. El

IES Jándula cuenta con una diversa oferta educativa:

- Enseñanza de Educación Secundaria Obligatoria y Bachillerato, donde Bachillerato posee la modalidad de Humanidades y Ciencias Sociales y Ciencias.
- Formación Profesional Básica de Informática y Comunicaciones.
- Ciclos Formativos de Grado Medio: Guía en el medio natural y de tiempo libre, y Sistemas microinformáticos y redes.
- Ciclos Formativos de Grado Superior: Desarrollo de Aplicaciones Multiplataforma, Desarrollo de Aplicaciones Web y Mecatrónica industrial.

Las instalaciones de las que consta el centro son: gimnasio, biblioteca, laboratorios de ciencia y tecnología, música, educación especial, pistas deportivas, salas de profesores, salón de actos y aulas de Informática, todas ellas equipadas con las herramientas necesarias en cada caso. El centro está liderado por un equipo directivo, formado por la Directora, el Vicedirector, el Secretario, el Jefe de Estudios y dos adjuntos. El Proyecto Educativo apuesta por una Formación Profesional conectada con el entorno productivo, la innovación, la orientación profesional, la formación en empresa u organismo equiparado, la empleabilidad y la internacionalización. La presente programación del módulo profesional Programación, perteneciente al ciclo CFGS Desarrollo de Aplicaciones Web, se desarrolla de acuerdo con estos principios y se adaptará a las características del alumnado, al perfil profesional y a la evolución del entorno productivo.

1.4. Integración con los planes y proyectos del centro

El centro viene desarrollando una serie de Planes y Proyectos educativos promovidos por la Consejería de Educación de la Junta de Andalucía con el fin de introducir mejoras en las prácticas educativas del profesorado para adecuarlas a las demandas y necesidades actuales de la sociedad. Estos proyectos están alineados en gran medida con el tratamiento de los temas transversales que más adelante se verán y algunos de ellos son: Plan de Transformación Digital Docente, Programa Aldea, Plan de Igualdad de género

en Educación en Andalucía, Plan de Convivencia, Proyecto STEAM 4.0, Proyecto Erasmus+, Programa Más Deporte.

2. Marco Legal

La presente programación se elabora conforme a la Ley Orgánica 3/2022, al Real Decreto 659/2023, al Decreto 147/2025, a la Orden de 18 de septiembre de 2025 que regula la evaluación de los grados D y E en Andalucía, a la normativa específica del título y currículo correspondiente y a los criterios establecidos en el Plan de Centro del IES Jándula. En caso de contradicción, prevalecerá la normativa específica aplicable a la oferta.

La elaboración de la Programación Didáctica del módulo profesional de Programación 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.

2.1. Normativa estatal de formación profesional

● El Real Decreto 686/2010, de 20 de mayo, por el que se establece el título de Técnico Superior en Desarrollo de Aplicaciones Web.

● Real Decreto 1147/2011, de 29 de julio, por el que se establece la ordenación general de la formación profesional del sistema educativo.

● 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 3/2022, de 31 de marzo, de ordenación e integración de la Formación Profesional.

● Real Decreto 659/2023, de 18 de julio, por el que se desarrolla la ordenación del Sistema de Formación Profesional.

● Real Decreto 500/2024, de 21 de mayo, por el que se modifican determinados reales decretos por los que se establecen títulos de Formación Profesional de grado superior y se fijan sus enseñanzas mínimas.

● Real Decreto 69/2025, de 4 de febrero, por el que se desarrollan los elementos integrantes y los instrumentos de gestión del Sistema Nacional de Formación Profesional, y se modifica el Real Decreto 375/1999, de 5 de marzo, por el que se crea el Instituto Nacional de las Cualificaciones.

● Real Decreto 532/2025, de 24 de junio, por el que se incluyen determinados estándares de competencias profesionales y se integran los estándares de competencias profesionales derivados de las antiguas unidades de competencia establecidas al amparo del Real Decreto 1128/2003, de 5 de septiembre, por el que se regula el Catálogo Nacional de Cualificaciones Profesionales, en el Catálogo Nacional de Estándares de Competencias Profesionales.

2.2. Normativa autonómica de formación profesional

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

● 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.

● 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.

● Resolución de 26 de junio de 2024, de la Dirección General de Formación Profesional, por la que se dictan Instrucciones para regular aspectos relativos a la organización y al funcionamiento del curso 2024/2025 en la Comunidad Autónoma de Andalucía.

● Decreto 147/2025, de 17 de septiembre de 2025, por el que se establece la ordenación de las enseñanzas de los Grados D y E del Sistema de Formación Profesional en la Comunidad Autónoma de Andalucía.

● Orden de 18 de septiembre de 2025, por la que se regula la evaluación, certificación, acreditación y titulación académica del alumnado que cursa enseñanzas de los grados D y E del Sistema de Formación Profesional en la Comunidad Autónoma de Andalucía.

● Orden de 26 de septiembre de 2025, por la que se regula la fase de formación en empresa u organismo equiparado de los grados D y E del Sistema de Formación Profesional de la Comunidad Autónoma de Andalucía.

●Instrucciones de 20 de mayo de 2026 de la Dirección General de Formación Profesional y Educación Permanente, por la que se establecen aspectos relativos a la evaluación final del alumnado de las enseñanzas de los grados D y E del Sistema de Formación Profesional para el curso 2025/2026 en los centros docentes de la Comunidad Autónoma de Andalucía.

● Resolución de 24 de julio de 2026 de la dirección general de formación profesional y educación permanente por la que se dictan instrucciones para determinar aspectos relativos a la distribución horaria y a la concreción curricular de las enseñanzas correspondientes a los grados d y e del sistema de formación profesional para el curso escolar 2026/2027 en la comunidad autónoma de Andalucía, conforme a lo dispuesto en el real decreto 659/2023, de 18 de julio, por el que se desarrolla la ordenación del sistema de formación profesional.

● Decreto 483/2026, de 31 de agosto, por el que se adoptan medidas para la mejora del funcionamiento del sistema educativo y de la gestión de la actividad docente en la enseñanza no universitaria en Andalucía.

3. Objetivos Generales del Ciclo Formativo

Los objetivos generales del ciclo, competencias profesionales y para la empleabilidad, resultados de aprendizaje y criterios de evaluación serán los establecidos en la normativa del título y currículo. Cuando estén incorporados en la fuente institucional utilizada, no se reproducirá innecesariamente. El trabajo curricular del departamento se centra en su secuenciación, ponderación, evidencias, unidades de trabajo y distribución centro-empresa.

3.1. Objetivos generales

Los objetivos generales de las enseñanzas correspondientes al título de formación profesional de Técnico Superior en Desarrollo de Aplicaciones Web son los siguientes:

a) Ajustar la configuración lógica del sistema 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 de protección de datos e infraestructuras.

c) Seleccionar y utilizar sistemas gestores de bases de datos interpretando su diseño lógico para administrar la información de forma eficiente y segura.

d) Evaluar y utilizar herramientas informáticas y entornos integrados de desarrollo (IDE) para la edición, prueba y depuración de código de aplicaciones web.

e) Seleccionar y utilizar lenguajes de programación, objetos y herramientas para desarrollar la arquitectura interna y la lógica de negocio de las aplicaciones.

f) Generar componentes de acceso a datos cumpliendo las especificaciones técnicas para integrar contenidos en la lógica de una aplicación web.

g) Utilizar lenguajes de marcas y estándares web, asumiendo manuales de estilo y diseño, para desarrollar interfaces de usuario adaptativas e interactivas.

h) Desarrollar componentes software en el entorno cliente integrando la interacción con el usuario, la usabilidad y el consumo de servicios web.

i) Desarrollar e integrar componentes software en el entorno del servidor web, empleando herramientas y lenguajes específicos para dar respuesta a los requerimientos del sistema.

j) Desarrollar servicios web para integrar sus funciones en el despliegue de otras aplicaciones.

k) Integrar componentes multimedia en la interfaz web para optimizar la experiencia de usuario y la accesibilidad.

l) Desplegar y distribuir aplicaciones web en distintos ámbitos de implantación (servidores web, servidores de aplicaciones, repositorios), verificando su comportamiento y rendimiento.

m) Elaborar y mantener la documentación técnica de los procesos de desarrollo utilizando herramientas de generación de documentación y control de versiones.

n) Evaluar y aplicar técnicas de prueba, verificación y depuración software para asegurar la calidad y optimización de las aplicaciones web.

