

Javascript moderno en Drupal sin desacoplar

Andrés Díaz Soto
Manatí



MANATÍ

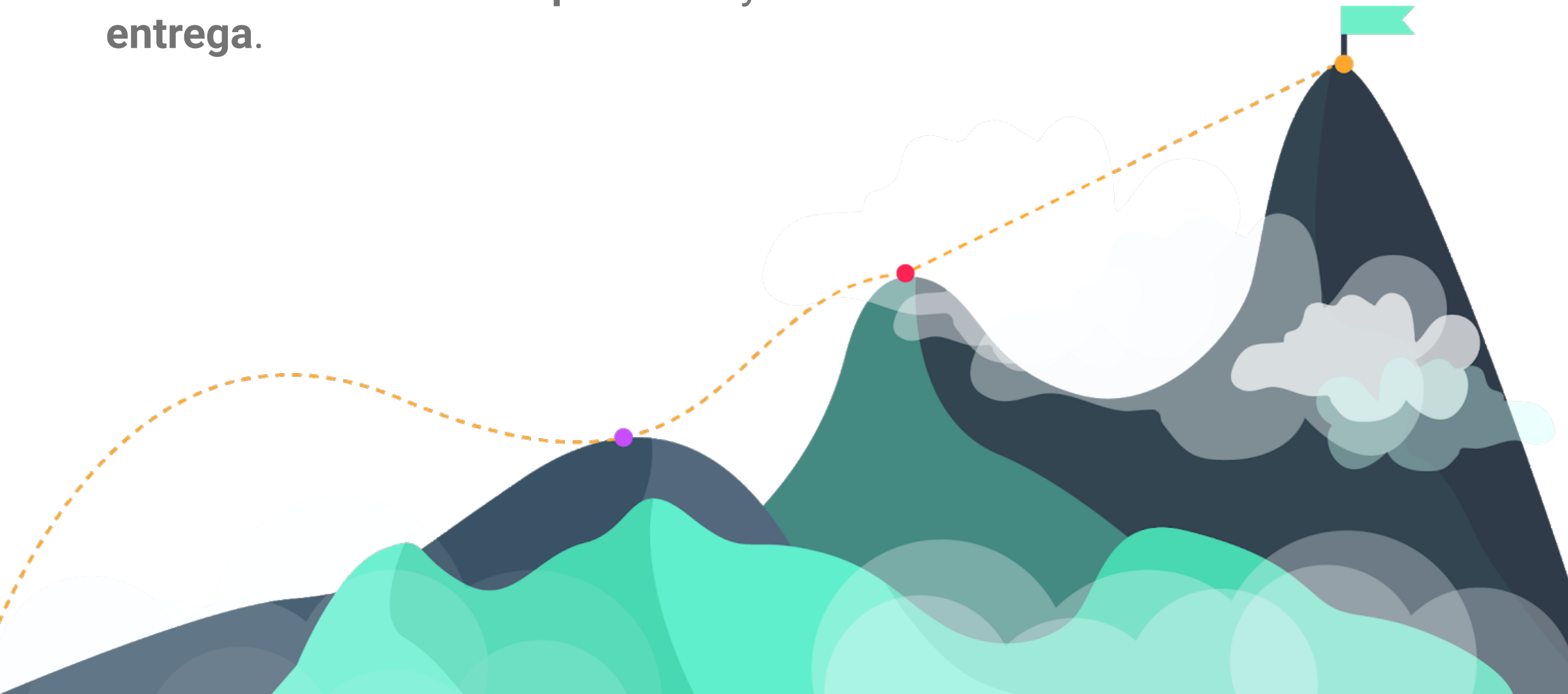
Andrés Díaz Soto

Frontend Developer en Manatí

@cienvaras

MANATÍ

Manatí es una agencia completa de **servicios de desarrollo web ágil** con un enfoque colaborativo de **diseño centrado en las personas** y excelencia en **entrega**.



