



UES

Universidad Estatal de Sonora
La Fuerza del Saber Estimulará mi Espíritu

MANUAL DE PRÁCTICAS DE LABORATORIO

Programación Web

Laboratorio de Programación

Programa Académico
Plan de Estudios
Fecha de elaboración
Versión del Documento

Ingeniero en Software
2021
30/06/2025
1.0



Dra. Martha Patricia Patiño Fierro
Rectora

Mtra. Ana Lisette Valenzuela Molina
**Encargada del Despacho de la Secretaría
General Académica**

Mtro. José Antonio Romero Montaña
Secretario General Administrativo

Lic. Jorge Omar Herrera Gutiérrez
**Encargado de Despacho de Secretario
General de Planeación**

Tabla de contenido

INTRODUCCIÓN.....	4
IDENTIFICACIÓN	6
<i>Carga Horaria de la asignatura</i>	<i>6</i>
<i>Consignación del Documento</i>	<i>6</i>
MATRIZ DE CORRESPONDENCIA	7
NORMAS DE SEGURIDAD Y BUENAS PRÁCTICAS	8
<i>Reglamento general del laboratorio</i>	<i>8</i>
<i>Uso adecuado del equipo y materiales.....</i>	<i>9</i>
<i>Procedimientos en caso de emergencia</i>	<i>9</i>
RELACIÓN DE PRÁCTICAS DE LABORATORIO POR ELEMENTO DE COMPETENCIA..	10
PRÁCTICAS.....	14
FUENTES DE INFORMACIÓN	55
NORMAS Y TÉCNICAS APLICABLES	56

INTRODUCCIÓN

Como parte de las herramientas esenciales para la formación académica de los estudiantes de la Universidad Estadal de Sonora, se definen manuales de práctica de laboratorio como elemento en el cual se define la estructura normativa de cada práctica y/o laboratorio, además de representar una guía para la aplicación práctica del conocimiento y el desarrollo de las competencias clave en su área de estudio. Su diseño se encuentra alineado con el modelo educativo institucional, el cual privilegia el aprendizaje basado en competencias, el aprendizaje activo y la conexión con escenarios reales.

Con el propósito de fortalecer la autonomía de los estudiantes, su pensamiento crítico y sus habilidades para la resolución de problemas, las prácticas de laboratorio integran estrategias didácticas como el aprendizaje basado en proyectos, el trabajo colaborativo, la experimentación guiada y el uso de tecnologías educativas. De esta manera, se promueve un proceso de enseñanza-aprendizaje dinámico, en el que los estudiantes no solo adquieren conocimientos teóricos, sino que también desarrollan habilidades prácticas y reflexivas para su desempeño profesional.

El propósito de este manual es guiar y estandarizar la realización de prácticas de laboratorio en el curso de Programación Web, asegurando que los estudiantes adquieran y apliquen los conocimientos y habilidades necesarias para desarrollar aplicaciones web dinámicas del lado del servidor utilizando PHP y MySQL, bajo estándares de calidad y seguridad.

El manual es fundamental para garantizar que las actividades prácticas se realicen de manera segura, ordenada y eficiente. Vincular la teoría con la práctica, permitiendo a los estudiantes aplicar los conceptos aprendidos en situaciones reales o simuladas además de cumplir con los estándares académicos y profesionales requeridos en el perfil de egreso, fortaleciendo la formación integral de los futuros ingenieros en software.

Competencias a Desarrollar

Competencias Blandas:

- Comunicación efectiva: Presentación clara de reportes y resultados.
- Trabajo en equipo: Colaboración en actividades grupales y solución de problemas.
- Responsabilidad y organización: Cumplimiento de normas, tiempos de entrega y cuidado del equipo.
- Uso de tecnologías digitales: Manejo de plataformas educativas y herramientas digitales para la elaboración de mapas mentales, cuadros sinópticos, etc.
- Aprendizaje autónomo: Investigación y autoformación continua.

Competencias Disciplinarias:

- Fundamentos de PHP: Identificación y aplicación de los conceptos básicos del lenguaje.
- Desarrollo de aplicaciones web: Programación de sitios web dinámicos con PHP y MySQL.
- Manejo de estructuras de control, funciones, sesiones y cookies en PHP.
- Instalación y configuración de entornos de desarrollo web.
- Aplicación de estándares de desarrollo y seguridad informática.

Competencias Profesionales

- Desarrollo de software para agilizar procesos y toma de decisiones en organizaciones, conforme a estándares nacionales e internacionales.
- Soporte y asistencia técnica: Resolución de problemas en sistemas de cómputo y optimización de recursos.
- Implementación y administración de sistemas informáticos y bases de datos, garantizando la integridad y seguridad de la información.
- Aplicación de soluciones innovadoras para la automatización de procesos en escenarios reales o simulados, alineadas con los principios y políticas organizacionales.
- Esta estructura permite que el manual sea una herramienta integral tanto para el aprendizaje autónomo como para el trabajo colaborativo, asegurando la formación de egresados competentes, responsables y alineados con las demandas del sector profesional.

IDENTIFICACIÓN

Nombre de la Asignatura		Programación Web	
Clave	061CE042	Créditos	6
Asignaturas Antecedentes	061CP018	Plan de Estudios	2021

Área de Competencia	Competencia del curso
Implementar soluciones e innovaciones tecnológicas, con el fin de contribuir a una planeación responsable de los recursos humanos, tecnológicos y financieros, con base en las necesidades, la problemática analizada y los estándares de calidad establecidos por la organización.	Desarrollar un sitio web del lado del servidor de contenido dinámico con la finalidad de administrar eficientemente la información de las organizaciones, bajo los estándares de desarrollo web con un enfoque de calidad, trabajo en equipo y responsabilidad.

Carga Horaria de la asignatura

Horas Supervisadas			Horas Independientes	Total de Horas
Aula	Laboratorio	Plataforma		
3	1	1	2	7

Consignación del Documento

Unidad Académica	Unidad Académica Navojoa
Fecha de elaboración	30/06/2025
Responsables del diseño	Mtra. Vianey Elena Valenzuela Gerardo
Validación	
Recepción	Coordinación de Procesos Educativos

MATRIZ DE CORRESPONDENCIA

Señalar la relación de cada práctica con las competencias del perfil de egreso

PRÁCTICA	PERFIL DE EGRESO
Mapa mental de fundamentos de PHP	Desarrollar software para agilizar procesos y toma de decisiones. Aplicar soluciones e innovaciones tecnológicas.
Reporte de instalación y configuración de PHP	Desarrollar soporte y asistencia técnica. Optimizar y usar responsablemente los recursos tecnológicos.
Reporte de código básico de estructura PHP	Desarrollar software bajo estándares de calidad nacional e internacional.
Práctica de sintaxis, etiquetas y comentarios	Desarrollar software de calidad, fácil mantenimiento y comprensión.
Investigación de tipos de datos en PHP	Desarrollar software robusto y seguro, adecuado a requerimientos organizacionales.
Mapa conceptual de variables y constantes	Aplicar buenas prácticas en desarrollo de software.
Cuadro sinóptico de operadores en PHP	Capacidad de análisis y resolución de problemas en sistemas de software.
Tabla descriptiva de estructuras de control	Desarrollo de software eficiente y adaptado a escenarios reales.
Investigación sobre funciones en PHP	Desarrollo de software modular y reutilizable, aplicando innovación tecnológica.
Ejercicios de funciones para manipular cadenas	Automatización de procesos y gestión eficiente de información.
Tabla sobre funciones de fecha y hora en PHP	Desarrollo de sistemas que requieren manejo de fechas y eventos para la toma de decisiones.
Reporte sobre funciones para administrar sesiones	Garantizar la seguridad de la información y la integridad de los datos en sistemas informáticos.
Página web con funciones para administrar inicio y fin de sesión	Desarrollo de sistemas seguros y confiables, alineados a políticas organizacionales y estándares de calidad.
Lectura crítica sobre cookies en PHP	Toma de decisiones informada sobre la seguridad y privacidad en el desarrollo de software para organizaciones.
Desarrollo de un sistema de gestión básico con PHP y MySQL	Desarrollar software para agilizar procesos y toma de decisiones.
Implementación de formularios para captura y consulta de datos	Desarrollar software bajo estándares de calidad. Desarrollar soporte y asistencia técnica.
Generación de reportes dinámicos a partir de la base de datos	Desarrollar software bajo estándares de calidad. Desarrollar soporte y asistencia técnica.
Integración de funciones de seguridad y control de acceso (sesiones y roles de usuario)	Garantizar la seguridad de la información y la integridad de los datos en sistemas informáticos.

NORMAS DE SEGURIDAD Y BUENAS PRÁCTICAS

Reglamento general del laboratorio

Acceso a Laboratorio de Programación

- El acceso a las salas de trabajo será sólo a las horas que no haya clases.
- Deberá entrar sólo con sus documentos de trabajo, el resto de sus cosas deberá dejarlo a la entrada del laboratorio de cómputo.
- No deberá entrar a las salas de trabajo con alimentos, bebidas o fumando.
- Queda estrictamente prohibido introducir animales.