ñ) Analizar y utilizar los recursos de aprendizaje autónomo y las oportunidades de formación continua vinculadas a la evolución tecnológica del sector TIC.

o) Evaluar situaciones de prevención de riesgos laborales y de protección ambiental asociadas a la actividad profesional en entornos informáticos.

p) Identificar los derechos y obligaciones derivados de las relaciones laborales dentro del marco legal del sector del desarrollo de software.

q) Reconocer e impulsar la iniciativa emprendedora, evaluando la viabilidad de proyectos para la creación de empresas informáticas o el trabajo autónomo.

r) Desarrollar habilidades de comunicación, trabajo en equipo, empatía y resolución conjunta de problemas en entornos colaborativos de desarrollo.

3.2. Competencias profesionales y para la empleabilidad del título

- **Configuración de Sistemas:** Configurar y explotar sistemas informáticos, adaptando la configuración lógica del sistema según las necesidades de uso y los criterios establecidos.
- **Seguridad Informática:** Aplicar técnicas y procedimientos relacionados con la seguridad en sistemas, servicios y aplicaciones, cumpliendo el plan de seguridad.
- **Gestión de Bases de Datos:** Gestionar bases de datos, interpretando su diseño lógico y garantizando la integridad, consistencia, accesibilidad y seguridad de la información.
- **Desarrollo Front-End:** Desarrollar elementos software en el entorno cliente, utilizando lenguajes de marcas, hojas de estilo y lenguajes de guión interactivos.
- **Desarrollo Back-End:** Desarrollar elementos software en el entorno servidor, empleando lenguajes de programación y herramientas específicas para integrar la lógica de negocio.
- **Diseño e Interfaces de Usuario:** Desarrollar interfaces de usuario interactivas, adaptativas y accesibles en aplicaciones web, garantizando la usabilidad (WCAG/WAI).
- **Integración Multimedia:** Integrar componentes multimedia en el desarrollo de aplicaciones web, optimizando su rendimiento y la experiencia de usuario.
- **Servicios Web e Interconexión:** Desarrollar e integrar servicios web y componentes distribuidos para extender las funcionalidades del software.

- **Despliegue y Explotación:** Desplegar y publicar aplicaciones web en servidores (Internet, Intranet y Extranet), verificando su rendimiento, disponibilidad y compatibilidad.
- **Aseguramiento de Calidad y Documentación:** Diseñar y realizar pruebas de software, depurar código y redactar la documentación técnica y los manuales de usuario para garantizar la mantenibilidad del proyecto.

Competencias Personales, Sociales y Metodológicas

- **Adaptación Tecnológica y Aprendizaje Continuo:** Mantener el espíritu de innovación y adaptarse a las constantes novedades tecnológicas y transformaciones del sector TIC.
- **Resolución de Problemas:** Diagnosticar fallos, tomar decisiones y resolver incidencias técnicas en el ámbito de sus competencias.
- **Trabajo Colaborativo en Equipos:** Liderar o coordinarse dentro de equipos de desarrollo software multidisciplinares, aplicando metodologías de trabajo colaborativo.
- **Comunicación Profesional:** Establecer comunicación fluida con clientes, usuarios y miembros del equipo técnico, adaptando el lenguaje tanto en español como en inglés profesional.
- **Ética y Responsabilidad:** Organizar y coordinar el trabajo propio con sentido de la responsabilidad, ética profesional, cumplimiento de la legislación de protección de datos y compromiso medioambiental.

Competencias para la Empleabilidad, Emprendimiento y PRL

- **Prevención de Riesgos Laborales:** Aplicar protocolos de seguridad, ergonomía y salud laboral aplicables a los puestos de trabajo en entornos informáticos.
- **Iniciativa Emprendedora:** Evaluar la viabilidad de proyectos de autoempleo o creación de empresas informáticas, analizando el marco comercial, financiero y legal.
- **Orientación e Inserción Laboral:** Identificar las oportunidades de empleo en el sector tecnológico andaluz y nacional, ejerciendo los derechos y cumpliendo los deberes derivados de la relación laboral.

3.3. Resultados de aprendizaje y criterios de evaluación

RA1. Reconocimiento de la estructura de programas:
Reconoce la estructura de un programa informático, identificando y relacionando sus elementos básicos (sintaxis, variables, tipos de datos, operadores, entornos de desarrollo e instrucciones básicas).

RA2. Uso de estructuras de control:
Escribe y prueba programas sencillos, reconociendo y aplicando las estructuras de control del lenguaje (estructuras condicionales de selección y bucles iterativos).

RA3. Programación Orientada a Objetos (POO) básica:
Desarrolla programas organizados en clases, aplicando los principios de la

programación orientada a objetos, modularidad, encapsulamiento, atributos y métodos para facilitar la reutilización y el mantenimiento del código.

RA4. Gestión de entrada y salida de datos (Persistencia básica):
Realiza operaciones de entrada y salida de información utilizando sistemas de ficheros (lectura/escritura) y las clases específicas del lenguaje de programación.

RA5. POO avanzada (Herencia y Polimorfismo):
Aplica técnicas avanzadas de la programación orientada a objetos, evaluando e implementando mecanismos de herencia, clases abstractas, interfaces y polimorfismo.

RA6. Tratamiento de excepciones y características avanzadas:
Desarrolla programas aplicando las características avanzadas del lenguaje y gestionando el control de errores mediante bloques de captura y elevación de excepciones.

RA7. Gestión de colecciones y estructuras de datos:
Utiliza las estructuras de datos avanzadas que ofrece el lenguaje (listas, conjuntos, mapas y colecciones dinámicas), aplicando operaciones sobre ellas de forma eficiente.

RA8. Acceso a bases de datos relacionales:
Desarrolla aplicaciones que acceden a bases de datos relacionales (SGBD), utilizando conectores, controladores e interfaces de programación de aplicaciones (API) específicas (p. ej., JDBC o mapeadores objeto-relacional).

RA9. Desarrollo de interfaces gráficas de usuario:
Desarrolla componentes y entornos visuales interactivos (GUI) basados en eventos, utilizando herramientas y librerías del entorno de desarrollo.

4. Organización, secuenciación y temporalización de los contenidos.

4.1. Contenidos

Los contenidos propuestos para este módulo profesional se agrupan en las siguiente Unidades didácticas:

1. Conceptos básicos
2. Programación estructurada en Java
3. Arrays en Java
4. Clases especiales en Java. Fechas, Cadenas.
5. Programación orientada a objetos
6. Interfaces
7. Ficheros
8. Colecciones.
9. Conexión a base de datos: JDBC
10. Interfaces gráficas.

4.2. Secuenciación de los contenidos

Teniendo en cuenta el calendario escolar establecido para el curso 26/27 y que este módulo profesional tiene una carga horaria de 256 horas, establecidas en 8 horas/semana, se realiza la planificación temporal de las Unidades didácticas con indicación de su duración establecida.

U.D.	Horas	1ª Eval.	2ª Eval.	3ª Eval.
1	16	X		
2	48	X		
3	18	X		

4	20	X		
5	48	X	X	
6	20		X	
7	26		X	X
8	26			X
9	20			X
10	14			X
Total	**256**			

4.3. Unidades didácticas

En las horas estimadas se incluyen las horas dedicadas a la realización de pruebas prácticas-teóricas.

Los objetivos, contenidos, resultados de aprendizaje y criterios de evaluación de la unidades didácticas definidas se recogen a continuación:

UD 1. Conceptos básicos Programación

Objetivos Didácticos

Establecer todos los conceptos básicos que son necesarios para el conocimiento de cualquier lenguaje de programación.
Concretar la importancia de los mecanismos básicos que usan los compiladores, como la declaración y asignación de variables, el establecimiento de tipos, etcétera.

Clasificar los distintos operadores, atendiendo a su uso y al resultado de la evaluación de sus expresiones.
Profundizar en el concepto de tipo primitivo, sus tamaños, rangos y conversiones, así como sus valores literales.
Asociar los tipos primitivos con sus operadores.
Asimilar las distintas formas de interacción con el usuario mediante la salida por consola y la entrada de datos desde el teclado.
Exponer el funcionamiento de herramientas (métodos) que proporciona la API para ciertas clases, la diferencia entre el uso estático y no estático de los métodos de la API, así como la clasificación de clases en los distintos paquetes.

