



UTPL

La Universidad Católica de Loja

Vicerrectorado de Modalidad Abierta y a Distancia

Desarrollo Web

Guía didáctica





Facultad Ingenierías y Arquitectura

Desarrollo Web

Guía didáctica

Carrera	PAO Nivel
Tecnologías de la Información	V

Autores:

Ramiro Leonardo Ramírez Coronel

Reestructurada por:

Diana Alexandra Torres Guarnizo



Desarrollo Web

Guía didáctica

Ramiro Leonardo Ramírez Coronel

Reestructurada por:

Diana Alexandra Torres Guarnizo

Diagramación y diseño digital

Ediloja Cía. Ltda.

Marcelino Champagnat s/n y París

edilocialtda@ediloja.com.ec

www.ediloja.com.ec

ISBN digital -978-9942-25-716-1

Año de edición: abril, 2020

Edición: primera edición reestructurada en agosto 2025 (con un cambio del 50%)

Loja-Ecuador



Los contenidos de este trabajo están sujetos a una licencia internacional Creative Commons **Reconocimiento-NoComercial-CompartirIgual** 4.0 (CC BY-NC-SA 4.0). Usted es libre de **Compartir** — copiar y redistribuir el material en cualquier medio o formato. Adaptar — remezclar, transformar y construir a partir del material citando la fuente, bajo los siguientes términos: Reconocimiento- debe dar crédito de manera adecuada, brindar un enlace a la licencia, e indicar si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que usted o su uso tienen el apoyo de la licenciante. No Comercial-no puede hacer uso del material con propósitos comerciales. Compartir igual-Si remezcla, transforma o crea a partir del material, debe distribuir su contribución bajo la misma licencia del original. No puede aplicar términos legales ni medidas tecnológicas que restrinjan legalmente a otras a hacer cualquier uso permitido por la licencia. <https://creativecommons.org/licenses/by-nc-sa/4.0/>

Índice

1. Datos de información	9
1.1 Presentación de la asignatura.....	9
1.2 Competencias genéricas de la UTPL.....	9
1.3 Competencias específicas de la carrera	9
1.4 Problemática que aborda la asignatura	9
2. Metodología de aprendizaje	10
3. Orientaciones didácticas por resultados de aprendizaje.....	11
Primer bimestre	11
Resultado de aprendizaje 1 a 3:.....	11
Contenidos, recursos y actividades de aprendizaje recomendadas.....	11
Semana 1	12
Unidad 1. Interfaces web	12
1.1 Integración e implementación de sitios web seguros	13
1.2 Integración de base de datos	16
1.3 Problemas de accesibilidad	17
Actividades de aprendizaje recomendadas	24
Autoevaluación 1	25
Resultado de aprendizaje 1, 4 y 5:	28
Contenidos, recursos y actividades de aprendizaje recomendadas.....	28
Semana 2	28
Unidad 2. Programación en PHP	28
2.1 Configuración del entorno de desarrollo	31
2.2 Primeros pasos con PHP.....	33
2.3 Variables	36
2.4 Constantes	37
2.5 Operadores	38
2.6 Estructuras de control.....	39
Contenidos, recursos y actividades de aprendizaje recomendadas.....	40



Semana 3 40

 Unidad 2. Programación en PHP 40

 2.7. Funciones 40

 2.8 Sesiones 42

 Actividades de aprendizaje recomendadas 43

 Autoevaluación 2..... 43

Resultado de aprendizaje 6 y 7:..... 46

Contenidos, recursos y actividades de aprendizaje recomendadas..... 46

Semana 4 46

 Unidad 3. PHP con MySQL 46

 3.1 Gestión de datos 47

 3.2 Creación de base de datos 47

 3.3 Listar las bases de datos de nuestro servidor 48

 3.4 Creación de tablas en la base de datos..... 49

 3.5 Manipulación de datos con SQL 50

 3.6 Gestión de Datos con PhpMyadmin 52

Contenidos, recursos y actividades de aprendizaje recomendadas..... 53

Semana 5 53

 Unidad 3. PHP con MySQL 53

 3.7 Conectar una base de datos con PHP 53

 3.8 Gestión de datos con PHP y MYSQL CRUD..... 55

 3.9 Formulario de registro 56

Contenidos, recursos y actividades de aprendizaje recomendadas..... 58

Semana 6 58

 Unidad 3. PHP con MySQL 58

 3.10 Borrar y actualizar registros 58

 Actividad de aprendizaje recomendada 64

 Autoevaluación 3..... 64

Resultado de aprendizaje 1 a 7:..... 67



Contenidos, recursos y actividades de aprendizaje recomendadas..... 67

Semana 7 67

 Actividades finales del bimestre 67

Contenidos, recursos y actividades de aprendizaje recomendadas..... 68

Semana 8 68

 Actividades finales del bimestre 68

Segundo bimestre..... 69

Resultado de aprendizaje 6, 8, 9 y 10: 69

Contenidos, recursos y actividades de aprendizaje recomendadas..... 69

Semana 9 70

 Unidad 4. Programación orientada a objetos 70

 4.1 Métodos y funciones para manejar eventos 70

Contenidos, recursos y actividades de aprendizaje recomendadas..... 75

Semana 10 75

 Unidad 4. Programación orientada a objetos 75

 4.2 Propagación de eventos 75

 4.3 Manejo de excepciones 75

 4.4 ¿Cuándo lanzar una excepción y cuándo un error? 76

 4.5 Métodos de una excepción 77

 Actividad de aprendizaje recomendada 77

 Autoevaluación 4..... 78

Resultado de aprendizaje 5, 7, 9 y 10: 81

Contenidos, recursos y actividades de aprendizaje recomendadas..... 81

Semana 11 81

 Unidad 5. Seguridad y vulnerabilidades 81

 5.2 Tipos de ataques XSS..... 83

Resultado de aprendizaje 3, 5, 9 y 11: 85

Contenidos, recursos y actividades de aprendizaje recomendadas..... 85

Semana 12..... 86



Unidad 5. Seguridad y vulnerabilidades	86
5.3 Prevención de ataques	86
5.4 Validación de datos.....	86
Actividad de aprendizaje recomendada	90
Autoevaluación 5.....	90
Resultado de aprendizaje 8 a 11:.....	93
Contenidos, recursos y actividades de aprendizaje recomendadas.....	93
Semana 13.....	93
Unidad 6. Seguridad web	93
6.1 Seguridad del servidor	94
6.2 Control del acceso al servidor con el archivo .htaccess	96
Actividad de aprendizaje recomendada	97
Resultado de aprendizaje 3, 5, 9 y 11:	99
Contenidos, recursos y actividades de aprendizaje recomendadas.....	99
Semana 14.....	99
Unidad 6. Seguridad web	99
6.3 Configuración del servidor ante ataques de servicio.....	99
6.4 ServeRoot	102
6.5 ServerName	102
6.6 Puesta en marcha las WebApps	103
Actividades de aprendizaje recomendadas	106
Autoevaluación 6.....	107
Resultado de aprendizaje 3, 5, 6, 7, 8, 9, 10 y 11:	109
Contenidos, recursos y actividades de aprendizaje recomendadas.....	109
Semana 15.....	109
Actividades finales del bimestre	109
Resultado de aprendizaje 3, 5, 6, 7, 8, 9, 10 y 11:	111
Contenidos, recursos y actividades de aprendizaje recomendadas.....	111
Semana 16.....	111



Actividades finales del bimestre	111
4. Solucionario	113
5. Glosario.....	120
6. Referencias bibliográficas	122





1. Datos de información

1.1 Presentación de la asignatura



1.2 Competencias genéricas de la UTPL

Orientación a la investigación e innovación.

1.3 Competencias específicas de la carrera

Diseñar aplicaciones de *software* que permitan mediante técnicas avanzadas de modelado dar solución a los requerimientos del cliente utilizando estándares de la industria.

1.4 Problemática que aborda la asignatura

Se identifica como tensión la escasez de proyectos para aprovechar la actividad financiera y bancaria que se caracterizan por sus altos niveles de aplicación de tecnología, en esta línea la carrera propone una formación específica en el ámbito de proyectos, arquitectura empresarial y formación avanzada en el uso de tecnologías *web*, *cloud*, móviles y desarrollo de sistemas.





2. Metodología de aprendizaje

La metodología de aprendizaje que se va a usar en el desarrollo de las actividades de la asignatura de Desarrollo web es una metodología de aprendizaje basada en TIC. Esta permite, a través de la utilización de medios tecnológicos, mejorar el proceso de aprendizaje utilizando el computador, *software* disponible en Internet como herramientas mediadoras del conocimiento.





3. Orientaciones didácticas por resultados de aprendizaje



Primer bimestre

Resultado de aprendizaje 1 a 3:

- Describe los problemas implicados en el desarrollo de una interfaz web.
- Resume la necesidad y las cuestiones implicadas en la ejecución y la integración de una página web.
- Explica la importancia de la interconexión entre sitios web para bases de datos subyacentes.

Para alcanzar los resultados de aprendizaje planteados, es esencial comprender y abordar los problemas relacionados con el desarrollo y la gestión de interfaces web. Esto implica identificar los desafíos en el diseño de interfaces, resumir la necesidad y las cuestiones clave en la ejecución e integración de páginas web, y explicar la importancia de la interconexión entre sitios web y bases de datos subyacentes. Además, se abordará la integración e implementación de sitios web seguros, la integración de bases de datos y los problemas de accesibilidad, ofreciendo una visión completa para resolver estos aspectos fundamentales.

Contenidos, recursos y actividades de aprendizaje recomendadas

Recuerde revisar de manera paralela los contenidos con las actividades de aprendizaje recomendadas y actividades de aprendizaje evaluadas.





Unidad 1. Interfaces web

Estimado alumno, iniciamos el estudio de la asignatura, mencionando temas fundamentales que se requiere conocer dentro del maravilloso mundo del desarrollo *web*. Comenzamos revisando algunos conceptos básicos como interfaces web, accesibilidad y usabilidad *web*; esto le ayudará a comprender lo importante de este tema.

La accesibilidad conlleva un trabajo extenso y detallado, ya que abarca una iteración múltiple de elementos como audio, video, imágenes y la estructura lógica de la aplicación. Representando una labor que va más allá de la presentación visual de la información.

En el proceso de diseño web, durante la fase de diseño de la aplicación, es fundamental considerar la usabilidad y la accesibilidad, ya que ambos son elementos clave para construir un sitio que cumpla con los principios mínimos de calidad.

- **Usabilidad:** hace referencia al tiempo, esfuerzo y nivel de comprensión que requiere un usuario para adaptarse eficazmente al uso de un sistema. Este concepto está relacionado con la cantidad de pasos necesarios para alcanzar un objetivo dentro de la aplicación, así como con los conocimientos previos que el usuario necesita para interactuar con la *WebApp* (aplicación web) de forma intuitiva y eficiente.
- **Accesibilidad:** una *WebApp* debe ser diseñada para estar al alcance de todos los usuarios, sin importar sus condiciones o el entorno desde el cual accedan. Esto implica considerar la diversidad de dispositivos, navegadores, idiomas, condiciones de conectividad, y posibles discapacidades. Estos factores pueden afectar la forma en que se presenta y funciona la aplicación, por lo que es esencial garantizar que la información sea percibida y utilizada correctamente por el mayor número de personas posible.





Tomando en cuenta los principios de usabilidad y accesibilidad, es fundamental desarrollar una interfaz web sólida que proporcione al usuario una experiencia cómoda, confiable y segura durante su interacción con la página.

Una *interfaz web* es el medio visual e interactivo a través del cual los usuarios se comunican y operan con una aplicación o sitio web desde un navegador, su propósito es permitir que el usuario pueda realizar tareas específicas de forma intuitiva, accesible y amigable, independientemente de su nivel de experiencia o del dispositivo que utilice para dicha tarea.

1.1 Integración e implementación de sitios web seguros

El uso de documentos HTML estáticos puede ser una solución adecuada cuando una aplicación web se limita a mostrar siempre la misma información o cuando es posible automatizar la actualización de dichos documentos. Sin embargo, la naturaleza dinámica de la web actual y las expectativas de los usuarios hacen necesaria la implementación de aplicaciones web seguras que generen contenido de forma dinámica con el objetivo de presentar la información según las necesidades específicas del momento.

Aunque técnicamente sería posible lograr esto con páginas HTML estáticas — si se cuenta con suficiente espacio en disco—, las aplicaciones web dinámicas ofrecen una ventaja significativa: capacidad de acceso a la base de datos que contienen información operativa actualizada, lo cual garantiza una experiencia más personalizada y oportuna para el usuario. Esta estrategia, de hecho, también puede contribuir a reducir la carga en la CPU de los servidores mediante mecanismos de caché o generación previa de contenido.

En este contexto, la seguridad se convierte en un aspecto esencial. Es común que los usuarios se pregunten si un sitio web es confiable para ingresar información personal o financiera. Por ello, es importante implementar protocolos y medidas de seguridad, como el uso de certificados digitales que



habiliten el protocolo HTTPS, políticas de control de acceso, cifrado de datos, y otras prácticas que garanticen la confidencialidad, integridad y disponibilidad de la información.

Control de acceso a la información

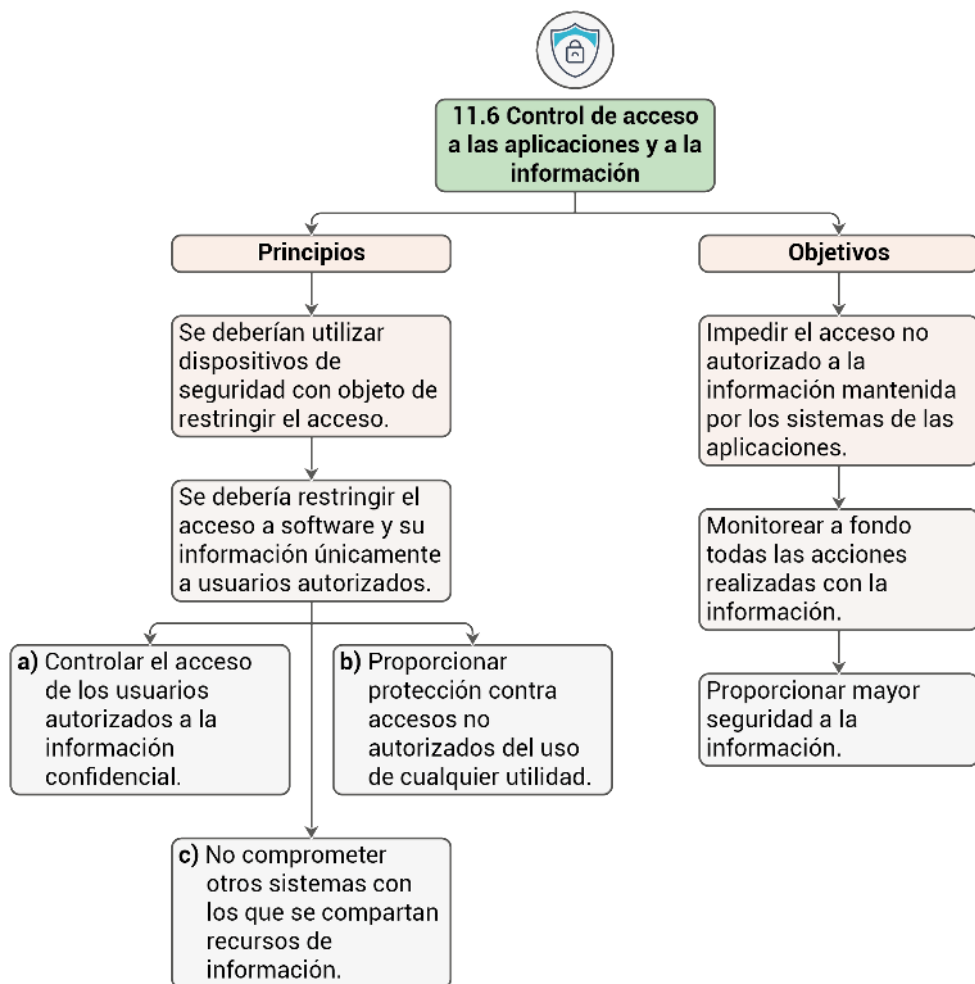
Las aplicaciones web tienen acceso abierto a un conjunto de información que explícitamente se hace pública. Sin embargo, en determinadas circunstancias, es interesante poder limitar el acceso a documentos reservados o útiles para un conjunto restringido de personas. Se pueden establecer dos tipos de restricciones:

1. **Limitación de acceso en función de direcciones IP o dominio:** solo los usuarios de un dominio u organización tendrán acceso a la información.
2. **Limitación de acceso por nombres de usuario y claves de acceso:** solo los usuarios que conozcan una clave de acceso válida pueden acceder a la información tal como se muestra en la figura 1.



Figura 1

Diagrama del control de acceso a la información, principios y objetivos



Nota. Ramírez, R., 2024.

Otro aspecto que está cobrando especial importancia es la seguridad de la información que se intercambia en la web.

La *Internet* exige disponer de sistemas de comunicación seguros, capaces de adaptarse a las necesidades de los nuevos servicios, como la compra electrónica o la banca electrónica. En estos servicios se manejan dos conceptos fundamentales: la **autenticación** y la **confidencialidad**.

El Control de acceso a la información se utiliza para limitar el acceso a determinados documentos de un servidor web, en función del origen y tipo de petición. La forma de hacerlo varía con el entorno en el que se publican las páginas (sistema operativo y servidor HTTP, principalmente); en general, todas las soluciones pasan por definir un fichero que contiene las diferentes limitaciones de acceso, en un formato característico del servidor HTTP. En algunos casos se utiliza un fichero global con las restricciones de acceso o bien un fichero por cada directorio al que se quiere limitar el acceso.

Cuando un cliente web accede a un fichero protegido, el servidor devuelve un **código de error** asociado a la falta de permisos para realizar la operación (código 401). Si desea conocer más sobre lo código de estado de respuesta HTTP, puede revisar la [documentación técnica oficial MDN Web Docs](#).



No olvide que debe acceder al Entorno Virtual de Aprendizaje para interactuar con el tutor y sus compañeros, por si le quedó alguna duda de los conceptos básicos sobre el tema de seguridad en una WebApp.

1.2 Integración de base de datos

La integración de datos es el proceso que permite combinar datos heterogéneos de varias fuentes en la forma y estructura de una única aplicación. Este proceso de integración de datos facilita que diferentes tipos de datos, tales como matrices de datos, documentos y tablas, sean fusionados por usuarios, organizaciones y aplicaciones para un uso personal, de procesos de negocio o de funciones.

La integración de datos soporta el procesamiento analítico de grandes conjuntos de datos, alineando, combinando y presentando cada conjunto de datos de departamentos organizacionales y fuentes de datos remotas y externas, para cumplir con los objetivos del integrador.



La integración de datos se implementa generalmente en una “*data warehouse*” mediante *software* especializado que aloja grandes repositorios de datos de recursos internos y externos. Los datos se extraen, se mezclan y se presentan de forma unificada. Por ejemplo, el conjunto completo de datos de un usuario puede incluir datos extraídos y combinados de *marketing*, ventas y operaciones, que se combinan para formar un informe completo (base de datos).

Un proyecto de integración de datos generalmente implica los siguientes pasos:

- **Acceso a los datos** desde todas las fuentes y localizaciones, tanto si se trata de locales, en la nube o de una combinación de ambos.
- **Integración de datos** de modo que los registros de una fuente de datos mapean registros en otra. Por ejemplo, incluso si un conjunto de datos utilizara “nombre, apellidos” y otro “nom, ape”, el conjunto integrado se asegurará de que en ambos casos los datos se almacenen en el lugar correcto. Se trata, pues, de un tipo de preparación de datos esencial para que las analíticas y otras aplicaciones sean capaces de utilizar los datos con éxito.
- **Entrega de datos integrados** al negocio justo en el momento en que la empresa los necesita, ya sea por lotes, casi en tiempo real o en tiempo real (base de datos).

1.3 Problemas de accesibilidad

Estimado alumno, iniciamos este apartado de la asignatura abordando temas fundamentales relacionados con la accesibilidad web, los cuales son esenciales para comprender su importancia y aplicación.

A lo largo de este bloque, se partirá de recordar el concepto de accesibilidad, los principios y buenas prácticas que permiten garantizar que los contenidos digitales sean accesibles para todos los usuarios, además de abordar el tema de cómo hacemos que un sitio sea accesible.



¿Qué es accesibilidad web?

Estimado alumno, para responder esta pregunta:

Revise el Capítulo 1 del texto "[Aplicación de Técnicas de Usabilidad y Accesibilidad en el Entorno del Cliente](#)", donde se presenta una definición clara y contextualizada del concepto de accesibilidad, así como su relevancia en el diseño y desarrollo de entornos digitales accesibles para todo tipo de usuarios.

Recuerden que el objetivo principal de la **accesibilidad web** es garantizar que las personas con diferentes capacidades, incluyendo aquellas con algún tipo de discapacidad, puedan utilizar los contenidos y servicios disponibles en la *web*.

Cuando hablamos de accesibilidad web, se hace referencia a un enfoque de diseño que permite a todos los usuarios independientemente de sus limitaciones físicas, sensoriales o cognitivas percibir, comprender, navegar e interactuar con los sitios web, así como también contribuir activamente con contenidos, como se muestra en la Figura 2. Este enfoque es esencial para construir una web inclusiva, equitativa y centrada en el usuario.



