

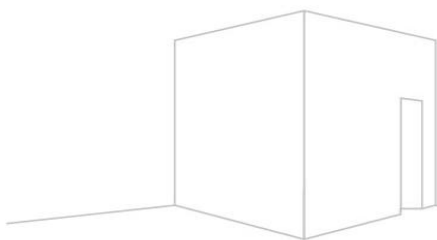
Informe Venta de Cursos online

Asignatura: Proyecto de Integración

Sección: IEI(1)-170-N6/D

Nombre del académico: Manuel Alfredo Sanchez Carcamo

Nombre de los integrantes del grupo: Víctor Jara, Nelson Slamanca, Esteban Reyes



Fecha de entrega

21-10-2025

Contenido

I.	Introducción	3
II.	Objetivos.....	4
III.	Desarrollo	5
1.	Especificación de Arquitectura.....	5
a)	Diseño de la Topología de comunicación.	5
b)	Diseño de la infraestructura.....	6
c)	Diseño de la arquitectura.....	7
2.	Descripción de las tecnologías.....	8
a)	Análisis comparativo cualitativo/cuantitativo.	8
b)	Selección de las herramientas.....	9
3.	Integración de las tecnologías.....	11
a)	Selección de los proveedores de tecnología.....	11
b)	Descripción de la seguridad y calidad.....	13
4.	Implementación de la solución.....	14
a)	Selección de plataformas.....	14
b)	Construcción de prototipo de la solución.....	15
5.	Aplicación de métodos, estándares y buenas prácticas.....	15
a)	Cumplimiento en los criterios de aceptación.....	15
b)	Cumplimiento de estándares normativos.....	16
c)	Cumplimiento de buenas prácticas.....	16
d)	Cumplimiento de rendimiento y eficiencia.....	16
e)	Cumplimiento de Seguridad.....	17
6.	Ajuste del Cronograma.....	18
IV.	Conclusiones	19

I. Introducción

El presente informe de desarrollo documenta la primera fase del proyecto "Plataforma de ventas de curso conecta saber", un sistema diseñado para la comercialización y entrega de contenido educativo en línea. Este documento se enfoca en la planificación técnica inicial, detallando la especificación de la arquitectura, la selección de las tecnologías más adecuadas y el esquema de integración de los diferentes componentes.

Se aborda el diseño de la arquitectura y la infraestructura, incluyendo la topología de comunicación que permite el flujo seguro de datos entre usuarios, servidores de aplicación y bases de datos. Posteriormente, se presenta un análisis comparativo que justifica la elección de herramientas y lenguajes clave, como Python con Django para el backend y React.js para el frontend, junto con PostgreSQL como sistema de gestión de bases de datos.

Finalmente, el informe establece los criterios para la implementación, seleccionando proveedores de servicios externos y plataformas de hosting. Se define un marco de seguridad integral, que incluye la protección de datos, la comunicación mediante HTTPS y la gestión segura de pagos. Este trabajo sienta las bases técnicas para la construcción de un prototipo de solución, asegurando el cumplimiento de estándares de calidad y rendimiento.

II. Objetivos

En este apartado se exponen los objetivos que guían el desarrollo y la documentación de la primera fase del proyecto de la plataforma de ventas de cursos.

Objetivo General

Documentar el diseño arquitectónico y justificar la selección tecnológica para la fase inicial de desarrollo del proyecto "Plataforma de ventas de curso conecta saber", asegurando la viabilidad técnica y operativa del prototipo.