Contenidos

1.1. Algoritmo
1.2. Lenguajes de programación
1.3. El programa principal
1.4. Palabras reservadas
1.5. Concepto de variable
1.6. Tipos primitivos
1.7. Variables de objeto
1.8. Constantes
1.9. Comentarios
1.10. API de Java
1.11. Operaciones básicas
1.12. Conversión de tipos

Resultados de Aprendizaje y Criterios de Evaluación

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

a) Se han identificado los bloques que componen la estructura de un programa informático.

b) Se han creado proyectos de desarrollo de aplicaciones.

c) Se han utilizado entornos integrados de desarrollo.

d) Se han identificado los distintos tipos de variables y la utilidad específica de cada uno.

e) Se ha modificado el código de un programa para crear y utilizar variables.

f) Se han creado y utilizado constantes y literales.

g) Se han clasificado, reconocido y utilizado en expresiones los operadores del lenguaje.

h) Se ha comprobado el funcionamiento de las conversiones de tipos explícitas e implícitas.

i) Se han introducido comentarios en el código.

UD 2. Programación Estructurada en Java

Objetivos Didácticos

Escribir y probar código que haga uso de estructuras de selección.

Clasificar, reconocer y utilizar los operadores del lenguaje en expresiones.

Reconocer la estructura de un programa informático, identificando y relacionando los elementos propios de Java.

Diseñar aplicaciones cuya ejecución no es siempre lineal, permitiendo, a partir de la entrada de datos, discriminar entre distintos escenarios.

Conocer y utilizar las estructuras de repetición.

Programar aplicaciones que repiten conjuntos de instrucciones mediante el uso de bucles.

Distinguir entre un bucle controlado por contador, un bucle precondición y un bucle postcondición.

Utilizar las estructuras adecuadas de control para conseguir que un programa funcione según las especificaciones funcionales.

Asimilar el concepto de función, las ventajas de su uso y la implicación en la mejora de mantenimiento de aplicaciones.

Entender y usar el concepto de parámetro de entrada, así como el mecanismo para generalizar el comportamiento de las funciones.

Escribir programas que hagan un uso adecuado de las funciones y del valor devuelto por estas.

Resolver problemas mediante el uso de funciones recursivas.

Contenidos

2.1. Expresiones lógicas relacionales

2.2. Condicional simple: if

2.3. Condicional doble: if-else

2.4. Condicional múltiple: switch

2.5. Conocer y utilizar las estructuras de repetición

2.6. Programar aplicaciones que repiten conjuntos de instrucciones mediante el uso de bucles

2.7. Distinguir entre un bucle controlado por contador, un bucle precondición y un bucle postcondición

2.8. Utilizar las estructuras adecuadas de control para conseguir que un programa funcione según las especificaciones funcionales

2.9. Bucles controlados por condición

2.10. Bucles controlados por contador: for

2.11. Salidas anticipadas

2.12. Bucles anidados

2.13. Conceptos básicos

2.14. Ámbito de las variables

2.15. Paso de información a una función

2.16. Valor devuelto por una función

2.17. Sobrecarga de funciones

Resultados de Aprendizaje y Criterios de Evaluación

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

b) Se han escrito programas simples.

i) Se ha utilizado el entorno integrado de desarrollo en la creación y compilación de programas simples

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

a) Se ha escrito y probado código que haga uso de estructuras de selección.

b) Se han utilizado estructuras de repetición.

c) Se han reconocido las posibilidades de las sentencias de salto.

e) Se han creado programas ejecutables utilizando diferentes estructuras de control.

f) Se han probado y depurado los programas.

g) Se ha comentado y documentado el código.

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

a) Se ha utilizado la consola para realizar operaciones de entrada y salida de información.

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

h) Se ha comentado y documentado el código.

UD 3. Arrays

Objetivos Didácticos

Conocer los arrays, que permiten almacenar múltiples valores en una variable.

Crear arrays de distinto tipo y longitudes.

Utilizar las operaciones básicas que se emplean con los arrays.

Diseñar programas que hagan uso de tablas, donde se almacenan los datos necesarios.

Modificar la longitud de un array en tiempo de ejecución sin pérdida de los datos que contiene.

Usar la API de Java relacionada con los arrays y aplicar su uso a la resolución de problemas.

Contenidos

3.1. Variables escalares versus tablas

3.2. Índices

3.3. Construcción de tablas

3.4. Referencias

3.5. Uso de tablas

3.6. Tablas como parámetros de funciones

3.7. Operaciones con tablas: la clase Arrays

3.8. Tablas n-dimensionales

Resultados de Aprendizaje y Criterios de Evaluación

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

c) Se han instanciado objetos a partir de clases predefinidas.

d) Se han utilizado métodos y propiedades de los objetos.

e) Se han escrito llamadas a métodos estáticos.

f) Se han utilizado parámetros en la llamada a los métodos.

g) Se han incorporado y utilizado librerías de objetos.

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

a) Se han escrito programas que utilicen arrays.

b) Se han reconocido las librerías de clases relacionadas con tipos de datos avanzados.

UD 4. Clases Especiales en Java: Fechas y Cadenas

Objetivos Didácticos

Utilizar el tipo primitivo `char`.

Conocer las funcionalidades que proporciona la clase `Character` para la manipulación de caracteres.

Usar la clase `String` como parte de las aplicaciones que se construyan.

Profundizar en el uso de operaciones avanzadas con texto y realizar implementaciones acordes a los requisitos del sistema.

Plantear distintas alternativas y elegir la solución óptima en cada caso, según el problema que se necesite resolver.

Conocer la API de Java que permite codificar aplicaciones que utilizan datos de tipo texto.

Exponer las ventajas e inconvenientes de las posibles herramientas que se usan para la creación y manipulación de texto.

Usar la API de fechas de Java.

Contenidos

4.1. Tipo primitivo `char`

4.2. Clase `Character`

4.3. Clase `String`

4.4. Cadenas y tablas de caracteres

4.5. Clase `StringBuffer`

4.6. Clase `StringBuilder`

4.7. Clases `LocalDateTime` y `LocalDate`

4.8. Clases de Zona y Lugar en Java

Resultados de Aprendizaje y Criterios de Evaluación

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

h) Se han utilizado constructores.

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

d) Se ha escrito código utilizando control de excepciones.

h) Se han creado excepciones.

i) Se han utilizado aserciones para la detección y corrección de errores durante la fase de desarrollo.

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

b) Se han aplicado formatos en la visualización de la información.

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

g) Se han utilizado expresiones regulares en la búsqueda de patrones en cadenas de texto.

UD 5. Programación Orientada a Objetos en Java

Objetivos Didácticos

Comprender y asimilar los conceptos de la POO.

Escribir programas que hagan uso de la POO para solucionar problemas, facilitando la tarea del programador.

Decidir si los métodos y atributos serán visibles o no para clases externas y vecinas.

Implementar métodos que permitan la asignación controlada de los atributos no visibles, así como otros que posibiliten consultar los valores de estos atributos.

Conocer la idea y la necesidad de la herencia entre clases.

Saber cuáles son los conceptos de subclase y superclase.

Utilizar la herencia como mecanismo de especialización.

Conocer las limitaciones de acceso a los miembros de una superclase.

Comprender y usar el mecanismo de la sustitución de métodos.

Utilizar el acceso a miembros sustituidos de una superclase.

Comprender y usar la selección dinámica de métodos en tiempo de ejecución.

Reconocer las principales funcionalidades definidas en la clase Object.

Usar la sustitución de los métodos de Object. En particular, la implementación de toString() y equals().

Conocer y utilizar las clases abstractas.

Usar las clases abstractas para la selección dinámica de métodos.

Contenidos

5.1. Definición de una clase

5.2. Crear una clase

5.3. Atributos

5.4. Objetos

5.5. Métodos

5.6. Atributos y métodos estáticos

5.7. Constructores

5.8. Paquetes

5.9. Modificadores de acceso

5.10. Enumerados

5.11. Subclase y superclase

5.12. Modificador de acceso para herencia

5.13. Redefinición de miembros heredados

5.14. La clase Object

5.15. Clases abstractas

Resultados de Aprendizaje y Criterios de Evaluación

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

a) Se han identificado los fundamentos de la programación orientada a objetos.

