



Universidad de Jaén
Centro de Estudios de Postgrado

Trabajo Fin de Máster

ARQUITECTURA DE PROGRAMACIÓN WEB: BACKEND

Alumno/a: Montávez Sánchez, Matías

Tutor/a: Prof. D. Rafael Segura Sánchez
Dpto: Informática

ÍNDICE DE CONTENIDOS

RESUMEN.	4
PALABRAS CLAVE.	4
ABSTRACT.	5
KEYWORDS.	5
1. FUNDAMENTACIÓN EPISTEMOLÓGICA.	6
1.1. Introducción.	6
1.2. Antecedentes y contextualización.	7
1.3. Descripción de la programación web: front-end y backend.	15
1.4. El backend.	18
1.4.1. Definición y estado actual.	18
1.4.2. Tendencias futuras.	22
1.5. Conclusión.	26
2. DESARROLLO DE UNA UNIDAD DIDÁCTICA.	27
2.1. Introducción.	28
2.2. Ubicación de la enseñanza.	29
2.3. Normativa.	30
2.4. Contexto.	30
2.4.1. Contexto socioeconómico del centro.	31
2.4.2. Características del centro.	32
2.4.3. Características del alumnado.	33
2.5. Objetivos.	33
2.5.1. Competencia general.	34
2.5.2. Objetivos generales del título.	34
2.5.3. Objetivos generales del módulo.	36
2.5.4. Competencias profesionales, personales y sociales del módulo.	37
2.5.5. Líneas de actuación del módulo.	38
2.5.6. Orientaciones pedagógicas del módulo.	39
2.5.7. Objetivos específicos de la unidad didáctica.	39

2.5.8.	Objetivos didácticos.....	40
2.6.	Resultados de aprendizaje.	40
2.7.	Contenidos.	41
2.7.1.	Contenidos generales del módulo.	41
2.7.2.	Contenidos específicos de la unidad didáctica.	44
2.7.3.	Interdisciplinariedad.	45
2.7.4.	Temas transversales.	45
2.8.	Metodología.	46
2.8.1.	Actividades de enseñanza-aprendizaje.	47
2.8.2.	Recursos.	49
2.8.3.	Espacio.	49
2.8.4.	Bibliografía.	50
2.9.	Evaluación.	51
2.9.1.	Criterios de evaluación.	51
2.9.2.	Instrumentos de evaluación.	52
2.9.3.	Recuperación.	52
2.9.4.	Evaluación del proceso de enseñanza.	52
2.10.	Atención a la diversidad.	53
2.10.1.	Ritmos de aprendizaje.	53
2.10.2.	ACNEAE.	53
2.10.3.	Actividades de refuerzo y ampliación.....	53
2.11.	Tabla resumen.	55
3.	BIBLIOGRAFÍA.....	57
	ANEXO I. Evaluación del proceso de enseñanza.	60

ÍNDICE DE ILUSTRACIONES

Ilustración 1: esquema web estática.	7
Ilustración 2: esquema web dinámica.	8
Ilustración 3: esquema programación cliente/servidor.	9
Ilustración 4: uso de lenguajes backend.....	10
Ilustración 5: versiones de PHP.....	13
Ilustración 6: evolución de PHP.	14
Ilustración 7: arquitectura de Angular.....	16
Ilustración 8: XML vs JSON.....	17
Ilustración 9: evolución de uso de Python.....	23
Ilustración 10: evolución de Docker.	24
Ilustración 11: futuro del lenguaje GO.	25
Ilustración 12: disposición del aula.....	50

RESUMEN.

Con la expansión de Internet de las últimas décadas y en lo referente al uso de la web, se ha pasado de ver una página web como una forma de presentar la información de forma estática dónde el usuario únicamente se limitaba a leer o visualizar la información representada a partir de un conjunto de código HTML inamovible, a su evolución a una web dinámica, donde ya nos encontramos elementos interactivos. El usuario puede interactuar con los elementos presentados, una web no es mero código HTML, se necesitan de nuevas tecnologías, o dicho de otra forma, una nueva forma de programar una página web, una programación front-end que se encargará de la parte de cliente, es decir lo que se ejecuta en el navegador y una programación backend para el lado del servidor que se encargará de las tareas más pesadas y gestionará los recursos de forma transparente al usuario final, a partir de dicho procesamiento presentará la información al cliente a través del navegador, es en esta parte en la que nos vamos a centrar en este texto.

PALABRAS CLAVE.

Programación web – Programación servidor – Backend – PHP – Servicios web

ABSTRACT.

With the expansion of the Internet in recent decades and in relation to the use of the web, it has gone from seeing a web page as a way of presenting information in a static way where the user was only limited to reading or viewing the information represented from a set of immovable HTML code, to its evolution to a dynamic web, where we already find interactive elements and the user can interact with the elements presented, a web is not only HTML code, new technologies are needed, or said In other words, a new way of programming a web page, a front-end programming that will take care of the client part, that is, what is executed in the browser, and a backend programming for the server side that will take care of the heavier tasks and will manage the resources in a transparent way to the end user, from said processing it will present the information to the client through the browser, it is in this part that we are going to focus on this text.

KEYWORDS.

Web programming – Server programming – Backend – PHP – Web services

1. FUNDAMENTACIÓN EPISTEMOLÓGICA.

A continuación, en los siguientes apartados se va a hacer un estudio del estado del arte de la programación web en general y de la programación web del servidor (backend) en particular.

Para ello vamos a ver los antecedentes e inicios de dicha programación, comparando los distintos lenguajes, viendo sus pros y sus contras para ver el estado actual de los mismos.

Por último, vamos a analizar las tendencias futuras, hasta donde vamos y hasta dónde podemos llegar con la evolución de esta tecnología.

1.1. Introducción.

A lo largo de las siguientes páginas vamos a analizar detenidamente los aspectos más relevantes de la programación en el lado servidor de una aplicación web centrándonos en:

- Antecedentes: se verán brevemente las tecnologías de la programación en el backend.
- Contextualización: se intentará analizar el porqué del éxito de PHP respecto a otras plataformas, sus distintas versiones, sus ventajas e inconvenientes. Para ello se van a analizar los lenguajes PHP, Java, JavaScript, Python y Ruby haciendo una comparativa de los mismos.
- Descripción de la programación del backend: se van a revisar los elementos más importantes de la programación en el servidor.
- Tendencias actuales y futuras: se hará un estudio de cómo ha evolucionado la programación backend y su estado actual, así como la previsión de futuro de la misma. Igualmente se verán distintos frameworks de trabajo haciendo una comparativa entre los mismos al igual que se realizó con los lenguajes de programación.
- Conclusión: en base a todo lo anterior se reflexionará de los pros y los contras de la programación web, de los lenguajes utilizados y de los frameworks en general.

1.2. Antecedentes y contextualización.

La programación web no surgió como la conocemos hoy en día, las primeras webs que aparecieron siempre mostraban la misma información, se almacenaban tal cual se creaban y su contenido nunca se modificaba a menos que el programador o administrador modificara manualmente el código HTML de la misma, es lo que se conoce como página web estática.

Como es obvio con la evolución de internet este tipo de webs tenían que evolucionar, el usuario demandaba unas prestaciones, un tipo de servicios que una página web estática no podía darles, de ahí que nacieran las páginas webs dinámicas.

A continuación, se van a describir las características de cada una de ellas (Jaime González, 2015).

Web estática.

En una página web estática siempre que se carguen va a mostrar la misma información por lo que son útiles para mostrar una información concreta que nunca va a cambiar.

El funcionamiento de la misma sigue el esquema de la Ilustración 1:

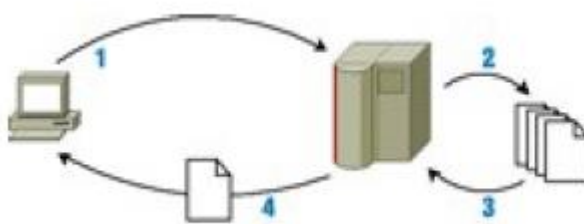


Ilustración 1: esquema web estática.

1. El cliente desde un navegador solicita una página web, dicha petición va a un servidor.
2. El servidor busca la página en su banco de páginas.
3. Si la encuentra la recupera.
4. El servidor manda la página al navegador para mostrarla al cliente.

Web dinámica.

En contraposición a las webs estáticas tenemos las webs dinámicas dónde el contenido de la misma puede ir variando dependiendo de la petición que se realice por parte del cliente.

El funcionamiento de la misma sigue el esquema de la ilustración 2:

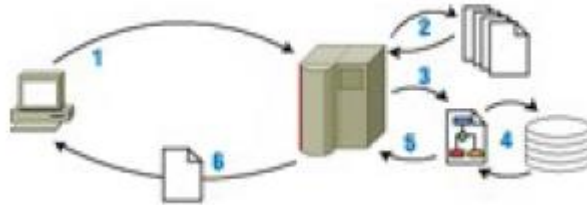


Ilustración 2: esquema web dinámica.

1. El cliente desde un navegador solicita una página web, dicha petición va a un servidor.
2. El servidor busca la página en su banco de páginas.
3. Como parte del proceso puede que se necesite ejecutar código en el servidor para crear la página web, por lo que se ejecutaría dicho código.
4. También suele ser normal que se acceda a una o varias bases de datos para crear la página web que se va a mostrar.
5. Una vez localizada la página, ejecutado el código en el servidor con el acceso correspondiente a las bases de datos si fuera necesario, ya tenemos la página web lista.
6. Mandamos la web al cliente para que la visualice en el navegador.

A continuación, y antes de introducirnos de lleno en los pormenores de un lenguaje de programación web específico, y ya que conocemos la diferencia entre web estática y dinámica vamos a centrarnos en la segunda, para ello es conveniente analizar en qué consiste la programación front-end y backend, nos vamos a ayudar de la ilustración 3 (Alegasa, 2016), y a partir de ella se va a ver cómo funciona una web dinámica (independientemente de los lenguajes usados).

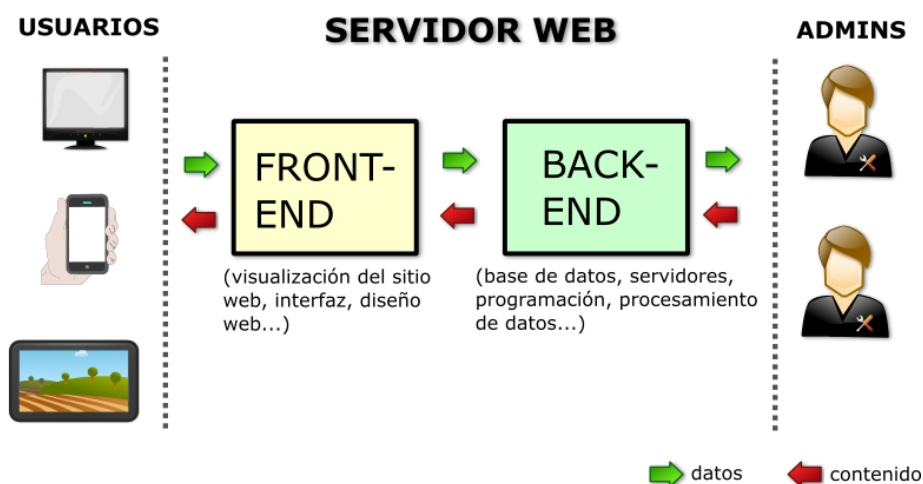


Ilustración 3: esquema programación cliente/servidor.

Como ejemplo veamos un flujo de trabajo de un usuario que se registra en una tienda online, por ejemplo.

- **Front-end:** una programación en esta parte se encarga de programar la visualización de la web, el formulario que rellenará el usuario y se encargará de enviar los datos que haya introducido el usuario (flecha verde) al servidor.
- **Backend:** en esta parte de la web se interactúan con las BB.DD. por ejemplo, por lo que una vez recibidos los datos del formulario que rellenó el usuario, estos se validan, se introducen en la BD pertinente y se manda una respuesta, dicha respuesta será la creación o modificación de la web original en base a la operación que se haya realizado con los datos, por ejemplo, si se han introducido con un mensaje "registro realizado" (flecha roja).

De esta forma se separa la parte usuario de la parte de servidor.

