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 típico juego para montar un puzzle de una imagen. En este enunciado se explica cómo debe ser el funcionamiento del juego que se pide para la práctica.

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.

Descripción del juego

El sitio web constará de las siguientes páginas:

- **`index.html`**. Esta será la página inicial de la práctica. En ella aparecerá una tabla de puntuación y permitirá seleccionar la imagen y la dificultad a utilizar para empezar a jugar. En el apartado de tareas a realizar se detalla todo lo que debe aparecer en esta página.
- **`juego.html`**. Será la página en la que se implementará el juego del puzzle 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..

El juego a desarrollar es el típico juego para montar un puzzle a partir de una imagen. En esta práctica se podrá elegir una de entre 6 imágenes y la dificultad. Las imágenes a utilizar se os proporcionan con el enunciado de la práctica, aunque podéis sustituirlas por otras (manteniendo el nombre de los ficheros) si queréis. Y la dificultad está relacionada con el número de piezas del puzzle: 4x4, 6x6, 8x8. Hasta que el usuario no haya elegido imagen y dificultad, no podrá empezar el juego.

Junto al enunciado de la práctica, además de las imágenes, se os proporciona un servidor RESTful (igual que se hizo en la práctica 2) muy simple y un script de creación de la base de datos del juego. La base de datos es muy sencilla. Simplemente guardará la información de las partidas jugadas (fecha, usuario, dificultad, número de jugadas). El servidor aceptará tres tipos de peticiones: una de tipo GET para obtener la tabla de puntuación y poder mostrarla en index; otra de tipo GET para pedir las imágenes; y otra petición de tipo POST para poder guardar nuevas puntuaciones. Al final de este enunciado se os proporciona la

lista de peticiones que podéis hacer al servidor.

Tareas a realizar

1) (0,5 puntos) Todas las páginas deben pasar satisfactoriamente y en todo momento la validación HTML en http://validator.w3.org.

2) (0,5 puntos) Todas las páginas tendrán en común las 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.

    b. Pie con, al menos, información de contacto, copyright, año de creación.
        i. 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) (0,5 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`**. Al cargar la página debe mostrar:

    a. **Tabla de puntuación**. Tendrá un título “Tabla de puntuación”, o similar.
        i. (0,75 puntos) Mostrará una tabla con las puntuaciones conseguidas por los jugadores. Para ello, se deberá hacer una petición de tipo GET a la url `api/puntuaciones` para pedir el listado de puntuaciones ordenado por nivel de dificultad descendente y número de jugadas ascendente (ver petición al final del enunciado). Para cada fila de la tabla se mostrará la posición, el nombre de la imagen utilizada para el puzzle, el nombre del jugador, el nivel de dificultad y las jugadas empleadas para montar el puzzle.
        ii. (0,5 puntos) Además de la tabla, habrá un par de botones que permitan moverse mediante paginación (solo opciones atrás y adelante) por las puntuaciones del juego. El tamaño de página lo podéis elegir vosotros, pero se recomienda que no sea mayor de 10 registros por página. El sistema de paginación es igual al utilizado en la práctica 2 (ver petición al final del enunciado).

    b. **Zona de selección de opciones para jugar**. Tendrá un título “Opciones de juego”, o similar, y mostrará las dos opciones posibles a elegir para poder jugar: la imagen a utilizar y la dificultad.
        i. (0,75 puntos) Lista de imágenes disponibles.
            - Se hará una petición de tipo GET a la url `api/imagenes` para pedir la información de las imágenes disponibles. Las imágenes se deberán

mostrar indicando, además, su nombre.
- Al pasar el ratón por encima de una imagen, ésta se deberá destacar de las demás, de alguna forma, mediante css.
- En esta zona el jugador seleccionará la imagen a utilizar pinchando sobre ella.Tras pinchar sobre una imagen, ésta pasará a estar seleccionada destacándose de las demás (por ejemplo, poniendo su borde en rojo). En todo momento sólo puede haber una imagen seleccionada.

ii. (0,25 puntos) Nivel de dificultad. Se mostrarán las tres posibles opciones de dificultad, que no son más que las dimensiones, en número de fichas, del puzzle: Fácil (4x4), Media (6x6), Alta (8x8). Sólo se puede elegir una opción.

iii. (0,5 puntos) Botón para jugar. Habrá un botón *Empezar*, o similar, que el jugador deberá pulsar para empezar a jugar.
- Si el jugador no ha seleccionado la imagen o no ha seleccionado el nivel de dificultad, se notificará mediante un mensaje modal y no se podrá comenzar el juego.
- Cuando el jugador ha seleccionado las dos opciones (imagen y dificultad), al pulsar el botón para empezar se guardará esta información en sessionStorage y se redirigirá a la página `juego.html`.

