

UNIVERSIDAD NACIONAL DE SAN AGUSTÍN
FACULTAD DE INGENIERIA DE PRODUCCIÓN Y SERVICIOS
ESCUELA PROFESIONAL DE INGENIERIA DE SISTEMAS



SISTEMA APRENDIENDO
PLATAFORMA EDUCATIVA DE DESARROLLO WEB

CURSO:
INTRODUCCIÓN AL DESARROLLO WEB

DOCENTE:
MARCO WILFREDO AEDO LOPEZ

PRESENTADO POR

Hancco Vargas Mark

AREQUIPA – PERÚ
2025

ÍNDICE

1. Resumen Ejecutivo
2. Introducción
3. Descripción del Proyecto
 1. Objetivos
 2. Alcance
 3. Tecnologías Utilizadas
4. Arquitectura del Sistema
 1. Frontend
 2. Backend
 3. Base de Datos
5. Funcionalidades Implementadas
 1. Frontend HTML y CSS
 2. JavaScript Interactivo
 3. Backend Python
 4. Base de Datos MySQL
6. Capturas de Pantalla
7. Manual de Usuario
 1. Requisitos del Sistema
 2. Instalación
 3. Uso del Sistema
8. Trabajo Futuro
9. Conclusiones
10. Bibliografía

1. RESUMEN EJECUTIVO

El Sistema APRENDIENDO es una plataforma educativa completa diseñada para facilitar el aprendizaje autodidacta de desarrollo web. El proyecto implementa una solución integral que combina contenido educativo estructurado con funcionalidades web modernas, incluyendo autenticación de usuarios, gestión de mensajes y una interfaz interactiva y responsiva.

El sistema está desarrollado utilizando tecnologías web fundamentales: HTML5, CSS3 y JavaScript para el frontend, Python puro (sin frameworks) para el backend, y MySQL para la persistencia de datos. La arquitectura modular permite una fácil expansión y mantenimiento del código.

El proyecto cuenta con 137 páginas HTML de contenido educativo organizado en categorías (Fundamentos, Lenguajes, Frameworks y Herramientas), componentes JavaScript interactivos, y un backend completo con autenticación y gestión de sesiones.

2. INTRODUCCIÓN

El desarrollo web ha evolucionado significativamente en los últimos años, convirtiéndose en una habilidad esencial en el mundo digital actual. El Sistema APRENDIENDO nace como respuesta a la necesidad de contar con una plataforma educativa completa, local y estructurada que permita el aprendizaje progresivo de las tecnologías web.

Este proyecto representa la culminación del curso de Desarrollo Web, integrando conocimientos de diseño frontend, programación backend, gestión de bases de datos y buenas prácticas de desarrollo. A diferencia de plataformas en línea que requieren conexión constante a internet, APRENDIENDO funciona completamente de manera local, garantizando acceso permanente al contenido educativo.

La plataforma está diseñada siguiendo principios de usabilidad, accesibilidad y diseño responsivo, asegurando una experiencia óptima en diferentes dispositivos. La implementación del backend en Python puro demuestra comprensión profunda de los protocolos HTTP y manejo de peticiones sin depender de frameworks de alto nivel.

Este proyecto nace con una filosofía de educación abierta y colaborativa. Su objetivo principal es que cualquier persona interesada en desarrollo web pueda aprender de manera gratuita y accesible. El proyecto es de código abierto (open source), lo que significa que toda la comunidad está invitada a contribuir con mejoras, correcciones, nuevos contenidos o funcionalidades. Esta apertura no solo enriquece el proyecto con diversas perspectivas y conocimientos, sino que también fomenta el aprendizaje colaborativo y el espíritu de compartir conocimiento que caracteriza a la comunidad de desarrollo de software.

3. DESCRIPCIÓN DEL PROYECTO

3.1 Objetivos

- Desarrollar una plataforma educativa completa para el aprendizaje de desarrollo web
- Implementar un sistema de autenticación seguro con gestión de sesiones
- Crear una interfaz de usuario intuitiva y responsiva

- Estructurar contenido educativo de forma modular y escalable
- Demostrar comprensión de arquitecturas cliente-servidor
- Aplicar buenas prácticas de desarrollo y documentación

3.2 Alcance

El sistema APRENDIENDO incluye:

- **Frontend:** 137 páginas HTML con contenido educativo estructurado
- **Estilos:** Sistema de diseño consistente con 7 archivos CSS modulares
- **Interactividad:** 4 componentes JavaScript para mejorar la experiencia de usuario
- **Backend:** Servidor HTTP implementado en Python puro
- **Base de Datos:** Sistema de gestión de usuarios y mensajes con MySQL
- **Autenticación:** Sistema completo de login/logout con sesiones persistentes
- **Contenido:** Material educativo sobre Fundamentos, Lenguajes, Frameworks y Herramientas

