

Manual tecnico

Grupo E



2 de julio de 2025

construcción de software

Universidad de Guayaquil

**Manual Técnico: Página Web de Venta y Compra de Cursos Online**

El manual técnico está dirigido a **desarrolladores, administradores de sistemas y personal de soporte técnico**. Debe ser exhaustivo, preciso y cubrir los detalles internos de cómo funciona y se mantiene la plataforma.

**1. Propósito**

* Proveer una comprensión profunda de la arquitectura y el funcionamiento interno de la plataforma.
* Servir como guía para la instalación, configuración, mantenimiento, depuración y escalabilidad.
* Facilitar futuras modificaciones y extensiones por parte del equipo de desarrollo.
* Asegurar la consistencia en el desarrollo y las operaciones.

**2. Estructura y Contenido Sugerido**

* **Página de Título:** Nombre del manual, nombre de la plataforma, versión, fecha y autores/equipo.
* **Índice/Tabla de Contenidos:** Muy detallado.
* **Introducción:**
  + Propósito del manual.
  + Descripción general de la plataforma desde una perspectiva técnica.
  + Audiencia objetivo.
* **Visión General de la Arquitectura:**
  + **Diagrama de Arquitectura:** Un diagrama de alto nivel que muestre los componentes principales (front-end, back-end, base de datos, APIs externas, servicios de terceros).
  + **Tecnologías Utilizadas:** Lenguajes de programación (Python, Node.js, PHP, etc.), frameworks (React, Angular, Vue, Django, Laravel), bases de datos (SQL, NoSQL), servidores web (Nginx, Apache), sistemas de caché, etc.
  + **Principios de Diseño:** Patrones de diseño aplicados, consideraciones de seguridad, escalabilidad, rendimiento.
* **Requisitos del Sistema e Instalación:**
  + **Prerrequisitos:** Hardware, software, sistema operativo.
  + **Guía de Instalación:** Pasos detallados para configurar el entorno de desarrollo y producción.
    - Clonación del repositorio.
    - Configuración de variables de entorno.
    - Instalación de dependencias.
    - Configuración de la base de datos.
    - Ejecución de migraciones.
    - Configuración del servidor web.
    - Configuración de certificados SSL.
    - Despliegue (CI/CD si aplica).
* **Estructura de la Base de Datos:**
  + **Diagrama Entidad-Relación (ERD):** Visualización de las tablas y sus relaciones.
  + **Descripción de Tablas:** Nombre de la tabla, propósito, campos (tipo de dato, restricciones, claves primarias/foráneas).
  + **Modelos de Datos:** Cómo se mapean los datos a los objetos/clases en el código.
* **Estructura del Código y Componentes:**
  + **Directorios y Archivos Clave:** Explicación de la organización del proyecto.
  + **Módulos/Servicios Principales:** Descripción de cada componente importante (ej. módulo de autenticación, gestión de cursos, pasarela de pagos, sistema de notificaciones).
  + **APIs Internas y Externas:** Descripción de los endpoints, métodos, parámetros, respuestas y autenticación.
  + **Control de Versiones:** Estándares de ramas, etiquetado, convenciones de *commits*.
* **Configuración y Parámetros:**
  + Variables de entorno.
  + Archivos de configuración.
  + Parámetros ajustables para diferentes entornos (desarrollo, *staging*, producción).
* **Seguridad:**
  + Vulnerabilidades conocidas y cómo se mitigan (SQL injection, XSS, CSRF).
  + Estrategias de autenticación y autorización.
  + Cifrado de datos.
  + Políticas de respaldo y recuperación.
* **Despliegue y Operaciones:**
  + **Estrategias de Despliegue:** Manual, automatizado (scripts, Docker, Kubernetes).
  + **Monitoreo y Logging:** Herramientas y procedimientos para monitorear el rendimiento y registrar eventos.
  + **Mantenimiento Rutinario:** Tareas de limpieza de base de datos, actualización de dependencias.
  + **Backup y Recuperación de Desastres:** Procedimientos para copias de seguridad y restauración.
* **Pruebas:**
  + Estrategias de pruebas (unitarias, integración, end-to-end).
  + Cómo ejecutar las pruebas.
* **Guía de Contribución (si es un equipo):**
  + Estándares de codificación.
  + Proceso de *code review*.
  + Cómo reportar bugs y solicitar nuevas características.
* **Solución de Problemas (Troubleshooting):**
  + Errores comunes y sus soluciones.
  + Cómo depurar la aplicación.
* **Glosario de Términos Técnicos:** Definiciones de la jerga específica del proyecto.

**3. Consejos para el Manual Técnico**

* **Precisión y Detalle:** Cada detalle cuenta. Sé lo más específico posible.
* **Diagramas y Gráficos:** Los diagramas de flujo, diagramas de componentes, ERDs y diagramas de secuencia son cruciales.
* **Ejemplos de Código:** Incluye fragmentos de código relevantes para ilustrar configuraciones o funciones complejas.
* **Mantener Actualizado:** Este manual es vital para el equipo de desarrollo. Debe ser un documento vivo que se actualiza con cada cambio significativo en la arquitectura o el código.
* **Control de Versiones:** Almacena el manual en el mismo sistema de control de versiones que el código (Git) para rastrear cambios.
* **Herramientas:** Considera herramientas como Sphinx, MkDocs, o Confluence para generar y organizar la documentación técnica.