

 Junta de Andalucía	 IES DOÑANA	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 2020/2021

Departamento de Informática y Comunicaciones

Jefe de Departamento: Ricardo Pérez López

Índice

1. Información general	3
2. Objetivos generales	3
3. Resultados de aprendizaje y criterios de evaluación	5
4. Instrumentos y procedimientos de evaluación y calificación	9
4.1. Valoración general de los contenidos	10
4.2. Calificación	10
4.2.1. Calificaciones parciales	11
4.2.2. Calificación final	12
4.3. Medidas de recuperación y mejora de las calificaciones	12
4.3.1. Recuperación del apartado TRA	12
4.3.2. Recuperación del apartado EXA	12
4.3.3. Mejora de las calificaciones	13
5. Contenidos y temporalización	13
5.1. Cuadro resumen	13
5.2. Esquema detalle	14
5.3. Correspondencia con resultados de aprendizaje y criterios de evaluación	35
6. Orientaciones pedagógicas	37
7. Orientaciones metodológicas	38
8. Recursos	39
8.1. Hardware	39
8.2. Software	39
8.3. Online	39
8.4. Bibliografía	40
8.4.1. Principal	40
8.4.2. Complementaria	40
9. Atención a la diversidad	40
10. Temas transversales	41
11. Actuaciones para desarrollar la perspectiva de género	42
11.1. Actuaciones generales permanentes	43

 Junta de Andalucía	 IES DOÑANA	MODELO DE PROGRAMACIÓN ANUAL		MD8101	
				VERSIÓN 0	Pág. 3 de 44

1. Información general

- **Normativa de aplicación:** [1], [2]
- **Equivalencia en créditos ECTS:** 12.
- **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 [1]:
 - c) Instalar módulos analizando su estructura y funcionalidad para gestionar servidores de aplicaciones.
 - d) Ajustar parámetros analizando la configuración para gestionar servidores de aplicaciones.
 - 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 empleados 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.

 Junta de Andalucía	 IES DOÑANA	MODELO DE PROGRAMACIÓN ANUAL		MD8101	
				VERSIÓN 0	Pág. 4 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 [1]:

- c) Gestionar servidores de aplicaciones adaptando su configuración en cada caso para permitir el despliegue de aplicaciones Web.
- d) Gestionar bases de datos, interpretando su diseño lógico y verificando integridad, consistencia, seguridad y accesibilidad de los datos.
- 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.

3. Resultados de aprendizaje y criterios de evaluación

Según se establece en [1], 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:

- CE1.a)* Se han caracterizado y diferenciado los modelos de ejecución de código en el servidor y en el cliente Web.
- CE1.b)* Se han reconocido las ventajas que proporciona la generación dinámica de páginas Web y sus diferencias con la inclusión de sentencias de guiones en el interior de las páginas Web.
- CE1.c)* Se han identificado los mecanismos de ejecución de código en los servidores Web.
- CE1.d)* Se han reconocido las funcionalidades que aportan los servidores de aplicaciones y su integración con los servidores Web.
- CE1.e)* Se han identificado y caracterizado los principales lenguajes y tecnologías relacionados con la programación Web en entorno servidor.
- CE1.f)* Se han verificado los mecanismos de integración de los lenguajes de marcas con los lenguajes de programación en entorno servidor.
- CE1.g)* Se han reconocido y evaluado las herramientas 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:

- CE2.a)* Se han reconocido los mecanismos de generación de páginas Web a partir de lenguajes de marcas con código embebido.
- CE2.b)* Se han identificado las principales tecnologías asociadas.
- CE2.c)* Se han utilizado etiquetas para la inclusión de código en el lenguaje de marcas.
- CE2.d)* Se ha reconocido la sintaxis del lenguaje de programación que se ha de utilizar.
- CE2.e)* Se han escrito sentencias simples y se han comprobado sus efectos en el documento resultante.
- CE2.f)* Se han utilizado directivas para modificar el comportamiento predeterminado.

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

CE2.g) Se han utilizado los distintos tipos de variables y operadores disponibles en el lenguaje.

CE2.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:

CE3.a) Se han utilizado mecanismos de decisión en la creación de bloques de sentencias.

CE3.b) Se han utilizado bucles y se ha verificado su funcionamiento.

CE3.c) Se han utilizado «arrays» para almacenar y recuperar conjuntos de datos.

CE3.d) Se han creado y utilizado funciones.

CE3.e) Se han utilizado formularios Web para interactuar con el usuario del navegador Web.

CE3.f) Se han empleado métodos para recuperar la información introducida en el formulario.

CE3.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:

CE4.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.

CE4.b) Se han utilizado sesiones para mantener el estado de las aplicaciones Web.

CE4.c) Se han utilizado «cookies» para almacenar información en el cliente Web y para recuperar su contenido.

CE4.d) Se han identificado y caracterizado los mecanismos disponibles para la autenticación de usuarios.

CE4.e) Se han escrito aplicaciones que integren mecanismos de autenticación de usuarios.