3.3 Tecnologías Utilizadas

Categoría	Tecnología	Versión/Especificación
Estructura	HTML	HTML5
Estilos	CSS	CSS3 (Flexbox, Grid)
Interactividad	JavaScript	ES6+
Backend	Python	3.x (http.server)
Base de Datos	MySQL	8.0+
Control de Versiones	Git	2.x

4. ARQUITECTURA DEL SISTEMA

4.1 Frontend

El frontend está estructurado en una arquitectura modular que separa el contenido educativo de los componentes de la aplicación. Se utilizan layouts modernos con Flexbox y Grid para garantizar una presentación responsiva en todos los dispositivos.

Estructura de Directorios Frontend

```

APRENDIENDO/
├── index.html           (Página de entrada)
├── home.html           (Dashboard principal)
├── login.html          (Autenticación)
├── register.html       (Registro de usuarios)
├── mensajes.html       (Panel de mensajes protegido)
├── assets/
│   ├── css/           (Estilos modulares)
│   │   ├── sidebar.css
│   │   ├── glosario.css
│   │   └── ejercicios.css

```

	—	toc.css	
	—	pages.css	
	—	fundamentos.css	
	—	keyboard-navigation.css	
	—	js/	(Componentes JavaScript)
	—	—	sidebar-component.js
	—	—	toc-component.js
	—	—	study-highlight.js
	—	—	keyboard-navigation.js
—	Fundamentos/		(Contenido: 4 módulos)
—	Lenguajes/		(Contenido: HTML, CSS, JS, Python, etc.)
—	Frameworks/		(Contenido: 14 frameworks)
—	Herramientas/		(Contenido: Git, Docker, etc.)

Sistema de Diseño

El sistema utiliza un tema oscuro minimalista con los siguientes componentes:

- **Colores:** Paleta dark (#0f0f0f, #1a1a1a, #2a2a2a) con acentos en gradiente púrpura (#667eea → #764ba2)
- **Tipografía:** Fuentes sans-serif para lectura óptima en pantalla
- **Layouts:** Flexbox para componentes, Grid para contenido
- **Responsividad:** Breakpoints para mobile (< 768px), tablet (768px-1024px) y desktop (> 1024px)

4.2 Backend

El backend está implementado en Python puro utilizando el módulo `http.server` de la biblioteca estándar, sin dependencias de frameworks externos. Esto demuestra comprensión profunda del protocolo HTTP y manejo manual de rutas y peticiones.

Estructura Backend

backend/	
—	server.py (Servidor HTTP principal)
—	db_config.py (Configuración y operaciones BD)
—	schema.sql (Script de creación de BD)
—	README.md (Documentación técnica)

Componentes del Backend

server.py - Servidor HTTP

Implementa un servidor HTTP personalizado con las siguientes características:

- Routing manual para peticiones GET y POST
- Servicio de archivos estáticos

- Gestión de sesiones con cookies
- Autenticación de usuarios
- APIs RESTful para operaciones CRUD
- Logging con timestamps

Rutas Implementadas

Método	Ruta	Descripción
GET	/	Página principal (index.html)
GET	/login.html	Formulario de login
GET	/mensajes.html	Panel de mensajes (protegido)
POST	/api/login	Autenticación de usuarios
POST	/api/register	Registro de nuevos usuarios
POST	/api/contacto	Guardar mensaje de contacto
POST	/api/mensaje/marcar-leido	Actualizar estado de mensaje
GET	/api/logout	Cerrar sesión

db_config.py - Gestión de Base de Datos

Módulo que centraliza todas las operaciones con MySQL:

- **Conexión:** Pool de conexiones a MySQL con manejo de errores
- **Usuarios:** Verificación de credenciales, creación, actualización
- **Sesiones:** Creación, validación y cierre de sesiones
- **Mensajes:** Guardado, consulta y actualización de mensajes de contacto

4.3 Base de Datos

La base de datos MySQL está diseñada con tres tablas principales que manejan la autenticación, sesiones y mensajes del sistema.

