

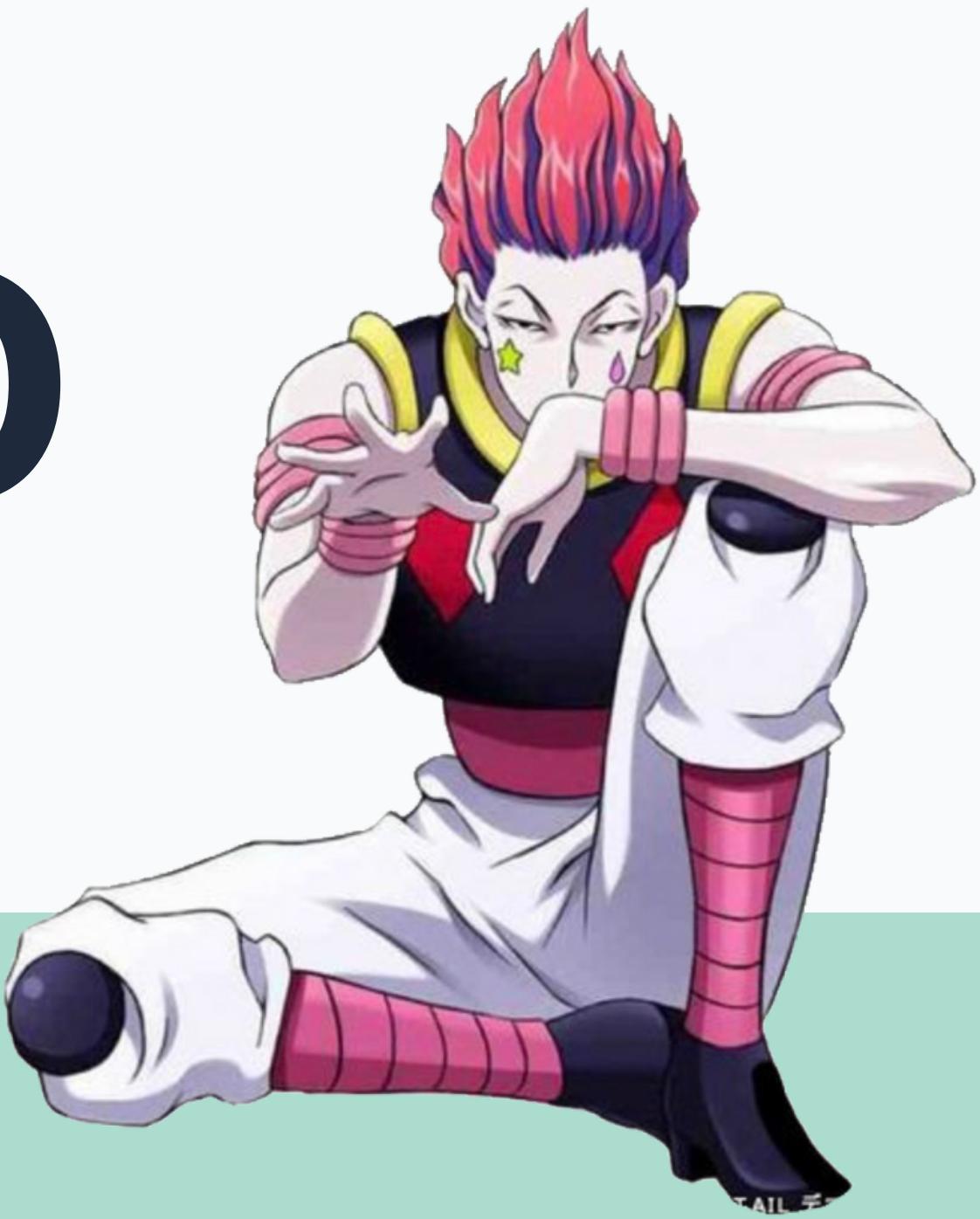
Web Components



Grupo 4



DEMO



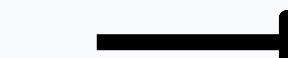
7 Componentes

4 Lit



- Rating
- Product
- Cart
- HeartIcon

3 Native



- TreeItem
- Modal
- ChevronIcon

Categorías

▼ Todos los productos

▼ Tecno

> Consolas y
Videojuegos

> Computadores

▼ Regalos

> Juegos

Peluches

▼ Ropa

Hombre

Mujer

Native

Total: \$



ShoppingCart

Show/Hide

Peluchi

Tablero

Apple V

Add new product

Name product

Price

Rating

Tecno

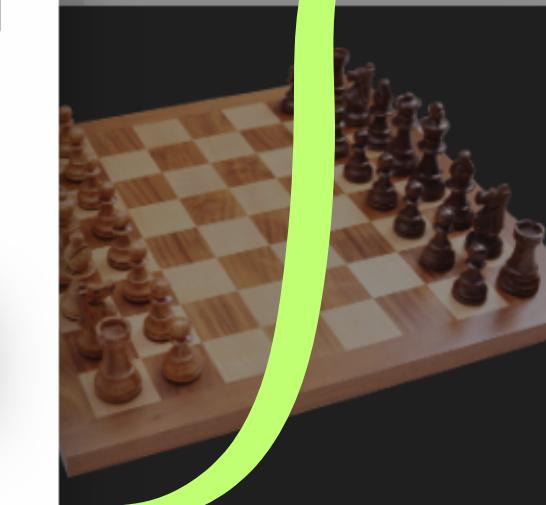
Discount

Seleccionar archivo Ningun... selec.

ADD

2
1
1

ADD NEW PRODUCT



Native

**Tablero
Ajedrez**

\$100.00



ADD TO CART

Apple Vision

Macbook Pro



Lit



ShoppingCart

Show/Hide

Peluche Miku	\$ 18.00	2
Tablero Ajedrez	\$ 100.00	1
Apple Vision Pro	\$ 3499.00	1

Total: \$3635.00

ADD NEW PRODUCT



Peluche Miku

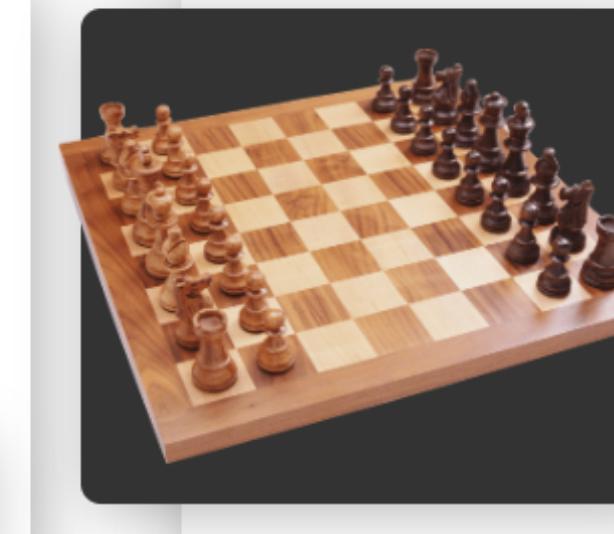
\$20.00

-%10

\$18.00



ADD TO CART



Tablero Ajedrez

\$100.00



ADD TO CART

Lit



[Lit] Product

Product → Rating → HeartIcon

[Lit] Product



```
export default class Product extends  
LitElement {  
  constructor() {  
    super();  
    this.addToCart = null;  
    this.categories = [];  
    this.name = "";  
    this.price = 0;  
    this.rating = 1;  
    this.discount = false;  
    this.img = "";  
  }  
  static get styles() {  
    return [style];  
  }  
  ...  
}  
const style = css`  
...  
`;
```



```
connectedCallback() {  
  super.connectedCallback();  
