

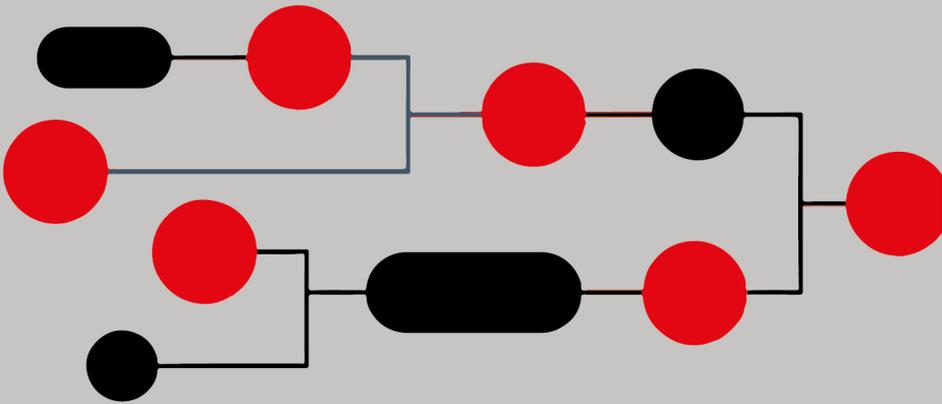


Religación  
Press  
Ideas desde el Sur Global

int Instituto  
Nelson  
Torres

# DESARROLLO DE APLICACIONES WEB

CON TECNOLOGÍAS MODERNAS



WILSON ALEXIS MÁRQUEZ COCA  
CARMEN VERÓNICA VALENZUELA CHICAIZA  
CARLOS ANDRÉS ACOSTA JARAMILLO  
NANCY MARICELA GAÓN ROJAS  
JUAN DAVID CHIMARRO AMAGUAÑA



| Colección Programación |

# **Desarrollo de aplicaciones web con tecnologías modernas**

Wilson Alexis Márquez Coca, Carmen Verónica Valenzuela Chicaiza,  
Carlos Andrés Acosta Jaramillo, Nancy Maricela Gaón Rojas,  
Juan David Chimarro Amaguaña

RELIGACION PRESS  
QUITO · 2023



### **Equipo Editorial**

Eduardo Díaz R. Editor Jefe  
Roberto Simbaña Q. Director Editorial  
Felipe Carrión. Director de Comunicación  
Ana Benalcázar. Coordinadora Editorial  
Ana Wagner. Asistente Editorial

### **Consejo Editorial**

Jean-Arsène Yao | Dilrabo Keldiyorovna Bakhronova | Fabiana Parra |  
Mateus Gamba Torres | Siti Mistima Maat | Nikoleta Zampaki | Silvina  
Sosa



**Religación Press**, es una iniciativa del Centro de Investigaciones CICSHAL-  
RELIGACIÓN.

Diseño, diagramación y portada: Religación Press.  
CP 170515, Quito, Ecuador. América del Sur.  
Correo electrónico: [press@religacion.com](mailto:press@religacion.com)  
[www.religacion.com](http://www.religacion.com)

# Desarrollo de aplicaciones web con tecnologías modernas

*Development of web applications with modern technologies*

*Desenvolvimento de aplicativos da Web com tecnologias modernas*

---

**Derechos de autor:** Wilson Alexis Márquez Coca©, Carmen Verónica Valenzuela Chicaiza©, Carlos Andrés Acosta Jaramillo©, Nancy Maricela Gaón Rojas©, Juan David Chimarro Amaguaña©, Religación Press©

Primera Edición: 2023

Editorial: Religación Press

Materia Dewey: 005 - Programación. programas. datos de computadores

Clasificación Thema: UMW - Programación web

UBW - Internet: obras generales

BISCA: COM051000 COMPUTERS / Programming / General

Público objetivo: Profesional/Académico

Colección: Programación

Soporte: Digital

Formato: Epub (.epub)/PDF (.pdf)

Publicado: 2023-12-07

ISBN: 978-9942-642-42-4

Este título se publica bajo una licencia de Atribución 4.0 Internacional (CC BY 4.0)



## Citar como (APA 7)

Márquez Coca, W.A., Valenzuela Chicaiza, C.V., Acosta Jaramillo, C.A., Gaón Rojas, N.M., y Chimarro Amaguaña, J.D. (2023). *Desarrollo de aplicaciones web con tecnologías modernas*. Religación Press. <https://doi.org/10.46652/ReligacionPress.108>

## Patrocinio:

Instituto Superior Tecnológico Nelson Torres. [Avenida Luis Cordero, Vía a Ayora. Cayambe, Ecuador].

ISBN: 978-9942-642-42-4



9 789942 642424

<https://press.religacion.com>



## **Revisión por pares / Peer Review**

Este libro fue sometido a un proceso de dictaminación por académicos externos. Por lo tanto, la investigación contenida en este libro cuenta con el aval de expertos en el tema, quienes han emitido un juicio objetivo del mismo, siguiendo criterios de índole científica para valorar la solidez académica del trabajo.

This book was reviewed by an independent external reviewers. Therefore, the research contained in this book has the endorsement of experts on the subject, who have issued an objective judgment of it, following scientific criteria to assess the academic soundness of the work.

## Sobre los autores/as



### **Wilson Alexis Márquez Coca**

Instituto Tecnológico Superior Nelson Torres | Cayambe | Ecuador  
<https://orcid.org/0000-0001-7923-9417>  
[alexis.marquez@intsuperior.edu.ec](mailto:alexis.marquez@intsuperior.edu.ec)

Ingeniero en Sistemas e Informática y con una sólida experiencia profesional, Jefe Departamental de Telemática en la Sede Ibarra de la Universidad Uniandes durante 5 años. Docente en el Instituto Superior Tecnológico Nelson Torres.



### **Carmen Verónica Valenzuela Chicaiza**

Instituto Tecnológico Superior Nelson Torres | Cayambe | Ecuador  
<https://orcid.org/0000-0002-3267-7273>  
[carmen.valenzuela@intsuperior.edu.ec](mailto:carmen.valenzuela@intsuperior.edu.ec)

Ingeniera en Economía, Mención finanzas, posee una maestría en Gestión de la Responsabilidad Social Corporativa, se ha desempeñado como docente de educación superior en varias universidades e institutos del Ecuador, ejerce la actividad de consultor independiente mediante la elaboración de proyectos y asesorías



### **Carlos Andrés Acosta Jaramillo**

Instituto Tecnológico Superior Nelson Torres | Cayambe | Ecuador  
<https://orcid.org/0000-0003-4336-259X>  
[carlos.acosta@intsuperior.edu.ec](mailto:carlos.acosta@intsuperior.edu.ec)

Docente Investigador en el área de Desarrollo de Software con una Maestría en Diseño y Gestión de proyectos Tecnológicos, cursando una Maestría en Ingeniería Matemática y Computación, y cursando un Doctorado en Educación, de profesión Ingeniero en Mecatrónica.



### **Nancy Maricela Gaón Rojas**

Instituto Tecnológico Superior Nelson Torres | Cayambe | Ecuador  
<https://orcid.org/0009-0001-0065-9871>  
[nancy.gaon@intsuperior.edu.ec](mailto:nancy.gaon@intsuperior.edu.ec)

Ingeniera en Administración de Empresas y Marketing, actualmente se encuentra culminando una Maestría en Administración y Dirección de Empresas, ha contribuido de manera significativa en el ámbito administrativo, desempeñando roles cruciales en el Gobierno Autónomo Descentralizado (GAD) de Montúfar.



### **Juan David Chamarro Amaguaña**

Instituto Tecnológico Superior Nelson Torres | Cayambe | Ecuador  
<https://orcid.org/0000-0001-9454-8357>  
[juan.chamarro@intsuperior.edu.ec](mailto:juan.chamarro@intsuperior.edu.ec)

Docente Investigador en el área de Desarrollo de Software con una Maestría en Telemática e Ingeniería en Mecatrónica, cursando un Doctorado en Educación con sólida y contrastada experiencia en el manejo de multiplataformas de desarrollo de software y lenguajes de programación, desarrollo de aplicaciones móviles en tiempo real, computación en la nube (Cloud Computing).

## Resumen

El libro abarca una serie de contenidos cruciales para el entendimiento y dominio de la programación web. Se exploran los fundamentos del desarrollo web, desde los conceptos generales hasta la publicación de sitios. Los estudiantes se adentran en la dinámica de los gestores de contenidos, comprenden los lenguajes de marcado y presentación (HTML, CSS), y se sumergen en la programación tanto del lado del cliente como del servidor, utilizando JavaScript y PHP. La interacción con bases de datos se aborda a través de SQL, mientras que los frameworks de desarrollo web se exploran en términos de ventajas y aplicaciones. La seguridad web se aborda desde las generalidades hasta herramientas de identificación de vulnerabilidades. Los estudiantes aprenden los pasos esenciales para la publicación de sitios web, abarcando elección de hosting, configuración de servidores y dominios. Cada contenido se complementa con actividades prácticas, que van desde la creación de sitios web simples hasta la simulación de pruebas de seguridad, preparando a los estudiantes para enfrentar los desafíos reales del desarrollo y despliegue web con solidez y confianza.

Palabras claves: desarrollo web, html, php, javascript.

## Abstract

The book covers a range of content crucial to understanding and mastering web programming. The fundamentals of web development are explored, from general concepts to site publishing. Students delve into the dynamics of content managers, understand markup and presentation languages (HTML, CSS), and dive into both client-side and server-side programming using JavaScript and PHP. Database interaction is addressed through SQL, while web development frameworks are explored in terms of benefits and applications. Web security is covered from generalities to vulnerability identification tools. Students learn the essential steps for web site publishing, covering choice of hosting, server configuration and domains. Each content is complemented with hands-on activities, ranging from creating simple websites to simulating security testing, preparing students to face the real challenges of web development and deployment with confidence.

Keywords: web development, html, php, javascript.

## Resumo

O livro abrange uma série de conteúdos cruciais para a compreensão e o domínio da programação da Web. São explorados os fundamentos do desenvolvimento da Web, desde os conceitos gerais até a publicação de sites. Os alunos se aprofundam na dinâmica dos sistemas de gerenciamento de conteúdo, compreendem as linguagens de marcação e apresentação (HTML, CSS) e mergulham na programação do lado do cliente e do lado do servidor usando JavaScript e PHP. A interação com o banco de dados é abordada por meio de SQL, enquanto as estruturas de desenvolvimento da Web são exploradas em termos de benefícios e aplicativos. A segurança na Web é abordada desde as generalidades até as ferramentas de identificação de vulnerabilidades. Os alunos aprendem as etapas essenciais para a publicação de sites, incluindo a escolha da hospedagem, a configuração do servidor e os domínios. Cada conteúdo é complementado por atividades práticas, que vão desde a criação de sites simples até a simulação de testes de segurança, preparando os alunos para enfrentar com confiança os desafios reais do desenvolvimento e da implantação da Web.

Palavras-chave: desenvolvimento web, html, php, javascript.

## Contenido

Revisión por pares / Peer Review	7
Sobre los autores	8
Resumen	9
Abstract	9
Resumo	9
Prólogo	19
Introducción	20
<b>Capítulo I</b>	
<b>Conceptos generales de desarrollo web</b>	<b>25</b>
Introducción al Desarrollo Web	28
Software para trabajar en desarrollo web	34
HTML	35
Etiquetas, DOM	37
Estructura de la Página Web	42
Etiqueta:	43
Atributo:	43
Valor:	43
Contenido:	43
Estructura del Sitio Web	44
Cabecera	46
Logotipo	47
Zonas de navegación	47
Buscador	48
Columnas laterales	48
Área de contenido principal	49
Pie de página	49
Formas de agregar estilos a HTML	49
Incluir CSS: estilo inline	50
Añadir CSS al principio del HTML	51
Integrar el CSS con un archivo externo	52
Atributos HTML	53
Atributos universales clásicos	57
Atributos HTML específicos	58
Nuevos atributos desde HTML5	59
Atributos de evento	60
Navegación entre secciones	61
El elemento de sección de navegación NAV	62
Reglas	63
Selectores	64
Combinación de Selectores:	65
Declaraciones, prioridades	66

Prioridades en CSS:	67
Los estilos incluidos por el navegador	68
Agregando clases a los componentes escritos en HTML	69
Gestores de Contenidos	70
Inspector de elementos	71
Lenguajes de marcado y de presentación. (HTML, CSS)	72
Estilos, CSS	72
Estilos en Cascada (CSS)	73
Propiedades CSS	74
Tipos de textos y Propiedades	77
Tipos de textos	77
Propiedades de texto	77
Tamaño de texto	78
Color de texto	78
Peso del texto	79
Estilo de texto	79
Dimensiones fijas para elementos	80
Backgrounds de color e imagen	82
Background de color en CSS:	82
Background de imagen en CSS:	83
Propiedades adicionales de background en CSS:	83
Background de gradiente en CSS:	84
Uso en diseño web:	84
Integración con HTML:	84
Lenguaje de marcado y de presentación:	85
Unidades de medida y colores	85
Unidades de Medida:	85
Colores:	86
Bordes, Margenes Rellenos	87
Bordes (CSS):	87
Márgenes (CSS):	88
Rellenos (CSS):	89
Uso en Diseño Web:	89
Combinación con Otros Estilos:	89
Conclusión	90
Autoevaluación	92

## **Capítulo 2**

<b>Programación del lado del cliente y del servidor</b>	<b>99</b>
Programación del Lado del Cliente:	100
Programación del Lado del Servidor:	101
Interacción Cliente-Servidor:	101
Arquitectura de aplicaciones multinivel	102
Tres Capas Fundamentales:	102

Beneficios:	103
Ejemplo en Aplicaciones Web:	104
PHP	104
Características Clave de PHP:	105
Usos Comunes de PHP en Aplicaciones Web:	105
Frameworks de PHP:	107
Integración PHP con HTML	107
Almacén de datos PHP	110
Interacción con bases de datos.	112
Operaciones Principales:	112
SELECT	114
INSERT (inserción)	114
DELETE (eliminación)	115
UPDATE (actualización)	115
Desarrollo Aplicación WEB con base de Datos	115
Consideraciones Importantes:	117
Conclusión	118
Autoevaluación	120

### **Capítulo 3**

#### **Frameworks de desarrollo**

**126**

Definiciones	127
Ruby on Rails:	127
Django:	127
Laravel:	128
Express.js:	128
Spring Boot:	128
Angular:	128
React:	129
Vue.js:	129
ASP.NET Core:	129
Flask:	130
Ventajas y Desventajas	131
Generalidades a la seguridad Web.	133
Herramientas de aplicación de seguridad Web	136
Open Web Application Security Project (OWASP):	136
Conclusión	138
Autoevaluación	140

### **Capítulo 4**

#### **Herramientas de identificación de vulnerabilidades**

**148**

Aspectos generales	149
Herramientas de aplicación	151
Tipos de Vulnerabilidades Detectadas:	151

Análisis de Código y Red:	152
Aplicaciones para verificar la accesibilidad de sitios Web	153
Publicación de sitios web.	156
Trabajo con la gestión de Proyectos	158
Seguimiento del Desarrollo de Proyectos	161
Conclusión	163
Autoevaluación	165

## **Referencias** **172**

## **Anexos** **176**

Anexo 1 Solucionario Unidad 1	177
Anexo 2. Solucionario Unidad 2	183
Anexo 3. Solucionario Unidad 3	188
Anexo 4. Solucionario Unidad 4	194

## Tablas

Tabla 1. Lista de etiquetas HTML.	37
Tabla 2. Atributos universales clásicos.	57
Tabla 3. Atributos HTML específicos.	58
Tabla 4. Nuevos atributos desde HTML5.	59
Tabla 5. Atributos de evento.	60

## Figuras

Figura 1. Título e identificación del sitio.	45
Figura 2. Cabecera.	46
Figura 3. Zona de navegación.	47
Figura 4. Padding.	56



| Colección Programación |

# **Desarrollo de aplicaciones web con tecnologías modernas**



## Prólogo

El presente libro de Programación de Aplicaciones Web con Tecnologías Modernas, contiene los fundamentos de la programación web, desde los aspectos básicos de lenguajes de marcado como HTML y CSS, responsables de la estructura y estética de las páginas, hasta los lenguajes de programación más dinámicos, como JavaScript y PHP, que infunden interacción y funcionalidad avanzada. Se aborda conceptos cruciales como el diseño moderno, garantizando una experiencia óptima en dispositivos de variados formatos, y conjuntamente trabajar con bases de datos, pilar del almacenamiento de información. Además, se sustentará las complejidades de la seguridad en línea, preservando tanto al usuario como a los datos sensibles de amenazas digitales.

Los objetivos de esta guía didáctica son de un alcance integral y específico. Enmarcados en el paradigma de la programación web, se direccionan hacia el desarrollo de aplicaciones web, abordando tanto la perspectiva del cliente como la del servidor. Para alcanzar esta finalidad, se recurre a la aplicación de lenguajes de programación web y la incorporación de algoritmos que abarcan un espectro amplio de funcionalidades, incluyendo búsqueda, ordenamiento, cálculos numéricos, análisis estadísticos y procedimientos criptográficos.

## Introducción

El presente libro de Programación de aplicaciones Web, contiene los fundamentos de la programación web, desde los aspectos básicos de lenguajes de marcado como HTML y CSS, responsables de la estructura y estética de las páginas, hasta los lenguajes de programación más dinámicos, como JavaScript y PHP, que infunden interacción y funcionalidad avanzada. Se aborda conceptos cruciales como el diseño moderno, garantizando una experiencia óptima en dispositivos de variados formatos, y conjuntamente trabajar con bases de datos, pilar del almacenamiento de información. Además, se sustentará las complejidades de la seguridad en línea, preservando tanto al usuario como a los datos sensibles de amenazas digitales.

Los objetivos de este material son de un alcance integral y específico. Enmarcados en el paradigma de la programación web, se direccionan hacia el desarrollo de aplicaciones web, abordando tanto la perspectiva del cliente como la del servidor. Para alcanzar esta finalidad, se recurre a la aplicación de lenguajes de programación web y la incorporación de algoritmos que abarcan un espectro amplio de funcionalidades, incluyendo búsqueda, ordenamiento, cálculos numéricos, análisis estadísticos y procedimientos criptográficos.

Este trabajo abarca una serie de contenidos cruciales para el entendimiento y dominio de la programación web. Se exploran los fundamentos del desarrollo web, desde los conceptos generales hasta la publicación de sitios. Los estudiantes se adentran en

la dinámica de los gestores de contenidos, comprenden los lenguajes de marcado y presentación (HTML, CSS), y se sumergen en la programación tanto del lado del cliente como del servidor, utilizando JavaScript y PHP. La interacción con bases de datos se aborda a través de SQL, mientras que los frameworks de desarrollo web se exploran en términos de ventajas y aplicaciones. La seguridad web se aborda desde las generalidades hasta herramientas de identificación de vulnerabilidades.

Los estudiantes aprenden los pasos esenciales para la publicación de sitios web, abarcando elección de hosting, configuración de servidores y dominios. Cada contenido se complementa con actividades prácticas, que van desde la creación de sitios web simples hasta la simulación de pruebas de seguridad, preparando a los estudiantes para enfrentar los desafíos reales del desarrollo y despliegue web con solidez y confianza.

Las actividades propuestas en esta asignatura utilizan varios recursos tecnológicos esenciales para garantizar un aprendizaje efectivo y una ejecución exitosa. Las herramientas que se han seleccionado para este propósito son:

Un computador personal, primordial para llevar a cabo las actividades propuestas, desde la escritura de código hasta la visualización de resultados, esta herramienta es la base sobre la cual los estudiantes interactúan con los conceptos y ejercicios de programación web. Un equipo con suficiente capacidad de procesamiento y memoria es preferible para garantizar una experiencia de aprendizaje fluida.

Editores de Código (Brackets, Sublime Text, Notepad++): Estos editores de código son pilares fundamentales para el estudiante de programación web. Brackets ofrece un entorno diseñado específicamente para desarrollo web, con características como vista en vivo y resaltado de sintaxis. Sublime Text es conocido por su velocidad y potencia, mientras que Notepad++ es apreciado por su simplicidad y versatilidad. Estas herramientas proporcionan interfaces intuitivas y facilitan la escritura, edición y depuración de código HTML, CSS y JavaScript.

### Objetivos

Desarrollar aplicaciones web tanto del lado del cliente como del servidor utilizando lenguajes de programación web y aplicando algoritmos de búsqueda, ordenamiento, numéricos, estadísticos y criptográficos.

### Orientaciones básicas para el estudio.

- **Planificación del Tiempo:** Establece un horario de estudio regular que te permita dedicar tiempo a la asignatura. Divide tu tiempo de estudio en sesiones más cortas y enfócate en tareas específicas para evitar la procrastinación.
- **Material de Estudio:** Familiarízate con los materiales de estudio proporcionados, como libros de texto, recursos en línea y material adicional proporcionado por tu docente. Utiliza recursos de calidad para reforzar tu comprensión de los conceptos.

- **Práctica Continua:** La programación web es una habilidad que se desarrolla a través de la práctica constante. Realiza ejercicios y proyectos para aplicar los conceptos aprendidos.
- **Investigación Adicional:** No dudes en investigar por tu cuenta. La web está llena de recursos y tutoriales útiles que pueden complementar tu aprendizaje.



# Capítulo I

## Conceptos generales de desarrollo web

A lo largo de esta sección, los educandos serán introducidos a los pilares fundamentales que definen la creación de sitios y aplicaciones web. Desde la comprensión de protocolos y tecnologías subyacentes que permiten la comunicación en línea, hasta la apreciación de las etapas de diseño, desarrollo y despliegue de un proyecto web, esta sección establece el marco en el que los estudiantes desarrollarán sus habilidades. Los conceptos de cliente y servidor, junto con el flujo de solicitud y respuesta, se exploran para proporcionar una comprensión sólida de cómo funcionan las interacciones web. Además, se expondrá la importancia de la usabilidad, la accesibilidad y la optimización, sentando las bases para la creación de experiencias web efectivas y eficientes.

El análisis del desarrollo web implica una evaluación completa de todos los aspectos relacionados con un sitio web, desde su diseño y contenido hasta su funcionamiento técnico y su rendimiento en línea. Un enfoque cuidadoso en cada uno de estos elementos es esencial para crear un sitio web exitoso y efectivo en la era digital.

De acuerdo a Molina et al., (2018), se deben tomar en cuenta los siguientes aspectos en el desarrollo web:

- **Objetivos del sitio web:** Antes de comenzar el desarrollo, es crucial comprender los objetivos del sitio web. ¿Es un sitio informativo, una tienda en línea, un blog, una plataforma de redes sociales? Definir claramente estos objetivos es fundamental.

- **Diseño y experiencia del usuario (UX):** El diseño del sitio web debe ser atractivo, intuitivo y centrado en el usuario. Esto implica considerar la navegación, la disposición de elementos, la legibilidad del contenido y la optimización para dispositivos móviles.
- **Arquitectura de la información:** Organizar la información de manera lógica y jerárquica es esencial para que los usuarios encuentren fácilmente lo que buscan. Esto incluye la estructura de menús, categorías y etiquetas.
- **Desarrollo técnico:** Aquí se evalúa la tecnología utilizada para construir el sitio web. Debe ser compatible con los navegadores, rápido en la carga y seguro. También es importante garantizar que el sitio sea accesible para personas con discapacidades.
- **SEO (Optimización de motores de búsqueda):** Para que el sitio web sea visible en los motores de búsqueda, se deben aplicar buenas prácticas de SEO, como la optimización de palabras clave, la creación de contenido de calidad y la optimización de metaetiquetas.
- **Contenido:** El contenido del sitio web debe ser relevante, actualizado y de alta calidad. Esto incluye texto, imágenes, videos y otros elementos multimedia. También se deben considerar las estrategias de marketing de contenido.

- **Seguridad:** La seguridad en línea es fundamental. Se deben tomar medidas para proteger el sitio web contra ataques cibernéticos, como la instalación de certificados SSL y la actualización regular de software.
- **Pruebas y control de calidad:** Antes de lanzar el sitio web, es necesario realizar pruebas exhaustivas para identificar y solucionar posibles errores y problemas de usabilidad.
- **Analítica web:** Una vez en funcionamiento, se deben implementar herramientas de análisis web, como Google Analytics, para realizar un seguimiento del rendimiento del sitio, como el tráfico, la conversión y el comportamiento del usuario.
- **Mantenimiento y actualización:** El desarrollo web es un proceso continuo. El sitio debe mantenerse actualizado con contenido fresco y tecnología actualizada. Esto incluye correcciones de errores, actualizaciones de seguridad y mejoras periódicas.

## Introducción al Desarrollo Web

La sección introductoria al desarrollo web constituye el punto de partida esencial en esta guía didáctica. Aquí, los estudiantes se adentran en el panorama global de la creación digital, explorando los fundamentos que conforman el tejido de la web moderna. Desde conceptos actuales, los educandos se familiarizan con

elementos clave como las URL y los navegadores, comprendiendo cómo interactúan para proporcionar la experiencia en línea que damos por sentada. Este módulo sienta las bases para el viaje de aprendizaje al destacar la evolución del desarrollo web y exponer a los estudiantes a los elementos esenciales que conforman la infraestructura de la internet.

El análisis del desarrollo web, centrado en el FrontEnd, se enfoca en las tecnologías y responsabilidades asociadas a esta parte esencial del proceso de creación de sitios web.

A continuación se desglosan algunos puntos clave según (Pérez et al., 2021):

- **FrontEnd y su Función:** El FrontEnd, en el contexto del desarrollo web, abarca todas las tecnologías que operan en el lado del cliente, es decir, en el navegador web. Estas tecnologías incluyen principalmente HTML, CSS y JavaScript. Los profesionales del FrontEnd trabajan con estas tres tecnologías, aunque también deben comprender el funcionamiento del BackEnd (lado del servidor), ya que esta comprensión es crucial para consumir datos y estructurar maquetas en HTML y CSS de manera efectiva.
- **Estilización y Presentación de la Página:** El FrontEnd se encarga de dar estilo a la página web de modo que esta pueda mostrar la información de manera atractiva y funcional para el usuario. Los desarrolladores FrontEnd deben dominar técnicas de experiencia de usuario

(UX) para mejorar la interacción usuario-página y conocimientos de diseño de interacción para organizar los elementos de forma que el usuario pueda encontrarlos de manera rápida y cómoda.

