

PROTOCOLOS DE COMUNICACIÓN  
Trabajo Práctico Especial

*Compra de películas online*

Gabriel Cosi 47.008, Emmanuel Teisaire 48.019  
& Villa Fernández Santiago 47.399

23 de junio de 2009

# Índice

<b>1. Introducción</b>	<b>3</b>
<b>2. Man Page de la aplicación</b>	<b>3</b>
2.a. Movie Store Server . . . . .	3
2.b. Payment lookup Server . . . . .	3
2.c. Payment Server . . . . .	4
2.d. Payment Server . . . . .	5
<b>3. Descripción de protocolos desarrollados</b>	<b>6</b>
3.a. Protocolo de comunicación entre Movie Store Server y Movie Store Client .	6
3.b. Protocolo de comunicación entre Movie Store Server y Payment Lookup Server	8
<b>4. Problemas encontrados</b>	<b>9</b>
<b>5. Limitaciones de la aplicación</b>	<b>9</b>
<b>6. Posibles extensiones</b>	<b>9</b>
<b>7. Conclusiones</b>	<b>9</b>
<b>8. Guía de instalación</b>	<b>10</b>
<b>9. Instrucciones para la configuración</b>	<b>11</b>
9.a. Movie Store Server . . . . .	11
9.a.1. Ejemplo de configuración . . . . .	11
9.b. Payment Lookup Server . . . . .	11
9.b.1. Ejemplo de configuración . . . . .	11
9.c. Payment Server . . . . .	12

## Resumen

El presente informe tiene como objetivo explicitar detalles de la implementación de un sistema para comprar películas online.

## 1. Introducción

El trabajo práctico implementa una aplicación cliente servidor. El objetivo es que varios usuarios se puedan conectar a un único servidor (MovieStoreServer), el cual a su vez se conecta a otros servidores. Estos servidores contienen información de cuenta de cada usuario. Un servidor LDAP guarda información en su base de datos de los usuarios autorizados a utilizar el servicio. También contamos con implementaciones de servidores de pago, los cuales almacenan información sobre la plata que posee cada usuario.

## 2. Man Page de la aplicación

### 2.a. Movie Store Server

msserver(7)

msserver(7)

#### NAME

msserver - Aplicación servidor que almacena películas para comprar

#### SYNOPSIS

msserver

#### DESCRIPTION

Esta aplicación almacena películas para ser descargadas por el Movie Store Client. Debe contar con la información para poder conectarse con un servidor LDAP para validar usuarios y con un Payment Lookup server para poder contactar servidores de pago.

#### AUTHOR

Gabriel Cosi 47.008, Emmanuel Teisaire 48.019 & Villa Fernández Santiago 47.399.

Versión: 1.0

23 de Junio de 2009

msserver(7)

### 2.b. Payment lookup Server

plserver(7)

plserver(7)

NAME

plserver - Aplicación para buscar servidores de pago

SYNOPSIS

plserver [puerto]

DESCRIPTION

Esta aplicación contiene información sobre los servidores de pago, para ser consultada por el Movie Store Server

OPTIONS

[puerto] - Puerto al cual se bindeara el payment lookup server. Si se omite, se bindeará por default al puerto 10539

AUTHOR

Gabriel Cosi 47.008, Emmanuel Teisaire 48.019 & Villa Fernández Santiago 47.399.

Versión: 1.0

23 de Junio de 2009

plserver(7)

## 2.c. Payment Server

pserver(7)

pserver(7)

NAME

pserver - Servidor de pago

SYNOPSIS

pserver [puerto]

DESCRIPTION

Esta aplicación contiene información sobre el crédito disponible de sus clientes

OPTIONS

[puerto] - Puerto al cual se bindeará el payment server. Si se omite, se bindeará por default al puerto 10001

AUTHOR

Gabriel Cosi 47.008, Emmanuel Teisaire 48.019 & Villa Fernández Santiago 47.399.

Versión: 1.0

23 de Junio de 2009

pserver(7)

## 2.d. Payment Server

msclient(7)

msclient(7)

### NAME

msclient - Aplicación para bajar películas

### SYNOPSIS

msclient

### DESCRIPTION

Esta aplicación se utiliza para comunicarse con un servidor Movie Store, mediante el cual se pueden bajar películas. El usuario debe estar registrado en un servidor de pago y en el servidor LDAP que utiliza el Movie Store Server.

### AUTHOR

Gabriel Cosi 47.008, Emmanuel Teisaire 48.019 & Villa Fernández Santiago 47.399.

Versión: 1.0

23 de Junio de 2009

msclient(7)

### 3. Descripción de protocolos desarrollados

#### 3.a. Protocolo de comunicación entre Movie Store Server y Movie Store Client

Todos los mensajes tienen los siguientes campos:

- Primer campo: Código de operacion
- Segundo campo: Nombre de usuario.
- Tercer Campo: Información variable referente al mensaje.

