

 	MODELO DE PROGRAMACIÓN ANUAL	MD8101	
		VERSIÓN 0	Pág. 1 de 44

Programación anual del módulo profesional

Desarrollo web en entorno servidor

Ricardo Pérez López

Curso 2024/2025

IES «Doñana», Sanlúcar de Bda.
Departamento de Informática y Comunicaciones

Jefe de Departamento: Ricardo Pérez López

Índice

1. Información general	4
2. Objetivos generales	4
3. Resultados de aprendizaje y criterios de evaluación	6
4. Contenidos y temporalización	10
4.1. Cuadro resumen	10
4.2. Esquema detalle	11
4.3. Correspondencia con resultados de aprendizaje y criterios de evaluación	26
5. Evaluación y calificación	28
5.1. Valoración general de los instrumentos de evaluación y calificación	30
5.2. Calificación	30
5.2.1. De un criterio de evaluación	30
5.2.2. De un resultado de aprendizaje	32
5.2.3. Calificaciones parciales	32
5.2.4. Calificación final	33
5.3. Ponderación de los criterios de evaluación	34
5.4. Medidas de recuperación y mejora de la calificación final	36
5.4.1. Prueba final	37
5.4.2. Mejora de la calificación final	37
6. Evaluación de la práctica docente	37
7. Orientaciones pedagógicas	38
8. Orientaciones metodológicas	39
9. Recursos	39
9.1. Hardware	39
9.2. Software	39
9.3. Online	40
9.4. Bibliografía	40
9.4.1. Principal	40
9.4.2. Complementaria	40
10. Atención a la diversidad	40

11. Temas transversales	41
12. Actuaciones para desarrollar la perspectiva de género	42
12.1. Actuaciones generales permanentes	43

 	MODELO DE PROGRAMACIÓN ANUAL	MD8101	
		VERSIÓN 0	Pág. 4 de 44

1. Información general

- **Normativa de aplicación:** [1], [2], [3], [4], [5], [6], [7].
- **Equivalencia en créditos ECTS:** 13.
- **Código:** 0613.
- **Duración total:** 168 (21 semanas).
- **Carga lectiva semanal:** 8 horas

2. Objetivos generales

1. La formación del módulo contribuye a alcanzar los **objetivos generales** de este ciclo formativo que se relacionan a continuación [3]:
 - c) Instalar módulos analizando su estructura y funcionalidad para gestionar servidores en distintos ámbitos de aplicación.
 - d) Ajustar parámetros analizando la configuración para gestionar servidores de aplicaciones en distintos ámbitos de aplicación.
 - 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.
 - l) Utilizar herramientas y lenguajes específicos, cumpliendo las especificaciones, para desarrollar e integrar componentes software en el entorno del servidor web.
 - m) Emplear herramientas específicas, integrando la funcionalidad entre aplicaciones, para desarrollar servicios empleables en aplicaciones web.
 - n) Evaluar servicios distribuidos ya desarrollados, verificando sus prestaciones y funcionalidad, para integrar servicios distribuidos en una aplicación web.
 - ñ) Verificar los componentes de software desarrollados, analizando las especificaciones, para completar el plan de pruebas.
 - q) Programar y realizar actividades para gestionar el mantenimiento de los recursos informáticos.

 	MODELO DE PROGRAMACIÓN ANUAL	MD8101	
		VERSIÓN 0	Pág. 5 de 44

- s) Desarrollar la creatividad y el espíritu de innovación para responder a los retos que se presentan en los procesos y organización de trabajo y de la vida personal.
- t) Tomar decisiones de forma fundamentada analizando las variables implicadas, integrando saberes de distinto ámbito y aceptando los riesgos y la posibilidad de equivocación en las mismas, para afrontar y resolver distintas situaciones, problemas o contingencias.
2. La formación del módulo contribuye a alcanzar las **competencias profesionales, personales y sociales** de este título que se relacionan a continuación [3]:
- c) Gestionar servidores en distintos ámbitos de implantación 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.
- 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.
- k) Desarrollar servicios para integrar sus funciones en otras aplicaciones web, asegurando su funcionalidad.
- l) Integrar servicios y contenidos distribuidos en aplicaciones web, asegurando su funcionalidad.
- m) Completar planes de pruebas verificando el funcionamiento de los componentes software desarrollados, según las especificaciones.
- n) Elaborar y mantener la documentación de los procesos de desarrollo, utilizando herramientas de generación de documentación y control de versiones.
- ñ) Desplegar y distribuir aplicaciones web en distintos ámbitos de implantación, verificando su comportamiento y realizando modificaciones.
- q) Resolver situaciones, problemas o contingencias con iniciativa y autonomía en el ámbito de su competencia, con creatividad, innovación y espíritu de mejora en el trabajo personal y en el de los miembros del equipo.

 	MODELO DE PROGRAMACIÓN ANUAL	MD8101	
		VERSIÓN 0	Pág. 6 de 44

3. Resultados de aprendizaje y criterios de evaluación

Los resultados de aprendizaje del módulo y sus criterios de evaluación asociados son los que se describen a continuación:

[RA₁] Selecciona las arquitecturas y tecnologías de programación web en entorno servidor, analizando sus capacidades y características propias.

Criterios de evaluación:

- CE_{1.a}) Se han caracterizado y diferenciado los modelos de ejecución de código en el servidor y en el cliente web.
- CE_{1.b}) Se han reconocido las ventajas que proporciona la generación dinámica de páginas web.
- CE_{1.c}) Se han identificado los mecanismos de ejecución de código en los servidores web.
- CE_{1.d}) Se han reconocido las funcionalidades que aportan los servidores de aplicaciones y su integración con los servidores web.
- CE_{1.e}) Se han identificado y caracterizado los principales lenguajes y tecnologías relacionados con la programación web en entorno servidor.
- CE_{1.f}) Se han verificado los mecanismos de integración de los lenguajes de marcas con los lenguajes de programación en entorno servidor.
- CE_{1.g}) Se han reconocido y evaluado las herramientas y frameworks de programación en entorno servidor.

[RA₂] Escribe sentencias ejecutables por un servidor web reconociendo y aplicando procedimientos de integración del código en lenguajes de marcas.

Criterios de evaluación:

- CE_{2.a}) Se han reconocido los mecanismos de generación de páginas web a partir de lenguajes de marcas con código embebido.
- CE_{2.b}) Se han identificado las principales tecnologías asociadas.
- CE_{2.c}) Se han utilizado etiquetas para la inclusión de código en el lenguaje de marcas.
- CE_{2.d}) Se ha reconocido la sintaxis del lenguaje de programación que se ha de utilizar.
- CE_{2.e}) Se han escrito sentencias simples y se han comprobado sus efectos en el documento resultante.
- CE_{2.f}) Se han utilizado directivas para modificar el comportamiento predeterminado.
- CE_{2.g}) Se han utilizado los distintos tipos de variables y operadores disponibles en el lenguaje.

 	MODELO DE PROGRAMACIÓN ANUAL	MD8101	
		VERSIÓN 0	Pág. 7 de 44

CE_{2.h}) Se han identificado los ámbitos de utilización de las variables.

[RA₃] Escribe bloques de sentencias embebidos en lenguajes de marcas, seleccionando y utilizando las estructuras de programación.

Criterios de evaluación:

CE_{3.a}) Se han utilizado mecanismos de decisión en la creación de bloques de sentencias.

CE_{3.b}) Se han utilizado bucles y se ha verificado su funcionamiento.

CE_{3.c}) Se han utilizado matrices (arrays) para almacenar y recuperar conjuntos de datos.

CE_{3.d}) Se han creado y utilizado funciones.

CE_{3.e}) Se han utilizado formularios web para interactuar con el usuario del navegador web.

CE_{3.f}) Se han empleado métodos para recuperar la información introducida en el formulario.

CE_{3.g}) Se han añadido comentarios al código.

[RA₄] Desarrolla aplicaciones web embebidas en lenguajes de marcas analizando e incorporando funcionalidades según especificaciones.

Criterios de evaluación:

CE_{4.a}) Se han identificado los mecanismos disponibles para el mantenimiento de la información que concierne a un cliente web concreto y se han señalado sus ventajas.

CE_{4.b}) Se han utilizado mecanismos para mantener el estado de las aplicaciones web.

CE_{4.c}) Se han utilizado mecanismos para almacenar información en el cliente web y para recuperar su contenido.

CE_{4.d}) Se han identificado y caracterizado los mecanismos disponibles para la autenticación de usuarios.

CE_{4.e}) Se han escrito aplicaciones que integren mecanismos de autenticación de usuarios.

CE_{4.f}) Se han utilizado herramientas y entornos para facilitar la programación, prueba y depuración del código.

[RA₅] Desarrolla aplicaciones web identificando y aplicando mecanismos para separar el código de presentación de la lógica de negocio.

Criterios de evaluación:

CE_{5.a}) Se han identificado las ventajas de separar la lógica de negocio de los aspectos de presentación de la aplicación.

 	MODELO DE PROGRAMACIÓN ANUAL	MD8101	
		VERSIÓN 0	Pág. 8 de 44

- CE_{5.b}) Se han analizado y utilizado mecanismos y frameworks que permiten realizar esta separación y sus características principales.
- CE_{5.c}) Se han utilizado objetos y controles en el servidor para generar el aspecto visual de la aplicación web en el cliente.
- CE_{5.d}) Se han utilizado formularios generados de forma dinámica para responder a los eventos de la aplicación web.
- CE_{5.e}) Se han identificado y aplicado los parámetros relativos a la configuración de la aplicación web.
- CE_{5.f}) Se han escrito aplicaciones web con mantenimiento de estado y separación de la lógica de negocio.
- CE_{5.g}) Se han aplicado los principios y patrones de diseño de la programación orientada a objetos.
- CE_{5.h}) Se ha probado y documentado el código.

