

Empacotando módulos de forma simples com Rollup

André Lins



Índice

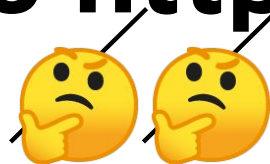
- Empacotadores de módulo;
- Conceitos importantes desse mundo;
- Rollup e mais rollup;
- Configurações;

**Imagine que você está
construindo um aplicativo
web**



```
<html>
  <script src="/src/foo.js"></script>
  <script src="/src/bar.js"></script>
  <script src="/src/baz.js"></script>
  <script src="/src/qux.js"></script>
  <script src="/src/quux.js"></script>
</html>
```

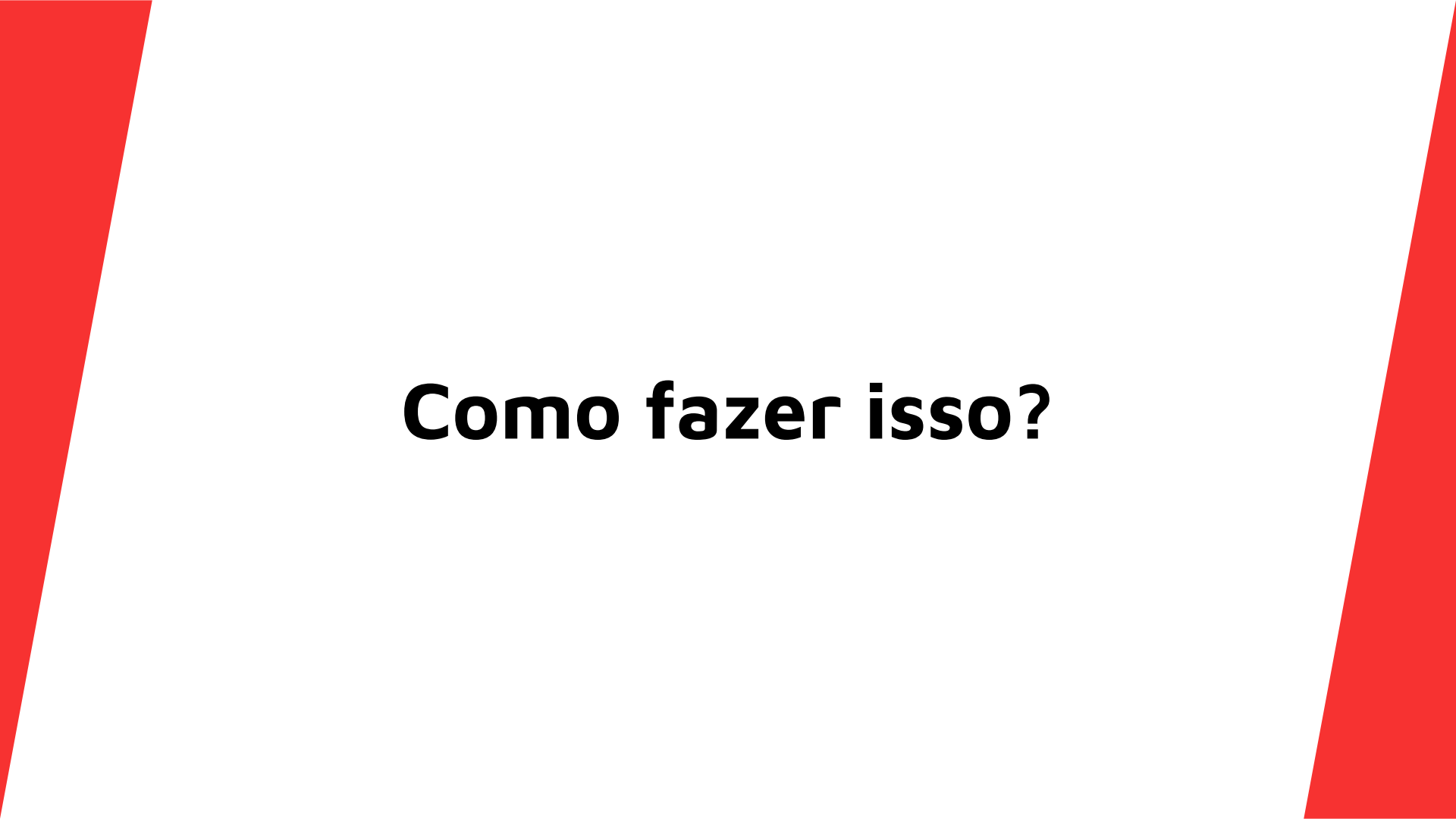
**Cada arquivo requer uma
solicitação http diferente**



