

# **IES Gran Capitán**

## **Departamento de Informática**

Ciclo Formativo de Grado Superior de  
Administración de Sistemas Informáticos

---

### **Módulo de Proyecto Integrado**

**Andrés Gómez, Salvador Venzala – 2º ASIR**

**Proyecto: Aplicación Web con Websockets HTML5**

**Curso 2012 / 2013**

---

### **Índice de contenido**

- 1.- Introducción
- 2.- Objetivos y requisitos del proyecto
- 3.- Estudio previo
  - 3.1.- Estado actual
- 4.- Plan de trabajo
- 5.- Diseño
  - 5.1.- Diseño general
  - 5.2.- Diseño detallado
- 6.- Implantación
- 7.- Recursos
  - 7.1.- Herramientas hardware
  - 7.2.- Herramientas software
  - 7.3.- Personal
  - 7.4.- Presupuesto
- 8.- Conclusiones
  - 8.1.- Grado de consecución de objetivos
  - 8.2.- Problemas encontrados
  - 8.3.- Futuras mejoras
- 9.- Referencias / bibliografía

# 1.-Introducción

A medida que el tiempo pasa, lo hace también la tecnología, surgen nuevas herramientas, nuevas formas de hacer las cosas, es el caso de HTML5, que no es una evolución del anterior HTML sino que se trata de un nuevo jugador con muchísimas posibilidades de crear Web, aplicaciones y otras herramientas. A este nuevo jugador se le han añadido multitud de mejoras como la posibilidad de dibujar mediante el uso de canvas, crear reproductores de audio y video sin necesidad de plug-in de terceros, se ha adaptado al tiempo actual haciéndolo compatible con dispositivos móviles, tablets, smart tv e intentado lograr un funcionamiento igualitario entre todos los navegadores del mercado. Se trata de un lenguaje muy potente que con el uso añadido de CSS3 y JS se pueden crear proyectos muy completos. Una de las funcionalidades en desarrollo son los Websockets.

Se trata de una tecnología que proporciona un canal bidireccional y de forma simultánea diseñada para implementar en navegadores y servidores Web. Trabaja con TCP, en los puertos 80 para ws y 443 para wss permitiendo al igual que https crear conexiones cifradas. Este tipo de tecnología funciona en tiempo real y permite que 2 usuarios conectados al mismo sitio intercambien información de forma instantánea, como ejemplos se puede usar para chat, juegos, pizarras o incluso generar un acuario digital usando JS y WebGL.

Para que esto sea posible existe detrás de este lenguaje una API (Interfaz de Programación de Aplicaciones). Existen multitud de API's para HTML5, pero la que nos ocupa más interés en el proyecto es la de Websockets, vamos a ampliar algo mas de información sobre ella, pero antes vamos a presentar algunas APIs de HTML5 y algunas más concretamente sobre comunicación.

Algunas de las nuevas API permiten realizar tareas que anteriormente habría que hacerlas con otros lenguajes y luego incorporarlas a HTML, veamos algunas de ellas.

- **Drag and Drop:** Permite realizar acciones de arrastrar y soltar.
- **Geolocalización:** Permite mediante el servicio de Google IP-Geocoding, desarrollar características de geolocalización en nuestra aplicación.
- **History:** La API de historia provee a Javascript una forma de cambiar la URL mostrada en el navegador sin recargar la página.
- **IndexedDB:** Es un almacén de datos persistente gestionado por el navegador, es decir una base de datos del lado del cliente.
- **localStorage:** Permite junto con sessionStorage guardar datos sin tener que utilizar cookies.
- **matchMedia:** Nos permite comprobar si el contenido multimedia será soportado por el navegador, útil para trabajar en sitios responsive design.
- **WebGL:** Hace posible crear gráficos en tiempo real, 2D y 3D.

Por otro lado las API de comunicaciones nos permiten el envío de datos entre los programas de manera asíncrona, veamos algunas de estas API.