  this.name = this.getAttribute("name");  
  this.price = this.getAttribute("price");  
  this.rating = this.getAttribute("rating");  
  this.categories = this.getAttribute("categories").split(",");  
  this.discount = this.getAttribute("discount") || false;  
  this.img = this.getAttribute("img");  
}
```

[Lit] Product



```
render() {
    return html`


    </div>
    <div class="info-product">
        <span class="name">${this.name}</span>
        ${this.renderPrice()}
        <rating-component rating="${this.rating}"></rating-component>
        <button class="btn" @click="${this.handleAddToCart}">
            Add to cart
        </button>
    </div>
`;


```

[Lit] Product



```
async function addProductToStore(product) {
  const productComponent = document.createElement("product-component");
  if (typeof product.img === "object") {
    productComponent.setAttribute("img", imgUploaded);
  }
  if (typeof product.img === "string") {
    productComponent.setAttribute("img", product.img);
  }
  productComponent.setAttribute("name", product.name);
  productComponent.setAttribute("price", product.price);
  productComponent.setAttribute("rating", product.rating);
  if (product.discount) {
    productComponent.setAttribute("discount", product.discount);
  }
  productComponent.setAttribute("categories", product.categories);
  productComponent.addToCart = addToCart;
  document.querySelector(".products-container").prepend(productComponent);
}
```

