

Factibilidad Tecnológica


*Desarrollo de una Aplicación Web con Machine Learning para
el Análisis de Estados Financieros y la Detección de Fraude en
la Empresa Ciclo Contable*

Fecha: 01/07/2025

1. Información del Proyecto

Empresa / Organización	Ciclo Contable
Proyecto	Desarrollo de una Aplicación Web con Machine Learning para el Análisis de Estados Financieros y la Detección de Fraude en la Empresa Ciclo Contable
Fecha de preparación	16/04/2025
Cliente	Contadores-Audidores de la Empresa Ciclo Contable
Patrocinador principal	Ciclo Contable
Gerente / Líder de Proyecto	Juan Ochoa
Gerente / Líder de Análisis de negocio y requerimientos	Juan Ochoa

2. Aprobaciones

Nombre y Apellido	Cargo	Departamento u Organización	Fecha	Firma
Juan Ochoa	Estudiante	ESPOCH	15/04/2025	
ING. Fredy Gavilánez	Director de Tesis	ESPOCH	15/04/2025	
ING. Natali Ochoa	Ingeniera en Contabilidad y Auditoria	Ciclo Contable	16/04/2025	

3. Factibilidad Tecnológica

a) Infraestructura y Herramientas Tecnológicas

El proyecto se desarrollará utilizando herramientas modernas, de código abierto y con soporte activo de la comunidad, lo que garantiza su viabilidad técnica y sostenibilidad en el tiempo.

Frontend (Interfaz de Usuario):

- **React.js:** Biblioteca JavaScript de código abierto ideal para interfaces dinámicas y escalables.
- **HTML5, CSS3 y JavaScript ES6:** Para la estructura, diseño y funcionalidades del sitio.

Backend (Lógica del servidor):

- **Python con Flask:** Microframework ligero y eficiente para desarrollo rápido de APIs RESTful.
- **PostgreSQL:** Sistema de gestión de bases de datos relacional, confiable y robusto, gratuito en fase de desarrollo.

Machine Learning:

- **Scikit-learn y TensorFlow:** Bibliotecas de Python ampliamente utilizadas para análisis de datos, entrenamiento y despliegue de modelos de ML.

Infraestructura en la nube:

- **Render o Heroku (planes gratuitos):** Para despliegue inicial.
- **Netlify o Vercel:** Para alojar el frontend de forma gratuita.
- **AWS o Azure (opcional):** En caso de requerirse mayor escalabilidad o disponibilidad.

Control de versiones y colaboración:

- **Git y GitHub:** Para gestión del código fuente, control de versiones y colaboración entre desarrolladores.

b) Desafíos Técnicos Identificados

1. Precisión del modelo de Machine Learning:

- Riesgo de errores si no se dispone de suficientes datos financieros reales.
- Solución: Uso de datasets públicos y colaboración con la empresa para obtener datos anonimizados.

2. Escalabilidad de la arquitectura:

- A medida que crecen los usuarios, pueden presentarse problemas de rendimiento.
- Solución: Arquitectura modular con contenedores Docker y migración progresiva a servicios cloud escalables.

3. Integración de múltiples tecnologías:

- Posible dificultad para integrar frontend, backend, base de datos y modelos ML de forma fluida.
- Solución: Desarrollo en fases, pruebas de integración continuas y uso de buenas prácticas DevOps.

4. Seguridad de datos sensibles:

- Riesgo de vulneraciones o fugas de datos.
- Solución: Implementar cifrado (TLS, AES-256), gestión segura de usuarios y auditorías de seguridad.

c) Desarrollo del Sistema

Fases del desarrollo tecnológico:

1. Fase de prototipo:

- Desarrollo básico de la interfaz web.
- Integración inicial con modelos de Machine Learning entrenados con datasets públicos.

2. Fase de integración:

- Conexión con base de datos PostgreSQL.
- Interconexión entre frontend y backend mediante API REST.

3. Fase de despliegue:

- Subida del sistema a servidores en la nube (Render, Netlify, etc.).
- Pruebas de funcionamiento y rendimiento.

4. Fase de validación:

- Evaluación del modelo ML con datos reales.
- Pruebas de usuarios y recolección de feedback.

Herramientas de desarrollo utilizadas:

- Visual Studio Code, Postman, Docker, GitHub.

) Evaluación de los Requerimientos Técnicos y Mantenimiento

Requerimientos mínimos del sistema:

- Navegador web moderno (Chrome, Firefox, Edge).
- Conexión a internet estable.
- Para el backend: Servidor con Python 3.8+, PostgreSQL y mínimo 2GB RAM.
- Para despliegue: Plataforma en la nube con soporte para Docker y Node.js (opcional para React build).

Plan de Mantenimiento:

1. Mantenimiento correctivo:

- Solución de errores detectados en producción.
- Atención a reportes de fallos por parte de usuarios.
- Frecuencia: continuo, con revisiones semanales.

2. Mantenimiento evolutivo:

- Mejoras en funcionalidades según necesidades del negocio o feedback de usuarios.
- Incorporación de nuevos algoritmos o visualizaciones.
- Frecuencia: mensual o por versiones.

3. Mantenimiento preventivo:

- Actualización de dependencias y librerías.
- Monitoreo del rendimiento e integridad del sistema.
- Backup regular de la base de datos y modelos entrenados.
- Frecuencia: cada 15 días.

4. Mantenimiento adaptativo:

- Ajustes necesarios por cambios en regulaciones legales o tecnológicas (por ejemplo, requisitos de la Ley de Protección de Datos).

- Cambios estructurales si se migra a otra nube o se incrementa el tráfico.

Conclusión:

El proyecto es tecnológicamente factible. Las herramientas utilizadas son compatibles entre sí, ampliamente documentadas y con bajo costo de implementación. Si bien existen desafíos técnicos, todos son abordables mediante buenas prácticas de desarrollo, uso de tecnologías escalables y una estrategia de mantenimiento planificada.