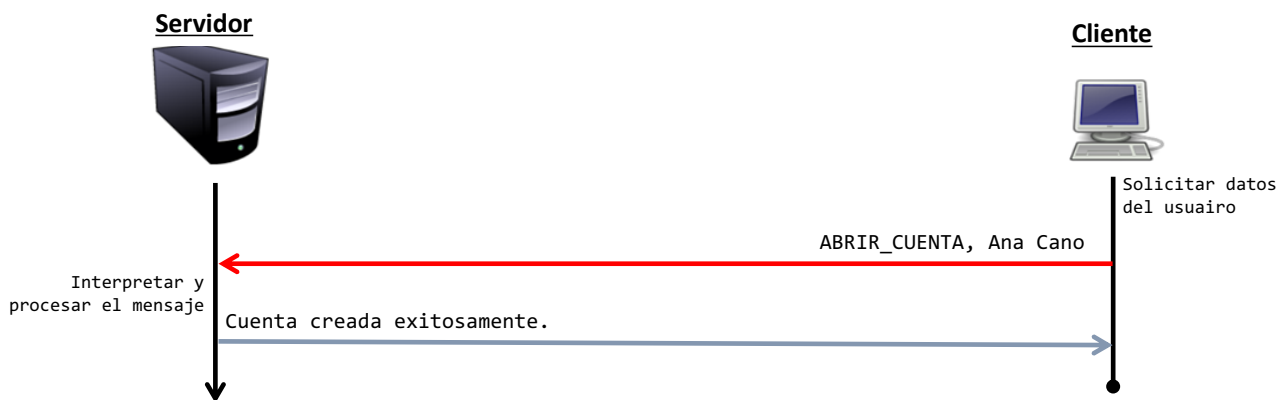


Infraestructura de Comunicaciones

Trabajo #2: Aplicación Cliente-Servidor para la simulación de las operaciones de un banco usando sockets TCP en Java

1. Trabajo a realizar

El trabajo consiste en el desarrollo de una aplicación que simule el comportamiento de algunas de las operaciones bancarias más comunes. Es una aplicación cliente – servidor en la cual los usuarios utilizan el programa cliente para interactuar con el banco, el cual usa el programa servidor. El programa cliente se comunica con el programa servidor a través de mensajes enviados por la red, utilizando sockets TCP en Java. Para almacenar los datos de los usuarios del banco, vamos a utilizar un objeto de tipo HashMap. Cada transacción realizada deberá llevar la fecha y hora en la que se realiza.



Servicios

1. **Abrir cuenta.** Es una cuenta de ahorros con los siguientes datos: (1) Número de la cuenta: un valor entero positivo que inicia en 0 (asignado por el servidor), (2) Nombre del usuario, (3) Saldo. Una cuenta se abre con saldo 0. El programa cliente enviará un mensaje con el siguiente contenido: ABRIR_CUENTA, nombre apellido. Por restricción, no se permitirán nombres repetidos. El servidor debe dar una respuesta de transacción exitosa o en error.
2. **Crear un bolsillo.** Es una cuenta dependiente de una cuenta de ahorros, en la cual el saldo es independiente. El número de esta cuenta es el mismo número de la cuenta de ahorros, a la que se le agrega la letra 'b'. Una cuenta bolsillo se abre con saldo 0. Cada usuario puede crear máximo un bolsillo. El programa cliente enviará un mensaje con el siguiente contenido: ABRIR_BOLSILLO, cuentaPrincipal. El servidor debe dar una respuesta de transacción exitosa o en error. En este caso, se debe especificar si el error es consecuencia de información insuficiente o de una cuenta no existente.
3. **Cancelar bolsillo.** Al cancelar un bolsillo, se traslada el saldo a la cuenta principal y esta cuenta se elimina de la estructura de datos. El usuario podría crear un nuevo bolsillo si lo desea. El programa cliente enviará un mensaje con el siguiente contenido: CANCELAR_BOLSILLO, cuentaBolsillo. El servidor debe dar una respuesta de