Para comunicación entre le Movie Store Server y el Movie Client se utilizaron los siguientes codigos (tipos) de paquetes:

Descripcion del mensaje	Codigo de Mensaje
*Mensaje para Conectar un nuevo usuario:	CON_USR
*Mensaje previo a comprar una pelicula:	PRE_BUY_MOVIE
*Mensaje para comprar una pelicula:	BUY_MOVIE
*Mensaje previo a cambiar un password:	PRE_CHG_PASS
*Mensaje para cambiar un password:	CHG_PASS
*Mensaje para iniciar transaccion de descargar:	DOWNINFO
*Mensaje para realizar transaccion de un archivos:	DOWNDATA
*Mensaje para iniciar listar Clientes:	PRE_LIST_CLIENTS
*Mensaje para realizar listar Clientes:	LIST_CLIENTS
*Mensaje para realizar listar Generos:	LIST_GEN
*Mensaje para realizar listar peliculas:	LIST_MOVIES
*Mensaje para realizar desconectar:	DISCONNECT

Los mensajes que tienen codigos previos son aquellos que deben ir encriptados, para estos casos se manda un mensaje previo para sacar del LDAP la contraseña del usuario y asi poder desencriptar el mensaje encriptado por DES.

##### Proceso para Conectar un usuario:

El mensaje que se envia posee el nombre de usuario, el código de operacion CON\_USR y en el campo de informacion variable, posee el password del usuario. Luego el servidor contesta con el mismo mensaje solo que en la parte de informacion variable, da un OK si esta todo bien o un KO si ocurrio algun error.

##### Proceso para Comprar Pelicula:

Primero se envia un mensaje con código PRE\_BUY\_MOVIE, y el nombre de usuario, para que el servidor busque la contraseña para desencriptar el proximo mensaje. Luego se manda el segundo mensaje, con código BUY\_MOVIE, nombre del usuario, y la parte de informacion variable encriptada, con el numero de cuenta y informacion necesaria para el servidor de pago. El servidor envia un mensaje con BUY\_MOVIE, nombre de

usuario y en la parte de informacion un numero de ticket, mas informacion relevante.

Proceso para cambiar un password:

Primero se envia un mensaje con código PRE\_CHG\_PASS, y el nombre de usuario, para que el servidor busque la contraseña para desencriptar el proximo mensaje. Luego se manda el segundo mensaje, con código CHG\_PASS, nombre del usuario, y la parte de informacion variable encriptada, con el viejo password y el nuevo password. El servidor envia un mensaje con CHG\_PASS, nombre de usuario y en la parte de informacion un el viejo y el nuevo password queda de la misma manera, y en el campo resp da un código de si fue exitoso o no.

Proceso para descargar una pelicula:

Primero se envia un mensaje con código DOWNINFO y el nombre de usuario, para que el servidor sepa que se quiere iniciar una descargar, y devuelva una estructura del mismo tipo diciendo si puede iniciar la descarga. Si la descarga fue autorizada, el servidor comienza a mandar mensajes con código DOWNDATA, los cuales las partes del archivo. El cliente va armando el archivo con los mensajes que va recibiendo.

Proceso de listar clientes:

Primero se envia un mensaje con código PRE\_CHG\_CLIENTS, y el nombre de usuario, para que el servidor busque la contraseña para desencriptar el proximo mensaje. Luego el cliente recibe mensajes con código LIST\_CLIENTS. En la estructura viene información con los clientes para que la aplicacion cliente la imprima en la pantalla.

Proceso de listar géneros:

Primero se envia un mensaje con código LIST\_GEN. El servidor lo recibe y manda (de a uno) los generos no repetidos que tiene.

Proceso de listar películas:

Primero se envia un mensaje con código LIST\_MOVIES, el cliente le envía el género específico. El servidor lo recibe y manda (de a uno) las películas que concuerdan con el género.

Proceso de desconexión: Se envía un único mensaje con código DISCONNECT. El servidor lo recibe y actualiza la información que tiene sobre ese usuario.

Otro detalle a destacar es que existe un cuarto campo llamado *Magic Cookie*, que es un numero que el cliente setea de manera “aleatoria” (mediante uso de la funcion `rand()`) cuando manda un mensaje de conectar un nuevo usuario, y este numero lo recibe el servidor, le aplica una función matematica (actualmente la función que se aplica es la identidad) y le devuelve el nuevo valor y el cliente verifica que el valor devuelto sea el correcto, sino lo es, devuelve un error. Existen funciones de seteo de valores tanto para el cliente como para el servidor.

### **3.b. Protocolo de comunicación entre Movie Store Server y Payment Lookup Server**

La comunicación entre el mss y el pls se realiza de la siguiente manera: Hay un sólo tipo de paquetes, que contiene la siguiente información: Nombre del payment server, su IP y puerto y su TTL. El paquete tiene una longitud constante, en nuestra implementación de 74 bytes, distribuidos de la siguiente forma: 50 bytes para el nombre del payment server (formato String), 16 bytes para la IP (formato String), 4 bytes para el puerto (formato Int), y 4 bytes para el campo TTL (formato Int).