- **Cross Document Messaging:** Permite la comunicación segura a través de las páginas web.
- **Channel Messaging:** Utiliza dos puertos para mensajes de muchos a muchos, puerto bidireccionales.

En resumen, aunque se trata de un proyecto en el que es fundamental la investigación, unificando las tecnologías de HTML5, los gráficos de CSS3, la API de Websockets junto con el polyfill Socket.io, podremos conseguir una aplicación muy interesante que como ya hemos explicado, la experiencia de usuario será en tiempo real e interactiva y atractiva visualmente.

## 2.-Objetivos y requisitos del proyecto

El problema que se nos presenta trata de, mediante el uso de un servidor Web, implementamos una conexión entre como mínimo 2 navegadores, creando una comunicación bidireccional y full-duplex, usando para ello Websockets e implementando una aplicación desarrollada con HTML5, CSS3 que puede ser un juego.

Objetivos:

- Profundizar en el conocimiento de HTML5
  - Estudio de las nuevas etiquetas y mejoras del nuevo estándar.
- Presentar las nuevas API de HTML5
  - Estudio de las API orientadas a la comunicación.
  - Estudio concreto de la API de Websockets.
- Elaborar un sitio que recogerá tanto la información para el estudio como la documentación técnica para poder elaborar este proyecto.
- Desarrollar una aplicación utilizando API Websockets y HTML5.
  - Utilizar las tecnologías estudiadas anteriormente.
  - Estudio de la base de datos de preguntas y respuestas desarrollada por alumnos del ciclo.
  - Diseño gráfico de la aplicación, usando criterios de usabilidad.
- Implantación de la aplicación.
- Puesta en producción.

Requisitos:

- Conexión a la red.
- Servidor aplicación compuesto por:
  - NodeJS
    - Express
    - Socket.io
    - Ejs
    - MySQL
- Servidor Base de Datos:
  - MySQL.
    - Gestor BBDD.
- Lenguajes de programación e interpretación:
  - HTML5
  - CSS3
    - Framework Bootstrap
  - JAVASCRIPT
    - NODE.JS
    - SOCKET.IO
- Disponibilidad equipos con cualquier sistema operativo.
  - Navegadores Web compatibles con Websockets.

### 3.- Estudio previo

#### 3.1.- Estado actual

Actualmente en el centro no existe ninguna solución implantada que utilice el nuevo estándar de HTML5 y que use una API de comunicación como es Websockets y tenga implantada sobre esto una aplicación online de preguntas y respuestas. Nuestra aplicación intentará proporcionar una solución para que en un futuro pueda ser aplicada a una nueva aplicación de forma funcional, utilizando esta tecnología en cualquier proyecto educativo que quiera realizar en el centro.

### 4.- Plan de trabajo

#### Abril:

<b>1</b> <i>Fase Investigación</i>	<b>2 Martes</b> <i>Fase Investigación</i>	<b>3</b> <i>Fase Investigación</i>	<b>4</b> <i>Fase Investigación</i>	<b>5</b> <i>Fase Investigación</i>
<b>8</b> <i>Fase Investigación</i>	<b>9 Martes</b> <i>Fase Investigación</i>	<b>10</b> <i>Fase Investigación</i>	<b>11</b> <i>Fase Investigación</i>	<b>12</b> <i>Fase Investigación</i>
<b>15</b> <i>Fase Investigación</i>	<b>16 Martes</b> <i>Fase Investigación</i>	<b>17</b> <i>Fase Investigación</i>	<b>18</b> <i>Fase Investigación</i>	<b>19</b> <i>Fase Investigación</i>
<b>22</b> <i>Fase Investigación</i>	<b>23 Martes</b> <i>Fase Investigación</i>	<b>24</b> <i>Fase Investigación</i>	<b>25</b> <i>Fase Investigación</i>	<b>26</b> <i>Fase Investigación</i>
<b>29</b> <i>Fase Investigación</i>	<b>30 Martes</b> <i>Fase Investigación</i>			

