

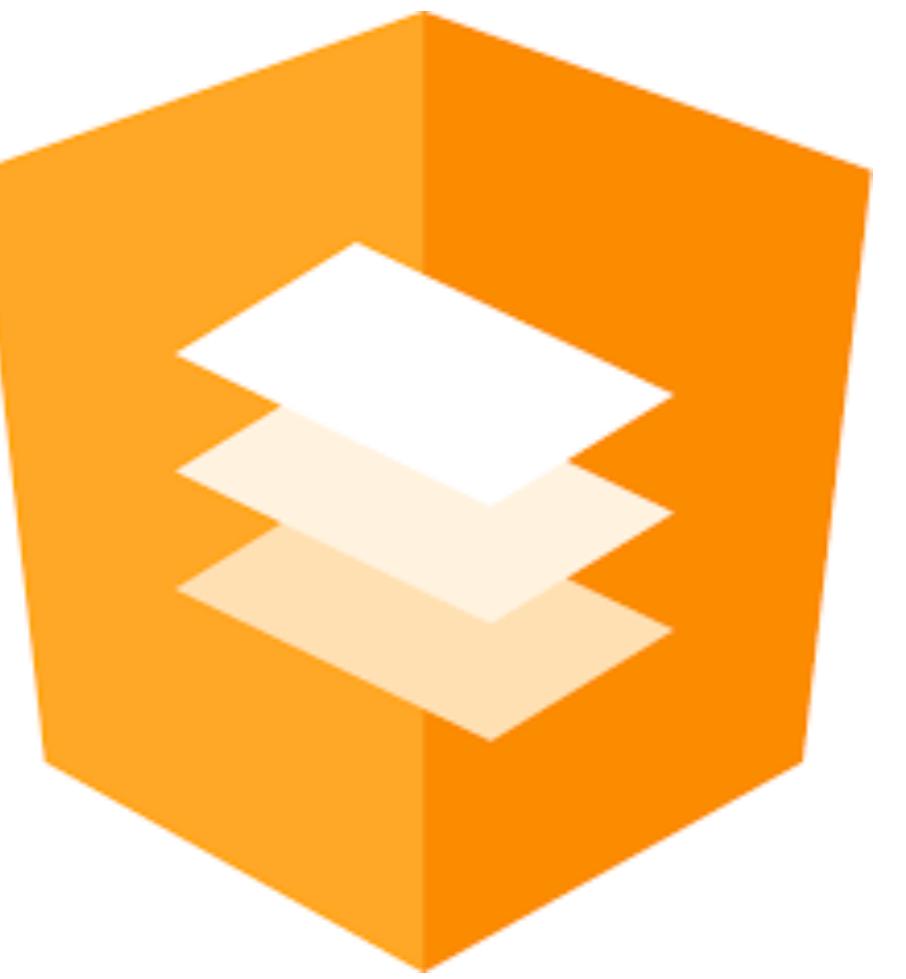
WEB COMPONENTS NATIVOS

“

Erase una vez...



<> RAISED



“

Y si pudieramos usar el mismo componente en
todas las tecnologías ? ...