Esquema de Base de Datos

Tabla: usuarios

Campo	Tipo	Descripción
id	INT (PK, AUTO_INCREMENT)	Identificador único
username	VARCHAR(50) UNIQUE	Nombre de usuario
password_hash	VARCHAR(255)	Contraseña hasheada (SHA-256)
email	VARCHAR(100) UNIQUE	Correo electrónico
nombre_completo	VARCHAR(100)	Nombre completo del usuario
rol	VARCHAR(20)	Rol en el sistema (default:

Campo	Tipo	Descripción
fecha_creacion	TIMESTAMP	'usuario') Fecha de creación (default: CURRENT_TIMESTAMP)

Tabla: mensajes

Campo	Tipo	Descripción
id	INT (PK, AUTO_INCREMENT)	Identificador único
nombre	VARCHAR(100)	Nombre del remitente
email	VARCHAR(100)	Email del remitente
asunto	VARCHAR(200)	Asunto del mensaje
mensaje	TEXT	Contenido del mensaje
ip_address	VARCHAR(45)	IP del remitente
user_agent	TEXT	Navegador utilizado
leido	BOOLEAN	Estado de lectura
fecha	TIMESTAMP	Fecha de envío

Tabla: sesiones

Campo	Tipo	Descripción
id	INT (PK, AUTO_INCREMENT)	Identificador único
usuario_id	INT (FK)	Referencia a usuarios
token	VARCHAR(255) UNIQUE	Token de sesión criptográfico
ip_address	VARCHAR(45)	IP de la sesión
user_agent	TEXT	Navegador de la sesión
fecha_creacion	TIMESTAMP	Fecha de creación (default: CURRENT_TIMESTAMP)
fecha_expiracion	TIMESTAMP	Fecha de expiración (24 horas)
activa	TINYINT(1)	Estado de la sesión (default: 1)

Datos de Prueba

La base de datos incluye usuarios de prueba precargados:

Usuario	Contraseña	Rol
admin@aprendiendo.com	test123	admin
usuario@aprendiendo.com	usuario123	usuario

5. FUNCIONALIDADES IMPLEMENTADAS

5.1 Frontend HTML y CSS

Páginas Principales

- **index.html:** Página de entrada con navbar y logo con gradiente
- **home.html:** Dashboard principal con sidebar colapsable y navegación estructurada
- **login.html:** Formulario de autenticación con validación client-side
- **register.html:** Formulario de registro con validación de contraseña
- **mensajes.html:** Panel protegido para visualizar mensajes recibidos

Contenido Educativo (137 páginas)

- **Fundamentos:** 35+ páginas sobre Diseño, Frontend, Backend y Base de Datos
- **Lenguajes:** 70+ páginas con glosarios de HTML (62 términos), CSS (70+ términos), JavaScript (60+ términos), Python, TypeScript, SQL, etc.
- **Frameworks:** 14 directorios con documentación de React, Vue, Angular, Django, Flask, Express, Next.js, Svelte, Astro, Bootstrap, Tailwind, Laravel, Spring Boot
- **Herramientas:** Documentación de Git/GitHub (45 comandos), Docker, Node.js

Sistema de Estilos

- **Tema Dark:** Paleta de colores oscuros (#0f0f0f, #1a1a1a, #2a2a2a)
- **Gradientes:** Acentos púrpura (#667eea → #764ba2)
- **Responsive:** Adaptación a mobile, tablet y desktop
- **Layouts Modernos:** Flexbox para componentes, CSS Grid para contenido

5.2 JavaScript Interactivo

Se implementaron 4 componentes JavaScript que superan el requisito mínimo de 2 funcionalidades:

1. Sidebar Colapsable (sidebar-component.js)

- Menú lateral interactivo que se expande y colapsa
- Persistencia del estado en localStorage
- Navegación jerárquica por categorías
- Animaciones suaves de transición

2. Tabla de Contenidos Dinámica (toc-component.js)

- Generación automática de tabla de contenidos
- Extracción de headings (h1, h2, h3) del documento
- Links de navegación a secciones
- Indicador visual de posición actual durante scroll
- Estilos responsivos

3. Sistema de Resaltado (study-highlight.js)

- Permite resaltar bloques de contenido importantes
- Bloques editables para personalización
- Guardado automático en localStorage
- Diseñado para mejorar la experiencia de estudio

4. Navegación por Teclado (keyboard-navigation.js)

- Navegación entre páginas usando teclas Espacio y Shift+Espacio
- Optimizado para flujo de estudio continuo
- Indicadores visuales de navegación disponible

5.3 Backend Python

El backend está implementado completamente en Python puro usando el módulo `http.server` de la biblioteca estándar, sin frameworks externos como Django o Flask.