- **Tecnologías Relacionadas:** Además de HTML, CSS y JavaScript, existen muchas otras tecnologías relacionadas que los desarrolladores FrontEnd deben conocer. Por ejemplo, para JavaScript, hay frameworks como Angular y BackboneJS, que pueden aprovechar bibliotecas como AnimateCSS y jQuery. También se utilizan lenguajes de transferencia de datos como XML, JSON y Ajax para realizar solicitudes al servidor sin recargar la página completa.
- **Mejora de la Experiencia del Usuario:** La labor del FrontEnd va más allá de la presentación visual; también se trata de mejorar la experiencia del usuario. Esto implica diseñar interfaces intuitivas, optimizar la velocidad de carga de la página y garantizar que el sitio sea responsivo (adaptable a diferentes dispositivos y tamaños de pantalla).
- **Interacción y Comunicación:** El FrontEnd también se encarga de gestionar la interacción entre el usuario y el sitio web. Esto incluye la validación de formularios, la manipulación de eventos y la comunicación con el BackEnd para obtener o enviar datos.

- **Evolución Constante:** El mundo del desarrollo web FrontEnd está en constante evolución. Los profesionales deben mantenerse al tanto de las últimas tendencias y tecnologías, como las bibliotecas de componentes y los marcos de trabajo que facilitan el desarrollo de aplicaciones web complejas.
- Dentro de esta sección, se alienta a los educandos a investigar y cuestionar, permitiendo una comprensión más profunda de los componentes subyacentes de la web. A medida que los estudiantes se sumergen en los conceptos generales de desarrollo web, pueden analizar cómo los navegadores interpretan el código, cómo se accede a diferentes recursos y cómo los protocolos guían las interacciones en línea. La introducción al desarrollo web se revela como un fascinante viaje a través del tiempo y la tecnología, que prepara el terreno para los temas más específicos que se abordarán posteriormente en la guía.
- El desarrollo web desempeña un papel de suma importancia en la actual era digital y engloba diversos aspectos cruciales que ejercen influencia en múltiples ámbitos.
- A continuación, se destacan algunos puntos fundamentales para comprender su relevancia tomando como referencia a (Casale, 2016):
- **Presencia en línea:** En un mundo cada vez más digitalizado, la presencia en línea se vuelve imprescindible para empresas, organizaciones e incluso particulares. El de-

sarrollo web posibilita la creación y el mantenimiento de sitios web que actúan como tarjetas de presentación digitales, simplificando la conexión con el público objetivo y posibles clientes.

- **Accesibilidad y Disponibilidad:** Un sitio web debidamente desarrollado garantiza que la información y los servicios estén accesibles las 24 horas del día, los 7 días de la semana, desde cualquier ubicación. Esto optimiza la accesibilidad y la comodidad para los usuarios.
- **Comunicación y Marketing:** La web constituye un canal altamente efectivo para la comunicación con la audiencia y la promoción de productos, servicios o ideas. A través del desarrollo web, es posible implementar estrategias de marketing digital, como la Optimización de Motores de Búsqueda (SEO) y el Marketing en Motores de Búsqueda (SEM), para incrementar la visibilidad y atraer tráfico.
- **Interactividad y Experiencia del Usuario:** Los sitios web que han sido diseñados y desarrollados con destreza ofrecen una experiencia interactiva atractiva para los usuarios. Elementos como animaciones, formularios dinámicos y una navegación intuitiva mejoran la interacción y la retención de los visitantes.
- **Comercio Electrónico y Transacciones en Línea:** El desarrollo web posibilita la creación de plataformas de comercio electrónico, lo que permite a las empresas vender

productos y servicios en línea. Esto amplía su alcance y brinda a los consumidores la comodidad de realizar compras desde sus propios hogares.

- **Automatización de Procesos:** A través del desarrollo web, es factible automatizar diversos procesos empresariales, desde la gestión de inventario hasta el servicio al cliente. Esto conlleva a una mayor eficiencia operativa y ahorro de tiempo.
- **Alcance Global y Diversificación:** Una presencia web sólidamente establecida habilita a las organizaciones para alcanzar a una audiencia global. Las barreras geográficas desaparecen, lo que brinda la oportunidad de diversificar y expandir la base de clientes.
- **Análisis y Toma de Decisiones Basadas en Datos:** A través de herramientas de análisis web, es posible rastrear el comportamiento de los usuarios en el sitio. Esto proporciona datos valiosos sobre la popularidad del contenido, la interacción de los visitantes y la posibilidad de mejorar las estrategias digitales.
- **Innovación Continua:** El desarrollo web está en constante evolución, con la introducción de nuevas tecnologías, frameworks y enfoques que permiten crear experiencias web más avanzadas y atractivas. Los desarrolladores web tienen la oportunidad de innovar y mejorar de manera continua.

El desarrollo web es esencial para establecer presencia en línea, fomentar una comunicación efectiva, ofrecer una experiencia interactiva de calidad y mejorar la eficiencia en la era digital actual. Facilita la conexión con el mundo, aumenta la visibilidad y capacita a las organizaciones para brindar mejores experiencias a su audiencia y clientes.

### **Software para trabajar en desarrollo web**

El apartado dedicado al software para el desarrollo web emerge como un pilar fundamental en esta guía, delineando el entorno de trabajo esencial para los aspirantes a programadores web. Con un enfoque especial en el editor de código Brackets, los estudiantes son introducidos a una herramienta versátil y amigable que optimiza la creación y edición de código HTML, CSS y JavaScript.

Para el desarrollo y prueba de aplicaciones web con tecnología PHP, se recomienda instalar el paquete XAMPP en su computadora, que incluye el servidor Apache y el servidor MySQL con una interfaz de administración conveniente para los servicios correspondientes (Bandiera, 2019).

Brackets, reconocido por su capacidad de vista en vivo y resaltado de sintaxis, permite a los estudiantes visualizar los cambios en tiempo real, agilizando el proceso de desarrollo y ofreciendo un ambiente intuitivo para novatos y profesionales por igual. Además, la guía invita a los estudiantes a explorar y seleccionar herramientas adicionales según sus preferencias y nece-

sidades, permitiéndoles ampliar su conjunto de herramientas y personalizar su flujo de trabajo.

Este tema no solo introduce a los estudiantes a la utilidad práctica de Brackets, sino que también fomenta la autonomía al permitirles elegir otras herramientas en consonancia con sus preferencias personales. A través de la exploración y elección de estas herramientas adicionales, como editores de texto, extensiones y complementos, los educandos amplían su comprensión del ecosistema del desarrollo web y experimentan con diferentes enfoques. Esta sección resalta la importancia de seleccionar herramientas que se alineen con las preferencias y necesidades individuales, al tiempo que establece una base sólida para abordar los desafíos más avanzados de la programación web en las siguientes etapas de la guía.

## **HTML**

El tema central del lenguaje de marcado HTML en esta guía didáctica se erige como una columna vertebral esencial para la comprensión y creación de páginas web. HTML, el lenguaje fundamental en la construcción de la estructura y contenido de los sitios, es explorado en profundidad en esta sección. Los estudiantes se sumergen en la sintaxis de etiquetas, atributos y elementos, aprendiendo cómo organizar jerárquicamente el contenido para lograr diseños y funcionalidades deseadas. Desde la creación de encabezados y párrafos hasta la inclusión de imágenes y enlaces, esta sección expone la riqueza de posibilidades que HTML ofrece para presentar información de manera estructurada y accesible.

El lenguaje HTML, según comenta Celaya (Celaya, 2019), es un lenguaje de marcas o etiquetas. Esto quiere decir que los elementos que conforman la página web se describen mediante palabras especiales que marcan el inicio y el final de los mismos, estas palabras se denominan etiquetas.

Por lo general las etiquetas se establecen por pares de la siguiente forma:

- Etiqueta de apertura:

carácter < nombre de la etiqueta (sin espacios en blanco) y terminado con el carácter >.

- Etiqueta de cierre:

carácter <, el carácter /, nombre de la etiqueta (sin espacios en blanco) y terminado con el carácter >.

Por tanto, la estructura típica de las etiquetas HTML es:

```
<nombre_etiqueta>  
...  
</nombre_etiqueta>
```

Mediante la construcción de páginas web simples y más complejas, los estudiantes se desafían a sí mismos para crear diseños visualmente atractivos y funcionales. Esta sección sienta las bases para la comprensión de otros conceptos clave, como la

separación de contenido y estilo a través de CSS, y la interacción dinámica habilitada por JavaScript. Con un énfasis en la práctica y la comprensión profunda, el segmento dedicado a HTML establece una base sólida para el aprendizaje continuo en el campo del desarrollo web.

## Etiquetas, DOM

El tópico central de las etiquetas y el Modelo de Objetos del Documento (DOM, por sus siglas en inglés) en esta guía didáctica se presenta como una perspicaz exploración de cómo las etiquetas HTML interactúan y estructuran el contenido dentro de una página web. Los estudiantes se adentran en el concepto de las etiquetas como elementos fundamentales para la creación y organización de contenido, entendiendo cómo atributos y valores afectan la presentación y el comportamiento. Además, se desentraña el intrigante concepto del DOM, un modelo en el que cada elemento de una página web es tratado como un objeto manipulable mediante lenguajes de programación como JavaScript. Mediante ejemplos y ejercicios, los educandos descubren cómo interactuar con el DOM para crear interactividad y dinamismo en sus proyectos web.

Tabla 1. Lista de etiquetas HTML.

Etiqueta	Función
<code>&lt;!--...--&gt;</code>	Define un comentario
<code>&lt;!DOCTYPE&gt;</code>	Define el tipo de documento

<b>Etiqueta</b>	<b>Función</b>
<a>	Define un hipervínculo
<abbr>	Define una abreviación
<address>	Define la información de contacto del autor / propietario del documento
<area>	Define un área dentro de un mapa de imagen
<article>	Define un artículo
<aside>	Define el contenido lateral del contenedor de una página
<audio>	Define contenido de sonido
<b>	Define texto en negrita
<base>	Especifica la base donde se abrirán todas las URL del documento
<bdi>	Aísla una parte del texto que puede tener un formato diferente del texto externo
<bdo>	Sobreescribe la dirección del texto
<blockquote>	Define una sección que tiene otra fuente
<body>	Define el cuerpo del documento
 	Define un salto de línea
<button>	Define un botón clickeable
<canvas>	Se usa para dibujar gráficos en pantalla
<caption>	Define el título de una tabla
<cite>	Define el título de un trabajo
<code>	Define un trozo de código de programación
<col>	Especifica las propiedades de la columna para cada columna del elemento <colgroup>
<colgroup>	Especifica un grupo de una o más columnas de una tabla
<command>	Define un botón command al que un usuario puede invocar
<datalist>	Especifica en un input una lista pre-definida de opciones
<dd>	Define la descripción de un ítem en una lista de definición

<b>Etiqueta</b>	<b>Función</b>
<del>	Define un texto que ha sido definido en un Mdocument
<details>	Define detalles adicionales que el usuario puede ver o esconder
<dfn>	Define el término de una definición
<dialog>	Define una caja o ventana de dialogo
<div>	Define una sección en un documento
<dl>	Define una lista de definición
<dt>	Define un término (un ítem) en una lista de definición
<em>	Define énfasis en un texto
<embed>	Define el contenedor de una aplicación externa (no html)
<fieldset>	Grupo de elementos relacionados en un formulario
<figcaption>	Define el título para una figura <figure>
<figure>	Especifica auto-contenido
<footer>	Define el pie de página de un documento
<form>	Define un formulario html
<h1> a <h6>	Define encabezados o títulos
<head>	Define información acerca del documento
<header>	Define la sección de encabezado del documnto
<hgroup>	Grupo de encabezado (<h1> a <6>)
<hr>	Define un cambio de temática a partir de una línea dibujada
<html>	Define la raíz del documento
<i>	Define una parte del texto de modo alternativo
<iframe>	Define un frame en línea
<img>	Define una imagen
<input>	Define un control de entrada de texto
<ins>	Define texto que ha sido insertado en un documento
<kbd>	Define entrada del teclado

<b>Etiqueta</b>	<b>Función</b>
<keygen>	Define un campo generador de claves para formularios
<label>	Define el rótulo para un elemento <input>
<legend>	Define un título para los elementos <fieldset>, <figure>, <details>
<li>	Define un ítem de una lista
<link>	Define la relación entre un documento y un recurso externo (generalmente con hojas de estilo)
<map>	Define un mapa de imagen del cliente
<mark>	Define texto resaltado o marcado
<menu>	Define la lista de un menú
<meta>	Define un metadato de un documento
<meter>	Define una medida escalar en un rango conocido
<nav>	Define un link de navegación
<noscript>	Define un contenido alternativo para los usuarios que no soportan scripts del cliente
<object>	Define un objeto embebido
<ol>	Define una lista ordenada
<optgroup>	Define un grupo de opciones relacionadas en una lista desplegable
<option>	Define una opción en una lista desplegable
<output>	Define el resultado de un cálculo
<p>	Define un párrafo
<param>	Define un parámetro para un objeto
<pre>	Define texto pre-formateado
<progress>	Representa el progreso de una tarea en una barra
<q>	Define una cita corta
<rp>	Define que debe mostrar en navegadores que no soportan scripts de ruby
<rt>	Define una pronunciación de caracteres
<ruby>	Define una notación de ruby
<s>	Define texto que no es correcto

<b>Etiqueta</b>	<b>Función</b>
<samp>	Define un ejemplo de salida de un programa
<script>	Define un script del lado cliente
<section>	Define una sección de undocumento
<select>	Define un drop-down list
<small>	Define texto pequeño
<source>	Define los recursos para elementos multimedia
<span>	Define una pequeña sección de un documento
<strong>	Define un texto en negrita
<style>	Define un estilo para la información de un documento
<sub>	Define un texto que es subíndice
<summary>	Define un encabezado visible para el elemento <details>
<sup>	Define un texto que es superíndice
<table>	Define una tabla
<tbody>	Define el cuerpo de una tabla
<td>	Define una celda en una tabla
<textarea>	Define un control de entrada de múltiples líneas
<tfoot>	Agrupar los footer contenidos en una tabla
<th>	Define una celda de encabezado en una tabla
<thead>	Agrupar los encabezados de una tabla
<time>	Define fecha / hora
<title>	Define un título para el documento
<tr>	Define una fila en una tabla
<track>	Define texto de la pista para elementos multimedia (vídeo y audio)
<ul>	Define una lista desordenada
<var>	Define una variable
<video>	Define un vídeo o película
<wbr>	Define un posible salto de línea

Esta sección no solo dota a los estudiantes con el conocimiento de cómo estructurar y etiquetar el contenido en HTML, sino que también sienta las bases para futuras exploraciones en la programación del lado del cliente. Al comprender cómo el DOM permite a los desarrolladores acceder y modificar elementos de una página web en tiempo real, los educandos se preparan para construir aplicaciones web interactivas y dinámicas. A través de esta comprensión profunda de las etiquetas y el DOM, los estudiantes adquieren la capacidad de transformar páginas estáticas en experiencias enriquecedoras y atractivas para los usuarios finales.

## **Estructura de la Página Web**

El tema central de la Estructura de la Página Web en esta guía didáctica presenta a los estudiantes un fundamento esencial para la creación de sitios web cohesivos y visualmente atractivos. Los educandos son guiados a través de la comprensión de cómo se organiza y jerarquiza el contenido en una página mediante el uso de elementos estructurales como encabezados, párrafos, listas y secciones. Se destaca la importancia de crear una estructura lógica y semánticamente coherente utilizando etiquetas HTML apropiadas, lo que no solo facilita la comprensión de la información por parte de los motores de búsqueda, sino que también mejora la accesibilidad para los usuarios con discapacidades.

La imagen siguiente es un esquema típico de una página web. Se debe tener en cuenta que este esquema sufre variaciones, aunque normalmente son pequeñas.

Con el lenguaje HTML damos forma al texto y al resto de elementos de una página web mediante etiquetas como ya sabemos. Las etiquetas HTML son marcas, escritas entre corchetes angulares (< >)

Esquema de una etiqueta html

```
< etiqueta atributo = "valor" > contenido </etiqueta>
```

Donde:

***Etiqueta:***

Es el nombre de la etiqueta <etiqueta>. </etiqueta> (indica final de la etiqueta)

***Atributo:***

Característica que modifica el comportamiento de una etiqueta. Siempre se pone dentro de la etiqueta de apertura. No todas las etiquetas tienen atributo.

***Valor:***

Es el valor que damos al atributo para modificar el comportamiento de una etiqueta.

***Contenido:***

Texto al que queremos dar formato.

El conjunto de las dos etiquetas con su atributo y el texto contenido en su interior conforman un elemento HTML.

Esta es una línea de código o elemento de HTML:

```
< p align = center > La Tierra es nuestro planeta </p>
```

Y su interpretación es:

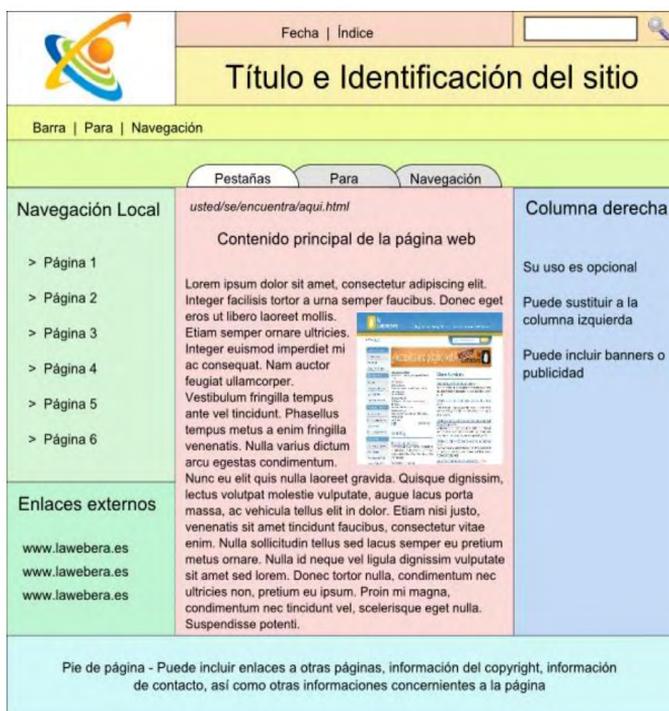
- `<p align = "center">` es la etiqueta de apertura. Indica un inicio de párrafo, es decir, a partir de aquí empieza un párrafo nuevo.
- `Align` es un atributo de la etiqueta `<p>` que sirve para fijar la alineación del párrafo
- `Center` es el valor que le adjudicamos al atributo `align` para hacer que el párrafo aparezca alineado en el centro de la página.
- `</p>` es la etiqueta de cierre. Aquí se acaba el párrafo que s inició con la etiqueta de apertura.
- Contenido “La Tierra es nuestro planeta” aparecerá centrado en la página.

## **Estructura del Sitio Web**

El tema central de la Estructura del Sitio Web en esta asignatura se enfoca en proporcionar a los estudiantes una comprensión profunda de cómo diseñar y organizar de manera efectiva

la arquitectura de un sitio web coherente y funcional. A lo largo de esta sección, los educandos explorarán los elementos esenciales que conforman una estructura web sólida, considerando aspectos como la navegación, la distribución de contenido y la usabilidad. Se aborda la importancia de diseñar una experiencia de usuario intuitiva y fluida, permitiendo a los visitantes interactuar con el sitio de manera eficiente y encontrar la información deseada sin dificultades. Además, los estudiantes se sumergen en conceptos de jerarquía de contenido y flujo de navegación, entendiendo cómo estas decisiones influyen en la percepción del sitio y su efectividad en la comunicación de información.

Figura 1. Título e identificación del sitio.



Desde la planificación de la disposición de las páginas hasta la consideración de las relaciones entre secciones y subpáginas, los educandos cultivarán la habilidad de traducir conceptos teóricos en diseños tangibles. A medida que los estudiantes internalizan cómo la estructura del sitio web impacta en la experiencia del usuario y la presentación de contenido, estarán equipados para diseñar y construir sitios web que no solo sean visualmente atractivos, sino también funcionales y accesibles. En última instancia, esta sección capacita a los estudiantes para crear sitios web que satisfagan las necesidades de los usuarios y logren sus objetivos comunicativos de manera efectiva.

## Cabecera

Figura 2. Cabecera.



De todas las partes de la estructura del diseño web, el encabezado es la que mayor tendencia tiene a repetirse entre páginas de un mismo sitio. La cabecera es una franja horizontal que ocupa todo el ancho de la página en la parte superior de la misma.

Partes que forman la cabecera.

## Logotipo

A la izquierda del encabezado suele ubicarse el logotipo del sitio, que ocupa un área importante dentro del mismo. Esta imagen debe tener un enlace a la página principal del sitio, ya que es uno de los estándares más respetados por los diseñadores y una de las formas que tienen los usuarios de volver al inicio.

## Zonas de navegación

Figura 3. Zona de navegación.



Desde el final del logotipo, ocupando el centro y la zona derecha de la cabecera, frecuentemente se sitúan una serie de zonas de navegación. En la zona superior, se encuentran algunos enlaces generales de la página, como acceso a usuarios registrados, registros de usuarios nuevos, acceso a perfiles, etc., así como ayudas y buscadores internos. La zona más ancha generalmente se encuentra ocupada por el título de la página, nombre de empresa, slogan, etc.

La zona inferior del encabezado se emplea para ubicar el menú de navegación principal, uno de los elementos más importantes a la hora de ayudar a los usuarios a encontrar las páginas componentes del sitio web.

## **Buscador**

Otra forma de colaborar con la exploración del sitio por parte de los usuarios es la instalación de un buscador, es aquí donde los usuarios buscan este tipo de elementos, ya que convencionalmente es aquí donde la mayoría de los diseñadores los ubican, aunque en algunos casos, puede encontrarse en la parte superior de alguna de las columnas laterales.

## **Columnas laterales**

El cuerpo principal de una página web, en el diseño moderno se encuentra dividido. Una de estas divisiones, consiste una o dos columnas laterales. Este tipo de columnas, se emplean para ubicar algunos elementos importantes de la página, como enlaces externos, navegación interna, publicidad, informaciones adicionales y contenidos secundarios.

Cuando se emplea una única columna, es indistinto para los usuarios si se coloca a derecha o izquierda del contenido, de acuerdo con estudios sobre el comportamiento de usuarios de sitios web, ya que su ubicación no plantea dificultad alguna. En el caso de que el diseño requiera el empleo de dos columnas, pue-

den ubicarse una a cada lado del cuerpo principal de la página o ambas a uno de los lados, siendo también indiferente en este caso si se ubican a la derecha o a la izquierda del contenido principal.

## **Área de contenido principal**

Esta zona del diseño de la página, es la que menos reglas tiene respecto de su diseño, ya que el hecho de que en ella se encuentren los contenidos principales de la página, el diseño depende en gran medida de los mismos, sobre todo en lo que respecta al tipo de contenidos, la cantidad, la disposición que se desea emplear, etc.

## **Pie de página**

Finalmente, debajo de todas las secciones, se coloca una faja horizontal en cuyo interior se colocan algunos elementos importantes, como el autor, copyright, acceso a diversas políticas del sitio y asuntos legales, datos de contacto, enlaces a sitios relacionados, enlaces internos, etc.

## **Formas de agregar estilos a HTML**

El tema central de las Formas de Agregar Estilos a HTML en esta asignatura busca sumergir a los estudiantes en el universo del diseño visual de páginas web. Aquí, los educandos explorarán las diversas metodologías para aplicar estilos a elementos HTML, transformando páginas en experiencias visualmente atractivas y

coherentes. Se introducirán conceptos de estilo en cascada (CSS), permitiendo a los estudiantes comprender cómo los estilos pueden ser declarados tanto inline, directamente en las etiquetas HTML, como internamente, dentro de la sección `<style>` de la página. Además, se explorará la potencia de las hojas de estilo externas, presentando cómo un solo archivo CSS puede influir en el diseño de múltiples páginas web. Los educandos comprenderán cómo utilizar selectores para apuntar a elementos específicos y cómo aplicar propiedades de estilo, como colores, tipografías y márgenes, para moldear la apariencia de los componentes de la página.

La vinculación de CSS en HTML es posible con hojas de estilo internas o externas. Con las hojas de estilo internas, las instrucciones CSS se encuentran dentro del documento HTML. Así, se puede integrar CSS al principio del archivo HTML o colocarlo en el código fuente. Con este método, una buena comprensión básica de HTML y su sintaxis es una ventaja. La forma más común y limpia de desarrollar páginas web es utilizar hojas de estilo CSS externas. El CSS se integra enlazando el documento HTML con un archivo CSS externo. A continuación, se presenta un resumen de los tres métodos:

### **Incluir CSS: estilo inline**

Con este método, las instrucciones de diseño se integran directamente en el código fuente mediante una etiqueta de estilo. Las propiedades deseadas solo se asignan a un elemento, de modo que es posible realizar diseños diferentes en todo el documento

HTML. En el ejemplo siguiente, el encabezamiento H1 debe aparecer en azul y con un tamaño de letra de 14:

```
<h1 style="color:blue; font-size:14px;">Esto es un encabezado</h1>
```

Con este tipo de integración, se pierden muchas de las ventajas de CSS. Entre ellas, la posibilidad de definir un único comando que luego se aplica, por ejemplo, a todos los elementos H1 del documento HTML. En algunas circunstancias, también hay un mayor esfuerzo de mantenimiento, porque se requiere la intervención directa en el código HTML.

### **Añadir CSS al principio del HTML**

En este caso, se incluye el CSS en la cabecera del documento HTML. La etiqueta de estilo se convierte así en parte del elemento de cabecera y se aplica a todo el archivo. Las instrucciones de diseño están contenidas en el documento, pero están separadas del código HTML. En el siguiente ejemplo, se aplica el mismo comando que en el ejemplo anterior. Esta vez, sin embargo, todos los encabezados H1 del archivo deben ser formateados de acuerdo con la información.

