# DISEÑO, DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA PARA
# EL APRENDIZAJE DE LINUX

## Aplicación de la Norma ISO/IEC/IEEE 15288:2015

---

**Integrantes:**

- David Santiago Chacón Herrera
- Cristian Cortés Bejarano
- Juanita Campos Cárdenas
- Danna Valentina Segura Forigua

**Asignatura:** Desarrollo de Aplicaciones Web

**Docente:** Luisa Felipe Herrera Quintero

**Universidad Piloto de Colombia**
Facultad Escuela TIC

Bogotá, 19 de septiembre de 2025

---

## Tabla de Contenidos

---

## Resumen

Se propone crear una aplicación web interactiva que facilite el aprendizaje de Linux a usuarios novatos. El sistema ofrecerá un entorno visual y lúdico que simule una terminal, brindando retroalimentación gráfica inmediata tras cada comando. Esto contrasta con la línea de comandos tradicional, la cual suele ser compleja y confusa para principiantes. Al proporcionar ejercicios prácticos con respuestas visuales, se busca aumentar la motivación y el compromiso de los estudiantes en el proceso de aprendizaje.

**Palabras clave:** Linux, educación interactiva, simulador de terminal, gamificación, aprendizaje web.

## Introducción

Linux es un sistema operativo libre y gratuito, ampliamente utilizado en servidores web, dispositivos móviles como Android, entre otras aplicaciones. Al ser de código abierto, permite una amplia personalización y existen distribuciones adaptadas a diferentes necesidades, incluyendo las educativas. Sin embargo, el aprendizaje inicial de Linux suele ser tedioso porque la interacción típica es por la línea de comandos. A pesar de que distribuciones de Linux como Debian tienen una interfaz amistosa con el usuario, si una persona quisiera aprender a usar la terminal mediante comandos, se frustraría con la retroalimentación que esta tiene. Por tanto, surge la necesidad de métodos alternativos interactivos donde el aprendizaje sea más accesible y atractivo para los usuarios.

## Marco Teórico

### Antecedentes y Justificación

#### Mercado y Oportunidad

Según estudios recientes del mercado tecnológico:

- Linux lidera el mercado de servidores con 62.7% de participación
- Linux ocupa más del 96% de los servidores web más importantes del mundo
- Linux alimenta aproximadamente el 91.5% de las supercomputadoras del mundo y el 70% de los servidores web

- Se proyecta que el mercado global de Linux alcanzará $15.64 billones para 2027

**Barrera de Entrada Actual**

A pesar de su dominio en servidores y sistemas empresariales:

- Linux solo tiene 3.9% de participación en el mercado de escritorio (comparado con 71.83% de Windows)
- En escritorio, Linux mantiene apenas un 2.09% de participación

**Demanda de Aprendizaje**

Los datos muestran una creciente necesidad de formación en Linux:

- Linux alimenta el 85% de los smartphones y ejecuta el 39.2% de los sitios web mundiales
- Existe un crecimiento constante: Linux alcanzó un récord histórico de 4.45% en julio 2024, subiendo desde 4.05% en junio
- Los estudiantes que fueron enseñados usando gamificación basada en desafíos aumentaron su rendimiento hasta un 89.45% comparado con aquellos que solo recibieron conferencias tradicionales (Negus, 2021)

## Planteamiento del Problema

El aprendizaje de un sistema operativo como Linux se ha convertido en una necesidad para estudiantes y profesionales en áreas como la administración de sistemas, la programación, la ciberseguridad y el manejo de servidores. Sin embargo, el acceso a este conocimiento se encuentra limitado debido a que los recursos existentes presentan diversas deficiencias:

1. Muchos cursos en línea son de pago, lo que restringe el acceso a quienes no cuentan con los recursos económicos suficientes
2. Los manuales gratuitos suelen ser teóricos y carecen de interactividad
3. La información disponible está dispersa, sin un orden pedagógico que facilite el avance progresivo del estudiante

A esto se suma que practicar directamente en un sistema Linux real puede generar riesgos técnicos, lo que desalienta a los principiantes y aumenta la curva de aprendizaje. En consecuencia, existe la necesidad de contar con una aplicación web gratuita, interactiva y estructurada que permita aprender Linux de manera segura, práctica y accesible.

El problema central que este proyecto busca resolver es:

**¿Cómo diseñar, desarrollar e implementar un sistema interactivo que responda a la necesidad de aprender Linux de manera segura, práctica y accesible para estudiantes y profesionales?**

# Objetivos

## Objetivo General

Diseñar, desarrollar e implementar una aplicación web responsiva y accesible que simule un entorno de terminal Linux para usuarios novatos, proporcionando lecciones estructuradas y ejercicios interactivos con retroalimentación gráfica inmediata, garantizando seguridad, usabilidad y evaluación del aprendizaje mediante pruebas con usuarios y métricas cuantificables.

## Objetivos Específicos

### 1. Analizar el estado del arte y definir requisitos funcionales y no funcionales

**Acción:** Investigar y comparar al menos 6 plataformas/sistemas educativos (p. ej. Linux Journey, OverTheWire, VIM Adventures, tutoriales interactivos) y consolidar hallazgos en una matriz comparativa; con base en eso, redactar el Documento de Requisitos (SRS).

**Entregable:** Informe de análisis (matriz comparativa de características y costos) + Documento SRS (requisitos funcionales y no funcionales, restricciones, casos de uso básicos).

**Métricas/aceptación:** Comparación de 6 o más plataformas; SRS aprobado por el equipo; lista priorizada de requisitos (MUST/SHOULD/CAN) y 10 casos de uso básicos identificados.

**Responsable:** David Santiago Chacón Herrera
**Fase:** Primer corte

### 2. Diseñar la arquitectura técnica y el prototipo de interfaz (UI/UX)

**Acción:** Especificar arquitectura (diagramas: componentes, API, DB, integraciones con librerías de terminal), definir stack tecnológico (Vue.js en el frontend, Nest.js en el backend, base de datis PostgreSQL) y entregar prototipos de alta fidelidad de las pantallas claves.

Para la simulación de la terminal se construirá un simulador propio:

- El frontend (Vue.js) mostrará una interfaz similar a una terminal real
- El backend (Nest.js) procesará los comandos en un entorno controlado, interpretando únicamente un conjunto limitado (ej. ls, cd, cat, chmod, mkdir, rm)

**Entregable:** Diagrama de arquitectura (componentes y flujo), especificación de APIs, prototipos en Figma con pantallas: (a) inicio/registro, (b) panel de comandos, (c) visor de resultados/retroalimentación, (d) actividad/desafío, (e) panel docente/estadísticas.

**Métricas/aceptación:** Prototipos navegables (mín. 5 pantallas) + diagrama de arquitectura validado; lista de endpoints API con contratos (inputs/outputs).

**Responsable:** Juanita Campos Cárdenas (diseño front) y David Santiago Chacón Herrera (requerimientos) **Fase:** Segundo corte

**3. Implementar el backend y la lógica del simulador**

**Acción:** Desarrollar en Nest.js el servicio que interprete/emule comandos, ejecute operaciones en un simulador controlado (simulación), gestione usuarios, sesiones y guardado de progresos; exponer APIs definidas.

**Entregable:** Servicio backend desplegable (repositorio con README), API documentada (OpenAPI/Swagger), scripts de inicialización de BD.

**Métricas/aceptación:** 95% de endpoints documentados; 80% de comandos básicos implementados; tiempos de respuesta < 1.5 s para simulaciones sencillas; pruebas unitarias con cobertura mínima 70%.

