**Asignatura: Ingeniería de Software**

**Sección: TI3043/D-IINF-N4-P2-C1/D**

**Nombre del docente:** José Luis Martinez Opazo

**Nombre de los integrantes del grupo:**

* **Cristopher Bocanegra**
* **Felipe Olmos**
* **Guillermo Cook**

**Fecha de entrega: 18/11/2025**

Kuromanga Store

Implementación, pruebas y calidad de software

Contenido

1. **Exploración de Tendencias Tecnológicas3**
2. Confiabilidad y Ética5
3. Construcción del Prototipo Funcional6
4. **Corrección y Testing del Prototipo11**
5. Referencias Bibliográficas12
6. Exploración de Tendencias Tecnológicas
   1. **Investigación de Conceptos Clave**

**Software as a Service (SaaS)**

* **Características:** Acceso vía navegador, modelo de suscripción, mantenimiento centralizado en un servidor remoto.
* **Ejemplos:** Salesforce, Google Workspace, Office 365.
* **Aplicación en el proyecto:**
  + El E-commerce podría ofrecer un **modelo SaaS para vendedores independientes** (marketplace), cobrando suscripción mensual o comisión.
  + También podría aplicarse como **panel administrativo accesible desde cualquier dispositivo**.

**Infrastructure as a Service (IaaS)**

* **Características:** Infraestructura virtualizada, pago por uso, alta escalabilidad.
* **Ejemplos:** AWS EC2, Google Compute Engine, Azure VM.
* **Aplicación en el proyecto:**
  + Hospedaje del backend en un **servidor virtual EC2**.
  + Almacenamiento de imágenes de productos en **AWS S3**.
  + Base de datos en **RDS MySQL**.

**Platform as a Service (PaaS)**

* **Características:** Plataformas listas para desplegar código, CI/CD integrado, escalabilidad automática.
* **Ejemplos:** Heroku, Firebase, Azure App Service.
* **Aplicación en el proyecto:**
  + Backend desplegado en **Render o Railway**.
  + Despliegue rápido de prototipos sin configurar infraestructura manualmente.
  + Integración automática con pipelines.
  1. **Análisis de Integración de Tecnologías**

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

**Decisiones arquitectónicas tomadas:**

* Frontend como **SaaS en Vercel** debido a velocidad de despliegue.
* Backend como **PaaS en Railway** por simplicidad para prototipos.
* Base de datos **MySQL en cloud** para disponibilidad y respaldo automático.
  1. Propuesta de Soluciones Cloud

**Arquitectura Propuesta:**

* **Frontend (React + TSX + Tailwind):** hospedado en **Vercel**
* **Backend (Express.js):** desplegado en **Railway / Render**
* **Base de datos MySQL:** Railway MySQL, PlanetScale o AWS RDS
* **Almacenamiento de imágenes:** AWS S3 o Cloudinary

**Seguridad y Privacidad Implementadas:**

* HTTPS obligatorio
* Encriptación de contraseñas con bcrypt
* Tokens JWT con expiración
* Logs de acceso
* Separación de roles (administrador / usuario)

**Evaluación de Proveedores Cloud:**

* **Vercel:** excelente para frontend estático/dinámico
* **Railway:** simple y rápido para Express.js
* **AWS:** robusto, pero más complejo para un prototipo

1. Confiabilidad y Ética
   1. **Análisis de Confiabilidad**

**Disponibilidad**

* SLA objetivo: **99%**
* Redundancia del backend mediante reinicios automáticos del proveedor
* Monitoreo con uptime robot o Railway health checks

**Integridad**

* Validación en backend de todos los datos enviados desde el frontend
* Integridad referencial en MySQL con claves foráneas
* Auditoría con logs de compras, cambios de stock y accesos

**Seguridad**

* Aplicación de principios “**security by design**”
* Autenticación con JWT
* Autorización basada en roles
* Prevención de ataques conocidos: XSS, SQL Injection, CSRF
  1. **Revisión de Códigos de Ética**

**ACM Code of Ethics**

* Priorizar bienestar del usuario y transparencia en manejo de datos
* Evitar prácticas engañosas (dark patterns)
* Sistemas accesibles e inclusivos

**IEEE Code of Ethics**

* Protección de datos personales
* Honestidad en recolección y uso de información
* Mejora continua de competencia técnica del equipo
  1. **Consideraciones Legales**
* Cumplimiento de **Ley de Protección de Datos Personales.**
* Almacenar datos mínimos necesarios.
* Se respetan **Derechos de Autor**: Imágenes de productos originales o con licencia.
* Condiciones de uso y política de privacidad vinculada al login.

1. Construcción del Prototipo Funcional
   1. **Preparación del Entorno**

* Repositorio GitHub con ramas: kuromanga-store, frontend/, backend/
* Pipeline CI/CD en Vercel (frontend) y Railway (backend)
* Entornos separados:
  + Desarrollo
  + Testing
  + Producción
  1. **Implementación Incremental**

Iteraciones desarrolladas:

1. Frontend básico:

Home: Interfaz de usuario gráfica, Texto, Aplicación, Sitio web

El contenido generado por IA puede ser incorrecto.

