

¡Javascript!

guayverd beta hub

Agenda del día



01

Introducción

Repaso Clase 23

02

Cart

Página de carrito.

Item de navegación.

Prohibir acceso a usuarios
anónimos.

Mapear.

Eliminar.

Total.

Vaciar carrito.

03

Ejercitación

Funcionalidad de
página cart.

.



daily

¿Cómo venimos?

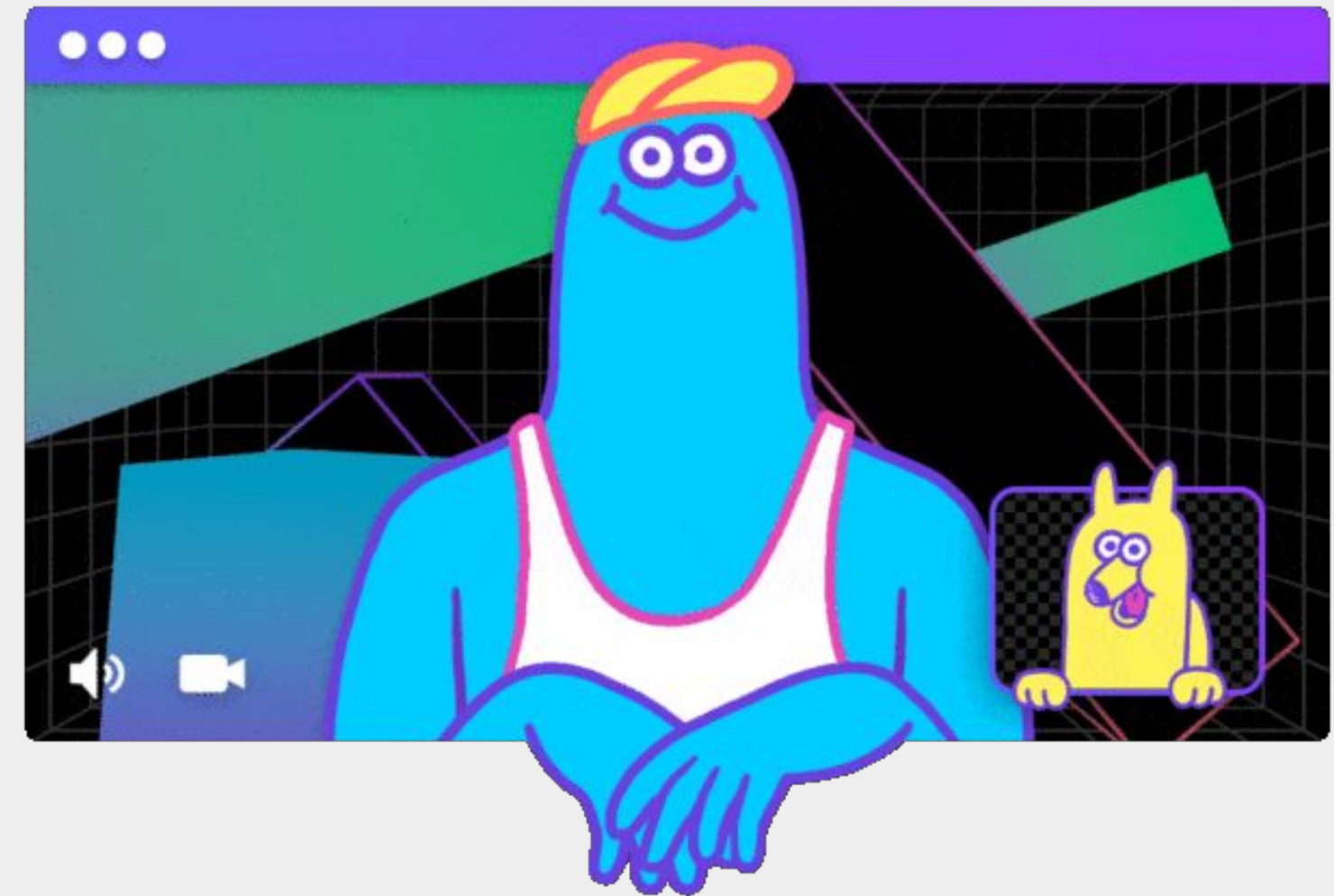
¿Algo nos bloquea?

¿Cómo seguimos?



