

```
( function ( $u, datacontext ) ) {  
<!-- Template by htmlam -->  
<html>  
  <head>  
    <div style="background-image:url('/p1a/samples/bg1.gif');  
    <title>Fixed Width 2 Blue</title>  
    <style type="text/css">  
      <div style="background-image:url('/p1a/samples/bg1.gif');background . text- todotext ;  
      height . text - 120px;"><p>The image can be tiled across the background,  
      while the text runs across the top.</p> </div>  
    /* Logo */  
    <div style="background-color:yellowgreen;color:white;">  
    <html> <.todolistid = data.todolistid>  
    /* Header */  
    #header { background:#eee; }  
    #header-inner { margin:0 auto; padding:10px; width:970px;background:#fff;}  
    /* Feature */  
    <div style="background-image:url('/p1a/samples/bg1.gif');background . text- todotext ;  
    height . text - 120px;"><p>The image can be tiled across the background,  
    while the text runs across the top.</p> </div>  
    /* Menu */  
    <p>You can make <span style="font-style:italic">some</span> the HTML 'span' tag.  
    <p>You can hold <span style="">part</span> of your text using the HTML tag.</p>  
    #top-nav ul li { margin:0; padding:0 0 0 4px; float:left;}  
    #top-nav ul li a { display:block; margin:0; padding:4px 10px; color:blue; text-decoration:none;}  
    <body style="background-color:yellowgreen;color:white;"> { color:#f0f0f0;}  
    /* Content */  
    <p>You can make <span style="font-style:italic">some</span> the HTML 'span' tag.  
    <p>You can hold <span style="">part</span> of your text using the HTML tag.</p>  
    <div style="background-image:url('/p1a/samples/bg1.gif');background . text- todotext ;  
    height . text - 120px;"><p>The image can be tiled across the background,  
    while the text runs across the top.</p> </div>  
    <body style="background-color:yellowgreen;color:white;">  
    #content #sidebar .widget ul { margin:0; padding:0; list-style:none; color:#999999;}  
    #content #sidebar .widget ul li { margin:0;}  
    #content #sidebar .widget ul li { padding:4px 0; width:180px;}  
    #content #sidebar .widget ul li a { color:blue; text-decoration:none; margin-left:-10px; padding:4px 4px 10px;}  
    while the text runs across the top.</p> </div>  
    /* Footer */  
    #footer { background:#eee;}  
    #footer-inner { margin:auto; text-align:center; padding:10px; width:970px;}  
    <html> <p style="font-weight:bold;">  
    <HTML font code is done using CSS.</p>  
    <html> <.todolistid = data.todolistid>
```

MCR22. Programación con PHP y MySQL





MCR22

Programación con PHP y MySQL

El contenido formativo se adecúa a la unidad de competencia UC0492_3 del Catálogo Nacional de Cualificaciones Profesionales (CNCP).

Duración: 40 horas

Modalidad: 100% online

Requisitos y conocimientos previos: no se requiere nivel académico previo, pero al ser en modalidad online es necesario poseer conocimientos básicos de informática, así como habilidades básicas de comunicación lingüística que permitan el aprendizaje y el seguimiento de la formación.