Permanencia

- Antes de trabajar se debe revisar el equipo de cómputo donde fue designado y reportar alguna anomalía al encargado de la sala; si posteriormente se detecta alguna falla no reportada, se le responsabilizará sobre ésta.
- No deberá mover el equipo de cómputo ni mobiliario de su lugar bajo ningún motivo; verifique en su apartado el equipo que va a necesitar para su trabajo.
- El usuario que dañe el equipo de cómputo, sus instalaciones o cambie su configuración del software, incluso protectores de pantalla o colores de ventanas quedará responsabilizado de pagar su costo de reparación o adquisición.
- El comportamiento de todo usuario no debe ir en contra de la moral y las buenas costumbres dentro de las diferentes salas de cómputo.
- El usuario deberá mantener limpia su área de trabajo.
- El uso del equipo de cómputo es exclusivamente para fines didácticos y de investigación, por lo que se prohíbe el uso de juegos, trabajos personales ó de terceros con fines de lucro.
- Solo se permitirá el trabajo individual por computadora, a excepciones de aquellos cursos ó clases impartidas por instructores ó profesores.
- Los usuarios no podrán ingresar ningún tipo de radio, grabadora ó elementos que produzcan ruido y/o elementos magnéticos.
- Los estudiantes deben guardar respeto mantener los canales de comunicación y demás normas establecidas con los monitores y responsables del centro de cómputo y estos para con ellos.
- El acceso a los laboratorios es para el usuario que haya solicitado el servicio, bajo ninguna circunstancia podrán estar dos personas trabajando conjuntamente en una computadora.

Uso adecuado del equipo y materiales

- El equipo periférico que se presta, es solo para uso durante la sesión de trabajo por lo que deberá de devolverlo al término de esta.
- Para asesorías ó dudas con respecto al uso del equipo, debe dirigirse al responsable en turno dentro de la sala donde se encuentre ó bien dirigirse al personal encargado del laboratorio
- El usuario no debe desconectar ni destapar los ratones de las computadoras, las impresoras ó cualquier equipo periférico que se tenga instalado.
- Es obligación del usuario saber operar el equipo de cómputo y periféricos; debe consultar con el personal del servicio social o leer los folletos sobre el uso de periféricos distribuidos para tal efecto; ya que el daño por el mal uso será responsabilidad del usuario.
- En caso de necesitar tóner la impresora notifique al responsable en turno ó bien al encargado del laboratorio, para que el la verifique y éste sea remplazado.

Procedimientos en caso de emergencia

- Evacuación ordenada por rutas señalizadas.
- Uso de extintores solo por personal capacitado.
- Reporte inmediato de accidentes al personal responsable.

RELACIÓN DE PRÁCTICAS DE LABORATORIO POR ELEMENTO DE COMPETENCIA

Elemento de Competencia al que pertenece la práctica	EC I
	Identificar los conceptos básicos de PHP, con la finalidad de conocer los aspectos esenciales del lenguaje para comenzar el desarrollo de aplicaciones web del lado del servidor, bajo los estándares para el desarrollo web.

PRÁCTICA	NOMBRE	COMPETENCIA
Práctica No. 1	Fundamentos de PHP	Elaborar un mapa mental sobre los fundamentos de PHP para organizar y comprender los conceptos esenciales del lenguaje, utilizando fuentes de información confiables y herramientas digitales, en el contexto del aula y plataforma educativa, fortaleciendo el aprendizaje autónomo y la capacidad de análisis.
Práctica No. 2	Instalación y configuración de PHP	Realizar la instalación y configuración del servicio PHP para verificar el funcionamiento del entorno de desarrollo, siguiendo instrucciones del facilitador y documentación oficial, en el laboratorio de cómputo, promoviendo la responsabilidad y el uso adecuado de la tecnología.
Práctica No. 3	Código básico de la estructura de PHP	Desarrollar un reporte de práctica sobre la estructura básica de código PHP para comprobar el funcionamiento correcto en el servidor, con base en fuentes confiables y la guía del facilitador, en el laboratorio y aula, fomentando la organización y la responsabilidad.
Práctica No. 4	Introducción a la sintaxis PHP, etiquetas y comentarios	Elaborar un ejemplo de código que utilice sintaxis, etiquetas y comentarios en PHP para documentar adecuadamente programas sencillos, utilizando referencias confiables, en el aula y laboratorio, desarrollando la comunicación efectiva y la claridad en la documentación.
Práctica No. 5	Tipos de datos en PHP	Investigar los tipos de datos existentes en

		PHP para identificar y diferenciar sus usos y características, consultando fuentes confiables y recursos digitales, en el contexto del aula y trabajo independiente, fortaleciendo el aprendizaje autónomo y la capacidad de síntesis.
Práctica No. 6	Variables y constantes	Elaborar un mapa conceptual sobre variables y constantes en PHP para organizar y visualizar sus diferencias y aplicaciones, utilizando herramientas digitales y fuentes confiables, en el aula y plataforma educativa, promoviendo la organización y el análisis.
Práctica No. 7	Operadores en PHP	Construir un cuadro sinóptico sobre los operadores en PHP para sintetizar y clasificar su funcionamiento y ejemplos, empleando recursos digitales y bibliografía recomendada, en el aula y plataforma, desarrollando la capacidad de síntesis y la organización de la información.
Práctica No. 8	Tabla descriptiva referente a las estructuras de control	Elaborar una tabla descriptiva sobre las estructuras de control en PHP para comparar su funcionamiento y aplicación mediante ejemplos, usando herramientas digitales y referencias confiables, en el aula y plataforma, fortaleciendo la organización y el análisis lógico.

Elemento de Competencia al que pertenece la práctica	EC II
	Realizar aplicaciones web básicas en el lenguaje PHP con el propósito de conocer las funciones principales para el desarrollo y administración del uso de sitios web, con un enfoque de calidad.

PRÁCTICA	NOMBRE	COMPETENCIA
Práctica No. 9	Funcionamiento de las funciones en PHP	Investigar el funcionamiento de las funciones en PHP para comprender su declaración, argumentos y retorno de valores, consultando documentación oficial y recursos digitales, en el aula y

		plataforma, promoviendo el aprendizaje autónomo y la capacidad de análisis.
Práctica No. 10	Ejercicios de funciones para manipular cadenas en PHP	Resolver ejercicios prácticos de manipulación de cadenas en PHP para aplicar funciones internas del lenguaje, siguiendo las indicaciones del facilitador y usando fuentes confiables, en el aula y plataforma, desarrollando la resolución de problemas y la responsabilidad.
Práctica No. 11	Tabla sobre las funciones de fecha y hora en PHP	Elaborar una tabla comparativa de funciones de fecha y hora en PHP para identificar su estructura y aplicación mediante ejemplos, utilizando bibliografía y recursos digitales, en el aula y trabajo independiente, promoviendo la organización y el análisis funcional.
Práctica No. 12	Reporte escrito de funciones para administrar sesiones en PHP	Investigar y reportar las funciones para administrar sesiones en PHP para comprender su aplicación en la gestión de usuarios, consultando documentación oficial y recursos digitales, en el aula y plataforma, fortaleciendo la responsabilidad y la capacidad de análisis.
Práctica No. 13	Página web con funciones PHP para administrar inicio y fin de sesión	Desarrollar una aplicación web que implemente funciones de inicio y cierre de sesión en PHP para gestionar la autenticación de usuarios, siguiendo ejemplos y referencias confiables, en el laboratorio y aula, promoviendo la responsabilidad y la autonomía.
Práctica No. 14	Uso y funcionamiento de cookies en PHP	Analizar críticamente el uso y funcionamiento de cookies en PHP para identificar sus ventajas y desventajas en el almacenamiento de datos, consultando fuentes confiables y documentación oficial, en el aula y trabajo independiente, desarrollando el pensamiento crítico y la toma de decisiones informada.

Elemento de Competencia al que pertenece la práctica	EC III
	Desarrollar sistemas de gestión con páginas dinámicas del lado del servidor utilizando PHP y MySQL, con el propósito de responder las necesidades de las organizaciones que le permitan automatizar una gran cantidad de sus tareas con enfoque de calidad.

PRÁCTICA	NOMBRE	COMPETENCIA
Práctica No. 15	Desarrollo de un sistema de gestión básico con PHP y MySQL	Desarrollar una aplicación web que permita registrar, consultar, modificar y eliminar información de empleados en una empresa, utilizando PHP para la lógica del servidor y MySQL como base de datos.
Práctica No. 16	Implementación de formularios para captura y consulta de datos	Crear un formulario web para registrar productos en una tienda y mostrar una lista de productos registrados.
Práctica No. 17	Generación de reportes dinámicos a partir de la base de datos	Programar la generación de un reporte en PDF o en pantalla que muestre el listado de empleados con filtros por área o salario.
Práctica No. 18	Integración de funciones de seguridad y control de acceso (sesiones y roles de usuario)	Implementar un sistema de autenticación donde solo los usuarios registrados puedan acceder a ciertas páginas, usando sesiones y roles (administrador, usuario).



PRÁCTICAS

NOMBRE DE LA PRÁCTICA	Fundamentos de PHP
COMPETENCIA DE LA PRÁCTICA	Elaborar un mapa mental sobre los fundamentos de PHP para organizar y comprender los conceptos esenciales del lenguaje, utilizando fuentes de información confiables y herramientas digitales, en el contexto del aula y plataforma educativa, fortaleciendo el aprendizaje autónomo y la capacidad de análisis.

