

## PROGRAMACIÓN HIPERMEDIA II. PRÁCTICA 3.

### Objetivos de la práctica

- Aprender a utilizar los nuevos APIs de HTML.
- Aprender a integrar todos los conocimientos adquiridos durante el curso en una única aplicación web y conseguir que interactúen entre sí de la manera más eficiente y productiva posible.

### Enunciado de la práctica

Esta práctica es independiente de las dos anteriores aunque, por simplicidad, utiliza parte del servidor *rest* de la práctica 2 y la misma base de datos. Se recomienda crear una carpeta nueva para la práctica 3 (por ejemplo, *practica3*) y copiar en ella la carpeta *rest* que se os proporciona y en la que se encuentran todos los ficheros necesarios de la parte del servidor para realizar la práctica. Recordad que debéis configurar el fichero *.htaccess* y *configbd.php* correctamente, tal y como se explicó en la práctica 2, para que hacer que el servidor *rest* funcione en esta práctica 3.

En esta práctica deberéis implementar el conocido juego “Hundir la flota” o también conocido como “Batalla naval” ([https://es.wikipedia.org/wiki/Batalla\\_naval\\_%28juego%29](https://es.wikipedia.org/wiki/Batalla_naval_%28juego%29)). Para esta práctica tendréis que implementar únicamente la parte del cliente, es decir la parte que ve el jugador. **La parte del servidor está implementada en los ficheros del servicio *rest* que se os proporcionan.** Al final del enunciado se os indican las peticiones, tanto GET como POST, que se pueden hacer, así como los parámetros que admiten. También **se os proporciona un script sql** que se encarga de crear en la misma base de datos que se utiliza para la práctica 2, las nuevas tablas que se necesitan para el juego. Simplemente debéis importarlo a la misma base de datos *ph2* que utilizasteis para la práctica 2.

**No se permite utilizar ningún framework, ni ninguna otra herramienta que no se haya visto en la asignatura. Tampoco se permitirán implementaciones de este juego que no se hayan realizado como se indica en este enunciado.**

El sitio web constará de las siguientes páginas web:

- ***index.html***. Esta será la página inicial de la práctica y en ella aparecerá una clasificación con las puntuaciones de los jugadores, ordenada por partidas jugadas en orden descendente y, a igualdad de partidas jugadas, por victorias también en orden descendente.
- ***juego.html***. Página principal de juego que contendrá todos los elementos necesarios para el desarrollo del mismo. A esta página sólo se podrá acceder si se está logueado.
- ***acerca.html***. Al igual que en el resto de las prácticas, esta página contendrá la información sobre el autor o autores de la práctica, así como también cualquier otra información relacionada con el desarrollo de la misma: partes realizadas y/o

---

sin realizar; problemas encontrados y solución encontrada; etc..