[RA₆] Desarrolla aplicaciones de acceso a almacenes de datos, aplicando medidas para mantener la seguridad y la integridad de la información.

Criterios de evaluación:

- CE_{6.a}) Se han analizado las tecnologías que permiten el acceso mediante programación a la información disponible en almacenes de datos.
- CE_{6.b}) Se han creado aplicaciones que establezcan conexiones con bases de datos.
- CE_{6.c}) Se ha recuperado información almacenada en bases de datos.
- CE_{6.d}) Se ha publicado en aplicaciones web la información recuperada.
- CE_{6.e}) Se han utilizado conjuntos de datos para almacenar la información.
- CE_{6.f}) Se han creado aplicaciones web que permitan la actualización y la eliminación de información disponible en una base de datos.
- CE_{6.g}) Se han probado y documentado las aplicaciones web.

[RA₇] Desarrolla servicios web reutilizables y accesibles mediante protocolos web, verificando su funcionamiento.

Criterios de evaluación:

- CE_{7.a}) Se han reconocido las características propias y el ámbito de aplicación de los servicios web.
- CE_{7.b}) Se han reconocido las ventajas de utilizar servicios web para proporcionar acceso a funcionalidades incorporadas a la lógica de negocio de una aplicación.

 	MODELO DE PROGRAMACIÓN ANUAL	MD8101	
		VERSIÓN 0	Pág. 9 de 44

CE7.c) Se han identificado las tecnologías y los protocolos implicados en el consumo de servicios web.

CE7.d) Se han utilizado los estándares y arquitecturas más difundidos e implicados en el desarrollo de servicios web.

CE7.e) Se ha programado un servicio web.

CE7.f) Se ha verificado el funcionamiento del servicio web.

CE7.g) Se ha consumido el servicio web.

CE7.h) Se ha documentado un servicio web.

[RA₈] Genera páginas web dinámicas analizando y utilizando tecnologías y frameworks del servidor web que añadan código al lenguaje de marcas.

Criterios de evaluación:

CE8.a) Se han identificado las diferencias entre la ejecución de código en el servidor y en el cliente web.

CE8.b) Se han reconocido las ventajas de unir ambas tecnologías en el proceso de desarrollo de programas.

CE8.c) Se han identificado las tecnologías y frameworks relacionadas con la generación por parte del servidor de páginas web con guiones embebidos.

CE8.d) Se han utilizado estas tecnologías y frameworks para generar páginas web que incluyan interacción con el usuario.

CE8.e) Se han utilizado estas tecnologías y frameworks para generar páginas web que incluyan verificación de formularios.

CE8.f) Se han utilizado estas tecnologías y frameworks para generar páginas web que incluyan modificación dinámica de su contenido y su estructura.

CE8.g) Se han aplicado estas tecnologías y frameworks en la programación de aplicaciones web.

[RA₉] Desarrollar aplicaciones web híbridas seleccionando y utilizando tecnologías, frameworks servidor y repositorios heterogéneos de información.

Criterios de evaluación:

CE9.a) Se han reconocido las ventajas que proporciona la reutilización de código y el aprovechamiento de información ya existente.

CE9.b) Se han identificado tecnologías y frameworks aplicables en la creación de aplicaciones web híbridas.

- CE_{9.c}) Se ha creado una aplicación web que recupere y procese repositorios de información ya existentes.
- CE_{9.d}) Se han creado repositorios específicos a partir de información existente en almacenes de información.
- CE_{9.e}) Se han utilizado librerías de código y frameworks para incorporar funcionalidades específicas a una aplicación web.
- CE_{9.f}) Se han programado servicios y aplicaciones web utilizando como base información y código generados por terceros.
- CE_{9.g}) Se han analizado y utilizado librerías de código relacionadas con Big Data e inteligencia de negocios (Business Intelligence), para incorporar análisis e inteligencia de datos proveniente de repositorios.
- CE_{9.h}) Se han probado, depurado y documentado las aplicaciones generadas.

4. Contenidos y temporalización

Cada unidad didáctica tiene una duración temporal de **una semana**, que podrá ampliarse o reducirse en función de las circunstancias cuando el profesor lo estime conveniente (por ejemplo, para la realización de actividades, ejercicios y prácticas en clase).

Los contenidos marcados con la etiqueta *#opcional* son contenidos complementarios que sólo se impartirán si hay tiempo suficiente para ello y nunca a costa de otros contenidos no opcionales. También podrán ser usados como elementos a desarrollar para el alumnado con altas capacidades o que manifieste un ritmo de aprendizaje superior al del resto del grupo/clase.

Todas las fechas se muestran en formato ISO 8601 (*año-mes-día*).

4.1. Cuadro resumen

Unidad didáctica	Inicio estimado
1. Introducción a la tecnología web <i>#ev1</i>	2024-09-16
2. Conceptos básicos de PHP (I) <i>#ev1</i>	2024-09-23
3. Conceptos básicos de PHP (II) <i>#ev1</i>	2024-09-30
4. Desarrollo de aplicaciones con PHP (I) <i>#ev1</i>	2024-10-07
5. Persistencia de datos con PHP <i>#ev1</i>	2024-10-14
6. Desarrollo de aplicaciones con PHP (II) <i>#ev1</i>	2024-10-21
7. Programación avanzada en PHP <i>#ev1</i>	2024-10-28
8. Interoperabilidad <i>#ev1</i>	2024-11-04

Unidad didáctica	Inicio estimado
9. Introducción a Laravel #ev1	2024-11-11
10. Arquitectura de Laravel #ev1	2024-11-18
11. Gestión de peticiones en Laravel #ev1	2024-11-25
12. Visualización de datos en Laravel #ev2	2025-01-08
13. Bases de datos en Laravel #ev2	2025-01-15
14. Mapeado objeto-relacional con Eloquent #ev2	2025-01-22
15. Creación y validación de formularios en Laravel #ev2	2025-01-29
16. Seguridad y cacheado en Laravel #ev2	2025-02-05
17. Características adicionales de Laravel #ev2	2025-02-12
18. Calidad #ev2	2025-02-19
19. Computación en la nube #ev2	2025-02-26
20. Servicios web con Laravel #ev2	2025-03-05
21. Aplicaciones web híbridas #ev2	2025-03-12

4.2. Esquema detalle

El título de cada unidad va etiquetado con los resultados de aprendizaje y los criterios de evaluación relacionados con dicha unidad. Cuando un resultado de aprendizaje no lleva asociado ningún criterio de evaluación, significa que dicho resultado de aprendizaje se trabaja en la unidad didáctica pero no es el elemento fundamental de evaluación.

1 **INTRODUCCIÓN A LA TECNOLOGÍA WEB** #ce1a #ce1b #ce1c #ce1d #ce1e #ce1g #ev1 #ra1 (est: 2024-09-16)

1.1. Introducción al desarrollo web

1.1.1. Conceptos básicos

- Navegadores y servidores web
- Agentes de usuario
- Web estática vs. dinámica
- Estructura vs. contenido
- Arquitectura multinivel

1.1.2. Ejemplos de aplicaciones web

- Redes sociales: Facebook, Twitter...
- Comercio electrónico: Amazon, eBay...
- Administración electrónica...
- Portales
- ERP, CRM

1.1.3. Tecnologías de desarrollo de aplicaciones web

- .NET
- Java
- Ruby/Rails
- Python/Django
- PHP
- El Kung-Fu de la Programación
 - Odo
 - PrestaShop
 - Drupal
 - WordPress

1.2. Arquitectura cliente/servidor

1.3. HTML 5 básico (recordatorio de primer curso)

1.4. Protocolo HTTP

1.4.1. URIs

- URL encoding

1.4.2. Peticiones (*HTTP requests*) y respuestas (*HTTP responses*)

1.4.3. Métodos: GET, POST

1.4.4. Versiones

- HTTP/1.0
- HTTP/1.1

1.4.5. Cabeceras HTTP

1.4.6. Códigos de estado

1.4.7. Experimentos

- telnet (a un servidor)
- netcat (desde un navegador)
- `curl -i -XPOST "http://..." | pygmentize -l http`
- http
- Google Chrome Developer Tools

1.4.8. Envío de datos al servidor

- Mediante GET
- Mediante POST
- Formularios HTML

1.4.9. Cookies

2 **CONCEPTOS BÁSICOS DE PHP (I)** #ce2d #ce2e #ce2f #ce2g #ce4f #ev1 #ra2 #ra3 #ra4 (est: 2024-09-23)

2.1. Introducción a PHP

2.1.1. Página web de PHP

2.1.2. Instalación de PHP

2.1.3. Configuración básica con php.ini

- error_reporting = E_ALL
- display_errors = On
- display_startup_errors = On
- date.timezone = 'UTC'

2.1.4. Módulos de extensión

2.1.5. Documentación y búsqueda de información

2.1.6. Modos de ejecución

- Por lotes
- Interactiva
 - php -a
 - PsySH

2.2. Sintaxis básica

2.2.1. Datos e instrucciones

2.2.2. Sentencias y comandos

- Comando echo

2.2.3. Expresiones, operadores y funciones

2.3. Funcionamiento del intérprete

2.3.1. Etiquetas <?php y ?>

2.3.2. Modo dual de operación

2.3.3. Comentarios

2.4. Variables

2.4.1. Conceptos básicos

2.4.2. Destrucción de variables

2.4.3. Asignación compuesta

2.4.4. Operadores de asignación por valor y por referencia

2.4.5. Variables predefinidas

2.5. Tipos básicos de datos

2.5.1. Lógicos (bool)

- Operadores lógicos

2.5.2. Numéricos

- Enteros (int)

- Números en coma flotante (float)
- Operadores
 - Operadores aritméticos
 - Operadores de incremento/decremento

