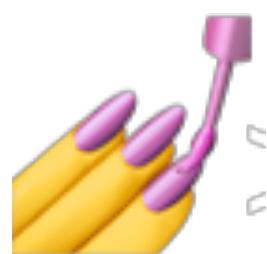




css in JS. **Estilos en React con styled-components**



Juan Carlos Ruiz Carrión
PcComponentes



css in JS. **Estilos en React con styled-components**

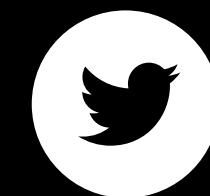
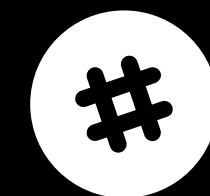
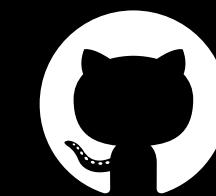


Juan Carlos Ruiz Carrión
PcComponentes



Juan Carlos Ruiz Carrión

UI / Front-end developer
@ PcComponentes



@klaufel

Panorama actual

A



CSS

Global
HD

Problemas frente a un mal uso de CSS

Cascada

Herencia

Especificidad

Cascada / Herencia / Especificidad



```
<div>
```

```
  <a class="button btn boton btn-primary link">My button</a>
```

```
</div>
```

Cascada / Herencia / Especificidad

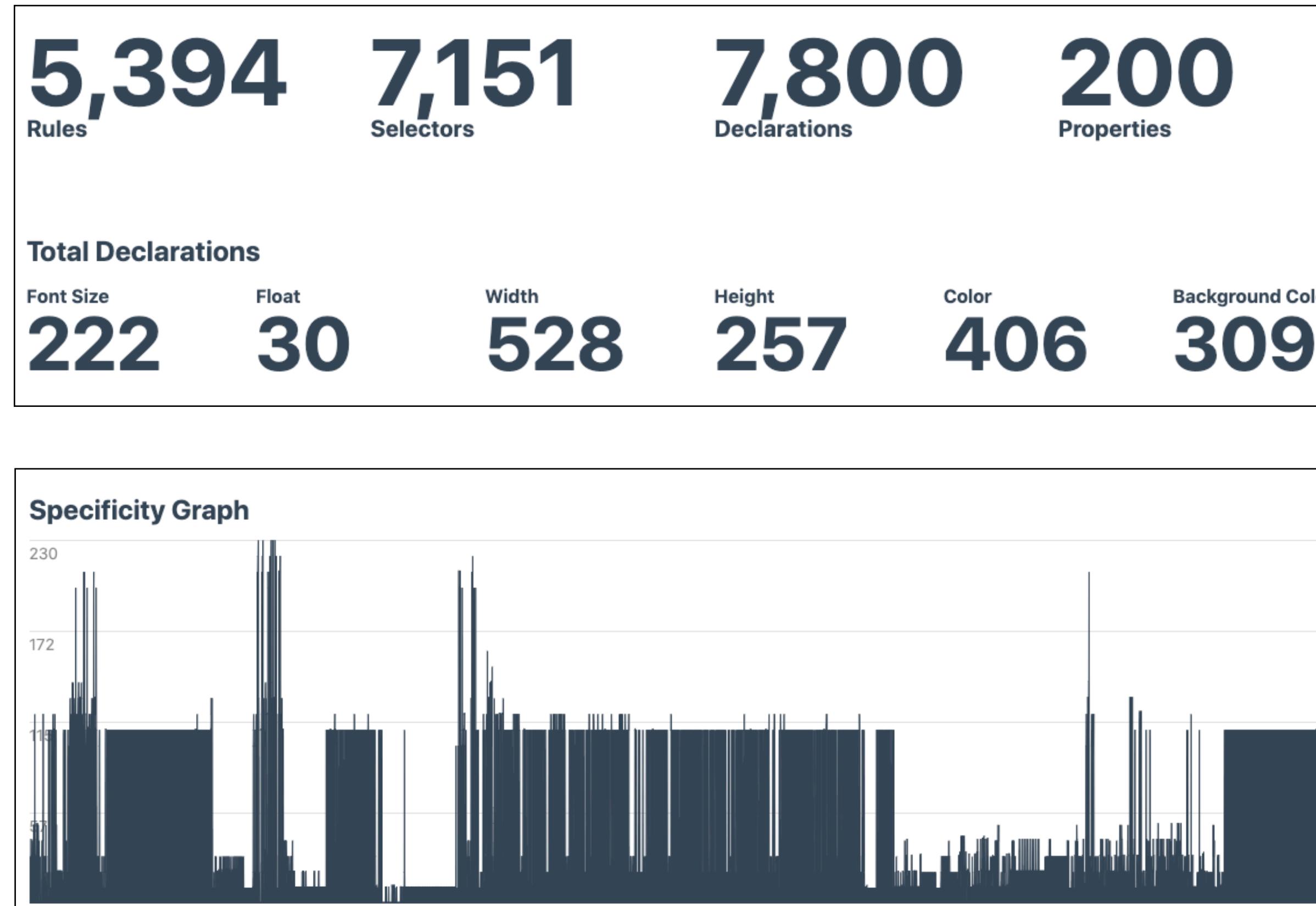
```
.button { ... }  
.btn { ... }  
div .btn { ... }  
div a.btn { ... }  
body div > .btn-primary { ... }  
.link { ... !important; }  
  
<div>  
  <a class="button btn boton btn-primary link">My button</a>  
</div>
```

¿Flexible? Si

¿Flexible? Si
¿Peligroso? También

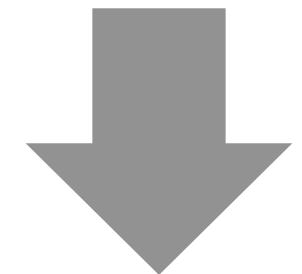
CSS stats

<https://es.yahoo.com/>



<https://cssstats.com/stats?url=http%3A%2F%2Fyahoo.com&name=Yahoo>

Un mal uso de la:
Cascada
Herencia
Especificidad



Arquitectura insostenible

Evolución y soluciones



Preprocesadores



Postprocesadores



Autoprefixer

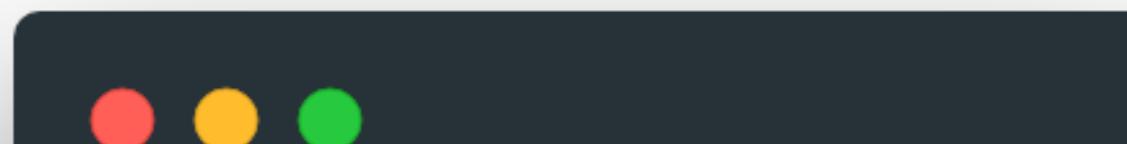


PostCSS

Metodologías y arquitecturas

Metodología BEM

(Block Element Modifier)



```
/* BEM methodology */

.button {
  background: red;
}

.button__icon {
  display: inline-block;
}

.button--fluid {
  width: 100%;
}
```

