



UNIVERSIDAD SIMÓN BOLÍVAR

Departamento de Computación y Tecnología de la Información

TRIMESTRE SEPTIEMBRE-DICIEMBRE 2016

**REDES DE COMPUTADORES II (CI 4835)**

### **ASIGNACIÓN N°1**

**Objetivo General:** Proveer al estudiante de un trabajo práctico de diseño y construcción de una aplicación final, que use el paradigma de programación Cliente – Servidor, para que este comprenda en general, el funcionamiento simple de aplicaciones y servicios en redes.

**Objetivos Específicos:** Al finalizar esta actividad el estudiante deberá estar en capacidad de:

- ✓ Comprender el uso y la programación de la Interfaz de Aplicaciones (API) Sockets de Berkeley.
- ✓ Aplicar los conceptos dados en clases para desarrollar el diseño e implementación de un protocolo de comunicación básico.

### **Planteamiento del Problema**

Un cajero automático es un computador personal (PC) conectado a un sistema central que trabaja en red. Tiene un puerto de comunicaciones multiprotocolo que le permite conectarse por módem (telefónico, de radio o celular), o vía satélite. Para entregar los recibos al cliente, el cajero utiliza una impresora (de 40 u 80 columnas) que trabaja a una velocidad de más de 400 caracteres por segundo.

El sistema está complementado con un lector de bandas magnéticas, un dispensador de billetes y un sensor que reconoce las diferentes denominaciones o documentos que maneja la máquina. Existen cajeros que funcionan únicamente como dispensadores de dinero (cash) y otros de funciones completas (full). Estos últimos son los que permiten realizar, además de retiros, otras transacciones como consignaciones, pagos y transferencias, entre otros.

A pesar de que en la actualidad se trabaja para dotar a los cajeros electrónicos con características multimedia y de videoconferencia, los fabricantes sostienen que este hecho podría demorar las transacciones, lo que iría en contra de la función de los equipos, que es agilizarlas.

Para muestra, un retiro, cuando un usuario entra al cajero e introduce la tarjeta, un lector registra la información de la banda magnética y determina si la transacción debe continuar o no. Si la tarjeta está muy deteriorada, lo más seguro es que el cajero no pueda leerla y cancele la operación.

La información viaja encriptada (en códigos ocultos para garantizar la seguridad del cliente) hasta la central, que devuelve un mensaje al cajero en el que se determina si la operación es válida.

Si la transacción finaliza con éxito, el cajero envía nuevamente un mensaje al computador central en el que confirma la operación. En caso de que la transmisión se interrumpa y, por ejemplo, la suma de un retiro sea debitada de la cuenta pero el cajero no entregue el dinero, el mensaje de éxito tampoco será enviado y la transacción queda sujeta a verificación.

Para confirmar si el reclamo de un cliente es cierto, el cajero cuenta con un sistema de auditoría interna en el disco duro (los modelos anteriores lo imprimían) que permite verificar las transacciones sin que el cajero salga de servicio.

La Universidad Simón Bolívar, dispone de una red de cajeros automáticos pertenecientes al Banco Simón Bolívar, éstos, se encuentran ubicados en la planta baja de los edificios: Básico I, Básico II y EME, permiten realizar transacciones diarias de depósito y retiro a los clientes. (Son cajeros tipo full)

Se quiere diseñar un sistema informático basado en el paradigma Cliente/Servidor que permita automatizar y controlar dicha red. Para ello se le ha contactado a usted como estudiante del curso de Redes de Computadores I, a fin que proponga e implemente una aplicación sencilla que incluya un Computador Central (CC) que va a controlar el sistema.

### **Funcionamiento del Sistema**

- El prototipo debe simular el funcionamiento del sistema las 24 horas del día, los 7 días a la semana (24/7)
- El “*Total Disponible*” para movimientos por cajero es de 80.000 bs
- Cuando se disponga de un “*Total Disponible*” con menos de 5.000 bs, deberá enviar un mensaje a la central comunicando la situación a fin de ser recargado el monto por dispositivo.
- Si el cajero cuenta con un “*Total Disponible*” igual o inferior a 5.000 bs, se informa al usuario de dicho evento. En la pantalla deberá mostrarse un mensaje de “*Dinero No Disponible*” dependiendo de los retiros que desee realizar.