A lo largo de la historia de la programación web han surgido numerosos lenguajes de programación, con mayor o menor éxito, si nos centramos en los lenguajes que se usan en la programación del lado servidor aún a día de hoy PHP es el más usado, o el que más se mantiene, casi el 80% de las webs usan PHP para programar su backend, o están programadas en PHP según muestra la Ilustración 4 (W3techs, 2022) donde se ve el uso de lenguajes de programación del servidor actualizado a 25/01/2022.

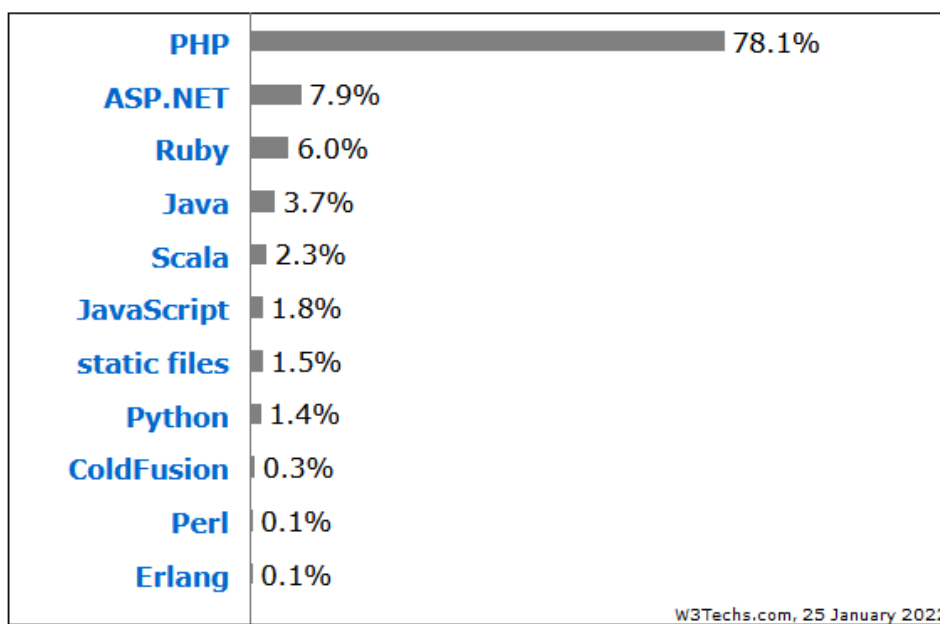


Ilustración 4: uso de lenguajes backend.

Vamos ahora a hacer un breve resumen de los lenguajes de programación web en el lado backend más populares.

Java

Otro lenguaje que se debe mencionar es Java que, aunque se conoce más como un lenguaje de programación orientado a objetos para el desarrollo de aplicaciones multiplataforma, también se puede utilizar como lenguaje en el lado servidor de una aplicación web.

Algunas de las características más relevantes son (Universidad de Alicante, 2012):

- Basado en la arquitectura Java EE, que es una plataforma orientada a la programación de aplicaciones usando Java. Dicha plataforma contiene gran número de librerías para la programación modular.
- Por sus especificaciones solo es efectiva para aplicaciones de tamaño mediano o grande debido a su complejidad en la programación.
- Se apoya en tecnologías como la creación de páginas JSP y servlets, tecnologías orientadas a la creación de forma dinámica de páginas webs.

El motivo principal que al principio hizo que Java no fuera el lenguaje más adecuado para la programación del backend era básicamente las brechas de seguridad que presentaba respecto a otros lenguajes como PHP, las aplicaciones escritas en Java eran mucho menos robustas y más vulnerables que sus análogas en otros lenguajes de programación (Waden & Doyle, 2010).

Estos problemas iniciales están actualmente solventados (Tokio School , 2020) presentando Java beneficios tales como la robustez en sus programas y la posibilidad de instalarse en cualquier plataforma.

ASP.NET

Las principales características de la programación de ASP.NET (Guérin, 2017) se podrían resumir en las siguientes:

- ASP.NET en si no es un lenguaje de programación ni un servidor web, es un marco de trabajo o framework que ha heredado todas las características del lenguaje .NET.
- Usa lenguajes como C# o VisualBasic habitualmente.
- Las aplicaciones se desarrollan en el entorno de programación Visual Studio.
- Un programa .NET funciona de forma similar a Java, se compilan dos veces, una vez en la distribución y otra en la ejecución (Arias, 2015).
- Al compilarse un programa .NET, independientemente del lenguaje utilizado se crea un código intermedio conocido como MSIL. (Arias, 2015).

ASP.NET nació para tener éxito, pero no llegó a su objetivo de mantenerse como el lenguaje de programación por excelencia en el servidor. Alguno de los motivos que hacen que ASP.NET no consiga la aceptación son los siguientes (Cedeño, 2020):

- IDE específico: al contrario que otros lenguajes pueden desarrollarse en la mayor parte de los IDE del mercado ASP.NET necesita de Visual Studio para su implementación.
- No es multiplataforma: funciona solamente bajo la plataforma Windows por lo que los proyectos no se pueden exportar a otras plataformas, en la actualidad que una aplicación sea o no multiplataforma forma parte de su éxito o fracaso en el mercado.
- Consumo de recursos: ASP.NET necesita más recursos en cuanto a RAM y CPU que otros lenguajes análogos.

Python

Python es uno de los lenguajes de programación que actualmente está experimentando mayor crecimiento en cuanto a su aceptación por parte de los programadores, y esto se ve reflejado también en su vertiente web, la popularidad de este lenguaje actualmente supera a la de sus competidores más cercanos (Rana, 2021).

Pero, ¿qué hace que este lenguaje de programación sea tan popular actualmente? Los principales motivos son los siguientes (Molina Ríos , Loja Mora , Zea Ordóñez, & Loaiza Sojos , 2016):

- Es un lenguaje con una sintaxis simple, limpia y muy intuitiva, a pesar de ello soporta programación de alto nivel, programación orientada a objetos e incluso incluye numerosas librerías para añadir funcionalidades.
- No requiere una compilación intermedia como otros lenguajes en el desarrollo de scripting, por lo que tiene una alta velocidad de procesamiento.
- Al igual que Java es multiplataforma.
- Tiene una gran comunidad de desarrolladores que lo apoyan, mejoran y hacen crecer continuamente.

Por todo esto Python es uno de los lenguajes en mayor crecimiento y que se convertirá en líder del mercado en programación web en un corto/medio plazo.

Ruby

En cuanto a Ruby (Flanagan & Matsumoto, 2008) es un lenguaje de programación orientado a la creación de webs dinámicas. En cuanto a sus características principales podemos destacar:

- La gramática del lenguaje es más compleja que otros lenguajes análogos aun así es similar a la que usa Java por ejemplo al ser un lenguaje totalmente orientado a objetos.
- Se basa en la programación funcional ya que hereda ideas de lenguajes como Lisp.
- Dispone de una amplia gama de librerías para completar las funciones básicas que ofrece de base.

En base a esto las ventajas y desventajas de Ruby respecto a otros lenguajes y tecnologías se podrían resumir en (Cedeño, 2020):

- La comunidad y en general el conocimiento que se tiene sobre este lenguaje es mucho menor que el que se tiene de otros como pueden ser PHP o Python.
- No existen una gran variedad de frameworks que permitan trabajar de forma más cómoda con este lenguaje, lo que hace que tenga menos éxito entre los programadores que no pueden automatizar partes del proceso de generación de código.
- En cambio, Ruby es un código totalmente libre y con sintaxis similar a otros lenguajes lo que hace que su uso sea más asequible.

PHP

Vamos a centrarnos ahora en el lenguaje más utilizado según hemos visto en la Ilustración, concretamente PHP y en su historia, vamos a hacer un breve repaso sobre las versiones de PHP que se han ido sucediendo a lo largo de los años como se ve en la Ilustración 5 (Evans, 2021):

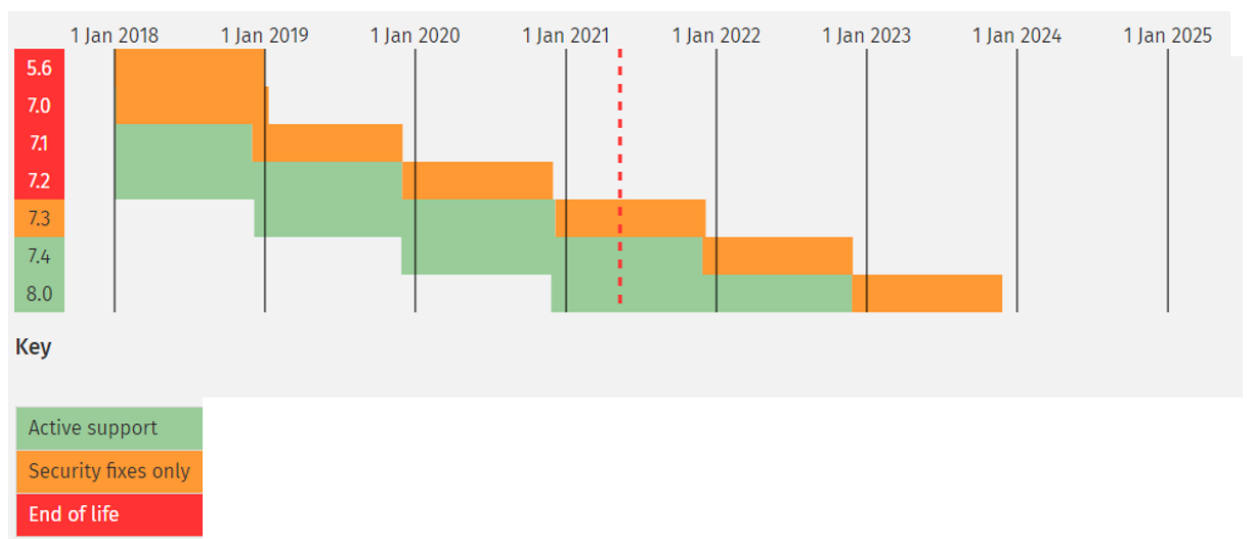


Ilustración 5: versiones de PHP.

A groso modo en PHP se han ido introduciendo mejoras y nuevas características a lo largo de sus versiones para ir adaptándose a las necesidades del usuario, a modo de resumen en la Ilustración 6 se muestra su evolución con los datos más relevantes a destacar de cada una de las versiones (www.php.net, 2021):

Versión	Lanzamiento	Datos relevantes
1.0	08/06/1995	Primer lanzamiento de PHP
2.0	01/11/1997	
3.0	06/06/1998	Se reescribe la base del lenguaje
4.0	22/05/2000	Se introducen las variables superglobales.
5.0	13/07/2004	Aunque la programación orientada a objetos aparecía ya en PHP 3 y 4 es en esta versión donde la POO ya es avanzada.
6		Versión retirada del mercado, fracaso comercial
7.0	03/11/2015	Mejoras de rendimiento
8.0	26/11/2020	Aún en desarrollo

Ilustración 6: evolución de PHP.

Como hitos relevantes, hay que destacar que PHP en si no se diseñó como un lenguaje orientado a objetos, y aunque las versiones 3 y 4 incluían cierta compatibilidad con este paradigma no fue hasta PHP 5.0 donde ya nos encontramos todos los elementos de la POO. Actualmente y desde 2020 se están desarrollando las versiones 8.x que van enfocadas a una mejor eficiencia y funcionamiento del lenguaje.

Ya hemos visto una breve historia de los lenguajes de programación dirigidos al backend de una aplicación web (ASP.NET, Python, Ruby Java y PHP), hemos visto por qué no han tenido el éxito que se esperaba de ellos analizando sus motivos y nos hemos centrado en el más usado, es decir PHP, pero cabe preguntarnos qué dicen del mismo sus detractores, cuáles son sus fortalezas y sus puntos débiles, a continuación, vamos a resumir ambos puntos (Sierra Sedeño & Espinoza Mina, 2018):

Ventajas:

- Facilidad de uso (aunque tiene una curva de aprendizaje bastante alta).
- Rapidez y compatibilidad con la mayor parte de las plataformas web.
- Posibilidad de utilizarlo como lenguaje orientado a objetos.
- Software libre y gratuito.
- Gran soporte y documentación, tiene una gran comunidad detrás y está en continua evolución.