Características del Servidor

- **Routing Manual:** Implementación de enrutamiento personalizado para peticiones GET y POST
- **Archivos Estáticos:** Servicio de HTML, CSS, JavaScript e imágenes
- **Gestión de Sesiones:** Sistema de sesiones con cookies seguras
- **Autenticación:** Login/logout con validación de credenciales
- **APIs RESTful:** Endpoints JSON para operaciones CRUD
- **Logging:** Registro de peticiones con timestamps
- **Manejo de Errores:** Respuestas HTTP apropiadas (200, 401, 404, 500)
-

Flujo de Autenticación

1. Usuario envía credenciales a `/api/login`
2. Backend verifica credenciales en base de datos
3. Si son válidas, genera un token de sesión único
4. Guarda la sesión en tabla sesiones con tiempo de expiración
5. Envía cookie con el token al cliente
6. Peticiones subsecuentes incluyen el token para verificación
7. Logout elimina la sesión de la base de datos

Seguridad Implementada

- Validación de entrada de usuario (sanitización básica)
- Protección de rutas con verificación de sesión
- Registro de IP y User-Agent para auditoría
- Sesiones con tiempo de expiración
- Tokens únicos generados con UUID

5.4 Base de Datos MySQL

Funcionalidades de Usuarios

- **Registro:** Creación de nuevos usuarios con validación de datos únicos
- **Login:** Verificación de credenciales (username/email + password)
- **Gestión de Sesiones:** Creación, verificación y cierre de sesiones
- **Roles:** Sistema de roles (admin/usuario) para futuras restricciones

Funcionalidades de Mensajes

- **Guardar Mensajes:** Almacenamiento de mensajes de contacto con metadatos
- **Listar Mensajes:** Consulta de mensajes con filtros (leídos/no leídos)
- **Marcar como Leído:** Actualización del estado de mensajes
- **Estadísticas:** Conteo de mensajes por estado

Operaciones CRUD Implementadas

Operación	Función	Descripción
Create	crear_usuario()	Insertar nuevo usuario
Create	guardar_mensaje()	Insertar nuevo mensaje
Create	crear_sesion()	Crear nueva sesión
Read	verificar_credenciales()	Consultar usuario por username
Read	obtener_usuario_por_id()	Consultar usuario por ID
Read	obtener_mensajes()	Listar mensajes con filtros
Read	verificar_sesion()	Validar token de sesión
Update	marcar_mensaje_leido()	Actualizar estado de mensaje
Delete	cerrar_sesion()	Eliminar sesión activa

6. CAPTURAS DE PANTALLA

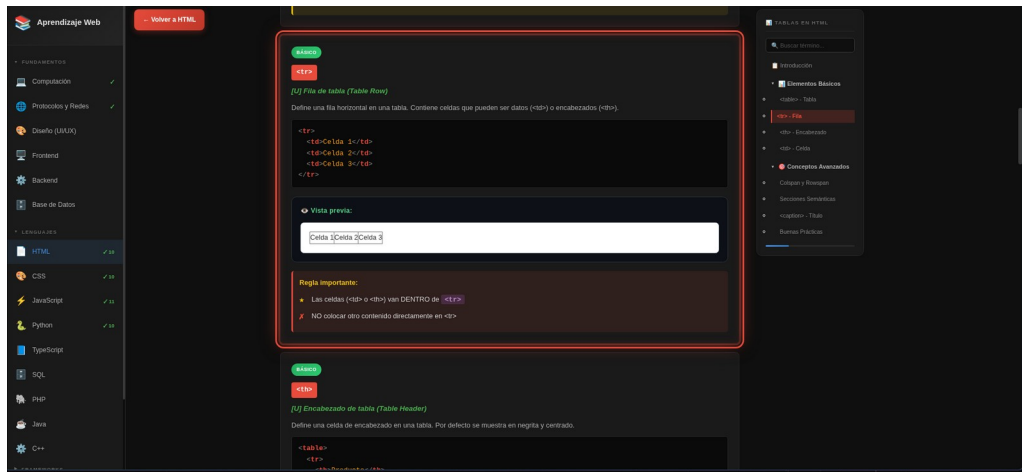
6.1 Página Principal

Figura 6.1: Captura de pantalla de la página principal (home.html)