Figura 2

Accesibilidad web



Nota. Adaptado de *Accesibilidad Web - Definición, características y ejemplos* [Ilustración], por Delgado, H., 2022, [diseñowebakus](https://diseñowebakus.com), CC BY 4.0.

La accesibilidad web si bien trata de buscar un acceso equitativo a usuarios con distintas discapacidades, incluyendo limitaciones visuales, auditivas, físicas, cognitivas, neurológicas y del habla. Sus beneficios no se limitan únicamente a este grupo de personas. También favorece a organizaciones y a usuarios que se desempeñan en entornos laborales con limitaciones, por ejemplo: acceso limitado a internet o idioma, entre otros.

Uno de los principios fundamentales sobre los que se basa la accesibilidad web es la flexibilidad de adaptarse a distintas necesidades, situaciones y preferencias de los usuarios.



Espero que la lectura que realizó haya sido entendida y comprendida, para ello acuda a su cuaderno de trabajo y escriba cuál es el concepto de accesibilidad web y responda las siguientes preguntas:

- ¿A quiénes beneficia la accesibilidad web?
- ¿Ayudará a personas sin discapacidad y por qué?

Para comprender mejor el concepto de accesibilidad web, es fundamental reconocer la importancia de incorporar estas técnicas en el desarrollo de nuestras aplicaciones.

¿Por qué es importante la accesibilidad web?:

Para comprender la importancia de la accesibilidad web, revise el Capítulo 2, denominado “Ventajas y dificultades en la implantación de la accesibilidad web”, del texto [Aplicación de Técnicas de Usabilidad y Accesibilidad en el Entorno del Cliente](#), donde se analiza en profundidad el impacto positivo que tiene la accesibilidad en el desarrollo web, así como los principales desafíos de su implementación. En este capítulo se proporciona una visión crítica sobre los beneficios sociales, técnicos y legales de la accesibilidad, junto con los obstáculos más comunes en su adopción.

Actualmente, la web ha cobrado mucha importancia, especialmente en áreas como educación, empleo, gobierno, comunicación, economía, entretenimiento, etc., por ello es necesario alcanzar sitios accesibles para proporcionar un acceso equitativo e igualdad de oportunidades a las personas con discapacidad y público en general.

Muy sencillo de entender el concepto. ¿Verdad?

¿Qué opina usted sobre el siguiente enunciado o aseveración?:

¿La accesibilidad web es importante porque permite el acceso equilibrado e igual oportunidad a las personas con discapacidad?



Con base en las preguntas propuestas se puede concluir que es muy relevante e importante construir aplicaciones web accesibles, ya que no debe existir ninguna restricción de acceso a la información a ninguna persona. Para ello, es importante cumplir los principios de accesibilidad web.

¿Cómo hacer un sitio accesible?

Una de las funciones de la Iniciativa de Accesibilidad Web (WAI: *Web Accessibility Initiative*) es desarrollar pautas y técnicas que proporcionen soluciones accesibles para el *software web* y para los desarrolladores web. Las pautas de WAI son consideradas como estándares internacionales de accesibilidad web.

Para acceder a estos recursos y profundizar en el tema, en el sitio web oficial del [W3C](#) y su Iniciativa de Accesibilidad Web (WAI), podrá consultar una amplia variedad de materiales especializados sobre accesibilidad web. Este portal ofrece guías, recomendaciones, herramientas y ejemplos prácticos para implementar buenas prácticas de accesibilidad en el desarrollo de sitios y aplicaciones web, basados en los estándares internacionales promovidos por el W3C.

Hacer un sitio web accesible puede ser un proceso sencillo o complejo, depende de muchos factores como: tipo de contenido, tamaño y complejidad del sitio, así como de las herramientas de desarrollo y el entorno que se use.

Varios aspectos relacionados con la accesibilidad en un sitio web pueden implementarse de manera sencilla si se consideran desde las etapas iniciales del desarrollo o al comienzo de un rediseño. Planificar con antelación permite integrar buenas prácticas sin complicaciones técnicas ni costos adicionales.

Por el contrario, modificar sitios web que no fueron diseñados con criterios de accesibilidad desde las fases pueden implicar un esfuerzo considerable, especialmente cuando no se utilizaron etiquetas semánticas y estándares definidos por HTML5 (*HyperText Markup Language 5*). Esta dificultad se incrementa en sitios que contienen contenido complejo, como elementos



multimedia, los cuales requieren medidas específicas para garantizar su accesibilidad (por ejemplo, subtítulos, descripciones de audio o controles alternativos).

Las nuevas etiquetas que se crearon en el lenguaje de etiquetado HTML5 ayudan a reemplazar el abuso excesivo de etiquetas <div> que habitualmente se utilizan con el fin de delimitar o emular las partes de un documento como cabecera, pie de página, contenedores, etc. Con el desarrollo y la utilización de las etiquetas HTML5 se logran estructuras de vistas más homogeneizadas y que el manejo de meta data sea más adecuado.

Este avance técnico contribuye directamente a el objetivo de avanzar hacia lo que se conoce como la "web semántica", un enfoque que busca que la presentación y estructura de los documentos web sean más coherentes, comprensibles y significativos, tanto para los usuarios como para las máquinas.



Continuemos con el aprendizaje revisando el siguiente video, donde se explica a detalle los objetivos de la [accesibilidad web](#).

Este video le permitirá comprender de manera clara y práctica la importancia de diseñar sitios web inclusivos, así como los beneficios que aporta la accesibilidad tanto a personas con discapacidad como al público en general.

1.3.1 Iniciativa de accesibilidad web

A continuación, en la siguiente infografía se muestra una lista de consideraciones a tomar en cuenta al momento del desarrollo de una aplicación web, ya que son puntos muy importantes por aplicar para que los resultados accesibles sean los correctos.

[Estructura y presentación en el desarrollo web](#)



Con la lectura y el análisis que acabó de realizar, acuda a su cuaderno de trabajo y mencione dos factores que intervienen en la accesibilidad de un sitio web.

Evaluación de la accesibilidad

Se recomienda realizar la evaluación de la accesibilidad desde las etapas iniciales del desarrollo de un proyecto y de forma progresiva a lo largo de todo el proceso, esto nos permite identificar a tiempo los problemas de accesibilidad, momento en el cual resulta más sencillo y menos costoso corregirlos.

Incluso mediante técnicas simples, como modificar la configuración del navegador o utilizar atajos de teclado, es posible comprobar si una página web cumple con ciertos aspectos básicos de las pautas de accesibilidad.

Existen diversas herramientas en línea que facilitan la evaluación técnica de sitios web. Sin embargo, es importante destacar que ninguna herramienta automatizada puede determinar por sí sola el nivel completo de accesibilidad de un sitio web. Para una evaluación precisa y contextualizada, siempre se requiere el trabajo conjunto de las herramientas y la intervención del juicio humano, especialmente en aspectos relacionados con la comprensión, la navegación y la usabilidad.

A continuación, una lista de herramientas en línea que ayudará a medir la accesibilidad de una *WebApp*:

- [Tawdis](#).
- [Sidar](#).
- [Validator](#).
- [Usableyaccesible](#).





Actividades de aprendizaje recomendadas

Es hora de reforzar los conocimientos adquiridos resolviendo las siguientes actividades:

1. Con la lectura rápida que haya realizado sobre accesibilidad, está en condiciones de completar la siguiente frase:

La evaluación temprana de la accesibilidad permite ...

Para dar respuesta a esta pregunta puede utilizar Internet.

2. Continuemos con el aprendizaje mediante la revisión de la siguiente herramienta web oficial denominada "[Validadores y herramientas para consultorías de accesibilidad y usabilidad](#)" y, mencione las diferentes herramientas que encontró.

Nota: por favor, complete las actividades en un cuaderno o documento Word.

3. Como práctica, puede realizar la validación del [portal de la UTPL](#), con mínimo 2 herramientas, una vez obtenido los resultados, méncionelos en el foro de accesibilidad del curso y posteriormente emita un criterio técnico para mejorar dichos resultados.
4. Estimado alumno, le invito a medir sus conocimientos del estudio de esta unidad con la siguiente autoevaluación, esta le ayudará a valorar cuál es su nivel de aprendizaje durante el transcurso de esta primera unidad, en caso de no obtener resultados satisfactorios le sugiero revisar nuevamente los contenidos en el cual debe afianzar ciertos conocimientos.





Autoevaluación 1

Conteste correctamente las preguntas según sea el caso:

1. Seleccione de la siguiente lista cuál es el objetivo de la accesibilidad web.
 - a. Acceso a aplicaciones restringidas para cualquier usuario.
 - b. Acceso a aplicaciones para cualquier usuario normal o con discapacidad.
 - c. Acceso a recursos ricos como PDF, doc.
2. La accesibilidad ha cobrado importancia en áreas como, seleccione más de una respuesta:
 - a. Educación.
 - b. Industria automotriz.
 - c. Comunicación digital.
 - d. Empleo.
3. Las siglas WAI en español significan: seleccione la respuesta correcta.
 - a. Accesibilidad Web Incorporada.
 - b. Iniciativa de Accesibilidad Web.
 - c. Importancia de la Accesibilidad Web.
4. Localice las técnicas de accesibilidad de estructura y presentación.
 - a. Sitio marcado semánticamente.
 - b. Proporcionan resúmenes a las tablas.
 - c. Tamaño de imágenes proporcionales al contenido.
 - d. Contenido organizado lógicamente.





5. De la siguiente lista, marque cuáles son los validadores de accesibilidad.

- a. <https://www.tawdis.net/>.
- b. www.accessvalidatorwai.com.
- c. <http://www.sidar.org/hera/>.
- d. <http://walidator.net/uwem/index.es.php>.

6. La característica principal de las etiquetas HTML5 es:

- a. Etiquetado semántico para las partes importantes de los elementos que componen la página *web*.
- b. Etiquetas que ayudan a mejorar la programación de la página *web*.
- c. Etiquetas, mejorar la representación semántica y ayuda a la accesibilidad *web*.

7. La etiqueta NAV sirve para representar:

- a. La cabecera de la página *web*.
- b. Los módulos internos de la página *web*.
- c. Los menús de navegación.

8. La etiqueta *section* sirve para:

- a. Colocar la cabecera de la página *web*.
- b. Maquetar las secciones de contenido de la página *web*.
- c. Colocar los pies de página de las páginas *web*.

9. La etiqueta *footer* sirve para:

- a. Colocar la cabecera de la página *web*.
- b. Maquetar las secciones de contenido de la página *web*.
- c. Colocar los pies de página de las páginas *web*.

10. La etiqueta *main* sirve para:

- a. Colocar la cabecera de la página *web*.

- b. Colocar el contenido principal de la página web.
- c. Colocar los pies de página de la página web.

[Ir al solucionario](#)

¡Muy bien!



Le felicito por mantener la energía y el compromiso con los que inició esta materia. A partir de ahora, abordaremos temas aún más interesantes que le permitirán profundizar en el mundo del desarrollo web.

¡Hemos finalizado la primera unidad!

¡Enhorabuena por este primer logro! Siga adelante con la misma dedicación; lo mejor está por venir.



Resultado de aprendizaje 1, 4 y 5:

- Describe los problemas implicados en el desarrollo de una interfaz web.
- Explica por qué los problemas de accesibilidad son una consideración importante en el desarrollo de páginas Web.
- Construye aplicaciones web accesibles y seguras.

Para alcanzar los resultados de aprendizaje planteados, es esencial comprender y abordar tanto los problemas en el desarrollo de interfaces web como la importancia de la accesibilidad. Esto incluye describir los desafíos en el diseño de interfaces, explicar por qué los problemas de accesibilidad deben ser una prioridad en el desarrollo de páginas web, y construir aplicaciones web que sean accesibles y seguras. Se estudiará la programación en PHP, desde la configuración del entorno de desarrollo hasta el manejo de variables, constantes, estructuras de control, fusiones y sesiones, ofreciendo una base integral para desarrollar aplicaciones web efectivas.

Contenidos, recursos y actividades de aprendizaje recomendadas

Recuerde revisar de manera paralela los contenidos con las actividades de aprendizaje recomendadas y actividades de aprendizaje evaluadas.



Semana 2

Unidad 2. Programación en PHP

Estimado alumno, continuamos con la segunda unidad, para adentrarnos en el desarrollo de las aplicaciones web. Este es un tema fundamental para comprender las siguientes unidades. Le recomendamos tomarse un tiempo prudencial para revisarlo porque en esta unidad se explica todo el proceso del diseño web, pero tranquilo, vamos paso a paso.



En la actualidad, la *web* está experimentando transformaciones significativas, lo que ha llevado a los expertos en el área a adoptar herramientas y técnicas propias de la ingeniería del *software*. Estas permiten garantizar tanto el correcto funcionamiento como una gestión eficiente de los sitios *web*.

Por ello, es indispensable que los sistemas, herramientas y servicios digitales cuenten con una serie de atributos y características que, en conjunto, conforman un concepto clave para alcanzar el éxito: la calidad.

En el caso de las aplicaciones *web*, disponer de atributos como usabilidad, navegabilidad, seguridad, mantenibilidad, entre otros, no solo mejora la eficiencia de la *WebApp*, sino que también incrementa significativamente la satisfacción del usuario final.

Para desarrollar una aplicación *web* (*WebApp*) es fundamental seguir un proceso sistemático que incluya etapas clave como la planificación, programación, seguridad y control de calidad. La calidad no puede ser añadida al final del proceso; debe integrarse desde el inicio mediante prácticas bien definidas.

Como resultado de un proceso de desarrollo centrado en la calidad, es común que surjan recomendaciones de mejora, que podrían llevar a decisiones estratégicas como lanzar una nueva versión del sitio *web* o modificar atributos ausentes o deficientemente implementados.

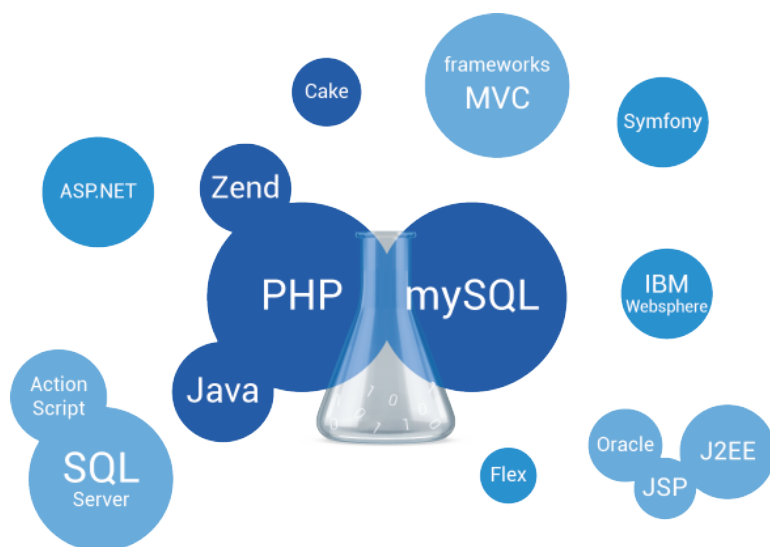
Es importante destacar que, el desarrollo *web* establece una clara distinción entre un sitio *web* y una aplicación *web*. Mientras que, un sitio *web* se enfoca en la publicación de contenido generalmente estático, con un nivel limitado de interactividad, una aplicación *web* ofrece un entorno más dinámico e interactivo, con funcionalidades que se asemejan a las de un *software* tradicional.



La **interactividad** es la característica principal que diferencia a las aplicaciones web. Por ejemplo, una aplicación web básica puede incluir formularios interactivos, mientras que las más complejas pueden manejar conexiones a bases de datos remotas, ofrecer gestores de contenido, e integrar funcionalidades avanzadas orientadas al usuario final.

Figura 3

Elementos del desarrollo web



Nota. Adaptado de *Web Engineering* [Ilustración], por idearí, 2013, [idearí](#), CC BY 4.0.

Entonces, el **Desarrollo Web** son procesos, disciplinados y cuantificables al desarrollo eficiente, operación y evolución de aplicaciones de alta calidad en la *World Wide Web*. En este sentido, se hace referencia a las técnicas y herramientas que se utilizan en el desarrollo de aplicaciones web complejas. Como se aprecia en la figura 3, los elementos del desarrollo web se pueden organizar en lenguajes de programación, bases de datos, *frameworks* (conjunto estandarizado de procesos), tecnologías, etc., estas son de gran dimensión, creando así *WebApps* de alta calidad, usabilidad, accesibilidad, etc.

Dentro de este ecosistema tecnológico, la programación de aplicaciones web en PHP constituye una de las prácticas más comunes y versátiles en el desarrollo web moderno.

PHP (*Hypertext Preprocessor*) es un lenguaje de programación del lado del servidor, de código abierto, ampliamente utilizado para crear sitios y aplicaciones dinámicas. Su integración nativa con bases de datos como MySQL, su facilidad de aprendizaje y su compatibilidad con la mayoría de los servidores web, lo convierten en una herramienta poderosa tanto para desarrolladores principiantes como para proyectos profesionales. Gracias a su flexibilidad y extensa comunidad, PHP sigue siendo una opción clave en la construcción de sistemas web escalables, seguros y eficientes.

2.1 Configuración del entorno de desarrollo

Con la introducción al tema de programación en PHP, le invito a revisar el siguiente libro: [Aplicaciones Web con PHP](#), específicamente el Capítulo 4: "Introducción a PHP", en el cual se abordan los conceptos fundamentales del lenguaje. En este capítulo, encontrará una explicación clara de los elementos básicos y los principales conceptos clave que le permitirán iniciarse en el mundo del desarrollo web con PHP, comprendiendo su sintaxis, funcionamiento y utilidad en la creación de aplicaciones dinámicas.

¿Qué opina de la lectura?, ¿Tiene dudas o inquietudes? ¡A continuación, las iremos resolviendo!

Configuración de un servidor web

Antes de comenzar a construir nuestro primer sitio web, es necesario levantar, configurar y personalizar el entorno de desarrollo sobre el cual vamos a trabajar. A continuación, en la figura 4 se presentan las herramientas XAMPP y, posteriormente, se presentan los pasos necesarios para preparar adecuadamente dicho entorno:



Figura 4

Herramientas XAMPP



Nota. Adaptado de ¿Qué es XAMPP? [Ilustración], por Apachefriends, s.f., [Apachefriends](#), CC BY 4.0.

La aplicación que utilizaremos para nuestro entorno de trabajo es XAMPP. XAMPP es un paquete completo de instalación que facilita la configuración de un entorno de desarrollo web local. Este software, es multiplataforma, permite instalar de manera sencilla los componentes esenciales para el desarrollo web:

- **Apache:** servidor web.
- **MySQL:** sistema gestor de bases de datos.
- **PHP:** lenguaje de programación del lado del servidor ampliamente utilizado para crear aplicaciones web dinámicas.

La instalación de XAMPP es muy sencilla, ya que solo requiere descargar el instalador y seguir unos pocos pasos. En la Figura 4 se muestra el logotipo de XAMPP junto con sus principales componentes.

Para obtener más información y acceder a versiones actualizadas, le invito a visitar el sitio oficial de [XAMPP](#).

El proceso de instalación la realizaremos en cuatro etapas:

1. Descargar el XAMPP.
2. Ejecutar el XAMPP para su instalación.
3. Levantar los servicios de Apache y MySQL.
4. Realizar Pruebas de los servicios.

Una vez configurado el entorno XAMPP, es indispensable contar con un editor de código fuente que facilite la escritura, organización y depuración de nuestros archivos. Entre los más recomendados se encuentran [Visual Studio Code](#) y [Sublime Text](#), ambos editores livianos, potentes y ampliamente utilizados en el desarrollo web.

Podemos concluir que el proceso de transformar nuestro computador en un servidor web es muy sencillo, siempre y cuando se utilice correctamente el *software* adecuado para el sistema operativo que utiliza.

Sin embargo, es importante tener en cuenta que, dado que XAMPP está diseñado para facilitar la instalación y configuración rápida de un entorno local, incluye configuraciones predeterminadas que no son seguras para un entorno de producción; por esta razón, su uso debe limitarse a la etapa de desarrollo, pruebas y aprendizaje.

2.2 Primeros pasos con PHP

Para avanzar con el desarrollo web empezaremos viendo los conceptos básicos y sintaxis del lenguaje de programación PHP, el cual nos ayudará a crear las *WebApps*.

Antes de comenzar, es importante conocer las configuraciones necesarias que debe tener el lenguaje de programación PHP. Estas se gestionan a través del archivo

```
php.ini
```



, el cual es fundamental para definir cómo se comporta el entorno de ejecución de PHP. En este archivo se pueden ajustar parámetros como:

- Límite de tamaño para la subida de archivos.
- Zona horaria predeterminada.
- Niveles de notificación de errores.
- Tiempo máximo de ejecución de scripts, entre otros.

Para profundizar en este tema, conocer la ubicación, funciones y cómo editar correctamente el archivo php.ini, puede apoyarse en el siguiente artículo tutorial: [“¿Qué es el archivo PHP.ini y en dónde está? Comprende los parámetros principales y edición”](#).

Estimado alumno, espero que todo el proceso de instalación y configuración se haya hecho sin mayor problema y tengamos lista la plataforma sobre la cual vamos a trabajar. Si usted tiene dudas de si la aplicación está funcionando correctamente, le recomiendo que realice la siguiente actividad, esta nos ayudará a verificar correctamente si todo lo instalado está funcionando a la perfección.