Desventajas:

- No es el lenguaje más apropiado para aplicaciones gráficas.
- Un script que se ha desarrollado en PHP en determinadas ocasiones puede funcionar con mayor lentitud que uno escrito en lenguaje de bajo nivel.
- Necesidad de un servidor de pruebas continuamente para ir comprobando el código que se va desarrollando.
- Falta de soporte por algunos IDEs al tener variables no tipadas.

1.3. Descripción de la programación web: front-end y backend.

Ya hemos dado un repaso a la historia y las distintas tecnologías de programación de un servidor web, pero no hemos entrado en detalle de qué es exactamente el front-end y el backend en este paradigma de programación, es lo que vamos a hacer a continuación (Pérez Ibarra, Quispe, Mullicundo, & Lamas, 2020).

¿Qué es el FrontEnd?

El front-end es la parte que se encarga de mostrar la información al usuario a través de una interfaz, dicha información no puede estar representada de cualquier forma, sino que debe mostrarse de forma que le sea agradable al usuario, que mejore su experiencia de cliente y que le incite a seguir en la página y no abandonarla (Pérez Ibarra, Quispe, Mullicundo, & Lamas, 2020).

En la programación del front-end, por lo tanto, es imprescindible tener conocimientos de diseño de interacción y de diseño de interfaces de usuario.

Actualmente existen un gran número de tecnologías y frameworks que permiten un correcto desarrollo y ayudan a los programadores a la programación en front-end, nos vamos a centrar en las dos más conocidas y con mayor expansión:

JavaScript

Podemos definir JavaScript (MDN contributors, 2022) como un lenguaje de programación que permite dar dinamismo a una página web, es decir modificar la forma en que un usuario visualiza en su navegador el contenido web.

Para eso se basa en dos pilares fundamentales, el código HTML de la propia web, así como el uso de estilos o CSS de la misma, estos son los tres elementos que permiten modificar la visualización de una página web, cómo se representa al usuario final.

JavaScript como lenguaje, se define como un lenguaje de scripting o secuencia de comandos que permite entre otras cosas (MDN contributors, 2022):

- Uso de variables para almacenar valores.

- Manejo de eventos. Dependiendo de ciertas acciones que el usuario pueda hacer sobre una web, JavaScript responde a las mismas y ejecuta cierto código asociado.
- Compatibilidad con la programación orientada a objetos.
- Manejo de cadenas de caracteres, elementos multimedia y mucho más.

Angular

Angular es en sí un framework de trabajo, se basa en el uso de módulos y otros componentes para crear aplicaciones webs completas usando HTML y Typescript. A groso modo la arquitectura de Angular sería la que se muestra en la Ilustración 7 (Angular, 2022):

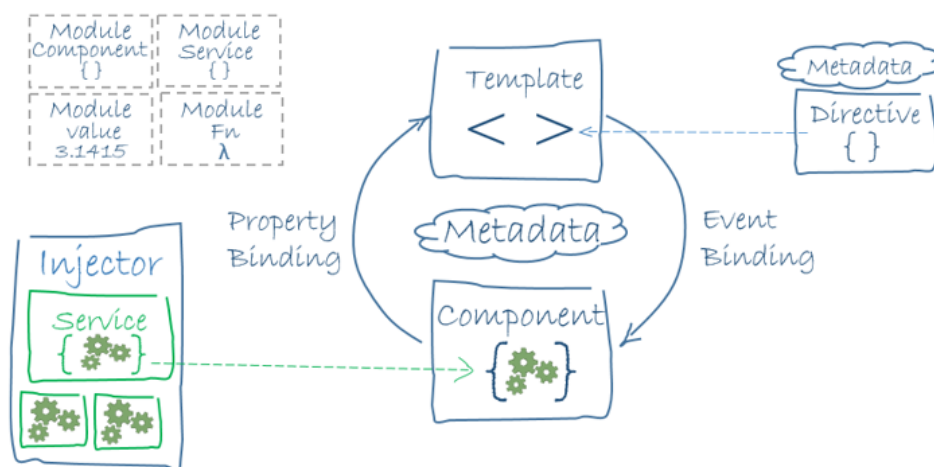


Ilustración 7: arquitectura de Angular.

Para crear una aplicación generamos templates o plantillas HTML y los controlamos con la lógica creada con el resto de elementos (Molina, 2018), todo esto nos lleva a la creación de una sola página HTML que irá cambiando en función a las peticiones del cliente (Web One Page).

En la programación front-end aparte de un lenguaje o una arquitectura de programación, igualmente se hace primordial el intercambio de información para ello se usan lenguajes de transferencia de información siendo los más usados:

JSON

JSON es un formato de intercambio de datos en internet (Json.org, 2022) fácilmente interpretable para los humanos. Es totalmente independiente del lenguaje de programación dónde se use y está constituido básicamente por una colección de pares nombre-valor.

XML

Un archivo XML es un archivo de intercambio de información en internet, no es tan intuitivo e interpretable para los humanos como lo es JSON ya que en lugar de colección de pares nombre-valor la información se encuentra anidada en una estructura tipo árbol mediante etiquetas partiendo de un nodo raíz.

A continuación, en la Ilustración 8 podemos ver las diferencias entre ambos lenguajes de intercambio de información (Dueñas Lerin, 2020):

XML	JSON
<pre><empinfo> <employees> <employee> <name>James Kirk</name> <age>40</age> </employee> <employee> <name>Jean-Luc Picard</name> <age>45</age> </employee> <employee> <name>Wesley Crusher</name> <age>27</age> </employee> </employees> </empinfo></pre>	<pre>{ "empinfo" : { "employees" : [{ "name" : "James Kirk", "age" : 40, }, { "name" : "Jean-Luc Picard", "age" : 45, }, { "name" : "Wesley Crusher", "age" : 27, }] } }</pre>

Ilustración 8: XML vs JSON

¿Qué es el BackEnd?

Por contrapartida al front-end tenemos el backend que se podría definir como la programación y el acceso a los datos que no son directamente accesibles para el usuario final (Pérez Ibarra, Quispe, Mullicundo, & Lamas, 2020), por lo que esta capa contiene toda la lógica de la aplicación.

Habitualmente una aplicación web se encuentra alojada en un servidor, y a la vez los datos a los que se debe de tener acceso también se encuentran en un servidor, que puede ser el mismo donde se encuentre la aplicación u otro, por lo cual la programación del backend de una aplicación web está íntimamente relacionada con el trabajo en el lado servidor de la misma a la vez que se debe manejar activamente el acceso a la información en una o varias bases de datos.

Ya hemos hecho un repaso a las tecnologías y lenguajes de programación más usuales en la programación del backend, vamos a hacer ahora un breve repaso sobre los motores de bases de datos que actualmente son los más usados.

MySql/MariaDB

Son los sistemas de bases de datos relacionales de código abierto más populares del mundo, la mayor parte de los sistemas y aplicaciones webs hacen uso de estos motores de bases de datos.

Oracle

Oracle es un sistema de bases de datos objeto-relacionales, ya que incorporan el paradigma de orientación a objetos. Está desarrollado por Oracle Corporation.

MongoDB

MongoDB no es un sistema SQL, es lo que se conoce como sistema gestor de base de datos NoSQL. Este gestor no utiliza tablas para el almacenamiento de datos sin colecciones, guardando los datos en documentos con una estructura similar a la especificada por JSON.

1.4. El backend.

En este punto ya hemos dado una visión general de la programación web, sus antecedentes, sus tecnologías y hemos diferenciado entre front-end y backend. Ahora nos vamos a centrar en los siguientes apartados en adentrarnos más en la definición de backend para analizar su estado actual y ver las tendencias futuras en los próximos años.

1.4.1. Definición y estado actual.

En la actualidad no se concibe la programación, o en particular la programación web sin la ayuda de un entorno de desarrollo o un framework, en este sentido vamos a analizar los distintos frameworks que podemos encontrar para el desarrollo de aplicaciones web en entorno servidor en función de varios criterios que iremos desglosando (Kaluža, Kalanj, & Vukelić, 2019).

- Generación de código (Tomassetti, 2018): generación automática de código que nos permita autocompletar pequeñas partes o incluso funcionalidades completas a partir de distintos lenguajes y usando distintas técnicas.

- Salud del proyecto (Raible, 2018): entendido como el número de comentarios etiquetas y entradas en StackOverflow.
- Percepción del desarrollador (Raible, 2018): o, dicho de otra forma, la satisfacción del desarrollador en el uso del framework.
- Disponibilidad del desarrollador (Raible, 2018): entendido como el número de desarrolladores que listan un framework en su perfil de LinkedIn.
- Tendencias comerciales (Raible, 2018): número de trabajos que solicitan el conocimiento del framework en cuestión.
- Plantillas (Raible, 2018): número de plantillas que posee el framework.
- Test (Raible, 2018): posibilidad de que acepte o no test.
- Validación (Raible, 2018): posibilidad o no de soportar validación de código.
- Soporte a plataformas móviles (Raible, 2018): posibilidad de integración con plataformas móviles Android y/o iOS.
- Licencia (Raible, 2018): licencia gratuita o con coste.
- Posibilidad de integración (Raible, 2018): capacidad de integrar partes del programa con otras partes o componentes desarrollados por terceros.
- Soporte a varios lenguajes (Raible, 2018): capacidad de escribir partes de código en distintos lenguajes de programación.
- Plugins (Raible, 2018): capacidad o no de añadir plugins o ad-ons en el framework.

En referencia a los criterios indicados previamente y dándole a cada uno de ellos un peso o valor entre 0 y 1 se llega a la siguiente tabla resumen mostrando los resultados del estudio en la Tabla 1 (Kaluža, Kalanj, & Vukelić, 2019)

Criterio	Laravel	ROR	Django	Spring
Generación de código.	1	1	1	1
Salud del proyecto.	1	1	1	1
Percepción del desarrollador.	1	1	1	1
Disponibilidad del desarrollador.	0,5	0,5	0,5	1
Tendencias comerciales.	0,5	0,5	0,5	1
Plantillas.	1	1	1	1
Test.	1	1	1	1
Validación.	1	1	1	1
Soporte a plataformas móviles.	1	1	1	1
Licencia.	1	1	1	1
Posibilidad de integración.	1	1	1	1
Soporte a varios lenguajes.	0	0	0	1
Plugins	1	1	1	1
TOTAL	12	12	12	14

Tabla 1: comparación de frameworks backend.

Como vemos todos los frameworks revisados tienen unas características y funcionalidades parecidas, vamos a analizarlos ahora un poco más en detalle.

Laravel

Laravel es un framework en el que se pueden desarrollar páginas webs completas mediante el lenguaje de programación PHP.

Algunas de sus características son las siguientes (Altube Vera, 2021):

- Se basa en el Modelo Vista Controlador.
- Tiene una arquitectura de proyecto y acceso a bases de datos intuitiva.
- Permite la ampliación con un gran número de módulos adicionales.

Ruby On Rails

Ruby on Rails es un framework de trabajo en el que se pueden desarrollar páginas webs mediante el lenguaje de programación Ruby.

Algunas de sus características son las siguientes (digital55, 2019):

- Está basado en las metodologías ágiles.
- Orientado en el patrón Modelo Vista Controlador.
- Dispone de un gran número de librerías OpenSource.

Django

Django es un framework de trabajo en el que se pueden desarrollar páginas webs mediante el lenguaje de programación Python.

Algunas de sus características son las siguientes (MDN contributors, 2021):

- Orientado en el patrón Modelo Vista Controlador.
- Una de sus filosofías es la reutilización de código y la extensibilidad de sus componentes.

Spring

Spring es un framework de trabajo en el que se pueden desarrollar páginas webs mediante el lenguaje de programación Java.

Algunas de sus características son las siguientes (Muradas , 2018):

- Genera un código liviano y de alto rendimiento.
- Al igual que Java está basado en la programación orientada a objetos.
- Se hace el procesamiento de datos por lotes.

Si nos fijamos en el estado actual de la programación web del backend, ya hemos visto que casi el 80% de las webs con programación servidor están programadas en PHP, pero en la actualidad ¿se sigue usando? ¿hacia dónde se dirige la programación en el nuevo contexto web en continua evolución que nos encontramos? Vamos a ver esa tendencia actual a través de los 5 puntos clave hacia donde se está dirigiendo la programación (Themerkraft, 2021):

1. Programación de IoT: a través de distintos frameworks se está incorporando PHP a la programación del “Internet de las cosas”.
2. Programación en la nube: otra de las tendencias actuales es integrar un servidor y la programación en la nube o cloud computing.
3. Optimización de búsqueda por voz: en uso de asistentes como Siri o Alexa.

