CYMETRIA Group S.A.S

Programa:
INTELIGENCIA ARTIFICIAL FULL STACK

Nombre del Proyecto:
Sistema de Detección y Prevención de Fraudes Financieros con Tecnología Blockchain e IA

Integrantes del Grupo:
- LEON MORENO
- ANYURY MUÑOZ
- JHONATAN BOLÍVAR
- SANTIAGO VASQUEZ
-BRAYAN ALEJANDRO MUÑOZ
-

Docente:
Felipe Alexander Garzón

Bogotá D.C.
Fecha: 22/11/2024

Tabla de contenido

1. Resumen

El presente proyecto aborda el problema del fraude financiero en transacciones digitales, una preocupación creciente en un contexto donde las operaciones electrónicas son fundamentales para usuarios y entidades financieras. Las instituciones enfrentan desafíos significativos para detectar y prevenir movimientos sospechosos en tiempo real, mientras los usuarios demandan mayor seguridad y transparencia en el manejo de su dinero.

Para resolver este problema, se diseñó e implementó un software integral de detección y prevención de fraudes basado en algoritmos avanzados y tecnología blockchain. Este sistema permite identificar patrones de fraude mediante el análisis en tiempo real de transacciones, generar alertas automáticas y registrar las operaciones en una blockchain para garantizar la inmutabilidad de los datos. Además, se desarrollaron interfaces intuitivas para clientes, administradores y analistas, adaptadas a las necesidades de cada perfil.

El impacto esperado del software incluye la reducción significativa de fraudes financieros, el aumento de la confianza del usuario en las aplicaciones financieras y la mejora de los procesos internos de monitoreo de seguridad. Este proyecto también sienta las bases para futuras integraciones con tecnologías emergentes, como inteligencia artificial, permitiendo una evolución continua frente a amenazas dinámicas en el sector financiero.

2. Introducción

En un mundo donde la transformación digital ha redefinido la forma en que se realizan las transacciones financieras, el fraude digital se ha convertido en una amenaza creciente que afecta tanto a instituciones financieras como a los usuarios finales. Según datos de la Asociación Latinoamericana de Seguridad Financiera (ALSEFI, 2021), las pérdidas por fraudes electrónicos superaron los $4.5 mil millones anuales en la región, evidenciando la necesidad de herramientas más efectivas para mitigar estos riesgos.

El problema radica en la incapacidad de muchos sistemas tradicionales para detectar y responder a patrones sospechosos en tiempo real. Esto deja a los usuarios vulnerables a actividades como transferencias no autorizadas, robo de identidad, y accesos fraudulentos a cuentas bancarias. A nivel institucional, el fraude no solo genera pérdidas económicas directas, sino que también compromete la confianza de los clientes en los servicios financieros digitales.

Este proyecto busca abordar este problema mediante el desarrollo de un software integral que combine algoritmos avanzados de detección de fraudes con la tecnología blockchain, garantizando seguridad y transparencia en las transacciones. La implementación de notificaciones en tiempo real, reglas

personalizables, y un registro inmutable de transacciones tiene el potencial de transformar la forma en que se monitorean y previenen los fraudes, beneficiando tanto a usuarios individuales como a grandes organizaciones. Este enfoque proactivo responde a la urgencia de fortalecer la confianza en los servicios financieros en la era digital.

3. Objetivo general

Desarrollar un software integral que permita la detección y prevención de fraudes financieros en tiempo real, mejorando la seguridad de las transacciones digitales mediante la integración de tecnología blockchain, algoritmos avanzados de análisis y notificaciones automáticas para aumentar la confianza de los usuarios y la eficiencia operativa de las instituciones financieras.

4. Objetivos específicos

 Analizar **el problema y sus requerimientos**:

- Recolectar información a través de encuestas, entrevistas y revisión de documentación técnica para comprender los patrones y riesgos asociados al fraude financiero.
- Identificar las funcionalidades clave que el software debe cumplir para satisfacer las necesidades de los usuarios y las instituciones financieras.

 Diseñar **el modelo del sistema**:

- Crear un Modelo Entidad-Relación (MER) que estructure la base de datos para gestionar clientes, cuentas, transacciones y alertas.
- Elaborar diagramas de casos de uso y diagramas de flujo que definan los procesos principales del sistema.

 Implementar **la arquitectura del software**:

- Desarrollar un backend robusto que procese las transacciones, aplique las reglas de detección de fraude y se comunique con blockchain.
- Crear un frontend intuitivo y responsivo que permita a los usuarios finales, administradores y analistas interactuar fácilmente con el sistema.

 Integrar **tecnologías avanzadas**:

- Incorporar blockchain para registrar transacciones de forma inmutable y garantizar transparencia.
- Implementar notificaciones en tiempo real para alertar de actividades sospechosas a los usuarios y administradores.

 Probar **y optimizar el sistema**:
- Realizar pruebas de funcionalidad, seguridad y rendimiento en todos los módulos del sistema.
- Ajustar los algoritmos y parámetros de detección para minimizar falsos positivos y garantizar la eficiencia del software.

 Documentar **y capacitar**:
- Elaborar una guía de uso clara para clientes, administradores y analistas.
- Ofrecer capacitación básica sobre las funcionalidades del sistema para maximizar su adopción y efectividad.

5. Desarrollo

Esta sección describe cada una de las fases del proyecto: análisis, diseño y desarrollo del software, incluyendo detalles sobre los métodos, herramientas y técnicas empleadas.