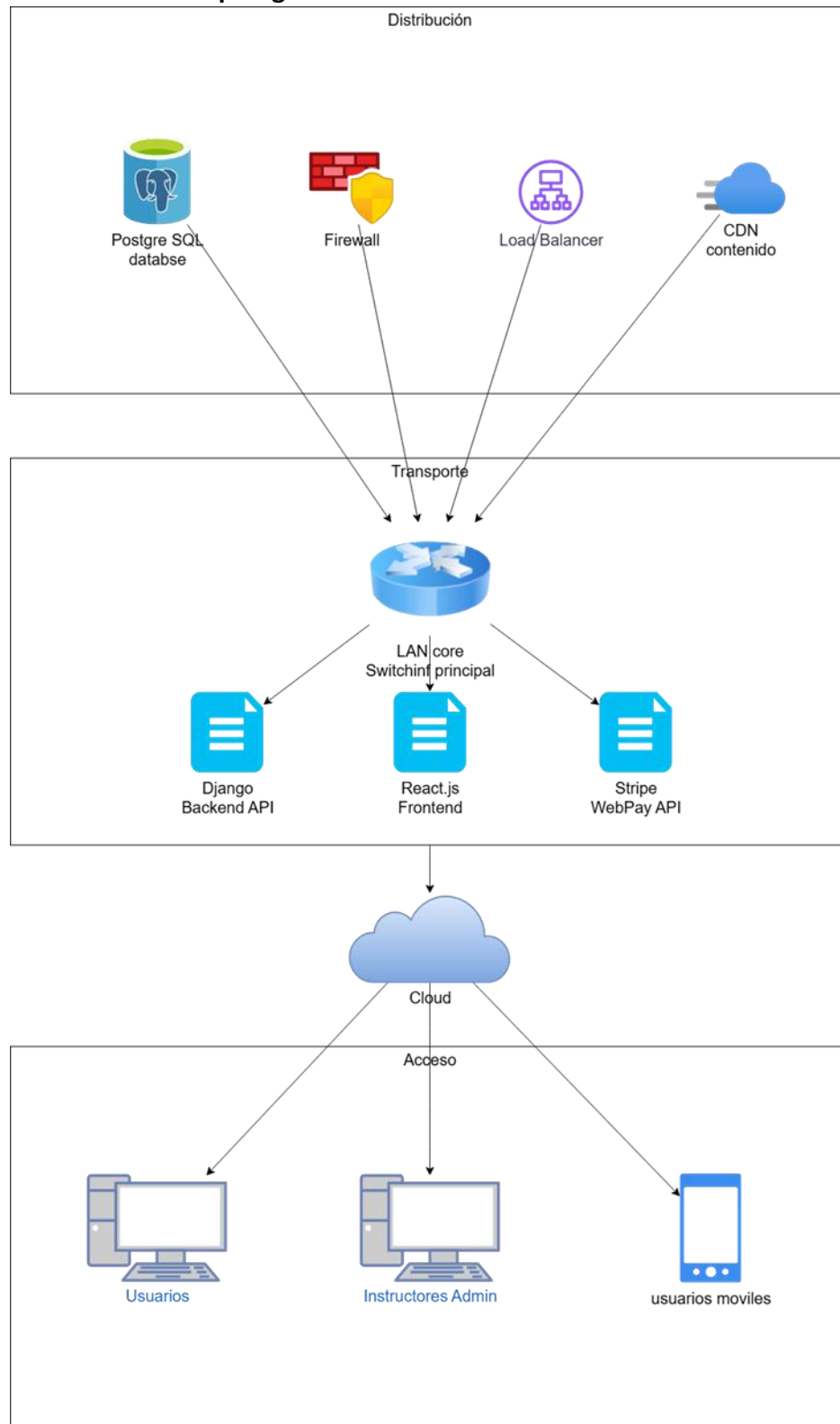
Objetivos Específicos

- Especificar el diseño de la arquitectura de software, la topología de comunicación y la infraestructura en la nube del proyecto.
- Realizar un análisis comparativo para seleccionar las herramientas de desarrollo, la base de datos y los servicios externos (pagos, video y correo electrónico).
- Definir los proveedores tecnológicos y establecer un marco de seguridad y calidad del software para la protección de la información y la estabilidad del sistema.
- Seleccionar las plataformas finales para la implementación del prototipo, incluyendo frontend, backend, base de datos y hosting.