FUNDAMENTO TEÓRICO
PHP es un lenguaje de programación de propósito general, ampliamente utilizado para el desarrollo de aplicaciones web del lado del servidor. Permite generar contenido dinámico, gestionar bases de datos y automatizar tareas en sitios web. Su sintaxis es sencilla y se integra fácilmente con HTML, lo que lo convierte en una herramienta fundamental para el desarrollo web moderno

MATERIALES, EQUIPAMIENTO Y/O REACTIVOS
<ul style="list-style-type: none"> • Computadora de laboratorio • Acceso a internet

PROCEDIMIENTO O METODOLOGÍA
<ul style="list-style-type: none"> • Investiga los conceptos: ¿Qué es PHP?, funcionamiento, importancia y usos. • Utiliza una herramienta digital (Lucidchart, Canva, MindMeister) para crear el mapa mental. • Incluye ramas principales: Definición, Funcionamiento, Importancia, Ejemplos. • Entrega el archivo en PDF a través de la plataforma educativa. • [Ejemplo de mapa mental de PHP](https://i.imgur.com/), puedes crear tu propio mapa mental con las herramientas sugeridas.

RESULTADOS ESPERADOS
Identificar correctamente los conceptos básicos de PHP y representaron visualmente la relación entre ellos. Se observó claridad en la organización de ideas y uso adecuado de recursos digitales.

ANÁLISIS DE RESULTADOS
El uso de mapas mentales facilita la comprensión global de los fundamentos de PHP y ayuda a estructurar el conocimiento de manera visual y ordenada.

CONCLUSIONES Y REFLEXIONES
Relación con la teoría y aplicación en el campo profesional

EVALUACIÓN Y EVIDENCIAS DE APRENDIZAJE

Criterios de evaluación	
Rúbricas o listas de cotejo para valorar desempeño	Rubrica de Mapa Mental
Formatos de reporte de prácticas	Reporte de práctica

NOMBRE DE LA PRÁCTICA	Instalación y configuración de PHP
COMPETENCIA DE LA PRÁCTICA	Realizar la instalación y configuración del servicio PHP para verificar el funcionamiento del entorno de desarrollo, siguiendo instrucciones del facilitador y documentación oficial, en el laboratorio de cómputo, promoviendo la responsabilidad y el uso adecuado de la tecnología.

FUNDAMENTO TEÓRICO
La instalación y configuración de PHP requiere de un servidor web (como Apache) y el propio intérprete de PHP. Este proceso es esencial para crear un entorno de desarrollo local que permita ejecutar y probar scripts PHP antes de su implementación en producción. Configurar correctamente el entorno garantiza el funcionamiento adecuado del software y la seguridad de las aplicaciones

MATERIALES, EQUIPAMIENTO Y/O REACTIVOS
<ul style="list-style-type: none"> • Computadora de laboratorio • Acceso a internet

PROCEDIMIENTO O METODOLOGÍA
<ul style="list-style-type: none"> • Descarga e instala XAMPP o WAMP (incluye Apache y PHP). • Abre el panel de control y activa Apache. • Crea un archivo info.php en la carpeta htdocs con: <pre><?php phpinfo(); ?></pre> • Accede a http://localhost/info.php en tu navegador y verifica la información de PHP. • Toma capturas de pantalla de cada paso y elabora un reporte en PDF. <p>Ejemplo de imágenes Instalación de XAMPP: ![[Instalador XAMPP]](https://i.imgur.com/control):** ![[Panel de Control XAMPP]](https://i.imgur.com/phpinfo):** ![[phpinfo()]](https://i.imgur.com/5N6pD7w.png resultados)</p>

RESULTADOS ESPERADOS
Instalar y configurar el entorno de desarrollo local, identificando correctamente la versión de PHP y el funcionamiento del servidor Apache.

ANÁLISIS DE RESULTADOS
Contar con un entorno de desarrollo correctamente configurado es esencial para el desarrollo y

prueba de aplicaciones web en PHP.

CONCLUSIONES Y REFLEXIONES

Verifica siempre la configuración del entorno antes de iniciar cualquier proyecto y guarda capturas de pantalla como evidencia para futuras referencias o solución de problemas.

ACTIVIDADES COMPLEMENTARIAS

EVALUACIÓN Y EVIDENCIAS DE APRENDIZAJE

Criterios de evaluación	
Rúbricas o listas de cotejo para valorar desempeño	Rubrica Reporte de practica
Formatos de reporte de prácticas	Reporte de Práctica

NOMBRE DE LA PRÁCTICA	Código básico de la estructura de PHP
COMPETENCIA DE LA PRÁCTICA	Desarrollar un reporte de práctica sobre la estructura básica de código PHP para comprobar el funcionamiento correcto en el servidor, con base en fuentes confiables y la guía del facilitador, en el laboratorio y aula, fomentando la organización y la responsabilidad.

FUNDAMENTO TÉCNICO

La sintaxis de PHP define cómo deben escribirse las instrucciones, el uso de etiquetas para delimitar el código PHP y los comentarios para documentar el código. Los comentarios permiten explicar el funcionamiento del código, facilitando el mantenimiento y la colaboración entre desarrolladores

MATERIALES, EQUIPAMIENTO Y/O REACTIVOS

- Computadora de laboratorio
- Acceso a internet

PROCEDIMIENTO O METODOLOGÍA

- Abre tu editor de texto y escribe el siguiente código:

```
<?php
echo "¡Hola, mundo!";
?>
```
- Guarda el archivo como hola.php en la carpeta htdocs.
- Accede a <http://localhost/hola.php> y verifica el resultado.
- Toma capturas del código y del resultado en el navegador.
- Elabora un reporte en PDF.
- Ejemplo de imágenes
Código fuente:
[Código PHP](<https://i.imgur.com/8w4J> navegador:**
[Hola Mundo en navegador](<https://i.imgur.com/1p6J>álisis de resultados

RESULTADOS ESPERADOS

Comprobar que los estudiantes entendieron la estructura básica de PHP y el proceso de despliegue en el servidor local.

ANÁLISIS DE RESULTADOS

Dominar la estructura básica de PHP es el primer paso para desarrollar aplicaciones web dinámicas y funcionales.

CONCLUSIONES Y REFLEXIONES

Practica frecuentemente la creación y ejecución de scripts básicos para familiarizarte con el entorno y la sintaxis de PHP.

ACTIVIDADES COMPLEMENTARIAS

EVALUACIÓN Y EVIDENCIAS DE APRENDIZAJE

Criterios de evaluación	
Rúbricas o listas de cotejo para valorar desempeño	Rubrica Reporte de practica
Formatos de reporte de prácticas	Reporte de Práctica

NOMBRE DE LA PRÁCTICA	Introducción a la sintaxis PHP, etiquetas y comentarios
COMPETENCIA DE LA PRÁCTICA	Elaborar un ejemplo de código que utilice sintaxis, etiquetas y comentarios en PHP para documentar adecuadamente programas sencillos, utilizando referencias confiables, en el aula y laboratorio, desarrollando la comunicación efectiva y la claridad en la documentación.

FUNDAMENTO TEÓRICO

La sintaxis de PHP define cómo deben escribirse las instrucciones, el uso de etiquetas para delimitar el código PHP y los comentarios para documentar el código. Los comentarios permiten explicar el funcionamiento del código, facilitando el mantenimiento y la colaboración entre desarrolladores

MATERIALES, EQUIPAMIENTO Y/O REACTIVOS

- Computadora de laboratorio
- Acceso a internet

PROCEDIMIENTO O METODOLOGÍA

- Escribe un archivo sintaxis.php con:


```
<?php
// Comentario de una línea
/*
Comentario
de varias líneas
*/
echo "Ejemplo de sintaxis básica en PHP";
?>
```
- Explica la función de los comentarios y las etiquetas en tu reporte.
- Incluye capturas de pantalla del código y del resultado.
- Ejemplo de imagen
- Código con comentarios:
- [Código con comentarios PHP](https://i.imgur.com/álisis de resultados)

RESULTADOS ESPERADOS

Aplicar correctamente los comentarios y las etiquetas, mejorando la documentación y legibilidad de sus códigos.

ANÁLISIS DE RESULTADOS

El uso adecuado de comentarios y etiquetas es fundamental para el mantenimiento y comprensión de los programas PHP

CONCLUSIONES Y REFLEXIONES

Documenta siempre tu código, aunque sea sencillo, para facilitar la colaboración y el mantenimiento futuro.

ACTIVIDADES COMPLEMENTARIAS

EVALUACIÓN Y EVIDENCIAS DE APRENDIZAJE

Criterios de evaluación

Rúbricas o listas de cotejo para valorar desempeño

Formatos de reporte de prácticas

Rubrica Reporte de practica

Reporte de Práctica

NOMBRE DE LA PRÁCTICA	Tipos de datos en PHP
COMPETENCIA DE LA PRÁCTICA	Investigar los tipos de datos existentes en PHP para identificar y diferenciar sus usos y características, consultando fuentes confiables y recursos digitales, en el contexto del aula y trabajo independiente, fortaleciendo el aprendizaje autónomo y la capacidad de síntesis.