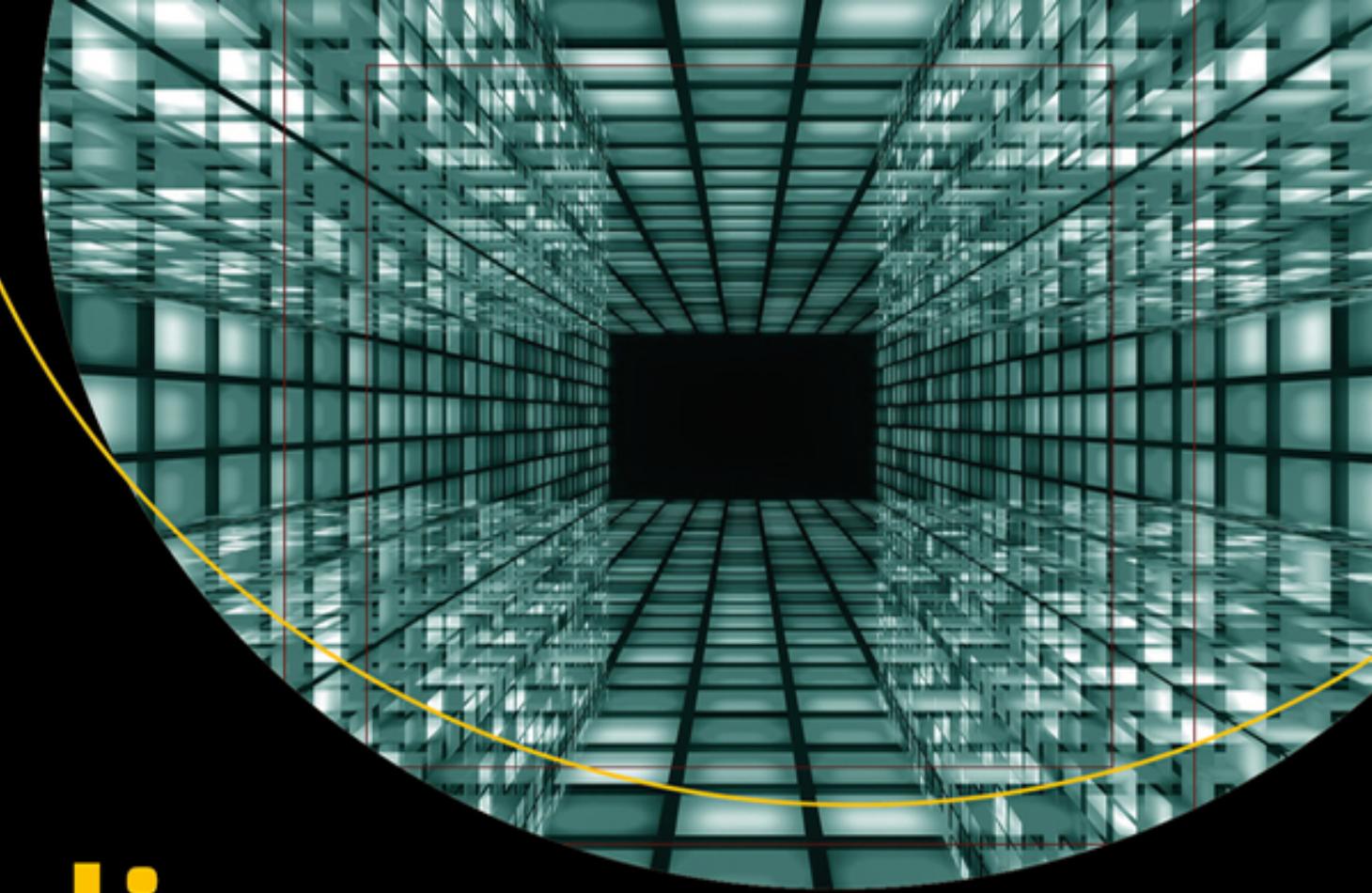
CARLOS ROJAS

@CARLOSROJAS_O



2020

<https://www.apress.com/la/book/9781484253335>



Building Progressive Web Applications with Vue.js

Reliable, Fast, and Engaging Apps
with Vue.js

—
Carlos Rojas

Apress®

2021

<https://www.apress.com/la/book/9781484259047>

Building Native Web Components

Front-End Development with Polymer
and Vue.js

—
Carlos Rojas

Apress®



PROBLEMA

FRONT-END



BACK-END



Willie
Zulu

LEGACY



ANGULARJS

FRAMEWORKS



TOOLING



webpack



GRUNT



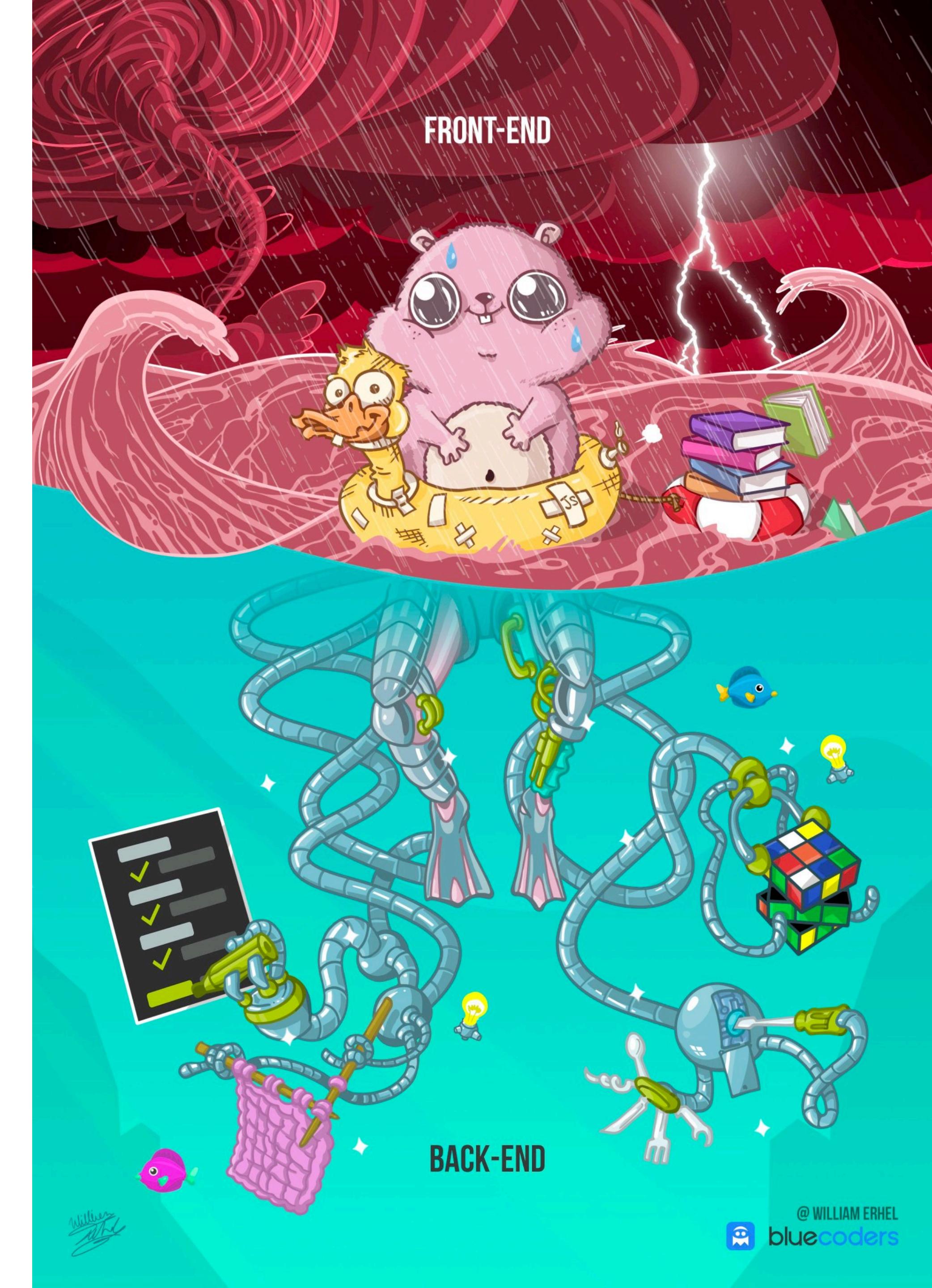
rollup.js



DEVS



JavaScript framework lifecycle



BACK-END



SOLUCIÓN

Componentes Nativos de los Navegadores

<IAMCARLOS></IAMCARLOS>

```
<HTML>
<HEAD></HEAD>
<BODY>
<IAMCARLOS></IAMCARLOS>
</BODY>
<SCRIPT TYPE="MODULE" SRC="NODE_MODULES/@CARLOS/CARLOS-EMOJI/CARLOS-
EMOJI.JS"
></SCRIPT>
</HTML>
```