6.2 Glosario de Lenguajes

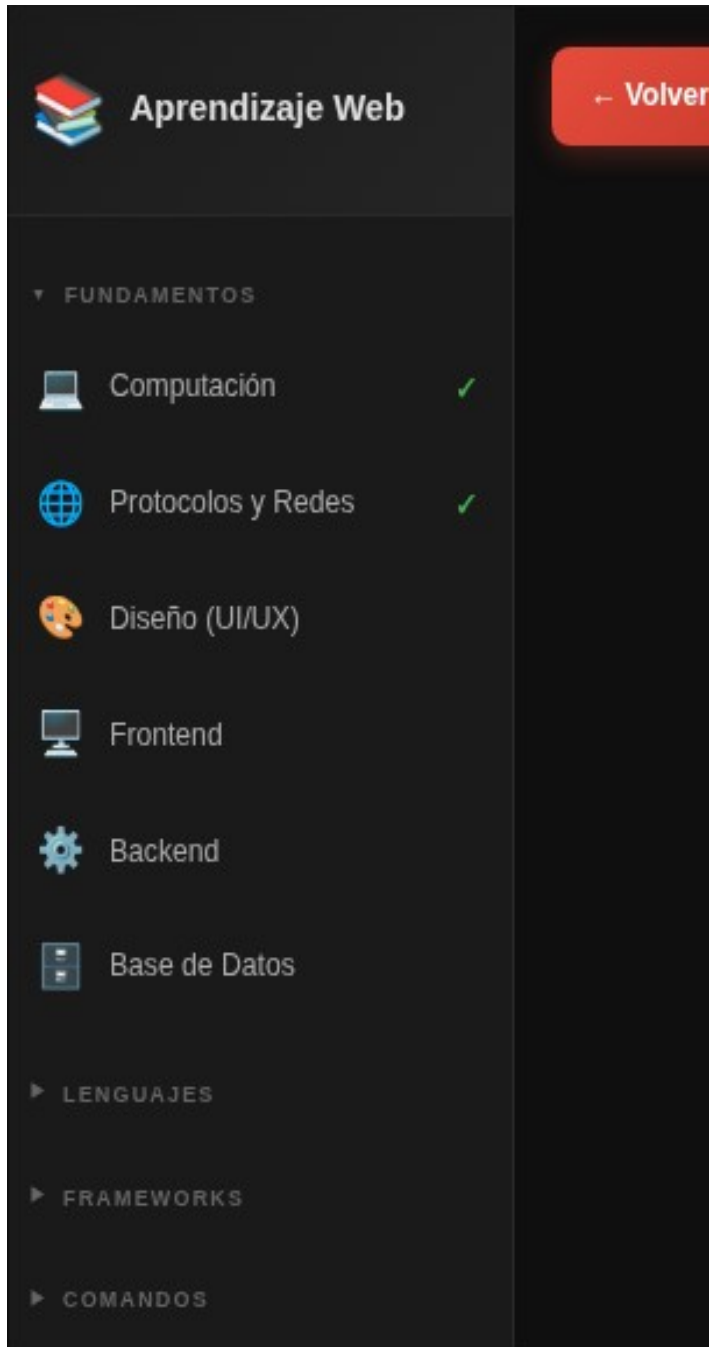
Figura 6.2: Ejemplo de glosario mostrando términos y definiciones



6.3


Sidebar Interactivo

Figura 6.3: Menú lateral (sidebar) con navegación expandida



6.4 Formulario de Login

Figura 6.4: Formulario de autenticación de usuarios



APRENDIENDO

Inicia sesión para acceder al sistema

Correo Electrónico

Contraseña

Iniciar Sesión

¿No tienes cuenta? [Crear cuenta](#)

[← Volver a la portada](#)

6.5 Panel de Mensajes

Figura 6.5: Vista del panel de mensajes protegido

Mensajes Recibidos

Usuario autenticado

Cancelar Sesión

TOTAL DE MENSAJES

3

MENSAJES LEIDOS

0

MENSAJES NUEVOS

3

Todos

No Leídos

Leídos

Implementación Completada

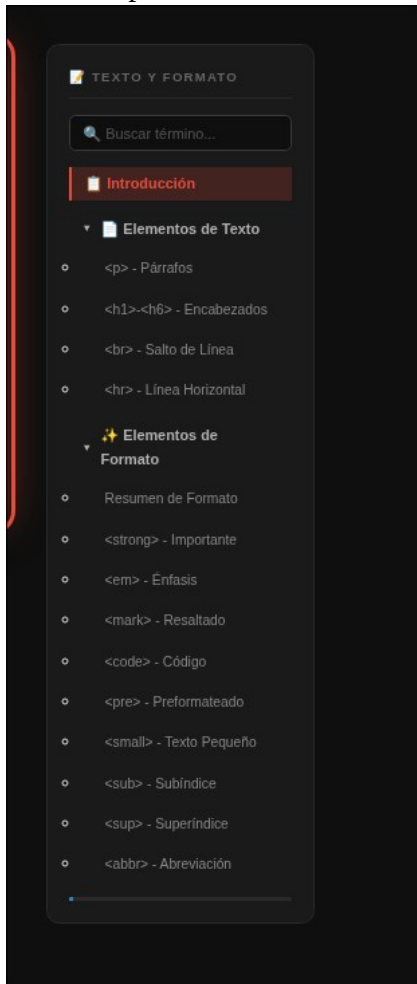
Esta es la página principal de mensajes. Para recibir los mensajes reales, necesitas agregar un proveedor (p.ej. messages) en el servicio, que consulte la base de datos usando el método `db.collection.messages.findOne()`.

