

WEB COMPONENTS

GRUPO 7 - DAHL



Objetivo 1

- Construir un componente que sirva para publicitar un producto.
- Lograrlo con los elementos estándar de Web Components y también con LitElement

Web Components App

¡Mundo Computación!



HP LAPTOP 15-EF1018LA /
AMD RYZEN™ 5 / 8GB
RAM / 256GB SSD / AMD
RADEON GRAPHICS / 15.6

\$599.990

Normal: ~~\$549.990~~



NOTEBOOK HP 14-
CK2091LA / INTEL®
CORE™ I3 / 4GB RAM /
128GB SSD / INTEL UHD
GRAPHICS / 14"

\$599.990

Normal: ~~\$549.990~~



NOTEBOOK HP 14-
CK2091LA / INTEL®
CORE™ I3 / 4GB RAM /
128GB SSD / INTEL UHD
GRAPHICS / 14"

\$599.990

Normal: ~~\$549.990~~

Objetivo 2

- Construir un TO-DO list donde se puedan añadir y eliminar elementos de la lista.
- Implementar con los elementos estándar de Web Components y también con LitElement

To do list

Write new task

To do

- | | | |
|-------------------------------------|--------------------|---|
| <input type="radio"/> | Segunda | X |
| <input checked="" type="checkbox"/> | Tercera | X |
| <input checked="" type="checkbox"/> | tarea 7 | X |

DEMO

Sell-item: component

```
const template = document.createElement("template");
template.innerHTML = `<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.
<style>
h1{
  color:red;
}
.col{
  display: inline-block;
}
</style>
<div class="col">
<div class="card" style="width: 18rem;">
  <img class="card-img-top" alt="Card image cap">
  <div class="card-body">
    <h5 class="card-title"></h5>
    <p class="card-text"><slot name="new_price" /></p>
    <p class="card-text"><slot name="old_price" /></p>
    <i class="bi bi-star-fill"></i>
  </div>
</div>
</div>`;

You, seconds ago | 2 authors (You and others)
class SellItem extends HTMLElement {
  constructor() {
    super();
    this.attachShadow({ mode: "open" });
  }

  connectedCallback() {
    // Se invoca cuando se llame al componente
    this.shadowRoot.appendChild(template.content.cloneNode(true));
    this.shadowRoot.querySelector("h5").innerText = this.getAttribute("name");
    this.shadowRoot.querySelector("img").src = this.getAttribute("img");
  }
}

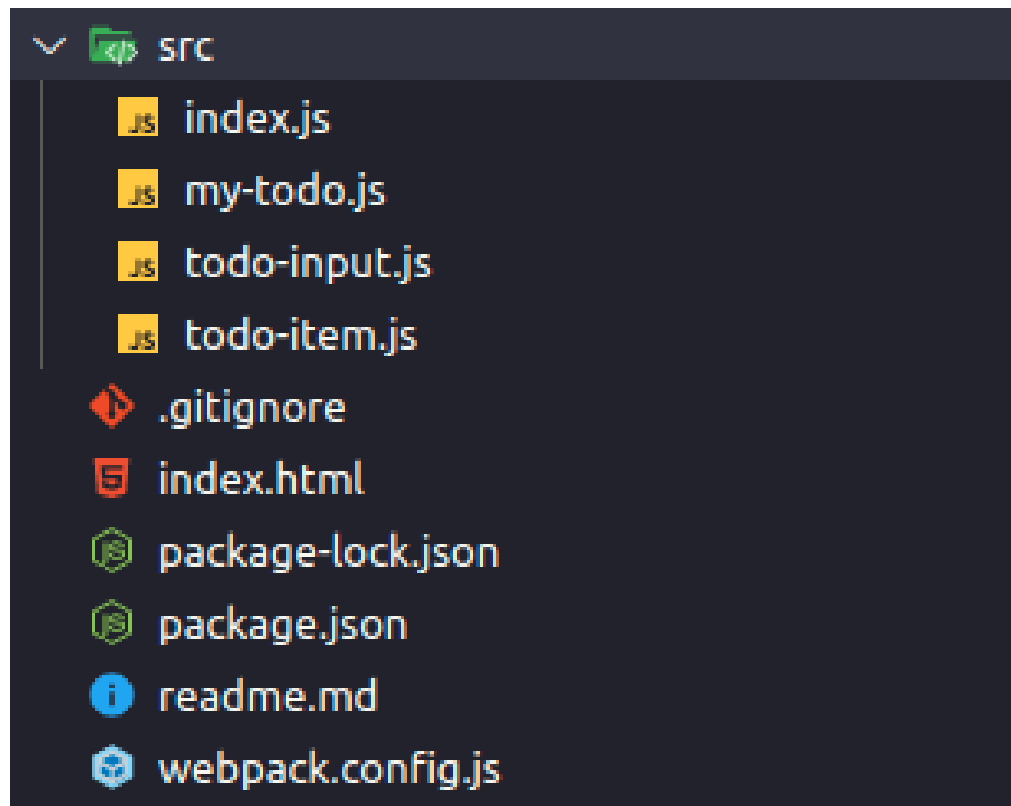
window.customElements.define("sell-item", SellItem);
```

