

Proyecto Integrador: *Juguetería Cósmica*

Etapa 2:

Consideraciones generales

- Fecha máxima para presentar el proyecto: jueves 22/09/2022 en cualquier horario.
- Todo el contenido desarrollado para esta segunda etapa deberá ser compatible con lo solicitado para la primera. Siguiendo con esa línea, podrán continuar haciendo todos los cambios que crean pertinentes o que no hayan llegado a hacer para la primera etapa. Esto incluye modificaciones en los estilos, agregado de elementos en las distintas vistas o incluso un rediseño completo, siempre y cuando se cumplan con todos los requisitos detallados para ambas etapas.
- No se requiere utilizar Node.js en esta entrega. Sin embargo, se admite su uso. En caso de utilizarlo, Los archivos entregados no deberán incluir el directorio con los node modules, pero sí el package.json correspondiente.

Funcionamiento del sitio

- Deberá elaborarse una SPA que permita acceder a cada una de las vistas, recargando solo el contenido del elemento `<main>`, manteniendo en todo momento los elementos `<header>` y `<footer>` (hijos de `<body>`) en el DOM, es decir, sin recargarlos.
- La comunicación con el servidor puede intercambiar datos de cualquier tipo, ya sea texto o HTML crudo, JSON, archivos binarios, etc, en cualquier vista. Podrán obtenerse datos del servidor, de forma asincrónica, no solo para las vistas principales, sino en cualquier momento en donde se requiera.
- Se podrá utilizar Live Server o Node.js + Express como servidor web (solo uno de ellos), a elección del alumno.
- Deberá utilizarse `fetch()` con `async/await` para las conexiones asincrónicas con el servidor.
- Al navegar por el sitio la URL debe ir cambiando. Todas las URL visibles por el usuario deben ser amigables y al acceder directamente a una deberá mostrarse el contenido correspondiente. Idealmente destacar de alguna manera el vínculo correspondiente a la vista activa.
- Todas las acciones que vaya realizando el usuario deben brindar algún tipo de feedback. Ej: los vínculos y botones deben reaccionar ante `hover`, `focus` y `active`, etc.
- Debe tenerse en cuenta que no todas las personas razonan igual, por lo que no siempre van a entender lo mismo ante los mismos elementos, por lo que es buena idea agregar ayudas contextuales, como atributos `title`, mensajes emergentes, u otro tipo de indicadores que marquen qué sucede al presionar tal botón o luego de realizar ciertas acciones, etc.

- Todas las acciones se deben poder realizar en cualquier resolución de pantalla y tipo de dispositivo. No olvidar los distintos tipos de interacción que existen en cada tipo de dispositivo (ej: pocos dispositivos táctiles reaccionan ante el hover). Recordar acomodar los elementos apropiadamente no solo en resoluciones chicas y medias, sino también en resoluciones muy grandes.

Elementos comunes a todo el sitio

Botón de carrito

Ubicado físicamente dentro del elemento `<header>` del `<body>`, podrá mostrarse arriba o en cualquier lugar de la página, gracias a los distintos mecanismos que CSS provee para tal fin. El botón deberá mostrar y ocultar, de manera alternada, un contenedor del tipo *modal* en el que se verá el listado de productos que se hayan agregado al carrito.

Carrito modal

Se mostrará cuando el botón del carrito del header sea presionado.

El diseño y tamaño del modal es libre, pudiendo incorporarse de manera centrada, en un lateral, etc. En todos los casos deberá funcionar como *modal*, por encima de los demás elementos de la página, de manera flotante.

Deberá funcionar en todas las páginas y su cierre se podrá efectuar de todas estas maneras:

- Presionando nuevamente en el botón del carrito
- Presionando en una x de cierre incorporada dentro del modal
- Presionando la tecla ESC del teclado
- Haciendo click en cualquier zona fuera del modal.

Dentro del modal se mostrarán los productos que hayan sido agregados al carrito a partir de las cards. Cada producto agregado al modal mostrará su foto, nombre, precio, cantidad y subtotal, es decir, su precio multiplicado por la cantidad de veces que haya sido agregado. Adicionalmente mostrará el total (la suma de todos los subtotales).