Página de Cart

App




SPRINT 3


Página de Cart

Avantt Cars

Home


Hola, rick@morty.com |  5 | Cerrar sesión

Carro de compras



Twice in a Lifetime

Precio	Cantidad	Total	Eliminar
\$55	4	\$220	



CSNY Déjà Vu

Precio	Cantidad	Total	Eliminar
\$52	1	\$52	

Seguir comprando

Checkout

Orden de Compra

Total :

\$ 745.2

Primero Repasemos



Querido diario:

Ya no quiero ser adulto,
todo está muy caro.

Mis estados de ánimo



Repasemos



Flirting With Disaster

Fusce posuere felis sed lacus. Morbi sem mauris, laoreet ut, rhoncus aliquet, pulvinar sed, nisl. Nunc rhoncus dui vel sem. Sed sagittis. N...

Precio: \$56

Stock: 38

Categoría: Vintage

+

1

-

Agregar al carrito

js/producto.js

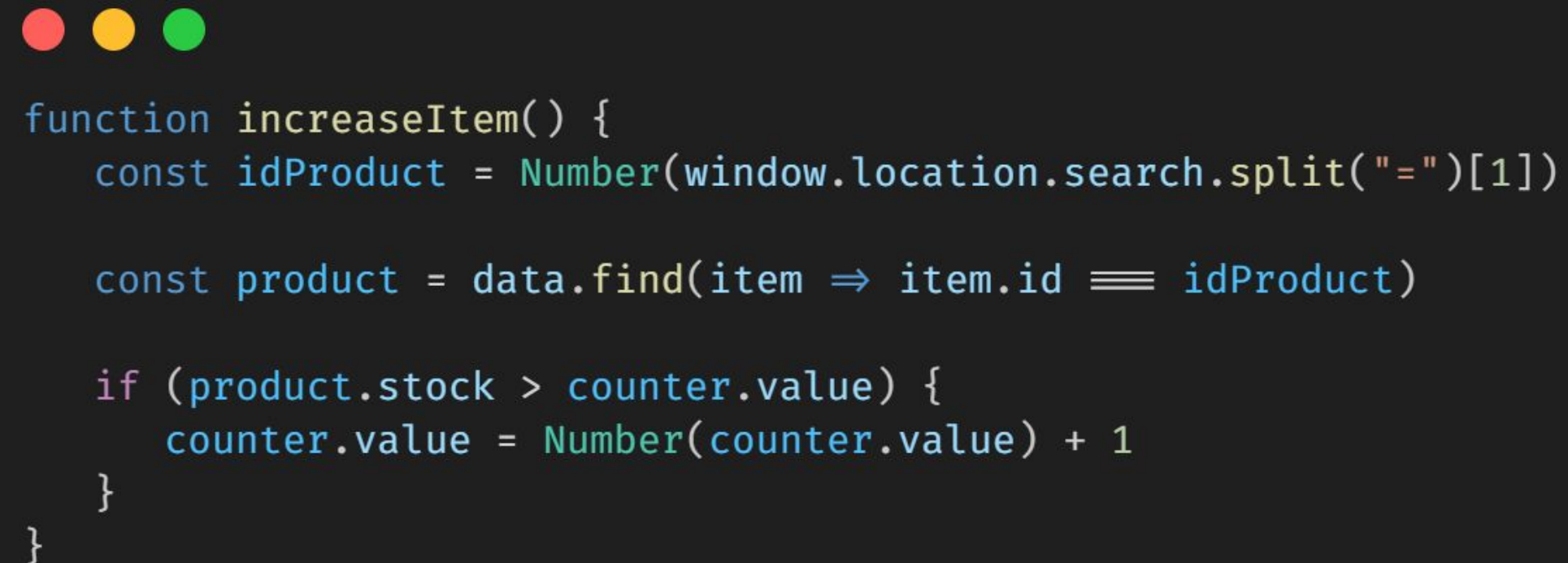
Agregamos a los botones eventos y controladores de eventos.



```
<div class="input-group">
  <button class="btn btn-danger" type="button" onclick="increaseItem()">+</button>
  <input type="number" class="form-control" value="0">
  <button class="btn btn-danger" type="button" onclick="decreaseItem()">-</button>
</div><a href="#" class="btn btn-primary col-12" onclick="addItem()">Comprar</a>
```

increaseItem


Esta función debe tener en cuenta el stock de producto para no superar ese valor.



```
function increaseItem() {  
  const idProduct = Number(window.location.search.split("=")[1])  
  
  const product = data.find(item => item.id === idProduct)  
  
  if (product.stock > counter.value) {  
    counter.value = Number(counter.value) + 1  
  }  
}
```

decreaseItem

Esta función debe tener en cuenta que el valor no puede ser menor a 1.



```
function decreaseItem() {  
  if (1 < counter.value) {  
    counter.value = Number(counter.value) - 1  
  }  
}
```

