

INFORME DEL MÉTODO DE LA INGENIERÍA

IDENTIFICACIÓN DEL PROBLEMA

Identificación de necesidades

- Tener un registro en tiempo real de dos filas, una para clientes con prioridad y otra para clientes generales.
- Cargar la información de la base de datos al programa.
- Buscar a un cliente dentro de la base de datos.
- Registrar a un nuevo cliente dentro de una fila.
- Añadir nuevos clientes a la base de datos
- Atender a los clientes y permitirles realizar un pago de tarjeta de crédito, cancelar su cuenta o realizar una consignación en su cuenta de ahorros.
- Deshacer cambios hechos para corregir errores humanos

Definición del problema

Un gran banco necesita un software que modele el funcionamiento de una de sus sedes principales para solventar las necesidades de sus clientes. Esto involucra todos los comportamientos que se presentan en un banco: Manejo de filas, visualización de bases de datos, operaciones financieras, etc. En específico, desean el manejo de dos filas, una para clientes generales y otra para clientes con prioridad. Para esto, se requiere poder registrar clientes a las filas, obtener la información del primer usuario en fila y retirarlo de la fila una vez atendido. Para las operaciones financieras se requiere poder atender consignaciones y retiros, cancelar cuentas y pagar tarjetas. Hay que tener en cuenta que los usuarios activos se guardan en una base de datos y los que son retirados en otra base distinta. Aparte, se requiere presentar a los clientes presentes dentro del banco en una tabla y poder organizarlos en base a 4 parámetros, y deshacer un cambio en el programa, por si se cometen errores en este.

RECOPILACIÓN DE INFORMACIÓN

Definiciones

Base de datos:

Una base de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido; una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. Actualmente, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital, siendo este un componente electrónico, por tanto se ha desarrollado y se ofrece un amplio rango de soluciones al problema del almacenamiento de datos.

Cuenta bancaria:

Una cuenta bancaria es un contrato financiero con una entidad bancaria en virtud del cual, se registran el balance y los subsiguientes movimientos de dinero del cliente.

Cuenta de ahorros:

Una cuenta de ahorro es un depósito ordinario a la vista (producto pasivo), en la que los fondos depositados por la cuenta, tienen disponibilidad inmediata y generan cierta rentabilidad durante un periodo determinado según el monto ahorrado.

Tarjeta de crédito:

La tarjeta de crédito es un instrumento material de identificación, que puede ser una tarjeta de plástico con una banda magnética, un microchip y un número en relieve. Es emitida por un banco o entidad financiera que autoriza a la persona a cuyo favor es emitida a utilizarla como medio de pago en los negocios adheridos al sistema, mediante su firma y la exhibición de la tarjeta. Es otra modalidad de financiación; por lo tanto, el usuario debe asumir la obligación de devolver el importe dispuesto y de pagar los intereses, comisiones bancarias y gastos

Requerimientos funciones

1. Registrar un cliente al momento de ingresar al banco: Se registra el nombre y la cédula de un cliente al momento en que ingresa al banco. Los clientes registrados serán acomodados en dos filas, una que es para clientes prioritarios y otra para clientes no prioritarios (personas con diferentes prioridades). Debido a que la cédula es única, no pueden existir dos clientes con la misma cédula haciendo fila al mismo tiempo.
2. Buscar un cliente: Se busca la información del cliente que esté al frente de la fila.
3. Realizar Retiro: El usuario podrá retirar un monto del dinero de su cuenta de ahorros, el usuario no podrá retirar un monto mayor al que se encuentra en su cuenta de ahorros.
4. Realizar una consignación en la cuenta de ahorros de un cliente: Se agrega un monto específico de dinero a la cuenta de ahorros de un cliente. La cantidad de dinero debe ser mayor a 50 pesos.
5. Cancelación de cuenta: Se cancela la cuenta y borra sus datos de la base de datos de clientes, así mismo estos datos pasan a otra base en la que se encuentran los que han desertado del banco, y se guardará tanto la fecha como el motivo de cancelación.
6. Realizar un pago de tarjeta de crédito: Se realiza el pago del crédito utilizado de una tarjeta de crédito. El pago puede ser realizado en efectivo, o extraerse el monto de la cuenta de ahorros. Una vez hecho el pago, se reduce la cantidad de crédito utilizado. No se puede pagar una cantidad mayor al crédito utilizado.
7. Mostrar una tabla con la información de todos los clientes presentes en la sede bancaria: se muestra una tabla con toda la información de todos los clientes

presentes en la sede bancaria, los cuales se podrán mostrar ordenadamente en base a cuatro aspectos.

