

FRONT-END PARA NAVEGADORES WEB

CONJUNTO DE PRÁCTICAS – 2022/2023

Las prácticas de las que consta la asignatura de “Front-end para Navegadores Web” (FENW) tendrán como objetivo afianzar los conocimientos adquiridos en clase acerca de los siguientes puntos, todos ellos encaminados a conseguir una construcción adecuada de la parte de “cliente” que es necesaria en los sistemas web:

- ✓ Diseño de la vista del sistema web que se ha de mostrar al usuario, basado en el desarrollo con HTML, CSS y librerías específicas como Bootstrap.
- ✓ Programación, mediante Javascript y AJAX, así como mediante la utilización de las nuevas APIs de programación que HTML5 proporciona, de los procesos necesarios para que la vista de usuario mencionada en el apartado anterior se comporte de acuerdo a las especificaciones establecidas.
- ✓ Utilización de una correcta “metodología” (patrones) de programación en javascript, así como el posible empleo de librerías de desarrollo específicas (jQuery) que puedan facilitar la labor del personal técnico encargado de dicho desarrollo.
- ✓ Utilización de un entorno de desarrollo concreto (Angular) creado para facilitar el diseño y desarrollo del sistema de acuerdo con los principios básicos de la ingeniería del software.

Para llevar a cabo esto, las prácticas consistirán en el desarrollo por fases, de la parte cliente de un sistema informático que implementará un juego de “parejas de cartas” o “matching game” y la gestión de usuarios y puntuaciones obtenidas por éstos. Cada una de las prácticas abordará una tarea específica del sistema de acuerdo con los objetivos a conseguir en ella.

PRIMERA PRÁCTICA – DISEÑO GRÁFICO DEL INTERFAZ – HTML Y CSS + BOOTSTRAP Y EJECUCIÓN EN CLIENTE – JAVASCRIPT, AJAX Y JQUERY

El objetivo de la primera práctica es que el alumno diseñe la parte visual del cliente del sistema web correspondiente al sitio del juego del “matching game” utilizando tecnologías asíncronas al mismo tiempo que diseña y programa la parte de proceso (en cliente) mediante el lenguaje de scripting Javascript o jQuery.

El alumno tendrá **libertad para diseñarlo en el formato que desee** y con la apariencia que prefiera. El objetivo del juego es conseguir detectar y señalar las “parejas” de imágenes iguales entre un conjunto de imágenes que aparecerán dadas la vuelta y que se mostrarán de dos en dos al jugador según vaya pinchando en ellas con el ratón. Aquellas parejas que se acierten puntuarán positivamente mientras que aquellas que no se correspondan se volverán a dar la vuelta. Se propone que inicialmente se correspondan con un conjunto limitado de cartas.

ATENCIÓN:

Para poder implementar ésta y la siguiente práctica se ha desarrollado una API REST que proporciona el servicio de “back-end” a la aplicación (gestión de usuarios y de puntuaciones). Esta API REST está a disposición de los alumnos para que puedan probar y validar el código generado antes de ser presentado para su evaluación. Las características básicas de la API REST son:

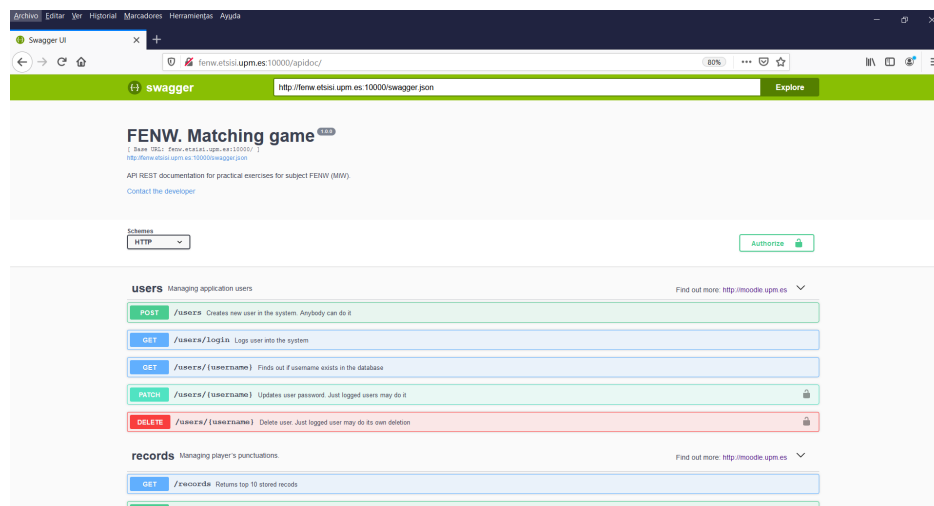
URL base: <http://fenw.etsisi.upm.es:10000> (Nótese el uso del puerto 10000)