Carrito en nav

Avantt Cars Home

Hola, rick@morty.com |  0 | Cerrar sesión

Carrito en nav

Agregamos la imagen y llamamos a un ítem de localStorage que aún no creamos en js/nav.js.



```
<span>|</span><li><b>${localStorage.getItem(
  "quantity"
)}</b></li><span>|</span>
```

Creamos ítems cart y quantity al iniciar sesión

cart será un array de productos, por ahora vacío. quantity, un string numérico.

```
const handleSubmitLogin = ev => {  
  ev.preventDefault()  
  let emailForm = ev.target.elements.email.value  
  let passwordForm = ev.target.elements.password.value  
  
  if (emailForm === USER_LOGIN.email && passwordForm === USER_LOGIN.password) {  
    localStorage.setItem("email", emailForm)  
    ▶ localStorage.setItem("cart", JSON.stringify([]))  
    ▶ localStorage.setItem("quantity", "0")  
    location.href = "./index.html"  
  } else {  
    message.innerText = "Por favor introduce credenciales válidas."  
  }  
}
```

addItem

- Esta función debe tener en cuenta el item “cart” de localStorage.
- Agregar el producto y su cantidad si no existe y si existe sumar las cantidades.
- Actualizar los items “cart” y “quantity” en localStorage.
- Actualizar el valor en el nav.
- Actualizar el valor del input en 1.

```
function addItem() {
  let cart = JSON.parse(localStorage.getItem("cart"))


  const idProduct = Number(window.location.search.split("=")[1])
  const product = data.find(item => item.id === idProduct)
  const existeIdEnCart = cart.some(item => item.product.id === idProduct)

  if (existeIdEnCart) {
    cart = cart.map(item => {
      if (item.product.id === idProduct) {
        return { ...item, quantity: item.quantity + Number(counter.value) }
      } else {
        return item
      }
    })
  } else {
    cart.push({ product: product, quantity: Number(counter.value) })
  }

  localStorage.setItem("cart", JSON.stringify(cart))
  let quantity = cart.reduce((acumulado, actual) => acumulado + actual.quantity, 0)
  localStorage.setItem("quantity", quantity)
  const quantityTag = document.querySelector("#quantity")
  quantityTag.innerText = quantity
  counter.value = "1"
}
```

Página cart.html

Descarga el template de: https://github.com/julioavantt/guayerd_login/blob/main/cart.html .