CE4.f) Se han realizado adaptaciones a aplicaciones Web existentes como gestores de contenidos u otras.

CE4.g) 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:

- CE5.a)* Se han identificado las ventajas de separar la lógica de negocio de los aspectos de presentación de la aplicación.
- CE5.b)* Se han analizado tecnologías y mecanismos que permiten realizar esta separación y sus características principales.
- CE5.c)* Se han utilizado objetos y controles en el servidor para generar el aspecto visual de la aplicación Web en el cliente.
- CE5.d)* Se han utilizado formularios generados de forma dinámica para responder a los eventos de la aplicación Web.
- CE5.e)* Se han identificado y aplicado los parámetros relativos a la configuración de la aplicación Web.
- CE5.f)* Se han escrito aplicaciones Web con mantenimiento de estado y separación de la lógica de negocio.
- CE5.g)* Se han aplicado los principios de la programación orientada a objetos.
- CE5.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:

- CE6.a)* Se han analizado las tecnologías que permiten el acceso mediante programación a la información disponible en almacenes de datos.
- CE6.b)* Se han creado aplicaciones que establezcan conexiones con bases de datos.
- CE6.c)* Se ha recuperado información almacenada en bases de datos.
- CE6.d)* Se ha publicado en aplicaciones Web la información recuperada.
- CE6.e)* Se han utilizado conjuntos de datos para almacenar la información.
- CE6.f)* Se han creado aplicaciones Web que permitan la actualización y la eliminación de información disponible en una base de datos.
- CE6.g)* Se han utilizado transacciones para mantener la consistencia de la información.
- CE6.h)* Se han probado y documentado las aplicaciones.

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

[RA7] Desarrolla servicios Web analizando su funcionamiento e implantando la estructura de sus componentes.

Criterios de evaluación:

- CE7.a)* Se han reconocido las características propias y el ámbito de aplicación de los servicios Web.
- CE7.b)* Se han reconocido las ventajas de utilizar servicios Web para proporcionar acceso a funcionalidades incorporadas a la lógica de negocio de una aplicación.
- CE7.c)* Se han identificado las tecnologías y los protocolos implicados en la publicación y utilización de servicios Web.
- CE7.d)* Se ha programado un servicio Web.
- CE7.e)* Se ha creado el documento de descripción del servicio Web.
- CE7.f)* Se ha verificado el funcionamiento del servicio Web.
- CE7.g)* Se ha consumido el servicio Web.

[RA8] Genera páginas Web dinámicas analizando y utilizando tecnologías 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 librerías y las tecnologías relacionadas con la generación por parte del servidor de páginas Web con guiones embebidos.
- CE8.d)* Se han utilizado estas tecnologías para generar páginas Web que incluyan interacción con el usuario en forma de advertencias y peticiones de confirmación.
- CE8.e)* Se han utilizado estas tecnologías, para generar páginas Web que incluyan verificación de formularios.
- CE8.f)* Se han utilizado estas tecnologías para generar páginas Web que incluyan modificación dinámica de su contenido y su estructura.
- CE8.g)* Se han aplicado estas tecnologías en la programación de aplicaciones Web.

[RA9] Desarrolla aplicaciones Web híbridas seleccionando y utilizando librerías de código 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 librerías de código y tecnologías aplicables en la creación de aplicaciones Web híbridas.

CE9.c) Se ha creado una aplicación Web que recupere y procese repositorios de información ya existentes.

CE9.d) Se han creado repositorios específicos a partir de información existente en Internet y en almacenes de información.

CE9.e) Se han utilizado librerías de código para incorporar funcionalidades específicas a una aplicación Web.

CE9.f) Se han programado servicios y aplicaciones Web utilizando como base información y código generados por terceros.

CE9.g) Se han probado, depurado y documentado las aplicaciones generadas.

4. Instrumentos y procedimientos de evaluación y calificación

La evaluación tendrá como finalidad determinar el nivel de competencia de los alumnos y la consecución de los objetivos. 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.
- 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.

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 un examen final. Asimismo, se requiere que el alumno **acceda al menos diariamente a la plataforma Ágora** 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**.

4.1. Valoración general de los contenidos

Los contenidos se ponderarán en base a los siguientes porcentajes:

Trabajos, actividades y ejercicios (casa, clase, grupo) (TRA)	30 %
Pruebas evaluativas (EXA)	70 %

Si, por algún motivo, no se pudiera evaluar uno de los dos apartados anteriores (**TRA** o **EXA**), el otro apartado restante soportaría el 100 % de la carga evaluativa, de forma que la calificación final resultaría únicamente de dicho apartado.

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.

4.2. Calificación

Se llevarán a cabo trabajos, actividades y/o ejercicios (apartado **TRA**) versados 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 (apartado **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, no es posible evaluar el segundo trimestre por separado del primero, por lo que los exámenes evaluarán todos los contenidos trabajados hasta el momento desde el comienzo del curso, de forma que:

- **En la primera evaluación**, el examen evaluará los contenidos trabajados en el primer trimestre.
- **En la segunda evaluación**, el examen evaluará los contenidos trabajados en el segundo trimestre pero pudiendo incluir aspectos del primer trimestre.