Javascript en Drupal 8

jQuery

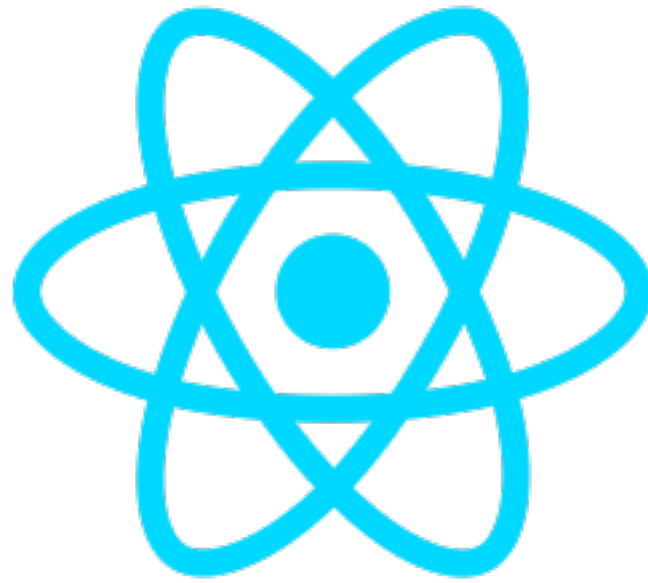
Backbone.js

Underscore.js

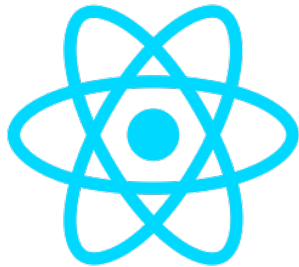
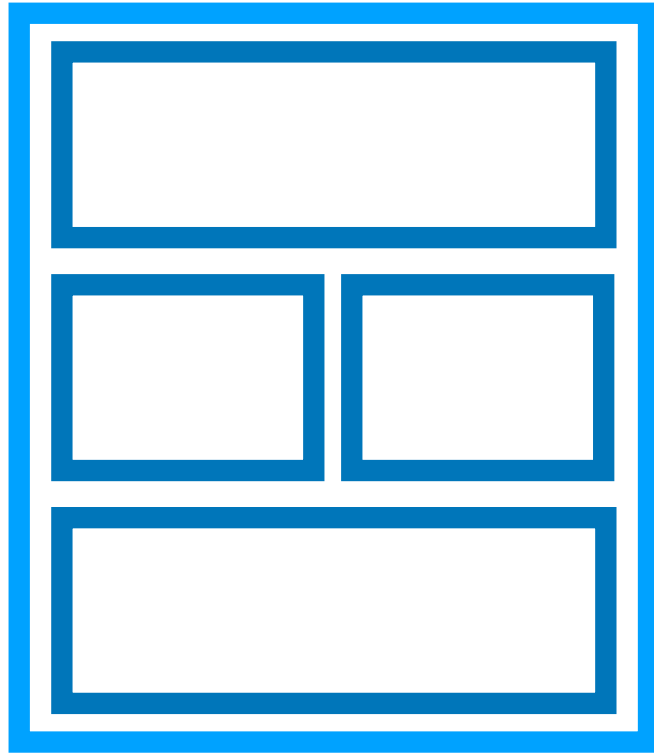
Javascript en Drupal 8

Manipulación “tradicional” del DOM

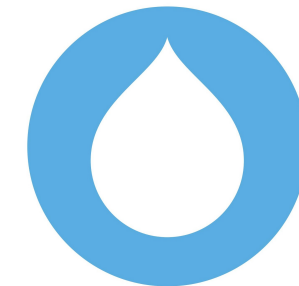
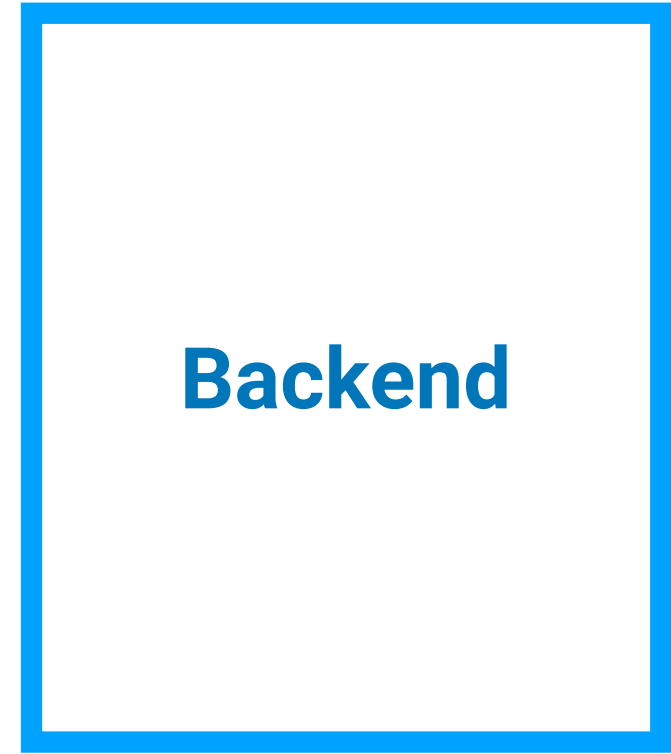
Creación de elementos de UI es complicada



