

Proyecto JavaScript tienda de ropa

Vamos a hacer una web de compra de productos usando HTML, JavaScript y SASS. Para ello vamos a usar la API <https://fakestoreapi.com/> para simular un servidor backend.

Requisitos generales:

- Siguiendo la filosofía AJAX, al interactuar con la página, las llamadas a la API que se hagan no deben producir recarga de página completa, sólo cambio de los componentes afectados.
- Debe estar en producción, es decir, alojada en algún servidor. (Aquí tienes una lista de algunos con plan gratuito: <https://stackdiary.com/free-hosting-for-developers/>)

Diseño: (3)

- Usar etiquetado HTML sintáctico
- Ser visualmente moderna y agradable
- Estar maquetada usando flex y grid
- Totalmente adaptable usando media queries ó flex/grid y clamp
- Usar SASS como lenguaje de maquetado con las estructuras y la organización que hemos visto.
- Usar iconos de alguna biblioteca de node como, por ejemplo, remixicons fort-awesome o feather icons para colocar, al menos, iconos de me gusta, no me gusta y favorito en cada producto, un icono para el carrito de la compra así como iconos de redes sociales en el pie de página.
- Usar node para automatizar el flujo de trabajo de prefijado, traducción y minimización
- Un menú principal con las secciones Registrar/iniciar sesión, carrito de la compra, un enlace por cada tipo de producto además de uno con información sobre cómo contactar con la empresa, donde se incluya un formulario para ponerse en contacto con la empresa (con todos sus campos validados) y un mapa, usando la API que prefieras, que permita mostrar dónde está la tienda física.

Funcionalidad: (7)

- La API proporciona algunos usuarios de prueba, pero no permite grabar nuevos. La aplicación web debe permitir iniciar sesión con los de prueba y registrar los propios mediante un formulario de registro. Éste debe pedir los datos típicos de nombre, apellidos, teléfono, DNI y edad. Hay que validar cada campo del formulario y no se permite pasar al siguiente hasta que se valide el actual. Los usuarios registrados de forma local deben almacenarse en localStorage
- Se debe presentar los productos al cliente en forma de lista o de tabla, como él elija, siendo posible el cambio en cualquier momento.
- Se debe usar una técnica de desplazamiento infinito para cargar productos.
- Se debe permitir que el usuario ordene los productos por categoría, ascendente o descendentemente.
- El listado de productos debe mostrar nombre, imagen y precio.
- Al pinchar en un producto, se debe mostrar una pantalla con más detalles sobre el mismo.
- Se debe poder agregar el producto a un carrito de compra, bien desde los detalles o bien desde el listado.

- Debe ser posible acceder al carrito para simular el pedido.
- Debe producirse alguna animación sencilla en la propia tarjeta del producto al pasar el cursor encima y otra en el carrito al añadirlo.
- El sistema debe recordar los productos favoritos del usuario y cuántas veces se ha pulsado cada icono de me gusta y no me gusta
- El carrito debe mostrar una línea por artículo con las unidades que el usuario haya añadido, permitiendo quitar o añadir unidades, incluso borrar el producto.
- Se debe almacenar en local el carrito de la compra y el inicio de sesión, para que se muestre tal y como lo dejó si cierra el navegador y vuelve.
- Usando la biblioteca nodemailer, la web debe mandar correos al usuario al registrarse y al finalizar pedido (puedes usar <https://emailfake.com> para crear un correo falso de prueba)

La entrega del proyecto se hará a través de un repositorio github llamado “apellidos nombre proyecto JavaScript DWEC”, indicando en el README la URL en producción. La nota final se consigue tras la defensa personalizada del proyecto para comprobar que el alumno entiende perfectamente su código y es capaz de realizar cambios sencillos conforme se le piden. La incapacidad para defender cualquier parte de su propio código supone el suspenso del proyecto.

Cualquier cosa no especificada se atiene a lo comentado y trabajado en clase