December Labs

Prueba técnica

Descripción

Implementar una API en Node JS, PHP, C# ó Ruby, que provea endpoints para consultar y ejecutar transacciones monetarias en base a ciertas reglas de negocio establecidas.

API endpoints (requeridos)

Método	Nombre	Parámetros/Body	Descripción
GET	/transactions	● From (Date, opcional) ● To (Date, opcional) ● SourceAccountID (number, opcional)	Obtiene las transacciones para el usuario logueado, opcionalmente se puede filtrar por un rango de fechas e incluso por identificador de cuenta origen.
POST	/transfer	Body: { accountFrom: <accountId>, accountTo: <accountId>, amount: <?>, date: <timestamp>, description: <string> }	Realiza la transferencia entre una cuenta de origen y una cuenta destino, las cuentas pueden ser del mismo usuario o no.

Base de datos (a elección)

Modelo

Un usuario puede tener múltiples cuentas.

Una cuenta tiene una divisa (moneda) y un capital (saldo).
Una transacción guarda la cuenta origen y destino, el monto, la fecha/hora y una descripción aportada por el usuario al momento de realizarla.
Las divisas serán tratadas con una tabla auxiliar donde al menos deberán existir : Pesos Uruguayos, Dólares Americanos y Euros.

Para obtener las últimas cotizaciones y poder convertir de una divisa a otra, se deberá utilizar la API https://apilayer.com/marketplace/fixer-api (tiene una suscripción free).
Asumir que este servicio tiene un coste por llamado.

Asumir que el volumen de transacciones en la Base de Datos está en el orden del millón.

Cuentas

Las cuentas pueden ser de distintas divisas. Una transacción entre cuentas de distinta divisa, asume que la suma “amount” se encuentra en la divisa origen y que hay que hacer la conversión a la divisa destino.

Si la transacción se realiza para un tercero (es decir que la cuenta de destino no pertenezca al usuario logueado) entonces se aplicará una comisión por transacción del 1% del monto transferido, la cual será aplicada sobre la cuenta origen.

Identificación de usuario

Dadas las limitaciones de tiempo, se deberá implementar un middleware que verifique que para una petición el usuario que está realizando el request es el mismo que está logueado (puede ser "hard-coded").

Fuera del Scope

- UI
- Tests