



Universidad Internacional San Isidro Labrador

(ISB-32) PROGRAMACIÓN AVANZADA.

Profesora: ESTEFANIA BOZA VILLALOBOS.

Estudiante: Efran Romero Fallas.

Avance #3

Proyecto Final:

Sistema Web Universitario.

Año 2025.

## Tabla de contenido

Índice de Tablas. ....	3
Índice de Figuras. ....	4
Información General del Proyecto. ....	6
Introducción. ....	6
Objetivo General. ....	7
Objetivos Específicos. ....	7
Justificación del Proyecto. ....	8
Alcances Esperados. ....	9
Requerimientos del Proyecto. ....	9
Diseño del Sistema (Pre-diseño visual). ....	10
Base de datos. ....	14
Tablas: ....	14
Conexión con la aplicación. ....	16
Backend del Sistema Web Universitario. ....	16
Conexión con la Base de Datos. ....	17
Comunicación con el Frontend. ....	20
Conclusiones. ....	21
Referencias. ....	23

## Índice de Tablas.

Tabla 1 Tabla Usuarios.

Campo	Tipo de dato	Descripción
id	INT(11)PK (Primary Key)	Identificador único del usuario.
username	VARCHAR(100)	Nombre de usuario (correo o usuario para iniciar sesión).
PASSWORD	VARCHAR(255)	Contraseña encriptada.
fullname	VARCHAR(150)	Nombre completo del usuario.
created_at	TIMESTAMP	Fecha de creación del usuario.

Tabla 2 Tabla Cursos.

Campo	Tipo de dato	Descripción
id	INT(11) PK (Primary Key)	Identificador único de la matrícula.
user_id	INT(11) FK (Foreign Key)	Hace referencia al usuario matriculado (users.id).
course_id	INT(11)	Identificador del curso (seguramente relacionado a una tabla courses, aunque no aparece en la imagen).
enrolled_at	TIMESTAMP	Fecha y hora de la inscripción.

# Índice de Figuras.

Ilustración 1 Login Usuarios.

### Iniciar sesión

Usuario

Contraseña

Entrar

[Crear cuenta](#)

Regresar al menú principal

Ilustración 2 DashBoard.



## Mis Cursos Matriculados

Aún no estás matriculado en ningún curso. [Ver cursos disponibles](#)

---



## Cursos Disponibles para Matricular

Programación

Matricularse

Matemáticas

Matricularse

Bases de Datos

Matricularse

Redes

Matricularse

Inteligencia Artificial

Matricularse

Ilustración 3 Primary Key entre tablas.

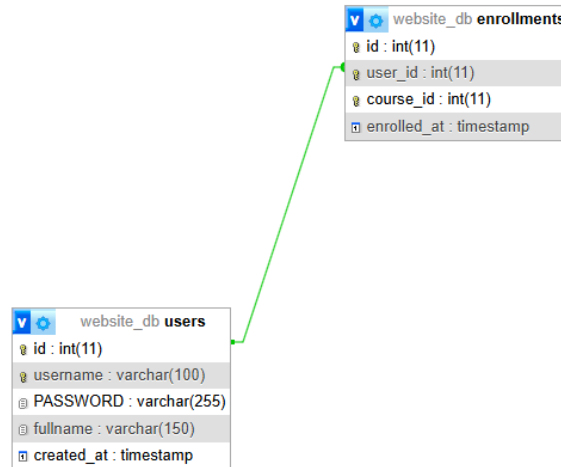


Ilustración 4 Obtener lista de estudiantes código.

```
<?php
header("Content-Type: application/json");
require_once "../config/Database.php";
$db = new Database();
$conn = $db->connect();
$sql = "SELECT * FROM estudiantes";
$stmt = $conn->prepare($sql);
$stmt->execute();
$estudiantes = $stmt->fetchAll(PDO::FETCH_ASSOC);
echo json_encode([
    "status" => "success",
    "data" => $estudiantes
]);
?>
```

# Información General del Proyecto.

## Introducción.

En el mundo actual, la gestión de información constituye un elemento fundamental para el funcionamiento eficiente de cualquier organización. La capacidad de almacenar, procesar y consultar datos en tiempo real se ha convertido en un requisito indispensable para la toma de decisiones oportunas y acertadas. Tradicionalmente, muchas instituciones han manejado sus datos de forma manual o mediante herramientas básicas, lo que genera errores, duplicidad de información y pérdida de tiempo en procesos administrativos.

