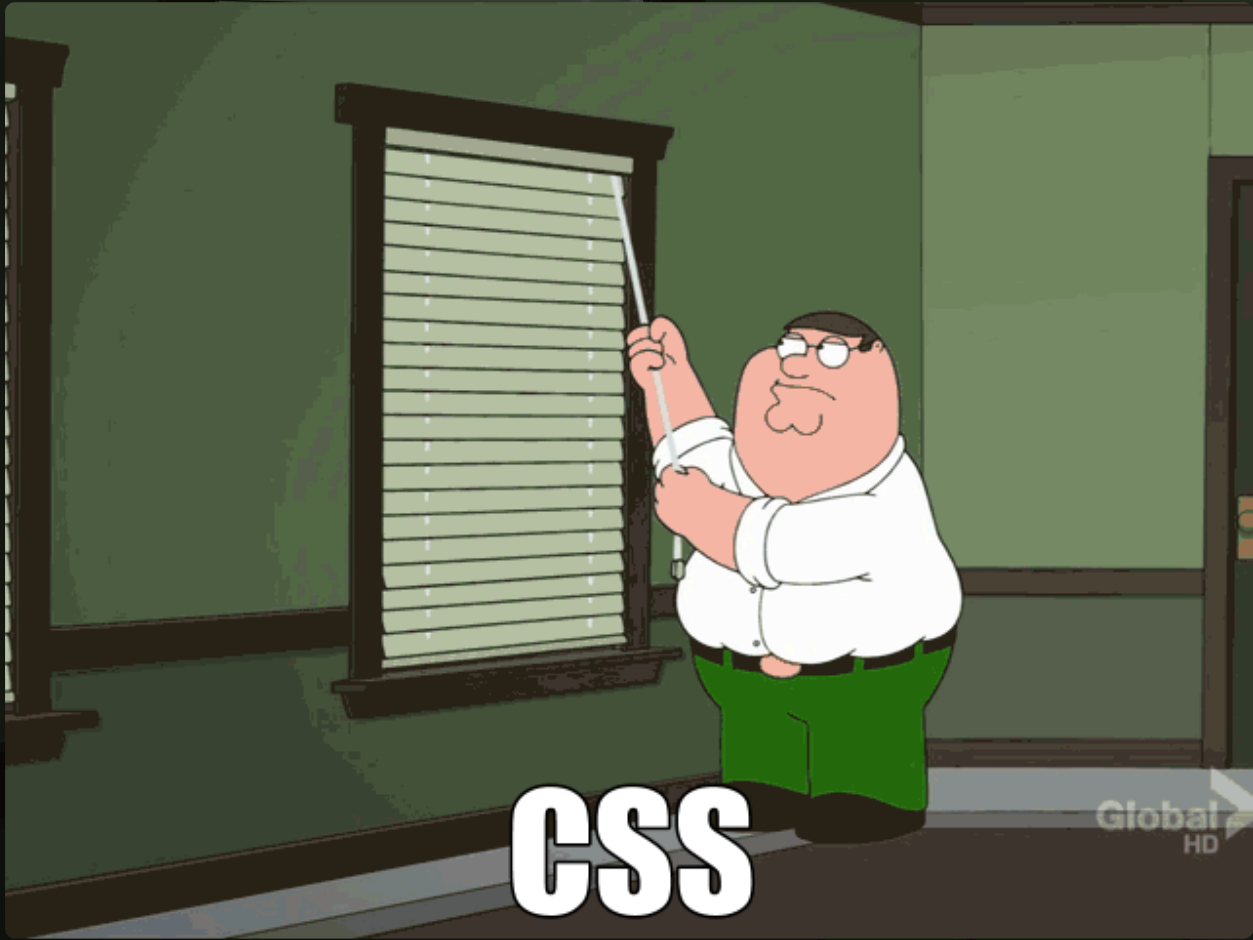


AS DIVERSAS FORMAS DO

# CSS Moderno.



**Uriell Viana**



CSS

Global HD

## Antes do CSS:

---

```
<MULTICOL COLS="3" GUTTER="25">  
  <P>  
    <FONT SIZE="4" COLOR="RED">  
      Texto quebrado em colunas  
    </FONT>  
  </P>  
</MULTICOL>
```



Channels Favorites History Search

Find It

Product Information  
Customer Support  
Technology & Research  
Developer World  
Groups & Interests  
Resources Online  
About Apple

Channels Favorites History Search

Find It

Product Information  
Customer Support  
Technology & Research  
Developer World  
Groups & Interests  
Resources Online  
About Apple

Apple Sites Worldwide

- Switzerland
- Taiwan
- Turkey
- UK & Ireland
- United States

Go

- Switzerland
- Taiwan
- Turkey
- UK & Ireland
- United States

Go

JULY 14

# Welcome to Apple



## Introducing CyberDrive

Register today for a free CD-ROM.