4.2.1. Calificaciones parciales

La **calificación de cada evaluación parcial** (primer y segundo trimestres por separado) se calculará de la siguiente forma:

Algoritmo 1 (Cálculo de la calificación parcial)

```

if (mínimo(TRA, EXA) ≥ 4) {
    NOTA = TRA * 0.3 + EXA * 0.7;
} else {
    NOTA = mínimo(TRA, EXA);
}
  
```

Donde:

TRA: Media aritmética de las calificaciones de los trabajos realizados en ese trimestre, valorados del 0 al 10.

EXA: Calificación del examen correspondiente a ese trimestre, valorada del 0 al 10.

La evaluación parcial se considera aprobada si **NOTA** ≥ 4,5.

Si durante el trimestre en cuestión no se realizaran trabajos, actividades y/o ejercicios (apartado **TRA**) dignos de calificación, entonces los porcentajes se redistribuirán de forma que la calificación de la evaluación parcial correspondiente resultará ser la nota del examen (apartado **EXA**), por lo que quedará simplemente de la siguiente forma:

$$\text{NOTA} = \text{EXA}$$

Asimismo, si durante el trimestre en cuestión no se realizaran exámenes (apartado **EXA**), entonces los porcentajes se redistribuirán de forma que la calificación de la evaluación parcial correspondiente resultará ser la nota de los trabajos, actividades y/o ejercicios (apartado **TRA**), por lo que quedará simplemente de la siguiente forma:

$$\text{NOTA} = \text{TRA}$$

Las **faltas de ortografía** en los exámenes serán **penalizadas** según acuerdo del Departamento, de la siguiente forma:

Número de faltas	Evaluación 1	Evaluación 2
< 5	−0, 25 puntos	−0, 50 puntos
≥ 5	−0, 50 puntos	−1, 00 puntos

En ningún caso este criterio podrá ser motivo de que el alumno no supere la prueba escrita.

4.2.2. Calificación final

Para superar el módulo será necesario obtener una calificación mínima en cada evaluación parcial, según se establece en el siguiente algoritmo, donde se determina cómo se calcula la calificación final del módulo:

Algoritmo 2 (Cálculo de la calificación final)

```

if (mínimo(EV1, EV2) ≥ 4) {
    NOTA = EV1 * 0.5 + EV2 * 0.5;
} else {
    NOTA = mínimo(EV1, EV2);
}

```

Donde:

EV₁: Calificación de la primera evaluación.

EV₂: Calificación de la segunda evaluación.

El módulo se considera aprobado si **NOTA** ≥ 4,5.

4.3. Medidas de recuperación y mejora de las calificaciones

4.3.1. Recuperación del apartado TRA

A lo largo del curso se abrirán nuevas ventanas temporales para que el alumnado entregue las correcciones necesarias en aquellas actividades que estén pendientes de evaluación positiva. La evaluación de tales correcciones podrá requerir, a criterio y discreción del profesor del módulo, la defensa de las mismas en una entrevista individual con el alumno para garantizar la autoría y la adquisición adecuada de las competencias correspondientes.

4.3.2. Recuperación del apartado EXA

Al final de cada evaluación, o a comienzos de la evaluación siguiente, se llevará a cabo una prueba de recuperación para aquellos alumnos que tengan algún contenido pendiente en esa evaluación. Al final del curso se hará una prueba final de recuperación para aquellos alumnos que tengan pendientes contenidos de alguna evaluación, debiendo presentarse únicamente a aquellas partes que tengan pendientes. La evaluación de tales pruebas de recuperación podrá requerir, a criterio y discreción del profesor del módulo, la defensa de las mismas en una entrevista individual con el alumno para garantizar la autoría y la adquisición adecuada de las competencias correspondientes.

4.3.3. Mejora de las calificaciones

El alumno con contenido superado que quieran mejorar sus calificaciones («subir nota») podrá, si así lo desea, someterse al protocolo de recuperación y realizar de nuevo las actividades del apartado **TRA** que desee, o bien las pruebas de recuperación del apartado **EXA** que desee (o bien ambas cosas) pero, en tal caso, **es consciente, entiende, asume y acepta que la calificación que finalmente constará en dicha actividad o prueba será la alcanzada en ese último intento, por lo que corre el riesgo de acabar con menos nota que antes.**

5. 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*).