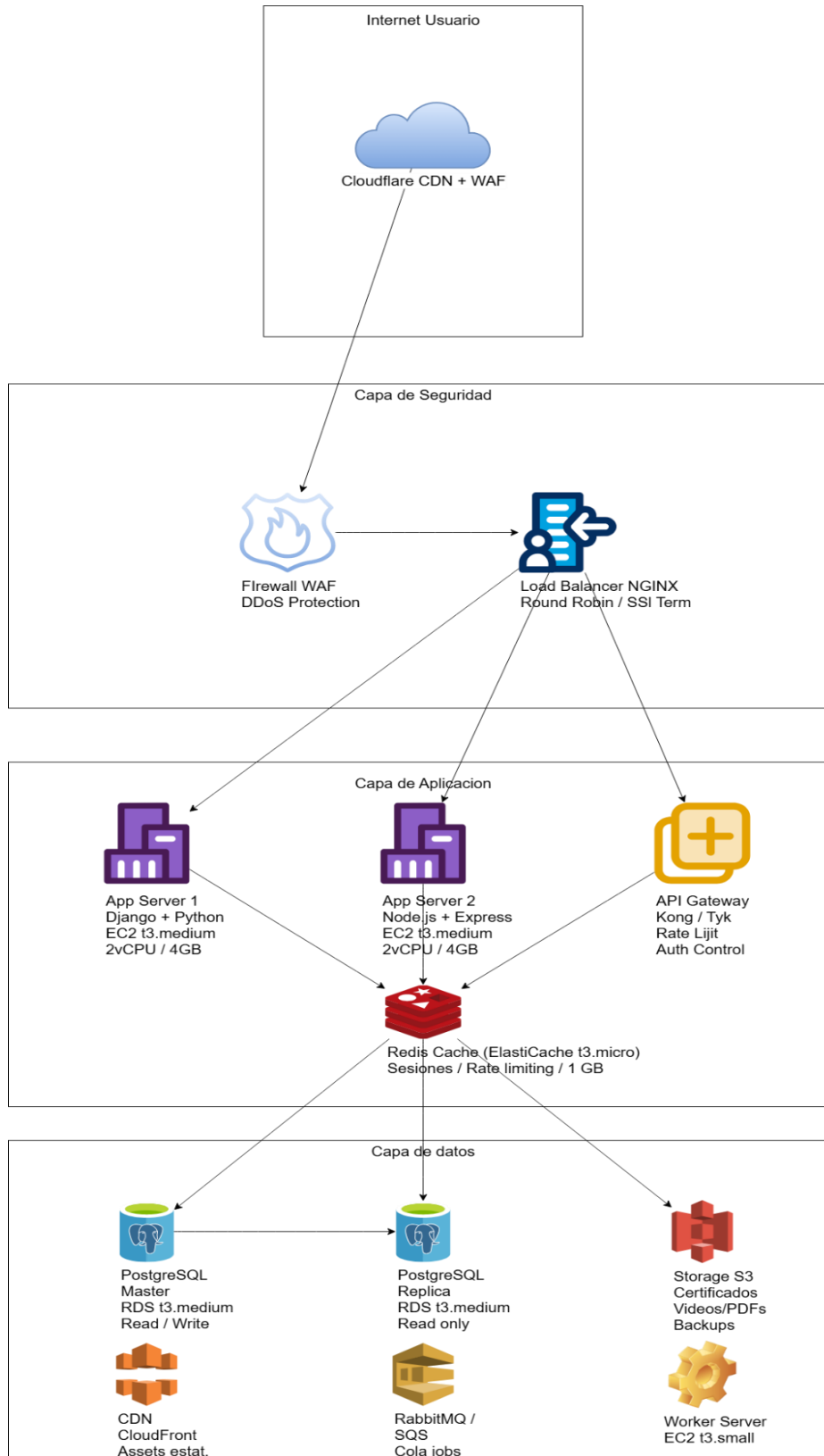
III. Desarrollo

1. Especificación de Arquitectura.

a) Diseño de la Topología de comunicación.



b) Diseño de la infraestructura.



c) Diseño de la arquitectura.



