



**UNIVERSIDAD PRIVADA DE TACNA**

**FACULTAD DE INGENIERIA**

**Escuela Profesional de Ingeniería de Sistemas**

**Proyecto *DocuCode-AI***

Curso: *PATRONES DE SOFTWARE*

Docente: *PATRICK JOSE CUADROS QUIROGA*

Integrantes:

***Farley Rodrigo Eduardo Viveros Blanco - 2020066896***

***Ronal Daniel Lupaca Mamani - 202006146***

**Tacna – Perú  
2025**

# **Sistema *DocuCode-AI*** **Informe de Factibilidad**

**Versión 2.0**

CONTROL DE VERSIONES					
Versión	Hecha por	Revisada por	Aprobada por	Fecha	Motivo
1.0	MPV	ELV	ARV	3/04/2025	Versión Original
2.0	MPV	ELV	ARV	30/05/2020	mejora

## INDICE GENERAL

1.	Descripción del Proyecto.....	4
2.	Riesgos.....	5
3.	Análisis de la Situación actual .....	5
4.	Estudio de Factibilidad .....	7
4.1	Factibilidad Técnica .....	7
4.2	Factibilidad económica.....	8
4.3	Factibilidad Operativa .....	9
4.4	Factibilidad Legal.....	10
4.5	Factibilidad Social.....	10
4.6	Factibilidad Ambiental .....	11
5.	Análisis Financiero.....	11
6.	Conclusiones.....	13

## **Informe de Factibilidad**

### **1. Descripción del Proyecto**

#### **1.1 Nombre del proyecto**

DocuCode-AI: Sistema Web para la Evaluación de Código y Generación de Diagramas UML

#### **1.2 Duración del proyecto**

El proyecto se desarrollará en un plazo estimado de 4 meses, incluyendo etapas de análisis, desarrollo, pruebas y despliegue.

#### **1.3 Descripción**

DocuCode-AI es un sistema web basado en Inteligencia Artificial que tiene como propósito asistir a docentes universitarios en la evaluación de prácticas de programación. El sistema permite analizar automáticamente archivos de código fuente mediante la API de OpenAI, generando comentarios explicativos, identificando duplicidad y malas prácticas, evaluando métricas de calidad, y construyendo diagramas UML como Clases, Casos de Uso, Secuencia, Actividad, Paquetes y Componentes.

La solución responde a una problemática común en los entornos académicos: el alto volumen de prácticas de programación que deben ser revisadas manualmente, lo que implica un proceso lento, subjetivo y repetitivo. Al integrar capacidades automatizadas y visuales, DocuCode-AI busca reducir significativamente el tiempo de evaluación y mejorar la retroalimentación entregada a los estudiantes.

La aplicación se desarrolla en PHP utilizando el patrón de diseño Fachada y herramientas como PlantUML para diagramas y Terraform para definir y simular la infraestructura de despliegue en la nube.

#### **1.4 Objetivos**

##### **1.4.1 Objetivo general**

Desarrollar un sistema web inteligente basado en PHP y tecnologías de Inteligencia Artificial que permita automatizar la evaluación, documentación y visualización del código fuente mediante comentarios,

métricas de calidad y diagramas UML, dirigido principalmente al ámbito educativo universitario.

#### 1.4.2 Objetivos Específicos

- Implementar un motor de IA para la generación de comentarios en el código.
- Incorporar herramientas para la creación automática de diagramas UML.
- Evaluar la calidad del código, identificando errores, código duplicado y malas prácticas.
- Integrar el sistema con plataformas educativas para facilitar su adopción.

## 2. Riesgos

- Dependencia de servicios externos (API de OpenAI):  
El sistema depende de la disponibilidad y costos de la API de OpenAI para el análisis del código. Un cambio en los precios, límites de uso o en la política de acceso podría afectar la sostenibilidad del proyecto.
- Seguridad de la información:  
El sistema manejará archivos de código fuente que pueden contener información sensible o privada. Es necesario garantizar la confidencialidad y protección de los datos mediante cifrado, autenticación segura y control de accesos.
- Compatibilidad con múltiples lenguajes de programación:  
Aunque el sistema está diseñado para analizar código en distintos lenguajes, podrían surgir dificultades con sintaxis particulares o estructuras inusuales que afecten la precisión del análisis automático.
- Conectividad a internet:  
Debido a que el análisis y los diagramas se procesan en la nube, una conexión lenta o inestable puede afectar el rendimiento y la experiencia del usuario.
- Curva de aprendizaje de nuevas herramientas (Terraform, PlantUML):  
El equipo de desarrollo debe adaptarse al uso de herramientas como Terraform y PlantUML, lo cual podría generar demoras en la primera fase del proyecto.
- Escalabilidad y demanda futura:  
Si la plataforma tiene una alta demanda por parte de usuarios concurrentes, será necesario escalar adecuadamente la infraestructura, lo cual puede representar un aumento en los costos.