Arquitecturas

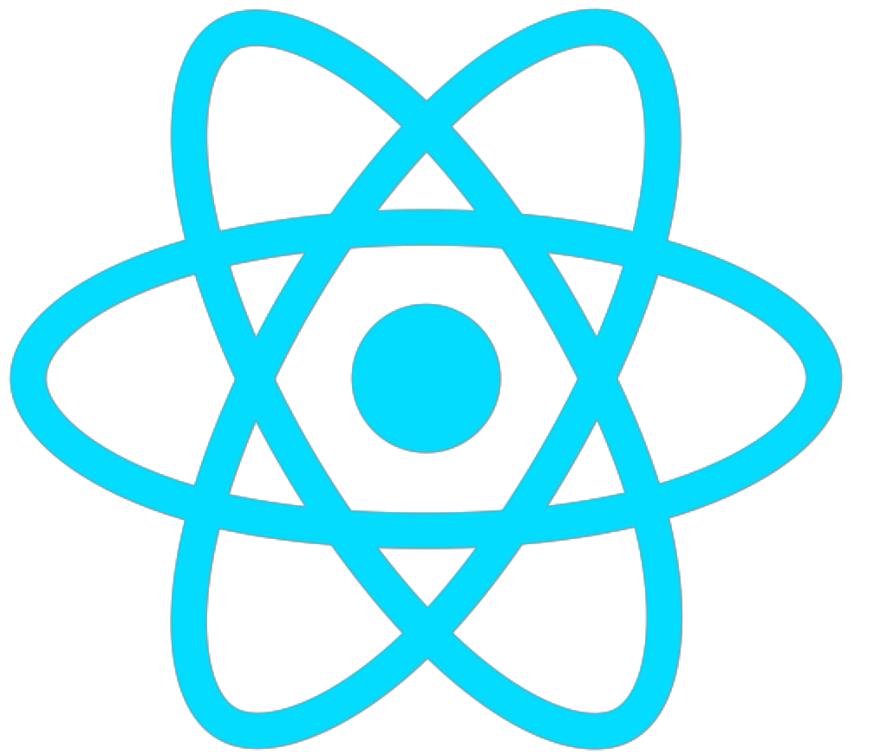
SMACSS

Scalable and Modular Architecture for CSS
(Base, Layout, Modules,
States and Theme)

ITCSS

Inverted Triangle CSS
(Settings, Tools, Generic, Elements,
Objects, Components, Trumps)

Filosofía de componentes



CSS

~~Cascading~~ Style Sheets

Component Style Sheets

CSS in JS

CSS in JS
NO SON
inline styles

CSS in JS

NO SIGNIFICA
dejar de escribir CSS

CSS in JS LO QUE SI ES

CSS in JS

Components everywhere

Soluciona carencias que tiene CSS como lenguaje y encaja perfectamente en la filosofía de componentes.

CSS in JS

Estilos aislados

No tendremos que preocuparnos del nombre que tienen las clases o de si habrá conflictos con otras.

Tendremos componentes completamente aislados listos para ser usados en cualquier parte de nuestra aplicación.

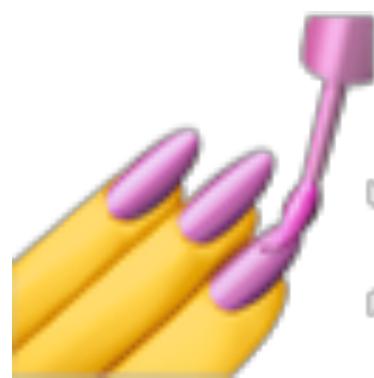
CSS in JS

Toda la potencia de JS

Es CSS, pero hipervitaminado con JS, el cual nos permitirá crear variables, funciones, hacer imports...

¿Nada nuevo? Entre otras cosas usar los imports de JS, nos permitirá optimizar el tamaño de nuestro CSS y cargar sólo el CSS mínimo necesario de los componentes que estén montados en ese momento.

Estilos en React con styled-components



¿Por qué styled-components?

styled-components

 Soporta todo el CSS

styled-components

- ✓ Soporta todo el CSS
- ✓ CSS global

styled-components

- ✓ Soporta todo el CSS
- ✓ CSS global
- ✓ Escribe CSS

styled-components

- ✓ Soporta todo el CSS
- ✓ CSS global
- ✓ Escribe CSS
- ✓ Estilos aislados

styled-components

- ✓ Soporta todo el CSS
- ✓ CSS global
- ✓ Escribe CSS
- ✓ Estilos aislados
- ✓ Auto prefixing

styled-components

- ✓ Soporta todo el CSS
- ✓ CSS global
- ✓ Escribe CSS
- ✓ Estilos aislados
- ✓ Auto prefixing
- ✓ No requiere ninguna configuración

Componente estilizado

Estilos condicionales

Animaciones

Theming

Estilos globales

Extensión de estilos



```
npm install styled-components
```

Componente estilizado

Componente estilizado



```
import styled from 'styled-components';

const Button = styled.a`  
  display: inline-block;  
  padding: 0.5rem 0;  
  background: black;  
  color: white;  
`;

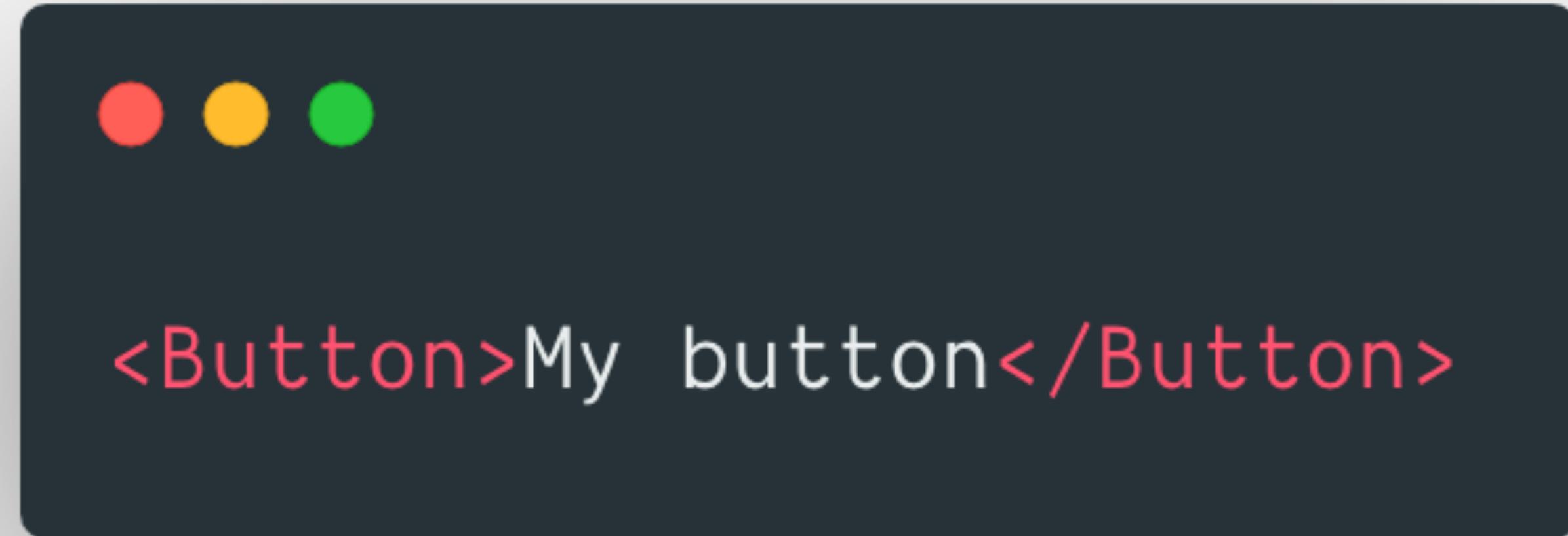
export default Button;
```

Componente estilizado

```
import styled from 'styled-components';

const Button = styled.a`  
  display: inline-block;  
  padding: 0.5rem 0;  
  background: black;  
  color: white;  
  
  &:hover {  
    background: blue;  
  }  
  
  &:focus {  
    background: red;  
  }  
`;  
  
export default Button;
```

Componente estilizado



Componente estilizado

```
▼<div id="root">
  <a class="eURmng">Default button</a>
</div>
```

```
.eURmng {           <style>...</style>
  display: inline-block;
  padding: ▶ 0.5rem 0;
  background: ▶ □ black;
  color: □ white;
}
```

