**Consignas para la resolución del examen**

1. **Conectarse a Miel.**
2. **Descargar el presente documento, junto con las Interfaces que serán utilizadas como el contrato de nuestro Sistema.**
3. **Crear un nuevo Proyecto en Eclipse.**
4. **Compartir el nuevo proyecto en un repositorio público de GitHub**
5. **Actualizar el proyecto siguiendo lo descripto en las siguientes páginas, usando la metodología TDD.**
6. **Entregar por prácticas de Miel la URL del repositorio público. Sólo se contemplarán los cambios subidos dentro del horario del examen.**

**Cyber Security**

**Con la llegada de la pandemia el uso de la banca móvil creció de manera exponencial. Las restricciones impuestas hicieron que incluso los más escépticos en el uso de la tecnología no tuvieran mas remedio que empezar a utilizarla, dado que no tenían la posibilidad de realizar sus trámites en las sucursales de los bancos o tenían que realizar sus compras a través de Internet. Pero por supuesto, de la mano con el incremento del uso de estas nuevas tecnologías, se incrementaron las estafas y los fraudes electrónicos. Para cada nueva funcionalidad, los defraudadores (o mejor dicho ciber defraudadores), encontraron una nueva oportunidad de fraude, siendo la ingeniería social y el phishing las más utilizadas.**

**Viendo esta situación, la Unlam nos propone el desarrollo de una herramienta que se encargue de MONITOREAR todas las TRANSACCIONES que realiza un Banco, de manera que esta herramienta vaya tomando la inteligencia suficiente para identificar cuando unan transacción resulta ser fraudulenta.**

**Esto con la intención de brindarles a los bancos públicos argentinos la posibilidad de protegerse contra este nuevo flagelo y por ende cuidar a sus clientes.**

**Obviamente este proyecto requerirá la conexión de todos los movimientos que se realizan en los bancos asociados con el nuevo software a desarrollar por parte de la universidad, pero inicialmente se plantea ir desarrollando el back-end que contenga la mayor cantidad de lógica de negocio que permita atender la necesidad planteada.**

**Lo primero que es importante saber, es que para saber si una transacción es fraudulenta, es necesario:**

1. **Conocer las características de los CLIENTES.**
2. **Conocer las características de los DISPOSITIVOS que usan los clientes.**
3. **Conocer las características de las TRANSACCIONES.**

**Dicho esto, va a ser necesario poder almacenar estos elementos fundamentales en el diseño de la solución.**

**De los clientes, necesitamos conocer su CUIT y nombre, pero también va a ser fundamental conocer las características de los DISPOSITIVOS que utilizan habitualmente. Esto es; si se conecta desde una PC, conocer su SO, dirección IP, y localidad desde donde hizo la conexión. Si se conecta desde un dispositivo móvil, a todos esos datos debemos incorporar el IMEI del dispositivo y si se registró mediante biometría (huella o rostro) o no.**

**De las TRANSACCIONES, es importante clasificarlas en dos tipos. Aquellas transacciones monetarias (Extracciones, Transferencias, PagosQR, PagosServicios) por un lado, y las transacciones no monetarias (Onboarding o alta de usuario, Consulta de saldo, Cambio de contraseña).**

**Las transacciones deben cumplir con el siguiente diseño. Algunas transacciones monetarias pueden ser RECHAZABLES (esto significa que se pueden rechazar en el caso que se detecte un alto riesgo de fraude) y ALERTABLES (significa que se pueden reportar para que un analista las revise, si se detecta un riesgo de fraude medio). Ahora, ambos tipos de transacciones (las monetarias y las no monetarias) deben ser MONITOREABLES, esto es que se deben poder monitorear para el caso que así se desee.**

**Para la primera versión del software se determina que las transacciones Extracción, Transferencia y PagosQR deben ser RECHAZABLES mientras que Extracción, Transferencia, PagosQR y PagosServicios deben ser ALERTABLES.**