Levante el servidor web en su máquina local y cree un archivo con extensión PHP y coloque el siguiente código:

```
<?php phpinfo();  
?>
```

Y hágalo correr en su servidor.

¿Qué resultados obtuvo? Si todo está bien, usted debe obtener un archivo similar al de la figura 5, donde se puede ver las versiones de las aplicaciones instaladas en el servidor web y las configuraciones de estas:



Datos de configuración, módulos y componentes de PHP



Puede descargar el ejemplo de la actividad anterior desde el [repositorio de código GitHub](#). Recuerde que, una vez descargado, debe colocarlo dentro de la carpeta htdocs de su entorno local. Por lo general, esta carpeta se encuentra en la siguiente ruta:

Para una mejor organización, se recomienda crear un directorio específico para sus prácticas, por ejemplo:

Esto le permitirá mantener sus archivos ordenados y acceder fácilmente desde el navegador mediante:

<http://localhost/practicadesarrolloweb>

Ahora sí, con la plataforma totalmente lista podemos empezar a desarrollar sitios web de una manera bastante sencilla, sigamos adelante con el siguiente tema, vamos a ver lo relacionado con la base de datos, cómo gestionar la información que se manipulará mediante nuestra aplicación web.

Conociendo el servidor MYSQL

Para profundizar en el tema de MySQL, le invito a revisar el siguiente Capítulo 8: “Acceso a Base de Datos” del libro [Aplicaciones Web con PHP](#). En este material encontrará una explicación detallada sobre cómo establecer una conexión entre PHP y una base de datos MySQL, así como las técnicas básicas para gestionar y manipular los datos almacenados. El contenido es ideal para iniciarse en el desarrollo de aplicaciones dinámicas con interacción con bases de datos.

¿Cómo le fue en la lectura? Muy interesante el capítulo, ¿verdad? Ahora usted ya tiene conocimiento de bases de datos y de SQL.

Para reforzar su lectura, acceda desde su equipo local al editor phpMyAdmin (entorno de administración de la base de datos MySQL), configure los usuarios de acceso y cree una base de datos de prueba. Si durante el desarrollo de este ejercicio encuentra alguna dificultad, no dude en plantear sus preguntas en el foro de la asignatura, donde podrá recibir orientación y apoyo.

¡Continuemos optimistas con la siguiente temática!

2.3 Variables

Pasamos a ver un nuevo tema en el desarrollo web, dentro del lenguaje PHP, lo que corresponde a variables y constantes.

Para conocer sobre el tema de variables en PHP, le recomiendo revisar la siguiente lección didáctica: “[Tipos de Variables en PHP](#)”. En este material se explica detalladamente cómo declarar variables en PHP. Además, incluye ejemplos prácticos que facilitan la comprensión del tema.



De acuerdo con lo expuesto anteriormente, le recomiendo desarrollar la siguiente actividad que le permitirá reforzar los conocimientos obtenidos en cuanto a la creación de variables en PHP.

Marque con un visto las variables que están creadas correctamente:



- \$var1=1;
- #var1=1
- Var1=1;
- \$var="2";

Nota: por favor, complete la actividad en un cuaderno o documento Word.

¿Cómo le pareció el tema de creación de las variables? ¡Muy sencillo, verdad! Es importante destacar que en PHP no se necesita convertir variables de enteros a caracteres, toda esta operación la realiza el compilador. Ahora que ya sabe cómo declarar variables en PHP.

Si tiene dudas, le invito a revisar el siguiente ejemplo, donde se declaran distintas [variables en PHP](#).

2.4 Constantes

La primera y gran diferencia que existe entre las variables y las constantes es que las últimas van a tener un valor fijo, es decir, su valor no se va a poder modificar durante la ejecución de una página. Por el contrario, una misma variable puede tomar varios valores en una misma ejecución.

Ahora, como práctica, le invito a declarar 4 constantes entre valores numéricos y valores de texto, en el Entorno de Desarrollo que instaló en su equipo local.



¿Cómo le fue con el desarrollo del ejercicio? ¿Tiene alguna duda? Si es así, puede descargar el siguiente ejemplo, donde se declaran [constantes en PHP](#). Esto le permitirá comparar con su propio código y detectar cualquier diferencia o novedad que pueda ayudarle a mejorar su comprensión.

Ahora que ya sabe cómo declarar variables y constantes en PHP, es hora de ver un nuevo tema:

2.5 Operadores

Para aprender sobre los operadores en PHP, le recomiendo revisar la siguiente lección didáctica: "[Operadores en PHP – Tipos y Uso](#)". En esta conocerá los distintos tipos de operadores disponibles y cómo aplicarlos correctamente en el desarrollo de una aplicación. Este material explica de manera detallada los operadores:

- Aritméticos
- De asignación
- De comparación
- Lógicos
- De incremento/decremento
- De cadenas

Adicionalmente, incluye ejemplos prácticos que le ayudarán a comprender cómo utilizar cada uno de ellos de forma efectiva.

¿Cómo le pareció el ejemplo?, espero que sea claro y entendible. Ahora ya sabe qué es un operador aritmético, operadores de comparación, operadores lógicos y operadores de unión de cadenas.

Está en capacidad de ejemplificar cada uno de ellos. Así pues, le recomiendo realizar la siguiente actividad:

Llene la siguiente tabla en su cuaderno o documento Word con un resumen de las características de cada uno de los operadores revisados.



Completar: Ejercicio de las características de los operadores

Operadores aritméticos	Operadores comparación	de	Operadores lógicos	Operadores unión
+, -, *, /, %, ++, --, **				
Operaciones matemáticas				
Incrementos				

Esta tabla le ayudará a resumir todos los tipos de operadores que existen y su forma de utilización y funcionamiento.

Si quiere afianzar sus conocimientos, puede revisar un ejemplo de aplicación de cada uno de ellos en el siguiente [repositorio de código GitHub](#).

Ahora que ya se tiene conocimiento de este tema, pasamos a un nuevo e interesante apartado: las estructuras de control.

2.6 Estructuras de control

Las estructuras de control son instrucciones utilizadas en programación para ejecutar una serie de acciones en las aplicaciones que creamos, en programación web nos ayudará de mucho para realizar algoritmos que impliquen condiciones de ejecución.

Es tiempo de apoyarnos en la siguiente lección didáctica: "[Estructuras de control](#)". Para ello, le invito a revisar los conceptos de cómo se gestionan estas estructuras en PHP.

Como ya habrá revisado, existen diferentes tipos de estructuras de control, entre ellas:

- Instrucciones condicionales.
- Instrucciones de bucle.
- Otras instrucciones.





Una vez detectados los tipos de estructuras de control, realice un resumen de las características de cada uno de ellos, ejemplifíquelos y publíquelos en el foro de la clase. No olvide incluir ejemplos diferentes a los que nos plantea la lección didáctica, esto nos ayudará a reforzar los conocimientos obtenidos mediante la lectura.

¿Tuvo errores al ejecutar los ejemplos? Si los datos no se muestran correctamente en su página web, no se preocupe. Es algo común cuando se está comenzando.

Para solucionar estos inconvenientes, puede revisar el siguiente [ejemplo funcional en PHP](#), donde encontrará un código funcional que le servirá como guía para comparar con su código y corregir posibles fallos.

Además, recuerde que cada semana contamos con un espacio dedicado para resolver dudas con el docente tutor. Consulte el horario de tutoría disponible en el entorno virtual de aprendizaje.

Contenidos, recursos y actividades de aprendizaje recomendadas



Semana 3

Unidad 2. Programación en PHP

2.7. Funciones

Estamos avanzando en contenidos interesantes de nuestra materia de Desarrollo web, le invito a seguir con ese mismo ánimo y cumplir los objetivos trazados al inicio del periodo de estudio.

Una función es un bloque de código que introducimos en nuestra página y que puede ser utilizado a lo largo de todo nuestro código PHP. La principal ventaja de las funciones es que nos permiten ahorrar y reutilizar código.



Para continuar con su aprendizaje en PHP, le recomiendo consultar el “capítulo 5: Funciones y objetos en PHP” del texto [Aprender PHP, MySQL y JavaScript con jQuery, CSS y HTML5](#). Este texto es un excelente apoyo para mejorar sus habilidades y estructurar mejor sus aplicaciones.

Es hora de poner en práctica los conocimientos obtenidos en este capítulo. Para ello, observe detenidamente el ejemplo planteado a continuación, en la figura 6, e identifique los errores que existen en el ejemplo de funciones.

Figura 6

Ejemplo de función

```
1  <?php
2      function calculoiva($monto){
3          $calculo=$monto * 0.12;
4          $total = $monto + $calculo;
5          print "El monto total a pagar es: ".$total;
6      }
7  >>
8  <!DOCTYPE html>
9  <html>
10 <head>
11     <title>Calculo de Iva</title>
12 </head>
13 <body>
14 <?php
15     print calculoiva(4000),
16 >>
17 </body>
18 </html>
```

Nota. Torres, D., 2025.

Con base en el ejemplo de la figura anterior, liste los errores encontrados de funciones y explique el problema y su solución.

Nota. Por favor, completar la actividad en su cuaderno de trabajo o en un documento de Word.

Ahora ya tiene conocimiento de los errores relacionados con las funciones y la forma de resolverlos y el porqué del problema. Esto le ayudará, con futuros trabajos, a implementar la mejor solución ante un problema de esta magnitud.



En su entorno local trabaje en un ejemplo donde se aplique el concepto de funciones, si surgen novedades en la ejecución, acuda al entorno virtual de aprendizaje o aproveche la tutoría síncrona con el docente tutor.

Estimado alumno, demos el siguiente paso y descubramos un nuevo tema:

2.8 Sesiones

Cuando se habla de sesiones, surge el término “*cookie*”, términos que, si bien están relacionados, no son lo mismo.

Las “*cookies*” son pequeños ficheros de texto que se generan en el servidor para conocer datos de los usuarios y emplearlos, si es necesario, en cada una de sus visitas; pero se almacenan en el navegador. Las *cookies* se pueden usar para, entre otras cosas, recordar los datos de acceso del usuario y las preferencias de usuario.

En cambio, la sesión almacena datos en el servidor, y usa una *cookie* de identificación (como

PHPSESSID

) para asociar al usuario con sus datos.

Para trabajar con sesiones en PHP, revise el siguiente video: “[Cómo trabajar con sesiones en PHP](#)”, donde aprenderá a mantener la información del usuario en su sitio *web* de forma segura y eficiente.

Si ya terminó de ver el video, Ud. ya está en capacidad de diferenciar lo que son las *cookies* y las sesiones; ambas nos ayudan para aplicar seguridad en nuestra aplicación *web*, ¿verdad?

Recuerde: las sesiones dependen de una *cookie* para identificar al usuario. Es decir: el navegador guarda la *cookie* con el ID de sesión, mientras que el servidor guarda los datos reales de esa sesión.





Actividades de aprendizaje recomendadas

Reforcemos el aprendizaje resolviendo las siguientes actividades:

1. Argumente lo que realiza la función `session_start()`, y cree un ejemplo completo de sesiones y comparta su experiencia en la tutoría síncrona con sus compañeros y tutor.

Nota. por favor, conteste la actividad en un cuaderno de apuntes o documento Word.

2. Antes de pasar al siguiente capítulo, le invito a desarrollar la siguiente autoevaluación para determinar el nivel de aprovechamiento del tema estudiado.



Autoevaluación 2

Conteste correctamente las siguientes preguntas:

1. Para el desarrollo web se utilizan metodologías:
 - a. Sistemáticas.
 - b. Multidisciplinarias.
 - c. Disciplinadas.
 - d. Cuantificables.
2. Para configurar su equipo personal como un servidor web necesita instalar:
 - a. Windows 10.
 - b. Apache.
 - c. PHP.
 - d. Photoshop.



3. Seleccione cuál de las alternativas es el gestor de base de datos que se instala con XAMPP:

- a. Oracle.
- b. SQL Server.
- c. MySQL.

4. Seleccione las etiquetas para ingresar código PHP:

- a. `<php> ... </php>`.
- b. `<?php ... ?>`.
- c. `<php? ... ?>`.

5. Seleccione la o las formas de declarar una variable en PHP:

- a. `$var1=1;`
- b. `&var1=1;`
- c. `%var1=1;`

6. Las constantes son:

- a. Variables aleatorias que cambian con el compilador del programa.
- b. Variables que son generadas por el gestor de base de datos y no cambian.
- c. Son variables que tienen un valor fijo en toda la ejecución del programa.

7. Seleccione el o los operadores de comparación:

- a. `=`.
- b. `-`.
- c. `==`.
- d. `j=`.

8. Seleccione el o los operadores de incremento:

- a. `&&`.



- b. ++.
- c. j-.
- d. %.

9. Seleccione la estructura correcta de una instrucción condicional:

- a. For($i=0; i>10; i++$).
- b. If($i==10$){ .. }.
- c. If($i=0; i>10; i++$){ .. }.

10. Una función es:

- a. Un ciclo repetitivo de código que se ejecuta al inicio de cada compilación del programa.
- b. Es un bloque de código que se ejecuta cuando es invocada en cualquier parte del programa.
- c. Una estructura de control para verificar una condición de una variable.

[Ir al solucionario](#)

Si no logró un buen resultado en la autoevaluación, no se preocupe, le recomiendo leer nuevamente el/los temas confusos y reforzar sus conocimientos. Y si aún tiene inquietudes, no dude en preguntar al tutor



¡Suerte! Siga adelante.

¡Hemos terminado la segunda unidad! Felicitaciones.



Resultado de aprendizaje 6 y 7:

- Explica la diferencia entre la programación orientada a eventos y programación de línea de comandos.
- Diseña, codifica, prueba y depura programas simples basados en eventos que respondan a eventos de usuario.

Para alcanzar los resultados de aprendizaje planteados, es fundamental entender las diferencias entre la programación orientada a eventos y la programación de línea de comandos, destacando cómo cada enfoque gestiona la interacción y el flujo de datos. Además, es esencial diseñar, codificar, probar y depurar programas simples que reaccionen a eventos de usuario. Utilizar PHP con MySQL permitirá crear aplicaciones interactivas que respondan eficientemente a estas acciones, integrando el manejo de eventos con la gestión de datos.

Contenidos, recursos y actividades de aprendizaje recomendadas

Recuerde revisar de manera paralela los contenidos con las actividades de aprendizaje recomendadas y actividades de aprendizaje evaluadas.



Semana 4

Unidad 3. PHP con MySQL

Empezamos un nuevo capítulo en el mundo del desarrollo *web*. Este se refiere al manejo de los datos con MySQL. De entre las muchas ventajas que se obtienen al crear una base de datos en una página *web*, una de las principales es que se consiguen páginas dinámicas.

Para usar y manejar la base de datos MySQL, vamos a emplear una aplicación muy usual y extendida entre los usuarios: **phpMyAdmin**, que es una herramienta gratuita y de código abierto que permite administrar bases de datos MySQL y MariaDB a través de una interfaz *web*.



3.1 Gestión de datos

Para comprender de forma clara el uso y la aplicación de phpMyAdmin, le recomiendo consultar la siguiente página web: "[phpMyAdmin en XAMPP](#)". En esta se explica:



- ¿Qué es phpMyAdmin?
- ¿Cómo iniciarlo correctamente desde XAMPP?
- ¿Cómo utilizar su interfaz para gestionar bases de datos MySQL de manera visual y eficiente?

Para iniciar el proceso de creación de una base de datos, es necesario primero activar el servicio de MySQL, lo cual puede hacer de dos formas:

1. A través del Panel de Control de XAMPP, activando el módulo MySQL.
2. Mediante la línea de comandos (CMD) de Windows, ejecutando los comandos correspondientes para iniciar MySQL.

Una vez que el servicio esté en ejecución, podrá acceder al entorno donde se puede ejecutar la sentencia SQL de creación de la base de datos, ya sea desde phpMyAdmin, la consola de MySQL o cualquier cliente que administre bases de datos.

En los siguientes puntos vamos a ver los comandos para gestionar una base de datos desde consola.

3.2 Creación de base de datos

La sentencia SQL para crear una base de datos es:

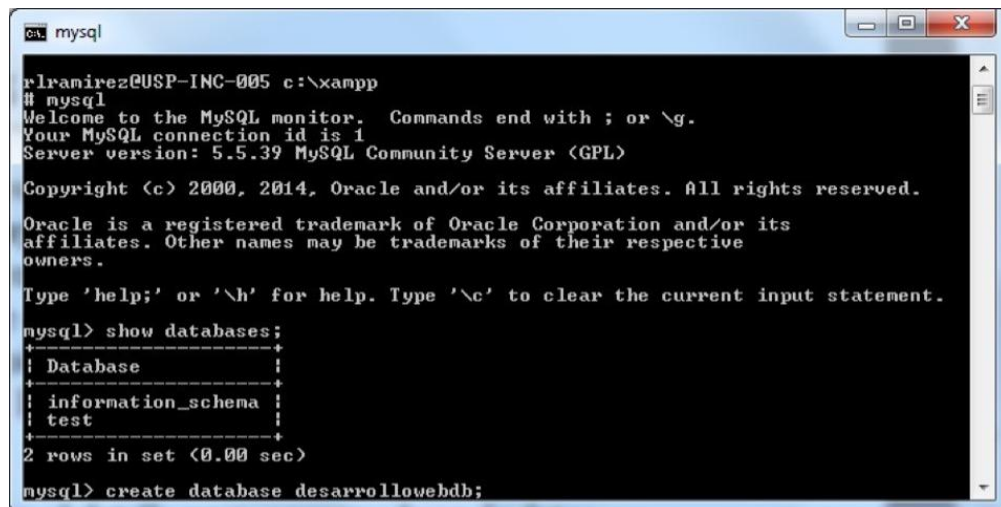
```
create database desarrollowebdb;
```

Como se muestra en la figura 7.



Figura 7

Consola MySQL, creación de base de datos



```
mysql
r1ramirez@USP-INC-005 c:\xampp
# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.5.39 MySQL Community Server (GPL)

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| test      |
+-----+
2 rows in set (0.00 sec)

mysql> create database desarrollowebdb;
```

Nota. Ramírez, R., 2020.

3.3 Listar las bases de datos de nuestro servidor

Los comandos para gestionar la base de datos son:

- Para listar las bases de datos es:

```
show databases;
```

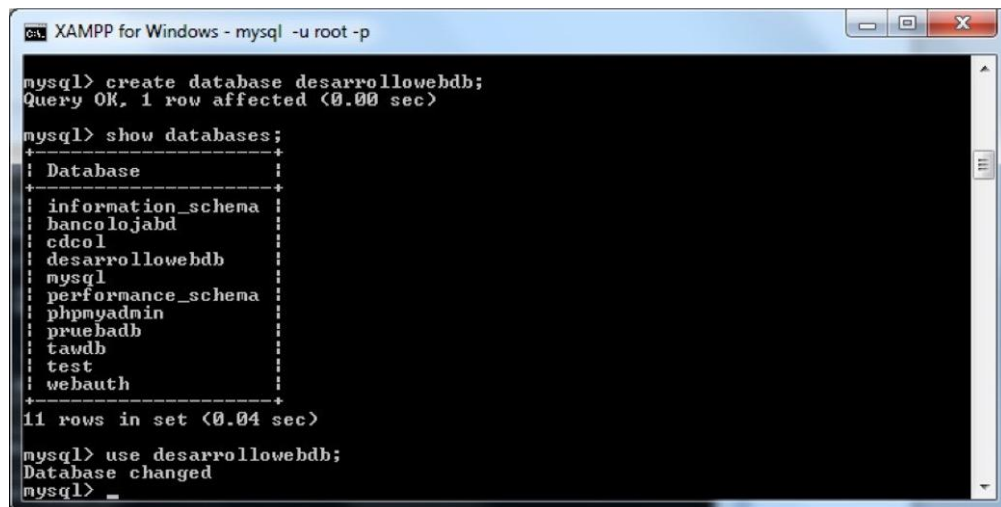
- Para ingresar a la edición de la base de datos es:

```
use desarrollowebdb;
```

Como se muestra en la figura 8.

Figura 8

Consola MySQL, listado y uso de base de datos



```
mysql> create database desarrollewebdb;
Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| banco_lojabd      |
| cdcol             |
| desarrollewebdb   |
| mysql              |
| performance_schema |
| phpmyadmin         |
| pruebadb           |
| tawdb              |
| test               |
| webauth            |
+-----+
11 rows in set (0.04 sec)

mysql> use desarrollewebdb;
Database changed
mysql>
```

Nota. Ramírez, R., 2020.