5.1 Análisis de Software

5.1.1 Recolección de Información

Se llevó a cabo un análisis exhaustivo mediante:

1. **Encuestas a usuarios finales**:
    - Preguntas centradas en la frecuencia de uso de aplicaciones financieras, problemas de seguridad enfrentados y preferencias respecto a la tecnología blockchain.
    - Herramienta: **Google Forms**.
    - **Resultados clave**:
        - Más del 50% de los encuestados han enfrentado problemas de seguridad en aplicaciones financieras.
        - Los usuarios priorizan notificaciones inmediatas de movimientos sospechosos.

- La implementación de blockchain fue considerada favorable para aumentar la confianza.

2. **Revisión documental**:
   - Análisis de investigaciones sobre ciberseguridad y fraude financiero, destacando la necesidad de monitoreo en tiempo real y herramientas de detección basadas en patrones.

3. **Entrevistas con expertos en ciberseguridad financiera**:
   - Opiniones de administradores y analistas sobre parámetros de seguridad clave (montos máximos, ubicaciones inusuales).

5.2 Diseño de Software

5.2.1 Modelo Entidad-Relación (MER)

El MER estructura la base de datos para capturar todas las interacciones y garantizar la trazabilidad de las transacciones.
Entidades y Relaciones Clave:

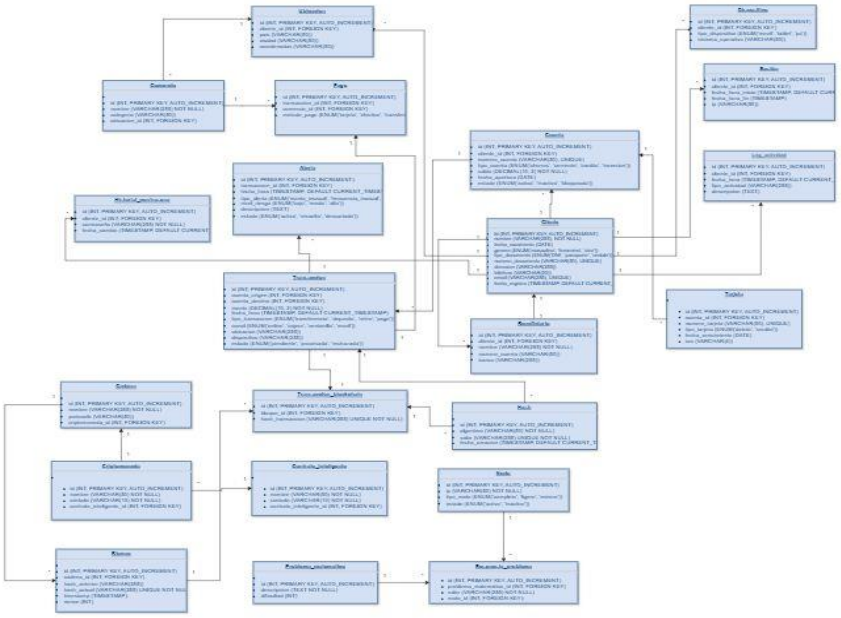
1. **Cliente**:
   - Campos: id, nombre, email, contraseña, fecha_registro.
   - Relación: 1:N con **Cuenta**.

2. **Cuenta**:
   - Campos: id, cliente_id, saldo, tipo_cuenta, estado.
   - Relación: 1:N con **Transacción**.

3. **Transacción**:
   - Campos: id, cuenta_origen, cuenta_destino, monto, fecha_hora, hash_blockchain.
   - Relación: 1:N con **Alerta**.

4. **Alerta**:
   - Campos: id, transaccion_id, tipo_alerta, descripcion, nivel_riesgo.

Diagrama MER: El diagrama muestra cómo las entidades están relacionadas y refleja la lógica del negocio.



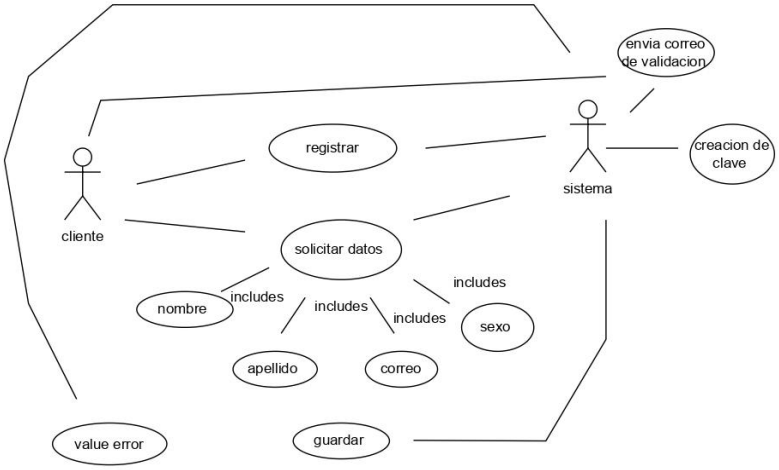
5.2.2 Casos de Uso

Caso de Uso 1: Registrar Cliente
- **Actor**: Cliente.
- **Descripción**: El cliente ingresa su información personal y se registra en el sistema.
- **Propósito**: Garantizar un acceso seguro y único.