FUNDAMENTO TÉCNICO
PHP soporta varios tipos de datos primitivos, como enteros, flotantes, cadenas, booleanos, arrays y objetos. Conocer los tipos de datos y su manejo es esencial para la correcta manipulación y almacenamiento de información en las aplicaciones web

MATERIALES, EQUIPAMIENTO Y/O REACTIVOS
<ul style="list-style-type: none"> • Computadora de laboratorio • Acceso a internet

PROCEDIMIENTO O METODOLOGÍA																								
<ul style="list-style-type: none"> • Indicaciones • Investiga los tipos de datos en PHP: String, Integer, Float, Boolean, Array, Object, NULL. • Elabora una tabla comparativa con definición y ejemplo de cada tipo. • Entrega el archivo en PDF. <p>Ejemplo de tabla</p> <table border="1"> <thead> <tr> <th>Tipo</th> <th>Descripción</th> <th>Ejemplo</th> </tr> </thead> <tbody> <tr> <td>String</td> <td>Cadena de texto</td> <td>"Hola"</td> </tr> <tr> <td>Integer</td> <td>Número entero</td> <td>42</td> </tr> <tr> <td>Float</td> <td>Número decimal</td> <td>3.14</td> </tr> <tr> <td>Boolean</td> <td>Verdadero/Falso</td> <td>true</td> </tr> <tr> <td>Array</td> <td>Conjunto de valores</td> <td>array(1,2,3)</td> </tr> <tr> <td>Object</td> <td>Instancia de clase</td> <td>new Persona()</td> </tr> <tr> <td>NULL</td> <td>Sin valor</td> <td>NULL</td> </tr> </tbody> </table>	Tipo	Descripción	Ejemplo	String	Cadena de texto	"Hola"	Integer	Número entero	42	Float	Número decimal	3.14	Boolean	Verdadero/Falso	true	Array	Conjunto de valores	array(1,2,3)	Object	Instancia de clase	new Persona()	NULL	Sin valor	NULL
Tipo	Descripción	Ejemplo																						
String	Cadena de texto	"Hola"																						
Integer	Número entero	42																						
Float	Número decimal	3.14																						
Boolean	Verdadero/Falso	true																						
Array	Conjunto de valores	array(1,2,3)																						
Object	Instancia de clase	new Persona()																						
NULL	Sin valor	NULL																						

RESULTADOS ESPERADOS
Identificar correctamente los tipos de datos, mostrando comprensión de su utilidad.

ANÁLISIS DE RESULTADOS

Conocer los tipos de datos es esencial para manipular información correctamente en PHP y evitar errores en la programación.

CONCLUSIONES Y REFLEXIONES

Consulta la documentación oficial de PHP para profundizar en el uso y comportamiento de cada tipo de dato.

ACTIVIDADES COMPLEMENTARIAS

EVALUACIÓN Y EVIDENCIAS DE APRENDIZAJE

Criterios de evaluación	
Rúbricas o listas de cotejo para valorar desempeño	Rubrica Reporte de practica
Formatos de reporte de prácticas	Reporte de Práctica

NOMBRE DE LA PRÁCTICA	Variables y constantes
COMPETENCIA DE LA PRÁCTICA	Elaborar un mapa conceptual sobre variables y constantes en PHP para organizar y visualizar sus diferencias y aplicaciones, utilizando herramientas digitales y fuentes confiables, en el aula y plataforma educativa, promoviendo la organización y el análisis.

FUNDAMENTO TÉCNICO
Las variables en PHP almacenan datos que pueden cambiar durante la ejecución del programa, mientras que las constantes mantienen un valor fijo. El uso adecuado de variables y constantes es clave para la organización y reutilización del código, así como para la gestión eficiente de la información

MATERIALES, EQUIPAMIENTO Y/O REACTIVOS
<ul style="list-style-type: none"> • Computadora de laboratorio • Acceso a internet

PROCEDIMIENTO O METODOLOGÍA
<ul style="list-style-type: none"> • Investiga qué son variables y constantes en PHP: definición, sintaxis, diferencias y ejemplos. • Elabora un mapa conceptual utilizando una herramienta digital (Lucidchart, Canva, MindMeister). • Entrega el archivo en PDF a través de la plataforma educativa.

RESULTADOS ESPERADOS
Identificar correctamente las diferencias y aplicaciones de variables y constantes, mostrando claridad en la organización visual.

ANÁLISIS DE RESULTADOS
El mapa conceptual facilita la comprensión y memorización de conceptos clave sobre variables y constantes, esenciales para la programación en PHP.

CONCLUSIONES Y REFLEXIONES
Utiliza mapas conceptuales para comparar conceptos similares y aclarar diferencias fundamentales en programación.

EVALUACIÓN Y EVIDENCIAS DE APRENDIZAJE	
Criterios de evaluación	
Rúbricas o listas de cotejo para valorar desempeño	Rubrica de mapa conceptual
Formatos de reporte de prácticas	Reporte de Práctica

NOMBRE DE LA PRÁCTICA	Operadores en PHP
COMPETENCIA DE LA PRÁCTICA	Construir un cuadro sinóptico sobre los operadores en PHP para sintetizar y clasificar su funcionamiento y ejemplos, empleando recursos digitales y bibliografía recomendada, en el aula y plataforma, desarrollando la capacidad de síntesis y la organización de la información.

FUNDAMENTO TEÓRICO
Los operadores en PHP permiten realizar operaciones aritméticas, lógicas, de comparación y asignación. Dominar el uso de operadores es fundamental para implementar condiciones, cálculos y manipulación de datos en los programas.

MATERIALES, EQUIPAMIENTO Y/O REACTIVOS
<ul style="list-style-type: none"> • Computadora de laboratorio • Acceso a internet

PROCEDIMIENTO O METODOLOGÍA			
<ul style="list-style-type: none"> • Investiga los distintos tipos de operadores en PHP: aritméticos, de asignación, comparación, lógicos, etc. • Elabora un cuadro sinóptico digital que clasifique y ejemplifique cada tipo. 			
Tipo de Operador	Operador	Descripción	Ejemplo
Aritméticos	+	Suma	$\$a + \b
	-	Resta	$\$a - \b
	*	Multiplicación	$\$a * \b
	/	División	$\$a / \b
	%	Módulo (residuo)	$\$a \% \b
Asignación	=	Asignación simple	$\$a = 5$
	+=	Suma y asigna	$\$a += 2$
	-=	Resta y asigna	$\$a -= 2$
	*=	Multiplica y asigna	$\$a *= 2$

	/=	Divide y asigna	\$a /= 2
	%=	Módulo y asigna	\$a %= 2
Comparación	==	Igual a	\$a == \$b
	===	Idéntico a (igual valor y tipo)	\$a === \$b
	!=	Diferente de	\$a != \$b
	<>	Diferente de	\$a <> \$b
	!==	No es idéntico a	\$a !== \$b
	>	Mayor que	\$a > \$b
	<	Menor que	\$a < \$b
	>=	Mayor o igual que	\$a >= \$b
	<=	Menor o igual que	\$a <= \$b
Lógicos	&&	Y lógico (AND)	(\$a && \$b)
	!	Negación lógica (NOT)	!\$a
Incremento/Decremento	++	Incrementa en uno	\$a++
	--	Decrementa en uno	\$a--
Concatenación	.	Une dos cadenas	\$a . \$b

- Entrega el archivo en PDF.

RESULTADOS ESPERADOS

Clasificar y ejemplificar correctamente los operadores, mostrando comprensión de su uso en distintos contextos.

ANÁLISIS DE RESULTADOS

El cuadro sinóptico es una herramienta útil para visualizar y recordar los diferentes operadores disponibles en PHP.

CONCLUSIONES Y REFLEXIONES

Consulta el cuadro sinóptico cuando tengas dudas sobre el uso de operadores en tus programas.

ACTIVIDADES COMPLEMENTARIAS

EVALUACIÓN Y EVIDENCIAS DE APRENDIZAJE

Criterios de evaluación	
Rúbricas o listas de cotejo para valorar desempeño	Rubrica cuadro sinóptico
Formatos de reporte de prácticas	Reporte de Práctica

NOMBRE DE LA PRÁCTICA	Tabla descriptiva referente a las estructuras de control
COMPETENCIA DE LA PRÁCTICA	Elaborar una tabla descriptiva sobre las estructuras de control en PHP para comparar su funcionamiento y aplicación mediante ejemplos, usando herramientas digitales y referencias confiables, en el aula y plataforma, fortaleciendo la organización y el análisis lógico.

FUNDAMENTO TEÓRICO

Las estructuras de control, como `if`, `else`, `switch`, `for`, `while` y `foreach`, permiten controlar el flujo de ejecución del programa. Su correcta aplicación es esencial para la toma de decisiones y la repetición de tareas en el desarrollo de software

MATERIALES, EQUIPAMIENTO Y/O REACTIVOS

- Computadora de laboratorio
- Acceso a internet

PROCEDIMIENTO O METODOLOGÍA

- Investiga las principales estructuras de control en PHP: `if`, `else`, `switch`, `for`, `while`, `foreach`.
- Elabora una tabla de tres columnas: nombre, funcionamiento, ejemplo.

Estructura	Funcionamiento	Ejemplo
<code>if</code>	Evalúa condición lógica	<code>if (\$a > 0) { ... }</code>
<code>for</code>	Repite bloque de código	<code>for (\$i=0;\$i<5;\$i++){}</code>

- Entrega el archivo en PDF.

RESULTADOS ESPERADOS

Describir adecuadamente el funcionamiento y aplicación de cada estructura de control.

ANÁLISIS DE RESULTADOS

Una tabla comparativa ayuda a identificar rápidamente la estructura de control adecuada para cada situación en programación.

CONCLUSIONES Y REFLEXIONES

Revisa la tabla antes de implementar estructuras de control en nuevos proyectos.