Cuando el Movie Store Server quiere comunicarse con el Payment look up server, arma un paquete con las características descritas anteriormente, poniendo en campo nombre, el nombre del payment server. Si el payment lookup server encuentra el payment server pedido, responde con el mismo paquete, con el nombre del payment server, su ip y puerto y su TTL. En caso de que el payment server pedido sea incorrecto, en el campo del nombre del payment server se pone el string "ERROR".



## 4. Problemas encontrados

Algunos de los problemas encontrados en la ultima version son:

- No verifica bien el MD5, cuando recibe una pelicula el cliente. A veces funciona.
- A veces no se guarda bien el estado de conexion del usuario.
- No esta probado capturar la señal SIGPIPE, cuando se cierra un socket, por algun inconveniente.

## 5. Limitaciones de la aplicación

La aplicacion no fue testeada con mas de cuatro clientes a la vez.

## 6. Posibles extenciones

Algunas posibles extenciones de la aplicacion dentro de las variadas que existen son:

- Se podría cambiar el sistema de guardar datos en el msserver por uno que implemente sqlite. Esto incrementaría la eficiencia.
- Se podrían agregar otros servidores de pago, con más información.
- Se podría agregar nuevos métodos de encriptación.
- Implementar retransmisión UDP entre el Movie Store Server y el Payment Lookup Server.

## 7. Conclusiones

En este trabajo practico hemos implementado conceptos vistos en la materia, por ejemplo sockets y señales. Los servidores son concurrentes, lo cual nos dificultó la programacion, pero nos permitio una mayor extensibilidad.

## 8. Guía de instalación

Para instalar la aplicación, se deben extraer todos los archivos del cd y colocarlo en un directorio. Luego mediante consola acceder a la carpeta que tiene el archivo titulado *Makefile*, y ejecutar la orden: **make again**.

Al ejecutar esta orden, se compilarán todos los archivos con código fuente y se obtendrán en la misma carpeta los siguientes ejecutables:

- **msclient** Aplicación Movie Store Client.
- **msserver** Aplicación Movie Store Server.
- **plserver** Aplicación Payment Lookup Server.
- **pserver** Aplicación Payment Server.

Una vez obtenido los archivos ejecutables, y creado los respectivos archivos de configuración en los casos que haga falta, se podrán ejecutar los distintos servidores y el cliente.

## 9. Instrucciones para la configuración

### 9.a. Movie Store Server

Al Movie Store Server no se le pasan argumentos por línea de comandos. La configuración se realiza por medio del archivo mssconfig que se encuentra dentro de la carpeta Config. Dicho archivo es de texto, y su estructura es la siguiente:

En la primer línea se introduce la información (IP y puerto) del servidor LDAP en la forma:

```
ldapServer=[IP];[Puerto]
```

En la segunda línea se introduce la información del Payment lookup server en la forma:

```
plServer=[IP];[Puerto]
```

En las líneas sucesivas se escriben las IPs y puertos a los cuales se bindeará el Movie Store Server. En cada línea se escribe una IP y un puerto de la siguiente forma:

```
mssendpoint=[IP];[Puerto]
```

Se validan en todas las ip de este archivo que sean ips bien formadas, y en los puertos que sean mayores a 1024 y menores a 65000.

#### 9.a.1. Ejemplo de configuración

```
ldapServer=10.6.0.159;389
plServer=127.0.0.1;10539
mssendpoint=127.0.0.1;10535
mssendpoint=127.0.0.1;11535
mssendpoint=127.0.0.1;12535
```

### 9.b. Payment Lookup Server

El Payment lookup server recibe opcionalmente por línea de comandos el puerto en el cual debe escuchar. Si no se le pasa dicho argumento, o se le pasa un puerto inválido toma como default el puerto 10539. La información de los payment servers las toma del archivo plsconfig. Dicho archivo es de texto, y contiene en cada línea la información sobre un payment server. La estructura de cada línea es:

```
[Nombre];[IP];[Puerto];[TTL]
```

#### 9.b.1. Ejemplo de configuración

```
Visa;127.0.0.1;10001;100
MasterCard;127.0.0.1;10002;200
AmericanExpress;127.0.0.1;10003;300
Diners;127.0.0.1;10004;400
Cabal;127.0.0.1;10005;500
```

### **9.c. Payment Server**

El payment server al igual que el payment lookup server recibe opcionalmente por línea de comandos el puerto en el cual escuchar. Si no se le pasa o si es incorrecto usará por default el puerto 10001. La información de los clientes en el payment server se lee del archivo de texto psconfig cuya estructura es la provista por la cátedra.