2.5.3. Cadenas (string)

- Operadores de cadenas
 - Concatenación
 - Acceso y modificación por caracteres
 - Operador de incremento *#opcional*
- Funciones de manejo de cadenas
- Extensión *mbstring*

2.5.4. Nulo (null)

2.6. Manipulación de datos

2.6.1. Precedencia de operadores

2.6.2. Operadores de asignación compuesta

2.6.3. Comprobaciones

- De tipos
 - `gettype()`
 - `is_*()`
- De valores
 - `is_numeric()`
 - `ctype_*()`

2.6.4. Conversiones de tipos

- Conversión explícita (*casting*) vs. implícita (automática)
- Conversión a bool
- Conversión a int
- Conversión a float
- Conversión de string a número
- Conversión a string
- Funciones de obtención de valores
 - `intval()`
 - `floatval()`
 - `strval()`
 - `boolval()`
- Funciones de formateo numérico
 - `number_format()`

- `money_format()`
- `setlocale()`
- `NumberFormatter`

2.6.5. Comparaciones

- Operadores de comparación
- `==` vs. `===`
- Ternario (`?:`)
- Fusión de `null` (`??`)
- Reglas de comparación de tipos

2.7. Constantes

- 2.7.1. `define()` y `const`
- 2.7.2. Constantes predefinidas
- 2.7.3. `defined()`

3 CONCEPTOS BÁSICOS DE PHP (II) *#ce2d #ce2e #ce2g #ce2h #ce3a #ce3b #ce3c #ce3d #ce3g #ce4g #ev1 #ra2 #ra3 #ra4* (est: 2024-09-30)

3.1. Flujo de control

3.1.1. Estructuras de control

- Secuencia
- Selección
- Iteración
- Sintaxis alternativa

3.1.2. Inclusión de scripts

- `include`, `require`
- `include_once`, `require_once`

3.2. Funciones predefinidas destacadas

- 3.2.1. `isset()`
- 3.2.2. `empty()`
- 3.2.3. `var_dump()`

3.3. Arrays

3.3.1. Operadores para arrays

- Acceso, modificación y agregación

3.3.2. Funciones de manejo de arrays

- Ordenación de arrays
- `print_r()`
- `'+'` vs. `array_merge()`

- `isset()` vs. `array_key_exists()`
- 3.3.3. `foreach`
 - 3.3.4. Conversión a array
 - 3.3.5. *Ejemplo*: `$argv` en CLI
- 3.4. Funciones definidas por el usuario
 - 3.4.1. Argumentos
 - Paso de argumentos por valor y por referencia
 - Argumentos por defecto
 - 3.4.2. Ámbito de variables
 - Ámbito simple al archivo
 - Variables locales
 - Uso de `global`
 - Variables superglobales
 - 3.4.3. Declaraciones de tipos
 - Declaraciones de tipo de argumento
 - Declaraciones de tipo de devolución
 - Tipos *nullable* (?) y `void`
 - Tipificación estricta
 - 3.5. Comentarios y documentación del código

4 DESARROLLO DE APLICACIONES CON PHP (I) *#ce1a #ce1b #ce1d #ce1f #ce2a #ce2b #ce2c #ce2d #ce2e #ce2f #ce2g #ce2h #ce3a #ce3b #ce3c #ce3d #ce3e #ce3f #ce3g #ce4g #ce5a #ce5d #ce5g #ce5h #ev1 #ra1 #ra2 #ra3 #ra4 #ra5* (est: 2024-10-07)

- 4.1. SAPIs
 - 4.1.1. CLI: Uso en línea de comandos
 - `$argc` y `$argv`
 - Flujos de entrada/salida *#opcional*
 - 4.1.2. Apache *#opcional*
 - Integración de PHP con Apache
 - PHP como lenguaje embebido
 - 4.1.3. CGI: PHP-FPM (FastCGI Process Manager) *#opcional*
- 4.2. Código embebido
 - 4.2.1. Etiqueta `<?=>`
 - 4.2.2. Servidor web interno
- 4.3. Manejo de datos de entrada: `$_GET` y `$_POST`
- 4.4. Funciones auxiliares de interés
 - 4.4.1. `extract()`

4.4.2. `compact()`

4.5. Aspectos básicos de la orientación a objetos

4.5.1. Objetos

- `new`
- `instanceof`
- `get_class`

4.5.2. Referencias

- Asignación por referencia (`=&`)

4.5.3. Clonación de objetos

4.5.4. Comparación de objetos

4.5.5. Propiedades

- Predeterminadas
- Dinámicas

4.5.6. Métodos

4.5.7. Constantes

- Operador de resolución de ámbito (`::`)

4.5.8. *Ejemplo*: manejo de fechas, horas, instantes e intervalos

4.6. Excepciones

4.6.1. Manejo de errores clásico en PHP

4.6.2. Errores vs. excepciones

4.6.3. La clase `Exception`

4.6.4. La clase `Error`

4.6.5. La clase `ErrorException`

4.6.6. Estructura de control `try ... catch`

4.7. Depuración

4.7.1. `var_dump()`, `print_r()`, `die()`

4.7.2. Depuración con `PsySH`

4.7.3. `Xdebug` *#opcional*

- Módulo `Xdebug`
- Aplicación `Xdebug` para Chrome
- Extensión `Xdebug Helper` para Chrome
- Paquete `php-debug` para Atom

5 PERSISTENCIA DE DATOS CON PHP #ce4a #ce4b #ce4c #ce4g #ce5f #ce5g #ce5h #ce6a #ce6b #ce6c #ce6e #ev1 #ra2 #ra3 #ra4 #ra5 #ra6 (est: 2024-10-14)

5.1. PDO (PHP Data Objects)

5.1.1. Clase PDO

- `__construct()`
- `query()`
- `exec()`
- `prepare()`

5.1.2. Clase PDOStatement

- `fetch()`
- `fetchAll()`
- `fetchColumn()`
- `execute()`
- `rowCount()`

5.1.3. Correspondencias de tipos entre SQL y PHP

5.1.4. Manejo de errores en PDO

- `PDO::errorInfo()`

5.1.5. Transacciones

- `$pdo->beginTransaction();`
- `$pdo->commit();`
- `$pdo->rollBack();`

5.2. Programación de *CRUD* básico

5.2.1. Ejemplo de aplicación: *Muéveme*

5.2.2. Ejemplo de aplicación: *FilmAffinity*

5.3. Post/Redirect/Get

5.4. `header()`

5.4.1. `output_buffering`

5.5. Cookies

5.5.1. `setcookie()`

5.5.2. Ejemplos de uso

6 DESARROLLO DE APLICACIONES CON PHP (II) #ce1a #ce4a #ce4b #ce4c #ce4d #ce4e #ce4f #ce5a #ce5b #ce5d #ce5f #ce5g #ce5h #ev1 #ra1 #ra4 #ra5 (est: 2024-10-21)

6.1. Filtrado de la entrada

6.1.1. Cómo *NO* se debe hacer

6.1.2. Extensión Filter

- `filter_input()`, `filter_has_var()`, `filter_var()`

- Filtros de validación y saneado
- 6.1.3. Expresiones regulares (PCRE)
- 6.2. Formateado de datos
 - 6.2.1. Números y cantidades monetarias
 - 6.2.2. Fechas
- 6.3. Sesiones
 - 6.3.1. Iniciar una sesión
 - `session_start()`
 - 6.3.2. Usar una sesión
 - `$_SESSION`
 - Ejemplos de uso
 - 6.3.3. Terminar una sesión
 - `session_destroy()`
 - `session_name()`
 - `session_id()`
 - `session_get_cookie_params()`
- 6.4. Seguridad básica
 - 6.4.1. Filtrar la entrada, escapar la salida
 - 6.4.2. Cross-Site Scripting (XSS)
 - No persistente
 - Persistente
 - Escapado de la salida
 - HTML Purifier
 - 6.4.3. `htmlspecialchars()`
- 6.5. Seguridad y persistencia
 - 6.5.1. Contraseñas
 - <https://www.md5online.org/>
 - <https://www.sha1online.org/>
 - `password_hash()`
 - `password_verify()`
 - 6.5.2. Inyección de código SQL
 - 6.5.3. Cross-Site Request Forgery (CSRF)

7 PROGRAMACIÓN AVANZADA EN PHP *#ce4g #ce5g #ce5h #ev1 #ra2 #ra3 #ra4 #ra5* (est: 2024-10-28)

- 7.1. Diseño de aplicaciones orientadas a objetos
 - 7.1.1. Clases

7.1.2. Propiedades

- Predeterminadas
- Dinámicas

7.1.3. Métodos

- Referencia `$this`
- Constructores y destructores

7.1.4. Constantes

- `self`

7.1.5. Herencia

- `parent`
- Sobreescritura de métodos

7.1.6. Miembros estáticos

- Constantes
- Métodos estáticos
- Propiedades estáticas
- Enlace estático en tiempo de ejecución

7.1.7. Interfaces

7.1.8. Traits

7.1.9. Contravarianza y covarianza

7.1.10. La clase `stdClass`

- Conversión de `array` a `object` y viceversa.

