

ELO322: Redes de Computadores I 1 sem. 2016

---

# Informe: “Order & Chaos”

## WebSocket + HTML5

---

23 junio, 2016

Barbarita Lara 201021048 – Jonathan Tapia 2973109-8

LINK PROYECTO GITHUB: [https://github.com/darkmeow/order\\_and\\_chaos](https://github.com/darkmeow/order_and_chaos)

### Resumen:

Luego de tener el manejo de los socket y la masiva utilización del entorno web, fue necesario la creación de los websocket y la actualización de html a html5, para ayudar a la multiplexación de puerto y permitir las múltiples conexiones servidor-cliente, así como la mejora en cuanto al lenguaje universal web, ahora más intuitivo a nivel de código y amigable para el usuario.

Hemos aprovechado estas características para la creación de un juego y mostrar a través de su control el uso de websocket y html5.

Este juego es order y chaos, derivado del "gato", donde dos jugadores deben enfrentarse para mantener el orden o el caos.

# Introducción

El protocolo websocket permite mantener una conexión **full-duplex** y con **estado** entre cliente y servidor web. Originalmente pensada para protocolos textuales, esta tecnología puede ser extendida para soportar protocolos binarios.

Comenzó siendo parte de la especificación de HTML 5, pero ahora se mantiene como un estándar a cargo de IETF(Internet Engineering Task Force) y de la W3C(World Wide Web Consortium). Surge de la necesidad de reemplazar las técnicas de **Comet**<sup>1</sup>, especialmente para aquellas aplicaciones web que generan contenido en tiempo real, como pueden ser chats, bolsas de comercio, herramientas colaborativas o aplicaciones de subastas. Es la manera de brindar un marco estandarizado para estas aplicaciones, en vez de desarrollarlas sobre técnicas que son verdaderos parches para el protocolo. Para Google ha sido de gran impacto e importancia en su industria, además de incorporar la tecnología en su navegador Chrome, la compañía da soporte a un sitio que muestra a los desarrolladores la forma de ponerlo en práctica. Ian Hickson, que lidera las especificaciones de HTML 5 para Google, escribió lo siguiente: "La reducción de datos desde kilobytes hasta 2 bytes [...] y reducir la latencia de 150 milisegundos a 50 milisegundos, es mucho más que algo marginal. De hecho, estos dos factores por sí solos son suficientes para hacer que WebSockets sea interesante para Google".

El proyecto "Order and Chaos" juego basado en HTML5 pretende demostrar las características de este protocolo: comunicación bidireccional utilizando una única conexión tcp para ambas direcciones.

# WebSocket

Las metas principales del WebSocket son: Permitir tanto al cliente como al servidor, transmitir información en cualquier momento, utilizar una única conexión TCP para las dos direcciones y reducir el overhead producido por los encabezados HTTP.

## Funcionamiento WebSocket<sup>2</sup>

1. **Frame(Trama):** Unidad básica de intercambio de información para el protocolo.
2. **Message(Mensaje):** Es un bloque de datos interrelacionados con límites bien específicos. Un mensaje puede abarcar varios frames y es la unidad de datos a nivel de aplicación.
3. **WebSocket Handshake (Negociación):** Es el proceso por el cual cliente y servidor negocian la conexión a establecer.
4. **WebSocket Communication Channel ( Canal de comunicación ):** Es el canal de comunicación bidireccional entre cliente y servidor.
5. **WebSocket Sub-Protocol (Subprotocolo):** Es el subprotocolo negociado para utilizar en el canal de comunicación. Este protocolo especifica el framing a utilizar, la codificación, el timing, etc.

## Requerimientos WebSocket Cliente - Servidor

**Clientes:** El cliente debe ser capaz de establecer una conexión websocket mediante una negociación (handshake) bien definida; El protocolo debe proveer un método para cerrar una conexión cuando el cliente lo solicite; Al mismo tiempo, el protocolo debe soportar pérdidas de conexión y cierres abruptos de una conexión por parte del usuario; El cliente debe poder especificar al servidor un subprotocolo específico durante el handshake; Debe poder enviar y recibir tanto datos binarios como en texto plano

**Servidor:** El servidor que acepta peticiones de WebSocket por parte de un cliente debe utilizar un handshake bien definido; Debe poder enviar y recibir tanto datos binarios como en texto plano

