

PROGRAMACIÓN DEL CLIENTE WEB. PRÁCTICA 3.

Objetivos de la práctica

- Aprender a utilizar los 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.

IMPORTANTE: 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.

Enunciado de la práctica

Esta práctica es independiente de las dos anteriores. Debéis implementar un juego de números llamado *NumTris*. El juego consiste en un tablero de 4x4 casillas (con un 20% de casillas *no jugables* elegidas de forma aleatoria) en las que, por turnos, dos jugadores irán colocando alternativamente números aleatorios del 1 al 9 (excepto el 5) que les serán proporcionados en grupos de tres. El objetivo es conseguir múltiplos de 5 sumando el número colocado en una casilla libre con el número que se encuentra en la casilla de arriba/abajo/izquierda/derecha. Se pueden conseguir hasta cuatro múltiplos de 5 en la misma jugada. Las reglas son las siguientes:

- El jugador que empieza la partida se elige de forma aleatoria.
- En cada jugada, sólo se puede colocar uno de los tres números proporcionados en una casilla vacía *jugable*. Primero se selecciona el número disponible a colocar y después se pincha en la casilla libre en la que se colocará.
- Si en una jugada el jugador consigue al menos un múltiplo de 5, recibirá tantos puntos como la suma de los números de las casillas utilizadas para conseguir dichos múltiplos de 5. A continuación, se eliminará el número utilizado de los disponibles, se limpiarán las casillas utilizadas para conseguir la puntuación y seguirá jugando el mismo jugador.
- Si al poner uno de los números disponibles, el jugador no consigue ningún múltiplo de 5, el turno pasa al otro jugador.
- Cuando se acaban los tres números disponibles, el juego genera un nuevo grupo de tres números disponibles.
- El juego finaliza cuando no quedan casillas libres para seguir colocando números y gana el jugador que ha conseguido la mayor puntuación.

A continuación se muestra un ejemplo de jugada en la que los números disponibles son 4, 8 y 7. Aunque hay diferentes posibilidades de conseguir múltiplos de 5, la mejor jugada, es decir la que más puntos suma, 17, es la que conseguimos colocando el 8 en la casilla o celda (1,2):

| Posición inicial | Mejor jugada | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|--------------|---|---|---|--|---|--|--|---|--|---|--|--|---|--|---|--|---|---|---|---|--|---|--|--|---|--|---|--|--|---|--|---|
| <table><tr><td>9</td><td></td><td>2</td><td>1</td></tr><tr><td></td><td>7</td><td></td><td></td></tr><tr><td>8</td><td></td><td>6</td><td></td></tr><tr><td></td><td>1</td><td></td><td>9</td></tr></table> | 9 | | 2 | 1 | | 7 | | | 8 | | 6 | | | 1 | | 9 | <table><tr><td>9</td><td>8</td><td>2</td><td>1</td></tr><tr><td></td><td>7</td><td></td><td></td></tr><tr><td>8</td><td></td><td>6</td><td></td></tr><tr><td></td><td>1</td><td></td><td>9</td></tr></table> | 9 | 8 | 2 | 1 | | 7 | | | 8 | | 6 | | | 1 | | 9 |
| 9 | | 2 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 7 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 8 | | 6 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 1 | | 9 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 9 | 8 | 2 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 7 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 8 | | 6 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 1 | | 9 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Descripción del sitio web del juego

El sitio web constará de las siguientes páginas:

- **index.html**. Esta será la página inicial de la práctica. En esta página, aparecerá la clasificación de las 10 puntuaciones más altas del día y un formulario en el que los dos jugadores puedan introducir sus nombres para empezar a jugar.
- **juego.html**. Será la página en la que se implementará el juego en sí.
- **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 sin realizar, si fuera el caso; problemas encontrados y solución encontrada; etc..

Nota: Para esta tercera práctica de la asignatura se proporciona servidor RESTful, pero no se proporciona base de datos puesto que no se va a utilizar.

