

FRONT-END PARA NAVEGADORES WEB

CONJUNTO DE PRÁCTICAS – 2018/2019

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 gestionará un club de pádel. 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 club de pádel y la parte de “login” 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.

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 reservas de pistas para poder jugar al pádel). 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:5555> (Nótese el uso del puerto 5555)

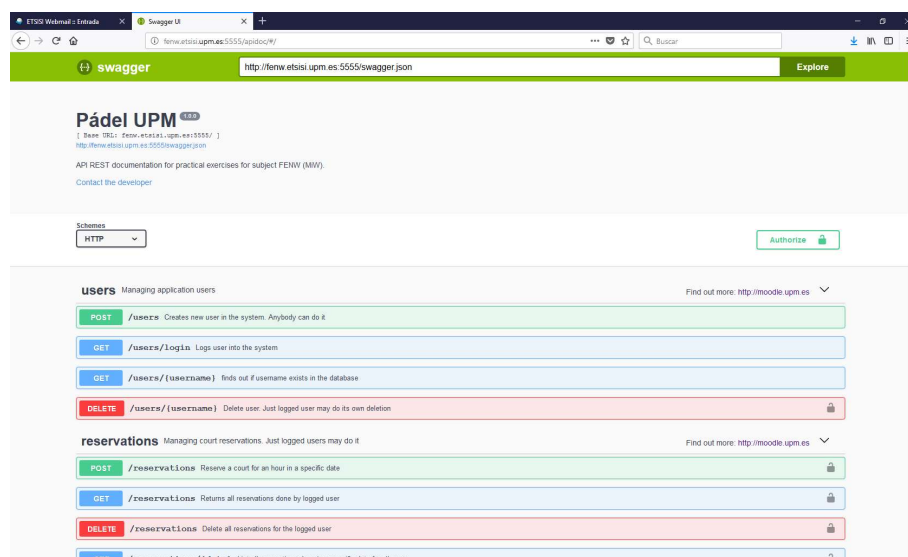
La API se encuentra suficientemente documentada a través de la herramienta “swagger” en la url <http://fenw.etsisi.upm.es:5555/apidoc>

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

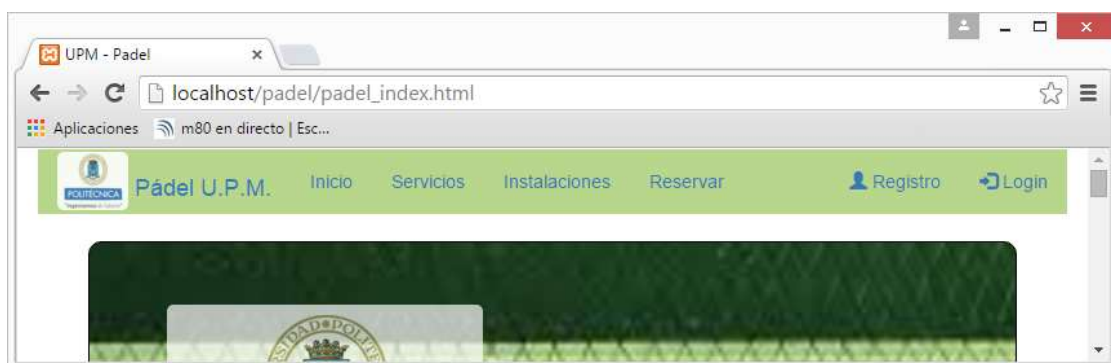
proporciona como respuesta; y por último, probar que se entiende su funcionamiento a través de la ejecución de los métodos disponibles en la misma página.

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, con el nombre indicado en la documentación (esto es, en esta API REST no se pasa ningún parámetro en formato json). 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.



