# INTRODUCCIÓN

El centro educativo implementará una plataforma de formación en línea autoalojada para ofrecer cursos especializados en Database Management (en inglés) y Redes de Ordenadores, utilizando exclusivamente infraestructura propia. La plataforma estará desarrollada con tecnologías ligeras y eficientes como PHP nativo, MySQL y servidores Apache bajo Ubuntu Server, garantizando total control sobre los datos y contenidos. Los materiales didácticos se crearán en formato Markdown, permitiendo incluir ejemplos interactivos de código SQL y configuraciones de red, con un sistema de visualización optimizado para enseñanza técnica. La arquitectura contempla dos servidores físicos principales -uno para el alojamiento web y otro para la base de datos- configurados con medidas de seguridad robustas como firewalls y certificados SSL autogestionados. Esta solución, diseñada para soportar hasta 100 conexiones simultáneas, ofrecerá ventajas clave frente a plataformas externas: menor costo operativo a largo plazo, personalización absoluta de los contenidos y adaptación específica a los requerimientos pedagógicos del centro. El proyecto priorizará la usabilidad para profesores y estudiantes, con funcionalidades como búsqueda semántica en los materiales y resaltado de sintaxis para código técnico

# ESTADO DEL ARTE

Este proyecto no representa un reto técnico elevado porque ya se han realizado trabajos similares anteriormente. En cuanto a **lenguajes de marcas**, se domina perfectamente HTML, CSS y JavaScript, que son más que suficientes para toda la parte frontend. Para el backend se usará PHP, que aunque no es un lenguaje que se conozca profundamente, se entienden sus conceptos básicos y será suficiente para este desarrollo.

En **bases de datos** se ha estudiado SQL, MySQL y PL/SQL, aunque con solo MySQL bastará para este proyecto. Respecto a **redes**, se tiene experiencia en:

* Configuración de routers y switches
* Medios de comunicación y cableado
* Creación de VLANs
* Conexión de redes LAN a Internet
* Administración básica de redes

En el ámbito de **hardware**, se sabe exactamente cómo montar un servidor físico desde cero. Además, en cuanto a **software**, se conocen:

* Los sistemas operativos necesarios
* Los programas requeridos
* La configuración básica de servidores web
* La gestión de almacenamiento

Todos estos conocimientos se han adquirido mediante proyectos prácticos previos como:

1. Montaje y configuración de equipos informáticos
2. Creación de páginas web funcionales
3. Implementación de bases de datos
4. Configuración de redes locales

La única novedad será trabajar con Markdown para los contenidos educativos, pero al ser un formato sencillo y existiendo experiencia previa con HTML, no supondrá mayor dificultad. Tampoco se prevén problemas con PHP, ya que se ha trabajado con lógica de programación similar en otros lenguajes.

# ESTUDIO DE VIABILIDAD

Se evalúa si el proyecto es viable o no, teniendo en cuenta aspectos técnicos, económicos y legales.

## Método DAFO/CAME.

El proyecto presenta un equilibrio favorable entre capacidades técnicas existentes y desafíos superables. Contamos con experiencia demostrada en desarrollo web (HTML5, CSS3, JavaScript) y administración de redes, lo que cubre los aspectos fundamentales. La infraestructura física del centro educativo elimina costos iniciales significativos. El principal reto es la curva de aprendizaje de PHP, aunque se compensa con conocimientos previos en lógica de programación.

**Fortalezas:**

* Dominio comprobado de tecnologías front-end básicas
* Infraestructura física disponible en el centro
* Costos operativos reducidos gracias al software open-source
* Experiencia práctica en configuración de redes LAN

**Debilidades:**

* Conocimiento limitado de PHP (aunque con base en otros lenguajes)
* Dependencia inicial de un único desarrollador
* Capacidad limitada de escalamiento en fase inicial

**Oportunidades:**

* Creciente demanda de formación técnica especializada
* Posibilidad de reutilizar la infraestructura para otros cursos
* Ventaja competitiva al ser plataforma institucional autoalojada

**Amenazas:**

* Competencia de plataformas consolidadas con mayor financiación
* Posibles problemas técnicos no previstos en fase de implementación
* Cambios en estándares tecnológicos

## Estudio de mercado.

El análisis del sector educativo online muestra un crecimiento sostenido del 25% anual en cursos técnicos especializados. Las plataformas globales (Udemy, Coursera) dominan el mercado generalista pero presentan limitaciones para instituciones educativas: costos elevados, falta de personalización curricular y dependencia de infraestructura externa. Nuestra solución ofrece ventajas diferenciadoras:

* Contenidos adaptados específicamente al plan de estudios
* Control total sobre datos y usuarios al evitar terceros
* Integración con recursos educativos existentes del centro
* Costos predecibles al eliminar licencias externas