7.2. Espacios de nombres

7.3. Funciones anónimas

7.3.1. Clausuras

7.4. Callables

7.4.1. `call_user_func()`

7.4.2. `array_map()` y `array_reduce()`

8 INTEROPERABILIDAD *#ce1e #ce1g #ce4g #ce5g #ce5h #ev1 #ra1 #ra4 #ra5* (est: 2024-11-04)

8.1. Versionado semántico

8.2. Composer

8.2.1. Paquetes

8.2.2. Packagist

8.2.3. Dependencias

- `composer.json` y `composer.lock`

8.2.4. Versiones y restricciones

- Versión exacta

- Rango (>, >=, <, <=, !=, , , , | |)
- Guión (-)
- Asterisco (*)
- Tilde (`)
- Circunflejo (^)
- Nombres de rama
 - dev-master
 - 5.1.x-dev
- Estabilidad mínima
- Comprobador online de restricciones

8.2.5. Comandos básicos

- require
- install
- update
- dump-autoload

8.2.6. Entornos de desarrollo y producción

8.3. Autocarga de clases

8.3.1. spl_autoload_register()

8.3.2. PSR-4

8.3.3. Autoloader de Composer

8.4. Ejemplos

8.4.1. mpdf/mpdf

8.4.2. ramsey/uuid

8.4.3. doctrine/inflector

8.4.4. nesbot/carbon

8.5. Recomendaciones PSR del PHP-FIG (Framework Interop Group)

8.5.1. PSR-1: Basic Coding Standard

8.5.2. PSR-2: Coding Style Guide

8.5.3. PSR-4: Autoloading Standard

8.5.4. PSR-5: PHPDoc Standard (borrador)

8.5.5. PSR-11: Extended Coding Style Guide (borrador)

8.5.6. PSR-19: PHPDoc tags (borrador)

8.6. Herramientas externas *#opcional*

8.6.1. PHP_CodeSniffer

8.6.2. PHP-CS-Fixer

9 INTRODUCCIÓN A LARAVEL #ce1d #ce1e #ce1g #ce4g #ev1 #ra1 #ra4 (est: 2024-11-11)

- 9.1. Frameworks, microframeworks y librerías
- 9.2. Patrón Modelo-Vista-Controlador (MVC)
 - 9.2.1. Modelos
 - 9.2.2. Vistas
 - 9.2.3. Controladores
 - 9.2.4. Rutas
- 9.3. Laravel
 - 9.3.1. ¿Qué es Laravel
 - 9.3.2. ¿Cómo es Laravel comparado con otros frameworks?
 - 9.3.3. Versiones de Laravel
- 9.4. Instalación, requisitos y puesta en marcha
 - 9.4.1. Requisitos previos
 - 9.4.2. Instalación de Laravel
- 9.5. ¡Hola, mundo!
- 9.6. Herramientas de desarrollo
 - 9.6.1. Clockwork
 - 9.6.2. Trazas de depuración
- 9.7. Estilo del código

10 ARQUITECTURA DE LARAVEL #ce1d #ce1e #ce1g #ce4g #ce5e #ev1 #ra1 #ra4 #ra5 (est: 2024-11-18)

- 10.1. El ciclo de vida de una petición
- 10.2. El contenedor de servicios
- 10.3. Proveedores de servicios
- 10.4. Fachadas

11 GESTIÓN DE PETICIONES EN LARAVEL #ce4g #ce5f #ce5g #ce5h #ev1 #ra4 #ra5 (est: 2024-11-25)

- 11.1. Enrutado
- 11.2. Middlewares
- 11.3. Peticiones
- 11.4. Respuestas
- 11.5. Controladores
- 11.6. Sesiones y cookies
- 11.7. Gestión de errores

12 VISUALIZACIÓN DE DATOS EN LARAVEL #ce2a #ce2b #ce2c #ce2d #ce2e #ce2g #ce2h #ce4g

 	MODELO DE PROGRAMACIÓN ANUAL	MD8101	
		VERSIÓN 0	Pág. 23 de 44

#ce5a #ce5b #ce5c #ce5d #ce5h #ce8a #ce8b #ce8c #ce8d #ce8e #ce8f #ce8g #ev2 #ra2 #ra3 #ra4 #ra5 #ra8 (est: 2025-01-08)

- 12.1. Vistas
- 12.2. Plantillas Blade
- 12.3. Compilación de assets
- 12.4. Generación de URLs

13 BASES DE DATOS EN LARAVEL *#ce2a #ce2b #ce2c #ce2d #ce2e #ce2g #ce2h #ce4g #ce5f #ce5g #ce5h #ce6a #ce6b #ce6c #ce6d #ce6e #ce6f #ce6g #ev2 #ra2 #ra3 #ra4 #ra5 #ra6* (est: 2025-01-15)

- 13.1. Conceptos básicos
- 13.2. Query Builder
- 13.3. Migraciones
- 13.4. Paginación
- 13.5. Semillas

14 MAPEADO OBJETO-RELACIONAL CON ELOQUENT *#ce2a #ce2b #ce2c #ce2d #ce2e #ce2g #ce2h #ce4g #ce5f #ce5g #ce5h #ce6a #ce6b #ce6c #ce6d #ce6e #ce6f #ce6g #ev2 #ra2 #ra3 #ra4 #ra5 #ra6* (est: 2025-01-22)

- 14.1. Conceptos básicos
- 14.2. Relaciones
 - 14.2.1. Uno a uno
 - 14.2.2. Uno a muchos
 - 14.2.3. Muchos a muchos
 - 14.2.4. Relaciones polimórficas
 - 14.2.5. Carga ansiosa
 - El problema del $N + 1$
- 14.3. Colecciones
- 14.4. Mutadores y castings

15 CREACIÓN Y VALIDACIÓN DE FORMULARIOS EN LARAVEL *#ce2a #ce2b #ce2c #ce2d #ce2e #ce2g #ce2h #ce3e #ce3f #ce4g #ce5b #ce5d #ce5h #ce6f #ce6g #ce8a #ce8b #ce8c #ce8d #ce8e #ce8f #ce8g #ev2 #ra2 #ra3 #ra4 #ra5 #ra6 #ra8* (est: 2025-01-29)

- 15.1. Creación de formularios
- 15.2. Protección contra CSRF
- 15.3. Validación de la entrada
- 15.4. Subida de archivos

16 **SEGURIDAD Y CACHEADO EN LARAVEL** *#ce2a #ce2b #ce2c #ce2d #ce2e #ce2g #ce2h #ce4d #ce4e #ce4g #ce5g #ce5h #ce6g #ev2 #ra2 #ra3 #ra4 #ra5 #ra6 #ra8* (est: 2025-02-05)

- 16.1. Autenticación
- 16.2. Autorización
- 16.3. Verificación de email
- 16.4. Cifrado
- 16.5. Hashing
- 16.6. Reinicio de contraseñas

17 **CARACTERÍSTICAS ADICIONALES DE LARAVEL** *#ce2a #ce2b #ce2c #ce2d #ce2e #ce2g #ce2h #ce4g #ce5d #ce5g #ce5h #ce6g #ce8a #ce8b #ce8c #ce8d #ce8e #ce8f #ce8g #ev2 #ra2 #ra3 #ra4 #ra5 #ra6 #ra8* (est: 2025-02-12)

- 17.1. Correo electrónico
- 17.2. Helpers
- 17.3. Eventos
- 17.4. Contratos
- 17.5. Colecciones
- 17.6. Colas
- 17.7. Planificación de tareas
- 17.8. Almacenamiento de archivos

18 **CALIDAD** *#ce3g #ce4g #ce5g #ce5h #ce6e #ce6g #ev2 #ra3 #ra4 #ra5 #ra6* (est: 2025-02-19)

- 18.1. Pruebas
 - 18.1.1. Pruebas HTTP
 - 18.1.2. Pruebas en consola
 - 18.1.3. Pruebas en navegador
 - 18.1.4. Bases de datos
 - 18.1.5. Mocking
- 18.2. Depuración
 - 18.2.1. dd()
 - 18.2.2. ddd()
 - 18.2.3. Clockwork
- 18.3. Documentación
 - 18.3.1. API documentation generator for Yii2
 - 18.3.2. GitHub Pages
- 18.4. Mantenimiento y calidad del código
 - 18.4.1. CodeSniffer

18.4.2. CS_Fixer

18.4.3. Code Climate

19 COMPUTACIÓN EN LA NUBE #ce1d #ce4g #ce5e #ev2 #ra1 #ra4 #ra5 (est: 2025-02-26)

19.1. Entornos de ejecución

19.1.1. Desarrollo

19.1.2. Producción

19.1.3. Pruebas

19.1.4. Preproducción

19.2. Cloud computing vs hosting

19.3. Cloud computing vs VPS

19.4. Servicios por capas

19.4.1. IaaS

19.4.2. PaaS

19.4.3. SaaS

19.5. 12 Factores

19.6. Railway

19.6.1. Railway CLI

19.6.2. Creación y despliegue de aplicaciones

19.6.3. Railway PostgreSQL

19.6.4. Variables de entorno

19.6.5. Despliegues

20 SERVICIOS WEB CON LARAVEL #ce1a #ce1b #ce2a #ce2b #ce2c #ce2d #ce2e #ce2g #ce2h #ce7a #ce7b #ce7c #ce7d #ce7e #ce7f #ce7g #ce7h #ev2 #ra1 #ra2 #ra3 #ra7 (est: 2025-03-05)

20.1. Recursos API

20.2. Serialización

20.3. Limitación de frecuencia de peticiones

21 APLICACIONES WEB HÍBRIDAS #ce2a #ce2b #ce2c #ce2d #ce2e #ce2g #ce2h #ce9a #ce9b #ce9c #ce9d #ce9e #ce9f #ce9g #ce9h #ev2 #ra2 #ra9 (est: 2025-03-12)

21.1. Reutilización de código e información

21.2. Utilización de información proveniente de repositorios

21.2.1. UDDI

21.3. Incorporación de funcionalidades específicas

21.4. Sindicación y formatos de redifusión

21.4.1. RSS

21.4.2. ATOM

4.3. Correspondencia con resultados de aprendizaje y criterios de evaluación

El símbolo «×» representa que en esa unidad didáctica se trabaja dicho resultado de aprendizaje pero no es el elemento fundamental de evaluación.