**Tareas a realizar:**

- 1) (0,25 puntos) Todas las páginas debe pasar satisfactoriamente y en todo momento la validación HTML en <http://validator.w3.org>. El html a validar incluirá el contenido generado con JavaScript.
- 2) (0,25 puntos) El diseño de las páginas se hará utilizando la técnica *Mobile First* y deberá tener, como mínimo, una cabecera en la que aparezca el nombre del juego y un pie de página que contendrá, al menos, un enlace a la página *acerca.html*.
- 3) (0,25 puntos) Todos los mensajes que se deban mostrar al usuario se mostrarán mediante mensaje emergente o parecido, nunca con `alert()`. El mensaje deberá tener un botón para cerrarlo.
- 4) Página ***index.html***. Mostrará, al menos, lo siguiente:
  - a) (0,5 puntos) **Clasificación de puntuaciones.** Clasificación de puntuaciones obtenidas por los jugadores en el juego. Cada fila de la clasificación mostrará el login del jugador, el número de partidas jugadas, número de partidas ganadas y porcentaje de victorias. Para obtener esta información se realizará una petición GET ajax al servidor (*rest/clasificacion/*) a la que se le pasará un parámetro *c* con el número de puntuaciones que se quiera.
  - b) **Opciones para login, registro y jugar.** Estas opciones pueden implementarse mediante un menú, botones o lo que consideréis oportuno.
    - i) (0,25 puntos) Debéis tener en cuenta que inicialmente, y si el usuario no está logueado, sólo se podrá realizar login y registro. Si el usuario está logueado, la única opción disponible que tendrá será la de jugar, que le llevará a la página *juego.html*.
    - ii) (0,5 puntos) Para hacer login y registro se mostrarán los formularios correspondientes. Las peticiones ajax a realizar son las mismas que en la práctica 2 (están explicadas en el enunciado de la práctica 2). Se recomienda reutilizar los formularios de login y registro de la práctica 2 pero incorporándolos a esta práctica 3 (no se podrá redireccionar a las páginas de la práctica 2). Para ello se mostrará una capa semitransparente sobre la página y en esta capa semitransparente aparecerá el formulario correspondiente. La única diferencia es que en esta práctica 3, y por simplificar, no hay que pedir la foto del usuario en el registro.
      - 1) Al hacer login, deberá guardarse en `sessionStorage`, al menos, el login del usuario (para saber que está logueado) y en la cabecera de la página deberá aparecer un texto con el login del usuario logueado junto a un enlace/botón/etc que

le permita hacer **logout**. Al pinchar en este botón de logout, se limpiará la información del juego almacenada en `sessionStorage` y se redirigirá a *index.html*.

- 2) Al hacer registro, si todo ha ido bien, se almacenará el login en `sessionStorage` (acción de login) y se mostrará un mensaje de bienvenida al usuario con un botón para cerrarlo. Cuando se cierre el mensaje, el usuario será redirigido a *juego.html*.

5) Página **juego.html**. Esta es la página principal de juego. A esta página no se podrá acceder si previamente no se ha logueado el usuario.

- a) Si al cargar la página el usuario no está logueado se debe redirigir automáticamente a *index.html*.
- b) Al cargar la página mostrará lo siguiente:

- i) (0,5 puntos) **Tablero de juego del usuario**. Será un **canvas** con el tamaño necesario para mostrar una rejilla de 11x11 celdas que representará el tablero de juego del usuario en el que colocará los barcos. Se recomienda que el tamaño de las celdas no sea muy grande (por ejemplo 28x28 píxeles) ya que se debe poder ver todo el tablero en tamaños de pantalla de móvil. Esta zona tendrá un título "Posición de tus barcos" o similar. La primera columna mostrará las letras de las filas de la A a la J y la primera fila mostrará los números de columna de la 1 a la 10, de la siguiente manera:

	1	2	3	4	5	6	7	8	9	10
A										
B										
C										
D										
E										
F										
G										
H										
I										
J										

- ii) (0,5 puntos) **Zona de barcos disponibles**. Zona en la que aparecerán los barcos disponibles para el usuario, inicialmente en posición horizontal (rotados 0°). En total, el usuario dispone de diez barcos: uno de tamaño 4; dos de tamaño 3; tres de tamaño 2; y cuatro de tamaño 1. Se recomienda implementar estos barcos también como canvas para poder reutilizar el código que dibuja los barcos en el canvas principal. Esta zona tendrá un título "Arrastra tus barcos a la cuadrícula", o similar.

- iii) (0,25 puntos) **Zona de botones de acción**. Junto al canvas

aparecerá una zona con los siguientes botones: uno para empezar a jugar y otro para rotar un barco. Inicialmente, y hasta que no estén todos los barcos colocados en el tablero de juego, el botón para empezar a jugar estará deshabilitado. El botón para rotar barco sólo estará habilitado cuando haya un barco seleccionado en el canvas.

6) **acerca.html**. Página que mostrará la siguiente información:

- a) Texto indicando que se trata de la práctica 3 de la asignatura Programación Hipermedia II, que pertenece al 3er curso de la titulación de Ingeniería Multimedia.
- b) Asimismo, aparecerá la información relativa al autor o autores de la práctica (dni, nombre, apellidos, grupo de prácticas y correo electrónico).