## Viabilidad técnica y económica

**Viabilidad Técnica :**  
El stack tecnológico propuesto (LAMP) es robusto y adecuado para los requerimientos:

* **Frontend:** HTML5, CSS3 y JavaScript vanilla (dominados)
* **Backend:** PHP 8.2 + Parsedown para Markdown
* **Base de datos:** MySQL 8.0 (experiencia previa suficiente)
* **Servidores:** Ubuntu Server 22.04 LTS (conocimiento disponible)

Los documentos formativos (.md) se almacenarán directamente en el servidor web bajo la ruta /var/www/cursos/, organizados por categorías, con permisos seguros que permiten solo acceso mediante la aplicación PHP.

**Viabilidad Económica:**

* **Inversión inicial:** €800 (actualización hardware)
* **Costes mensuales:** €90 (mantenimiento + electricidad)
* **Retorno esperado:** 24 meses (proyección conservadora)
* **Ventaja económica:** Elimina costos recurrentes de plataformas externas

## Recursos HW

**Servidor Principal (Web + Contenidos):**

* Procesador: Intel Xeon E-2234 (4 núcleos / 8 hilos)
* Memoria: 16GB DDR4 ECC
* Almacenamiento: 512GB SSD (sistema + aplicaciones) + 2TB HDD (/var/www/cursos/)
* Red: Dual Gigabit Ethernet
* Backup: NAS con replicación diaria

**Servidor de Base de Datos:**

* Procesador: Intel Xeon E-2236 (6 núcleos / 12 hilos)
* Memoria: 32GB DDR4 ECC
* Almacenamiento: 1TB SSD en RAID 1
* Red: Dual Gigabit Ethernet (conexión dedicada al servidor web)

**Red:**

* Switch gestionable 24 puertos (VLAN support)
* Router empresarial básico
* Sistema UPS para protección eléctrica

## Recursos SW

**Stack Principal:**

* Sistema Operativo: Ubuntu Server 22.04 LTS
* Servidor Web: Apache 2.4 + PHP-FPM 8.2
* Base de Datos: MySQL 8.0 Community Edition
* Librerías Clave: Parsedown (para procesamiento de Markdown)

**Herramientas Desarrollo:**

* Entorno: VS Code + Git
* Monitorización: NetData + Fail2Ban
* Backup: Automatizado con rsync + cron

## Recursos humanos

Según el tema de Recursos Humanos vamos a necesitar dos personas principales que se van a encargar de desarrollar y mantener nuestra plataforma igual que los dispositivos físicos.

**Desarrollador Principal (Full-Stack):**

* **Responsabilidades técnicas:**
  + Desarrollo frontend (HTML5, CSS3, JavaScript)
  + Implementación backend (PHP + MySQL)
  + Configuración inicial del servidor web (Apache/Ubuntu)
  + Gestión de archivos .md y estructura de contenidos
  + **Configuración básica de red:**
    - Asignación de IPs locales
    - Configuración de reglas básicas de firewall (UFW)
    - Pruebas de conectividad servidor-web/servidor-DB
  + Documentación técnica completa

**Soporte TI del Centro:**

* **5 horas semanales de apoyo**
* **Áreas de soporte:**
  + Conexión física de servidores (cableado/racks)
  + Configuración avanzada de red (VLANs, switches)
  + Apertura segura de puertos en firewall institucional
  + Monitorización de ancho de banda
  + Resolución de incidencias complejas de conectividad
  + Gestión de backups automatizados

**Nota sobre responsabilidades de red:**

* El desarrollador manejará la configuración lógica de red (IPs, firewall básico)
* El Soporte TI se encargará de la infraestructura física y configuración avanzada
* Coordinación semanal para alinear requerimientos de red

## Viabilidad temporal

Completar el proyecto en un mes mediante una planificación ultra optimizada y dirección clara de tareas y sin gasto de tiempo innecesario :

Plan supuesto:

1. **Semana 1 - Configuración Crítica**
   * **Servidor Web:**
     + Instalación de Ubuntu Server + Apache + PHP
     + Configuración básica de seguridad (firewall)
     + Estructura inicial para archivos .md
   * **Servidor MySQL:**
     + Instalación y configuración básica
     + Creación de usuario para conexión remota
     + Tablas esenciales (cursos, lecciones)
   * **Red:**
     + Asignación IPs fijas (coordinado con Soporte TI)
     + Pruebas de conectividad entre servidores
2. **Semana 2-3 - Desarrollo Acelerado**
   * **Frontend:**
     + Interfaz mínima viable (1 plantilla reutilizable)
     + Navegación básica sin JavaScript complejo
   * **Backend:**
     + Sistema de lectura de archivos .md
     + Conexión básica a la base de datos
     + 3-5 lecciones por curso como demostración
3. **Semana 4 - Integración Express**
   * Pruebas funcionales básicas
   * Ajustes de rendimiento mínimos
   * Documentación esencial

