

# Simulador gráfico de comunicaciones TCP



Universidad de Alcalá

Guía de usuario

# Contenido

<b>1. La aplicación .....</b>	<b>2</b>
1.1. Consideraciones .....	2
1.2. Parámetros que el usuario puede introducir .....	2
1.3. Interfaz de usuario .....	3
1.3.1. La simulación .....	5
1.3.2. Información/ayuda .....	7
<b>2. TCP .....</b>	<b>8</b>

# 1. La aplicación

Esta aplicación web te permite simular, a través de unos parámetros que el propio usuario puede elegir, una comunicación TCP entre un cliente y un servidor, mostrándose como resultado un diagrama de comunicaciones. En la sección [2. TCP](#) puede conocer brevemente más sobre este protocolo.

Para un correcto funcionamiento de la aplicación se recomienda el uso de [Google Chrome](#) o de navegadores basados en Chromium, como [Vivaldi](#), [Microsoft Edge](#) u [Opera](#).

## 1.1. Consideraciones

Para la simulación se tiene en cuenta lo siguiente:

- El tiempo de propagación será de "medio tic".
- La longitud de datos de SYN, SYN ACK, ACK, FIN será de 0 bytes.
- Las comunicaciones serán entre dos entidades, "cliente" (que comienza la comunicación y envía datos) y "servidor" (que responde con más datos).

## 1.2. Parámetros que el usuario puede introducir

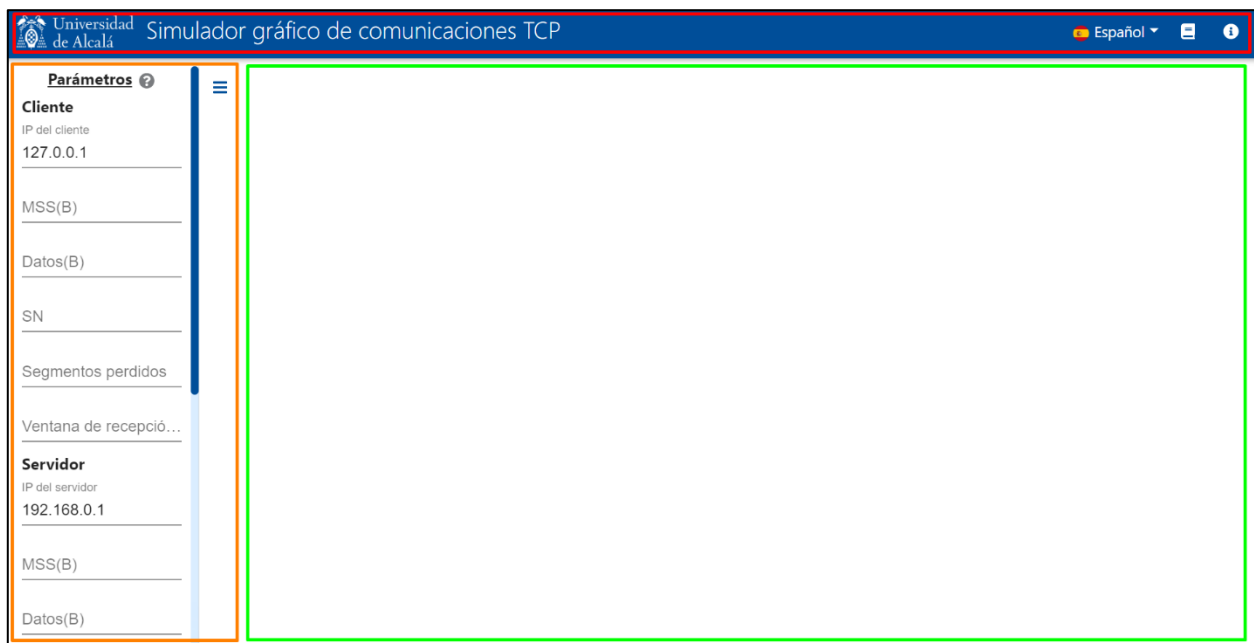
Los parámetros que el usuario puede introducir son los siguientes (\* obligatorio):