Solución:
Desacoplar



Services



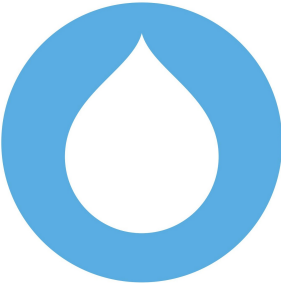
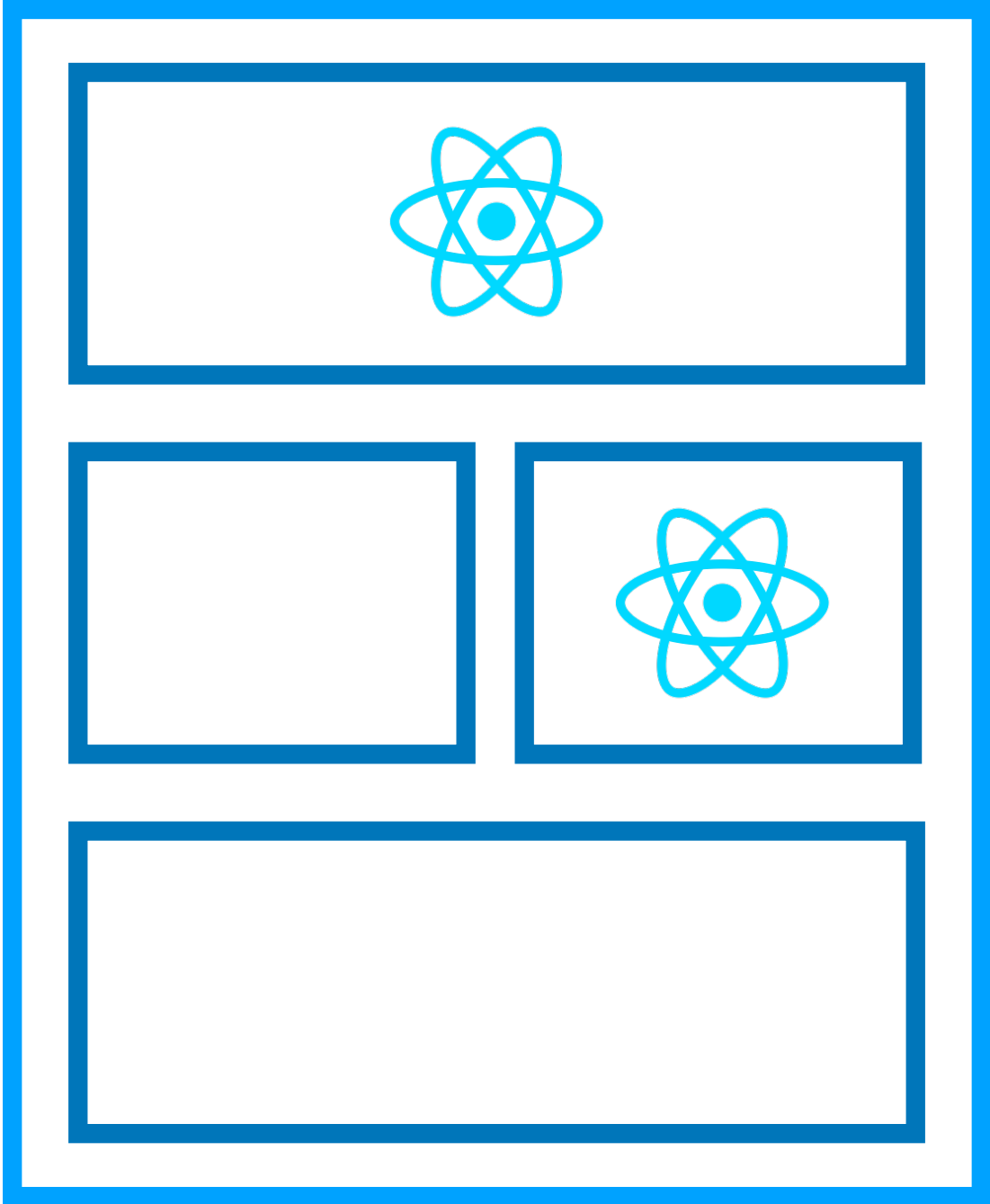
Sitios desacoplados

Múltiples codebases

Hosting puede complicarse

Es necesario construir toda la UI desde cero,
no pueden aprovecharse elementos que
provee Drupal

Otra solución:
Desacople parcial



Desacople parcial

Sólo se desacoplan componentes que lo necesitan

Todo está dentro del mismo codebase de Drupal

Permite aprovechar lo mejor de ambos mundos

Construyendo un
componente parcialmente
desacoplado

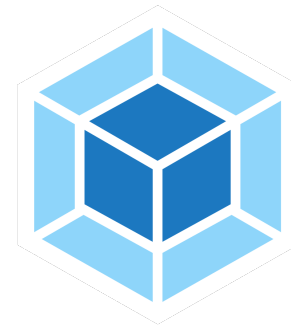
1. Definir un contenedor
2. Configurar un entorno de desarrollo para Javascript
3. Definir una biblioteca de Drupal
4. Cargar la biblioteca
5. Agregar la aplicación al contenedor