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {color:blue; font-size:14px;}
</style>
</head>
<body>
<h1> Esto es un encabezado </h1>
<p>Esto es un párrafo</p>
</body>
</html>
```

## Integrar el CSS con un archivo externo

Este es posiblemente el mejor método para enlazar CSS en HTML. Un sitio web suele constar de muchas páginas, por lo que tiene sentido guardar las instrucciones de diseño en un archivo separado. Esto permite una separación limpia entre el contenido y el diseño y facilita el mantenimiento. El archivo exportado está simplemente vinculado al documento HTML. Se guarda la hoja de estilos externa con la extensión .css y luego se utiliza una etiqueta de enlace para incluirla en el archivo HTML. En el siguiente ejemplo, el archivo CSS se llama stylesheet.css.

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="stylesheet.css">
</head>
<body>
<h1> Esto es un encabezado </h1>
<p>Esto es un párrafo</p>
</body>
</html>
```

El atributo `rel` define el tipo de relación lógica y `href` hace referencia al archivo CSS que se va a incrustar. Ten en cuenta que el elemento `link` puede tomar otros atributos. Con `media = "print"` específicas, por ejemplo, que la hoja de estilos solo se utiliza en la vista de impresión. La hoja de estilos externa no contiene ninguna etiqueta HTML, solo el selector respectivo y los corchetes con las instrucciones, como en el siguiente ejemplo:

```
H1 {
color: blue;
font-size:14px;
}
```

## Atributos HTML

El tema focal de los Atributos HTML en esta guía didáctica aborda uno de los componentes fundamentales en la construcción de páginas web. Los estudiantes se sumergen en la comprensión y aplicación de los atributos, que enriquecen las etiquetas HTML al brindarles información adicional y funcionalidad específica. A lo largo de esta sección, los educandos explorarán una variedad de atributos, desde los utilizados para mejorar la accesibilidad y la semántica, hasta aquellos que permiten la incrustación de contenido multimedia o la creación de hipervínculos. Se les enseñará cómo emplear atributos como “alt” para descripciones de imágenes, “href” para enlaces y “src” para recursos multimedia, otorgando a las etiquetas HTML un propósito más allá de su estructura básica.

Para clarificar estos conceptos, ejemplos prácticos serán presentados para cada tipo de atributo, permitiendo a los estudiantes ver la aplicación real de estas funcionalidades en un contexto web.

Al entender cómo los atributos influyen en la interacción de los usuarios con la página web, los estudiantes estarán mejor equipados para diseñar experiencias web más ricas y funcionales.

Las etiquetas pueden ir acompañadas de atributos, ahora bien, si necesitamos modificar alguno de los atributos de la etiqueta párrafo (<p>) mostrado, por ejemplo el tamaño de la fuente, alineación o tipografía deberemos aplicar dicha modificación a la etiqueta en cuestión. La forma general para referenciar los atributos es la siguiente:

```
<etiqueta atributo_1="valor_1" atributo_2="valor_2">  
Contenido  
</etiqueta>
```

También puede utilizarse el atributo `style` para la asignación de otros atributos, por ejemplo:

```
<etiqueta style="atributo_1:valor_1; atributo_2:valor_2;">  
Contenido  
</etiqueta>
```

Observar que a diferencia de la forma anterior los valores se asignan mediante dos puntos ( `:` ) en vez de igual ( `=` ), además luego de cada atributo se coloca punto y coma ( `;` ). A la forma vista de asignación de atributos aplicados mediante hojas de estilos en cascada (CSS). Aplicando lo visto al ejemplo del párrafo si debemos asignarle al mismo fuente tamaño 30px, color verde con tipografía verdana.

```
<p style="color: red; font-size: 30px; font-family: verdana;">
```

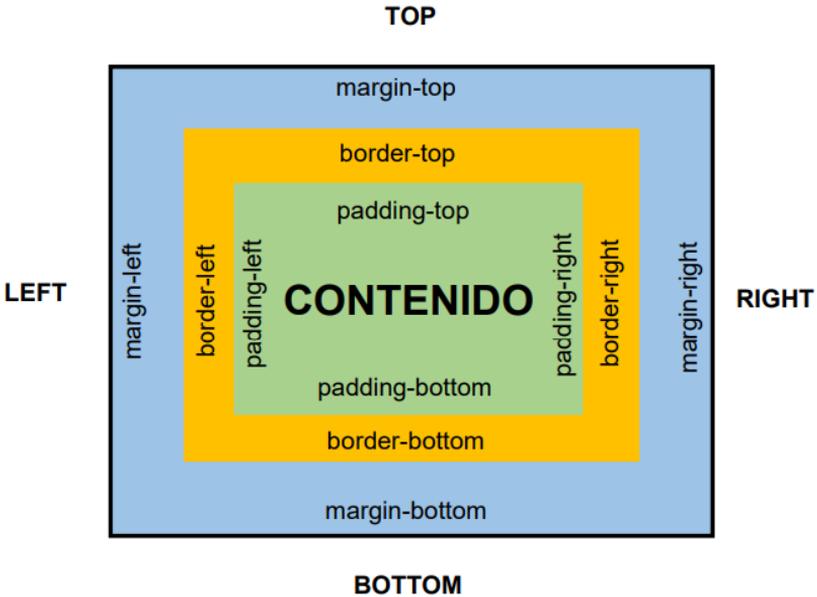
Este es un párrafo con los atributos asignados `</p>`, para la asignación de algunos atributos debemos tener en cuenta la forma en que los interpreta el navegador, por ejemplo, para:

- Margin

- Border
- Padding

Observar el modelo de caja CSS, ya que para la aplicación de los citados atributos lo deberemos realizar de la siguiente manera:

Figura 4. Padding.



Hay muchos atributos HTML diferentes, aunque, desde la introducción de HTML5, algunas opciones ya no son viables. A grandes rasgos, los atributos HTML pueden dividirse en cinco grupos diferentes:

- **Atributos universales clásicos**, que se pueden utilizar en casi todos los elementos y normalmente cumplen la misma función. Sin embargo, en algunos elementos no tienen ningún efecto.
- **Atributos específicos**, que afectan a elementos concretos y definen parámetros para estos.
- **Atributos que se utilizan desde la introducción de HTML5**; estos generan en parte nuevas posibilidades o sustituyen a los atributos HTML más antiguos.
- **Atributos de evento**, que activan un evento definido para un elemento cuando los usuarios realizan una determinada acción.
- **Atributos de datos**, que pueden contener información, pero no son visibles para los usuarios.

### **Atributos universales clásicos**

Los atributos universales clásicos están contemplados en la mayoría de las etiquetas HTML. Estos son los más conocidos:

Tabla 2. Atributos universales clásicos.

<b>Atributo HTML</b>	<b>Descripción</b>	<b>Ejemplo</b>
id	Sirve para identificar de forma individual a cada elemento HTML.	id="ejemplo"

Atributo HTML	Descripción	Ejemplo
class	Permite asignar un elemento a una o varias clases.	class="coches"
style	Define el estilo de un elemento HTML y puede determinar, por ejemplo, su color, fuente o tamaño.	style="color: blue; font-size: 2em"
title	Contiene información adicional sobre el contenido de un elemento; se muestra, por ejemplo, en una ventana separada cuando se pasa el ratón por encima del elemento en cuestión.	title="texto de ejemplo"
lang	Establece el idioma base de un documento.	<html lang="es">
dir	Define el sentido de lectura de un texto, bien sea de izquierda a derecha o viceversa.	<html dir="ltr">e- jemplo</html>

## Atributos HTML específicos

Existen numerosos atributos HTML específicos para definir elementos concretos. Estos son los más utilizados:

Tabla 3. Atributos HTML específicos.

Atributo HTML	Descripción	Ejemplo
alt	Muestra un texto alternativo si no se puede cargar o mostrar una imagen.	
height	Establece la altura de un elemento en px.	<img alt="el coche rojo" height="220">
width	Establece el ancho de un elemento en px.	<img alt="el coche rojo" width="220">
href	Define la URL de un enlace.	<a href="https://www.paginaejemplo.es" title="Más información">

Atributo HTML	Descripción	Ejemplo
hreflang	Establece el idioma del documento vinculado.	<code>&lt;a href="https://www.paginaejemplo.es" lang="es" hreflang="en"&gt;Información en inglés&lt;/a&gt;</code>
target	Define dónde se debe abrir un enlace.	<code>&lt;a href="https://www.paginaejemplo.es" target="_blank"&gt;</code>
rel	Define la relación entre el documento target y su vínculo.	<code>&lt;a rel="nofollow" href="https://www.paginaejemplo.es/"&gt;pagina de ejemplo&lt;/a&gt;</code>
src	Define el origen de un elemento.	<code>&lt;img src="https://www.ionos.es/digitalguide/cocherojo.jpg" alt="el coche rojo"&gt;</code>
autoplay	Garantiza que un determinado contenido multimedia se reproduzca automáticamente; sin embargo, la mayoría de los navegadores pueden anular esta configuración.	<code>&lt;reproducción automática de vídeo&gt;</code>
poster	Define una imagen que previsualizar cuando se inserta un vídeo en HTML.	<code>&lt;video controls poster="previsualización.png"&gt;</code>

## Nuevos atributos desde HTML5

Desde la introducción de HTML5 y HTML 5.1 hay algunos atributos HTML de nueva incorporación. Entre ellos se encuentran los siguientes:

Tabla 4. Nuevos atributos desde HTML5.

Atributo HTML	Descripción	Ejemplo
contenteditable	Define si el contenido de un elemento puede ser editado; sus posibles valores son true y false.	<code>&lt;p contenteditable="false"&gt;</code>

Atributo HTML	Descripción	Ejemplo
hidden	Atributo universal que puede ocultar un elemento.	<code>&lt;p hidden&gt;</code> Este texto quedará oculto <code>&lt;/p&gt;</code>
dropzone	Define si en el proceso de arrastrar y soltar el elemento este se copia (copy), se enlaza (link) o se mueve (move).	<code>&lt;p dropzone="move"&gt;</code>
draggable	Atributo universal que define si el contenido de un elemento puede ser arrastrado y soltado; sus posibles valores son true, false y auto.	<code>&lt;p draggable="false"&gt;</code>
loading	Establece cómo se cargan los medios externos; los valores permitidos son auto, eager y lazy.	<code>&lt;img src="https://www.ionos.es/digitalguide/cocherojo.jpg" alt="el coche rojo" loading="lazy"&gt;</code>
spellcheck	Determina si se debe activar un corrector ortográfico; los valores permitidos son true y false.	<code>&lt;p contenteditable="true" spellcheck="false"&gt;</code>

## Atributos de evento

Hay muchos atributos HTML diferentes, que activan eventos en un navegador. Los siguientes atributos HTML son, por tanto, solo una pequeña selección de diferentes eventos que pueden ser activados a través del JavaScript incluido en HTML.

Tabla 5. Atributos de evento.

Atributo HTML	Descripción	Ejemplo
click	Activa un evento en JavaScript con un clic de ratón.	<code>&lt;button onclick="Ejemplo de función ()"&gt;</code> "Haz clic aquí" <code>&lt;/button&gt;</code>

Atributo HTML	Descripción	Ejemplo
onscroll	Se activa al mover la rueda del ratón sobre él.	<code>&lt;div onscroll="Ejemplo de función () "&gt;</code>
onkeydown	Se activa cuando se pulsa una tecla.	<code>&lt;input type="text" onkeydown="Ejemplo de función () "&gt;</code>
onsearch	Activa un JavaScript en cuanto se introduce una búsqueda en la ventana de búsqueda.	<code>&lt;input type="búsqueda" onsearch="Ejemplo de función () "&gt;</code>
onerror	Puede ejecutar un JavaScript en caso de error.	<code>&lt;img src="https://www.ionos.es/digital-guide/cocherojo.jpg" onerror="Ejemplo de función () "&gt;</code>
oncopy	Se activa cuando se copia un texto.	<code>&lt;input type="text" oncopy="Ejemplo de función () " value="Copia este texto"&gt;</code>
onselect	Activa un JavaScript en cuanto se selecciona un elemento del input.	<code>&lt;input type="text" onselect="Ejemplo de función () " value="Texto de ejemplo"&gt;</code>

## Navegación entre secciones

La Navegación entre Secciones en esta guía didáctica tiene como objetivo capacitar a los estudiantes en la creación de una navegación fluida y efectiva dentro de las páginas web. Los educandos explorarán estrategias y técnicas para permitir que los usuarios se desplacen fácilmente entre diferentes secciones de un sitio, mejorando la experiencia de navegación. A lo largo de esta sección, se presentarán conceptos clave como los enlaces internos y las anclas HTML, que permiten a los visitantes saltar directamente a secciones específicas en una página larga.

Los estudiantes aprenderán cómo utilizar etiquetas como `<nav>` para crear menús de navegación claros y accesibles, y cómo aplicar estilos CSS para resaltar los elementos activos y mejorar la usabilidad. Al comprender cómo optimizar la navegación, los estudiantes estarán mejor preparados para crear sitios web que ofrezcan una experiencia de usuario intuitiva y eficiente, lo que a su vez mejorará la retención de visitantes y la accesibilidad general del sitio.

## El elemento de sección de navegación NAV

El elemento HTML `<nav>` representa una sección de una página cuyo propósito es proporcionar enlaces de navegación, ya sea dentro del documento actual o a otros documentos. Ejemplos comunes de secciones de navegación son menús, tablas de contenido e índices.

En este ejemplo, un bloque `<nav>` es usado para contener una lista no ordenada (`<ul>`) de enlaces. Con el CSS apropiado, esto puede ser presentado como una barra lateral, una barra de navegación o un menú desplegable.

```
<nav class="menu">
<ul>
<li><a href="#">Inicio</a></li>
<li><a href="#">Sobre nosotros</a></li>
<li><a href="#">Contacto</a></li>
</ul>
</nav>
```

## Reglas

El tema de las Reglas en esta guía didáctica abarca un aspecto crítico del diseño web al explorar cómo aplicar estilos a elementos específicos utilizando selectores y propiedades en CSS. Los estudiantes se sumergen en la comprensión de cómo las reglas CSS permiten definir el aspecto visual de elementos individuales o grupos de elementos en una página web. A lo largo de esta sección, los educandos aprenden a utilizar selectores para apuntar a elementos específicos, ya sean etiquetas HTML, clases o identificadores, y cómo aplicar propiedades de estilo como colores, fuentes, márgenes y dimensiones.

La comprensión de cómo las reglas pueden ser organizadas y jerarquizadas para establecer prioridades también se aborda, permitiendo a los estudiantes desarrollar un control preciso sobre el diseño visual de su contenido. A continuación, la lista de reglas principales que se cumplen al desarrollar con html:

- Delimitar el ámbito de las etiquetas (salvo excepciones)
- Etiqueta interna se cierra antes que externa
- Etiquetas a nivel de bloque pueden contener etiquetas a nivel de línea, pero no al revés
- Etiquetas no sensitivas a mayúsculas
- En vez de espacios agregar raya al piso
- Cerrar todos los tags, ie: `<p> ... </p>`, `<li> ... </li>`
- Incluir marca de cierre para tags vacíos, ie: `<hr />`, `<br />`

- Poner entre comillas todos los atributos, ie: ``
- Todos los tags en minúsculas
- Terminar con `</html>` (González, 2016)

## Selectores

Los selectores son una parte fundamental en CSS (Cascading Style Sheets) y se utilizan para identificar y aplicar estilos a elementos HTML específicos en una página web. Los selectores permiten a los desarrolladores web apuntar a uno o varios elementos en el documento HTML y aplicar reglas de estilo, como colores, fuentes, márgenes y más. Los tipos de selectores son:

**Selector de Tipo:** Este es el selector más simple y se usa para seleccionar elementos basados en su nombre de etiqueta. Por ejemplo, `p` seleccionaría todos los párrafos en la página.

**Selector de Clase:** Se utiliza para seleccionar elementos que tienen un atributo `class` específico. Por ejemplo, `.boton` seleccionaría todos los elementos con la clase "boton".

**Selector de ID:** Se utiliza para seleccionar un elemento en función de su atributo `id`. Por ejemplo, `#encabezado` seleccionaría el elemento con el ID "encabezado".

**Selector Universal (\*):** Este selector selecciona todos los elementos en la página.

**Selector de Atributo:** Se utiliza para seleccionar elementos que tienen un atributo específico. Por ejemplo, `[href]` seleccionaría todos los elementos que tienen un atributo `href`.

**Selector de Pseudo-clase:** Estos selectores permiten aplicar estilos a elementos en estados específicos. Por ejemplo, `:hover` se aplica cuando el usuario pasa el mouse sobre un elemento.

**Selector de Descendencia:** Permite seleccionar elementos que son descendientes de otros elementos. Por ejemplo, `ul li` seleccionaría todos los elementos `li` que están dentro de una lista desordenada `ul`.

**Selector de Clase Anidado:** Este selector se utiliza para seleccionar elementos que tienen una clase específica dentro de otro elemento con una clase específica. Por ejemplo, `.contenedor .boton` seleccionaría un botón dentro de un contenedor específico.

### **Combinación de Selectores:**

Los selectores se pueden combinar para crear reglas de estilo más específicas y dirigidas. Por ejemplo, `h1.destacado` seleccionaría todos los encabezados `h1` que tienen la clase “destacado”.

### **Importancia del Selector Correcto:**

Elegir el selector correcto es esencial en CSS, ya que afecta la especificidad y la cascada de estilos en una página web. Los selectores más específicos anulan a los menos específicos, y la cascada de estilos determina qué reglas se aplican cuando existen conflictos.

### Ejemplo de Uso:

Supongamos que deseas seleccionar todos los elementos a (enlaces) dentro de un elemento con la clase “menu”. El selector se vería así:

```
css
.menu a {
/* Estilos para enlaces dentro de un menú */
}
```

### Declaraciones, prioridades

#### Declaraciones en CSS:

Una declaración en CSS consta de dos partes: un selector y un conjunto de propiedades y valores separados por dos puntos. Por ejemplo:

```
Css
p {
color: blue;
font-size: 16px;
}
```

En este ejemplo, el selector es p, y las propiedades son color y font-size con sus respectivos valores.

## Prioridades en CSS:

Cuando se aplican múltiples reglas de estilo a un mismo elemento HTML, se utiliza el sistema de prioridades de CSS para determinar cuáles estilos prevalecerán. El sistema de prioridades se basa en varios factores:

**Especificidad del Selector:** Los selectores más específicos tienen prioridad sobre los selectores menos específicos. Por ejemplo, un selector de ID (#elemento) es más específico que un selector de clase (.clase).

**Orden de Declaración:** Si dos reglas CSS tienen la misma especificidad, la regla que aparece más abajo en el archivo CSS (o que se carga más tarde en caso de estilos en línea o incrustados) prevalecerá.

**Importancia:** Las declaraciones con !important tienen la máxima prioridad y anulan cualquier otra regla. Sin embargo, se recomienda usar !important con moderación, ya que puede complicar la resolución de conflictos.

### Ejemplo de Prioridades:

Supongamos que tienes el siguiente HTML y CSS:

html
<p id="párrafo" class="texto">Este es un párrafo.</p>

```
css
p {
color: red;
font-size: 14px;
}
.texto {
color: blue;
}
#parrafo {
color: green !important;
font-size: 16px;
}
```

En este ejemplo, el elemento `<p>` tiene un ID (“parrafo”) y una clase (“texto”). La regla `#parrafo` tiene la mayor prioridad debido al uso de `!important`, por lo que el texto será verde y tendrá un tamaño de fuente de 16px. Aunque hay otras reglas, la regla `#parrafo` anula todas las demás debido a su alta prioridad.

## Los estilos incluidos por el navegador

Los Estilos Incluidos por el Navegador en esta guía didáctica explora una faceta crucial del diseño web al abordar cómo los navegadores web aplican estilos predeterminados a elementos HTML. En esta sección, los estudiantes entenderán que los navegadores otorgan un conjunto predefinido de estilos a elementos básicos como encabezados, párrafos y enlaces, lo que puede afectar la apariencia visual de una página. Los educandos aprenderán

cómo los estilos del navegador pueden influir en la estructura y el diseño de la página, y cómo estos estilos predeterminados pueden ser sobrescritos y personalizados mediante CSS propio. Se enfatiza la importancia de conocer y manejar los estilos del navegador para lograr una experiencia web consistente y coherente en diferentes plataformas y dispositivos.

### **Agregando clases a los componentes escritos en HTML**

Agregar Clases a los Componentes Escritos en HTML en esta guía didáctica explora un aspecto fundamental en el diseño y la estilización de páginas web. Los estudiantes se sumergirán en la práctica de asignar clases a elementos HTML, lo que les permitirá aplicar estilos específicos a grupos de elementos que comparten características similares. A lo largo de esta sección, los educandos entenderán cómo las clases actúan como identificadores para elementos HTML, permitiendo que las reglas de CSS sean dirigidas a múltiples elementos a la vez. Se enfatiza la importancia de la modularidad y la reutilización del código, ya que las clases permiten aplicar estilos coherentes a diferentes partes de un sitio web.

Mediante ejemplos prácticos y ejercicios, los estudiantes experimentarán con la asignación de clases a elementos HTML y su posterior estilización a través de CSS. A través de estas actividades, los educandos aprenderán a crear reglas de estilo específicas para clases y cómo estas clases pueden ser reutilizadas en múltiples lugares de la página. La comprensión de cómo agregar clases a los componentes HTML potencia las habilidades de los estu-

diantes para lograr consistencia y modularidad en el diseño web, permitiéndoles construir sitios visualmente cohesivos y eficientes en términos de estilización. Esta sección provee las bases esenciales para que los estudiantes desarrollen habilidades clave en el diseño web y optimicen la presentación visual de sus proyectos.

## **Gestores de Contenidos**

A lo largo de esta sección, los educandos comprenderán la importancia de los sistemas de gestión de contenido (CMS, por sus siglas en inglés) en la simplificación y agilización del proceso de construcción y mantenimiento de sitios web. Se destacarán ejemplos populares de CMS, como WordPress, Joomla y Drupal, para mostrar cómo estas plataformas proporcionan interfaces amigables que permiten a los usuarios sin experiencia en programación crear y administrar contenido web dinámico. Los estudiantes explorarán la capacidad de los gestores de contenido para gestionar artículos, imágenes, videos y otros elementos multimedia, así como la facilidad para personalizar el diseño y la funcionalidad de los sitios mediante temas y complementos.

A través de ejemplos de uso y ejercicios prácticos, los estudiantes experimentarán la creación y personalización de sitios web utilizando un CMS. Mediante la comprensión de cómo los gestores de contenido simplifican tareas complejas y permiten una colaboración efectiva entre múltiples usuarios, los educandos estarán mejor preparados para enfrentar proyectos de desarrollo web del mundo real. Esta sección ofrece una visión enriquecedora de cómo los gestores de contenido impactan en la eficiencia y

la flexibilidad del desarrollo web, brindando a los estudiantes una perspectiva valiosa sobre cómo simplificar la creación y administración de contenido en sus futuros proyectos.

## **Inspector de elementos**

El Inspector de Elementos introduce a los estudiantes a una herramienta esencial en el desarrollo web, destinada a comprender y manipular en tiempo real la estructura y el estilo de una página web. A lo largo de esta sección, los educandos explorarán cómo utilizar el inspector de elementos, una función integrada en los navegadores web modernos, para inspeccionar y modificar el código HTML y CSS de una página en vivo. Los estudiantes aprenderán cómo abrir el inspector, seleccionar elementos específicos en la página y explorar sus atributos, contenido y estilos. Además, se mostrará cómo realizar ediciones temporales en el código para visualizar instantáneamente cómo los cambios afectan el diseño y la apariencia.

Mediante ejemplos prácticos y ejercicios interactivos, los estudiantes adquirirán habilidades prácticas para utilizar el inspector de elementos de manera efectiva. Desde ajustar el tamaño de los elementos hasta modificar colores y márgenes, los educandos experimentarán con la manipulación de propiedades CSS en tiempo real, lo que les permitirá comprender cómo se construye la presentación visual de una página web. Esta sección empodera a los estudiantes con la capacidad de explorar y experimentar con el código de una página web existente, lo que a su vez contribuye a una comprensión profunda de cómo los elementos y estilos interactúan para crear la experiencia web final.

## **Lenguajes de marcado y de presentación. (HTML, CSS)**

En la sección dedicada a Lenguajes de Marcado y de Presentación (como HTML y CSS), los estudiantes se sumergirán en los pilares fundamentales del diseño y desarrollo web. Explorarán la función esencial de HTML (Lenguaje de Marcado de Hipertexto) en la estructuración y organización de contenido en una página web. Aprenderán a utilizar etiquetas para definir encabezados, párrafos, listas y otros elementos, estableciendo la base para la creación de una estructura semántica y coherente. Además, se abordará la importancia de atributos que mejoran la accesibilidad y la usabilidad, proporcionando a los educandos la capacidad de crear contenido web enriquecedor y significativo.

En paralelo, se introducirá CSS (Hojas de Estilo en Cascada) como el lenguaje de presentación que permite estilizar y dar vida a los elementos marcados en HTML. Los estudiantes explorarán cómo CSS les permite definir colores, fuentes, márgenes y otros estilos visuales, brindándoles el poder de crear una presentación atractiva y coherente. A medida que los estudiantes comprenderán cómo estos lenguajes trabajan en conjunto, estarán mejor equipados para diseñar y desarrollar páginas web que ofrezcan tanto contenido estructurado como una presentación visual impactante.

### **Estilos, CSS**

Network (2023) menciona que los estilos se refieren a cómo se presenta y se muestra el contenido en una página web. Los estilos son esenciales para dar formato y apariencia a un sitio web.

## Estilos en Cascada (CSS)

CSS es un lenguaje de hojas de estilo que se utiliza para controlar el aspecto de las páginas web. Se utiliza junto con HTML para definir el diseño y la apariencia de los elementos de una página web, como el color, el tamaño, la fuente y la posición.