4. Interfaces de usuario: con el fin de la mejora de experiencia de usuario en el acceso y uso de la web.
5. Headless Development: es la mayor tendencia actual entendido cómo la programación de una web sin front-end en sí mismo.

A continuación, vamos a repasar algunas de las tendencias actuales en programación backend más interesantes:

SAAS (software como servicio).

El software como servicio (SaaS) es una forma de producción de software en la que el desarrollador del mismo entrega un software en la nube, que será el que el usuario utilice y/o compre (Chnar & Subhi , 2021).

Esto lleva asociado un término menos conocido que es el término de BaaS (Backend como servicio) que es el que se encarga del mantenimiento de toda esa infraestructura en la nube que hace el correcto funcionamiento de todo el sistema completo.

Debido al impulso que están teniendo hoy en día tecnologías de la computación en la nube o cloud computing es una de las mayores tendencias actuales en cuanto a la programación web.

La arquitectura de software basada en microservicios

En la actualidad hemos pasado de la arquitectura software basada en productor/consumidor en la que un backend o productor produce lo que un consumidor le va pidiendo, a una arquitectura basada en microservicios.

Dicha arquitectura consiste el desarrollo de una aplicación a partir de un conjunto de pequeños servicios que están ejecutándose continua e independientemente. Dichos servicios normalmente se comunican entre ellos a través de alguna API que se encarga de dar respuesta a través de una petición HTTP (Pareja Valerio & Burgos Robles, 2019).

Este enfoque de programación es más versátil que el clásico productor consumidor lo que hace que sea el más utilizado en la actualidad en la programación web.

1.4.2. Tendencias futuras.

En cuanto al futuro la programación en general está destinada a mejorar la experiencia cliente, a la optimización y facilitar el trabajo del programador y desarrollador, en este sentido podemos destacar las siguientes tendencias (Nubecolectiva, 2021):

Python

Una de las mayores tendencias en la programación y por consiguiente en la programación web es el aumento de la popularidad entre los programadores del lenguaje de programación Python.

Esta popularidad está basada en su concepto de más por menos, con mucho menos código se permite hacer un programa o aplicación que con otro lenguaje necesitaría muchas más líneas lo que lo hace totalmente eficiente.

Por otro lado, también dispone de un gran número de desarrolladores y un framework en continua evolución como es Django que a la vez es intuitivo y de fácil manejo, con una curva de aprendizaje bastante baja.

Hoy en día se hace primordial el estudio del lenguaje Python para cualquier desarrollador de aplicaciones ya que en un corto/medio plazo va a ser el lenguaje de referencia en la mayor parte de los campos.

Los usos y aplicaciones de Python se encuentran mucho más allá de los campos mencionados, desde el desarrollo de juegos hasta la visualización de datos, desde la creación de redes hasta el desarrollo de software en general. Las aplicaciones de Python son numerosas.

En la Ilustración 9 que corresponde a un informe realizado por web Stackoverflow, se comprueba que desde el 2012 todos los lenguajes se han mantenido estables en cuanto a su evolución, todos excepto Python que ha seguido una evolución ascendente, superando al resto y continuando su crecimiento exponencial desde el año 2018.

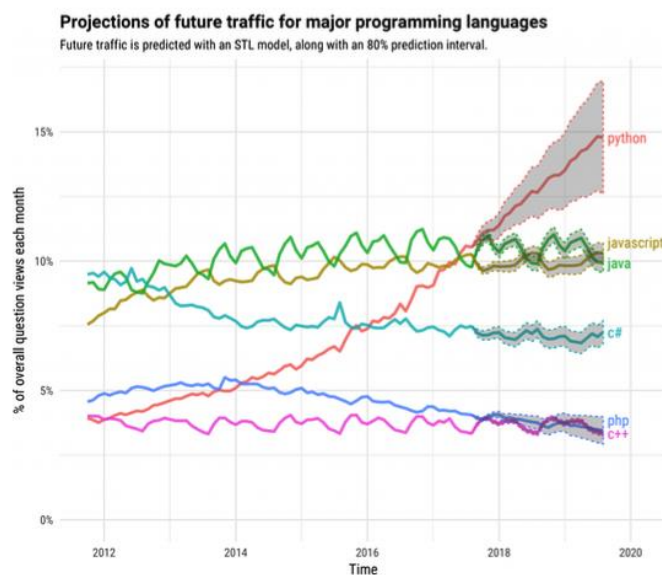


Ilustración 9: evolución de uso de Python

Tecnología de Contenedores

Un contenedor es una forma de virtualización de un sistema operativo y en él se puede ejecutar cualquier cosa o programa que el usuario del mismo desee por lo que la ejecución en un servidor ya no tiene por qué realizarse en un hardware físico sino que se puede realizar en sistema virtualizado (Bernstein, 2014).

En este sentido Docker es la tecnología con mayor proyección de futuro y en la que actualmente más están invirtiendo las empresas como se observa en la Ilustración 10, ya en 2017 el 15% de los host se ejecutaban sobre Docker (Lucas , 2017):

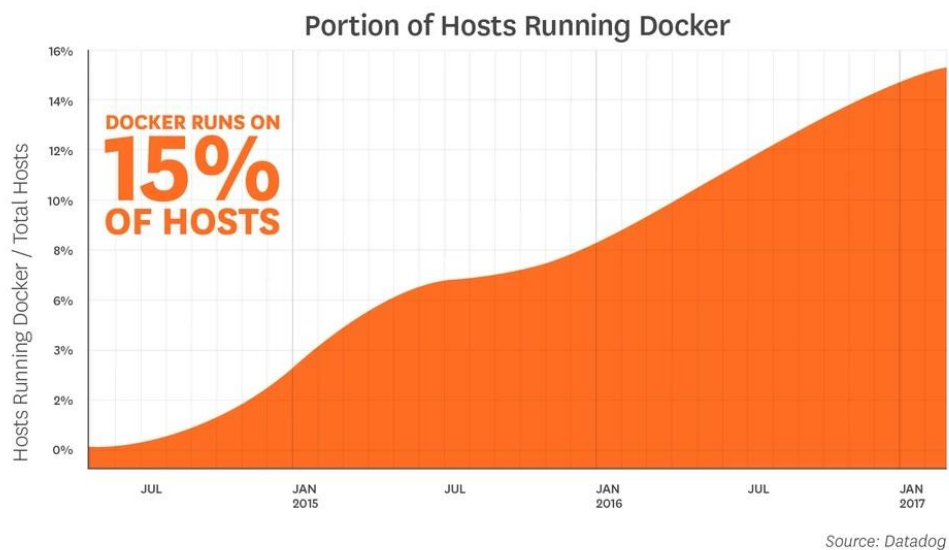


Ilustración 10: evolución de Docker.

Blockchain

La tecnología Blockchain es una metodología que, aunque no es nueva ha empezado a tener impacto en los últimos años, convirtiéndose en una tendencia futura en el mundo de la programación.

Blockchain si lo traducimos directamente al español es una cadena de bloque, y es la forma de trabajar de esta tecnología, se programa mediante bloques que contienen información encriptada, a partir de estos bloques se pueden realizar infinidad de aplicaciones (Nubecolectiva, 2021).

Go

El lenguaje Python que hemos comentado anteriormente no es el único que tiene una fuerte proyección de futuro, en este sentido también tenemos que comentar la evolución del lenguaje de programación GO.

Dicho lenguaje, comparado con otros es relativamente nuevo, ya que nace en 2009 desarrollado por Google.

Alguna de las características que le dan esa proyección de futuro en la programación de aplicaciones web en lado servidor son (KeepCoding, 2020):

- Simplicidad y facilidad de uso, uno de los paradigmas de sus creadores ha sido crear un lenguaje asequible para todos los usuarios con una curva de aprendizaje baja.
- Funcionalidades adicionales que permiten realizar una programación de alto nivel en todos sus sentidos.
- En cierto modo se basa en la programación con C y se dice por muchos expertos que es la continuación natural del mismo.

En la Ilustración 11 se muestra un estudio realizado por la web hackernoon, en la que se hace evidente la evolución de GO en el futuro, si nos fijamos en la programación backend en dicho estudio ya el 11% de los desarrolladores planean usar dicho lenguaje para sus desarrollos (Sidorenko, 2019):

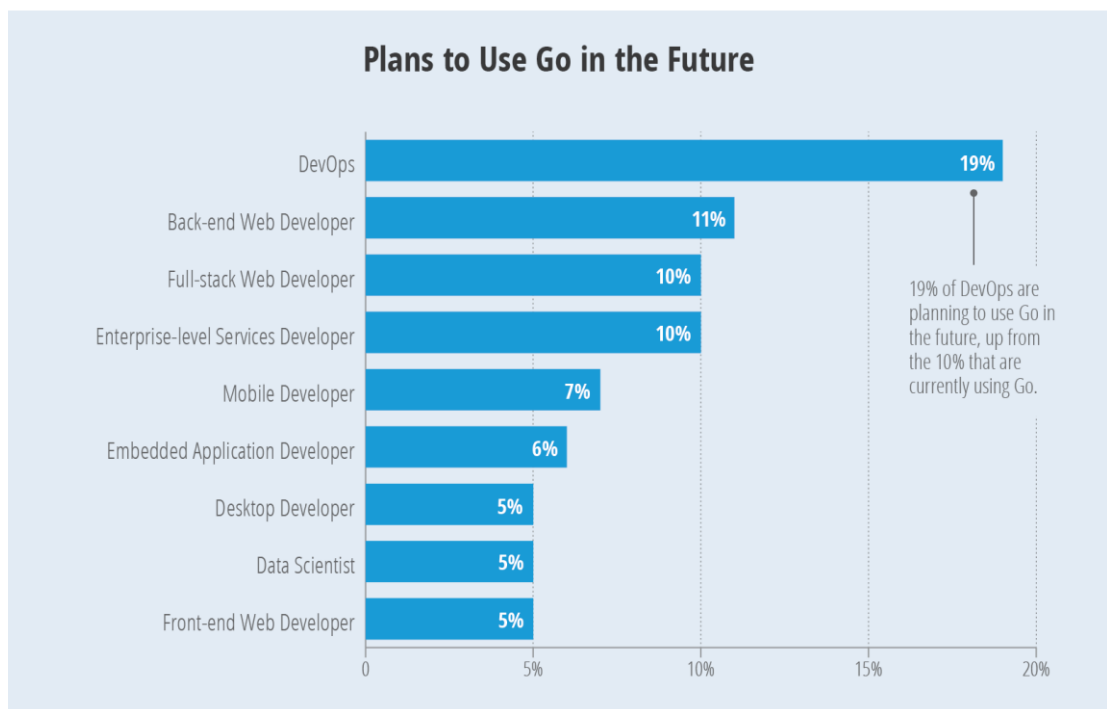


Ilustración 11: futuro del lenguaje GO.

1.5. Conclusión.

A lo largo de los apartados anteriores se ha querido dar una visión global y generalizada de lo que es la programación web, dando unas pinceladas a la parte del front, pero adentrándonos de lleno en la parte del back.

Se han abordado varias tecnologías, se han repasado distintos frameworks, se ha estudiado el estado actual del mercado y se ha previsto las posibles tendencias futuras, hacia dónde va la programación web.

En el mundo incierto y cambiante en el que siempre se mueve el ámbito de la informática, la programación web no es menos, y más en un campo que va dirigido hacia el cliente, los clientes cada vez son más exigentes, necesitan que les facilitemos el trabajo dándole respuesta sus peticiones en cada vez un tiempo más corto, e independientemente de la tecnología, framework o tendencia esto es lo que se busca al final con una programación correcta del servidor, asegurar que se da el servicio que el cliente nos está demandando con el menor número de recursos necesarios, invirtiendo el menor tiempo posible y dando una respuesta de calidad que haga al usuario repetirla en un futuro.

2. DESARROLLO DE UNA UNIDAD DIDÁCTICA.

Una vez vista la fundamentación epistemológica vamos de la programación web o más concretamente la programación web en entorno backend, vamos a englobar dichos contenidos dentro de unos estudios actuales.