**Responsable:** Cristian Cortés Bejarano
**Fase:** Tercer corte

**4. Implementar el frontend y la experiencia interactiva**

**Acción:** Construir la interfaz en Vue.js con el panel de comandos (simulador de terminal), área de visualización gráfica de resultados y sistema de navegación entre lecciones/retos.

**Entregable:** SPA funcional (build) alojable, integración con APIs, prototipo responsive validado.

**Métricas/aceptación:** Compatible con navegadores modernos y responsive en móvil/desktop; pruebas de accesibilidad básicas; pruebas end-to-end con éxito ≥ 90%.

**Responsable:** Juanita Campos Cárdenas (implementación front) con apoyo de Cristian Cortés Bejarano
(APIs)
**Fase:** Tercer corte

**5. Integrar simulador de terminal y manejar seguridad**

**Acción:** Integrar el simulador propio en el frontend con Vue.js y asegurar que las ejecuciones se realicen en un entorno aislado:

- Lista blanca de comandos permitidos
- Limitación de parámetros peligrosos
- Timeouts en ejecuciones largas
- Posibilidad de extender la simulación con contenedores controlados en el futuro

**Entregable:** Módulo de terminal integrado + documento de seguridad (lista de riesgos y controles).

**Métricas/aceptación:** 0 ejecuciones con acceso al sistema host; listas blancas/negras de comandos aplicadas; tiempo máximo por ejecución configurado; pruebas de penetración básicas.

**Responsable:** Cristian Cortés Bejarano (seguridad backend) y Juanita Campos Cárdenas (integración frontend)
**Fase:** Tercer corte

**6. Diseñar y producir contenido pedagógico (lecciones y ejercicios)**

**Acción:** Crear un plan de lecciones modular (niveles: básico e intermedio) con actividades, retos y retroalimentación automática.

- Cada lección estará alineada con los comandos soportados en el simulador propio (ej. ls, cd, cat, chmod, mkdir, rm)
- Se diseñarán al menos 20 ejercicios prácticos, cada uno con: enunciado, solución esperada, retroalimentación automática y criterios de evaluación
- El curso básico tendrá una duración estimada de 20 horas distribuidas en 4 semanas, mientras que los módulos intermedios cubrirán aproximadamente 30 horas adicionales
- El contenido estará basado en referencias confiables, principalmente *Linux Bible* (Negus, 2021)
- Se incorporarán elementos de gamificación (sistema de rachas, recompensas virtuales, estadísticas de progreso)

**Entregable:** Banco de ejercicios (en formato JSON/DB), guías de lección y rúbrica de evaluación asociada a cada comando/actividad.

**Métricas/aceptación:** 20 o más ejercicios clasificados por dificultad. Por cada ejercicio: enunciado, input esperado, output esperado y casos de prueba automáticos.

**Responsable:** Equipo (Danna Valentina Segura Forigua lidera contenido pedagógico y pruebas)
**Fase:** Segundo → Tercer corte

**7. Pruebas de usabilidad y validación pedagógica con usuarios reales**

**Acción:** Planificar y ejecutar pruebas con usuarios (n mínimo 15-20) para medir usabilidad (SUS o cuestionario equivalente), tasa de éxito en tareas, tiempo en completar ejercicios y mejora en post-test respecto a pre-test.

**Entregable:** Informe de pruebas (metodología, datos recogidos, análisis estadístico, lista de ajustes).

**Métricas/aceptación:** SUS ≥ 68 (o meta acordada); tasa de éxito en tareas ≥ 80% tras iteraciones; mejora promedio en puntuación de conocimientos ≥ 20% entre pre/post test.

**Responsable:** Danna Valentina Segura Forigua (coordinación pruebas) con apoyo del resto
**Fase:** Piloto dentro del Tercer corte; análisis final antes de entrega

**8. Despliegue, condiciones de operación y documentación final**

**Acción:** Desplegar la aplicación en un proveedor (Docker-AWS/hosting elegido), configurar dominio y SSL, crear guía de usuario y documentación técnica (README, manual de mantenimiento).

**Entregable:** URL operativa (o instrucción de despliegue reproducible), manual de usuario y manual técnico.

**Métricas/aceptación:** Aplicación accesible públicamente; SSL configurado; script CI/CD o guía paso a paso; checklist de entrega aprobado por el equipo.

**Responsable:** Cristian Cortés Bejarano (deploy) y David Santiago Chacón Herrera (documentación SRS/entregables)
**Fase:** Tercer corte (despliegue final)

**9. Evaluación final y reporte del proyecto**

**Acción:** Consolidar resultados técnicos y pedagógicos, comparar desempeño con métodos tradicionales (si hay datos), y redactar el informe final y la presentación para evaluación.

**Entregable:** Informe final (metodología, resultados cuantitativos y cualitativos, lecciones aprendidas) y presentación (slides).

**Métricas/aceptación:** Informe entregado con métricas clave: número de usuarios en piloto, SUS, % mejora en aprendizaje, cobertura de comandos, cobertura de pruebas automatizadas.

**Responsable:** Equipo (Danna Valentina Segura Forigua coordina la redacción del informe)
**Fase:** Entrega final (post Tercer corte)

## Hipótesis

La hipótesis principal es que una plataforma web interactiva de aprendizaje de Linux incrementará la motivación y la eficacia del aprendizaje en usuarios novatos, en comparación con métodos tradicionales. Se espera que al brindar retroalimentación visual inmediata los usuarios comprendan mejor los conceptos. El sistema permitirá cierto grado de personalización en el aprendizaje, adaptando la dificultad y los retos en función del progreso del estudiante. Se registrarán métricas de desempeño que permitirán ofrecer recomendaciones de ejercicios adicionales en caso de dificultad.

## Alcance

### Incluye:

- Desarrollo de una plataforma web accesible desde navegadores
- Lecciones básicas de Linux: navegación de archivos, permisos, gestión de paquetes, etc.
- Simulador de terminal en línea
-
-

Ejercicios interactivos, problemas y retos con retroalimentación gráfica

Diseño para aprendizaje de nivel inicial

### No incluye:

- No se desarrollará un nuevo sistema operativo ni una distribución completa de Linux
- No se abordarán temas de administración de servidores avanzados ni scripting complejo
- Tampoco funcionalidades empresariales más allá del aprendizaje didáctico
- No se centra en la gestión de entornos de producción

## Metodología

### Metodología de la Investigación

La investigación es de tipo **aplicada**, con enfoque **descriptivo y experimental**, orientada a resolver un problema práctico de aprendizaje de Linux mediante una aplicación web interactiva.

#### 1. Enfoque metodológico

Se combinará un enfoque descriptivo (análisis del estado del arte) y experimental (validación con usuarios).

#### 2. Método de investigación

Se adopta la investigación aplicada, integrando fundamentos pedagógicos con el desarrollo tecnológico.

#### 3. Técnicas e instrumentos

- Revisión documental de fuentes y plataformas existentes
- Encuestas y entrevistas a estudiantes
- Observación y pruebas de usabilidad

#### 4. Procedimiento

1. Análisis de requerimientos y antecedentes
2. Diseño de la arquitectura y prototipos
3. Desarrollo de la aplicación con tecnologías web
4. Implementación y pruebas piloto con usuarios

**5. Variables de estudio**

- **Independiente:** Uso de la aplicación
- **Dependiente:** Motivación y eficacia en el aprendizaje de Linux
- **Competencias:** Administración de archivos, gestión de permisos, navegación en directorios y manejo de comandos esenciales

**6. Análisis de datos**

Datos cuantitativos (estadísticas de uso, aciertos, tiempos) y cualitativos (opiniones y sugerencias).