## Planificación temporal o agenda de trabajo.

Tabla 1- Planificacion Temporal

| **Semana** | **Foco Principal** | **Tareas Clave** | **Horas** |
| --- | --- | --- | --- |
| **1** | Infraestructura | - Instalación servidores - Configuración red | 40h |
| **2** | Desarrollo Frontend/Backend | - Interfaz mínima - Lector .md | 60h |
| **3** | Contenidos Básicos | - Creación 2 cursos demo | 30h |
| **4** | Lanzamiento Controlado | - Pruebas - Ajustes finales | 20h |

# ANÁLISIS DE REQUISITOS

## Descripción de requisitos.

**Funcionalidades principales:**

1. **Visualización de contenidos educativos**
   * Mostrar archivos en formato Markdown (.md) con:
     + Resaltado de sintaxis para código SQL y comandos de redes
     + Soporte para imágenes y diagramas incrustados
   * Organización por cursos y lecciones
2. **Navegación entre contenidos**
   * Menú lateral con índice de cursos
   * Acceso rápido a lecciones recientes
3. **Búsqueda básica**
   * Buscar términos en títulos y contenidos
   * Filtrado por categorías (Database/Redes)

**Requisitos técnicos:**

1. **Compatibilidad**
   * Navegadores modernos (Chrome, Firefox, Edge)
   * Dispositivos móviles (tablets y smartphones)
2. **Rendimiento**
   * Carga inicial en menos de 3 segundos
   * Soporte para 50 usuarios simultáneos