Tareas a realizar

- 1) (0.25 puntos) Todas las páginas se implementarán siguiendo el estándar HTML y CSS, debiendo pasar correctamente la validación del HTML en <https://validator.w3.org/> y del CSS en <https://jigsaw.w3.org/css-validator/>.
- 2) (0.25 puntos) Todas las páginas tendrán en común las siguientes partes, además de las propias de cada una:
 - Cabecera con: **logotipo**, nombre/título del sitio web y un pequeño texto explicativo del objetivo del juego.
 - Pie con, al menos, información de contacto y año de creación. La información de contacto será un enlace a la página **acerca.html**, en la que se mostrará la información completa del autor o autores de la práctica y la documentación de la misma.

- 3) (0.25 puntos) El diseño de las páginas se hará usando la técnica *Mobile First*, permitiendo la correcta visualización de todo el contenido de la página en todo momento, sin que se solapen o se pierdan elementos al cambiar el ancho de la ventana del navegador. Para obtener la puntuación de este apartado, al cambiar el tamaño de la ventana del navegador se deben apreciar claramente cambios en la disposición de los elementos de la página web mediante el uso de reglas media query.
- 4) Página `index.html`.
- (0.25 puntos) Al cargar la página se debe comprobar si se está jugando una partida, en cuyo caso se redireccionará a la página `juego.html`. La forma de comprobar si se está jugando una partida será consultando `sessionStorage` y buscando la información de la partida. Si no existe información de la partida en `sessionStorage` significará que no se está jugando ninguna partida. La información de la partida que se guardará en `sessionStorage` será el estado del tablero de juego (se recomienda utilizar una matriz de javascript), los nombres y puntuaciones de los jugadores, qué jugador tiene el turno y los números disponibles.
 - b. Clasificación de las 10 puntuaciones más altas del día.**
 - (0.5 puntos) El diseño será el siguiente: Tendrá un título y mostrará las 10 puntuaciones más altas ordenadas de mayor a menor. En cada fila de la tabla se mostrará la posición, el nombre del jugador y la puntuación obtenida. Si hubiera menos de 10 posiciones almacenadas, sólo se mostrará la tabla para las puntuaciones almacenadas. En ningún caso se mostrarán filas vacías hasta completar 10 filas. A continuación se muestra un ejemplo de lo que se pide:

Top 10 puntuaciones de hoy

| Pos. | Usuario | Puntuación |
|------|----------|------------|
| 1 | Juan | 230 |
| 2 | Ana | 196 |
| 3 | Pedro | 190 |
| 4 | Jose | 118 |
| 5 | Patricia | 108 |

La primera vez que arranque el sitio web, lógicamente no habrá ninguna puntuación almacenada, por lo que sólo se mostrará el título, la cabecera de la tabla y un texto similar a "Todavía no hay puntuaciones guardadas. ¡¡¡ Sé el primero en conseguir una puntuación máxima !!!", tal y como se muestra en la siguiente imagen:

Top 10 puntuaciones de hoy

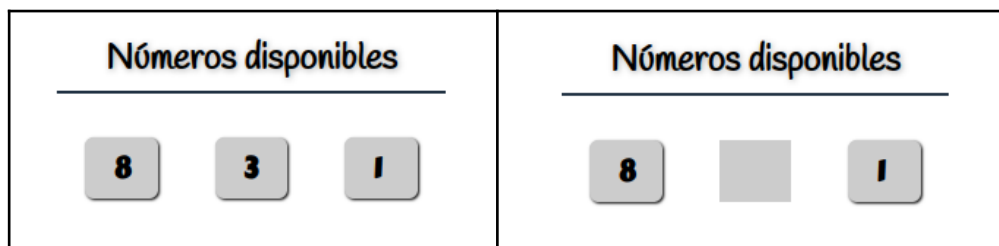
| Pos. | Usuario | Puntuación |
|------|---------|------------|
|------|---------|------------|

Todavía no hay puntuaciones guardadas.
!!! Sé el primero en conseguir una puntuación máxima !!!