#### Mayo:

		<b>1</b> <i>Fase Desarrollo</i> - Desarrollo del Websockets - Gráficos	<b>2</b> <i>Fase Desarrollo</i> - Desarrollo del Websockets - Gráficos	<b>3</b> <i>Fase Desarrollo</i> - Desarrollo del Websockets - Gráficos
<b>6</b> <i>Fase Desarrollo</i> - Desarrollo Websockets - Gráficos	<b>7 Martes</b> <i>Fase Desarrollo</i> -Desarrollo Websockets - Entrega BBDD	<b>8</b> <i>Fase Desarrollo</i> -Desarrollo Websockets - Estudio BBDD	<b>9</b> <i>Fase Desarrollo</i> -Desarrollo Websockets - Estudio BBDD	<b>10</b> <i>Fase Desarrollo</i> -Desarrollo Websockets - Estudio BBDD
<b>13</b> <i>Fase Desarrollo</i> -Desarrollo Websockets - Estudio BBDD	<b>14 Martes</b> <i>Fase Desarrollo</i> -Desarrollo Websockets - Desarrollo de App	<b>15</b> <i>Fase Desarrollo</i> - Desarrollo de App	<b>16</b> <i>Fase Desarrollo</i> - Desarrollo de App	<b>17</b> <i>Fase Desarrollo</i> - Desarrollo de App

<b>20</b> <i>Fase Desarrollo</i> - Desarrollo de App	<b>21 Martes</b> <i>Fase Desarrollo</i> - Desarrollo de App	<b>22</b> <i>Fase Desarrollo</i> - Desarrollo de App	<b>23</b> <i>Fase Desarrollo</i> - Desarrollo de App	<b>24</b> <i>Fase Desarrollo</i> - Desarrollo de App
<b>27</b> <i>Fase Desarrollo</i> - Desarrollo de App	<b>28 Martes</b> <i>Fase Desarrollo</i> - Desarrollo de App	<b>29</b> <i>Implementación</i>	<b>30</b> <i>Implementación</i>	<b>31</b> <i>Implementación</i>

## Junio:

<b>3</b> <i>Implementación</i>	<b>4 Martes</b> <i>Implementación</i>	<b>5</b> <i>Corrección</i>	<b>6</b> <i>Corrección</i>	<b>7</b> <i>Corrección</i>
<b>10</b> <i>Corrección</i>	<b>11 Martes</b> <i>Corrección</i>	<b>12</b> <i>En Producción.</i>	<b>13</b> <i>En Producción.</i>	<b>14</b> <i>En Producción.</i>
<b>17</b> <i>En Producción.</i>	<b>18 Martes</b> <i>Entrega</i>			

## 5.- Diseño

### 5.1.- Diseño general

La solución propuesta consiste en que usando las tecnologías estudiadas implantemos un servidor con los requisitos necesarios para funcionar. Dicho servidor debe de ser capaz de interpretar una aplicación desarrollada por nosotros, la cual se utiliza como nexo de comunicación entre ambos extremos de los Websockets que estudiaremos.

### 5.2.- Diseño detallado

#### Parte del servidor:

Se ejecutará el servidor de Websockets, este contendrá los protocolos que harán posible la conexión entre los usuarios, proporcionará una dirección desde la cual los usuarios podrán acceder para usar la aplicación, mientras tanto estará a la espera de que alguien se conecte.

Una vez el usuario se conecte a la dirección indicada, el servidor creará la partida del juego y comenzará a generar las preguntas que serán escogidas de manera aleatoria desde la base de datos. A medida que se conecten nuevos jugadores se le enviará la pregunta generada.