Por el contenido práctico y su clara orientación al mercado laboral, está claro que la programación web podría encajar a la perfección en estudios de formación profesional, si nos centramos en los de informática y comunicaciones actualmente nos encontramos los siguientes:

- Técnico en Sistemas Microinformáticos y Redes.
- Técnico Superior en Administración de Sistemas Informáticos en Red.
- Técnico Superior en Desarrollo de Aplicaciones Multiplataforma.
- Técnico Superior en Desarrollo de Aplicaciones Web.

De estos cuatros títulos de formación profesional uno corresponde a estudios de grado medio (Técnico en Sistemas Microinformáticos y Redes), aunque ya por el nombre resultaría lógico en cuál de ellos podríamos encajar la programación web en entorno backend revisamos la competencia general de cada uno de ellos establecida en su correspondiente RD y nos encontramos (ver apartado 4.5.1.) con que en el título de Técnico Superior en Desarrollo de Aplicaciones Web entre otras cosas el RD dice:

“La competencia general de este título consiste en desarrollar, implantar, y mantener aplicaciones web, con independencia del modelo empleado y utilizando tecnologías específicas...”

Por lo cual este sería el título dónde vamos a incluir la UD, ahora queda por decidir el módulo dentro del título, para ello nos remitimos nuevamente al Real Decreto 686/2010, de 20 de mayo, por el que se establece el título de Técnico Superior en Desarrollo de Aplicaciones Web y se fijan sus enseñanzas mínimas y vemos los que módulos que incluye son:

- 0483. Sistemas informáticos.
- 0484. Bases de datos.
- 0485. Programación.
- 0373. Lenguajes de marcas y sistemas de gestión de información.
- 0487. Entornos de desarrollo.
- 0612. Desarrollo web en entorno cliente.
- 0613. Desarrollo web en entorno servidor.

- 0614. Despliegue de aplicaciones web.
- 0615. Diseño de interfaces WEB.
- 0616. Proyecto de desarrollo de aplicaciones web.
- 0617. Formación y orientación laboral.
- 0618. Empresa e iniciativa emprendedora.
- 0619. Formación en centros de trabajo.

Claramente hay un módulo específico para el temo que nos ocupa que en este caso sería:

- 0613. Desarrollo web en entorno servidor.

Aquí es donde vamos a ubicar nuestra unidad didáctica.

2.1. Introducción.

Actualmente la programación es fundamental para acrecentar el avance tecnológico dentro de las industrias que, para llevar a cabo sus funciones necesitan de sitios y aplicaciones creadas a partir de páginas webs.

No es novedad que vivimos en un mundo computarizado donde la tecnología forma parte de todo lo que hacemos. En este contexto se hace aún más necesaria e importante la programación de aplicaciones web dinámicas, para ello se hace imprescindible el conocimiento de la programación del servidor de una aplicación web que sea capaz de adaptarse a las necesidades del usuario de forma dinámica.

En este contexto la programación web y concretamente la programación del servidor o backend se hacen imprescindibles cuestiones como:

- Autenticación y seguridad: necesitamos tener un control de los accesos a un servicio o aplicación web, independientemente del tipo de aplicación que sea (tienda virtual, gestor de contenidos etcétera) no solo ya para llevar el control de acceso sino para cerciorarnos del uso correcto del servicio y por supuesto evitar todas las posibles brechas de seguridad.
- Sesiones y cookies: en la navegación web la velocidad de respuesta es fundamental, por lo que se necesita en cierto modo agilizar todo lo posible las peticiones de los clientes, en este contexto juegan un papel importante el almacenamiento temporal de información en un entorno de programación web.
- Depuración de código: dar agilidad a una aplicación web no está reñido con la calidad de la misma, para garantizar un correcto funcionamiento de cualquier

servicio que un servidor web pueda ofrecer nos debemos de asegurar que el código sea limpio y eficiente, para ello se hace primordial el uso de herramientas que nos permitan realizar estas labores.

A partir de estos principios básicos de autenticación, seguridad, sesiones, cookies y depuración se va a crear nuestra unidad didáctica, que (aunque se irá especificando en apartados posteriores) será la correspondiente unidad 4 dentro del módulo anteriormente indicado (desarrollo web en entorno servidor).

Aclarar que una vez llegado a este punto el alumno ya dispone de ciertos conocimientos que se pasan a relatar a continuación:

- Conoce el concepto de programación web en entorno servidor y sabe diferenciarla de la programación web en entorno cliente.
- Ha estudiado e interiorizado los elementos de un lenguaje propio de programación web en entorno servidor, en este caso PHP, sabiendo su estructura, tipos, sentencias, programación orientada a objetos y en definitiva todo lo que el lenguaje puede ofrecer.
- Ha practicado el acceso a bases de datos relacionales a través de programación en el servidor con el lenguaje de programación PHP.

2.2. Ubicación de la enseñanza.

El módulo de Desarrollo Web en Entorno Servidor es uno de los módulos profesionales del Ciclo Formativo de Grado Superior de Desarrollo de Aplicaciones Web. Pertenece al segundo curso de dicho ciclo formativo.

Este módulo profesional, debido a lo extenso de sus contenidos y a la enorme importancia que tiene en la adquisición de las capacidades terminales del ciclo formativo, se desglosa en 8 unidades de trabajo, en este texto se desarrollará la UD04, siendo el resto las siguientes:

UD01: Programación web en entorno servidor. Introducción y conceptos básicos.

UD02: Características de un lenguaje de programación web. PHP.

UD03: Bases de datos con PHP.

UD04: Desarrollo de aplicaciones web con PHP.

UD05: Programación avanzada en PHP.

UD06: Servicios web.

UD07: Aplicaciones web dinámicas.

UD08: Aplicaciones web híbridas.

A este módulo profesional le corresponden 168 horas de clase o lectivas de las cuales tenemos 8 horas semanales repartidas de la siguiente forma:

- 3 horas lunes de 09:00 a 12:00.
- 3 horas miércoles de 09:00 a 12:00.
- 2 horas viernes de 10:00 a 12:00.

En cuanto a la UD desarrollada en este texto, se impartirá en la segunda parte del segundo trimestre y se compondrá de las sesiones y horarios anteriormente descritos.

2.3. Normativa.

Para la realización de la unidad didáctica que nos ocupa se ha tenido en cuenta la siguiente normativa:

- **RD:** Real Decreto 686/2010, de 20 de mayo, por el que se establece el título de Técnico Superior en Desarrollo de Aplicaciones Web y se fijan sus enseñanzas mínimas.
- **RD:** Real Decreto 1147/2011, de 29 de julio, por el que se establece la ordenación general de la formación profesional del sistema educativo.
- **ORDEN:** 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.
- **LOE:** Ley Orgánica 2/2006, de 3 de mayo, de Educación.
- **LOMCE:** Ley Orgánica 8/2013, de 9 de diciembre, para la mejora de la calidad educativa.
- **LEA:** Ley 17/2007, de 10 de diciembre, de Educación de Andalucía.

2.4. Contexto.

A continuación, se va a analizar en profundidad el entorno profesional, socioeconómico y cultural del centro, así como las características del alumnado al que se va a impartir la unidad didáctica.

Estas características son el eje en el cual basar el desarrollo y la planificación de todos los procesos de aprendizaje, el profesorado se tendrá que adaptar a estos elementos para una correcta planificación de la enseñanza y así realizar correctamente las pertinentes programaciones.

2.4.1. Contexto socioeconómico del centro.

El instituto se encuentra situado a las afueras de la ciudad de Jaén. Esta ciudad es la capital de provincia y en el censo realizado en 2018 cuenta con un total de 113.457 habitantes.

Ubicado en una zona en expansión con nivel socioeconómico medio - alto.

Al estar ubicado en una zona en expansión a las afueras de la ciudad el centro también es un instituto de referencia para todas las poblaciones colindantes. Además, dispone de numerosas empresas cercanas para poder realizar el módulo de FCT sin que los alumnos se tengan que desplazar.

Así mismo tiene cuatro centros de educación primaria adscritos.

Si analizamos los datos sociales del centro, vamos a ver en primer lugar la nacionalidad de los padres y madres del alumnado:

- Nacionalidad española: alrededor del 85%.
- Nacionalidad no española: alrededor del 15%.

Según los estudios realizados la mayor parte de las familias del alumnado del centro cuentan con más de tres miembros, alrededor del 95% de las mismas.

En cuanto a los datos económicos según los datos actuales de la zona donde se encuentra el instituto, se cuenta con los siguientes datos:

- Población activa del 57%.
- Tasa de empleo del 48%.
- Tasa de desempleo del 12%.

Se han realizado también estudios sobre la situación profesional y laboral de las familias de los alumnos obteniéndose los siguientes resultados:

- Padres con trabajo remunerado: 66%.
- Padres en situación de desempleo: 30%.
- Padres inactivos por diversos motivos (jubilación, incapacidad): 4%.

2.4.2. Características del centro.

El centro empezó su trayectoria en el curso 2004-2005 y está diseñado para 520 alumnos y alumnas contando en el curso actual con 425 alumnos y alumnas matriculados.

En cuanto a la organización tiene los siguientes cargos unipersonales:

- Director.
- Vicedirector.
- Jefe de Estudios.
- Jefe de Estudios Adjunto.
- Jefe de Estudios Adjunto de FP.
- Secretario.

En el centro de referencia se imparten las siguientes enseñanzas:

- ESO: teniendo los cursos de 1º a 4º en todas sus modalidades.
- Bachillerato: 1º y 2º en todas sus modalidades.
- FP Básica.
- Formación Profesional de grado medio:
 - Técnico en Sistemas Microinformáticos y Redes.
- Formación Profesional de grado superior:
 - Técnico Superior en Desarrollo de Aplicaciones Multiplataforma.
 - Técnico Superior en Desarrollo de Aplicaciones Web.
- Formación Profesional de grado superior a distancia:
 - Técnico Superior en Desarrollo de Aplicaciones Multiplataforma.
 - Técnico Superior en Desarrollo de Aplicaciones Web.

El instituto dispone de los siguientes recursos disponibles:

- 5 aulas de informática con equipos renovados el curso 2019-2020. Dichos equipos disponen de conexión de internet de altas capacidades.
- Todas las aulas disponen de pizarra digital y tradicional además de proyectores.
- Mobiliario necesario para impartir la docencia.
- Todo el material con los alumnos se comparte a través de la Moodle del centro, dicha plataforma está adscrita a la junta de Andalucía y los alumnos acceden a

ella mediante su clave de iPasen, la plataforma es accesible desde la URL:
<https://educacionadistancia.juntadeandalucia.es/centros/jaen/login/index.php>

2.4.3. Características del alumnado.

Si nos centramos ahora en el alumnado nos encontramos que la clase donde se va a impartir la unidad didáctica que nos ocupa consta de 20 alumnos y alumnas de entre 18 y 25 años de edad de los cuales:

- 85% son hombres un total de 17.
- 15% son mujeres un total de 3.

La formación o procedencia de dichos alumnos y alumnas es diversa desglosándose de la siguiente forma es:

- 55% provienen del ciclo de grado medio de SMR, un total de 11.
- 10% ya han cursado el ciclo de grado superior de DAM, un total de 2.
- 30% provienen de Bachillerato, un total de 6.
- 5% provienen del mundo laboral, un total de 1.

Por lo cual tienen un nivel de conocimientos previos medio/bajo, sobre todo en el campo de desarrollo de aplicaciones web en el lado del servidor, aunque si disponen de otros conocimientos informáticos y un razonamiento lógico.

Uno de los alumnos del grupo presenta movilidad reducida al tener que desplazarse mediante silla de ruedas.

Por último, según los sondeos realizados los intereses de los alumnos en cuánto al estudio del título en cuestión serían los siguientes:

- 75% incorporación al mundo laboral.
- 20% ampliar una formación previa.
- 5% otros motivos.

2.5. Objetivos.

Una vez analizado en detalle el centro y los alumnos y alumnas objeto de la unidad didáctica que se está desarrollando vamos a adentrarnos en analizar sus objetivos, para tener una visión global vamos a ver objetivos, competencias y contenidos generales del título y del módulo para luego centrarnos en los específicos de la unidad didáctica.

