

## 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.

### Enunciado de la práctica

Esta práctica es independiente de las dos anteriores. Debéis implementar el juego conocido como 1010 ó 10x10 ([Aquí](#) podéis ver una versión). A continuación se explicará el funcionamiento del juego.

**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:

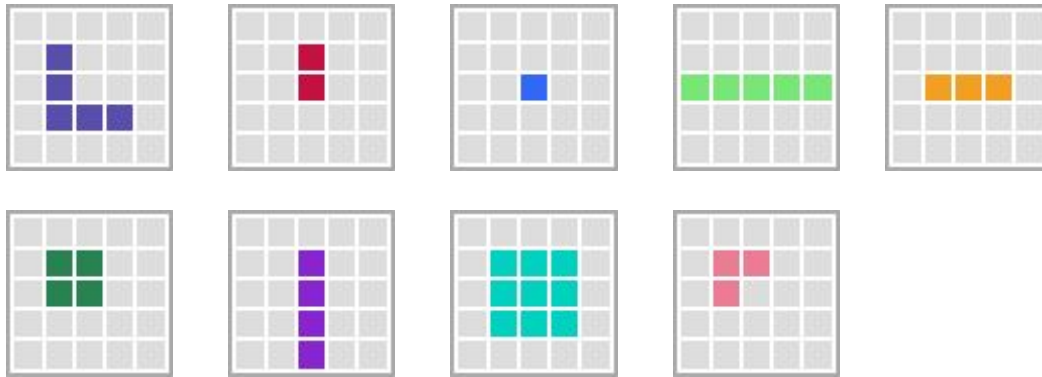
- **index.html**. Esta será la página inicial de la práctica y en ella aparecerá todo lo necesario para el juego. En el apartado de tareas a realizar se detalla todo lo que debe aparecer en esta página.
- **acerca.html**. Al igual que en el resto de las prácticas, esta página contendrá la información 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..

### Descripción del juego

El juego a desarrollar es un típico juego de puzzles, muy parecido al tetris, pero con algunas diferencias. En este caso se dispone de un panel de juego de 10x10 casillas. Junto al panel de juego irán apareciendo piezas, de tres en tres y elegidas aleatoriamente, que habrá que encajar en las casillas vacías del panel de juego. Cuando una fila y/o columna se rellene, se eliminará del panel, dejando las correspondientes casillas libres para colocar nuevas piezas.

- Piezas

Hay un total de 9 piezas. Al seleccionar las tres piezas al azar, éstas podrán aparecer rotadas, también de forma aleatoria, en 2D, en ángulos de 90°, 180° y 270°, siempre y cuando existan dichas rotaciones. Las 9 piezas son las siguientes:



- Puntuación

Cada vez que se coloque una pieza, la puntuación del jugador aumentará en el número de casillas ocupadas, es decir, el número de casillas que forma la pieza. Además, si al poner la pieza se rellena, y elimina, una fila o columna, se sumarán otros 10 puntos. Si tras colocar la pieza, el número de filas y/o columnas que se eliminan es mayor de 1, la puntuación que suman las filas y/o columnas eliminadas se multiplica por dos.

- Fin de juego

El juego finalizará cuando en el panel de juego no haya sitio para colocar ninguna de las fichas que le quedan al jugador.

Junto al enunciado de la práctica se os proporcionará un servidor RESTful (igual que se hizo en la práctica 2) muy simple, cuya única función será llevar la clasificación de puntuaciones conseguidas por los usuarios. Para ello, tendrá acceso a una base de datos, que también se os proporciona junto al enunciado, en la que se guardarán, únicamente, las puntuaciones junto al nombre del jugador que la consiguió.

### 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) (0,5 puntos) Todas las páginas tendrán en común la siguientes partes, además de las propias de cada una:
  - a) Cabecera con: logotipo, nombre/título del sitio web y un pequeño texto explicativo del objetivo del juego.
    - i) En el caso de la página `acerca.html`, el logotipo será un enlace a la página `index.html`.
  - b) Pie con, al menos, información de contacto, copyright, año de creación.
    - i) En el caso de la página `index.html`, 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.
- ii) En el caso de la página `acerca.html`, el pie de página incluirá un botón/enlace que permitirá volver a la página `index.html`.
- 3) (0,25 puntos) El diseño de las páginas se hará utilizando la técnica *Mobile First*, permitiendo la correcta visualización de todo el contenido de la página en todo momento.
- 4) Página `index.html`. Debe haber dos zonas: una para el panel de juego; y la otra para las tres piezas que tendrá el usuario para colocar.
- a) **Panel de juego.** Tendrá:
- i) (0,25 puntos) Un título (“Panel de juego”, o similar)
  - ii) (0,5 puntos) Un `<canvas>`, que será el panel de juego. Mostrará las divisiones que formarán las 10x10 casillas.
  - iii) (0,25 puntos) Un apartado para mostrar la puntuación que lleva el jugador.
  - iv) (0,25 puntos) El panel de juego se implementará en JavaScript mediante una matriz (objeto Array) de ocupación de dos dimensiones, que servirá para gestionar la ocupación de casillas al colocar las distintas piezas.
- b) **Piezas a colocar.** Tendrá:
- i) (0,25 puntos) Un título (“*Piezas disponibles*”, o similar)
  - ii) (0,5 puntos) Tres elementos `<canvas>`, uno por cada una de las tres piezas (seleccionadas aleatoriamente) a colocar. Estos elementos canvas mostrarán las divisiones que formarán una rejilla de 5x5 casillas.
  - iii) (0,5 puntos) Las piezas deberán implementarse en JavaScript como objetos con, al menos, dos propiedades: una que contendrá la forma base de la pieza y otra que será el color de la pieza.
- 5) (0,25 puntos) Página `acerca.html`. Al igual que en las prácticas anteriores, tendrá la siguiente información:
- a) 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.
  - b) Asimismo, aparecerá la información relativa al autor o autores de la práctica (dni, nombre, apellidos, grupo de prácticas y correo electrónico).
  - c) 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.
- 6) **Funcionamiento del juego.**
- **Inicio**
    - a) Al cargar la página `index.html`:
      - i) (0,25 puntos) Sobre el contenido de la página aparecerá una ventana
-