**7. Validación**

Pruebas de funcionalidad, usabilidad y comparación con métodos tradicionales.

## Cronograma

### Primer Corte (Semanas 1-5)

- Análisis del estado del arte
- Definición de requisitos
- Documento SRS inicial
- Investigación de usuarios

### Segundo Corte (Semanas 6-10)

- Diseño de arquitectura
- Prototipos UI/UX
- Inicio de desarrollo de contenido pedagógico
- Especificación de APIs

### Tercer Corte (Semanas 11-16)

- Implementación backend
- Implementación frontend
- Integración de componentes
- Pruebas con usuarios
- Despliegue final
- Documentación completa

# Presupuesto

## Resumen de Costos

| Categoría | Descripción | Costo (COP) | Porcentaje |
|---|---|---|---|
| **Personal** | | | |
| Ingeniero de requerimientos (David) | Levantamiento de requisitos, análisis | $4,000,000 | 18.9% |
| Ingeniero de front-end (Juanita) | Desarrollo interfaz, diseño visual | $3,500,000 | 16.5% |
| Ingeniero de back-end (Cristian) | Desarrollo lógica del sistema, API | $2,500,000 | 11.8% |
| Ingeniero de pruebas (Valentina) | Planificación, ejecución de pruebas | $3,000,000 | 14.2% |
| **Subtotal Personal** | | **$13,000,000** | **61.5%** |
| **Infraestructura Tecnológica** | | | |
| Hosting en la nube (AWS, Azure) | 4 meses de servicio | $2,000,000 | 9.5% |
| Dominio web (.com/.org) | Registro anual | $120,000 | 0.6% |
| Certificado SSL | Seguridad y cifrado | $300,000 | 1.4% |
| Herramientas de desarrollo | IDEs, librerías, software | $1,000,000 | 4.7% |
| **Subtotal Infraestructura** | | **$3,420,000** | **16.2%** |
| **Capacitación y Certificaciones** | | | |
| Certificación Linux Foundation | Validación técnica (1 persona) | $1,200,000 | 5.7% |
| Cursos complementarios | Cursos online de apoyo | $800,000 | 3.8% |
| **Subtotal Capacitación** | | **$2,000,000** | **9.5%** |
| **Otros Costos** | | | |
| Materiales de apoyo | Papelería, manuales | $300,000 | 1.4% |
| Conectividad y energía | Internet y electricidad (4 meses) | $600,000 | 2.8% |
| Contingencias (10%) | Fondo de imprevistos | $1,832,000 | 8.7% |
| **Subtotal Otros** | | **$2,732,000** | **12.9%** |
| **Categoría** | **Descripción** | **Costo (COP)** | **Porcentaje** |
| **TOTAL GENERAL** | | **$21,152,000** | **100%** |

*Nota: Las contingencias se calcularon como el 10% del subtotal de las primeras tres categorías. Todos los valores están en pesos colombianos (COP) e incluyen IVA donde aplique.*

# Aplicación de la Norma ISO/IEC/IEEE 15288

## Mapeo de Procesos del Ciclo de Vida del Sistema

La presente sección describe la aplicación de la norma ISO/IEC/IEEE 15288:2015 — Sistemas e Ingeniería de Software — Procesos del Ciclo de Vida del Sistema, garantizando el cumplimiento de buenas prácticas de ingeniería de sistemas.

### Procesos Técnicos (Cláusula 6.4)

| Cláusula | Proceso | Implementación en el Proyecto |
|---|---|---|
| **6.4.1** | **Proceso de Análisis de Misión o Negocio** | Análisis de mercado y oportunidad educativa en Linux. Identificación de la brecha entre demanda de profesionales Linux y recursos de aprendizaje disponibles. ROI proyectado basado en modelo de sostenibilidad por publicidad. |
| **6.4.2** | **Proceso de Definición de Necesidades y Requisitos de las Partes Interesadas** | Planteamiento del problema, hipótesis y objetivos. Identificación de stakeholders: estudiantes novatos, profesionales en transición, instituciones educativas. Encuestas y entrevistas para captura de necesidades. |
| **6.4.3** | **Proceso de Definición de Requisitos del Sistema** | Objetivo específico 1: Documento SRS con requisitos funcionales, no funcionales, casos de uso, criterios de aceptación. Priorización MoSCoW (MUST/SHOULD/CAN/WON'T). Trazabilidad de requisitos. |
| **6.4.4** | **Proceso de Definición de la Arquitectura** | Objetivo específico 2: Arquitectura de 3 capas (presentación Vue.js, lógica Nest., datos PostgreSQL). Diagramas UML de componentes, secuencia y despliegue. Patrones de diseño: MVC, Repository, Factory. |
| **6.4.5** | **Proceso de Definición del Diseño** | Objetivos 3 y 4: Diseño detallado de módulos. Especificación de interfaces API REST. Diseño de base de datos (modelo E-R). Prototipos de UI/UX en Figma. |
| **6.4.6** | **Proceso de Análisis del Sistema** | Análisis de rendimiento esperado. Modelado de casos de uso críticos. Análisis de carga (usuarios concurrentes). Análisis de seguridad (OWASP Top 10). |
| **6.4.7** | **Proceso de Implementación** | Desarrollo iterativo con sprints de 2 semanas. Integración continua con Git. Pruebas unitarias (cobertura >70%). Code reviews obligatorios. |
| **6.4.8** | **Proceso de Integración** | CI/CD pipeline con Jenkins/GitHub Actions. Integración frontend-backend progresiva. Pruebas de integración automatizadas. Ambientes: desarrollo, staging, producción. |
| **6.4.9** | **Proceso de Verificación** | Pruebas automatizadas multinivel: unitarias, integración, E2E. Verificación contra SRS. Auditorías de código con SonarQube. Pruebas de rendimiento con JMeter. |
| **6.4.10** | **Proceso de Transición** | Despliegue en Docker/AWS. Migración de datos de prueba a producción. Capacitación de administradores. Plan de rollback documentado. |
| **6.4.11** | **Proceso de Validación** | Pruebas con usuarios (n=20). Métricas SUS ≥ 68. Pre-test y post-test de conocimientos. Análisis estadístico de mejora ≥20%. |
| **6.4.12** | **Proceso de Operación** | Monitoreo con herramientas APM. Logs centralizados (ELK Stack). Métricas de uso y engagement. Soporte nivel 1 y 2 definido. |

| Cláusula | Proceso | Implementación en el Proyecto |
|---|---|---|
| **6.4.13** | **Proceso de Mantenimiento** | Plan de mantenimiento preventivo/correctivo. Versionado semántico. Actualizaciones de seguridad mensuales. Backups automatizados diarios. |
| **6.4.14** | **Proceso de Eliminación** | Política de retención de datos (2 años). Procedimiento de exportación de datos de usuario. Desactivación gradual con notificación 60 días. Plan de archivo histórico. |

