**Descripción del Problema**

Los hackeos de cuentas bancarias se han convertido en una preocupación creciente para los clientes del Banco de Venezuela. Los ciberdelincuentes emplean diversas técnicas, como el phishing, el malware y el robo de credenciales, para obtener acceso no autorizado a las cuentas de los usuarios y realizar transacciones fraudulentas. Esta situación genera pérdidas económicas significativas para los afectados, erosiona la confianza en el sistema bancario digital y exige la implementación de medidas de seguridad más robustas y efectivas.

El problema se manifiesta de las siguientes maneras:

* **Aumento de casos de fraude:** Los clientes del banco reportan un incremento en la frecuencia de transacciones no reconocidas en sus cuentas, lo que indica un aumento en la actividad de los ciberdelincuentes.
* **Vulnerabilidad de los sistemas de seguridad:** Los métodos de autenticación y autorización actuales pueden no ser suficientes para proteger las cuentas de los usuarios de las técnicas de ataque más sofisticadas.
* **Falta de conciencia de los usuarios:** Muchos clientes pueden no estar al tanto de los riesgos de seguridad en línea ni de las mejores prácticas para proteger sus credenciales y su información financiera.
* **Impacto en la reputación del banco:** Los incidentes de hackeo pueden dañar la reputación del Banco de Venezuela y generar desconfianza entre sus clientes.

**Posibles Soluciones:**

Para mitigar el problema de los hackeos de cuentas bancarias, se pueden implementar las siguientes medidas:

* **Autenticación multifactor (MFA):** Requerir a los usuarios que proporcionen múltiples formas de verificación de identidad, como contraseñas, códigos enviados a sus teléfonos móviles o datos biométricos, para acceder a sus cuentas. Esto dificulta que los ciberdelincuentes puedan acceder a las cuentas, incluso si han robado las contraseñas.
* **Detección de anomalías:** Implementar sistemas que analicen el comportamiento de las transacciones de los usuarios y detecten patrones inusuales que puedan indicar fraude. Estos sistemas pueden utilizar inteligencia artificial y aprendizaje automático para identificar actividades sospechosas en tiempo real.
* **Actualizaciones de seguridad:** Mantener actualizados todos los sistemas y software del banco para corregir las vulnerabilidades conocidas que los ciberdelincuentes pueden aprovechar. Esto incluye la aplicación de parches de seguridad a los servidores, las bases de datos y las aplicaciones móviles.
* **Monitoreo continuo:** Supervisar continuamente los sistemas de seguridad del banco para detectar y responder rápidamente a cualquier intento de ataque. Esto puede implicar el uso de herramientas de detección de intrusiones, firewalls y otras tecnologías de seguridad.

## Tipo De Modelo

Si bien el proyecto se centra en la detección de fraude, la implementación de una solución efectiva en un entorno bancario dinámico se beneficiaría enormemente de la adopción de una metodología DevOps. DevOps promueve la colaboración, la automatización y la mejora continua, lo cual es esencial para responder a la evolución de las amenazas y garantizar la confiabilidad del sistema.

**Este proyecto utilizará la siguiente metodología:**

* **Detección de fraude en tiempo real:** La naturaleza de los ataques de fraude exige un sistema que pueda responder con rapidez a las nuevas amenazas. DevOps permite la implementación de actualizaciones de seguridad y ajustes de algoritmos de forma continua, lo que garantiza que el sistema de detección de fraude se mantenga eficaz.
* **Adaptabilidad a las amenazas:** El panorama del fraude está en constante evolución, con los ciberdelincuentes desarrollando continuamente nuevas técnicas. DevOps facilita la adaptación rápida del sistema a estos cambios mediante la integración continua y la entrega continua (CI/CD).
* **Confiabilidad y disponibilidad:** El sistema de detección de fraude debe estar disponible en todo momento para proteger las transacciones de los usuarios. DevOps garantiza la confiabilidad del sistema mediante la automatización de las pruebas, el monitoreo continuo y la rápida recuperación ante fallos.
* **Escalabilidad:** El sistema debe ser capaz de manejar grandes volúmenes de transacciones y crecer junto con las necesidades del banco. DevOps permite una escalabilidad horizontal sencilla mediante la infraestructura como código (IaC) y la automatización de la implementación.
* **Mejora continua:** DevOps promueve una cultura de aprendizaje y experimentación, lo que permite al equipo analizar continuamente el rendimiento del sistema, identificar áreas de mejora e implementar nuevas estrategias de detección de fraude.

**Análisis De La Viabilidad**

 **Viabilidad Técnica:** La disponibilidad y calidad de los datos, la infraestructura tecnológica existente y la capacidad del equipo de desarrollo son cruciales. Si el banco tiene datos de alta calidad y la infraestructura necesaria, y cuenta con un equipo capacitado, la viabilidad técnica es alta.