- ii. (0.5 puntos) Al no disponer de base de datos y tratarse de las puntuaciones diarias, los datos de las puntuaciones se almacenarán en `sessionStorage`. De esta manera, cada vez que se carga la página `index.html` se deberá consultar la información de las puntuaciones en `sessionStorage` y mostrar el diseño de la tabla en función de lo explicado en el punto anterior.
 - c. (0.5 puntos) **Formulario de identificación de jugadores.** El formulario tendrá dos input de tipo text para recoger los nombres de los dos jugadores, un botón para comenzar el juego y un botón para limpiar los campos del formulario. Se debe tener en cuenta que los campos para introducir los nombres no pueden quedar vacíos, por lo que si se pincha el botón para jugar y alguno de ellos está vacío no se hará nada.
 - d. (0.25 puntos) Botón para comenzar juego. Al pinchar en el botón comenzar juego, se deberá guardar en `sessionStorage` los nombres de los jugadores y se redirigirá a `juego.html`.
- 5) Página **juego.html**. Esta página será la principal del juego. En ella aparecerá el tablero de juego, el marcador y los números disponibles.
- a. (0.25 puntos) Al cargar la página, si no se está jugando una partida (no hay datos de los jugadores en `sessionStorage`) se redirigirá a `index.html`.
 - b. Si hay datos de los jugadores en `sessionStorage`, se hará lo siguiente:
 - i. (0.5 puntos) Si hay datos de la partida en `sessionStorage`, se deberá mostrar el estado de la misma: tablero de juego, marcador del juego y números disponibles.
 - ii. (1 punto) Si no hay datos de la partida en `sessionStorage` es porque se va a iniciar la partida, por lo que se hará lo siguiente:
 - Una petición ajax/fetch de tipo GET a la url `api/tablero` para que el servidor genere y devuelva un tablero de juego aleatorio (ver apartado de peticiones al final del enunciado).
 - Se generarán aleatoriamente tres números entre el 1 y 9, excluyendo el 5, que serán los números disponibles para empezar a jugar.
 - Se elegirá aleatoriamente el turno del jugador que comienza a jugar y se mostrará un mensaje modal indicándoselo a los jugadores.
 - Se deberá guardar en `sessionStorage` toda la información






necesaria (estado del tablero, nombre y puntuaciones de los jugadores, qué jugador tiene el turno y números disponibles) para jugar una partida. A continuación, se mostrará en la página toda la información de la partida, de la misma forma que en el apartado 5bi. (Nota: La idea de guardar toda la información de la partida en `sessionStorage` es para no perderla si se recarga la página y así poder seguir jugando la misma partida.)

- c. (0.5 puntos) **Barra de opciones.** La página deberá tener una barra de opciones o similar, en la que se mostrará una opción para terminar la partida y otra para mostrar ayuda con las instrucciones del juego. Para las instrucciones del juego podéis utilizar el siguiente texto: *“El juego consiste en ir colocando en las casillas vacías del tablero los números que se proporcionan en grupos de tres. Juegan dos jugadores por turnos. Si al colocar un número en una celda vacía, sumándole el que tiene arriba/abajo/izquierda/derecha se obtiene un múltiplo de 5, se limpian las casillas correspondientes y el resultado de la suma son los puntos que acumula el jugador, manteniendo el turno. El juego finaliza cuando ya no quedan casillas vacías en el tablero, ganando el jugador con mayor puntuación.”*
- Opción *Terminar partida*. Limpiará todos los datos almacenados de la partida (excepto la lista de las 10 puntuaciones más altas, que no forma parte de la información de la partida), y redirigirá a `index.html`.
 - Opción *Ayuda*. Mostrará un mensaje modal en el que se mostrarán las instrucciones del juego
- d. (0.5 puntos) **Marcador del juego.** El marcador de juego mostrará el nombre de ambos jugadores, junto a sus correspondientes puntuaciones, y de alguna manera indicará a quién le corresponde el turno de juego.
- e. (0.5 puntos) **Tablero de juego.** Se representará mediante una cuadrícula de tamaño 4x4 (se deberá utilizar un elemento `<canvas>`) en la que se mostrarán las celdas y los números que hubiera en ellas. Las celdas *no jugables* deberán distinguirse de las *jugables* de alguna manera.
- f. (0.5 puntos) **Números disponibles.** Se mostrarán los tres números disponibles. Si no hubiese tres números disponibles porque ya se ha utilizado alguno de ellos, se mostrará el hueco vacío de aquellos números disponibles ya utilizados junto a los números disponibles que quedaran. A continuación se muestra un ejemplo de lo que se pide:



En la parte derecha todos los números disponibles. En la parte izquierda los números disponibles una vez utilizado el número 3.

6) Dinámica de juego.

- a. (0.75 puntos) Selección de número disponible. El jugador que tiene el turno, primero deberá hacer clic sobre uno de los números disponibles teniendo en cuenta lo siguiente:
 - Al pasar el ratón por encima de un número disponible, el cursor del ratón cambiará a  (valor pointer de la propiedad css cursor) y se destacará de los demás números disponibles mediante css.
 - Al hacer clic sobre el número disponible, éste quedará destacado de los demás mediante css. Sólo puede haber un máximo de un número disponible destacado en todo momento.
 - Si el cursor del ratón pasa sobre un hueco vacío de un número disponible, el cursor pasará a  (valor not-allowed de la propiedad css cursor) y no se destacará ni hará nada aunque el usuario haga clic sobre el hueco.
- b. (0.75 puntos) Ubicación en el tablero del número disponible seleccionado.
 - Si no hay ningún número disponible seleccionado, al pasar el cursor del ratón sobre el canvas, éste cambiará a  y, aunque se haga clic sobre el canvas, no se hará nada más.
 - Si hay un número disponible seleccionado, si el cursor del ratón pasa sobre una celda no disponible o una celda ya ocupada por un número, el cursor del ratón cambiará a  y, aunque se haga clic sobre el canvas, no se hará nada más. Si el cursor del ratón pasa por una celda disponible y vacía, el cursor cambiará a , la celda se destacará de las demás y si el usuario hace clic se colocará el número en ella haciendo lo que se indica en el siguiente apartado.
- c. (0.75 puntos) Número colocado en el tablero. Se hará lo siguiente:
 - Se eliminará el número de los números disponibles, dejando su hueco vacío. Hay que tener en cuenta que este hueco vacío dejará de estar destacado.
 - Se guardará el número en la correspondiente posición de la matriz del tablero y se realizará una petición ajax/fetch de tipo POST a la url

api/comprobar, a la que se le pasará la matriz del tablero. El servidor comprobará si se ha conseguido algún múltiplo de 5 y devolverá un array con las posiciones (fila,columna) de las celdas que sumen múltiplos de 5 (ver apartado de peticiones al final del enunciado) y se hará lo siguiente:

- Si no se ha conseguido sumar múltiplos de 5, el array vendrá vacío y simplemente se repintará el canvas con el nuevo tablero. A continuación se comprobará si hay fin de juego (ver apartado 7). Si no hay fin de juego, el turno pasará al otro jugador.
- Si se ha conseguido sumar algún múltiplo de 5, se sumarán los valores de las celdas indicadas en el array devuelto y sus posiciones en la matriz de juego se limpiarán, dejándolas libres para que se puedan volver a colocar números en ellas. A continuación se repintará el canvas con el nuevo tablero y la cantidad de puntos obtenida se sumará a la puntuación del jugador, que mantendrá el turno de juego.