Funcionalidades implementadas:

- ✓ Autenticación con servicio
- ✓ Manejo de esta página (login)
- ✓ Contador de mensajes completos
- ✓ Lista de mensajes con filtros
- ✓ Estadísticas
- ✓ Carga de mensajes desde BD (requiere endpoint API)

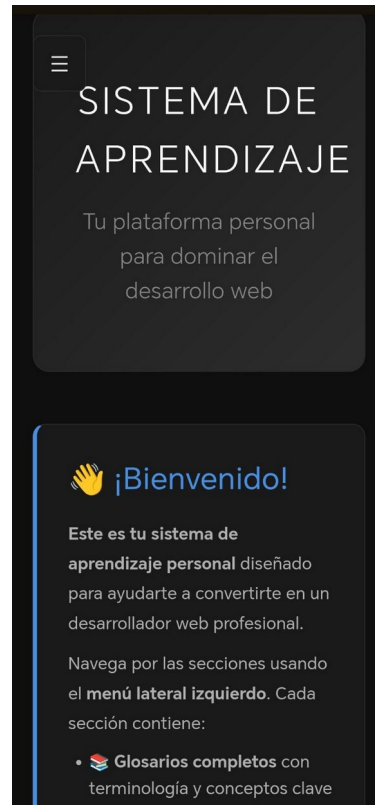
6.6 Tabla de Contenidos Dinámica

Figura 6.6: Componente de tabla de contenidos generado automáticamente



6.7 Responsive Design

Figura 6.7: Vista del sistema en diferentes dispositivos



7. MANUAL DE USUARIO

7.1 Requisitos del Sistema

Hardware Mínimo

- Procesador: 1 GHz o superior
- RAM: 2 GB mínimo
- Espacio en disco: 100 MB
- Conexión a internet: No requerida (sistema local)

Software Requerido

- Python 3.7 o superior
- MySQL 8.0 o superior

- Navegador web moderno (Chrome, Firefox, Edge, Safari)
- Librería Python: mysql-connector-python

7.2 Instalación

Paso 1: Configurar la Base de Datos

1. Iniciar MySQL Server
2. Abrir terminal MySQL o cliente gráfico
3. Ejecutar el script de base de datos:

```
mysql -u root -p < backend/schema.sql
```

Esto creará:

- Base de datos aprendiendo_db
- Tablas usuarios, mensajes y sesiones
- Usuarios de prueba precargados

Paso 2: Instalar Dependencias Python

```
pip install mysql-connector-python
```

Paso 3: Configurar Conexión a Base de Datos

Editar backend/db_config.py si es necesario ajustar los parámetros de conexión:

```
DB_CONFIG = {
    'host': 'localhost',
    'port': 3306,
    'user': 'aprendiendo',
    'password': 'aprendiendo123',
    'database': 'aprendiendo_db',
    'charset': 'utf8mb4'
}
```

Paso 4: Iniciar el Servidor

1. Navegar a la carpeta del proyecto
2. Ejecutar el servidor Python:

```
cd backend
python server.py
```

El servidor iniciará en `http://localhost:8000`

7.3 Uso del Sistema

Navegación Principal

1. Abrir navegador web
2. Acceder a `http://localhost:8000`
3. Se mostrará la página principal con el menú de navegación
4. Usar el sidebar para explorar las diferentes secciones:

- **Fundamentos:** Conceptos básicos de desarrollo web
- **Lenguajes:** Glosarios de HTML, CSS, JavaScript, etc.
- **Frameworks:** Documentación de frameworks populares
- **Herramientas:** Guías de Git, Docker, etc.

Registro e Inicio de Sesión

1. Hacer clic en "Registro" o "Login" en el navbar
2. Para registro:
 - Completar formulario con username, email, password y nombre completo
 - Hacer clic en "Registrar"
 - Si el registro es exitoso, será redirigido al login
3. Para login:
 - Ingresar username o email y contraseña
 - Hacer clic en "Iniciar Sesión"
 - Si las credenciales son correctas, será redirigido a la página principal

Uso de Componentes Interactivos

Sidebar Colapsable

- Hacer clic en el botón de menú (≡) para expandir/colapsar
- El estado se guarda automáticamente
- Navegar por las categorías organizadas jerárquicamente

Navegación por Teclado

- **Espacio:** Ir a la siguiente página
- **Shift + Espacio:** Ir a la página anterior

Sistema de Resaltado

- Seleccionar texto importante en las páginas de estudio
- Hacer clic en el botón de resaltado
- El contenido resaltado se guarda automáticamente