5.1. Cuadro resumen

Unidad didáctica	Inicio estimado
1. Introducción a la tecnología web <i>#ev1</i>	2020-09-21
2. Conceptos básicos de PHP (I) <i>#ev1</i>	2020-09-28
3. Conceptos básicos de PHP (II) <i>#ev1</i>	2020-10-05
4. Desarrollo de aplicaciones con PHP (I) <i>#ev1</i>	2020-10-12
5. Persistencia de datos con PHP <i>#ev1</i>	2020-10-19
6. Desarrollo de aplicaciones con PHP (II) <i>#ev1</i>	2020-10-26
7. Programación avanzada en PHP <i>#ev1</i>	2020-11-02
8. Interoperabilidad <i>#ev1</i>	2020-11-09
9. Introducción a Yii <i>#ev1</i>	2020-11-16
10. Estructura de una aplicación Yii <i>#ev1</i>	2020-11-23
11. Gestión de peticiones en Yii <i>#ev2</i>	2021-01-11
12. Acceso a bases de datos en Yii <i>#ev2</i>	2021-01-18
13. Creación y validación de formularios en Yii <i>#ev2</i>	2021-01-25
14. Visualización de datos en Yii <i>#ev2</i>	2021-02-01
15. Seguridad y cacheado en Yii <i>#ev2</i>	2021-02-08

Unidad didáctica	Inicio estimado
16. Características adicionales de Yii #ev2	2021-02-15
17. Calidad #ev2	2021-02-22
18. Computación en la nube #ev2	2021-03-02
19. Servicios web con REST en Yii #ev2	2021-03-09
20. Contenedores #ev2 #opcional	2021-03-16

5.2. 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: 2020-09-21)

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

- Odoo
- 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

1.5. Apache básico *#opcional*

1.5.1. Instalación

1.5.2. Configuración básica

1.5.3. Sitios virtuales

2 CONCEPTOS BÁSICOS DE PHP (I) *#ce2c #ce2d #ce2e #ce2f #ce2g #ce2h #ce4g #ev1 #ra2 #ra3 #ra4* (est: 2020-09-28)

2.1. Introducción a PHP

2.1.1. Página web de PHP

2.1.2. Instalación de PHP

2.1.3. Documentación y búsqueda de información

2.1.4. 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 formateado numérico
 - `number_format()`
 - `money_format()`
 - `setlocale()`

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 #ce3a #ce3b #ce3c #ce3d #ce3g #ce4g #ev1 #ra2 #ra3 #ra4 (est: 2020-10-05)

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 #ce5d #ce5g #ce5h #ev1 #ra1 #ra2 #ra3 #ra4 #ra5* (est: 2020-10-12)

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
- Etiqueta <?=
- Servidor web interno

4.1.3. CGI: PHP-FPM (FastCGI Process Manager)

4.1.4. Configuración básica con php.ini

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

4.1.5. Módulos de extensión

4.2. Manejo de datos de entrada: \$_GET y \$_POST

4.3. Funciones auxiliares interesantes

4.3.1. extract()

4.3.2. compact()

4.4. Aspectos básicos de la orientación a objetos

4.4.1. Objetos

- new
- instanceof

4.4.2. Referencias

- Asignación por referencia (= &)

4.4.3. Clonación de objetos

4.4.4. Comparación de objetos

4.4.5. Propiedades

- Predeterminadas
- Dinámicas

4.4.6. Métodos

4.4.7. Constantes

- Operador de resolución de ámbito (::)

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

4.5. Excepciones

4.5.1. Manejo de errores clásico en PHP

4.5.2. Errores vs. excepciones

4.5.3. La clase Exception

4.5.4. La clase Error

4.5.5. La clase RuntimeException

4.5.6. Estructura de control try ... catch

4.6. Depuración

4.6.1. var_dump(), print_r(), die()

4.6.2. Xdebug

4.6.3. 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 #ce6g #ev1 #ra2 #ra3 #ra4 #ra5 #ra6* (est: 2020-10-19)

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. Transacciones

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

5.2. Cookies

5.2.1. `setcookie()`

5.2.2. Ejemplos de uso

5.3. Sesiones

5.3.1. Iniciar una sesión

- `session_start()`

5.3.2. Usar una sesión

- `$_SESSION`
- Ejemplos de uso

5.3.3. Terminar una sesión

- `session_destroy()`
- `session_name()`
- `session_id()`
- `session_get_cookie_params()`

5.4. Seguridad y persistencia

5.4.1. Contraseñas

- <https://www.md5online.org/>
- <https://www.sha1online.org/>
- `password_hash()`
- `password_verify()`

5.4.2. Inyección de código SQL

5.4.3. Cross-Site Request Forgery (CSRF)

6 DESARROLLO DE APLICACIONES CON PHP (II) (est: 2020-10-26)

6.1. Programación de *CRUD* básico

6.1.1. Ejemplo de aplicación: *Muéveme*

6.1.2. Ejemplo de aplicación: *FilmAffinity*

6.2. Post/Redirect/Get

6.3. header()

6.3.1. output_buffering

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
 - htmlspecialchars()
 - HTML Purifier

6.4.3. Filtrado de la entrada

- Cómo NO se debe hacer
- Extensión Filter
 - filter_input(), filter_has_var(), filter_var()
 - Filtros de validación y saneado
- Expresiones regulares (PCRE)

7 PROGRAMACIÓN AVANZADA EN PHP #ce4g #ce5g #ce5h #ev1 #ra2 #ra3 #ra4 #ra5 (est: 2020-11-02)

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. La clase stdClass

- Conversión de array a object

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 #ce1g #ce4g #ce5g #ce5h #ce9a #ce9b #ce9e #ev1 #ra1 #ra4 #ra5 #ra9 (est: 2020-11-09)