Los objetivos de CSS son:

- Estandarizar el aspecto de las páginas web. CSS permite definir los estilos de los elementos de una página web de forma independiente del navegador que se utilice.
- Facilitar el mantenimiento de las páginas web. CSS permite agrupar los estilos de los elementos de una página web en una sola hoja de estilo, lo que facilita el mantenimiento y la actualización de los estilos.
- Mejorar la accesibilidad de las páginas web. CSS permite definir los estilos de los elementos de una página web de forma accesible para personas con discapacidades.

**Sintaxis de CSS:** La sintaxis de CSS es sencilla y fácil de aprender. Las reglas de CSS se definen utilizando una sintaxis de tipo “propiedad: valor”. La propiedad define el atributo que se va a modificar, y el valor es el valor que se va a asignar al atributo.

Por ejemplo, para definir el color de fondo de una página web, se puede utilizar la siguiente regla:

```
CSS
body {
background-color: #ffffff;
}
```

En esta regla, la propiedad es “background-color” y el valor es “#ffffff”, que es el código hexadecimal del color blanco.

## Propiedades CSS

CSS tiene una gran cantidad de propiedades que se pueden utilizar para modificar el aspecto de los elementos de una página web. Algunas de las propiedades CSS más comunes son:

- Color: define el color de un elemento.
- Tamaño: define el tamaño de un elemento.
- Fuente: define la fuente de un elemento.
- Posición: define la posición de un elemento.
- Margenes: define los espacios entre un elemento y otros elementos.
- Bordes: define los bordes de un elemento.
- Estructura de las hojas de estilo CSS

Las hojas de estilo CSS se pueden colocar en el mismo documento HTML que contiene el contenido de la página web, o en un archivo externo.

Si la hoja de estilo se coloca en el mismo documento HTML, se debe colocar dentro de la etiqueta `<style>`.

Por ejemplo, el siguiente código muestra cómo colocar una hoja de estilo CSS en el mismo documento HTML que contiene el contenido de la página web:

HTML
<pre>&lt;!DOCTYPE html&gt; &lt;html lang="es"&gt; &lt;head&gt; &lt;title&gt;Mi página web&lt;/title&gt; &lt;style&gt; body { background-color: #ffff; } &lt;/style&gt; &lt;/head&gt; &lt;body&gt; &lt;h1&gt;Este es el título de mi página web&lt;/h1&gt; &lt;p&gt;Este es un párrafo de mi página web&lt;/p&gt; &lt;/body&gt; &lt;/html&gt;</pre>

Si la hoja de estilo se coloca en un archivo externo, se debe importar al documento HTML utilizando la etiqueta `<link>`.

Por ejemplo, el siguiente código muestra cómo importar una hoja de estilo CSS externa al documento HTML:

HTML
<pre>&lt;!DOCTYPE html&gt; &lt;html lang="es"&gt; &lt;head&gt; &lt;title&gt;Mi página web&lt;/title&gt; &lt;link rel="stylesheet" href="estilos.css"&gt; &lt;/head&gt; &lt;body&gt; &lt;h1&gt;Este es el título de mi página web&lt;/h1&gt; &lt;p&gt;Este es un párrafo de mi página web&lt;/p&gt; &lt;/body&gt; &lt;/html&gt;</pre>

En la asignatura Programación de Aplicaciones Web, los estudiantes aprenden a utilizar CSS para controlar el aspecto de las páginas web. Los estudiantes aprenden a utilizar las propiedades CSS más comunes para modificar el color, el tamaño, la fuente, la posición, los márgenes, los bordes y otros atributos de los elementos de una página web.

Los estudiantes también aprenden a utilizar hojas de estilo CSS internas y externas para importar estilos a sus páginas web.

## Tipos de textos y Propiedades

CSS es un lenguaje de hojas de estilo que se utiliza para controlar el aspecto de las páginas web. Una de las características más importantes de CSS es que permite modificar el texto de una página web según (W3Schools, 2023).

### Tipos de textos

CSS permite controlar el tipo de texto que se utiliza en una página web. Esto se puede hacer utilizando la propiedad `font-family`.

La propiedad `font-family` acepta un valor que es una lista de nombres de fuentes separadas por comas. El navegador elegirá la primera fuente que esté instalada en el sistema del usuario.

Por ejemplo, el siguiente código establece el tipo de letra del texto de un elemento `h1` en “Arial, sans-serif”:

CSS
<pre>h1 {   font-family: "Arial", sans-serif; }</pre>

### Propiedades de texto

CSS también permite controlar otras propiedades del texto, como el tamaño, el color, el peso y el estilo.

## Tamaño de texto

La propiedad `font-size` se utiliza para establecer el tamaño del texto. El valor de esta propiedad puede ser un valor absoluto, como “12px”, o un valor relativo, como “1.5em”.

Por ejemplo, el siguiente código establece el tamaño del texto de un elemento `p` en 16 píxeles:

CSS
<pre>p {   font-size: 16px; }</pre>

## Color de texto

La propiedad `color` se utiliza para establecer el color del texto. El valor de esta propiedad puede ser un valor hexadecimal, como “#ffffff”, o un valor de nombre de color, como “red”.

Por ejemplo, el siguiente código establece el color del texto de un elemento `h2` en rojo:

CSS
<pre>h2 {   color: red; }</pre>

## Peso del texto

La propiedad `font-weight` se utiliza para establecer el peso del texto. Los valores posibles son “normal”, “bold” y “bolder”.

Por ejemplo, el siguiente código establece el peso del texto de un elemento `a` en negrita:

CSS
<pre>a { font-weight: bold; }</pre>

## Estilo de texto

La propiedad `text-decoration` se utiliza para establecer el estilo del texto. Los valores posibles son “none”, “underline”, “overline”, “line-through” y “blink”.

Por ejemplo, el siguiente código establece el estilo del texto de un elemento `em` en subrayado:

CSS
<pre>em { text-decoration: underline; }</pre>

## Dimensiones fijas para elementos

**Dimensiones en HTML:** En HTML, los elementos generalmente tienen dimensiones que se definen en píxeles o unidades relativas. Puedes especificar dimensiones fijas para elementos, como imágenes, mediante atributos como `width` (ancho) y `height` (alto) en las etiquetas HTML.

html
<pre>&lt;img src="imagen.jpg" width="300" height="200" alt="Una imagen"&gt;</pre>

En este ejemplo, la imagen tiene un ancho fijo de 300 píxeles y un alto fijo de 200 píxeles.

**Dimensiones fijas en CSS:** CSS se utiliza para controlar las dimensiones y el diseño de elementos HTML de manera más flexible. Puedes definir dimensiones fijas utilizando las propiedades `width` y `height`.

css
<pre>.caja { width: 400px; height: 250px; }</pre>

En este caso, todos los elementos con la clase “caja” tendrán un ancho de 400 píxeles y un alto de 250 píxeles.

**Ventajas y desventajas:** El uso de dimensiones fijas en elementos puede ofrecer control preciso sobre la apariencia de una página web. Sin embargo, puede generar problemas en diseños responsivos, ya que los elementos no se ajustarán automáticamente al tamaño de la pantalla del dispositivo.

**Unidades de medida:** CSS admite diversas unidades de medida, como píxeles (px), porcentaje (%), em (relativo al tamaño de fuente) y rem (relativo al tamaño de fuente de la raíz del documento). Al elegir dimensiones, considera qué unidades son las más adecuadas para tus necesidades y si deseas que los elementos sean escalables.

**Diseño responsivo:** En la programación de aplicaciones web modernas, es común utilizar enfoques de diseño responsivo en los que se evitan dimensiones fijas en favor de unidades relativas. Esto permite que los elementos se adapten al tamaño de la pantalla del dispositivo, mejorando la experiencia del usuario en dispositivos móviles y de escritorio.

**Lenguajes de marcado y de presentación:** HTML se considera un lenguaje de marcado y se utiliza para estructurar y organizar el contenido de una página web, mientras que CSS se considera un lenguaje de presentación y se utiliza para definir la apariencia y el estilo de ese contenido.

HTML se centra en etiquetas y elementos que describen la estructura y el significado del contenido, como párrafos, encabezados, enlaces, listas, etc.

CSS se centra en reglas de estilo que se aplican a los elementos HTML para controlar su diseño, colores, fuentes y otros aspectos visuales.

**Integración de HTML y CSS:** HTML y CSS suelen trabajar en conjunto. Puedes utilizar CSS para definir dimensiones fijas en elementos HTML y ajustar su apariencia visual.

## Backgrounds de color e imagen

### *Background de color en CSS:*

En CSS, puedes establecer el fondo de un elemento HTML con un color sólido utilizando la propiedad `background-color`.

```
css
.mi-clase {
background-color: #FF5733; /* Color de fondo en formato hexadecimal */
}
```

Puedes utilizar nombres de colores comunes (por ejemplo, red para rojo) o códigos de colores en formato hexadecimal (como #FF5733) para definir el color de fondo.

## Background de imagen en CSS:

CSS también te permite establecer una imagen como fondo para un elemento utilizando la propiedad `background-image`.

```
css
.mi-clase {
background-image: url('imagen.png'); /* URL de la imagen de fondo */
}
```

La propiedad `background-image` requiere la URL de la imagen que deseas utilizar como fondo. Puedes especificar imágenes locales o imágenes alojadas en la web.

## Propiedades adicionales de background en CSS:

Puedes controlar varios aspectos del fondo de un elemento utilizando propiedades adicionales de CSS, como `background-repeat` (para repetir la imagen de fondo), `background-size` (para ajustar el tamaño de la imagen de fondo) y `background-position` (para definir la posición de la imagen de fondo).

```
css
.mi-clase {
background-image: url('imagen.png');
background-repeat: no-repeat;
background-size: cover;
background-position: center;
}
```

## Background de gradiente en CSS:

Además de colores sólidos e imágenes, CSS te permite crear fondos de gradiente utilizando la propiedad `background-image` con el valor `linear-gradient` o `radial-gradient`. Los gradientes pueden ser lineales o radiales y pueden combinar varios colores.

```
css
.mi-clase {
background-image: linear-gradient(to right, #FF5733, #FFC300); /* Gradiente lineal */
}
```

## Uso en diseño web:

Los backgrounds de color e imagen son fundamentales para el diseño web, ya que permiten personalizar la apariencia de un sitio web, desde cambiar el fondo de una sección hasta crear efectos visuales interesantes.

Es importante tener en cuenta la accesibilidad y la legibilidad al elegir colores y fondos, especialmente en texto sobre fondos de colores o imágenes.

## Integración con HTML:

Para aplicar backgrounds a elementos HTML, debes vincular las clases o identificadores CSS adecuados a los elementos HTML correspondientes. Por ejemplo:

```
html
<div class="mi-clase">
Contenido con fondo de color o imagen.
</div>
```

## Lenguaje de marcado y de presentación:

El uso de backgrounds de color e imagen está relacionado con CSS, que es un lenguaje de presentación utilizado para definir la apariencia de los elementos HTML. Mientras que HTML se encarga de la estructura y el contenido de la página web, CSS se encarga de su presentación.

## Unidades de medida y colores

### *Unidades de Medida:*

**Píxeles (px):** Los píxeles son la unidad de medida más común en el diseño web. Un píxel representa un punto en la pantalla. Por ejemplo, puedes definir el ancho de un elemento usando píxeles, como `width: 200px;`.

**Porcentaje (%):** Las unidades de porcentaje se utilizan para crear diseños web flexibles y adaptables. Puedes definir el ancho de un elemento en términos de un porcentaje del ancho de su contenedor. Por ejemplo, `width: 50%;` hará que el elemento ocupe la mitad del ancho de su contenedor.

**Em (em):** El “em” es una unidad de medida relativa al tamaño de fuente de un elemento. Por ejemplo, si defines `font-size:`

16px; en un elemento y luego estableces `width: 2em`; en otro elemento dentro de él, el segundo elemento tendrá un ancho de 32 píxeles (2 veces el tamaño de fuente del elemento padre).

**Rem (rem):** Similar al “em”, pero relativo al tamaño de fuente de la raíz del documento en lugar del elemento padre. Esto hace que el “rem” sea especialmente útil para mantener un diseño más predecible.

**Centímetros (cm), Milímetros (mm), Pulgadas (in):** Estas unidades son menos comunes en diseño web, pero se utilizan ocasionalmente cuando es necesario especificar tamaños en unidades del sistema métrico o imperial.

**Viewport Width (vw) y Viewport Height (vh):** Estas unidades se basan en el tamaño de la ventana gráfica del navegador. Por ejemplo, `width: 50vw`; hará que un elemento ocupe el 50% del ancho de la ventana gráfica.

## Colores:

1. **Notación de Color en CSS:** Los colores en CSS se pueden especificar de varias maneras, incluyendo:

**Nombres de color:** CSS proporciona nombres de colores predefinidos, como “red”, “blue”, “green”, etc.

**Códigos de color hexadecimales:** Los colores se pueden especificar utilizando valores hexadecimales, como #FF5733.

**Códigos de color RGB:** Se pueden utilizar valores de rojo (red), verde (green) y azul (blue) para definir un color, por ejemplo, `rgb(255, 87, 51)`.

**Códigos de color RGBA:** Similar a RGB, pero con un canal alfa que controla la transparencia del color, como `rgba(255, 87, 51, 0.5)`.

- 2. Gradientes de Color:** CSS permite crear fondos de gradiente utilizando la propiedad `background-image`. Puedes definir gradientes lineales y radiales con una combinación de colores.
- 3. Modelo de color HSL:** CSS también admite el modelo de color HSL (Hue, Saturation, Lightness), que permite definir colores en términos de matiz, saturación y luminosidad. Por ejemplo, `hsl(20, 50%, 75%)` representa un color con un matiz de 20 grados, saturación del 50% y luminosidad del 75%.
- 4. Colores transparentes:** Puedes usar valores de transparencia en el canal alfa para crear colores transparentes. Por ejemplo, `rgba(255, 87, 51, 0.5)` representa un color semi-transparente.

## **Bordes, Margenes Rellenos**

### ***Bordes (CSS):***

Los bordes se utilizan para definir los límites de un elemento en una página web. En CSS, puedes aplicar bordes a elementos

HTML utilizando propiedades como `border`, `border-width`, `border-style` y `border-color`.

```
css
.mi-clase {
border: 2px solid #333; /* Un borde sólido de 2 píxeles de ancho, color gris oscuro */
}
```

Los estilos de borde comunes incluyen “solid” (sólido), “dashed” (guionado), “dotted” (punteado) y “double” (doble línea), entre otros.

### **Márgenes (CSS):**

Los márgenes son espacios en blanco que rodean un elemento y permiten controlar su separación con otros elementos en la página. Puedes definir márgenes utilizando propiedades como `margin-top`, `margin-right`, `margin-bottom` y `margin-left`.

```
css
.mi-clase {
margin: 10px; /* Márgenes de 10 píxeles en todos los lados */
}
```

Los márgenes pueden ser positivos (para alejar un elemento de otros) o negativos (para superponer elementos).

## Rellenos (CSS):

Los rellenos se utilizan para definir el espacio entre el contenido de un elemento y su borde. Puedes definir rellenos utilizando propiedades como `padding-top`, `padding-right`, `padding-bottom` y `padding-left`.

```
css
.mi-clase {
padding: 20px; /* Rellenos de 20 píxeles en todos los lados */
}
```

Los rellenos son útiles para ajustar el espaciado interno de un elemento y controlar cómo se muestra su contenido.

## Uso en Diseño Web:

Los bordes, márgenes y rellenos son fundamentales en el diseño web, ya que permiten controlar la apariencia y la disposición de los elementos en una página. Se utilizan para crear diseños con espaciado adecuado, separación entre elementos, resaltado de elementos clave y para mejorar la legibilidad y la estética de una página web.

## Combinación con Otros Estilos:

Estos conceptos se combinan frecuentemente con otras propiedades de CSS, como colores de fondo, fuentes y tamaños de texto, para lograr un diseño web completo y coherente.

## **Lenguaje de Marcado y Presentación:**

En el contexto de HTML y CSS, el lenguaje de marcado (HTML) se utiliza para estructurar y organizar el contenido, mientras que el lenguaje de presentación (CSS) se utiliza para definir la apariencia y el estilo de ese contenido. Los bordes, márgenes y rellenos son ejemplos de cómo se controla la presentación de elementos HTML mediante CSS.

## **Propiedades Relacionadas:**

Junto con bordes, márgenes y rellenos, otras propiedades de CSS relacionadas incluyen `border-radius` (para crear bordes redondeados), `box-shadow` (para agregar sombras a elementos) y `outline` (para crear bordes enfocables).

La comprensión y el uso efectivo de estos conceptos son esenciales para el diseño y la maquetación de páginas web. Estos elementos permiten a los desarrolladores web controlar la presentación y la interacción de los elementos en sus aplicaciones web.

## **Conclusión**

Los contenidos presentados en la asignatura de Programación de Aplicaciones Web abarcan una amplia gama de conceptos y herramientas esenciales para el desarrollo web. Comenzando con una introducción al desarrollo web y la explicación de conceptos generales, los estudiantes adquieren una base sólida para comprender los elementos esenciales de la creación de sitios web. Se exploran en profundidad aspectos clave como HTML, el len-

guaje de marcado fundamental para la estructura web, y cómo aplicar etiquetas y el Modelo de Objeto del Documento (DOM) para manipular el contenido.

Además, se aborda la estructura de páginas y sitios web, lo que es fundamental para la organización y la navegación efectiva entre diferentes secciones. La introducción a la adición de estilos a HTML permite a los estudiantes comprender cómo dar vida a sus sitios web mediante el uso de reglas, selectores, propiedades y declaraciones CSS.

La asignatura también se centra en la navegación entre secciones, lo que es crucial para la experiencia del usuario, y explora en detalle las clases, atributos y atributos HTML que permiten una mayor personalización y estilo. Además, se discute la interacción con los estilos incluidos por el navegador.

En conjunto, estos contenidos proporcionan a los estudiantes una base sólida para el desarrollo web y les permiten comprender cómo crear sitios web estructurados y visualmente atractivos. A medida que avanzan en su educación, podrán aplicar estos conocimientos para desarrollar aplicaciones web más avanzadas y sofisticadas. La asignatura sienta las bases para futuros estudios y proyectos en el campo del desarrollo web.

## Autoevaluación

**Pregunta 1: ¿Cuál de los siguientes no es un aspecto clave a considerar en el desarrollo web, según Molina et al., (2018)?**

- a) Tecnología utilizada
- b) Diseño de interacción
- c) Tendencias de marketing
- d) Contenido de alta calidad
- e) Optimización de servidores

**Pregunta 2: ¿Qué lenguaje de marcado se utiliza en la construcción de la estructura y contenido de las páginas web?**

- a) JavaScript
- b) CSS
- c) HTML
- d) PHP
- e) Python

**Pregunta 3: ¿Qué función cumple el Modelo de Objetos del Documento (DOM) en el desarrollo web?**

- a) Definir la estructura de la página web
- b) Almacenar datos en el servidor
- c) Manipular elementos de la página en tiempo real
- d) Diseñar la interfaz de usuario
- e) Encriptar datos sensibles

**Pregunta 4: Según el texto, ¿por qué es importante crear una estructura lógica y semánticamente coherente en una página web?**

- a) Para aumentar el rendimiento del servidor
- b) Para facilitar la comprensión de los usuarios con discapacidades

- c) Para acelerar la carga de la página
- d) Para mejorar la seguridad del sitio web
- e) Para obtener un mejor posicionamiento en los motores de búsqueda

**Pregunta 5: ¿Cuál de las siguientes etiquetas HTML se utiliza para definir una lista desordenada?**

- a) <li>
- b) <ul>
- c) <ol>
- d) <p>
- e) <dl>

**Pregunta 6: ¿Cuál de las siguientes tecnologías se asocia principalmente con el FrontEnd en desarrollo web?**

- a) PHP
- b) HTML
- c) Node.js
- d) MySQL
- e) Java

**Pregunta 7: ¿Cuál de las siguientes tecnologías se utiliza para dar estilo a una página web?**

- a) HTML
- b) JavaScript
- c) CSS
- d) PHP
- e) XML

**Pregunta 8: ¿Qué se entiende por “SEO” en el desarrollo web?**

- a) Search Engine Optimization (Optimización de Motores de Búsqueda)
- b) Sistema de Enlaces y Opciones
- c) Servicio de Edición en Línea
- d) Sociedad de Empresas Online
- e) Software de Entretenimiento en Línea

**Pregunta 9: ¿Qué etiqueta HTML se utiliza para definir un título para un documento web?**

- a) <title>
- b) <head>
- c) <h1>
- d) <p>
- e) <header>

**Pregunta 10: ¿Cuál de las siguientes tecnologías se utiliza para crear aplicaciones web interactivas y dinámicas en el lado del cliente?**

- a) HTML
- b) CSS
- c) MySQL
- d) DOM
- e) JavaScript

**Pregunta 11: ¿Cuál es la principal área del sitio web donde se suele ubicar el logotipo y el enlace a la página principal?**

- a) Columna lateral izquierda
- b) Área de contenido principal
- c) Cabecera
- d) Pie de página
- e) Zona de navegación superior

**Pregunta 12: ¿Qué atributo de HTML se utiliza para proporcionar una descripción de una imagen en una etiqueta de imagen (<img>)?**

- a) src
- b) alt
- c) href
- d) style
- e) title

**Pregunta 13: ¿Cuál de las siguientes opciones no es una forma de enlazar hojas de estilo CSS a un documento HTML?**

- a) Hoja de estilo interna
- b) Añadir CSS al principio del HTML
- c) Añadir CSS al final del HTML
- d) Integrar el CSS con un archivo externo
- e) Hoja de estilo inline

**Pregunta 14: ¿Qué atributo se utiliza comúnmente en las etiquetas de enlace (<a>) para especificar la URL de destino?**

- a) src
- b) style
- c) alt
- d) href
- e) rel

**Pregunta 15: ¿En qué parte de la estructura de una página web se ubica generalmente el menú de navegación principal?**

- a) Zona de navegación superior
- b) Área de contenido principal
- c) Columna lateral derecha
- d) Pie de página
- e) Columna lateral izquierda

**Pregunta 16: ¿Cuál de los siguientes atributos HTML es utilizado para asignar un estilo de color, fuente o tamaño a un elemento?**

- A) id
- B) class
- C) style
- D) title
- E) lang

**Pregunta 17: ¿Cuál es el selector utilizado para aplicar estilos a elementos HTML basados en su atributo de clase?**

- A) Selector de Tipo
- B) Selector de Clase
- C) Selector de ID
- D) Selector Universal (\*)
- E) Selector de Atributo

**Pregunta 18: ¿Cuál es el propósito de los gestores de contenido (CMS) en el desarrollo web?**

- a) Simplificar y acelerar la construcción de sitios web.
- b) Programar aplicaciones web complejas.
- c) Controlar la gestión de servidores web.
- d) Optimizar la velocidad de carga de la página.
- e) Realizar análisis de datos en tiempo real.

**Pregunta 19: ¿Cuál es el objetivo de la herramienta “Inspector de Elementos” en el desarrollo web?**

- a) Editar videos en tiempo real.
- b) Comprobar el clima actual de la ubicación de un sitio web.
- c) Comprender y modificar la estructura y el estilo de una página web en vivo.

- d) Dibujar gráficos y animaciones interactivas.
- e) Crear contenido de redes sociales.

**Pregunta 20: ¿Qué unidad de medida se utiliza comúnmente en diseño web para establecer dimensiones flexibles?**

- a) Pulgadas (in).
- b) Milímetros (mm).
- c) Píxeles (px).
- d) Kilogramos (kg).
- e) Centímetros (cm).



## **Capítulo 2**

Programación del lado del cliente y del servidor

Axarnet (2023) menciona acerca del lenguaje del lado del servidor:

### **Programación del Lado del Cliente:**

**Definición:** La programación del lado del cliente se refiere a la escritura de código y la lógica que se ejecuta en el navegador web del usuario. Esto significa que el cliente (es decir, el navegador del usuario) se encarga de procesar el código y gestionar la interacción con el usuario.

**Tecnologías clave:** La programación del lado del cliente generalmente se realiza utilizando lenguajes como HTML, CSS y JavaScript. Estos lenguajes se utilizan para definir la estructura de la página web (HTML), su estilo y diseño (CSS), y para agregar interactividad y dinamismo (JavaScript).

**Responsabilidades:** En el lado del cliente, se gestionan tareas como la validación de formularios, la interacción con el usuario a través de eventos, la manipulación del DOM (Document Object Model), la carga de contenido dinámico y la comunicación con el servidor a través de solicitudes HTTP (por ejemplo, AJAX).

**Ventajas:** La programación del lado del cliente permite una experiencia de usuario más ágil y receptiva, ya que la mayoría de las operaciones se realizan en el navegador del usuario sin necesidad de recargar la página completa.

## **Programación del Lado del Servidor:**

**Definición:** La programación del lado del servidor implica la escritura de código y la lógica que se ejecuta en el servidor web. El servidor se encarga de procesar las solicitudes del cliente, realizar operaciones en la base de datos y devolver resultados al cliente.

**Tecnologías clave:** La programación del lado del servidor puede involucrar una variedad de lenguajes y tecnologías, como PHP, Python, Ruby, Java, Node.js y otros. También implica la interacción con bases de datos mediante lenguajes como SQL.

**Responsabilidades:** En el lado del servidor, se gestionan tareas como el procesamiento de formularios, la autenticación de usuarios, la gestión de sesiones, la recuperación y almacenamiento de datos en la base de datos y la generación dinámica de páginas web.

**Ventajas:** La programación del lado del servidor proporciona mayor seguridad, control y escalabilidad a las aplicaciones web. También permite el almacenamiento de datos confidenciales y la administración de recursos compartidos, como bases de datos.

## **Interacción Cliente-Servidor:**

**Comunicación:** La programación del lado del cliente y del servidor se comunica a través de protocolos como HTTP (Hypertext Transfer Protocol). El cliente envía solicitudes al servidor, que procesa la solicitud y devuelve una respuesta.