### Procesos de Gestión Técnica (Cláusula 6.3)

| Cláusula | Proceso | Implementación en el Proyecto |
|---|---|---|
| **6.3.1** | **Proceso de Planificación del Proyecto** | WBS detallado con 9 objetivos específicos. Cronograma de 16 semanas. Asignación de recursos y responsabilidades. Hitos y entregables definidos. |
| **6.3.2** | **Proceso de Evaluación y Control del Proyecto** | Reuniones semanales de seguimiento. Dashboard de métricas del proyecto. Control de cambios con CCB. Informes de progreso quincenales. |
| **6.3.3** | **Proceso de Gestión de Decisiones** | Matriz RACI para toma de decisiones. Registro de decisiones arquitecturales (ADR). Análisis de alternativas documentado. Criterios de decisión ponderados. |
| **6.3.4** | **Proceso de Gestión de Riesgos** | Registro de riesgos con matriz probabilidad/impacto. Planes de mitigación y contingencia. Riesgos principales: seguridad de comandos, adopción de usuarios, escalabilidad. Revisión semanal de riesgos. |
| **6.3.5** | **Proceso de Gestión de la Configuración** | Git con branching strategy (GitFlow). Versionado semántico. Control de cambios en requisitos. Gestión de releases documentada. |
| **6.3.6** | **Proceso de Gestión de la Información** | Wiki del proyecto en Confluence/Notion. Repositorio documental estructurado. Gestión de conocimiento con lecciones aprendidas. Backup de información crítica. |
| **6.3.7** | **Proceso de Medición** | KPIs del proyecto: velocidad, burn-down, deuda técnica. Métricas de producto: usuarios activos, tasa de completación, NPS. Métricas técnicas: disponibilidad, latencia, errores. |
| **6.3.8** | **Proceso de Aseguramiento de la Calidad** | Plan de calidad del proyecto. Auditorías de proceso mensuales. Revisiones técnicas formales. Gestión de no conformidades. |

### Procesos de Acuerdo (Cláusula 6.1)

| Cláusula | Proceso | Implementación en el Proyecto |
|---|---|---|
| **6.1.1** | **Proceso de Adquisición** | Selección de proveedores de hosting (RFP). Evaluación de herramientas de desarrollo. Contratos de licenciamiento de software. SLAs con proveedores cloud. |
| **6.1.2** | **Proceso de Suministro** | Entrega de la plataforma a usuarios finales. Términos de servicio y privacidad. Modelo freemium con publicidad. Soporte técnico definido. |

**Procesos Organizacionales Habilitadores del Proyecto (Cláusula 6.2)**

| Cláusula | Proceso | Implementación en el Proyecto |
|---|---|---|
| **6.2.1** | **Proceso de Gestión del Modelo de Ciclo de Vida** | Modelo iterativo incremental. Adaptación de Scrum con sprints de 2 semanas. Gates de calidad entre fases. Retrospectivas y mejora continua. |
| **6.2.2** | **Proceso de Gestión de la Infraestructura** | Infraestructura cloud (IaaS). Ambientes segregados (dev/test/prod). Herramientas de desarrollo estandarizadas. Gestión de licencias centralizada. |
| **6.2.3** | **Proceso de Gestión del Portafolio** | Alineación con objetivos educativos institucionales. Priorización de features por valor. Roadmap de evolución del producto. Análisis de portafolio de cursos. |

## Checklist de Artefactos Requeridos (ISO/IEC/IEEE 15288)

Para garantizar el cumplimiento completo de la norma, se han identificado los siguientes artefactos obligatorios:

- [ ] **Documentación de Gestión**
  - [ ] Plan de Proyecto (Project Management Plan)
  - [ ] Registro de Riesgos (Risk Register) con planes de mitigación
  - [ ] Matriz de Trazabilidad de Requisitos
  - [ ] Registro de Decisiones Arquitecturales (ADR)
  - [ ] Plan de Gestión de la Configuración
  - [ ] Plan de Aseguramiento de la Calidad (QA Plan)
  - [ ] Informes de Avance (Progress Reports) - mensuales
- [ ] **Documentación Técnica**
  - [ ] Documento de Especificación de Requisitos del Sistema (SRS) - versionado

Documento de Diseño de la Arquitectura del Sistema (SAD)

Especificación de Interfaces (IRS)

Documento de Diseño Detallado (DDD)

- [ ] Manual de Operación del Sistema
- [ ] Manual de Usuario Final

- [ ] **Artefactos de Desarrollo**

- [ ] Repositorio Git con control de versiones
- [ ] Código fuente documentado (JavaDoc, JSDoc)
- [ ] Scripts de base de datos versionados
- [ ] Configuración de CI/CD pipeline

Contratos de API (OpenAPI/Swagger)

Prototipos navegables en Figma

- [ ] **Documentación de Pruebas**

- [ ] Plan de Pruebas del Sistema
- [ ] Casos de Prueba documentados
- [ ] Reporte de Pruebas Unitarias (cobertura >70%)
- [ ] Reporte de Pruebas de Integración

Reporte de Pruebas de Usuario (SUS, métricas)

Reporte de Pruebas de Seguridad

- [ ] **Documentación de Validación**

- [ ] Protocolo de Validación con Usuarios
- [ ] Análisis Pre-test/Post-test
- [ ] Informe de Métricas de Usabilidad

Registro de Feedback de Usuarios

Certificado de Aceptación del Sistema

- [ ] **Documentación de Transición y Operación**

- [ ] Plan de Despliegue (Deployment Plan)
- [ ] Procedimientos de Respaldo y Recuperación

Plan de Continuidad del Negocio
SLA definidos y documentados
Manual de Mantenimiento
Política de Retención y Eliminación de Datos

# Gestión del Proyecto

## Estructura de Desglose del Trabajo (WBS)

1. Gestión del Proyecto
   1.1 Iniciación
   1.2 Planificación
   1.3 Ejecución y Control
   1.4 Cierre

2. Análisis y Diseño
   2.1 Análisis del Estado del Arte
   2.2 Definición de Requisitos
   2.3 Diseño de Arquitectura
   2.4 Diseño de UI/UX

3. Desarrollo
   3.1 Configuración del Entorno
   3.2 Desarrollo Backend
   3.3 Desarrollo Frontend
   3.4 Integración de Componentes

4. Contenido Pedagógico
   4.1 Diseño Curricular
   4.2 Creación de Lecciones
   4.3 Desarrollo de Ejercicios
   4.4 Sistema de Gamificación

5. Pruebas y Validación
   5.1 Pruebas Unitarias
   5.2 Pruebas de Integración
   5.3 Pruebas de Usuario
   5.4 Validación Pedagógica

6. Despliegue
   6.1 Configuración de Infraestructura
   6.2 Migración de Datos
   6.3 Puesta en Producción
   6.4 Documentación Final

## Matriz RACI

| Actividad | David (Req) | Cristian (Back) | Juanita (Front) | Valentina (QA) |
|---|---|---|---|---|
| Análisis de Requisitos | **R**/A | C | C | I |
| Diseño de Arquitectura | A | **R** | C | I |
| Desarrollo Backend | C | **R**/A | I | C |
| **Actividad** | **David (Req)** | **Cristian (Back)** | **Juanita (Front)** | **Valentina (QA)** |
| Desarrollo Frontend | C | C | **R**/A | C |
| Diseño UI/UX | C | I | **R**/A | C |
| Contenido Pedagógico | C | C | C | **R**/A |
| Pruebas de Sistema | I | C | C | **R**/A |
| Documentación Técnica | **R** | C | C | C |
| Despliegue | C | **R**/A | C | I |