## 3. Análisis de la Situación actual

### 3.1 Planteamiento del problema

En las instituciones educativas, especialmente en carreras de ingeniería y programación, los docentes enfrentan serias dificultades al momento de revisar prácticas y proyectos de código fuente. Este proceso suele ser manual, consume mucho tiempo, es subjetivo y propenso a errores. La falta de documentación adecuada en los archivos entregados por los estudiantes complica la comprensión del código, lo que puede derivar en evaluaciones inconsistentes o poco claras.

Además, debido al volumen de trabajos que se deben revisar y a la diversidad de lenguajes y estructuras de programación utilizados, se vuelve inviable realizar una evaluación profunda y personalizada sin herramientas de automatización.

En este contexto, surge la necesidad de implementar un sistema web inteligente que permita automatizar las tareas de análisis, documentación y retroalimentación de código fuente, facilitando así la labor docente y mejorando la calidad de la enseñanza.

### 3.2 Consideraciones de hardware y software

#### ***Hardware disponible***

- ***Servidor VPS en la nube (Elastika):***
  - *2 vCPUs*
  - *2 GB de RAM*
  - *40 GB SSD*
  - *1 IP pública*
  - *Tráfico ilimitado*
  - *Soporte autogestionado*
  - *Costo aproximado: S/ 20.00 mensuales / S/ 100.00 anuales*
- ***Equipos cliente (docentes y usuarios):***
  - *Computadoras o laptops con navegador actualizado (Chrome, Firefox, Edge).*
  - *Conexión estable a internet.*

#### ***Software utilizado***

- ***Backend:*** *PHP con patrón de diseño Fachada, complementado con módulos en Python para análisis de código.*
- ***Frontend:*** *HTML, CSS, JavaScript con interfaz responsiva.*

- **Base de datos:** MySQL.
- **APIs externas:** OpenAI API (para generación de comentarios en código)
- **Librerías UML:** PlantUML para generación automática de diagramas de clases, casos de uso, secuencia, actividad, componentes y paquetes.
- **Infraestructura como código:** Terraform, para definición y simulación de la infraestructura del sistema.
- **Herramientas de desarrollo:** Visual Studio Code, Postman, GitHub, GitHub Actions (CI/CD).

#### 4. Estudio de Factibilidad

*El presente estudio tiene como propósito determinar la viabilidad integral del desarrollo del sistema DocuCode-AI, evaluando aspectos técnicos, económicos, operativos, legales, sociales y ambientales. Esta evaluación permite identificar riesgos, recursos requeridos y beneficios esperados, justificando así la implementación del sistema.*

*Para ello, se llevaron a cabo las siguientes actividades:*

- Revisión de recursos tecnológicos disponibles y compatibles.
- Estimación de costos y análisis financiero preliminar.
- Análisis de normativas legales y requerimientos institucionales.
- Identificación del impacto social y ambiental.
- Simulación de infraestructura con Terraform para estimación de costos operativos.

##### 4.1 Factibilidad Técnica

*Este apartado analiza los recursos tecnológicos disponibles y su aplicabilidad a DocuCode-AI, asegurando que el sistema pueda ser desarrollado y mantenido sin limitaciones técnicas.*

*Evaluación de Hardware y Servidores*

- Servidor en la nube: Se usará Elastika, con capacidad escalable según demanda.
- Especificaciones del servidor: 2 vCPUs, 4GB RAM, 50GB SSD.
- Requerimientos de conectividad: Acceso estable a internet con latencia mínima.

*Evaluación de Software*

- Lenguajes de programación:
  - Backend: PHP y Python.
  - Frontend: HTML, CSS, JavaScript.
- Base de datos: MySQL/PostgreSQL.
- APIs y Librerías:
  - OpenAI API: Generación de comentarios en código.
  - PlantUML: Creación de diagramas UML.

