

The background of the slide features a large, semi-transparent clapperboard on the left side, tilted diagonally. The clapperboard has text such as 'PROD', 'ROLL', 'SCENE', 'DIRECTOR', and 'RATE'. To the right of the clapperboard is a director's chair with a black seat and backrest, and a gold-colored frame. The backrest of the chair has the word 'DIRECTOR' written on it. In front of the chair is a black traffic cone. The overall background is a light, neutral color.

Funcionamiento de Servicios Web, FTP

Tema 2.- Nivel de aplicación en Internet

*Dr. Daniel Morató
Redes de Computadores
Ingeniero Técnico en Informática de
Gestión, 2º curso*

Material adaptado del libro *Computer Networking: A Top Down Approach Featuring the Internet*,
3rd edition. Jim Kurose, Keith Ross, Ed. Addison-Wesley, Julio 2004

Temario

0.- Presentación de la asignatura

1.- Introducción

2.- Nivel de aplicación en Internet

3.- Nivel de transporte en Internet

4.- Nivel de red en Internet

5.- Nivel de enlace



Temario

0.- Presentación de la asignatura

1.- Introducción

2.- Nivel de aplicación en Internet

- Principios
- Funcionamiento de servicios
- Diseño y programación de servicios

3.- Nivel de transporte en Internet

4.- Nivel de red en Internet

5.- Nivel de enlace



Tema 2: Servicios

Objetivos:

» Aprender con el ejemplo: Funcionamiento de protocolos de nivel de aplicación

- Web y HTTP
- FTP



Tema 2: Servicios

Objetivos:

» Aprender con el ejemplo: Funcionamiento de protocolos de nivel de aplicación

- Web y HTTP
- FTP



Web y HTTP

Términos

- » Una **Página Web** está compuesta por **objetos**
- » Un objeto puede ser un fichero HTML, una imagen JPEG, un applet JAVA, un fichero de sonido...
- » La página Web está compuesta por un **fichero HTML base** que hace referencia a otros objetos
- » Se hace referencia a cada objeto mediante un **URL**
- » Ejemplo de URL:

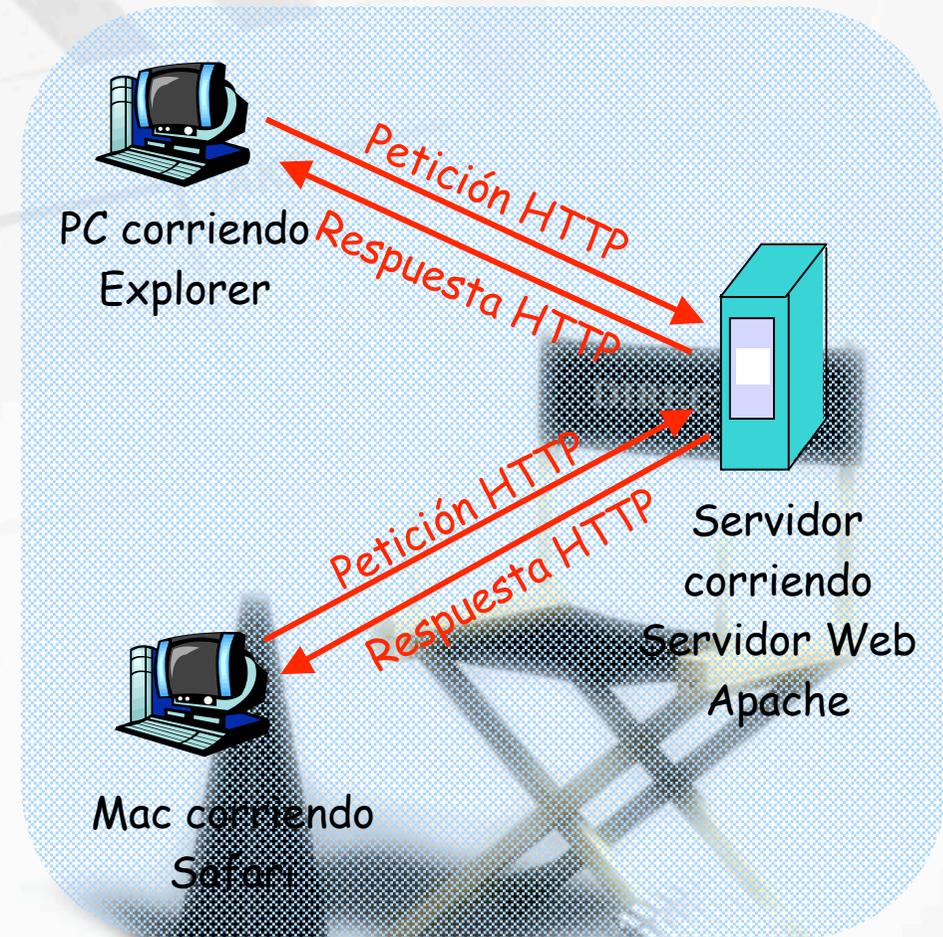
`http://www.tlm.unavarra.es/daniel/index.html`