*Leyenda: R = Responsable, A = Aprobador, C = Consultado, I = Informado*

## Gestión de Riesgos

Matriz de Riesgos

| ID | Riesgo | Probabilidad | Impacto | Severidad | Mitigación |
|---|---|---|---|---|---|
| R01 | Ejecución de comandos maliciosos | Media | Alto | **Crítico** | Implementar sandbox, lista blanca estricta |
| R02 | Baja adopción de usuarios | Media | Alto | **Alto** | Plan de marketing, alianzas institucionales |
| R03 | Problemas de escalabilidad | Baja | Medio | **Medio** | Arquitectura cloud elástica, cache distribuido |
| R04 | Retrasos en desarrollo | Media | Medio | **Medio** | Buffer de tiempo, desarrollo ágil |
| R05 | Falta de contenido pedagógico de calidad | Baja | Alto | **Alto** | Revisión por expertos, referencias confiables |
| R06 | Vulnerabilidades de seguridad | Media | Alto | **Crítico** | Auditorías de seguridad, OWASP compliance |
| R07 | Incompatibilidad entre navegadores | Baja | Bajo | **Bajo** | Pruebas cross-browser, polyfills |
| R08 | Sobrecostos del proyecto | Media | Medio | **Medio** | Control presupuestario semanal |

## Métricas de Éxito del Proyecto

### KPIs Técnicos

- Disponibilidad del sistema: >99.5%
- Tiempo de respuesta promedio: <2 segundos
- Cobertura de pruebas: >70%
- Bugs críticos en producción: 0
- Cumplimiento de sprints: >85%

### KPIs de Negocio

- Usuarios registrados en el primer mes: >500
- Tasa de retención mensual: >40%
- NPS (Net Promoter Score): >50
- Tasa de completación de cursos: >60%
- Ingresos por publicidad: Cubrir costos operativos

### KPIs Pedagógicos

- Mejora promedio pre-test/post-test: >20%
- Satisfacción del usuario (SUS): ≥68
- Tiempo promedio para completar ejercicio básico: <10 minutos
- Tasa de éxito en ejercicios: >80%
- Engagement diario (DAU/MAU): >25%

# Sostenibilidad

## Modelo de Negocio

La plataforma adoptará un modelo de sostenibilidad híbrido:

### Fase 1: Lanzamiento (Meses 1-6)

- **Versión básica gratuita** con acceso completo al contenido
- **Publicidad no intrusiva** (Google AdSense, banners) mostrada al finalizar lecciones
- **Meta de ingresos:** Cubrir costos de hosting ($500 USD/mes)

### Fase 2: Crecimiento (Meses 7-12)

- **Alianzas institucionales** con universidades y colegios técnicos

**Patrocinios corporativos** de empresas tecnológicas

**Certificados verificables** (opcional, costo: $20 USD)

**Meta de ingresos:** Cubrir costos operativos completos

**Fase 3: Escalamiento (Año 2+)**

- **Plan Premium** sin publicidad con features avanzados
- **Licencias institucionales** para organizaciones
- **Marketplace de contenido** creado por la comunidad
- **Meta de ingresos:** Rentabilidad del 20%

## Plan de Continuidad

1. **Respaldo de Datos**
   - Backups automáticos diarios (base de datos)
   - Snapshots semanales del sistema completo
   - Almacenamiento redundante en múltiples regiones
2. **Escalabilidad Técnica**
   - Arquitectura de microservicios para crecimiento modular
   - Auto-scaling en cloud para manejar picos de demanda
   - CDN para distribución global de contenido
3. **Actualización de Contenido**
   - Revisión trimestral del contenido pedagógico
   - Actualización anual basada en nuevas versiones de Linux
   - Incorporación de feedback de usuarios
4. **Comunidad y Soporte**
   - Foro de usuarios para soporte peer-to-peer
   - Base de conocimiento colaborativa
   - Sistema de tickets para soporte técnico

## Referencias

American Psychological Association. (2020). *Publication manual of the American Psychological Association* (7th ed.). https://doi.org/10.1037/0000165-000

ComputerHoy. (2022, 10 de junio). *Las mejores webs para aprender Linux de forma fácil y divertida*.

ComputerHoy. https://computerhoy.com/tutoriales/mejores-webs-aprender-linux-facil-divertida1073033

EducaciónIT. (2021, 30 de marzo). *10 razones por las que Linux es mejor para programadores y desarrolladores de software*. EducaciónIT Blog. https://blog.educacionit.com/2021/03/30/10-razoneslinux-mejor-para-programadores-desarrolladores-software/

HackerRank. (2025). *Solve Linux Shell HackerRank*. https://www.hackerrank.com/domains/shell

Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado. (2018). *Linux y educación:*
*Guía de software libre en educación*. INTEF. https://descargas.intef.es/recursos_educativos/it/guia_linux/guia_linux.pdf

International Organization for Standardization. (2015). *ISO/IEC/IEEE 15288:2015 Systems and software engineering — System life cycle processes*. ISO.

LabEx. (2025). *Linux Journey: Learn Linux with Free Linux Tutorial & Course*. https://labex.io/linuxjourney

Negus, C. (2021). *Linux bible* (10th ed.). Wiley.

OverTheWire. (2023). *Wargames*. https://overthewire.org/wargames/

VIM Adventures. (2025). *Learn VIM while playing a game - VIM Adventures*. https://vim-adventures.com

# Anexos

## Anexo A: Glosario de Términos

**API (Application Programming Interface):** Conjunto de definiciones y protocolos para desarrollar e integrar software de aplicaciones.

**CI/CD (Continuous Integration/Continuous Deployment):** Práctica de desarrollo que requiere que los desarrolladores integren código en un repositorio compartido varias veces al día.

**DAU/MAU (Daily Active Users/Monthly Active Users):** Métricas de engagement que miden usuarios activos diarios sobre usuarios activos mensuales.

**MoSCoW:** Técnica de priorización (Must have, Should have, Could have, Won't have).

**NPS (Net Promoter Score):** Métrica de lealtad del cliente que mide la disposición a recomendar.

**OWASP (Open Web Application Security Project):** Comunidad que produce artículos, metodologías y herramientas de seguridad web.

**RACI:** Matriz de asignación de responsabilidades (Responsible, Accountable, Consulted, Informed).

**Sandbox:** Entorno de prueba aislado donde se puede ejecutar código sin afectar el sistema principal.

**SLA (Service Level Agreement):** Acuerdo de nivel de servicio entre proveedor y cliente.

**SPA (Single Page Application):** Aplicación web que interactúa con el usuario reescribiendo dinámicamente la página actual.

**SRS (Software Requirements Specification):** Documento que describe completamente el comportamiento del software a desarrollar.