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

a) Se ha reconocido la sintaxis, estructura y componentes típicos de una clase.

b) Se han definido clases.

c) Se han definido propiedades y métodos.

d) Se han creado constructores.

e) Se han desarrollado programas que instancien y utilicen objetos de las clases creadas anteriormente.

f) Se han utilizado mecanismos para controlar la visibilidad de las clases y de sus miembros.

g) Se han definido y utilizado clases heredadas.

h) Se han creado y utilizado métodos estáticos.

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

a) Se han identificado los conceptos de herencia, superclase y subclase.

b) Se han utilizado modificadores para bloquear y forzar la herencia de clases y métodos.

c) Se ha reconocido la incidencia de los constructores en la herencia.

d) Se han creado clases heredadas que sobrescriban la implementación de métodos de la superclase.

e) Se han diseñado y aplicado jerarquías de clases.

f) Se han probado y depurado las jerarquías de clases.

g) Se han realizado programas que implementen y utilicen jerarquías de clases.

UD 6. Interfaces

Objetivos Didácticos

Conocer la idea y la necesidad de las interfaces y distinguirlas de las clases abstractas.

Definir métodos abstractos en una interfaz.

Implementar una o más interfaces en una clase.

Implementar métodos de extensión en una interfaz.

Implementar métodos privados como auxiliares en una interfaz.

Diseñar interfaces para operaciones específicas en clases diversas.

Utilizar variables de tipo interfaz para conseguir la selección dinámica de métodos.

Conocer qué son la herencia simple y múltiple de interfaces.

Conocer algunas interfaces importantes de la API.

Saber implementar clases anónimas.

Implementar criterios de comparación y aplicarlos a procesos de búsqueda y ordenación.

Contenidos

6.1. Concepto de interfaz

6.2. Atributos de una interfaz

6.3. Métodos implementados en una interfaz

6.4. Herencia

6.5. Variables de tipo interfaz

6.6. Clases anónimas

6.7. Acceso entre miembros de una interfaz

6.8. Sintaxis general

6.9. Un par de interfaces de la API

Resultados de Aprendizaje y Criterios de Evaluación

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

f) Se han creado clases y métodos genéricos.

h) Se han identificado las clases relacionadas con el tratamiento de documentos escritos en diferentes lenguajes de intercambio de datos.

i) Se han realizado programas que realicen manipulaciones sobre documentos escritos en diferentes lenguajes de intercambio de datos.

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

i) Se han identificado y evaluado los escenarios de uso de interfaces.

j) Se han identificado y evaluado los escenarios de utilización de la herencia y la composición.

UD 7. Ficheros

Objetivos Didácticos

Conocer el concepto de excepción y sus tipos.

Manipular excepciones.

Crear excepciones de usuario.

Saber qué es el concepto de flujos de entrada y salida.

Crear flujos de entrada de texto.

Conocer y aplicar funciones de lectura en flujos de entrada de texto.

Utilizar la entrada formateada desde archivos de texto con Scanner.

Crear flujos de salida de texto.

Conocer y aplicar funciones de escritura en flujos de salida de texto.

Cerrar flujos de entrada y salida de texto.

Leer y escribir entre archivos XML y aplicaciones Java con JAXB.

Abrir y cerrar archivos binarios.

Escribir datos primitivos y objetos en archivos binarios.

Leer datos primitivos y objetos de archivos binarios.

Conocer las distintas excepciones que se pueden arrojar durante la apertura y cierre de archivos binarios.

Identificar las distintas excepciones que se pueden arrojar durante la escritura y lectura de datos en archivos binarios.

Gestionar sistemas complejos donde se guarda y recupera información.

Contenidos

7.1. Excepciones

7.2. Flujos de entrada de texto

7.3. Scanner y flujos de entrada

7.4. Flujos de salida de texto

7.5. Ficheros XML y Java: API JAXB

7.6. Flujos de salida binarios

7.7. Flujos de entrada binarios

7.8. Ficheros binarios y objetos complejos

Resultados de Aprendizaje y Criterios de Evaluación

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

c) Se han reconocido las posibilidades de entrada/salida del lenguaje y las librerías asociadas.

d) Se han utilizado ficheros para almacenar y recuperar información.

e) Se han creado programas que utilicen diversos métodos de acceso al contenido de los ficheros.

UD 8. Colecciones

Objetivos Didácticos

Implementar clases, interfaces y métodos con tipos genéricos.

Conocer la interfaz Collection y sus métodos básicos y globales.

Conocer la interfaz List y sus métodos.

Utilizar las implementaciones de List: ArrayList y LinkedList.

Conocer la interfaz Set.

Usar las implementaciones de Set: HashSet, TreeSet y LinkedHashSet.

Emplear las conversiones entre distintas implementaciones de List y Set.

Conocer la interfaz Map.

Usar los métodos de Map a través de sus implementaciones HashMap, TreeMap y LinkedHashMap.

Emplear las vistas Collection de los mapas.

Contenidos

8.1. Tipos parametrizados o genéricos

8.2. Interfaz Collection

8.3. Métodos específicos de la interfaz List

8.4. Interfaz Set

8.5. Conversiones entre colecciones

8.6. Clase Collections

8.7. Interfaz Map

Resultados de Aprendizaje y Criterios de Evaluación

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

i) Se han creado y utilizado conjuntos y librerías de clases.

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

c) Se han utilizado listas para almacenar y procesar información.

d) Se han utilizado iteradores para recorrer los elementos de las listas.

e) Se han reconocido las características y ventajas de cada una de las colecciones de datos disponibles.

j) Se han utilizado operaciones agregadas para el manejo de información almacenada en colecciones.

UD 9. Conexión a Base de Datos: JDBC

Objetivos Didácticos

Realizar la persistencia de los datos de una aplicación mediante los servicios de un SGBD.

Configurar el driver JDBC para poder acceder a SGBD de distintos fabricantes.

Crear una conexión entre un programa Java y un SGBD.

Ejecutar instrucciones de un SGBD desde una aplicación que funciona como cliente.

Insertar los datos de una aplicación en una base de datos.

Recuperar información almacenada en una base de datos, para su posterior uso, desde un programa Java.

Realizar el mapeo objeto-relacional de clases simples.

Contenidos

9.1. API JDBC

9.2. Driver

9.3. Conexión

9.4. Ejecución de sentencias

9.5. Clase ResultSet

9.6. SQL Injection

9.7. Sentencias parametrizadas

9.8. Operaciones CRUD

9.9. Objeto de Acceso a Datos (DAO)

Resultados de Aprendizaje y Criterios de Evaluación

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

a) Se han identificado las características de las bases de datos orientadas a objetos.

b) Se ha analizado su aplicación en el desarrollo de aplicaciones mediante lenguajes orientados a objetos.

c) Se han instalado sistemas gestores de bases de datos orientados a objetos.

d) Se han clasificado y analizado los distintos métodos soportados por los sistemas gestores para la gestión de la información almacenada.

e) Se han creado bases de datos y las estructuras necesarias para el almacenamiento de objetos.

f) Se han programado aplicaciones que almacenen objetos en las bases de datos creadas.

g) Se han realizado programas para recuperar, actualizar y eliminar objetos de las bases de datos.

h) Se han realizado programas para almacenar y gestionar tipos de datos estructurados, compuestos y relacionados.

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

a) Se han identificado las características y métodos de acceso a sistemas gestores de bases de datos.

b) Se han programado conexiones con bases de datos.

c) Se ha escrito código para almacenar información en bases de datos.

d) Se han creado programas para recuperar y mostrar información almacenada en bases de datos.

e) Se han efectuado borrados y modificaciones sobre la información almacenada.

f) Se han creado aplicaciones que muestran la información almacenada en bases de datos.

g) Se han creado aplicaciones para gestionar la información presente en bases de datos.

UD 10. Interfaces Gráficas

Objetivos Didácticos

Utilizar herramientas del entorno de desarrollo para crear interfaces gráficas de usuario simples.

Programar controladores de eventos.

Realizar programas que utilicen interfaces gráficas para la entrada y salida de la información.

Contenidos

10.1. Creación de interfaces gráficas simples

10.2. Controladores de eventos

10.3. Interfaces gráficas para la entrada y salida de la información

Resultados de Aprendizaje y Criterios de Evaluación

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

f) Se han utilizado las herramientas del entorno de desarrollo para crear interfaces gráficas de usuario simples.

g) Se han programado controladores de eventos.

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

4.4 Interdisciplinariedad

La interdisciplinariedad se concibe como una coordinación curricular activa con el resto de módulos del primer curso. Al ser el módulo con mayor carga horaria (256 horas totales / 8 horas semanales), actúa como el eje vertebrador del perfil técnico del título.