ACTIVIDADES COMPLEMENTARIAS

EVALUACIÓN Y EVIDENCIAS DE APRENDIZAJE

Criterios de evaluación	
Rúbricas o listas de cotejo para valorar desempeño	Rubrica Reporte de practica
Formatos de reporte de prácticas	Reporte de Práctica

NOMBRE DE LA PRÁCTICA

Funcionamiento de las funciones en PHP

COMPETENCIA DE LA PRÁCTICA

Investigar el funcionamiento de las funciones en PHP para comprender su declaración, argumentos y retorno de valores, consultando documentación oficial y recursos digitales, en el aula y plataforma, promoviendo el aprendizaje autónomo y la capacidad de análisis.

FUNDAMENTO TEÓRICO

Las funciones en PHP permiten agrupar instrucciones bajo un nombre, facilitando la reutilización y modularidad del código. Existen funciones internas del lenguaje y funciones definidas por el usuario, que pueden recibir argumentos y devolver valores.

MATERIALES, EQUIPAMIENTO Y/O REACTIVOS

- Computadora de laboratorio
- Acceso a internet

PROCEDIMIENTO O METODOLOGÍA

- Investiga qué son las funciones en PHP, tipos (internas y de usuario), argumentos y valores de retorno.
- Incluye ejemplos de declaración y uso.
php
function suma(\$a, \$b) {
 return \$a + \$b;
}
- Elabora un reporte escrito y entrégalo en PDF.

RESULTADOS ESPERADOS

Diferenciar entre funciones internas y de usuario, y ejemplificaron correctamente su uso.

ANÁLISIS DE RESULTADOS

Dominar el uso de funciones permite crear código modular, reutilizable y más fácil de mantener.

CONCLUSIONES Y REFLEXIONES

Utiliza funciones para evitar la repetición de código y mejorar la organización de tus programas.

ACTIVIDADES COMPLEMENTARIAS

EVALUACIÓN Y EVIDENCIAS DE APRENDIZAJE

Criterios de evaluación	
Rúbricas o listas de cotejo para valorar desempeño	Rubrica de cuadro Sinóptico
Formatos de reporte de prácticas	Reporte de Práctica

NOMBRE DE LA PRÁCTICA	Ejercicios de funciones para manipular cadenas en PHP
COMPETENCIA DE LA PRÁCTICA	Resolver ejercicios prácticos de manipulación de cadenas en PHP para aplicar funciones internas del lenguaje, siguiendo las indicaciones del facilitador y usando fuentes confiables, en el aula y plataforma, desarrollando la resolución de problemas y la responsabilidad.

FUNDAMENTO TEÓRICO

PHP ofrece funciones específicas para manipular cadenas de texto, como concatenación, búsqueda, reemplazo y transformación de mayúsculas/minúsculas. Estas funciones son esenciales para el procesamiento y presentación de información en aplicaciones web

MATERIALES, EQUIPAMIENTO Y/O REACTIVOS

- Computadora de laboratorio
- Acceso a internet

PROCEDIMIENTO O METODOLOGÍA

- Realiza ejercicios prácticos usando funciones como `strlen()`, `strtoupper()`, `substr()`, etc.

- Presenta el código y el resultado en un reporte PDF.

```
php
$texto = "Hola Mundo";
echo strtoupper($texto); // HOLA MUNDO
```

Ejercicios de funciones para manipular cadenas en PHP

1. Obtener la longitud de una cadena (**strlen**)

```
php
<?php
$texto = "Universidad Estatal de Sonora";
echo "La longitud es: " . strlen($texto); // Resultado: 29
?>
```

2. Convertir a mayúsculas (**strtoupper**)

```
php
<?php
$texto = "php es divertido";
echo strtoupper($texto); // Resultado: PHP ES DIVERTIDO
?>
```

3. Convertir a minúsculas (**strtolower**)

```
php
<?php
```

```
$texto = "PROGRAMACIÓN WEB";  
echo strtolower($texto); // Resultado: programación web  
?>
```

4. Extraer una subcadena (**substr**)

```
php  
<?php  
$texto = "Programación Web";  
echo substr($texto, 0, 12); // Resultado: Programación  
?>
```

5. Buscar la posición de una palabra o carácter (**strpos**)

```
php  
<?php  
$frase = "Bienvenido a PHP";  
echo strpos($frase, "PHP"); // Resultado: 13  
?>
```

6. Reemplazar texto en una cadena (**str_replace**)

```
php  
<?php  
$frase = "Hola Mundo";  
echo str_replace("Mundo", "PHP", $frase); // Resultado: Hola PHP  
?>
```

7. Revertir una cadena (**strrev**)

```
php  
<?php  
$texto = "UES";  
echo strrev($texto); // Resultado: SEU  
?>
```

8. Eliminar espacios al inicio y final de una cadena (**trim**)

```
php  
<?php  
$texto = " Hola PHP ";  
echo trim($texto); // Resultado: Hola PHP  
?>
```

9. Contar cuántas veces aparece una palabra (**substr_count**)

```
php  
<?php  
$texto = "PHP es genial, PHP es poderoso";  
echo substr_count($texto, "PHP"); // Resultado: 2  
?>
```

10. Dividir una cadena en un arreglo (**explode**)

```
php  
<?php
```

```
$frase = "uno,dos,tres,cuatro";
$arreglo = explode(",", $frase);
print_r($arreglo); // Resultado: Array ( [0] => uno [1] => dos [2] => tres [3] => cuatro )
?>
```

RESULTADOS ESPERADOS

Aplicar correctamente las funciones de manipulación de cadenas, mostrando comprensión práctica.

ANÁLISIS DE RESULTADOS

Las funciones para cadenas son esenciales para el procesamiento y presentación de información textual en aplicaciones web.

CONCLUSIONES Y REFLEXIONES

Consulta la documentación oficial de PHP para descubrir más funciones útiles para manipular cadenas.

ACTIVIDADES COMPLEMENTARIAS

EVALUACIÓN Y EVIDENCIAS DE APRENDIZAJE

Criterios de evaluación	
Rúbricas o listas de cotejo para valorar desempeño	Rubrica solución de ejercicios
Formatos de reporte de prácticas	Reporte de Práctica

NOMBRE DE LA PRÁCTICA	Tabla sobre las funciones de fecha y hora en PHP
COMPETENCIA DE LA PRÁCTICA	Elaborar una tabla comparativa de funciones de fecha y hora en PHP para identificar su estructura y aplicación mediante ejemplos, utilizando bibliografía y recursos digitales, en el aula y trabajo independiente, promoviendo la organización y el análisis funcional.

FUNDAMENTO TÉCNICO
Las funciones de fecha y hora en PHP permiten obtener, formatear y manipular valores temporales, lo que es fundamental para gestionar eventos, registros y operaciones dependientes del tiempo en las aplicaciones web.

MATERIALES, EQUIPAMIENTO Y/O REACTIVOS
<ul style="list-style-type: none"> • Computadora de laboratorio • Acceso a internet

PROCEDIMIENTO O METODOLOGÍA									
<ul style="list-style-type: none"> • Investiga las funciones de fecha y hora en PHP: date(), time(), strtotime(), etc. • Elabora una tabla con estructura, funcionamiento y ejemplo. • Entrega el archivo en PDF. 									
<table border="1"> <thead> <tr> <th>Función</th> <th>Funcionamiento</th> <th>Ejemplo</th> </tr> </thead> <tbody> <tr> <td>date()</td> <td>Devuelve fecha formateada</td> <td>date("Y-m-d")</td> </tr> <tr> <td>time()</td> <td>Timestamp actual</td> <td>time()</td> </tr> </tbody> </table>	Función	Funcionamiento	Ejemplo	date()	Devuelve fecha formateada	date("Y-m-d")	time()	Timestamp actual	time()
Función	Funcionamiento	Ejemplo							
date()	Devuelve fecha formateada	date("Y-m-d")							
time()	Timestamp actual	time()							

RESULTADOS ESPERADOS
Identificar correctamente la utilidad y sintaxis de las funciones de fecha y hora.

ANÁLISIS DE RESULTADOS
Estas funciones son indispensables para gestionar eventos y registros en aplicaciones web.

CONCLUSIONES Y REFLEXIONES
Usar el formato de fecha/hora adecuado según los requerimientos del sistema.

ACTIVIDADES COMPLEMENTARIAS

EVALUACIÓN Y EVIDENCIAS DE APRENDIZAJE

Criterios de evaluación	
Rúbricas o listas de cotejo para valorar desempeño	Rubrica Reporte de practica
Formatos de reporte de prácticas	Reporte de Práctica

NOMBRE DE LA PRÁCTICA	Reporte escrito de funciones para administrar sesiones en PHP
COMPETENCIA DE LA PRÁCTICA	Investigar y reportar las funciones para administrar sesiones en PHP para comprender su aplicación en la gestión de usuarios, consultando documentación oficial y recursos digitales, en el aula y plataforma, fortaleciendo la responsabilidad y la capacidad de análisis.

FUNDAMENTO TEÓRICO

Las sesiones en PHP permiten almacenar información del usuario en el servidor durante su interacción con el sitio web. Las funciones de sesión son esenciales para implementar autenticación, personalización y seguimiento de usuarios

MATERIALES, EQUIPAMIENTO Y/O REACTIVOS

- Computadora de laboratorio
- Acceso a internet

PROCEDIMIENTO O METODOLOGÍA

- Investiga las funciones para manejo de sesiones: `session_start()`, `$_SESSION`, `session_destroy()`.
- Elabora un reporte explicando cada función y su uso.
php
`session_start();`
`$_SESSION["usuario"] = "Juan";`
- Entrega el archivo en PDF.