Unidades didácticas	#ra1	#ra2	#ra3	#ra4	#ra5	#ra6	#ra7	#ra8	#ra9
1. Introducción a la tecnología web	#ce1a #ce1b #ce1c #ce1d #ce1e #ce1g								
2. Conceptos básicos de PHP (I)		#ce2d #ce2e #ce2f #ce2g	×	#ce4f					
3. Conceptos básicos de PHP (II)		#ce2d #ce2e #ce2g #ce2h	#ce3a #ce3b #ce3c #ce3d #ce3g	#ce4g					
4. Desarrollo de aplicaciones con PHP (I)	#ce1a #ce1b #ce1d #ce1f	#ce2a #ce2b #ce2c #ce2d #ce2e #ce2f #ce2g #ce2h	#ce3a #ce3b #ce3c #ce3d #ce3e #ce3f #ce3g	#ce4g	#ce5a #ce5d #ce5g #ce5h				
5. Persistencia de datos con PHP		×	×	#ce4a #ce4b #ce4c #ce4g	#ce5f #ce5g #ce5h	#ce6a #ce6b #ce6c #ce6e			
6. Desarrollo de aplicaciones con PHP (II)	#ce1a			#ce4a #ce4b #ce4c #ce4d #ce4e #ce4f	#ce5a #ce5b #ce5d #ce5f #ce5g #ce5h				
7. Programación avanzada en PHP		×	×	#ce4g	#ce5g #ce5h				

Unidades didácticas	#ra1	#ra2	#ra3	#ra4	#ra5	#ra6	#ra7	#ra8	#ra9
8. Interoperabilidad	#ce1e #ce1g			#ce4g	#ce5g #ce5h				
9. Introducción a Laravel	#ce1d #ce1e #ce1g			#ce4g					
10. Arquitectura de Laravel	#ce1d #ce1e #ce1g			#ce4g	#ce5e				
11. Gestión de peticiones en Laravel				#ce4g	#ce5f #ce5g #ce5h				
12. Visualización de datos en Laravel		#ce2a #ce2b #ce2c #ce2d #ce2e #ce2g #ce2h	×	#ce4g	#ce5a #ce5b #ce5c #ce5d #ce5h			#ce8a #ce8b #ce8c #ce8d #ce8e #ce8f #ce8g	
13. Bases de datos en Laravel		#ce2a #ce2b #ce2c #ce2d #ce2e #ce2g #ce2h	×	#ce4g	#ce5f #ce5g #ce5h	#ce6a #ce6b #ce6c #ce6d #ce6e #ce6f #ce6g			
14. Mapeado objeto-relacional con Eloquent		#ce2a #ce2b #ce2c #ce2d #ce2e #ce2g #ce2h	×	#ce4g	#ce5f #ce5g #ce5h	#ce6a #ce6b #ce6c #ce6d #ce6e #ce6f #ce6g			
15. Creación y validación de formularios en Laravel		#ce2a #ce2b #ce2c #ce2d #ce2e #ce2g #ce2h	#ce3e #ce3f	#ce4g	#ce5b #ce5d #ce5h	#ce6f #ce6g		#ce8a #ce8b #ce8c #ce8d #ce8e #ce8f #ce8g	

Unidades didácticas	#ra1	#ra2	#ra3	#ra4	#ra5	#ra6	#ra7	#ra8	#ra9
16. Seguridad y cacheado en Laravel		#ce2a #ce2b #ce2c #ce2d #ce2e #ce2g #ce2h	×	#ce4d #ce4e #ce4g	#ce5g #ce5h	#ce6g		×	
17. Características adicionales de Laravel		#ce2a #ce2b #ce2c #ce2d #ce2e #ce2g #ce2h	×	#ce4g	#ce5d #ce5g #ce5h	#ce6g		#ce8a #ce8b #ce8c #ce8d #ce8e #ce8f #ce8g	
18. Calidad			#ce3g	#ce4g	#ce5g #ce5h	#ce6e #ce6g			
19. Computación en la nube	#ce1d			#ce4g	#ce5e				
20. Servicios web con Laravel	#ce1a #ce1b	#ce2a #ce2b #ce2c #ce2d #ce2e #ce2g #ce2h	×				#ce7a #ce7b #ce7c #ce7d #ce7e #ce7f #ce7g #ce7h		
21. Aplicaciones web híbridas		#ce2a #ce2b #ce2c #ce2d #ce2e #ce2g #ce2h						#ce9a #ce9b #ce9c #ce9d #ce9e #ce9f #ce9g #ce9h	

5. Evaluación y calificación

La evaluación tendrá como finalidad determinar el nivel de competencia de los alumnos y la consecución de los objetivos del módulo. Se desarrollará de forma continua, y atenderá a los siguientes aspectos:

- Aprendizaje autónomo, viendo la capacidad del alumno para interiorizar, gestionar y participar en los procesos de aprendizaje propios.

 	MODELO DE PROGRAMACIÓN ANUAL	MD8101	
		VERSIÓN 0	Pág. 29 de 44

- Comprensión del lenguaje común.
- Adquisición de conceptos básicos del módulo profesional que permiten al alumno incluirlos como un elemento más de su realidad profesional.
- Participación y trabajo en grupo, viendo la capacidad que tiene este de escuchar y debatir las diferentes soluciones de un problema.
- Nivel de abstracción alcanzado.

Durante el desarrollo del curso existirán, al menos, una evaluación inicial, dos evaluaciones parciales (al final de cada trimestre) y una evaluación final.

La evaluación se realizará atendiendo a los criterios de evaluación asociados a cada resultado de aprendizaje del módulo. Para ello, se usarán instrumentos de evaluación que medirán si el alumnado cumple con los criterios de evaluación y en qué grado.

- En cada **evaluación parcial** se medirá el grado de *cumplimiento* de los *criterios de evaluación* trabajados y evaluados en el trimestre correspondiente.

Se entenderá que el alumno **cumple** un determinado **criterio de evaluación** si supera al menos un instrumento de evaluación que mida el grado de cumplimiento de dicho criterio de evaluación.

- En la **evaluación final** se determinará si el alumnado *alcanza*, y en qué medida, los *resultados de aprendizaje* del módulo a partir de los criterios de evaluación trabajados y evaluados que estén asociados a cada resultado de aprendizaje y en función de la ponderación que tenga cada criterio de evaluación en su correspondiente resultado de aprendizaje, según se establece en el apartado 5.3.

Para la superación del módulo se requerirá que el alumno **alcance** todos los **resultados de aprendizaje** del módulo.

Para que el seguimiento de dicha evaluación sea factible, el alumno deberá asistir con regularidad a las clases y participar activamente en las mismas, de forma que una sistemática y frecuente falta de asistencia a clase supondrá para el alumno la **pérdida de la evaluación continua** y sólo tendrá derecho a una prueba final. Asimismo, se requiere que el alumno **acceda al menos diariamente a las plataformas iPasen y Moodle Centros** y que **revise diariamente su correo en el dominio @iesdonana.org** para informarse puntualmente de las novedades que pudieran darse en el módulo, quedando claro que **es responsabilidad del alumno informarse activamente sobre las mismas**.

 	MODELO DE PROGRAMACIÓN ANUAL	MD8101	
		VERSIÓN 0	Pág. 30 de 44

5.1. Valoración general de los instrumentos de evaluación y calificación

Los instrumentos de evaluación y calificación usados durante el curso para medir el grado de cumplimiento de los criterios de evaluación se clasifican en dos grandes tipos:

- Trabajos, actividades y ejercicios (casa, clase, grupo) (tipo **TRA**).
- Pruebas evaluativas (tipo **EXA**).

La evaluación de cualquiera de los trabajos, actividades, ejercicios y pruebas evaluativas podrá requerir, a criterio y discreción del profesor del módulo, la defensa de los mismos en una entrevista individual con el alumno para garantizar la autoría y la adquisición adecuada de las competencias correspondientes.

Los trabajos, actividades y/o ejercicios (**TRA**) versarán sobre los contenidos trabajados en una unidad didáctica o bloque de unidades didácticas conceptualmente relacionadas. (Esto significa, en consecuencia, que no es obligatoria la realización de trabajos, actividades y/o ejercicios en cada unidad didáctica, sino que a tales efectos se pueden agrupar varias unidades didácticas.)

Dentro de este grupo podrá incluirse alguna prueba (tipo test o de respuestas cortas) a responder de forma individual sobre conocimientos teóricos de determinados aspectos básicos asociados a unidades (o conjunto de unidades) didácticas concretas.

Asimismo, se realizará un examen al final de cada evaluación parcial (**EXA**), coincidiendo aproximadamente con el final del trimestre correspondiente. Debido al carácter de evaluación continua del módulo, así como del hecho de que cada contenido trabajado se asienta sobre los anteriores, es posible que el alumno requiera, para la evaluación positiva de un trimestre, el conocimiento necesario de contenidos de trimestres anteriores.

5.2. Calificación

5.2.1. De un criterio de evaluación

La calificación de cada criterio de evaluación, una vez trabajado y evaluado, se valorará de 0,0 a 10,0 y se calculará como la calificación obtenida en el instrumento de evaluación y calificación usado para medir el grado de cumplimiento de ese criterio de evaluación.

Se entenderá que el alumno **cumple un criterio de evaluación** si la calificación que ha obtenido en el mismo es igual o superior a 4,5.

Si, a lo largo del curso, se usaran varios instrumentos para evaluar y calificar el mismo criterio de evaluación, la calificación de dicho criterio de evaluación se calcularía de la siguiente forma:

1. Se agruparán las calificaciones obtenidas en los diferentes instrumentos que hayan medido

ese criterio de evaluación en función del tipo de instrumento usado para evaluarlo (**TRA** o **EXA**). Se obtendrán así dos grupos de calificaciones, uno por cada tipo de instrumento.

