**Guía3. Informe final Proyecto APT**

**Asignatura Capstone**

**Integrantes:**

* **Elvis Andrade**
* **Maximiliano Betta**
* **Rodrigo Mendoza**

|  |
| --- |
| **1. Informe final Proyecto APT** |
|  |

|  |  |
| --- | --- |
| Nombre del proyecto | MentoraLMS - Plataforma de Gestión de Aprendizaje para Instituciones Educativas |
| Área (s) de desempeño(s) | • Desarrollo y gestión de software web.  • Innovación en entornos digitales educativos.  • Administración de proyectos tecnológicos.  • Diseño y administración de bases de datos. |
| Competencias | • Realizar pruebas de certificación de productos y procesos usando buenas prácticas de la industria.  • Gestionar proyectos informáticos ofreciendo alternativas para la toma de decisiones.  • Construir modelos de datos escalables y seguros.  • Desarrollar soluciones de software siguiendo técnicas que sistematizan el desarrollo y mantenimiento.  • Comunicar información técnica y documentación en inglés. |

|  |  |
| --- | --- |
| **Contenidos del informe final** | |
| 1. Relevancia del proyecto APT | **1. Relevancia del proyecto APT**  **Problema que busca solucionar:**  MentoraLMS aborda la **falta de acceso a plataformas educativas digitales propias en instituciones educativas de recursos limitados**. Muchos colegios dependen de sistemas fragmentados (correos, planillas Excel, WhatsApp) para gestionar procesos académicos, generando desorganización, falta de seguimiento centralizado, comunicación ineficiente, y barreras económicas para acceder a plataformas comerciales costosas. Esto profundiza la brecha tecnológica educativa entre estudiantes de diferentes contextos socioeconómicos.  **Relevancia para el campo laboral:**  Este proyecto es altamente relevante para Ingeniería en Informática porque:   * Responde a la creciente demanda del sector educativo digital. * Aplica integralmente múltiples competencias técnicas: desarrollo full-stack Laravel, diseño de bases de datos, seguridad web, metodologías ágiles, despliegue cloud. * Proporciona experiencia en desarrollo de software con impacto social real. * Desarrolla habilidades directamente valoradas en la industria del software.   **Ubicación del problema:**  El proyecto está dirigido a instituciones educativas, específicamente colegios municipales, particulares subvencionados, centros de capacitación y organizaciones educativas sin fines de lucro. Según el Ministerio de Educación (2024), más del 60% de los establecimientos no cuentan con plataforma LMS propia. La problemática es extensible a otros países de América Latina con contextos similares.  **Impacto:**  La solución beneficia a tres grupos principales:  **Estudiantes:** Acceso centralizado a materiales y evaluaciones 24/7, mayor autonomía en el aprendizaje, mejor comunicación con docentes, y desarrollo de competencias digitales.  **Docentes:** Organización y centralización del trabajo académico, gestión eficiente de múltiples cursos, distribución digital de materiales, sistema integrado de evaluaciones y calificaciones, seguimiento efectivo del progreso estudiantil.  **Administradores:** Visibilidad completa de procesos académicos, datos para toma de decisiones, control centralizado de usuarios y cursos, reducción de costos operativos.  **Impacto cuantitativo estimado:**   * 40% de reducción en tiempo administrativo docente * 50% de aumento en acceso estudiantil a materiales fuera de horario   **Aporte de valor:**  MentoraLMS se diferencia por su **accesibilidad económica** (costo mínimo de hosting vs. licencias de $1,000-5,000 USD anuales), **diseño contextualizado** al sistema educativo, **tecnología moderna** (Laravel 10, arquitectura escalable), **funcionalidad integral** (cubre procesos académicos esenciales), y **contribución a la equidad educativa** reduciendo la brecha tecnológica. |