- **IP (*Internet Protocol*)\***: Identificador de la interfaz red de un dispositivo. Se usará IPv4 (32 bits), con formato [0-255].[0-255].[0-255].[0-255].
- **MSS (*Maximum Segment Size*)\***: Tamaño máximo, en bytes, que un dispositivo pueda recibir en un único segmento sin fragmentar.
- **Datos\***: Tamaño, en bytes, de los datos que se van a enviar.
- **SN (*Sequential Number*)\***: Número inicial de secuencia, intercambiado al inicio por ambos dispositivos, son los identificadores de los datos dentro del flujo de bytes con la finalidad de identificar y contar los bytes de la aplicación.
- **Segmentos perdidos**: Segmentos de datos a enviar que no llegan al destinatario. En la simulación no habrá pérdidas de paquetes en la fase de conexión y cierre.
- **Ventana de recepción\***: Búfer, en bytes, que posee el dispositivo en el que almacenar tramas recibidas.
- **Timeout**: Tiempo máximo, en tics, para considerar que un paquete se ha perdido al no haber recibido su confirmación correspondiente.

- **Algoritmo de congestión\***: Algoritmo a usar cuando hay congestión en la red. En este caso se puede elegir entre TCP Tahoe y TCP Reno. Aún se encuentran en fase de desarrollo, por lo que no se tendrán en cuenta y se simulará solo con el algoritmo de Evitación de la Congestión.
- **Umbral/Ssthresh\***: Límite de la ventana de congestión a partir del cual se pasará al modo de Evitación de la Congestión (EC).
- **Cierre de conexión\***: Qué dispositivo iniciará el fin de conexión.

### 1.3. Interfaz de usuario

La interfaz consta de tres áreas claramente definidas:



- **Barra superior (en rojo)**: Logo de la Universidad de Alcalá que redirige a su web, título de la aplicación (visible u oculto dependiendo del tamaño de la pantalla) y botones de selección de idioma (actualmente disponible en español e inglés), manual de usuario (este documento) e información de la aplicación (“acerca de”).

- Selección de idioma:






- Manual de usuario:



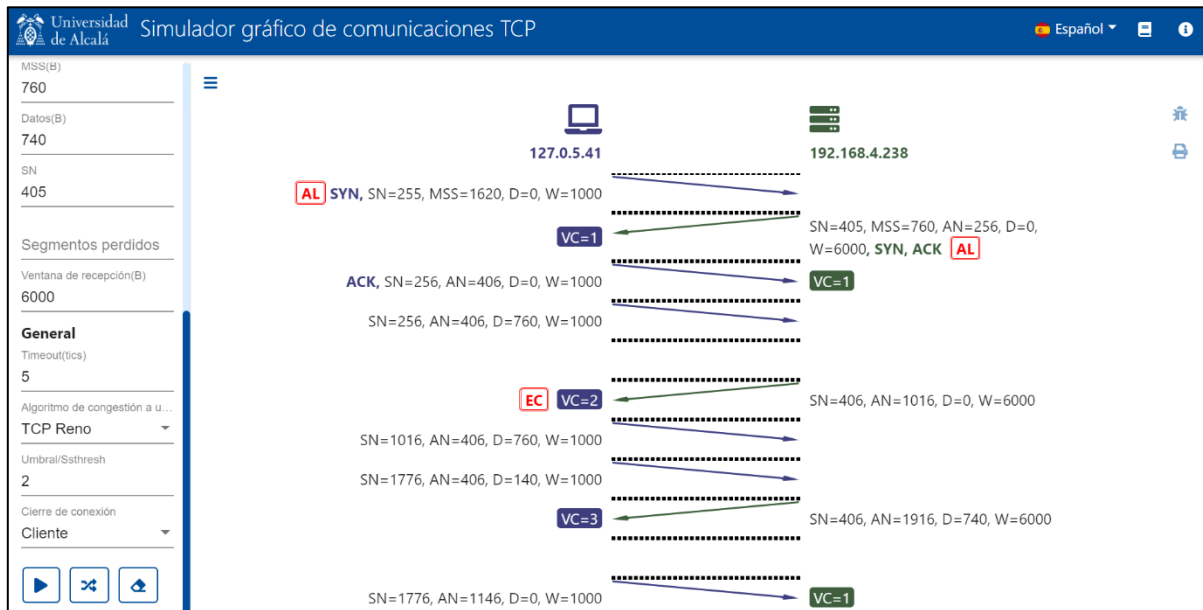
- Información de la aplicación:



- **Menú de parámetros (en naranja):** Junto a la palabra “Parámetros” incluye un botón con las consideraciones que se han tenido en cuenta para la simulación e información sobre los parámetros a introducir, iguales a los puntos [1.1. Consideraciones](#) y [1.2. Parámetros que el usuario puede introducir](#). Al final del menú, desplazándose con la barra lateral, se encuentran tres botones:
  - Simular (): Al pulsarlo, si se han rellenado los parámetros necesarios para realizar la simulación, se procede a realizar y mostrar el diagrama de comunicaciones. Si algún parámetro faltase o fuese erróneo, aparecerían alertas indicando qué está mal o falta por rellenar.
  - Rellenar con datos aleatorios (): Completa el formulario de los parámetros con valores aleatorios, excepto los segmentos perdidos, tanto para cliente como para servidor.
  - Limpiar formulario (): Vuelve la aplicación a su estado de inicio, limpiando el formulario y eliminando la simulación si hubiese mostrándose una.
- **Simulación (en verde):** En esta parte se muestran las aletas generadas en el caso de que las hubiese al intentar ejecutar la simulación o, si todo es correcto, el diagrama de comunicaciones resultante.

### 1.3.1. La simulación


Cuando todos los parámetros son correctos, se calcula la simulación y se muestra el diagrama de comunicaciones junto a dos botones al derecha de la pantalla:



Los datos que podemos ver en la simulación son los siguientes:

- **AL, EC y RR:** algoritmos usados, los cuales son son Arranque Lento (AL), Evitación de la Congestión (EC) y Recuperación Rápida (RR).
- **SYN, ACK, FIN:** *Flags* del segmento enviado, de sincronización (SYN), de reconocimiento (ACK) y de finalización (FIN).
- **SN:** Número de secuencia.
- **AN:** Número de reconocimiento.
- **MSS:** Tamaño máximo de segmento.
- **D:** Datos.
- **W:** Ventana de recepción.
- **VC:** Ventana de congestión.


Respecto a los dos botones a la derecha:

- **Reportar un error en la simulación** (

Asunto Reportar un error en la simulación

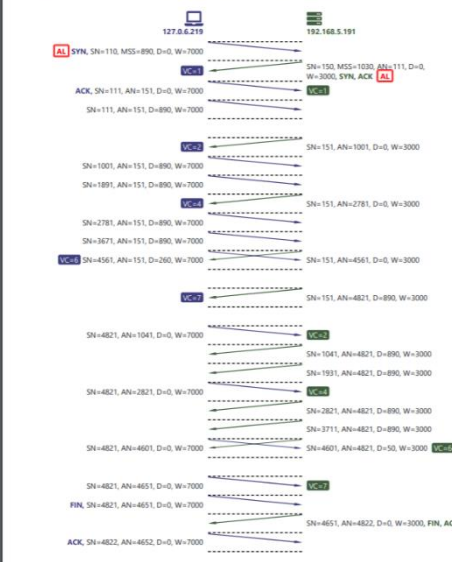
Parámetros  
{ "ipcli": "127.0.5.41", "msscli": 1620, "datoscli": 1660, "sncli": 255, "segpercli": null, "wcli": 1000, "ipserv": "192.168.4.238", "msscli": 1620, "datoscli": 1660, "sncli": 255, "segpercli": null, "wcli": 1000 }

-----Escriba a partir de aquí-----

- **Imprimir simulación** (


Simulador gráfico de comunicaciones TCP

Parámetros  
Cliente  
IP del cliente: 127.0.5.41 MSS(8): 890 Datos(8): 4710 SN: 110 Ventana de recepción(8): 7000  
Servidor  
IP del servidor: 192.168.5.191 MSS(8): 1030 Datos(8): 4500 SN: 150 Ventana de recepción(8): 3000  
General  
Timeout(tico): 7 Algoritmo de congestión: TCP Tahoe Umbra/Threshold: 8 Cierre de conexión: Cliente



10

Imprimir 2 páginas

Destino  Guardar como PDF

Páginas Todas

Páginas por hoja 1

Márgenes Predeterminado

Opciones

  - ☒ Encabezado y pie de página
  - ☒ Gráficos de fondo

Guardar Cancelar

### 1.3.2. Información/ayuda

Todo elemento en la aplicación posee información sobre lo que trata (*tooltip*) con solo posar el puntero del ratón sobre él, desde los botones hasta los segmentos de la simulación. Por ejemplo:

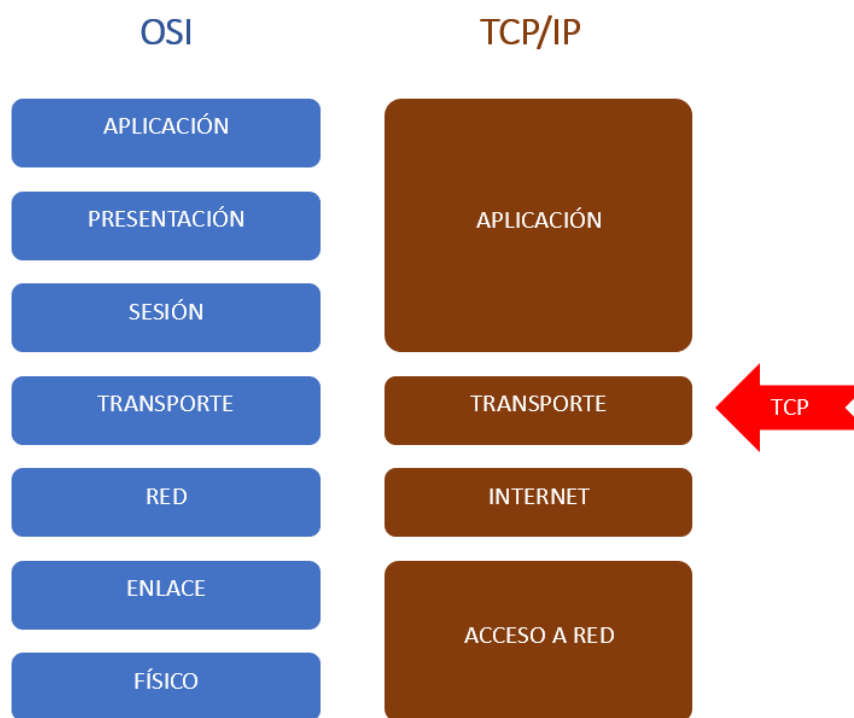




## 2. TCP

Para aprender sobre TCP, así como el funcionamiento de las redes de computadoras, se recomienda el libro “Redes de computadoras. Un enfoque descendente”, de James F. Kurose y Keith W. Ross, editorial en Pearson (ISBN: 9788490355282).

En resumen, TCP son las siglas de *Transmission Control Protocol* (Protocolo de Control de Transmisión, en español) y es ni más, ni menos, que uno de los múltiples protocolos utilizados en la red. Fue creado por las mismas personas que crearon el modelo que lleva su nombre, Vinton Cerf y Robert E. Kahn, a mediados de la década de los 70 y se encuentra en la capa que comunica la capa de red (IP, *Internet Protocol*) con la capa de aplicación, es decir, conocida como capa de transporte.



*Comparación entre el modelo OSI y el modelo TCP/IP*

A diferencia del protocolo UDP (*User Datagram Protocol*, o Protocolo de Datagramas de Usuario), TCP está orientado a conexión, lo que quiere decir que ambas máquinas, emisor y receptor, deben reconocerse mutuamente y establecer una comunicación fiable, para lo que se distinguen tres fases: establecimiento de conexión, transferencia de datos y finalización de la conexión. Esto implica una conexión punto a punto, un único emisor y receptor, con todas las ventajas e inconvenientes que ello conlleva.

Podríamos destacar como ventaja el control sobre los paquetes, ya que si un paquete se pierde se vuelve a enviar, así como el control de la congestión para evitar que la red se sature. Por su parte, como inconveniente tiene que, al ser un protocolo orientado a conexión, la transferencia de datos es más lenta que en UDP, siendo menos preferible su uso en programas que requieran extrema inmediatez o baja latencia.

TCP, para asegurarse un correcto funcionamiento, utiliza varios algoritmos de control de congestión. Esto son, si no hay segmentos perdido, Arranque Lento (Slow-start), aplicado en la fase de conexión y utilizado hasta que se alcanza el umbral de congestión; y Evitación de la Congestión (Congestion Avoidance), donde, cuando se ha alcanzado el umbral de congestión, la ventana se incrementa de forma lineal. En el caso de que el receptor reciba un segmento con un número de secuencia que no debería por la pérdida de un segmento, se utiliza el algoritmo de Recuperación Rápida (Fast Recovery). Además, mejorando los algoritmos antes mencionados, hay disponibles un amplio abanico de ellos, como son TCP Reno y TCP Tahoe, pero no son los únicos<sup>1</sup>.

---

<sup>1</sup> Wikipedia. (2020). TCP congestion control, de Wikipedia. Sitio web: [https://en.wikipedia.org/wiki/TCP\\_congestion\\_control#Algorithms](https://en.wikipedia.org/wiki/TCP_congestion_control#Algorithms)