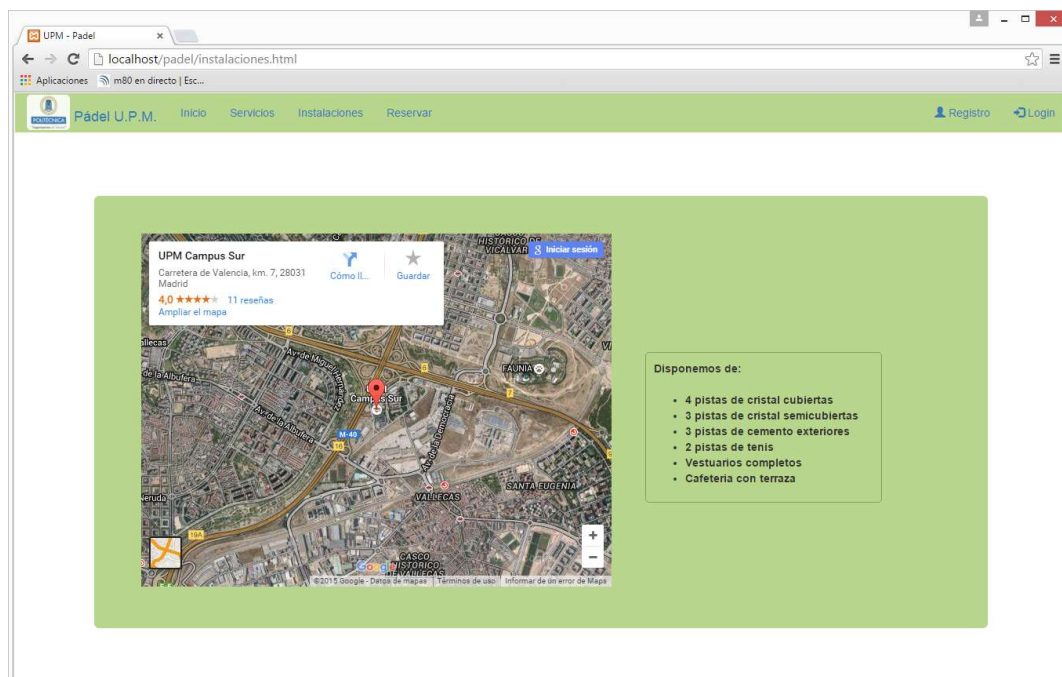
La página principal del sitio web mostrará una presentación a gusto del diseñador, tratando de recrear en la medida de lo posible, un caso real. En esta primera práctica, aunque se describa la funcionalidad del sistema, hay que tener en cuenta que las opciones de “Registro” y “Reserva” que se van a presentar **NO** tienen 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 información genérica del club



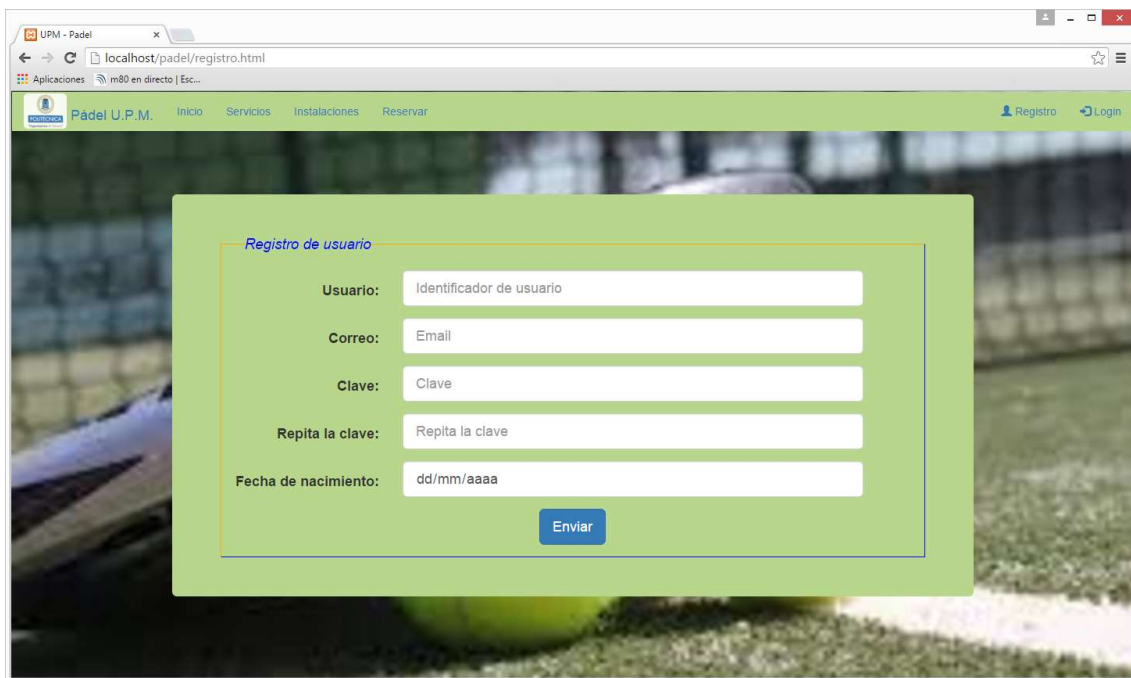
Instalaciones. – Esta opción dará paso a una página en la que se mostrará, inmerso en un mapa de google, la ubicación de las instalaciones y enumerará los recursos de los que dispone el club (de manera estática):