3.4 Creación de tablas en la base de datos

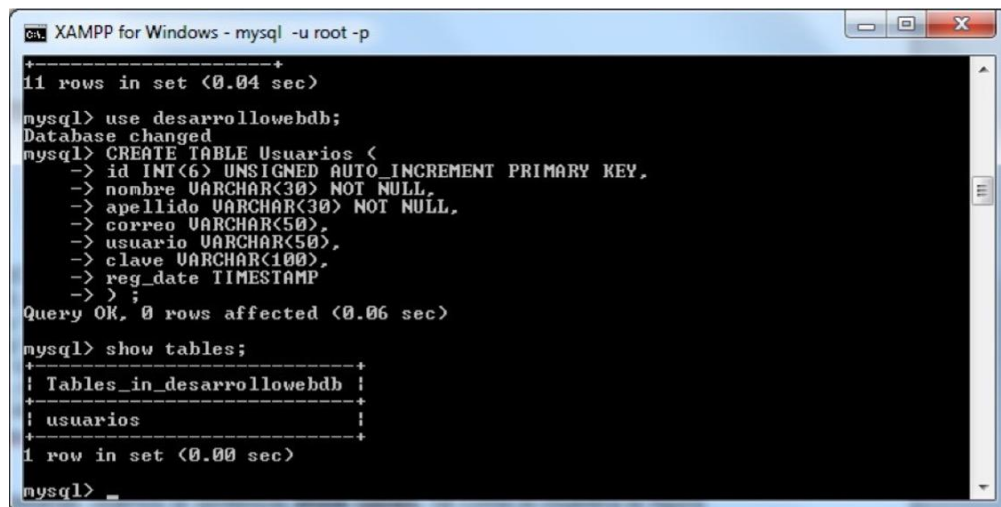
Para crear una base de datos en consola necesitamos estructurar la sentencia SQL de la siguiente forma:

```
CREATE TABLE usuarios(
  id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY
  KEY, nombre VARCHAR(30) NOT NULL,
  apellido VARCHAR(30) NOT NULL,
  correo VARCHAR(50),
  usuario VARCHAR(50),
  clave VARCHAR(100),
  reg_date TIMESTAMP
);
```

Y para poder listar las tablas creadas, usamos la sentencia **"show tables;"**, como se aprecia en la figura 9:

Figura 9

Consola MySQL, creación de tablas



```
XAMPP for Windows - mysql -u root -p
+-----+
11 rows in set (0.04 sec)

mysql> use desarrollowebdb;
Database changed
mysql> CREATE TABLE Usuarios (
  -> id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
  -> nombre VARCHAR(30) NOT NULL,
  -> apellido VARCHAR(30) NOT NULL,
  -> correo VARCHAR(50),
  -> usuario VARCHAR(50),
  -> clave VARCHAR(100),
  -> reg_date TIMESTAMP
  -> );
Query OK, 0 rows affected (0.06 sec)

mysql> show tables;
+-----+
! Tables_in_desarrollowebdb !
+-----+
! usuarios !
+-----+
1 row in set (0.00 sec)

mysql> _
```

Nota. Ramírez, R., 2020.

3.5 Manipulación de datos con SQL

• Insertar y listar datos en una tabla

Para insertar los datos en la base de datos MySQL se usa la siguiente sentencia SQL:

```
INSERT INTO usuarios (nombre, apellido, email, usuario,
clave) VALUES ('Ramiro', 'Ramírez',
'rlramirez@utpl.edu.ec', 'rlramirez', '12345');
```

De la misma forma, la sentencia SQL para listar los registros de la base de datos es:

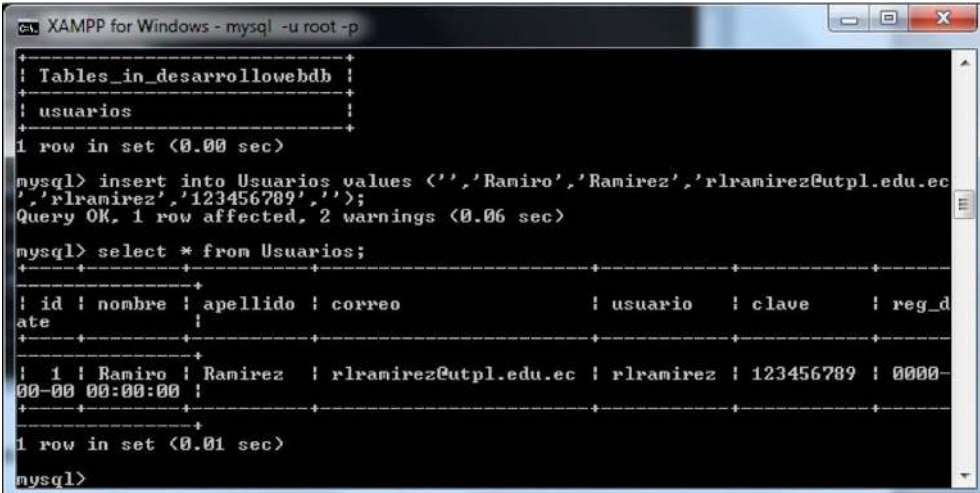
```
select * from usuarios;
```

Los resultados los tenemos en la figura 10.



Figura 10

Consola MySQL, insertar datos en una tabla



```
XAMPP for Windows - mysql -u root -p
+-----+
! Tables_in_desarrollowebdb !
+-----+
! usuarios                    !
+-----+
1 row in set (0.00 sec)

mysql> insert into Usuarios values ('','Raniro','Ramirez','rlramirez@utpl.edu.ec',
'rlramirez','123456789','');
Query OK, 1 row affected, 2 warnings (0.06 sec)

mysql> select * from Usuarios;
+-----+
! id ! nombre ! apellido ! correo                ! usuario ! clave ! reg_d
ate !
+-----+
! 1 ! Raniro ! Ramirez ! rlramirez@utpl.edu.ec ! rlramirez ! 123456789 ! 0000-
00-00 00:00:00 !
+-----+
1 row in set (0.01 sec)

mysql>
```

Nota. Ramírez, R., 2020.

• Borrar datos en una tabla

Para borrar los datos o registros en la base de datos MySQL, se emplea la siguiente sentencia SQL:

```
delete from Usuarios where id=1;
```

Es importante colocar el condicional, ya que se corre el riesgo de eliminar todos los registros en la tabla.

• Actualizar datos en una tabla

Para actualizar los datos o registros en la base de datos MySQL, se utiliza la siguiente sentencia SQL:

```
update usuarios set nombre='Carlos',
apellido='Jaramillo' where id=1;
```

Es importante colocar el condicional, ya que se corre el riesgo de actualizar todos los registros de la tabla.



En este último tema de la semana vamos a conocer cómo se gestiona una base de datos desde la aplicación **phpMyAdmin**.

3.6 Gestión de Datos con PhpMyadmin

phpMyAdmin es una herramienta web que le permite administrar bases de datos MySQL o MariaDB de forma visual, sin escribir código SQL directamente. Viene incluida con XAMPP y otros entornos de desarrollo local.

Sus ventajas son:

- Ahorra tiempo y evita errores de sintaxis.
- Ideal para principiantes que aún no dominan SQL.
- Permite explorar y entender mejor la estructura de una base de datos.



Para aprender a utilizar esta herramienta, puede apoyarse en el siguiente video: "[Crear base de datos con tablas y registros, en MySQL](#)".

Este video le guía paso a paso en el uso de la herramienta, mostrando de forma práctica cómo:

- Acceder a **phpMyAdmin** desde XAMPP.
- Crear y administrar bases de datos y tablas.
- Insertar, editar y eliminar registros sin necesidad de escribir código SQL.
- Realizar tareas comunes como importar, exportar y respaldar datos.

Además, recuerde que cada semana contamos con un espacio dedicado para resolver dudas con el docente tutor. Consulte el horario de tutoría disponible en el entorno virtual de aprendizaje.





Unidad 3. PHP con MySQL

3.7 Conectar una base de datos con PHP



Para entender un poco más la conexión entre MySQL y PHP, le sugiero revisar el “capítulo 14: PHP y MySQL” del texto [Creación de un sitio web con PHP con MySQL](#) y realizar una lectura comprensiva y práctica de los temas y ejemplos presentados. Esto le ayudará a entender mejor el funcionamiento.

Es importante saber los parámetros requeridos para poder conectarse a la base de datos MYSQL con PHP, tal como lo muestra el ejemplo del texto, son 4. Estos parámetros dependerán de la configuración de su ordenador.

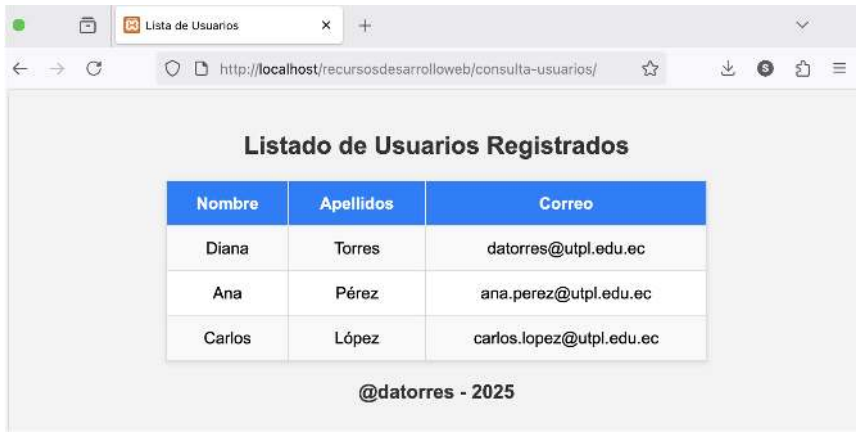
En el siguiente ejemplo, se desarrolla una aplicación web básica que incluye un formulario HTML sencillo. Este formulario permite seleccionar usuarios registrados en una tabla de base de datos que contiene los campos: nombre, apellidos, correo y clave.

Le invito a revisar cuidadosamente el [código](#) para comprender la lógica detrás del funcionamiento de esta aplicación. Observará cómo se integran el *frontend* (formulario) y el *backend* (consulta a la base de datos) para lograr un resultado funcional.

El resultado esperado se muestra en la figura 11, como referencia visual para validar que el desarrollo es correcto.



Figura 11
Vista de usuarios desde el navegador web



Nombre	Apellidos	Correo
Diana	Torres	datorres@utpl.edu.ec
Ana	Pérez	ana.perez@utpl.edu.ec
Carlos	López	carlos.lopez@utpl.edu.ec

@datorres - 2025

Nota. Torres, D., 2025.

En la figura 12, se muestra una sección del código usado en la implementación de la aplicación web que usamos como ejemplo. En esta sección se puede ver el código PHP que construye la tabla y, mediante una estructura

```
while
```

, va presentando los registros de la tabla usuarios.



Figura 12

Código PHP del ejemplo de la figura 11

```
<?php
if ($resultado->num_rows > 0) {
    echo "<table>";
    echo "<tr><th>Nombre</th><th>Apellidos</th><th>Correo</th></tr>";

    while ($fila = $resultado->fetch_assoc()) {
        echo "<tr>";
        echo "<td>" . htmlspecialchars($fila['nombre']) . "</td>";
        echo "<td>" . htmlspecialchars($fila['apellidos']) . "</td>";
        echo "<td>" . htmlspecialchars($fila['correo']) . "</td>";
        echo "</tr>";
    }

    echo "</table>";
} else {
    echo "<p style='text-align:center;'>No hay usuarios registrados.</p>";
}

$conexion->close();
?>
```

Nota. Torres, D., 2025.

3.8 Gestión de datos con PHP y MYSQL CRUD

Estamos avanzando en el estudio del lenguaje de desarrollo PHP, ahora gestionando los datos con la base de datos MySQL, le invito a seguir trabajando en este emocionante mundo del desarrollo de *WebApps*.

Pasamos ahora al desarrollo de un CRUD, apoyándonos de PHP y HTML, para ello acuda nuevamente al texto sugerido en la sección anterior y continuemos revisando el capítulo 14.

En la sección anterior ya vimos cómo conectarnos a una base de datos y cómo listar los registros. Ahora trabajaremos complementando la práctica anterior para que se puedan registrar los usuarios de la tabla, a través de un formulario en HTML.

3.9 Formulario de registro

Vamos a crear un formulario que nos permita registrar los campos de la tabla usuarios. En la figura 13, se muestra la sección del código que nos permite realizar esta actividad.

Figura 13

Código de un formulario en HTML

```
<h2>Formulario de Registro de Usuario</h2>

<form action="guardar_usuario.php" method="POST">
  <label for="nombre">Nombre:</label><br>
  <input type="text" name="nombre" required><br><br>

  <label for="apellidos">Apellidos:</label><br>
  <input type="text" name="apellidos" required><br><br>

  <label for="correo">Correo:</label><br>
  <input type="email" name="correo" required><br><br>

  <label for="clave">Contraseña:</label><br>
  <input type="password" name="clave" required><br><br>

  <input type="submit" value="Registrar Usuario">
</form>
```

Nota. Torres, D., 2025.

El formulario posee un botón tipo

submit

que va a apuntar al archivo que tiene la funcionalidad de guardar los datos en la base de datos.

Recuerde que cada caja de texto tiene un nombre, este será el nombre de la variable en el cual nosotros haremos referencia para obtener el valor de cada variable.

En la figura 14, se muestra el código en PHP, el cual recepta las variables y las guarda en la base de datos.

Figura 14

Sección de código de inserción datos en la tabla

```
// Recoger los datos del formulario
$nombre = $_POST['nombre'];
$apellidos = $_POST['apellidos'];
$correo = $_POST['correo'];
$clave = $_POST['clave'];

// Validar si el correo ya está registrado
$verificar = $conexion->prepare("SELECT id FROM usuarios WHERE correo = ?");
$verificar->bind_param("s", $correo);
$verificar->execute();
$verificar->store_result();

if ($verificar->num_rows > 0) {
    echo "<p style='text-align:center; color:red;'>El correo ya está registrado.</p>";
} else {
    // Insertar el nuevo usuario
    $stmt = $conexion->prepare("INSERT INTO usuarios (nombre, apellidos, correo, clave)
    $stmt->bind_param("ssss", $nombre, $apellidos, $correo, $clave);
```

Nota. Torres, D., 2025.

En la sección del código de la figura anterior se recogen los datos enviados por el formulario

`registrar_usuario.html`

usando el método POST. Antes de insertar, consulta si ya hay un usuario con ese correo para evitar duplicados. Posteriormente, inserta el nuevo usuario usando una sentencia preparada (

`prepare()`

), lo que protege contra inyecciones SQL. Si todo va bien, muestra el mensaje: "Usuario registrado correctamente". En caso contrario, si el correo ya existe o hay un error, se presenta un mensaje de error.

Estimados alumnos, espero que sea de comprensión todo lo expuesto hasta ahora. Si existe alguna duda, pueden acudir al foro de actividades y plantear la inquietud. Estaré gustoso de responder sus preguntas.

Recuerde que cada semana contamos con un espacio dedicado para resolver dudas con el docente tutor. Consulte el horario de tutoría disponible en el entorno virtual de aprendizaje.





Semana 6

Unidad 3. PHP con MySQL

3.10 Borrar y actualizar registros

Para implementar la funcionalidad de eliminar y actualizar usuarios, partimos del código previamente desarrollado en la sección anterior, donde listamos los usuarios registrados desde la base de datos.

En esta etapa, reutilizamos ese módulo de listado y lo modificamos para agregar dos enlaces adicionales por cada registro:

- **“Borrar”** → para eliminar el usuario seleccionado.
- **“Actualizar”** → para editar los datos.

El resultado de esta modificación se muestra en la figura 15, donde ahora se visualizan todos los usuarios en una tabla, acompañados de enlaces para su gestión.

Figura 15

Lista de usuarios con las opciones de actualizar y eliminar registros

The screenshot shows a web browser window with the title 'Lista de Usuarios'. The address bar shows the URL 'http://localhost/recursosdesarrolloweb/gestion-usuarios/listar_usuarios.php'. The main content area displays a table titled 'Usuarios Registrados'.

Nombre	Apellidos	Correo	Acciones
Diana	Torres	datorres@utpl.edu.ec	Actualizar Borrar
Ana	Pérez	ana.perez@utpl.edu.ec	Actualizar Borrar
Carlos	López	carlos.lopez@utpl.edu.ec	Actualizar Borrar

Nota. Torres, D., 2025.

A continuación, se presenta la sección del código que permite construir la tabla que se mostró en la figura anterior.



Figura 16

Código del listado de los datos y CRUD

```
if ($resultado->num_rows > 0) {
    echo "<table>";
    echo "<tr><th>Nombre</th><th>Apellidos</th><th>Correo</th><th>Acciones</th></tr>";
    while ($fila = $resultado->fetch_assoc()) {
        echo "<tr>";
        echo "<td>{$fila['nombre']}</td>";
        echo "<td>{$fila['apellidos']}</td>";
        echo "<td>{$fila['correo']}</td>";
        echo "<td>";
        echo "<a href='editar.php?id={$fila['id']}'>Actualizar</a> |";
        echo "<a href='borrar.php?id={$fila['id']}' onclick='return confirm(\"¿Deseas eliminar este usuario?\")'>Borrar</a>";
        echo "</td>";
        echo "</tr>";
    }
    echo "</table>";
} else {
    echo "<p>No hay usuarios registrados.</p>";
}
$conexion->close();
?>
```

Nota. Torres, D., 2025.

En la sección de código se puede observar que, al momento de crear los enlaces para las acciones de actualizar y eliminar, se incluye intencionalmente el

`id`

del usuario en la URL. Esto es fundamental, ya que el

`id`

actúa como identificador único del registro en la base de datos, lo que permite al sistema acceder directamente al usuario específico sobre el cual se desea realizar la acción (modificación o eliminación).

En la figura 17 se presenta el código PHP para eliminar un registro.



Figura 17

Código para eliminar datos de la tabla

```
<?php
if (isset($_GET['id'])) {
    $conexion = new mysqli("localhost", "root", "", "usuariosdb");
    if ($conexion->connect_error) die("Error de conexión");

    $id = intval($_GET['id']);
    $stmt = $conexion->prepare("DELETE FROM usuarios WHERE id = ?");
    $stmt->bind_param("i", $id);

    if ($stmt->execute()) {
        header("Location: listar_usuarios.php");
    } else {
        echo "Error al borrar el usuario.";
    }
    $stmt->close();
    $conexion->close();
}
?>
```

Nota. Torres, D., 2025.

Como se puede ver en el código, luego de eliminar el registro se redirecciona a la lista de usuarios registrados.

En la figura 18 se presenta el código que permite obtener los datos de un registro en específico.



Figura 18

Obtener los datos de un registro

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $conexion = new mysqli("localhost", "root", "", "usuariosdb");
    if ($conexion->connect_error) die("Error de conexión");

    $id = intval($_POST['id']);
    $nombre = $_POST['nombre'];
    $apellidos = $_POST['apellidos'];
    $correo = $_POST['correo'];

    $stmt = $conexion->prepare("UPDATE usuarios SET nombre = ?, apellidos = ?, correo = ? WHERE id = ?");
    $stmt->bind_param("sssi", $nombre, $apellidos, $correo, $id);

    if ($stmt->execute()) {
        header("Location: listar_usuarios.php");
    } else {
        echo "Error al actualizar el usuario.";
    }

    $stmt->close();
    $conexion->close();
}
?>
```

Nota. Torres, D., 2025.

Como pudo observar en el código, en algunas secciones se usa el ? en lugar de enviar directamente la variable en la consulta.

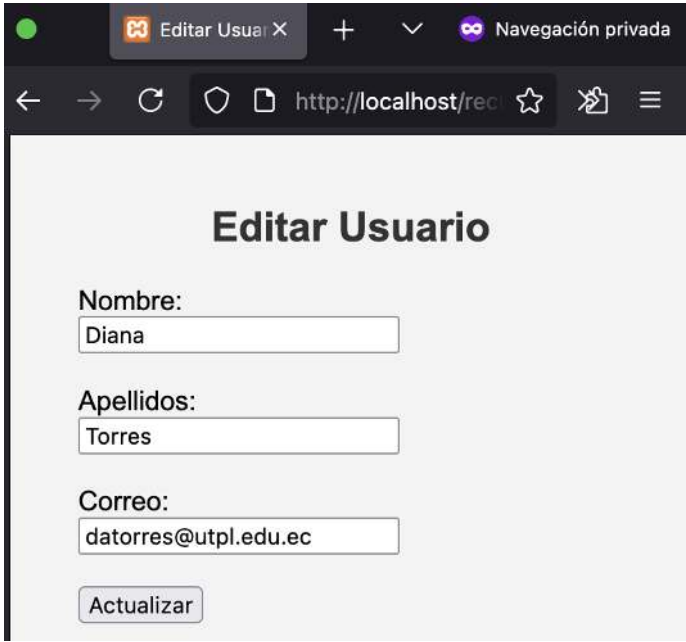
Para complementar esta temática, le invito a investigar cuál es el objetivo de colocar el ? en lugar de la variable, y a compartir esta investigación con sus compañeros en la tutoría. Por favor, tome notas en su documento o cuaderno sobre dudas o puntos de discusión

Como resultado tenemos el formulario actualizado tal como lo muestra la figura 19.



Figura 19

Formulario con los datos a modificar de un registro



The screenshot shows a web browser window with a single tab titled 'Editar Usuario'. The address bar displays 'http://localhost/reci'. The page content features a heading 'Editar Usuario' in bold. Below the heading are three labeled input fields: 'Nombre:' containing 'Diana', 'Apellidos:' containing 'Torres', and 'Correo:' containing 'datorres@utpl.edu.ec'. At the bottom of the form is a button labeled 'Actualizar'.

Nota. Torres, D., 2025.

Finalmente, cuando el cliente presiona el botón “Actualizar”, el código evalúa el contenido de las variables y ejecuta la sentencia de actualización, como se puede ver en el código de la figura 20.



Figura 20

Código para actualizar los datos de una tabla

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $conexion = new mysqli("localhost", "root", "", "usuariosdb");
    if ($conexion->connect_error) die("Error de conexión");

    $id = intval($_POST['id']);
    $nombre = $_POST['nombre'];
    $apellidos = $_POST['apellidos'];
    $correo = $_POST['correo'];

    $stmt = $conexion->prepare("UPDATE usuarios SET nombre = ?, apellidos = ?, correo = ? WHERE id = ?");
    $stmt->bind_param("sssi", $nombre, $apellidos, $correo, $id);

    if ($stmt->execute()) {
        header("Location: listar_usuarios.php");
    } else {
        echo "Error al actualizar el usuario.";
    }

    $stmt->close();
    $conexion->close();
}
?>
```

Nota. Torres, D., 2025.