2. Descripción de las tecnologías.

a) Análisis comparativo cualitativo/cuantitativo.

Herramientas y lenguajes de desarrollo

Tipo	Elemento	Análisis Cualitativo	Análisis Cuantitativo
Lenguaje Backend	Python (Django) o Node.js (Express)	Permiten construir APIs robustas y seguras. Django facilita la autenticación, manejo de usuarios y administración. Node.js ofrece escalabilidad y buen rendimiento.	Desarrollo del backend: ~30% del esfuerzo total. Requiere al menos 1 desarrollador backend con 3 meses de trabajo.
Lenguaje Frontend	HTML5, CSS3, JavaScript (React o Vue)	Tecnologías estándar para interfaz web moderna y responsive. React permite componentes reutilizables y alto rendimiento.	Frontend: ~25% del esfuerzo total. Al menos 1 desarrollador frontend.
Base de datos	PostgreSQL o MySQL	Sistemas relacionales confiables, ideales para gestión de usuarios, cursos, compras y progreso.	Licencia gratuita (open source).
Entorno de desarrollo	VS Code	IDEs populares y compatibles con los lenguajes seleccionados.	Sin costo en versiones Community. Instalación en todos los equipos de desarrollo.

Componentes de hardware

Componente	Análisis Cualitativo	Análisis Cuantitativo
Servidor Web / Hosting	Debe ser capaz de ejecutar servicios web 24/7, con soporte HTTPS y almacenamiento de base de datos. Se recomienda usar AWS, Azure o Google Cloud.	Recurso inicial: 2 CPU, 4 GB RAM, 80 GB almacenamiento. Costo mensual estimado: 30–50 USD.
Equipos de desarrollo	Computadoras para el equipo técnico (programadores, diseñador, tester).	Mínimo: CPU i5, 8 GB RAM, 256 GB SSD. Costo aprox. 600–800 USD por equipo.
Red / Conectividad	Internet estable, necesaria para sincronización con repositorios y despliegues en la nube.	Conexión ≥ 100 Mbps, costo promedio 40 USD/mes.

Servicio TI externos

Servicio	Análisis Cualitativo	Análisis Cuantitativo
Pasarela de pago (Stripe, PayPal, WebPay)	Permite procesar pagos automáticos, simulados o reales. Integración vía API.	Comisión por transacción: 2.9% + 0.30 USD.
Correo electrónico (SendGrid, Mailgun o Gmail API)	Para notificaciones automáticas de registro, compra y certificados.	100–500 correos gratuitos al mes; luego ~10 USD por cada 10.000.
Plataforma de video (YouTube, Vimeo o Zoom)	Alojamiento externo para cursos grabados o en vivo. Evita sobre costos de almacenamiento.	Uso gratuito o plan premium desde 15 USD/mes por instructor.

Herramientas de gestión y documentación

Herramienta	Análisis Cualitativo	Análisis Cuantitativo
Trello / Jira / Notion	Gestión ágil de tareas, backlog y control del avance del proyecto.	Gratis o 10–15 USD/mes por usuario (según plan).
Google Workspace / Microsoft 365	Comunicación y colaboración en línea (documentos, reuniones, etc.).	Desde 6 USD/mes por usuario.

b) Selección de las herramientas. Lenguaje frontend.js con React.js

Selección:

Se selecciona React.js para la construcción del frontend interactivo.

Justificación técnica:

1. Permite crear interfaces dinámicas y modulares, con componentes reutilizables.
2. Excelente integración con API REST (backend Django).
3. Facilita el desarrollo de una interfaz responsive compatible con dispositivos móviles.

Justificación económica:

1. Framework gratuito y open source.

2. Amplia documentación y disponibilidad de desarrolladores.

Base de datos: PostgreSQL

Selección:

Sistema de base de datos relacional PostgreSQL para almacenar usuarios, cursos, compras, progreso y certificados.

Justificación técnica:

1. Cumple con estándares ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad).
2. Soporta consultas complejas, ideal para métricas de finalización o ventas.
3. Integración directa con Django ORM.