## Diagramas visuales

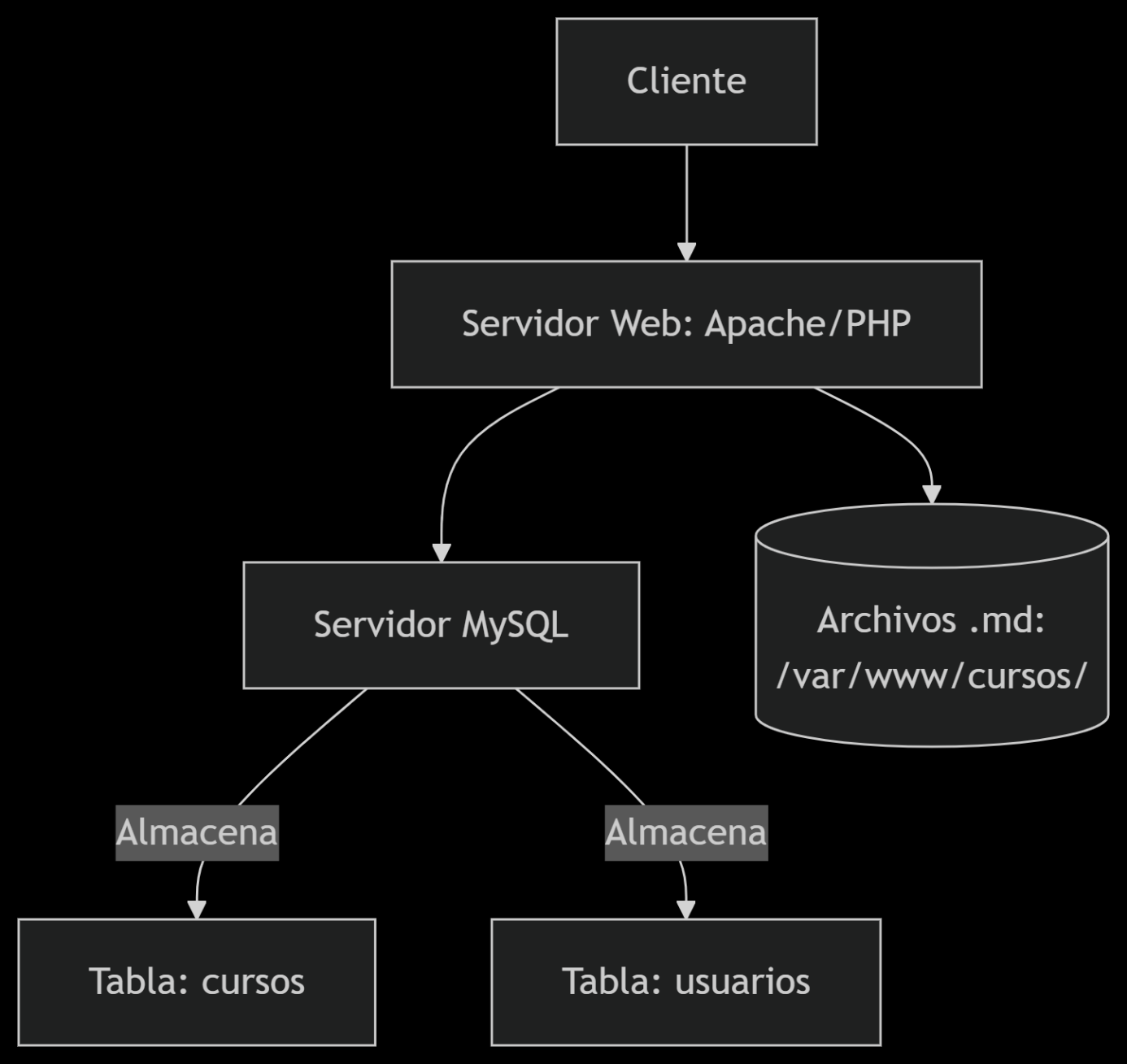


IMAGEN 1 - Arquitectura fisica

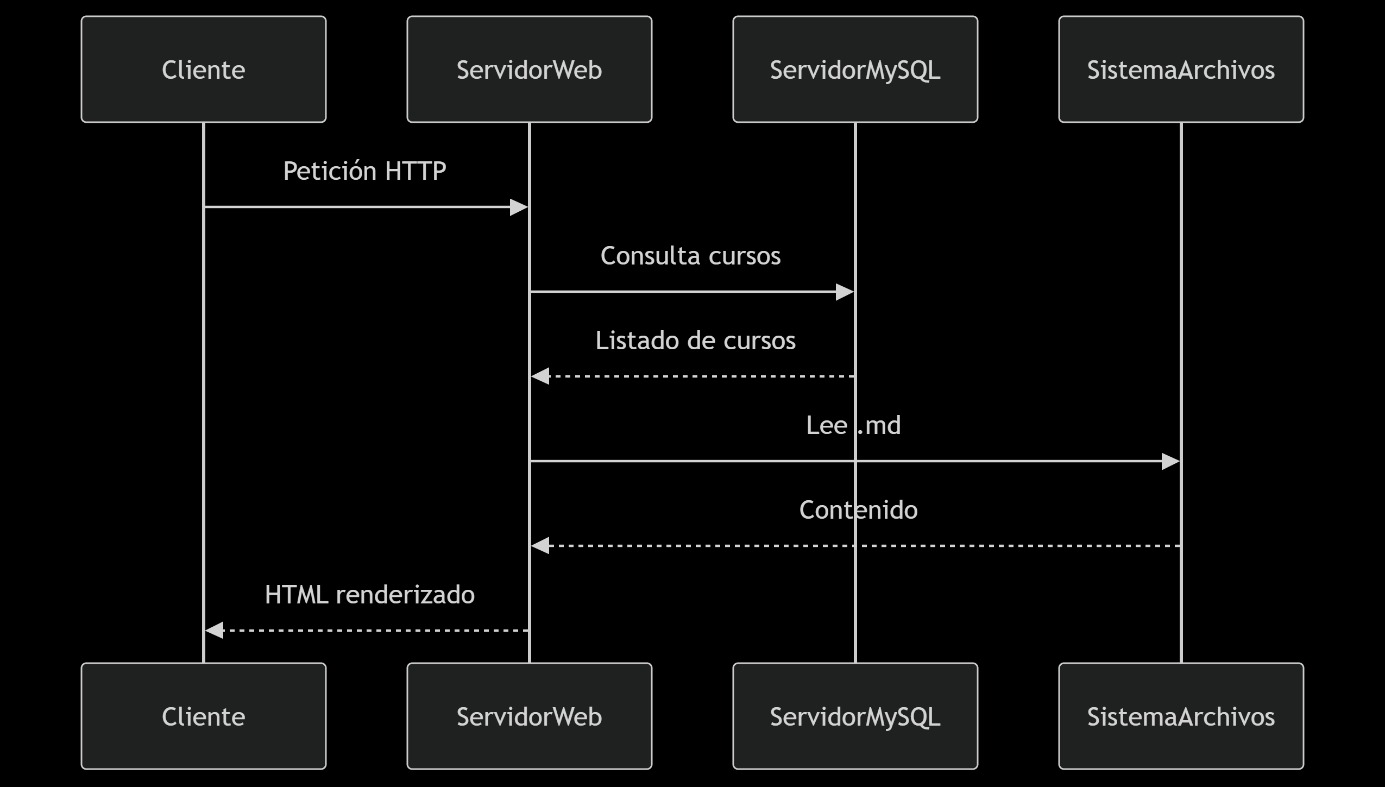


IMAGEN 2 - Arquitectura de datos

## Análisis de riesgos y vulnerabilidades

**Principales desafíos técnicos:**

1. **Formato de archivos**
   * Riesgo: Inconsistencias en archivos .md
   * Solución: Validación con script PHP antes de cargar
2. **Conectividad entre servidores**
   * Riesgo: Latencia en consultas MySQL
   * Solución: Optimizar índices y usar conexiones persistentes
3. **Seguridad**
   * Riesgo: Acceso no autorizado a archivos
   * Solución: Configuración estricta de permisos en /var/www/cursos/

**Plan de mitigación:**

* Pruebas diarias de integración
* Monitoreo con herramientas como NetData
* Backups automáticos de base de datos y contenidos

# DISEÑO (DEPENDE DEL PROYECTO)

## Diseño de la arquitectura de red

El diseño de red se basa en una arquitectura cliente-servidor con dos nodos principales: el servidor web (Apache/PHP) y el servidor de base de datos (MySQL). La comunicación entre estos servidores se realizará a través de la red local del centro educativo, utilizando el puerto 3306 para las consultas MySQL. Se implementará un sistema de direccionamiento IP fijo para garantizar la estabilidad de las conexiones, asignando 192.168.1.10 al servidor web y 192.168.1.20 al servidor de base de datos.