El protocolo de websocket está diseñado para que haya la menor cantidad de datos de framing y así ahorrar espacio. Conceptualmente, websocket es una capa justo por encima de TCP que agrega:

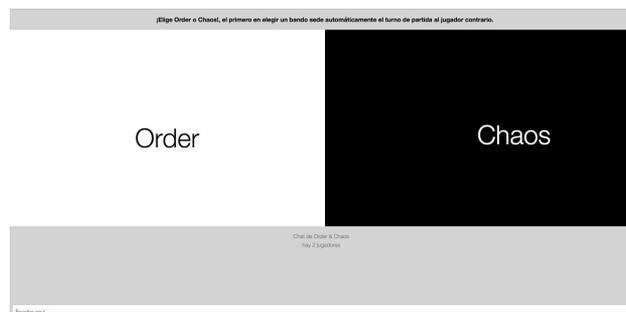
- Un modelo de seguridad basado en origen para los navegadores web
- Un mecanismo para soportar varios servicios en un mismo puerto y varios nombres de host en una misma dirección IP.
- Una capa de encima de TCP para trabajar con paquetes -a la manera de IP-, pero sin un límite en el tamaño.

El objetivo es darle al desarrollador algo lo más parecido a un socket TCP. La única relación entre WebSocket y HTTP es lo que los servidores web interpretan el handshake y hacen el upgrade al protocolo WebSocket. Por recomendación de IANA ( Internet Assigned Numbers Authority ) utiliza por default los puertos TCP/80 y TCP/443 para las conexiones sin cifrar y cifradas, respectivamente.

# Order & Chaos

.Este juego de mesa está basado en “tic-tac-toe” o más conocido en Chile como “gato”. La idea es que el jugador Order debe crear una línea vertical, horizontal o diagonal de 5 fichas iguales continuas y Chaos debe impedirlo, todo esto en un tablero de 6x6. Las piezas no se pueden mover una vez posicionadas en el tablero, es por esto, que al igual que el “gato”, se puede jugar incluso con un lápiz y papel.

En nuestra implementación hemos modificado levemente las reglas, cambiando la forma en que se inicia el juego, originalmente Order es el jugador que comienza, lo que fue modificado por una interfaz amigable en HTML5 donde se pregunta al jugador si quiere ser Order o Chaos, pero con la limitación de que en el momento que un jugador elija primero su rol, el otro jugador inmediatamente asume el rol contrario y comienza su turno.



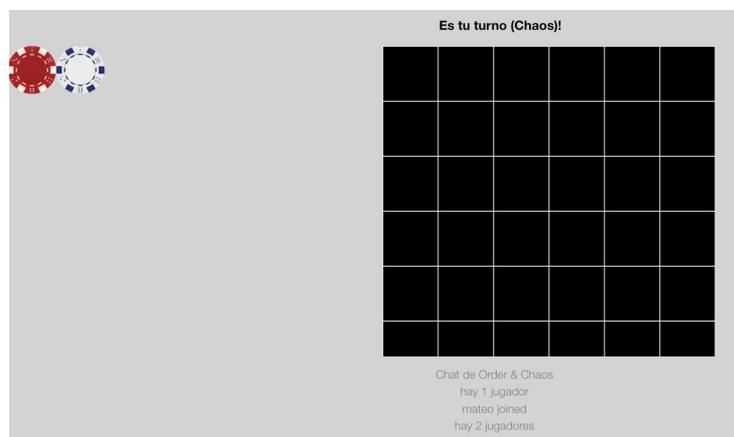
A través de PHPSocketIO,<sup>1</sup> Framework en PHP que facilita y abstrae la creación de socket a través de la implementación de WebSocket, hemos creado una interfaz Servidor-Cliente que permite controlar la multiplexación de puertos a través del puerto 2020 y el servidor Web en el puerto 7000.