7) **Funcionamiento del juego:**

- a) El juego no comenzará hasta que el usuario haya colocado todos los barcos en la cuadrícula y esto lo podrá realizar de dos maneras:
  - i) (1 punto) Primera: seleccionando el barco en la zona de barcos disponibles haciendo click sobre él. Una vez seleccionado el barco, el usuario hará click en la celda en la que desee colocarlo. Esa celda será la primera de la nueva ubicación y el barco se extenderá hacia la derecha, siempre y cuando la posición sea correcta. Más adelante se explicará cuándo se considera correcta la ubicación de un barco.
    - 1) Al seleccionar el barco en la zona de barcos disponibles, éste debe destacarse de los demás de alguna forma, por ejemplo, poniendo su borde de otro color.
    - 2) Si se pincha en otro barco de la zona de barcos disponibles, éste pasará a ser el barco seleccionado. En todo momento, como mucho sólo puede existir un barco seleccionado.
    - 3) Al pinchar en una celda del canvas, primero se comprobará si la ubicación del barco es correcta. Si lo es, el barco se dibujará en la cuadrícula y se eliminará de la zona de barcos disponibles. Si la ubicación del barco no es correcta, no se hará nada.
  - ii) (1 punto) Segunda: utilizando drag & drop. Para ello, los barcos de la zona de barcos disponibles deberán tener el atributo *draggable* a *true*. Tendréis que implementar el código necesario para que se puedan arrastrar al canvas. Al *soltar* un barco sobre el canvas, se calculará la posición en la que se ubicará y se deberá realizar el mismo proceso que se indica en 7.a.iii.
    - 1) Al arrastrar (sin soltar) el barco sobre el canvas se deberá resaltar en la cuadrícula (por ejemplo, dibujando el borde exterior del área de celdas que ocuparía) la posición en la que se ubicaría el barco. Si la posición es correcta, el color a

utilizar para resaltar sería verde (#0a0) y si no fuera correcta se utilizaría color rojo (#a00).

- b) (0,5 puntos) **Rotar barcos en canvas.** Los barcos colocados en el canvas se podrán rotar siempre y cuando no haya empezado el juego. Para rotar un barco en el canvas:
- i) primero habrá que pinchar sobre él para seleccionarlo. El barco seleccionado en el canvas se resaltará de alguna manera, por ejemplo pintando en rojo (#a00) el borde exterior del área de celdas que ocupa. Si se pincha en otro barco del canvas, este barco pasará a ser el seleccionado. Como mucho sólo puede haber un barco seleccionado en el canvas. Si se pincha en una celda que no hay barco, el barco que estuviera seleccionado dejará de estarlo. El botón de rotar barco deberá estar habilitado siempre que haya un barco seleccionado en el canvas y deshabilitado en caso contrario.
  - ii) Una vez seleccionado el barco en el canvas, se podrá rotar pinchando el botón de rotar barco. Las posibles rotaciones de un barco serán 0° (horizontal) y -90° (vertical hacia arriba). Al pinchar sobre el botón de rotar, se deberá comprobar si con la nueva rotación la ubicación del barco es correcta. Si lo es, se realizará la rotación y el barco se dibujará en el canvas en la nueva ubicación. Si la nueva ubicación no fuera correcta, no se haría nada.
- c) (0,5 puntos) **Modificar ubicación de los barcos en el canvas.** Mientras no haya comenzado el juego, el usuario tendrá la posibilidad de cambiar la ubicación de sus barcos en el tablero, bien sea moviéndolos o rotándolos.
- i) Para la rotación de los barcos se aplica lo indicado en 7.b.
  - ii) Para mover los barcos en el canvas. Primero se ha de seleccionar el barco a mover mediante click de ratón sobre él y, a continuación, se hará click en la celda destino. Una vez hecho click en la celda destino, se procederá de forma similar a lo que se indica en 7.a.i.3 (salvo la parte de su eliminación de barcos disponibles), es decir, se comprobará si la nueva ubicación del barco (teniendo en cuenta su rotación) es correcta o no. En caso de ser correcta se procederá a moverlo, y en caso de no ser correcta no se hará nada.
- d) (0,75 puntos) **Empezar a jugar.** Sólo se podrá empezar a jugar cuando todos los barcos estén correctamente colocados en el tablero de juego (canvas).
- i) Una vez colocados todos los barcos en el tablero de juego, el botón de empezar juego pasará a estar habilitado. Mientras el usuario no pinche en el botón de empezar juego, podrá seguir modificando la ubicación de sus barcos en el tablero. Cuando se haga click sobre el botón de empezar juego se hará lo siguiente:
    - 1) Se ocultarán todos los botones de la zona de botones de acción del tablero de juego.
    - 2) Aunque el usuario haga click sobre el tablero de juego, ya no se seleccionará ningún barco, ni se desencadenará ninguna otra acción.