host path

HTTP

HTTP: HyperText Transfer Protocol

- » Protocolo de nivel de aplicación de la Web
- » Modelo cliente/servidor
 - **cliente:** browser (navegador) que solicita, recibe y muestra objetos de la Web
 - **servidor:** el servidor Web envía objetos en respuesta a peticiones
- » HTTP 1.0: RFC 1945
- » HTTP 1.1: RFC 2068



HTTP

Usa TCP:

- » El cliente inicia una conexión TCP al servidor, puerto 80
- » El servidor acepta la conexión TCP del cliente
- » Cada uno tiene un socket conectado con el otro
- » Se intercambian mensajes HTTP entre el navegador y el servidor Web
- » Se cierra la conexión TCP

HTTP es “sin estado”

- » El servidor no mantiene ninguna información de peticiones anteriores del cliente

Nota

Los protocolos que mantienen “estado” son complejos

- Debe mantener la historia pasada (estado)
- Si el cliente/servidor falla, el estado entre ambos puede volverse incoherente

Empleo de las conexiones

HTTP no persistente

- » En cada conexión TCP se envía como máximo un objeto
- » HTTP/1.0

HTTP persistente

- » En la misma conexión TCP se pueden enviar varios objetos entre el servidor y el cliente
- » HTTP/1.1, funcionamiento por defecto



HTTP no persistente

Supongamos que el usuario solicita el URL

`www.tlm.unavara.es/~daniel/index.html`

(contiene texto y
1 referencia a
una imagen JPEG)

1a. El cliente HTTP inicia la conexión TCP con el (proceso) servidor de HTTP en `www.tlm.unavarra.es` puerto 80

1b. El servidor HTTP en el host `www.tlm.unavarra.es` espera conexiones al puerto 80. Acepta la conexión, notificando al cliente

2. El cliente HTTP envía un **mensaje de petición** (contiene el URL) a través de la conexión TCP (empleando el socket). El mensaje indica que el cliente quiere el objeto `/~daniel/index.html`

3. El servidor HTTP recibe el mensaje de petición, forma un **mensaje de respuesta** que contiene el objeto solicitado y lo envía a través de su socket

tiempo

HTTP no persistente

5. El cliente HTTP recibe el mensaje de respuesta que contiene el fichero HTML. Lo muestra y al interpretarlo encuentra la referencia a un objeto jpeg

