

INFORME

Final



Año: 2021.

Alumnos: Bettega Nicolas, Costedoat Alexis.

Materia: Aplicaciones para Dispositivos Móviles.

Turno: Noche.

Comisión: A.

Index

Se implementó Vue.js, Vue-Router, Vuetify y Vue Cli para el desarrollo del proyecto final. También, animate.Css para las transiciones de las vistas y transition en el slider principal. En el índice llamamos al componente Nav y Footer, a través de router view mostramos las distintas vistas.



Bienvenido

Te deseamos que tengas una experiencia positiva en tienda Mi Reloj.

Tenemos el objetivo de convertirnos en un referente a nivel nacional en la venta de relojes. A partir de esto, se han incorporado diversas marcas para poder cumplir con las necesidades de nuestros clientes.

¡Participa de la encuesta!

Si quieres obtener un 20% de descuento en tu próxima compra, participa de la encuesta y automáticamente te enviaremos el código de descuento

Nombre	<input type="text"/>	0 / 10
E-mail	<input type="text"/>	0 / 10
Telefono	<input type="text"/> dd/mm/aaaa	0 / 10
Genero	<input type="text"/>	Tipo de reloj
Frecuencia de compra	<input type="text"/>	Factor de compra
Canal de compra	<input type="text"/>	Probabilidad
<input type="button" value="Enviar"/>		

Medios de pago



Efectivo



Transferencia



Deposito



Tarjeta

Desarrollado por:
Nicolas Bettega,
Alexis Costedoat.
Materia: APDM
Turno: Noche.
Comisión: A.
Año: 2021.



Formulario de encuesta

Se desarrolló el componente “FORM” que está compuesto por la siguiente data nombre, mail, teléfono, fecha de nacimiento, genero, tipo de reloj, frecuencia de compra, factor de compra, canal de compra, probabilidad de recomendación todos inicializados en null.

El mismo cuenta con etiquetas de vuetify como por ejemplo V-container, V-row, V-col, V-text-field, etc.

¡Participa de la encuesta!

Si quieres obtener un 20% de descuento en tu próxima compra, participa de la encuesta y automáticamente te enviaremos el código de descuento

Nombre

0 / 30

E-mail

0 / 30

Teléfono

dd/mm/aaaa

0 / 10

Genero

>

Tipo de reloj

<

Frecuencia de compra

>

Factor de compra

<

Canal de compra

>

Probabilidad

<

Enviar

Medios de pago

Efectivo

Transferencia

Deposito

Tarjeta

También un objeto llamado campos el cual aloja el mensaje de error cuando el usuario no ingreso un dato en el campo designado, y propiedad esValido en null, que va a ser usada para validad si el campo esta completo o no, aplicando una clase rojo o verde por medio del v-bind implementado con un ternario en cada campo del formulario.

Ejemplo de Validación campos incompletos con la clase rojo

Nombre 0 / 10

E-mail 0 / 10

Telefono 0 / 10 dd/mm/aaaa

Genero ▼ Tipo de reloj ▼

Frecuencia de compra ▼ Factor de compra ▼

Canal de compra ▼ Probabilidad ▼

Enviar

- El nombre es obligatorio.
- El mail es obligatorio
- El telefono es obligatorio
- El fecha nacimiento es obligatorio
- El campo genero es obligatorio
- El campo tipo de reloj es obligatorio
- El campo frecuencia de compra es obligatorio
- El campo factor de compra es obligatorio
- El campo canal de compra es obligatorio
- El campo probabilidad de compra es obligatorio

Ejemplo de Validación con los campos completos con la clase Verde

Nombre
Nicolas bettega

Email
nicolas96bettega@gmail.com

Teléfono 1168344316 Fecha de Nacimiento 23/01/1996

Genero Masculino Tipo de reloj Digital

Frecuencia de compra Una vez al mes. Factor de compra Promociones

Canal de compra Internet Probabilidad Muy probable.

Enviar

Luego al enviar los datos los mismos son alojados en localStorage por medio del método enviar que es ejecutado por un v-on en el botón de enviar, el cual antes de enviar ejecuta la función validación() que comprueba que cada campo este completo dándole que le retorna a esValido() un booleano que aplica una clase si es True (Verde) || False (Rojo).

Siguiendo con las propiedades del componente pasamos al template que contiene la estructura HTML, SCSS, Vuetify y Vue.js. Cada campo tiene un v-on.blur con la función validación() que recibe el value del campo por medio del v-model. Además cada uno tiene un v-bind con su respectivo ternario que al validad si esta completo de aplica la clase rojo o verde respectivamente.

Nombre
alexis

6 / 10

E-mail
alexiscostedoat@gmail.com

25 / 10

Telefono
1126751814

10 / 10

01/09/1935

Genero
Masculino

Tipo de reloj
Analogico

Frecuencia de compra
Una vez al año

Factor de compra
Procedencia

Canal de compra
Free shop

Probabilidad
Poco probable

Enviar

Encuesta enviada

Esperamos verte pronto en nuestra tienda.
¡Gracias!