#### Parte del cliente:

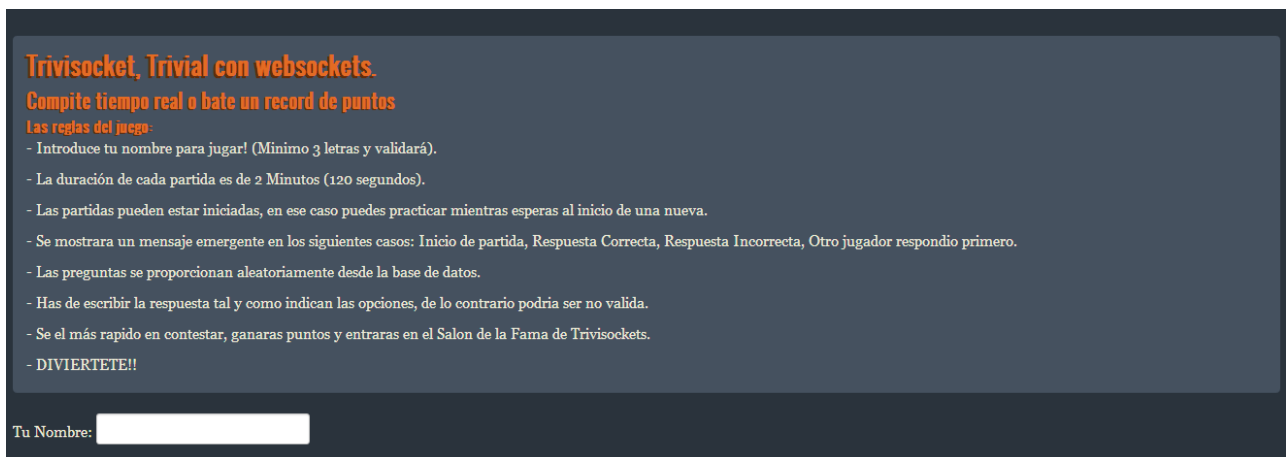
Una vez el jugador acceda a la dirección en la que se encuentra implantada la aplicación se encontrará con la pagina inicial que indicara por un lado el estado en que se encuentra el servidor de Websockets pudiendo ser Conectado o Desconectado. Además de unas instrucciones sobre el juego y una casilla en la que deberá introducir su nombre. (Dicha casilla cumple la función de al menos 3 caracteres, hecho esto pasara a la pagina del juego).



Captura detalle Estado Websockets Conectado.