La API se encuentra suficientemente documentada a través de la herramienta “swagger” en la url <http://fenw.etsisi.upm.es:10000/apidoc>

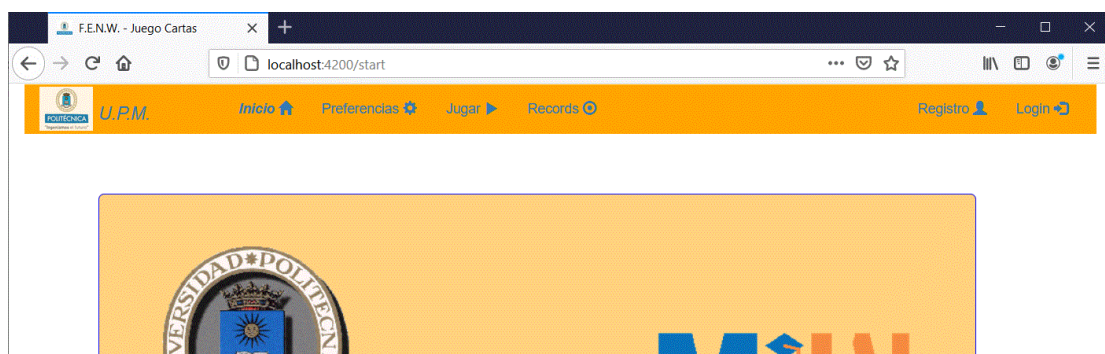
Esta página de especificación permite, por una parte, comprobar las rutas y métodos accesibles; por otra, comprobar los parámetros necesarios para su ejecución así como los resultados que la misma proporciona como respuesta; y por último, probar que se entiende su funcionamiento a través de la ejecución de los métodos disponibles en la misma página.

Hay que observar que en esta primera práctica solamente se hará uso de la ruta concerniente a la identificación de usuarios (**/users/login** a través del método **GET**). Tal y como se especifica más tarde, este método obtiene del servidor un *token* de acceso que se puede introducir en la opción “Authorize” de la página para tener acceso al resto de rutas que precisan del mismo en la cabecera.

Resulta **importante** destacar que todos los parámetros pasados a la API, sean en la query o en el formulario, se pasan como parámetros independientes (**urlencoded**), con el nombre indicado en la documentación. Sin embargo, todas las respuestas recibidas desde la API REST estarán en formato **json** (incluso aunque sea una simple cadena de caracteres o un número). Evidentemente, el token recibido en la cabecera *Authorization* será una excepción, tal y como se detalla más adelante.