Componente estilizado



```
<Button /> // <a> element  
<Button href="..." target="_blank" /> // <a> element  
<Button as="span" /> // <span> element  
<Button as="button" /> // <button> element  
<Button as="button" type="submit" /> // <button type="submit"> element  
  
<Button as="OtherComponent" {...props} /> // Other element
```



Estilos condicionales

Estilos condicionales

```
● ● ●

import styled from 'styled-components';

const Button = styled.a`  

  display: inline-block;  

  padding: 0.5rem 0;  

  color: white;  

  // Ternary operator  

  background: ${props => props.secondary ? 'red' : 'black'};  

`;  

export default Button;  

<Button />  

<Button secondary />
```

Estilos condicionales

```
import styled, { css } from 'styled-components';

const Button = styled.a`  
  display: inline-block;  
  padding: 0.5rem 0;  
  background: black;  
  color: white;  
  
  // Logical operator  
  ${props => props.secondary && css`  
    background: red;  
  `}  
`;

export default Button;

<Button />
<Button secondary />
```

Animaciones

Animaciones



```
import styled, { keyframes } from 'styled-components';

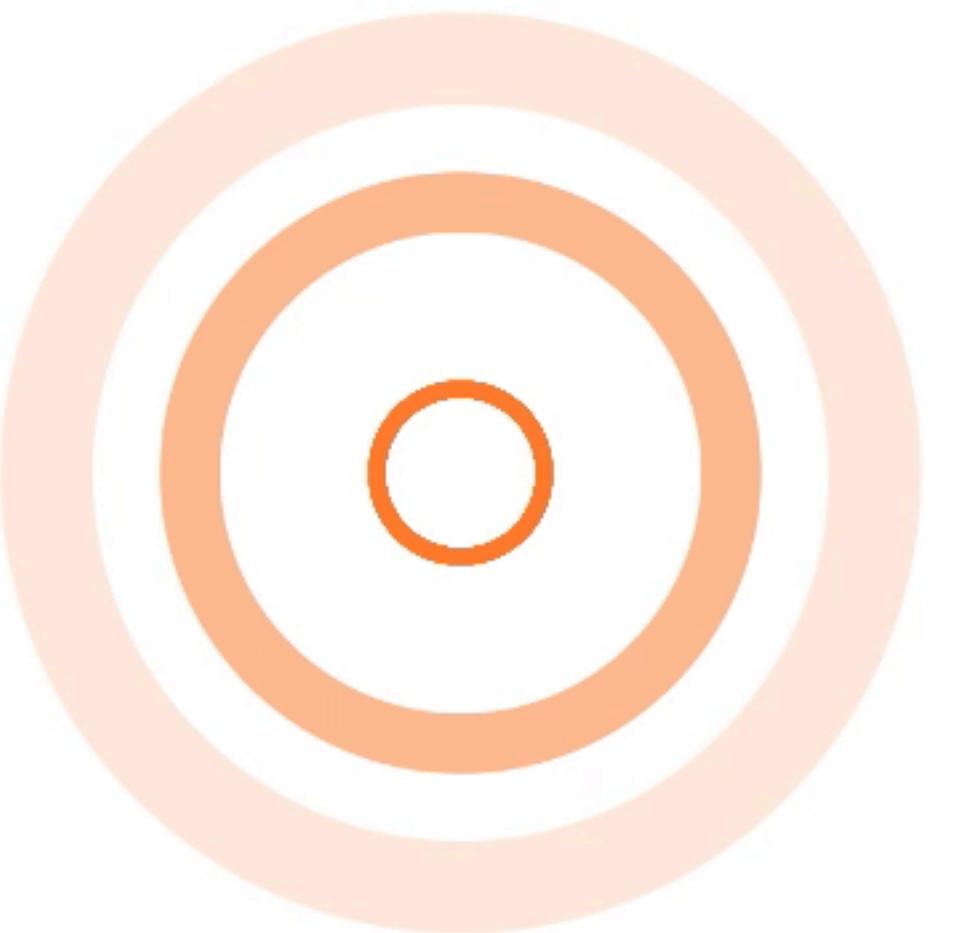
const animationRipple = keyframes`  
  0% {  
    transform: scale(0);  
    opacity: 1;  
  }  
  100% {  
    transform: scale(1);  
    opacity: 0;  
  }  
`;
```

Animaciones



```
const Item = styled.div`  
  animation: ${animationRipple} 1.5s cubic-bezier(0, 0.2, 0.8, 1) infinite;  
`;
```

Animaciones



```
import React from 'react';
import styled, { keyframes } from 'styled-components';

const animationRipple = keyframes` 
  0% {
    transform: scale(0);
    opacity: 1;
  }
  100% {
    transform: scale(1);
    opacity: 0;
  }
`;

const Wrapper = styled.div`...`;

const Item = styled.div`
  animation: ${animationRipple} 1.5s cubic-bezier(0, 0.2, 0.8, 1) infinite;

  &:nth-child(2) {
    animation-delay: -0.5s;
  }
  &:nth-child(3) {
    animation-delay: -1s;
  }
`;

const Loader = () => (
  <Wrapper>
    <Item />
    <Item />
    <Item />
  </Wrapper>
);

export default Loader;
```

Theming

Theming

```
// Define what props.theme will look like
const theme = {
  colors: {
    main: '#ff6000',
    second: '#cc4d00',
    dark: '#212529',
    gray: '#6c757d',
    light: '#d8d8d8'
  },
  layout: {
    container: '2rem',
    gutter: '1rem'
  },
  fonts: {
    family: "'Roboto', sans-serif",
    size: '1rem'
  },
  breakpoints: {
    xs: '20rem',
    sm: '36rem',
    md: '48rem',
    lg: '62rem',
    xl: '75rem'
  }
  // ...more config theme styles
};

export default theme;
```

Theming

```
import React from 'react';
import ReactDOM from 'react-dom';
import { ThemeProvider } from 'styled-components';
import theme from './theme';
import App from './components/App';

ReactDOM.render(
  <ThemeProvider theme={theme}>
    <App />
  </ThemeProvider>,
  document.getElementById('root'),
);
```

Theming

```
import styled, { css } from 'styled-components';

const Button = styled.button`  
background: ${props => props.theme.colors.main};  
border-radius: ${props => props.theme.styles.radius};  
color: #ffffff;  
cursor: ${props => (props.disabled ? 'not-allowed' : 'pointer')};  
display: inline-block;  
font-family: inherit;  
font-size: ${props => props.theme.fonts.size};  
font-weight: 400;  
min-height: ${props => (props.tiny ? '20px' : '45px')};  
padding: ${props => (props.tiny ? '2px 6px' : '12px 30px')};  
  
/* Secondary button variation styles */  
${props => props.secondary && css`  
background-color: #eeeeee;  
border-color: ${props.theme.colors.main};  
color: ${props.theme.colors.main},  
`}  
  
&:hover {  
background-color: ${props => props.theme.colors.second};  
border-color: ${props => props.theme.colors.second};  
box-shadow: ${props => props.theme.styles.shadow};  
color: #ffffff;  
}  
;  
  
export default Button;
```

Estilos globales

Estilos globales

Reinicio de estilos (Normalize)

Fuentes del proyecto

Estilos genéricos

Estilos globales

```
import { createGlobalStyle } from 'styled-components';

const GlobalStyle = createGlobalStyle`  
  
  @import url('https://fonts.googleapis.com/css?family=Roboto:400,700');  
  
  body {  
    -webkit-font-smoothing: antialiased;  
    -moz-osx-font-smoothing: grayscale;  
    font-family: ${props => props.theme.fonts.family};  
    margin: 0;  
    padding: 0;  
    background: #fff;  
  }  
  
  // More styles...  
`;  
  
export default GlobalStyle;
```