## Diseño de la topología de red

La topología física sigue un esquema en estrella, donde ambos servidores se conectarán a un switch gestionable central. Este switch estará a su vez conectado al router principal del centro educativo, permitiendo el acceso a los usuarios internos. Para garantizar el rendimiento, se utilizará cableado CAT6 en todas las conexiones físicas y se configurará QoS (Calidad de Servicio) para priorizar el tráfico de la plataforma educativa.

## Diseño de las soluciones de seguridad.

Las medidas de seguridad incluyen:

1. Configuración de firewall en ambos servidores, permitiendo solo los puertos esenciales (80, 443, 3306)
2. Implementación de reglas de iptables para proteger contra ataques DDoS básicos
3. Uso de certificados SSL autofirmados para el tráfico web
4. Política de permisos estrictos en los directorios de archivos (/var/www/cursos/)
5. Configuración de backups automáticos diarios para la base de datos y los contenidos
6. **Segmentación VLAN:**

Configuración en el switch para aislar los servidores en VLAN 100.

Reglas en el firewall para restringir el tráfico inter-VLAN.

## Diagrama de despliegue.

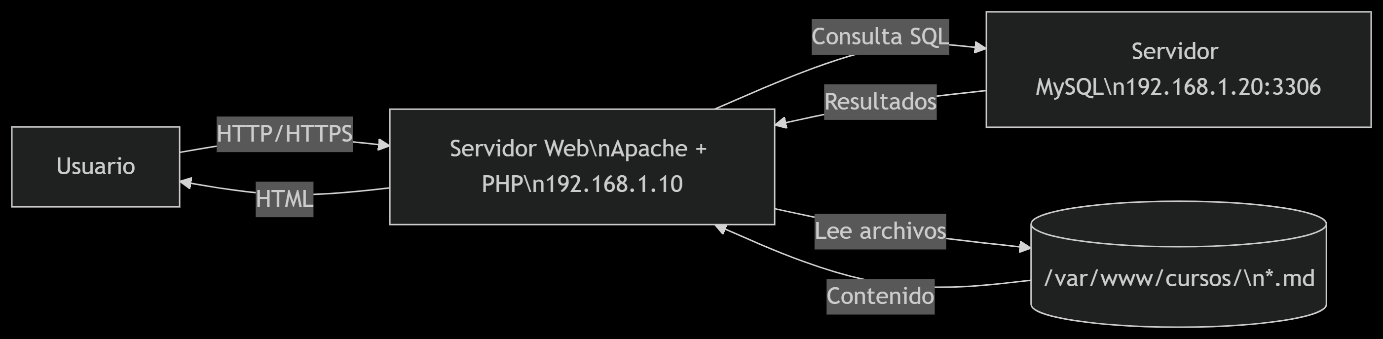


IMAGEN 3 - diagrama de despligue

El usuario accede via navegador al servidor web (Apache/PHP).