- Pylint y AST: Análisis de código en Python.
- Conclusión: DocuCode-AI es técnicamente viable con los recursos y tecnologías disponibles.

#### 4.2 Factibilidad Económica

*El análisis económico busca evaluar si los costos del proyecto son sostenibles en relación con los beneficios que ofrece.*

Definir los siguientes costos:

##### 4.2.1 Costos Generales

Concepto	Costo Estimado
Servidor en la nube (Elastika VPS)	S/ 20.00 mensuales / S/ 100.00 anuales
Dominio web	S/ 12.00 anuales
Certificado SSL (Let's Encrypt)	S/ 0.00
OpenAI API (GPT-4)	~S/ 20.00 mensuales (estimado por uso)
<b>Total estimado anual</b>	<b>S/ 150.00 – S/ 250.00</b>

##### 4.2.2 Costos operativos durante el desarrollo

No se requieren oficinas físicas, ya que el equipo trabajará de forma remota.

Uso de herramientas gratuitas para desarrollo: **VS Code, GitHub, Postman.**

##### 4.2.3 Costos del ambiente

- Servidor en la nube con Docker y balanceador de carga.
- Conexión estable a internet.

##### 4.2.4 Costos de personal

Rol	Horas estimadas	Costo estimado
Estudiante – Backend	300 horas	S/ 0.00 (Desarrollo académico)
Estudiante – Frontend	250 horas	S/ 0.00 (Desarrollo académico)
Estudiante – Integración IA	200 horas	S/ 0.00 (Desarrollo académico)

##### 4.2.5 Costos totales del desarrollo del sistema

*El costo total del proyecto durante su etapa de desarrollo y primer año de operación se estima entre S/ 150.00 y S/ 250.00, cubriendo infraestructura, dominio y consumo promedio de la API de OpenAI. Este monto es asumible dentro de un entorno académico.*



#### 4.2.6 Análisis Económico utilizando Terraform

Como parte del análisis económico del proyecto DocuCode-AI, se desarrolló un archivo de infraestructura como código (`main.tf`) utilizando Terraform con el objetivo de automatizar el cálculo de los costos anuales del sistema.

Este enfoque permite:

- Simular y visualizar automáticamente los costos mensuales y anuales del sistema.
- Mantener la trazabilidad de la infraestructura y el gasto asociado.
- Promover el uso de herramientas DevOps dentro del desarrollo académico.

El archivo `main.tf` considera:

- Costo mensual del VPS (Elastika).
- Consumo estimado mensual de la API de OpenAI.
- Registro del dominio y certificado SSL.

> Este archivo se encuentra disponible en el repositorio del proyecto, dentro de la carpeta `/infraestructura/`.

Ejemplo de salida al ejecutar el archivo:

Costo mensual total (S/.) = 40

Costo anual total (S/.) = 492

#### 4.3 Factibilidad Operativa

*DocuCode-AI representa una solución tecnológica viable desde el punto de vista operativo. El sistema ofrece múltiples beneficios que mejoran significativamente el proceso de evaluación de código en instituciones educativas, reduciendo la carga docente y facilitando la comprensión del software entregado por los estudiantes.*

##### **Beneficios del producto**

- *Evaluación automática del código mediante IA, ahorrando tiempo de revisión.*
- *Generación de diagramas UML de manera visual e inmediata.*
- *Interfaz web accesible, multiplataforma y de fácil uso.*
- *Reducción de errores humanos al evaluar grandes volúmenes de prácticas.*
- *Historial de análisis por usuario autenticado, facilitando el seguimiento.*

##### **Capacidad del cliente (docente/universidad)**

- *Los docentes poseen el conocimiento necesario para interpretar los resultados del sistema.*
- *El sistema no requiere conocimientos técnicos para su uso, más allá de un manejo básico de plataformas web.*
- *La universidad puede mantener el sistema funcionando mediante un VPS accesible y autogestionado, con mínimo soporte técnico.*

#### **Impacto en los usuarios**

- *Aumenta la eficiencia y precisión en la evaluación del código fuente.*
- *Mejora la retroalimentación a los estudiantes.*
- *Promueve la calidad y buenas prácticas de programación en los entornos de enseñanza.*