Estilos globales



```
import React from 'react';
import ReactDOM from 'react-dom';
import { ThemeProvider } from 'styled-components';
import theme from './theme';
import GlobalStyle from './GlobalStyle';
import App from './components/App';

ReactDOM.render(
  <ThemeProvider theme={theme}>
    <>
      <GlobalStyle />
      <App />
    </>
  </ThemeProvider>,
  document.getElementById('root'),
);
```

Extensión de estilos

Extensión de estilos



```
import styled from 'styled-components';

const Input = styled.input`  
background-color: #ffffff;  
background-image: none;  
border-radius: ${props => props.theme.styles.radius};  
border: 1px solid ${props => props.theme.colors.light};  
color: ${props => props.theme.colors.dark};  
display: block;  
font-family: inherit;  
font-size: ${props => props.theme.fonts.size};  
`;  
  
export default Input;
```

Extensión de estilos

```
import styled from 'styled-components';
import Input from '../Input';

const Select = styled(Input)
  .attrs({
    as: 'select',
  })
  .background-image: url('data:image/svg+xml; charset=UTF-8, ...');
  .background-position: center right 15px;
  .background-repeat: no-repeat;
  .cursor: pointer;
`;

export default Select;
```

Extensión de estilos

(Componentes de terceros)



```
import styled from 'styled-components';
import Button from 'react-admin';

const MyButton = styled(Button)`  

  background: ${props => props.theme.colors.main};  

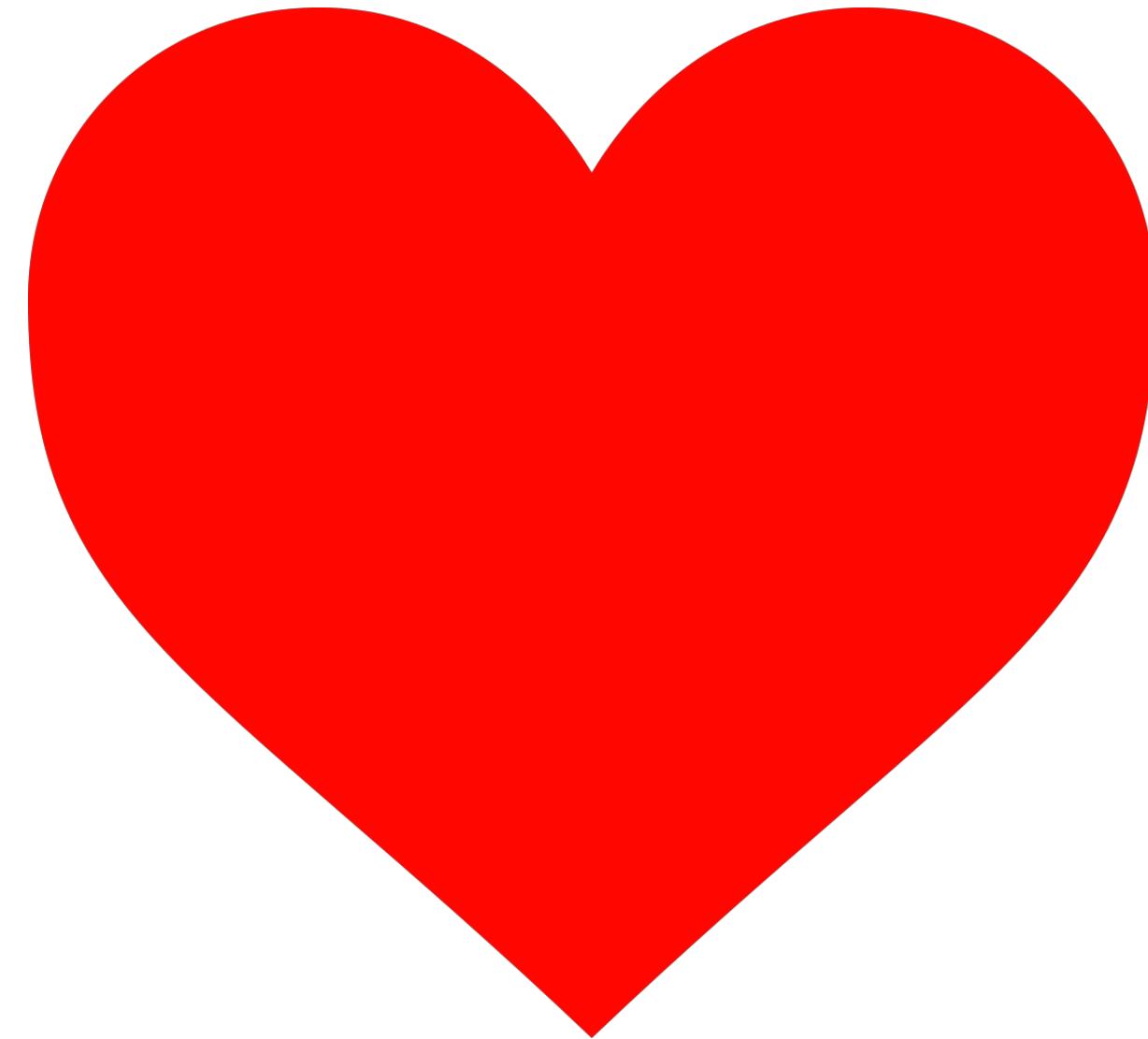
`;  
  

export default MyButton;
```

Extensión de estilos (Componentes de terceros)

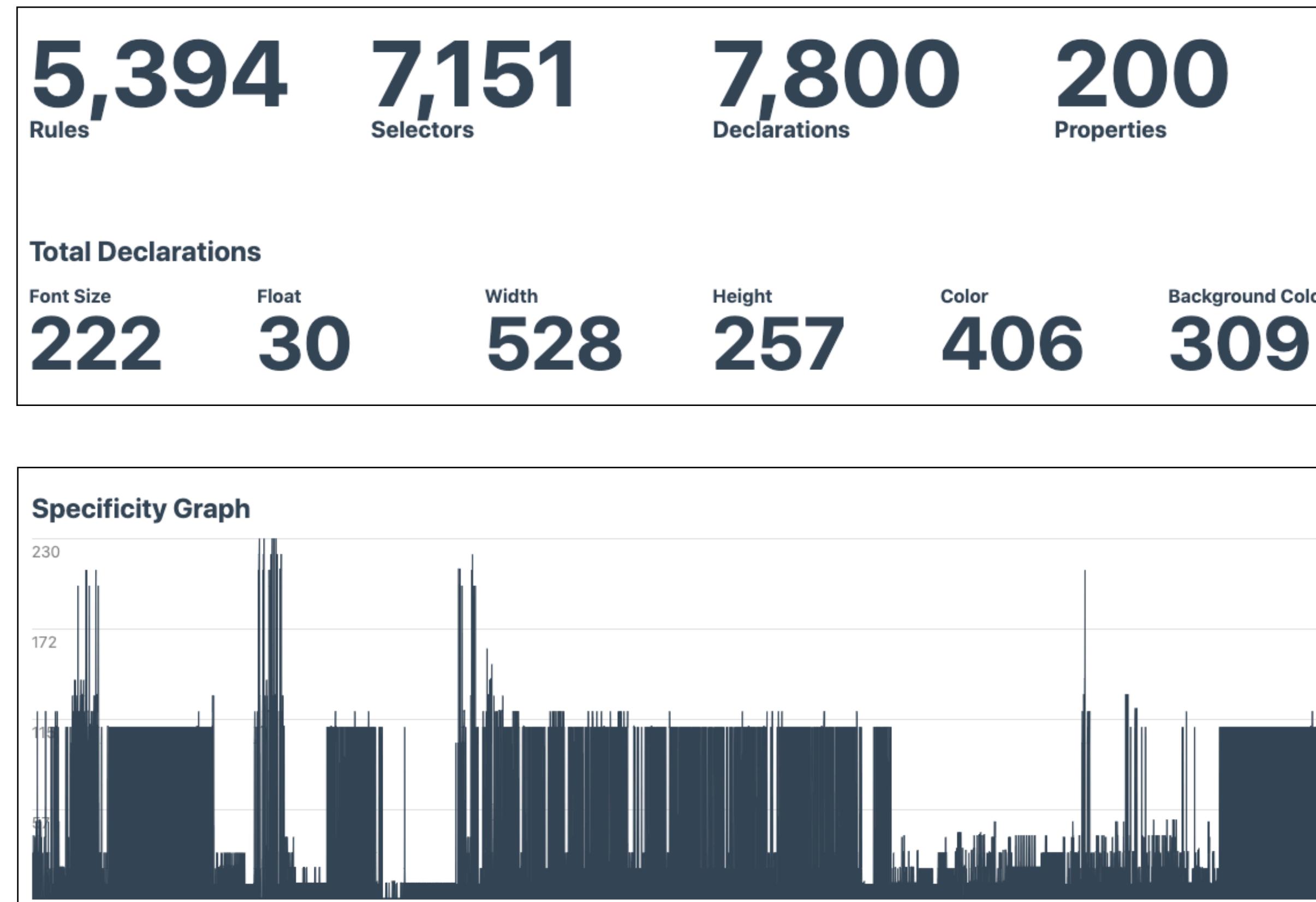
```
import styled from 'styled-components';
import Wysiwyg from 'react-admin';

const MyWysiwyg = styled(Wysiwyg)`\n  width: 100%;\n  display: block;\n  border: 1px solid ${props => props.theme.colors.main};\n\n  .panel {\n    background: ${props => props.theme.colors.main};\n  }\n\n  h1, h2, h3 {\n    color: ${props => props.theme.colors.dark};\n  }\n`;\n\nexport const MyWysiwyg;
```



