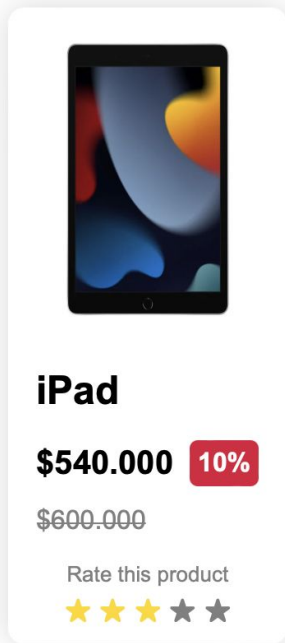
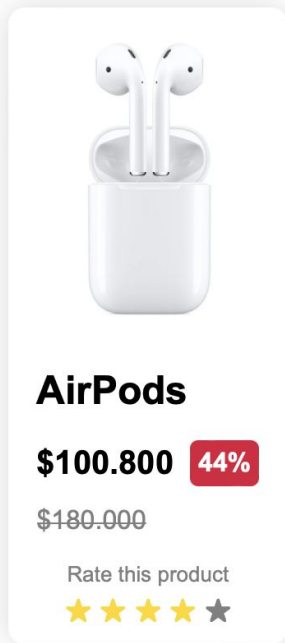


Web Components

Grupo 12

Componente 1: <product-element/>



```
<product-element name="AirPods" price=180000 discount=44
  imageSrc="https://.../604974-0000-001.jpg"
  imageHoverSrc="https://.../604974-0000-002.jpg"
  starCount="4"></product-element>
<product-element name="iPad" price=600000 discount=10
  imageSrc="https://.../275238-0000-002.jpg"
  imageHoverSrc="https://.../275238-0000-001.jpg"
  starCount="3"></product-element>
```

product-element.js

```
export class ProductElement extends LitElement {
  static styles = css`
    ...
  `;

  static properties = {
    name: { type: String },
    price: { type: Number },
    starCount: { type: Number },
    imageSrc: { type: String },
    discount: { type: Number },
  };

  constructor() {
    super();
    this.name = 'iPhone 14';
    this.starCount = 3;
    this.price = 974990;
    this.imageSrc = "https://.../iphone_14_morad_ok_1.png";
  }
}
```

product-element.js

```
/**
 * Incrementa el conteo de estrellas en uno.
 */
__increment() {
  this.starCount += 1;
}

/**
 * Marca las estrellas según la calificación.
 * @param {number} idx - Índice de la estrella a marcar.
 * @returns {string} Clase CSS para marcar la estrella.
 */
marked(idx) {
  if (idx <= this.starCount) {
    return "star-checked";
  } else {
    return ""
  }
}

/**
 * Actualiza la calificación del producto.
 * @param {Event} e - Evento de click.
 */
__rate(e) {
  const newRate = parseInt(e.target.id.slice(-1));
  console.log(newRate);
  this.starCount = newRate;
}
```

```
/**
 * Obtiene el precio original del producto.
 * @returns {number} Precio original.
 */
getOriginalPrice() {
  if (this.discount) {
    return this.price;
  } else {
    return "";
  }
}

/**
 * Obtiene el precio del producto con el descuento aplicado.
 * @returns {number} Precio con descuento.
 */
getPrice() {
  if (this.discount) {
    return this.price - (this.price * this.discount / 100);
  } else {
    return this.price;
  }
}
```


Componente 2: <tree-item/>



```
<tree-item>
  Tecnología
  <tree-item>
    Celulares
    <tree-item>
      Smartphones
      <tree-item>iPhone</tree-item>
      <tree-item>Samsung</tree-item>
      <tree-item>Xiaomi</tree-item>
      <tree-item>
        Accesorios
        <tree-item>
          Audífonos
          <tree-item>Inalámbricos</tree-item>
          <tree-item>Cableados</tree-item>
        </tree-item>
        <tree-item>Cargadores</tree-item>
        <tree-item>Adaptadores</tree-item>
        <tree-item>Baterías</tree-item>
        <tree-item>Carcasas</tree-item>
      </tree-item>
    </tree-item>
    <tree-item>Básicos</tree-item>
  </tree-item>
  <tree-item>
    Computadores
```

tree-item.js

```
const template = document.createElement('template');

template.innerHTML = `
  <style>
    ...
  </style>
  <div class="toggle"></div>
  <div class="tree-item">
    <slot></slot>
  </div>
`;

export class TreeItem extends HTMLElement {
  constructor() {
    super();
    this.expanded = false;
    this.attachShadow({ mode: 'open' });
    this.shadowRoot.appendChild(template.content.cloneNode(true));
    this.$toggle = this.shadowRoot.querySelector('.toggle');
    this.$children = this.shadowRoot.querySelector('.tree-item');
  }

  connectedCallback() {
    setTimeout(() => {
      this.label = this.getLabel();
      this.$toggle.addEventListener('click', () => this.toggle());
      this.render();
    });
  }
}
```

```
getLabel() {
  let childNodes = Array.from(this.childNodes);
  let node = childNodes.find(node => node.nodeType === Node.TEXT_NODE &&
    node.nodeValue !== '');
  let label = node.nodeValue.trim();
  this.removeChild(node);
  return label;
}

toggle() {
  this.expanded = !this.expanded;
  this.render();
}

render() {
  this.$toggle.textContent = this.getIcon() + this.label;
  this.$children.style.display = this.expanded ? 'block' : 'none';
}

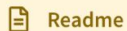
getIcon() {
  if (this.hasChildren()) {
    if (this.expanded) {
      return '▼';
    } else {
      return '▶';
    }
  } else {
    return ' ';
  }
}

hasChildren() {
  let childNodes = Array.from(this.childNodes);
  return childNodes.some(node => node.nodeType !== Node.TEXT_NODE);
}
```