El servidor web:

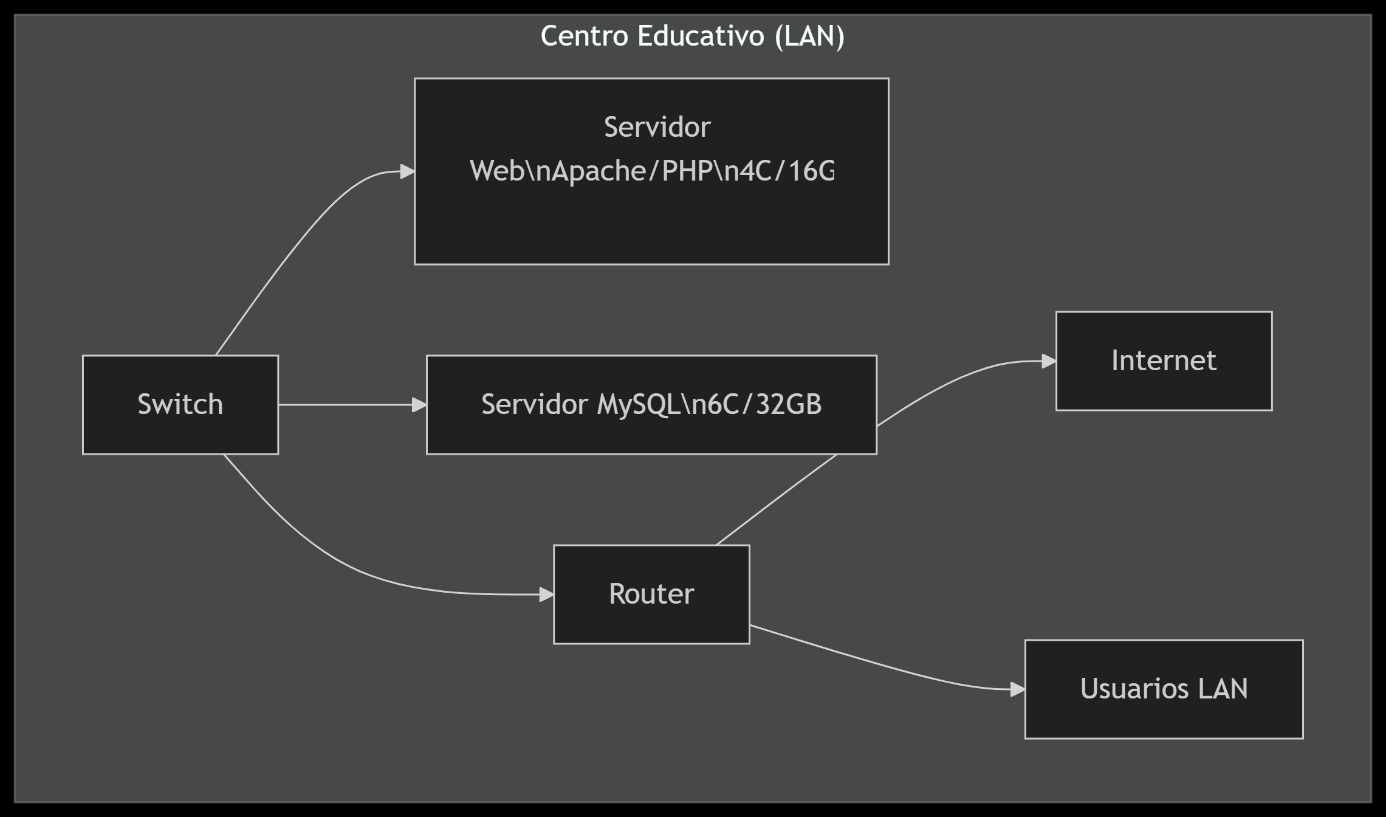
* + Consulta metadatos en MySQL (cursos, lecciones).
  + Lee contenidos Markdown del sistema de archivos local.

Devuelve HTML renderizado al usuario.

**Detalles técnicos**:

* + **Protocolos**: HTTP (puerto 80) y HTTPS (puerto 443).
  + **Conexión MySQL**: Usa autenticación segura con usuario restringido.
  + **Archivos .md**: Permisos restringidos a www-data.

## Diagrama de infraestructura.



**Componentes**:

* **Switch**: Conecta servidores y router (VLAN).
* **Router**: Gestiona tráfico interno/externo (NAT para acceso web).
* **Servidores**: Configuración física

# IMPLEMENTACIÓN

En esta sección se detalla cómo se llevará a cabo la implementación del proyecto. Se especifican las tareas y acciones necesarias para poner en marcha el sistema.

## Instalación y configuración de los dispositivos de red.

El proceso comienza con el montaje físico de los servidores en el rack del centro educativo, asegurando una adecuada ventilación y organización del cableado estructurado CAT6. Se procede a configurar el switch gestionable, creando una VLAN dedicada exclusivamente para el tráfico de los servidores educativos. Cada servidor recibe una dirección IP fija dentro del rango privado institucional, con el servidor web como puerta de enlace principal.