CSS stats

<https://es.yahoo.com/>

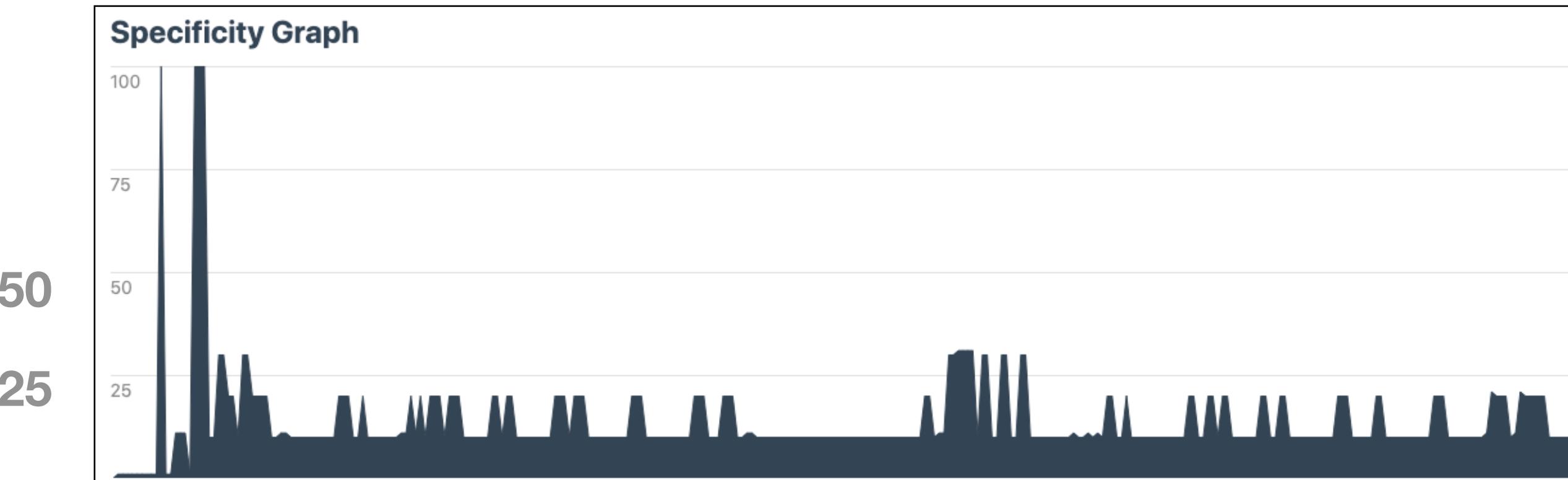
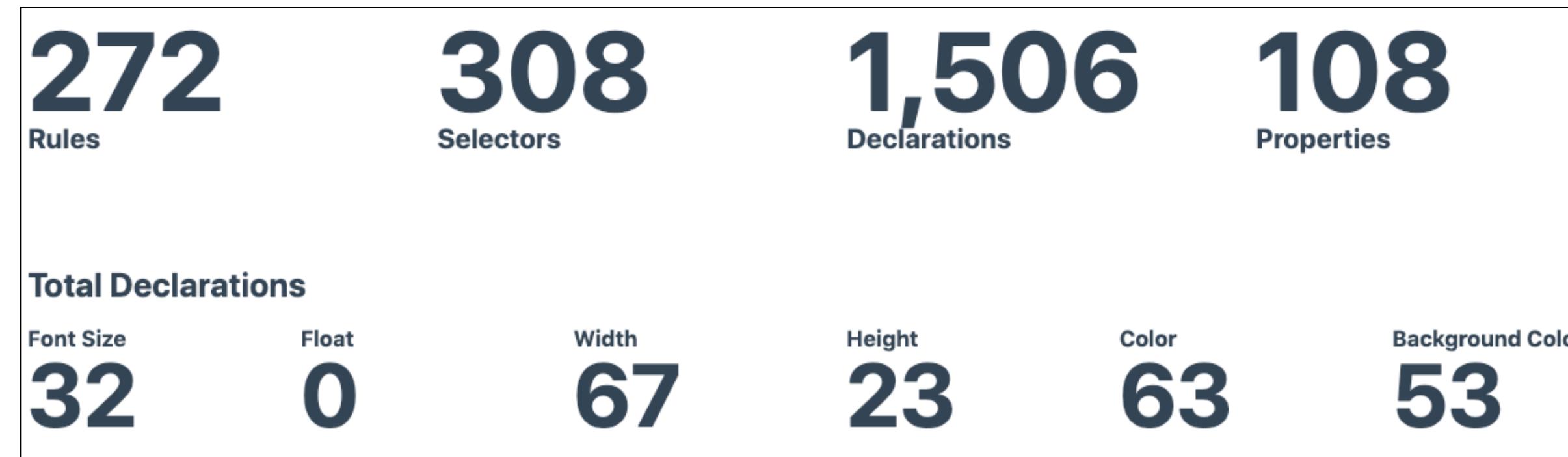


<https://cssstats.com/stats?url=http%3A%2F%2Fyahoo.com&name=Yahoo>

CSS stats



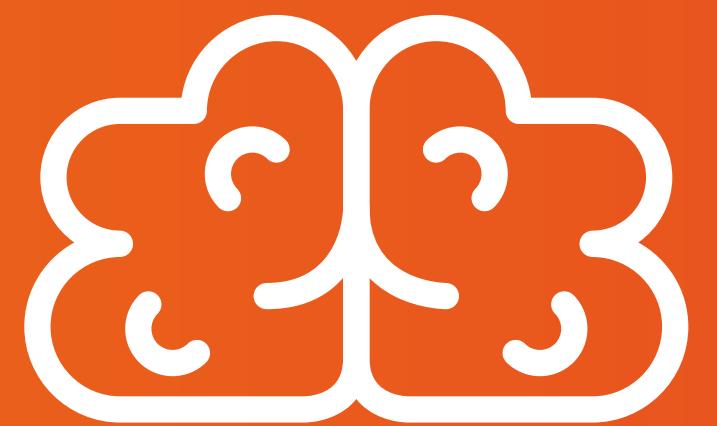
<https://www.ticketmaster.es>



<https://cssstats.com/stats?url=https%3A%2F%2Fwww.ticketmaster.es&ua=Browser%20Default>