CUSTOM ELEMENTS



```
class ErrorComponent extends HTMLElement {  
  constructor(...)  
}  
customElements.define('error-component', ErrorComponent);
```

SHADOW DOM



```
this.root = this.attachShadow( {mode: 'open'});
```

HTML TEMPLATES



```
<template class="none-type">
  <style>
    .none {
      background-color: gray;
      padding: 15px;
      color: black;
    }
  </style>
  <div class="none">
    <slot>Error component<slot>
  </div>
</template>
```

ES MODULES



```
<!-- HTML -->

<script type="module" src="./math-constants.js"></script>

/* JS */

import {pi, euler} from "./math-constants.js";
```

MAJOR BROWSERS

Chrome Firefox Safari Edge Opera



Custom Elements



HTML Templates

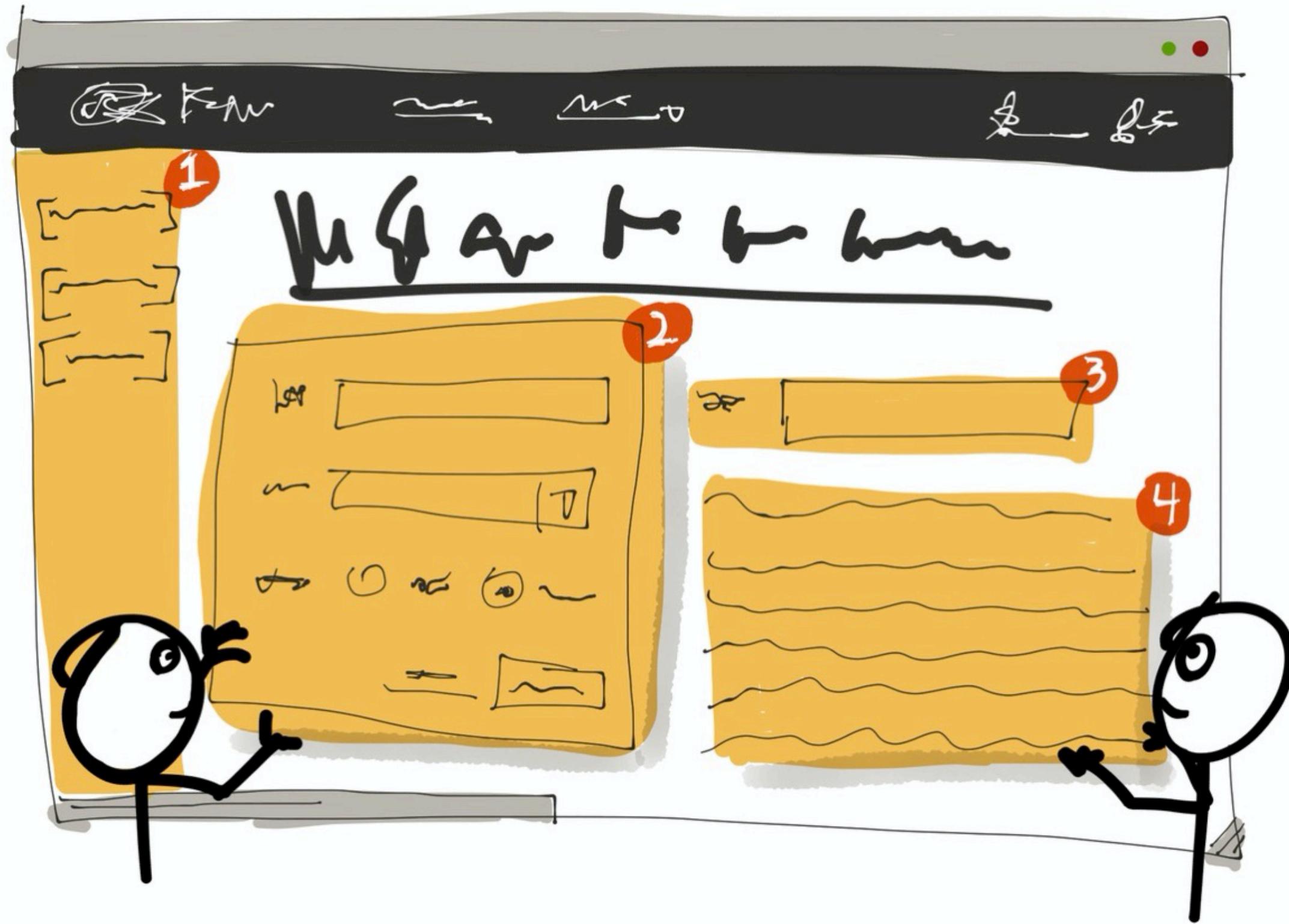


Shadow DOM



ES Modules





CDD, COMPONENT-DRIVEN DEVELOPMENT

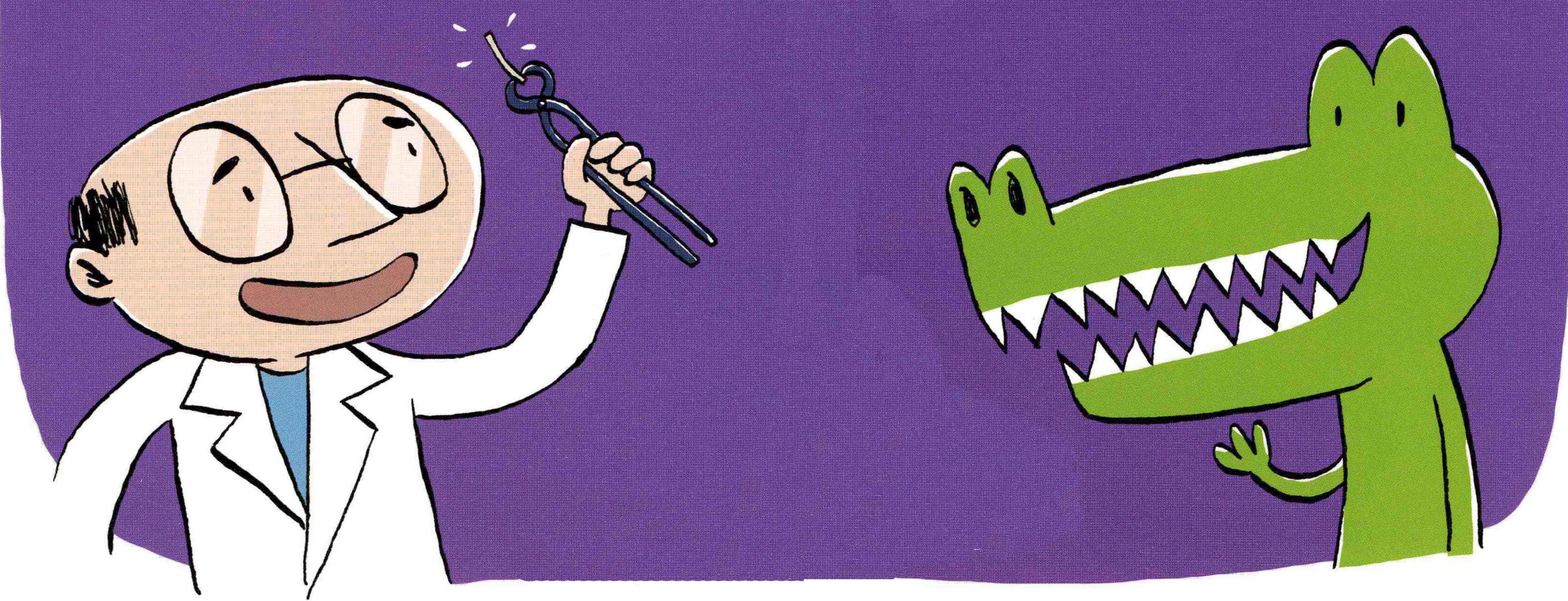
BENEFICIOS.

- Enfoque en el desarrollo.
- Aumenta el cubrimiento en el UI.
- Enfocado en retroalimentación.
- Construye una librería de componentes.
- Paraleliza desarrollo.
- Prueba visualmente.

PRINCIPIO FIRST

FIRST

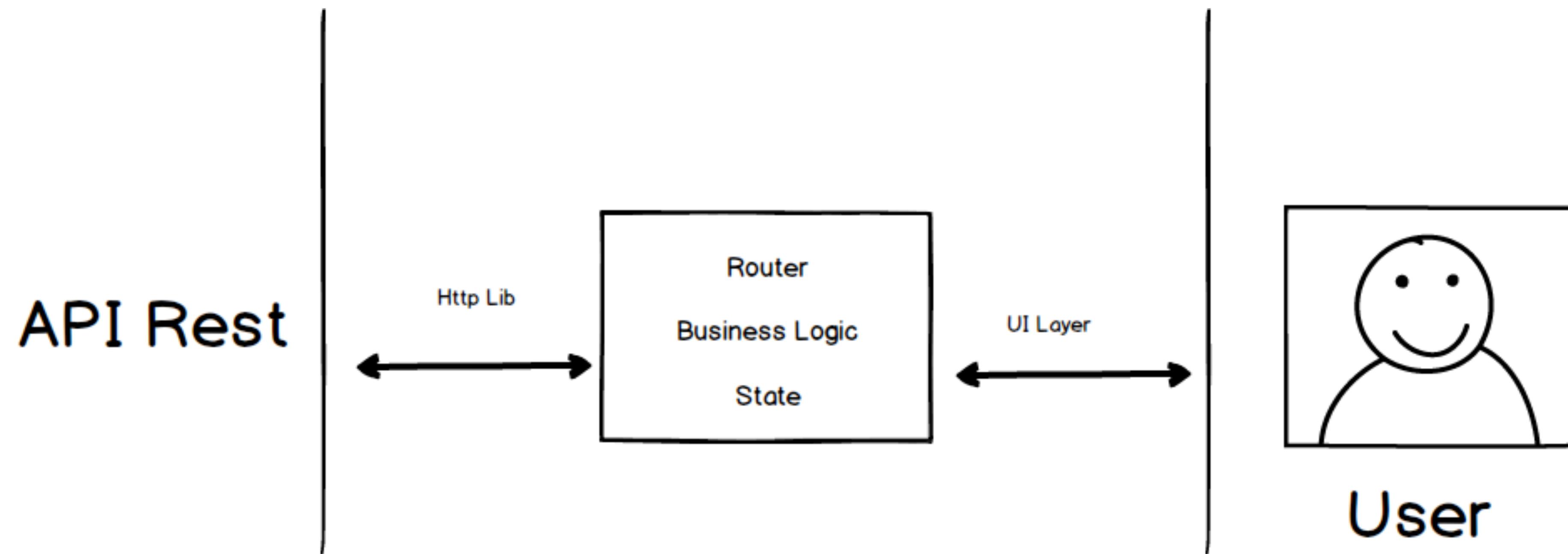
- Keep it (F)ocused.
- Keep it (I)ndependent.
- Keep it (R)eusable.
- Keep it (S)mall.
- Keep it (T)estable.