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

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

8.5. 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. Paquetes de Atom y herramientas externas *#opcional*

8.6.1. PHP_CodeSniffer

8.6.2. PHP-CS-Fixer

8.6.3. Yii2-Shell

9 INTRODUCCIÓN A YII *#ce1e #ce1g #ce4g #ce5a #ce5b #ce5c #ce5d #ce5e #ce5g #ce5h #ce6a #ce6b #ce6c #ce6d #ce6e #ce6f #ce6g #ce6h #ce9b #ce9c #ce9e #ce9f #ce9g #ev1 #ra1 #ra4 #ra5 #ra6 #ra9* (est: 2020-11-16)

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. Yii 2

9.3.1. ¿Qué es Yii?

9.3.2. ¿En qué es mejor Yii?

9.3.3. ¿Cómo es Yii comparado con otros frameworks?

9.3.4. Versiones de Yii

9.4. Instalación, requisitos y puesta en marcha

9.4.1. Requisitos previos

9.4.2. Instalación de Yii 2

- Instalación mediante Composer

9.4.3. Plantillas de proyecto

- Plantilla básica vs. avanzada
- Plantilla básica modificada

9.5. ¡Hola, mundo!

9.6. Formularios

9.7. Bases de datos

9.8. Generador de código Gii

9.9. Herramientas de desarrollo

9.9.1. Barra de depuración

9.9.2. Trazas de depuración

9.10. Estilo del código

10 ESTRUCTURA DE UNA APLICACIÓN YII *#ce4g #ce5g #ce5h #ce9e #ce9f #ce9g #ev1 #ra2 #ra3 #ra4 #ra5 #ra9* (est: 2020-11-23)

10.1. A pequeña escala

10.1.1. Componentes

- La clase yii
 - base
 - BaseObject
 - Propiedades
 - Configuraciones
 - Asignación masiva
 - Creación de nuevas instancias
 - Normas de creación de componentes
 - Diferencias entre new y Yii::createObject()
- La clase yii
 - base
 - Component
 - Eventos
 - De instancia
 - Eventos de instancia
 - Manejadores de eventos de instancia

- De clase
 - Eventos de clase
 - Manejadores de eventos de clase
- Comportamientos

10.1.2. Alias

10.1.3. Autoloading de clases

10.1.4. Localizador de servicios

10.1.5. Contenedor de inyección de dependencias

10.2. A gran escala

10.2.1. Introducción

10.2.2. Scripts de entrada

10.2.3. Aplicaciones

10.2.4. Componentes de aplicación

10.2.5. Controladores

- Acciones
- Filtros

10.2.6. Modelos

10.2.7. Vistas

- Widgets

10.2.8. Otros componentes

- Módulos
- Assets
- Extensiones

11 GESTIÓN DE PETICIONES EN YII #ce4g #ce5d #ce5f #ce5g #ce5h #ev2 #ra4 #ra5 (est: 2021-01-11)

11.1. Introducción

11.2. Arranque (bootstrapping)

11.3. Enrutado y creación de URLs

11.4. Peticiones

11.5. Respuestas

11.6. Sesiones y cookies

12 ACCESO A BASES DE DATOS EN YII #ce4g #ce5f #ce5g #ce5h #ce6a #ce6b #ce6c #ce6d #ce6e #ce6f #ce6g #ce6h #ce9e #ce9f #ce9g #ev2 #ra2 #ra3 #ra4 #ra5 #ra6 #ra9 (est: 2021-01-18)

12.1. DAO

12.1.1. yii

db

Connection

12.1.2. yii

db

Connection::createCommand()

12.1.3. Consultas SQL

- queryAll()
- queryOne()
- queryColumn()
- queryScalar()

12.1.4. Sentencias no SELECT

- execute()
- insert()
- update()
- delete()

12.2. Query Builder

12.2.1. yii

db

Query

12.2.2. Creación de consultas

- select()
- from()
- Condiciones y filtrado de filas
 - where()
 - Formatos de condiciones
 - De cadena
 - De array
 - De operadores
 - andWhere()
 - orWhere()
 - filterWhere()
 - andFilterWhere()
 - orFilterWhere()
- orderBy()
- groupBy()
- Condiciones y filtrado de grupos
 - having()

- filterHaving()
- andFilterHaving()
- orFilterHaving()
- limit()
- offset()
- Combinaciones
 - join()
 - innerJoin()
 - leftJoin()
 - rightJoin()
- union()

12.2.3. Recogida de resultados

- all()
- one()
- column()
- scalar()
- exists()
- count()
- Funciones de grupo
 - sum()
 - average()
 - max()
 - min()
- indexBy()

12.2.4. Consultas por lotes

- batch()
- each()

12.3. Active Record

12.3.1. findOne()

12.3.2. findAll()

12.3.3. save()

12.3.4. ActiveQuery

- find()