Con *Entornos de Desarrollo* (Código 0487):

- **Uso unificado de herramientas (IDE):** Coordinación en el manejo de los entornos de desarrollo integrados (*IntelliJ IDEA*, *Eclipse* o *NetBeans*) para edición y ejecución de código Java.
- **Control de versiones:** Aplicación práctica de *Git* y *GitHub* para el control de versiones de los proyectos y prácticas de programación creados por el alumnado.
- **Modelado y diseño UML:** Utilización de diagramas de clases UML trabajados en Entornos de Desarrollo como paso previo a la implementación del código orientado a objetos en Programación.
- **Depuración y calidad:** Aplicación de técnicas de depuración e inspección de variables sobre el código desarrollado.

Con *Bases de Datos* (Código 0484):

- **Persistencia relacional y acceso a datos:** Coordinación entre el diseño de tablas y modelos relacionales (SQL) en Bases de Datos y su manipulación desde el código Java en la UT 10 de Programación mediante la API **JDBC**.
- **Operaciones CRUD y seguridad:** Implementación de sentencias SQL parametrizadas (PreparedStatement) para la consulta, inserción, modificación y borrado de registros, así como la prevención de vulnerabilidades como la *SQL Injection*.

Con *Lenguajes de Marcas y Sistemas de Gestión de Información* (Código 0488):

- **Estructuración e intercambio de datos:** Integración de formatos como **XML y JSON** analizados en Lenguajes de Marcas para el almacenamiento y la lectura de información persistente en las aplicaciones.
- **Serialización de objetos:** Procesamiento de ficheros XML desde programas Java mediante tecnologías de mapeo y lectura/escritura (API JAXB) en la UT 7.

Con *Sistemas Informáticos* (Código 0483):

- **Entorno de ejecución:** Configuración del entorno de desarrollo Java (JDK, máquina virtual JVM y variables de entorno como JAVA_HOME y PATH).
- **Gestión de recursos y ficheros:** Comprensión de los mecanismos del sistema operativo para la gestión de memoria, permisos de lectura/escritura y rutas en el sistema de archivos.

Con *Itinerario Personal para la Empleabilidad / FOL*:

- **Prevención de riesgos ergonómicos:** Adopción de normas ergonómicas e higiene postural frente a pantallas de visualización de datos (PVD) durante la codificación

4.5. Elementos transversales

Para desarrollar y trabajar los temas transversales a lo largo del módulo, se plantea el desarrollo habitual de las clases y las actividades correspondientes, utilizando técnicas que lo permitan y haciendo referencia en fechas señaladas, como en el Día de la Mujer, el Día contra la Violencia de Género, el Día de Andalucía, el Día de la Constitución. Concretamente, se trabajarán los siguientes elementos transversales:

- Educación moral y cívica: para adoptar hábitos positivos y solidarios en la convivencia en sociedad, se rechazará al uso de software obtenido de manera ilegal, y se valorará el tratamiento adecuado de información sensible con lo establecido en la Ley Orgánica de Protección de Datos de Carácter Personal.
- Educación para la paz: durante las sesiones de clase, el alumnado aportará sus ideas y compartirá sus opiniones sobre cuestiones que el profesor plantee. En las intervenciones, deberán respetarse unos a otros y ser tolerantes con las opiniones ajenas. No se permitirán conductas racistas, discriminatorias, sexistas o intolerantes.
- Educación para la igualdad de oportunidades de ambos sexos: se fomentará en el alumnado la reflexión sobre este tema, a través del uso de términos que hacen referencia a personas sin diferencia de sexo, concienciar el trato igualitario entre hijos e hijas en el entorno familiar y escolar, etcétera.
- Educación ambiental: el alumnado deberá tratar debidamente el material, hacer una utilización responsable de este, apagando monitores y ordenadores cuando no se esté haciendo uso de estos, e intentar que los documentos que se utilicen sean en formato digital.
- Educación para la salud: se pretenderá que el alumnado reflexione en aspectos que inciden en su salud, manteniendo un buen estado tanto físico como mental a través de recomendaciones como la corrección de la posición al sentarse ante el ordenador, el ajuste del asiento de forma cómoda y confortable, el uso aconsejable del ratón y el teclado, estiramientos de hombros y brazos, la distancia y la disposición del monitor ante el alumnado.
- Patrimonio cultural: el alumnado participa y conoce algunas manifestaciones culturales y artísticas de su entorno, fomentando el desarrollo de una actitud de interés y aprecio hacia la Cultura Andaluza y de valoración y respeto a la pluralidad cultural.

Asimismo, debemos de prestar atención a las áreas prioritarias o de especial interés, existentes en la Formación Profesional: TIC, idiomas y prevención de riesgos laborales.

5. Fase de Formación en Empresa (Dualización)

5.1. Marco General y Coordinación Centro-Empresa

La formación en empresa u organismo equiparado se integrará en el proceso formativo conforme a la planificación del ciclo y a la normativa vigente. La programación identificará los **Resultados de Aprendizaje (RA)** y **Criterios de Evaluación (CE)** que podrán desarrollarse total o parcialmente en la empresa, garantizando la coordinación entre el centro educativo y el entorno profesional.

- **Tutoría y evaluación:** La tutoría dual de la empresa informará y valorará la adquisición de los RA previstos durante la estancia. El profesorado responsable del módulo incorporará dicha valoración a la evaluación y calificación conforme a lo establecido en la programación.

- **Gestión documental:** La empresa concreta, los calendarios, periodos, tutorías, rotaciones y datos individualizados se recogerán en el *Plan de Formación Inicial*, los *Planes de Formación Individuales*, la plataforma *Séneca* y los documentos oficiales correspondientes, evitando duplicidades en el cuerpo de esta programación.

5.2. Régimen y Distribución Horaria (Ley Orgánica 3/2022)

En cumplimiento de la **Ley Orgánica 3/2022**, de ordenación e integración de la Formación Profesional, el alumnado desarrollará parte de su aprendizaje en la empresa tanto en el primer curso como en el segundo curso:

- **Modalidad:** Régimen General (comprendido entre un **25% y un 35%** de la carga horaria total del ciclo, repartidas entre ambos cursos).

- **Carga horaria:** Se impartirán aproximadamente **500 horas de formación en empresa**, lo que representa el **25% de la carga horaria total** del título.

5.3. Concreción en el Módulo de Programación (1.º DAW)

De acuerdo con la planificación desarrollada por el Departamento de Informática para el curso de 1.º de DAW, se ha fijado el periodo de estancia y la selección del Resultado de Aprendizaje a dualizar:

- **Periodo de formación:** Desarrollado en la programación general del departamento.

- **Resultado de Aprendizaje (RA) dualizado:**

RA 8: *Utiliza bases de datos orientadas a objetos, analizando sus características y aplicando técnicas para mantener la persistencia de la información.*

6. Principios Pedagógicos y Metodológicos

6.1. Orientaciones Metodológicas

La metodología del módulo tendrá un **carácter competencial, aplicado y profesionalizador**, orientado directamente a la adquisición de los Resultados de Aprendizaje y de las competencias asociadas al perfil profesional:

- **Conexión con el entorno productivo:** Se favorecerá el aprendizaje mediante la aplicación práctica, la resolución de problemas profesionales, la autonomía, la responsabilidad, el trabajo en equipo y la adaptación a situaciones reales o simuladas.

- **Estrategias didácticas:** Según la naturaleza del módulo, podrán emplearse unidades de trabajo, proyectos, retos, simulaciones, casos prácticos, actividades técnicas, prácticas de taller o laboratorio, actividades vinculadas a la empresa o proyectos intermodulares.

- **Dinámica de aprendizaje:** Se combinarán la explicación y demostración con la práctica guiada y autónoma, el manejo de herramientas/equipos y la realización de productos, procesos o desempeños reales.

- **Ejes transversales prioritarios:** Se prestará especial atención a la seguridad, la prevención de riesgos laborales, la innovación, la actualización tecnológica y la empleabilidad.

6.2. Actividades de enseñanza-aprendizaje

Se entiende por metodología el conjunto de decisiones organizativas y pedagógicas adaptadas para llevar a cabo el proceso de enseñanza-aprendizaje (*cómo enseñar*). El método de trabajo en el aula se fundamenta en los siguientes aspectos estratégicos:

1. **Presentación del módulo:** Exposición inicial por parte del docente, detallando sus características, contenidos, metodología de trabajo y los criterios de evaluación que se van a aplicar.