- 3) Se deberá realizar una petición POST ajax al servidor (*rest/juego/empezar/*) con el parámetro *login* cuyo valor es el login del usuario registrado. Si todo ha ido bien, el servidor habrá preparado en la base de datos sus barcos y estará preparado para jugar (más adelante se explica cómo).
- 4) Se eliminará la zona de barcos disponibles (que estará vacía) y se mostrará otra cuadrícula (canvas) igual a la de la zona de juego en la que el usuario podrá anotar sus disparos. El título de esta zona cambiará y será algo similar a "Tus disparos". Junto a esta cuadrícula se mostrará un botón que permitirá obtener el disparo del enemigo cuando sea su turno. Inicialmente, este botón estará deshabilitado.

Una vez se ha pinchado en el botón de empezar a jugar y se han realizado todas las acciones indicadas en 7.d) comienza el juego, siendo el usuario el primero en *disparar*.

- e) (1 punto) **Disparo de usuario**. El usuario podrá realizar un disparo cuando sea su turno haciendo click en la celda correspondiente de la rejilla del canvas de disparos. Para ello:
  - i) Al mover el ratón por encima del canvas de disparos, la celda sobre la que se encuentre el ratón se resaltará pintando su borde de color azul (#00a).
  - ii) Al hacer click sobre la celda, si ésta ya tiene un disparo anotado no se hará nada. Si no tiene disparo anotado, se hará una petición POST ajax al servidor (*rest/juego/disparo/jugador/*) pasándole los parámetros *login*, *columna* y *fila*, cada uno con el valor correspondiente. Los valores de los parámetros de *columna* y *fila* son numéricos del 1 al 10. Si la respuesta del servidor indica que se ha alcanzado un barco del enemigo (servidor), el usuario tiene derecho a disparar de nuevo. Si el resultado del disparo ha sido AGUA, el turno pasa al enemigo (servidor) y el usuario no podrá interactuar (hacer click) con el canvas de disparos. Más adelante se explican las distintas respuestas del servidor.
  - iii) El disparo se marcará en el canvas de disparos, distinguiendo los disparos con resultado AGUA y los disparos con resultado TOCADO. Por ejemplo, se puede dibujar un aspa de distinto color según el resultado.
- f) (1 punto) **Disparo del enemigo (servidor *rest*)**. Cuando el turno de disparo corresponda al enemigo (servidor), el botón de obtener el disparo del enemigo estará habilitado y al pinchar sobre él:
  - i) se realizará una petición POST ajax al servidor (*rest/juego/disparo/enemigo/*) con el parámetro *login*, cuyo valor será el login del usuario logueado.
  - ii) Si todo ha ido bien, se recibirá como respuesta una posición (columna, fila) de la rejilla del jugador. Habrá que comprobar si en