- transacción exitosa o en error. En este caso, se debe especificar si el error es consecuencia de información insuficiente, inconsistente, o de una cuenta no existente.
4. **Cancelar cuenta de ahorros.** Al cancelar una cuenta de ahorros, esta cuenta se elimina de la estructura de datos. Solo se puede cancelar una cuenta de ahorros que no esté asociada a un bolsillo y que su saldo sea 0. El usuario podría crear una nueva cuenta de ahorros si lo desea. El programa cliente enviará un mensaje con el siguiente contenido: CANCELAR_CUENTA, cuentaAhorros. El servidor debe dar una respuesta de transacción exitosa o en error. En este caso, se debe especificar si el error es consecuencia de información insuficiente, inconsistente, de una cuenta no existente, o si el saldo no es suficiente.
 5. **Depositar dinero en una cuenta.** El programa cliente enviará un mensaje con el siguiente contenido: DEPOSITAR, cuentaAhorros, valor. El banco deberá calcular el nuevo saldo de la cuenta. El servidor debe dar una respuesta de transacción exitosa o en error. En este caso, se debe especificar si el error es consecuencia de información insuficiente, inconsistente, o de una cuenta no existente.
 6. **Retirar dinero de una cuenta.** El programa cliente enviará un mensaje con el siguiente contenido: RETIRAR, cuentaAhorros, valor. El banco deberá calcular el nuevo saldo de la cuenta. Es necesario validar que el saldo sea suficiente para el retiro solicitado. El servidor debe dar una respuesta de transacción exitosa o en error. En este caso, se debe especificar si el error es consecuencia de información insuficiente, inconsistente, de una cuenta no existente, o si el saldo no es suficiente.
 7. **Trasladar dinero a un bolsillo.** El programa cliente enviará un mensaje con el siguiente contenido: TRASLADAR, cuentaAhorros, valor. El banco deberá calcular el nuevo saldo de la cuenta y el nuevo saldo del bolsillo. Es necesario validar que el saldo sea suficiente para el traslado solicitado. El servidor debe dar una respuesta de transacción exitosa o en error. En este caso, se debe especificar si el error es consecuencia de información insuficiente, inconsistente, de una cuenta no existente, o si el saldo no es suficiente.
 8. **Consultar saldo.** El programa cliente enviará un mensaje con el siguiente contenido: CONSULTAR, cuenta. En este caso, cuenta puede ser el número de una cuenta de ahorros o un bolsillo. El servidor debe dar una respuesta de transacción exitosa o en error. En este caso, se debe especificar si el error es consecuencia de información insuficiente, inconsistente o de una cuenta no existente.

Etapa 1: Modo texto (50%)

Aparece un menú y cada una de las opciones de los servicios ofrecidos por el banco. De acuerdo con la opción solicitada, el programa solicita por teclado los datos. El programa crea el mensaje y lo envía. Recibe la respuesta y la imprime.

Etapa 2: Modo gráfico (20%)

Desarrollar una interfaz de usuario que deberá tener un mecanismo para cada una de las transacciones a realizar, bien sea mediante menús o botones. En cada caso, debe ser posible ingresar los datos necesarios para crear el mensaje, enviarlo al servidor y recibir la respuesta.

Etapa 3: Carga automática (20%)

Carga automática de al menos 10 transacciones, de manera que se pueda probar el sistema a partir de la ejecución de esas transacciones. El programa cliente enviará un mensaje con el siguiente contenido: CARGA, seguido del nombre del archivo que almacena los mensajes de las transacciones que el cliente le enviará al servidor.

2. Entregables

(a) Evidencias de la ejecución

Realizar un informe amplio en el que se puedan ver **todos** los servicios ofrecidos por la aplicación, incluyendo capturas de pantalla y comentarios explicativos que dejen la evidencia del trabajo realizado. En este documento deben aparecer tanto la etapa en modo texto como la etapa en modo gráfico. Dentro de las evidencias se debe incluir el uso de la aplicación desde las redes caseras de los distintos integrantes del grupo mediante la aplicación ngrok. (10%)

(b) Código fuente del proyecto

Entregar el código fuente del proyecto comprimido en formato **.zip**. **Entregar el código fuente comprimido en otro formato, tal como .7z, .rar, .tar, o .tgz, da lugar a una disminución de la nota.**

3. Condiciones generales

- Grupos de mínimo de 3 y máximo de 4 estudiantes.
- Fecha de entrega: lunes 26 de julio de 2021.
- Forma de entrega: A través de Moodle.
- Asesoría: Correo electrónico y Cisco Webex.
- Demostración y sustentación: Según sea programado luego de la entrega. El profesor seleccionará uno o dos estudiantes dependiendo si el grupo es de tres o cuatro personas respectivamente. Las otras dos personas del grupo no pueden participar de la reunión de sustentación. La nota obtenida en la sustentación será la nota de todo el grupo.