- a. Cedula.
 - b. Nombre.
 - c. Tiempo de vinculación del cliente.
 - d. Monto en cuenta bancaria.
8. Deshacer una acción realizada en el sistema. Se podrán deshacer todas las acciones realizadas en el momento en el que se usa el programa. Una vez se cierra el programa y se vuelve a ejecutar, no se podrán deshacer las acciones realizadas previas a la ejecución.

BÚSQUEDA DE SOLUCIONES CREATIVAS

Soluciones para cargar la base de datos al programa

Solución 1: Hacer un array del tamaño necesario para cargar todos los usuarios y guardarlos en este desde un csv.

Solución 2: Hacer una linkedlist e ir guardando los usuarios hasta que se carguen todos en la lista desde un csv.

Solución 3: Implementar un árbol binario de búsqueda en el cual guardamos todos los usuarios, los cuales serán cargados desde un csv.

Solución 4: Hacer uso de una queue para cargar todos los usuarios desde un csv.

Solución 5: Hacer uso de un stack para cargar todos los usuarios desde un csv.

Solución 6: Implementar una hash table y guardar todos los usuarios en esta desde un csv.

Solución 7: Hacer una lista doblemente enlazada e ir guardando los usuarios hasta que se carguen todos en la lista desde un csv.

Soluciones para el uso de estructuras de datos

Solución 1: Hacer uso de una queue para la fila general y una priority queue para la fila prioritaria, un stack para la opción de undo y dos hashtables para la base de datos de clientes activos y la base de datos de clientes retirados.

Solución 2: Hacer uso de dos arreglos para las filas de espera, un stack para la opción de undo y dos hashtables para la base de datos de clientes activos y la base de datos de clientes retirados.

Solución 3: Hacer uso de dos listas enlazadas para las filas de espera, una queue para la opción de undo y dos stacks para la base de datos de clientes activos y la base de datos de clientes retirados.

Solución 4: Hacer uso de dos queues para las filas de espera, un array para la opción de undo y dos listas enlazadas para la base de datos de clientes activos y la base de datos de clientes retirados.

Solución 5: Hacer uso de dos stacks para las filas de espera, un stack para la opción de undo y dos queues para la base de datos de clientes activos y la base de datos de clientes retirados.

Solución 6: Hacer uso de dos queues para las filas de espera, una queue para la opción de undo y dos arreglos para la base de datos de clientes activos y la base de datos de clientes retirados.

Solución 7: Hacer uso de dos queues para las filas de espera, un stack para la opción de undo y dos árboles binarios de búsqueda para la base de datos de clientes activos y la base de datos de clientes retirados.

Solución 8: Hacer uso de dos listas enlazadas para las filas de espera, un stack para la opción de undo y dos árboles binarios de búsqueda para la base de datos de clientes activos y la base de datos de clientes retirados.

TRANSICIÓN DE LA FORMULACIÓN DE IDEAS DE DISEÑOS PRELIMINARES

Descripción de soluciones para cargar la base de datos al programa

Descripción solución 1: Hacer un array del tamaño necesario para cargar todos los usuarios en este desde un csv por medio de un bucle que inserte cada usuario en un espacio del array.

Descripción solución 2: Hacer una linkedlist e ir guardando los usuarios por medio de un bucle que utilice el método de agregar hasta que se carguen todos en la lista desde un csv.

Descripción solución 3: Implementar un árbol binario de búsqueda en el cual guardamos todos los usuarios, los cuales serán cargados desde un csv por medio de un bucle que utilice el método de agregar, siguiendo el parámetro número de cédula y hasta que se carguen todos al árbol.

Descripción solución 4: Hacer uso de una queue para cargar todos los usuarios desde un csv por medio de un bucle que utilice el método de agregar, hasta que se carguen todos al queue.