Con la llegada de las tecnologías de la información y la comunicación (TIC), las aplicaciones web se han posicionado como una solución accesible, flexible y escalable para resolver estas problemáticas. En este contexto, el desarrollo de sistemas con tecnologías de código abierto como PHP y MySQL ofrece una alternativa de bajo costo, altamente adaptable y con una amplia comunidad de soporte, lo cual facilita su implementación y mantenimiento.

Este proyecto plantea la construcción de un sistema de información basado en la web, diseñado para gestionar datos de manera eficiente y segura. A través de una interfaz intuitiva y un backend robusto, el sistema busca optimizar procesos internos, reducir tiempos de operación y ofrecer a los usuarios una plataforma confiable para el manejo de la información. Además, se pretende sentar las bases para futuras integraciones y ampliaciones que respondan a las necesidades cambiantes de la organización, implementando los mejores fundamentos de programación.

## Objetivo General.

Desarrollar un sistema de información que facilite la gestión, consulta y mantenimiento de datos en una plataforma web dinámica, utilizando PHP como lenguaje de programación y MySQL con XAMPP como sistema de gestión de bases de datos.

## Objetivos Específicos.

1. Diseñar la arquitectura del sistema que permita la correcta interacción entre usuarios, base de datos y servicios internos.
2. Implementar una base de datos normalizada que garantice integridad, consistencia y seguridad de la información.
3. Desarrollar las funcionalidades del backend utilizando un enfoque modular y escalable.
4. Establecer una interfaz gráfica intuitiva que mejore la experiencia de usuario.

## Justificación del Proyecto.

La necesidad de contar con sistemas de información confiables y eficientes es cada vez mayor en un entorno caracterizado por la digitalización de procesos y la alta demanda de acceso a la información. Muchas organizaciones, especialmente las de tamaño pequeño y mediano, enfrentan el reto de modernizar sus procesos internos para mantenerse competitivas y responder con agilidad a las demandas del mercado.

En este escenario, un sistema desarrollado con PHP y MySQL representa una solución viable y estratégica. PHP es un lenguaje de programación ampliamente utilizado en el desarrollo web por su flexibilidad, facilidad de aprendizaje y compatibilidad con múltiples plataformas. MySQL, por su parte, es un sistema de gestión de bases de datos relacional robusto y seguro, que garantiza la integridad y consistencia de los datos.

La implementación de este proyecto no solo permitirá centralizar la información en una plataforma única y accesible, sino también mejorar la eficiencia operativa mediante la automatización de tareas repetitivas, la reducción de errores humanos y la disponibilidad de datos en tiempo real. Asimismo, la escalabilidad del sistema permitirá incorporar nuevas funcionalidades en el futuro, como reportes avanzados, integración con otros sistemas o módulos especializados, asegurando la sostenibilidad tecnológica a largo plazo.

En conclusión, este proyecto se justifica por la necesidad de modernizar la gestión de información, aprovechar herramientas tecnológicas de bajo costo y ofrecer a los usuarios una solución que potencie la productividad y la toma de decisiones estratégicas dentro de la organización.



## Alcances Esperados.

1. El sistema permitirá registrar, consultar y gestionar la información almacenada en la base de datos mediante una interfaz web.
2. Generación de reportes y estadísticas para la toma de decisiones.
3. Acceso a la información en tiempo real, facilitando la gestión de datos desde diferentes ubicaciones.
4. Integración con módulos futuros según las necesidades de la organización, garantizando la escalabilidad del sistema.

## Requerimientos del Proyecto.

### Requerimientos funcionales:

1. Registro de usuarios y autenticación.
2. Gestión de información mediante operaciones CRUD (Crear, Leer, Actualizar, Eliminar).
3. Generación de reportes personalizados.
4. Envío de notificaciones y alertas básicas en pantalla.

### Requerimientos no funcionales:

1. Seguridad en el acceso y en el manejo de datos.
2. Buen rendimiento en la respuesta del sistema.
3. Escalabilidad para permitir futuras ampliaciones.
4. Interfaz amigable y fácil de usar.

### Infraestructura mínima:

1. Servidor web con soporte para PHP (Apache/Nginx).
2. Sistema gestor de base de datos MySQL.
3. Navegador web actualizado para la interacción de usuarios.
4. Frameworks de apoyo como Bootstrap para interfaz, librerías PHP para seguridad y consultas.