[Native]

Tree-item



```
<tree-item>
    Todos los productos
    <tree-item>
        Tecno
        <tree-item>
            Consolas y Videojuegos
            <tree-item> Playstation </tree-item>
            <tree-item> Nintendo </tree-item>
            <tree-item> Otros </tree-item>
        </tree-item>
        <tree-item>
            Computadores
            <tree-item> Notebooks </tree-item>
            <tree-item> Ipads y tablets </tree-item>
        </tree-item>
    </tree-item>
    <tree-item>
        Regalos
        <tree-item>
            Juegos
            <tree-item> Juegos de Mesa </tree-item>
        </tree-item>
        <tree-item> Peluches </tree-item>
    </tree-item>
    <tree-item>
        Ropa
        <tree-item> Hombre </tree-item>
        <tree-item> Mujer </tree-item>
    </tree-item>
</tree-item>
```

[Native] Tree-item



```
template.innerHTML = `

<style>
  ...
</style>

<div class="container">
  <button id="main-button"></button>
  <slot></slot>
</div>
`;
```



```
export default class TreeItem extends HTMLElement {
  constructor() {
    super();
    this._shadowRoot = this.attachShadow({ mode: "open" });
    this._shadowRoot.appendChild(template.content.cloneNode(true));

    this.$children = this._shadowRoot.querySelectorAll("slot");
    this.$button = this._shadowRoot.querySelector("#main-button");
    this.$addButton = this._shadowRoot.querySelector("#add-child");

    const assignedNodes = this.$children.assignedNodes();
    this.text = this.formatText(assignedNodes[0].textContent);
    if (assignedNodes.length == 1) {
      this.$button.innerHTML =
        `
          ${this.text}
        
      `;
      assignedNodes[0].textContent = "";
    } else if (assignedNodes.length > 1) {
      this.$button.innerHTML =
        `</chevron-icon>
        <span>${this.text}</span>
      `;
      this.$icon = this._shadowRoot.querySelector("chevron-icon");
      assignedNodes[0].textContent = "";
    }
    this.$button.addEventListener("click", () => {
      this.showChildren();
      if (this.handleClick) {
        this.handleClick(this.text);
        this.$button.classList.add("selected");
        if (this.$icon) this.$icon.toggleColor();
      }
    });
  }
}
```

[Native] Tree-item



```
export default class TreeItem extends HTMLElement {
  constructor() {
    super();
    this._shadowRoot = this.attachShadow({ mode: "open" });
    this._shadowRoot.appendChild(template.content.cloneNode(true));

    this.$container = this._shadowRoot.querySelector(".container")
    this.$button = this._shadowRoot.querySelector("#main-button");
    this.$children = this._shadowRoot.querySelector("slot");
  ...
}
```

```
export default class TreeItem extends HTMLElement {
  constructor() {
    super();
    this._shadowRoot = this.attachShadow({ mode: "open" });
    this._shadowRoot.appendChild(template.content.cloneNode(true));

    this.$container = this._shadowRoot.querySelector(".container")
    this.$button = this._shadowRoot.querySelector("#main-button");
    this.$children = this._shadowRoot.querySelector("slot");
}
```

[Native] Tree-item



```
const assignedNodes = this.$children.assignedNodes();
this.text = this.formatText(assignedNodes[0].textContent);
if (assignedNodes.length == 1) {
  this.$button.innerHTML =
    <span class="left-margin">
      ${this.text}
    </span>
  ;
  assignedNodes[0].textContent = "";
} else if (assignedNodes.length > 1) {
  this.$button.innerHTML =
    <chevron-icon direction="0"></chevron-icon>
    <span>${this.text}</span>
  ;
  this.$icon = this._shadowRoot.querySelector("chevron-icon");
  assignedNodes[0].textContent = "";
}
```



```
const assignedNodes = this.$children.assignedNodes();
this.text = this.formatText(assignedNodes[0].textContent);
if (assignedNodes.length == 1) {
  this.$button.innerHTML =
    <span class="left-margin">
      ${this.text}
    </span>
  ;
  assignedNodes[0].textContent = "";
} else if (assignedNodes.length > 1) {
  this.$button.innerHTML =
    <chevron-icon direction="0"></chevron-icon>
    <span>${this.text}</span>
  ;
  this.$icon = this._shadowRoot.querySelector("chevron-icon");
  assignedNodes[0].textContent = "";
}
```