2. En cada grupo, se seleccionará la calificación máxima dentro de ese grupo.
3. Se calculará la media ponderada (M) de las dos calificaciones resultantes del punto anterior, atendiendo a las siguientes ponderaciones:

TRA	50 %
EXA	50 %

Si, por algún motivo, no se usaran instrumentos de uno de los dos tipos (**TRA** o **EXA**) durante la evaluación, el otro apartado soportaría el 100 % de la carga evaluativa, de forma que la evaluación y calificación resultaría únicamente de dicho apartado.

4. Se calcula la calificación del criterio de evaluación de la siguiente forma, que garantiza que un criterio de evaluación califique como cumplido cuando haya al menos un instrumento con una calificación mayor o igual a 4,5 que mida ese criterio de evaluación:

$$CE_{i,j} = \begin{cases} \max(M_{ij}; 4,5) & \text{si } \max I_{ij} \geq 4,5 \\ M_{ij} & \text{e. o. c.} \end{cases}$$

Donde:

$CE_{i,j}$ = Calificación del criterio de evaluación j asociado al resultado de aprendizaje i .

M_{ij} = Media ponderada calculada en el punto 3 anterior para el criterio de evaluación j asociado al resultado de aprendizaje i .

I_{ij} = Conjunto de todas las calificaciones obtenidas por el alumno en todos los instrumentos de evaluación y calificación que miden el criterio de evaluación j asociado al resultado de aprendizaje i .

En consecuencia:

- La calificación de un determinado criterio de evaluación podría ir cambiando a lo largo del curso en función de los instrumentos de evaluación y calificación realizados por el alumno para evaluar, recuperar, mejorar su calificación, etcétera.
- La calificación de un criterio de evaluación no podrá ser inferior a 4,5 si el alumno ha obtenido una calificación igual o superior a 4,5 en algún instrumento de evaluación y calificación que haya medido ese criterio de evaluación.

5.2.2. De un resultado de aprendizaje

La calificación de cada resultado de aprendizaje, según se recoge en el algoritmo 1, se valorará de 0,0 a 10,0 y se calculará como una media ponderada de los criterios de evaluación trabajados y evaluados que estén asociados a ese resultado de aprendizaje, siendo requisito imprescindible para alcanzar un resultado de aprendizaje el cumplir todos los criterios de evaluación trabajados y evaluados; en caso contrario, la calificación del resultado de aprendizaje no podrá ser superior a 4,0.

Algoritmo 1 (Cálculo de la calificación de un resultado de aprendizaje)

$$M_i = \frac{\sum_{j \in C_i} CE_{i,j} \cdot p_{ij}}{\sum_{j \in C_i} p_{ij}}$$

$$RA_i = \begin{cases} M_i & \text{si } \min_{c \in C_i} \{c\} \geq 4,5 \\ \min(M_i; 4,0) & \text{e. o. c.} \end{cases}$$

Donde:

M_i = Media ponderada de los criterios de evaluación asociados al resultado de aprendizaje i .

RA_i = Calificación del resultado de aprendizaje i .

$CE_{i,j}$ = Calificación del criterio de evaluación j asociado al resultado de aprendizaje i .

C_i = Conjunto de las calificaciones de los criterios de evaluación trabajados y evaluados, asociados al resultado de aprendizaje i .

p_{ij} = Ponderación del criterio de evaluación j en el resultado de aprendizaje i , según se determina en el apartado 5.3.

Se entenderá que el alumno ha **alcanzado** un **resultado de aprendizaje** si la calificación que ha obtenido en el mismo es igual o superior a 4,5.

5.2.3. Calificaciones parciales

La calificación del módulo en cada evaluación parcial (primer y segundo trimestres por separado), según se recoge en el algoritmo 2, se valorará de 0,0 a 10,0 y se calculará como la media

aritmética de las calificaciones de los criterios de evaluación trabajados y evaluados durante el trimestre correspondiente, siendo requisito imprescindible para aprobar una evaluación parcial el cumplir todos los criterios de evaluación trabajados y evaluados; en caso contrario, la calificación de la evaluación parcial no podrá ser superior a 4,0.

Algoritmo 2 (Cálculo de una calificación parcial)

$$M_i = \frac{\sum_{c \in C_i} c}{|C_i|}$$

$$P_i = \begin{cases} M_i & \text{si } \min_{c \in C_i} \{c\} \geq 4,5 \\ \min(M_i; 4,0) & \text{e. o. c.} \end{cases}$$

Donde:

M_i = Media aritmética de las calificaciones de los criterios de evaluación trabajados en el trimestre $i \in \{1,2\}$.

C_i = Conjunto de las calificaciones de los criterios de evaluación trabajados y evaluados en el trimestre $i \in \{1,2\}$.

P_i = Calificación parcial del trimestre $i \in \{1,2\}$.

Se entenderá que el alumno ha **aprobado** una **evaluación parcial** si la calificación que ha obtenido en la misma es igual o superior a 4,5. La calificación introducida en el sistema Séneca será la calificación parcial redondeada al entero más cercano, y no podrá ser inferior a 1,0.

5.2.4. Calificación final

La calificación del módulo en la evaluación final se calculará como la media aritmética de las calificaciones de los diferentes resultados de aprendizaje, según se recoge en el algoritmo 3.

Algoritmo 3 (Cálculo de la calificación final)

$$M = \frac{\sum_{r \in R} r}{|R|}$$

$$F = \begin{cases} M & \text{si } \min_{r \in R} \{r\} \geq 4,5 \\ \min(M; 4,0) & \text{e. o. c.} \end{cases}$$

Donde:

M = Media aritmética de las calificaciones de los resultados de aprendizaje del módulo.

R = Conjunto de las calificaciones de los resultados de aprendizaje del módulo.

F = Calificación final del módulo.

La calificación introducida en el sistema Séneca será la calificación final redondeada al entero más cercano, y no podrá ser inferior a 1,0. Asimismo, si el alumno no ha alcanzado todos los resultados de aprendizaje, la calificación final del módulo no podrá ser superior a 4,0.

5.3. Ponderación de los criterios de evaluación

Los criterios de evaluación de cada resultado de aprendizaje se ponderan con los siguientes pesos relativos:

Resultado de aprendizaje (RA _{<i>i</i>})	Criterio de evaluación (CE _{<i>i</i>·<i>j</i>})	Porcentaje (<i>p_{ij}</i>)
RA ₁ #ra1	CE _{1·a} #ce1a	14,2 %
RA ₁ #ra1	CE _{1·b} #ce1b	14,2 %
RA ₁ #ra1	CE _{1·c} #ce1c	14,2 %
RA ₁ #ra1	CE _{1·d} #ce1d	14,2 %
RA ₁ #ra1	CE _{1·e} #ce1e	14,2 %
RA ₁ #ra1	CE _{1·f} #ce1f	14,2 %
RA ₁ #ra1	CE _{1·g} #ce1g	14,2 %
RA ₂ #ra2	CE _{2·a} #ce2a	12,5 %
RA ₂ #ra2	CE _{2·b} #ce2b	12,5 %
RA ₂ #ra2	CE _{2·c} #ce2c	12,5 %
RA ₂ #ra2	CE _{2·d} #ce2d	12,5 %
RA ₂ #ra2	CE _{2·e} #ce2e	12,5 %

Resultado de aprendizaje (RA _i)	Criterio de evaluación (CE _{i·j})	Porcentaje (p _{ij})
RA ₂ #ra2	CE _{2·f} #ce2f	12,5 %
RA ₂ #ra2	CE _{2·g} #ce2g	12,5 %
RA ₂ #ra2	CE _{2·h} #ce2h	12,5 %
RA ₃ #ra3	CE _{3·a} #ce3a	14,2 %
RA ₃ #ra3	CE _{3·b} #ce3b	14,2 %
RA ₃ #ra3	CE _{3·c} #ce3c	14,2 %
RA ₃ #ra3	CE _{3·d} #ce3d	14,2 %
RA ₃ #ra3	CE _{3·e} #ce3e	14,2 %
RA ₃ #ra3	CE _{3·f} #ce3f	14,2 %
RA ₃ #ra3	CE _{3·g} #ce3g	14,2 %
RA ₄ #ra4	CE _{4·a} #ce4a	16,6 %
RA ₄ #ra4	CE _{4·b} #ce4b	16,6 %
RA ₄ #ra4	CE _{4·c} #ce4c	16,6 %
RA ₄ #ra4	CE _{4·d} #ce4d	16,6 %
RA ₄ #ra4	CE _{4·e} #ce4e	16,6 %
RA ₄ #ra4	CE _{4·f} #ce4f	16,6 %
RA ₅ #ra5	CE _{5·a} #ce5a	12,5 %
RA ₅ #ra5	CE _{5·b} #ce5b	12,5 %
RA ₅ #ra5	CE _{5·c} #ce5c	12,5 %
RA ₅ #ra5	CE _{5·d} #ce5d	12,5 %
RA ₅ #ra5	CE _{5·e} #ce5e	12,5 %
RA ₅ #ra5	CE _{5·f} #ce5f	12,5 %
RA ₅ #ra5	CE _{5·g} #ce5g	12,5 %
RA ₅ #ra5	CE _{5·h} #ce5h	12,5 %
RA ₆ #ra6	CE _{6·a} #ce6a	14,2 %
RA ₆ #ra6	CE _{6·b} #ce6b	14,2 %
RA ₆ #ra6	CE _{6·c} #ce6c	14,2 %
RA ₆ #ra6	CE _{6·d} #ce6d	14,2 %
RA ₆ #ra6	CE _{6·e} #ce6e	14,2 %
RA ₆ #ra6	CE _{6·f} #ce6f	14,2 %
RA ₆ #ra6	CE _{6·g} #ce6g	14,2 %
RA ₇ #ra7	CE _{7·a} #ce7a	12,5 %
RA ₇ #ra7	CE _{7·b} #ce7b	12,5 %
RA ₇ #ra7	CE _{7·c} #ce7c	12,5 %
RA ₇ #ra7	CE _{7·d} #ce7d	12,5 %
RA ₇ #ra7	CE _{7·e} #ce7e	12,5 %
RA ₇ #ra7	CE _{7·f} #ce7f	12,5 %
RA ₇ #ra7	CE _{7·g} #ce7g	12,5 %
RA ₇ #ra7	CE _{7·h} #ce7h	12,5 %
RA ₈ #ra8	CE _{8·a} #ce8a	14,2 %
RA ₈ #ra8	CE _{8·b} #ce8b	14,2 %