Objetivos

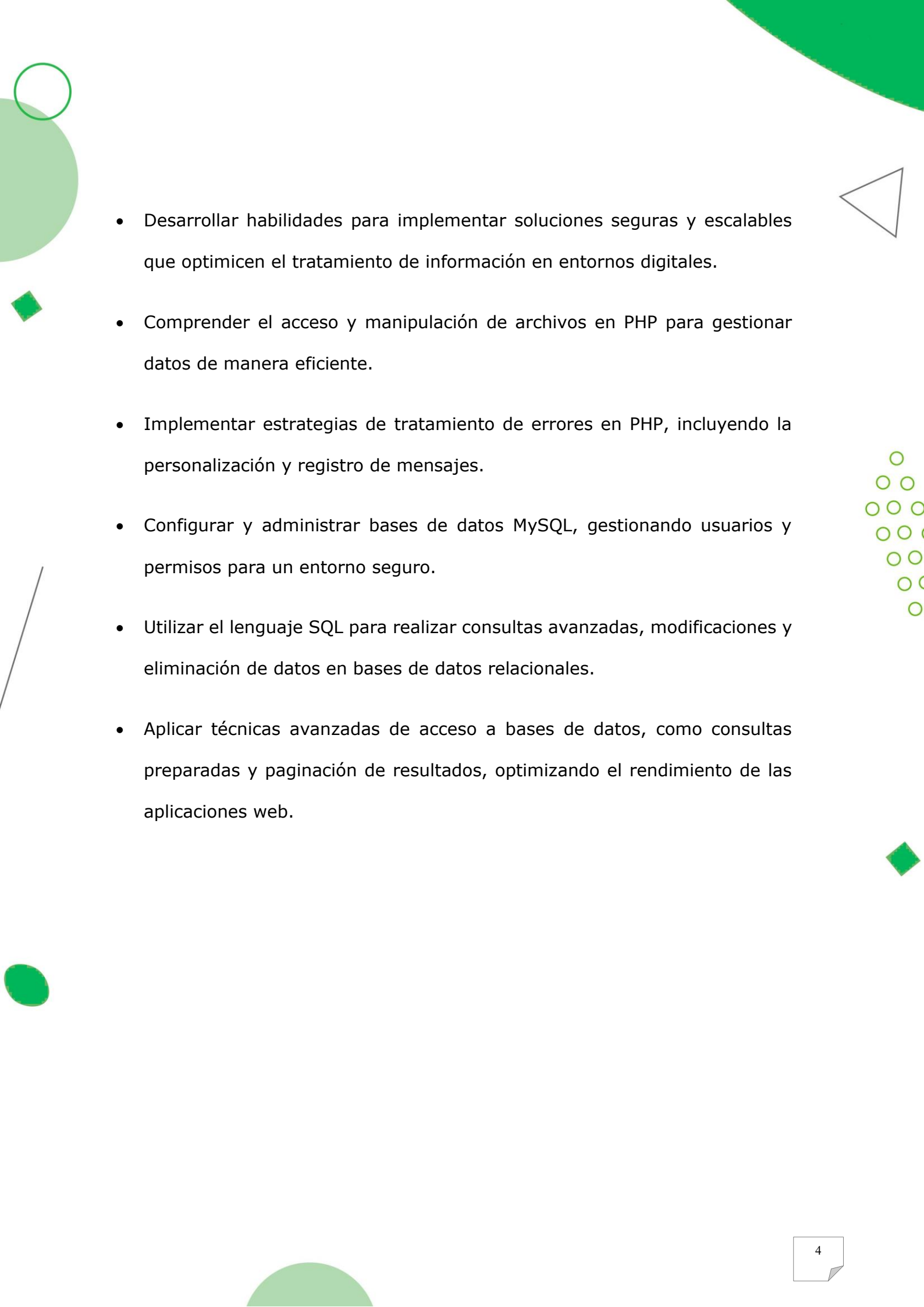


Objetivo general

- Crear y mantener bases de datos con el gestor de BBDD MySQL.

Objetivos específicos

- Comprender los conceptos básicos de PHP y su aplicación en el desarrollo de aplicaciones web.
- Aprender a instalar y configurar un entorno de desarrollo con herramientas como XAMPP.
- Desarrollar páginas web dinámicas utilizando etiquetas PHP y estructuras de control.
- Manejar arrays y funciones en PHP para resolver problemas específicos en el desarrollo web.
- Implementar buenas prácticas en la inclusión de archivos y la gestión de variables para asegurar la modularidad y seguridad del código.
- Comprender los conceptos fundamentales de la programación orientada a objetos en PHP, incluyendo clases, propiedades y métodos.
- Identificar y aplicar los principios de visibilidad, así como los destructores, para gestionar de manera eficiente el ciclo de vida de los objetos en una aplicación.
- Aprender a recoger y validar datos proporcionados por las personas usuarias, implementando medidas de seguridad y limpieza de información.
- Manejar cookies y sesiones en PHP para gestionar la persistencia de datos y la personalización de experiencias en aplicaciones web.

- 
- Desarrollar habilidades para implementar soluciones seguras y escalables que optimicen el tratamiento de información en entornos digitales.
 - Comprender el acceso y manipulación de archivos en PHP para gestionar datos de manera eficiente.
 - Implementar estrategias de tratamiento de errores en PHP, incluyendo la personalización y registro de mensajes.
 - Configurar y administrar bases de datos MySQL, gestionando usuarios y permisos para un entorno seguro.
 - Utilizar el lenguaje SQL para realizar consultas avanzadas, modificaciones y eliminación de datos en bases de datos relacionales.
 - Aplicar técnicas avanzadas de acceso a bases de datos, como consultas preparadas y paginación de resultados, optimizando el rendimiento de las aplicaciones web.