Justificación económica:

1. Licencia gratuita (open source).
2. Soporte en múltiples servicios cloud (AWS RDS, Azure Database, etc.).

Pasarela de pago: Stripe / PayPal / WebPay

Selección:

Uso de Stripe o WebPay (dependiendo del país) para procesar pagos en línea.

Justificación técnica:

1. APIs seguras y bien documentadas.
2. Permite simular pagos en entorno de pruebas.
3. Facilita activación automática de cursos tras validación del pago.

Justificación económica:

1. Sin costos fijos; solo comisión por transacción (2.9% + 0.30 USD).

Plataforma de video embebido: YouTube / Vimeo / Zoom

Selección:

Integración de contenidos externos mediante videos embebidos o enlaces privados para cursos en vivo.

Justificación técnica:

1. Evita almacenamiento local de video (reduce carga del servidor).

2. Permite emitir clases en vivo o grabadas

Justificación económica:

1. Gratuito (YouTube) o bajo costo (Vimeo, Zoom desde 15 USD/mes).

Lenguaje de programación backend: Python con Django

Selección:

Se opta por Python (framework Django) para el desarrollo del backend, dado que ofrece un ecosistema maduro, seguro y con una amplia comunidad. Django incorpora autenticación, manejo de sesiones, ORM para base de datos y panel de administración nativo.

Justificación técnica:

1. Facilita la gestión de usuarios, pagos y accesos de forma estructurada.
2. Reduce los tiempos de desarrollo por su enfoque “batteries included”.
3. Altamente compatible con PostgreSQL y servicios cloud (AWS, Azure, Heroku).

3. Integración de las tecnologías.

a) Selección de los proveedores de tecnología.

Infraestructura y hosting en la nube

Proveedor	Costo aproximado	Evaluación global	Desventajas
Amazon Web Services (AWS)	EC2 (servidores), RDS (bases de datos), S3 (almacenamiento), SES (correo).	Altamente escalable, soporte global, gran variedad de servicios, integración con Django.	Configuración compleja, curva de aprendizaje alta.
Microsoft Azure	Hosting, SQL Database, App Services, Storage, Mail.	Integración fácil con entornos Windows, buena documentación.	Ligeramente más costoso que AWS.

Google Cloud Platform (GCP)	Compute Engine, Cloud SQL, Firebase Hosting.	Muy buena integración con frontend (React/Firebase).	Límite de almacenamiento gratuito bajo.
Heroku (Salesforce)	Despliegue rápido de apps Django/Node.js.	Ideal para MVPs, configuración simple, despliegue automático.	Menor control de infraestructura, limitado para gran escala.

Pasarela de pago

Proveedor	Ventajas	Desventajas	Costos / Comisiones
Stripe	Fácil integración con Django y React, soporte global, panel de control intuitivo.	No disponible en todos los países latinoamericanos.	2.9% + 0.30 USD por transacción.
PayPal	Reconocimiento mundial, alta seguridad.	Integración más compleja, mayores tarifas.	3.5% + 0.30 USD.
WebPay (Transbank - Chile)	Local, adaptado al mercado chileno, soporte a bancos nacionales.	Documentación limitada, integración más manual.	2.5–3.0% por transacción.
MercadoPago	Muy usado en Latinoamérica, fácil integración.	Comisiones variables, retención de fondos en algunos casos.	4–5% por transacción.

Herramientas del desarrollo y gestión de proyectos

Proveedor	Herramienta	Ventajas	Costo
GitHub	Control de versiones y repositorios.	Gratuito, integración con CI/CD.	Gratis o 4 USD/usuario/mes.
Trello / Notion	Gestión de tareas y backlog.	Visual, simple, adaptable a equipos pequeños.	Gratis o 10 USD/usuario/mes.
Slack / Discord	Comunicación del equipo.	Eficiente y multiplataforma.	Gratis (básico) o 6 USD/usuario/mes.