5) Página **`juego.html`**. Esta página será la principal del juego. En ella aparecerán dos secciones: una con la imagen elegida para el puzzle y la otra con el puzzle.
   a. (0,25 puntos) **Zona de la imagen**. Tendrá un título “Imagen a conseguir” o similar y un canvas en el que se mostrará la imagen elegida en la página `index.html`.
   b. (0,25 puntos) **Zona del puzzle**. Se mostrará un título “Puzzle” o similar y un canvas con las divisiones correspondientes al nivel de dificultad seleccionado en `index.html`. Además:
      - Habrá un botón “Terminar” o similar, que permitirá al usuario abortar la partida y volver a `index.html`.
      - Mostrará el contador de jugadas (intercambios de piezas) realizadas.

   Nota: Ver funcionamiento del juego en el apartado 7.

6) (0,5 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.

7) **Funcionamiento del juego.**

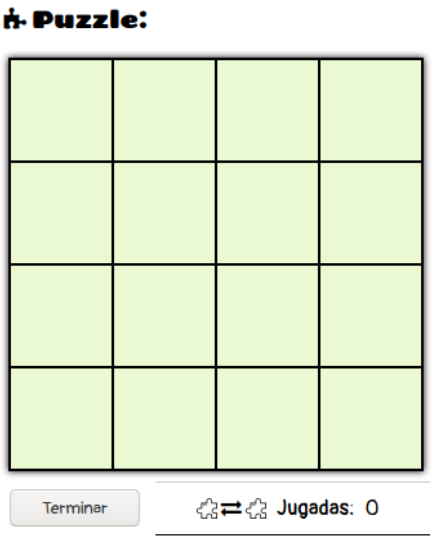
- Página `juego.html`.

a. (0,25 puntos) Al cargar la página, se debe comprobar que en sessionStorage se encuentra la información relativa a la imagen y nivel de dificultad seleccionado. Si no existe esta información, se debe redireccionar a `index.html`.

b. Creación del puzzle. Con la información obtenida de sessionStorage (imagen y nivel de dificultad) se debe crear el puzzle. Para ello:

i. (0,25 puntos) Primero se mostrará la imagen seleccionada en el canvas de la zona de la imagen.

ii. (1 punto) A continuación, se debe "trocear" la imagen en tantas piezas como indique el nivel de dificultad y, posteriormente, "mezclarlas" (Nota: Se recomienda realizar esta parte utilizando arrays de javascript, por ser más sencillo). En el canvas de la zona del puzzle aparecerán todas las piezas "boca abajo". Para conseguir este efecto, se dibujarán las divisiones sobre un fondo del mismo color, quedando similar a lo que se puede ver en la siguiente imagen:



Ejemplo de la zona de puzzle con todos sus elementos

c. (1,5 puntos) Movimientos. Una vez generado el puzzle, el contador de jugadas se pondrá a 0 y empezará el juego. El usuario irá haciendo clic sobre las piezas del puzzle teniendo en cuenta lo siguiente:

i) Si la pieza está "volteada" permitiendo ver la región de la imagen que "contiene", no se hará nada.

ii) Si la pieza no está volteada se "volteará", permitiendo ver la región de la imagen que "contiene" y se hará lo siguiente:

1. Si la pieza está en su posición correcta, se dejará boca arriba y no se hará nada más. Este caso no supone intercambio de piezas, por lo que no se debe incrementar el contador de jugada.

2. Si la pieza no está en su posición correcta:
    i. Si no hay ninguna otra pieza seleccionada, ésta pasará a estar "seleccionada", destacándola de alguna manera de las demás (por ejemplo, poniendo su borde en rojo).
    ii. Si ya hay una pieza seleccionada, se intercambiarán ésta y la pieza seleccionada y, transcurrido 1 segundo, se volverán a poner "boca abajo" salvo que, tras el intercambio, la pieza esté en la posición correcta en el puzzle. A continuación, dejará de haber pieza seleccionada (ya no aparecerá ninguna pieza destacada de las demás) y se contará una jugada, ya que se ha realizado un intercambio de piezas.

d. Fin de juego. El fin de juego se puede dar porque el jugador haya armado el puzzle o porque haya pulsado el botón de terminar antes de finalizar.

    i. (1,5 puntos) El jugador arma el puzzle. Tras dejar "boca arriba" una pieza que se encuentra en su lugar, se debe comprobar si se ha finalizado el juego, es decir, si todas las piezas están visibles porque están en su lugar.
        - Si se ha finalizado el juego se debe mostrar un mensaje modal al jugador del tipo "¡¡Enhorabuena!! Has montado el puzzle en X jugadas", siendo X el número de jugadas realizadas (intercambios de dos piezas). Además, en el mensaje modal se debe mostrar un texto del tipo "Introduce tu nombre para guardar tu puntuación" y un input de tipo text para recoger el nombre del jugador.
        - Al pinchar en el botón de Aceptar del mensaje modal, si el jugador ha escrito su nombre se hará una petición POST a la url `api/puntuaciones` para guardar la puntuación. Esta petición deberá incluir los parámetros nombre, dificultad y número de jugadas (ver apartado de peticiones al final del enunciado). Si la puntuación se ha guardado correctamente, se debe mostrar un mensaje modal al jugador indicándoselo. Si el usuario no ha escrito su nombre, se reiniciará el juego (limpiando lo guardado en sessionStorage) y se redirigirá a `index.html`.