Todo el código completo del proyecto se encuentra disponible en el repositorio de GitHub. Para revisarlo, acceda a la carpeta denominada "[Gestión de Usuarios](#)", donde tendrá acceso a todos los archivos necesarios del sistema.

Recuerde que con el código disponible podrá:

- Estudiar el código en detalle.
- Comprender su estructura y lógica.
- Descargarlo o clonarlo para realizar pruebas o modificaciones en su entorno local.

Esta es una excelente oportunidad para repasar y consolidar lo aprendido, observando cómo se integran los diferentes componentes del proyecto en un entorno real.



Actividad de aprendizaje recomendada

Continuemos con el aprendizaje mediante su participación en la actividad que se describe a continuación:

Estimado alumno, le invito a medir sus conocimientos de esta unidad con la siguiente autoevaluación. Esta le ayudará a valorar cuál es su nivel de conocimientos obtenidos durante el transcurso de este periodo transcurrido. En caso de no obtener resultados satisfactorios, le sugiero revisar nuevamente los contenidos en los cuales tiene problemas.



Autoevaluación 3

Conteste correctamente las siguientes preguntas de la unidad 3:

1. El editor de base de datos MySQL que se instala con XAMPP es:
 - a. xamppMyAdmin.
 - b. mysqlMyAdmin.
 - c. phpMyAdmin.
 - d. mysqlAdmin.
2. Seleccione la sentencia correcta para crear una base de datos en MySQL:
 - a. Create table proyecto1db;
 - b. Create database proyecto1db;
 - c. Create databases proyecto1db;
3. Seleccione la sentencia correcta para listar las bases de datos de nuestro servidor:
 - a. Select * from databases;
 - b. Select show databases;
 - c. Show databases;



4. La propiedad para crear una llave primaria de una tabla es:

- a. Auto_incremente.
- b. Primary key.
- c. Unsigned auto_increment.

5. La estructura correcta para insertar en una base de datos es:

- a. Insert into tabla values("","");
- b. Insert into tabla (campos) values("","");
- c. Insert tabla values("","");

6. La estructura correcta para listar los datos de una tabla es:

- a. Select * from tabla;
- b. Select * tabla;
- c. Select nombre, apellido from tabla;

7. La estructura correcta para eliminar los datos de una tabla es:

- a. Delete * from usuarios;
- b. Delete from usuarios where id=1;
- c. Delete usuarios where id=1;

8. La estructura correcta para actualizar los datos de un registro es:

- a. Update usuarios values (campos);
- b. Update usuarios set (capos) where id=2;
- c. Update usuarios set nombre='nuevo nombre' where id=1;

9. La sentencia para seleccionar una base de datos en PHP es:

- a. Mysqlconnect();
- b. Mysqlquery();
- c. Mysqlselectdb();



10. Para ejecutar una consulta en PHP de MYSQL es:

- a. Mysqlconnect();
- b. Mysqlquery();
- c. Mysqlfield();

[Ir al solucionario](#)

Si no logró un buen resultado en la autoevaluación, no se preocupe, le recomiendo leer nuevamente el/los temas confusos y reforzar sus conocimientos. Y si aún tiene inquietudes, no dude en preguntar al tutor.



Le sigo invitando a que continuemos con el mismo ritmo de trabajo con que empezamos, estamos por finalizar este bimestre.



Resultado de aprendizaje 1 a 7:

- Describe los problemas implicados en el desarrollo de una interfaz web.
- Resume la necesidad y las cuestiones implicadas en la ejecución y la integración de una página web.
- Explica la importancia de la interconexión entre sitios web para bases de datos subyacentes.
- Explica por qué los problemas de accesibilidad son una consideración importante en el desarrollo de páginas Web.
- Construye aplicaciones web accesibles y seguras.
- Explica la diferencia entre la programación orientada a eventos y programación de línea de comandos.
- Diseña, codifica, prueba y depura programas simples basados en eventos que respondan a eventos de usuario.

Contenidos, recursos y actividades de aprendizaje recomendadas



Semana 7

Actividades finales del bimestre

Durante esta semana nos enfocaremos en repasar las temáticas estudiadas en las semanas anteriores, con el objetivo de reforzar los conocimientos adquiridos antes de la evaluación presencial.

Le recomiendo ingresar al Entorno Virtual de Aprendizaje (EVA) para:

- Revisar las actividades calificadas.
- Identificar posibles dudas o temas que necesiten reforzamiento.
- Prepararse con mayor seguridad para la evaluación.

Este espacio de repaso será clave para consolidar su aprendizaje y resolver cualquier inquietud pendiente.





Recuerde que cada semana contamos con un espacio dedicado para resolver dudas con el docente tutor. Consulte el horario de tutoría disponible en el Entorno Virtual de Aprendizaje.

Contenidos, recursos y actividades de aprendizaje recomendadas



Semana 8

Actividades finales del bimestre

En la última semana del primer bimestre. Así que nos vamos a enfocar en la revisión del contenido, previo a la evaluación presencial. Se recomienda aprovechar esta semana revisando nuevamente, cada uno de los temas y si surgieran dudas, por favor, contactarse con el docente tutor para resolverlas.;





Segundo bimestre

Resultado de aprendizaje 6, 8, 9 y 10:

- Explica la diferencia entre la programación orientada a eventos y programación de línea de comandos.
- Desarrolla código que responde a las condiciones de excepción planteadas durante la ejecución.
- Describe los métodos de seguridad para cookies y las formas en las que pueden ser usadas para poner en peligro la privacidad del usuario.
- Identifica los sitios web que están utilizando gráficos de páginas web como iconos web.

Para alcanzar los resultados de aprendizaje planteados, es crucial diferenciar entre la programación orientada a eventos y la programación de línea de comandos, y desarrollar código que maneje de manera efectiva las excepciones que surgen durante la ejecución. Además, es importante entender cómo asegurar cookies y los riesgos para la privacidad que pueden derivarse de su mal uso. También se debe identificar los sitios web que usan gráficos de páginas web como iconos. La programación orientada a objetos ofrece herramientas clave para gestionar eventos y excepciones, incluyendo métodos para manejar eventos, decidir cuándo lanzar excepciones frente a errores, y gestionar adecuadamente las excepciones.

Contenidos, recursos y actividades de aprendizaje recomendadas

Recuerde revisar de manera paralela los contenidos con las actividades de aprendizaje recomendadas y actividades de aprendizaje evaluadas.





Unidad 4. Programación orientada a objetos

Estimado alumno, empezamos el cuarto capítulo de la asignatura abordando temas fundamentales que se requiere entender en Programación Orientada a Objetos en Desarrollo web.



Para iniciar el estudio del tema de POO, le recomendamos leer la sección de [Programación orientada a objetos](#) del Manual oficial de PHP.

¿Qué le pareció la lectura?, ¿tiene algunas dudas? No se preocupe, las vamos a ir resolviendo.

4.1 Métodos y funciones para manejar eventos

Para el manejo de métodos y funciones en PHP, trabajaremos en el desarrollo de una clase orientada a objetos que se encargue de realizar toda la gestión de la base de datos con MySQL. Esta clase será responsable de:

- Establecer la conexión con la base de datos.
- Ejecutar las consultas SQL necesarias (inserción, actualización, eliminación, selección).
- Exponer sus funcionalidades mediante métodos que puedan ser invocados desde un objeto.

Este enfoque le permitirá comprender cómo encapsular la lógica de acceso a datos y reutilizarla de forma estructurada dentro de una aplicación web.

La Programación Orientada a Objetos (POO) consiste en la creación de clases y la definición de objetos con el objetivo de organizar y reutilizar el código de manera eficiente a lo largo de toda nuestra aplicación web.



4.1.1 Objeto

Un objeto es una entidad independiente que posee sus propios datos (atributos) y comportamientos (métodos). Elementos cotidianos como ventanas, menús o carpetas en un sistema operativo pueden ser representados como objetos, ya que tienen características propias y realizan acciones específicas.

Por ejemplo, en el caso de un automóvil:

- El motor puede considerarse un objeto.
- Sus atributos describen características físicas (como tipo, potencia, cilindrada).
- Sus métodos definen su funcionamiento interno y cómo interactúa con otros componentes del vehículo, los cuales también son objetos como: ruedas, frenos, sistema eléctrico, entre otros).

4.1.2 Clase

Es un término genérico que, como su nombre lo indica, representa una clase, plantilla o conjunto; que define un conjunto de atributos y métodos que caracterizan a cualquier objeto de dicha clase.

En la figura 21 se muestra el código de creación de una clase ilustrativa en PHP. La clase se llama Auto, donde se definen: variables, constructores y métodos.



Figura 21

Código de creación de una clase auto en PHP

```
<?php

// Definición de la clase
class Auto {
    // Atributos (propiedades)
    public $marca;
    public $color;
    public $velocidad = 0;

    // Método constructor (se ejecuta al crear el objeto)
    public function __construct($marca, $color) {
        $this->marca = $marca;
        $this->color = $color;
    }

    // Método para acelerar
    public function acelerar($incremento) {
        $this->velocidad += $incremento;
        echo "El auto ha acelerado a " . $this->velocidad . " km/h<br>";
    }

    // Método para frenar
    public function frenar($reduccion) {
        $this->velocidad -= $reduccion;
        if ($this->velocidad < 0) {
            $this->velocidad = 0;
        }
        echo "El auto ha frenado a " . $this->velocidad . " km/h<br>";
    }

    // Método para mostrar información
    public function mostrarInformacion() {
        echo "Marca: " . $this->marca . "<br>";
        echo "Color: " . $this->color . "<br>";
        echo "Velocidad actual: " . $this->velocidad . " km/h<br>";
    }
}

?>
```

Nota. Torres, D., 2025.

Cuando se crean clases en PHP, las propiedades pueden ser públicas o privadas:

- a. `public` quiere decir que la propiedad se puede acceder y modificar desde fuera de la clase.
- b. `private` indica que no es posible ni leer ni reescribir la propiedad fuera de la clase y esto nos trae un nuevo concepto muy importante llamado encapsulamiento.

Una vez que se crea una clase en php, se puede instanciar objetos de esa clase con el objetivo de reutilizar código.

En la figura 22 se muestra el código PHP para instanciar objetos de la clase `auto.php`

Figura 22

Código de instanciación la clase `auto` en PHP

```
<?php
// Incluir la clase Auto
include 'Auto.php';

// Crear un objeto basado en la clase Auto
$miAuto = new Auto("Toyota", "Rojo");

// Mostrar información inicial
$miAuto->mostrarInformacion();

// Acelerar el auto
$miAuto->acelerar(50);

// Frenar el auto
$miAuto->frenar(20);

// Mostrar información final
$miAuto->mostrarInformacion();
?>
```

Nota. Torres, D., 2025.

Analizando el código de los ejemplos anteriores se pueden destacar los siguientes elementos:

- **Clase:** `Auto` define cómo será cualquier objeto de tipo auto.
- **Atributos:** `marca, color, velocidad` son datos internos del objeto.
- **Métodos:** `acelerar(), frenar(), mostrarInformacion()` son acciones que el objeto puede realizar.
- **`$this`:** Se usa para referirse a los atributos del propio objeto.
- **Objeto:** `$miAuto` es una instancia real de la clase, con sus propios datos.

Resumiendo, podemos decir que, con el uso de clases y objetos, es posible agrupar información y acciones de forma estructurada y reutilizable.

- Una clase funciona como una plantilla que define qué atributos y métodos tendrá un objeto. **Ejemplo:** Clase `Auto`
- Un objeto es una instancia concreta de esa clase. **Ejemplo:** `Toyota` es una instancia de la clase `Auto`.

Por tanto, al diseñar una clase, lo primero que debemos definir es un nombre genérico que represente el concepto o entidad que estamos modelando. Esto nos permitirá crear múltiples objetos basados en esa clase, cada uno con sus propios datos, pero compartiendo la misma estructura y comportamiento.





Unidad 4. Programación orientada a objetos

4.2 Propagación de eventos



Para iniciar el estudio de este tema, le sugiero que lea la guía de aprendizaje oficial denominada: “[MDN Web Docs - Introducción a los eventos](#)”. En esta guía, su sección de eventos cubre la propagación de manera concisa y clara, ofreciendo una buena base para entender cómo funcionan los eventos en un contexto real.

Con base en su lectura, podemos resumir que la propagación de eventos es una forma de permitir que los objetos se comuniquen sin estar fuertemente acoplados. Esto favorece la reutilización, escalabilidad y separación de responsabilidades en el diseño del *software*.

Este concepto es fundamental, principalmente, en entornos gráficos o interactivos (como interfaces de usuario *web*), y hace referencia al orden en que los eventos son gestionados por los diferentes objetos en una jerarquía visual o lógica.

Muchos lenguajes y *frameworks* actuales permiten detener la propagación de un evento en cualquier punto. Esto es útil si un objeto maneja completamente el evento y no quiere que los objetos superiores también reaccionen a él.

4.3 Manejo de excepciones

Para conocer sobre el manejo de errores y excepciones, puede revisar el [Curso de programación lógica básica a avanzada](#). Esta lección ofrece una explicación clara y concisa sobre qué son las excepciones, por qué son



importantes en la programación y cómo se manejan. Es una buena base para comprender los conceptos fundamentales del manejo de errores en cualquier lenguaje de programación.

Con base en su lectura, se puede decir que una excepción es un evento que ocurre durante la ejecución de un programa y requiere de la ejecución controlada de un bloque de código fuera del flujo normal de ejecución.

Esta técnica es una herramienta muy potente a la hora de realizar una manipulación de una situación ante cualquier posible imprevisto. Lo primero que hay que comprender es que una excepción no es un error. Es una situación fuera de lo común que se experimenta en un bloque de código y que este no es capaz de manejar.

4.4 ¿Cuándo lanzar una excepción y cuándo un error?

La decisión entre lanzar una excepción o manejar un error depende del contexto en el que se encuentra el código.

Si se está trabajando en una clase de alto nivel, estrechamente acoplada a la lógica específica de la aplicación, lo más adecuado suele ser gestionar el error localmente, utilizando mecanismos de control de errores definidos por la propia aplicación. Esto permite una respuesta inmediata y contextual, sin propagar el problema innecesariamente.

En cambio, si se trabaja en una clase de bajo nivel o genérica, diseñada para ser reutilizable y desacoplada del resto del sistema, lo recomendable es lanzar una excepción (

```
throw
```

). De esta forma, se delega la responsabilidad de manejo a las capas superiores, que podrán capturarla y responder adecuadamente según el flujo de la aplicación.

Cuando se lanza una excepción en PHP, el motor intentará encontrar un bloque

```
catch {}
```



que la capture. Si no se encuentra ninguno, se generará un error fatal con el mensaje “*Uncaught Exception*”, y se interrumpirá la ejecución del *script*. Para manejar adecuadamente una excepción, el código debe estar envuelto dentro de un bloque

```
try {}
```

seguido de al menos un bloque

```
catch {}
```

, donde se define cómo responder ante la excepción.

Veamos un ejemplo.



```
<?php function foo(){ try {  
    throw new Exception(< Upps !!>);  
} catch (Exception $e) {  
    hecho <ErrorException> . $e->getMessage();  
}  
}  
foo();
```

4.5 Métodos de una excepción

Una excepción es un objeto. A continuación, listamos los métodos que están disponibles en el lenguaje PHP para la correcta gestión de excepciones:

- `getMessage()` devuelve el mensaje de la excepción.
- `getCode()` devuelve el código del error.
- `getFile()` devuelve el nombre del fichero que lanzó la excepción.
- `getLine()` devuelve la línea del fichero que lanzó la excepción.
- `getTrace()` devuelve la pila de ejecución en formato de `array`.
- `getTraceAsString()` devuelve la pila de ejecución en formato de `string`.



Actividad de aprendizaje recomendada

Es hora de reforzar los conocimientos adquiridos resolviendo la siguiente actividad:

Estimado alumno, le invito a medir sus conocimientos de esta unidad con la siguiente autoevaluación. Esta le ayudará a valorar cuál es el nivel de conocimientos obtenidos durante el transcurso de este período transcurrido. En caso de no obtener resultados satisfactorios, le sugiero revisar nuevamente los contenidos en los cuales tiene problemas.



Autoevaluación 4

Conteste las siguientes preguntas de la unidad 4:

1. La Programación Orientada a Objetos es:

- a. La Programación Orientada a Objetos es la creación de clases y definimos objetos, esto con el fin de reutilizar código en toda nuestra WebApp.
- b. La Programación Orientada a Objetos con funciones que se crean para ser invocadas desde nuestra WebApp.
- c. La Programación Orientada a Objetos son instrucciones que crean para la gestión de la base de datos.

2. Un objeto es:

- a. Es un estilo de crear una función.
- b. Es una variable con características de valores numéricos.
- c. Es una entidad independiente con sus propios datos y programación.

3. Una clase es:

- a. Es un conjunto de objetos.
- b. Es un término genérico que, como su nombre lo indica, representa una clase, tipo o conjunto.
- c. Es una forma de gestión de base de datos.

4. En una clase las propiedades pueden ser:

- a. *Public*.



b. *Unsigned*.

c. *Private*.

5. Para acceder a las propiedades internas de las clases usamos:

a. `This->`.

b. `$this->`.

c. `$$this->`.

6. Para utilizar los métodos de una clase utilizamos el siguiente código:

a. `$miconexion = new clase_mysql;`

`$miconexion->conectar`

b. `$miconexion = clase_mysql;`

`$miconexion->conectar`

c. `$miconexion = new clase_mysql;`

`$ clase_mysql ->conectar`

7. Instanciar un objeto en PHP de la forma correcta es:

a. `$miconexion->consulta();`

b. `$miconexion->this->consulta();`

8. Para consumir los métodos internos de una misma clase, se realiza de la siguiente manera:

a. `Consulta();`

b. `This->consulta();`

c. `$this->consulta();`

9. Una excepción es:

a. Es un evento que ocurre durante la ejecución de un programa y requiere de la ejecución controlada de un bloque de código.



- b. Es un incidente que ocurre durante la ejecución de un programa y requiere de la ejecución controlada de un bloque de código.
- c. Es un evento que ocurre durante la ejecución de un programa.

10. Existen varios métodos de una excepción, seleccione los que sean correctos:

- a. getMessage().
- b. get\$Code().
- c. getFile().
- d. getLine().

[Ir al solucionario](#)

Si no logró un buen resultado en la autoevaluación, no se preocupe, le recomiendo leer nuevamente el/los temas confusos y reforzar sus conocimientos. Y si aún tiene inquietudes, no dude en preguntar al tutor.



Resultado de aprendizaje 5, 7, 9 y 10:

- Construye aplicaciones web accesibles y seguras.
- Diseña, codifica, prueba y depura programas simples basados en eventos que respondan a eventos de usuario.
- Describe los métodos de seguridad para cookies y las formas en las que pueden ser usadas para poner en peligro la privacidad del usuario.
- Describe formas de aumentar la fiabilidad de un sitio web, tales como certificados de seguridad.

Para alcanzar los resultados de aprendizaje planteados, es fundamental desarrollar aplicaciones web que sean accesibles y seguras. Esto implica diseñar, codificar, probar y depurar programas simples que respondan a eventos del usuario de manera efectiva. Además, es importante conocer los métodos de seguridad para cookies y cómo pueden comprometer la privacidad del usuario si no se manejan adecuadamente. También es necesario entender cómo aumentar la fiabilidad de un sitio web, utilizando medidas como certificados de seguridad. La comprensión de aspectos relacionados con la seguridad y vulnerabilidades, incluyendo ataques XSS y sus variantes no persistentes, es clave para proteger y fortalecer las aplicaciones web.

Contenidos, recursos y actividades de aprendizaje recomendadas

Recuerde revisar de manera paralela los contenidos con las actividades de aprendizaje recomendadas y actividades de aprendizaje evaluadas.



Semana 11

Unidad 5. Seguridad y vulnerabilidades

En el desarrollo de aplicaciones web, la seguridad es un factor clave. No es solo dar una milla adicional, sino una parte fundamental para proteger tanto a los usuarios como a los datos que manejamos.



Las vulnerabilidades son los “puntos débiles” en nuestras aplicaciones. Son fallos que un atacante puede explotar para robar información, causar problemas, entre otras amenazas. Identificarlas y corregirlas a tiempo es crucial para evitar inconvenientes.

En esencia, construir aplicaciones web seguras significa entender estas vulnerabilidades y aplicar las defensas necesarias desde las primeras fases de desarrollo del proyecto.