**Arquitecturas:** Hay varias arquitecturas de aplicaciones web, como el modelo cliente-servidor tradicional, la arquitectura de tres capas (cliente, servidor de aplicaciones y base de datos) y la arquitectura sin servidor (serverless), que delega gran parte de la lógica en servicios de terceros.

**Ejemplos:** Un ejemplo común de interacción cliente-servidor es cuando un usuario completa un formulario en un sitio web (cliente), envía los datos al servidor, donde se procesan y almacenan en una base de datos, y se devuelve una página actualizada al cliente.

## **Arquitectura de aplicaciones multinivel**

**Definición:** La arquitectura de aplicaciones multinivel, también conocida como arquitectura de tres capas, es un enfoque de diseño de software en el que una aplicación se divide en tres niveles o capas lógicas distintas, cada una con un conjunto específico de responsabilidades y funcionalidades. Estos tres niveles son el nivel de presentación, el nivel de lógica de negocio y el nivel de acceso a datos (*Aplicaciones Multinivel*, 2023).

### **Tres Capas Fundamentales:**

**Nivel de Presentación:** Esta capa es la interfaz de usuario de la aplicación y se encarga de la presentación de datos al usuario. Puede incluir componentes de interfaz de usuario, páginas web, formularios y elementos visuales. En el contexto de aplicaciones web, esta capa suele estar compuesta por HTML, CSS y JavaScript.

**Nivel de Lógica de Negocio:** Aquí se encuentra la lógica central de la aplicación. Es responsable de procesar datos, aplicar reglas de negocio y tomar decisiones. En aplicaciones web, esta capa a menudo se implementa utilizando lenguajes de programación del lado del servidor, como PHP, Java, Python, o Node.js. También puede interactuar con bases de datos y servicios externos.

**Nivel de Acceso a Datos:** Esta capa se ocupa de la gestión de datos y el acceso a fuentes de datos, como bases de datos relacionales, sistemas de archivos, servicios web u otros almacenes de datos. Se encarga de realizar operaciones de lectura y escritura en la base de datos y de garantizar la integridad de los datos.

### **Beneficios:**

**Separación de Responsabilidades:** La arquitectura de tres capas permite una clara separación de responsabilidades, lo que facilita el mantenimiento y la escalabilidad de la aplicación. Cada capa se encarga de una tarea específica, lo que simplifica la gestión y el desarrollo.

**Reutilización de Código:** La lógica de negocio y las operaciones de acceso a datos se pueden reutilizar en diferentes partes de la aplicación y en diferentes interfaces de usuario, lo que ahorra tiempo y esfuerzo.

**Escalabilidad:** Es más sencillo escalar cada capa de manera independiente para satisfacer las demandas de rendimiento de la aplicación. Por ejemplo, si se necesita aumentar la capacidad de la base de datos, se puede hacer sin afectar las demás capas.

## **Ejemplo en Aplicaciones Web:**

En una aplicación web, el nivel de presentación está compuesto por las páginas HTML, CSS y JavaScript que el usuario ve en su navegador.

El nivel de lógica de negocio se implementa en el lado del servidor, donde se procesan las solicitudes del usuario, se aplican reglas de negocio y se generan respuestas dinámicas.

El nivel de acceso a datos se encarga de la gestión de la base de datos, incluyendo consultas, inserciones y actualizaciones de datos.

Tecnologías y Frameworks: La elección de las tecnologías y frameworks específicos para implementar una arquitectura multinivel en una aplicación web dependerá de los requisitos y preferencias del proyecto. Algunos ejemplos comunes incluyen el uso de frameworks de desarrollo web como Ruby on Rails, Django, Spring (Java), o Express.js (Node.js).

## **PHP**

PHP es un lenguaje de programación de código abierto que se utiliza principalmente en el desarrollo de aplicaciones web. Es muy popular y ampliamente adoptado para la creación de sitios web dinámicos y aplicaciones web (PHP, 2023).

## **Características Clave de PHP:**

**Interpretado:** PHP es un lenguaje interpretado, lo que significa que el código se ejecuta en el servidor web y genera HTML que se envía al navegador del usuario.

**Incrustación en HTML:** Una de las ventajas de PHP es su facilidad de integración con HTML. Puedes incluir código PHP directamente en documentos HTML, lo que facilita la creación de páginas web dinámicas.

**Multiplataforma:** PHP es compatible con una variedad de sistemas operativos, incluyendo Windows, macOS y sistemas Unix/Linux.

**Amplia Comunidad:** PHP cuenta con una gran comunidad de desarrolladores y una amplia variedad de recursos, lo que facilita el aprendizaje y la solución de problemas.

## **Usos Comunes de PHP en Aplicaciones Web:**

**Creación de Sitios Web Dinámicos:** PHP se utiliza para crear sitios web dinámicos que pueden mostrar contenido diferente a los usuarios según su interacción o elecciones.

**Formularios y Procesamiento de Datos:** PHP es fundamental para el procesamiento de formularios en línea. Puedes recopilar datos de formularios web, validarlos y guardarlos en una base de datos utilizando PHP.

**Interacción con Bases de Datos:** PHP se integra fácilmente con sistemas de gestión de bases de datos como MySQL, PostgreSQL y SQLite, permitiendo la creación, lectura, actualización y eliminación de datos.

**Autenticación y Control de Acceso:** PHP se utiliza para gestionar sistemas de autenticación y autorización de usuarios, lo que garantiza que solo usuarios autorizados tengan acceso a ciertas partes de una aplicación.

**Generación de Contenido Dinámico:** PHP permite la generación de contenido web en tiempo real, lo que es esencial para la creación de blogs, foros, tiendas en línea y otros tipos de aplicaciones.

Ejemplo de Código PHP:

```
php
<?php
$nombre = "Juan";
echo "¡Hola, " . $nombre . "!";
?>
```

Este código PHP define una variable \$nombre con el valor "Juan" y luego lo imprime en la página web. Cuando se ejecuta en el servidor, el navegador del usuario verá "¡Hola, Juan!" en la página.

## Frameworks de PHP:

Además de la programación en PHP “puro”, también puedes utilizar frameworks de desarrollo web de PHP, como Laravel, Symfony, CodeIgniter y Zend Framework, que ofrecen una estructura organizada y herramientas útiles para el desarrollo de aplicaciones web.

## Integración PHP con HTML

Integración de HTML, CSS y PHP, 2023 enfatiza que:

**Interacción Cliente-Servidor:** En una aplicación web, el servidor web se encarga de procesar las solicitudes de los clientes (navegadores web) y responder con contenido HTML y otros recursos. PHP es un lenguaje del lado del servidor que se utiliza para generar dinámicamente el contenido HTML que se envía al navegador del usuario.

**Incrustación de PHP en HTML:** Una de las características clave de PHP es su capacidad de integración con HTML. Puedes incrustar código PHP directamente dentro de documentos HTML utilizando etiquetas especiales, como `<?php` y `?>`. Esto permite mezclar contenido estático (HTML) con contenido dinámico (generado por PHP) en la misma página.

**Variables PHP en HTML:** Puedes usar variables PHP para incrustar datos dinámicos en un documento HTML. Las variables PHP se definen con el signo `$`, y su valor puede ser impreso directamente en el HTML.

Por ejemplo:

```
php
<?php
$nombre = "Juan";
?>
<h1>Hola, <?php echo $nombre; ?>!</h1>
```

Cuando este código se ejecuta en el servidor, el navegador mostrará “Hola, Juan!” en la página, con el nombre obtenido de la variable PHP.

**Estructuras de Control:** PHP también permite el uso de estructuras de control como condicionales (if, else, switch) y bucles (for, while) en el contexto de HTML. Esto es útil para controlar el flujo de contenido y generar contenido dinámico basado en condiciones o iteraciones.

```
Php
<?php
$edad = 25;
if ($edad >= 18) {
echo "Eres mayor de edad.";
} else {
echo "Eres menor de edad.";
}
?>
```

**Funciones y Lógica de Negocio:** En lugar de imprimir directamente en HTML, también puedes utilizar PHP para definir funciones y lógica de negocio que generen o procesen datos antes de mostrarlos en la página. Por ejemplo, puedes conectarte a una base de datos, recuperar información y presentarla en una tabla HTML.

**Formularios Web:** PHP es ampliamente utilizado para procesar datos de formularios web. Puedes recibir datos enviados por el usuario a través de formularios HTML, validarlos, realizar acciones en función de los datos y proporcionar respuestas personalizadas.

**Ventajas:** La integración de PHP con HTML permite crear páginas web dinámicas y personalizadas en función de las necesidades de los usuarios. Esto es esencial para aplicaciones web interactivas y sitios web que deben mostrar información actualizada o responder a las acciones del usuario.

Ejemplo Completo de Página Web PHP + HTML:

Php
-----

```
<!DOCTYPE html>
<html>
<head>
<title>Página Web PHP</title>
</head>
<body>
<h1>Bienvenido a mi sitio web</h1>
<p>Hoy es <?php echo date("d/m/Y"); ?></p>
<?php
$edad = 25;
if ($edad >= 18) {
echo "<p>Eres mayor de edad.</p>";
} else {
echo "<p>Eres menor de edad.</p>";
}
?>
</body>
</html>
```

En este ejemplo, se mezcla HTML y PHP para mostrar un mensaje de bienvenida y la edad del usuario en una página web.

## Almacén de datos PHP

**Bases de Datos:** PHP se utiliza comúnmente para interactuar con bases de datos, como MySQL, PostgreSQL, SQLite y otros sistemas de gestión de bases de datos. Puedes usar PHP para conectarte a una base de datos, realizar consultas, insertar datos,

actualizar registros y recuperar información. Esto es fundamental para aplicaciones web que requieren almacenamiento y recuperación de datos.

**Sesiones y Cookies:** PHP permite gestionar sesiones y cookies en aplicaciones web. Las sesiones se utilizan para mantener el estado de un usuario a lo largo de múltiples solicitudes, lo que es útil para la autenticación y el seguimiento del usuario. Las cookies son pequeños fragmentos de datos que se almacenan en el navegador del usuario.

**Ficheros y Almacenamiento en el Servidor:** PHP te permite trabajar con ficheros y directorios en el servidor. Puedes subir, descargar, modificar y gestionar ficheros, lo que es esencial en aplicaciones que manipulan contenido multimedia, archivos o datos específicos.

**APIs y Servicios Web:** PHP se usa para consumir y proporcionar datos a través de APIs y servicios web. Esto permite que una aplicación web se comunique con otros sistemas y recupere o envíe datos a través de solicitudes HTTP.

**Frameworks y Bibliotecas:** Muchos frameworks y bibliotecas de PHP, como Laravel, Symfony y CodeIgniter, ofrecen herramientas y componentes para el almacenamiento y manipulación de datos. Estos facilitan la creación de aplicaciones web sofisticadas.

## **Interacción con bases de datos.**

**Concepto Fundamental:** La interacción con bases de datos se refiere a la capacidad de una aplicación web para conectarse a una base de datos, realizar operaciones como la inserción, actualización, eliminación y consulta de datos, y luego utilizar esos datos en la aplicación (Readthedocs, 2023).

**Tipos de Bases de Datos:** En el desarrollo web, las bases de datos más comunes son las bases de datos relacionales como MySQL, PostgreSQL y SQL Server, así como bases de datos NoSQL como MongoDB y Firebase. La elección de la base de datos depende de las necesidades específicas del proyecto.

**Lenguaje SQL:** En el caso de bases de datos relacionales, la interacción con la base de datos generalmente se realiza a través de lenguajes como SQL (Structured Query Language). SQL permite realizar consultas y manipular datos en la base de datos.

**PHP y Bases de Datos:** PHP es un lenguaje de programación muy utilizado en el desarrollo web y es especialmente conocido por su capacidad para interactuar con bases de datos. Para conectarse a una base de datos desde PHP, se pueden usar extensiones como mysqli o PDO.

### **Operaciones Principales:**

**Inserción (INSERT):** Permite agregar nuevos registros o filas a la base de datos. Por ejemplo, para agregar un nuevo usuario a una tabla de usuarios.

**Actualización (UPDATE):** Se utiliza para modificar datos existentes en la base de datos. Por ejemplo, para actualizar la información de un producto en una tienda en línea.

**Eliminación (DELETE):** Permite eliminar registros o filas de la base de datos. Por ejemplo, para eliminar un comentario en un blog.

**Consulta (SELECT):** Se utiliza para recuperar datos de la base de datos. Por ejemplo, para mostrar una lista de productos en una tienda en línea.

**Preparación de Consultas:** Es importante utilizar consultas preparadas (prepared statements) al interactuar con bases de datos desde PHP. Esto ayuda a prevenir ataques de inyección SQL y garantiza la seguridad de la aplicación.

**ORM (Mapeo Objeto-Relacional):** En lugar de escribir consultas SQL directamente, muchas aplicaciones web modernas utilizan ORM, que permite mapear objetos de la aplicación a tablas de base de datos. Ejemplos populares de ORM para PHP incluyen Eloquent (Laravel) y Doctrine.

**Escalabilidad y Rendimiento:** La interacción eficiente con bases de datos es fundamental para el rendimiento de una aplicación. Esto incluye la indexación adecuada, el uso de cachés y la optimización de consultas.

**Conexión Segura:** Es importante asegurarse de que la conexión a la base de datos sea segura y se sigan las mejores prácticas de seguridad para proteger los datos.

**Bases de Datos NoSQL:** En el caso de bases de datos NoSQL, la interacción puede ser diferente, ya que no siguen un modelo relacional. Estas bases de datos a menudo utilizan formatos como JSON y se centran en la escalabilidad y la flexibilidad.

### Acceso a bases de datos Sentencias SQL

*MySQL*, 2023 considera que en el desarrollo web, las bases de datos relacionales son comunes y se utilizan para almacenar datos estructurados en tablas. Algunos ejemplos populares de sistemas de gestión de bases de datos relacionales son *MySQL*, *PostgreSQL*, *SQL Server* y *SQLite*.

## SELECT

La sentencia `SELECT` se utiliza para recuperar datos de la base de datos. Puedes especificar qué columnas deseas recuperar y aplicar condiciones para filtrar los resultados.

Sql
-----

<code>SELECT nombre, edad FROM empleados WHERE departamento = 'Ventas';</code>
--

## INSERT (inserción)

La sentencia `INSERT` se utiliza para agregar nuevos registros o filas a una tabla de la base de datos. Por ejemplo, para agregar un nuevo usuario a una tabla de usuarios.

Sql
-----

<code>INSERT INTO usuarios (nombre, correo) VALUES ('Juan', 'juan@example.com');</code>
---

## DELETE (eliminación)

La sentencia DELETE se utiliza para eliminar registros o filas de una tabla de la base de datos. Por ejemplo, para eliminar un comentario en un blog.

Sql
DELETE FROM comentarios WHERE id = 101;

## UPDATE (actualización)

La sentencia UPDATE permite modificar datos existentes en la base de datos. Por ejemplo, para actualizar la información de un producto en una tienda en línea.

Sql
UPDATE productos SET precio = 50 WHERE id = 1;

## Desarrollo Aplicación WEB con base de Datos

(Guerra, 2021) considera los siguientes pasos de Desarrollo Aplicación WEB con base de Datos:

**Definición de Requisitos:** Comienza por comprender los requisitos de tu aplicación web. ¿Qué tipo de datos necesitas almacenar? ¿Cómo se debe interactuar con ellos? ¿Cuáles son las funcionalidades requeridas? Esto te ayudará a definir la estructura de la base de datos y las características de la aplicación.

**Elección de la Tecnología:** Selecciona la tecnología que utilizarás para desarrollar la aplicación web y gestionar la base de datos. Algunas opciones populares incluyen lenguajes de programación (como PHP, Python, Ruby, Node.js), sistemas de gestión de bases de datos (como MySQL, PostgreSQL, MongoDB), y frameworks web (como Ruby on Rails, Django, Laravel).

**Diseño de la Base de Datos:** Diseña la estructura de la base de datos, incluyendo tablas, campos y relaciones. Define cómo se organizarán los datos para satisfacer los requisitos de la aplicación. Utiliza diagramas ER (Entidad-Relación) o herramientas de modelado de bases de datos si es necesario.

**Desarrollo de la Aplicación Web:** Escribe el código de la aplicación web. Esto incluye la creación de páginas web, la implementación de lógica de negocio, la validación de formularios y la interacción con la base de datos. Asegúrate de seguir buenas prácticas de desarrollo web y seguridad.

**Conexión a la Base de Datos:** La aplicación web debe establecer una conexión con la base de datos. Utiliza API o bibliotecas proporcionadas por la tecnología que estás utilizando para interactuar con la base de datos.

**Operaciones CRUD:** Implementa operaciones CRUD (Crear, Leer, Actualizar y Eliminar) para interactuar con los datos en la base de datos. Esto implica el uso de sentencias SQL (o el equivalente en la tecnología que estés utilizando) para insertar, consultar, actualizar y eliminar registros.

**Seguridad:** Asegúrate de que la aplicación web sea segura. Implementa mecanismos de autenticación y autorización para proteger los datos de la base de datos. Evita la inyección de SQL y otros ataques comunes.

**Pruebas y Depuración:** Realiza pruebas exhaustivas de la aplicación web para garantizar que funcione correctamente. Soluciona errores y realiza pruebas de rendimiento para optimizar la velocidad de respuesta.

**Despliegue:** Publica la aplicación web en un servidor web para que los usuarios puedan acceder a ella. Configura la base de datos en un servidor accesible. Asegúrate de tener copias de seguridad regulares de los datos.

**Mantenimiento y Actualizaciones:** Mantén la aplicación web y la base de datos actualizadas. Realiza mantenimiento regular, soluciona errores y agrega nuevas funcionalidades según sea necesario.

### **Consideraciones Importantes:**

**Escalabilidad:** Diseña la base de datos y la aplicación con la escalabilidad en mente. A medida que la cantidad de datos y usuarios aumenta, la aplicación debe poder manejar la carga sin problemas.

**Respaldo de Datos:** Implementa políticas de respaldo de datos para garantizar la seguridad de la información almacenada en la base de datos. Esto es esencial para la recuperación de datos en caso de problemas.

**Cumplimiento Legal:** Asegúrate de cumplir con las leyes y regulaciones de protección de datos y privacidad que se apliquen a tu aplicación y a los datos que maneja.

**Documentación:** Documenta el diseño de la base de datos, la estructura de las tablas, las relaciones y las operaciones disponibles en la aplicación. Esto facilitará el mantenimiento y la colaboración con otros desarrolladores.

## Conclusión

Los contenidos cubiertos en la asignatura de Programación de Aplicaciones Web, que incluyen programación tanto del lado del cliente como del lado del servidor, arquitectura de aplicaciones multinivel, PHP, integración de PHP con HTML, almacenamiento de datos, interacción con bases de datos y sentencias SQL (SELECT, INSERT, DELETE, UPDATE), desempeñan un papel fundamental en la creación de aplicaciones web dinámicas y funcionales.

En esta asignatura, los estudiantes han adquirido un conocimiento sólido de cómo funcionan las aplicaciones web desde una perspectiva tanto técnica como arquitectónica. Han aprendido a utilizar el lenguaje de programación PHP para desarrollar aplicaciones en el lado del servidor, lo que les permite gestionar datos y lógica empresarial de manera eficiente. Además, han comprendido cómo integrar PHP con HTML para generar contenido dinámico y personalizado en las páginas web.

El almacenamiento de datos y la interacción con bases de datos son aspectos críticos de la asignatura. Los estudiantes han explorado cómo diseñar bases de datos, cómo conectarse a ellas utilizando PHP y cómo realizar operaciones esenciales como SELECT para recuperar datos, INSERT para agregar nuevos registros, DELETE para eliminar datos y UPDATE para modificar registros existentes. Estos conceptos son fundamentales para el desarrollo de aplicaciones web que requieren una gestión efectiva de datos.

La asignatura también ha proporcionado una comprensión de la arquitectura de aplicaciones multinivel, que es esencial para el diseño y la escalabilidad de aplicaciones web. Los estudiantes han aprendido a desarrollar aplicaciones web que siguen una estructura lógica y organizada, lo que mejora la mantenibilidad y la experiencia del usuario.

Los contenidos de esta asignatura han preparado a los estudiantes para diseñar, desarrollar y mantener aplicaciones web interactivas y funcionales. Han adquirido habilidades y conocimientos esenciales en programación web, arquitectura de aplicaciones, gestión de bases de datos y desarrollo de aplicaciones web con base de datos. Esta base sólida sienta las bases para futuros estudios y proyectos en el emocionante campo de la programación de aplicaciones web.

## Autoevaluación

**Pregunta 1: ¿Cuál de los siguientes lenguajes se utiliza principalmente en la programación del lado del cliente?**

- a) Java
- b) PHP
- c) HTML
- d) Python

**Pregunta 2: ¿Qué capa de la arquitectura de aplicaciones multinivel se encarga de la presentación de datos al usuario?**

- a) Nivel de Acceso a Datos
- b) Nivel de Lógica de Negocio
- c) Nivel de Presentación
- d) Nivel de Comunicación

**Pregunta 3: ¿Cuál de las siguientes afirmaciones sobre PHP es correcta?**

- a) PHP es un lenguaje de programación del lado del cliente.
- b) PHP se utiliza principalmente para la creación de páginas web estáticas.
- c) PHP es un lenguaje interpretado que se ejecuta en el servidor web.
- d) PHP no se integra con HTML en absoluto.

**Pregunta 4: ¿Qué función cumplen las sesiones y las cookies en aplicaciones web desarrolladas con PHP?**

- a) Mantener la base de datos
- b) Gestionar el almacenamiento de archivos
- c) Mantener el estado del usuario a lo largo de múltiples solicitudes
- d) Validar formularios

**Pregunta 5: ¿Qué ventaja ofrece la integración de variables PHP en documentos HTML?**

- a) Permite a los navegadores ejecutar código PHP directamente.
- b) Hace que la página sea estática y rápida.
- c) Permite mezclar contenido estático con contenido dinámico.
- d) Añade complejidad innecesaria a las páginas web.

**Pregunta 6: ¿Cuál de las siguientes tecnologías es esencial para la programación del lado del cliente en una aplicación web?**

- a) CSS
- b) Python
- c) SQL
- d) Ruby

**Pregunta 7: ¿En cuál de las siguientes capas se encuentra la lógica central de una aplicación en la arquitectura de tres capas?**

- a) Nivel de Presentación
- b) Nivel de Lógica de Negocio
- c) Nivel de Acceso a Datos
- d) Nivel de Comunicación

**Pregunta 8: ¿Cuál es una característica clave de PHP?**

- a) Es un lenguaje de programación de código cerrado.
- b) No se puede utilizar en el desarrollo de aplicaciones web.
- c) Se ejecuta en el navegador del usuario.
- d) Permite la incrustación en documentos HTML.

**Pregunta 9: ¿Para qué se utilizan comúnmente las cookies en aplicaciones web?**

- a) Para decorar las páginas web.
- b) Para gestionar el almacenamiento de archivos.
- c) Para mantener el estado del usuario a lo largo de múltiples solicitudes.
- d) Para ejecutar código JavaScript.

**Pregunta 10: ¿Cuál es el propósito principal de integrar PHP con HTML en una página web?**

- a) Para hacer que la página sea más lenta.
- b) Para permitir a los navegadores ejecutar código PHP directamente.
- c) Para mantener el estado de un usuario a lo largo de múltiples solicitudes.
- d) Para crear páginas web estáticas sin contenido dinámico.

**Pregunta 11: ¿Cuál de las siguientes no es una operación principal en la interacción con bases de datos?**

- a) Inserción (INSERT)
- b) Actualización (UPDATE)
- c) Eliminación (DELETE)
- d) Compartir (SHARE)
- e) Consulta (SELECT)

**Pregunta 12: ¿Qué lenguaje se utiliza comúnmente para interactuar con bases de datos relacionales?**

- a) JSON
- b) Python
- c) SQL
- d) PHP
- e) HTML

**Pregunta 13: ¿Cuál de las siguientes sentencias se utiliza para agregar nuevos registros a una tabla de la base de datos?**

- a) INSERT
- b) UPDATE
- c) DELETE
- d) SELECT
- e) JOIN

**Pregunta 14: ¿Cuál es la sentencia que se utiliza para eliminar registros de una tabla de la base de datos?**

- a) SELECT
- b) DELETE
- c) INSERT
- d) DROP
- e) MODIFY

**Pregunta 15: ¿Qué paso en el desarrollo de una aplicación web involucra la definición de la estructura de la base de datos y las características de la aplicación?**

- a) Elección de la Tecnología
- b) Diseño de la Base de Datos
- c) Pruebas y Depuración
- d) Conexión a la Base de Datos
- e) Despliegue

**Pregunta 16: ¿Qué tipo de base de datos es MongoDB?**

- a) Relacional
- b) NoSQL
- c) SQLite
- d) MySQL
- e) PostgreSQL

**Pregunta 17: ¿Cuál es el propósito de las consultas preparadas (prepared statements) al interactuar con bases de datos?**

- a) Acelerar la conexión a la base de datos.
- b) Realizar transacciones de bases de datos.
- c) Prevenir ataques de inyección SQL.
- d) Crear copias de seguridad de la base de datos.
- e) Desarrollar aplicaciones en el servidor.

**Pregunta 18: ¿Qué operación se utiliza para modificar datos existentes en la base de datos?**

- a) INSERT
- b) DELETE
- c) UPDATE
- d) SELECT
- e) CREATE

**Pregunta 19: ¿Qué paso implica escribir el código de la aplicación web y la interacción con la base de datos?**

- a) Pruebas y Depuración
- b) Definición de Requisitos
- c) Desarrollo de la Aplicación Web
- d) Mantenimiento y Actualizaciones
- e) Elección de la Tecnología

**Pregunta 20: ¿Qué es una consideración importante en la interacción con bases de datos relacionada con el aumento de datos y usuarios?**

- a) Seguridad
- b) Escalabilidad
- c) Respaldo de Datos
- d) Documentación
- e) Cumplimiento Legal



## **Capítulo 3**

Frameworks de desarrollo

Los frameworks de desarrollo son herramientas esenciales en el mundo de la programación web, ya que simplifican y aceleran el proceso de creación de aplicaciones web al proporcionar estructuras predefinidas, bibliotecas y patrones de diseño (IMMUNE, 2023).