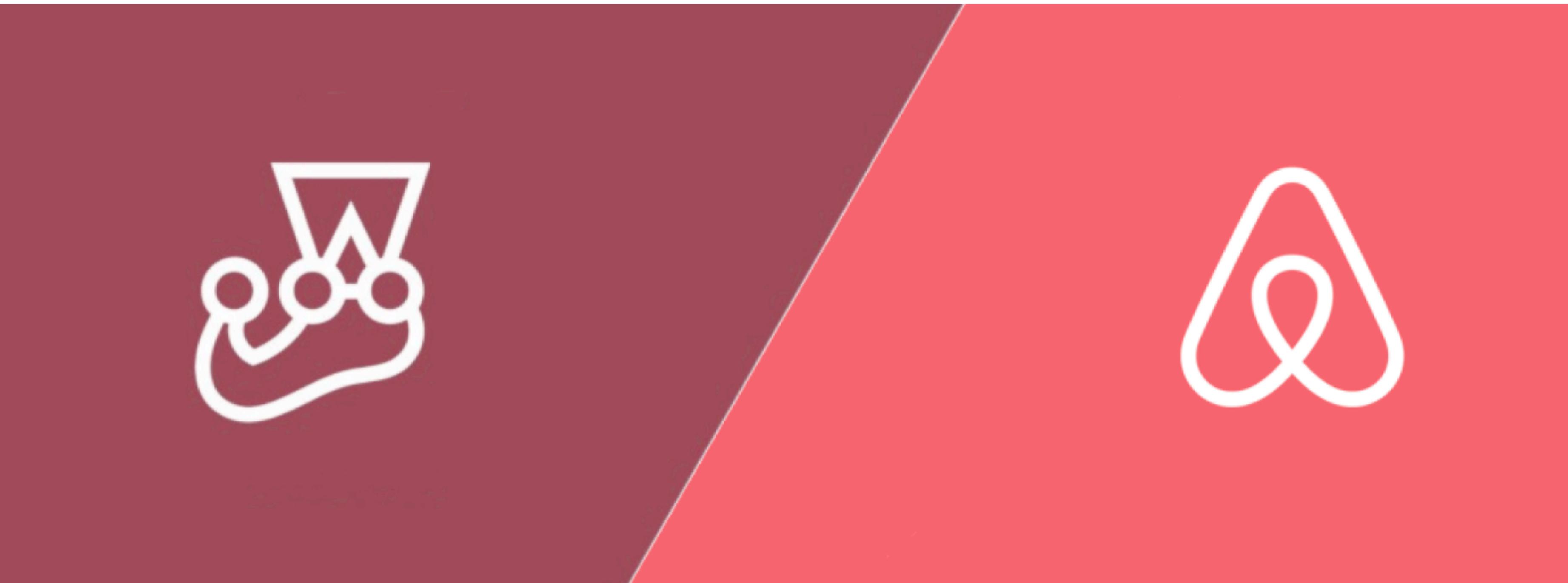


Testing de styled-components



Testing

(jest & enzyme)



Testing

(jest & enzyme)



```
npm install jest jest-styled-components  
npm install enzyme enzyme-adapter-react-16
```

Testing

(jest & enzyme)



```
"scripts": {  
  "test": "jest",  
  "test:update": "jest --updateSnapshot"  
}
```

Fichero de test

```
● ● ●  
  
import React from 'react';  
import 'jest-styled-components';  
import Adapter from 'enzyme-adapter-react-16';  
import Enzyme, { render } from 'enzyme';  
import Button from '../';  
  
Enzyme.configure({ adapter: new Adapter() });  
  
describe('Button', () => {  
  it('it works default button', () => {  
    const component = render(<Button />);  
    expect(component).toMatchSnapshot();  
  });  
});
```

Testing (npm run test)



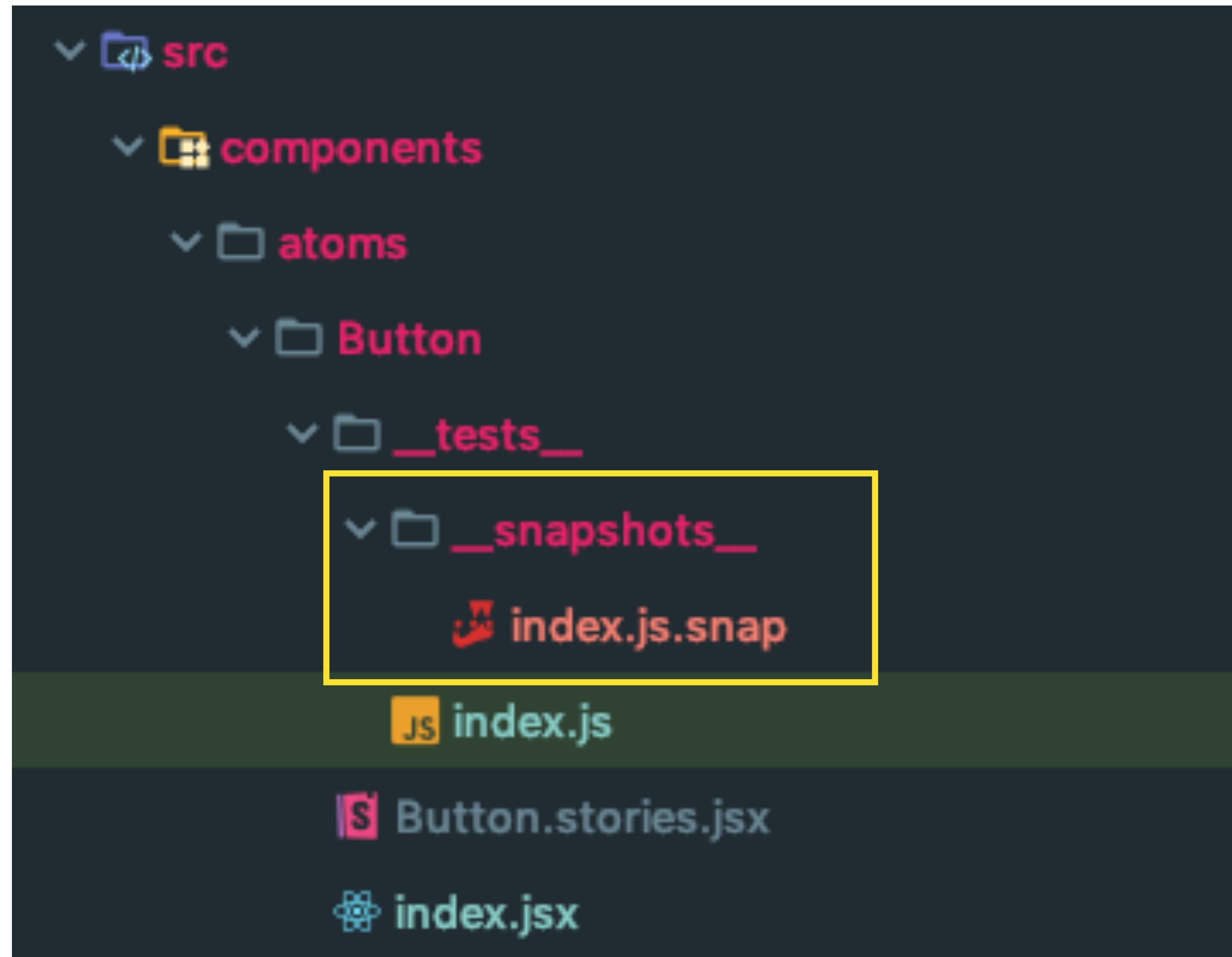
```
pccstorybook — pccommacbook1@MacBook-Pro-de-pccom — ..../pccstorybook — -zsh — 120x28
[ pccommacbook1@MacBook-Pro-de-pccom ~ /src/pccstorybook ⌘ tests •+ npm run test

> pccstorybook@1.0.0 test /Users/pccommacbook1/src/pccstorybook
> jest

PASS  src/components/atoms/Button/__tests__/index.js
  Button
    ✓ 1. it works default component (25ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   1 passed, 1 total
Time:        1.698s, estimated 2s
Ran all test suites.
pccommacbook1@MacBook-Pro-de-pccom ~ /src/pccstorybook ⌘ tests •+
```