**Seria melhor se pudéssemos
ter todos esses scripts em
um único arquivo**



```
<html>  
  <script src="/dist/bundle.js"></script>  
</html>
```



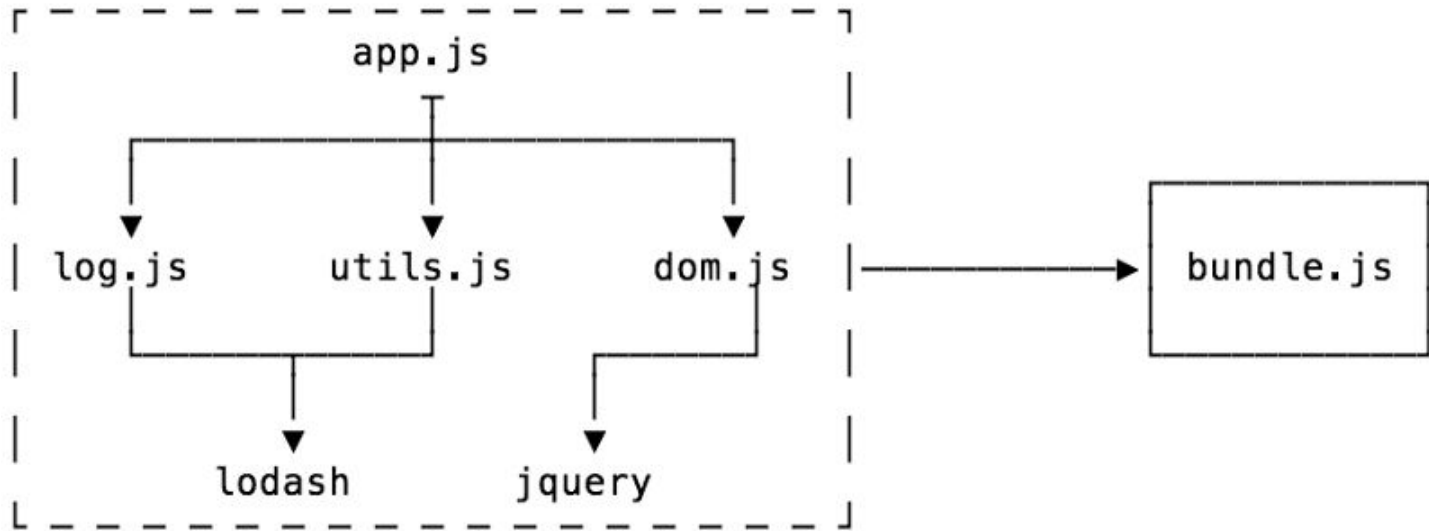
Como fazer isso?

Empacotadores de Módulo

O que é um empacotador de módulo?

É uma ferramenta que pega pedaços de JavaScript e suas dependências e os empacota em um único arquivo, geralmente para uso no navegador.

[Adam Kelly]

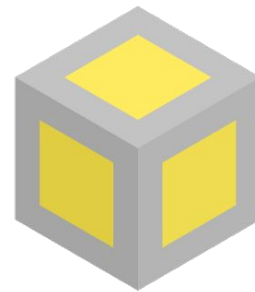


Por que é necessário usar?

- O navegador não suporta o sistema de módulos, embora isso não seja totalmente verdade hoje em dia;
- É interessante e mais rápido ter um arquivo único com todo seu código javascript;
- Ele ajuda você a gerenciar o relacionamento de dependência do seu código, carregando os módulos em ordem de dependência para você;
- Ajuda a carregar seus *assets* em ordem de dependência, *asset* de imagem, *asset* css, etc;