DEMO



INCEPTION



UI LAYER

WEB COMPONENTS

ROUTER

SIMROU

BUSINESS LOGIC

CONTROLADORES

STATE

REDUX

FLUX

HTTP LIB

AXIOS

FETCH



ESTRATEGIA DE ADOPCIÓN

FUNCIONALIDAD SIMPLE

EXTENSIBLE

UX/UI APROBADO

EQUIPO DEDICADO

VERSION

v1.0.1

UNIT TESTED

DOCUMENTADO

Usage

[ANGULAR](#)[JAVASCRIPT](#)[REACT](#)[VUE](#)[TYPESCRIPT](#)

```
import { Component } from '@angular/core';
import { AlertController } from '@ionic/angular';

@Component({
  selector: 'alert-example',
  templateUrl: 'alert-example.html',
  styleUrls: ['./alert-example.css'],
})
export class AlertExample {

  constructor(public alertController: AlertController) {}

  async presentAlert() {
    const alert = await this.alertController.create({
      header: 'Alert',
      subHeader: 'Subtitle',
      message: 'This is an alert message.',
      buttons: ['OK']
    });

    await alert.present();
  }

  async presentAlertMultipleButtons() {
    const alert = await this.alertController.create({
      header: 'Alert',
      subHeader: 'Subtitle',
```

DEMOS

ion-alert

CONTENTS

[Usage](#)

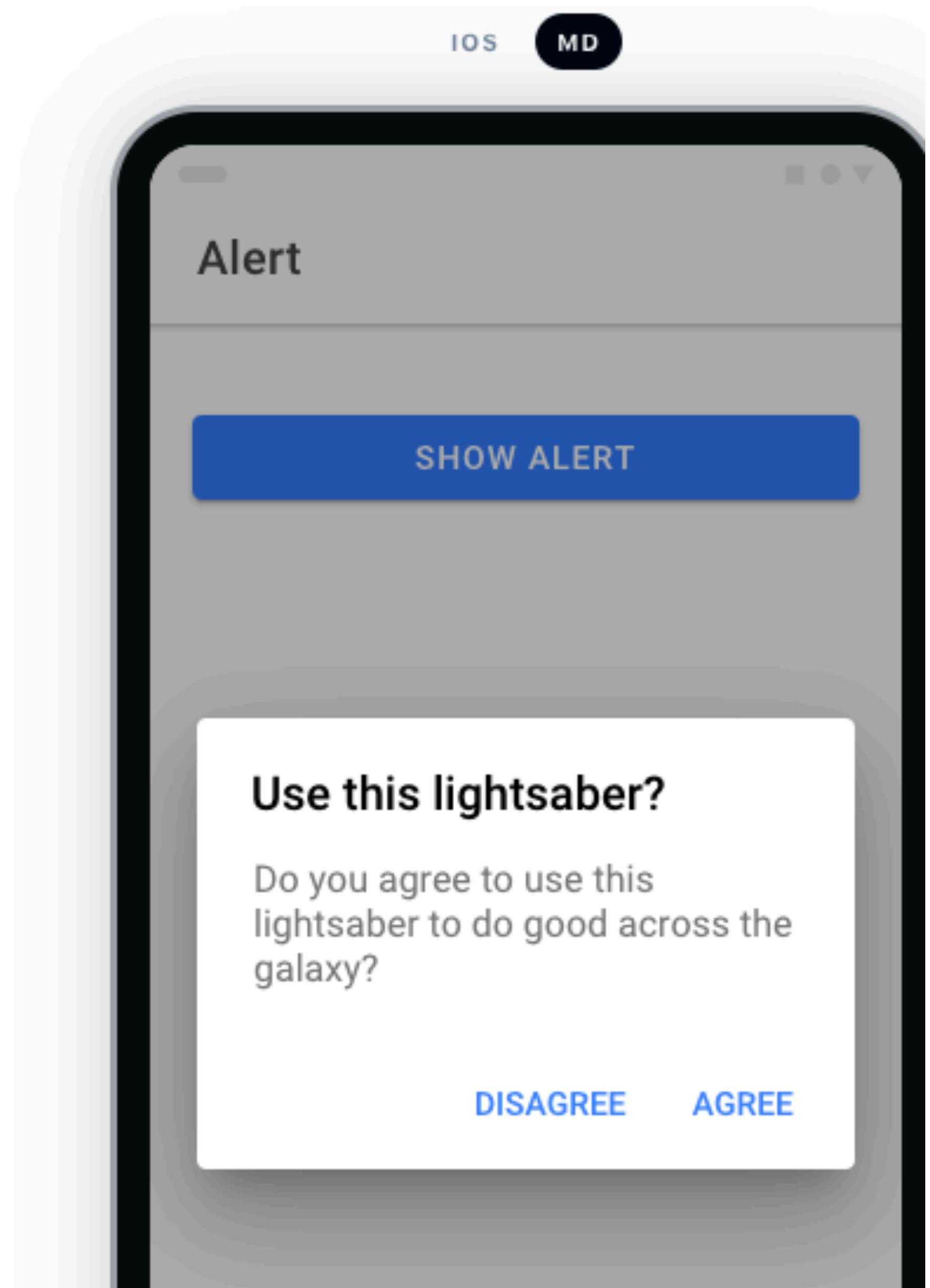
[Properties](#)

[Events](#)

[Methods](#)

[CSS Custom Properties](#)

An Alert is a dialog that presents users with information or collects information from the user using inputs. An alert appears on top of the app's content, and must be manually dismissed by the user before they can resume interaction with the app. It can also optionally have a `header` , `subHeader` and `message` .