12.3.5. Atributos sucios

12.3.6. Relaciones

- Encadenamiento de relaciones

12.3.7. Carga ansiosa y carga perezosa

12.3.8. `joinWith()`

12.3.9. Atributos virtuales

- Siete técnicas
 - Calcular a mano cuando/donde haga falta
 - Usar vistas SQL
 - Sobreescibir el método `find()` del modelo para que se use siempre en lugar del heredado de `ActiveRecord`
 - Sobreescibir el método `afterFind()` para rellenar el atributo a mano cada vez que se hace un `find()`
 - Capturar el evento `EVENT_AFTER_FIND` del modelo
 - Usar una propiedad con *getter* y *setter*
 - Crear un método `findEspecial()` que se usará en lugar de `find()` cuando haga falta
 - La mejor opción, en la mayoría de los casos: combinar las dos anteriores
 - Ejemplo

13 CREACIÓN Y VALIDACIÓN DE FORMULARIOS EN YII #ce3e #ce3f #ce4g #ce5b #ce5d #ce5h #ce6f #ce6h #ce8a #ce8b #ce8c #ce8d #ce8e #ce8f #ce8g #ce9e #ce9f #ce9g #ev2 #ra2 #ra3 #ra4 #ra5 #ra6 #ra8 #ra9 (est: 2021-01-25)

13.1. Creación de formularios

13.1.1. ActiveForm

13.1.2. yii

helpers

Html

13.2. Validación de la entrada

13.2.1. Declaración de reglas

- Validadores principales
- `skipOnEmpty`
- `skipOnError`
- Personalizar mensajes de error
- Validación condicional
- Filtrado (saneado) de datos
- Manejo de entradas vacías

13.2.2. Validadores en línea

13.2.3. Validaciones en el cliente #opcional

13.3. Subida de archivos

14 **VISUALIZACIÓN DE DATOS EN YII** *#ce4g #ce5b #ce5c #ce5d #ce5h #ce6f #ce6h #ce8a #ce8b #ce8c #ce8d #ce8e #ce8f #ce8g #ce9e #ce9f #ce9g #ev2 #ra2 #ra3 #ra4 #ra5 #ra6 #ra8 #ra9* (est: 2021-02-01)

14.1. Formateado de datos

14.1.1. yii
i18n
Formatter

14.2. Paginación

14.2.1. yii
data
Pagination

14.2.2. Entrada

- totalCount
- pageSize
- page

14.2.3. Salida

- limit
- offset

14.2.4. yii
widgets
LinkPager

14.3. Ordenación

14.3.1. yii
data
Sort

14.3.2. Entrada

- attributes
- sort

14.3.3. Salida

- orders

14.3.4. yii
data
Sort::link()

14.4. Proveedores de datos

14.4.1. Entrada

- pagination

- sort

14.4.2. Salida

- models
- count
- totalCount

14.4.3. ActiveDataProvider

- query

14.4.4. SqlDataProvider

- sql
- params
- totalCount

14.4.5. ArrayDataProvider

- allModels

14.5. Widgets de datos

14.5.1. DetailView

14.5.2. ListView

14.5.3. GridView

- dataProvider
- columns
 - DataColumn
 - SerialColumn
 - ActionColumn
- Ordenación de columnas
- Filtrado de datos
 - yii
 - grid
 - GridView::\$filterModel
- Relaciones
 - Ordenación con relaciones
 - Filtrado con relaciones
 - yii
 - db
 - ActiveRecord::getAttribute(\$name)

14.5.4. El problema de las fechas/horas/instantes

14.5.5. Otros widgets no oficiales

- Krajee Yii Extensions
 - yii2-datecontrol

– yii2-number

14.6. Scripts de cliente

14.7. Trabajar con ActiveForm desde el cliente

14.8. Temas *#opcional*

15 SEGURIDAD Y CACHEADO EN YII *#ce4d #ce4e #ce4g #ce5e #ce5g #ce5h #ce6h #ce9e #ce9f #ce9g #ev2 #ra2 #ra3 #ra4 #ra5 #ra6 #ra8 #ra9* (est: 2021-02-08)

15.1. Autenticación

15.1.1. Componente de aplicación user

15.1.2. Clase identidad e interfaz yii

web

IdentityInterface

15.1.3. Métodos de *login* y *logout*.

15.2. Contraseñas

15.3. Autorización

15.4. Niveles de caché

15.4.1. Cacheado de datos

15.4.2. Cacheado de fragmentos

15.4.3. Cacheado de páginas

15.4.4. Cacheado HTTP

16 CARACTERÍSTICAS ADICIONALES DE YII *#ce4g #ce5d #ce5g #ce5h #ce6h #ce8a #ce8b #ce8c #ce8d #ce8e #ce8f #ce8g #ce9e #ce9f #ce9g #ev2 #ra2 #ra3 #ra4 #ra5 #ra6 #ra8 #ra9* (est: 2021-02-15)