La configuración de red incluye la implementación de QoS para priorizar el tráfico educativo y la segmentación lógica mediante VLANs para aislar los servidores de otros dispositivos en la red institucional. Se establecen conexiones redundantes mediante los puertos Gigabit Ethernet duales de cada servidor.

## Configuración de las soluciones de seguridad.

La seguridad se implementa en capas, comenzando con la configuración del firewall en ambos servidores para restringir el acceso únicamente a los puertos esenciales (HTTP, HTTPS y MySQL). Se aplican reglas estrictas de filtrado de paquetes y protección contra ataques DDoS básicos.

En el servidor de base de datos, se configura autenticación segura con acceso restringido únicamente desde la IP del servidor web. Se implementa Fail2Ban para protección contra intentos de acceso no autorizados, configurando umbrales sensibles para los servicios SSH, HTTP y MySQL.

Los permisos de archivos se configuran con el principio de mínimo privilegio, especialmente en el directorio que contiene los materiales educativos, donde solo el usuario del servicio web tiene permisos de lectura.

## Configuración de las soluciones de virtualización.

Aunque el proyecto se basa principalmente en infraestructura física, se prepara el entorno para posibles necesidades futuras de virtualización. Esto incluye la instalación del hipervisor KVM y la configuración de redes virtuales aisladas que podrían utilizarse para entornos de desarrollo o pruebas.

Se define una estructura de redes virtuales que replica la arquitectura física, permitiendo migraciones futuras si fuera necesario. Se configuran plantillas base de máquinas virtuales con los mismos estándares de seguridad que los servidores físicos.

## Configuración de las soluciones de almacenamiento en red.

El sistema de almacenamiento se implementa utilizando **FTP/SFTP** como método principal para la transferencia segura de archivos, eliminando la necesidad de configuraciones complejas de NAS o Samba.

1. **Almacenamiento Principal**
   * Los materiales educativos (archivos Markdown, imágenes, etc.) se alojan en el servidor web (/var/www/cursos/).
   * Acceso restringido mediante **FTP seguro (SFTP/FTPS)** para administradores y profesores autorizados.
2. **Seguridad en FTP**
   * **Encriptación**: Uso obligatorio de FTPS (FTP + SSL) o SFTP para evitar transferencias en texto plano.
   * **Autenticación**:
     + Usuarios con acceso limitado a sus directorios asignados (chroot).
     + No se permite FTP anónimo.
   * **Registro de actividad**: Logs detallados de conexiones y transferencias.

## Configuración de las copias de seguridad.

El plan de backups sigue la metodología 3-2-1 (tres copias, en dos medios diferentes, una fuera del sitio). Se implementan:

1. Backups completos diarios de la base de datos con retención de 7 días
2. Backups incrementales de los contenidos educativos cada 6 horas
3. Snapshots semanales del estado completo del sistema

Los backups se verifican automáticamente mediante scripts que comprueban la integridad de los archivos y la capacidad de restauración. Se establece un protocolo claro de recuperación ante desastres, documentando paso a paso el proceso para restaurar el servicio en caso de fallo crítico.

Todos los procesos de backup se registran en un sistema centralizado de monitoreo, con alertas configuradas para cualquier fallo en el proceso de copia de seguridad. Se programan pruebas trimestrales de restauración completa para validar la efectividad del sistema.

# ADMINISTRACIÓN

## Gestión de usuarios y permisos.

Se implementará un sistema de acceso basado en roles:

* **Administradores**: Acceso completo (FTP, MySQL, servidores).
* **Profesores**: Permisos para subir/editar materiales (FTP/SFTP en /var/www/cursos/).
* **Estudiantes**: Solo lectura (acceso web sin permisos de modificación).

**Acciones clave**:  
✔ Creación de usuarios con adduser y asignación a grupos (www-data, ftp-users).  
✔ Configuración de permisos en archivos (chmod 750 para profesores, chmod 644 para contenido público).  
✔ Uso de **SFTP/FTPS** para transferencia segura de archivos (no FTP plano).

## Monitoreo y mantenimiento de la red.

**Herramientas y prácticas**:

* **Monitoreo**:
  + vnstat para análisis de ancho de banda.
  + iftop para tráfico en tiempo real.
  + Alertas por correo ante caídas de servicio (configuradas en netdata).
* **Mantenimiento**:
  + Actualización mensual de firmware en switch/router.
  + Pruebas de latencia entre servidores (ping, traceroute).

## Políticas de seguridad

**Normas obligatorias**:

* **Contraseñas**:
  + Mínimo 12 caracteres, con mayúsculas, números y símbolos.
  + Cambio cada 90 días.
* **Accesos**:
  + SSH solo con claves públicas (no password).
  + FTP restringido a IPs internas + VPN.
* **Auditorías**:
  + Scans semanales con lynis para detectar vulnerabilidades.