Catálogo: Interfaz de usuario gráfica, Aplicación, Sitio web

El contenido generado por IA puede ser incorrecto.

Detalle de producto: Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

1. Autenticación

Login:

Interfaz de usuario gráfica, Aplicación, Teams

El contenido generado por IA puede ser incorrecto.

Registro:

Interfaz de usuario gráfica, Aplicación, Teams

El contenido generado por IA puede ser incorrecto.

Perfiles:

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

1. Carrito y pagos simulados

Carrito: Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Con Productos:

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

* 1. **Buenas Prácticas OWASP**
* **A01 – Broken Access Control:**
  + Middleware para validar JWT en rutas protegidas
  + Roles admin vs usuario
* A02 – Cryptographic Failures:
  + HTTPS y contraseñas hasheadas
  + JWT con expiración y refresh tokens
* A03 – Injection:
  + Consultas parametrizadas en MySQL
  + Santización de inputs
  1. **Ley de Delitos Informáticos**
* Registro de intentos de accesos fallidos
* Prevención de acceso no autorizado
* Códigos y logs almacenados con integridad
  1. **Justificación de Decisiones Técnicas**
* React + TSX: Desarrollo modular y rápido.
* Express.JS: Ligereza y flexibilidad.
* MySQL: Estructura relacional e ideal para catálogo.
* TailwindCSS: Velocidad y Consistencia Visual.
* Vercel/Railway: Despliegue simple y eficiente.

1. Corrección y Testing del Prototipo
   1. **Pruebas Funcionales**

**Pruebas Unitarias:**

* Framework: Jest
* Cobertura: 85% del backend

**Pruebas de Integración:**

* Validación de endpoints: Auth, Productos, Carrito

**Pruebas del Sistema:**

* Flujo completo de compra probado
* Performance: 200 – 300 ms promedio por request.
  1. **Pruebas de Seguridad**
* Pentesting con OWASP ZAP
* Resultados:
  + Vulnerabilidades Críticas: 0
  + Vulnerabilidades medias: 1 (Corregida)
  1. **Cumplimiento ISO-27000**
* Políticas de seguridad documentadas
* Control de Accesos Implementado
* Evaluación de Riesgos Inicial Realizada
  1. **Corrección por Defectos**
* Se priorizó por severidad
* Se corrigieron los errores de cálculo del carrito
* Se ajustaron validaciones de inputs
  1. **Documentación de Resultados**

**Reporte de Testing del Proyecto:**

* Pruebas Unitarias: 54/58 exitosas (93%)
* Pruebas de Integración: 22/24 (91%)
* Vulnerabilidades Críticas: 0
* Promedio de Respuesta API: 230 ms
* Disponibilidad estimada: 99%

1. Referencias bibliográficas

React. (2024). React documentation. Meta Platforms. <https://react.dev>

TailwindCSS. (2024). TailwindCSS documentation. Tailwind Labs. <https://tailwindcss.com>

Node.js Foundation. (2024). Node.js documentation. <https://nodejs.org>

Express.js. (2024). Express documentation. OpenJS Foundation. <https://expressjs.com>

MySQL. (2024). MySQL 8.0 Reference Manual. Oracle Corporation. <https://dev.mysql.com/doc>

JWT. (2024). JSON Web Tokens documentation. Auth0. <https://jwt.io>

OWASP Foundation. (2024). OWASP Top Ten 2021: The Ten Most Critical Web Application Security Risks. <https://owasp.org/www-project-top-ten>

ISO. (2013). ISO/IEC 27001:2013 – Information Security Management Systems. International Organization for Standardization.

ISO. (2022). ISO/IEC 27002:2022 – Information Security Controls. International Organization for Standardization.

Stallings, W. (2021). Cryptography and Network Security: Principles and Practice (8th ed.). Pearson.

Sharma, R. (2020). Web Application Security: Exploitation and Countermeasures for Modern Web Applications. Apress.

Northwood, T. (2022). Securing Node Applications: Best Practices for Server-Side JavaScript. O’Reilly Media.

Amazon Web Services. (2024). AWS Cloud Documentation. https://aws.amazon.com/documentation

Google Cloud. (2024). Google Cloud Documentation. <https://cloud.google.com/docs>

Microsoft Azure. (2024). Azure Documentation. <https://learn.microsoft.com/azure>

Heroku. (2024). Heroku Developer Documentation. Salesforce. <https://devcenter.heroku.com>

Myers, G. J., Sandler, C., & Badgett, T. (2011). The Art of Software Testing (3rd ed.). Wiley.

Jest. (2024). Jest documentation. Meta. <https://jestjs.io>

OWASP Foundation. (2017). OWASP Web Security Testing Guide. <https://owasp.org/www-project-web-security-testing-guide>

ACM. (2018). ACM Code of Ethics and Professional Conduct. Association for Computing Machinery. <https://www.acm.org/code-of-ethics>

IEEE. (2020). IEEE Code of Ethics. Institute of Electrical and Electronics Engineers. https://www.ieee.org/content/ieee-code-of-ethics

Gobierno de Chile. (2020). Ley N.º 19.628 sobre Protección de la Vida Privada. Biblioteca del Congreso Nacional.