## Diseño del Sistema (Pre-diseño visual).

El diseño presentado corresponde a un pre-diseño visual del sistema web universitario, el cual muestra de manera general cómo se espera que luzca la interfaz antes de su implementación definitiva. Este diseño busca reflejar la estructura y estilo visual que guiarán el desarrollo del sistema, manteniendo la coherencia con los objetivos del proyecto y la experiencia de usuario deseada.

## Elementos de Diseño.

Estructura de pantallas:

El sistema contará con una organización visual clara y jerárquica de sus pantallas principales. Iniciará con una pantalla de inicio de sesión (login) donde los usuarios podrán autenticarse de manera segura mediante su correo institucional y contraseña. Una vez validados, accederán al panel principal o dashboard, el cual presentará un resumen general del sistema y accesos directos a los módulos más importantes.

Con el menú de navegación lateral o superior, se podrá moverse con facilidad y de manera intuitiva entre las diferentes áreas, tales como gestión de usuarios, matrícula, reportes, configuración y soporte.

Para mejorar la experiencia del usuario, cada módulo incorporará formularios de actualización y registro con campos claramente definidos, mensajes de validación y botones de acción que sean visibles.

Además, se incorporarán pantallas de consulta y reportes que presentarán la información en tablas dinámicas y que tendrán opciones para filtrar, imprimir y exportar datos con el objetivo de respaldar el control administrativo y la toma de decisiones.

#### Colores:

La paleta cromática se basará principalmente en tonos turquesa y gris claro, complementados con acentos en blanco y negro para lograr contraste y equilibrio. El color turquesa aportará una sensación de frescura, tecnología y confianza, mientras que el gris claro brindará sobriedad y profesionalismo. Esta combinación permitirá mantener una interfaz moderna, limpia y coherente visualmente en todos los módulos del sistema.

#### Tipografía:

Se utilizarán fuentes modernas, legibles y compatibles con navegadores actuales, tales como Segoe UI, Roboto o Arial. Estas tipografías se seleccionan por su claridad en entornos digitales y su adecuada lectura en distintos tamaños de pantalla, garantizando accesibilidad tanto en dispositivos de escritorio como móviles. El uso consistente de tipografía contribuirá a la identidad visual del sistema y mejorará la experiencia del usuario final.

#### Imágenes e íconos:

Con el fin de fortalecer la percepción visual de las funciones, se aplicará una iconografía minimalista y representativa. Cada módulo tendrá iconos que representen su propósito; por ejemplo, un gráfico para el módulo de reportes o un ícono de usuario para la administración de cuentas.

Se utilizarán bibliotecas estándar como Lucide-React o FontAwesome, las cuales proporcionan un extenso repertorio de íconos escalables y personalizables, para asegurar una apariencia homogénea y profesional en toda la interfaz. Además, las imágenes empleadas en el sistema serán ligeras y optimizadas para optimizar el desempeño sin sacrificar la calidad visual.

#### Slogan:

El sistema podría incorporar el lema “Tu información académica, siempre a tu alcance”. El cual resume la misión del proyecto: simplificar las tareas administrativas y garantizar la protección de la información. Este slogan refuerza la identidad del sistema, transmitiendo confianza y eficiencia a los usuarios institucionales.

## Explicación.

El pre-diseño visual sirve como una guía que permitirá a los usuarios y al equipo de desarrollo comprender la estructura y apariencia del sistema antes de su construcción. Este esquema inicial facilita la validación del diseño, la coherencia con los requerimientos funcionales y la identificación temprana de mejoras en la experiencia de usuario.

Imágenes de referencia:

## Index:

En este inicio de la portada web, se utiliza este panel como referencia, donde tendrá un apartado para lo que sería la institución, los cursos disponibles para matricular. Aparte de un apartado para contactar a soporte, y recibir ayuda con cualquier duda que le pueda surgir al usuario.

## Login de Usuarios:

La pantalla de inicio de sesión representa el primer punto de contacto entre el usuario y el sistema. Su diseño prioriza la claridad, seguridad y facilidad de uso, asegurando que el acceso al sistema sea rápido e intuitivo. En esta sección, el usuario deberá ingresar sus credenciales institucionales, nombre de usuario o correo y contraseña, para autenticarse correctamente. Se implementarán mecanismos de validación de campos y mensajes de error en caso de credenciales incorrectas o campos vacíos.

Visualmente, el formulario de login estará centrado en la pantalla y enmarcado en un contenedor con bordes suaves y colores acordes a la paleta institucional (turquesa y gris claro). El fondo puede incluir el logotipo de la universidad o una imagen tenue que refuerce la identidad visual del sistema.