Tabla de Contenidos

- Aparece automáticamente en páginas con múltiples secciones
- Hacer clic en cualquier sección para navegar directamente
- Indicador visual muestra la sección actual mientras se hace scroll

Cerrar Sesión

1. Hacer clic en "Logout" en el navbar
2. La sesión se cerrará y será redirigido a la página de login

7.4 Solución de Problemas

Problema: El servidor no inicia

- Verificar que Python 3.7+ está instalado: `python --version`
- Verificar que el puerto 8000 no está en uso
- Revisar los logs en la consola para mensajes de error

Problema: Error de conexión a base de datos

- Verificar que MySQL está corriendo
- Verificar credenciales en `db_config.py`
- Verificar que la base de datos `aprendiendo_db` existe
- Verificar que `mysql-connector-python` está instalado

Problema: Login no funciona

- Verificar conexión a base de datos
- Usar credenciales de prueba: `admin@aprendiendo.com/test123` o `usuario@aprendiendo.com/usuario123`
- Revisar consola del navegador para errores JavaScript
- Revisar logs del servidor Python

Problema: Páginas no se cargan correctamente

- Limpiar caché del navegador
- Verificar que todos los archivos CSS y JS están presentes
- Abrir consola del navegador (F12) para ver errores

8. TRABAJO FUTURO

8.1 Mejoras de Contenido

- Completar glosarios de Python, TypeScript y SQL
- Agregar más ejercicios prácticos interactivos
- Incluir videos tutoriales embebidos
- Agregar ejemplos de código ejecutables en el navegador
- Expandir documentación de frameworks modernos

8.2 Funcionalidades de Usuario

- Sistema de progreso de aprendizaje por usuario
- Marcadores personalizados en páginas favoritas
- Notas y anotaciones por usuario
- Historial de páginas visitadas
- Perfil de usuario con estadísticas

8.3 Gamificación

- Sistema de puntos por completar módulos
- Badges y logros por hitos de aprendizaje
- Tablas de clasificación (leaderboards)
- Desafíos y quizzes interactivos
- Recompensas por racha de estudio diario

8.4 Mejoras Técnicas

- Implementar HTTPS con certificados SSL
- Mejorar seguridad con hashing de contraseñas (bcrypt/argon2)
- Agregar rate limiting para prevenir ataques
- Implementar sistema de caché para mejor rendimiento
- Optimizar imágenes con lazy loading
- Agregar PWA (Progressive Web App) para uso offline

8.5 Panel de Administración

- Dashboard de administrador con estadísticas
- Gestión de usuarios (activar/desactivar, roles)
- Gestión de contenido (agregar/editar/eliminar páginas)
- Visualización de mensajes de contacto
- Logs de actividad del sistema
- Exportación de datos a CSV/PDF

8.6 Internacionalización

- Soporte para múltiples idiomas (Español, Inglés, Portugués)
- Selector de idioma en navbar
- Traducción de contenido educativo
- Localización de fechas y formatos

8.7 Integración con Herramientas Externas

- Integración con GitHub para proyectos prácticos
- Conectar con plataformas de práctica (LeetCode, HackerRank)
- API pública para consumir contenido desde otras aplicaciones
- Exportar progreso a LinkedIn Learning o similar

8.8 Mobile App

- Desarrollar aplicación móvil nativa (React Native o Flutter)
- Sincronización de progreso entre web y móvil
- Notificaciones push para recordatorios de estudio
- Modo offline completo en móvil

9. CONCLUSIONES

El desarrollo del Sistema APRENDIENDO ha permitido consolidar conocimientos fundamentales de desarrollo web en sus tres capas principales: frontend, backend y base de datos. A través de este proyecto se logró implementar una arquitectura completa que demuestra comprensión profunda de los protocolos web, manejo de estado, persistencia de datos y diseño de interfaces.

La decisión de implementar el backend en Python puro, sin frameworks, representó un desafío técnico significativo que resultó en una comprensión más profunda del protocolo HTTP, manejo de peticiones, routing y gestión de sesiones. Esta aproximación de bajo nivel proporciona una base sólida para trabajar posteriormente con frameworks de alto nivel como Django o Flask.

El frontend desarrollado demuestra aplicación de principios modernos de diseño web: responsividad, accesibilidad, componentización y separación de responsabilidades. Los componentes JavaScript interactivos mejoran significativamente la experiencia de usuario, transformando un sitio estático en una aplicación web dinámica.

La base de datos MySQL está diseñada siguiendo principios de normalización, integridad referencial y optimización con índices apropiados. El esquema permite escalabilidad futura para agregar nuevas funcionalidades como roles avanzados, permisos granulares y auditoría completa.