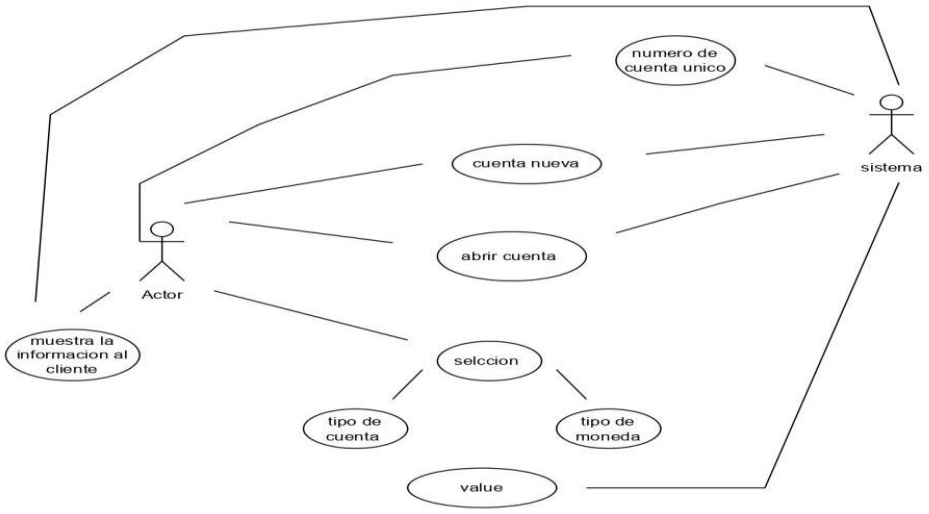
Caso de Uso 2: Realizar Transferencia
- **Actor**: Cliente.
- **Descripción**: El cliente transfiere dinero entre cuentas.
- **Flujo**:
    1. Validar fondos disponibles.
    2. Procesar transferencia y registrar en blockchain.
    3. Notificar al cliente.

Caso de Uso 3: Configurar Parámetros
- **Actor**: Administrador.
- **Descripción**: El administrador establece umbrales de alerta (e.g., montos máximos, frecuencia de transacciones).