- esa posición hay algún barco del usuario, en cuyo caso el turno seguirá siendo del servidor, por lo que el botón para obtener el disparo del servidor seguirá habilitado y el usuario seguirá sin poder marcar su disparo en la zona de disparos del usuario.
- iii) El disparo se marcará en el tablero del jugador, distinguiendo si el resultado ha sido AGUA o TOCADO. Se debe marcar de la misma manera que en canvas de disparos.
  - g) (1 punto) **Comprobación de fin de juego.** Cada vez que se realice un disparo (ya sea por parte del usuario o por parte del servidor) y el resultado sea TOCADO, habrá que comprobar si con el disparo realizado se han hundido todos los barcos del receptor del mismo. Si se ha conseguido hundir todos los barcos, el juego habrá finalizado y habrá ganado el que realizó el disparo. En ese caso se debe realizar lo siguiente:
    - i) Enviar una petición POST ajax al servidor (*rest/juego/ganada/*), con el parámetro login cuyo valor es el login de usuario, en caso de que el usuario haya ganado la partida; o bien, enviar la petición POST ajax a *rest/juego/perdida/*, también con el parámetro login cuyo valor es el login de usuario, si el usuario ha perdido la partida.
    - ii) Si la respuesta del servidor es correcta (RESULTADO = 'ok'), habrá que mostrar un mensaje con capa semitransparente sobre toda la página. El mensaje a mostrar será similar a "Enhorabuena!!! Has ganado la partida", si el usuario ha ganado; o bien, "Lo siento, has perdido la partida". Además, el mensaje incorporará un botón para cerrarlo, tras lo cual se redirigirá a la página *index.html*.

**Nota:** Se considera que la **posición de un barco** es **correcta** en la cuadrícula cuando está totalmente dentro de la misma, no se solapa con ningún otro barco y está rodeado de celdas vacías (AGUA).

## Entrega

- El plazo de entrega de la práctica finalizará el **domingo 29 de mayo de 2016, a las 23:55h.**
- La práctica debe ir acompañada de una documentación en la que figure, como mínimo, los siguientes apartados:
  - Nombre, DNI y grupo del autor o autores de la práctica.
  - Apartado de posibles incompatibilidades y problemas de los navegadores. Acompañar esta lista de posibles incompatibilidades y problemas de la solución utilizada para solventarlos (si se ha podido).

**Nota:** La documentación se puede hacer en un documento independiente, o bien incluirla en la página ***acerca.html***.

- La entrega se realizará a través de la plataforma Moodle mediante la opción habilitada para ello y consistirá en un único fichero comprimido que deberá incluir lo siguiente:
  - Documentación de la práctica (si se ha realizado en un documento

independiente).

- Código completo de la práctica. Se debe comprimir la carpeta completa del sitio web.



## Peticiones del servidor *rest* de la práctica

### ERROR

Para todas las peticiones, si se produce un error se devuelve:

```
{"RESULTADO":"error", "CODIGO":"código del error", "DESCRIPCION":"Descripción del error"}
```

### Método GET

- **Obtener clasificación de puntuaciones:** [rest/clasificacion/](#)

- Con parámetros:

- [rest/clasificacion/?c={NÚMERO}](#)

Respuesta:

- Respuesta correcta:

- ```
{"RESULTADO":"ok", "CODIGO":"200", "FILAS":[ {"LOGIN":"usu2",  
"JUGADAS":"17", "GANADAS":"10"}, ... ]}
```

- **Obtener barcos del enemigo (servidor *rest*):** [rest/barco/?login={USUARIO}](#)

Devuelve la lista de los barcos colocados por el servidor para la partida con el usuario que se le pasa. Esta llamada está disponible sólo a modo de ayuda a la hora de desarrollar la práctica, para la práctica en si no la necesitáis.

