

# **APLICACIÓN SERVIDOR/CLIENTE TIENDA ONLINE**

**Fundamentos de Redes**

Daniel Bolaños Martínez  
José María Borrás Serrano

## Descripción de la aplicación:

Para la realización de la aplicación Cliente/Servidor, vamos a utilizar un protocolo TCP para simular la realización de compras en una tienda online.

Es necesario usar TCP en lugar de UDP, ya que no podemos permitir que en la comunicación entre el cliente y el servidor se produzcan errores o haya pérdidas. Debemos ser capaces de asegurar que los pedidos se han realizado correctamente.

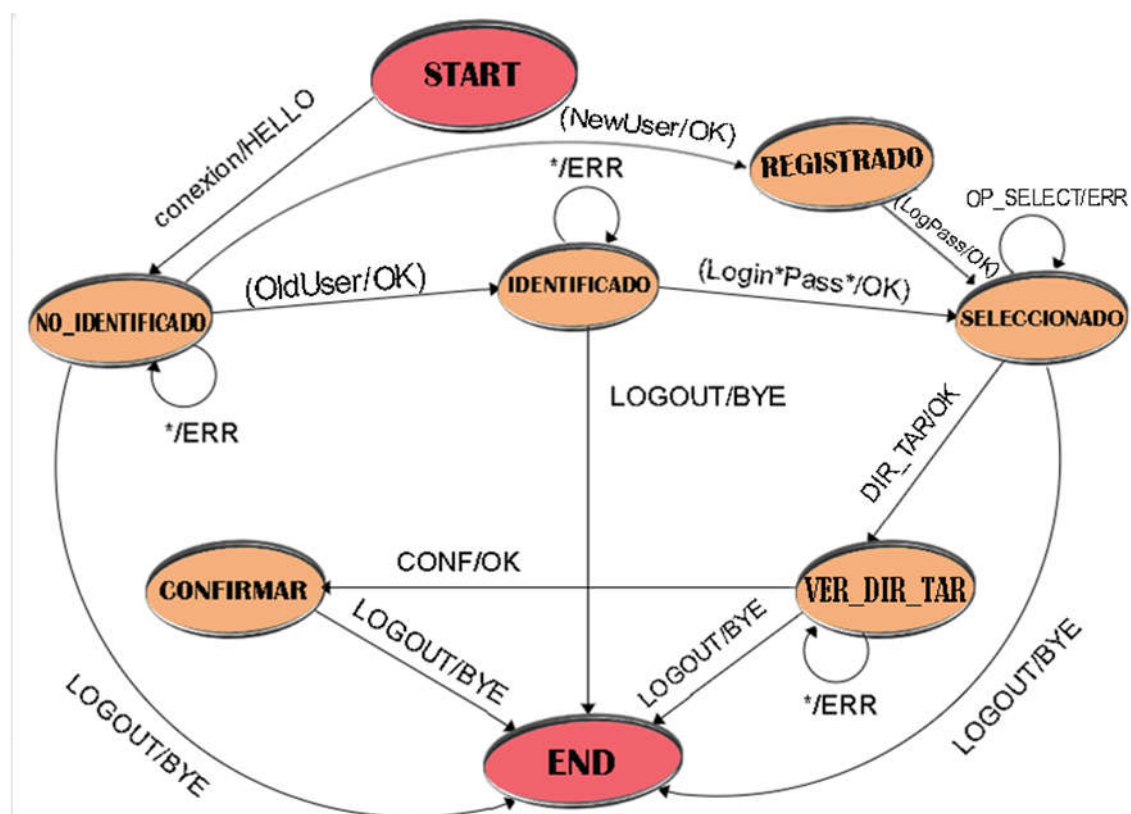
El funcionamiento sería el siguiente:

El cliente se conecta al servidor de la tienda y, a continuación, se identifica mediante su nombre de usuario y su contraseña. Luego elige un producto, pone su dirección de correo y tarjeta y, por último, confirma su pedido. Hay que tener en cuenta que una vez que el cliente se ha conectado con el servidor de la tienda, en cualquier momento puede decidir desconectarse.

En cada paso el servidor le mandará al cliente una señal indicando si el mensaje que ha recibido es correcto, y por tanto, pasa al siguiente estado o si el mensaje no es válido y le envía un código de error.