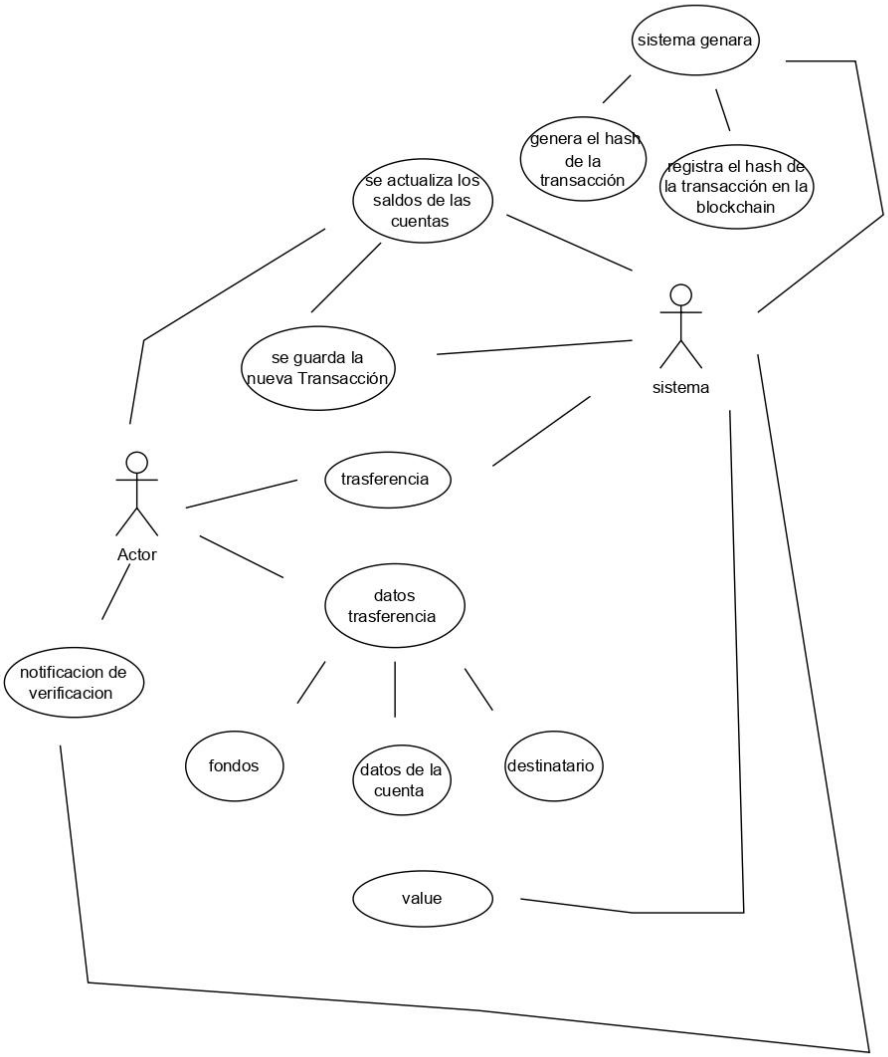
Caso1 - Registro Cliente



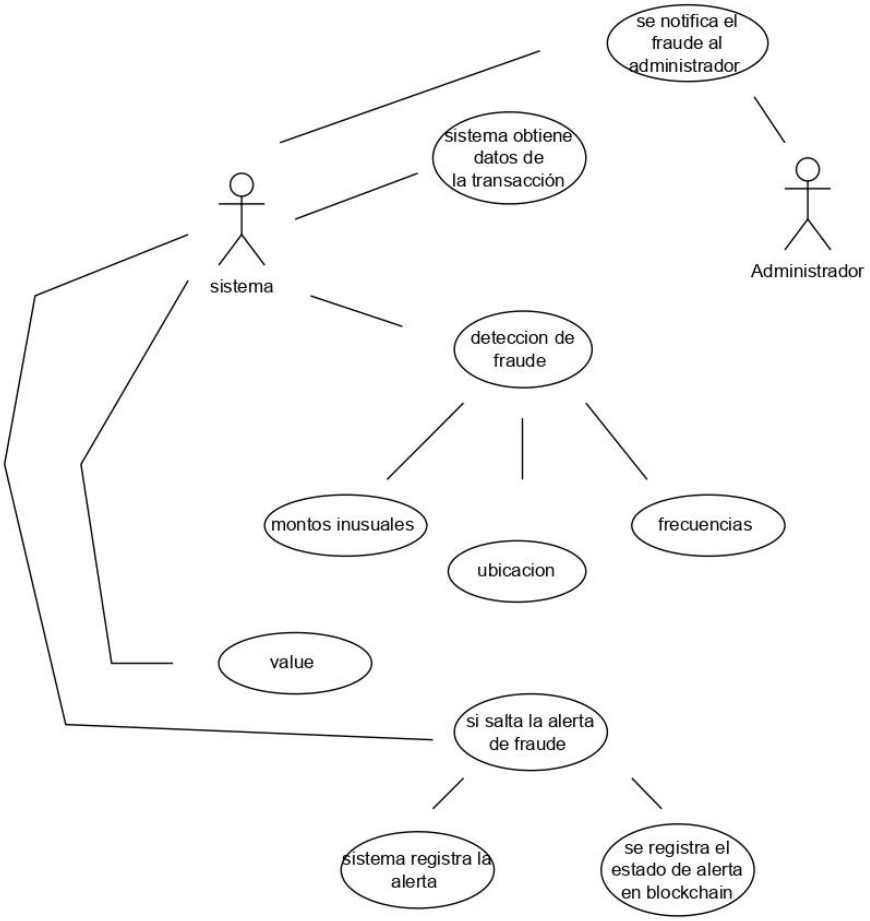
Caso 2- Abrir cuenta



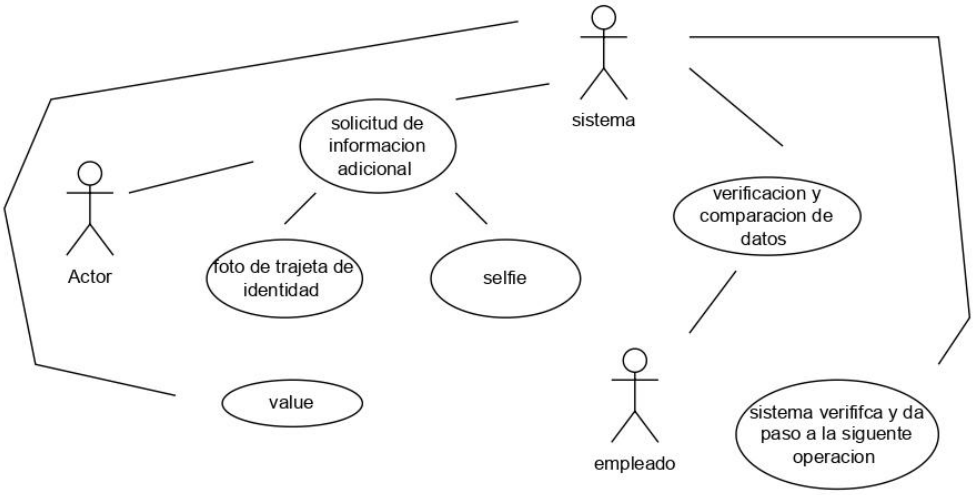
Caso 3 - Realizar transacción

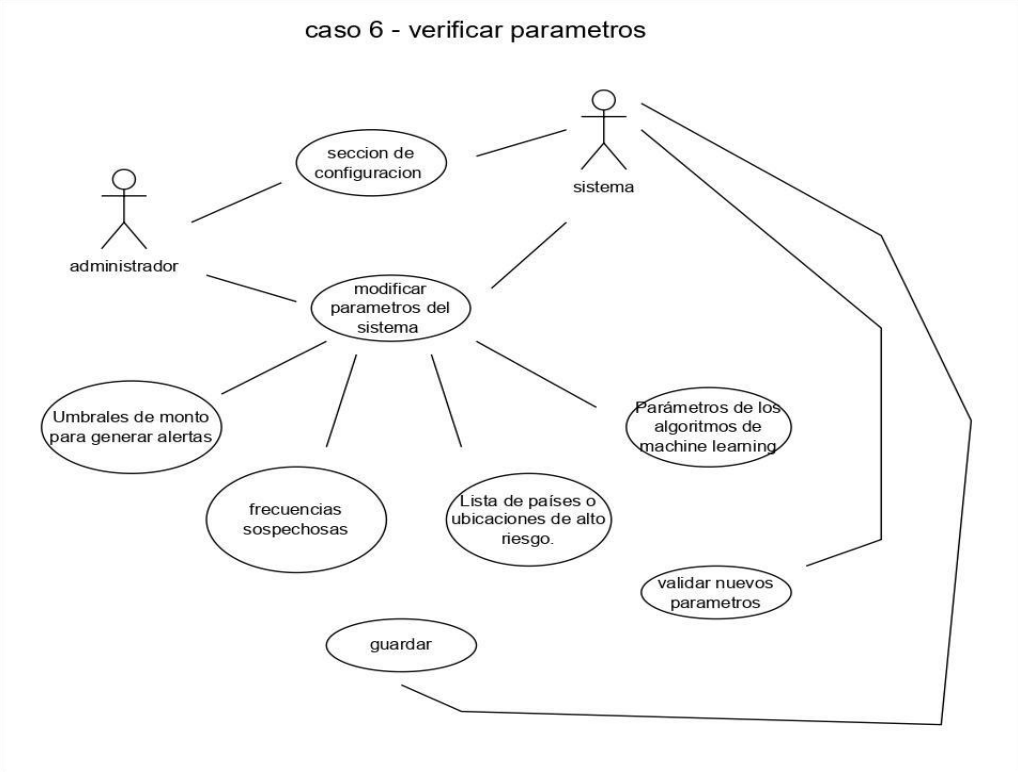


Caso 4 - Ananlizar transacción