Generación de tests



Snapshots

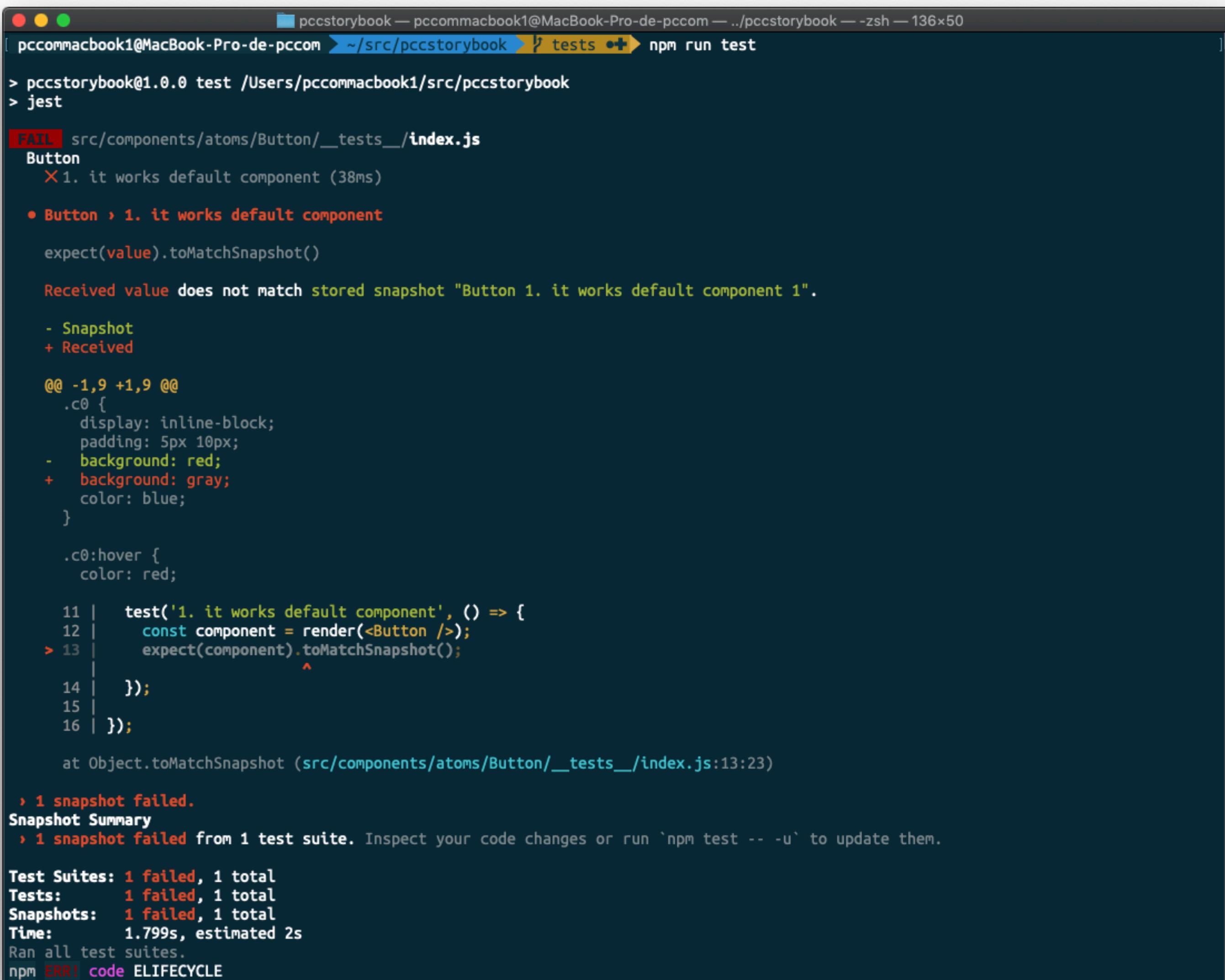
```
// Jest Snapshot v1, https://goo.gl/fbAQLP

exports[`Button 1. it works default component 1`] =
.c0 {
  display: inline-block;
  padding: 5px 10px;
  background: gray;
  color: blue;
}

.c0:hover {
  color: red;
  background: black;
  font-weight: bold;
  border-color: red;
}

<button
  class="c0"
/>
`;
```

Modificación y test



A screenshot of a terminal window on a Mac OS X system. The title bar shows the path: pccstorybook — pccmacbook1@MacBook-Pro-de-pccom —/pccstorybook — -zsh — 136x50. The command run is npm run test. The output shows a failing Jest test for the Button component. The error message indicates that the received value does not match the stored snapshot, specifically regarding the background color. The test code uses a snapshot comparison to assert the component's state.

```
pccstorybook@1.0.0 test /Users/pccmacbook1/src/pccstorybook
> jest

 FAIL  src/components/atoms/Button/__tests__/index.js
  Button
    ✕ 1. it works default component (38ms)

  ● Button > 1. it works default component

    expect(value).toMatchSnapshot()

    Received value does not match stored snapshot "Button 1. it works default component 1".
      - Snapshot
      + Received

      @@ -1,9 +1,9 @@
      .c0 {
        display: inline-block;
        padding: 5px 10px;
      - background: red;
      + background: gray;
        color: blue;
      }

      .c0:hover {
        color: red;

    11 |   test('1. it works default component', () => {
    12 |     const component = render(<Button />);
    > 13 |     expect(component).toMatchSnapshot();
          ^
    14 |   });
    15 |
    16 | });

  at Object.toMatchSnapshot (src/components/atoms/Button/__tests__/index.js:13:23)

  > 1 snapshot failed.

  Snapshot Summary
  > 1 snapshot failed from 1 test suite. Inspect your code changes or run `npm test -- -u` to update them.

  Test Suites: 1 failed, 1 total
  Tests:       1 failed, 1 total
  Snapshots:  1 failed, 1 total
  Time:        1.799s, estimated 2s
  Ran all test suites.
  npm ERR! code ELIFECYCLE
```