    ii. (0,25 puntos) El jugador pulsa el botón de terminar. Se deberá mostrar un mensaje modal al jugador con un mensaje similar a "¡¡Lástima!! No has podido armar el puzzle tras X jugadas", donde X es el número de jugadas realizadas (intercambios de dos piezas). A continuación, se reiniciará el juego (limpiando lo guardado en sessionStorage) y se redirigirá a `index.html`.

Entrega

- El plazo de entrega de la práctica finalizará el **domingo 30 de mayo de 2021**, **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.

(Ver listado de peticiones al servidor en la página siguiente)

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/imagenes`
  - **Pedir la lista de imágenes**: `api/imagenes`

    Parámetros de la petición: Ninguno

    Respuesta:
    - Si se ha podido realizar correctamente la petición:
      `{"CODIGO":200,"RESULTADO":"OK","FILAS":[{"id":1,"nombre":"Imagen 1","fichero":"img1.jpg"},{"id":2,"nombre":"Imagen 2","fichero":"img2.jpg"},{"id":3,"nombre":"Imagen 3","fichero":"img3.jpg"},{"id":4,"nombre":"Imagen 4","fichero":"img4.jpg"},{"id":5,"nombre":"Imagen 5","fichero":"img5.jpg"},{"id":6,"nombre":"Imagen 6","fichero":"img6.jpg"}]}`
  - **Pedir toda la información sobre una imagen concreta**: `api/imagenes/`***`{ID_IMAGEN}`***

    `{ID_IMAGEN}` es el ID de la que se pide la información.

    Parámetros de la petición: Ninguno

    Respuesta:
    - Si se ha podido realizar correctamente la petición:
      `{"CODIGO":200,"RESULTADO":"OK","FILAS":[{"id":2,"nombre":"Imagen 2","fichero":"img2.jpg"}]}`

- **RECURSO**: `api/puntuaciones`
  - **Pedir la lista de puntuaciones**: `api/puntuaciones`

    Parámetros de la petición: Ninguno

7Respuesta:

- Si se ha podido realizar correctamente la petición:
  `{"RESULTADO":"OK","CODIGO":200,"FILAS":[{"id":1,"id_imagen":4,"usuario":"Pedro","dificultad":"4x4","jugadas":32,"fecha_hora":"2021-03-14 10:21:14","nombre":"Imagen 4","fichero":"img4.jpg"}, ...`

Con parámetros en la url:

- **`api/puntuaciones?ord=`*`{ORDEN}`***

  Devuelve las puntuaciones ordenadas según lo que se indique en *{ORDEN}*, mediante los siguientes valores: `i`=imagen, `d`=dificultad, `j`=jugadas, `f`=fecha. Para indicar el orden ascendente o descendente, se añade al indicador del campo por el que ordenar la letra `a` para ascendente o la `d` para descendente. Se puede especificar más de un orden separándolos por coma. Por ejemplo:
  - `api/puntuaciones?ord=i` => devuelve los datos ordenados por imagen ascendente
  - `api/puntuaciones?ord=i,dd,ja` -> devuelve los datos ordenados por imagen ascendente, dificultad descendente y número de jugadas ascendente.

- **`api/rutas?pag={pagina}&lpag={registros_por_pagina}`**

  Se utiliza para pedir los datos con **paginación**. Devuelve los registros que se encuentren en la página ***`{pagina}`***, teniendo en cuenta un tamaño de página de ***`{registros_por_pagina}`***. La primera página es la 0.

Se pueden combinar los parámetros mediante el operador **`&`** y utilizar más de un parámetro en la misma petición. El resultado es una combinación de condiciones mediante AND.

Peticiones POST

- **RECURSO**: **`api/puntuaciones`**
  - **Guardar una nueva puntuación en la base de datos**: **`api/puntuaciones`**

    Parámetros de la petición:
    - **`id_imagen`**: id de la imagen
    - **`nombre`**: nombre del jugador
    - **`dificultad`**: dificultad empleada
    - **`jugadas`**: número de jugadas (intercambios de piezas) realizadas

    Respuesta:
    - Si se ha podido realizar correctamente la petición:
      `{"RESULTADO":"OK","CODIGO":201,"DESCRIPCION":"Puntuaci\u00f3n guardada correctamente"}`