Datos enviados

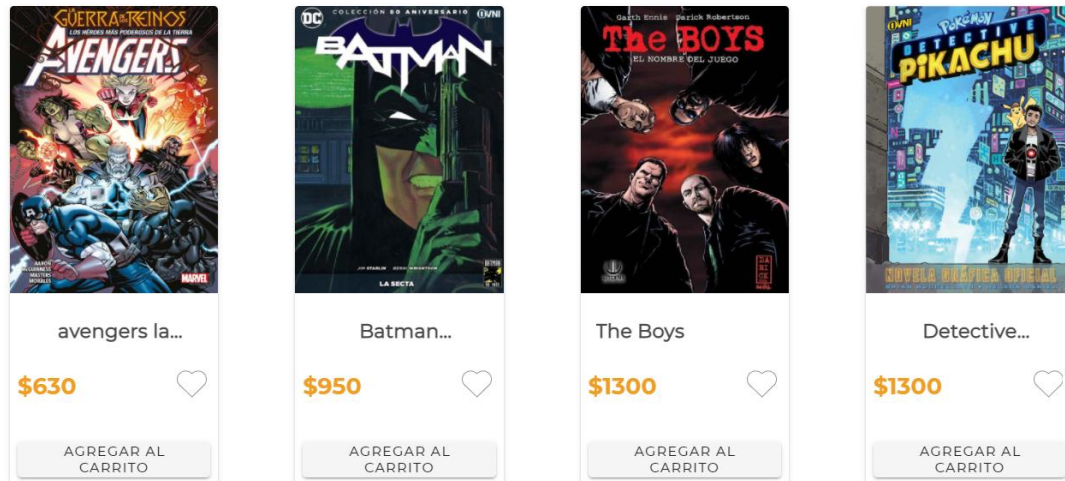
- ALEXIS
- alexiscostedoat@gmail.com
- 1126751814
- 1935-09-01
- Masculino
- Analogico
- Una vez al año
- Procedencia
- Free shop
- Poco probable

Posteriormente se desarrolló la estructura que permite que el usuario vea los mensajes de error o el mensaje enviado con sus datos correspondientes. Esto se comprueba por medio de un V-IF que pregunta si el objeto hayErrores va alojar el valor true mientras haya datos incompletos mostrando los errores por medio de un v-for que recorre el objeto campos y por el contrario va a devolver false que por medio de otro V-IF va a estar mostrando un mensaje de agradecimiento y los datos ingresados por el usuario en los cuales se aplicaron los Vue.Filters Collection de terceros a la hora de mostrar los mismos.

Productos

Se creó una sección llamada productos que contiene un v-for que recorre todos los mismo, obteniéndolos de un archivo externo PHP alojado en un hosting que es consumidos mediante Ajax (Fetch).

Nuestros productos

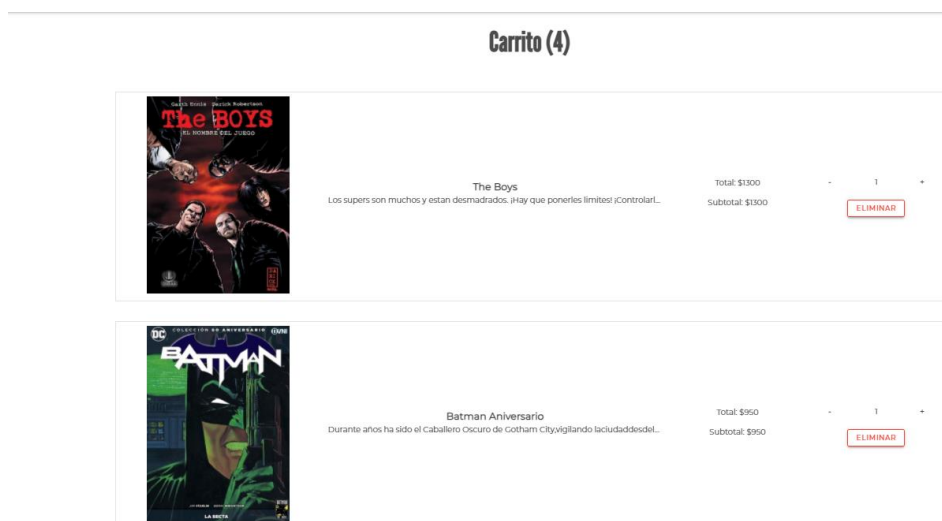


Además, tiene un template con la estructura Vuetify de la tarjeta del producto mostrando así una imagen, el título y precio del mismo.

También tiene methods que permiten agregar productos al carrito y a su vez los guarda en el localStorage. Además, tiene un botón que permite agregar el producto a favoritos.

Carrito

En la sección carrito se llama por medio de Vue Router. En el que se van a mostrar los productos que se guardaron en localStorage desde la sección de productos.



Dentro del carrito se pueden visualizar de forma horizontal los productos en desktop, el mismo presenta btn para sumar o restar productos del carrito y si el usuario desea eliminar el articulo cuenta con esa opción, todos los botones accionan los respectivos methods que actualizan los datos y los guardan nuevamente en el localStorage.

También tiene métodos computed que permiten visualizar de manera dinámica la cantidad de productos guardados en total y el precio total.