2.5.1. Competencia general.

En Real Decreto 686/2010, de 20 de mayo, por el que se establece el título de Técnico Superior en Desarrollo de Aplicaciones Web y se fijan sus enseñanzas mínimas se establece la competencia general del título en cuestión y sería la siguiente:

“La competencia general de este título consiste en desarrollar, implantar, y mantener aplicaciones web, con independencia del modelo empleado y utilizando tecnologías específicas, garantizando el acceso a los datos de forma segura y cumpliendo los criterios de accesibilidad, usabilidad y calidad exigidas en los estándares establecidos.”

2.5.2. Objetivos generales del título.

De conformidad con lo establecido en el artículo 9 del Real Decreto 686/2010, de 20 de mayo, por el que se establece el título de Técnico Superior en Desarrollo de Aplicaciones Web y se fijan sus enseñanzas mínimas, los objetivos generales de las enseñanzas correspondientes al mismo son:

- a) Ajustar la configuración lógica analizando las necesidades y criterios establecidos para configurar y explotar sistemas informáticos.
- b) Identificar las necesidades de seguridad verificando el plan preestablecido para aplicar técnicas y procedimientos relacionados.
- c) Instalar módulos analizando su estructura y funcionalidad para gestionar servidores de aplicaciones.
- d) Ajustar parámetros analizando la configuración para gestionar servidores de aplicaciones.
- e) Interpretar el diseño lógico, verificando los parámetros establecidos para gestionar bases de datos.
- f) Seleccionar lenguajes, objetos y herramientas, interpretando las especificaciones para desarrollar aplicaciones web con acceso a bases de datos.
- g) Utilizar lenguajes, objetos y herramientas, interpretando las especificaciones para desarrollar aplicaciones web con acceso a bases de datos.
- h) Generar componentes de acceso a datos, cumpliendo las especificaciones, para integrar contenidos en la lógica de una aplicación web.
- i) Utilizar lenguajes de marcas y estándares web, asumiendo el manual de estilo, para desarrollar interfaces en aplicaciones web

- j) Emplear herramientas y lenguajes específicos, siguiendo las especificaciones, para desarrollar componentes multimedia.
- k) Evaluar la interactividad, accesibilidad y usabilidad de un interfaz, verificando los criterios preestablecidos, para Integrar componentes multimedia en el interfaz de una aplicación.
- l) Utilizar herramientas y lenguajes específicos, cumpliendo las especificaciones, para desarrollar e integrar componentes software en el entorno del servidor web.
- m) Emplear herramientas específicas, integrando la funcionalidad entre aplicaciones, para desarrollar servicios empleables en aplicaciones web.
- n) Evaluar servicios distribuidos ya desarrollados, verificando sus prestaciones y funcionalidad, para integrar servicios distribuidos en una aplicación web.
- ñ) Verificar los componentes de software desarrollados, analizando las especificaciones, para completar el plan de pruebas.
- o) Utilizar herramientas específicas, cumpliendo los estándares establecidos, para elaborar y mantener la documentación de los procesos.
- p) Establecer procedimientos, verificando su funcionalidad, para desplegar y distribuir aplicaciones.
- q) Programar y realizar actividades para gestionar el mantenimiento de los recursos informáticos.
- r) Analizar y utilizar los recursos y oportunidades de aprendizaje relacionadas con la evolución científica, tecnológica y organizativa del sector y las tecnologías de la información y la comunicación, para mantener el espíritu de actualización y adaptarse a nuevas situaciones laborales y personales.
- 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.
- u) Desarrollar técnicas de liderazgo, motivación, supervisión y comunicación en contextos de trabajo en grupo para facilitar la organización y coordinación de equipos de trabajo.

- v) Aplicar estrategias y técnicas de comunicación adaptándose a los contenidos que se van a transmitir, la finalidad y a las características de los receptores, para asegurar la eficacia en los procesos de comunicación.
- x) Evaluar situaciones de prevención de riesgos laborales y de protección ambiental, proponiendo y aplicando medidas de prevención personales y colectivas, de acuerdo a la normativa aplicable en los procesos del trabajo, para garantizar entornos seguros.
- y) Identificar y proponer las acciones profesionales necesarias para dar respuesta a la accesibilidad universal y al diseño para todos
- z) Identificar y aplicar parámetros de calidad en los trabajos y actividades realizados en el proceso de aprendizaje para valorar la cultura de la evaluación y de la calidad y ser capaces de supervisar y mejorar procedimientos de gestión de calidad.
- aa) Utilizar procedimientos relacionados con la cultura emprendedora, empresarial y de iniciativa profesional, para realizar la gestión básica de una pequeña empresa o emprender un trabajo.
- ab) Reconocer sus derechos y deberes como agente activo en la sociedad, teniendo en cuenta el marco legal que regula las condiciones sociales y laborales para participar como ciudadano democrático.

2.5.3. Objetivos generales del módulo.

Centrándonos ahora en el módulo 0613. Desarrollo web en entorno servidor, la formación del dicho módulo contribuye a alcanzar de los objetivos generales del título indicados en el apartado 4.5.2. los que se relacionan a continuación:

- 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.
- 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.

2.5.4. Competencias profesionales, personales y sociales del módulo.

La formación del módulo 0613. Desarrollo web en entorno servidor, también contribuye a alcanzar de las competencias profesionales, personales y sociales del título descritas en el artículo 9 del Real Decreto 686/2010, de 20 de mayo, por el que se establece el título de Técnico Superior en Desarrollo de Aplicaciones Web que se relacionan a continuación:

- 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.

2.5.5. Líneas de actuación del módulo.

Por otro lado, las líneas de actuación en el proceso de enseñanza-aprendizaje que permiten alcanzar los objetivos del módulo profesional, descritas en la 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, versarán sobre:

- a) El análisis de los métodos de generación dinámica de documentos Web.
- b) La integración del lenguaje de marcas con el código ejecutable en el servidor Web.
- c) El análisis, diferenciación y clasificación de las características y funcionalidades incorporadas en los entornos y lenguajes de programación de los servidores Web más difundidos.
- d) La utilización de características y funcionalidades específicas de los lenguajes de programación seleccionados.
- e) La modificación del código existente en soluciones Web heterogéneas para su adaptación a entornos específicos.

- f) El análisis y la utilización de funcionalidades aportadas por librerías generales y específicas de programación web en entorno servidor.
- g) La utilización de librerías para incorporar interactividad a los documentos Web generados de forma dinámica.

2.5.6. Orientaciones pedagógicas del módulo.

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 definida en la 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, incluye aspectos como:

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

2.5.7. Objetivos específicos de la unidad didáctica.

De los objetivos del módulo descritos en el apartado 4.5.3. en esta unidad didáctica se trabajan los siguientes:

- 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.

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.

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.

2.5.8. Objetivos didácticos.

Los objetivos didácticos que se pretenden conseguir al finalizar esta unidad serían:

1. Conocer por qué es importante la autenticación de usuarios, tipos de usuarios y tecnologías disponibles.
2. Ser capaz de aplicar una autenticación de usuarios sencilla a una aplicación web.
3. Implementar correctamente las cookies en una aplicación web.
4. Manejar y conocer de forma completa las sesiones en PHP, como manejarlas y sacarles el máximo partido.
5. Diferenciar distintas tecnologías de depuración de código en entorno servidor.
6. Aplicar una tecnología (Xdebug) de depuración de código de PHP.

2.6. Resultados de aprendizaje.

El título según el Real Decreto 686/2010, de 20 de mayo, por el que se establece el título de Técnico Superior en Desarrollo de Aplicaciones Web y se fijan sus enseñanzas Mínimas, se compone de los siguientes resultados de aprendizaje:

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

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

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

RA4. Desarrolla aplicaciones Web embebidas en lenguajes de marcas analizando e incorporando funcionalidades según especificaciones.

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

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

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

RA8. Genera páginas Web dinámicas analizando y utilizando tecnologías del servidor Web que añadan código al lenguaje de marcas.

RA9. Desarrolla aplicaciones Web híbridas seleccionando y utilizando librerías de código y repositorios heterogéneos de información.

De los cuales en esta unidad didáctica se trabaja el siguiente:

RA4. Desarrolla aplicaciones Web embebidas en lenguajes de marcas analizando e incorporando funcionalidades según especificaciones.

2.7. Contenidos.

Una vez vistos los objetivos tanto del título como del módulo y específicamente de la unidad didáctica ya sabemos a dónde queremos llegar, vamos ahora a definir los contenidos para ver cómo vamos a llegar al cumplimiento de dichos objetivos.

Para ello se va a hacer lo mismo que en el apartado anterior, vamos a ir de los contenidos generales del módulo a los específicos de la unidad didáctica.

2.7.1. Contenidos generales del módulo.

Los contenidos básicos que deben impartirse en este módulo y que así están especificados la 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. en son los siguientes.

Selección de arquitecturas y herramientas de programación:

- Modelos de programación en entornos cliente/servidor. Características.
- Generación dinámica de páginas Web. Ventajas.

- Lenguajes de programación en entorno servidor. Tipos. Características.
- Integración con los lenguajes de marcas.
- Integración con los servidores Web.
- Herramientas de programación. Tipos.

Inserción de código en páginas Web:

- Mecanismos de generación de páginas Web. Lenguajes embebidos en HTML.
- Tecnologías asociadas: PHP, ASP, JSP, «Servlets», entre otras.
- Obtención del lenguaje de marcas para mostrar en el cliente.
- Sintaxis del lenguaje.
- Etiquetas para inserción de código.
- Directivas.
- Tipos de datos. Conversiones entre tipos de datos.
- Variables. Tipos. Ámbito de una variable.

Programación basada en lenguajes de marcas con código embebido:

- Sentencias. Tipos. Bloques.
- Comentarios.
- Tomas de decisión.
- Bucles.
- Tipos de datos compuestos: Arrays.
- Funciones. Parámetros.
- Recuperación y utilización de información proveniente del cliente Web.
- Procesamiento de la información introducida en un formulario. Métodos POST y GET.

Desarrollo de aplicaciones Web utilizando código embebido:

- Mantenimiento del estado. Sesiones. Cookies.
- Seguridad: usuarios, perfiles, roles.
- Mecanismos de autenticación de usuarios.
- Adaptación a aplicaciones Web: Gestores de contenidos y tiendas virtuales entre otras.
- Pruebas y depuración. Herramientas y entornos.

Generación dinámica de páginas Web:

- Mecanismos de separación de la lógica de negocio.
- Patrones de diseño: el patrón MVC u otros.
- Controles de servidor.
- Mecanismos de generación dinámica del interface Web.
- Generación dinámica de formularios.
- Prueba y documentación del código.

Utilización de técnicas de acceso a datos:

- Establecimiento de conexiones.
- Recuperación y edición de información.
- Utilización de conjuntos de resultados.
- Ejecución de sentencias SQL.
- Transacciones.
- Utilización de otros orígenes de datos.

Programación de servicios Web:

- Arquitecturas de programación orientadas a servicios.
- Mecanismos y protocolos implicados.
- Generación de un servicio Web.
- Descripción del servicio.
- Interface de un servicio Web.
- Uso del servicio.

Generación dinámica de páginas Web interactivas:

- Procesamiento en el servidor y en el cliente.
- Librerías y tecnologías relacionadas.
- Generación dinámica de páginas interactivas.
- Controles con verificación de información en el cliente.
- Obtención remota de información.
- Modificación de la estructura de la página Web.

Desarrollo de aplicaciones Web híbridas:

- Reutilización de código e información.
- Repositorios de código. Utilización de información proveniente de repositorios.
- Frameworks. Tipos y características.
- Incorporación de funcionalidades específicas.
- Prueba y documentación de aplicaciones Web.

2.7.2. Contenidos específicos de la unidad didáctica.

De los contenidos del módulo que se han descrito en el apartado anterior, la unidad didáctica que se está desarrollando contemplaría los siguientes:

Desarrollo de aplicaciones Web utilizando código embebido:

- Mantenimiento del estado. Sesiones. Cookies.
- Seguridad: usuarios, perfiles, roles.
- Mecanismos de autenticación de usuarios.
- Adaptación a aplicaciones Web: Gestores de contenidos y tiendas virtuales entre otras.
- Pruebas y depuración. Herramientas y entornos.

A partir de dichos contenidos se ha realizado y estructurado nuestra unidad didáctica de la siguiente forma:

1. Control de acceso mediante autenticación.
 - a. Autenticación.
 - b. Aplicación web segura. Métodos de autenticación.
 - c. Cookies en PHP.
2. Sesiones en PHP.
 - a. Conceptos básicos.
 - b. Iniciar y finalizar sesiones.
 - c. Manejo de información mediante sesiones.
3. Depuración de código.
 - a. Herramientas de depuración.
 - b. Depuración en PHP.

