"Mi primer desafío bancario para Consultoría Global"

Problema: se hicieron movimientos que descontaron montos del saldo de los clientes, pudiendo haberse generado valores negativos.

Consignas:

1: procesar los movimientos realizados y generar como salida un nuevo archivo que contenga aquellas cuentas que tienen saldo negativo.

2: Mostrar el archivo e informar cuántas cuentas quedaron con saldo menor a -$10.000.

3: Mostrar la salida con pic de edición.

Archivos de entrada:

- Cuentas.dat -> contiene las cuentas de los clientes:

. NroCue, N, 8

. Titular, A, 25

. Saldo, N, 6.2

- Movimientos.dat -> contiene las operaciones realizadas en cada cuenta:

. NroCue, N, 8

. Movimiento, C, 1 ("D" o "E")

. Monto, N, 6.2

Archivo de salida:

- Errores.dat -> contiene el resultado de la ejecución del proceso:

. NroCue, N, 8

. Titular, A, 25

. CantExtr, N, 8

. MontoTot, N, 6.2

. SaldoFinal, N, 6.2

En los 3 casos, asumo que el formato de los archivos es de texto, con tabulaciones de separación.

Presentación de la solución:

En un principio me incliné a abordar el tema con un único proceso en Node js que, asumiendo que existen los dos archivos de entrada en la misma carpeta, los recorra en 4 pasos básicos:

1 - Levantar el archivo de cuentas y generarle un índice, siendo que los registros vienen desordenados.

2 - Crear un array en memoria que irá guardando las cuentas y acumulando la cantidad de extracciones, montos y saldos finales.

3 - Barrer el archivo de movimientos. Por cada registro, obtener la información relacionada (nro. de cuenta, titular, saldo inicial) del archivo de cuentas e ir acumulando en el array los valores que cambien.

4 - Generar el archivo de salida recorriendo el array generado, pero incluyendo solamente aquellas cuentas que quedaron con saldo negativo. Durante el recorrido ir contando aquellas cuentas con saldo menor a -$10.000.

Finalmente descarté esa idea, ya que me pareció demasiado rápida de resolver. Siendo que me dieron 1 semana de plazo, decidí implementar otro abordaje, el cual presento. Opté por una solución frontend (React) + backend (Node js), en la que se permite al usuario seleccionar los dos archivos de entrada e iniciar a petición el proceso. Luego, la lógica que corre automáticamente es la siguiente:

1 – El frontend -> envía al backend (por HTTP, ruta /accounts) el archivo de cuentas. Este lo recibe, lo recorre y agrega sus registros a una tabla de base de datos PostgreSQL, generando un índice primario por número de cuenta. Este índice facilitará las búsquedas en el siguiente paso.

2 – El frontend -> envía al backend (por HTTP, por ruta /movim) el archivo de movimientos. Este lo recibe y lo recorre de la siguiente manera:

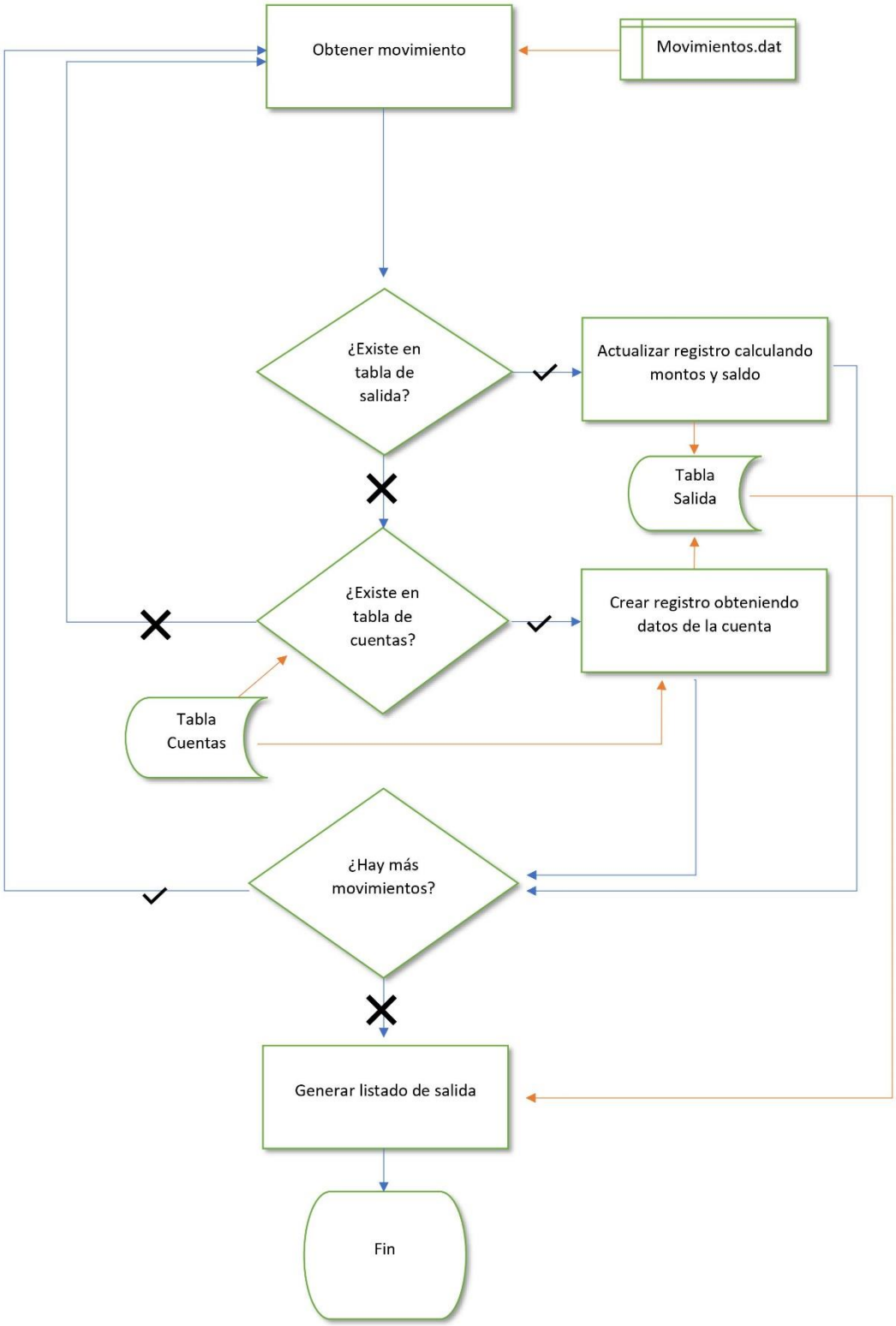
2.1 - Si existe la cuenta en la tabla de salida (de la misma base de datos PostgreSQL), se toman los valores del registro y se actualizan los campos de la tabla de salida cantidad de extracciones, monto total y saldo final. Los cálculos varían dependiendo de si se trata de un depósito o extracción.

2.2 - Si no existe la cuenta en la tabla de salida, se agrega el nuevo registro en la misma, obteniéndose los datos relacionados desde la tabla de cuentas (nro. de cuenta, titular, saldo inicial) y se registran los datos del movimiento: cantidad de extracciones, monto total y saldo final. Los cálculos varían dependiendo de si se trata de un depósito o extracción.

3 - Finalizada la recorrida se contabiliza el total de registros de la tabla de salida cuyo saldo es menor a -$10.000.

4 – El frontend -> envía al backend (por ruta /results) la petición del archivo de salida. El backend recibe el pedido y realiza un proceso de descarga, previo filtrado por monto negativo,

que finalizará con el archivo Errores.dat en la carpeta de descargas del equipo. Los montos que superen los 6.2 dígitos quedarán truncados.



Se adjunta más documentación, incluyendo archivos de prueba e instrucciones de instalación y ejecución, en el README.md del proyecto, ubicado en el repositorio PauloDamianVinci/listador (github.com)

Quedo a disposición en caso de que tengan dificultades para ejecutarlo. Se podría organizar una videollamada de ejecución en vivo.

Con este proyecto se completa la consigna 1, y también la parte de la 2 en donde se solicita mostrar los contadores de movimientos menores a -$10.000.

Para lo restante, hice consultas con colegas y en internet, ya que desconozco la terminología "salida pic", y llegamos a concluir parcialmente que podrían estar asociadas a necesidades de entornos no gráficos, en donde se requieren procesos adicionales para mostrar e imprimir la salida. Esto no es necesario en un entorno visual, ya que la localización e impresión del archivo resultante es una tarea trivial.

Se entrega un deploy del proyecto en https://listador-dlil.vercel.app. Tener en cuenta la privacidad de los datos de entrada antes de ejecutar en este sitio público. En la subcarpeta */Files* del repositorio, hay versiones de ejemplo para utilizar.

Por motivos de seguridad decidí no implementar otro método de procesamiento más veloz de los archivos de entrada, como ser el uso de Cloudinary, donde la subida anticipada a la nube y posterior descarga por parte del backend mejoraría los tiempos previos a la recorrida de movimientos, aunque aumentaría los riesgos también.

Los conocimientos aplicados en este proyecto fueron obtenidos en el curso Fullstack developer, el cual estoy cercano a finalizar en www.soyhenry.com. También recurrí a un colega de confianza para consultas técnicas sobre la transferencia de archivos vía http, ya que fue mi primera experiencia con esa funcionalidad puntual. Hice más de un intento consultando sobre el tema a ChatGPT, pero sólo me produjo recomendaciones parciales, desactualizadas, que no me sirvieron de mucho. Otra fuente consultada para el proyecto fue www.stackoverflow.com.

Agradecimiento especial a mi colega Eliana Santa Cruz por el soporte desde España para el deploy.