Caso 5 - Verificar identidad

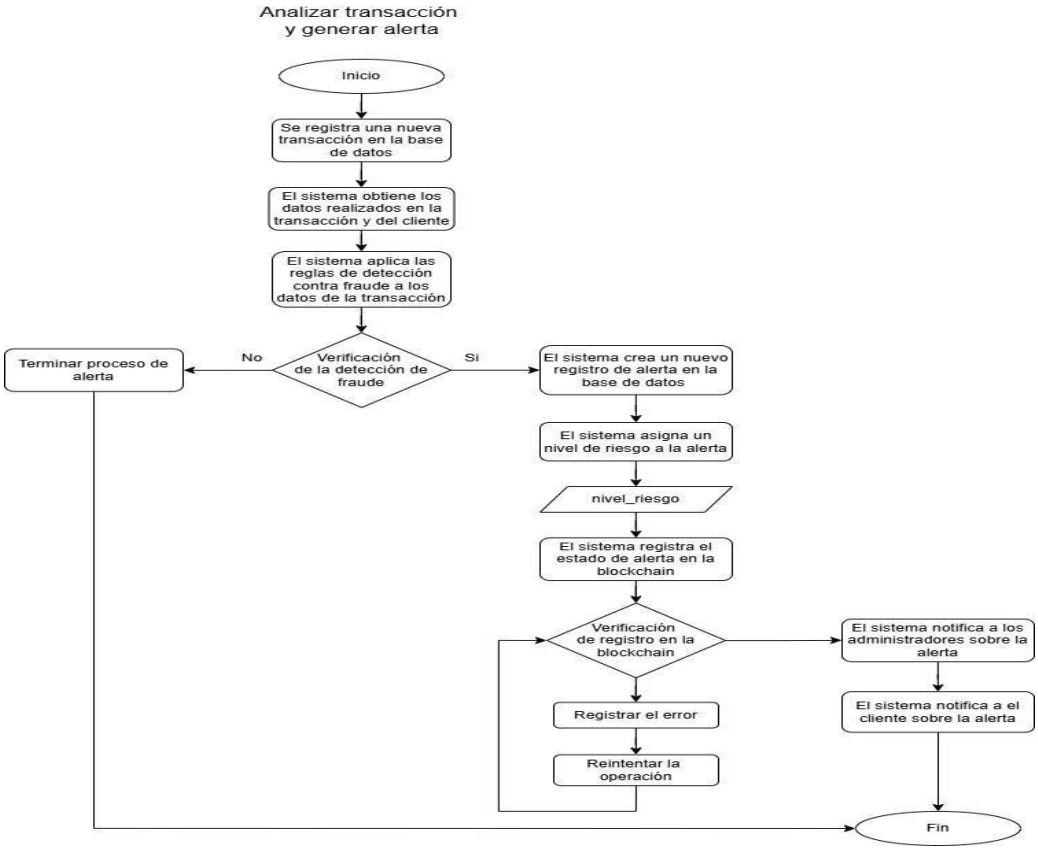


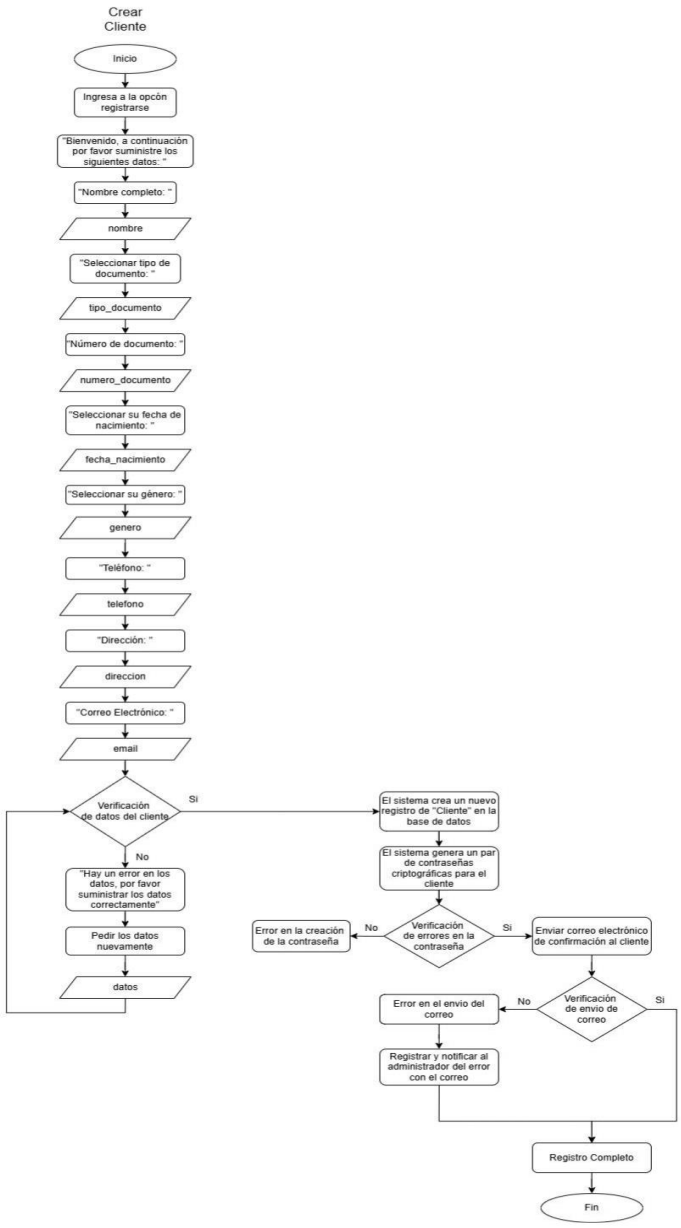


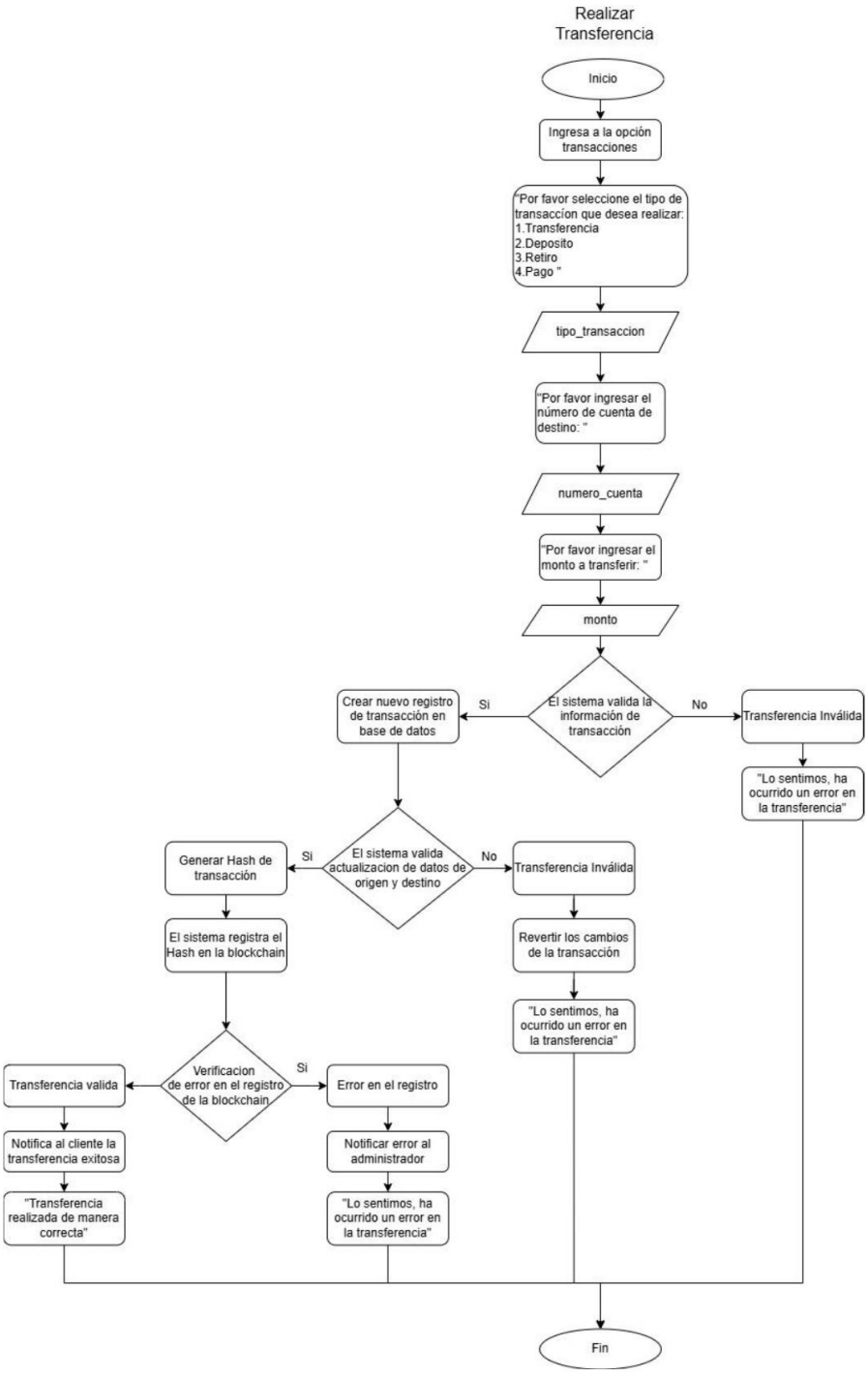
5.2.3 Diagrama de Flujo

Ejemplo: **Flujo para la Realización de Transferencias**

1. El cliente accede a la opción "Transferir".
2. Ingresan los detalles de la transferencia (cuenta destino, monto).
3. El sistema valida los datos:
    - Fondos disponibles.
    - Cuenta destino válida.
4. Si es válido:
    - Se registra la transacción.
    - Se genera un hash en blockchain.
    - Se actualizan los saldos.
5. Si falla, se notifica al cliente.