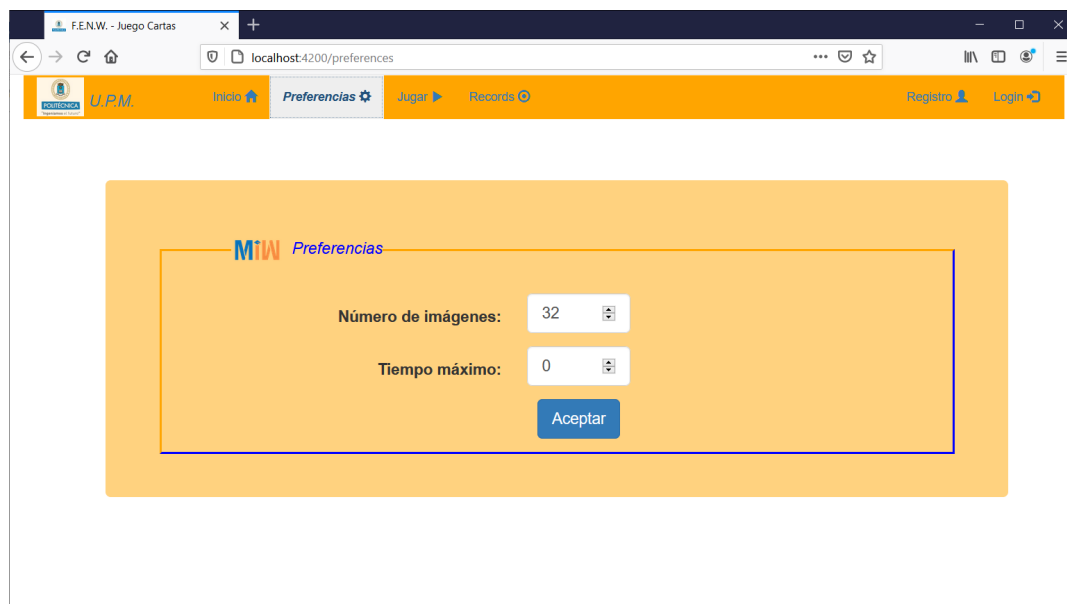
La página principal del sitio web mostrará una presentación a gusto del diseñador, teniendo en cuenta que la elaboración de la misma se tendrá en cuenta a la hora de valorar el trabajo. En esta primera práctica, aunque se describa la funcionalidad del sistema, hay que tener en cuenta que la opción de “Registro” que se va a presentar **NO** tiene ningún tipo de proceso más allá de mostrar el formulario. Obligatoriamente, esta página deberá mostrar algún tipo de menú con, como mínimo, las siguientes opciones:



Inicio. – Al pulsarla, el sistema mostrará, siempre, la página de presentación/bienvenida. Se puede observar un ejemplo en la figura en la que en la parte superior aparece el menú con las opciones divididas en dos grupos (izquierda: funciones y derecha, *registro y login*) y en la parte inferior aparece una presentación que el alumno deberá elaborar a su gusto.

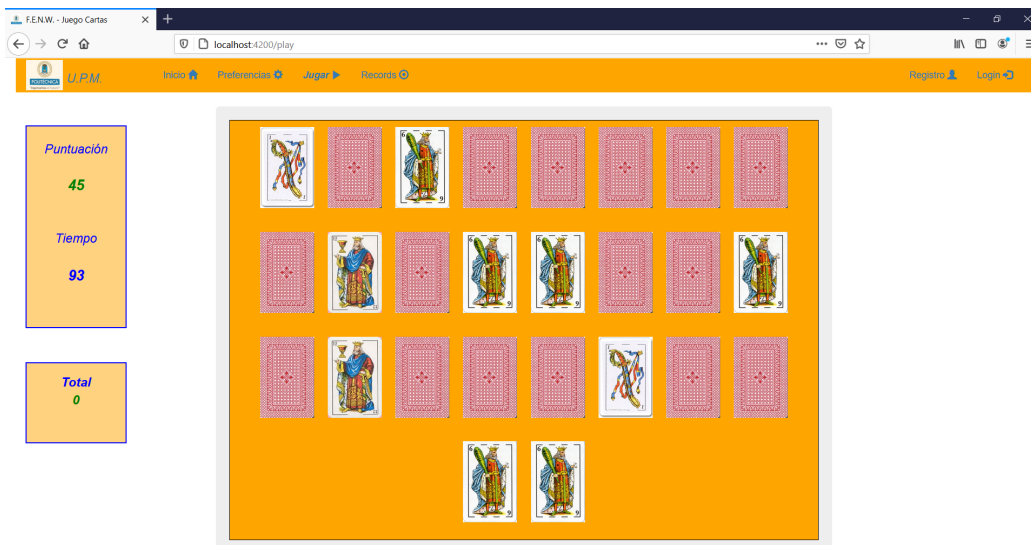


Preferencias. – Esta opción dará paso a un formulario en el que el jugador podrá elegir tanto el número de imágenes con las que va a jugar como el tiempo del que va a disponer. El conjunto de imágenes podrá ser elegido entre **20, 26 ó 32**. Por su parte, el tiempo podrá ser **cero**, que indicará **“sin límite de tiempo”**, o bien **60, 90, 120 o 150**, que serán el número de segundos de los que dispondrá el jugador para terminar el juego antes de que el sistema le impida seguir jugando. Estas preferencias deberían ser **almacenadas** localmente para poder ser consultadas a la hora de jugar.