Además, se incluirán elementos adicionales como:

1. Un enlace para recuperar contraseña en caso de olvido.
2. Botones claramente diferenciados para “Iniciar sesión” y, si se habilita, “Matricular cursos” o “Registrarse”.
3. En versiones futuras, la posibilidad de autenticación mediante roles (estudiantes, profesores, administradores), mostrando diferentes paneles según el tipo de usuario.

### Dashboard:

Tras el inicio de sesión exitoso, el usuario será dirigido al Dashboard o panel principal, que funcionará como el centro de control del sistema. Desde aquí se podrá visualizar un resumen general de la información más relevante y acceder a los diferentes módulos mediante un menú lateral o superior de navegación.

El dashboard mostrará indicadores clave del sistema, como:

1. Cantidad de usuarios registrados.
2. Cursos o materias activas.
3. Reportes recientes o notificaciones.
4. Accesos rápidos a tareas frecuentes (por ejemplo, registrar datos, generar reportes o modificar información).

Visualmente, se organizará mediante tarjetas informativas con íconos representativos, colores suaves y tipografía clara, siguiendo el diseño moderno establecido. Cada tarjeta o bloque podrá incluir animaciones ligeras o resaltados al pasar el cursor, para mejorar la interacción y la usabilidad.

El objetivo del panel principal es ofrecer una vista general del sistema que facilite la toma de decisiones y la navegación intuitiva, permitiendo al usuario acceder rápidamente a las funciones más utilizadas. Además, servirá como base para futuras integraciones, como gráficos dinámicos, estadísticas o notificaciones en tiempo real.

## Base de datos.

### Tablas:

Tabla: users (usuarios):

La tabla users representa a los usuarios registrados dentro del sistema. Su objetivo es almacenar la información necesaria para que una persona pueda autenticarse e interactuar con el sistema. Esta tabla funciona como entidad principal para el módulo de usuarios.

Tabla: enrollments (matrículas / inscripciones):

La tabla enrollments se utiliza para relacionar usuarios con cursos (o módulos del sistema, dependiendo del contexto). Funciona como entidad intermedia en una relación de muchos a muchos, donde:

- Un usuario puede estar inscrito en varios cursos.
- Un curso puede tener varios usuarios inscritos.

### Llaves primarias y foráneas.

Para garantizar integridad y consistencia en los datos, se definieron llaves primarias (PK) y llaves foráneas (FK).

### Llaves primarias (Primary Key).

En la tabla users, la llave primaria es id. Esto garantiza que cada usuario registrado sea único y fácilmente identificable. En la tabla enrollments, la llave primaria también es id, garantizando que cada registro de matrícula sea único. Las llaves primarias utilizan AUTO\_INCREMENT, lo cual facilita el registro de información sin necesidad de insertar manualmente un identificador.

## Llaves foráneas (Foreign Key).

La tabla enrollments contiene dos llaves foráneas:

user\_id → hace referencia a users.id

course\_id → hace referencia a la tabla courses (esto permite enlazar usuarios con cursos)

La razón de usar llaves foráneas es:

Mantener la integridad referencial entre las tablas, evitando inscripciones inválidas o usuarios inexistentes.

Gracias a esto:

- Un usuario puede tener varias matrículas (relación 1:N).
- Un curso puede tener varios usuarios inscritos.

Relación entre tablas (lo que muestra la línea verde)

- users.id → enrollments.user\_id

Es decir, un usuario puede tener muchas matrículas, pero cada matrícula pertenece a un solo usuario.

## Conexión con la aplicación.

La conexión entre la aplicación web y la base de datos se implementa mediante el lenguaje PHP, utilizando el motor de base de datos MySQL administrado con phpMyAdmin. Para gestionar dicha conexión se utiliza la tecnología PDO (PHP Data Objects), ya que permite realizar consultas de forma segura mediante sentencias preparadas y facilita el manejo de excepciones en caso de errores.

PDO es la opción recomendada en proyectos modernos por su capacidad de trabajar con distintos motores de base de datos, ofrecer mayor seguridad contra ataques de inyección SQL y permitir una programación más estructurada orientada a objetos.

## Backend del Sistema Web Universitario.

El backend del Sistema Web Universitario constituye la capa central donde se ejecutan todos los procesos internos del sistema, integrando los elementos definidos en las fases previas: análisis, diseño, arquitectura y modelo de base de datos. Esta etapa es fundamental, ya que garantiza que las funcionalidades del sistema operen de manera confiable, segura y eficiente.