2. **Evaluación inicial y esquema conceptual:** Al comienzo de cada unidad se presentará un esquema de los contenidos a tratar y se propiciará un debate o introducción guiada para detectar los conocimientos previos del alumnado.

3. **Explicación sintética y ejemplificación:** Exposiciones claras y escuetas a cargo del docente, respaldadas por abundantes ejemplos prácticos y supuestos que sinteticen la materia.

4. **Trabajo práctico individual y grupal:** A partir de las explicaciones y demostraciones del docente, el alumnado resolverá problemas y supuestos planteados en el aula.

5. **Fomento de la participación activa:** Impulso de la cooperación e implicación intensa del alumnado durante el desarrollo formativo de la clase.

6. **Aclaración de dudas y ejercitación continua:** Existencia de un espacio dedicado expresamente a la resolución de dudas, destinando el resto del tiempo a la realización de ejercicios (individuales o colectivos) que permitan consolidar lo aprendido.

7. **Recursos de apoyo en las exposiciones:** Uso complementario de recursos didácticos digitales como vídeos, documentación web y aplicaciones informáticas.

8. **Comprobación de la asimilación práctica:** Verificación de la asimilación teórica y práctica mediante la resolución de cuestionarios tipo test y supuestos prácticos en ordenador.

9. **Observación sistemática:** Valoración continua de la atención, actitud y participación activa del alumnado como elemento integrante de la evaluación.

6.3. Recursos y Materiales Didácticos

Para el desarrollo de la práctica docente se emplearán los siguientes medios tecnológicos y entornos:

- **Recursos audiovisuales y multimedia:** Pizarra interactiva digital, equipos informáticos e Internet de alta velocidad.

- **Plataforma Moodle (Aula Virtual):** Entorno virtual para la consulta de materiales didácticos (apuntes, ejercicios, boletines prácticos), canal de comunicación oficial y medio de entrega y seguimiento de actividades de evaluación.

Java con JDK compatible con el entorno del centro; Visual Studio Code.

7. Evaluación

Principios Generales y Características de la Evaluación

La evaluación del módulo se caracterizará por ser **objetiva, continua, formativa e integradora** de las competencias adquiridas tanto en el centro educativo como en el entorno empresarial.

- **Referentes de evaluación:** Los referentes exclusivos para determinar la superación del módulo serán los **Resultados de Aprendizaje (RA)** y sus correspondientes **Criterios de Evaluación (CE)**.
- **Criterios de aplicación:** Las técnicas e instrumentos de evaluación serán variados, flexibles, accesibles y adaptados a la naturaleza de los aprendizajes que se deban comprobar en cada momento.

7.1. Técnicas, Instrumentos y Recogida de Evidencias

Para la obtención de evidencias sobre el desempeño y adquisición de competencias por parte del alumnado, se podrán emplear las siguientes herramientas y actividades:

- **Pruebas de conocimiento e implementación:** Pruebas teóricas o prácticas, resolución de supuestos profesionales e incidencias técnicas.
- **Proyectos y entregables:** Proyectos de desarrollo de software, productos, procesos, documentación técnica y portfolios de evidencias.
- **Prácticas y simulaciones:** Ejercicios prácticos de aula, simulaciones de entorno laboral real y presentaciones orales/técnicas.
- **Observación y seguimiento:** Observación sistemática del desempeño en el aula-taller, entrevistas profesionales y otros medios técnicamente adecuados.

7.2 Ponderación RA/CE, calificación y unidades didácticas.

La calificación se obtendrá a partir del grado de adquisición de los RA, determinado mediante la valoración de sus CE y las ponderaciones establecidas. Cuando existan varias evidencias de un mismo CE se integrarán atendiendo a suficiencia, calidad, continuidad, evolución e información más actual. No se superpondrán porcentajes genéricos por tipos de instrumentos que alteren la ponderación RA/CE.

Resultado de Aprendizaje (RA)	Criterios de Evaluación Asociados (CE)	Ponderación (%)
RA1. Reconoce la estructura de un programa informático, identificando y relacionando los elementos propios del lenguaje de programación utilizado.	**CE1.a)** Identificación de los bloques de la estructura de un programa. **CE1.b)** Creación de proyectos de desarrollo. **CE1.c)** Manejo de IDEs. **CE1.d)** Identificación de tipos de variables. **CE1.e)** Creación y uso de variables. **CE1.f)** Uso de constantes y literales.	**9%**

	CE1.g) Uso de operadores en expresiones. **CE1.h)** Conversiones de tipo explícitas e implícitas. **CE1.i)** Inserción de comentarios en el código.	
RA2. Escribe y prueba programas sencillos, reconociendo y aplicando los fundamentos de la programación orientada a objetos.	**CE2.a)** Identificación de fundamentos de POO. **CE2.b)** Escritura de programas simples. **CE2.c)** Instanciación de objetos. **CE2.d)** Uso de métodos y propiedades. **CE2.e)** Llamadas a métodos estáticos.	**11%**

	CE2.f) Uso de parámetros. **CE2.g)** Uso de librerías de objetos. **CE2.h)** Uso de constructores. **CE2.i)** Compilación e IDE en programas simples.	

RA3. Escribe y depura código, analizando y utilizando las estructuras de control del lenguaje.	**CE3.a)** Estructuras de selección. **CE3.b)** Estructuras de repetición. **CE3.c)** Sentencias de salto. **CE3.d)** Control de excepciones. **CE3.e)** Programas con estructuras de control. **CE3.f)** Depuración y pruebas de código. **CE3.g)** Documentación y comentarios.	**11%**

	CE3.h) Creación de excepciones. **CE3.i)** Aserciones para detección de errores.	
RA4. Desarrolla programas organizados en clases analizando y aplicando los principios de la programación orientada a objetos.	**CE4.a)** Sintaxis y componentes de una clase. **CE4.b)** Definición de clases. **CE4.c)** Definición de propiedades y métodos. **CE4.d)** Creación de constructores. **CE4.e)** Instanciación y uso de clases propias. **CE4.f)** Control de visibilidad y encapsulamiento.	**12%**

	CE4.g) Definición de clases heredadas. **CE4.h)** Creación de métodos estáticos. **CE4.i)** Creación y uso de librerías de clases.	
RA5. Realiza operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases.	**CE5.a)** Entrada y salida por consola. **CE5.b)** Formato en visualización de información. **CE5.c)** Librerías de E/S del lenguaje. **CE5.d)** Almacenamiento y lectura en ficheros.	**15%**

	CE5.e) Métodos de acceso a ficheros. **CE5.f)** Creación de GUI simples. **CE5.g)** Programación de controladores de eventos. **CE5.h)** E/S en interfaces gráficas.	

RA6. Escribe programas que manipulen información seleccionando y utilizando tipos avanzados de datos.	**CE6.a)** Programas con arrays. **CE6.b)** Librerías de tipos de datos avanzados. **CE6.c)** Uso de listas para procesar datos. **CE6.d)** Recorrido con iteradores. **CE6.e)** Selección de colecciones de datos. **CE6.f)** Clases y métodos genéricos. **CE6.g)** Expresiones regulares (Regex).	**15%**

	CE6.h) Documentos XML/JSON e intercambio. **CE6.i)** Manipulación de lenguajes de datos. **CE6.j)** Operaciones agregadas en colecciones.	
RA7. Desarrolla programas aplicando características avanzadas de los lenguajes orientados a objetos y del entorno de programación.	**CE7.a)** Conceptos de herencia y jerarquía. **CE7.b)** Bloqueo y forzado de herencia. **CE7.c)** Incidencia de constructores en herencia. **CE7.d)** Sobrescritura de métodos. **CE7.e)** Diseño de jerarquías de clases.	**14%**

	CE7.f) Depuración de jerarquías de clases. **CE7.g)** Aplicación de jerarquías completas. **CE7.h)** Documentación técnica del código. **CE7.i)** Escenarios de uso de interfaces. **CE7.j)** Elección entre herencia y composición.	