Analizar transacción y generar alerta
Inicio
Se registra una nueva transacción en la base de datos
El sistema obtiene los datos realizados en la transacción y del cliente
El sistema aplica las reglas de detección contra fraude a los datos de la transacción
Verificación de la detección de fraude
No
Si
Terminar proceso de alerta
El sistema crea un nuevo registro de alerta en la base de datos
El sistema asigna un nivel de riesgo a la alerta
nivel_riesgo
El sistema registra el estado de alerta en la blockchain
Verificación de registro en la blockchain
Registrar el error
Reintentar la operación
El sistema notifica a los administradores sobre la alerta
El sistema notifica a el cliente sobre la alerta
Fin




Realizar Transferencia
Inicio
Ingresa a la opción transacciones
"Por favor seleccione el tipo de transaccíon que desea realizar:
1.Transferencia
2.Deposito
3.Retiro
4.Pago "
tipo_transaccion
"Por favor ingresar el número de cuenta de destino: "
numero_cuenta
"Por favor ingresar el monto a transferir: "
monto
El sistema valida la información de transacción
Si
No
Crear nuevo registro de transacción en base de datos
Transferencia Inválida
"Lo sentimos, ha ocurrido un error en la transferencia"
El sistema valida actualizacion de datos de origen y destino
Si
No
Generar Hash de transacción
Transferencia Inválida
El sistema registra el Hash en la blockchain
Revertir los cambios de la transacción
"Lo sentimos, ha ocurrido un error en la transferencia"
Verificacion de error en el registro de la blockchain
Si
Transferencia valida
Error en el registro
Notifica al cliente la transferencia exitosa
Notificar error al administrador
"Transferencia realizada de manera correcta"
"Lo sentimos, ha ocurrido un error en la transferencia"
Fin

5.3 Desarrollo de Software

5.3.1 Implementación del Código

Tecnologías Utilizadas:

- **Backend**: Flask con Flask-SocketIO.
- **Frontend**: React.js con Material-UI.
- **Base de Datos**: MySQL.
- **Blockchain**: Ethereum (mediante Web3.py).

Ejemplo de Endpoint del Backend:
Python

```
@app.route('/registrar-cliente', methods=['POST'])
def registrar_cliente():
    data = request.json
    conn = db_connection()
    cursor = conn.cursor()
    cursor.execute(
        "INSERT INTO Cliente (nombre, email, contraseña) VALUES (%s, %s, %s)",
        (data['nombre'], data['email'], data['contraseña'])
    )
    conn.commit()
    return jsonify({"message": "Cliente registrado exitosamente"}), 201
```

Ejemplo de Componente Frontend:
javascript

```
function RegistrarCliente() {
    const [nombre, setNombre] = useState('');
    const [email, setEmail] = useState('');
    const [contraseña, setContraseña] = useState('');

    const registrar = async () => {
        const response = await fetch('http://localhost:5000/registrar-cliente', {
            method: 'POST',
            headers: { 'Content-Type': 'application/json' },
```

```
        body: JSON.stringify({ nombre, email, contraseña }),
      });
      const data = await response.json();
      alert(data.message || 'Error al registrar cliente');
    };

    return (
      <div>
        <input placeholder="Nombre" onChange={(e) =>
setNombre(e.target.value)} />
        <input placeholder="Email" onChange={(e) => setEmail(e.target.value)}
/>
        <input placeholder="Contraseña" type="password" onChange={(e) =>
setContraseña(e.target.value)} />
        <button onClick={registrar}>Registrar Cliente</button>
      </div>
    );
}
```

5.3.2 Configuración de la Base de Datos
Esquema SQL Principal:

sql

```
CREATE TABLE Cliente (
    id INT PRIMARY KEY AUTO_INCREMENT,
    nombre VARCHAR(255) NOT NULL,
    email VARCHAR(255) UNIQUE NOT NULL,
    contraseña VARCHAR(255) NOT NULL,
    fecha_registro TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE Cuenta (
    id INT PRIMARY KEY AUTO_INCREMENT,
    cliente_id INT,
    saldo DECIMAL(10, 2) DEFAULT 0.00,
    tipo_cuenta ENUM('ahorros', 'corriente'),
    estado ENUM('activa', 'inactiva') DEFAULT 'activa',
    FOREIGN KEY (cliente_id) REFERENCES Cliente(id)
);
```

```
CREATE TABLE Transaccion (
    id INT PRIMARY KEY AUTO_INCREMENT,
    cuenta_origen INT,
    cuenta_destino INT,
    monto DECIMAL(10, 2),
    fecha_hora TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    hash_blockchain VARCHAR(255),
    FOREIGN KEY (cuenta_origen) REFERENCES Cuenta(id),
    FOREIGN KEY (cuenta_destino) REFERENCES Cuenta(id)
);

CREATE TABLE Alerta (
    id INT PRIMARY KEY AUTO_INCREMENT,
    transaccion_id INT,
    tipo_alerta ENUM('monto_inusual', 'frecuencia'),
    descripcion TEXT,
    nivel_riesgo ENUM('bajo', 'medio', 'alto'),
    FOREIGN KEY (transaccion_id) REFERENCES Transaccion(id)
);
```

5.3.3 Desafíos y Soluciones

1. **Problema: Registro de hashes en blockchain**:
   - **Solución**: Implementar colas de procesamiento en segundo plano para minimizar la latencia.

2. **Problema: Optimización de reglas de fraude**:
   - **Solución**: Ajustar dinámicamente los umbrales basados en datos históricos.

6. Conclusión

El desarrollo del sistema de detección y prevención de fraudes financieros logró cumplir con los objetivos planteados al inicio del proyecto, aportando una solución robusta y escalable para abordar un problema crítico en el sector financiero.