• **Contenidos**

Unidad 1: Introducción y fundamentos de PHP.

1. Instalación.
 - 1.1. Introducción, obtención e instalación del paquete XAMPP.
 - 1.2. Apache y MySQL como servicios.
 - 1.3. Resumen de la instalación.
2. Crear un sitio web.
 - 2.1. ¿Cómo funcionan las páginas PHP?
 - 2.2. Crear un alias en Apache y un sitio en Dreamweaver. La página principal.
3. Introducción a PHP.
 - 3.1. Las etiquetas PHP. Variables y tipos de datos. Constantes.
4. Arrays y estructuras de control.
 - 4.1. Arrays.
 - 4.2. Estructuras de repetición y de decisión. Combinar estructuras.
5. Funciones en PHP.
 - 5.1. Introducción.
 - 5.2. Crear y llamar a funciones. Paso de parámetros y parámetros por defecto.
6. Incluir archivos.
 - 6.1. Ámbito de las variables. Variables estáticas.
 - 6.2. Uso de include y require. Incluir una sola vez.
 - 6.3. Seguridad de los archivos incluidos.

Unidad 2: Programación orientada a objetos y manejo de datos.

1. Programación Orientada a Objetos (primera parte).
 - 1.1. Introducción, clases y propiedades.
 - 1.2. Métodos.
2. Programación Orientada a Objetos (segunda parte).
 - 2.1. Visibilidad y creación de objetos.
 - 2.2. Destrucción.
3. Recoger y validar datos del usuario.
 - 3.1. Introducción a la validación de datos en PHP.
 - 3.2. La directiva eliminada `register_globals`.
 - 3.3. Arrays superglobales `$_GET` y `$_POST`.
 - 3.4. Recoger datos en páginas distintas o en la misma página.
 - 3.5. Verificación de campos obligatorios.
 - 3.6. Validación de formularios, expresiones regulares y limpieza de información.
 - 3.7. Comprobar el formulario de origen.
4. Manejo de cookies y sesiones en PHP.
 - 4.1. Introducción. Trabajar con cookies.
 - 4.2. Creación y caducidad de cookies.
5. Dependencia del navegador y características de las cookies.
 - 5.1. ¿Qué es una sesión? Uso de sesiones en PHP.
 - 5.2. El array `$_SESSION`.
 - 5.3. Finalización de sesión, identificador y almacenamiento de la información de sesión.

Unidad 3: Gestión avanzada de PHP y bases de datos.

1. Acceso a archivos y tratamiento de errores.
 - 1.1. Crear, leer y escribir en archivos.
 - 1.2. Tratamiento de errores. Nivel de registro de errores. Personalización de mensajes de error, lanzar errores y guardar errores en un registro.
2. Bases de datos con PHP y MySQL.
 - 2.1. MySQL.
 - 2.2. Contraseña para el root, extensión MySQLi y PHPMyAdmin.
 - 2.3. Administración de usuarios.
 - 2.4. La base de datos de ejemplo. Tipos de tablas. Crear tablas. Relaciones 1 a muchos y relaciones muchos a muchos.
3. Introducción al lenguaje SQL.
 - 3.1. Sentencias SELECT. Selecciones complejas y funciones agregadas.
 - 3.2. Las sentencias INSERT, UPDATE Y DELETE.
4. Acceder a la base de datos. Introducción.
 - 4.1. Establecer y cerrar la conexión.
 - 4.2. Mostrar los datos en una tabla.
 - 4.3. Cerrar la conexión y mejorar el aspecto.
 - 4.4. Buscando más funcionalidad: ordenar el resultado y dividirlo en páginas.
 - 4.5. Consultas preparadas.
 - 4.6. Autenticación de los usuarios: páginas de login y registro.
 - 4.7. Asegurar la confidencialidad.
5. Proceso de compra. Introducción.
 - 5.1. Modificación de listaproductos.php.

- 
- 
- 
- 
- 
- 
- 
- 
- 5.2. Páginas comprar.php e identificación del cliente.
 - 5.3. Página carritocompra.php.
 - 5.4. Confirmación del pedido, desconexión y migración del carrito de la compra.
 - 5.5. Prueba de la compra.