

Práctica 2: Javascript

1. Objetivos

- Realizar una primera toma de contacto con **JavaScript**.

Grupo	Sesión 1	Sesión 2
P2 (Martes)	12/03	19/03
P1 (Miércoles)	13/03	20/03
P3 (Jueves)	07/03	14/03

Ponderación nota final de prácticas: 10 %

2. Descripción de la práctica

El objetivo de la práctica es el diseño de un espacio para comentarios que los usuarios del sitio pueden hacer asociados a cada uno de las actividades mostradas en el sitio. Los cambios han de realizarse en la página `actividad.html`. Debe tener la siguiente funcionalidad:

- El panel de comentarios debe estar oculto en un lateral, y desplegarse al pulsar un botón (o imagen) de comentarios.
- Cuando se despliegue, deberá contar con dos comentarios predefinidos (como si hubieran sido introducidos por usuarios anteriores de la web).
- Cada comentario tendrá **autor**, **fecha y hora** del comentario y **texto** del comentario.
- A continuación de los dos comentarios existentes, habrá un formulario con los siguientes campos:
 - Nombre.
 - E-mail.
 - Texto del comentario.
 - Botón de enviar.
- Al pulsar el botón de enviar, el texto del comentario se incluirá de forma similar a los ya existentes a continuación de estos usando **JavaScript**. Obviamente, al no estar guardando los comentarios en ningún sitio (ni base de datos ni nada similar) dichos comentarios nuevos desaparecerán al recargar la página.

- Al pulsar el botón de enviar, *antes* de incluir el texto del comentario se comprobará que todos los campos han sido rellenados. En caso negativo se avisará con un dialogo modal y *NO* se incluirá el comentario. Además se realizará también una validación del e-mail (que sea un e-mail que parezca válido).
- Conforme el usuario escribe el comentario, el sistema detectará mediante **JavaScript** la aparición de palabras “prohibidas” (se pueden definir hasta 10 palabras prohibidas en el script). Cada uno de los caracteres de cada palabra prohibida se sustituirá por un * en tiempo de escritura.

3. Opcional (para subir nota)

Hoy en día es muy típico usar bibliotecas o *frameworks* CSS para crear nuestros diseños. En esta práctica se propone incluir alguna funcionalidad de algún *frameworks* CSS como **Bootstrap**. Por ejemplo, el formulario propuesto (y/o los diálogos modales) podrían hacerse con alguno de los componentes de **Bootstrap** (ver la documentación al respecto).

Además, si en la primera práctica no te gustó como quedó el aspecto de tu página o no es *responsive*, puedes usar los *layouts* de **Bootstrap** para mejorarla. Si conoces o prefieres otro *framework* CSS distinto a **Bootstrap**, puedes usarlo.

4. Bibliografía básica

- Javascript Tutorial: <https://www.w3schools.com/js/default.asp>
- Guía de JavaScript: <https://developer.mozilla.org/es/docs/Web/JavaScript/Guide>
- Documentación de Bootstrap: <https://getbootstrap.com/docs/5.0/getting-started/introduction/>

5. Evaluación de la práctica

Se tendrán en cuenta los siguientes aspectos:

- Se cumple con toda la funcionalidad descrita.
- Se realizan correctamente las validaciones.
- El código está bien estructurado, separando correctamente los scripts del HTML.

6. Aclaración a la práctica

Se pide que el campo de comentarios “evite” que se pongan palabras “prohibidas” sustituyendo dichas palabras por unos asteriscos (*). Pero esa sustitución se debe hacer *MIENTRAS* el usuario está escribiendo en el campo, no cuando pulsamos el botón de enviar (que es cuando haremos las otras comprobaciones

del email correcto y que no haya campos vacíos). Por ejemplo, si una palabra prohibida fuera **tractor**, en el momento en el que el usuario pulse la tecla **r** que finaliza la palabra es cuando está se sustituiría por *********. Para eso se puede hacer uso de los eventos **onchange**, **onkeyup** o similares que podemos asignarle al campo de texto.

Está claro que hay casuísticas en las que el sistema en su forma más básica no va a funcionar bien. Ejemplo típico: Si una palabra prohibida fuera **puta**, cuando quisiéramos escribir **diputación** acabaríamos teniendo **di****ción**. Esto se puede arreglar teniendo en cuenta que la palabra estuviera separada por otros caracteres que no fueran letras (antes y después), pero esa implementación más sofisticada **no** se pide en la práctica.