- Al cierre del día el Computador Central debe tener un registro de todas las operaciones realizadas indicando fecha, hora, código del usuario, el evento (Depósito/Retiro) y Total Disponible.
- Cuando el CC recibe un mensaje de “*Retiro*” decrementa el contador de *Total Disponible* y responde al cajero adecuado con un mensaje indicando la hora y fecha de retiro, un código (número entero) que identifica al usuario.
- Con la fecha, hora, operación y código, el cajero automático imprime el ticket del usuario.
- El usuario podrá realizar máximo tres (3) retiros por un monto máximo de 3.000 bs cada uno.
- Cuando un usuario desea realizar un “*Depósito*”, introduce el código de usuario y monto del depósito.
- Cada vez que un usuario deposita en su cuenta, el CC debe incrementar el “*Total Disponible*” en cajero e imprimir ticket con la información (fecha, hora, operación y código)
- Considere que puede haber servicios simultáneos

### **Sintaxis de la Invocación de Comandos**

La sintaxis para la invocación de la ejecución del módulo que ofrece el servicio del CC deberá ajustarse a la siguiente indicación:

`bsb_svr -l <puerto_bsb_svr> -i <bitácora_entrada> -o <bitácora_salida>`

Dónde:

<puerto\_bsb\_svr> Es el número de puerto local en que el computador central ofrecerá el servicio

<bitácora\_depósito> Es el nombre y dirección relativa o absoluta de un archivo de texto que almacena las operaciones de depósito del cajero.

<bitácora\_retiro> Es el nombre y dirección relativa o absoluta de un archivo de texto que almacena las operaciones de retiro del cajero.

La sintaxis del módulo que ejecutan los cajeros automáticos:

`bsb_cli -d <nombre_módulo_atención> -p <puerto_bsb_svr> -c <op> -i <codigo_usuario>`

Dónde:

<nombre\_módulo\_atención>: Es el nombre de dominio o la dirección IP (versión 4) del equipo en donde se deberá ejecutar el módulo de atención centralizada.

<puerto\_bsb\_svr>: Es el número de puerto remoto en que el módulo de servicio atenderá la comunicación solicitada.

<op>: Indica si el usuario va a depositar o retirar dinero del cajero, puede tener dos valores d ó r

<código\_usuario> Es un número serial que identifica un usuario de forma única

### **Diseño del Protocolo de Comunicación**

Para la definición del protocolo se deben tener en cuenta los siguientes aspectos y como parte de la documentación se deberá describir en forma clara, cada uno de ellos:

1. ¿Qué tipo de sockets decidió emplear? Justifique su respuesta.
2. Identifique todos los mensajes del sistema, indicando: el formato del mismo, su tamaño en bytes, quién genera el mensaje y quién lo recibe y procesa.
3. Realice el diseño completo del protocolo de comunicación que construya y describa como opera el mismo. Incluya el punto de vista de los cambios de estado de las entidades que se comunican. (Se sugiere usar diagramas de máquinas de estados finitos y facilitar la explicación de cómo es la actividad del protocolo)
4. Describa aspectos del proyecto que funcionan según el enunciado y cuáles no. Cualquier requerimiento no desarrollado o que contenga fallas, deberá ser señalado claramente.
5. La implementación es en lenguaje C.

### **Condiciones de Entrega**

1. Esta sección debe imprimirse y será entregada el lunes de semana 8 (31 de octubre de 2016) a los docentes del laboratorio, o en su defecto, puede dejarlo en el casillero ubicado en el Departamento de Computación y TI, de 8:30 a.m. a 3:45 p.m. (cada quien en su sección)