Descripción solución 5: Hacer uso de un stack para cargar todos los usuarios desde un csv por medio de un bucle que utilice el método de agregar, hasta que se carguen todos al stack.

Descripción solución 6: Hacer uso de una hashtable para cargar todos los usuarios desde un csv por medio de un bucle que utilice el método de agregar por medio de direccionamiento directo, hasta que se carguen todos a la hash table.

Descripción solución 7: Hacer uso de una lista doblemente enlazada para cargar todos los usuarios desde un csv por medio de un bucle que utilice el método de agregar, hasta que se carguen todos a la lista.

Descripción de soluciones para el uso de estructuras de datos

Descripción solución 1: La queue utilizada para la fila de clientes generales permiten registrar clientes, insertándolos en la última posición de la queue. También permiten eliminarlos, removiendo al primero de la queue una vez atendido. Por último permite obtener la información del primero de la queue. Por otro lado, la priority queue añade clientes prioritarios, insertándose en el último espacio disponible, y organizándose en base a su nivel de prioridad, poniendo al más prioritario en el último espacio. El stack utilizado para el undo guarda el estado del cliente antes de que se realice el cambio deseado, insertándose como el tope del stack, y cuando se pida, se restablecerá el estado anterior más reciente a la base de datos, extrayendo el tope del stack. Las bases de datos de clientes activos y clientes retirados usarán una hashtable que contendrá una tupla clave, cliente, usando exploración lineal para agregar nuevos clientes.

Descripción solución 2: Los dos arreglos utilizados para filas permiten registrar clientes, insertándolos en la primera posición del arreglo, y si hay previamente clientes, se desplazan un espacio hacia la derecha. También permiten eliminarlos, removiendo al cliente en la última posición. El arreglo para clientes prioritarios además los organiza de menor a mayor prioridad cada vez que un cliente entra. Por último permite obtener la información del último cliente del arreglo. El stack utilizado para el undo guarda el estado del cliente antes de que se realice el cambio deseado, insertándose como el tope del stack, y cuando se pida, se restablecerá el estado anterior más reciente a la base de datos, extrayendo el tope del stack. La base de datos usará un hashtable que contendrá una tupla clave, cliente, usando exploración lineal para agregar nuevos clientes.

Descripción solución 3: Las dos listas enlazadas utilizadas para filas permiten registrar clientes, insertándolos en la primera posición del arreglo, y si hay previamente clientes, se desplazan un espacio hacia la derecha. También permiten eliminarlos, removiendo al último de la lista. Por último permite obtener la información del último cliente de la lista. La queue utilizada para el undo guarda el estado del cliente antes de que se realice el cambio deseado, insertándose en la última posición de la queue. y cuando se pida, se restablecerá el estado anterior más reciente a la base de datos, insertando todos los estados en otra queue menos el más recientemente agregado. La base de datos usará un stack que contendrá clientes.

Descripción solución 4: Las dos queues utilizadas para filas permiten registrar clientes, insertándolos en la última posición de la queue. También permiten eliminarlos, removiendo al primero de la queue una vez atendido. Por último permite obtener la información del primero de la queue. El array utilizado para el undo guarda el estado del cliente antes de que se realice el cambio deseado, insertándose a la última posición disponible del arreglo, y cuando se pida, se restablecerá el estado anterior más reciente a la base de datos, retirando el último estado en el arreglo. La base de datos usará una lista enlazada que contendrá clientes.

Descripción solución 5: Los dos stacks utilizados para filas permiten registrar clientes, insertándolos en el tope del stack. También permiten eliminarlos, insertando todos los clientes en un nuevo stack, menos el ultimo, que sera el eliminado, y se vuelven a insertar de nuevo en otro stack para tener el orden original. Por último permite obtener la información del primero de la fila, insertando todos los clientes en un nuevo stack, obteniendo la información y reinsertando todos los clientes en un nuevo stack para obtener el orden anterior. El stack utilizado para el undo guarda el estado del cliente antes de que se realice el cambio deseado, insertándose como el tope del stack, y cuando se pida, se restablecerá el estado anterior más reciente a la base de datos, extrayendo el tope del stack. La base de datos usará una queue que contendrá clientes. La base de datos usará una lista enlazada que contendrá clientes.