El backend fue desarrollado bajo principios de modularidad, escalabilidad, seguridad, limpieza de código y eficiencia, de manera que cada módulo cumpla responsabilidades específicas sin generar dependencia excesiva entre ellos. Esto facilita la mantenibilidad del sistema y permite la incorporación de nuevas funciones sin alterar los componentes ya implementados.

Además, esta capa actúa como puente entre el frontend (interfaz gráfica) y la base de datos, procesando solicitudes, validaciones, cálculos y respuestas mediante API endpoints, asegurando así una comunicación robusta y organizada.



## Arquitectura en Capas

- Capa de Acceso a Datos (DAL):  
Contiene las clases responsables de ejecutar consultas SQL, almacenar procedimientos almacenados y gestionar la conexión con la base de datos. Garantiza una comunicación eficiente con SQL Server/MySQL (según el proyecto).
- Capa de Lógica de Negocio (BLL):  
Implementa las reglas de negocio para estudiantes, cursos, usuarios, matrículas y roles.  
Aquí se validan los datos, se aplican reglas y se coordinan operaciones complejas.
- Capa de Servicios / API (Controllers):  
Expone los endpoints accesibles desde el frontend.  
Todas las operaciones CRUD viajan a través de controladores organizados por entidad.

## Conexión con la Base de Datos.

La conexión con la base de datos se implementó respetando los principios de seguridad y eficiencia. Para ello se utilizó:

- Cadenas de conexión protegidas mediante archivo de configuración (config.php).
- Uso de PDO o MySQLi con consultas preparadas para evitar inyecciones SQL.
- Implementación de procedimientos almacenados para manipulación masiva o crítica de datos.
- Manejo de excepciones para capturar errores y mantener la integridad del sistema.

## Lógica de Negocio Implementada.

Cada módulo del sistema contiene un archivo o clase que administra todas las reglas de negocio necesarias para garantizar la coherencia del sistema.

### Módulo de Usuarios y Autenticación

#### Registro de usuarios

- Inicio de sesión con validación de credenciales
- Hash y verificación de contraseñas (password\_hash y password\_verify)
- Control de roles: Administrador, Profesor y Estudiante
- Sesiones seguras

#### Módulo de Estudiantes

- Formatos de nombres y cédulas
- Emails únicos y válidos
- Restricciones en edición y eliminación
- Integración con matrículas

#### Módulo de Cursos

- El sistema gestiona:
- Creación de cursos con código único
- Validación de cupos
- Relación entre cursos y profesores

## Seguridad del Backend

El backend fue construido bajo lineamientos modernos de seguridad para minimizar vulnerabilidades comunes:

### Medidas Implementadas

- Uso exclusivo de consultas preparadas para evitar SQL Injection.
- Hash de contraseñas con algoritmos de última generación.
- Gestión de sesiones segura utilizando tokens y expiración automática.
- Validación exhaustiva en backend, incluso si el frontend valida.
- Control de permisos según el rol del usuario autenticado.
- Reglas CORS para impedir comunicaciones no autorizadas.

### Protección de Endpoints.

- Todos los controladores de la API verifican:
- Que el usuario esté autenticado.
- Que dicho usuario cuente con permisos suficientes para realizar la operación.
- Que los datos enviados cumplan con el formato esperado.

## Comunicación con el Frontend.

El backend se comunica con el frontend mediante API REST, utilizando:

JSON para intercambio de datos.

Métodos HTTP estándar:

- GET → obtener datos
- POST → crear
- PUT → actualizar
- DELETE → eliminar

## Conclusiones.

El desarrollo del sistema de información constituye una respuesta integral a las necesidades actuales de modernización y optimización de los procesos organizacionales. La propuesta presentada no solo aborda la centralización y administración eficiente de los datos, sino que también integra principios de usabilidad y accesibilidad que potencian la interacción entre los usuarios y la plataforma web. De esta manera, el proyecto contribuye a fortalecer la estructura tecnológica de la organización y a mejorar sus dinámicas operativas.

La selección de PHP como lenguaje de programación y MySQL como gestor de bases de datos se presenta como una decisión acertada desde el punto de vista técnico, económico y funcional. Estas tecnologías poseen una amplia trayectoria en el ámbito del desarrollo web, ofrecen estabilidad y cuentan con una gran comunidad de soporte. Asimismo, su carácter open source favorece la sostenibilidad del proyecto, permitiendo un desarrollo escalable, adaptable y libre de costos asociados a licenciamientos.