## Diagrama de estados:

El diagrama de estados de nuestro modelo cliente/servidor aplicado a la aplicación de la tienda online simplificada queda de la siguiente forma:



De forma resumida, el cliente manda petición al servidor para conectarse, este le envía un mensaje sobre si quiere registrarse o autenticarse, el cliente realiza la opción que necesite en cada caso. En el siguiente paso, el cliente elige los productos que quiera comprar, luego verifica sus datos personales y método de pago y finalmente confirma la compra.

En cualquier momento el cliente puede salir de la aplicación y la aplicación tendrá el tratamiento de errores que vea oportuno si en cada caso el cliente introduce los datos incorrectos. Nosotros optaremos por un bucle para pedir de nuevo los datos si en algún momento están equivocados.

## Mensajes que intervienen:

Los mensajes que intervienen en nuestra aplicación son los siguientes:

### Cliente:

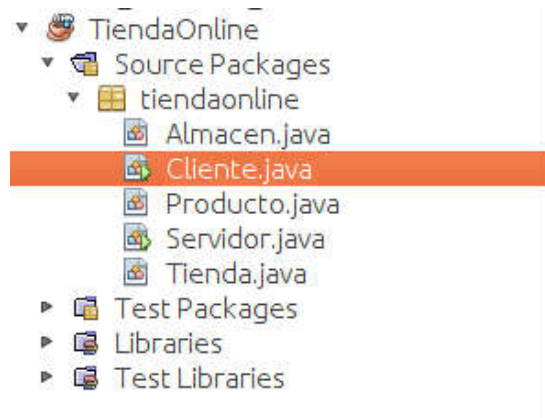
Código	Cuerpo	Descripción
2010	CONEXIÓN + dirección y puerto	Petición de conexión al servidor de la tienda
2050	NEW_USER	Petición de registro de nuevo usuario
2150	LOGIN + new_user + PASS + password	Mensaje de registro para el servidor
2200	OLD_USER	Petición de identificación de usuario existente
2250	LOGIN + old_user + PASS + password	Mensaje de identificación para el servidor
2511	OP_SELECT + producto	Mensaje para la selección de un producto
2553	VER_DIR_TAR + dirección_ facturación + tarjeta	Mensaje para enviar la dirección a la que debe enviarse el pedido y la tarjeta
2770	CONF + confirmación	Mensaje de confirmación del pedido
2999	LOGOUT	Mensaje para desconectarse del servidor

### Servidor:

Código	Cuerpo	Descripción
200	HELLO	Mensaje de confirmación de conexión al servidor
300	OK_LOG	Mensaje de elección para la identificación
310	ERR + "usuario o contraseña inválidos"	Mensaje de error de identificación
400	OK_LOG_IN	Mensaje de usuario y password aceptadas
410	OK_NEW	Mensaje de elección de registro nuevo
500	OK_REG_IN	Mensaje de nuevo usuario y contraseña aceptadas
510	OK_SELECT	Mensaje de confirmación de selección de producto
600	ERR + "selección no válida"	Mensaje de error de selección de producto
610	OK_VER_DIR_TAR	Mensaje de confirmación de la dirección de facturación y tarjeta
700	ERR + "dirección o tarjeta incorrectos"	Mensaje de error de la dirección de facturación y/o tarjeta
710	OK_CONF	Mensaje de confirmación del pedido
1000	BYE	Mensaje de confirmación de desconexión al servidor

## Funcionamiento de la aplicación:

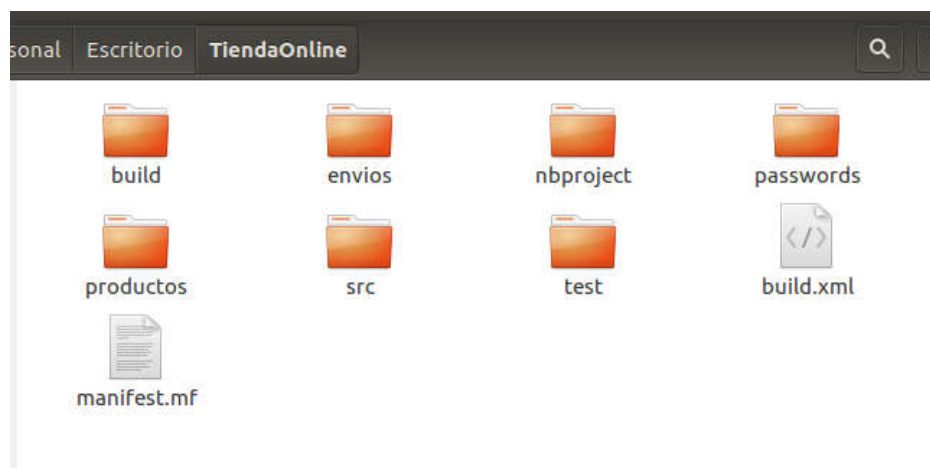
La aplicación la hemos programado en Java usando lo aprendido en la Práctica 2 con los protocolos TCP para conectar cliente y servidor.