| 2. Objetivos | Objetivo General:  Diseñar, desarrollar e implementar MentoraLMS, una plataforma LMS basada en PHP Laravel 10 y MySQL, alojada en Hostinger, que permita a instituciones educativas de recursos limitados organizar eficientemente sus procesos académicos mediante herramientas digitales accesibles, intuitivas y escalables.  Objetivos Específicos:   1. Diseñar una arquitectura de base de datos normalizada que soporte gestión escalable de usuarios, cursos, materiales, evaluaciones y foros. 2. Implementar sistema robusto de autenticación y autorización con tres roles (estudiante, profesor, administrador) usando Laravel. 3. Desarrollar módulos funcionales principales: gestión de cursos, materiales educativos, evaluaciones, calificaciones y foros de discusión. 4. Crear interfaz responsiva e intuitiva para web escritorio, tablet y móvil. 5. Configurar y desplegar en Hostinger con prácticas de seguridad (HTTPS, validaciones) y backups automáticos. 6. Ejecutar pruebas exhaustivas: unitarias, integración y usuario con casos reales. 7. Elaborar documentación técnica completa: arquitectura, base de datos, código, manuales de usuario, en español. |
| 3. Metodología | **Metodología Utilizada:**  Implementamos **metodología ágil Scrum adaptada** a equipo de 3 integrantes, con sprints de 2 semanas, entregas incrementales y prácticas modernas de desarrollo (Git, integración continua, revisiones de código).  **Estructura de Sprints:**   * **Sprint Planning** (2h): Selección de historias, estimación, asignación de tareas * **Daily Stand-ups adaptados** (cada 3-4 días, 15-20 min): Sincronización y detección de obstáculos * **Sprint Review** (1h): Demo de funcionalidades al profesor guía * **Sprint Retrospective** (30 min): Mejora continua del proceso   **Fases del Proyecto:**  **FASE 1: Planificación (Semana 1):** Definición de alcance, roles, Product Backlog (45 historias de usuario), configuración de herramientas (GitHub, Slack, Discord), levantamiento de requerimientos con usuarios potenciales.  **FASE 2: Análisis y Diseño (Semanas 2-5):** Diseño de modelo entidad-relación con 12 tablas normalizadas, prototipos de alta fidelidad en Figma, definición de arquitectura MVC y guía de estilos.  **FASE 3: Configuración Base (Semanas 3-6):** Instalación Laravel 10, configuración MySQL, migraciones, Laravel, estructura de vistas y componentes.  **FASE 4: Desarrollo de Módulos (Semanas 7-12):**   * Sistema de roles y permisos. * CRUD completo de cursos con inscripciones * Sistema de carga de materiales con Laravel Storage * Módulo de evaluaciones con diferentes tipos de preguntas * Sistema de calificaciones con gráficos * Foros con comentarios anidados   **FASE 5: Pruebas y Optimización (Semanas 13-16):** Pruebas unitarias integración, seguridad, usabilidad (8 usuarios reales), corrección de bugs, optimización de consultas SQL, mejoras de accesibilidad.  **FASE 6: Despliegue y Documentación (Semanas 13-18):** Configuración servidor Hostinger, dominio, SSL, migración a producción, backups automáticos, documentación técnica completa, manuales de usuario por rol, video demostrativo.  **Justificación:**  Scrum fue pertinente porque permitió adaptabilidad a cambios, entregas incrementales validables, visibilidad constante del progreso, colaboración efectiva del equipo, y gestión de riesgos mediante detección temprana de problemas. |
| 4. Desarrollo | **Etapas del Proyecto:**  El desarrollo siguió las 6 fases descritas en la metodología, con énfasis en iteraciones cortas y validación continua.  **Facilitadores:**   * **Experiencia previa del equipo** en Laravel y desarrollo web aceleró la curva de aprendizaje. * **Documentación abundante** de Laravel y comunidades activas facilitaron resolución rápida de problemas técnicos. * **Herramientas colaborativas** (GitHub, Figma, Slack, Discord) mejoraron coordinación del equipo. * **Apoyo docente** con retroalimentación oportuna en etapas críticas. * **Recursos accesibles** (Hostinger económico, Laravel gratuito) eliminaron barreras de presupuesto.   **Dificultades y soluciones:**   1. **Carga académica simultánea** (Semanas 4-5): Otros ramos generaron conflicto de tiempo. **Solución:** Reorganización de tareas priorizando actividades críticas, redistribución temporal de responsabilidades, reuniones más frecuentes pero cortas. 2. **Complejidad del sistema de permisos** (Semana 7): Diseño de permisos granulares resultó más complejo de lo anticipado. **Solución:** Asignación de 3-4 días adicionales, consulta intensiva de documentación Laravel, validación con profesor guía para evitar refactorizaciones. 3. **Curva de aprendizaje en Tailwind CSS**: Desarrollador frontend necesitó familiarizarse más. **Solución:** Tutoriales específicos, uso de componentes pre-diseñados de Tailwind UI para acelerar desarrollo. 