Plataforma de video y contenido multimedia

Proveedor	Ventajas	Desventajas	Costo estimado
-----------	----------	-------------	----------------

YouTube (privado o no listado)	Gratuito, alta disponibilidad, sin límite de almacenamiento.	Control limitado sobre privacidad y branding.	Gratis.
Vimeo Pro	Control de acceso, alta calidad de video, sin publicidad.	Costo mensual, límites de almacenamiento.	Desde 15 USD/mes.

b) Descripción de la seguridad y calidad.

Enfoque en la seguridad

El sistema está diseñado bajo un esquema de seguridad integral que abarca la protección de datos, el control de accesos y la comunicación segura entre los diferentes componentes:

Autenticación y control de acceso

Se implementa un sistema de autenticación basado en JSON Web Tokens (JWT), que permite identificar y controlar el acceso de cada usuario según su rol (administrador, instructor, estudiante o visitante). Esto garantiza que solo los usuarios autorizados puedan acceder a los cursos comprados o a los paneles de administración.

Cifrado y protección de contraseñas

Las credenciales de usuario se almacenan cifradas mediante algoritmos seguros como bcrypt o Argon2, evitando vulnerabilidades ante accesos no autorizados.

Comunicación segura (HTTPS)

Todo el tráfico entre el cliente (navegador) y el servidor se realiza bajo el protocolo SSL/TLS, protegiendo la confidencialidad de los datos financieros y personales durante las transacciones

Seguridad en pagos

Al integrarse con pasarelas de pago certificadas (WebPay o Stripe), se delega la gestión de datos sensibles a proveedores que cumplen con los estándares PCI DSS, garantizando la seguridad de las operaciones financieras.

Protección del contenido educativo

El acceso a los videos y materiales está controlado mediante enlaces embebidos privados (YouTube no listado o Vimeo privado), reduciendo el riesgo de distribución no autorizada.

Enfoque en la calidad del software

La calidad del sistema se aborda desde las fases de diseño y desarrollo, aplicando buenas prácticas de ingeniería y herramientas que aseguran la mantenibilidad y la estabilidad del producto final:

Arquitectura modular (Django + React)

La separación entre frontend y backend permite una mayor mantenibilidad, escalabilidad y facilidad de actualización sin afectar al sistema completo.

Pruebas automatizadas y control de versiones

El uso de GitHub y pruebas unitarias en Django permite detectar errores antes del despliegue y mantener la trazabilidad de los cambios.

Estandarización del código

Se siguen convenciones internacionales (PEP8 para Python, ESLint para JavaScript), lo que mejora la legibilidad y reduce defectos.

Gestión ágil y control de calidad continua

Herramientas como Trello o Jira facilitan la planificación, priorización de tareas y revisión periódica del avance, fomentando un ciclo de mejora continua.

4. Implementación de la solución.

a) Selección de plataformas.

Para implementar la solución se necesitarán variadas herramientas y plataformas que permitan el correcto desarrollo de aplicaciones web.

- **FrontEnd: React.js**
Se utilizará react gracias a su capacidad para crear interfaces dinámicas, modulares y compatibles con múltiples dispositivos.
- **Backend: Python con Django**
Es un framework seguro que facilita la gestión de usuarios, autenticación, administración y conexión con bases de datos relacionales.
- **Base de datos: PostgreSQL**
Sistema de gestión de base de datos relacional con alta compatibilidad con Django. Permite almacenar información sobre los usuarios, pagos y certificados.
- **Hosting: Heroku**

Esta plataforma ofrece completo soporte para Django, con escalabilidad automática y respaldo en la nube.

- Pasarela de pago: Webpay (Transbank)
Permite el procesamiento seguro de los pagos, con validación automática de transacciones y es configurable para la activación inmediata de los cursos.
- Plataforma de video: Youtube (video no listado)
Es una solución gratuita para la publicación y reproducción segura del contenido educativo, evitando también sobrecarga en el servidor.
- Gestión del proyecto: Github
Se utilizará para el control de versiones y seguimiento del progreso de desarrollo.

b) Construcción de prototipo de la solución.