## **Definiciones**

Algunos de los frameworks de desarrollo más populares utilizados en la asignatura de Programación de Aplicaciones Web son:

### **Ruby on Rails:**

Lenguaje: Ruby

Descripción: Ruby on Rails, comúnmente llamado Rails, es un framework de desarrollo web de código abierto conocido por su simplicidad y productividad. Sigue el principio de “convención sobre configuración” y utiliza el patrón de diseño Modelo-Vista-Controlador (MVC).

### **Django:**

Lenguaje: Python

Descripción: Django es un framework de desarrollo web de alto nivel que enfatiza la facilidad de uso y la productividad. Ofrece un ORM (Object-Relational Mapping) incorporado y sigue el patrón MVC.

## **Laravel:**

Lenguaje: PHP

Descripción: Laravel es un popular framework de desarrollo web PHP que se centra en la elegancia y la simplicidad. Proporciona características como el enrutamiento, el ORM Eloquent y el sistema de plantillas Blade.

## **Express.js:**

Lenguaje: JavaScript (Node.js)

Descripción: Express.js es un framework minimalista y flexible para el desarrollo de aplicaciones web basadas en Node.js. Se utiliza comúnmente para crear aplicaciones web y APIs.

## **Spring Boot:**

Lenguaje: Java

Descripción: Spring Boot es un framework de desarrollo de aplicaciones empresariales en Java. Facilita la creación de aplicaciones web, microservicios y aplicaciones escalables.

## **Angular:**

Lenguaje: TypeScript

**Descripción:** Angular es un framework de desarrollo web front-end mantenido por Google. Proporciona una estructura sólida para crear aplicaciones web interactivas y dinámicas.

### **React:**

Lenguaje: JavaScript

**Descripción:** React es una biblioteca de JavaScript utilizada para construir interfaces de usuario interactivas. Aunque no es un framework completo, se utiliza ampliamente en el desarrollo web moderno.

### **Vue.js:**

Lenguaje: JavaScript

**Descripción:** Vue.js es otro framework de desarrollo web front-end que se centra en ser fácil de aprender y usar. Es adecuado para la creación de interfaces de usuario interactivas y componentes reutilizables.

### **ASP.NET Core:**

Lenguaje: C#

**Descripción:** ASP.NET Core es un framework de desarrollo web de Microsoft que permite crear aplicaciones web escalables y seguras. Se ejecuta en Windows, Linux y macOS.

## **Flask:**

Lenguaje: Python

**Descripción:** Flask es un framework de desarrollo web minimalista de Python que se utiliza para crear aplicaciones web ligeras y rápidas.

### **Consideraciones al Elegir un Framework:**

**Tipo de Aplicación:** Elige un framework que se adapte a la naturaleza de tu aplicación. Algunos son más adecuados para aplicaciones grandes y complejas, mientras que otros son ideales para proyectos más pequeños.

**Lenguaje de Programación:** Asegúrate de que el lenguaje de programación utilizado por el framework sea compatible con tus habilidades y las necesidades del proyecto.

**Comunidad y Soporte:** Un framework con una comunidad activa y una amplia base de usuarios tiende a tener una mejor documentación y más recursos disponibles.

**Características Específicas:** Examina las características específicas de cada framework, como la gestión de base de datos, la seguridad y las capacidades de desarrollo front-end, y elige según las necesidades de tu proyecto.

**Escalabilidad:** Asegúrate de que el framework sea escalable y pueda manejar un crecimiento futuro si es necesario.

## Ventajas y Desventajas

Torrejón (2023) considera las siguientes ventajas de los Frameworks de Desarrollo:

### Ventajas de los Frameworks de Desarrollo

**Productividad Mejorada:** Los frameworks proporcionan estructuras predefinidas y bibliotecas que permiten a los desarrolladores ahorrar tiempo en tareas comunes, lo que aumenta la velocidad de desarrollo.

**Mantenimiento Simplificado:** Las aplicaciones construidas con frameworks suelen ser más fáciles de mantener, ya que siguen patrones de diseño y convenciones establecidos.

**Reutilización de Código:** Los frameworks promueven la reutilización de código, lo que significa que puedes utilizar componentes o módulos desarrollados previamente en otros proyectos.

**Escalabilidad:** Muchos frameworks están diseñados para ser escalables, lo que facilita la gestión del crecimiento de una aplicación a medida que más usuarios la utilizan.

**Seguridad Incorporada:** Los frameworks a menudo incluyen características de seguridad integradas, como la prevención de ataques de inyección SQL y protección contra vulnerabilidades comunes.

**Comunidad Activa:** Los frameworks populares tienen comunidades activas de desarrolladores que ofrecen soporte, documentación y complementos.

**Estándares de Desarrollo:** Los frameworks suelen imponer estándares de desarrollo que ayudan a mantener un código limpio y consistente.

**Optimización del Rendimiento:** Muchos frameworks incluyen características para la optimización del rendimiento, como el almacenamiento en caché y la carga diferida.

### **Desventajas de los Frameworks de Desarrollo:**

Robles (2023) determina las siguientes desventajas de los Frameworks de Desarrollo:

**Curva de Aprendizaje:** Algunos frameworks pueden tener una curva de aprendizaje empinada, especialmente para desarrolladores principiantes.

**Flexibilidad Limitada:** Los frameworks imponen una estructura y patrones de diseño, lo que puede limitar la flexibilidad y la creatividad en algunos casos.

**Sobrecarga de Abstracción:** Algunos frameworks pueden introducir una sobrecarga de abstracción, lo que significa que se necesita más tiempo para comprender cómo funciona el framework.

**Tamaño del Código:** Los frameworks a menudo generan un código más grande que una aplicación personalizada, lo que puede afectar el rendimiento en proyectos pequeños.

**Actualizaciones y Cambios:** Las actualizaciones o cambios importantes en un framework pueden requerir que se realicen modificaciones significativas en la aplicación.

**Dependencia del Framework:** Al utilizar un framework, la aplicación se vuelve dependiente de ese framework, lo que puede dificultar la migración a otro framework en el futuro.

**Recursos Adicionales:** Algunos frameworks pueden requerir recursos adicionales, como más memoria y capacidad de procesamiento.

## **Generalidades a la seguridad Web.**

Se consideran las siguientes generalidades acerca del seguridad web según lo que establece (*Seguridad informática, 2023*):

### **1. Importancia de la Seguridad Web:**

La seguridad web es fundamental para proteger la confidencialidad, integridad y disponibilidad de los datos y servicios ofrecidos por una aplicación web.

Los ataques cibernéticos, como la inyección SQL, el cross-site scripting (XSS) y el cross-site request forgery (CSRF), pueden explotar vulnerabilidades y comprometer la seguridad de una aplicación web.

### **2. Principales Amenazas y Vulnerabilidades:**

**Inyección SQL:** Un ataque en el que se inserta código SQL malicioso en las entradas de una aplicación web para manipular la base de datos.

**XSS (Cross-Site Scripting):** Permite a un atacante ejecutar scripts maliciosos en el navegador de un usuario.

**CSRF (Cross-Site Request Forgery):** Un ataque en el que se engaña al usuario para que realice acciones no deseadas en una aplicación web sin su consentimiento.

**Inyección de Comandos:** Similar a la inyección SQL, pero se utilizan comandos del sistema para ejecutar acciones no autorizadas.

**Autenticación y Sesiones Débiles:** Las contraseñas débiles y las sesiones no seguras pueden poner en riesgo la seguridad de los usuarios.

**Errores de Configuración:** La mala configuración de servidores web y bases de datos puede exponer información sensible.

### **3. Mejores Prácticas de Seguridad Web:**

**Validación de Datos de Entrada:** Valida y filtra todas las entradas del usuario para prevenir inyecciones y otros ataques.

**Escapado de Datos de Salida:** Escapa los datos antes de mostrarlos en el navegador para prevenir ataques XSS.

**Uso de Parámetros Preparados:** Utiliza consultas preparadas o consultas parametrizadas para evitar la inyección SQL.

**Autenticación y Autorización:** Implementa sistemas de autenticación seguros y controles de acceso adecuados.

**Sesiones Seguras:** Utiliza tokens de sesión seguros y establece límites de tiempo para las sesiones.

**Control de Errores:** No reveles información sensible en mensajes de error públicos.

**Actualizaciones Regulares:** Mantén actualizados los componentes de tu aplicación, incluyendo el sistema operativo, el servidor web y el software de terceros.

**Pruebas de Seguridad:** Realiza pruebas de seguridad regulares, como pruebas de penetración, para identificar y corregir vulnerabilidades.

**Educación y Concienciación:** Capacita a los desarrolladores y usuarios sobre las mejores prácticas de seguridad y los posibles riesgos.

#### **4. Herramientas de Seguridad Web:**

**Web Application Firewalls (WAF):** Ayudan a proteger las aplicaciones web contra ataques comunes mediante la detección y el bloqueo de tráfico malicioso.

**Escáneres de Vulnerabilidades:** Herramientas que identifican vulnerabilidades en una aplicación web, como Nessus y OWASP ZAP.

**Gestores de Contraseñas:** Fomentan el uso de contraseñas fuertes y ayudan a almacenarlas de forma segura.

**Autenticación de Doble Factor (2FA):** Añade una capa adicional de seguridad al requerir una segunda forma de autenticación además de la contraseña

## Herramientas de aplicación de seguridad Web

Hostinger, 2023 hace mención a las herramientas de aplicación de seguridad web:

### **Nessus:**

Descripción: Nessus es una herramienta de escaneo de vulnerabilidades que realiza auditorías de seguridad en aplicaciones y redes. Identifica posibles vulnerabilidades y proporciona recomendaciones para solucionar los problemas detectados.

### **Open Web Application Security Project (OWASP):**

Descripción: OWASP es una organización que proporciona una serie de recursos y herramientas de código abierto para mejorar la seguridad de las aplicaciones web. Esto incluye listas de las 10 principales vulnerabilidades web y herramientas como OWASP ZAP (Zed Attack Proxy), que es una suite de pruebas de seguridad de aplicaciones web.

### **Burp Suite:**

Descripción: Burp Suite es una herramienta de prueba de seguridad ampliamente utilizada para la evaluación de aplicaciones web. Ofrece características como escaneo de seguridad, prueba de penetración, interceptación de solicitudes y respuestas, y análisis de vulnerabilidades.

**Acunetix:**

Descripción: Acunetix es una herramienta de escaneo de seguridad que se utiliza para identificar y solucionar vulnerabilidades en aplicaciones web. Realiza pruebas de seguridad automatizadas y proporciona informes detallados sobre los problemas encontrados.

**Nikto:**

Descripción: Nikto es una herramienta de código abierto que realiza escaneos de seguridad en servidores web para identificar vulnerabilidades y problemas de configuración. Es especialmente útil para detectar problemas comunes en servidores web.

**ModSecurity:**

Descripción: ModSecurity es un firewall de aplicaciones web (WAF) de código abierto que se instala en el servidor web y protege contra ataques comunes, como inyección SQL, XSS y CSRF. Puede ser personalizado para adaptarse a las necesidades específicas de una aplicación.

**W3af:**

Descripción: W3af es un marco de pruebas de seguridad de aplicaciones web que ayuda a identificar y solucionar vulnerabilidades en aplicaciones web. Ofrece escaneo de seguridad automatizado y una amplia gama de complementos.

**Qualys Web Application Scanning (WAS):**

Descripción: Qualys WAS es una herramienta de escaneo y

evaluación de seguridad que detecta y soluciona vulnerabilidades en aplicaciones web. Proporciona informes detallados y recomendaciones de corrección.

### **Zed Attack Proxy (ZAP):**

Descripción: ZAP es una herramienta de código abierto desarrollada por OWASP para pruebas de seguridad de aplicaciones web. Permite el escaneo, la interceptación y el análisis de tráfico web para identificar vulnerabilidades.

### **SecurityHeaders.io:**

Descripción: SecurityHeaders.io es una herramienta en línea que analiza las cabeceras HTTP de una aplicación web y proporciona recomendaciones para mejorar la seguridad. Ayuda a garantizar que las políticas de seguridad adecuadas estén en su lugar.

## **Conclusión**

Los contenidos abordados en la asignatura de Programación de Aplicaciones Web, que incluyen frameworks de desarrollo, definiciones, ventajas y desventajas, así como generalidades de seguridad web y herramientas de aplicación de seguridad web, proporcionan a los estudiantes una comprensión sólida de cómo desarrollar aplicaciones web de manera efectiva y segura en un mundo cada vez más digital y conectado.

Los frameworks de desarrollo, que ofrecen una estructura y un conjunto de herramientas predefinidos, permiten a los de-

sarrolladores acelerar el proceso de desarrollo y mantener una organización coherente en sus proyectos. Los estudiantes han aprendido sobre las definiciones y características clave de estos frameworks, lo que les permitirá elegir y utilizar la herramienta adecuada para sus proyectos web.

Además, han explorado las ventajas y desventajas de los frameworks de desarrollo. Esto les brinda una comprensión equilibrada de cuándo y cómo utilizar estas herramientas, teniendo en cuenta factores como la velocidad de desarrollo, la reutilización de código y la flexibilidad.

En cuanto a la seguridad web, los estudiantes han adquirido conocimientos esenciales sobre las amenazas y vulnerabilidades comunes que enfrentan las aplicaciones web. Han comprendido la importancia de la seguridad web y han aprendido a utilizar herramientas de aplicación de seguridad web para identificar y mitigar estas amenazas. Esto es fundamental para proteger la integridad y la privacidad de los datos en aplicaciones web y garantizar la confianza de los usuarios.

Los contenidos de esta asignatura proporcionan una visión integral de cómo desarrollar aplicaciones web de manera efectiva, eficiente y segura. Los estudiantes han adquirido conocimientos técnicos y conceptuales que les permitirán enfrentar los desafíos en constante evolución del desarrollo web y mantenerse al tanto de las mejores prácticas en seguridad. Esta base sólida los prepara para desempeñar un papel fundamental en la creación y protección de aplicaciones web en el mundo digital actual.

## Autoevaluación

**Pregunta 1: ¿Qué son los frameworks de desarrollo en el contexto de la programación web?**

- a) Herramientas para el diseño gráfico.
- b) Estructuras predefinidas para acelerar el desarrollo de aplicaciones web.
- c) Programas de análisis de datos.
- d) Sistemas operativos para servidores web.
- e) Bases de datos en la nube.

**Pregunta 2: ¿Cuál de los siguientes no es un framework de desarrollo web?**

- a) Ruby on Rails
- b) Django
- c) Spring Boot
- d) MySQL
- e) Laravel

**Pregunta 3: ¿Cuál de los siguientes frameworks utiliza el lenguaje Ruby?**

- a) Django
- b) Laravel
- c) Spring Boot
- d) Ruby on Rails
- e) Express.js

**Pregunta 4: ¿Cuál de los siguientes frameworks se enfoca en el desarrollo web con PHP?**

- a) Django
- b) Laravel
- c) Spring Boot
- d) Ruby on Rails
- e) Express.js

**Pregunta 5: ¿Cuál de los siguientes frameworks es mantenido por Google y se utiliza para el desarrollo web front-end?**

- a) React
- b) Flask
- c) Vue.js
- d) Express.js
- e) Spring Boot

**Pregunta 6: ¿Cuál de los siguientes frameworks se utiliza comúnmente para crear aplicaciones web y APIs con JavaScript (Node.js)?**

- a) Ruby on Rails
- b) Laravel
- c) Spring Boot
- d) Ruby on Rails
- e) Express.js

**Pregunta 7: ¿Cuál de los siguientes frameworks es un framework de desarrollo web minimalista de Python?**

- a) Django
- b) Flask

- c) Spring Boot
- d) Ruby on Rails
- e) Laravel

**Pregunta 8:** ¿Qué consideración es importante al elegir un framework de desarrollo?

- a) Color de la interfaz de usuario.
- b) La popularidad del framework.
- c) Tamaño del monitor.
- d) Tipo de CPU.
- e) Tipo de aplicación.

**Pregunta 9:** ¿Qué desventaja común de los frameworks de desarrollo se refiere a que pueden limitar la flexibilidad y la creatividad?

- a) Sobrecarga de abstracción.
- b) Curva de aprendizaje empinada.
- c) Tamaño del código.
- d) Actualizaciones y cambios.
- e) Dependencia del Framework.

**Pregunta 10:** ¿Cuál de las siguientes no es una mejor práctica de seguridad web?

- a) Validación de datos de entrada.
- b) Autenticación y autorización.
- c) Sesiones seguras.
- d) Escapado de datos de salida.
- e) Publicación de información sensible en mensajes de error.

**Pregunta 11: ¿Qué herramienta se utiliza para realizar auditorías de seguridad en aplicaciones y redes y proporciona recomendaciones para solucionar problemas?**

- a) Burp Suite.
- b) Nikto.
- c) Nessus.
- d) Zed Attack Proxy (ZAP).
- e) Flask.

**Pregunta 12: ¿Cuál de las siguientes herramientas es un firewall de aplicaciones web (WAF) de código abierto que protege contra ataques comunes?**

- a) Acunetix.
- b) ModSecurity.
- c) W3af.
- d) Burp Suite.
- e) Vue.js.

**Pregunta 13: ¿Cuál de las siguientes herramientas es una suite de pruebas de seguridad de aplicaciones web desarrollada por OWASP?**

- a) Burp Suite.
- b) Nikto.
- c) Nessus.
- d) Zed Attack Proxy (ZAP).
- e) SecurityHeaders.io.

**Pregunta 14: ¿Qué herramienta es ampliamente utilizada para la evaluación de aplicaciones web, incluyendo escaneo de seguridad y prueba de penetración?**

- a) Nessus.
- b) Acunetix.
- c) Burp Suite.
- d) W3af.
- e) ModSecurity.

**Pregunta 15: ¿Cuál de las siguientes herramientas se utiliza para identificar y solucionar vulnerabilidades en aplicaciones web, con pruebas de seguridad automatizadas?**

- a) Qualys Web Application Scanning (WAS).
- b) Zed Attack Proxy (ZAP).
- c) SecurityHeaders.io.
- d) Acunetix.
- e) Nikto.

**Pregunta 16: ¿Cuál de las siguientes herramientas se utiliza para realizar escaneos de seguridad en servidores web y detectar vulnerabilidades?**

- a) Nikto.
- b) Zed Attack Proxy (ZAP).
- c) Burp Suite.
- d) ModSecurity.
- e) Qualys Web Application Scanning (WAS).

**Pregunta 17: ¿Cuál de las siguientes organizaciones proporciona recursos y herramientas de código abierto para mejorar la seguridad de las aplicaciones web?**

- a) Nessus.
- b) Acunetix.
- c) OWASP.
- d) Qualys.
- e) W3af.

**Pregunta 18: ¿Qué tipo de amenaza web se refiere a un ataque en el que se inserta código SQL malicioso en las entradas de una aplicación web?**

- a) Inyección SQL.
- b) XSS.
- c) CSRF.
- d) Inyección de comandos.
- e) Autenticación débil.

**Pregunta 19: ¿Qué desventaja común de los frameworks de desarrollo se refiere al hecho de que pueden generar un código más grande en comparación con una aplicación personalizada?**

- a) Sobrecarga de abstracción.
- b) Curva de aprendizaje empinada.
- c) Tamaño del código.
- d) Actualizaciones y cambios.
- e) Dependencia del Framework.

**Pregunta 20: ¿Cuál de las siguientes no es una ventaja de los frameworks de desarrollo?**

- a) Mantenimiento simplificado.
- b) Escalabilidad.
- c) Flexibilidad ilimitada.
- d) Reutilización de código.
- e) Estándares de desarrollo.



## **Capítulo 4**

Herramientas de identificación de vulnerabilidades

Los desarrolladores y profesionales de seguridad deben entender cómo funcionan estas herramientas y cómo interpretar sus resultados para garantizar que sus aplicaciones sean resistentes a las amenazas cibernéticas (CODSTER, 2023).

## Aspectos generales

### 1. Tipos de Vulnerabilidades Detectadas:

Las herramientas de identificación de vulnerabilidades pueden detectar una amplia gama de problemas, como inyección SQL, XSS (Cross-Site Scripting), CSRF (Cross-Site Request Forgery), autenticación débil, configuración insegura, entre otros.

### 2. Escaneo Automatizado:

La mayoría de estas herramientas realizan escaneos automatizados de la aplicación, lo que significa que pueden evaluar grandes cantidades de código y configuración en poco tiempo. Esto es especialmente valioso en aplicaciones web complejas.

### 3. Análisis de Código y Red:

Algunas herramientas se centran en el análisis estático del código fuente de la aplicación, mientras que otras se enfocan en el análisis dinámico de la aplicación en tiempo de ejecución. Algunas herramientas también pueden combinar ambos enfoques.

### 4. Integración en el Ciclo de Desarrollo:

En un enfoque de desarrollo seguro, estas herramientas se utilizan durante todo el ciclo de vida del desarrollo de software. Se pueden integrar en las fases de desarrollo, pruebas y produc-

ción para identificar y solucionar problemas de seguridad a lo largo del tiempo.

### **5. Generación de Informes:**

Las herramientas de identificación de vulnerabilidades suelen generar informes detallados que incluyen una descripción de las vulnerabilidades encontradas, su gravedad y recomendaciones para corregirlas. Estos informes son valiosos para los equipos de desarrollo y seguridad.

### **6. Escaneo Continuo:**

Algunas herramientas permiten el escaneo continuo o programado de la aplicación, lo que facilita la detección de nuevas vulnerabilidades a medida que se introducen cambios en el código o en la configuración.

### **7. Ajuste y Personalización:**

Las herramientas suelen permitir cierto grado de ajuste y personalización para adaptarse a las necesidades específicas de la aplicación o el entorno. Esto puede incluir la definición de políticas de seguridad personalizadas.

### **8. Evolución y Actualizaciones:**

Dado que las amenazas y las vulnerabilidades evolucionan con el tiempo, las herramientas también se actualizan regularmente para abordar nuevas amenazas y proporcionar un escaneo más preciso.

## **9. Acceso a Base de Datos:**

Algunas herramientas tienen la capacidad de acceder a la base de datos de la aplicación, lo que les permite detectar vulnerabilidades de inyección SQL y otras amenazas relacionadas con la base de datos.

## **10. Compatibilidad con Diferentes Tecnologías:**

Las herramientas de identificación de vulnerabilidades son compatibles con una variedad de tecnologías web, incluidos lenguajes de programación, marcos de desarrollo y sistemas operativos.

## **Herramientas de aplicación**

### ***Tipos de Vulnerabilidades Detectadas:***

Las herramientas de identificación de vulnerabilidades pueden detectar una amplia gama de problemas, como inyección SQL, XSS (Cross-Site Scripting), CSRF (Cross-Site Request Forgery), autenticación débil, configuración insegura, entre otros.

### **Escaneo Automatizado:**

La mayoría de estas herramientas realizan escaneos automatizados de la aplicación, lo que significa que pueden evaluar grandes cantidades de código y configuración en poco tiempo. Esto es especialmente valioso en aplicaciones web complejas.

### **Análisis de Código y Red:**

Algunas herramientas se centran en el análisis estático del código fuente de la aplicación, mientras que otras se enfocan en el análisis dinámico de la aplicación en tiempo de ejecución. Algunas herramientas también pueden combinar ambos enfoques.

### **Integración en el Ciclo de Desarrollo:**

En un enfoque de desarrollo seguro, estas herramientas se utilizan durante todo el ciclo de vida del desarrollo de software. Se pueden integrar en las fases de desarrollo, pruebas y producción para identificar y solucionar problemas de seguridad a lo largo del tiempo.

### **Generación de Informes:**

Las herramientas de identificación de vulnerabilidades suelen generar informes detallados que incluyen una descripción de las vulnerabilidades encontradas, su gravedad y recomendaciones para corregirlas. Estos informes son valiosos para los equipos de desarrollo y seguridad.

### **Escaneo Continuo:**

Algunas herramientas permiten el escaneo continuo o programado de la aplicación, lo que facilita la detección de nuevas vulnerabilidades a medida que se introducen cambios en el código o en la configuración.

### **Ajuste y Personalización:**

Las herramientas suelen permitir cierto grado de ajuste y

personalización para adaptarse a las necesidades específicas de la aplicación o el entorno. Esto puede incluir la definición de políticas de seguridad personalizadas.

### **Evolución y Actualizaciones:**

Dado que las amenazas y las vulnerabilidades evolucionan con el tiempo, las herramientas también se actualizan regularmente para abordar nuevas amenazas y proporcionar un escaneo más preciso.

### **Acceso a Base de Datos:**

Algunas herramientas tienen la capacidad de acceder a la base de datos de la aplicación, lo que les permite detectar vulnerabilidades de inyección SQL y otras amenazas relacionadas con la base de datos.

### **Compatibilidad con Diferentes Tecnologías:**

Las herramientas de identificación de vulnerabilidades son compatibles con una variedad de tecnologías web, incluidos lenguajes de programación, marcos de desarrollo y sistemas operativos.

## **Aplicaciones para verificar la accesibilidad de sitios Web**

La accesibilidad web es un aspecto crítico del diseño y desarrollo de sitios web, ya que garantiza que todas las personas, incluidas aquellas con discapacidades, puedan acceder y utilizar los

sitios de manera efectiva. Para verificar la accesibilidad de sitios web, existen herramientas y aplicaciones específicas que permiten evaluar y mejorar la conformidad con las pautas de accesibilidad, como las Pautas de Accesibilidad al Contenido en la Web (WCAG) lo establece (SocialCo, 2023).

### **WAVE (Web Accessibility Evaluation Tool):**

Descripción: WAVE es una extensión de navegador que proporciona una evaluación detallada de la accesibilidad de una página web. Identifica problemas de accesibilidad y sugiere correcciones.

### **axe DevTools (anteriormente aXe):**

Descripción: axe DevTools es una extensión de navegador que se integra con las herramientas de desarrollador de Chrome y Firefox. Ofrece análisis de accesibilidad en tiempo real y ayuda a identificar y solucionar problemas de accesibilidad.