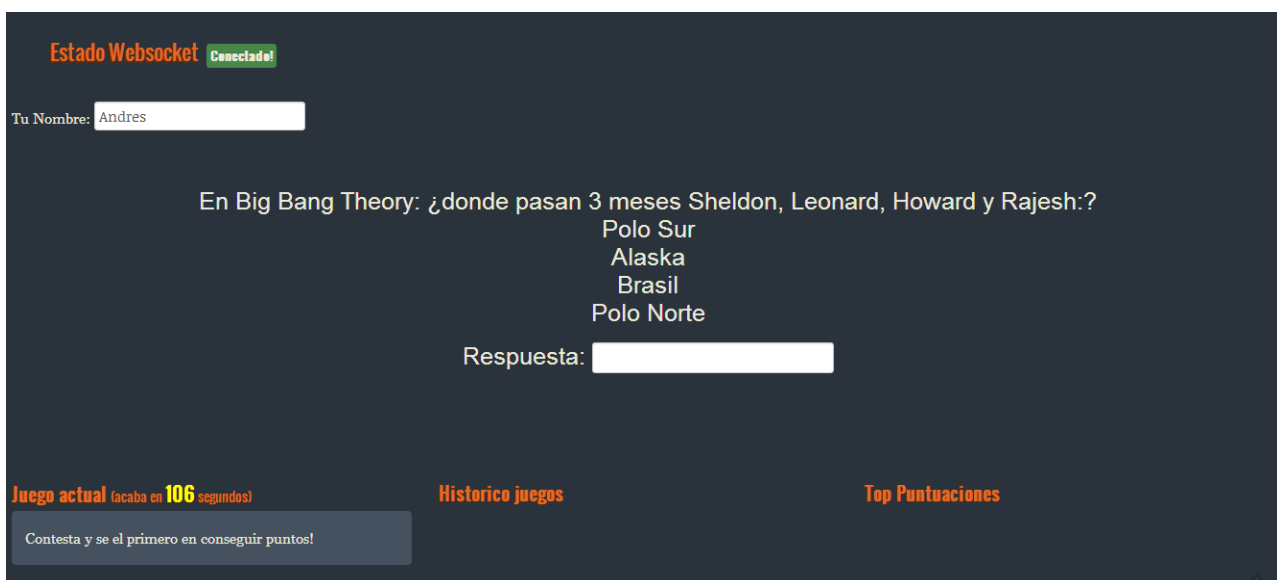


Captura detalle Estado Websockets Desconectado.



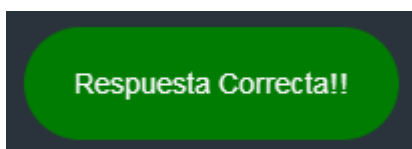
Captura detalle página inicial, Título, reglas y casilla para introducir el nombre del jugador.

Una vez el jugador introduzca su nombre pasara a la página del juego, en la que encontrara una pregunta con sus correspondientes opciones de respuesta, una casilla donde deberá introducir la respuesta correcta.

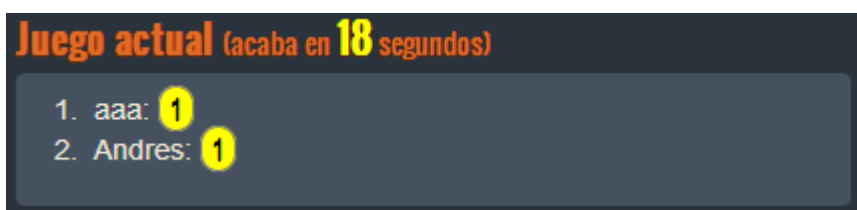


Captura detalle pagina de juego con pregunta y respuesta, tiempo de juego actual, histórico y top de puntos.

En el juego existen varias interacciones con el juego, en caso de que el jugador introduzca una respuesta correcta se le mostrará un mensaje que informará que acertó y se añadirá su nombre y el número de puntos que acumula.

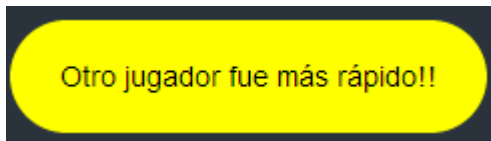


Captura detalla Respuesta Correcta.



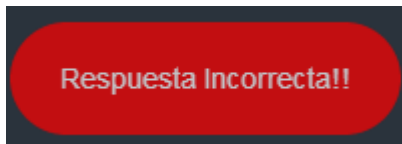
Captura detalle Juego actual, puntos, tiempo restante.

En caso de que caso de que sea otro jugador el que acierte dicha pregunta se mostrará un mensaje de que otro fue más rápido y se añadirá su nombre y número de puntos en el apartado de Juego actual, como se muestra en la captura anterior.



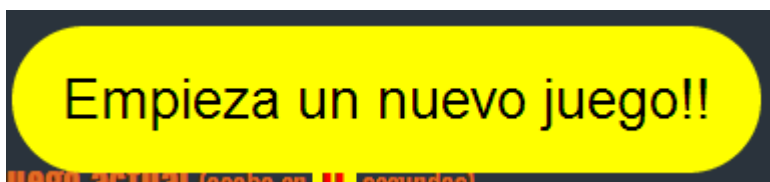
Captura detalle Otro jugador respondió antes.

En caso de que la respuesta sea incorrecta se mostrará un aviso que informara que hemos fallado.