Definir un contenedor

```
class ReactVideo extends BlockBase {  
  
    ...  
  
    public function build() {  
        $build = [];  
        $build['#markup'] = '<div id="react-video"></div>';  
        return $build;  
    }  
}
```

modules/custom/react_video/src/Plugin/Block/ReactVideo.php

Configurar un entorno de desarrollo para Javascript



webpack



ESLint

package.json

```
{
  "scripts": {
    "dev": "webpack --mode=development -w",
    "build": "webpack --mode=production -p"
  },
  "devDependencies": {
    "@babel/core": "^7.5.0",
    "@babel/preset-env": "^7.5.0",
    "@babel/preset-react": "^7.0.0",
    "babel-eslint": "^10.0.2",
    "babel-loader": "^8.0.6",
    "eslint": "^5.3.0",
    "eslint-config-airbnb": "^17.1.1",
    "eslint-plugin-import": "^2.18.0",
    "eslint-plugin-jsx-a11y": "^6.2.3",
    "eslint-plugin-react": "^7.14.2",
    "webpack": "^4.35.3",
    "webpack-cli": "^3.3.5"
  },
  "dependencies": {
    "axios": "^0.19.0",
    "prop-types": "^15.7.2",
    "react": "^16.8.6",
    "react-dom": "^16.8.6"
  }
}
```

Webpack

```
const path = require('path');

module.exports = {
  entry: {
    'react-video': path.resolve(__dirname, 'src/index.js'),
  },
  output: {
    path: path.resolve(__dirname, 'dist'),
    filename: '[name].js',
  },
  devtool: 'source-map',
  resolve: {
    extensions: ['.js', '.jsx', '.json'],
    alias: {
      '@': path.resolve(__dirname, 'src'),
    },
  },
  module: {
    rules: [
      {
        test: /\.jsx?$/,
        exclude: /node_modules/,
        use: ['babel-loader', 'eslint-loader'],
      },
    ],
  },
};
```

modules/custom/react_video/js/webpack.config.js

Definir una biblioteca de Drupal

```
react-video:  
  version: 1.x  
  js:  
    js/dist/react-video.js: {}
```

modules/custom/react_video/react_video.libraries.yml

Cargar la biblioteca

(En un bloque)

```
class ReactVideo extends BlockBase {  
  
    ...  
  
    public function build() {  
        $build = [];  
        $build['#markup'] = '<div id="react-video"></div>';  
        $build['#attached']['library'][] = 'react_video/react-video';  
        return $build;  
    }  
}
```

modules/custom/react_video/src/Plugin/Block/ReactVideo.php

Cargar la biblioteca

(En una vista)

```
function react_filters_views_pre_render(ViewExecutable $view) {  
    if ($view→id() === 'events' && $view→current_display === 'page_1') {  
        $view→element['#attached']['library'][] = 'react_filters/react-filters';  
    };  
}
```

modules/custom/react_filters/react_filters.module

Agregar la aplicación al contenedor

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App';

const reactElement = document.getElementById('react-video');

ReactDOM.render(<App />, reactElement);
```

modules/custom/react_video/js/src/index.js

Agregar la aplicación al contenedor

(Si no tenemos un contenedor)

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App';

// Gets view element.
const view = document.querySelector('.view-id-events');

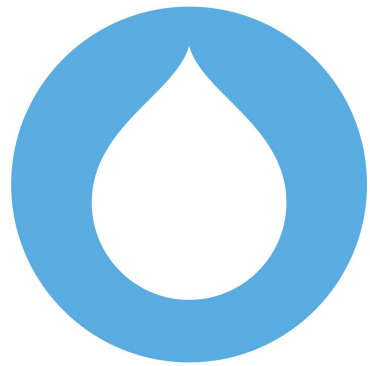
// Creates react container and appends it to the view element.
const reactElement = document.createElement('div');
reactElement.setAttribute('id', 'react-app');
view.insertAdjacentElement('beforebegin', reactElement);

ReactDOM.render(<App />, reactElement);
```

modules/custom/react_filters/js/src/index.js

Comunicándose con Drupal

Services



**Restful API
Services**

```
{ json:api }
```



GraphQL

drupalSettings

```
$data = {  
  // datos  
}  
  
$build['#attached']['drupalSettings']['myReactModule']['config'] = $data;
```

Juntándolo todo: Demostración

<https://github.com/cienvaras/drupalcamp2019-jsmoderno>