### **Pally:**

Descripción: Pally es una herramienta de línea de comandos que permite automatizar pruebas de accesibilidad. Puede ser utilizado en la línea de comandos o como parte de un flujo de trabajo de integración continua.

### **Siteimprove:**

Descripción: Siteimprove es una plataforma de gestión de la calidad web que incluye un módulo de accesibilidad. Proporciona informes de accesibilidad detallados y seguimiento continuo.

**SortSite:**

Descripción: SortSite es una herramienta de escritorio que verifica la accesibilidad y la calidad de un sitio web. Ofrece informes detallados y recomendaciones de corrección.

**Tenon.io:**

Descripción: Tenon.io es una plataforma en línea que realiza pruebas de accesibilidad en sitios web. Ofrece una interfaz de usuario amigable y proporciona informes detallados.

**AInspector Sidebar:**

Descripción: AInspector Sidebar es una extensión de Firefox que evalúa la accesibilidad de una página web y muestra los resultados en un panel lateral.

**Accessibility Insights:**

Descripción: Accessibility Insights es una extensión de navegador desarrollada por Microsoft que proporciona información detallada sobre la accesibilidad de una página web y ofrece consejos para solucionar problemas.

**WAI-ARIA Validator:**

Descripción: WAI-ARIA Validator es una herramienta en línea que verifica el uso adecuado de atributos ARIA (Accessible Rich Internet Applications) en una página web para mejorar la accesibilidad.

### **Contrast Checker:**

**Descripción:** Esta herramienta se enfoca en verificar el contraste de colores en una página web para garantizar que el texto sea legible para personas con discapacidades visuales.

### **Publicación de sitios web.**

Theastrology. (2023) determina acerca de la publicación de sitios web:

#### **1. Elección de un Proveedor de Alojamiento Web:**

Antes de publicar un sitio web, debes seleccionar un proveedor de alojamiento web. Estos proveedores ofrecen servidores donde puedes alojar tu sitio web y hacerlo accesible en línea. Algunos ejemplos populares incluyen Bluehost, HostGator, SiteGround y AWS.

#### **2. Registro de un Dominio:**

Para que tu sitio web sea accesible a través de una dirección web única, debes registrar un nombre de dominio. Esto permite a los usuarios acceder a tu sitio escribiendo el nombre de dominio en un navegador web. Puedes registrar dominios a través de registradores de dominios, como GoDaddy o Namecheap.

#### **3. Preparación de Archivos del Sitio:**

Asegúrate de que todos los archivos y recursos necesarios para tu sitio web estén listos. Esto incluye el código HTML, CSS, imágenes, archivos JavaScript y otros activos multimedia.

#### **4. Configuración del Servidor Web:**

En el servidor web que has elegido, debes configurar la infraestructura necesaria, como servidores HTTP (como Apache o Nginx), bases de datos (si es necesario), y otros servicios relevantes para tu aplicación.

#### **5. Transferencia de Archivos:**

Para publicar tu sitio web, debes transferir los archivos desde tu entorno local al servidor web. Esto se puede hacer a través de FTP (Protocolo de Transferencia de Archivos), SSH (Secure Shell) o utilizando herramientas integradas de gestión de archivos en tu panel de control de alojamiento web.

#### **6. Configuración de DNS:**

Debes configurar los registros DNS (Sistema de Nombres de Dominio) para que el nombre de dominio apunte a la dirección IP de tu servidor web. Esto se hace a través del panel de control de tu registrador de dominios o de tu proveedor de alojamiento web.

#### **7. Pruebas y Depuración:**

Antes de anunciar tu sitio web al público, realiza pruebas exhaustivas para asegurarte de que todo funcione correctamente. Verifica enlaces rotos, errores de código y la compatibilidad con múltiples navegadores.

#### **8. Implementación de Seguridad:**

Asegúrate de implementar medidas de seguridad en tu ser-

vidor web para proteger tu sitio contra amenazas cibernéticas. Esto puede incluir la instalación de certificados SSL para habilitar HTTPS y la configuración de cortafuegos y sistemas de prevención de intrusiones.

### **9. Monitorización y Mantenimiento Continuo:**

Después de publicar tu sitio web, es importante monitorear su rendimiento y seguridad de forma continua. Realiza actualizaciones de software y realiza copias de seguridad regulares.

### **10. Promoción y SEO:**

- Una vez que tu sitio web esté en línea, promócelo a través de canales de redes sociales, publicidad en línea y estrategias de SEO (Optimización de Motores de Búsqueda) para aumentar su visibilidad.

## **Trabajo con la gestión de Proyectos**

Asana, 2022 define acerca del trabajo con la gestión de proyectos:

### **1. Definición de Objetivos y Alcance del Proyecto:**

Antes de comenzar un proyecto, es esencial definir sus objetivos y alcance. Esto incluye determinar qué características se desarrollarán, cuáles son los requisitos y las expectativas del cliente, y cuál es el calendario previsto.

## **2. Planificación del Proyecto:**

La planificación es una parte crucial de la gestión de proyectos. Debes definir tareas, asignar recursos y estimar el tiempo necesario para completar cada fase del proyecto. Las herramientas de gestión de proyectos, como Microsoft Project o Trello, pueden ser útiles para crear planes detallados.

## **3. Gestión de Recursos:**

La gestión de recursos implica asignar adecuadamente a los miembros del equipo, determinar qué tecnologías y herramientas se utilizarán y garantizar que haya suficientes recursos disponibles para completar el proyecto a tiempo.

## **4. Seguimiento y Control:**

A medida que el proyecto avanza, es importante realizar un seguimiento de su progreso y controlar cualquier desviación con respecto al plan original. Esto implica evaluar el estado del proyecto, gestionar riesgos y tomar medidas correctivas cuando sea necesario.

## **5. Comunicación Efectiva:**

La comunicación es esencial en la gestión de proyectos. Debes mantener a todas las partes interesadas informadas sobre el progreso, los problemas y las decisiones importantes. Las reuniones regulares, informes de estado y herramientas de colaboración son útiles para mantener una comunicación efectiva.

## **6. Gestión de Riesgos:**

Identificar y gestionar riesgos es una parte integral de la gestión de proyectos. Debes anticipar posibles obstáculos y tener planes de contingencia en caso de que ocurran problemas inesperados.

## **7. Control de Calidad y Pruebas:**

La gestión de proyectos debe incluir la planificación y ejecución de pruebas de calidad para garantizar que la aplicación web cumpla con los estándares de rendimiento y funcionalidad. Esto puede incluir pruebas de seguridad, pruebas de usabilidad y pruebas de rendimiento.

## **8. Entrega y Despliegue:**

La fase de entrega y despliegue implica la implementación de la aplicación web en un entorno de producción. Esto incluye la configuración de servidores, la instalación de software y la verificación de que el sitio web esté funcionando correctamente.

## **9. Evaluación Posterior al Proyecto:**

Después de que el proyecto esté completo, se debe realizar una evaluación para identificar lecciones aprendidas y áreas de mejora. Esto es esencial para la mejora continua de los procesos de gestión de proyectos.

## **10. Documentación y Mantenimiento:**

La documentación detallada de la aplicación web y los procesos utilizados en el proyecto son esenciales. Además, el mante-

nimiento continuo del sitio web, que incluye actualizaciones de seguridad y correcciones de errores, es importante después de la implementación.

## **Seguimiento del Desarrollo de Proyectos**

### **1. Planificación Inicial:**

Antes de comenzar cualquier proyecto, es fundamental realizar una planificación detallada. Esto incluye definir los objetivos del proyecto, los entregables esperados, el alcance, el cronograma y los recursos necesarios. Un plan bien estructurado servirá como punto de referencia para el seguimiento posterior.

### **2. Establecimiento de Indicadores Clave de Rendimiento (KPI):**

Identifica los KPI que se utilizarán para medir el progreso y el éxito del proyecto. Los KPI pueden incluir plazos de entrega, calidad del código, número de errores encontrados y otros indicadores específicos para tu proyecto.

### **3. Uso de Herramientas de Gestión de Proyectos:**

Las herramientas de gestión de proyectos, como Microsoft Project, Trello, Asana o Jira, son útiles para realizar un seguimiento del progreso del proyecto. Estas herramientas permiten crear planes, asignar tareas, establecer fechas límite y colaborar en equipo.

### **4. Reuniones de Seguimiento Regulares:**

Programa reuniones de seguimiento regulares con el equipo de desarrollo para evaluar el progreso y abordar cualquier pro-

blema que pueda surgir. Estas reuniones proporcionan un espacio para la comunicación y la resolución de problemas.

### **5. Control de Versiones y Repositorios de Código:**

Utiliza sistemas de control de versiones, como Git, y repositorios de código, como GitHub o GitLab, para rastrear los cambios en el código fuente. Esto ayuda a mantener un registro de las modificaciones y facilita la colaboración entre desarrolladores.

### **6. Seguimiento del Avance del Proyecto:**

Haz un seguimiento regular del progreso del proyecto en relación con el plan original. Esto incluye el seguimiento del cumplimiento de las fechas límite, la realización de pruebas y la corrección de errores.

### **7. Control de Calidad y Pruebas:**

Realiza pruebas de calidad en cada etapa del desarrollo para garantizar que el proyecto cumple con los estándares de rendimiento y funcionalidad. Identifica y soluciona problemas a medida que se descubren.

### **8. Documentación:**

Documenta todas las etapas del desarrollo, incluidos los cambios realizados, las decisiones clave y los problemas resueltos. La documentación es esencial para el seguimiento y la evaluación del proyecto.

### **9. Evaluación de Riesgos:**

Identifica y evalúa los riesgos a lo largo del proyecto y desarrolla planes de contingencia para abordarlos en caso de que se materialicen.

### **10. Comunicación Efectiva:**

Mantén una comunicación efectiva con todas las partes interesadas, incluidos los clientes y los miembros del equipo. Proporciona actualizaciones periódicas y aborda cualquier preocupación o solicitud.

### **11. Análisis de Desviaciones:**

Analiza las desviaciones del plan original y determina las causas de estas desviaciones. Esto te permitirá tomar medidas correctivas para mantener el proyecto en el camino correcto.

### **12. Actualización del Plan:**

A medida que el proyecto avanza, actualiza el plan de proyecto según sea necesario. Asegúrate de que refleje con precisión el estado actual del proyecto y las fechas límite revisadas.

## **Conclusión**

Los contenidos abordados en la asignatura de Programación de Aplicaciones Web relacionados con frameworks de desarrollo, definiciones, ventajas y desventajas, así como generalidades de seguridad web y herramientas de aplicación de seguridad web, han proporcionado a los estudiantes una comprensión integral de aspectos críticos en el desarrollo web y la protección de aplicaciones en línea.

Los frameworks de desarrollo se presentan como herramientas esenciales para simplificar y agilizar el proceso de creación de aplicaciones web, brindando una estructura predefinida

y conjuntos de herramientas que permiten a los desarrolladores trabajar de manera más eficiente. Los estudiantes han adquirido una apreciación de la diversidad de frameworks disponibles y cómo seleccionar el más adecuado para sus proyectos específicos.

Además, han explorado en profundidad las ventajas y desventajas asociadas con el uso de frameworks, lo que les permite tomar decisiones informadas sobre cuándo y cómo aplicar estas herramientas en sus proyectos. La comprensión de las ventajas, como la velocidad de desarrollo y la consistencia, junto con las desventajas, como la curva de aprendizaje y la posible restricción de flexibilidad, les brinda una visión equilibrada y crítica.

La seguridad web se ha destacado como un aspecto fundamental en la creación de aplicaciones web. Los estudiantes han adquirido conocimientos sobre las amenazas comunes que enfrentan las aplicaciones web y cómo aplicar medidas de seguridad efectivas. La conciencia de la importancia de la seguridad web y el conocimiento de herramientas específicas para identificar y abordar vulnerabilidades son competencias cruciales en la actualidad, dada la creciente sofisticación de las amenazas cibernéticas.

Esta asignatura ha equipado a los estudiantes con conocimientos y habilidades esenciales para desarrollar aplicaciones web de manera eficaz, al tiempo que se centra en la seguridad y la protección de datos. Los conocimientos adquiridos en esta asignatura sientan las bases para futuros proyectos y carreras en el campo de la programación de aplicaciones web, que requiere un equilibrio sólido entre funcionalidad y seguridad en un entorno digital en constante evolución.

## Autoevaluación

**Pregunta 1: ¿Qué tipos de vulnerabilidades pueden detectar las herramientas de identificación de vulnerabilidades?**

- a) Vulnerabilidades de diseño gráfico.
- b) Vulnerabilidades de base de datos.
- c) Vulnerabilidades de escalabilidad.
- d) Vulnerabilidades de inyección SQL.
- e) Vulnerabilidades de hardware.

**Pregunta 2: ¿Qué enfoque de análisis utilizan algunas herramientas de identificación de vulnerabilidades?**

- a) Análisis estático del código fuente.
- b) Análisis dinámico de la aplicación en tiempo de ejecución.
- c) Análisis de diseño gráfico.
- d) Análisis de redes sociales.
- e) Análisis de configuración de hardware.

**Pregunta 3: ¿Qué etapas del ciclo de desarrollo de software pueden beneficiarse de la integración de herramientas de identificación de vulnerabilidades?**

- a) Fase de desarrollo.
- b) Fase de prueba.
- c) Fase de producción.
- d) Todas las anteriores.
- e) Ninguna de las anteriores.

**Pregunta 4: ¿Qué información se incluye en los informes generados por las herramientas de identificación de vulnerabilidades?**

- a) Recetas de cocina.
- b) Gravedad de las vulnerabilidades encontradas.

- c) Pronósticos del tiempo.
- d) Noticias de entretenimiento.
- e) Opiniones de películas.

**Pregunta 5: ¿Qué función pueden desempeñar algunas herramientas de identificación de vulnerabilidades en relación con la base de datos de una aplicación?**

- a) Acceder y gestionar la base de datos.
- b) Evaluar la velocidad de la base de datos.
- c) Redactar informes financieros.
- d) Realizar tareas de diseño gráfico.
- e) Preparar café.

**Pregunta 6: ¿Qué herramientas se utilizan para evaluar y mejorar la conformidad de un sitio web con las pautas de accesibilidad, como WCAG?**

- a) Herramientas de diseño gráfico.
- b) Herramientas de análisis financiero.
- c) Herramientas de análisis de velocidad de carga.
- d) Herramientas de identificación de vulnerabilidades.
- e) Herramientas de verificación de la accesibilidad.

**Pregunta 7: ¿Cuál de las siguientes herramientas de verificación de la accesibilidad es una extensión de navegador que proporciona una evaluación detallada de la accesibilidad de una página web?**

- a) Wave (Web Accessibility Evaluation Tool).
- b) Siteimprove.
- c) Tenon.io.
- d) SortSite.
- e) AInspector Sidebar.

**Pregunta 8: ¿Qué fase del desarrollo de software implica la configuración del servidor web?**

- a) Fase de planificación.
- b) Fase de análisis.
- c) Fase de implementación.
- d) Fase de diseño gráfico.
- e) Fase de prueba.

**Pregunta 9: ¿Cuál de las siguientes no es una etapa común en el ciclo de desarrollo de proyectos?**

- a) Planificación del proyecto.
- b) Implementación de seguridad.
- c) Evaluación posterior al proyecto.
- d) Seguimiento del progreso del proyecto.
- e) Evaluación de riesgos.

**Pregunta 10: ¿Qué debe hacerse antes de transferir los archivos de un sitio web desde un entorno local al servidor web?**

- a) Realizar pruebas exhaustivas.
- b) Realizar una evaluación posterior al proyecto.
- c) Regresar al análisis de riesgos.
- d) Configurar el servidor web.
- e) Hacer una copa de café.

**Pregunta 11: ¿Cuál es el propósito de la comunicación efectiva en la gestión de proyectos?**

- a) Mejorar la velocidad de carga del sitio web.
- b) Evaluar la conformidad con las pautas de accesibilidad.
- c) Comunicar los resultados de la evaluación de vulnerabilidades.
- d) Mantener a todas las partes interesadas informadas so-

bre el progreso y los problemas del proyecto.

e) Realizar pruebas de calidad.

**Pregunta 12: ¿Qué se utiliza para rastrear los cambios en el código fuente de un proyecto?**

a) Herramientas de análisis financiero.

b) Repositorios de código.

c) Análisis de rendimiento del servidor web.

d) Herramientas de análisis de velocidad de carga.

e) Medidores de temperatura del hardware.

**Pregunta 13: ¿Qué fase del desarrollo implica la configuración de servidores, la instalación de software y la verificación de que el sitio web funcione correctamente?**

a) Fase de implementación de seguridad.

b) Fase de evaluación posterior al proyecto.

c) Fase de pruebas y depuración.

d) Fase de entrega y despliegue.

e) Fase de control de calidad.

**Pregunta 14: ¿Por qué es importante la evaluación posterior al proyecto en la gestión de proyectos?**

a) Para mantener a todas las partes interesadas informadas sobre el progreso.

b) Para evaluar la conformidad con las pautas de accesibilidad.

c) Para identificar lecciones aprendidas y áreas de mejora.

d) Para configurar servidores.

e) Para realizar pruebas de rendimiento.

**Pregunta 15: ¿Qué es esencial para el seguimiento y la evaluación del proyecto?**

- a) Herramientas de análisis financiero.
- b) Comunicación efectiva.
- c) Análisis de velocidad de carga del sitio web.
- d) Herramientas de diseño gráfico.
- e) Programar reuniones de seguimiento.

**Pregunta 16: ¿Cuál es una función de las reuniones de seguimiento en la gestión de proyectos?**

- a) Analizar el contraste de colores en una página web.
- b) Identificar y solucionar problemas de vulnerabilidad.
- c) Evaluar el rendimiento de la base de datos.
- d) Evaluar el progreso del proyecto y abordar problemas.
- e) Realizar análisis de riesgos.

**Pregunta 17: ¿Qué hacen las herramientas de identificación de vulnerabilidades?**

- a) Evaluación de velocidad de carga.
- b) Evaluación de la accesibilidad.
- c) Pruebas de seguridad en línea.
- d) Evaluación de riesgos.
- e) Evaluación de vulnerabilidades de seguridad.

**Pregunta 18: ¿Cuál es una responsabilidad importante en la fase de publicación de sitios web?**

- a) Planificación del proyecto.
- b) Realizar pruebas exhaustivas.
- c) Configuración del servidor web.
- d) Registro de un dominio.
- e) Promoción y SEO.

**Pregunta 19: ¿Cuál de las siguientes tareas es parte de la planificación de un proyecto?**

- a) Realizar una evaluación posterior al proyecto.
- b) Configuración de servidores.
- c) Planificación de la fase de entrega y despliegue.
- d) Definición de objetivos y alcance del proyecto.
- e) Implementación de seguridad.

**Pregunta 20: ¿Cuál es una función esencial de las herramientas de verificación de accesibilidad de sitios web?**

- a) Evaluación de velocidad de carga.
- b) Identificación de vulnerabilidades de seguridad.
- c) Verificar el contraste de colores en una página web.
- d) Evaluar la conformidad con las pautas de accesibilidad.
- e) Realizar análisis de riesgos.



## **Referencias**

- Axarnet. (2023). *Lenguajes del lado del servidor (Características)*. <https://axarnet.es/blog/lenguajes-del-lado-del-servidor#>
- Casale, J.C. (2016). Introducción a la programación. Manual del Desarrollador. *RU RedUsers*. <https://n9.cl/yrvvc5>
- Casillas Santillán, L.A., Ginestà, M.G., y Pérez Mora, Ó. (2023). Bases de datos en MySQL. *El Hacker*. <https://acortar.link/ZFucVg>
- EINA Virtual. (s.f.). *Integración de HTML, CSS y PHP*. (2023). <https://einavirtual.com/php/integracion-php>
- Gutiérrez, E. (s.f.). ¿Cuáles son las herramientas de evaluación de vulnerabilidades? *CODSTER*. <https://acortar.link/rWsjm>
- Hostinger Tutoriales. (2023, 13 de febrero). *Seguridad en aplicaciones web: Qué es, cómo funciona y los mejores servicios*. <https://www.hostinger.es/tutoriales/seguridad-en-aplicaciones-web>
- IMMUNE Technology Institute. (2023, 6 de marzo). *¿Qué es un framework? Mejores opciones para aprender en 2023*. <https://immune.institute/blog/que-es-un-framework-desarrollo/>
- Molina, J., Zea, M., Redrován, F., Loja, N., Valarezo, M., & Honores, J. (2018). *SNAIL, Una metodología híbrida para el desarrollo de aplicaciones web*. 3Ciencias.
- Mozilla Developer Network. (2023). *Fundamentos de texto y fuentes tipográficas*. [https://developer.mozilla.org/es/docs/Learn/CSS/Styling\\_text/Fundamentals](https://developer.mozilla.org/es/docs/Learn/CSS/Styling_text/Fundamentals)
- Pérez, S., Quispe, J., Mullicundo, F., & Lamas, D. (2021). *Herramientas y tecnologías para el desarrollo web desde el frontend al backend*. Universidad Nacional de la Plata. <http://sedici.unlp.edu.ar/handle/10915/119487>
- PHP. (s.f.). *¿Qué es PHP?* <https://www.php.net/manual/es/intro-what-is.php>

- Raeburn, A. (2023, 8 de noviembre). Los 12 mejores software de gestión de proyectos en 2024. *Asana*. <https://asana.com/es/resources/best-project-management-software>
- Readthedocs. (s.f.). *Gestión de Bases de Datos*. <https://gestionbases-datos.readthedocs.io/es/latest/Tema1/Teoria.html>
- Robles, V. (2023). *Pros y contras de los frameworks de desarrollo web*. <https://acortar.link/C83kSy>
- SocialCo. (2022, 14 de marzo). *¿Qué herramientas nos pueden ayudar a mejorar la accesibilidad en un sitio web?* <https://acortar.link/HCwm9E>
- Theastrology. (2023). *¿Qué es la publicación web?–Definición de techopedia*. <https://es.theastrologypage.com/web-publishing>
- Torrejón, M. (2023, marzo 30). Ventajas de los frameworks. *El blog de Omatech*. <https://www.omatech.com/blog/2023/03/30/ventajas-frameworks/>
- Vegas J. (2002). Aplicaciones Multinivel. *Infor*. <https://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node21.html>
- W3Schools. (s.f.). *CSS Text*. [https://www.w3schools.com/css/css\\_text.asp](https://www.w3schools.com/css/css_text.asp)



## **Anexos**

## Anexo 1 Solucionario Unidad 1

**Pregunta 1: ¿Cuál de los siguientes no es un aspecto clave a considerar en el desarrollo web, según Molina et al., (2018)?**

- a) Tecnología utilizada
- b) Diseño de interacción
- c) Tendencias de marketing
- d) Contenido de alta calidad
- e) Optimización de servidores

Respuesta correcta: c) Tendencias de marketing

**Pregunta 2: ¿Qué lenguaje de marcado se utiliza en la construcción de la estructura y contenido de las páginas web?**

- a) JavaScript
- b) CSS
- c) HTML
- d) PHP
- e) Python

Respuesta correcta: c) HTML

**Pregunta 3: ¿Qué función cumple el Modelo de Objetos del Documento (DOM) en el desarrollo web?**

- a) Definir la estructura de la página web
- b) Almacenar datos en el servidor
- c) Manipular elementos de la página en tiempo real
- d) Diseñar la interfaz de usuario
- e) Encriptar datos sensibles

Respuesta correcta: c) Manipular elementos de la página en tiempo real

**Pregunta 4: Según el texto, ¿por qué es importante crear una estructura lógica y semánticamente coherente en una página web?**

- a) Para aumentar el rendimiento del servidor
- b) Para facilitar la comprensión de los usuarios con discapacidades

- c) Para acelerar la carga de la página
- d) Para mejorar la seguridad del sitio web
- e) Para obtener un mejor posicionamiento en los motores de búsqueda

Respuesta correcta: b) Para facilitar la comprensión de los usuarios con discapacidades

**Pregunta 5: ¿Cuál de las siguientes etiquetas HTML se utiliza para definir una lista desordenada?**

- a) <li>
- b) <ul>
- c) <ol>
- d) <p>
- e) <dl>

Respuesta correcta: b) <ul>

**Pregunta 6: ¿Cuál de las siguientes tecnologías se asocia principalmente con el FrontEnd en desarrollo web?**

- a) PHP
- b) HTML
- c) Node.js
- d) MySQL
- e) Java

Respuesta correcta: b) HTML

**Pregunta 7: ¿Cuál de las siguientes tecnologías se utiliza para dar estilo a una página web?**

- a) HTML
- b) JavaScript
- c) CSS
- d) PHP
- e) XML

Respuesta correcta: c) CSS

**Pregunta 8: ¿Qué se entiende por “SEO” en el desarrollo web?**

- a) Search Engine Optimization (Optimización de Motores de Búsqueda)
- b) Sistema de Enlaces y Opciones
- c) Servicio de Edición en Línea
- d) Sociedad de Empresas Online
- e) Software de Entretenimiento en Línea

Respuesta correcta: a) Search Engine Optimization (Optimización de Motores de Búsqueda)

**Pregunta 9: ¿Qué etiqueta HTML se utiliza para definir un título para un documento web?**

- a) <title>
- b) <head>
- c) <h1>
- d) <p>
- e) <header>

Respuesta correcta: a) <title>

**Pregunta 10: ¿Cuál de las siguientes tecnologías se utiliza para crear aplicaciones web interactivas y dinámicas en el lado del cliente?**

- a) HTML
- b) CSS
- c) MySQL
- d) DOM
- e) JavaScript

Respuesta correcta: e) JavaScript

**Pregunta 11: ¿Cuál es la principal área del sitio web donde se suele ubicar el logotipo y el enlace a la página principal?**

- a) Columna lateral izquierda
- b) Área de contenido principal
- c) Cabecera
- d) Pie de página

e) Zona de navegación superior

Respuesta correcta: c) Cabecera

**Pregunta 12: ¿Qué atributo de HTML se utiliza para proporcionar una descripción de una imagen en una etiqueta de imagen (<img>)?**

a) src

b) alt

c) href

d) style

e) title

Respuesta correcta: b) alt

**Pregunta 13: ¿Cuál de las siguientes opciones no es una forma de enlazar hojas de estilo CSS a un documento HTML?**

a) Hoja de estilo interna

b) Añadir CSS al principio del HTML

c) Añadir CSS al final del HTML

d) Integrar el CSS con un archivo externo

e) Hoja de estilo inline

Respuesta correcta: c) Añadir CSS al final del HTML

**Pregunta 14: ¿Qué atributo se utiliza comúnmente en las etiquetas de enlace (<a>) para especificar la URL de destino?**

a) src

b) style

c) alt

d) href

e) rel

Respuesta correcta: d) href

**Pregunta 15: ¿En qué parte de la estructura de una página web se ubica generalmente el menú de navegación principal?**

a) Zona de navegación superior

b) Área de contenido principal

- c) Columna lateral derecha
- d) Pie de página
- e) Columna lateral izquierda

Respuesta correcta: a) Zona de navegación superior

**Pregunta 16: ¿Cuál de los siguientes atributos HTML es utilizado para asignar un estilo de color, fuente o tamaño a un elemento?**

- A) id
- B) class
- C) style
- D) title
- E) lang

Respuesta correcta: C) style

**Pregunta 17: ¿Cuál es el selector utilizado para aplicar estilos a elementos HTML basados en su atributo de clase?**

- A) Selector de Tipo
- B) Selector de Clase
- C) Selector de ID
- D) Selector Universal (\*)
- E) Selector de Atributo

Respuesta correcta: B) Selector de Clase

**Pregunta 18: ¿Cuál es el propósito de los gestores de contenido (CMS) en el desarrollo web?**

- a) Simplificar y acelerar la construcción de sitios web.
- b) Programar aplicaciones web complejas.
- c) Controlar la gestión de servidores web.
- d) Optimizar la velocidad de carga de la página.
- e) Realizar análisis de datos en tiempo real.

