

Proyecto Final: Simulador de un "Monedero Virtual con sistema de Puntos"

Objetivo: Desarrollar una aplicación que simule un monedero virtual avanzado, permitiendo realizar transacciones entre clientes como depósitos, retiros, transferencias, consultas, reportes, gastos, además de incorporar funcionalidades complejas como transacciones programadas, análisis de patrones de gasto, sistema de puntos por transacciones y optimización de algoritmos según el tipo de estructura de datos utilizada.

Descripción:

El sistema debe permitir la gestión de cuentas virtuales de los clientes, donde cada cliente puede realizar diversas transacciones y consultar distintos tipos de información relacionada con sus cuentas y gastos. Además de las funcionalidades básicas, el sistema debe incluir un componente de gamificación con un sistema de puntos, incentivando a los clientes a usar el monedero virtual.

Funcionalidades Básicas:

1. **Depósitos:** Los clientes pueden agregar dinero a su cuenta.
2. **Retiros:** Los clientes pueden retirar dinero de su cuenta, siempre que tengan saldo suficiente.
3. **Transferencias:** Los clientes pueden transferir dinero entre diferentes cuentas de otros clientes.
4. **Consultas de Saldo:** Los clientes pueden consultar su saldo actual.
5. **Consultas de Historial de Transacciones:** Los clientes pueden ver todas las transacciones realizadas.

Funcionalidades Avanzadas:

1. **Sistema de Puntos por Transacciones:**
 - Cada transacción realizada por un cliente (depósito, retiro, transferencia) genera una cantidad específica de puntos dependiendo del tipo y el monto de la transacción. Por ejemplo:
 - **Depósitos:** 1 punto por cada 100 unidades de moneda depositadas.
 - **Retiros:** 2 puntos por cada 100 unidades retiradas.
 - **Transferencias:** 3 puntos por cada 100 unidades transferidas.
 - Los puntos acumulados pueden ser canjeados por beneficios como descuentos en transacciones futuras o privilegios adicionales en el sistema.
 - Implementa una estructura eficiente como **árboles binarios de búsqueda** para gestionar los puntos acumulados por cada cliente.
2. **Rangos de Clientes:**
 - Con base en los puntos acumulados, los clientes serán clasificados en diferentes rangos (por ejemplo, **Bronce, Plata, Oro, Platino**). Cada rango puede tener beneficios adicionales, como tarifas reducidas para transferencias o bonos de puntos adicionales por ciertas transacciones.
 - Usa **árboles balanceados** para mantener actualizados los rangos de los clientes en función de sus puntos acumulados.
3. **Transacciones Programadas:**
 - Los clientes pueden programar transacciones recurrentes (depósitos automáticos, transferencias periódicas).

- Usa **colas de prioridad** para procesar estas transacciones de manera ordenada y eficiente según la fecha de ejecución.
- 4. **Análisis de Patrones de Gasto:**
 - Genera informes avanzados sobre los patrones de gasto de los clientes. Usa **grafos dirigidos** para analizar las relaciones entre categorías de gasto y transferencias entre clientes.
- 5. **Alertas y Notificaciones:**
 - Envía notificaciones automáticas al cliente, como alertas de saldo bajo o recordatorios de transacciones programadas.
 - Usa **listas circulares** para gestionar las notificaciones de manera eficiente.
- 6. **Multimonedero:**
 - Los clientes pueden manejar múltiples monederos dentro de su cuenta (uno para ahorros, otro para gastos diarios).
 - Utiliza **grafos dirigidos** para representar las relaciones entre los monederos y las transacciones realizadas entre ellos.
- 7. **Verificación de Integridad de las Transacciones:**
 - Garantiza la validez de todas las transacciones realizadas utilizando algoritmos recursivos.
- 8. **Interfaz de Usuario Avanzada:**
 - Proporciona una interfaz para realizar transacciones, consultar puntos acumulados, ver rangos, y canjear beneficios.

Requisitos del Sistema de Puntos:

1. **Acumulación de Puntos:**
 - Cada cliente tiene un registro de los puntos acumulados y un historial de transacciones relacionadas con estos puntos.
 - Los puntos deben ser recalculados dinámicamente en caso de que se revierta alguna transacción.
2. **Canje de Puntos:**
 - Los clientes pueden canjear puntos por beneficios específicos. Ejemplo:
 - **100 puntos:** Reducción del 10% en la comisión por transferencias.
 - **500 puntos:** Un mes sin cargos por retiros.
 - **1000 puntos:** Bono de saldo de 50 unidades en el monedero.
3. **Clasificación de Rangos:**
 - Implementar una función que calcule el rango de cada cliente con base en sus puntos acumulados:
 - **Bronce:** 0–500 puntos.
 - **Plata:** 501–1000 puntos.
 - **Oro:** 1001–5000 puntos.
 - **Platino:** Más de 5000 puntos.

Requisitos del Proyecto:

- **Estructuras de Datos:**
 - **Listas:** Para gestionar las transacciones por cliente.
 - **Pilas:** Para manejar transacciones reversibles.
 - **Colas:** Para gestionar las transacciones programadas y las notificaciones.

- **Árboles:** Para almacenar puntos acumulados y rangos de los clientes.
- **Grafos:** Para analizar transferencias entre clientes y patrones de gasto.
- **Tablas Hash:** Para acceder rápidamente a los datos de los puntos acumulados por cliente.
- **Optimización de Algoritmos:**
 - Usa algoritmos eficientes para calcular y actualizar puntos, manejar rangos, y generar reportes.

Informe Final: Incluye un análisis técnico detallado de cómo se implementaron las estructuras de datos y los algoritmos optimizados, además de ejemplos prácticos del sistema de puntos y su impacto en el uso del monedero virtual.