### EMATE 300

Mobile, Affordable, & Smart



### MOVIES FROM MARS



QuickTime VR Takes You Out of this World

#### What's Hot

**Preorder Mac OS 8**  
Now you can [preorder Mac OS 8](#), described by Macworld as "the most comprehensive update to the Mac OS in years, sporting a bold new look, a speedier Finder, more shortcuts and integrated Internet functions."

**Be the First to Know**  
Learn about new Macintosh software releases the moment they become available. Check [Hot Mac Products](#) to hear about programs like Speed Demon, ReBirth RB-338 and QuickCRC.

**Want a PowerBook?**  
Qualify to win a [PowerBook 3400/200](#) by [entering](#) this month's Apple

**Newton Connects**  
Newton, Inc., will enhance network connectivity for Newton-based devices this fall via [Newton Internet Enabler](#).

speedier Finder, more shortcuts and integrated Internet functions."

**Want a PowerBook?**  
Qualify to win a [PowerBook 3400/200](#) by [entering](#) this month's Apple

ReBirth RB-338 and QuickCRC.

**Newton Connects**  
Newton, Inc., will enhance network connectivity for Newton-based devices this fall via [Newton Internet Enabler](#)



MOVIES FROM MARS  
QuickTime VR Takes You Out of this World

Só que, alguns dos  
problemas da época,  
**acontecem até hoje.**

## **3 Problemas persistentes do CSS:**

- **Namespace global**
- **Arquivos muito grandes**
- **Isolamento**

Mas como esses problemas  
**foram resolvidos?**

# Metodologias comumente utilizadas:

- **SMACSS**  
Guia & boas práticas  
Escalável & modular
- **OOCSS**  
Estrutura **!=** aparência  
Princípio de responsabilidade **única**
- **BEM**  
Escopo **controlado**  
Estrutura simples e **robusta**



Veio a necessidade de fazer  
**MORE** with **LESS** 😎

# Pré- processamento de CSS:

- SASS  
[sass-lang.com](http://sass-lang.com)
- LESS  
[lesscss.org](http://lesscss.org)
- Stylus  
[stylus-lang.com](http://stylus-lang.com)

Com isso veio o hype de **variáveis, aninhamento, mixins e funções.**

Mas nós acabamos  
**duplicando código** 🤔

Por exemplo...

## HTML estruturado com o BEM:

---

```
<table class="Table">
  <thead class="TableHeader">
    <tr class="TableRow">
      <th class="TableCell TableCell--header">Product</th>
      <th class="TableCell TableCell--header">Price</th>
    </tr>
  </thead>
  <tbody class="TableBody">
    <tr class="TableRow">
      <td class="TableCell">Shirt</td>
      <td class="TableCell">$10</td>
    </tr>
    <tr class="TableRow">
      <td class="TableCell">Pants</td>
      <td class="TableCell">$20</td>
    </tr>
  </tbody>
</table>
```

## CSS feito na sintaxe do SASS:

---

```
.Table {
  padding: 5px;

  .TableCell {
    min-width: 100px;
  }

  .TableBody {
    .TableRow {
      .TableCell {
        border-bottom: 1px solid silver;
      }
    }
  }
}
```

## Resultado do SASS anterior:

---

```
.Table {  
  padding: 5px;  
}  
.Table .TableCell {  
  min-width: 100px;  
}  
.Table .TableBody .TableRow .TableCell {  
  border-bottom: 1px solid silver;  
}
```



Então, precisamos  
**reduzir nossos estilos!**

# Pós- processamento de CSS:

- **PurgeCSS**  
Remoção de código não utilizado.
- **Stylelint**  
Análise estática de código.  
Indica potenciais problemas.
- **PostCSS**  
Um mundo de transformações e  
plugins pra estender seu CSS.