## Plan de mantenimiento preventivo y correctivo

Tabla 2 - Plan de mantenimiento y correctivo

| **Tipo** | **Frecuencia** | **Acciones** |
| --- | --- | --- |
| **Preventivo** | Semanal | - Actualizaciones de seguridad (apt update && apt upgrade). |
|  | Mensual | - Verificación de logs (/var/log/apache2/, /var/log/mysql/). |
| **Correctivo** | Según fallos | - Restauración de backups ante corrupción de datos. |
|  |  | - Reemplazo de hardware defectuoso (discos, fuentes de alimentación). |

## Implementación de backups y recuperación ante desastres.

**Estrategia 3-2-1**:

* **3 copias**:
  1. Local (en servidor web).
  2. Remota (NAS interno vía SFTP).
  3. Externa (USB cifrado, guardado fuera del sitio).
* **Automatización**:
  1. Scripts diarios (mysqldump + rsync para archivos).
  2. Verificación semanal de integridad.
* **Recuperación**:
  1. Tiempo máximo estimado: 2 horas (base de datos + contenidos).

## Plan de contingencia y recuperación ante incidentes

**Escenarios críticos y respuestas**:

1. **Ataque DDoS**:
   * Activación de reglas de firewall emergentes (limitar conexiones/segundo).
2. **Pérdida de datos**:
   * Restauración desde backups + análisis forense.
3. **Caída de servidor**:
   * Migración temporal a máquina virtual de respaldo.

## Soporte a la aplicación

**Canales de ayuda**:

* **Sistema de tickets** (GLPI o similar) para reportar:
  + Errores en materiales (Markdown mal formateado).
  + Problemas de acceso (FTP, web).
* **Helpdesk prioritario**:
  + Profesores: Respuesta en <24h.
  + Estudiantes: Respuesta en <48h.

**Documentación**:

* Manuales en formato Markdown accesibles desde la plataforma.
* FAQs para problemas comunes (ej: "Cómo subir archivos via SFTP").

# HERRAMIENTAS DE APOYO

Son aquellas que se utilizan para facilitar la ejecución del proyecto, como control de versiones, gestión de pruebas, entre otros.

## Control de versiones

**Herramienta principal:** **Git**  [“GITHUB”](https://github.com/TIKIDAXX/Plataforma-educativa.git)

**Implementación:**  
✔ **Estructura del repositorio:**

Copy

Download

/

── backend/ # Código PHP y lógica de la plataforma

── frontend/ # HTML, CSS, JavaScript

── docs/ # Documentación técnica

── scripts/ # Automatizaciones (backups, despliegues)

── cursos/ # Materiales educativos (Markdown)

## Gestión de pruebas

**Estrategia de testing:**

Tabla 3 - Gestion de pruebas

| **Tipo de Prueba** | **Herramientas** | **Frecuencia** | **Objetivo** |
| --- | --- | --- | --- |
| **Unitarias** | PHPUnit (backend) | Tras cada commit | Verificar funciones críticas (ej: parser MD). |
| **Integración** | Selenium (frontend) | Semanal | Probear interacción frontend-backend-DB. |
| **Rendimiento** | Apache JMeter | Mensual | Evaluar carga (100 usuarios simultáneos). |
| **Seguridad** | OWASP ZAP | Trimestral | Detectar vulnerabilidades (SQLi, XSS). |

# CONCLUSIONES.

* 1. Conclusiones sobre el trabajo realizado

Es una sección en donde se explica el trabajo realizado, se destacan los logros, se presentan las limitaciones, y se plantean posibles mejoras.

* 1. Conclusiones personales

Es una sección en donde el autor del informe expresa su opinión personal sobre el proyecto, y su experiencia al desarrollarlo.

* 1. Posibles ampliaciones y mejoras

Es una sección en donde se plantean ideas para seguir mejorando el proyecto en el futuro.

# BIBLIOGRAFÍA

Es una sección en donde se incluyen las fuentes utilizadas para elaborar el proyecto, incluyendo libros, artículos, apuntes, y direcciones web. Es importante que se haga una breve reseña de cada una de ellas, y se indique su importancia para el proyecto.

* 1. Libros, artículos y apuntes
  2. Direcciones web

# ANEXOS

Se distinguirán con letras mayúsculas consecutivas con el estilo **Título 2**; ejemplo: Anexo A, Anexo B, … En ellos se puede incluir:

* Manual Instalación/Configuración/Uso
* Partes de código/Ficheros de configuración importantes.
* …