#### **Lista de interesados**

- *Docentes: Usuarios directos del sistema.*
- *Estudiantes: Beneficiarios de los análisis y retroalimentación.*
- *Administradores de sistemas: Encargados del soporte técnico.*
- *Universidad: Institución que promueve la innovación educativa.*

*Conclusión: El sistema es operativamente viable, accesible y fácil de implementar.*

#### **4.4 Factibilidad Legal**

- *En la evaluación legal del proyecto **DocuCode-AI**, no se han identificado conflictos con normativas locales ni internacionales. El sistema cumple con los principios básicos de protección de datos y seguridad digital requeridos para aplicaciones web modernas.*

##### *Consideraciones legales*

- **Protección de datos personales:** *Se implementará cifrado SSL para garantizar la confidencialidad del código subido por los usuarios.*
- **Conducta de negocio y licenciamiento:** *El sistema será liberado bajo una licencia de software libre para fines académicos, con posibilidad de uso comercial futuro bajo términos claros.*
- **Cumplimiento normativo:** *El sistema puede adaptarse a marcos legales como el GDPR en caso de implementarse en otros países.*

*Conclusión: DocuCode-AI es legalmente viable. No infringe ninguna norma local o internacional aplicable.*

#### **4.5 Factibilidad Social**

*El proyecto DocuCode-AI tiene un impacto social positivo, al integrarse en el ámbito académico y promover la innovación en la enseñanza de la programación.*

#### *Evaluación social y cultural*

- *Códigos de ética y conducta: El sistema respeta los principios de equidad educativa, al ofrecer un entorno justo de evaluación para todos los estudiantes.*
- *Clima político y educativo: Se alinea con los objetivos de mejora de la calidad educativa promovidos por las instituciones públicas y privadas.*
- *Inclusión: Permite a docentes con poca experiencia técnica utilizar tecnologías de vanguardia sin dificultad.*

*Conclusión: No existen barreras sociales que limiten la implementación del sistema. Su impacto es favorable y fomenta la transformación digital en la educación.*

#### 4.6 Factibilidad Ambiental

*La solución propuesta tiene un bajo impacto ambiental, dado que se desarrolla y opera íntegramente en la nube y evita el uso de recursos físicos.*

##### *Consideraciones ambientales*

- *Menor uso de papel: El sistema genera documentación automática, eliminando la necesidad de imprimir prácticas o reportes.*
- *Infraestructura digital: No requiere equipos físicos locales, ya que se ejecuta en servidores remotos optimizados.*
- *Bajo consumo energético: Al operar en un VPS compartido, el impacto energético se reduce frente a infraestructuras tradicionales.*

*Conclusión: DocuCode-AI es ambientalmente sostenible y se alinea con prácticas responsables de desarrollo digital.*

#### 5. Análisis Financiero

*El plan financiero de DocuCode-AI se basa en la estimación de beneficios académicos y operativos frente a los gastos asociados al desarrollo, despliegue y mantenimiento del sistema. Aunque no tiene fines comerciales en esta etapa, este análisis permite justificar la inversión de tiempo, esfuerzo y recursos, así como anticipar situaciones financieramente inadecuadas.*

*Se han proyectado beneficios durante cinco años, considerando principalmente el ahorro de tiempo docente en la evaluación manual de prácticas de programación, contrastando con el bajo costo de mantener la infraestructura tecnológica del sistema.*

##### 5.1 Justificación de la Inversión

*La inversión está plenamente justificada por el impacto positivo que genera el sistema en la dinámica educativa. Automatizar la revisión de código y la generación de documentación técnica mejora la productividad docente,*

*garantiza una evaluación más objetiva y proporciona retroalimentación inmediata al estudiante.*

#### 5.1.1 Beneficios del Proyecto

##### **Beneficios Tangibles:**

- *Reducción del tiempo invertido por los docentes en revisión de código fuente (de horas a minutos).*
- *Disminución de errores humanos y subjetividad en las evaluaciones.*
- *Eliminación del uso de papel y reducción de recursos físicos.*
- *Ahorro en contratación de asistentes académicos o evaluadores externos.*

##### **Beneficios Intangibles:**

- *Mejora significativa en la eficiencia del área académica de programación.*
- *Aumento en la calidad y confiabilidad de las evaluaciones.*
- *Disponibilidad inmediata de reportes e historial por estudiante.*
- *Valor agregado a la experiencia formativa del estudiante.*
- *Impulso a la transformación digital de la enseñanza universitaria.*
- *Alineamiento con tendencias de automatización y educación inteligente.*