**Mas então...**

Metodologias, pré & pós processadores  
**resolveram os 3 problemas, certo?**

Sim,  
**mas e na prática?**

## CSS no Vue:

---

```
<template>
  <div class="hello">
    <h2>Essentials</h2>
    <ul>
      <li>Core Docs</li>
      <li class="highlighted">Forum</li>
    </ul>
  </div>
</template>

<style scoped>
  h1, h2 { font-weight: normal }
  ul { list-style-type: none; padding: 0 }
  a { color: #42b983 }
  .highlighted { font-weight: bold }
</style>
```

## CSS Inline no React:

---

```
const style = {  
  borderRadius: "3px",  
  WebkitBackdropFilter: "blur(5px)",  
  backgroundColor: "rgba(255, 255, 255, 0.85)"  
};
```

```
const Card = () => (  
  <div className="Card" style={style}>  
    Conteúdo  
  </div>  
);
```

Trazendo pro lado do React,  
**o que é mais popular?**

## Exemplo do JSS:

---

```
const styles = {
  button: {
    "&:hover": {
      backgroundColor: "blue"
    }
  },
  ctaButton: {
    extend: "button",
    "&:hover": {
      border: "1px solid white"
    }
  }
};

const { classes } = jss.createStyleSheet(styles).attach();

document.body.innerHTML = `
  <button class="${classes.button}">Button</button>
  <button class="${classes.ctaButton}">CTA Button</button>
`;
```



## Resultado gerado pelo CSS Modules:

---

```
<div class="Grid_Grid__3Y92C">
  <div class="Grid_GridRow__2KjFd">
    <div class="Grid_GridColumn__2Du6a"></div>
    <div class="Grid_GridColumn__2Du6a"></div>
    <div class="Grid_GridColumn__2Du6a"></div>
  </div>
  <div class="Grid_GridRow__2KjFd">
    <div class="Grid_GridColumn__2Du6a"></div>
    <div class="Grid_GridColumn__2Du6a"></div>
    <div class="Grid_GridColumn__2Du6a"></div>
  </div>
  <div class="Grid_GridRow__2KjFd">
    <div class="Grid_GridColumn__2Du6a"></div>
    <div class="Grid_GridColumn__2Du6a"></div>
    <div class="Grid_GridColumn__2Du6a"></div>
  </div>
</div>
```



# styletron

[github.com/stereobooster/css-in-js-101](https://github.com/stereobooster/css-in-js-101)



 styletron



 linaria



# fela

CSS  
MODULES



## CSS in JS:

---

```
import styled from "styled-components";

const Button = styled.button`
  background-color: blue;
  border-radius: ${props => (props.rounded ? "9999px" : "3px")};
  border: 1px solid white;
  font-size: 16px;
  color: white;
`;

export default Button;
```

## O que o CSS Funcional resolve:

---

```
.button {  
  color: #19223c;  
  padding: 5px 10px;  
  border: 1px solid #19223c;  
}
```

```
.input {  
  color: #19223c;  
  border: none;  
  padding: 5px 10px;  
  background-color: white;  
}
```

Agora abra sua mente,  
**essa proposta é uma  
quebra de paradigmas.**

## A proposta do CSS Funcional:

---

```
<div class="pa2 br2 ba b--gray-100 dark-900">  
  <label class="dark-900">E-mail</label>  
  <input type="email" class="pa2 br1 ba b--gray-50 dark-900" />  
</div>
```

Mas nenhuma é  
**melhor que a outra.**

## E podem ser combinadas:

---

```
const base = "helvetica f6 br-pill mh1 ph3 pv1 mb2 ";

const variants = {
  primary: "bn bg-blue white pointer",
  outline: "ba bg-white blue pointer b--blue",
  disabled: "bn bg-black-20 black-40"
};

const Button = styled.button.attrs(({ variant, ...rest }) => ({
  className: base + variants[variant],
  disabled: variant === "disabled",
  ...rest
}));
```



# Concluindo, qual foi o ganho?

- Atacamos 3 dos maiores problemas em qualquer aplicação;
- Conhecemos a maioria das soluções relevantes no mercado pra CSS;
- Descobrimos alguns prós, contras e pontos de atenção na hora de aplicar.

# Links:

## Metodologias:

- [SMACSS](#)
- [OOCSS](#)
- [BEM](#)

## Pós-processadores:

- [PurgeCSS](#)
- [Stylelint](#)
- [PostCSS](#)

## Pré-processadores:

- [SASS](#)
- [LESS](#)
- [Stylus](#)

## Bibliotecas:

- [CSS Modules](#)
- [CSS in JS 101](#)
- [CSS Funcional](#)

dito

# Obrigado!

 [in.uriell.dev](https://in.uriell.dev)

 [github.uriell.dev](https://github.uriell.dev)

 [modern-css.uriell.dev](https://modern-css.uriell.dev)