4. **Coordinación de horarios**: Integrantes con jornada laboral dificultaron encuentros. **Solución:** Establecimiento de 2 horarios fijos semanales (miércoles 19:00, sábado 10:00), comunicación asíncrona por Discord para consultas menores. 5. **Optimización de consultas N+1** (Semana 15): Problemas de rendimiento con datos grandes. **Solución:** Implementación de eager loading, uso de Laravel Debugbar para identificar queries lentas, índices en base de datos.   **Ajustes realizados:**   * Extensión de 3 días en diseño de base de datos por validación adicional con profesor. * Iteración adicional de prototipos por feedback de usuarios potenciales. * Aumento de frecuencia de reuniones (cada 3-4 días vs. semanal) en primeras 6 semanas. * Incorporación de Trello como complemento a GitHub. * Priorización de funcionalidades críticas sobre nice-to-have cuando surgieron retrasos menores (2-3 días). |
| 5. Evidencias | Las siguientes evidencias respaldan el desarrollo exitoso del proyecto:   1. Repositorio GitHub completo con 150+ commits distribuidos equitativamente entre integrantes, mostrando código funcional, migraciones, pruebas unitarias y documentación técnica. 2. Diagrama entidad-relación normalizado con 12 tablas y documentación de decisiones de diseño. 3. Prototipos de alta fidelidad en Figma con pantallas validadas por usuarios potenciales. 4. Plataforma MentoraLMS desplegada en Hostinger, accesible públicamente con todos los módulos funcionales operativos. 5. Capturas de pantalla del sistema funcionando: dashboards por rol, gestión de cursos, carga de materiales, evaluaciones, calificaciones, foros. 6. Testing de pruebas unitarias con PHPUnit. 7. Reporte de pruebas de usabilidad con 8 usuarios reales, incluyendo hallazgos, correcciones implementadas y feedback positivo. 8. Documentación técnica completa: arquitectura del sistema, especificación de base de datos, documentación de API, guías de instalación. 9. Manuales de usuario diferenciados por rol (estudiante, profesor, administrador) con capturas de pantalla y flujos paso a paso. 10. Video demostrativo de 5 minutos mostrando funcionalidades principales del sistema. 11. Registros en Trello de tareas completadas, sprints cerrados y evolución del proyecto. |
| 6. Intereses y proyecciones profesionales | Reflexión:  Este proyecto nos permitió aplicar de manera integral los conocimientos adquiridos durante la carrera, enfrentando desafíos reales de desarrollo de software desde la concepción hasta el despliegue. La experiencia de trabajar con metodologías ágiles, gestionar un equipo, tomar decisiones técnicas complejas, y entregar una solución funcional con impacto social real, reforzó significativamente nuestra preparación profesional.  Aprendimos que el desarrollo de software va más allá del código: requiere comprensión profunda de las necesidades del usuario, planificación estratégica, comunicación efectiva, y capacidad de adaptación ante obstáculos imprevistos. La experiencia de validar nuestro trabajo con usuarios reales y recibir feedback positivo sobre la utilidad de MentoraLMS fue particularmente gratificante y motivadora.  El proyecto consolidó nuestro interés por el desarrollo full-stack, la arquitectura de software escalable, donde la tecnología puede generar impacto social tangible reduciendo inequidades y mejorando el acceso a educación de calidad.  Proyecciones laborales:  A partir de este proyecto, identificamos varias oportunidades profesionales:   1. Especialización en desarrollo Laravel y PHP: Profundizaremos en frameworks modernos, arquitectura de microservicios. 2. Vemos oportunidades en startups y empresas de tecnología educativa, un sector en crecimiento en América Latina. 3. Desarrollo de proyectos con impacto social: Buscamos posiciones donde la tecnología resuelva problemas reales de comunidades vulnerables. 4. Consultoría en transformación digital educativa: Asesorar instituciones en implementación de soluciones tecnológicas educativas. 5. Emprendimiento: MentoraLMS tiene potencial comercial como producto SaaS para colegios, lo que abre posibilidades de emprendimiento tecnológico.   El proyecto MentoraLMS no solo cumplió los objetivos académicos, sino que nos proporcionó un portafolio profesional demostrable, experiencia práctica valiosa, y una red de contactos en el sector educativo que facilitarán nuestra inserción laboral exitosa. |