Servicios (o similares). – El alumno puede sentirse libre de incorporar en esta opción o aquellas que considere convenientes, cualquier elemento de diseño que aporte valor extra a su práctica.

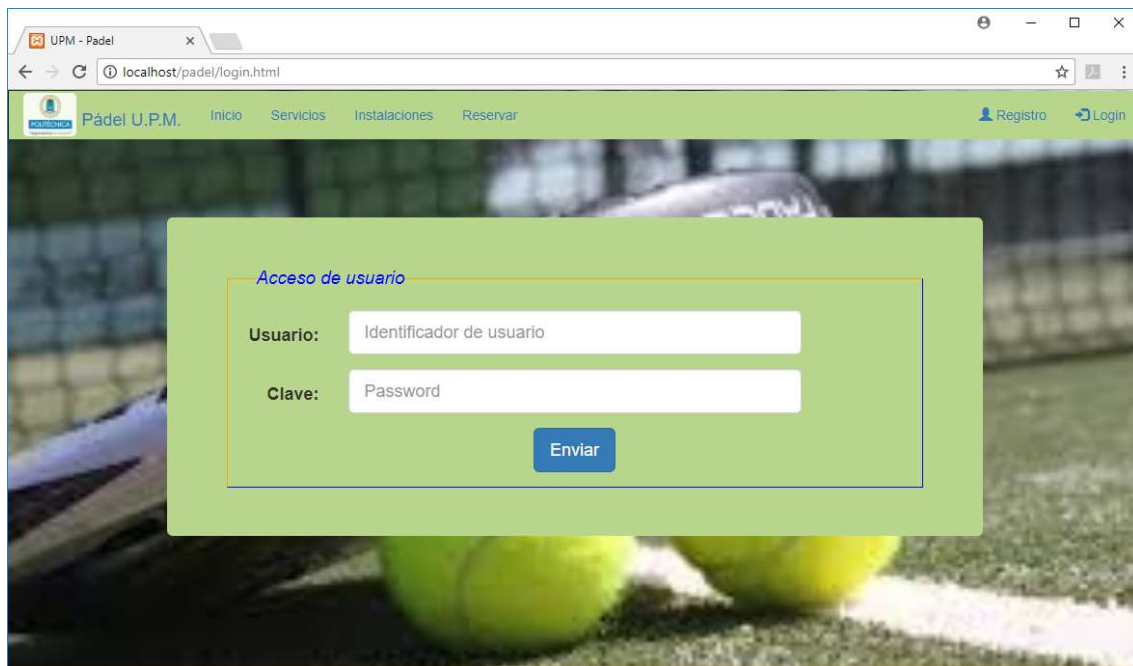
Registro. – Permite que un usuario del sistema se pueda registrar en el mismo, lo que le dará capacidad para poder realizar reservas de pistas para jugar. Un usuario que quiera registrarse en el sistema deberá introducir un **nombre de usuario, una dirección de correo, una clave para poder identificarse y su fecha de nacimiento** (todas obligatorias menos la fecha de nacimiento, que será optativa). Dado que en la siguiente práctica nos ocuparemos de que el registro sea efectivo, el usuario deberá introducir dos veces la clave **y el sistema deberá verificar la coincidencia**.