Sell-item: utilización

```
<script src="./app.js"></script>
```

```
<div class="card-group">
  <sell-item
    name="HP LAPTOP 15-EF1018LA / AMD RYZEN™ 5 / 8GB RAM / 256GB SSD / AMD RADEON GRAPHICS / 15.6"
    img="https://home.ripley.cl/store/Attachment/WOP/D113/2000383107424/2000383107424-1.jpg"
  >
    <div slot="new_price">$599.990</div>
    <div slot="old_price">Normal: <del>$549.990</del></div>
  </sell-item>
  <sell-item
    name="NOTEBOOK HP 14-CK2091LA / INTEL® CORE™ I3 / 4GB RAM / 128GB SSD / INTEL UHD GRAPHICS / 14""
    img="https://home.ripley.cl/store/Attachment/WOP/D113/2000381074056/2000381074056-1.jpg"
  >
    <div slot="new_price">$469.990</div>
    <div slot="old_price">Normal: <del>$399.990</del></div>
  </sell-item>
</div>
```

Todo-list: component



```
import Todo from './my-todo';
import TodoInput from './todo-input';
import TodoItem from './todo-item';

window.customElements.define('todo-list', Todo);
window.customElements.define('todo-input', TodoInput);
window.customElements.define('todo-item', TodoItem);
```

```
<body>
  <todo-list></todo-list>
  <script src='./dist/main.js'></script>
</body>
```

Todo-list: MyTodo

```
const templateTodo = document.createElement('template');
> templateTodo.innerHTML = `...
`;

You, seconds ago | 1 author (You)
export default class MyTodo extends HTMLElement {
  constructor() {
    super();
    this._root = this.attachShadow({ 'mode': 'open' });
    // initial state
    this._list = [
      { text: 'Primera', checked: false },
      { text: 'Segunda', checked: false },
      { text: 'Tercera', checked: true },
    ];
  }

  connectedCallback() { ... }

  addItem(e) { ... }

  removeItem(e) { ... }

  toggleItem(e) { ... }

  disconnectedCallback() { ... }

  _render() { ... }
}
```

```
<section>
  <todo-input></todo-input>
  <h3>To do</h3>
  <ul id="list-container"></ul>
</section>
```


Todo-list: input

```
const templateTodoInput = document.createElement('template');
> templateTodoInput.innerHTML = `
  `;

You, a day ago | 1 author (You)
export default class TodoInput extends HTMLElement {
  constructor() {
    super();
    this._root = this.attachShadow({ mode: 'open' });
  }

  connectedCallback() {
    this._root.appendChild(templateTodoInput.content.cloneNode(true));
    this.$form = this._root.querySelector('form');
    this.$input = this._root.querySelector('input');
    this.$form.addEventListener("submit", (e) => {
      e.preventDefault();
      if (!this.$input.value) return;
      this.dispatchEvent(new CustomEvent('onSubmit', { detail: this.$input.value }));
      this.$input.value = '';
    });
  }

  disconnectedCallback() {
  }
}
```

```
<form id="new-todo-form">
  <input id="new-todo" type="text" placeholder="Write new task" />
</form>
```

