

# Apuntes de websockets que debes de saber antes de usarlo.



Autor:

Freddy Alcarazo Ibáñez.

[www.github.com/alacarazolabs](http://www.github.com/alacarazolabs)

Los websockets son una sesión interactiva entre el navegador del usuario y un servidor. Se puede enviar mensajes al servidor y recibir respuestas basadas en eventos sin tener que sondear “pull” al servidor para obtener respuesta.

- ✓ Los websockets realizan comunicación full-duplex a través de una única conexión TCP.
- ✓ HTTP y Websockets están ubicados en la “capa de transporte” del modelo OSI.
- ✓ Esta diseñado para trabajar en los puertos 80 y 443 y con proxy.
- ✓ Los websockets se ven obligados a usar proxy TCP y no HTTP (Los proxys TCP no pueden inyectar encabezados, reescribir URL's o realizar muchas funciones que un proxy HTTP hace). El límite de conexiones para solicitudes es de alrededor de seis. Con websockets se tiene un limite de 50 conexiones, ejemplo 10 pestañas del navegador por 50 es igual a 500 conexiones.
- ✓ Los websockets no se multiplexan automáticamente a través de conexiones HTTP/2.
- ✓ La multiplexación de multiplexación personalizada tanto en el servidor y cliente es complicada.

## Los websockets y el equilibrio de carga (Load Balancing)

- ❖ No es posible mover conexiones de socket a un servidor diferente si el actual experimenta una carga alta. Las conexiones deben de cerrarse y reabrirse.

## Los websockets y el DOS

- ❖ El DOS se controla mediante proxys HTTP. Usando TCP no puede manejar esto ya que son necesarios para websockets. Se puede conectar al socket e inundar el servidor con datos. Websockets es vulnerable a todo tipo de ataques.

## Reinventar la rueda

- ❖ Con websockets uno debe manejar muchos problemas que con HTTP ya están resueltos.

## Casos de éxito de websockets

- ✓ Los casos de éxito de los websockets son los juegos multijugador y chats, aquí los beneficios superan los problemas. Su beneficio es la comunicación full-duplex.

## Impacto

- ❖ Alta sobrecarga operativa en términos de desarrollo y pruebas y escalado con polling y websockets.
- **Problemas en dispositivos móviles:**  
El diseño de hardware de estos dispositivos mantiene la conexión abierta y al mantener la antena viva y la conexión a la red telefónica se ocasiona una reducción en la vida útil de la batería, calentamientos y carga adicional por consumo de datos.

## ¿Cómo funciona una red telefónica?

Los dispositivos móviles tienen una antena de baja potencia que recibe datos de una celda, una vez que un dispositivo recibe datos de una llamada entrante se arranca la antena full-duplex para establecer la llamada. La misma antena se usa para hacer llamadas o conectarse a Internet (Si no hay wifi disponible).

La antena full-duplex necesita establecer una conexión a la red telefónica y realizar alguna autenticación, una vez que establece la conexión se crea una comunicación entre el teléfono y la red telefónica para poder realizar nuestra solicitud a la red, luego nos redirigen al proxy interno del proveedor de servicios móviles que maneja las solicitudes de internet y a partir de ese momento el procedimiento ya es conocido, pregunta un DNS esta la web [www.google.com](http://www.google.com), recibe la URI del recurso y finalmente es redirigido a él.

- ❖ El proceso consume mucha energía de la batería, por eso los proveedores telefónicos ofrecen un tiempo límite de conversación de unas horas.
- ❖ Sin wifi, tanto los websockets y polling “sondeo” requieren que la antena full-duplex funcione casi constantemente, por lo tanto se tiene un mayor consumo de datos, de energía y calor.