2. La entrega de la aplicación debe hacerse de manera electrónica a través del Aula Virtual, hasta la fecha y hora indicada: domingo 30 de octubre de 2016 (semana 7) hasta las 11:50 p.m. El acceso para subir los proyectos será cerrado luego de la hora establecida, por lo tanto se recomienda no esperar hasta la última hora. (Deberá enviar también al correo electrónico del docente que corresponda)
3. Cualquier demora podrá dar la potestad al docente de no aceptar el trabajo.
4. Sea organizado en el documento que será entregado, incluya portada, contenido, descripción del problema a resolver, entre otros; se evaluará la presentación y estructura.
5. Cualquier falla de versión inapropiada, falta de algún elemento u otra situación anómala, podrá dar potestad al docente de colocar cero (0) como calificación.
6. Recuerde, el docente no está obligado a ajustar o adaptar su sistema. Usted debe preparar todo para facilitar la ejecución de su Proyecto sin esfuerzos. No piense que tendrá otra oportunidad para arreglar cualquier falla.
7. Los equipos deben ser de dos (2) estudiantes, en caso de ser necesario sus integrantes deberán asegurarse de poder explicar, a cabalidad, la distribución del trabajo.
8. Cada miembro del equipo debe estar en capacidad de comprender y desarrollar cualquier parte del trabajo, incluso aunque no le haya sido asignada originalmente ya que podrá ser interrogado al respecto durante la corrección del mismo.
9. Aquellas personas que no muestren un dominio de los detalles del proyecto no tendrán puntos en la evaluación.
10. Ambos deben estar inscritos en el curso y en aula virtual Si estas condiciones no se cumplen, el evaluador del proyecto podrá reprobar al integrante que no cumpla con las mismas.
11. Cualquier caso de plagio será severamente castigado, no será evaluado ninguno de los proyectos involucrados y serán aplicadas las sanciones correspondientes establecidas en los reglamentos de la universidad.
12. La entrega será con un archivo comprimido que incluya toda la documentación y los programas y se identificará de la siguiente manera:

**Apellido1\_Apellido2\_X.tar.gz**

(ver <http://ldc.usb.ve/~figueira/Cursos/ci3825/taller/material/TGZ.html> de cómo hacer un archivo tar.gz), donde X será reemplazado por el término Sockets y Apellido1 y Apellido2 corresponden a los apellidos de los

integrantes del grupo (por ejemplo, Alvarez\_Torrealba\_Sockets.tar.gz) con los siguientes archivos:

- ✓ Archivos fuente. Por ejemplo, si se trata de un proyecto en lenguaje C, debe incluir todos los ".c" y ".h".
- ✓ Makefile
- ✓ Un archivo LEEME.txt, que explique el contenido del "tar.gz":

Tenga cuidado de no incluir ejecutables, archivos objeto (.o) o cualquier otro archivo generado automáticamente a partir de los fuentes.

Para efectos de desarrollo puede utilizarse una máquina para correr todos los componentes (clientes y servidores). Una vez funcione en una máquina, debe probarlo en al menos dos máquinas para verificar que funcione bien remotamente. La corrección se hará usando varias máquinas.

### **Código**

El código deberá incluir un documento del tipo "Leeme.txt" con al menos los siguientes elementos:

1. Nombres y apellidos de los integrantes del grupo
2. Número de carnet
3. ¿Qué archivos lo componen y qué tiene cada archivo?
4. ¿Qué hace el programa?
5. ¿Cómo se ejecuta?
6. ¿Qué condiciones particulares tiene?, por ejemplo la entrada XX tiene máximo 50 caracteres" o "el máximo número de registros YY es 20"

Por su parte, el código deberá estar debidamente documentado, siguiendo los estándares de documentación y los encabezados de las funciones implementadas. Adicionalmente, todos sus programas deben seguir las buenas prácticas de estilo de programación en C y todas las llamadas al sistema deben ser correctamente manejadas.

### **Consideraciones Adicionales**

- ✓ Durante el funcionamiento del sistema puede ser que el servidor no esté siempre operativo. En ese caso el cliente hará la petición hasta 3 veces, si el servidor no le responde, entonces enviará un mensaje a la pantalla indicando que el tiempo de repuesta se agotó.
- ✓ El servidor deberá operar en forma continua.
- ✓ Recuerde que los servidores deben soportar obligatoriamente la concurrencia.
- ✓ Todos los parámetros estarán siempre presentes, pero pueden ser invocados en un orden diferente al que se indicó en la sintaxis. Aun así, el sistema deberá operar correctamente.
- ✓ Todo lo que usted asuma justifíquelo adecuadamente.
- ✓ Proyecto sin Leeme.txt , NO será corregido

### **Plataforma donde será evaluado su trabajo**

- ✓ Los proyectos serán evaluados en los equipos del Laboratorio de Computación (LDC). Deberá correr obligatoriamente en estos equipos.
- ✓ Para las pruebas de su proyecto utilice como número de puertos uno de 5 cifras que comience por 2 y que las siguientes 4 cifras coincidan con las últimas cuatro cifras del número de carnet de alguno de los integrantes del equipo. Si se requiere más de un puerto, repita el procedimiento pero con el número de carnet de otro integrante.

**ÉXITO!**