	Waterproof Mobile Phone		
	Precio	Cantidad	Total
	\$450	2	\$900

Seguir comprando

Checkout

Orden de Compra

Total :	\$ 745.2
---------	----------

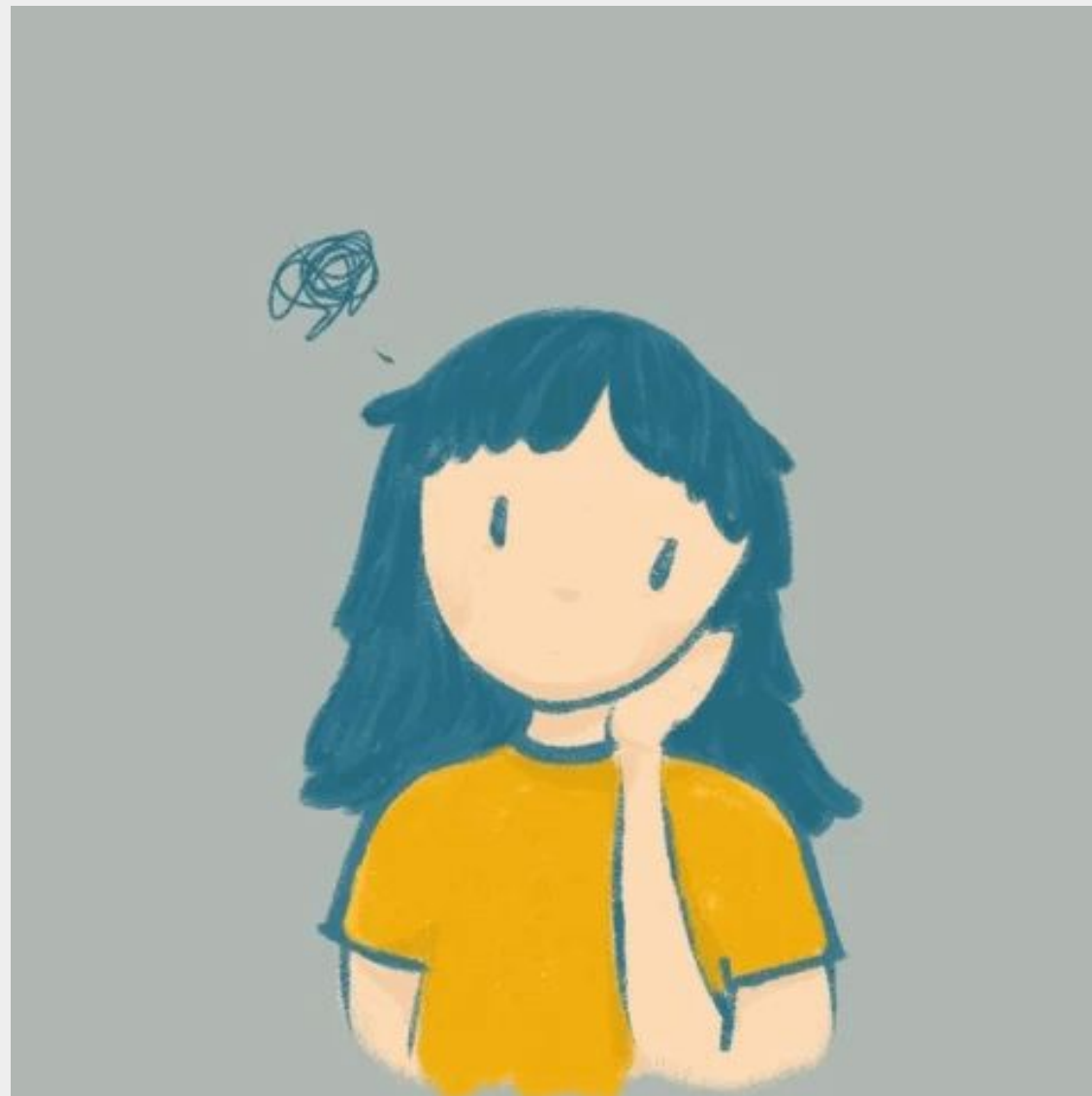
Vincular el carrito del nav con la página

No sé qué más poner...



js/isNoAuthRedirectHome.js

Creamos la función que nos redirige a la Home si intentamos ir a cart.html y no tenemos sesión **activa**.



Veamos nuestro item cart

Es un string al que debemos aplicar JSON.parse para convertirlo en un array de objetos.

```
> localStorage.cart  
< ' [{"product":{"id":2,"title":"Twice in a Lifetime","detail":"Cras mi pede, ma  
lesuada in, imperdiet et, commodo vulputate, justo. In blandit ultrices enim.  
Lorem ipsum dolor sit amet, consectetur adipiscing elit.\\n\\nProin interdum  
mauris non ligula pellentesque ultrices. Phasellus id sapien in sapien iaculi  
s congue. Vivamus metus arcu, adipiscing molestie, hendrerit at, vulputate vi  
tae, nisl.\\n\\nAenean lectus. Pellentesque eget nunc. Donec quis orci eget o  
rci vehicula condimentum."},"price":55,"stock":36,"category":"Vintage"},"quant  
ity":3}, {"product":{"id":6,"title":"Nine to Five (a.k.a. 9 to 5)","detail":"C  
ras mi pede, malesuada in, imperdiet et, commodo vulputate, justo. In blandit  
ultrices enim. Lorem ipsum dolor sit amet, consectetur adipiscing elit.\\n  
\\nProin interdum mauris non ligula pellentesque ultrices. Phasellus id sapie  
n in sapien iaculis congue. Vivamus metus arcu, adipiscing molestie, hendreri  
t at, vulputate vitae, nisl."},"price":27,"stock":20,"category":"Moderno"},"qu  
antity":4}]'
```


js/cart.js

En este archivo luego de parsear el item “cart”, mapeamos las cards de los autos y lo insertamos en el DOM de cart.html.