webpack



rollup.js



PARCEL

Blazing fast, zero configuration web application bundler



FUSEBOX

JAVASCRIPT TOOLS

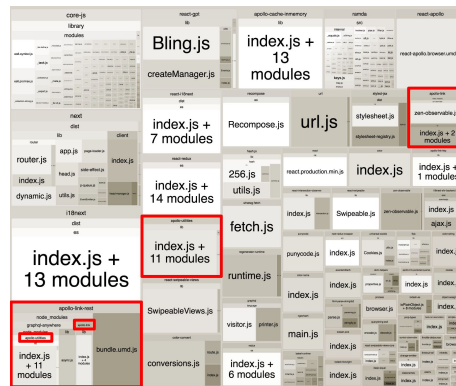
Conceitos importantes

Module Format

- **CommonJS(CJS):**
 - Definido pelo RequireJs;
 - Feito para nodejs;
 - Não funcionando no navegador;
- **AMD:**
 - Leitura de forma assíncrona;
 - Feito para frontend;
- **UMD:**
 - Definição universal de módulo;
 - Funciona para frontend e backend;
 - Serve mais como um padrão para configurar sistemas de módulo distintos;
- **ES Modules:**
 - Proposta do javascript para criar módulos padrão;
 - Melhor do mundo CJS e AMD;
 - Funciona para frontend e backend;

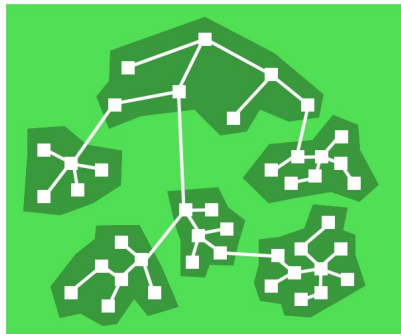
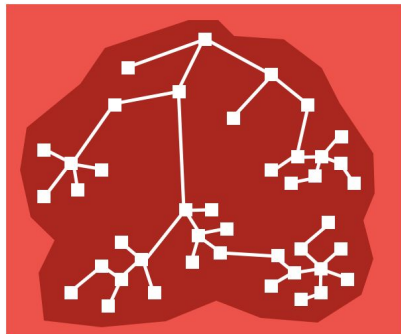
Tree-shaking

- Elimina código não utilizado em módulos importados.
- Possível a partir da criação do ES Modules.
- Bundle menor e refletindo o que de fato é utilizado.
- Práticas como **lodash.debounce** não são mais necessárias.



Code splitting

- Divide seu código em vários pacotes menores que podem ser carregados sob demanda ou em paralelo.
- Tamanho do bundle principal reduzido
- Maior velocidade do navegador para carregar pacotes.





**Agora vamos ao assunto
desta apresentação...**





rollup.js



Next-generation ES module bundler



Rollup

- Criado em 2015;
- Autor: Rich Harris (criador do svelte);
- Seu pensamento inicial era construir bibliotecas javascript de forma mais eficiente utilizando o ES2015;
- Versão atual: 2.*;



18,8K



245



4,332



907

Vantagens do rollup

- Formato ES6 nativo;
- Simplicidade nas configurações;
- Pacote menor em comparação a outras ferramentas;
- Tree-shaking;
- Code splitting;
- Simplicidade na criação de bibliotecas que potencialize a tecnologia (Conceito de plugins);
- Perfeito para criar bundle de bibliotecas;

**Como fica o arquivo de
configuração?**

Configurando...

- 1) Instale o rollup;
- 2) Crie o arquivo **rollup.config.js**;
- 3) Digite a seguinte configuração e se apaixone ❤️❤️;

```
import commonjs from "rollup-plugin-commonjs";

export default {
  input: "src/main.js",
  output: {
    file: "dist/bundle.js"
  },
  plugins: [
    commonjs()
  ]
};
```


Configurando ambiente Vue

```
import resolve from 'rollup-plugin-node-resolve';
import commonjs from 'rollup-plugin-commonjs';
import { terser } from 'rollup-plugin-terser';
import vue from 'rollup-plugin-vue';
import css from 'rollup-plugin-css-only';
import autoprefixer from 'autoprefixer';

export default {
  input: 'src/main.js',
  output: { name: 'app', file: 'public/build/bundle.js' },
  plugins: [
    vue({ css: false, style: { postcssPlugins: [autoprefixer] } }),
    css({ entry: 'src/main.js', dest: 'public/build/bundle.css' }),
    resolve({ mainFields: ['module', 'jsnext', 'main', 'browser'] }),
    commonjs(),
    terser()
  ]
};
```