CATALOGADO

- A Angular**
- A Angular**
- JS JavaScript**
- React**
- V Vue**

Controller

ALERT

ion-alert

ion-alert-controller

BADGE

ion-badge

BUTTON

ion-button

ion-ripple-effect

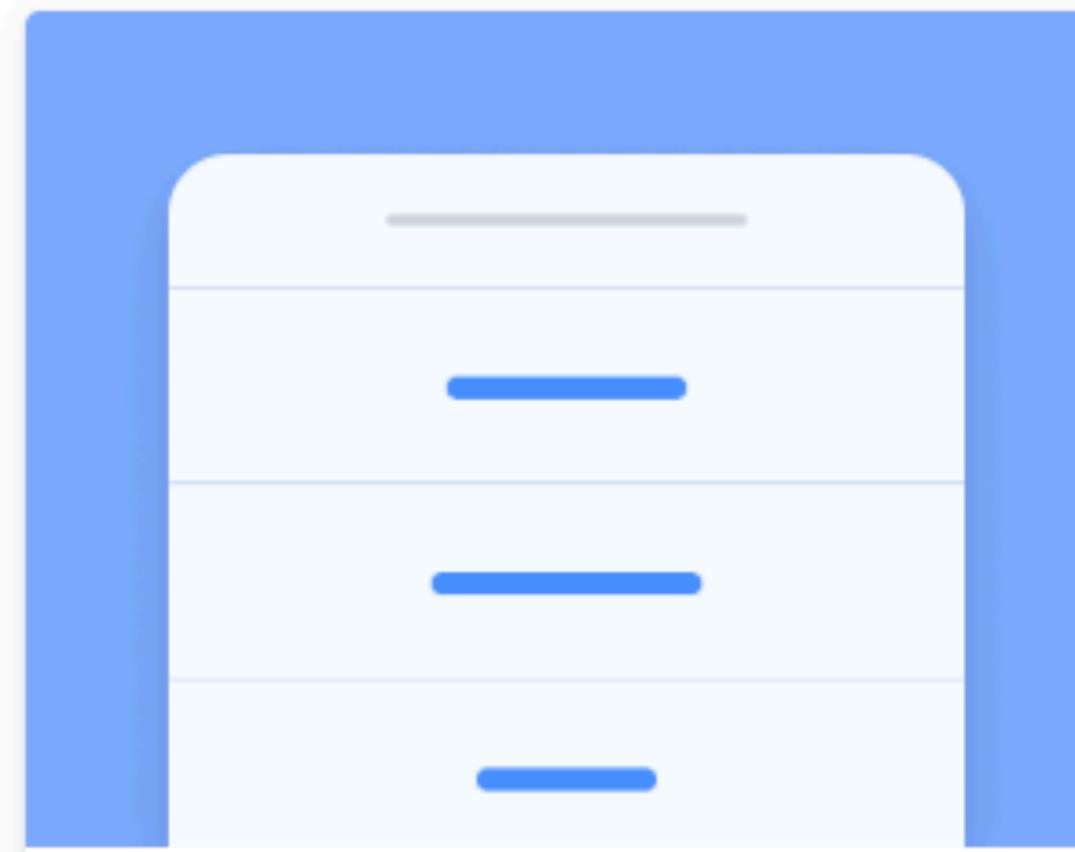
CARD

ion-card

ion-card-content

ion-card-header

...



Action Sheet

Action Sheets display a set of options with the ability to confirm or cancel an action.



Alert

Alerts are a great way to offer the user the ability to choose a specific action or list of actions.



Button

Buttons let your users take action. They're an essential way to interact with and navigate through an app.



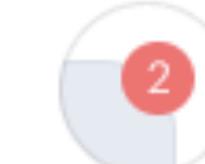
Checkbox

Checkboxes can be used to let the user know they need to make a binary decision.



Chip

Chips are a compact way to display data or actions.



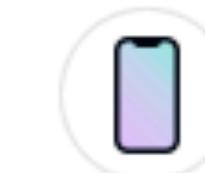
Badge

Badges are a small component that typically communicate a numerical value to the user.



Card

Cards are a great way to display an important piece of content, and can contain images, buttons, text, and more.



Content

Content is the quintessential way to interact with and navigate through an app.

FRAMEWORKS Y WEB COMPONENTS

Spoiler Alert! React no se ve muy bien acá

[**HTTPS://CUSTOM-ELEMENTS-EVERYWHERE.COM/**](https://custom-elements-everywhere.com/)

LIBRARY	SCORE	BASIC TESTS	ADVANCED TESTS
Angular 8.0.0	100%	16/16	14/14

Handling data

Angular's default binding syntax will always set properties on an element. This works well for rich data, like objects and arrays, and also works well for primitive values so long as the Custom Element author has mapped any exposed attributes to corresponding properties.

Angular also provides binding syntax specifically for setting an attribute, if a developer would prefer to communicate with an element that way.