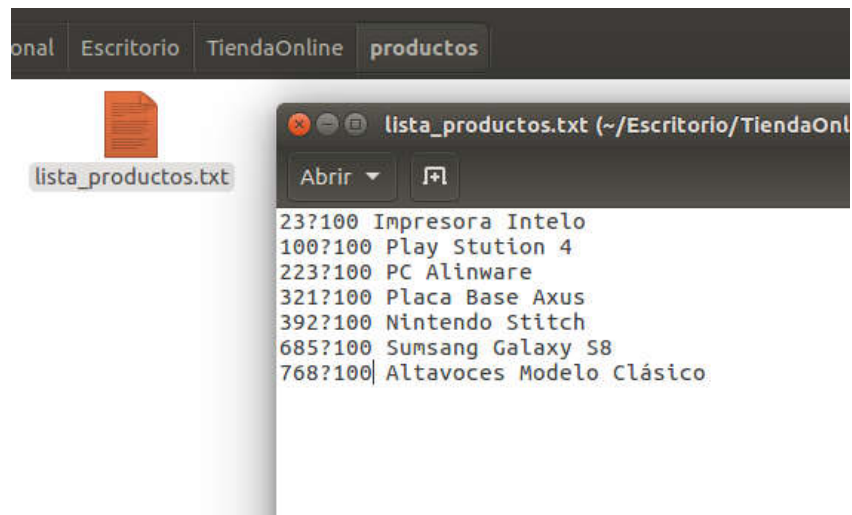
Hemos creado una clase Producto que funciona como elemento a vender y que tiene tres campos (código de identificación, descripción del producto y unidades disponibles).

La clase Almacén, que es global para cada cliente y funciona como una Lista de Productos, es la que servirá de escaparate para ver los productos que puede comprar el cliente.

La clase Tienda tiene algunas funciones que usan el cliente y servidor para llevar a cabo la compra venta de productos. Finalmente las clases Cliente y Servidor funcionan de forma similar a las vistas en la Práctica, creando cauces de lectura y escritura en modo texto con `PrintWriter` y `BufferedReader` para intercambiar los mensajes entres ambos.



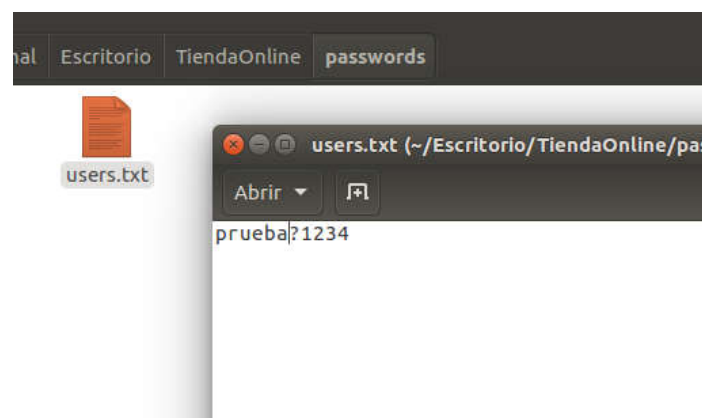
De forma adicional hemos creado 3 carpetas productos (que contiene la lista de productos disponibles en un formato para leer y escribir internamente desde la aplicación), passwords (que contiene los usuarios junto con sus contraseñas) y envíos (que genera los envíos de los productos a los clientes que hayan usado la aplicación).



Primeramente, iniciaremos el servidor y luego los clientes. La aplicación soporta concurrencia entre clientes. La simulación la haremos con dos clientes.