---

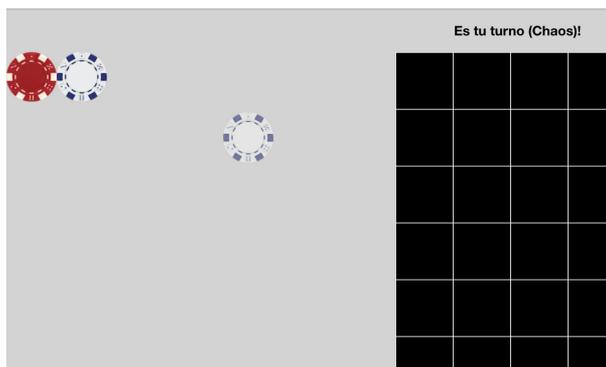
<sup>1</sup> [ref003]

```
meow@mail:~/orderandchaos$ php server.php start
Workerman[server.php] start in DEBUG mode
-----
WORKERMAN
-----
Workerman version:3.3.1      PHP version:5.6.20-0+deb8u1
-----
WORKERS
-----
user      worker      listen      processes  status
meow      PHPsocketIO  socketIO://0.0.0.0:2020  1           [OK]
meow      WebServer    http://0.0.0.0:7000     1           [OK]
-----
Press Ctrl-C to quit. Start success.
```

Luego de ingresar los dos jugadores podemos ver el siguiente escenario:

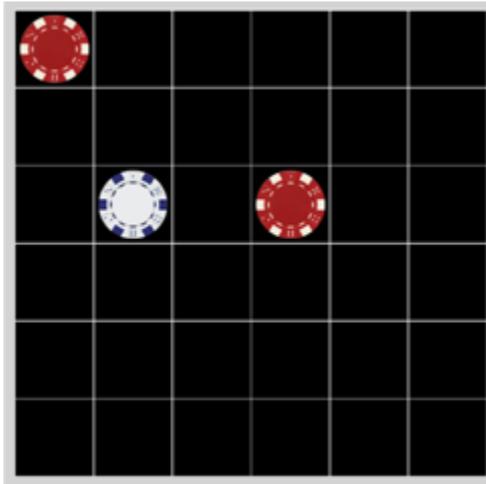


De esta misma forma, Order puede ver el tablero e incluso usar el chat implementado en la misma plataforma para comunicarse en tiempo real con el otro jugador.



Gracias a las características de HTML5, cada jugador puede arrastrar las fichas al tablero (drag and drop).

## ¿Cómo identificamos la posición de una ficha?



Las fichas azules se identifican con un 1 y las rojas con un 2, además los espacios vacíos son considerados como 0.

Internamente identificamos el posicionamiento de fichas como una matriz 6x6:

```
-
200000
000000
010200
000000
000000
000000
-
```

Esta matriz se actualiza en tiempo real al momento de posicionar nuevas fichas y llevamos el control de esto en nuestra interfaz.

## ¿Cómo identificamos a un ganador?

```
Array
(
  [0] => 200000
  [1] => 000000
  [2] => 010200
  [3] => 000000
  [4] => 000000
  [5] => 000000
  [6] => 200000
  [7] => 001000
  [8] => 000000
  [9] => 002000
  [10] => 000000
  [11] => 000000
  [12] => 000000
  [13] => 000200
  [14] => 00200
  [15] => 01000
  [16] => 200000
  [17] => 00200
)
```

Generamos un arreglo que se actualiza en tiempo real donde guardamos la posición de las fichas para buscar si se cumple la regla de 5 fichas iguales contiguas o bien, si gana Chaos.

Para encontrar efectivamente al ganador generamos un arreglo de strings con 18 posiciones de la siguiente forma:

- [0 - 5]: información horizontal del tablero
- [6 - 11]: información vertical del tablero
- [12 - 17]: información diagonal del tablero

Recorriendo este arreglo podemos encontrar al ganador.

## ¿Cómo se puede jugar en nuestro servidor?

Sólo es necesario que se conecten a la ip de nuestro servidor **23.92.30.236** o a la dirección **emercom.cl** a través de un navegador, pero apuntando al puerto 7000.

Es necesario acotar que debe estar corriendo la interfaz “servidor-cliente” y una vez terminado el juego debe reiniciarse.

# Referencias

[rf001] **Comet**: describe un modelo de aplicación web en el que una petición HTTP mantenida abierta permite a un servidor web enviar datos a un navegador por Tecnología Push, sin que el navegador los solicite explícitamente.

[rf002] **RFC The WebSocket protocol**: <https://tools.ietf.org/html/rfc6455>

[rf003] **PhpSocketIO** <https://github.com/walkor/phpsocket.io>

# Anexo

- Banchoff Tzancoff, Matías, “Websocket: comparación de performance e implementación de aplicaciones web”, Andres Barbieri, 2011.  
Disponble en: <http://sedici.unlp.edu.ar/handle/10915/47014>