Logros del Proyecto

1. **Detección Eficiente de Fraudes**:
    - Implementación de algoritmos en tiempo real para detectar patrones sospechosos en las transacciones, reduciendo significativamente el tiempo de respuesta frente a posibles fraudes.
2. **Uso de Blockchain para Seguridad y Transparencia**:
    - El registro de transacciones en blockchain garantiza la inmutabilidad de los datos, aumentando la confianza de los usuarios y facilitando auditorías.
3. **Interfaz Intuitiva y Responsiva**:
    - Diseño de módulos específicos para cada actor del sistema (clientes, administradores y analistas), permitiendo un uso sencillo y una gestión eficiente de las operaciones.
4. **Notificaciones en Tiempo Real**:
    - Integración de WebSockets para alertas instantáneas, mejorando la capacidad de respuesta ante eventos críticos.
5. **Adaptación a Necesidades Dinámicas**:
    - Permitir a los administradores ajustar los parámetros del sistema (e.g., umbrales de fraude, reglas de monitoreo) ofrece una solución flexible frente a amenazas emergentes.

Impacto del Software

El sistema no solo mejora la seguridad de las transacciones financieras, sino que también genera confianza entre los usuarios, al brindarles herramientas para monitorear y controlar sus operaciones. Para las instituciones financieras,

el software representa una solución innovadora que combina eficiencia operativa y cumplimiento normativo.

Reflexiones y Aprendizajes

1. **Importancia de la Colaboración**:
    - El éxito del proyecto fue posible gracias a la interacción entre desarrolladores, expertos en seguridad financiera y usuarios finales. Este enfoque permitió comprender las necesidades reales y diseñar una solución adaptada.
2. **Adopción de Nuevas Tecnologías**:
    - La integración de blockchain demostró ser un componente clave para garantizar la seguridad y la trazabilidad. Este aprendizaje subraya el valor de explorar tecnologías emergentes.
3. **Escalabilidad desde el Diseño**:
    - Diseñar con una arquitectura modular permitió no solo cumplir con los requerimientos actuales, sino también preparar el sistema para futuras expansiones.

Futuras Mejoras y Adaptaciones

1. **Integración con Inteligencia Artificial**:
    - Incorporar modelos de aprendizaje automático para mejorar la detección de fraudes, basándose en análisis predictivo de patrones históricos.
2. **Soporte Multiplataforma**:
    - Ampliar el sistema para incluir aplicaciones móviles nativas, aumentando su alcance y accesibilidad.
3. **Monitoreo Global**:

- Permitir la integración con sistemas internacionales de monitoreo, ampliando la cobertura del software a redes financieras globales.

4. **Optimización de Procesamiento en Blockchain**:

   - Investigar alternativas como blockchain privada o híbrida para reducir costos y mejorar la velocidad de registro de transacciones.

El proyecto representa un avance significativo en la lucha contra el fraude financiero, destacándose por su capacidad de adaptarse a las necesidades cambiantes del sector y su potencial para integrarse con tecnologías más avanzadas en el futuro. Este desarrollo reafirma la importancia de combinar innovación tecnológica con un enfoque centrado en el usuario para resolver problemas complejos y de alto impacto.

7. Referencias

 National Institute of Standards and Technology. (2019). *Framework for Improving Critical Infrastructure Cybersecurity, Version 1.1*. Recuperado de https://nvlpubs.nist.gov

 Sayeed, S., Marco-Gisbert, H., & Hasan, R. (2020). Assessing Blockchain's Applicability in Combatting Fraud: A Comprehensive Survey. *IEEE Access*, 8, 21091-21112. https://doi.org/10.1109/ACCESS.2020.2968372

 Romero, G., & Hernández, A. (2023). *La ciberseguridad financiera en la era digital*. Universidad de Valencia, España.

 Asociación Latinoamericana de Seguridad Financiera. (2021). *Tendencias de Fraude en Transacciones Digitales*. Recuperado de https://alsefi.org/reportes

 SurveyMonkey. (2024). *Resultados de Encuestas sobre Seguridad Financiera de los Usuarios*. Datos recolectados mediante formulario digital, enero de 2024.

 Microsoft Excel. (2024). Análisis estadístico de los resultados obtenidos en las encuestas sobre seguridad financiera. Herramienta utilizada para analizar los datos recolectados.

- Lucid chart. (2024). Diagramas de flujo para el análisis de procesos de detección de fraude. Recuperado de https://lucidchart.com

- Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. Recuperado de https://bitcoin.org/bitcoin.pdf

- OpenAI. (2024). Modelos avanzados para la detección de anomalías en transacciones financieras. Consultado el 22 de noviembre de 2024, desde https://openai.com

- Hyperledger Foundation. (2024). *Uso de Blockchain en la Gestión de Transacciones Financieras*. Recuperado de https://hyperledger.org

- Flask-SocketIO. (2024). *Documentación oficial para la implementación de notificaciones en tiempo real*. Consultado desde https://flask-socketio.readthedocs.io

- React.js. (2024). *Guía para el desarrollo de interfaces de usuario dinámicas y responsivas*. Documentación oficial: https://reactjs.org

- Web3.py. (2024). *Integración de Blockchain con aplicaciones Python*. Consultado desde https://web3py.readthedocs.io

- Chart.js. (2024). *Visualización de datos financieros mediante gráficos interactivos*. Recuperado de https://chartjs.org

- MySQL Documentation. (2024). *Guía para la gestión de bases de datos relacionales en sistemas financieros*. Consultado desde https://dev.mysql.com/doc

8. Nota final

Este campo está diseñado exclusivamente para los jurados esperamos un excelente ficback!