**Por otro lado, el sistema debe tener la posibilidad de informar datos que hayan formado parte de un evento de fraude. Es bajo esa condición que aparece el concepto de DENUNCIABLE, y los elementos denunciables pueden ser clientes, a partir de un número de CUIT. Dispositivos, a partir de una dirección IP o IMEI. Los elementos denunciables son aquellos que participaron de un evento delictivo.**

**La operatoria del sistema, consiste obviamente en monitorear las transacciones de las entidades bancarias, y cuando la transacción que se está monitoreando es RECHAZABLE, se le debe asignar un SCORE (puntaje) que determine el riesgo de la misma. Este SCORE se determina en función de las características de la transacción. Dicho SCORE va de 0 a 100, siendo 0 el riego de fraude mas bajo. Si el SCORE da mas de 80, la transacción se debe lanzar un evento de FRAUDE que bloquee la transacción. Si el SCORE da entre 60 y 79, se debe generar un caso de análisis para estudiar la transacción, pero la misma no es bloqueada y debe seguir su curso. Luego un analista debe actualizar dicho caso de análisis informando si la transacción se trató de un FRAUDE o no. Para el cálculo del SCORE se deben considerar los siguientes aspectos:**

1. **Si el CUIT del cliente de la transacción formó parte de un fraude se suma 80 puntos al SCORE.**
2. **Si la dirección IP o IMEI del dispositivo estuvo involucrado en un evento de FRAUDE, se suma 80 puntos al SCORE.**
3. **Si el cliente hizo un cambio de contraseña inmediatamente antes (última operación realizada) a realizar una transacción RECHAZABLE, suma 20 puntos al SCORE.**
4. **Si el monto de la TRANSFERENCIA coincide con el total de fondos del cliente (su saldo quedaría en 0) suma 40 puntos al SCORE.**
5. **Si el cliente está operando desde un dispositivo que nunca utilizó previamente, suma 20 puntos al SCORE.**

**Al momento de monitorear una transacción RECHAZABLE, el sistema debe calcular el SCORE de esta. Si el SCORE se encuentra entre 60 y 79, la transacción se debe permitir, pero se debe registrar un nuevo caso de análisis para que un analista lo revise y determine si existió FRAUDE o no. Si el analista marca el caso como FRAUDE, se deben registrar los datos involucrados en el fraude dentro de las listas negras en cuestión (Cliente y dispositivo).**

**Condiciones de aprobación.**

* **Se considerará desaprobado todo proyecto que:**
  1. **no compile, o,**
  2. **no se identifiquen los atributos necesarios para resolver lo solicitado, o,**
  3. **los tipos de datos elegidos para los atributos no sean los adecuados, o,**
  4. **no cumpla con las condiciones mínimas descriptas a continuación.**
* **Se considerará aprobado, aquel examen que ejecute de manera correcta al menos el 70% basado en la ponderación de los siguientes test:**

|  |  |
| --- | --- |
| **queSePuedaCrearUnCliente** | **5%** |
| **queSePuedaCrearUnDispositivo** | **5%** |
| **queSePuedaMonitorearUnaExtraccion** | **5%** |
| **queSePuedaMonitorearUnaTransferencia** | **5%** |
| **queSePuedaMonitorearUnPagoConQR** | **5%** |
| **queSePuedaMonitorearUnPagoDeServicio** | **5%** |
| **queSePuedaMonitorearUnAltaDeUsuario** | **5%** |
| **queSePuedaMonitorearUnCambioDeContraseña** | **5%** |
| **queElScoreDeUnaTransaccionRechazableSinAntecedentesDeCero** | **10%** |
| **queUnaTransaccionAlertablePuedaSerMarcadaComoFraudulenta** | **10%** |
| **queElScoreDeUnaTransaccionRechazableConNuevoDispositivoDe20Puntos** | **10%** |
| **QueUnaTransaccionDeMasDe60PuntosYMenosDeOchentaSeaAlertadaPeroAprobada** | **15%** |
| **queUnaTransaccionDeMasDe80PuntosLanceLaExcepcionFraudeException** | **15%** |
|  | **100%** |