5. Aplicación de métodos, estándares y buenas prácticas.

a) Cumplimiento en los criterios de aceptación.

Pruebas de interfaz: El sistema les permite a los usuarios poder visualizar un catálogo de los respectivos cursos grabados y en vivo con sus respectivos descripción y el precio correspondiente.

Pruebas de Ingreso: Los usuarios pueden registrarse, iniciar sesión y poder acceder según su respectivo rol (visitantes, estudiantes, instructores o administradores).

Pruebas de autenticación: Solo los estudiantes que compraron el curso pueden acceder a su contenido.

Simulación de pago: El estudiante puede realizar una compra en línea y obtener el acceso inmediato al curso.

Pruebas de seguimiento: El estudiante puede marcar módulos como completados y puede visualizar el progreso.

Verificación de base de datos: Las contraseñas deben de almacenarse cifradas.

Pruebas de carga: La plataforma soporta al menos 100 usuarios de manera simultánea.

Comparación con proceso mensuales: Se reduce el tiempo de validación de pagos y entrega de accesos.

b) Cumplimiento de estándares normativos.

ISO/IEC 25010 (calidad de software): Define los atributos como funcionalidad, fiabilidad, seguridad, usabilidad, rendimiento y mantenimiento.

ISO/IEC 830 (requisitos de software): Guía para redactar especificaciones funcionales o no funcionales. [2]

ISO/IEC 27001 (seguridad de la información): Se aplican los principios de confidencialidad, integridad y disponibilidad en el manejo de datos personales y pagos.

ISO/IEC 12027 (ciclo de vida del software): Estructura del proceso de análisis, diseño, implementación, pruebas y mantenimiento.

c) Cumplimiento de buenas prácticas.

Control de versiones con GitHub: se utiliza un repositorio centralizado para mantener trazabilidad de los cambios, control de ramas y revisión de código colaborativa.

Metodología ágil (Scrum): se organizan iteraciones semanales que permiten evaluar el progreso y priorizar tareas de acuerdo con los requerimientos del cliente.

Estandarización del código: se aplican normas como PEP8 (Python) y ESLint (JavaScript) para garantizar un estilo uniforme, mejorar la legibilidad y prevenir errores comunes.

Documentación técnica continua: se mantiene documentación actualizada de la arquitectura, los casos de uso y los diagramas de componentes, lo que facilita la futura escalabilidad.

Pruebas automatizadas y validación manual: antes de cada despliegue se realizan pruebas unitarias en Django y pruebas de integración con la API para garantizar la estabilidad.

d) Cumplimiento de rendimiento y eficiencia.

Optimización del backend Django: uso de consultas ORM optimizadas y almacenamiento en caché para reducir la carga del servidor.

Frontend eficiente con React.js: renderizado por componentes y carga dinámica (lazy loading) para mejorar la velocidad de respuesta.

Compresión de archivos y minificación: aplicación de técnicas de compresión GZIP y minificación de recursos CSS/JS para acelerar la carga del sitio.

Uso de CDN (Content Delivery Network): permite distribuir los recursos estáticos y videos de manera eficiente, reduciendo la latencia en distintos dispositivos.

Monitoreo de rendimiento: integración de herramientas como Heroku Metrics o Google Lighthouse para evaluar tiempo de carga, consumo de recursos y experiencia del usuario.

e) Cumplimiento de Seguridad.

Autenticación basada en JWT (JSON Web Token): cada usuario se valida mediante un token único, evitando accesos no autorizados.

Cifrado de contraseñas: uso de algoritmos seguros como bcrypt o Argon2 para el almacenamiento en la base de datos.

Comunicación cifrada: todo el tráfico entre cliente y servidor se realiza bajo el protocolo HTTPS (TLS/SSL).

Pagos seguros: integración con pasarelas certificadas (WebPay o Stripe) que cumplen con los estándares PCI DSS para el manejo de información financiera.

Control de roles y permisos: los usuarios solo pueden acceder a las funciones correspondientes a su rol (administrador, instructor o estudiante).