**SUS (System Usability Scale):** Escala de usabilidad del sistema, cuestionario estandarizado de 10 ítems.

**WBS (Work Breakdown Structure):** Descomposición jerárquica del trabajo total del proyecto.

## Anexo B: Plantilla de Caso de Uso

**ID:** UC-001
**Nombre:** Ejecutar comando básico en terminal
**Actor Principal:** Estudiante
**Precondiciones:** Usuario autenticado y lección iniciada
**Flujo Principal:**

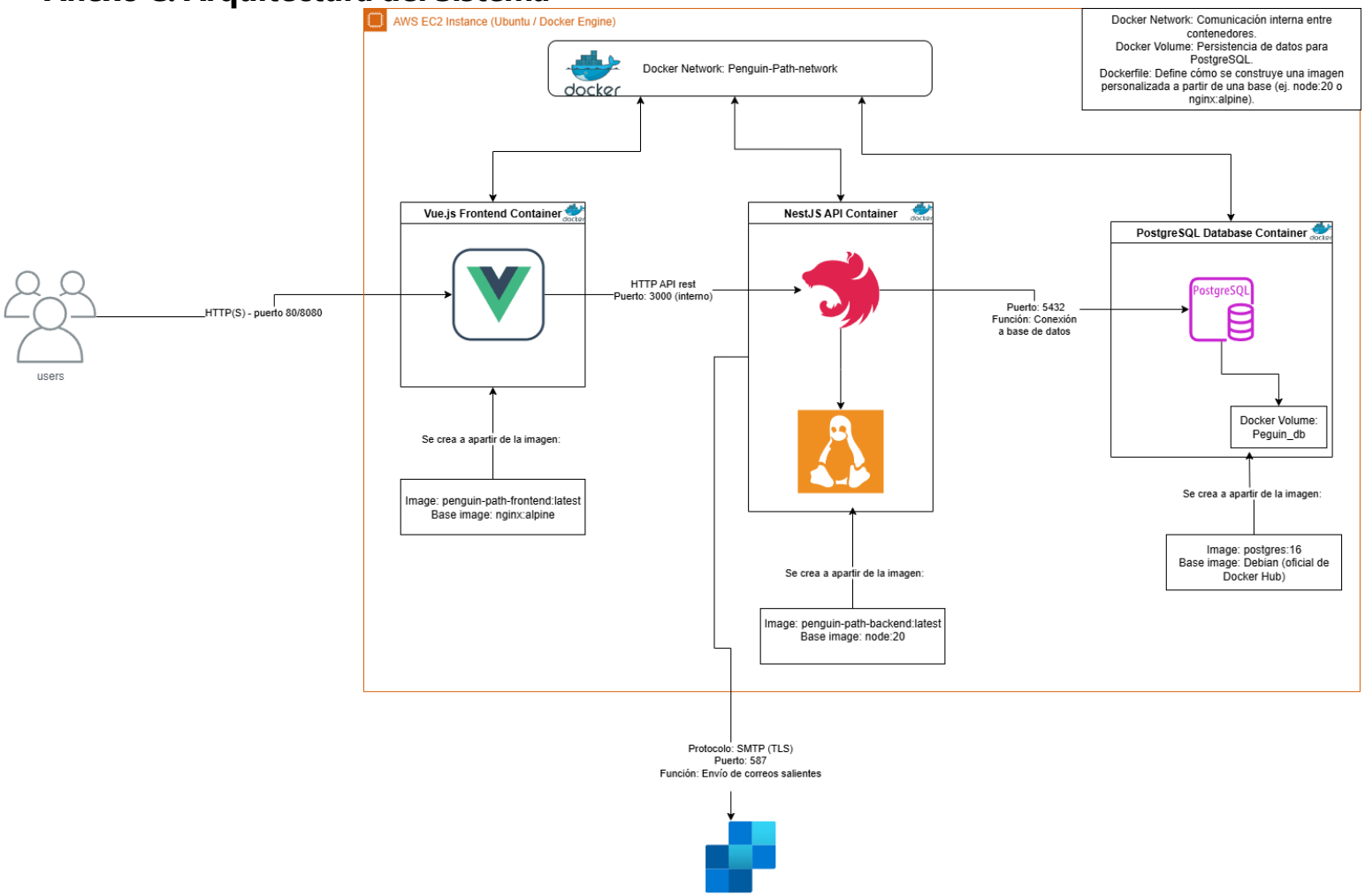
1. El estudiante ingresa un comando en la terminal simulada
2. El sistema valida el comando contra la lista blanca
3. El sistema ejecuta el comando en el sandbox
4. El sistema muestra el resultado con retroalimentación visual
5. El sistema registra el progreso del estudiante

**Flujos Alternativos:**

- 2a. Comando no permitido: mostrar mensaje de error educativo
- 3a. Timeout de ejecución: detener proceso y notificar

**Postcondiciones:** Comando ejecutado y progreso actualizado

# Anexo C: Arquitectura del Sistema



## Anexo D: Matriz de Trazabilidad de Requisitos (Ejemplo)

| Req ID | Descripción | Caso de Uso | Módulo | Prueba | Estado |
|---|---|---|---|---|---|
| RF-001 | Login de usuario | UC-001 | Auth | TC-001 | ✓ |
| RF-002 | Ejecutar comando ls | UC-002 | Terminal | TC-002 | ✓ |
| RF-003 | Guardar progreso | UC-003 | Progress | TC-003 | X |
| RNF-001 | Tiempo respuesta <2s | - | All | TP-001 | ✓ |
| RNF-002 | Disponibilidad 99.5% | - | Infra | TP-002 | X |

*Leyenda: ✓ Completado, ⧗ En progreso, ✗ Pendiente*

---

**Nota Final:** Este documento es un entregable vivo que se actualizará durante todo el ciclo de vida del proyecto, siguiendo los procesos establecidos en la norma ISO/IEC/IEEE 15288:2015.

## Anexo E: Preguntas

1. La plataforma combina teoría con laboratorios prácticos interactivos, máquinas virtuales, ¿simuladores?

Respuesta: Sí. En el Resumen y en el Alcance se dice que será una aplicación web interactiva, con un simulador de terminal en línea y ejercicios prácticos con retroalimentación gráfica. No obstante, como ya se ha mencionado no se usarán máquinas virtuales, sino una simulación de la terminal.

2. ¿Requiere instalar Linux en una máquina local o incluye entornos listos en la nube?

Respuesta: En el Planteamiento del problema se aclara que practicar en un Linux real puede ser riesgoso, por eso el sistema usará un entorno web seguro y simulado.

3. ¿Se podría internamente usar IA en la aplicación?

Respuesta:  No está pensado en el proyecto inicial, pero se podría escalar a usarla como un tutor que te de lecciones personalizadas

4. ¿Dan el ejemplo “racha” se debe usar la aplicación diariamente?

Respuesta: En el Diseño pedagógico (punto 6 – Acción) se incluye la gamificación:
La aplicación implementará un sistema de rachas como incentivo para la práctica diaria. Sin embargo, su uso no será obligatorio: los estudiantes podrán avanzar a su propio ritmo, pero mantener la racha servirá como motivador adicional para reforzar la constancia.

5. ¿Esa aplicación se podría usar con celular o netamente pc?

Respuesta: Sí. En los Objetivos específicos (punto 4) y el Alcance se indica que será responsive y accesible desde navegadores en móvil y desktop

6. ¿Qué herramientas usaran para reforzar la seguridad?

Respuesta: En los Objetivos específicos (punto 5) se menciona:

- Sandbox controlado
- Limitación de comandos
- Timeouts
- Pruebas de penetración básicas

Además, en el Presupuesto se incluye certificado SSL.

7. ¿Qué elementos usaran para el manejo de información por usuario?

Respuesta: En el Objetivo específico 3 (backend y lógica del simulador) se menciona la gestión de usuarios, sesiones y guardado de progresos.

8. ¿Cómo garantizan que los usuarios que usen la aplicación tendrán acceso al sistema operativo?

Respuesta: En el Planteamiento del problema y el Alcance: el acceso se hace a través del simulador de terminal en línea que replica comandos de Linux sin instalar el Sistema Operativo real.

9. ¿De qué manera enseñaran el uso de los lenguajes necesarios?

Respuesta: En el Diseño pedagógico (punto 6 – Acción) y en el Alcance.

Se enseñará principalmente el lenguaje de comandos de Linux (Bash) mediante un simulador de terminal en línea, acompañado de lecciones modulares, retos prácticos y retroalimentación automática. El enfoque no es en lenguajes de programación avanzados, sino en el uso de la terminal y comandos básicos.