Resultado de aprendizaje (RA _i)	Criterio de evaluación (CE _{i.j})	Porcentaje (p _{ij})
RA ₈ #ra8	CE _{8.c} #ce8c	14,2 %
RA ₈ #ra8	CE _{8.d} #ce8d	14,2 %
RA ₈ #ra8	CE _{8.e} #ce8e	14,2 %
RA ₈ #ra8	CE _{8.f} #ce8f	14,2 %
RA ₈ #ra8	CE _{8.g} #ce8g	14,2 %
RA ₉ #ra9	CE _{9.a} #ce9a	12,5 %
RA ₉ #ra9	CE _{9.b} #ce9b	12,5 %
RA ₉ #ra9	CE _{9.c} #ce9c	12,5 %
RA ₉ #ra9	CE _{9.d} #ce9d	12,5 %
RA ₉ #ra9	CE _{9.e} #ce9e	12,5 %
RA ₉ #ra9	CE _{9.f} #ce9f	12,5 %
RA ₉ #ra9	CE _{9.g} #ce9g	12,5 %
RA ₉ #ra9	CE _{9.h} #ce9h	12,5 %

5.4. Medidas de recuperación y mejora de la calificación final

A lo largo del curso (idealmente, a comienzos del trimestre) se establecerán instrumentos de recuperación destinados a que el alumnado cumpla los criterios de evaluación trabajados y evaluados en trimestres anteriores que no hayan podido cumplir hasta ahora al no haber podido superar los instrumentos de evaluación correspondientes.

Los criterios de evaluación que se trabajen en unidades didácticas temporalizadas en varios trimestres podrían no necesitar de instrumentos de recuperación, ya que el alumno tendría varias oportunidades para cumplir el mismo criterio de evaluación a lo largo de varios trimestres con los instrumentos de evaluación propios de las unidades didácticas programadas para esos trimestres.

Los instrumentos de recuperación utilizados para que el alumnado pueda cumplir los criterios de evaluación que aún no ha cumplido, podrán coincidir o no con los utilizados anteriormente para evaluar el cumplimiento de tales criterios, a discreción del profesor. En particular, se podrán usar pruebas evaluativas (instrumentos del tipo **EXA**) como instrumentos de recuperación para que el alumnado pueda cumplir criterios de evaluación que aún no haya cumplido y que previamente se habían medido con instrumentos del tipo **TRA**.

Los alumnos que tengan criterios de evaluación pendientes de cumplir en la tercera evaluación, deberán presentarse directamente a la prueba final tal como se establece en el apartado 5.4.1. No hay, por tanto, pruebas de recuperación específicas para los contenidos de la tercera evaluación.

5.4.1. Prueba final

La normativa vigente establece la existencia de un examen o prueba a final de curso (usualmente en junio) para aquellos alumnos que no hayan podido superar el módulo mediante el procedimiento habitual de evaluación continua y que, por tanto, tengan pendiente alcanzar algún resultado de aprendizaje que no se haya podido alcanzar ni siquiera mediante los instrumentos de recuperación indicados anteriormente.

Asimismo, y según establece la misma normativa, esta prueba final permite al alumnado la superación del módulo incluso en el caso de no haber asistido con regularidad a clase a lo largo del curso. Es, por tanto, el único instrumento que posibilita el que un alumno que cuente con un número de faltas de asistencia superior al permitido (y que, por consiguiente, haya perdido el derecho a la evaluación continua) pueda demostrar la consecución de los objetivos del módulo y, en consecuencia, su superación.

Dicha prueba o examen final constará de varias partes separadas. De ellas, las partes que el alumno deberá realizar y superar dependerán de las calificaciones que haya obtenido en los diferentes resultados de aprendizaje a lo largo del curso, de manera que el alumno deberá realizar las partes correspondientes a los resultados de aprendizaje que aún no haya alcanzado.

Finalmente, la calificación obtenida en la prueba final se tendrá en cuenta para calcular la **calificación final del módulo** según se establece en 5.2.4.

El alumno que, debiendo presentarse a la prueba final, no lo haga, será calificado como «No evaluado» (NE).

5.4.2. Mejora de la calificación final

El alumno que, habiendo superado el módulo mediante el procedimiento habitual de evaluación continua, desee mejorar su calificación final («subir nota»), podrá hacerlo en la prueba final, manteniéndose la calificación final más alta de entre las obtenidas antes y después de presentarse a subir nota en dicha prueba final. A tal efecto, se indicará en la prueba final qué partes deberá realizar el alumnado que se haya presentado a subir nota.

6. Evaluación de la práctica docente

Se realizarán entre el alumnado dos encuestas de valoración trimestrales y anónimas, al final de cada trimestre, formadas por preguntas cuantitativas y cualitativas relacionadas con aspectos que afecten al módulo, su organización, su evaluación, los materiales empleados, el día a día de su práctica docente y el profesorado encargado de su docencia. A lo largo del curso se analizará la evolución de los resultados obtenidos globalmente y por cada pregunta para observar su

 	MODELO DE PROGRAMACIÓN ANUAL	MD8101	
		VERSIÓN 0	Pág. 38 de 44

progresión positiva o negativa y así poder tomar las acciones que se estime oportunas llegado el caso.

7. Orientaciones pedagógicas

Según queda recogido en [3], este módulo profesional contiene la formación necesaria para desempeñar la función de **desarrollo de aplicaciones y servicios destinados a su ejecución por servidores en entornos Web**.

La **función** de desarrollo de aplicaciones para servidores Web incluye aspectos como:

- La creación de aplicaciones de servidor que generan interfaces Web como resultado de su ejecución.
- La programación de métodos para almacenar, recuperar y gestionar mediante documentos Web información disponible en almacenes de datos.
- La generación de servicios reutilizables y accesibles mediante protocolos Web.
- El desarrollo de aplicaciones basadas en información y funcionalidades distribuidas.

Las **actividades profesionales** asociadas a esta función se aplican en el desarrollo y la adaptación de servicios y aplicaciones para servidores de aplicaciones y servidores web.

Las **líneas de actuación** en el proceso de enseñanza-aprendizaje que permiten alcanzar los objetivos del módulo profesional versarán sobre:

- El análisis de los métodos de generación dinámica de documentos Web.
- La integración del lenguaje de marcas con el código ejecutable en el servidor Web.
- El análisis, diferenciación y clasificación de las características y funcionalidades incorporadas en los entornos y lenguajes de programación de los servidores Web más difundidos.
- La utilización de características y funcionalidades específicas de los lenguajes de programación seleccionados.
- La modificación del código existente en soluciones Web heterogéneas para su adaptación a entornos específicos.
- El análisis y la utilización de funcionalidades aportadas por librerías generales y específicas de programación web en entorno servidor.
- La utilización de librerías para incorporar interactividad a los documentos Web generados de forma dinámica.

 	MODELO DE PROGRAMACIÓN ANUAL	MD8101	
		VERSIÓN 0	Pág. 39 de 44

8. Orientaciones metodológicas

El módulo es totalmente práctico, y el proceso de enseñanza-aprendizaje se fundamenta en la interacción continua y total de los alumnos con las herramientas software utilizadas y estudiadas a lo largo del curso.

El módulo construye el aprendizaje de forma progresiva, comenzando con el estudio de la tecnología web, separando el entorno cliente del servidor, seguido de la profundización en el desarrollo de aplicaciones web con lenguajes de script embebidos tanto directamente como mediante el uso de un framework apropiado para tal fin, finalizando con una introducción a las nuevas tendencias en el despliegue de aplicaciones web desde la perspectiva del entorno servidor.

De forma transversal, se hará hincapié continuamente en el aseguramiento de la calidad del producto resultante, así como del proceso de desarrollo y el código fuente desarrollado, poniendo especial interés en la metodología *TDD (Test-Driven Development)* y las pruebas automáticas.

La enseñanza se basará casi por completo en la realización de ejercicios y supuestos de aplicación, que obliguen al alumno a enfrentarse con las herramientas software necesarias para solucionarlos.

Las actividades propuestas podrán ser individuales o grupales, aplicando donde corresponda el concepto de *programación en pareja*¹ como vehículo de colaboración y aprendizaje compartido entre varios alumnos.

Finalmente, se incentivará al alumno para que mejore su comprensión mediante el auto-aprendizaje y la elaboración propia de ejercicios y desarrollos.

9. Recursos

9.1. Hardware

- Un ordenador para cada alumno, conectado a la red local del aula y esta, a su vez, a la troncal del Centro.
- Conexión a Internet de banda ancha.
- Cañón retroproyector.

9.2. Software

- Sistema operativo GNU/Linux (Ubuntu o Debian GNU/Linux, preferentemente).

¹ https://es.wikipedia.org/wiki/Programaci%C3%B3n_en_pareja

 	MODELO DE PROGRAMACIÓN ANUAL	MD8101	
		VERSIÓN 0	Pág. 40 de 44

- El resto de herramientas y aplicaciones necesarias se instalarán a través de Internet a lo largo del curso.

9.3. Online

- Plataforma **Moodle Centros**² de la Junta de Andalucía para el seguimiento general del módulo, incluyendo distribución de material y entrega de ejercicios y exámenes.
- **GitHub**³ como herramienta centralizada para compartir código y para la gestión integral de todo elemento satélite del mismo (control de versiones, desarrollo colaborativo, incidencias, etcétera).