Handling events

Angular components can listen to native DOM events dispatched from Custom Elements. It supports all styles of events (lowercase, camelCase, kebab-case, etc).

[View the tests](#)

Related Issues



Yay! No open issues!

LIBRARY	SCORE	BASIC TESTS	ADVANCED TESTS
React 16.8.6	71%	16/16	0/14

Handling data

React passes all data to Custom Elements in the form of HTML attributes. For primitive data this is fine, but the system breaks down when passing rich data, like objects or arrays. In these instances you end up with stringified values like `some-attr="[object Object]"` which can't actually be used.

Handling events

Because React implements its own synthetic event system, it cannot listen for DOM events coming from Custom Elements without the use of a workaround. Developers will need to reference their Custom Elements using a `ref` and manually attach event listeners with `addEventListener`. This makes working with Custom Elements cumbersome.

[View the tests](#)

Related Issues

RFC: Plan for custom element attributes/properties in React 17

#11347 opened Oct 23 by robododson

Bypass synthetic event system for Web Component events

#7901 opened Oct 6, 2016 by staltz

Passing props to custom elements as properties instead of attributes

#8755 opened Jan 11 by joeldenning

Attributes and properties for Custom Components

#7249 opened Jul 12, 2016 by edoardocavazza

Boolean attributes on Web Components

#9230 opened Mar 21 by nickdima

Event Handler on React Component not invoked when React Component is rendered inside a Web Component

#9242 opened on Mar 23 by nilshartmann

LIBRARY	SCORE	BASIC TESTS	ADVANCED TESTS
Vue 2.6.10	100%	16/16	14/14

Handling data

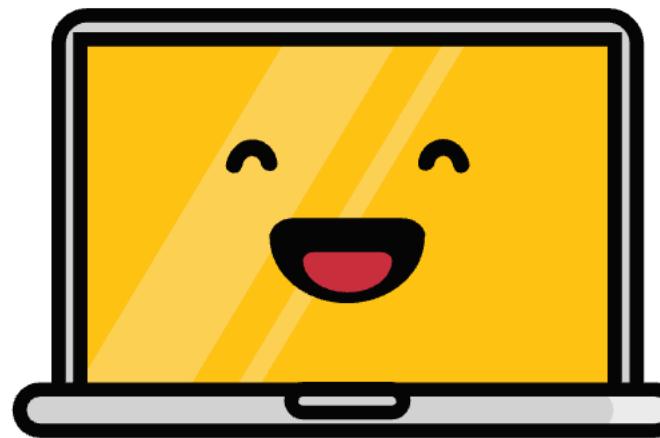
By default, Vue passes all data to Custom Elements as attributes. However, Vue also provides syntax to instruct its bindings to use properties instead. To bind to a Custom Element property use `:foo.prop="bar"`.

Handling events

Vue can listen to native DOM events dispatched from Custom Elements. It supports all styles of events (lowercase, camelCase, kebab-case, etc).

[View the tests](#)

Related Issues



Yay! No open issues!