 **Viabilidad Económica:** El presupuesto disponible, el ROI esperado y los costos de mantenimiento son consideraciones importantes. El proyecto será viable si el presupuesto es suficiente, el ROI es positivo y los costos de mantenimiento son manejables.

 **Viabilidad Legal:** El cumplimiento de las regulaciones y leyes de protección de datos es esencial. El proyecto debe diseñarse para cumplir con todos los requisitos legales aplicables para evitar problemas en el futuro.

 **Viabilidad Operacional:** La integración con los sistemas existentes, los procesos internos del banco y la capacitación del personal son factores importantes. El proyecto será viable si se puede integrar sin problemas en la infraestructura existente del banco, se alinea con los procesos internos y el personal puede ser capacitado para utilizarlo de manera efectiva.

**Objetivo General:**

Desarrollar un sistema inteligente y eficiente para la validación de transacciones financieras que minimice el riesgo de fraude, mediante la integración de técnicas de inteligencia artificial y la implementación de medidas de seguridad robustas.

**Objetivos Específicos:**

1. Diseñar e implementar una API RESTful utilizando FastAPI para la gestión y validación de transacciones.
2. Integrar modelos de aprendizaje automático para la detección de patrones de fraude y la priorización de preguntas de seguridad.
3. Desarrollar un prototipo de interfaz de usuario para la simulación y demostración del sistema de validación de transacciones.
4. Garantizar la seguridad de los datos sensibles mediante la implementación de técnicas de encriptación avanzadas (AES-256).
5. Realizar pruebas exhaustivas del sistema, incluyendo pruebas unitarias y de carga, para asegurar su correcto funcionamiento y escalabilidad.
6. Desplegar el sistema en un entorno de producción y realizar ajustes basados en el feedback y el rendimiento real.
7. Documentar el proceso de desarrollo, la arquitectura del sistema y las decisiones técnicas tomadas.

**DESCRIPCIÓN DE ACTIVIDADES EDT**

1. Fase de Preparación (2 meses)

Objetivo:Bases técnicas, legales y de datos.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Actividad | Tareas Específicas | Entregables |
| 1.1 | Cumplimiento Legal | - Revisar regulaciones (Ley venezolana, PCI DSS).  - Acuerdos con el banco para datos. | Documento de requisitos legales. |
| 1.2 | Infraestructura | - Configurar AWS Lightsail. GitHub Actions (CI/CD).  Instalar Prometheus. | Entorno cloud operativo. |
| 1.3 | Preparación de Datos | - Evaluar datasets reales.  Limpiar datos con Pandas.  -Crear datasets sintéticos. | Dataset listo para entrenamiento. |

2. Fase de Desarrollo (5 meses)

Objetivo: API funcional + Modelo ML.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Actividad | Tareas Específicas | Entregables |
| 2.1 | API FastAPI | - Endpoint `/detectar-fraude`.  - JWT + MFA (Twilio).  - Documentar con Swagger. | API básica funcional. |
| 2.2 | Modelo ML | - Algoritmo LightGBM.  - Entrenamiento.  - Validar Recall 85%. | Modelo entrenado (.pkl). |
| 2.3 | Integración API-ML | - Conectar modelo a endpoint.  - Optimizar latencia 200ms. | Endpoint operativo. |
| 2.4 | Seguridad | - Encriptar datos (AES-256).  - AWS KMS para claves. | Reporte de seguridad. |

3. Fase de Pruebas (1 mes)

Objetivo: Validar funcionalidad y seguridad.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Actividad | Tareas Específicas | Entregables |
| 3.1 | Pruebas Unitarias | - Ejecutar `pytest`.  - Validar respuestas del modelo. | Reporte de cobertura > 90%. |
| 3.2 | Pruebas de Carga | - Simular 1,000 transacciones/min (Locust). | Reporte de rendimiento. |
| 3.3 | Seguridad | - Pentesting básico (OWASP ZAP).  - Validar encriptación. | Reporte de vulnerabilidades. |

4. Fase de Despliegue (1 mes)

Objetivo: Lanzamiento controlado.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Actividad | Tareas Específicas | Entregables |
| 4.1 | Despliegue Gradual | - Staging (10% tráfico).  - Monitoreo con Grafana.  - Escalar a 100%. | Sistema en producción. |
| 4.2 | Capacitación | - Manual técnico.  - Tutoriales de uso. | Documentación final. |

5. Monitoreo Continuo

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Actividad | Tareas Específicas | Entregables |
| 5.1 | Monitoreo | - Alertas para transacciones sospechosas.  - Dashboard Grafana. | Reportes mensuales. |
| 5.2 | Mantenimiento | - Actualizar modelo ML trimestral.  - Parches de seguridad. | Registro de actualizaciones. |