Respuesta Correcta: a) Simplificar y acelerar la construcción de sitios web.

**Pregunta 19: ¿Cuál es el objetivo de la herramienta “Inspector de Elementos” en el desarrollo web?**

- a) Editar videos en tiempo real.
- b) Comprobar el clima actual de la ubicación de un sitio web.
- c) Comprender y modificar la estructura y el estilo de una página web en vivo.
- d) Dibujar gráficos y animaciones interactivas.
- e) Crear contenido de redes sociales.

Respuesta Correcta: c) Comprender y modificar la estructura y el estilo de una página web en vivo.

**Pregunta 20: ¿Qué unidad de medida se utiliza comúnmente en diseño web para establecer dimensiones flexibles?**

- a) Pulgadas (in).
- b) Milímetros (mm).
- c) Píxeles (px).
- d) Kilogramos (kg).
- e) Centímetros (cm).

Respuesta Correcta: c) Píxeles (px).

## Anexo 2. Solucionario Unidad 2

**Pregunta 1: ¿Cuál de los siguientes lenguajes se utiliza principalmente en la programación del lado del cliente?**

- a) Java
- b) PHP
- c) HTML
- d) Python

Respuesta correcta: c) HTML

**Pregunta 2: ¿Qué capa de la arquitectura de aplicaciones multinivel se encarga de la presentación de datos al usuario?**

- a) Nivel de Acceso a Datos
- b) Nivel de Lógica de Negocio
- c) Nivel de Presentación
- d) Nivel de Comunicación

Respuesta correcta: c) Nivel de Presentación

**Pregunta 3: ¿Cuál de las siguientes afirmaciones sobre PHP es correcta?**

- a) PHP es un lenguaje de programación del lado del cliente.
- b) PHP se utiliza principalmente para la creación de páginas web estáticas.
- c) PHP es un lenguaje interpretado que se ejecuta en el servidor web.
- d) PHP no se integra con HTML en absoluto.

Respuesta correcta: c) PHP es un lenguaje interpretado que se ejecuta en el servidor web.

**Pregunta 4: ¿Qué función cumplen las sesiones y las cookies en aplicaciones web desarrolladas con PHP?**

- a) Mantener la base de datos
- b) Gestionar el almacenamiento de archivos
- c) Mantener el estado del usuario a lo largo de múltiples solicitudes
- d) Validar formularios

Respuesta correcta: c) Mantener el estado del usuario a lo largo de múltiples solicitudes

**Pregunta 5: ¿Qué ventaja ofrece la integración de variables PHP en documentos HTML?**

- a) Permite a los navegadores ejecutar código PHP directamente.
- b) Hace que la página sea estática y rápida.
- c) Permite mezclar contenido estático con contenido dinámico.
- d) Añade complejidad innecesaria a las páginas web.

Respuesta correcta: c) Permite mezclar contenido estático con contenido dinámico.

**Pregunta 6: ¿Cuál de las siguientes tecnologías es esencial para la programación del lado del cliente en una aplicación web?**

- a) CSS
- b) Python
- c) SQL
- d) Ruby

Respuesta correcta: a) CSS

**Pregunta 7: ¿En cuál de las siguientes capas se encuentra la lógica central de una aplicación en la arquitectura de tres capas?**

- a) Nivel de Presentación
- b) Nivel de Lógica de Negocio
- c) Nivel de Acceso a Datos
- d) Nivel de Comunicación

Respuesta correcta: b) Nivel de Lógica de Negocio

**Pregunta 8: ¿Cuál es una característica clave de PHP?**

- a) Es un lenguaje de programación de código cerrado.
- b) No se puede utilizar en el desarrollo de aplicaciones web.
- c) Se ejecuta en el navegador del usuario.
- d) Permite la incrustación en documentos HTML.

Respuesta correcta: d) Permite la incrustación en documentos HTML.

**Pregunta 9: ¿Para qué se utilizan comúnmente las cookies en aplicaciones web?**

- a) Para decorar las páginas web.
- b) Para gestionar el almacenamiento de archivos.
- c) Para mantener el estado del usuario a lo largo de múltiples solicitudes.
- d) Para ejecutar código JavaScript.

Respuesta correcta: c) Para mantener el estado del usuario a lo largo de múltiples solicitudes.

**Pregunta 10: ¿Cuál es el propósito principal de integrar PHP con HTML en una página web?**

- a) Para hacer que la página sea más lenta.
- b) Para permitir a los navegadores ejecutar código PHP directamente.
- c) Para mantener el estado de un usuario a lo largo de múltiples solicitudes.
- d) Para crear páginas web estáticas sin contenido dinámico.

Respuesta correcta: c) Para mantener el estado de un usuario a lo largo de múltiples solicitudes.

**Pregunta 11: ¿Cuál de las siguientes no es una operación principal en la interacción con bases de datos?**

- a) Inserción (INSERT)
- b) Actualización (UPDATE)
- c) Eliminación (DELETE)
- d) Compartir (SHARE)
- e) Consulta (SELECT)

Respuesta correcta: d) Compartir (SHARE)

**Pregunta 12: ¿Qué lenguaje se utiliza comúnmente para interactuar con bases de datos relacionales?**

- a) JSON
- b) Python
- c) SQL
- d) PHP
- e) HTML

Respuesta correcta: c) SQL

**Pregunta 13: ¿Cuál de las siguientes sentencias se utiliza para agregar nuevos registros a una tabla de la base de datos?**

- a) INSERT
- b) UPDATE
- c) DELETE
- d) SELECT
- e) JOIN

Respuesta correcta: a) INSERT

**Pregunta 14: ¿Cuál es la sentencia que se utiliza para eliminar registros de una tabla de la base de datos?**

- a) SELECT
- b) DELETE
- c) INSERT
- d) DROP
- e) MODIFY

Respuesta correcta: b) DELETE

**Pregunta 15: ¿Qué paso en el desarrollo de una aplicación web involucra la definición de la estructura de la base de datos y las características de la aplicación?**

- a) Elección de la Tecnología
- b) Diseño de la Base de Datos
- c) Pruebas y Depuración
- d) Conexión a la Base de Datos
- e) Despliegue

Respuesta correcta: b) Diseño de la Base de Datos

**Pregunta 16: ¿Qué tipo de base de datos es MongoDB?**

- a) Relacional
- b) NoSQL
- c) SQLite
- d) MySQL

e) PostgreSQL

Respuesta correcta: b) NoSQL

**Pregunta 17: ¿Cuál es el propósito de las consultas preparadas (prepared statements) al interactuar con bases de datos?**

- a) Acelerar la conexión a la base de datos.
- b) Realizar transacciones de bases de datos.
- c) Prevenir ataques de inyección SQL.
- d) Crear copias de seguridad de la base de datos.
- e) Desarrollar aplicaciones en el servidor.

Respuesta correcta: c) Prevenir ataques de inyección SQL.

**Pregunta 18: ¿Qué operación se utiliza para modificar datos existentes en la base de datos?**

- a) INSERT
- b) DELETE
- c) UPDATE
- d) SELECT
- e) CREATE

Respuesta correcta: c) UPDATE

**Pregunta 19: ¿Qué paso implica escribir el código de la aplicación web y la interacción con la base de datos?**

- a) Pruebas y Depuración
- b) Definición de Requisitos
- c) Desarrollo de la Aplicación Web
- d) Mantenimiento y Actualizaciones
- e) Elección de la Tecnología

Respuesta correcta: c) Desarrollo de la Aplicación Web

**Pregunta 20: ¿Qué es una consideración importante en la interacción con bases de datos relacionada con el aumento de datos y usuarios?**

- a) Seguridad
- b) Escalabilidad

- c) Respaldo de Datos
- d) Documentación
- e) Cumplimiento Legal

Respuesta correcta: b) Escalabilidad

### **Anexo 3. Solucionario Unidad 3**

**Pregunta 1: ¿Qué son los frameworks de desarrollo en el contexto de la programación web?**

- a) Herramientas para el diseño gráfico.
- b) Estructuras predefinidas para acelerar el desarrollo de aplicaciones web.
- c) Programas de análisis de datos.
- d) Sistemas operativos para servidores web.
- e) Bases de datos en la nube.

Respuesta correcta: b) Estructuras predefinidas para acelerar el desarrollo de aplicaciones web.

**Pregunta 2: ¿Cuál de los siguientes no es un framework de desarrollo web?**

- a) Ruby on Rails
- b) Django
- c) Spring Boot
- d) MySQL
- e) Laravel

Respuesta correcta: d) MySQL

**Pregunta 3: ¿Cuál de los siguientes frameworks utiliza el lenguaje Ruby?**

- a) Django
- b) Laravel
- c) Spring Boot

d) Ruby on Rails

e) Express.js

Respuesta correcta: d) Ruby on Rails

**Pregunta 4: ¿Cuál de los siguientes frameworks se enfoca en el desarrollo web con PHP?**

a) Django

b) Laravel

c) Spring Boot

d) Ruby on Rails

e) Express.js

Respuesta correcta: b) Laravel

**Pregunta 5: ¿Cuál de los siguientes frameworks es mantenido por Google y se utiliza para el desarrollo web front-end?**

a) React

b) Flask

c) Vue.js

d) Express.js

e) Spring Boot

Respuesta correcta: a) React

**Pregunta 6: ¿Cuál de los siguientes frameworks se utiliza comúnmente para crear aplicaciones web y APIs con JavaScript (Node.js)?**

a) Ruby on Rails

b) Laravel

c) Spring Boot

d) Ruby on Rails

e) Express.js

Respuesta correcta: e) Express.js

**Pregunta 7: ¿Cuál de los siguientes frameworks es un framework de desarrollo web minimalista de Python?**

- a) Django
- b) Flask
- c) Spring Boot
- d) Ruby on Rails
- e) Laravel

Respuesta correcta: b) Flask

**Pregunta 8: ¿Qué consideración es importante al elegir un framework de desarrollo?**

- a) Color de la interfaz de usuario.
- b) La popularidad del framework.
- c) Tamaño del monitor.
- d) Tipo de CPU.
- e) Tipo de aplicación.

Respuesta correcta: e) Tipo de aplicación.

**Pregunta 9: ¿Qué desventaja común de los frameworks de desarrollo se refiere a que pueden limitar la flexibilidad y la creatividad?**

- a) Sobrecarga de abstracción.
- b) Curva de aprendizaje empinada.
- c) Tamaño del código.
- d) Actualizaciones y cambios.
- e) Dependencia del Framework.

Respuesta correcta: a) Sobrecarga de abstracción.

**Pregunta 10: ¿Cuál de las siguientes no es una mejor práctica de seguridad web?**

- a) Validación de datos de entrada.
- b) Autenticación y autorización.
- c) Sesiones seguras.
- d) Escapado de datos de salida.

e) Publicación de información sensible en mensajes de error.

Respuesta correcta: e) Publicación de información sensible en mensajes de error.

**Pregunta 11: ¿Qué herramienta se utiliza para realizar auditorías de seguridad en aplicaciones y redes y proporciona recomendaciones para solucionar problemas?**

- a) Burp Suite.
- b) Nikto.
- c) Nessus.
- d) Zed Attack Proxy (ZAP).
- e) Flask.

Respuesta correcta: c) Nessus.

**Pregunta 12: ¿Cuál de las siguientes herramientas es un firewall de aplicaciones web (WAF) de código abierto que protege contra ataques comunes?**

- a) Acunetix.
- b) ModSecurity.
- c) W3af.
- d) Burp Suite.
- e) Vue.js.

Respuesta correcta: b) ModSecurity.

**Pregunta 13: ¿Cuál de las siguientes herramientas es una suite de pruebas de seguridad de aplicaciones web desarrollada por OWASP?**

- a) Burp Suite.
- b) Nikto.
- c) Nessus.
- d) Zed Attack Proxy (ZAP).
- e) SecurityHeaders.io.

Respuesta correcta: d) Zed Attack Proxy (ZAP).

**Pregunta 14: ¿Qué herramienta es ampliamente utilizada para la evaluación de aplicaciones web, incluyendo escaneo de seguridad y prueba de penetración?**

- a) Nessus.
- b) Acunetix.
- c) Burp Suite.
- d) W3af.
- e) ModSecurity.

Respuesta correcta: c) Burp Suite.

**Pregunta 15: ¿Cuál de las siguientes herramientas se utiliza para identificar y solucionar vulnerabilidades en aplicaciones web, con pruebas de seguridad automatizadas?**

- a) Qualys Web Application Scanning (WAS).
- b) Zed Attack Proxy (ZAP).
- c) SecurityHeaders.io.
- d) Acunetix.
- e) Nikto.

Respuesta correcta: a) Qualys Web Application Scanning (WAS).

**Pregunta 16: ¿Cuál de las siguientes herramientas se utiliza para realizar escaneos de seguridad en servidores web y detectar vulnerabilidades?**

- a) Nikto.
- b) Zed Attack Proxy (ZAP).
- c) Burp Suite.
- d) ModSecurity.
- e) Qualys Web Application Scanning (WAS).

Respuesta correcta: a) Nikto.

**Pregunta 17: ¿Cuál de las siguientes organizaciones proporciona recursos y herramientas de código abierto para mejorar la seguridad de las aplicaciones web?**

- a) Nessus.
- b) Acunetix.

- c) OWASP.
- d) Qualys.
- e) W3af.

Respuesta correcta: c) OWASP.

**Pregunta 18: ¿Qué tipo de amenaza web se refiere a un ataque en el que se inserta código SQL malicioso en las entradas de una aplicación web?**

- a) Inyección SQL.
- b) XSS.
- c) CSRF.
- d) Inyección de comandos.
- e) Autenticación débil.

Respuesta correcta: a) Inyección SQL.

**Pregunta 19: ¿Qué desventaja común de los frameworks de desarrollo se refiere al hecho de que pueden generar un código más grande en comparación con una aplicación personalizada?**

- a) Sobrecarga de abstracción.
- b) Curva de aprendizaje empinada.
- c) Tamaño del código.
- d) Actualizaciones y cambios.
- e) Dependencia del Framework.

Respuesta correcta: c) Tamaño del código.

**Pregunta 20: ¿Cuál de las siguientes no es una ventaja de los frameworks de desarrollo?**

- a) Mantenimiento simplificado.
- b) Escalabilidad.
- c) Flexibilidad ilimitada.
- d) Reutilización de código.
- e) Estándares de desarrollo.

Respuesta correcta: c) Flexibilidad ilimitada

## Anexo 4. Solucionario Unidad 4

**Pregunta 1: ¿Qué tipos de vulnerabilidades pueden detectar las herramientas de identificación de vulnerabilidades?**

- a) Vulnerabilidades de diseño gráfico.
- b) Vulnerabilidades de base de datos.
- c) Vulnerabilidades de escalabilidad.
- d) Vulnerabilidades de inyección SQL.
- e) Vulnerabilidades de hardware.

Respuesta correcta: d) Vulnerabilidades de inyección SQL.

**Pregunta 2: ¿Qué enfoque de análisis utilizan algunas herramientas de identificación de vulnerabilidades?**

- a) Análisis estático del código fuente.
- b) Análisis dinámico de la aplicación en tiempo de ejecución.
- c) Análisis de diseño gráfico.
- d) Análisis de redes sociales.
- e) Análisis de configuración de hardware.

Respuesta correcta: a) Análisis estático del código fuente.

**Pregunta 3: ¿Qué etapas del ciclo de desarrollo de software pueden beneficiarse de la integración de herramientas de identificación de vulnerabilidades?**

- a) Fase de desarrollo.
- b) Fase de prueba.
- c) Fase de producción.
- d) Todas las anteriores.
- e) Ninguna de las anteriores.

Respuesta correcta: d) Todas las anteriores.

**Pregunta 4: ¿Qué información se incluye en los informes generados por las herramientas de identificación de vulnerabilidades?**

- a) Recetas de cocina.
- b) Gravedad de las vulnerabilidades encontradas.

- c) Pronósticos del tiempo.
- d) Noticias de entretenimiento.
- e) Opiniones de películas.

Respuesta correcta: b) Gravedad de las vulnerabilidades encontradas.

**Pregunta 5: ¿Qué función pueden desempeñar algunas herramientas de identificación de vulnerabilidades en relación con la base de datos de una aplicación?**

- a) Acceder y gestionar la base de datos.
- b) Evaluar la velocidad de la base de datos.
- c) Redactar informes financieros.
- d) Realizar tareas de diseño gráfico.
- e) Preparar café.

Respuesta correcta: a) Acceder y gestionar la base de datos.

**Pregunta 6: ¿Qué herramientas se utilizan para evaluar y mejorar la conformidad de un sitio web con las pautas de accesibilidad, como WCAG?**

- a) Herramientas de diseño gráfico.
- b) Herramientas de análisis financiero.
- c) Herramientas de análisis de velocidad de carga.
- d) Herramientas de identificación de vulnerabilidades.
- e) Herramientas de verificación de la accesibilidad.

Respuesta correcta: e) Herramientas de verificación de la accesibilidad.

**Pregunta 7: ¿Cuál de las siguientes herramientas de verificación de la accesibilidad es una extensión de navegador que proporciona una evaluación detallada de la accesibilidad de una página web?**

- a) Wave (Web Accessibility Evaluation Tool).
- b) Siteimprove.
- c) Tenon.io.
- d) SortSite.
- e) AInspector Sidebar.

Respuesta correcta: a) Wave (Web Accessibility Evaluation Tool).

**Pregunta 8: ¿Qué fase del desarrollo de software implica la configuración del servidor web?**

- a) Fase de planificación.
- b) Fase de análisis.
- c) Fase de implementación.
- d) Fase de diseño gráfico.
- e) Fase de prueba.

Respuesta correcta: c) Fase de implementación.

**Pregunta 9: ¿Cuál de las siguientes no es una etapa común en el ciclo de desarrollo de proyectos?**

- a) Planificación del proyecto.
- b) Implementación de seguridad.
- c) Evaluación posterior al proyecto.
- d) Seguimiento del progreso del proyecto.
- e) Evaluación de riesgos.

Respuesta correcta: b) Implementación de seguridad.

**Pregunta 10: ¿Qué debe hacerse antes de transferir los archivos de un sitio web desde un entorno local al servidor web?**

- a) Realizar pruebas exhaustivas.
- b) Realizar una evaluación posterior al proyecto.
- c) Regresar al análisis de riesgos.
- d) Configurar el servidor web.
- e) Hacer una copa de café.

Respuesta correcta: a) Realizar pruebas exhaustivas.

**Pregunta 11: ¿Cuál es el propósito de la comunicación efectiva en la gestión de proyectos?**

- a) Mejorar la velocidad de carga del sitio web.
- b) Evaluar la conformidad con las pautas de accesibilidad.
- c) Comunicar los resultados de la evaluación de vulnerabilidades.
- d) Mantener a todas las partes interesadas informadas sobre el progreso y los problemas del proyecto.

e) Realizar pruebas de calidad.

Respuesta correcta: d) Mantener a todas las partes interesadas informadas sobre el progreso y los problemas del proyecto.

**Pregunta 12: ¿Qué se utiliza para rastrear los cambios en el código fuente de un proyecto?**

- a) Herramientas de análisis financiero.
- b) Repositorios de código.
- c) Análisis de rendimiento del servidor web.
- d) Herramientas de análisis de velocidad de carga.
- e) Medidores de temperatura del hardware.

Respuesta correcta: b) Repositorios de código.

**Pregunta 13: ¿Qué fase del desarrollo implica la configuración de servidores, la instalación de software y la verificación de que el sitio web funcione correctamente?**

- a) Fase de implementación de seguridad.
- b) Fase de evaluación posterior al proyecto.
- c) Fase de pruebas y depuración.
- d) Fase de entrega y despliegue.
- e) Fase de control de calidad.

Respuesta correcta: d) Fase de entrega y despliegue.

**Pregunta 14: ¿Por qué es importante la evaluación posterior al proyecto en la gestión de proyectos?**

- a) Para mantener a todas las partes interesadas informadas sobre el progreso.
- b) Para evaluar la conformidad con las pautas de accesibilidad.
- c) Para identificar lecciones aprendidas y áreas de mejora.
- d) Para configurar servidores.
- e) Para realizar pruebas de rendimiento.

Respuesta correcta: c) Para identificar lecciones aprendidas y áreas de mejora.

**Pregunta 15: ¿Qué es esencial para el seguimiento y la evaluación del proyecto?**

- a) Herramientas de análisis financiero.
- b) Comunicación efectiva.
- c) Análisis de velocidad de carga del sitio web.
- d) Herramientas de diseño gráfico.
- e) Programar reuniones de seguimiento.

Respuesta correcta: b) Comunicación efectiva.

**Pregunta 16: ¿Cuál es una función de las reuniones de seguimiento en la gestión de proyectos?**

- a) Analizar el contraste de colores en una página web.
- b) Identificar y solucionar problemas de vulnerabilidad.
- c) Evaluar el rendimiento de la base de datos.
- d) Evaluar el progreso del proyecto y abordar problemas.
- e) Realizar análisis de riesgos.

Respuesta correcta: d) Evaluar el progreso del proyecto y abordar problemas.

**Pregunta 17: ¿Qué hacen las herramientas de identificación de vulnerabilidades?**

- a) Evaluación de velocidad de carga.
- b) Evaluación de la accesibilidad.
- c) Pruebas de seguridad en línea.
- d) Evaluación de riesgos.
- e) Evaluación de vulnerabilidades de seguridad.

Respuesta correcta: e) Evaluación de vulnerabilidades de seguridad.

**Pregunta 18: ¿Cuál es una responsabilidad importante en la fase de publicación de sitios web?**

- a) Planificación del proyecto.
- b) Realizar pruebas exhaustivas.
- c) Configuración del servidor web.
- d) Registro de un dominio.

e) Promoción y SEO.

Respuesta correcta: e) Promoción y SEO.

**Pregunta 19: ¿Cuál de las siguientes tareas es parte de la planificación de un proyecto?**

- a) Realizar una evaluación posterior al proyecto.
- b) Configuración de servidores.
- c) Planificación de la fase de entrega y despliegue.
- d) Definición de objetivos y alcance del proyecto.
- e) Implementación de seguridad.

Respuesta correcta: d) Definición de objetivos y alcance del proyecto.

**Pregunta 20: ¿Cuál es una función esencial de las herramientas de verificación de accesibilidad de sitios web?**

- a) Evaluación de velocidad de carga.
- b) Identificación de vulnerabilidades de seguridad.
- c) Verificar el contraste de colores en una página web.
- d) Evaluar la conformidad con las pautas de accesibilidad.
- e) Realizar análisis de riesgos.

Respuesta correcta: d) Evaluar la conformidad con las pautas de accesibilidad.



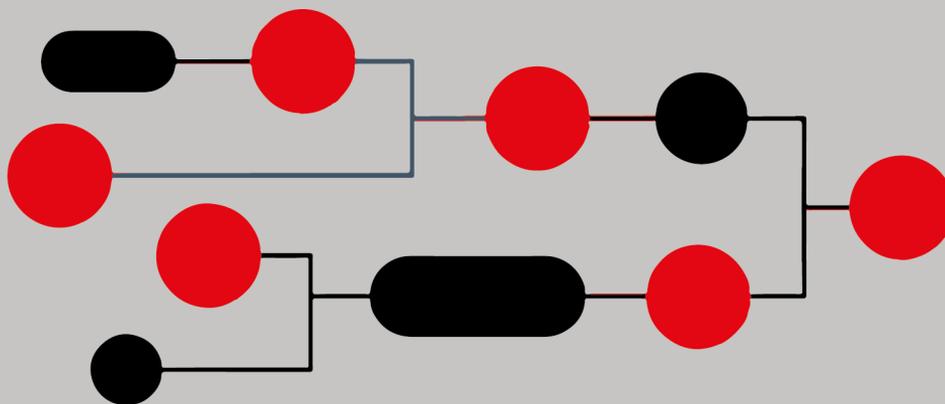


Religación  
**Press**  
Ideas desde el Sur Global



Religación  
**Press**  
Ideas desde el Sur Global

**int** Instituto  
Nelson  
Torres



ISBN: 978-9942-642-36-3



9 | 7 8 9 9 4 2 | 6 4 2 3 6 3 |