Opcionalmente, una vez que el jugador pulsa **“Aceptar”**, puede presentarse directamente la pantalla correspondiente a la opción de **“Jugar”**

Jugar. – Esta opción presenta una pantalla en la que se muestran las imágenes (cartas) dadas la vuelta. El número de cartas se corresponderá con el elegido en las opciones y si se ha elegido un tiempo distinto de cero, se mostrará y se indicará, en todo momento, el tiempo restante en una cuenta atrás hasta cero.

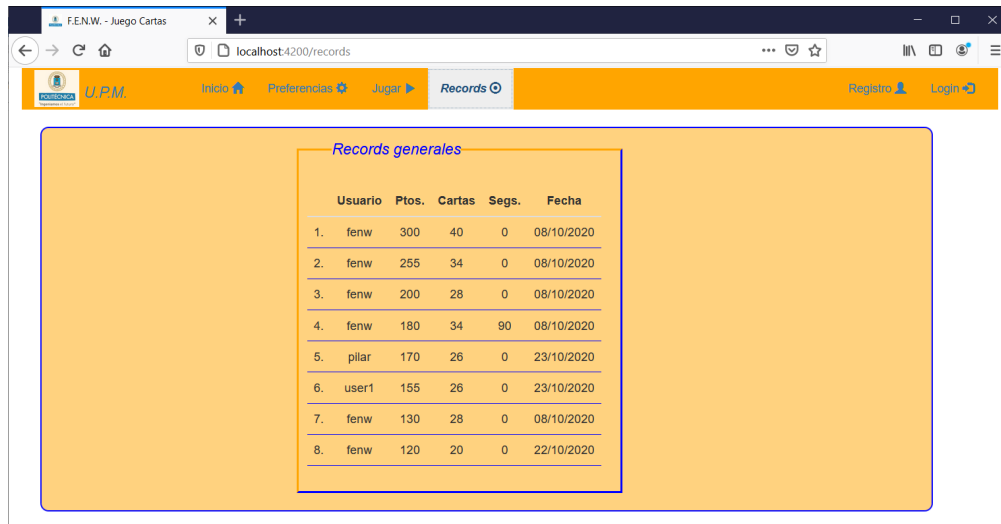


Juego: El jugador pinchará en una de las imágenes, que se dará la vuelta mostrando su contenido. Esta carta se mantendrá en esta posición hasta que el jugador pulse una segunda carta, que se dará la vuelta también cuando el jugador la pulse. Si ambas cartas son iguales (mismo contenido de la imagen), entonces ambas cartas se quedarán dadas la vuelta (mostrando su contenido) y se **sumarán 15 puntos** a la puntuación del jugador; si son diferentes, ambas volverán a su estado inicial (mostrando su reverso) al cabo de 700 milisegundos desde que se ha mostrado la segunda y se **restarán 5 puntos** de la puntuación del jugador. El juego se termina o bien cuando se haya terminado el tiempo (se debe impedir que el jugador pueda seguir jugando cuando haya terminado la cuenta atrás) o bien, si no hay límite de tiempo, cuando se hayan descubierto todas las parejas y todas se hayan emparejado.

Al acabar, debe aparecer la puntuación final que se debe corresponder con la **puntuación obtenida al jugar a la que se suman las cantidades:**

- **25 puntos** si se han elegido 26 cartas, **50 puntos** si se han elegido 32 cartas.
- **100 puntos** si se han elegido 60 segundos, **75 puntos** si son 90, **50 puntos** si son 120 segundos y **25 puntos** si se han elegido 150 segundos. Evidentemente, la opción “sin límite de tiempo” no tiene bonificación.