10. ¿con qué herramienta mostrarían la interfaz del programa?

Respuesta: En el Objetivo específico (punto 2) se dice que usarán Vue + simulador propio para la simulación de la terminal y el frontend.

11. ¿Funciona en diferentes dispositivos (PC, tablet, móvil)?

Respuesta: Sí. En los Objetivos específicos (punto 4): SPA funcional responsive en móvil/desktop.

12. ¿Requiere instalar Linux en una máquina local o incluye entornos listos en la nube?

Respuesta: Ya respondido en la 2: es en la nube/simulación, no local.

13. ¿Qué tan actualizado se mantiene el contenido con respecto a cambios en distribuciones y herramientas?

Respuesta: En la Hipótesis y en el Diseño pedagógico (punto 6 – Acción), con el añadido de la Linux Bible como referencia.
El contenido se mantendrá actualizado mediante la revisión periódica de bibliografía confiable, principalmente la Linux Bible (Negus, 2021), junto con otras fuentes listadas en las Referencias. Esto permitirá incorporar cambios relevantes en distribuciones y herramientas de Linux.

14. ¿A qué se diferencian de los cursos online?

Respuesta: En el Planteamiento del problema se dice que los cursos existentes suelen ser teóricos, dispersos o de pago, mientras que esta aplicación será gratuita, interactiva y con retroalimentación inmediata.

15. ¿Como garantizaran el aprendizaje de los usuarios?

Respuesta: En Objetivo general y en Objetivo específico (punto 7) se menciona: pruebas con usuarios, métricas cuantificables (SUS, tasa de éxito, pre-test y post-test).
Eso garantiza aprendizaje con datos medibles.

16. ¿Qué tanto tiempo será necesario para tomar todo lo aprendido en Linux?

Respuesta: En la parte de objetivos específicos (punto 6) esta explicado. Se menciona que El curso básico estará diseñado para completarse en aproximadamente 20 horas distribuidas en 4 semanas, mientras que los módulos intermedios tendrán una duración estimada de 30 horas.

17. A la hora de realizar un proyecto por parte de la plataforma quien o que es la persona encargada de verificar que lo que escriba el estudiante este correcto, ya que puede haber varias formas de responder un problema.

Respuesta: En Métricas/aceptación del punto 6 de los objetivos se aclara que habrá múltiples soluciones correctas, evaluadas mediante patrones de comandos y casos de prueba automáticos.

18. ¿Cuál es el auto sustento que va a tener la plataforma?

Respuesta: En el apartado Sostenibilidad del proyecto:
La plataforma será gratuita y se sostendrá mediante anuncios integrados (ejemplo: Google Ads o banners no invasivos) que se mostrarán al finalizar cada lección. Estos anuncios generarán ingresos para cubrir costos de hosting, mantenimiento y actualizaciones.

19. ¿Quién va a ser el encargado de actualizar temas proponer nuevos temas etc....?

Respuesta: En el Diseñar y producir contenido pedagógico (punto 6 – Acción) se indica que el contenido obedecerá a fuentes actualizadas y confiables como la obra Linux Bible (Negus, 2021).

20. ¿Porque debería elegirlos y no elegir algo más técnico como un libro que explica de manera más directa y profesional?

Respuesta: En el Planteamiento del problema se dice: los libros/manuales suelen ser teóricos, dispersos y poco interactivos, mientras que aquí hay práctica con retroalimentación inmediata.

21. ¿Cuál es la diferencia de aprendizaje entre un profesor y su proyecto?

Respuesta: En la Hipótesis y en el Planteamiento del problema: la aplicación busca mayor motivación y accesibilidad que las clases tradicionales, además es gratuita y práctica.

22. ¿Cuál es el experto en Linux que guiará el proceso?

Respuesta: Ya no se requiere un experto externo. Según el punto 6 – Acción, el proceso se guía con base en fuentes bibliográficas confiables, especialmente la Linux Bible, lo cual garantiza respaldo técnico y actualización sin depender de una sola persona.

23. ¿Los sistemas de aprendizajes son personalizados o general, Si un estudiante aprende diferente manera?

Respuesta: En la Hipótesis se establece que el sistema sí tendrá personalización parcial:

La dificultad y los retos se adaptarán al progreso del estudiante.

Se usarán métricas de desempeño para ofrecer ejercicios adicionales en caso de dificultad.

24. cuál es la importancia de enseñar de esta manera y no con otras plataformas ya existentes

Respuesta: En el Planteamiento del problema se dice que los cursos existentes suelen ser teóricos, dispersos o de pago, mientras que esta app será gratuita, interactiva y con retroalimentación inmediata.

25. ¿Quién está capacitado o tiene los conocimientos para hacer las prácticas de Linux?

Respuesta: Ya no se requiere una persona capacitada. Según el punto 6 – Acción, el proceso se guía con base en fuentes bibliográficas confiables, especialmente la Linux Bible, lo cual garantiza respaldo técnico y actualización sin depender de una sola persona.

26. ¿La plataforma tendrá su propia máquina de uso o se debe instalar Linux en un computador?

Respuesta: En el Alcance: simulador de terminal en línea, no requiere instalar Linux localmente.

27. ¿A quién está dirigido específicamente este sistema?

Respuesta: En el Planteamiento del problema: a estudiantes y profesionales en áreas de sistemas, programación, ciberseguridad, servidores.

28. ¿Cómo planeas garantizar la seguridad del sistema durante su implementación, especialmente considerando que está destinado a enseñar Linux, un sistema que involucra trabajar con operaciones que pueden ser arriesgadas?

Respuesta: En Objetivo específico (punto 5): sandbox, listas blancas/negras de comandos, timeouts, pruebas de seguridad.

29. ¿Cómo planean simular el entorno de Linux que sea suficiente realista para los usuarios novatos?

Respuesta: En los Objetivos específicos (puntos 3 y 5) se define construir un simulador propio de terminal. Así, los usuarios novatos tendrán una experiencia realista, segura y con retroalimentación pedagógica.

30. ¿Cómo la plataforma web va a generar ingresos?

Respuesta: En el apartado Sostenibilidad del proyecto, se menciona que la plataforma será gratuita y se sostendrá mediante anuncios integrados que se mostrarán al finalizar cada lección.

31. ¿Qué mejora genera el uso de simulación de terminal?

Respuesta: Permite practicar comandos y flujos reales en un entorno seguro (sin riesgo de dañar el sistema real), ofrece retroalimentación inmediata y trazable (salidas y logs), facilita ejercicios evaluables automáticamente y reduce la curva de aprendizaje al combinar explicación contextual con práctica.
En: Objetivos específicos / Diseño pedagógico — sección de actividades prácticas y simulador.

32. ¿Qué son requisitos SRS?

Respuesta: El SRS (Software Requirements Specification) es el documento formal que especifica los requisitos funcionales y no funcionales, interfaces, restricciones, criterios de aceptación y prioridades (MUST/SHOULD/CAN) para el sistema. Sirve como base para diseño, pruebas y validación.
En: Objetivo específico 1 — Entregable: Documento SRS / Levantamiento de requisitos.

33. ¿Qué es certificado SSL?

Respuesta: Un certificado SSL/TLS habilita HTTPS en el dominio, cifrando la comunicación cliente-servidor para garantizar confidencialidad, integridad y autenticidad del sitio; es requisito de seguridad para el despliegue en producción.
En: Despliegue y condiciones de operación — sección de infraestructura y seguridad.