La arquitectura del sistema propuesta evidencia un diseño orientado a la eficiencia, la modularidad y la interacción coherente entre los distintos componentes de la plataforma. Esta estructura facilita el flujo de información entre los usuarios, el backend y la base de datos, reduciendo posibles inconsistencias y promoviendo un procesamiento de datos más seguro y confiable. En consecuencia, se sientan las bases para un sistema sólido que responda a las demandas operativas de manera oportuna.

La implementación de una base de datos normalizada se consolida como uno de los pilares fundamentales del proyecto, dado que garantiza la integridad, consistencia y correcta organización de la información. Un modelo de datos estructurado adecuadamente minimiza errores, evita redundancias y fortalece la seguridad de la información almacenada. Esto resulta especialmente relevante en sistemas donde la precisión de los datos es crucial para la toma de decisiones estratégicas.

El enfoque orientado hacia una interfaz gráfica intuitiva y amigable contribuye significativamente a la experiencia del usuario final. Una interfaz bien diseñada no solo facilita la interacción con el sistema, sino que también incrementa la eficiencia con la que se gestionan los datos, independientemente del nivel de conocimiento técnico de los usuarios. Este aspecto resulta esencial para garantizar la adopción del sistema dentro de la organización y promover su uso continuo.

Los alcances esperados y los requerimientos funcionales y no funcionales definidos reflejan la intención de construir un sistema robusto, escalable y alineado con las mejores prácticas de desarrollo de software. La inclusión de características como la escalabilidad, la seguridad, el rendimiento y la modularidad permiten prever la incorporación futura de nuevas funcionalidades, módulos o integraciones con otros sistemas institucionales, asegurando la vigencia y adaptabilidad del proyecto en el largo plazo.

Finalmente, el proyecto se posiciona como una herramienta que contribuye a la mejora continua de los procesos internos de la organización, gracias a la automatización de tareas y la disponibilidad de información en tiempo real. Elementos como la generación de reportes, la reducción de errores, la eficiencia en el manejo de datos y la capacidad de consulta inmediata fortalecen la toma de decisiones estratégicas y aumentan la productividad global. En este sentido, la implementación del sistema no solo beneficia las operaciones actuales, sino que también prepara a la organización para futuros retos tecnológicos.

## Referencias.

Albornoz, D., & Albornoz, D. (2024, 3 septiembre). ¿Qué es PHP? Aprende todo sobre el lenguaje de scripting. Tutoriales Hostinger. <https://www.hostinger.com/co/tutoriales/que-es-php>

Beautiful Replacement for Javascript Popups by KeenThemes. (s. f.). Good By Keenthemes. <https://preview.keenthemes.com/html/good/docs/general/sweetalert>

colaboradores de Wikipedia. (2025, 10 noviembre). PHP. Wikipedia, la Enciclopedia Libre. <https://es.wikipedia.org/wiki/PHP>

Contributors, M. o. J. T. A. B. (s. f.-a). Bootstrap. <https://getbootstrap.esdocu.com/>

Contributors, M. o. J. T. A. B. (s. f.-b). Get started with Bootstrap. <https://getbootstrap.com/docs/5.3/getting-started/introduction/>

DSouza, D., & DSouza, D. (2025, 24 abril). How to create an XAMPP MySQL Database in 9 Easy Steps. Learn | Hevo. <https://hevodata.com/learn/xampp-mysql/>

Eddy. (2025, 5 junio). Set up and work with MySQL on Windows with XAMPP. SHIFT ASIA | Dev Blog. <https://shiftasia.com/community/set-up-and-work-with-mysql-on-window-with-xampp/>

¿Qué es el backend? Definición y explicación. (2024, 7 junio). IONOS Digital Guide. <https://www.ionos.com/es-us/digitalguide/paginas-web/creacion-de-paginas-web/que-es-el-backend/>

JustBlossom. (2013, 24 julio). How do I use MySQL through XAMPP? Stack Overflow. <https://stackoverflow.com/questions/17823189/how-do-i-use-mysql-through-xampp>

Qué es el Backend de una web y por qué es tan importante. (s. f.). Rafa Arjonilla. <https://rafarjonilla.com/que-es/backend/>

Suárez, D. (2025, 27 agosto). Qué es Bootstrap y cómo usarlo. Raiola Networks - Dominios y Alojamiento Web de Calidad. <https://raiolanetworks.com/blog/bootstrap/>

Wwlpublish. (s. f.). Creación de páginas web con HTML y CSS para principiantes - Training. Microsoft Learn. <https://learn.microsoft.com/es-es/training/paths/build-web-pages-html-css-for-beginners/>