2.7.3. Interdisciplinariedad.

Este módulo no se puede ver como un ente solitario, sino que está íntimamente relacionado con otros del título para componer el conocimiento global que va a necesitar el alumnado y que le permita desarrollar correctamente su carrera profesional. En este sentido este módulo de desarrollo web en entorno servidor está relacionado con los siguientes módulos del mismo título:

- Bases de datos: es primordial el acceso a los datos almacenados para una aplicación web, no se concibe la misma sin el manejo de una base de datos.
- Lenguajes de marcas: al final lo que se presenta al cliente es una web, dicha web está compuesta por un código HTML que es un lenguaje de marcas, por lo cual están íntimamente relacionados los dos módulos.
- Programación: para poder cursar el módulo que nos ocupa de forma adecuada el alumno debe tener ya adquiridos unos conceptos de programación previos en los que no se profundizan, para ello es primordial la relación en el módulo de programación.
- Entornos de desarrollo: el alumno necesita conocer y manejar los entornos dónde va a realizar la programación.
- Desarrollo de interfaces web: la forma en que un servidor presenta la información al cliente a través de una interfaz es crucial para el éxito o el fracaso de una aplicación web.
- Desarrollo web en entorno cliente: hoy en día no se concibe la programación web en entorno servidor sin contemplar también la programación web en entorno cliente, son dos tecnologías distintas, pero íntimamente relacionadas entre sí y que conviven en la mayor parte de las aplicaciones webs actuales.

2.7.4. Temas transversales.

La unidad didáctica que nos ocupa se va a impartir en fechas cercanas a la feria del libro que se celebra el 23 de abril por lo que se va a trabajar el fomento de la lectura y día del libro como tema transversal.

Para ello se van a diseñar diversas actividades de enseñanza-aprendizaje (ver apartado 4.8.1.), pero básicamente la finalidad es la que se va a comentar a continuación.

Aprovechando el estudio de esta unidad donde se ven métodos de autenticación, cookies y sesiones, se realizará como actividad que englobe el tema transversal previamente mencionado la realización de una web de intercambio de libros que permita las siguientes funcionalidades:

- Login de usuario: habrá dos tipos de usuario alumno y administrador.
- Usuario alumno: podrá logarse o registrarse para poder solicitar un libro de los que previamente estén registrados en una pequeña base de datos.
- Usuario administrador: se encarga de la gestión de la base de datos, registrando las nuevas donaciones y retirando los libros que se vayan donando.

El funcionamiento es el siguiente:

1. Un alumno dona un libro que ya haya leído o no necesite, lo lleva al administrador (puede ser el mismo profesor) que lo registra.
2. Un alumno puede consultar la web para ver que han donado otros alumnos.
3. Si algún libro le interesa lo añade al “carro de la compra” y envía un pedido.
4. El administrador entrega al alumno su pedido.
5. Para poder retirar un libro previamente se tiene que haber donado, es decir por cada libro donado puedes retirar otro.

Entre todas las aplicaciones que hagan los alumnos se elegirá mediante una votación de los propios alumnos la que se va a implantar para llevar a cabo la aplicación real.

2.8. Metodología.

Una vez conocidos los objetivos y los contenidos vamos a centrarnos en cómo vamos a impartir y evaluar dichos contenidos, es decir, la metodología.

En esta unidad como en el resto del módulo a rasgos generales la metodología va orientada a:

- Pensamiento lógico: entendiéndose como la capacidad del alumno para entender un problema relacionándolo con el contexto y con lo que le rodea para llegar a entender el mismo y así poder tener una solución.
- Resolución de problemas: fomentar la capacidad analítica para la resolución de problemas matemáticos.
- Aprendizaje online: aprender a usar las plataformas online y las nuevas tecnologías.
- Trabajo en equipo: en la informática se trabaja con proyectos lo que implica un equipo. Se pretende que el alumno sea capaz de asimilar el concepto de grupo cuando la situación la requiera para trabajar en común para un proyecto o problema dejando en segundo plano la individualidad.

2.8.1. Actividades de enseñanza-aprendizaje.

Para desarrollar en el aula la unidad didáctica que nos ocupa, de forma genérica, se van a llevar a cabo las siguientes actividades de enseñanza-aprendizaje:

- Act1: Actividad introductoria. Para introducir un tema e iniciar al alumnado en la materia que se va a ver en cada una de las sesiones.
- Act2: Actividad de desarrollo de contenidos. Con la finalidad desarrollar los contenidos propuestos para la consecución de los objetivos y la adquisición de las competencias profesionales.
- Act3: Actividad de elementos transversales. Con los el tema del día del libro y el fomento de la lectura.
- Act4: Actividad expositiva. El profesor transmite los conocimientos a los alumnos que desea que estos adquieran.
- Act5: Actividad práctica. En la que el alumno realiza una tarea práctica asignada por el profesor.
- Act6: Actividad de exposición. En la que el alumno expone ante los compañeros un tema previamente acordado con el profesor.
- Act7. Actividad de evaluación. Realización de prueba de evaluación.
- Act8. Actividad de consolidación. Realización de alguna actividad para consolidar los conocimientos.

A partir de las actividades mencionadas anteriormente, la organización de las sesiones 8 sesiones de la unidad sería la siguiente:

Sesiones		
1	Act1. Se introduce al alumno el concepto de control de acceso mediante autenticación. - 30' Act4. El profesor explicará en qué consiste la autenticación de usuarios y como llevarla a cabo – 130' Act8. Kahoot para afianzar los conocimientos. - 20'	180'
2	Act4. Actividad expositiva. El profesor explica en qué consisten las cookies. - 100' Act2. Se propone que cada alumno individualmente investigue cómo trata el lenguaje PHP las cookies. - 80'	180'
3	Act6. Puesta en común de la información recopilada en la sesión 2. - 60' Act5. Ejercicio práctico sobre cookies. - 60'	120'
4	Act4. El profesor explicará toda la teoría relacionada con las sesiones. - 180'	180'
5	Act2. Ejercicio práctico sobre sesiones. El alumno deberá implementar una aplicación que guarde información del usuario en una sesión y luego sea capaz de recuperarla. – 180'	180'
6	Act4. El profesor explicará toda la teoría relacionada con la depuración de código. – 100' Act2. Se propone que cada alumno individualmente investigue las opciones de depuración de código de PHP. - 20'	120'
7	Act7. Prueba de evaluación práctica en clase que englobe todos los conceptos estudiados (autenticación, cookies, sesiones y depuración) - 150' Act3. Explicación de la actividad transversal, se dividirá a la clase en grupos de 3-4 alumnos para la realización de la aplicación web descrita en el apartado 4.7.4. Dicha actividad será realizada en casa por el grupo de alumnos - 30'	180'
8	Act3. Exposición de todas las aplicaciones creadas y elección de la que se va a implantar. - 180'	180'

2.8.2. Recursos.

Para la correcta realización y desarrollo de esta unidad didáctica se utilizarán los siguientes recursos:

Aula:

- Aula con capacidad de 20 alumnos con los equipos informáticos adecuados y los periféricos pertinentes.

Software:

- IDE NetBeans 8.0.1.
- IDE Visual Studio Code.
- Cualquier otro entorno de desarrollo que pueda proceder.
- XAMPP.
- MySQL Workbench.
- Navegador web.
- Xdebug para la depuración de PHP.

Otros recursos:

- Ordenador sobremesa con sistema operativo Windows o Ubuntu.
- Proyector para las clases expositivas.
- Pizarra para las clases expositivas.
- Conexión a internet.

2.8.3. Espacio.

El espacio donde se impartirá la unidad didáctica será en uno de los laboratorios de informática del que dispone el instituto. En dicho laboratorio nos encontramos con 20 posiciones, cada una de ellas con un ordenador sobremesa equipado con el software necesario.

También dispone de un espacio adaptado al lado de la entrada para el alumno con adaptación de acceso.

El plano del aula sería el que se muestra en la ilustración 12:



Ilustración 12: disposición del aula.

2.8.4. Bibliografía.

Para el desarrollo del contenido teórico-práctico que se va a dar a los alumnos, así como para la realización de la documentación de consulta de los mismos se han utilizado los siguientes recursos:

Libros:

- PHP 5 Power Programming. Andy Gutmans. Prentice-Hall.
- Desarrollo web con PHP y MySQL, Luke Welling y Laura Thomson. Anaya Multimedia.
- Introducción a Ajax con PHP. Lee Babin, Anaya.
- Desarrollo Web con PHP y MySQL, Anaya.
- Learning PHP, MySQL, JavaScript, CSS & HTML5: A Step-by-Step Guide to Creating Dynamic Websites.
- Apuntes del professor.

Páginas web:

- Web oficial de Java: <https://www.oracle.com/java/>.
- Manual de php: <https://www.php.net/manual/es/>.
- Moodle del centro:
<https://educacionadistancia.juntadeandalucia.es/centros/jaen/login/index.php>

2.9. Evaluación.

Una vez definidos los objetivos, los contenidos y la metodología vamos a proceder a definir como será evaluado el alumno, o lo que es lo mismo que criterios vamos a seguir para verificar que el alumno ha llegado a los objetivos propuestos.

Esto lo vamos a realizar mediante unos criterios e instrumentos de evaluación que se expondrán a continuación.

2.9.1. Criterios de evaluación.

En primer lugar, nos vamos a fijar en los criterios de evaluación que la 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 establece para este resultado de aprendizaje, estos serían los siguientes:

- 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.
- b) Se han utilizado sesiones para mantener el estado de las aplicaciones Web.
- c) Se han utilizado «cookies» para almacenar información en el cliente Web y para recuperar su contenido.
- d) Se han identificado y caracterizado los mecanismos disponibles para la autenticación de usuarios.
- e) Se han escrito aplicaciones que integren mecanismos de autenticación de usuarios.
- f) Se han realizado adaptaciones a aplicaciones Web existentes como gestores de contenidos u otras.
- g) Se han utilizado herramientas y entornos para facilitar la programación, prueba y depuración del código.

2.9.2. Instrumentos de evaluación.

Para la evaluación de la unidad didáctica se van a utilizar los siguientes instrumentos de evaluación:

- Ejercicios prácticos: la unidad didáctica es meramente práctica por lo que la mitad de la nota de la misma vendrá de uno o varios supuestos prácticos que se propondrán al alumno. Dichos supuestos deben de ser entregados con fecha máxima el día de la evaluación final.
- Observación directa: participación en clase, resolución de dudas, ayuda a compañeros etcétera. Supondrá el 10% de la nota final.
- Prueba práctica: al finalizar la UD se realizará una pequeña prueba teórico-práctica para verificar los conocimientos. Supondrá el 25% de la nota final.
- Trabajo en grupo: los alumnos van a realizar un trabajo en grupo como elemento transversal que supondrá un 15% de la nota de la evaluación.

La unidad didáctica se aprobará si la nota ponderada de todas las actividades e instrumentos es mayor o igual que 5. Para ello se seguirá una evaluación continua, basada en ejercicios prácticos realizados en clase una vez el profesor explica los contenidos teóricos.

Además, se utilizarán exposiciones de trabajos autónomos realizados por el alumnado.

2.9.3. Recuperación.

Si el alumno no ha conseguido superar satisfactoriamente los contenidos de la unidad mediante los medios descritos en el apartado 4.9.2. se dispondrá para la recuperación de una práctica guiada individual.

Dicha práctica será individual y personalizada para cada alumno y similar a los ejercicios que se han ido realizando en clase.

Englobará todos los contenidos, siendo el 100% de la nota y debiendo obtener una calificación mayor o igual que 5 para superarla.

2.9.4. Evaluación del proceso de enseñanza.

La evaluación de la práctica docente se realizará mediante un cuestionario pasado al alumno al final del trimestre.

Dicho cuestionario se adjunta en el ANEXO I.

2.10. Atención a la diversidad.

En la clase encontramos un alumno con movilidad reducida que necesita desplazamiento en silla de ruedas, la única adaptación necesaria será adaptación de acceso, pero aun así para la atención a la diversidad se trabajará siempre con contacto continuo con el departamento de orientación del centro.