RESULTADOS ESPERADOS

Comprender la importancia de las sesiones para la gestión de usuarios y seguridad.

ANÁLISIS DE RESULTADOS

El manejo de sesiones es fundamental para implementar autenticación y personalización en aplicaciones web.

CONCLUSIONES Y REFLEXIONES

Siempre destruye la sesión al cerrar la aplicación para proteger la información del usuario.

ACTIVIDADES COMPLEMENTARIAS

EVALUACIÓN Y EVIDENCIAS DE APRENDIZAJE

Criterios de evaluación	
Rúbricas o listas de cotejo para valorar desempeño	Rubrica Reporte escrito
Formatos de reporte de prácticas	Reporte de Práctica

NOMBRE DE LA PRÁCTICA	Página web con funciones PHP para administrar inicio y fin de sesión
COMPETENCIA DE LA PRÁCTICA	Desarrollar una aplicación web que implemente funciones de inicio y cierre de sesión en PHP para gestionar la autenticación de usuarios, siguiendo ejemplos y referencias confiables, en el laboratorio y aula, promoviendo la responsabilidad y la autonomía.

FUNDAMENTO TEÓRICO

El manejo de sesiones permite controlar el acceso y la seguridad de los usuarios en aplicaciones web, asegurando que solo usuarios autenticados puedan acceder a ciertas funcionalidades. Implementar inicio y cierre de sesión es una práctica estándar en el desarrollo web seguro

MATERIALES, EQUIPAMIENTO Y/O REACTIVOS

- Computadora de laboratorio
- Acceso a internet

PROCEDIMIENTO O METODOLOGÍA

- Desarrolla una aplicación web sencilla que permita iniciar y cerrar sesión usando PHP.
- Implementa los archivos necesarios (login.php, logout.php).

Ejemplo de código

login.php

```
<?php
session_start();
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $usuario = $_POST['usuario'];
    $password = $_POST['password'];
    // Ejemplo sencillo: usuario=admin, password=1234
    if ($usuario == "admin" && $password == "1234") {
        $_SESSION['usuario'] = $usuario;
        header("Location: bienvenida.php");
        exit();
    }
}
```

```

    } else {
        $error = "Usuario o contraseña incorrectos";
    }
}
?>
<!DOCTYPE html>
<html>
<head>
    <title>Iniciar Sesión</title>
</head>
<body>
    <h2>Iniciar Sesión</h2>
    <?php if(isset($error)) echo "<p style='color:red;'>$error</p>"; ?>
    <form method="post" action="">
        Usuario: <input type="text" name="usuario" required><br>
        Contraseña: <input type="password" name="password" required><br>
        <input type="submit" value="Entrar">
    </form>
</body>
</html>

```

bienvenida.php

```

<?php
session_start();
if (!isset($_SESSION['usuario'])) {
    header("Location: login.php");
    exit();
}
?>
<!DOCTYPE html>
<html>
<head>
    <title>Bienvenida</title>
</head>
<body>
    <h2>Bienvenido, <?php echo $_SESSION['usuario']; ?>!</h2>
    <a href="logout.php">Cerrar sesión</a>
</body>
</html>

```

logout.php

```

<?php
session_start();
session_destroy();
header("Location: login.php");
exit();
?>

```

- Presenta capturas de pantalla del funcionamiento y entrega el código en PDF.

RESULTADOS ESPERADOS

Implementación correcta de la autenticación básica, comprendiendo el flujo de inicio y cierre de

sesión.

ANÁLISIS DE RESULTADOS

La gestión de sesiones es clave para controlar el acceso y la seguridad en sistemas web.

CONCLUSIONES Y REFLEXIONES

Implementa validaciones y controles adicionales para fortalecer la seguridad de los sistemas de autenticación.

ACTIVIDADES COMPLEMENTARIAS

EVALUACIÓN Y EVIDENCIAS DE APRENDIZAJE

Criterios de evaluación	
Rúbricas o listas de cotejo para valorar desempeño	Rubrica solución de ejercicios
Formatos de reporte de prácticas	Reporte de Práctica

NOMBRE DE LA PRÁCTICA

Uso y funcionamiento de cookies en PHP

COMPETENCIA DE LA PRÁCTICA

Analizar críticamente el uso y funcionamiento de cookies en PHP para identificar sus ventajas y desventajas en el almacenamiento de datos, consultando fuentes confiables y documentación oficial, en el aula y trabajo independiente, desarrollando el pensamiento crítico y la toma de decisiones informada.

FUNDAMENTO TEÓRICO

Las cookies son pequeños archivos almacenados en el navegador del usuario que permiten guardar información entre sesiones. Su uso adecuado es crucial para la personalización de la experiencia del usuario y la gestión de preferencias, pero implica consideraciones de seguridad y privacidad

MATERIALES, EQUIPAMIENTO Y/O REACTIVOS

- Computadora de laboratorio
- Acceso a internet

PROCEDIMIENTO O METODOLOGÍA

- Investiga las ventajas y desventajas de utilizar cookies en el navegador con PHP.
- Realiza un ejemplo práctico donde se cree, lea y elimine una cookie.

Ejemplo de código PHP

Crear una cookie

```
php
<?php
// Crear una cookie llamada "usuario" con valor "Juan" que expira en 1 hora
setcookie("usuario", "Juan", time() + 3600);
echo "Cookie 'usuario' creada.<br>";
?>
```

Leer una cookie

```
php
<?php
if (isset($_COOKIE["usuario"])) {
    echo "El valor de la cookie 'usuario' es: " . $_COOKIE["usuario"];
} else {
    echo "La cookie 'usuario' no está definida.";
}
?>
```

Eliminar una cookie

```
php
<?php
// Para eliminar una cookie, se establece una fecha de expiración en el pasado
setcookie("usuario", "", time() - 3600);
echo "Cookie 'usuario' eliminada.";
?>
```

- Elabora un reporte breve con el análisis, el código y capturas de pantalla del funcionamiento.

RESULTADOS ESPERADOS

Identificar correctamente los riesgos y beneficios del uso de cookies en aplicaciones web.

ANÁLISIS DE RESULTADOS

Las cookies son útiles para personalizar la experiencia del usuario, pero requieren un manejo responsable por temas de privacidad.

CONCLUSIONES Y REFLEXIONES

Informa siempre a los usuarios sobre el uso de cookies y cumple con la normativa de protección de datos.

ACTIVIDADES COMPLEMENTARIAS

EVALUACIÓN Y EVIDENCIAS DE APRENDIZAJE

Criterios de evaluación

Rúbricas o listas de cotejo para valorar desempeño

Formatos de reporte de prácticas

Rubrica Reporte de practica

Reporte de Práctica

NOMBRE DE LA PRÁCTICA	Desarrollo de un sistema de gestión básico con PHP y MySQL
COMPETENCIA DE LA PRÁCTICA	Desarrollar una aplicación web que permita registrar, consultar, modificar y eliminar información de empleados en una empresa, utilizando PHP para la lógica del servidor y MySQL como base de datos.

FUNDAMENTO TEÓRICO	
La integración de PHP y MySQL permite crear aplicaciones web dinámicas que gestionan grandes volúmenes de datos de manera eficiente y segura. Las operaciones CRUD (Create, Read, Update, Delete) son fundamentales para cualquier sistema de gestión, automatizando procesos y facilitando la administración de la información.	

MATERIALES, EQUIPAMIENTO Y/O REACTIVOS	
<ul style="list-style-type: none"> • Computadora de laboratorio • Acceso a internet 	

PROCEDIMIENTO O METODOLOGÍA	
<ul style="list-style-type: none"> • Diseñar la base de datos en MySQL (tabla empleados: id, nombre, puesto, salario). • Crear formularios en HTML para capturar y mostrar información. • Programar scripts PHP para conectar con la base de datos y realizar operaciones CRUD. • Validar los datos recibidos del usuario. • Mostrar mensajes de éxito o error según la operación realizada. • Documentar el código y tomar capturas de pantalla del funcionamiento. 	
Procedimiento y código de ejemplo	
a) Crear la base de datos y la tabla	
<pre> sql -- Ejecuta esto en phpMyAdmin o tu cliente MySQL CREATE DATABASE gestion; USE gestion; CREATE TABLE empleados (id INT AUTO_INCREMENT PRIMARY KEY, nombre VARCHAR(100) NOT NULL, puesto VARCHAR(50) NOT NULL, salario DECIMAL(10,2) NOT NULL); </pre>	
b) Conexión a la base de datos (conexion.php)	
<pre> php <?php \$conexion = new mysqli("localhost", "root", "", "gestion"); if (\$conexion->connect_error) { </pre>	

```

    die("Conexión fallida: " . $conexion->connect_error);
}
?>

```

c) Formulario para agregar empleados (agregar.php)

```

php
<!DOCTYPE html>
<html>
<head>
    <title>Agregar Empleado</title>
</head>
<body>
    <h2>Agregar Empleado</h2>
    <form method="post" action="guardar.php">
        Nombre: <input type="text" name="nombre" required><br>
        Puesto: <input type="text" name="puesto" required><br>
        Salario: <input type="number" name="salario" step="0.01" required><br>
        <input type="submit" value="Guardar">
    </form>
    <a href="listar.php">Ver empleados</a>
</body>
</html>