Respuesta:

- Respuesta correcta:

- ```
{"RESULTADO":"ok", "CODIGO":"200",  
"FILAS":[{"COLUMNNA":"3","FILA":"3","TAMANYO":"3","ANGULO":"0","LOGIN  
":"usu1","HITS":"0"}, {"COLUMNNA":"3","FILA":"6","TAMANYO":"3","ANGULO":  
"0","LOGIN":"usu1","HITS":"0"}, ... ]}
```

### Método POST

- **Hacer login:** [rest/login/](#)

Parámetros de la petición:

- **login:** login de usuario
  - **pwd:** contraseña

Respuesta:

- Si se ha podido realizar el login:

- ```
{"RESULTADO":"ok","CODIGO":"200","CLAVE":"1225286697fbb5acbab746b  
2973de1e3", "LOGIN":"usu1","NOMBRE":"Usuario 1",  
"EMAIL":"usu1@ph2.es", "ULTIMO_ACCESO":"2016-03-01 16:31:21",  
"FOTO":"usu1.jpg"}
```

**Importante:** En esta tercera práctica sólo es necesario el login del usuario para posteriores peticiones.

● **Registro de nuevo usuario:** [rest/usuario/](#)

Parámetros de la petición:

- **login:** login del usuario.
- **pwd:** contraseña.
- **pwd2:** contraseña repetida.
- **nombre:** nombre del usuario.
- **email:** correo electrónico del usuario.
- **foto:** fichero de la foto de perfil del usuario. Debe ser el contenido de un input type="file".

Nota: En el caso de esta tercera práctica no hace falta incluir la foto en el registro.

Respuesta:

- Si se ha podido crear el usuario:  
{ "RESULTADO": "ok", "CODIGO": "200", "LOGIN": "usu4", "FOTO": "usu4.jpg" }

● **Empezar juego:** [rest/juego/empezar/](#)

Parámetros de la petición:

- **login:** login del usuario.

Respuesta:

- Si no ha habido ningún problema:  
{ "RESULTADO": "ok", "CODIGO": "200", "DESCRIPCION": "Barcos del servidor generados" }

● **Enviar al servidor disparo de usuario:** [rest/juego/disparo/jugador/](#)

Parámetros de la petición:

- **login:** login del usuario.
- **columna:** columna de la celda del disparo.
- **fila:** fila de la celda del disparo.

Respuesta:

- Si no ha habido ningún problema:
  - Si el disparo no ha alcanzado ningún barco del enemigo:  
{ "RESULTADO": "ok", "CODIGO": "200", "DISPARO": "-1", "DESCRIPCION": "AGUA" }
  - Si el disparo ha alcanzado un barco del enemigo pero no lo ha hundido:  
{ "RESULTADO": "ok", "CODIGO": "200", "DISPARO": "1", "DESCRIPCION": "TOCADO" }
  - Si el disparo ha alcanzado y hundido un barco del enemigo:  
{ "RESULTADO": "ok", "CODIGO": "200", "DISPARO": "2", "DESCRIPCION": "TOCADO Y HUNDIDO", "BARCO": { "COLUMNA": "3", "FILA": "3", "TAMANYO": "3", "ANGULO": "0", "LOGIN": "usu1", "HITS": "3" } }Si el barco se ha hundido también se devuelve la posición del barco.

● **Pedir disparo del servidor (enemigo):** [rest/juego/disparo/enemigo/](#)

Parámetros de la petición:

- **login:** login del usuario.

Respuesta:

- Si no ha habido ningún problema:  
`{"RESULTADO":"ok", "CODIGO":"200",  
"DISPARO":{"COLUMNA":"4","FILA":"6"}}`

● **Anotar partida ganada por el usuario:** [rest/juego/ganada/](#)

Parámetros de la petición:

- **login:** login del usuario.

Respuesta:

- Si no ha habido ningún problema:  
`{"RESULTADO":"ok", "CODIGO":"200", "DESCRIPCION":"Clasificación  
actualizada correctamente"}`

● **Anotar partida perdida por el usuario:** [rest/juego/perdida/](#)

Parámetros de la petición:

- **login:** login del usuario.

Respuesta:

- Si no ha habido ningún problema:  
`{"RESULTADO":"ok", "CODIGO":"200", "DESCRIPCION":"Clasificación  
actualizada correctamente"}`