Todo-list: item

```
export default class TodoItem extends HTMLElement {  
  constructor() {  
    super();  
    this._root = this.attachShadow({ 'mode': 'open' });  
    this._checked = this.checked;  
    this._index = this.index;  
    this._text = '';  
  }  
}
```

```
<li class="item">  
  <input type="checkbox">  
  <label></label>  
  <button class="destroy">x</button>  
</li>
```

Lit-Element



Lit

```
<div class="container">
  <h1>Lit-Element Components App</h1>
  <h5>¡Mundo Computación!</h5>
  <div class="row">
    <div class="card-group">
      <sell-item
        name="HP LAPTOP 15-EF1018LA / AMD RYZEN™ 5 / 8GB RAM / 256GB SSD / AMD RADEON GRAPHICS / 15.6"
        img= "https://home.ripley.cl/store/Attachment/WOP/D113/2000383107424/2000383107424-1.jpg"
        new_price="$549.990"
        old_price="$599.990"
      >
      </sell-item>
    </div>
  </div>
  <div class="row">
    <div class="card-group">
      <todo-list></todo-list>
    </div>
  </div>
</div>
```

sell-item

```
import { LitElement, html } from 'lit-element';

export default class SellItem extends LitElement {

  static get properties() {
    return {
      name: {
        type: String,
        attrName: 'name'
      },
      img: {
        type: String,
        attrName: 'img'
      },
      new_price: {
        type: String,
        attrName: 'new_price'
      },
      old_price: {
        type: String,
        attrName: 'old_price'
      }
    }
  }
}
```

sell-item

```
render() {  
  return html`  
    <div class="col">  
      <div class="card" style="width: 18rem;">  
          
        <div class="card-body">  
          <h5 class="card-title">${this.name}</h5>  
          <p class="card-text">${this.new_price}</p>  
          <p class="card-text">Normal: <del>${this.old_price}</del></p>  
          <i class="bi bi-star-fill"></i>  
        </div>  
      </div>  
    </div>  
  `;  
}
```

todo-list

```
import { LitElement, css, html } from 'lit-element';

export default class TodoList extends LitElement {
  constructor() {
    super();
    this.list = [
      { text: 'Primera', checked: false },
      { text: 'Segunda', checked: false },
      { text: 'Tercera', checked: true }
    ];
    this.addItem = this.addItem.bind(this);
    this.removeItem = this.removeItem.bind(this);
    this.toggleItem = this.toggleItem.bind(this);
  }

  > addItem(e, text) { ...
  }

  > removeItem(e, index) { ...
  }

  > toggleItem(e, index) { ...
  }
```

todo-list

```
static get styles() {
  return css`
    section {
      background: #f5f5f5;
      margin: 30px 0 40px 0;
      position: relative;
      box-shadow: 0 2px 4px 0 rgba(0, 0, 0, 0.2), 0 25px 50px 0
    }
    #list-container {
      margin: 0 0 0 0;
      padding-top: 0;
      list-style: none;
      border-top: 1px solid #e6e6e6;
    }
  `;
}
```

```
render() {
  return html`
    <h1>Todo - Lit Element</h1>
    <section>
      <todo-input @submit="${(e) => this.addItem(e, e.detail)}">
      </todo-input>
      ${this.list.map((item, index) => html`
        <todo-item
          .text=${item.text}
          .checked=${item.checked}
          .index=${index}
          @remove="${(e) => this.removeItem(e, e.detail)}"
          @toggle="${(e) => this.toggleItem(e, e.detail)}"
        ></todo-item>`)}
      <ul id="list-container"></ul>
    </section>`;
}
```

Conclusiones

- Web components de forma nativa sintaxis más tediosa.
- Lit Element es autocontenido y con sintaxis más simple.
- Permite añadir componentes web sin la necesidad de desarrollar directamente en una aplicación.
- Reutilización de código entre aplicaciones.
- Desarrollo más rápido.

**MUCHAS
GRACIAS**

GRUPO 7 - DAHL