```

d) Guardar empleado en la base de datos (guardar.php)

```

php
<?php
include 'conexion.php';
$nombre = $_POST['nombre'];
$puesto = $_POST['puesto'];
$salario = $_POST['salario'];

$sql = "INSERT INTO empleados (nombre, puesto, salario) VALUES ('$nombre', '$puesto', '$salario')";
if ($conexion->query($sql) === TRUE) {
    echo "Empleado guardado correctamente.<br>";
} else {
    echo "Error: " . $conexion->error;
}
echo '<a href="agregar.php">Agregar otro</a> | <a href="listar.php">Ver empleados</a>';
?>

```

e) Listar empleados (listar.php)

```

php
<?php
include 'conexion.php';
$resultado = $conexion->query("SELECT * FROM empleados");
?>
<!DOCTYPE html>
<html>
<head>
    <title>Lista de Empleados</title>
</head>
<body>
    <h2>Empleados Registrados</h2>
    <table border="1">

```

```

<tr>
  <th>ID</th>
  <th>Nombre</th>
  <th>Puesto</th>
  <th>Salario</th>
</tr>
<?php while($fila = $resultado->fetch_assoc()) { ?>
<tr>
  <td><?php echo $fila['id']; ?></td>
  <td><?php echo $fila['nombre']; ?></td>
  <td><?php echo $fila['puesto']; ?></td>
  <td><?php echo $fila['salario']; ?></td>
</tr>
<?php } ?>
</table>
<a href="agregar.php">Agregar empleado</a>
</body>
</html>

```

RESULTADOS ESPERADOS

El sistema permite agregar, consultar, modificar y eliminar empleados, la información se almacena y recupera correctamente desde MySQL y el sistema muestra mensajes claros al usuario sobre el estado de cada operación.

ANÁLISIS DE RESULTADOS

La mayoría de los equipos logra implementar correctamente las operaciones CRUD y la interacción entre PHP y MySQL, evidenciando comprensión de la arquitectura cliente-servidor y el manejo de bases de datos.

CONCLUSIONES Y REFLEXIONES

El desarrollo de sistemas de gestión con PHP y MySQL es esencial para automatizar tareas y mejorar la eficiencia en organizaciones. Se debe considerar validar siempre los datos de entrada para evitar errores y vulnerabilidades así como comentar el código para facilitar el mantenimiento y futuras mejoras.

ACTIVIDADES COMPLEMENTARIAS

EVALUACIÓN Y EVIDENCIAS DE APRENDIZAJE	
Criterios de evaluación	
Rúbricas o listas de cotejo para valorar desempeño	Rubrica Reporte de practica
Formatos de reporte de prácticas	Reporte de Práctica

NOMBRE DE LA PRÁCTICA	Implementación de formularios para captura y consulta de datos
COMPETENCIA DE LA PRÁCTICA	Crear un formulario web para registrar productos en una tienda y mostrar una lista de productos registrados.

FUNDAMENTO TEÓRICO

Los formularios HTML permiten la interacción entre el usuario y la aplicación web. PHP procesa los datos enviados por el formulario y los almacena en la base de datos. La validación de datos es crucial para garantizar la integridad y seguridad de la información.

- MATERIALES, EQUIPAMIENTO Y/O REACTIVOS**
- Computadora de laboratorio
 - Acceso a internet

PROCEDIMIENTO O METODOLOGÍA

- Diseñar un formulario HTML para capturar nombre, precio y cantidad de productos.
- Programar PHP para procesar el formulario y guardar los datos en MySQL.
- Crear una página que muestre la lista de productos registrados.
- Validar los datos antes de almacenarlos.

Procedimiento

a) Crear la base de datos y la tabla
 sql
CREATE DATABASE tienda;
USE tienda;

CREATE TABLE productos (
 id **INT AUTO_INCREMENT PRIMARY KEY**,
 nombre **VARCHAR(100)** NOT NULL,
 precio **DECIMAL(10,2)** NOT NULL,
 cantidad **INT** NOT NULL
);

b) Archivo de conexión (conexion.php)
 php
 <?php
\$conexion = new mysqli("localhost", "root", "", "tienda");
if (\$conexion->connect_error) {
die("Conexión fallida: " . \$conexion->connect_error);
}
 ?>

c) Formulario de registro (registrar.php)
 php
 <!DOCTYPE html>
 <html>
 <head>
 <title>Registrar Producto</title>

```

</head>
<body>
  <h2>Registrar Producto</h2>
  <form method="post" action="guardar.php">
    Nombre: <input type="text" name="nombre" required><br>
    Precio: <input type="number" name="precio" step="0.01" required><br>
    Cantidad: <input type="number" name="cantidad" required><br>
    <input type="submit" value="Guardar">
  </form>
  <a href="consultar.php">Ver productos</a>
</body>
</html>

```

d) Guardar producto en la base de datos (guardar.php)

```

php
<?php
include 'conexion.php';
$nombre = $_POST['nombre'];
$precio = $_POST['precio'];
$cantidad = $_POST['cantidad'];

$sql = "INSERT INTO productos (nombre, precio, cantidad) VALUES ('$nombre', '$precio', '$cantidad)";
if ($conexion->query($sql) === TRUE) {
  echo "Producto registrado correctamente.<br>";
} else {
  echo "Error: " . $conexion->error;
}
echo '<a href="registrar.php">Registrar otro</a> | <a href="consultar.php">Ver productos</a>';
?>

```

e) Consultar productos (consultar.php)

```

php
<?php
include 'conexion.php';
$resultado = $conexion->query("SELECT * FROM productos");
?>
<!DOCTYPE html>
<html>
<head>
  <title>Lista de Productos</title>
</head>
<body>
  <h2>Productos Registrados</h2>
  <table border="1">
    <tr>
      <th>ID</th>
      <th>Nombre</th>
      <th>Precio</th>
      <th>Cantidad</th>
    </tr>
    <?php while($fila = $resultado->fetch_assoc()) { ?>

```

```

<tr>
  <td><?php echo $fila['id']; ?></td>
  <td><?php echo $fila['nombre']; ?></td>
  <td><?php echo $fila['precio']; ?></td>
  <td><?php echo $fila['cantidad']; ?></td>
</tr>
<?php } ?>
</table>
<a href="registrar.php">Registrar producto</a>
</body>
</html>

```

RESULTADOS ESPERADOS

El formulario registra productos correctamente, la lista de productos se actualiza en tiempo real y se evita el registro de datos vacíos o incorrectos.

ANÁLISIS DE RESULTADOS

Los estudiantes comprenden la importancia de la validación y la interacción entre el frontend (formulario) y el backend (PHP/MySQL).

CONCLUSIONES Y REFLEXIONES

El uso de formularios es fundamental para la captura y gestión de datos en aplicaciones web. Se recomienda implementar validaciones tanto en el cliente (JavaScript) como en el servidor (PHP) así como diseñar formularios claros y accesibles para el usuario.

ACTIVIDADES COMPLEMENTARIAS

EVALUACIÓN Y EVIDENCIAS DE APRENDIZAJE

Criterios de evaluación	
Rúbricas o listas de cotejo para valorar desempeño	Rubrica Reporte de practica
Formatos de reporte de prácticas	Reporte de Práctica

NOMBRE DE LA PRÁCTICA	Generación de reportes dinámicos a partir de la base de datos
COMPETENCIA DE LA PRÁCTICA	Programar la generación de un reporte en PDF o en pantalla que muestre el listado de empleados con filtros por área o salario.

FUNDAMENTO TEÓRICO
La generación de reportes permite analizar y visualizar la información almacenada en la base de datos, apoyando la toma de decisiones en la organización. PHP puede generar reportes en diferentes formatos, como HTML, PDF o CSV.

MATERIALES, EQUIPAMIENTO Y/O REACTIVOS
<ul style="list-style-type: none"> • Computadora de laboratorio • Acceso a internet

PROCEDIMIENTO O METODOLOGÍA
<ul style="list-style-type: none"> • Consultar la base de datos con PHP usando filtros seleccionados por el usuario. • Mostrar los resultados en una tabla HTML o generar un archivo PDF usando una librería como FPDF. • Permitir la descarga o impresión del reporte. <p>Ejemplo: Reporte de empleados filtrado por puesto</p> <p>1. Base de datos y tabla de ejemplo</p> <pre> sql CREATE DATABASE gestion; USE gestion; CREATE TABLE empleados (id INT AUTO_INCREMENT PRIMARY KEY, nombre VARCHAR(100) NOT NULL, puesto VARCHAR(50) NOT NULL, salario DECIMAL(10,2) NOT NULL); -- Datos de ejemplo INSERT INTO empleados (nombre, puesto, salario) VALUES ('Ana López', 'Gerente', 25000), ('Carlos Ruiz', 'Desarrollador', 18000), ('María Pérez', 'Desarrollador', 17500), ('Luis Torres', 'Soporte', 12000); </pre> <p>2. Conexión a la base de datos (conexion.php)</p> <pre> php <?php \$conexion = new mysqli("localhost", "root", "", "gestion"); if (\$conexion->connect_error) { die("Conexión fallida: " . \$conexion->connect_error); } </pre>

```
}
?>
```