Modificación y test

```
Button
  ✘ 1. it works default component (38ms)

● Button › 1. it works default component

  expect(value).toMatchSnapshot()

  Received value does not match stored snapshot "Button 1.

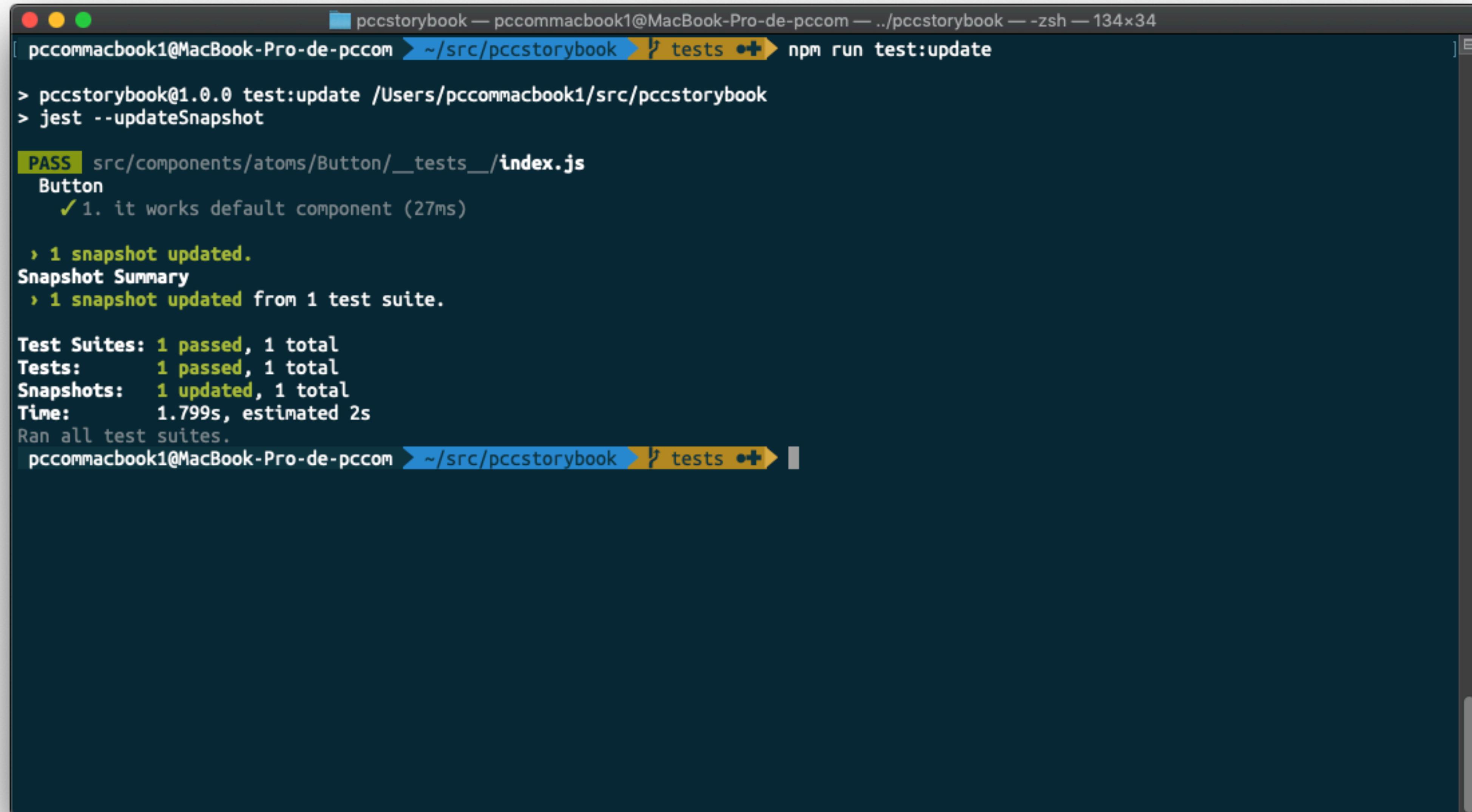
    - Snapshot
    + Received

@@ -1,9 +1,9 @@
.c0 {
  display: inline-block;
  padding: 5px 10px;
- background: red;
+ background: gray;
  color: blue;
}

.c0:hover {
```

Testing

(npm run test:updated)



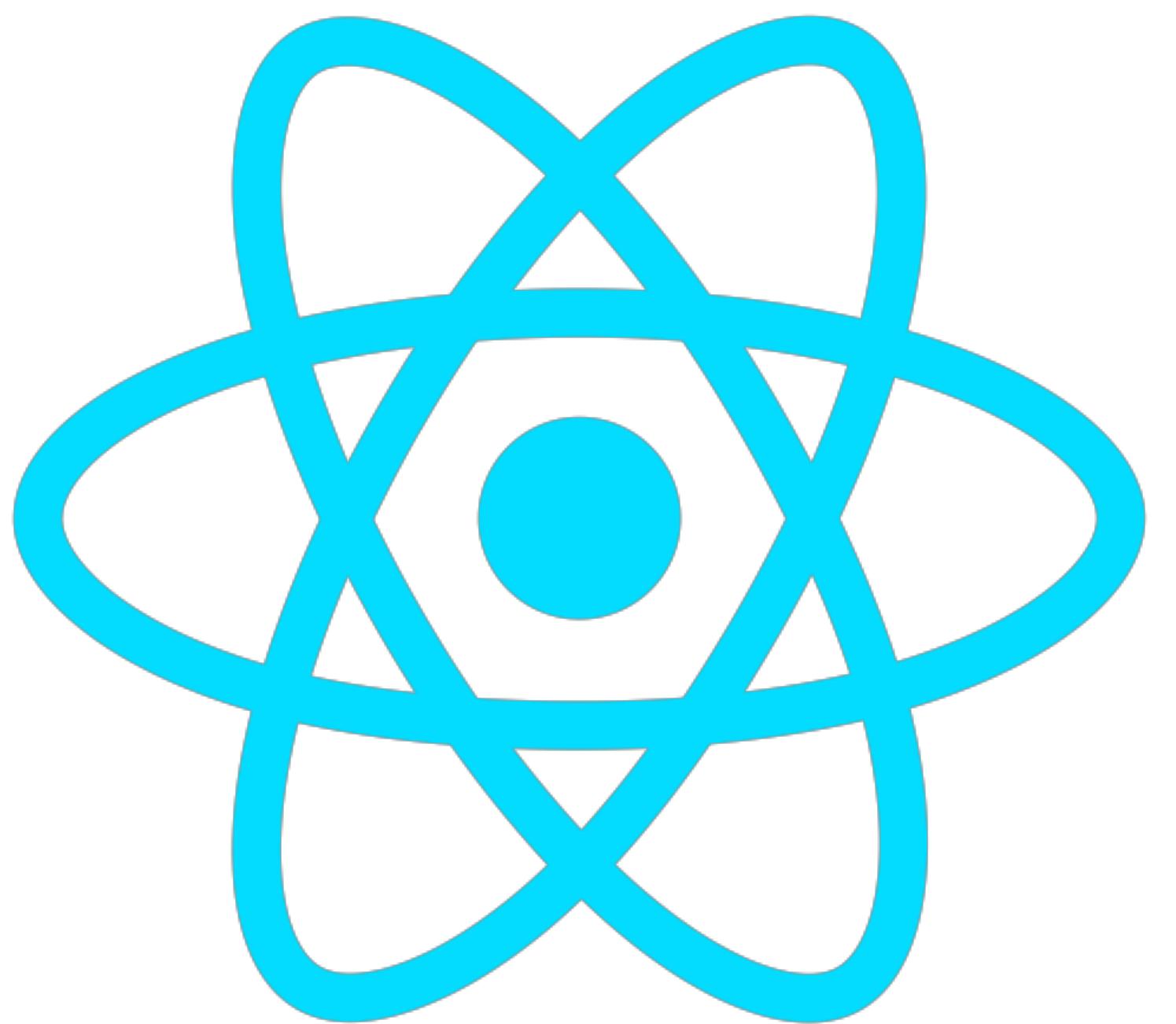
```
pccstorybook — pccommacbook1@MacBook-Pro-de-pccom — ..../pccstorybook — -zsh — 134x34
[ pccommacbook1@MacBook-Pro-de-pccom ~ /src/pccstorybook tests •+ npm run test:update

> pccstorybook@1.0.0 test:update /Users/pccommacbook1/src/pccstorybook
> jest --updateSnapshot

PASS  src/components/atoms/Button/__tests__/index.js
  Button
    ✓ 1. it works default component (27ms)

    > 1 snapshot updated.
  Snapshot Summary
    > 1 snapshot updated from 1 test suite.

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   1 updated, 1 total
Time:        1.799s, estimated 2s
Ran all test suites.
pccommacbook1@MacBook-Pro-de-pccom ~ /src/pccstorybook tests •+ |
```





<https://github.com/styled-components/vue-styled-components>



LA RESIDENCIA



¡GRACIAS!

¿Dudas o preguntas?