Carro de compras



Twice in a Lifetime

Precio
\$55

Cantidad
3

Total
\$165

Eliminar



Nine to Five (a.k.a. 9 to 5)

Precio
\$27

Cantidad
4

Total
\$108

Eliminar


```

const cardsSection = document.querySelector("#cart #cards")

function getCart(cards) {
  const list = cards.map(
    card => `<div class="card border shadow-none mb-2">
    <div class="card-body">
      <div class="d-flex align-items-start">
        <div class="me-4">
          
        </div>
        <div class="flex-grow-1 overflow-hidden">
          <h5 class="text-truncate font-size-18">${card.product.title}</h5>
          <div class="row">
            <div class="col-md-3">
              <div class="mt-3">
                <p class="text-muted mb-2">Precio</p>
                <h5 class="mb-0 mt-2">${card.product.price}</h5>
              </div>
            </div>
            <div class="col-md-4">
              <div class="mt-3">
                <p class="text-muted mb-2">Cantidad</p>
                <h5 class="mb-0 mt-2">${card.quantity}</h5>
              </div>
            </div>
            <div class="col-md-3">
              <div class="mt-3">
                <p class="text-muted mb-2">Total</p>
                <h5>${card.product.price * card.quantity}</h5>
              </div>
            </div>
            <div class="col-md-2">
              <div class="mt-3">
                <p class="text-muted mb-2" onclick="removeItem(${card.product.id})">Eliminar</p>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>`
  )

  cardsSection.innerHTML = list.join("")
}


getCart(JSON.parse(localStorage.getItem("cart")))

```

js/cart.js


Calcular el total.

Carro de compras



Twice in a Lifetime

Precio	Cantidad	Total	Eliminar
\$55	3	\$165	



Nine to Five (a.k.a. 9 to 5)

Precio	Cantidad	Total	Eliminar
\$27	4	\$108	


Orden de Compra

Total :

\$273

js/cart.js => total

Calcular el total.



```
function total(cards) {  
  let cartTotal = document.querySelector("#cart-total")  
  
  let total = cards.reduce(  
    (acumulado, actual) => acumulado + actual.product.price * actual.quantity,  
    0  
  )  
  cartTotal.innerText = "$" + total  
}  
  
total(JSON.parse(localStorage.getItem("cart")))
```

js/cart.js

Eliminar producto.



Twice in a Lifetime

Precio	Cantidad	Total	Eliminar
\$55	3	\$165	

js/cart.js => removeItem

- Parseamos el item cart.
- Filtramos para remover.
- Actualizamos el item cart.
- Volvemos a renderizar las cards y el total.
- Recalculamos el total del nav.

```
function removeItem(id) {  
  const cards = JSON.parse(localStorage.getItem("cart"))  
  
  const newCards = cards.filter(card => card.product.id !== id)  
  
  localStorage.setItem("cart", JSON.stringify(newCards))  
  
  getCart(newCards)  
  
  total(newCards)  
  
  let quantity = newCards.reduce((acumulado, actual) => acumulado + actual.quantity, 0)  
  localStorage.setItem("quantity", quantity)  
  const quantityTag = document.querySelector("#quantity")  
  quantityTag.innerText = quantity  
}
```

js/cart.js

Vaciar carrito. Es parecido a la función removeItem, pero mucho más simple.

Carro de compras



Tribe, The (Plemya)

Precio

Cantidad

Total

Eliminar

\$93

5

\$465



Nine to Five (a.k.a. 9 to 5)

Precio

Cantidad

Total

Eliminar

\$27

3

\$81

Orden de Compra

Total :

\$546

Seguir comprando

Vaciar carrito

Checkout



```
function clearCart() {  
  let quantityTag = document.querySelector("#quantity")  
  quantityTag.innerText = "0"  
  localStorage.setItem("cart", JSON.stringify([]))  
  getCart([])  
  total(0)  
}
```

js/cart.js

Carro de compras

Seguir comprando

Vaciar carrito

Checkout

Orden de Compra

Total : \$0







¡Manos a la obra!

SPRINT 3

Cart 2

Entregable

Hacer **sólo** lo visto en clase (Hacer la vista del carrito e iterar las funciones.).

Template de cart.html:

https://github.com/julioavantt/guayerd_login/blob/main/cart.html





retro

¿Cómo nos fué?

¿Qué cosas no quedaron claras y
necesitamos repasar la próxima?