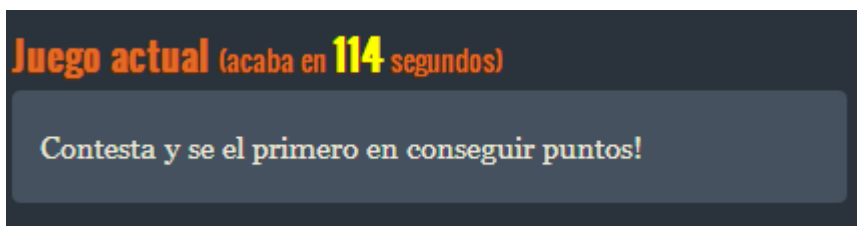


Captura detalle Respuesta incorrecta.

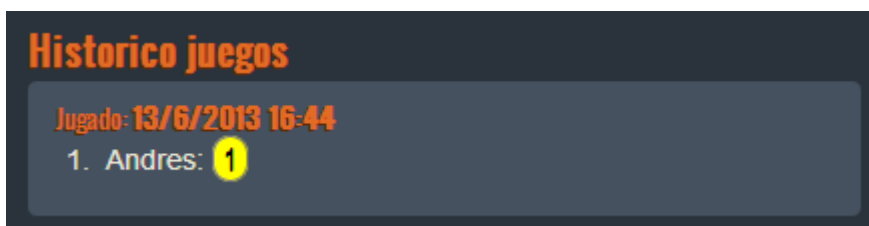
Cuando pase el tiempo estipulado para cada juego (2 minutos) comenzará un nuevo juego, mostrando otro mensaje de alerta, se reinician las puntuaciones del juego actual, se mostrará un histórico de los juegos anteriores con información sobre día, hora, jugadores y puntos. Además se mostrará un Top puntuaciones con las 10 mejores puntuaciones, fecha y hora de cuando se produjo.



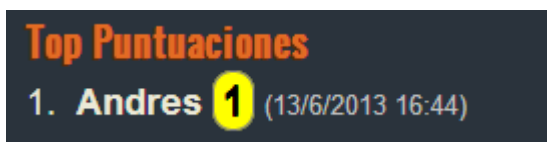
Captura detalle Nuevo juego.



Captura detalle Reseteo de puntos juego actual.



Captura detalle Histórico de juegos anteriores.



Captura detalle Top puntuaciones, con nombre y detalles.

En caso de que el servidor se desconectara se bloqueara el juego y mostrará el estado en Desconectado. Para la desconexión del usuario simplemente cerraremos el navegador.

## 6.- Implantación

Para implantar este proyecto necesitamos una maquina con cualquiera de estos sistemas: Windows, Linux, Mac. Comenzamos con la instalación de NodeJS, para ello desde Windows o Mac descargaremos el instalador, se instala sin complicación, para Linux desde el terminal basta con desde el terminal ejecutar la orden:

***“sudo apt-get install nodejs”***

Una vez hecho este paso, en Windows debemos añadir una variable de entorno al sistema llamada NODE\_PATH y localizada en C:\Program Files\nodejs. En el resto de sistemas se añaden automáticamente.

A continuación debemos instalar los paquetes dependientes de la aplicación, node instala además una herramienta llamada npm que nos permite instalar multitud de paquetes con los que node interacciona. En nuestro caso son 3 los paquetes que debemos instalar, no importa el orden pero si es imprescindible que se haga en la carpeta en la cual tenemos la aplicación, desde la terminal o desde el símbolo de sistema, nos movemos a la carpeta y realizamos la instalación de la siguiente forma.

***“npm install ejb”***

***“npm install express”***

***“npm install mysql@2.0.0-alpha8”***

***“npm install socket.io”***

Hecho estos pasos, simplemente escribimos la siguiente orden ***“node server.js”*** El terminal comenzará a devolver líneas la primera nos informa de que estamos trabajando con Websockets y la segunda que el servidor se encuentra en ejecución y el número de puerto por el que lo hace. Abrimos el navegador y nos conectamos a la dirección en este caso [http://IP\\_Servidor:Puerto](http://IP_Servidor:Puerto)

En el momento que el jugador se conecta el terminal comienza a escribir información del mismo, como id, nombre que introduzca, etc.