Descripción solución 6: Las dos queues utilizadas para filas permiten registrar clientes, insertándolos en la última posición de la queue. También permiten eliminarlos, removiendo al primero de la queue una vez atendido. Por último permite obtener la información del primero de la queue. La queue utilizada para el undo guarda el estado del cliente antes de que se realice el cambio deseado, insertándose en la última posición de la queue. y cuando se pida, se restablecerá el estado anterior más reciente a la base de datos, insertando todos los estados en otra queue menos el más recientemente agregado. La base de datos usará un arreglo que contendrá clientes.

Descripción solución 7: Las dos queues utilizadas para filas permiten registrar clientes, insertándolos en la última posición de la queue. También permiten eliminarlos, removiendo al primero de la queue una vez atendido. Por último permite obtener la información del primero de la queue. El stack utilizado para el undo guarda el estado del cliente antes de que se realice el cambio deseado, insertándose como el tope del stack, y cuando se pida, se restablecerá el estado anterior más reciente a la base de datos, extrayendo el tope del stack. La base de datos usará un árbol binario que contendrá clientes, guardados por su cedula de ciudadanía.

Descripción solución 8: Las dos listas enlazadas utilizadas para filas permiten registrar clientes, insertándolos en la primera posición del arreglo, y si hay previamente clientes, se desplazan un espacio hacia la derecha. También permiten eliminarlos, removiendo al último de la lista. Por último permite obtener la información del último cliente de la lista.

Hacer uso de dos listas enlazadas, una para cada una de las filas de espera, un stack para la opción de undo y una árbol binario de búsqueda para la base de datos. El stack utilizado para el undo guarda el estado del cliente antes de que se realice el cambio deseado, insertándose como el tope del stack, y cuando se pida, se restablecerá el estado anterior más reciente a la base de datos, extrayendo el tope del stack. La base de datos usará un árbol binario que contendrá clientes, guardados por su cedula de ciudadanía.

Evaluación y solución de la posible solución

Solución	Conocimiento sobre el tema	Necesidad o valor para el cliente	Facilidad de desarrollo	Flexibilidad	Total	Aprobado/No aprobado
Soluciones para cargar la base de datos al programa						
Alternativa 2	5	3	5	2	13	NO APROBADO
Alternativa 3	5	4	4	4	17	NO APROBADO
Alternativa 6	4	5	3	4	16	APROBADO
Soluciones para el uso de estructuras de datos						
Alternativa 1	4	4	3	4	15	APROBADO
Alternativa 7	5	3	4	3	15	NO APROBADO
Alternativa 8	5	3	5	3	16	NO APROBADO

Tras evaluar las 3 alternativas más prometedoras para la solución del problema, se ha decidido utilizar la alternativa 6 para cargar la base de datos y la alternativa 1 para el uso de las estructuras de datos.

Se eligió la alternativa 6 para la base de datos por su eficiencia en la inserción y remoción a la hora de lidiar con los clientes, que es lo que principalmente nos interesa en este proyecto.

Por otro lado, se eligió la alternativa 1 para el uso de estructuras de datos por varios motivos:

- La queue y priority queue son las mejores estructuras para las filas, pues de por sí imitan precisamente el comportamiento de una fila en la vida real, lo que hace que su manejo sea más intuitivo, y por ende, fácil.
- El stack como estructura de la función undo es la mejor ya que es la que permite el mejor manejo sobre los cambios a retornar. Otras estructuras simplemente tienen muchos pasos extra que pueden ser evitados con el stack.
- Las hash tables como bases de datos para clientes activos y retirados es la mejor por su eficiente forma de insertar y eliminar clientes en ellas.

DIAGRAMA (DE CLASES)

Última página

REFERENCIAS

- https://es.wikipedia.org/wiki/Base_de_datos
- https://es.wikipedia.org/wiki/Cuenta_bancaria
- https://es.wikipedia.org/wiki/Cuenta_de_ahorro
- https://es.wikipedia.org/wiki/Tarjeta_de_cr%C3%A9dito