[Native] Tree-item



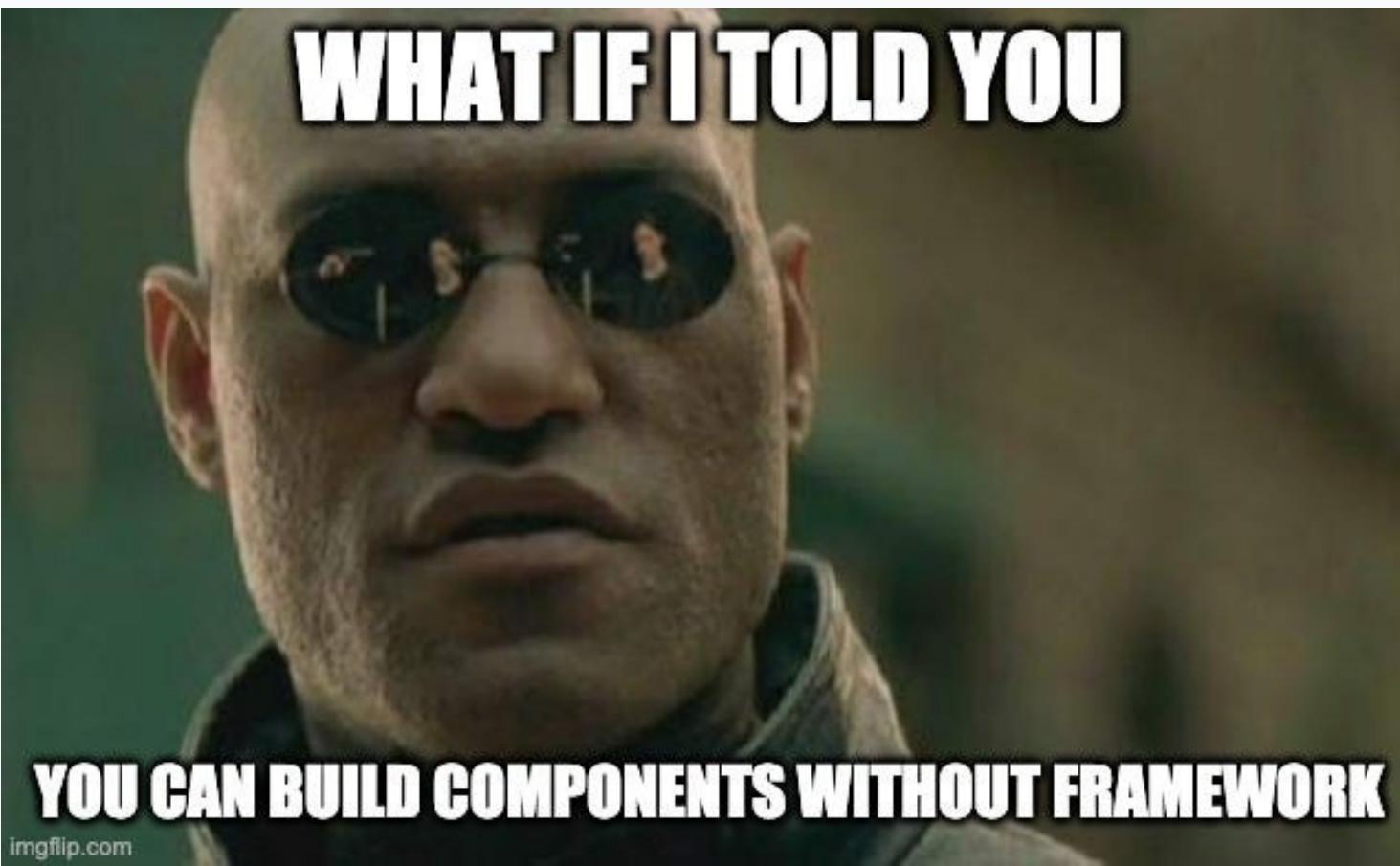
```
this.$button.addEventListener("click", () => {  
  this.showChildren();  
  if (this.handleClick) {  
    this.handleClick(this.text);  
    this.$button.classList.add("selected");  
    if (this.$icon) this.$icon.toggleColor();  
  }  
});  
...
```

```
this.$button.addEventListener("click", () => {  
  this.showChildren();  
  if (this.handleClick) {  
    this.handleClick(this.text);  
    this.$button.classList.add("selected");  
    if (this.$icon) this.$icon.toggleColor();  
  }  
});
```

Conclusiones

- Componentes nativos no son nada trivial
- Independencia de frameworks
- Encapsulación:
amiga o enemiga?





Web Components

Grupo 4