- 
- modal con la clasificación de las diez mejores puntuaciones.
- ii) (0,25 puntos) La lista de las diez mejores puntuaciones se obtiene de la base de datos del juego mediante la correspondiente petición al servidor RESTful que se os proporciona.
- b) La ventana modal tendrá un botón *Jugar*. Al pulsar este botón, la ventana modal se cerrará y empezará el juego:
- iii) (0,25 puntos) Se mostrará el panel de juego vacío (sin piezas), tal y como se pide en el apartado 4.a.ii, y la puntuación a cero.
  - iv) (0,75 punto) Se seleccionarán las primeras tres piezas al azar (posibles rotaciones incluidas) y se mostrarán en la zona de piezas disponibles, cada pieza en un canvas.
- **Movimiento de las piezas al panel.** Para mover las piezas de la zona de piezas disponibles al panel, se seguirán los siguientes pasos:
    - a) (0,25 puntos) Primero se hará click en la pieza a colocar. Esta pieza pasará a destacarse de las otras dos, de alguna manera, para indicar que es la pieza seleccionada para colocar en el panel.
    - b) (0,75 puntos) A continuación, se moverá el cursor del ratón al panel de juego y la pieza seleccionada se irá dibujando en el panel de control, acorde a la posición del cursor del ratón y siguiendo su movimiento. La idea es mostrar al usuario la posición en la que se colocaría la pieza si se hiciera click sobre el panel.
  - **Colocación de la pieza.** Para colocar la pieza en el panel bastará con hacer click sobre él teniendo una pieza, de las disponibles, seleccionada.
    - a) Si se hace click sobre el panel y no hay ninguna pieza seleccionada, no se hará nada.
    - b) Si se hace click sobre el panel y hay una pieza seleccionada:
      - i) (0,5 puntos) Si la pieza está completamente dentro del panel y, además, las casillas que ocupa están libres, se pintará en el panel la pieza con su color correspondiente.
      - ii) (0,5 puntos) Se marcarán las correspondientes casillas como ocupadas en la matriz de ocupación, guardando el color de la ficha en cada posición.
      - iii) Tras la colocación de una pieza hay que realizar las siguientes tareas:
        - 1) (0,5 puntos) Eliminar la pieza de la zona de piezas disponibles, dejando vacío (solo con las divisiones) el canvas en el que estaba.
        - 2) (0,25 puntos) Actualizar la puntuación del usuario. Se debe comprobar si se ha completado fila y/o columna para calcular el aumento final de la puntuación, tal y como se explica en el apartado *Puntuación* de la descripción del juego.
        - 3) (0,5 puntos) Si se han completado filas y/o columnas, eliminarlas del panel y volver a poner como libres las correspondientes casillas de la
-

matriz de ocupación.  
4) Comprobar fin de juego

- **Comprobación de fin de juego.**

- a) (0,5 puntos) El juego finalizará cuando ninguna de las piezas disponibles que le queden al usuario se pueda poner en el panel.
- b) Se mostrará una ventana modal en la que:
  - i) (0,25 puntos) Aparecerá un mensaje indicando al jugador que se ha acabado el juego porque no puede colocar más piezas.
  - ii) (0,25 puntos) Se le indicará el número de puntos conseguidos.
  - iii) (0,25 puntos) Se le pedirá el nombre, mediante un elemento `<input>` de tipo texto, para guardar su puntuación en la base de datos.
  - iv) Habrá un botón (o similar) con el texto *Volver a jugar*. Al pulsar en este botón:
    - 1) (0,25 puntos) Si el usuario escribió su nombre en el campo de texto, se debe hacer una petición al servidor RESTful para guardar el nombre junto a su puntuación en la base de datos, utilizando la correspondiente petición que se os proporciona.
    - 2) Cerrará la ventana modal y reiniciará el juego, haciendo lo mismo que se pide en el apartado 6.b tras pulsar el botón *Jugar*.

## Entrega

- El plazo de entrega de la práctica finalizará el **domingo 26 de mayo de 2019, a las 23:59h**.
- 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 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/puntuaciones](#)
  - Petición de la lista de puntuaciones: [api/puntuaciones](#)

Respuesta:

- Devuelve la lista de puntuaciones ordenadas de mayor a menor.

### Peticiones POST

- **RECURSO:** [api/puntuaciones](#)
  - Guardar puntuación: [api/puntuaciones](#)

Parámetros de la petición:

- **nombre:** nombre del jugador
- **puntos:** puntuación a guardar junto al nombre del jugador

Respuesta:

- Si la puntuación se ha guardado correctamente:  

```
{"RESULTADO": "OK", "CODIGO": 201, "DESCRIPCION": "Puntuación guardada  
correctamente", "NOMBRE": "Ana", "PUNTOS": 264}
```