RA8. Utiliza bases de datos orientadas a objetos, analizando sus características y aplicando técnicas para mantener la persistencia de la información.	**CE8.a)** Características de BDOO. **CE8.b)** Aplicación de BDOO en lenguajes OO. **CE8.c)** Instalación de SGBD orientados a objetos. **CE8.d)** Métodos de gestión en SGBDOO. **CE8.e)** Estructuras para almacenamiento de objetos. **CE8.f)** Persistencia de objetos en BDOO. **CE8.g)** Operaciones CRUD sobre objetos.	**4%**

	CE8.h) Tipos de datos estructurados/complejos.	
RA9. Gestiona información almacenada en bases de datos relacionales manteniendo la integridad y consistencia de los datos.	**CE9.a)** Métodos de acceso a SGBD relacionales. **CE9.b)** Conexiones a BD mediante JDBC. **CE9.c)** Escritura de código de almacenamiento. **CE9.d)** Consulta y recuperación de datos. **CE9.e)** Borrados y modificaciones (UPDATE/DELETE). **CE9.f)** Mapeo e interfaces para mostrar BD.	**9%**

	CE9.g) Gestión completa de BD desde la aplicación.	
Total		**100%**

7.3 Matriz de ponderación de Criterios de Evaluación y Unidades Didácticas.

RA	Criterio de Evaluación (CE)	Ponderación CE en Módulo (%)	Unidad Didáctica (UD) Asociada
RA1 *(9%)*	**CE1.a** Estructura de bloques de un programa	1,00%	**UD 1** (Conceptos básicos)
	CE1.b Creación de proyectos de aplicaciones	1,00%	**UD 1** (Conceptos básicos)
	CE1.c Uso de entornos integrados (IDE)	1,00%	**UD 1** (Conceptos básicos)
	CE1.d Tipos de variables y utilidad	1,00%	**UD 1** (Conceptos básicos)
	CE1.e Creación y modificación de variables	1,00%	**UD 1** (Conceptos básicos)

	CE1.f Uso de constantes y literales	1,00%	**UD 1** (Conceptos básicos)
	CE1.g Clasificación y uso de operadores	1,00%	**UD 1** (Conceptos básicos)
	CE1.h Conversiones de tipos de datos	1,00%	**UD 1** (Conceptos básicos)
	CE1.i Inserción de comentarios en código	1,00%	**UD 1** (Conceptos básicos)
RA2 *(11%)*	**CE2.a** Fundamentos de POO	1,22%	**UD 5** (POO en Java)
	CE2.b Escritura de programas simples	1,22%	**UD 2** (Prog. Estructurada)
	CE2.c Instanciación de objetos predefinidos	1,22%	**UD 3** (Arrays)
	CE2.d Métodos y propiedades de objetos	1,22%	**UD 3** (Arrays)
	CE2.e Llamadas a métodos estáticos	1,22%	**UD 3** (Arrays)
	CE2.f Uso de parámetros en llamadas	1,22%	**UD 3** (Arrays)

	CE2.g Incorporación de librerías de objetos	1,22%	**UD 3** (Arrays)
	CE2.h Uso de constructores	1,22%	**UD 4** (Fechas y Cadenas)
	CE2.i Compilación y ejecución en IDE	1,22%	**UD 2** (Prog. Estructurada)
RA3 *(11%)*	**CE3.a** Estructuras de selección (if, switch)	1,22%	**UD 2** (Prog. Estructurada)
	CE3.b Estructuras de repetición (bucles)	1,22%	**UD 2** (Prog. Estructurada)
	CE3.c Sentencias de salto y control	1,22%	**UD 2** (Prog. Estructurada)
	CE3.d Uso de control de excepciones	1,22%	**UD 4** (Fechas y Cadenas)
	CE3.e Programas con estructuras de control	1,22%	**UD 2** (Prog. Estructurada)
	CE3.f Depuración y pruebas de programas	1,22%	**UD 2** (Prog. Estructurada)
	CE3.g Documentación y comentarios de código	1,22%	**UD 2** (Prog. Estructurada)

	CE3.h Creación de excepciones personalizadas	1,22%	**UD 4** (Fechas y Cadenas)
	CE3.i Uso de aserciones en desarrollo	1,22%	**UD 4** (Fechas y Cadenas)
RA4 *(12%)*	**CE4.a** Sintaxis y estructura de clases	1,33%	**UD 5** (POO en Java)
	CE4.b Definición de clases propias	1,33%	**UD 5** (POO en Java)
	CE4.c Definición de atributos y métodos	1,33%	**UD 5** (POO en Java)
	CE4.d Creación de constructores en clases	1,33%	**UD 5** (POO en Java)
	CE4.e Instanciación de objetos propios	1,33%	**UD 5** (POO en Java)
	CE4.f Control de visibilidad (encapsulación)	1,33%	**UD 5** (POO en Java)
	CE4.g Definición de clases heredadas	1,33%	**UD 5** (POO en Java)
	CE4.h Uso de métodos y miembros estáticos	1,33%	**UD 5** (POO en Java)

	CE4.i Creación y uso de conjuntos de clases	1,33%	**UD 8** (Colecciones)
RA5 *(15%)*	**CE5.a** Operaciones de E/S por consola	1,875%	**UD 2** (Prog. Estructurada)
	CE5.b Formatos de visualización de datos	1,875%	**UD 4** (Fechas y Cadenas)
	CE5.c Librerías de E/S del lenguaje	1,875%	**UD 7** (Ficheros)
	CE5.d Almacenamiento y lectura en ficheros	1,875%	**UD 7** (Ficheros)
	CE5.e Métodos de acceso a ficheros	1,875%	**UD 7** (Ficheros)
	CE5.f Creación de interfaces gráficas (GUI)	1,875%	**UD 10** (Interfaces gráficas)
	CE5.g Programación de eventos en GUI	1,875%	**UD 10** (Interfaces gráficas)
	CE5.h E/S mediante interfaces gráficas	1,875%	**UD 10** (Interfaces gráficas)
RA6 *(15%)*	**CE6.a** Desarrollo de programas con arrays	1,50%	**UD 3** (Arrays)

	CE6.b Librerías de datos avanzados	1,50%	**UD 3** (Arrays)
	CE6.c Listas para almacenamiento de datos	1,50%	**UD 8** (Colecciones)
	CE6.d Recorrido de datos con iteradores	1,50%	**UD 8** (Colecciones)
	CE6.e Selección de colecciones de datos	1,50%	**UD 8** (Colecciones)
	CE6.f Clases y métodos genéricos	1,50%	**UD 6** (Interfaces)
	CE6.g Uso de expresiones regulares (Regex)	1,50%	**UD 4** (Fechas y Cadenas)
	CE6.h Identificación de librerías XML/JAXB	1,50%	**UD 6** (Interfaces) / **UD 7**
	CE6.i Manipulación de ficheros XML/JAXB	1,50%	**UD 6** (Interfaces) / **UD 7**
	CE6.j Operaciones agregadas sobre colecciones	1,50%	**UD 8** (Colecciones)
RA7 *(14%)*	**CE7.a** Herencia, superclases y subclases	1,40%	**UD 5** (POO en Java)

	CE7.b Modificadores de herencia (final)	1,40%	**UD 5** (POO en Java)
	CE7.c Constructores en jerarquías de herencia	1,40%	**UD 5** (POO en Java)
	CE7.d Sobrescritura de métodos	1,40%	**UD 5** (POO en Java)
	CE7.e Diseño de jerarquías de clases	1,40%	**UD 5** (POO en Java)
	CE7.f Depuración de jerarquías de clases	1,40%	**UD 5** (POO en Java)
	CE7.g Programas con jerarquías complejas	1,40%	**UD 5** (POO en Java)
	CE7.h Documentación de código orientado a objetos	1,40%	**UD 2** (Prog. Estructurada)
	CE7.i Definición e implementación de interfaces	1,40%	**UD 6** (Interfaces)
	CE7.j Evaluación de herencia vs. composición	1,40%	**UD 6** (Interfaces)
RA8 *(4%)*	**CE8.a** Características de BDOO	0,50%	**UD 9** (JDBC y BDOO / Empresa)