2.10.1. Ritmos de aprendizaje.

Se consideran tres ritmos de aprendizaje distintos en el aula con las siguientes actividades para cada uno de ellos:

- Rápido: se facilitarán actividades de ampliación.
- Normal: solo se facilitarán actividades en caso necesario. Es el ritmo esperado en la clase.
- Lento: se facilitarán actividades guiadas individuales por parte del profesor.

2.10.2. ACNEAE.

Como se ha indicado anteriormente, en el aula tenemos un alumno con dificultades de acceso debido a la necesidad del uso de silla de ruedas.

La única adaptación necesaria es facilitarle la movilidad y acceso al aula y los recursos necesarios para ello:

- El alumno puede valerse por sí mismo por lo que solo se le va a prestar ayuda en caso de que lo necesite o pida expresamente.
- Dispondremos de una ubicación específica al final del laboratorio donde el alumno tenga libertad de movimiento para poder acceder al ordenador que va a necesitar durante el aula.
- Además, dicha ubicación se encuentra cercana a la zona de acceso y abandono del aula para facilitar esta labor al alumno.

2.10.3. Actividades de refuerzo y ampliación.

Durante el desarrollo del curso se facilitarán actividades de refuerzo para aquellos alumnos o alumnas que no hayan conseguido asimilar los contenidos, así mismo se facilitarán actividades de ampliación la clase cuando se considere necesario que se deban reforzar los conocimientos.

Dichas actividades siempre consistirán en actividades prácticas guiadas por el profesor e individuales.

2.11. Tabla resumen.

Desarrollo de aplicaciones web con PHP.							
N.º unidad	Etapas	Módulo		Curso	Trimestre	Temporalización	Sesiones
4	CFGS	DWEC		2º	2º	11/04/2021 a 27/04/2021	6 sesiones de 3 horas y 2 sesiones de 2 horas.
RA	OOGG	CPPS	% Anual	% Trim.	Recursos		
4	f, g, h, l, ñ, q, s, t	d, f, m, q	12%	35%	IDE NetBeans 8.0.1XAMPPMySQL Workbench		PC Proyector Pizarra
Orien. pedagógicas		Líneas de actuación					
a, b		b, d					

Contenidos	Objetivos didácticos	
Desarrollo de aplicaciones Web utilizando código embebido: - Mantenimiento del estado. Sesiones. Cookies. - Seguridad: usuarios, perfiles, roles. - Mecanismos de autenticación de usuarios. - Adaptación a aplicaciones Web: Gestores de contenidos y tiendas virtuales entre otras. - Pruebas y depuración. Herramientas y entornos.	1. Conocer por qué es importante la autenticación de usuarios, tipos de usuarios y tecnologías disponibles. 2. Ser capaz de aplicar una autenticación de usuarios sencilla a una aplicación web. 3. Implementar correctamente las cookies en una aplicación web.	4. Manejar y conocer de forma completa las sesiones en PHP, como manejarlas y sacarles el máximo partido. 5. Diferenciar distintas tecnologías de depuración de código en entorno servidor. 6. Aplicar una tecnología (Xdebug) de depuración de código de PHP.
	Temas transversales	Fomento de la lectura

Atención a la diversidad		ACNEAE
Actividades de refuerzo	Actividades de ampliación.	Adaptación de acceso.

Interdisciplinaridad
Bases de datos - Lenguajes de marcas - Programación - Entornos de desarrollo Desarrollo de interfaces web - Desarrollo web en entorno cliente

Evaluación				
Criterios de evaluación				Instrumentos de evaluación
RA 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.	15%	Act2. Act7. Act6.	Ejercicios prácticos - 25% Observación directa - 10% Prueba práctica - 40% Trabajo en grupo - 40%
	b) Se han utilizado sesiones para mantener el estado de las aplicaciones Web.	15%	Act2. Act5.	
	c) Se han utilizado «cookies» para almacenar información en el cliente Web y para recuperar su contenido.	15%	Act2. Act3. Act7.	
	d) Se han identificado y caracterizado los mecanismos disponibles para la autenticación de usuarios.	15%	Act3.Act7.	
	e) Se han escrito aplicaciones que integren mecanismos de autenticación de usuarios.	15%	Act3. Act7.	
	f) Se han realizado adaptaciones a aplicaciones Web existentes como gestores de contenidos u otras.	10%	Act2. Act7.	
	g) Se han utilizado herramientas y entornos para facilitar la programación, prueba y depuración del código.	15%	Act2. Act7.	

3. BIBLIOGRAFÍA.

Real Decreto 686/2010, de 20 de mayo, por el que se establece el título de Técnico Superior en Desarrollo de Aplicaciones Web y se fijan sus enseñanzas mínimas.

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

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.

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

Ley Orgánica 8/2013, de 9 de diciembre, para la mejora de la calidad educativa.

Ley 17/2007, de 10 de diciembre, de Educación de Andalucía.

Lucas , J. (2017). <https://openwebinars.net>. Obtenido de <https://openwebinars.net:https://openwebinars.net/blog/8-datos-sorprendentes-sobre-el-uso-de-docker/>

Alegasa, L. (08 de 02 de 2016). <http://www.alegsa.com.ar>. Obtenido de Definición de backend: <http://www.alegsa.com.ar/Dic/back-end.php>

Altube Vera, R. (2021). Qué es Laravel: Características y ventajas.

Angular. (2022). <https://angular.io/>. Obtenido de <https://angular.io:https://angular.io/>

Arias, Á. (2015). *Aprende a Programar ASP .NET y C#*.

Bernstein, D. (2014). Containers and Cloud: From LXC to Docker to Kubernetes. *IEE*.

Cedeño, A. (03 de 05 de 2020). <https://blogueropro.com>. Obtenido de <https://blogueropro.com/blog/ventajas-y-desventajas-de-usar-asp-net-en-la-programaci%C3%B3n-web>

Cedeño, A. (2020). Ventajas y desventajas de usar Ruby en la programación web. *Desarrollo web*.

Chnar , M., & Subhi , R. (2021). Sufficient Comparison Among Cloud Computing Services: IaaS, PaaS, and SaaS: A Review. *International Journal of Science and Business*.

digital55. (26 de 04 de 2019). <https://digital55.com>. Obtenido de <https://digital55.com:https://digital55.com/ruby-on-rails-desarrollo-software/>

Dueñas Lerin, J. (2020). <https://duenaslerin.com/>. Obtenido de <https://duenaslerin.com/:https://duenaslerin.com/pmdm/acceso-a-datos/02-acceso-api-rest-json/>

- Evans, S. (25 de 05 de 2021). <https://digitalcommunications.wp.st-andrews.ac.uk/2021/05/25/php-8-0-whats-new/>. Obtenido de PHP 8.0 – what’s new: <https://digitalcommunications.wp.st-andrews.ac.uk/2021/05/25/php-8-0-whats-new/>
- Flanagan, D., & Matsumoto, Y. (2008). *The Ruby Programming Language: Everything You Need to Know*. O`REILLY.
- Guérin, B.-A. (2017). *ASP.NET con C# en Visual Studio*. Ediciones ENI.
- <https://laravel.com/>. (2022). <https://laravel.com/>. Obtenido de <https://laravel.com/>: <https://laravel.com/>
- Jaime González, C. (2015). Programación de web estático.
- Json.org. (2022). <http://www.json.org/>. Obtenido de <http://www.json.org/>: <http://www.json.org/json-es.html>
- Kaluža, M., Kalanj, M., & Vukelić, B. (2019). A Comparison of Back-End Frameworks for Web Application Development.
- KeepCoding. (13 de 01 de 2020). <https://keepcoding.io>. Obtenido de <https://keepcoding.io/blog/lenguaje-de-programacion-go-caracteristicas/>
- MDN contributors. (11 de 02 de 2021). <https://developer.mozilla.org>. Obtenido de <https://developer.mozilla.org>: <https://developer.mozilla.org/es/docs/Learn/Server-side/Django/Introduction>
- MDN contributors. (03 de 06 de 2022). <https://developer.mozilla.org>. Obtenido de <https://developer.mozilla.org>: https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/What_is_JavaScript
- Molina Ríos , J., Loja Mora , N., Zea Ordóñez, M., & Loaiza Sojos , E. (2016). Evaluación de los Frameworks en el Desarrollo de Aplicaciones Web con Python.
- Molina, T. (26 de 09 de 2018). <https://medium.com/>. Obtenido de <https://medium.com/>: <https://medium.com/angular-chile/angular-arquitectura-del-framework-e46204f38fef>
- Muradas , Y. (2018). Qué es Spring Framework y por qué usarlo.
- Nubecolectiva. (29 de 11 de 2021). <https://blog.nubecolectiva.com>. Obtenido de <https://blog.nubecolectiva.com>: <https://blog.nubecolectiva.com/5-tendencias-que-se-daran-en-el-desarrollo-backend-para-el-2022/>
- Pareja Valerio, C., & Burgos Robles, L. (2019). La arquitectura de software basada en microservicios: Una revisión sistemática de la literatura.

- Pérez Ibarra, S., Quispe, J., Mullicundo, F., & Lamas, D. (2020). Herramientas y tecnologías para el desarrollo web desde el frontend al backend.
- Raible, M. (2018). <https://raibledesigns.com/rd/>. Obtenido de <https://raibledesigns.com/rd/>: <https://raibledesigns.com/rd/>
- Rana, M. (27 de 12 de 2021). <https://www.europeanbusinessreview.com>. Obtenido de <https://www.europeanbusinessreview.com>: <https://www.europeanbusinessreview.com/top-8-backend-languages-that-will-make-wave-in-2022/>
- Sidorenko, I. (26 de 02 de 2019). <https://hackernoon.com>. Obtenido de <https://hackernoon.com>: <https://hackernoon.com/should-i-go-the-pros-and-cons-of-using-go-programming-language-8c1daf711e46>
- Sierra Sedeño, A., & Espinoza Mina, M. (2018). Análisis comparativo entre ASP.NET y PHP. *INNOVA Research Journal*, ISSN 2477.
- Themerkraft. (2021). <https://themekraft.com>. Obtenido de 5 Hottest Trends of PHP Development in 2021: <https://themekraft.com/5-hottest-trends-of-php-development-in-2021/>
- Tokio School . (25 de 03 de 2020). <https://www.tokioschool.com/>. Obtenido de <https://www.tokioschool.com/>: <https://www.tokioschool.com/noticias/7-razones-usar-aplicaciones-java/>
- Tomassetti, G. (21 de 11 de 2018). <https://tomassetti.me/code-generation/>. Obtenido de <https://tomassetti.me/code-generation/>: <https://tomassetti.me/code-generation/>
- Universidad de Alicante. (2012). *Introducción a Java EE*.
- W3techs. (25 de 01 de 2022). <https://w3techs.com/>. Obtenido de Server-side Programming Languages Market Report: https://w3techs.com/technologies/report/programming_language
- Waden, J., & Doyle, M. (2010). Java vs. PHP: Security Implications of Language Choice for Web Applications.
- www.php.net. (2021). <https://www.php.net>. Obtenido de News Archive - 2021: <https://www.php.net/archive/2021.php#2021-12-17-1>

ANEXO I. Evaluación del proceso de enseñanza.

Nombre Profesor/a:				Curso:	
Grupo:		Materia:		Departamento:	

1. Evaluación: valore del 0 al 10 su grado de acuerdo siendo 0 totalmente en desacuerdo y 10 totalmente de acuerdo.

	0	1	2	3	4	5	6	7	8	9	10	NS/NC
Conozco los objetivos de la materia.												
Coordinación entre teoría y práctica.												
El volumen de trabajo ha sido el adecuado.												
Las clases han estado bien organizadas.												
La metodología ha sido la adecuada.												
Los recursos han sido los adecuados.												
Los criterios de evaluación han sido los adecuados.												
El tiempo empleado ha sido el adecuado.												
Me ha resultado fácil seguir el curso.												
La materia ha permitido la interacción de los padres en la misma.												
He podido realizar aportaciones.												
El clima ha sido el adecuado.												
El profesor dominaba la materia.												

2. Valoración global: valore del 0 al 10 su grado de acuerdo siendo 0 totalmente en desacuerdo y 10 totalmente de acuerdo.

[illegible]

3. Sugerencias: sugerencias para mejorar la asignatura.

--