16.1. AJAX y PJAX

16.1.1. Validaciones Ajax

16.1.2. PJAX *#opcional*

16.1.3. CORS *#opcional*

16.2. Correo electrónico

16.3. Aplicación de consola

16.4. Migraciones

16.5. Extensiones

16.6. Paquetes

17 CALIDAD *#ce3g #ce4g #ce5g #ce5h #ce6e #ce6h #ce9e #ce9g #ev2 #ra3 #ra4 #ra5 #ra6 #ra9* (est: 2021-02-22)

17.1. Pruebas

17.1.1. Tipos de pruebas

- Unitarias
- Funcionales
- De aceptación

17.1.2. Herramientas

- PHPUnit *#opcional*
- Codeception
 - Ejecutar pruebas
 - Crear pruebas en formato Cest
- Fixtures
 - `./yii fixture/generate <nombre>`
 - `./yii fixture/load <nombre>`

17.1.3. Integración continua: Travis CI

17.1.4. Cobertura de código *#opcional*

17.2. Depuración

17.2.1. `var_dump()` mejorado

17.2.2. Consola integrada

17.2.3. Barra de depuración

17.2.4. Depuración con PsySH *#opcional*

17.3. Documentación

17.3.1. API documentation generator for Yii2

17.3.2. GitHub Pages

17.4. Mantenimiento y calidad del código

17.4.1. CodeSniffer

17.4.2. CS_Fixer

17.4.3. Code Climate

18 COMPUTACIÓN EN LA NUBE *#ce1a #ce1c #ce1d #ce1e #ce5h #ce6h #ce9e #ce9f #ce9g #ev2 #ra1 #ra5 #ra6 #ra9* (est: 2021-03-02)

18.1. Entornos de ejecución

18.1.1. Desarrollo

18.1.2. Producción

18.1.3. Pruebas

- 18.1.4. Preproducción
- 18.2. Cloud computing vs hosting
- 18.3. Cloud computing vs VPS
- 18.4. Servicios por capas
 - 18.4.1. IaaS
 - 18.4.2. PaaS
 - 18.4.3. SaaS
- 18.5. 12 Factores
- 18.6. Heroku
 - 18.6.1. Heroku CLI
 - 18.6.2. Creación y despliegue de aplicaciones
 - 18.6.3. Heroku Postgres
 - 18.6.4. Variables de entorno
 - 18.6.5. Releases
- 18.7. Escalabilidad *#opcional*
- 18.8. Alta disponibilidad *#opcional*

19 SERVICIOS WEB CON REST EN YII *#ce1a #ce1b #ce7a #ce7b #ce7c #ce7d #ce7e #ce7f #ce7g #ce8b #ce8f #ce8g #ce9c #ce9e #ce9f #ce9g #ev2 #ra1 #ra2 #ra3 #ra7 #ra8 #ra9* (est: 2021-03-09)

- 19.1. Introducción
- 19.2. Recursos
- 19.3. Controladores
- 19.4. Enrutado
- 19.5. Formateo de la respuesta
- 19.6. Autenticación
- 19.7. Limitación de frecuencia de peticiones
- 19.8. Versionado
- 19.9. Gestión de errores

20 CONTENEDORES *#ce1a #ce1c #ce1d #ce1e #ce9e #ce9f #ce9g #ev2 #opcional #ra1 #ra9* (est: 2021-03-16)

- 20.1. Vagrant
 - 20.1.1. PuPHPet
- 20.2. Docker
 - 20.2.1. Docker Hub
 - 20.2.2. Dockerfiles

20.2.3. Docker Compose

- docker-compose.yml

20.3. PHPDocker

5.3. 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)		#ce2c #ce2d #ce2e #ce2f #ce2g #ce2h	×	#ce4g					
3. Conceptos básicos de PHP (II)		#ce2d #ce2e #ce2g	#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	#ce5d #ce5g #ce5h				
5. Persistencia de datos con PHP		×	×	#ce4a #ce4b #ce4c #ce4g	#ce5f #ce5g #ce5h	#ce6a #ce6b #ce6c #ce6e #ce6g			

Unidades didácticas	#ra1	#ra2	#ra3	#ra4	#ra5	#ra6	#ra7	#ra8	#ra9
6. Desarrollo de aplicaciones con PHP (II)	#ce1a			#ce4a #ce4b #ce4c #ce4d #ce4e #ce4f #ce4g	#ce5a #ce5b #ce5d #ce5f #ce5g #ce5h	#ce6a #ce6b #ce6c #ce6d #ce6e #ce6f #ce6g #ce6h			
7. Programación avanzada en PHP		×	×	#ce4g	#ce5g #ce5h				
8. Interoperabilidad	#ce1e #ce1g			#ce4g	#ce5g #ce5h				#ce9a #ce9b #ce9e
9. Introducción a Yii	#ce1e #ce1g			#ce4g	#ce5a #ce5b #ce5c #ce5d #ce5e #ce5g #ce5h	#ce6a #ce6b #ce6c #ce6d #ce6e #ce6f #ce6g #ce6h			#ce9b #ce9c #ce9e #ce9f #ce9g
10. Estructura de una aplicación Yii		×	×	#ce4g	#ce5g #ce5h				#ce9e #ce9f #ce9g
11. Gestión de peticiones en Yii				#ce4g	#ce5d #ce5f #ce5g #ce5h				
12. Acceso a bases de datos en Yii		×	×	#ce4g	#ce5f #ce5g #ce5h	#ce6a #ce6b #ce6c #ce6d #ce6e #ce6f #ce6g #ce6h			#ce9e #ce9f #ce9g
13. Creación y validación de formularios en Yii		×	#ce3e #ce3f	#ce4g	#ce5b #ce5d #ce5h	#ce6f #ce6h		#ce8a #ce8b #ce8c #ce8d #ce8e #ce8f #ce8g	#ce9e #ce9f #ce9g