Configurando ambiente Svelte

```
import svelte from "rollup-plugin-svelte";
import resolve from "rollup-plugin-node-resolve";
import commonjs from "rollup-plugin-commonjs";
import { terser } from "rollup-plugin-terser";

export default [
  {
    input: "src/dev/main.js",
    output: { format: "iife", name: "app", file: "public/build/bundle.js" },
    plugins: [
      svelte({
        dev: false,
        css: css => {
          css.write("public/build/bundle.css");
        }
      }),
      resolve({
        browser: true,
        dedupe: importee =>
          importee === "svelte" || importee.startsWith("svelte/")
      }),
      commonjs(),
      terser()
    ],
    watch: { clearScreen: false }
  }
];
```



Talk is cheap. Show me the code.

(Linus Torvalds)

izquotes.com

Rollup x Webpack

Comparando Rollup com Webpack

Rollup

- 5 anos
- ES Modules
- Plugins
- Único padrão de configuração
- Destaque: Tree shaking

Webpack

- 8 anos
- CommonJS
- Loader, Resolver e Plugins
- Diversos padrões de configuração
- Destaque: Module Federation

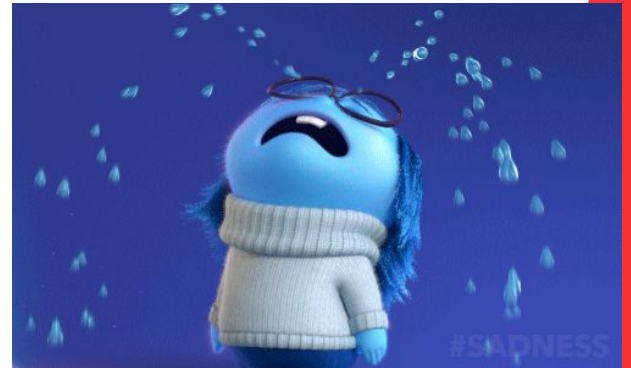


TOOL	DEV BUILD TIME ONE	DEV BUILD TIME TWO	DEV BUILD TIME THREE	DEV BUILD TIME AVG
PARCEL	14.92 S	5.22 S	4.36 S	8.16 AVG S
ROLLUP	0.570 S	0.482 S	0.487 S	0.513 AVG S
WEBPACK	3.608 S	3.328 S	3.844 S	3.59 AVG S

TOOL	PROD BUILD TIME ONE	PROD BUILD TIME TWO	PROD BUILD TIME THREE	PROD BUILD TIME AVG
PARCEL	738.509 S	35.364 S	35.592 S	269.82 AVG S
ROLLUP	0.712 S	0.665 S	0.714 S	0.697 AVG S
WEBPACK	3.636 S	3.805 S	4.305 S	3.915 AVG S

Análise retirada de um artigo de Adam Gerard da X-team

Porém nem tudo são flores...



Desvantagens do Rollup

- Dificuldade de entender dependências circulares;
- Quantidade reduzida de plugins em comparação ao webpack;
- A configuração pode ficar complexa com um número de entradas e saídas alto;

Quer saber mais sobre esse conteúdo?

- <https://rollupjs.org/>
- <https://medium.com/@ajmeyghani/javascript-bundlers-a-comparison-e63f01f2a364>
- <https://lihautan.com/what-is-module-bundler-and-how-does-it-work/>
- <https://blog.codecasts.com.br/ecossistema-javascript-parte-05-bundlers-builders-6809b17ddcf8>
- <https://blog.codecasts.com.br/ecossistema-javascript-parte-05-bundlers-builders-6809b17ddcf8>
- <https://x-team.com/blog/rollup-webpack-parcel-comparison/>



ANDRÉ LINS

- Engenheiro FrontEnd Sênior na PagSeguro
- Graduado em Ciência da Computação pela UFRPE
- Pós-Graduando em Engenharia de Software pela PUC Minas
- Viciado em OpenSource
- Desenvolvedor de algumas bibliotecas para Svelte, React, Vue, Node, ReasonML, Rust entre outras tecnologias.
- Tentando não ser evangelista Javascript

<https://twitter.com/andremlins>

<https://github.com/andremlins>