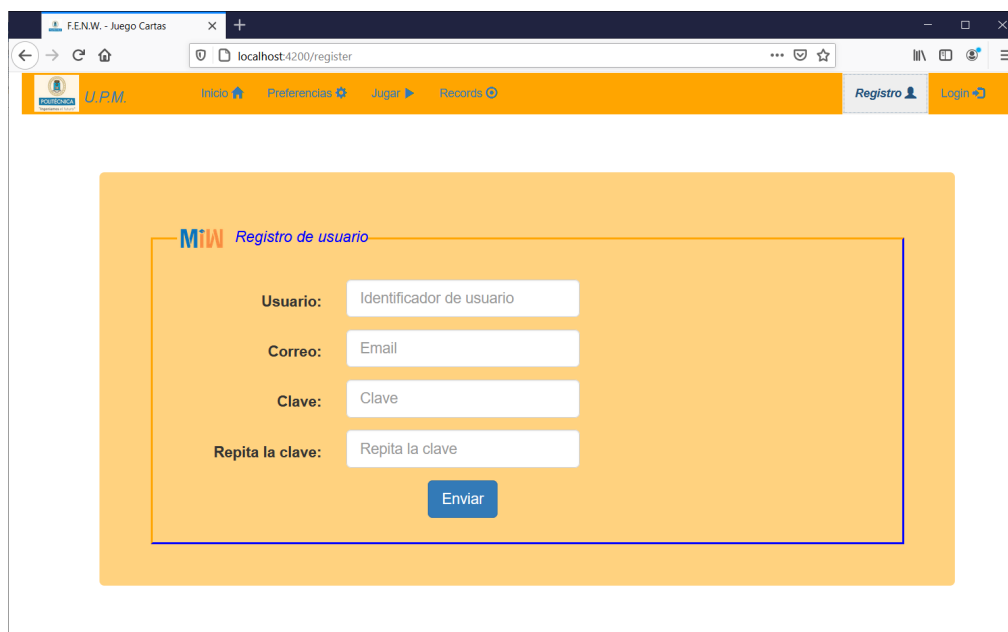
Records. – Esta opción presenta al jugador un listado con los 10 records más altos que hay en el servidor (si no hay 10 registrados se mostrarán los que hay). Para ello se solicitará dicha lista al servidor tal y como se indica en la documentación de la API REST (método **get** en <http://fenw.etsisi.upm.es:10000/records>).



Records generales

	Usuario	Ptos.	Cartas	Segs.	Fecha
1.	fenw	300	40	0	08/10/2020
2.	fenw	255	34	0	08/10/2020
3.	fenw	200	28	0	08/10/2020
4.	fenw	180	34	90	08/10/2020
5.	pilar	170	26	0	23/10/2020
6.	user1	155	26	0	23/10/2020
7.	fenw	130	28	0	08/10/2020
8.	fenw	120	20	0	22/10/2020

Registro. – Permite que un usuario del sistema se pueda registrar en el mismo, lo que le dará capacidad para poder registrar sus puntuaciones. Un usuario que quiera registrarse en el sistema deberá introducir un **nombre de usuario, una dirección de correo y una clave para poder identificarse** (todas obligatorias). En esta práctica **NO** se desarrollará el proceso asociado a esta opción en la API REST, aunque **sí se implementará el formulario** con objeto de prepararlo para la siguiente práctica.



Registro de usuario

Usuario:

Correo:

Clave:

Repita la clave:

Login: Esta opción permitirá a un usuario registrado poder acceder al sistema para proceder a registrar sus puntuaciones. Para poder realizar el “login”, el usuario deberá identificarse mediante un nombre de usuario acompañándolo por la palabra clave que también registró:

The screenshot shows a web browser window with the address bar displaying 'localhost:4200/login'. The page has a navigation bar with links: 'Inicio', 'Preferencias', 'Jugar', 'Records', 'Registro', and 'Login'. The main content area features a login form with the title 'Identificación de usuario'. The form includes two input fields: 'Usuario' with the placeholder text 'Identificador de usuario' and 'Clave' with the placeholder text 'Password'. Below these fields is a blue button labeled 'Enviar'.

En esta práctica **NO** se desarrollará el proceso asociado a esta opción en la API REST, aunque **sí se implementará el formulario** con objeto de prepararlo para la siguiente práctica.

El funcionamiento de las opciones de **“Registro”** y **“Login”** se detallarán en el enunciado de la siguiente práctica.

Se pide:

Se debe entregar (subir a la plataforma) un fichero comprimido que contenga todos aquellos elementos para que, una vez descomprimido, la práctica pueda visualizarse sin problemas. La descompresión del fichero debe dar lugar a la jerarquía de directorios necesaria para situar cada fichero donde corresponda, incluyendo aquellos que puedan pertenecer a las librerías (bootstrap, jquery, etc) si no se utiliza CDN para su incorporación.

Se pueden utilizar libremente aquellos elementos de librerías que aporten elementos no estándar de HTML o CSS (incluido Bootstrap).

CALIFICACIÓN

- ✓ La corrección de la práctica se realizará valorando los aspectos indicados en la rúbrica correspondiente publicada en la plataforma.