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.



The screenshot shows a web browser window with the address bar displaying 'localhost/padel/registro.html'. The page has a green header with the logo 'Pádel U.P.M.' and navigation links: 'Inicio', 'Servicios', 'Instalaciones', and 'Reservar'. On the right side of the header are links for 'Registro' and 'Login'. The main content area features a light green box titled 'Registro de usuario' containing a registration form. The form includes the following fields: 'Usuario:' with a placeholder 'Identificador de usuario', 'Correo:' with a placeholder 'Email', 'Clave:' with a placeholder 'Clave', 'Repita la clave:' with a placeholder 'Repita la clave', and 'Fecha de nacimiento:' with a placeholder 'dd/mm/aaaa'. A blue 'Enviar' button is located at the bottom right of the form.

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



Funcionalidad de la opción de “login”

Una vez pulsada la opción de *login*, al usuario le debe aparecer el formulario diseñado. El sistema deberá obligar al usuario, antes de proceder al envío de los datos, a rellenar el campo de identificador de usuario y la palabra clave.

Dado que el servidor al que se accede nos proporciona una API REST, el sistema deberá abortar el envío normal del formulario y proceder a la petición asíncrona. Los parámetros, tal y como se puede ver en la especificación de la API REST serán **username** y **password**, en la *query*, en formato *urlencoded* (no json). Igualmente se puede comprobar en dicha especificación, que la url a la que realizar la petición resultaría ser: <http://fenw.etsisi.upm.es:5555/users/login>.

Ante una identificación positiva, la API REST El sistema devuelve un *token* de seguridad (JWT) en la **cabecera** *http* de la respuesta denominada **Authorization**. Tal y como determina el estándar, el contenido de la cabecera será el *token* precedido de la cadena “**Bearer**” y un **espacio en blanco**. Este *token* habrá sido generado por el servidor de la API REST con una validez de **10 minutos**. El alumno deberá guardarlo de manera local durante la sesión de usuario ya que tendrá que enviarlo en dicha cabecera (a la API REST) cada vez que desee acceder a alguna de las opciones que necesitan de la identificación (en esta práctica ninguna, pero la opción de *Reserva* necesitará de identificación previa).

Atención: Normalmente el sistema no devolvería ningún mensaje en el cuerpo (código http 200 –OK). Sin embargo en este caso y a modo de comprobación, la API REST proporciona una copia del *token* recibido en el cuerpo del mensaje. Esta copia solamente deberá ser utilizada como comprobación, si es necesario, durante la fase de desarrollo de la práctica.

En el caso de una autenticación positiva, y habiendo recibido el token de seguridad, el sistema **deberá almacenar de manera local dicho token** ya que, dado que la API proporcionada sigue las especificaciones REST, el sistema deberá enviarlo a modo de identificación cuando se desarrolle la parte de la opción de “**Reserva**” en la **siguiente práctica**.

Si el usuario no se ha identificado de manera correcta, el sistema **debe advertírsele** mediante el mensaje correspondiente mientras que si se ha recibido el “token” de seguridad con normalidad, en el menú de opciones del sistema, la opción de “login” debe cambiarse por “logout” o “desconexión” que funcionará, cuando sea seleccionada, **destruyendo de manera local** el token.

ATENCIÓN: El alumno podrá **probar** el funcionamiento de su código con los usuarios y claves que se detallan a continuación:

Usuario: **user1** clave: **user1**

Usuario: **user2** clave: **user2**

Logout: Esta opción, que no producirá ninguna llamada a la API REST, **destruirá el token en la parte cliente** y enviará al usuario a la página de Inicio, con la opción de volver a hacer *login*.

El funcionamiento de las opciones de “**Registro**” y de “**Reserva**” se detallará 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.