7) **Fin de juego.** Al recibir la respuesta del servidor tras la petición POST a api/comprobar, uno de los campos recibidos indica el número de celdas *jugables* que quedan en el tablero.

a. (0.5 puntos) Si el número de celdas jugables es 0 y el array con las posiciones de las celdas que suman múltiplos de 5 está vacío (es decir, no se ha conseguido ningún múltiplo de 5), se habrá alcanzado el fin de juego y se hará lo siguiente:

- Se mostrará un mensaje modal diciendo que se ha alcanzado el fin de juego, indicando qué jugador ha sido el ganador y mostrando la puntuación de cada jugador. El mensaje modal tendrá un botón *Aceptar* que cerrará el mensaje modal.
- Al pinchar en el botón *Aceptar* y cerrar el mensaje modal, se deberán guardar las puntuaciones de los jugadores en la lista de puntuaciones, siempre y cuando estén dentro de las 10 puntuaciones más altas (deben ser mayores que la más baja de las puntuaciones guardadas). A continuación se borrará toda la información de la partida de `sessionStorage` y se redirigirá a `index.html`.

b. (0.25 puntos) Si no se cumplen las dos condiciones anteriores para que se dé el fin de juego, se seguirá jugando la partida, por lo que si no quedan números disponibles habrá que volver a generar tres números disponibles aleatorios y mostrarlos en la zona de números disponibles.

8) Página **acerca.html**.

a. (0,25 puntos) Al igual que en las prácticas anteriores, tendrá la siguiente información maquetada lo más correctamente posible con elementos html:

- Texto indicando que se trata de la práctica 3 de la asignatura

Programación del Cliente Web, que pertenece al 2o curso de la titulación del Grado en Ingeniería Multimedia.

- Asimismo, aparecerá la información relativa al autor o autores de la práctica (dni, nombre, apellidos, grupo de prácticas y correo electrónico).
- Cualquier comentario al respecto del desarrollo de la práctica que se considere relevante, como pueden ser problemas de compatibilidad entre navegadores, partes de la práctica no realizadas, etc.

- b. (0.25 puntos) **Barra de opciones**. Deberá tener una única opción que si se está jugando una partida permitirá volver al juego y si no se está jugando una partida, la opción permitirá volver a `index.html`.

Entrega

- El plazo de entrega de la práctica finalizará el **domingo 28 de mayo de 2023, a las 23:59h**.
- La entrega se realizará a través de la plataforma Moodle mediante la opción habilitada para ello y consistirá en un único fichero comprimido que contendrá todos los ficheros de la práctica para su correcto funcionamiento. Se recomienda comprimir la carpeta completa de la práctica.

Peticiones al servidor *RESTful* de la práctica 3

ERROR

Para todas las peticiones, si se produce un error se devuelve el siguiente texto en formato JSON:

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

Peticiones GET

- **RECURSO:** [api/tablero](#)
 - Generación de tablero de juego: [api/tablero](#)

Respuesta:

```
{"CODIGO":200,"RESULTADO":"OK","TABLERO":[[0,0,0,0],[0,0,-1,0],[0,0,-1,-1],[0,0,0,0]]}
```

Las celdas con valor 0 son las celdas vacías (celdas *jugables*) y las celdas con valor -1 son las celdas *no jugables*.

Peticiones POST

- **RECURSO:** [api/comprobar](#)
 - **Comprobación del tablero de juego:** [api/comprobar](#)

Parámetros de la petición:

- **tablero:** matriz del tablero de juego en formato json.

Respuesta:

- Si no se ha conseguido ningún múltiplo de 5:
`{"CODIGO":200,"RESULTADO":"OK","CELDAS_SUMA":[],"JUGABLES":11}`
- Si se ha conseguido algún múltiplo de 5:
`{"CODIGO":200,"RESULTADO":"OK","CELDAS_SUMA":[{"fila":2,"col":3},{"fila":1,"col":3}],"JUGABLES":10}`