6. Los pasos 1-5 se repiten para el objeto jpeg

4. El servidor HTTP cierra la conexión TCP

tiempo



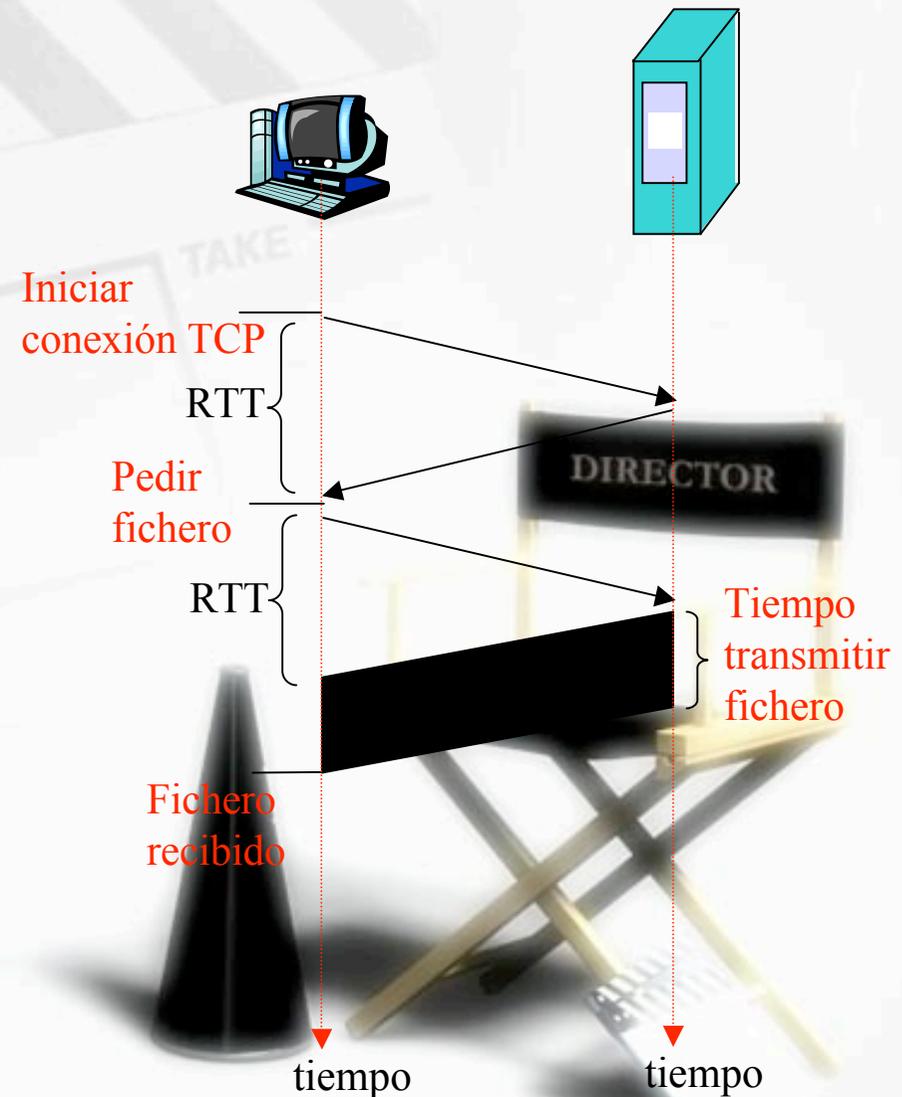
Modelo del tiempo de respuesta

Definición de RTT: tiempo para que un paquete pequeño viaje de cliente a servidor y vuelta

Tiempo de respuesta:

- » Un RTT para iniciar la conexión
- » Un RTT para la petición HTTP y el comienzo de la respuesta
- » Tiempo de transmisión del fichero