9.4. Bibliografía

9.4.1. Principal

- Apuntes y documentación *online*⁴ suministrados por el profesor.
- Documentación de Git⁵ y GitHub⁶.
- Documentación de PHP⁷.
- Documentación de Laravel Framework⁸.
- Documentación de PostgreSQL⁹.

9.4.2. Complementaria

- López Sanz, M. y otros (2012). *Desarrollo web en entorno servidor*. Editorial RA-MA.

10. Atención a la diversidad

Se llevarán a cabo actividades de refuerzo o ampliación para aquellos alumnos que así lo requieran en función de las necesidades detectadas:

- Para los alumnos que muestren dificultades de aprendizaje, se propondrán **actividades**

² <https://educacionadistancia.juntadeandalucia.es/centros/cadiz>

³ <https://github.com>

⁴ <https://dwese.iesdonana.org>

⁵ <https://git-scm.com/book/en/v2>

⁶ <https://help.github.com>

⁷ <https://php.net>

⁸ <https://laravel.com>

⁹ <https://www.postgresql.org>

 	MODELO DE PROGRAMACIÓN ANUAL	MD8101	
		VERSIÓN 0	Pág. 41 de 44

de refuerzo destinadas a afianzar los aspectos conceptuales y procedimentales en los que el alumnado presente carencias.

- Para los alumnos que muestren un mayor grado de adquisición de competencias, se propondrán **actividades de ampliación** que supongan la investigación autónoma de contenidos opcionales (aquellos marcados con la etiqueta *#opcional*).

11. Temas transversales

Con el objeto de fomentar entre los alumnos el hábito de la lectura, se plantearán actividades individuales y en grupo en las que, para su resolución, se necesite leer información de distintas fuentes escritas, como artículos, blogs, páginas web, tutoriales, etc.

La evolución experimentada por la informática en los últimos años tiene como consecuencia su influencia inevitable en todos los aspectos de las relaciones entre las personas y entre éstas y el entorno. Además ha demostrado ser un medio valiosísimo para la educación cualquiera que sea el ámbito en el que se use. En concreto, en cuanto a los temas transversales propuestos:

- **Educación ambiental:** La utilización de la informática, en general, y sobre todo en los negocios, hace que grandes volúmenes de información puedan ser almacenados en soportes informáticos, discos, CD, ... y enviados de unos lugares a otros a través de las redes informáticas, evitándose de esta manera el consumo de grandes cantidades de papel y, por consiguiente, la destrucción de bosques, contribuyendo de alguna manera a la preservación de los medios naturales y medioambientales.
- **Educación del consumidor:** El análisis y la utilización de diferentes herramientas informáticas favorecen la capacidad del alumnado para decidir sobre los productos informáticos que debe adquirir y utilizar de manera ventajosa.
- **Educación para la salud:** Cuando se utilizan equipos informáticos se procura que el alumnado conozcan una serie de normas de higiene y seguridad en el trabajo, así como sobre las precauciones necesarias en el empleo de los equipos. De esta manera, se intenta que el alumnado conozca los principios de la ergonomía del puesto de trabajo, para que cualquier trabajo frente al ordenador resulte lo más agradable posible y no le cause ningún problema. En este sentido, resultan de interés las instrucciones elaboradas por el Instituto Nacional de Seguridad e Higiene en el Trabajo [8].
- **Educación para la igualdad de oportunidades entre ambos sexos:** Desde este módulo contamos con elementos para concienciar al alumnado sobre la igualdad de oportunidades para alumnos y alumnas:
 - Formando grupos mixtos de trabajo.

- Distribuyendo las tareas a realizar en la misma medida entre el alumnado de ambos sexos.
- Haciendo que todos utilicen los mismos o equivalentes equipos.
- Fomentando la participación de todos, sin distinciones de sexo.
- **Educación para el trabajo:** Respecto a este módulo encontramos los siguientes elementos:
 - Técnicas de trabajo en grupo: sujeción a unas reglas corporativas.
 - Colaboración de varias personas para la realización de un único trabajo.
- **Educación para la paz y la convivencia:** Se trabajan los elementos siguientes:
 - Acuerdos para la utilización de los mismos estándares en toda la comunidad internacional.
 - Respeto por las opiniones de los demás.
 - Aprender a escuchar.

12. Actuaciones para desarrollar la perspectiva de género

El conocimiento de la realidad existente es el primer paso a realizar para incorporar la perspectiva de género. De esta manera, se descubrirá la existencia de situaciones de desequilibrio entre mujeres y hombres en el desempeño de la actividad docente.

La perspectiva de género es trabajada de manera transversal y permanente en todas las Unidades Didácticas que componen esta programación. El IES Doñana como organización social en aplicación de esta óptica favorece, entre otros aspectos, la detección de estereotipos y la asignación de roles y responsabilidades, la evaluación del uso y control de los recursos puestos a disposición de hombres y mujeres con la finalidad última de introducir las modificaciones y medidas correctoras necesarias para eliminar las desigualdades detectadas en cualquier ámbito de la vida del centro y particularmente dentro del aula.

En el marco de esta programación, el análisis de estas circunstancias permite identificar las diferentes necesidades, intereses y perspectivas de mujeres y hombres sobre las que diseñar estrategias que equiparen las oportunidades de ambas partes en las distintas actuaciones que lo integran. Fundamentalmente en los siguientes círculos se realizan las actuaciones:

- Profesores–profesores
- Alumnos–alumnos y
- Alumnos–profesores

Implica tener en cuenta las siguientes cuestiones:

1. Valorar la situación de partida de hombres y mujeres.
2. Analizar las necesidades y obligaciones relacionadas con la actividad cotidiana en el centro y la posición social de hombres y mujeres en el centro.
3. Velar por el cumplimiento de la condición de igualdad de género en todos los ámbitos de actuación como cuestión de justicia y responsabilidad social.

12.1. Actuaciones generales permanentes

1. Revisión del material curricular para la eliminación de la transmisión de estereotipos o modelos de conductas determinados por el género, tipo identificación cultural de funciones realizadas tradicionalmente por hombres o mujeres.
2. Detectar las desigualdades y discriminaciones de género existentes en el centro para su tratamiento/denuncia pertinente.
3. Garantizar la participación equilibrada de hombres y mujeres en las distintas actividades en el aula y en el centro.
4. Velar porque el contenido gráfico y lingüístico de las acciones, materiales y dispositivos de formación y difusión carezca de cualquier carácter o pretensión discriminatoria.
5. Participación en las actividades propuestas por el Plan de Igualdad del centro articulado a través de actuaciones propias o la acción tutorial:
 - a) 25 de noviembre: Día Internacional de la Eliminación de la Violencia contra la Mujer.
 - b) 30 de enero: Resolución de conflictos de forma pacífica. Día de la Paz.
 - c) 8 de marzo: Día Internacional de la Mujer.

Desde el primer momento se advertirá al alumnado que el uso del vocabulario y expresiones propias del lenguaje hablado y escrito se llevará a cabo de forma extensiva a ambos géneros, de manera que cuando hablamos del «administrador» o el «programador» lo hacemos siempre considerando que dichos roles son de aplicación a hombres y mujeres por igual. Así pues, resultará innecesario y, por tanto, se evitará el uso de fórmulas tales como «administrador o administradora», que recargan el lenguaje sin aportar información adicional. Ello además va en consonancia con lo manifestado por la Real Academia Española, al afirmar que:

*«El español dispone de un mecanismo inclusivo: el masculino gramatical, que, como término no marcado de la oposición de género, puede referirse a grupos formados de hombres y mujeres y, en contextos genéricos o inespecíficos, a personas de uno u otro sexo».*¹⁰

¹⁰ <https://twitter.com/RAEinforma/status/1111565711653113856>

 	MODELO DE PROGRAMACIÓN ANUAL	MD8101	
		VERSIÓN 0	Pág. 44 de 44

Por otra parte, en el planteamiento y realización de tareas y ejercicios, se procurará el equilibrio en cuanto a presencia de actores de ambos géneros.

Referencias

- [1] Real Decreto 1147/2011, de 29 de julio (BOE n° 182 de 30 de julio) por el que se establece la ordenación general de la formación profesional del sistema educativo.
- [2] Real Decreto 686/2010, de 20 de mayo (BOE n° 143 de 12 de junio), por el que se establece el título de Técnico Superior en Desarrollo de Aplicaciones Web.
- [3] *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 (págs. 130 a 133 del BOJA n° 149 del 1 de agosto)¹¹.
- [4] *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 (BOJA n° 202 del 15 de octubre).
- [5] Ley Orgánica 3/2022, de 31 de marzo (BOE n.º 78 de 1 de abril), de ordenación e integración de la Formación Profesional.
- [6] Real Decreto 405/2023, de 29 de mayo (BOE n.º 132 de 3 de junio), por el que se actualizan los títulos de la formación profesional del sistema educativo de Técnico Superior en Desarrollo de Aplicaciones Multiplataforma y Técnico Superior en Desarrollo de Aplicaciones Web, de la familia profesional Informática y Comunicaciones, y se fijan sus enseñanzas mínimas.
- [7] Real Decreto 659/2023, de 18 de julio (BOE n.º 174 de 22 de julio), por el que se desarrolla la ordenación del Sistema de Formación Profesional.
- [8] Instituto Nacional de Seguridad e Higiene en el Trabajo. *Instrucción básica para el trabajador usuario de pantallas de visualización de datos.*¹²

¹¹ <http://www.juntadeandalucia.es/boja/boletines/2011/149/d/updf/d23.pdf\T1\textbackslash#page=17>

¹² http://www.insht.es/InshtWeb/Contenidos/Documentacion/TextosOnline/Guias_Ev_Riesgos/Instruccion_Pantallas/Instruccion_basica.pdf