**Ingeniería de Software**

**PROYECTO WEB SCRAPER**

**EVALUACIÓN DE CALIDAD**

Mas información para añadir aquí.

**Contenido**

[Resumen Ejecutivo 2](#_Toc201945095)

[Ejercicio 1: Evaluación de la Calidad del Proceso y Producto 2](#_Toc201945096)

[1.1 Calidad del Proceso 2](#_Toc201945097)

[1.2 Calidad del Producto 3](#_Toc201945098)

[Ejercicio 2: Auditoría ISO/IEC 25010 4](#_Toc201945099)

[Lista de Verificación 4](#_Toc201945100)

[Ejercicio 3: Evaluación de Madurez (CMMI/TSP) 4](#_Toc201945101)

[Autoevaluación CMMI 4](#_Toc201945102)

[Nivel Actual: 2 (Gestionado) 4](#_Toc201945103)

[Plan de Mejora 5](#_Toc201945104)

## 

## Resumen Ejecutivo

Este documento presenta una evaluación integral del proceso de desarrollo y la calidad del producto final (backend y app móvil), aplicando estándares internacionales (ISO/IEC 25010) y modelos de madurez (CMMI, TSP). Incluye:

* Análisis del ciclo de vida del proyecto.
* Auditoría de calidad basada en ISO 25010.
* Autoevaluación de madurez según CMMI y plan de mejora.

## Ejercicio 1: Evaluación de la Calidad del Proceso y Producto

### 1.1 Calidad del Proceso

#### Ciclo de Vida Implementado:

1. Diseño:
   * *Actividades*: Diagramas UML, definición de arquitectura (MVP).
   * *Roles*: Arquitecto de software, diseñador UX.
   * *Artefactos*: Wireframes, documentación de API.
2. Implementación:
   * *Actividades*: Desarrollo en sprints (2 semanas), revisiones de código.
   * *Roles*: Desarrolladores backend/frontend.
   * *Artefactos*: Código fuente en GitHub, commits con convención semántica.
3. Pruebas:
   * *Actividades*: Unitarias (pytest), rendimiento (JMeter), seguridad (OWASP ZAP).
   * *Roles*: QA Engineer.
   * *Artefactos*: Reportes de cobertura (85%), logs de pruebas.
4. Despliegue:
   * *Actividades*: CI/CD con GitHub Actions, despliegue en AWS.
   * *Roles*: DevOps.
   * *Artefactos*: Dockerfiles, pipeline YAML.

#### Efectividad del QA en GitHub:

* *Aciertos*:
  + Integración continua con pruebas automáticas.
  + Revisión por pares obligatoria (2 aprobaciones).
* *Mejoras*:
  + Implementar gates de seguridad en PRs (ej: bloqueo si ZAP detecta vulnerabilidades).

### 1.2 Calidad del Producto

| Característica | Evaluación | Evidencia |
| --- | --- | --- |
| Funcionalidad | Cumple (90%) | Todos los EPIC del backlog implementados. |
| Fiabilidad | Cumple parcialmente | Pruebas de estrés mostraron caídas con >300 usuarios. |
| Usabilidad | Cumple | Feedback positivo en tests de usuario (85% satisfacción). |
| Eficiencia | Cumple | Tiempo de respuesta promedio: 1.2s (pruebas con 50 usuarios). |
| Seguridad | Cumple parcialmente | pytest detectó 2 XSS (corregidos en PR #45). |

## Ejercicio 2: Auditoría ISO/IEC 25010

### Lista de Verificación

1. Funcionalidad

* [✔] Cumple: Todos los requisitos funcionales están implementados.
* [✔] Cumple: Los mensajes de error son claros (ej: "URL inválida").

2. Usabilidad

* [✔] Cumple: Interfaz móvil sigue guidelines de Material Design.
* [✘] No cumple: Falta tutorial onboarding.

3. Eficiencia

* [✔] Cumple: Uso de caché reduce consultas a la API.
* [✔] Cumple parcialmente: CPU alcanza 80% bajo carga máxima.

4. Seguridad

* [✔] Cumple: HTTPS y validación de inputs.
* [✘] No cumple: Falta rate limiting para prevenir DDoS.

## Ejercicio 3: Evaluación de Madurez (CMMI/TSP)

### Autoevaluación CMMI

### Nivel Actual: 2 (Gestionado)

* *Justificación*:
  + Planificación documentada (sprints).
  + Control de versiones (Git).
  + Pruebas automatizadas.
* *Faltas para Nivel 3*:
  + No hay métricas formales de calidad.

### Plan de Mejora

1. Adoptar TSP:
   * Autogestión de tareas con daily meetings.
2. Métricas CMMI Nivel 3:
   * Trackear defectos por cada 1k líneas de código.
3. Procesos MoProSoft:
   * Certificar a un miembro en el estándar.

## Conclusiones

* La combinación de estándares (ISO 25010) y modelos (CMMI) mejora la trazabilidad y calidad.
* Las pruebas automatizadas son críticas para fiabilidad.
* **Lección clave**: Implementar gates de seguridad desde el diseño.

## Bibliografía (APA)

* ISO/IEC. (2011). *25010: Systems and software quality requirements and evaluation*.
* CMMI Institute. (2018). *CMMI for Development Version 2.0*.