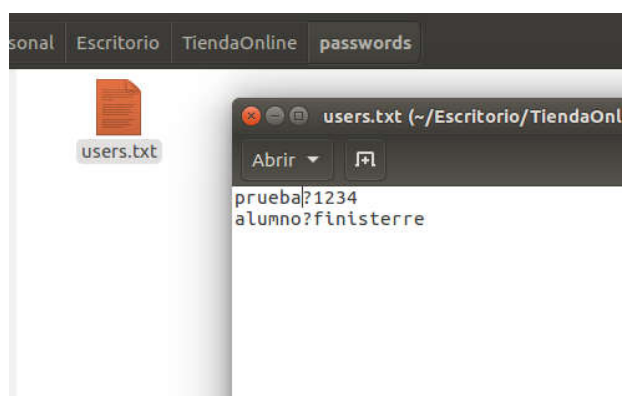


En el cliente nos da a elegir entre registrarnos y autenticarnos, hemos introducido en la carpeta de passwords un usuario y contraseña previos (user : prueba, pass : 1234). Así que en uno nos identificaremos con eso y en otro nos registraremos.



tiendaOnline (run-single) x	TiendaOnline (run-single) #2 x	TiendaOnline (run-single) x	TiendaOnline (run-single) #2 x
<pre>ant -f /home/danibolanos/Escritorio/TiendaOnline -Dja init: Deleting: /home/danibolanos/Escritorio/TiendaOnline/bi deps-jar: Updating property file: /home/danibolanos/Escritorio/ Compiling 1 source file to /home/danibolanos/Escritor compile-single: run-single: Elija una opción:   1 Registrar un nuevo usuario   2 Autenticarse 2 Introduzca su usuario prueba Introduzca su contraseña 1234  La lista de productos disponibles es:  Id: 23. Impresora Intelo. Uds: 100 Id: 100. Play Stution 4. Uds: 100 Id: 223. PC Alinware. Uds: 100 Id: 321. Placa Base Axis. Uds: 100 Id: 392. Nintendo Stitch. Uds: 100 Id: 685. Sumsang Galaxy S8. Uds: 100 Id: 768. Altavoces Modelo Clásico. Uds: 100  Introduzca el número del producto que desea:  </pre>		<pre>ant -f /home/danibolanos/Escritorio/TiendaOnline init: Deleting: /home/danibolanos/Escritorio/TiendaOnl deps-jar: Updating property file: /home/danibolanos/Escrit Compiling 1 source file to /home/danibolanos/Esc compile-single: run-single: Elija una opción:   1 Registrar un nuevo usuario   2 Autenticarse 1 Introduzca su nuevo usuario alumno Introduzca su nueva contraseña finisterre  La lista de productos disponibles es:  Id: 23. Impresora Intelo. Uds: 100 Id: 100. Play Stution 4. Uds: 100 Id: 223. PC Alinware. Uds: 100 Id: 321. Placa Base Axis. Uds: 100 Id: 392. Nintendo Stitch. Uds: 100 Id: 685. Sumsang Galaxy S8. Uds: 100 Id: 768. Altavoces Modelo Clásico. Uds: 100  Introduzca el número del producto que desea:  </pre>	

El de la izquierda ha usado el usuario y contraseña existente y el de la derecha se ha identificado con (user : alumno, pass : finisterre), que automáticamente han pasado a formar parte del archivo users.txt de la carpeta passwords.



```
Elija una opción:
  1 Registrar un nuevo usuario
  2 Autenticarse
2
Introduzca su usuario
hola
Introduzca su contraseña
mundo
Usuario o contraseña no válidos
Introduzca su usuario
|
```

Si nos autenticásemos con un usuario y contraseña no existentes en el fichero, nos daría un error y tendríamos que volver a introducir el usuario o clave o salir de la aplicación para registrarnos.



Como podemos ver, después de autenticarse o registrarse, aparece la lista de productos, que podremos elegir indicando su Id y el número de unidades que queremos obtener. Finalmente se nos preguntará si queremos seguir comprando o no.

Al ser Almacén una clase global, todos los cambios que se hagan en cualquier hebra cliente, se harán en el fichero de lista\_productos.txt por lo que siempre será coherente la información respecto a los productos disponibles.