## **7.- Recursos**

### **7.1.- Herramientas hardware**

En cuanto a requisitos hardware no es necesario un gran equipo o servidor, pero si lo suficiente para que soporte las conexiones de los usuarios que lleguen a conectarse.

### **7.2.- Herramientas software**

Sobre herramientas software necesitaremos:

- Sistema Operativo
  - Linux, Mac o Windows.
- Navegador Web
  - Chrome, Firefox, Opera
- Servidor MySQL
- Servidor Aplicación
  - Node
    - Paquete Express
    - Paquete EJS
    - Paquete Socket.io
    - Paquete MySQL
- Editor Gedit

### **7.3.- Personal**

El personal que ha participado en la investigación, estudio, desarrollo y testeo de la aplicación se compone de 2 personas.

- Andrés J. Gómez
- Salvador Venzala

### **7.4.- Presupuesto**

El presupuesto para el desarrollo de este proyecto es prácticamente 0€ si usamos un sistema operativo gratuito como por ejemplo Ubuntu, el editor de texto podemos usar cualquiera en este caso se ha usado gedit, también gratuito, al igual que los navegadores web y el servidor de la aplicación y los paquetes que son totalmente gratuitos.

## 8.- Conclusiones

### 8.1.- Grado de consecución de objetivos

Objetivos:

- Profundizar en el conocimiento de HTML5
  - Estudio de las nuevas etiquetas y mejoras del nuevo estándar.
  - Presentar las nuevas API de HTML5
  - Estudio de las API orientadas a la comunicación
  - Estudio concreto de la API de Websockets.
  - **Realizado**
- Elaborar un sitio que recogerá tanto la información para el estudio como la documentación técnica para poder elaborar este proyecto.
  - **Realizado**
- Estudio de la base de datos de preguntas y respuestas desarrollada por alumnos del ciclo.
  - **Realizado**
- Desarrollar una aplicación utilizando API Websockets y HTML5.
  - Utilizar las tecnologías estudiadas anteriormente.
  - Diseño gráfico de la aplicación, usando criterios de usabilidad.
  - **Realizado**
- Implantación de la aplicación.
  - **Realizado**
- Puesta en producción.
  - **Realizado**

### 8.2.- Problemas encontrados

- Desarrollo de una aplicación responsive: para ello la solución implantada ha sido el uso de un framework css como es Bootstrap.
- Generación de gráficos: la intención era mediante imágenes insertar un tablero con fichas y movimiento, pero ante la imposibilidad generamos los gráficos mediante CSS3.

### 8.3.- Futuras mejoras

- Implementación del sistema de turnos.
- Implementación de los gráficos con animación de las fichas.
- Corrección errores como mostrar la primera pregunta que actualmente se muestra como vacía.
- Implementación de un sistema para finalizar el juego por completo.

## 9.- Referencias / bibliografía

Información general, API sobre Websockets:

- [RFC Websockets](#)
- [Presentación sobre API Websockets HTML5](#)
- [Tutorial de instalacion de websockets con XAMPP](#)
- [HTML5 con Websockets PHP](#)
- [Websockets en HTML5](#)
- [JQuery Graceful Websockets: Plugin JQuery para Websockets](#)
- [Websockets Tutorial](#)
- [Introducción a los Websockets](#)
- [Websockets API](#)
- <http://www.websocket.org/index.html>
- [Node-Mysql](#)

Libros:

- [El gran libro de HTML5, CSS3 y Javascript.](#)
- [El Manual de AJAX](#)
- [Introducción a Javascript](#)
- [Introducción a AJAX](#)

Ejemplos:

- [Echo Test](#)
- [Kaazing.me](#)
- [Pizarra Multiusuario](#)
- [Wordsquare](#)
- [Acuario WebGL](#)
- [Video Sincronizado con Websockets](#)
- [Rumpetroll](#)
- [BrowserQuest](#)

Diario de proyecto:

- [Websockets 2 ASIR](#)