Cada producto mostrado en el modal permitirá ser eliminado del mismo, actualizando el monto total, en consecuencia. Optativamente, se podrán agregar botones para sumar o restar la cantidad o editar directamente el número de unidades. Los eventos necesarios para llevar a cabo estas acciones deberán estar programados sobre el modal u otro contenedor, y no sobre cada producto, mediante el uso de delegación de eventos, con el fin de evitar que cada producto tenga registrado un evento por separado.

Vistas individuales

Todas las vistas deberán recibirse de manera asíncrona, tal y como se definió previamente.

Home

Dentro de la misma, se mostrará el listado de cards, al igual que se hizo en la primera etapa, pero siendo las mismas generadas en el HTML a partir de un array definido con JS, es decir que las cards serán generadas de forma programática. Se deberá utilizar Handlebars para la creación de las mismas. Cada card deberá contar con un botón que permita agregar el producto que se detalla en ella al carrito. Al hacerse esto se deberá informar al usuario con un mensaje, no invasivo, que se agregó el producto al carrito. Al suceder esto, se adicionará el producto al modal del carrito como uno nuevo, si no existía previamente. Al agregar al carrito más unidades del mismo producto, se deberá incrementar la cantidad de unidades dentro, sin repetir el producto, actualizando el subtotal y total.

Adicionalmente, y de forma optativa, cada card puede contar con otros elementos interactivos, como botones, u otro tipo de respuesta (hover, etc), que permita adicionar información a la misma, como puede ser una descripción larga del producto, fotos adicionales, etc, las cuales podrán cargarse de forma asincrónica, según sea necesario.

Deberá utilizarse delegación de eventos, evitando que cada card tenga registrado un evento por separado, programando solo a un contenedor para que reaccione cuando sea presionando o cuando corresponda, verificando sobre qué botón descendiente se hizo click, por ejemplo, y tomando las acciones que sean necesarias.

Alta

El formulario definido deberá contar con las validaciones que se detallan a continuación. Además de los requeridos, se podrán agregar otros campos al formulario, si lo desean. Cada campo, al perder el foco, deberá informar, debajo suyo o a un costado, si tiene errores de validación. Los errores también deberán mostrarse si se intenta enviar el formulario y al menos un campo no cumple con los requisitos. En dicho caso, cada campo deberá mostrar los errores correspondientes y el formulario no deberá enviarse. Solo se permitirá el envío de los datos si todos los campos fueron validados exitosamente. Se deberá mostrar al usuario qué tipo de validaciones se están realizando, con el fin de que sepa qué correcciones debe realizar al completar el formulario.

Nombre

Obligatorio. Campo alfanumérico. Soporta caracteres utilizados en el idioma español, con y sin acentuaciones. También números, puntos, comas, comillas simples y dobles, espacios, barras, guiones medios y bajos. 30 caracteres como máximo.

Precio

Obligatorio. Campo numérico positivo con o sin decimales.

Stock

Obligatorio. Campo numérico positivo o cero, sin decimales.

Marca

Misma validación que para el nombre, pero con 40 caracteres como máximo.

Categoría

Misma validación que para el nombre, pero con 50 caracteres como máximo.

Descripción corta

Obligatorio. Texto libre de hasta 80 caracteres como máximo.

Descripción larga

Obligatorio. Texto libre de hasta 2000 caracteres como máximo.

Envío sin cargo

Puede ser un checkbox, en ese caso sería optativo marcar la casilla.

Si es radio con 2 opciones es obligatorio.

Edad desde

Obligatorio. Campo numérico positivo, sin decimales.*

Edad hasta

Obligatorio. Campo numérico positivo, sin decimales.*

* Idear un mecanismo que permita que el usuario defina si la edad que se está indicando son meses o años. Esto podrá variar de producto en producto.

Foto

Obligatorio. Un único archivo, si se mostrará una sola imagen en las cards. Permitir múltiples si la intención es mostrar múltiples imágenes para cada producto. Permitir elegir solo archivos de imágenes.

Contacto

Deberá contar con el mismo comportamiento que el formulario de Alta, quedando las validaciones correspondientes a criterio del alumno.

De igual manera, además de los requeridos, se podrán agregar otros campos al formulario, si lo desean.

Nombre

A criterio del alumno.

E-mail

A criterio del alumno.

Comentarios

A criterio del alumno.