CONCLUSIONES

RENDIMIENTO

TodoMVC Gzipped Bundle Size

 2.0KB

Stencil One

 3.6KB

Svelte

 8.8KB

LitElement

 13.3KB

Angular Ivy

 39.1KB

React

TRABAJA DONDE SEA

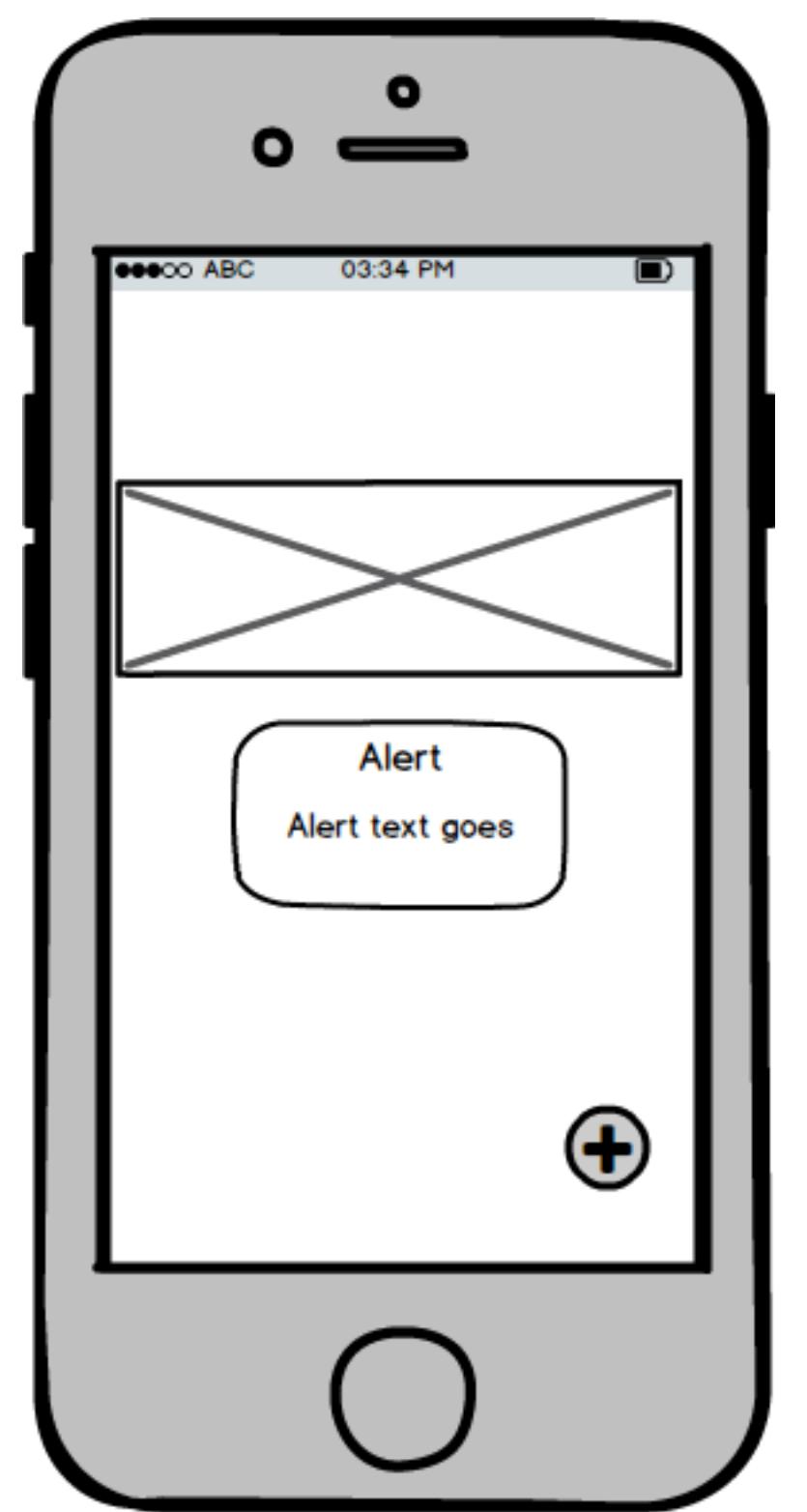
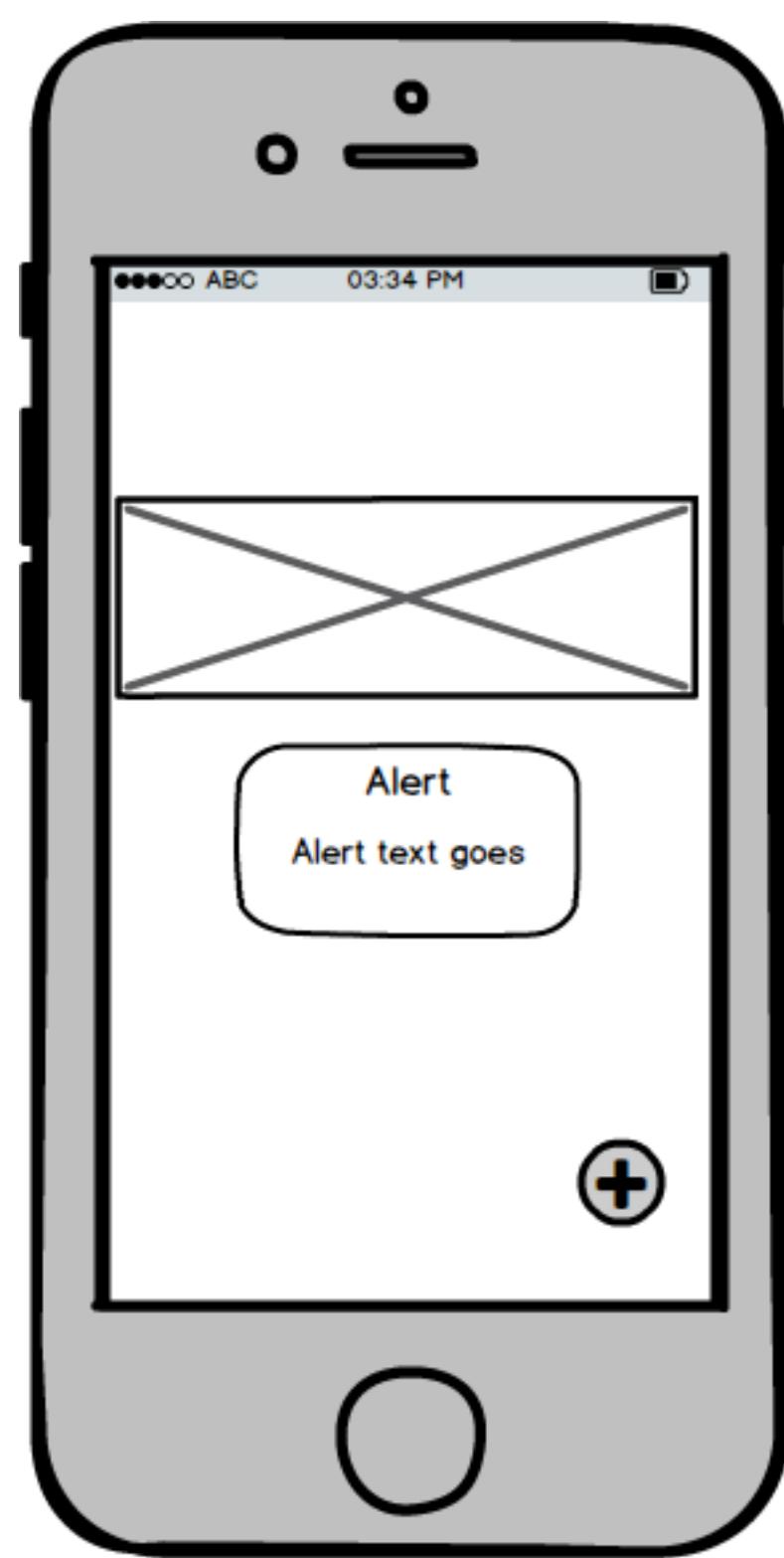
PERSONALIZABLES

PREPARADO PARA EL FUTURO

WC LIBRARIES

BROWSER SUPPORT

...UNA ULTIMA COSA



\$NPM INSTALL