[illegible]

IV. Conclusiones

El trabajo desarrollado ha permitido establecer una base sólida y bien fundamentada para la construcción de la plataforma de ventas de cursos. La definición de la arquitectura de tres zonas (Seguridad DMZ, Aplicación y Datos) garantiza una estructura modular, escalable y con seguridad distribuida, facilitando el desarrollo y la mantenibilidad a largo plazo.

La selección de la pila tecnológica, basada en Python con Django y React.js, se justifica no solo por la robustez y seguridad que ofrece el framework de backend, sino también por la eficiencia y el alto rendimiento que permite la creación de interfaces dinámicas en el frontend. El uso de PostgreSQL y la integración de servicios externos como Stripe/WebPay y plataformas de video embebido, demuestran un enfoque en la reducción de costos operativos y la delegación de responsabilidades complejas (como la seguridad de pagos) a proveedores especializados.

Desde la perspectiva de seguridad y calidad, el diseño incorpora prácticas esenciales como la autenticación con JWT, el cifrado de contraseñas y el cumplimiento de estándares internacionales (ISO/IEC 25010, 27001), lo que asegura la protección de los datos de los usuarios y la fiabilidad del sistema.

Como aportes al proyecto, se ha establecido un conjunto claro de criterios de aceptación y se han definido las herramientas clave (Heroku para hosting y Webpay para pagos) que orientarán directamente la fase de construcción del prototipo. A futuro, sería beneficioso profundizar en el diseño detallado de los microservicios si el proyecto escalara a grandes volúmenes de usuarios, y evaluar una estrategia de migración multicloud, más allá de Heroku, para optimizar costos de infraestructura a largo plazo.

V. Referencias bibliográficas

ISO/IEC 27001 (Seguridad de la Información) International Organization for Standardization. (2022).

ISO/IEC 27001: Information security, cybersecurity and privacy protection — Information security management systems — Requirements. ISO. Recuperado de <https://www.iso.org/standard/81223.html>

ISO/IEC 25010 (Calidad de Software) International Organization for Standardization. (2018). ISO/IEC 25010: Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models. ISO. Recuperado de <https://www.iso.org/standard/69921.html>

ISO/IEC 830 (Requisitos de Software) International Organization for Standardization. (1993). ISO/IEC 830: Information technology — Specification of software requirements. ISO. Recuperado de <https://www.iso.org/standard/5225.html>

ISO/IEC 12027 (Ciclo de Vida del Software) International Organization for Standardization. (1998). ISO/IEC TR 12027: Information technology — Guide for the application of the process assessment standard to the improvement of the software life cycle processes. ISO. Recuperado de <https://www.iso.org/standard/20261.html>

ProcessOn. (2025, febrero 19). ¿Qué es un diagrama de topología de red? Conceptos, tipos y herramientas de dibujo. <https://www.processon.io/es/blog/network-topology-es>

Visual Paradigm. (s.f.). Software de diagramas de red. <https://online.visual-paradigm.com/es/diagrams/features/network-diagram-software/>

MyMap.AI. (s.f.). Creador de diagramas de red gratuito. <https://www.mymap.ai/es/network-diagram-maker>

Vargas, B. (2024, febrero 13). Guía completa sobre la creación de APIs: Paso a paso para desarrollar tu propia interfaz de programación. <https://www.byronvargas.com/web/como-se-crea-un-api/>

Lucidchart. (s.f.). Crear web API paso a paso. <https://www.lucidchart.com/blog/es/como-crear-una-api>

Apidog. (2025, junio 23). Cómo crear una API: Guía ilustrada paso a paso. <https://apidog.com/es/blog/how-to-build-an-api-a-step-by-step-illustrated-guide-es/>

Wix.com.. (s.f.). Crear página web gratis. <https://es.wix.com/>

Canva. (s.f.). Crea páginas web gratis con Canva. <https://www.canva.com/es-es/constructor-sitios-web/>

WebCreativa.cl.. (s.f.). Diseño web para todo Chile. <https://webcreativa.cl/>