34. ¿El acceso a la plataforma educativa tendrá algún costo o cómo se financiará?

Respuesta: El plan inicial contempla una versión básica gratuita. La sostenibilidad propuesta incluye ingresos por anuncios no invasivos, alianzas o patrocinios institucionales y, como extensión futura, planes premium o servicios de valor agregado.
En: Apartado Sostenibilidad / Modelo de negocio.

35. ¿Cualquier persona puede participar?

Respuesta: Sí; la plataforma está diseñada para acceso abierto (navegador) orientado a estudiantes y profesionales novatos, sin requisitos de instalación ni exclusiones por perfil académico.
En: Planteamiento del problema / Alcance del proyecto.

36. ¿Van a tener un centro de ayuda si no se algo sobre algún comando? O sea, explicándole cada comando

Respuesta: Sí; se incluye documentación integrada: fichas por comando (descripción, flags comunes, ejemplos), tutoriales paso a paso, ayuda contextual dentro del simulador, banco de ejercicios y FAQ.
En: Diseño pedagógico — Materiales y recursos didácticos / Objetivos de contenido.

37. ¿Como van a llevar a que utilicen su aplicación y no la de otras existentes?

Respuesta: Estrategia de diferenciación: simulador visual y seguro con retroalimentación inmediata; diseño pedagógico alineado al currículo; gamificación (puntos, rachas, badges) para retención; accesibilidad (versión gratuita); y alianzas con instituciones para adopción.
En: Análisis de competencia y propuesta de valor / Objetivos de difusión y retención.

38. ¿Saldrá uno certificado o solamente es para aprender?

Respuesta: En la fase inicial no se contempla certificación formal; la plataforma ofrecerá badges y reportes de progreso internos. La emisión de certificados oficiales se considera como extensión futura sujeta a convenios.
En: Diseño pedagógico — Sección de acreditación y extensiones (notas sobre certificación).

39. ¿Van a utilizar algún método de puntuación al hacer los ejercicios dados?
    ¿Darán certificado?

Respuesta: Sí; se implementará un sistema de puntuación basado en aciertos, tiempo y eficiencia, más rachas y badges, para feedback y ajuste adaptativo de la dificultad.
En: Diseño pedagógico / Gamificación y evaluación formativa.

40. ¿Qué mecanismos de evaluación usarán para medir el progreso del aprendizaje?

Respuesta: No en el alcance inicial; se entregarán reconocimientos internos (badges, reportes). La certificación oficial queda como posible fase posterior tras validar el piloto.
En: Sección de acreditación / Notas sobre alcance y extensiones.

41. ¿Planean incluir certificaciones oficiales como valor agregado del proyecto?

Respuesta: Pre-test y post-test por módulo, ejercicios automáticos con casos de prueba y salidas esperadas, métricas de uso (tiempo, tasa de éxito), cuestionarios de percepción/usabilidad y pruebas de usuario para medir ganancia de aprendizaje.
En: Objetivo específico — Validación pedagógica y pruebas de usabilidad / Métricas y evaluación.

42. ¿Como garantizan la efectividad de los métodos de calificación?

Respuesta: Es una posibilidad para fases posteriores: se estudiarán convenios con instituciones o certificadores tras la fase piloto y evaluación de efectividad. No está incluido en la versión inicial.
En: Planeación de extensiones / Roadmap y sostenibilidad.

43.  está enfocado a alguna institución o en un curso de estudio autónomo?

Respuesta: Mediante validación empírica: pruebas automatizadas (casos de prueba), análisis pre/post para medir ganancia de conocimientos, métricas de desempeño del piloto y ajustes iterativos basados en datos y feedback de usuarios.
En: Objetivos de evaluación / Metodología de validación y análisis de resultados.

44.  ¿Integra foros para retroalimentaciones?

Respuesta: Diseñado para ambos: uso autónomo por estudiantes y profesionales, y compatibilidad para integración institucional (panel docente, estadísticas agregadas y recursos para docentes).

En: Alcance y usuarios objetivo / Planteamiento del problema y objetivos de integración institucional.

45. ¿Su plataforma funcionaría solo en computador o también en celular?

Respuesta: En los Objetivos específicos (punto 4): SPA funcional responsive en móvil/desktop.

46. ¿Cómo harán para mantener los temas siempre actualizados?

Respuesta: En la Hipótesis y en el Diseño pedagógico (punto 6 – Acción):
Los contenidos estarán basados en fuentes confiables como *Linux Bible* y revisados periódicamente por el equipo pedagógico.

47. ¿Por qué creen que su plataforma es mejor que un curso online que ya existe?

Respuesta: En el Planteamiento del problema:

Los cursos online existentes suelen ser teóricos, dispersos o de pago.

La plataforma es gratuita, interactiva, con simulador y retroalimentación inmediata.

48. ¿Qué mecanismos de evaluación usarán para medir si el estudiante realmente aprendió?

Respuesta: En el Objetivo específico 7 – Pruebas de usabilidad y validación pedagógica:

Pruebas pre-test y post-test.

Métricas de desempeño (tiempo, tasa de éxito, SUS ≥ 68).

49. ¿Cómo van a mantener actualizados los contenidos frente a nuevas versiones de Linux y software relacionado?

Respuesta: En el Diseño pedagógico (punto 6 – Acción):
Revisión periódica de la Linux Bible y otras fuentes de referencia en la sección Referencias.

50. ¿Van a certificar a los usuarios que completen el programa?

Respuesta: No habrá certificación formal, el enfoque está en la práctica interactiva y la motivación con rachas y puntuaciones.

51. ¿Van a utilizar algún método de puntuación al hacer los ejercicios dados?

Respuesta: En el Diseño pedagógico (punto 6 – Acción) con la gamificación añadida:
Sí, se incluirán puntuaciones y rachas como motivación, además de retroalimentación automática.

52. ¿Qué tan actualizado se mantiene el contenido con respecto a cambios en distribuciones y herramientas?

Respuesta: Ya respondido antes: Diseño pedagógico (punto 6 – Acción) y Referencias.

53. ¿Analizaron las competencias?

Respuesta: Sí, en el objetivo 1 analizamos plataformas como Linux Journey y OverTheWire, y a partir de esa competencia definimos cómo diferenciarnos con interactividad y retroalimentación inmediata.

54. ¿Cómo van a simular un terminal para probar?

Respuesta: En el objetivo 2 se menciona que la terminal se probará con un simulador propio que interpreta comandos en un sistema de archivos virtual controlado, evitando riesgos en el sistema real.

55. ¿Con que lenguaje harán el proyecto?

Respuesta: En el Objetivo específico 2 – Diseñar la arquitectura técnica:
Stack tecnológico: Vue.js (frontend), Nest.js(backend), PostgreSQL(Base de datos).

56. ¿Cuál es la diferencia entre la plataforma de LINUX SURVIVAL, LINUX JOURNEY?

Respuesta: Sí, en el Objetivo específico 1 – Analizar el estado del arte y definir requisitos se compararon plataformas como Linux Survival y Linux Journey; ambas son útiles pero teóricas o poco interactivas. Nuestra propuesta se diferencia al ofrecer práctica en un simulador con retroalimentación inmediata y gamificación.

57. ¿Como será el aprendizaje aplicado a nivel de colegio o de universidad?

Respuesta: En el Planteamiento del problema y Alcance:
Dirigido a estudiantes y profesionales novatos. Se puede usar en colegios/universidades como recurso didáctico, porque no requiere instalación de Linux real y es accesible desde navegador.