#### 5.1.2 Criterios de Inversión

*A pesar de que el proyecto DocuCode-AI no tiene fines comerciales, es posible evaluar su viabilidad económica mediante indicadores adaptados al contexto académico, utilizando como beneficio el ahorro operativo, el tiempo docente y el valor institucional generado por la automatización de tareas clave en el proceso de enseñanza de la programación.*

##### 5.1.2.1 Relación Beneficio/Costo (B/C)

La relación B/C compara el valor presente de los beneficios proyectados frente a los costos estimados.

##### **Resultado obtenido:**

**B/C = 16.24**

Esto indica que por cada sol invertido en la operación del sistema, se generan aproximadamente **S/ 16.24 en beneficios académicos**, tales como ahorro de tiempo docente, reducción de uso de papel, y mejora en la calidad del servicio educativo.

Indicador	Resultado	Criterio de aceptación
B/C	16.24	$B/C > 1 \rightarrow$ Proyecto viable

*Conclusión:* El proyecto es **claramente rentable y justificado** en términos educativos.

#### 5.1.2.2 Valor Actual Neto (VAN)

El VAN representa el valor neto de los beneficios descontados, considerando una tasa social de descuento del 3%, menos la inversión inicial.

**Resultado obtenido:**

**VAN = S/ 18,558.64**

Este valor refleja los ahorros y mejoras acumuladas que el sistema ofrecerá durante los próximos 5 años en comparación con el costo total de operación.

Indicador	Resultado	Criterio de aceptación
VAN	S/ 18,558.64	$VAN > 0 \rightarrow$ Proyecto viable

*Conclusión:* La inversión inicial se recupera rápidamente y el sistema genera **un valor significativo para la institución**.

#### 5.1.2.3 Tasa Interna de Retorno (TIR)

La TIR indica la rentabilidad relativa del proyecto en comparación con un costo de oportunidad (COK) estimado para el sector educativo entre 3% y 10%.

**Resultado obtenido:**

**TIR = 76.45%**

Esto evidencia un alto retorno del valor educativo en relación con los recursos invertidos.

Indicador	Resultado	Criterio de aceptación
TIR	76.45%	$TIR > COK \rightarrow$ Proyecto viable

*Conclusión:* El sistema **DocuCode-AI** representa una inversión de alta eficiencia, con un retorno institucional considerable.

## 6. Conclusiones

*El análisis de factibilidad realizado para el proyecto DocuCode-AI:  
Sistema Web para la Evaluación de Código y Generación de Diagramas*

*UML demuestra que su desarrollo e implementación son plenamente viables desde múltiples enfoques.*

*Desde el punto de vista técnico, el sistema puede ser construido utilizando tecnologías ampliamente conocidas (PHP, Python, MySQL, OpenAI API, PlantUML), sobre una infraestructura de nube asequible como Elastika, lo cual garantiza su funcionalidad, escalabilidad y compatibilidad.*

*En términos económicos, el proyecto requiere una inversión inicial baja, estimada entre S/ 200.00 a S/ 250.00 anuales, principalmente en servicios de nube y APIs. A cambio, ofrece beneficios académicos que superan los S/ 3,600.00 por año, generando una relación Beneficio/Costo de 16.24, un VAN de S/ 18,558.64 y una TIR del 76.45%, indicadores que lo posicionan como altamente rentable desde una perspectiva institucional y educativa.*

*En cuanto a la factibilidad operativa, se concluye que el sistema puede ser adoptado fácilmente por docentes universitarios sin requerir una curva de aprendizaje elevada, gracias a su interfaz amigable y accesible vía navegador.*

*A nivel legal, social y ambiental, no se identifican barreras que impidan su implementación. Al contrario, el sistema promueve prácticas sostenibles, fomenta la digitalización de la enseñanza y respeta la privacidad de los datos mediante cifrado SSL.*

*En resumen, el proyecto DocuCode-AI es técnica, económica, operativa, legal, social y ambientalmente viable. Su implementación representa una mejora significativa en la evaluación del aprendizaje de programación, aportando eficiencia, objetividad y valor pedagógico al proceso educativo.*