total = 2RTT + tiempo transmisión



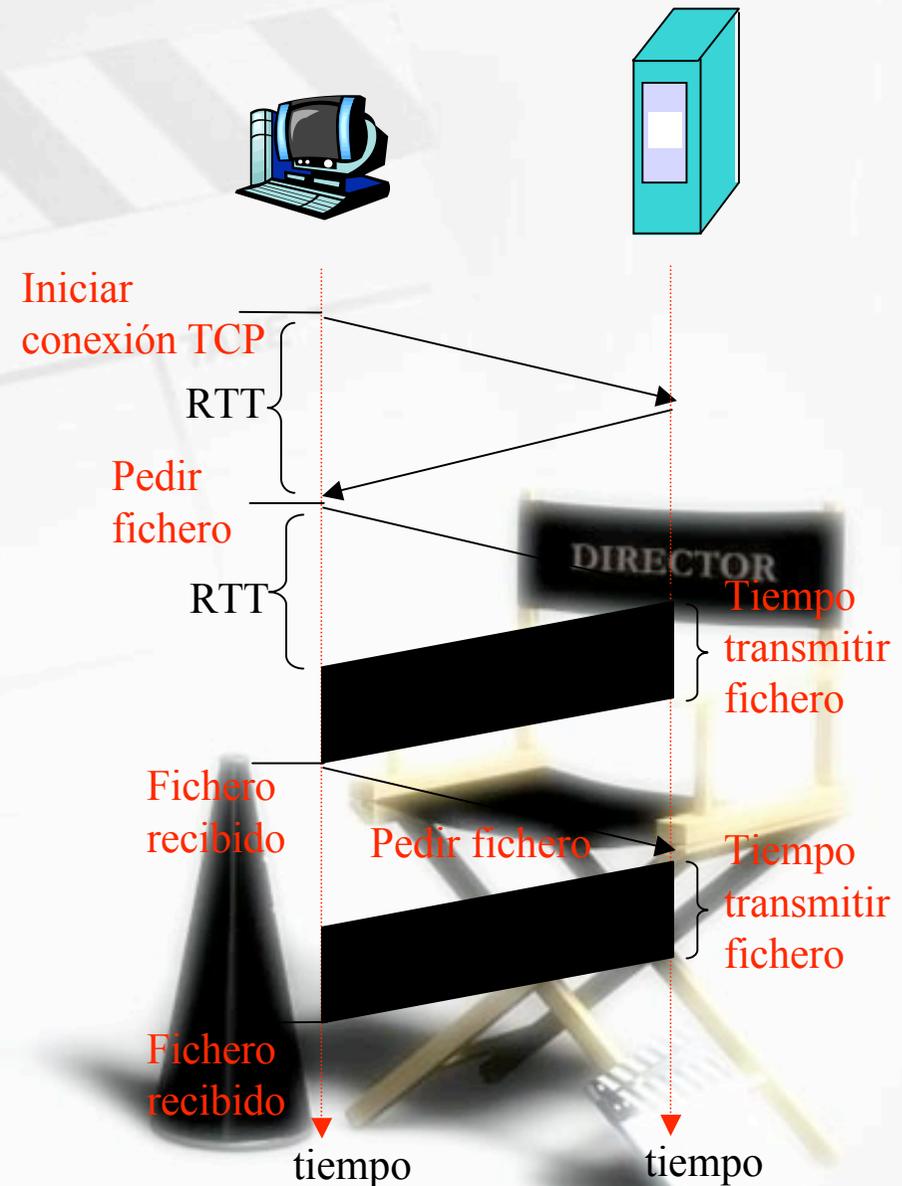
HTTP persistente

Con HTTP no persistente:

- » Requiere 2 RTTs por objeto
- » OS debe reservar recursos para cada conexión TCP
- » Pero el navegador suele abrir varias conexiones TCP en paralelo

HTTP persistente:

- » El servidor deja la conexión abierta tras enviar la respuesta
- » Los siguientes mensajes HTTP entre cliente y servidor van por la misma conexión