	CE8.b Aplicación de BDOO en lenguajes OO	0,50%	**UD 9** (JDBC y BDOO / Empresa)
	CE8.c Instalación de SGBDOO	0,50%	**UD 9** (JDBC y BDOO / Empresa)
	CE8.d Métodos de gestión en SGBDOO	0,50%	**UD 9** (JDBC y BDOO / Empresa)
	CE8.e Estructuras para almacenamiento	0,50%	**UD 9** (JDBC y BDOO / Empresa)
	CE8.f Almacenamiento de objetos en BDOO	0,50%	**UD 9** (JDBC y BDOO / Empresa)
	CE8.g Operaciones CRUD sobre BDOO	0,50%	**UD 9** (JDBC y BDOO / Empresa)
	CE8.h Gestión de datos estructurados	0,50%	**UD 9** (JDBC y BDOO / Empresa)
RA9 *(9%)*	**CE9.a** Características de acceso a SGBD	1,285%	**UD 9** (Conexión a BD: JDBC)
	CE9.b Conexión con SGBD relacionales	1,285%	**UD 9** (Conexión a BD: JDBC)
	CE9.c Inserción y persistencia de datos	1,285%	**UD 9** (Conexión a BD: JDBC)

	CE9.d Consultas SQL y recuperación de datos	1,285%	**UD 9** (Conexión a BD: JDBC)
	CE9.e Modificaciones y borrados (CRUD)	1,285%	**UD 9** (Conexión a BD: JDBC)
	CE9.f Interfaces para mostrar información	1,285%	**UD 9** (Conexión a BD: JDBC)
	CE9.g Gestión completa de la base de datos	1,285%	**UD 9** (Conexión a BD: JDBC)
Total	**100%**	**100%**	

7.4 Sistema de recuperación, subida de notas y evaluación Final 1ª y 2ª

7.4.1 Criterio de superación del módulo y RA imprescindibles.

Se requiere la superación de todos los resultados de aprendizaje. Cada RA debe alcanzar al menos 5 sobre 10: sus funciones son complementarias y no se considera que la ausencia de una pueda compensarse con otra. No se establece una selección reducida de RA imprescindibles. Superados todos los RA, la nota final se obtiene con las ponderaciones publicadas y se expresa de 1 a 10 sin decimales. Se redondea al entero más próximo, con 0,5 al alza, sin utilizar el redondeo para acreditar un RA que no alcance 5. Si queda un RA pendiente, la calificación será inferior a 5. No se aplicarán descuentos por ausencias, comportamiento o sospecha de uso de IA.

7.4.2 Recuperación y mejora.

La recuperación se integrará en el proceso de enseñanza y evaluación y se dirigirá específicamente a los RA/CE no suficientemente acreditados. Podrá realizarse mediante actividades prácticas, tareas, proyectos, supuestos profesionales, pruebas, demostraciones, entrevistas u otras evidencias adecuadas. No será necesario repetir el mismo instrumento ni volver a acreditar aprendizajes ya suficientemente demostrados cuando no resulte

necesario. Las actuaciones de mejora de calificación estarán vinculadas a RA/CE concretos y requerirán nuevas evidencias válidas. No consistirán en la asignación de puntos adicionales desvinculados de los referentes curriculares. Procedimientos de recuperación: observación individual adicional, pequeña prueba manuscrita, tarea práctica, examen escrito, oral o práctico y revisión de productos, según los CE pendientes. Se comunicará un plan con criterios, actividades, instrumentos y fechas. Habrá oportunidades durante cada evaluación y en los periodos finales del centro. La mejora se acordará para CE concretos mediante nuevas evidencias, sin puntos extra ajenos al currículo ni límite de nota por recuperar.

7.4.3 Evaluaciones parciales y finales

Las calificaciones parciales reflejarán el grado de adquisición acreditado hasta ese momento respecto de los RA/CE efectivamente desarrollados. En las evaluaciones finales se determinará el grado de adquisición alcanzado a partir de las evidencias disponibles y de las obtenidas mediante los procesos de recuperación o mejora. La calificación final se determinará aplicando los criterios y ponderaciones establecidos en esta programación.

7.4.4 Pérdida del derecho a evaluación continua

En modalidad presencial que es esta enseñanza, la evaluación continua requiere una asistencia regular y obligatoria, tanto en el centro como en la formación en empresa, de al menos el 80 % de la duración total del módulo, ámbito o proyecto desde la fecha efectiva de matrícula. Conforme al Proyecto Educativo del IES Jándula, se producirá la pérdida del derecho a evaluación continua cuando las ausencias superen el 20 % de la duración correspondiente y la asistencia efectiva resulte inferior al 80 %. El cómputo se realizará de forma independiente para cada módulo, ámbito o proyecto e incluirá las horas de centro y, cuando corresponda, las vinculadas a la formación en empresa. La pérdida no supone la pérdida del derecho a ser evaluado.

7.4.5 Formación en empresa y evaluación

La tutoría dual de empresa valorará los RA previstos durante la estancia y trasladará su valoración al centro. El profesorado responsable de cada módulo ajustará su evaluación y posterior calificación en consonancia con dicha información. La calificación del módulo integrará la valoración del centro y de la empresa conforme a la programación y será responsabilidad final del equipo docente y del centro.

7.4.6 Insuficiencia de evidencias, conservación y garantías

Cuando las evidencias resulten insuficientes podrán recabarse nuevas evidencias mediante actividades prácticas, pruebas, demostraciones, entrevistas profesionales, supuestos o revisión de procesos. Las evidencias se identificarán y conservarán conforme a los criterios institucionales para garantizar objetividad, trazabilidad y los derechos de aclaración, revisión y reclamación.

8. Atención a la diversidad

8.1. Marco General y Principios de la Respuesta Educativa

La respuesta educativa garantizará el **acceso, participación, aprendizaje, permanencia y adquisición de las competencias profesionales** mediante medidas metodológicas, tecnológicas, organizativas, de comunicación, de acceso a espacios/equipamientos, materiales, tiempos y procedimientos de evaluación.

- **Preservación de estándares curriculares:** Las medidas aplicadas facilitarán el acceso y la evaluación manteniendo integrales los Resultados de Aprendizaje (RA), Criterios de Evaluación (CE) y competencias profesionales establecidos, salvo previsión normativa específica.

- **Registro documental:** Las medidas individualizadas se documentarán oportunamente en las fuentes e instrumentos institucionales correspondientes.

- **Marco normativo en Formación Profesional:** En la enseñanza de FP **no se contemplan las adaptaciones curriculares significativas**. Por ello, la atención individualizada se canaliza exclusivamente a través de **medidas no significativas de acceso al currículo**.

8.2. Medidas de Adaptación Curricular No Significativa

- **Medidas de Acceso:**
  Modificación o provisión de recursos espaciales, materiales y de comunicación que faciliten el desarrollo del currículo ordinario al alumnado que lo requiera.

  Gestión flexible y cooperativa de la organización del aula.

- **Medidas de Refuerzo y Ampliación:**

  Suministro de materiales complementarios de refuerzo o ampliación para ajustar el proceso de enseñanza-aprendizaje a los distintos ritmos de trabajo.

  Programación de actividades específicas de refuerzo para un seguimiento más personalizado.
  -
- **Atención a la Incorporación y Admisión Tardía:**
  Ante la fluctuación de matriculaciones y altas/bajas habituales durante las primeras semanas de curso, se aplican los siguientes protocolos:

  Acceso a contenidos: Garantía de acceso inmediato a todo el material e información impartida con anterioridad a la incorporación.

Tutorización y apoyo: Acompañamiento docente para resolver dudas y superar dificultades iniciales sobre las unidades ya trabajadas.

Flexibilidad en la evaluación: Siempre que la incorporación tardía responda a causas justificadas, el alumnado podrá realizar las pruebas de evaluación y prácticas que hayan tenido lugar antes de su llegada. En caso no justificado, se aplicarán los criterios comunes de evaluación.

Información al grupo: Recordatorio general de los criterios de evaluación y calificación en el aula una vez que la composición del grupo se haya estabilizado.

- **Atención al Alumnado con Altas Capacidades:**

  Actividades de ampliación: Propuesta de tareas complejas y complementarias que profundicen en los contenidos abordados o en áreas tecnológicas afines.

  Aprendizaje colaborativo (Tutoría entre iguales): Participación como monitores de apoyo a sus compañeros en aquellas actividades donde muestren mayor destreza, promoviendo la cohesión del grupo.

8.3. Diagnóstico de la Evaluación Inicial

Tras la realización de la **evaluación inicial**, se ha detectado alumnado con necesidades específicas de atención a la diversidad en el grupo actual, repetidor con este módulo en concreto. Vamos a tener en cuenta su NEAE dando más tiempo para realizar las pruebas escritas.

9.- Bibliografía

- Dorrego Martín, M. (s.f) Programación Síntesis.
- Jiménez Marín, A & Pérez Montes, F (2021). Programación. Ediciones Paraninfo.