La lista de productos disponibles es:

```
Id: 23. Impresora Intelo. Uds: 100
Id: 100. Play Stution 4. Uds: 100
Id: 223. PC Alinware. Uds: 100
Id: 321. Placa Base Axus. Uds: 100
Id: 392. Nintendo Stitch. Uds: 100
Id: 685. Sumsang Galaxy S8. Uds: 100
Id: 768. Altavoces Modelo Clásico. Uds: 100
```

```
Introduzca el número del producto que desea:
23
Introduzca el número de unidades que desea de dicho producto:
5
¿Desea seguir comprando más productos? (Si/No):
No
Introduzca la dirección de facturación
Calle Periodista Daniel Saucedo Aranda
Introduzca el número de su tarjeta de crédito
1234543465e3234e
Datos incorrectos
Introduzca la dirección de facturación
Calle Periodista Daniel Saucedo Aranda
Introduzca el número de su tarjeta de crédito
1234543234565432
Confirmar Pedido(Si/No):
Si
Gracias por utilizar nuestra aplicación
```

Con el usuario prueba hemos comprado 5 Impresoras Intelo, hemos introducido la dirección y tarjeta (la primera vez incorrecta, ya que tienen que ser 16 números) y finalmente hemos confirmado la compra.

Con el usuario alumno hemos comprado 1 Play Stution 4 y hemos seguido comprando 2 Impresoras. Como podemos observar, como el cliente “prueba” hemos seleccionado antes las 5 impresoras, cuando realiza la acción de seguir comprando, el usuario “alumno”, se encuentra la lista de productos actualizada con 5 impresoras menos.

Finalmente hemos introducido la dirección y tarjeta correctas y hemos confirmado la compra.

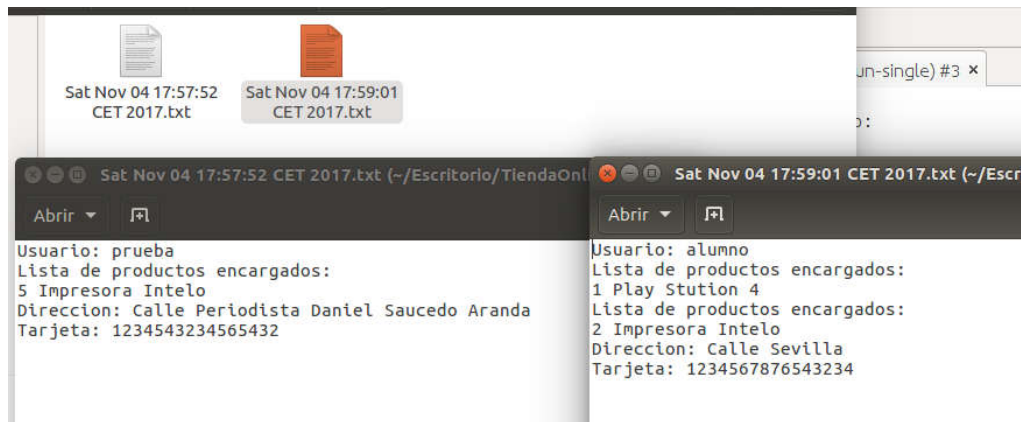
```
100
Introduzca el número de unidades que desea de dicho producto:
1
¿Desea seguir comprando más productos? (Si/No):
Si

La lista de productos disponibles es:

Id: 23. Impresora Intelo. Uds: 95
Id: 100. Play Stution 4. Uds: 99
Id: 223. PC Alinware. Uds: 100
Id: 321. Placa Base Axus. Uds: 100
Id: 392. Nintendo Stitch. Uds: 100
Id: 685. Sumsang Galaxy S8. Uds: 100
Id: 768. Altavoces Modelo Clásico. Uds: 100

Introduzca el número del producto que desea:
23
Introduzca el número de unidades que desea de dicho producto:
2
¿Desea seguir comprando más productos? (Si/No):
No
Introduzca la dirección de facturación
Calle Sevilla
Introduzca el número de su tarjeta de crédito
1234567876543234
Confirmar Pedido(Si/No):
Si
Gracias por utilizar nuestra aplicación
```

Una vez que confirmamos la compra, se generan en la carpeta envíos, dos archivos de texto con la hora de compra como nombre que contienen la información relativa a cada cliente. Si la respuesta a confirmación es “No”, este archivo no se generará.



Finalmente, podemos comprobar, desde el mismo archivo lista\_productos.txt o creando otro cliente y viendo la lista de productos. De que los productos comprado han decrementado sus unidades correctamente. Cuando un producto llega a 0 unidades, no deja comprarlo.