Para profundizar en este tema, puede consultar el artículo “[Strike - OWASP Top 10 2025: comprendiendo las vulnerabilidades comunes de aplicaciones web y los riesgos de seguridad](#)”. Este artículo da una visión general sobre las posibles tendencias y el enfoque de una futura actualización del OWASP Top 10.

En los siguientes temas vamos a abordar las técnicas de XSS más comunes en los últimos años.

5.1 Ataques XSS Cross- Site Scripting

¿Qué es *cross-site scripting*?

XSS ocurre cuando un atacante es capaz de inyectar un script, normalmente *JavaScript*, en el *output* de una aplicación web, de forma que se ejecuta en el navegador del cliente, es un problema de seguridad web en el que los cibercriminales ejecutan *scripts* maliciosos en sitios web legítimos o confiables. Los ataques se producen principalmente por validar incorrectamente los datos de usuario, y los *scripts* maliciosos se suelen inyectar mediante un formulario web o mediante un enlace alterado.

Los desarrolladores muchas veces subestiman el potencial peligro que conlleva este tipo de ataques. Si un atacante puede inyectar *JavaScript* en el *output* de una aplicación web y ejecutarlo, podrá ejecutar cualquier código *JavaScript* en el navegador de un usuario.



Algunos de los objetivos que se quieren conseguir con ataques XSS son:

- Robar *cookies* y sesiones de usuarios.
- Modificar el sitio *web*.
- Realizar HTTP *Requests* con la sesión del usuario.
- Redireccionar a usuarios a sitios dañinos.
- Atacar al navegador o instalar *malware*.
- Reescribir o manipular extensiones de navegador, etc.

Desde la perspectiva del navegador, el *script* es originado por la aplicación *web*, por lo que se asume como fuente fiable. Esta es una de las causas que permiten que estos ataques se puedan llevar a cabo.

Para ello, detallamos los tipos de ataques XSS que existen:

5.2 Tipos de ataques XSS



Para entender el tema, puede apoyarse en el siguiente artículo "[Excess XSS disponible](#)", donde se da una explicación detallada de todo lo que abarca el tema.

En resumen, existen tres tipos principales:

1. XSS reflejado (no persistente)

Es el tipo más común y sencillo. El código malicioso se envía como parte de una solicitud (puede ser, un enlace), y el servidor lo refleja directamente en la respuesta. Los atacantes, por lo general, usan esta técnica en enlaces engañosos o correos electrónicos fraudulentos, redirigiendo al usuario a una página donde el *script* se ejecuta de inmediato.

Al no almacenarse en el servidor, el atacante debe reenviar constantemente el enlace para afectar a nuevas víctimas.

2. XSS almacenado (persistente)



Es el tipo más peligroso y dañino, ya que el atacante inyecta el *script* directamente en la base de datos del sitio *web*, aprovechando entradas no validadas como formularios de comentarios o perfiles de usuario.

Cuando otro usuario accede a la página afectada, el *script* se ejecuta automáticamente sin que la víctima tenga que interactuar con nada. Este tipo de ataque puede propagarse fácilmente y tener un impacto más amplio, ya que el código permanece almacenado en el servidor.

3. XSS basado en DOM

Este ataque ocurre directamente en el navegador del usuario sin intervención del servidor. El atacante manipula el DOM (Modelo de Objetos del Documento) de la página, aprovechando cómo la aplicación maneja datos proporcionados por el usuario (como parámetros en la URL o fragmentos de *hash*).

Como la carga de datos no pasa por el servidor, estos ataques son más difíciles de detectar mediante *firewalls* o registros. Son comunes en aplicaciones modernas con mucho JavaScript, donde los datos del usuario se insertan dinámicamente en el contenido de la página.

En conclusión, los ataques XSS representan una de las amenazas más comunes y peligrosas en aplicaciones *web*. Cada tipo de ataque aprovecha distintos puntos vulnerables del sistema y requiere estrategias de mitigación específicas, como la validación y sanitización de entradas, así como el uso de políticas de contenido.

Recuerde que cada semana contamos con un espacio dedicado para resolver dudas con el docente tutor. Consulte el horario de tutoría disponible en el Entorno Virtual de Aprendizaje.



Resultado de aprendizaje 3, 5, 9 y 11:

- Explica la importancia de la interconexión entre sitios web para bases de datos subyacentes.
- Construye aplicaciones web accesibles y seguras.
- Describe los métodos de seguridad para cookies y las formas en las que pueden ser usadas para poner en peligro la privacidad del usuario.
- Describe formas de aumentar la fiabilidad de un sitio web, tales como certificados de seguridad.

Para alcanzar los resultados de aprendizaje planteados, es esencial entender la importancia de la interconexión entre sitios web y sus bases de datos subyacentes. Además, es fundamental desarrollar aplicaciones web que sean accesibles y seguras, abordando la seguridad de cookies y los riesgos que pueden comprometer la privacidad del usuario. También es necesario conocer estrategias para aumentar la fiabilidad de un sitio web, como el uso de certificados de seguridad. La gestión de seguridad y vulnerabilidades, incluyendo la prevención de ataques y la validación de datos, es clave para proteger y fortalecer las aplicaciones web.

Contenidos, recursos y actividades de aprendizaje recomendadas

Recuerde revisar de manera paralela los contenidos con las actividades de aprendizaje recomendadas y actividades de aprendizaje evaluadas.





Unidad 5. Seguridad y vulnerabilidades

5.3 Prevención de ataques

Así como un atacante puede afectar fácilmente un sitio web que no está protegido contra ataques *Cross-Site Scripting* (XSS), un desarrollador también puede implementar medidas efectivas para prevenirlos. La seguridad debe considerarse desde las etapas iniciales del desarrollo, incluso antes de escribir una sola línea de código.

La regla o política más básica que ha de tenerse siempre en cuenta es simple: nunca confíe en datos que vienen de usuarios o de cualquier otra fuente externa. Cualquier dato debe ser validado o escapado para su *output*.

Las medidas a tomar se pueden dividir en tres: *data validation*, *data sanitization* y *output escaping*.

5.4 Validación de datos



Para profundizar en este tema, puede consultar el artículo de MDN Web Docs "[Validación de formularios de datos](#)", donde se abordan los conceptos fundamentales de la validación de formularios, así como la importancia de no depender exclusivamente de la validación del lado del cliente para la seguridad, ya que esta puede ser eludida por usuarios malintencionados.

La validación de datos es el proceso de asegurarse de que su aplicación analiza el tipo de datos correctos. Si su *script* PHP espera un *integer* de un *input*, cualquier otro tipo de dato debe de rechazarse. Cada dato debe ser validado cuando se recibe para asegurarse de que es del tipo correcto, y debe ser rechazado si no pasa el proceso de validación.



Si quiere validar un número de teléfono, por ejemplo, deberá rechazar cualquier *string* que contenga letras, porque solo consistirá en dígitos. También puede tener en cuenta la longitud que deberán tener estos dígitos. Siendo más permisivo, se pueden aceptar algunos otros símbolos como +, (), y - que a veces se utilizan al indicar números de teléfono:

```
// Comprobar un número de teléfono en Estados Unidos:
$telefono = '1-909-466-4344';
if (preg_match('/^((1-)?\d{3})-\d{3}-\d{4}/', $telefono)){
    echo "El teléfono $telefono es válido";
} else {
    echo «El teléfono $telefono NO es válido»;
}
```

Data sanitization

La **sanitización de datos** se centra en manipular los datos para asegurarse de que son confiables, eliminando cualquier parte indeseable y normalizándolos en la forma correcta.

Según Ashwani (2025), este procedimiento es fundamental para proteger la información sensible y garantizar la privacidad.

Por ejemplo, si se espera un texto *string* de los usuarios, se puede querer evitar cualquier tipo de *markup* HTML:

```
// Sanitizar comentario de usuario.
$comentario = strip_tags($_POST[«comentario»]);
$telefono = «1234567»;
$telefono = preg_replace(</[^\d]/>, «», $telefono);
$length = strlen($telefono);
if ($length = 7 || $length = 10 || $length = 11){
    echo «$telefono es un formato válido»;
}
```

Recuerde que implementar políticas adecuadas de sanitización de datos es crucial para las organizaciones, ya que ayuda a prevenir filtraciones de información, cumplir con las regulaciones de protección de datos y mantener la confianza de los clientes.



Output escaping

Para proteger la integridad de los **datos que se devuelven**, el *output data*, se debe escapar cualquier dato que se devuelve al usuario.

Esto evita que el navegador malinterprete alguna **secuencia especial de caracteres**:

```
echo «Has buscado la palabra: « . htmlspecialchars($_GET[«query»]);
```

Puede emplearse también la función *htmlentities()*. La diferencia entre ambas es que *htmlspecialchars()* solo traduce los símbolos &, «», ¢, < y > en **entidades HTML**; en cambio, *htmlentities()* traduce todos los caracteres posibles que tengan su equivalencia en HTML. Normalmente, vale con *htmlspecialchars()* a no ser que use algún tipo de codificación diferente a **ISO-8859-1** o **UTF-8**.

Ejemplo de prevención contra ataques XSS

Mezclando un poco las tres **formas de prevenir ataques XSS**, vamos a ver una sección de código de un sencillo sistema de comentarios:

```
// Validar el comentario.
$comentario = trim($_POST["comentario"]);
if(empty($comentario)){
    exit("Debes proporcionar un comentario");
}
// Sanitizar comentario.
$comentario = strip_tags($comentario);
// El comentario ya se puede guardar de forma segura
file_put_contents("comentarios.txt", $comentario, FILE_APPEND);
// Escapar comentarios antes de mostrarlos
$comentarios = file_get_contents("comentarios.txt");
echo htmlspecialchars($comentarios);
```

Primero, nos aseguramos de que no se guardan **comentarios vacíos**. Después se *sanitizan* los datos, eliminando cualquier posible **etiqueta HTML** que pudiera contener. Finalmente, los comentarios se devuelven **filtrados**. La función *_striptags* hace que no sea posible insertar enlaces en los



comentarios, ya que estos utilizan una etiqueta que será eliminada. Para que puedan insertarse, se pueden utilizar *htmlentities* o *htmlspecialchars* en su lugar.



Si desea descargar el código para revisarlo, lo puede consultar en el [repositorio de código GitHub](#).

Hay que tener en cuenta que ninguna solución es fiable al 100 %, y que es conveniente estar al tanto de novedades respecto a los **ataques Cross-Site Scripting**, ya que van evolucionando a medida que lo hacen las plataformas que los facilitan (navegadores, HTML, etc.).

Ataques CSRF *Cross-Site Request Forgeries*

Los ataques *Cross-Site Request Forgeries*, también conocido como falsificación de solicitudes entre sitios, se producen cuando el atacante provoca que el usuario ejecute una acción de forma no intencionada en una aplicación en la que había iniciado sesión.

Por ejemplo, cuando el usuario “logueado” en su sitio favorito, procede a hacer clic en un enlace que parece inofensivo. En el fondo, su información de perfil está siendo actualizada con la dirección de *email* del atacante. El atacante puede ahora usar la opción de **recuperar la contraseña** en el sitio web para cambiar la contraseña de la cuenta mediante *email*.

Cualquier acción que pueda realizar un usuario cuando está logueado en un sitio web, la puede realizar también el atacante, como, por ejemplo:

- Actualizar su perfil.
- Añadir objetos a la cesta de la compra.
- Postear mensajes en un foro o cualquier otra cosa.

Recuerde que cada semana contamos con un espacio dedicado para resolver dudas con el docente tutor. Consulte el horario de tutoría disponible en el Entorno Virtual de Aprendizaje.





Actividad de aprendizaje recomendada

Es momento de aplicar su conocimiento a través de la actividad que se ha planteado a continuación:

Estimado alumno, le invito a medir sus conocimientos de esta unidad con la siguiente autoevaluación. Esta le ayudará a valorar cuál es el nivel de conocimientos obtenidos durante el transcurso de este período transcurrido. En caso de no obtener resultados satisfactorios, le sugiero revisar nuevamente los contenidos en los cuales tiene problemas.



Autoevaluación 5

Conteste la siguiente autoevaluación de la unidad 5:

1. El Cross Site Scripting es:

- a. Una medida de seguridad en el momento del desarrollo de las *WebApps*.
- b. Un ataque de concurrencia a nuestra aplicación web.
- c. Un ataque capaz de inyectar un script en las salidas de nuestra *WebApp*.

2. Los ataques se producen principalmente por:

- a. Uso incorrecto de la metodología de desarrollo.
- b. Inyectar código malicioso mediante un formulario web o mediante un enlace alterado.
- c. Una mala compra de los servidores de alojamiento.

3. Los objetivos de los ataques XSS son:

- a. Robar cookies o sesiones de los usuarios.
- b. Obtener datos como números de tarjetas de crédito.
- c. Modificar el sitio *web*.



4. Los tipos de ataques XSS pueden ser:

- a. Auto-persistent.
- b. Non-persistent XSS.
- c. Reflected XSS.

5. El ataque persistent XSS es:

- a. El código malicioso ya ingresó por el proceso de validación y está almacenado en los datos.
- b. Es el código malicioso que está buscando ingresar por algún puerto inseguro.
- c. Un código dañino en la configuración del servidor de base de datos.

6. La regla o política básica que se debe tener en cuenta ante estos ataques es:

- a. Dejar la seguridad en manos del servidor *web*.
- b. Nunca confiar en la configuración del servidor.
- c. Nunca confiar en los datos que vienen del usuario o fuente externa.

7. Las medidas a tomar en la prevención de los ataques pueden ser:

- a. *Data validation*.
- b. *Data sharing*.
- c. *Output escaping*.

8. La validación de datos es:

- a. La validación de datos es el proceso de asegurarse que tu aplicación analiza el tipo de datos correcto.
- b. La validación de datos es el proceso de analizar las propiedades de los formularios, ya que son estos por donde ingresa el usuario los datos.
- c. Ninguna de las anteriores.



9. En PHP, la función para sanitizar los datos es:

- a. Decode_utf8();
- b. Strleng();
- c. Strip_tags();

10. Los ataques CSRF tienen como objetivo:

- a. Enviar conexiones al servidor haciéndolo colapsar.
- b. El usuario, ejecute una acción de forma no intencionada.
- c. Eliminar archivos de configuración del servidor.

[Ir al solucionario](#)

Si no logró un buen resultado en la autoevaluación, no se preocupe, le recomiendo leer nuevamente el/los temas confusos y reforzar sus conocimientos. Y si aún tiene inquietudes, no dude en preguntar al tutor.



¡Hemos terminado la quinta unidad! Felicitaciones.



Resultado de aprendizaje 8 a 11:

- Desarrolla código que responde a las condiciones de excepción planteadas durante la ejecución.
- Describe los métodos de seguridad para cookies y las formas en las que pueden ser usadas para poner en peligro la privacidad del usuario.
- Identifica los sitios web que están utilizando gráficos de páginas web como iconos web.
- Describe formas de aumentar la fiabilidad de un sitio web, tales como certificados de seguridad.

Para alcanzar los resultados de aprendizaje planteados, es esencial desarrollar código que maneje adecuadamente las excepciones durante la ejecución. Además, es importante entender los métodos de seguridad para cookies y cómo pueden afectar la privacidad del usuario si no se utilizan correctamente. Identificar los sitios web que emplean gráficos como iconos también es crucial, así como describir las estrategias para aumentar la fiabilidad de un sitio web, como la implementación de certificados de seguridad. La seguridad web, que incluye la protección del servidor y el control de acceso mediante el archivo .htaccess, es fundamental para mantener la integridad y la seguridad de las aplicaciones.

Contenidos, recursos y actividades de aprendizaje recomendadas

Recuerde revisar de manera paralela los contenidos con las actividades de aprendizaje recomendadas y actividades de aprendizaje evaluadas.



Semana 13

Unidad 6. Seguridad web

Estimado alumno, empezamos el sexto capítulo de la asignatura, abordando temas fundamentales que se requieren entender dentro de la seguridad web.





Para iniciar el estudio de este tema, le invito a revisar el artículo titulado "[Evaluación multicriterio sobre herramientas de análisis de seguridad en aplicaciones web](#)" y consultar el *blog* "[Servidor Apache](#)".

Una vez que haya revisado estos materiales educativos, para comprender mejor el tema de la seguridad en aplicaciones web, le invito a consultar la guía educativa "[Seguridad de sitios web](#)" de Mozilla Developer Network. En este sitio encontrará una introducción clara y didáctica sobre los principales conceptos de seguridad en el desarrollo del lado del servidor, incluyendo prácticas recomendadas, amenazas comunes y estrategias para proteger su sitio web.

Después de estas lecturas, ¿qué opina de lo aprendido?, ¿tiene inquietudes o dudas? ¡A continuación, las iremos resolviendo!

6.1 Seguridad del servidor

La seguridad en un servidor web es fundamental porque este es el punto de acceso principal a los sitios y aplicaciones en línea. Cualquier vulnerabilidad puede tener consecuencias críticas tanto para los usuarios como para la organización que lo gestiona.

Las aplicaciones web tienen acceso abierto a un conjunto de información que explícitamente se hace pública. Sin embargo, en determinadas circunstancias, es interesante poder limitar el acceso a documentos reservados o útiles para un conjunto restringido de personas. Se pueden establecer dos tipos de restricciones:

1. Limitación de acceso **en función de direcciones IP o dominio**. Solo los usuarios de un dominio u organización tendrán acceso a la información.
2. Limitación de acceso **por nombres de usuario y claves de acceso**. Solo los usuarios que conozcan una clave de acceso válida pueden acceder a la información.



Otro aspecto que está cobrando especial importancia es la seguridad de la información que se intercambia en la web.

La *Internet* exige disponer de sistemas de comunicación seguros, capaces de adaptarse a las necesidades de los nuevos servicios, como la compra o la banca electrónica. En estos servicios, se manejan dos conceptos fundamentales: la autenticación y la confidencialidad.

Gracias a los sistemas de comunicación actualmente en uso, es técnicamente posible interceptar un enlace de comunicaciones y acceder al contenido de los datos que usan el protocolo TCP/IP que por él se transmiten. Cuando se envía información privada, por ejemplo, un número de tarjeta de crédito en un formulario de compra, es vital garantizar que la información sea recibida exclusivamente por su destinatario, y que la identidad es la esperada.

Por esta razón, la implementación de mecanismos de seguridad, como el cifrado y los certificados digitales, resulta esencial para proteger la confidencialidad e integridad de los datos durante su transmisión.

Control de acceso a la información

Se utiliza para limitar el acceso a determinados documentos de un servidor *web*, en función del origen y tipo de petición. La forma de hacerlo varía con el entorno en el que se publican las páginas (sistema operativo y servidor HTTP, principalmente). En general, todas las soluciones pasan por definir un fichero que contiene las diferentes limitaciones de acceso, en un formato característico del servidor HTTP. En algunos casos, se utiliza un fichero global con las restricciones de acceso o bien un fichero por cada directorio al que se quiere limitar el acceso.

Cuando un cliente *web* intenta acceder a un recurso protegido sin proporcionar las credenciales adecuadas, el servidor responde con un código de estado

HTTP 401

(Unauthorized). Este código indica que la solicitud no ha sido procesada debido a la falta de autenticación válida.



En otras palabras, el servidor reconoce que el recurso solicitado existe, pero requiere que el cliente se autentique antes de permitir el acceso.

Este mecanismo forma parte de las prácticas estándar de seguridad en aplicaciones *web* para controlar el acceso a contenidos confidenciales y restringirlos de usuarios no autorizados.

Para reforzar esta temática, realice una lectura del artículo [Consejos de seguridad](#) y comente los temas principales en el foro del EVA.



Asimismo, en el espacio de tutoría síncrona, comente cuál es el objetivo de estas líneas de código.

```
<Directory"/">
```

```
Require all denied
```

```
</Directory>
```

Tras la lectura realizada, usted ya cuenta con los conocimientos necesarios para asignar permisos adecuados en los directorios de un servidor *web*, así como para proteger las carpetas sensibles que puedan contener información crítica. Estos aspectos son fundamentales para establecer un nivel básico pero eficaz de seguridad en la configuración del servidor.

Si todo ha quedado claro hasta este punto, estamos listos para avanzar al siguiente tema: Control de acceso en el servidor Apache.

6.2 Control del acceso al servidor con el archivo `.htaccess`

```
.htaccess
```

(Acceso de hipertexto) es el nombre por defecto del archivo de configuración de directorios de Apache. Este archivo provee la habilidad para **personalizar la configuración de las directivas** definidas en el archivo de configuración principal. Las directivas de configuración necesitan estar en el contexto de

```
.htaccess
```

y el usuario debe tener los permisos apropiados.



A través de este archivo, es posible implementar mecanismos de seguridad esenciales, como restringir el acceso a determinadas carpetas, definir reglas de autenticación mediante usuario y contraseña, redirigir páginas, bloquear direcciones IP, y establecer políticas de acceso personalizadas.

A continuación, en la infografía se muestran las configuraciones generales que suelen hacer con este archivo.

[Configuraciones esenciales para controlar el acceso con el archivo .htaccess.](#)

Para reforzar sus conocimientos sobre los archivos de configuración del servidor Apache, le recomiendo revisar la siguiente guía práctica del sitio oficial del proyecto Apache: "[Tutorial del Servidor Apache HTTP: Ficheros .htaccess](#)". En esta guía encontrará una explicación detallada sobre el uso del archivo

`.htaccess`

, su propósito dentro del servidor, las directivas que se pueden aplicar y las mejores prácticas para su implementación.