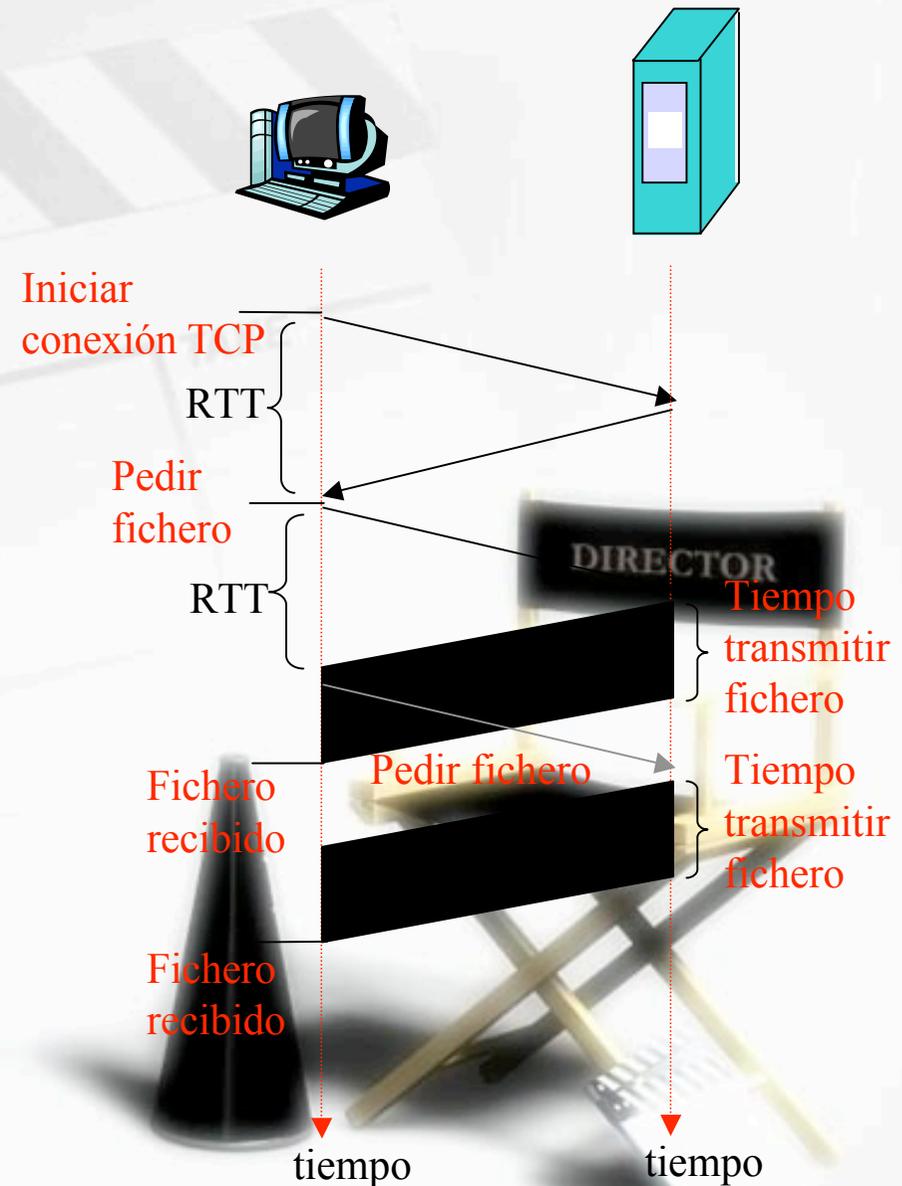
HTTP persistente

Persistente sin pipelining:

- » El cliente manda la nueva petición cuando ha terminado de recibir la respuesta anterior
- » Al menos un RTT por cada objeto

Persistente con pipelining:

- » *default* en HTTP/1.1
- » El cliente envía petición tan pronto como encuentra una referencia a objeto
- » Solo un RTT para todos los objetos referenciados en la página base



HTTP request message

- » Dos tipos de mensajes messages: *request*, *response*
- » Mensaje HTTP request :
 - ASCII (formato legible por humanos)

línea de petición
(comandos GET,
POST, HEAD)

líneas de
cabecera

Retorno del carro,
fín de linea
indica fin del mensaje

```
GET /~daniel/index.html HTTP/1.1
Host: www.tlm.unavarra.es
User-agent: Mozilla/4.0
Connection: close
Accept-language: es
```

HTTP response message

línea de estado
(código de estado
frase de estado)

```
HTTP/1.1 200 OK
Connection close
Date: Thu, 06 Aug 1998 12:00:15 GMT
Server: Apache/2.0.47 (Unix)
Last-Modified: Mon, 22 Jun 1998
Content-Length: 6821
Content-Type: text/html
```

cabecera

datos, ej.,
fichero HTML
solicitado

```
datos datos datos datos datos...
```

Probando HTTP desde el cliente

1. Telnet a su servidor Web favorito:

```
telnet www.tlm.unavarra.es 80
```

Abre una conexión TCP al puerto 80 (puerto por defecto del servidor HTTP) de `www.tlm.unavarra.es`

Lo que se escriba se envía por la conexión TCP

2. Escribir una petición GET de HTTP:

```
GET /~daniel/ HTTP/1.1  
Host: www.tlm.unavarra.es
```

Escribiendo esto (y retorno del carro dos veces) se envía un petición HTTP 1.1 mínima pero completa al servidor

3. Vea el mensaje de respuesta del servidor

Tema 2: Servicios

Objetivos:

» Aprender con el ejemplo: Funcionamiento de protocolos de nivel de aplicación

- Web y HTTP
- **FTP**



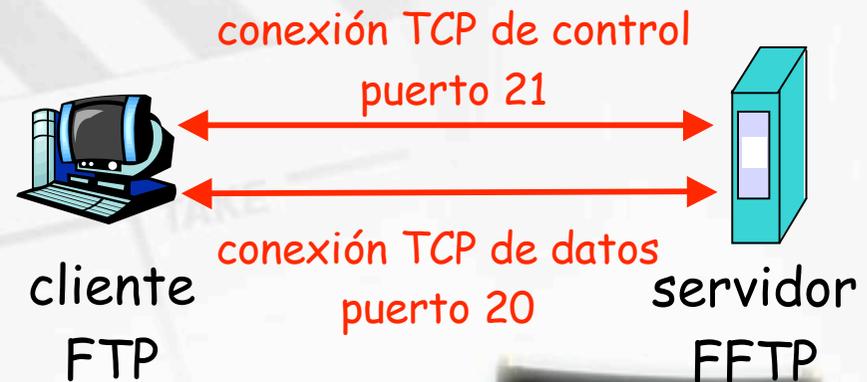
FTP: File Transfer Protocol



- » Transferencia de fichero hacia/desde host remoto
- » modelo cliente-servidor
 - *cliente*: extremo que inicia la transferencia (bien sea desde o hacia el extremo remoto)
 - *servidor*: host remoto
- » FTP: RFC 959
- » Servidor FTP: TCP puerto 21

FTP: conexiones de datos y control separadas

- » El cliente FTP contacta con el servidor en el puerto 21 empleando TCP
- » El cliente se autentica a través de esta conexión de control
- » El cliente puede explorar los directorios remotos enviando comandos por la conexión de control
- » Cuando el servidor recibe un comando para una transferencia de fichero abre una conexión TCP con el cliente
- » Tras transferir el fichero cierra esa conexión de datos



- » El servidor abre una segunda conexión TCP para transferir el fichero
- » Conexión de control "out of band"
- » El servidor FTP mantiene el "estado": directorio actual, autenticación

Comandos y respuestas FTP

Comandos de ejemplo:

- » Enviados como texto ASCII por el canal de control
- » **USER *username***
- » **PASS *password***
- » **LIST** devuelve una lista de los ficheros en el directorio actual
- » **RETR *filename***
Obtiene el fichero
- » **STOR *filename***
Almacena el fichero en el host remoto

Códigos de respuesta:

- » Código de estado y frase (como en HTTP)
- » **331 Username OK, password required**
- » **125 data connection already open; transfer starting**
- » **425 Can't open data connection**
- » **452 Error writing file**

Ejemplo de FTP

```
[daniel]$ ftp tlm13
Connected to tlm13.net.tlm.unavarra.es
220 tlm13.net.tlm.unavarra.es FTP
Name (tlm13:daniel): daniel
331 Password required for daniel.
Password:
230 User daniel logged in.
Remote system type is UNIX.
Using binary mode to transfer file
ftp> cd tesis
250 CWD command successful.
ftp> ls
200 PORT command successful.
150 Opening ASCII mode data connection
total 65
drwxr-xr-x  2 daniel  staff
drwx----- 48 daniel  staff
-rw-r--r--  1 daniel  staff
226 Transfer complete.
ftp> get todasfases.fig
local: todasfases.fig remote: toda
200 PORT command successful.
150 Opening BINARY mode data connection
226 Transfer complete.
4187 bytes received in 0.0101 seconds
ftp> bye
221-You have transferred 4187 bytes
221-Total traffic for this session
221-Thank you for using the FTP se
1320 Goodbye.
```

The screenshot shows an FTP client window with the following components:

- Host:** tlm13.net.tlm.unavarra.es
- Port:** 21
- User:** daniel
- Pass:** *****

Local [All Files] Listing:

Filename	Size	User
..	1,024	ro
...	1,024	ro
.AbiSuite	1,024	da
.ddd	1,024	da
.e-conf	1,024	da
.ee	1,024	da
.enlightenment	2,048	da
.freeciv	1,024	da
.fullcircle	1,024	da
.gftp	1,024	da
.gimp	1,024	da
.gnome	1,024	da

Remote [All Files] Listing:

Filename	Size	User
..	3,072	danie
...	3,072	danie
cnxfases.old.eps	9,694	danie
cnxfases.old.fig	3,944	danie
cnxfases.old.fig.ba	3,780	danie
cnxfases.ps	9,751	danie
todasfases.eps	11,120	danie
todasfases.fig	4,187	danie
todasfases.fig.bak	5,514	danie
todasfases.ps	11,180	danie

Terminal Window:

```
PWD:
257 "/opt3/staff/daniel/tesis" is current directory.
PASV:
227 Entering Passive Mode (1,1,1,13,26,101)
LIST -L:
150 Opening BINARY mode data connection for /bin/ls.
226 Transfer complete.
```

Próximo día

Más aplicaciones y protocolos de nivel de aplicación