Componentes en npm

product-element

0.0.17 • Public • Published 7 days ago

[Readme](#)[Code](#)[Beta](#)[1 Dependency](#)[0 Dependents](#)[18 Versions](#)[Settings](#)

<product-element>

This webcomponent follows the **open-wc** recommendation.

Installation

```
npm i product-element
```

Usage

```
<script type="module">
  import 'product-element/product-element.js';
</script>

<product-element></product-element>
```

Install

```
> npm i product-element
```

↓ Weekly Downloads

208

Version

License

0.0.17

MIT

Unpacked Size

Total Files

15.1 kB

11

Last publish

7 days ago

Collaborators

Publicación de componentes

npm publish

```
{  
  "name": "product-element",  
  "version": "0.0.2",  
  "type": "module",  
  "main": "index.js",  
  "module": "index.js"  
}
```

package.json del componente

```
<!doctype html>  
...  
<script type="module" src="https://cdn.skypack.dev/product-element"></script>  
...  
<body>  
  <product-element  
    name="AirPods" price=180000 discount=44  
    imageSrc="https://www.paris.cl/..."  
    starCount="4">  
  </product-element>  
</body>  
</html>
```

Otros proyectos

Documentación de componentes

<product-element>

Docs Demo product-element

Componente de producto con calificación y precio.

PROPERTIES

Name	Attribute	Type
name	name	string = 'iPhone 14'

METHODS

Description

Nombre del producto.

Name	Attribute	Type
starCount	starCount	number = 3

Description

Número de estrellas para la calificación del producto.

Name	Attribute	Type
price	price	number = 974990

Description

Precio del producto.

Name	Attribute	Type
imageSrc	imageSrc	string =
"https://tiendaempresas.entel.cl/media/catalog/product/cache/e83b319fe15d087a014"		

Description

URL de la imagen del producto.


Name	Attribute	Type
discount	discount	number

Description

Descuento en el producto.

<product-element>

Docs Demo product-element



iPhone 14

\$974.990

Rate this product

★★★★☆

SOURCE

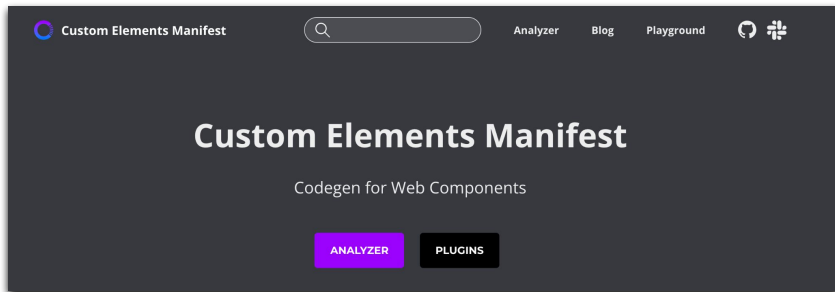
<product-element></product-element>

COPY

KNOBS

Documentación de componentes

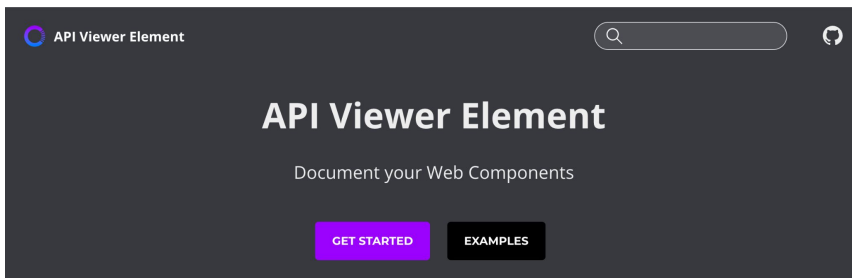
custom-elements-manifest analyze



```
...
/**
 *
 * Componente de producto con calificación y precio.
 *
 * @element product-element
 *
 * @property {string} name - Nombre del producto.
 * @property {number} price - Precio del producto.
 * @property {number} starCount - Número de estrellas para la calificación del producto.
 * @property {string} imageSrc - URL de la imagen del producto.
 * @property {number} discount - Descuento en el producto.
 *
 */
export class ProductElement extends LitElement {
  ...
}
```

Sintaxis JsDoc

Documentación de componentes



```
<!DOCTYPE html>
<html>
  <head>
    <script type="module">
      import 'api-viewer-element';
    </script>
    <script type="module">
      import '../product-element.js';
    </script>
  </head>
  <body>
    <api-viewer src="../custom-elements.json"></api-viewer>
  </body>
</html>
```

Web Components

Grupo 12