Le resultó muy sencilla la lectura, ¿verdad? Ahora pasemos a realizar la siguiente actividad para afianzar su aprendizaje.



Actividad de aprendizaje recomendada

Implemente este siguiente código .htaccess en una aplicación de su servidor apache, para ello:

- Cree una carpeta llamada aplicacionweb1.
- Coloque algunos archivos como imágenes, PDF, etc.
- Ejecute la aplicación son 127.0.0.1/ aplicaiconweb1.
- Implemente el siguiente código en el .htaccess.
 - **RewriteEngine on.**
 - **RewriteRule ^\$ webroot/ [L].**
 - **RewriteRule (.*) webroot/\$1 [L].**
 - **Options -Indexes.**



- Socialice el resultado en el foro del EVA y emita comentarios.

Continuemos con algunos temas más de seguridad.



Recuerde que cada semana contamos con un espacio dedicado para resolver dudas con el docente tutor. Consulte el horario de tutoría disponible en el Entorno Virtual de Aprendizaje.



Resultado de aprendizaje 3, 5, 9 y 11:

- Explica la importancia de la interconexión entre sitios web para bases de datos subyacentes.
- Construye aplicaciones web accesibles y seguras.
- Describe los métodos de seguridad para cookies y las formas en las que pueden ser usadas para poner en peligro la privacidad del usuario.
- Describe formas de aumentar la fiabilidad de un sitio web, tales como certificados de seguridad.

Para alcanzar los resultados de aprendizaje planteados, es fundamental comprender la importancia de la interconexión entre sitios web y sus bases de datos subyacentes. También es crucial desarrollar aplicaciones web que sean accesibles y seguras, gestionando adecuadamente la seguridad de las cookies para proteger la privacidad del usuario. Además, es necesario identificar estrategias para mejorar la fiabilidad de un sitio web, como la implementación de certificados de seguridad. La seguridad web abarca la configuración del servidor para defenderlo contra ataques de servicio y la correcta puesta en marcha de aplicaciones web para garantizar su funcionamiento seguro y eficiente.

Contenidos, recursos y actividades de aprendizaje recomendadas

Recuerde revisar de manera paralela los contenidos con las actividades de aprendizaje recomendadas y actividades de aprendizaje evaluadas.



Semana 14

Unidad 6. Seguridad web

6.3 Configuración del servidor ante ataques de servicio

El servidor Apache, que se instala con **XAMPP**, contiene su propio archivo de configuración que se llama



httpd.conf

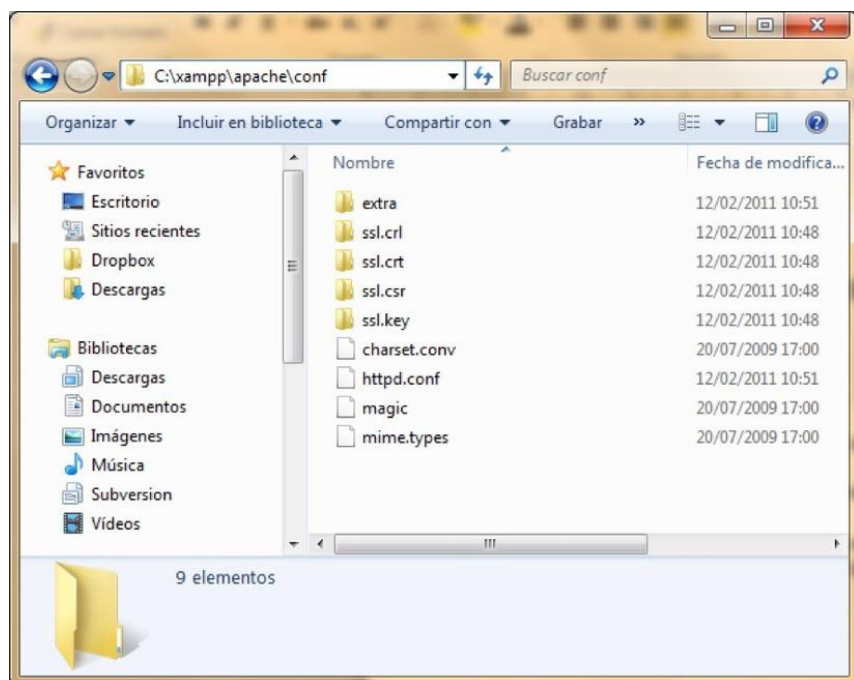
. En equipos Windows, este archivo se encuentra ubicado en

C:\xampp\apache\conf

, como se muestra en la figura 23:

Figura 23

Directorio del archivo httpd.conf



Nota. Ramírez, R., 2020.

El archivo httpd.conf está dividido en 3 partes:

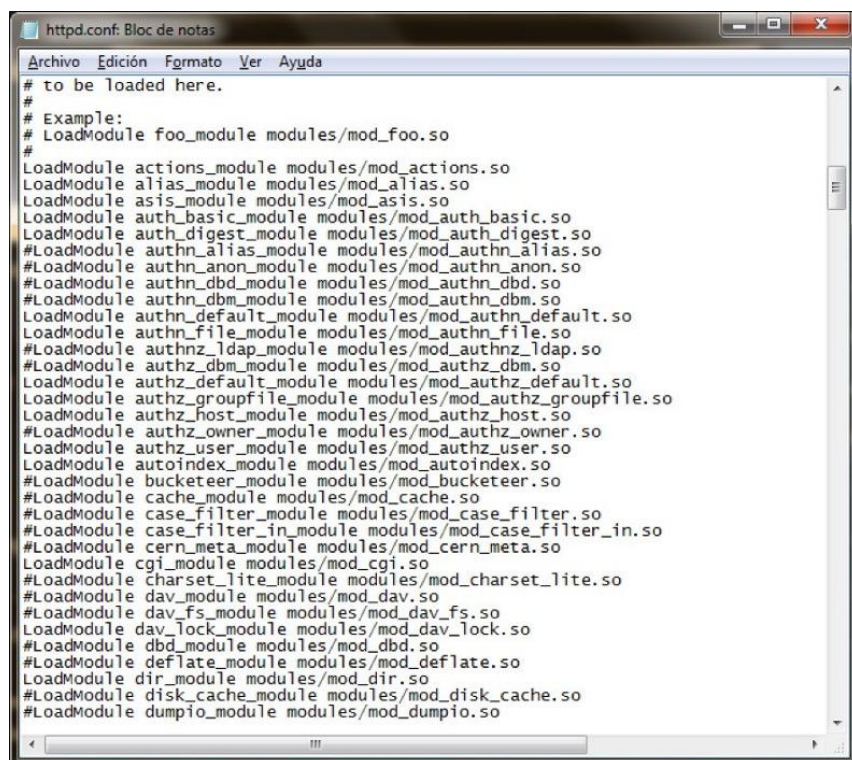
- Configuración de parámetros globales.
- Directivas de funcionamiento.
- Configuración de los *hosts* virtuales.

Por lo regular, el archivo contiene parámetros de configuración que vienen por defecto en la instalación. Pero también existen parámetros que se los pueden configurar a conveniencia del usuario.

Dentro de este archivo podemos activar o desactivar librerías tal como lo muestra la figura 24:

Figura 24

Contenido del archivo *httpd.conf*



```
# to be loaded here.
#
# Example:
# LoadModule foo_module modules/mod_foo.so
#
LoadModule actions_module modules/mod_actions.so
LoadModule alias_module modules/mod_alias.so
LoadModule asis_module modules/mod_asis.so
LoadModule auth_basic_module modules/mod_auth_basic.so
LoadModule auth_digest_module modules/mod_auth_digest.so
#LoadModule authn_alias_module modules/mod_authn_alias.so
LoadModule authn_anon_module modules/mod_authn_anon.so
#LoadModule authn_dbd_module modules/mod_authn_dbd.so
LoadModule authn_dbm_module modules/mod_authn_dbm.so
LoadModule authn_default_module modules/mod_authn_default.so
LoadModule authn_file_module modules/mod_authn_file.so
#LoadModule authnz_ldap_module modules/mod_authnz_ldap.so
LoadModule authz_dbm_module modules/mod_authz_dbm.so
LoadModule authz_default_module modules/mod_authz_default.so
LoadModule authz_groupfile_module modules/mod_authz_groupfile.so
LoadModule authz_host_module modules/mod_authz_host.so
#LoadModule authz_owner_module modules/mod_authz_owner.so
LoadModule authz_user_module modules/mod_authz_user.so
LoadModule autoindex_module modules/mod_autoindex.so
#LoadModule bucketeer_module modules/mod_bucketeer.so
#LoadModule cache_module modules/mod_cache.so
#LoadModule case_filter_module modules/mod_case_filter.so
#LoadModule case_filter_in_module modules/mod_case_filter_in.so
#LoadModule cern_meta_module modules/mod_cern_meta.so
LoadModule cgi_module modules/mod_cgi.so
#LoadModule charset_lite_module modules/mod_charset_lite.so
#LoadModule dav_module modules/mod_dav.so
#LoadModule dav_fs_module modules/mod_dav_fs.so
LoadModule dav_lock_module modules/mod_dav_lock.so
#LoadModule dbd_module modules/mod_dbd.so
#LoadModule deflate_module modules/mod_deflate.so
LoadModule dir_module modules/mod_dir.so
#LoadModule disk_cache_module modules/mod_disk_cache.so
#LoadModule dumpio_module modules/mod_dumpio.so
```

Nota. Ramírez, R., 2020.

Eliminando el signo # desconectamos la librería para luego reiniciar los servicios y se active con dicha opción.

Otros parámetros pueden ser:

- **<Directory>**: los parámetros que se encuentran dentro de esta sección, solo se aplicarán al directorio especificado y a sus subdirectorios.
- **<DirectoryMatch>**: igual que *Directory*, pero acepta en el nombre del directorio expresiones regulares.
- **<Files>**: los parámetros de configuración proporcionan.

- **<FilesMatch>**: igual que *Files*, pero acepta expresiones regulares en el nombre del fichero.
- **<Location>**: proporciona un control de acceso de los ficheros por medio de la URL.
- **<LocationMatch>**: igual que *Location*, pero acepta expresiones regulares en el nombre del fichero.

Algunas veces, las directivas de funcionamiento de las secciones anteriores se pueden cruzar, en cuyo caso tienen el siguiente orden de preferencia:

1. <Directory> y .htaccess (.htaccess prevalece frente a <Directory>)
2. <DirectoryMatch> y <Directory>
3. <Files> y <FilesMatch>
4. <Location> y <LocationMatch>

Timeout

Timeout define, en segundos, el tiempo que el servidor esperará por recibir y transmitir durante la comunicación. *Timeout* está configurado por defecto a **300 segundos**, lo cual es apropiado para la mayoría de las situaciones.

6.4 ServeRoot

La directriz **ServerRoot** especifica el directorio de nivel superior que tiene el contenido web. Por defecto, **ServerRoot** está configurado a "C:\xampp\htdocs" cuando se instala Apache vía Xampp.

6.5 ServerName

Use la directriz **ServerName** para configurar un nombre de servidor y un número de puerto (que coincida con la *directriz Listen*) para el servidor. El **ServerName** no necesita coincidir con el nombre real de la máquina. Por ejemplo, el servidor web puede ser **www.example.com**, pero el nombre del servidor es en realidad **foo.example.com**.



El valor especificado en `ServerName` debe ser un nombre del Servicio de Nombres de Dominio (*Domain Name Service*, DNS) válido que pueda ser resuelto por el sistema.

Lo siguiente es una directriz `ServerName` de ejemplo: `www.servidorweb.com:80`.



Para reforzar aún más este tema le invito a desarrollar una práctica de configuración del servidor local, active la librería para la base de datos postgres y cambie su `serverName` a `www.ejemploweb.com`. Capture la pantalla de la práctica y súbalas al EVA y emita criterios de la misma.

6.6 Puesta en marcha las *WebApps*

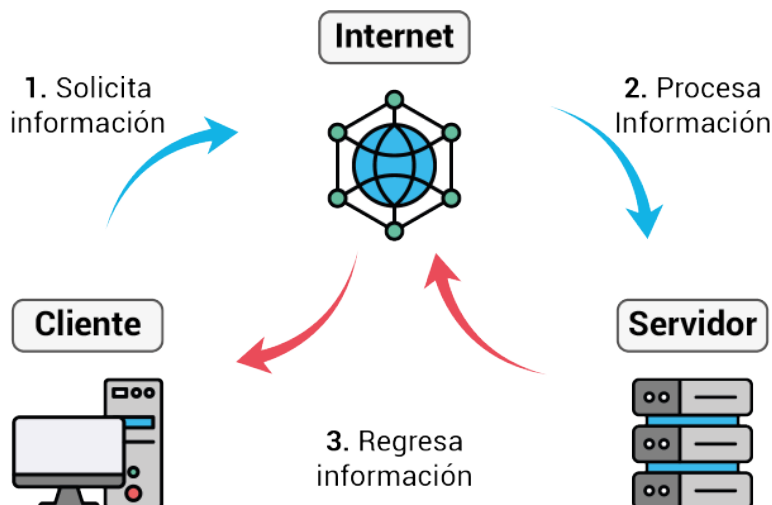
Una vez desarrollada y probada una aplicación web (*WebApp*) en el entorno local, llega el momento de su puesta en marcha o despliegue en un entorno de producción. Este proceso implica trasladar la aplicación desde el entorno de desarrollo al servidor definitivo donde estará disponible para los usuarios finales.

Es importante conocer cuál es el proceso de implementación de nuestra *WebApp*, qué se debe comprar y dónde se puede realizar, así como las ventajas y desventajas del proceso de implementación. A continuación, en la figura 25 se describe la forma de cumplir este objetivo.



Figura 25

Proceso de la petición de una WebApp con el servidor



Nota. Ramírez, R., 2024.

El *Web Hosting* es una computadora con características especiales donde se guarda toda la información de la *WebApp*, correos, bases de datos, etc.

En la figura 25 se describe el proceso que se realiza cuando el cliente solicita una aplicación web por Internet, ejemplo www.utpl.edu.ec, la computadora del usuario se conecta con otra computadora en internet para descargar la información que va a mostrar, a esta otra computadora se le llama **Servidor**, porque es la encargada de brindar la información, y a tu computadora se le llama **Cliente**, porque es quien recibe la información.

El servidor que brinda el *Hosting*, que es todo lo necesario para que una página pueda ser utilizada desde *Internet*, tiene varios servicios, como son:

- Espacio en disco para almacenar la página.
- Espacio para guardar los correos electrónicos.
- Recibir y enviar correos electrónicos.
- Guardar y administrar bases de datos.
- Guardar estadísticas sobre las visitas a la página de Internet.

Dominios

Los **dominios** son «nombres» de las páginas en *Internet*. Cuando se quiere entrar a una *WebApp*, normalmente se escribe el nombre, por ejemplo: www.utpl.edu.ec.

El nombre de dominio está compuesto por dos partes:

1. El nombre: utpl
2. TLD (Top Level Domain): .edu.ec.

La terminación o extensión puede ser **.com**, **.com.ec**, **.org**, **.ec**, **.edu**, **.edu. ec**, etc. Esta terminación ayuda a indicar el motivo o la razón de su empresa y ubicación:

- **.com** → comercial
- **.com.ec** → comercial en Ecuador.
- **.org** → organización sin fines de lucro.
- **.tv** → canal de televisión o televisora.
- **.edu** → educativo
- **.edu.ec** educativas de Ecuador.

Hay reglas para escoger un nombre de dominio:

- **Tiene que ser único:** si alguien ya registró un dominio, no se puede repetir.
- **Pueden tener los siguientes caracteres:** letras de la A a la Z (con la excepción de la ñ) o números del 1 al 9 o guion medio “-”.
- Cualquier otro carácter, como espacios en blanco, acentos, signos de puntuación, paréntesis, guion bajo “_”, etc., no pueden ser parte de un dominio.
- No pueden empezar con un guion medio.

Para poder ver la disponibilidad del dominio, le invito a ingresar al sitio web [Network Solutions](#) y buscar si está disponible el dominio deseado.



Luego de realizar la lectura sugerida, ¿le parece aplicable? Reflexione sobre lo que tiene en su entorno y aplique lo indicado. Si le quedaron dudas vuelva a leer detenidamente el tema, formule preguntas y consúltelas con su tutor.

Recuerde que la puesta en marcha de una *WebApp* incluye varias tareas críticas, como:

- Configuración del servidor web (por ejemplo, Apache o Nginx).
- Carga de archivos al servidor mediante FTP, Git u otras herramientas.
- Ajustes en la base de datos y archivos de configuración.
- Establecimiento de medidas de seguridad como: HTTPS, control de acceso y protección de datos.
- Pruebas finales para verificar el correcto funcionamiento de la aplicación.



Realizar esta etapa de forma organizada y segura es esencial para garantizar que la aplicación funcione correctamente en el entorno real, ofrezca una buena experiencia al usuario y esté protegida frente a posibles vulnerabilidades.



Actividades de aprendizaje recomendadas

Es momento de aplicar sus conocimientos a través de las actividades que se han planteado a continuación:

1. Con base en el conocimiento adquirido, conste las siguientes preguntas:

- Liste qué es un servidor y un cliente.
- Qué es el *hosting*.
- Qué es un dominio.
- Cuáles son las terminologías que pueden tener.
- Con ayuda del foro del EVA comente qué empresa vende dominios con la terminología .ec.

Nota: por favor, copie las preguntas y dé respuestas en un documento Word o cuaderno.



2. Realice un presupuesto para la implementación del cms instalado anteriormente en la *Internet*, que abarque lo que es el dominio .ec y el *hosting*. Puede comentarlo en el foro del EVA.
3. Estamos avanzando en los contenidos, por ello, lo invito a medir sus conocimientos, desarrollando la autoevaluación. Esta le ayudará a autoevaluarse y ver su estado en la materia.

¡Suerte en la autoevaluación! Siga adelante.



Autoevaluación 6

Seleccione la opción correcta:

1. Dentro de la seguridad, se encuentra el control de acceso de información. Localice el tipo de restricciones:
 - a. Limitación de acceso en función de direcciones IP o dominio.
 - b. Solo los usuarios de un dominio u organización tendrán acceso a la información.
 - c. Limitación de acceso por sistema operativo.
2. El significado del archivo .htaccess en español es:
 - a. Acceso de protocolo de hipertexto.
 - b. Acceso de hipertexto.
 - c. Acceso de hipermedia.
3. Para la restricción de una carpeta en el archivo .htaccess es:
 - a. Allow from all.
 - b. Deny from all.
 - c. Allow dem /carpeta.
4. Para permitir el acceso desde un rango de IP es:
 - a. Deny from 10.2.2.2.
 - b. Allow from 192.167.2.3/34.-.



c. Deny from all.

5. El archivo de configuración del servidor Apache se denomina:

- a. Htpd.conf.
- b. Httpd.ini.
- c. Httpd.conf.

6. De la siguiente lista, marque las partes en que se divide el archivo httpd:

- a. Configuración de parámetros internos.
- b. Configuración de parámetros globales.
- c. Directivas de funcionamiento.
- d. Restricciones de dominios.

7. Localice el proceso de una petición web de la siguiente lista:

- a. *Servidor* – cliente- *Internet* – servidor- cliente.
- b. Cliente – servidor – *Internet* – servidor - cliente.
- c. Petición del cliente – *Internet* – servidor – *Internet* – cliente.

8. () Coloque una V si es verdadero o una F si es falso. Lo siguiente: el *web hosting* es el alojamiento de mi aplicación web en otra máquina local.

9. () Coloque una V si es verdadero o una F si es falso a lo siguiente: la terminología .ec significa que son de Ecuador.

10. () Coloque una V si es verdadero o una F si es falso, a lo siguiente: el dominio son los nombres de mi página web.

[Ir al solucionario](#)

Si no logró un buen resultado en la autoevaluación, no se preocupe, le recomiendo leer nuevamente el/los temas confusos y reforzar sus conocimientos. Y si aún tiene inquietudes, no dude en preguntar al tutor.



Resultado de aprendizaje 3, 5, 6, 7, 8, 9, 10 y 11:

- Explica la importancia de la interconexión entre sitios web para bases de datos subyacentes.
- Construye aplicaciones web accesibles y seguras.
- Explica la diferencia entre la programación orientada a eventos y programación de línea de comandos.
- Diseña, codifica, prueba y depura programas simples basados en eventos que respondan a eventos de usuario.
- Desarrolla código que responde a las condiciones de excepción planteadas durante la ejecución.
- Describe los métodos de seguridad para cookies y las formas en las que pueden ser usadas para poner en peligro la privacidad del usuario.
- Identifica los sitios web que están utilizando gráficos de páginas web como iconos web.
- Describe formas de aumentar la fiabilidad de un sitio web, tales como certificados de seguridad.

Contenidos, recursos y actividades de aprendizaje recomendadas



Semana 15

Actividades finales del bimestre

Durante esta penúltima semana del curso se realizará una revisión general de los temas abordados anteriormente, con el objetivo de reforzar los conocimientos adquiridos antes de la evaluación presencial.

Al estar dedicada a la consolidación del aprendizaje, le animo a repasar cuidadosamente cada uno de los contenidos revisados. Si surge alguna duda o inquietud, no dude en contactar al tutor para recibir orientación o aclaraciones oportunas.