3. Formulario de filtro y generación de reporte (reporte.php)

```
php
<?php
include 'conexion.php';

$filtero = "";
if (isset($_POST['puesto']) && $_POST['puesto'] != "") {
    $puesto = $conexion->real_escape_string($_POST['puesto']);
    $filtro = "WHERE puesto = '$puesto'";
}

$sql = "SELECT * FROM empleados $filtro";
$resultado = $conexion->query($sql);
?>

<!DOCTYPE html>
<html>
<head>
    <title>Reporte de Empleados</title>
</head>
<body>
    <h2>Reporte Dinámico de Empleados</h2>
    <form method="post" action="">
        Filtrar por puesto:
        <select name="puesto">
            <option value="">-- Todos --</option>
            <option value="Gerente">Gerente</option>
            <option value="Desarrollador">Desarrollador</option>
            <option value="Soporte">Soporte</option>
        </select>
        <input type="submit" value="Filtrar">
    </form>
    <br>
    <table border="1">
        <tr>
            <th>ID</th>
            <th>Nombre</th>
            <th>Puesto</th>
            <th>Salario</th>
        </tr>
        <?php while($fila = $resultado->fetch_assoc()) { ?>
        <tr>
            <td><?php echo $fila['id']; ?></td>
            <td><?php echo $fila['nombre']; ?></td>
            <td><?php echo $fila['puesto']; ?></td>
            <td><?php echo $fila['salario']; ?></td>
        </tr>
        <?php } ?>
    </table>
```

</body>
</html>

RESULTADOS ESPERADOS

El reporte muestra la información solicitada de manera clara y ordenada y el usuario puede filtrar los datos según criterios definidos.

ANÁLISIS DE RESULTADOS

Los estudiantes logran generar reportes útiles y comprensibles, mejorando la presentación y análisis de datos.

CONCLUSIONES Y REFLEXIONES

Saber generar reportes dinámicos es clave para el análisis y la gestión eficiente de la información. Se debe utilizar librerías especializadas para mejorar la presentación de los reportes y verificar la exactitud de los datos antes de generar el reporte.

ACTIVIDADES COMPLEMENTARIAS

EVALUACIÓN Y EVIDENCIAS DE APRENDIZAJE

Criterios de evaluación

Rúbricas o listas de cotejo para valorar desempeño

Formatos de reporte de prácticas

Rubrica Reporte de practica

Reporte de Práctica

NOMBRE DE LA PRÁCTICA	Integración de funciones de seguridad y control de acceso (sesiones y roles de usuario)
COMPETENCIA DE LA PRÁCTICA	Implementar un sistema de autenticación donde solo los usuarios registrados puedan acceder a ciertas páginas, usando sesiones y roles (administrador, usuario).

FUNDAMENTO TEÓRICO
Las sesiones en PHP permiten mantener información del usuario a lo largo de su navegación, mientras que los roles controlan los permisos de acceso. Esto es fundamental para la seguridad y personalización de aplicaciones web.

MATERIALES, EQUIPAMIENTO Y/O REACTIVOS
<ul style="list-style-type: none"> • Computadora de laboratorio • Acceso a internet

PROCEDIMIENTO O METODOLOGÍA
<ul style="list-style-type: none"> • Crear un formulario de inicio de sesión. • Validar las credenciales contra la base de datos. • Iniciar una sesión PHP y almacenar el rol del usuario. • Restringir el acceso a páginas según el rol. • Implementar la opción de cerrar sesión. <p>Ejemplo: Sistema de login con control de acceso por roles</p> <p>1. Base de datos y tabla de usuarios</p> <p>sql</p> <pre>CREATE DATABASE seguridad; USE seguridad; CREATE TABLE usuarios (id INT AUTO_INCREMENT PRIMARY KEY, usuario VARCHAR(50) NOT NULL, password VARCHAR(255) NOT NULL, rol ENUM('admin','usuario') NOT NULL); -- Contraseñas cifradas con password_hash INSERT INTO usuarios (usuario, password, rol) VALUES ('admin', '\$2y\$10\$elmiTXuWVxfM37uY4JANjQ==', 'admin'), -- Ejemplo, reemplaza por hash real ('juan', '\$2y\$10\$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', 'usuario');</pre> <p>Nota: Genera los hashes de contraseña reales con <code>password_hash('tu_clave', PASSWORD_DEFAULT)</code> en PHP.</p> <p>2. Conexión a la base de datos (conexion.php)</p> <p>php</p> <pre><?php \$conexion = new mysqli("localhost", "root", "", "seguridad"); if (\$conexion->connect_error) { die("Conexión fallida: " . \$conexion->connect_error); }</pre>

```
}
?>
```

3. Formulario de login (login.php)

php

```
<?php
session_start();
$error = "";
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    include 'conexion.php';
    $usuario = $_POST['usuario'];
    $password = $_POST['password'];
    $sql = "SELECT * FROM usuarios WHERE usuario = '$usuario'";
    $res = $conexion->query($sql);
    if ($res->num_rows == 1) {
        $row = $res->fetch_assoc();
        if (password_verify($password, $row['password'])) {
            $_SESSION['usuario'] = $row['usuario'];
            $_SESSION['rol'] = $row['rol'];
            header("Location: dashboard.php");
            exit();
        } else {
            $error = "Contraseña incorrecta.";
        }
    } else {
        $error = "Usuario no encontrado.";
    }
}
}
?>
```

```
<!DOCTYPE html>
<html>
<head><title>Login Seguro</title></head>
<body>
    <h2>Iniciar Sesión</h2>
    <?php if($error) echo "<p style='color:red;'>$error</p>"; ?>
    <form method="post" action="">
        Usuario: <input type="text" name="usuario" required><br>
        Contraseña: <input type="password" name="password" required><br>
        <input type="submit" value="Entrar">
    </form>
</body>
</html>
```

4. Panel de control con control de acceso por rol (dashboard.php)

php

```
<?php
session_start();
if (!isset($_SESSION['usuario'])) {
    header("Location: login.php");
    exit();
}
?>
```

```

<!DOCTYPE html>
<html>
<head><title>Panel de Control</title></head>
<body>
  <h2>Bienvenido, <?php echo $_SESSION['usuario']; ?> (<?php echo $_SESSION['rol']; ?>)</h2>
  <?php if ($_SESSION['rol'] == 'admin') { ?>
    <p>Acceso a funciones de administrador.</p>
    <!-- Aquí puedes poner enlaces a páginas de administración -->
  <?php } else { ?>
    <p>Acceso a funciones de usuario estándar.</p>
  <?php } ?>
  <a href="logout.php">Cerrar sesión</a>
</body>
</html>

```

5. Cerrar sesión (logout.php)

php

```

<?php
session_start();
session_destroy();
header("Location: login.php");
exit();
?>

```

RESULTADOS ESPERADOS

Solo los usuarios autenticados pueden acceder a las páginas protegidas y el sistema diferencia entre roles y muestra opciones según el tipo de usuario.

ANÁLISIS DE RESULTADOS

La mayoría de los estudiantes implementa correctamente la autenticación y el control de acceso, comprendiendo la importancia de la seguridad en aplicaciones web.

CONCLUSIONES Y REFLEXIONES

El control de acceso mediante sesiones y roles es esencial para proteger la información y garantizar la integridad de los sistemas web. Se recomienda nunca almacenar contraseñas en texto plano; usar técnicas de hash y cerrar la sesión correctamente para evitar riesgos de seguridad.

ACTIVIDADES COMPLEMENTARIAS

EVALUACIÓN Y EVIDENCIAS DE APRENDIZAJE

Criterios de evaluación	
Rúbricas o listas de cotejo para valorar desempeño	Rubrica Reporte de practica
Formatos de reporte de prácticas	Reporte de Práctica

FUENTES DE INFORMACIÓN

Libros y manuales:

Borges, E. L., & Puertas, J. P. (2015). Creación de un sitio web con PHP y MySQL (5a ed.). RA-MA Editorial.

Escarcena, M. (2020). Programación páginas web: JavaScript y PHP (1a ed.). RA-MA Editorial. <https://elibro.net/es/ereader/ues/222656?page=1>

López Sanz, M. (2015). Programación web en el entorno servidor. RA-MA Editorial. <https://elibro.net/es/lc/ues/titulos/106487>

Pavón Puertas, J., & Llarena Borges, E. (2015). Creación de un sitio web con PHP y MySQL (5a ed.). RA-MA Editorial. <https://elibro.net/es/lc/ues/titulos/106491>

Pérez Rodríguez, M. D. (Coord.). (2013). Programación de páginas web dinámicas con Apache, Base de Datos MySQL y PHP (2a ed.). Editorial ICB. <https://elibro.net/es/lc/ues/titulos/113229>

Sitios web y documentación:

Documentation. (s.f.). Php.net. <https://www.php.net/docs.php>

Crea mapas mentales online gratis. (s.f.). Lucidchart.com. Recuperado el 13 de junio de 2024, de <https://www.lucidchart.com/pages/es/ejemplos/mapa-mental-online>

NORMAS Y TÉCNICAS APLICABLES

- Estándares de desarrollo web: W3C, PHP-FIG (PSR).
- Normas de seguridad informática: ISO/IEC 27001.
- Buenas prácticas de programación: PSR-1, PSR-2, PSR-12.
- Normas de manejo y disposición de residuos electrónicos: NOM-161-SEMARNAT-2011 (México).
- Reglamento interno del laboratorio institucional.



UES

Universidad Estatal de Sonora
La Fuerza del Saber Estimulará mi Espíritu