Unidades didácticas	#ra1	#ra2	#ra3	#ra4	#ra5	#ra6	#ra7	#ra8	#ra9
14. Visualización de datos en Yii		×	×	#ce4g	#ce5b #ce5c #ce5d #ce5h	#ce6f #ce6h		#ce8a #ce8b #ce8c #ce8d #ce8e #ce8f #ce8g	#ce9e #ce9f #ce9g
15. Seguridad y cacheado en Yii		×	×	#ce4d #ce4e #ce4g	#ce5e #ce5g #ce5h	#ce6h		×	#ce9e #ce9f #ce9g
16. Características adicionales de Yii		×	×	#ce4g	#ce5d #ce5g #ce5h	#ce6h		#ce8a #ce8b #ce8c #ce8d #ce8e #ce8f #ce8g	#ce9e #ce9f #ce9g
17. Calidad			#ce3g	#ce4g	#ce5g #ce5h	#ce6e #ce6h			#ce9e #ce9g
18. Computación en la nube	#ce1a #ce1c #ce1d #ce1e				#ce5h	#ce6h			#ce9e #ce9f #ce9g
19. Servicios web con REST en Yii	#ce1a #ce1b	×	×				#ce7a #ce7b #ce7c #ce7d #ce7e #ce7f #ce7g	#ce8b #ce8f #ce8g	#ce9c #ce9e #ce9f #ce9g
20. Contenedores	#ce1a #ce1c #ce1d #ce1e								#ce9e #ce9f #ce9g

6. Orientaciones pedagógicas

Según queda recogido en [1], 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.

7. 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, será 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.

8. Recursos

8.1. 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.

8.2. Software

- Sistema operativo GNU/Linux (Ubuntu o Debian GNU/Linux, preferentemente).
- El resto de herramientas y aplicaciones necesarias se instalarán a través de Internet a lo largo del curso.

8.3. 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.

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

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

- **GitHub y GitHub Classroom**³ como herramientas centralizadas para compartir código y para la gestión integral de todo elemento satélite del mismo (control de versiones, desarrollo colaborativo, incidencias, etcétera).
 - Los alumnos disponen, de forma totalmente gratuita, del **GitHub Student Developer Pack**⁴, que les ofrece servicios y herramientas con descuentos de hasta el 100 % con respecto al precio de mercado.

8.4. Bibliografía

8.4.1. Principal

- Apuntes y documentación *online*⁵ suministrados por el profesor.
- Documentación de Git⁶ y GitHub⁷.
- Documentación de PHP⁸.
- Documentación de Yii 2 Framework⁹.
- Documentación de PostgreSQL¹⁰.

8.4.2. Complementaria

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

9. 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 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*).

³ <https://classroom.github.com>

⁴ <https://education.github.com/pack>

⁵ <https://dwese.iesdonana.org>

⁶ <https://git-scm.com/book/en/v2>

⁷ <https://help.github.com>

⁸ <https://php.net>

⁹ <https://www.yiiframework.com>

¹⁰ <https://www.postgresql.org>

10. 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 [3].
- **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.

11. 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.

11.1. 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: Violencia de Género.
 - b) 30 de enero: Resolución de conflictos de forma pacífica. Día de la Paz.
 - c) 8 de marzo: Día de la Mujer Trabajadora.

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».*¹¹

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.

¹¹ <https://twitter.com/RAEinforma/status/1111565711653113856>

Referencias

- [1] *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 nº 149 del 1 de agosto)¹².
- [2] *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 nº 202 del 15 de octubre).
- [3] Instituto Nacional de Seguridad e Higiene en el Trabajo. *Instrucción básica para el trabajador usuario de pantallas de visualización de datos*.¹³
- [4] *Instrucciones de 23 de abril de 2020*, de la Viceconsejería de Educación y Deporte, relativas a las medidas educativas a adoptar en el tercer trimestre del curso 2019/2020.

¹² <http://www.juntadeandalucia.es/boja/boletines/2011/149/d/updf/d23.pdf\T1\textbackslash#page=17>

¹³ http://www.insht.es/InshtWeb/Contenidos/Documentacion/TextosOnline/Guias_Ev_Riesgos/Instruccion_Pantallas/Instruccion_basica.pdf