El contenido educativo estructurado en 137 páginas HTML representa un recurso valioso para el aprendizaje autodidacta. La organización modular por categorías (Fundamentos, Lenguajes, Frameworks, Herramientas) facilita la navegación y el aprendizaje progresivo.

Los principales logros del proyecto incluyen:

- Sistema de autenticación completo con gestión de sesiones persistentes
- Interfaz responsiva que funciona correctamente en todos los dispositivos
- Componentes JavaScript reutilizables y modulares
- Backend robusto con manejo apropiado de errores
- Base de datos normalizada con integridad referencial
- Documentación técnica exhaustiva
- Control de versiones con Git

Las áreas de mejora identificadas incluyen la implementación de hashing seguro de contraseñas, mejora en la validación de entrada de usuario, implementación de HTTPS, y expansión del contenido educativo en lenguajes como Python y TypeScript.

Este proyecto no solo cumple con los requisitos académicos del curso, sino que representa una plataforma funcional que puede continuar desarrollándose y mejorándose. Las bases arquitecturales están sólidas para implementar las funcionalidades futuras propuestas en la sección 8.

En conclusión, el Sistema APRENDIENDO demuestra competencia en desarrollo web full-stack, desde el diseño de interfaces hasta la implementación de lógica de negocio y persistencia de datos. El proyecto sirve tanto como portafolio técnico como herramienta educativa práctica para el aprendizaje continuo de tecnologías web.

Finalmente, es importante destacar que este proyecto está concebido como una iniciativa de

código abierto con fines educativos. La misión es democratizar el acceso al conocimiento de desarrollo web, permitiendo que cualquier persona, independientemente de su ubicación o recursos económicos, pueda aprender y crecer como desarrollador. Se invita a toda la comunidad de desarrolladores, educadores y estudiantes a contribuir al proyecto: ya sea agregando nuevo contenido educativo, mejorando el código existente, reportando errores, sugiriendo nuevas funcionalidades o traduciendo el contenido a otros idiomas. Cada contribución, por pequeña que sea, ayuda a construir una mejor herramienta de aprendizaje para todos. El futuro del proyecto depende de la colaboración abierta y el compromiso compartido de hacer que la educación en tecnología sea accesible para todos.

10. BIBLIOGRAFÍA

Documentación Oficial

1. Mozilla Developer Network (MDN). (2025). *HTML: HyperText Markup Language*. <https://developer.mozilla.org/en-US/docs/Web/HTML>
2. Mozilla Developer Network (MDN). (2025). *CSS: Cascading Style Sheets*. <https://developer.mozilla.org/en-US/docs/Web/CSS>
3. Mozilla Developer Network (MDN). (2025). *JavaScript*. <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
4. Python Software Foundation. (2025). *Python Documentation*. <https://docs.python.org/3/>
5. Python Software Foundation. (2025). *http.server — HTTP servers*. <https://docs.python.org/3/library/http.server.html>
6. Oracle Corporation. (2025). *MySQL 8.0 Reference Manual*. <https://dev.mysql.com/doc/refman/8.0/en/>

Libros de Referencia

7. Duckett, J. (2014). *HTML and CSS: Design and Build Websites*. John Wiley & Sons.
8. Haverbeke, M. (2018). *Eloquent JavaScript: A Modern Introduction to Programming* (3rd ed.). No Starch Press.
9. Lutz, M. (2013). *Learning Python* (5th ed.). O'Reilly Media.
10. Beighley, L., & Morrison, M. (2009). *Head First SQL*. O'Reilly Media.

Recursos Web

11. W3C. (2025). *Web Content Accessibility Guidelines (WCAG) 2.1*. <https://www.w3.org/WAI/WCAG21/quickref/>
12. CSS-Tricks. (2025). *A Complete Guide to Flexbox*. <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
13. CSS-Tricks. (2025). *A Complete Guide to Grid*. <https://css-tricks.com/snippets/css/complete-guide-grid/>
14. Can I Use. (2025). *Browser Support Tables for Modern Web Technologies*. <https://caniuse.com/>

Tutoriales y Cursos

15. freeCodeCamp. (2025). *Responsive Web Design Certification*. <https://www.freecodecamp.org/>
16. The Odin Project. (2025). *Full Stack JavaScript Path*. <https://www.theodinproject.com/>
17. Real Python. (2025). *Python Web Development Tutorials*. <https://realpython.com/>

Herramientas y Tecnologías

18. Git. (2025). *Git Documentation*. <https://git-scm.com/doc>
19. Visual Studio Code. (2025). *Documentation for Visual Studio Code*. <https://code.visualstudio.com/docs>
20. MySQL Connector/Python. (2025). *MySQL Connector/Python Developer Guide*. <https://dev.mysql.com/doc/connector-python/en/>