Asimismo, es importante que siga con atención todas las instrucciones proporcionadas de cada actividad, ya que esto le permitirá cumplir de manera efectiva con las tareas asignadas y prepararse adecuadamente para la evaluación final.



Resultado de aprendizaje 3, 5, 6, 7, 8, 9, 10 y 11:

- Explica la importancia de la interconexión entre sitios web para bases de datos subyacentes.
- Construye aplicaciones web accesibles y seguras.
- Explica la diferencia entre la programación orientada a eventos y programación de línea de comandos.
- Diseña, codifica, prueba y depura programas simples basados en eventos que respondan a eventos de usuario.
- Desarrolla código que responde a las condiciones de excepción planteadas durante la ejecución.
- Describe los métodos de seguridad para cookies y las formas en las que pueden ser usadas para poner en peligro la privacidad del usuario.
- Identifica los sitios web que están utilizando gráficos de páginas web como iconos web.
- Describe formas de aumentar la fiabilidad de un sitio web, tales como certificados de seguridad.

Contenidos, recursos y actividades de aprendizaje recomendadas



Semana 16

Actividades finales del bimestre

Durante esta última semana del curso, el enfoque estará centrado en el cierre del proceso formativo y la realización de la evaluación presencial. Es el momento de reforzar los aprendizajes clave, consolidar lo revisado y aplicar los conocimientos adquiridos a lo largo de la asignatura.

Le recomiendo consultar nuevamente los materiales educativos más relevantes, revisar las actividades desarrolladas, y repasar en aquellos conceptos que hayan requerido mayor atención. Si aún tiene dudas pendientes, esta es la oportunidad final para aclararlas con el tutor antes de la evaluación.



Finalmente, asegúrese de cumplir con todas las actividades previstas, atender los lineamientos de la evaluación, y mantener una actitud reflexiva sobre lo aprendido. Este cierre le permitirá evaluar su propio progreso y prepararse con seguridad para concluir satisfactoriamente el estudio de la asignatura.





4. Solucionario

Autoevaluación 1

Pregunta	Respuesta	Retroalimentación
1	b	Porque la accesibilidad web pretende llegar a todo tipo de personas sin discriminación de ningún tipo.
2	a, c	La accesibilidad garantiza que servicios educativos y canales digitales lleguen a todas las personas, favoreciendo la igualdad de oportunidades en la enseñanza y la comunicación.
3	b	El acrónimo WAI proviene del inglés <i>Web Accessibility Initiative</i> ; su traducción habitual al español es Iniciativa para la Accesibilidad en la Web.
4	a, b, d	Aplicar etiquetado semántico en el HTML, añadir resúmenes o descripciones para tablas y ajustar el tamaño de las imágenes para mantener coherencia con el contenido.
5	a, c	Aplicar etiquetado semántico en el HTML, añadir resúmenes o descripciones para tablas y ajustar el tamaño de las imágenes para mantener coherencia con el contenido.
6	a, c	HTML5 incorpora etiquetas semánticas que identifican secciones (títulos, módulos, contenidos), mejorando la estructura del documento y, por tanto, la accesibilidad.
7	c	La etiqueta <nav> agrupa los enlaces de navegación principales de la página, facilitando el acceso rápido a las secciones relevantes.
8	b	Se emplea para delimitar bloques temáticos de contenido en la página; la etiqueta facilita la organización y navegación del documento.
9	c	HTML5 ofrece una etiqueta específica para ubicar la información situada al final de la página (por ejemplo, notas, créditos o contactos).
10	b	La etiqueta <main> identifica el bloque central del documento —el contenido principal—, lo que ayuda a motores y lectores a priorizar la información más relevante.



[Ir a la autoevaluación](#)



Autoevaluación 2

Pregunta	Respuesta	Retroalimentación
1	b	Porque intervienen diferentes áreas como diseño, desarrollo, control de calidad, etc.
2	b, c	Entre los programas que tiene XAMPP está PHP como lenguaje de programación y Apache como servidor web.
3	c	XAMPP incluye un gestor de bases de datos para desarrollo local (MySQL; en distribuciones recientes suele aparecer como MariaDB).
4	b	La etiqueta que el compilador de PHP entiende para leer su código es <code><?php ... ?></code>
5	a	Siempre se utiliza el signo de dólar para la creación de variables en PHP, ejemplo <code>\$var=0;</code>
6	c	Una constante es un identificador asociado a un valor inmutable que permanece igual durante toda la ejecución del programa.
7	c, d	Entre los operadores de comparación están <code>==</code> (igualdad) y <code>!=</code> (desigualdad); <code>=</code> corresponde a la asignación.
8	b	Dentro de los operadores del incremento se encuentran los signos <code>++</code> que indican que la variable aumentará de 1 en 1.
9	b	Las condiciones son representadas en programación con el IF y la estructura correcta de esta es como lo indica la opción b.
10	b	Una función son bloques de códigos que son ejecutadas siempre que sean llamadas o invocadas; estas pueden enviar y retornar parámetros.

[Ir a la autoevaluación](#)



Autoevaluación 3

Pregunta	Respuesta	Retroalimentación
1	c	Porque el editor <i>web</i> para la gestión de bases de datos que se instala con XAMPP es phpMyAdmin.
2	b	La forma correcta para crear una base de datos a nivel de consola es <code>create database nombre;</code> ; esta debe ser ejecutada en el editor o en la consola para su ejecución.
3	c	Una vez creada la base de datos y podemos visualizar las bases de datos que se tienen en el servidor, la sentencia correcta es <code>show databases.</code>
4	b	En base de datos, las tablas deben tener una llave primaria para poder identificar sus registros, para ello se debe colocar al campo que cumpla como valor único e irrepetible con la propiedad de <code>primary key.</code>
5	a, b	Existen diferentes formas para construir una sentencia SQL para insertar datos en la base de datos, entre ellas pueden ser las opciones a y b, ya que no poseen errores de estructura.
6	a, c	La forma correcta de obtener los datos de las tablas de una base de datos son las opciones a y c, ya que no poseen problemas en la estructura de la sentencia SQL.
7	b	La forma correcta para eliminar registros de una tabla es con la opción b, ya que posee una cláusula <code>where</code> que identifica qué registro borrar; caso contrario, se eliminarían todos los valores de la tabla.
8	c	Dentro de la gestión de los datos en una base de datos está la opción <code>update</code> la cual permite la actualización de los registros, en la opción c, se muestra la estructura correcta mencionando que se debe tener una cláusula <code>where</code> para identificar el registro a modificar.
9	c	Dentro de las funciones reservadas en PHP para la selección de la base de datos está <code>mysql_select_db</code> que nos permite escoger la base de datos a utilizar en nuestra WebApp.
10	b	<code>mysql_query</code> es una función reservada dentro de PHP para la ejecución de una sentencia SQL.

[Ir a la autoevaluación](#)



Autoevaluación 4

Pregunta	Respuesta	Retroalimentación
1	a	La POO tiene como objetivo la realización de código mediante el uso de clases y funciones para ser utilizados en cualquier parte de la aplicación.
2	c	Se trata de un ente abstracto usado en programación que permite separar los diferentes componentes de un programa, simplificando así su elaboración, depuración y posteriores mejoras. Los objetos se integran, a diferencia de los métodos procedurales, tanto los procedimientos como las variables y datos referentes al objeto.
3	b	Una clase es la descripción de un conjunto de objetos similares; consta de métodos y de datos que resumen las características comunes de dicho conjunto.
4	a, c	Dentro de las propiedades de las clases para su invocación pueden ser públicas o privadas.
5	b	Para acceder a las propiedades internas, como los métodos o funciones de la clase se utiliza <code>\$this-></code> con ello podemos reutilizar propiedades internas de la clase.
6	a	La forma correcta para crear una instancia de la clase MySQL es la opción a, ya que esta no tiene ningún problema de estructura.
7	b	La manera correcta de invocar un método sobre una instancia es <code>\$miconexion->consulta()</code> ; Dentro de la propia clase se usaría <code>\$this->consulta()</code> .
8	c	La forma correcta de reutilizar un método de la clase es con la propiedad <code>\$this-></code> más el método o la función a utilizar.
9	a	Una excepción se refiere a un evento que sucede mientras un programa está en ejecución y que interrumpe el flujo normal del programa.
10	a, c, d	En PHP, los métodos más comunes para manejar excepciones incluyen <code>getMessage()</code> , que devuelve el mensaje de la excepción; <code>getFile()</code> , que proporciona el nombre del archivo donde ocurrió el error, y <code>getLine()</code> , que indica la línea del archivo en la que se produjo la excepción.

[Ir a la autoevaluación](#)



Autoevaluación 5

Pregunta	Respuesta	Retroalimentación
1	c	Un Cross Site son ataques que sufren las aplicaciones web al momento de presentar los valores solicitados al servidor.
2	b	Los ataques por lo regular suelen ocurrir cuando existe código malicioso en el servidor web o es incrustado por vías como los formularios o el navegador.
3	a, c	Los objetivos comunes son robar <i>cookies</i> o sesiones de usuarios y modificar el contenido de la aplicación para ejecutar acciones no autorizadas.
4	b, c	Dentro de las vulnerabilidades a las aplicaciones web están las opciones b y c que son ataques de terceros y están dentro de la XSS para la modificación del sitio web.
5	a	El ataque persistente XSS ejecuta código malicioso en el proceso de validación del usuario para poder suplantar la identidad y poder tener acceso a la información restringida.
6	c	Existen muchas reglas a tomar en cuenta en el momento de aplicar seguridad, pero una importante es la opción c, ya que se debe estar preparado ante cualquier dato que ingrese el usuario final.
7	a, c	Dentro de las estrategias para ver la información que ingresa el usuario y la validación de los datos es la opción a y c, ya que garantiza un nivel de seguridad mayor.
8	a	La validación de los datos es la utilización de estrategias como las del navegador en el lado del cliente, como el análisis del dato en el lado del <i>back-end</i> para garantizar la integridad de la información que llega al servidor.
9	c	En el lenguaje PHP nos ayuda a garantizar la integridad de los datos que ingresan del lado del cliente mediante la función <code>strip_tags()</code> .
10	b	Dentro de las vulnerabilidades de una aplicación web se encuentran las que el usuario ejecuta de una forma no intencionada, para ello se debe restringir al máximo las opciones al usuario.

[Ir a la autoevaluación](#)



Autoevaluación 6

Pregunta	Respuesta	Retroalimentación
1	a	Otra forma de brindar seguridad a la aplicación web es garantizar seguridad al servidor mediante la restricción a los archivos de la app.
2	b	El significado del archivo .htaccess es acceso de hipertexto que es un archivo de configuración del servidor.
3	b	Dentro de los parámetros que se pueden configurar en el archivo .htaccess está la restricción de los directorios que se configura con la opción b.
4	b	Dentro de los parámetros que se pueden configurar en el archivo .htaccess es la restricción de las direcciones IPS que se configura con la opción b.
5	a	El archivo general del servidor Apache se denomina httpd.conf. Este archivo es crucial para la configuración del servidor, permitiendo ajustar parámetros y directivas que controlan el funcionamiento de Apache.
6	b, c	El archivo httpd.conf se compone de dos partes que son la configuración de parámetros globales del servidor y las directivas de funcionamiento. Ambas partes son esenciales para una configuración adecuada y segura del servidor Apache, permitiendo una administración flexible y eficiente del servidor web.
7	c	El proceso de petición desde el cliente hasta el servidor sigue el patrón cliente – Internet – servidor – Internet – cliente, con el fin de obtener los datos solicitados por el cliente y consumidos en el servidor.
8	f	Es falsa, ya que el <i>hosting</i> debe ser público para el acceso de todo el público de la app.
9	v	Todos los dominios que terminan en .ec, ejemplo: utpl.edu.ec, indican que son de procedencia de Ecuador.
10	v	Es correcto porque es el nombre con el cual el usuario podrá reconocer y acceder a la aplicación web para poder consumir la información que brinda la app.

[Ir a la autoevaluación](#)





5. Glosario

Código fuente: conjunto de instrucciones que forman un programa o subprograma informático.

CSS: hojas de estilo en cascada (*Cascading Style Sheets*) es el lenguaje utilizado para describir la presentación de documentos HTML o XML.

PHP: lenguaje de desarrollo para la construcción de aplicaciones *web*.

Hipervínculo: es un enlace de una página *web* o un archivo a otra página *web* u otro archivo.

HTML: Lenguaje de Marcado para Hipertextos (*HyperText Markup Language*) es el elemento de construcción más básico de una página *web*.

HTML5: es la última versión de HTML, con nuevos elementos, atributos y comportamientos.

HTTPS: Protocolo de Transferencia de Hipertexto (en inglés: *Hypertext Transfer Protocol* o HTTP) es el protocolo de seguridad.

JS: es un archivo de texto plano que contiene *scripts* de JavaScript, y que puede, por tanto, ser modificado con cualquier editor de textos.

Link: apuntador de hipertexto que tiene como misión llevarnos de una información a otra, de una página a otra, o de un servidor a otro.

Navegador web: es un *software*, aplicación o programa que permite el acceso a la *web*, que se instala en los clientes.

CMS: Sistemas Manejadores de Archivos que ayudan a la creación de aplicaciones *web*.

Regla: es cada uno de los estilos que componen una hoja de estilos CSS.



Selector: indica el elemento o elementos HTML a los que se aplica la regla CSS.

Servidor web: es un programa informático que procesa una aplicación del lado del servidor, realizando conexiones bidireccionales o unidireccionales y síncronas o asíncronas con el cliente.

URL: es una sigla del idioma inglés correspondiente a *Uniform Resource Locator* (Localizador Uniforme de Recursos); es una secuencia de caracteres que sigue un estándar y que permite denominar recursos dentro de la *web*.

Valor: establece el nuevo valor de la característica modificada en el elemento.

W3C: el Consorcio *World Wide Web* es una comunidad internacional donde las organizaciones miembros [ing, personal [inglés] a tiempo completo y el público en general trabajan conjuntamente para desarrollar estándares *web* [inglés].





6. Referencias bibliográficas

Básica

Pavón, J. Llanera, E. (2016). Creación de un sitio web con PHP y mysql. Madrid. España: RA-MA Editorial. ISBN: 978-84-9964- 567-4

Este libro posee un importante contenido ilustrativo e intuitivo que nos ayudará a cumplir nuestros objetivos planteados a lo largo de nuestra materia, como es el desarrollo de aplicaciones en PHP, así como también la gestión de los datos en una base de datos relacional como lo es MySQL.

Ramírez, R. (2017): Guía didáctica de Desarrollo Web, Loja-Ecuador, Editorial Universidad Técnica Particular de Loja.

En la guía didáctica encontrará los lineamientos necesarios para que se pueda guiar, además encontrará explicaciones adicionales sobre algunos temas que le permitirán entenderlos mejor.

Complementaria

PRESSMAN R.(2010), Ingeniería de Software, un Enfoque Práctico.

México, 7.^a edición. McGraw-Hill. ISBN 970-10-5473-3. Texto de estudio de la asignatura de Ingeniería de software, en esta obra nos acerca y profundiza a los contenidos de Ingeniería de software. Este texto es de lectura y le servirá de mucho durante el transcurso de este período de estudios.

SOFTWARE ENGINEERING FOR INTERNET APPLICACIONES. Eve Andersson, Philip Greenspun, and Andrew Grumet. 6.^a edición. [en línea] Disponible en: <http://ndl.ethernet.edu.et/bitstream/123456789/4863/1/18pdf.pdf>, [consultado a: 11/12/2017].



Recurso digital donde encontrará los contenidos relacionados con el proceso de desarrollo de aplicaciones web.

SOFTWARE ENGINEERING FOR INTERNET APPLICACIONES. Eve Andersson, Philip Greenspun, and Andrew Grumet. 6.^a edición. [en línea] Disponible en: <http://www.kiarash.net/Teaching/22-24-118/Resources/Software%20Engineering%20for%20Internet%20Applications.pdf> [consultado a: 2 de junio del 2017].

Recurso digital donde encontrará los contenidos de la metodología de desarrollo de aplicaciones web, fases o procesos a seguir para el desarrollo de las apps.

Accesibilidad Web [en línea] Disponible en: <https://blog.hubspot.es/website/que-es-accesibilidad-web> [consultado a: 12 de diciembre del 2017].

Recurso digital donde elementos básicos para el desarrollo web accesible, con estándares.

Codigofacilito (2017). Curso de Introducción a HTML 5. Disponible en: <https://codigofacilito.com/cursos/HTML5>

En este sitio web hay disponibles varios videos cortos pero explicativos sobre los nuevos elementos HTML 5, con ejemplos en código y explicaciones detalladas.

Gómez, M. R. (2013). HTML5, CSS3 y JavaScript. Anaya Multimedia-Anaya Interactiva.

En este libro se explican a detalle los elementos y estructura del Lenguaje de etiquetado HTML5 con sus componentes en CSS3 y JavaScript.



Darío Andrés Silva, Construyendo aplicaciones web con una metodología de diseño orientado a objetos. [en línea] Disponible en: <http://revistas.unab.edu.co/index.php?journal=rcc&page=article&op=viewArticle&path%5B%5D=1116> [consultado a: 12 de diciembre del 2017].

Recurso digital donde se explica la metodología de diseño orientado a objetos orientado a aplicaciones web.

Apache, [en línea] Disponible en: https://httpd.apache.org/docs/trunk/es/misc/security_tips.html [consultado a: 12 de diciembre del 2017].

Recurso digital donde se explica lo que se debe tener en cuenta para contar con una buena seguridad en nuestro servidor y algunos consejos prácticos en la configuración del servidor Apache.

Base de Datos: Guía de base de datos. Recuperado de <https://www.powerdata.es/integracion-de-datos>

Guía Digital: Usabilidad web. Recuperado de <https://rockcontent.com/es/blog/usabilidad/>

Guía y configuración del servidor web Apache. Recuperado de https://httpd.apache.org/docs/trunk/es/misc/security_tips.html

Guía de accesibilidad web. Recuperado de <https://userway.org/es/blog/accesibilidad-web/>

Validadores de accesibilidad web. Recuperado de https://www.usableyaccessible.com/recurso_misvalidadores.php

Desarrollo de aplicaciones con en HTML5, maquetado semántico. Recuperado de http://www.aulaclie.es/articulos/html5-semantica_1.html



Seguridad Web: PHP con Mysql. Recuperado de <https://diego.com.es/seguridad-web-en-php>

Desarrollo de aplicaciones web. Recuperado de https://rua.ua.es/dspace/bitstream/10045/90894/1/daw-18_19-prac8-PHPbd1.pdf

PHP con base de datos Mysql. Recuperado de https://www.w3schools.com/php/php_mysql_intro.asp

MDN Web Docs. (s. f.). Códigos de estado de respuesta HTTP. En Mozilla Developer Network. Recuperado el 24 de junio de 2025, de <https://developer.mozilla.org/es/docs/Web/HTTP/Reference/Status>

W3C – Web Accessibility Initiative. (s.f.). Web Accessibility Initiative (WAI). Recuperado el 22 de junio de 2025, de <https://www.w3.org/WAI/>

Ganzábal García, X. (2024). Aplicaciones técnicas de usabilidad y accesibilidad en el entorno cliente. Ediciones Paraninfo, S.A.

González, J. (2015). Accesibilidad web: Diseño y evaluación de sitios web accesibles. Ediciones Paraninfo.

Apache Friends. (s.f.). XAMPP: Instalador Apache fácil de usar. Recuperado el 23 de junio de 2025, de <https://www.apachefriends.org/es/index.html>

Microsoft. (s.f.). Visual Studio Code. Recuperado el 23 de junio de 2025, de <https://code.visualstudio.com/>

Sublime HQ. (s.f.). Sublime Text: A sophisticated text editor for code, markup and prose. Recuperado el 23 de junio de 2025, de <https://www.sublimetext.com/>



Hostinger. (s.f.). ¿Qué es php.ini y cómo configurarlo correctamente?
Recuperado el 23 de junio de 2025, de <https://www.hostinger.com/es/tutoriales/que-es-php-ini>

MDN Web Docs. (s.f.). Introducción a los eventos. Recuperado de https://developer.mozilla.org/es/docs/Learn_web_development/Core/Scripting/Events

Strike. (2025, 25 de abril). OWASP Top 10 2025: Comprendiendo las vulnerabilidades comunes de aplicaciones web y los riesgos de seguridad. <https://strike.sh/es/blog/owasp-top-10-2025-comprendiendo-las-vulnerabilidades-comunes-de-aplicaciones-web-y-los-riesgos-de-seguridad>

Kallin, J., & Lobo Valbuena, I. (2013). Excess XSS: A comprehensive tutorial on cross-site scripting. Recuperado el 30 de junio de 2025, de <https://excess-xss.com/excess-xss.com+5excess-xss.com+5github.com+5>

Departamento de Ciencias de la Computación, Universidad de Chile. (s.f.). Cross-Site Scripting. Taller de Hacking Competitivo. Recuperado el 30 de junio de 2025, de <https://tallerdehacking.dcc.uchile.cl/docs/apps-web/xss/tallerdehacking.dcc.uchile.cl+2tallerdehacking.dcc.uchile.cl+2tallerdehacking.dcc.uch>

DesarrolloWeb. (s.f.). Validación de un formulario con Javascript. Recuperado el 30 de junio de 2025, de <https://desarrolloweb.com/articulos/1767.php>

