



UNIVERSIDADE FEDERAL DO CEARÁ

Code smells em frameworks front-end

Disciplina: Qualidade de Software

Equipe: Miquéias Bento da Silva e Daniel Christopher Souza Rodrigues

Projeto: [Official components for Angular](#)

Fork: [components-code-smells-refactoring](#)

Responsável: Miquéias Bento

Eu estou atualmente trabalhando na refatoração do seguinte code smell:

Overusing-Any-Type, nos arquivos:

../docs/scenes/src/app/scene-viewer/scene-viewer.ts

../docs/scenes/src/app/scenes/slider/slider-scene.ts

../docs/src/app/material-docs-app.ts

../docs/src/app/pages/component-page-header/component-page-header.spec.ts

../docs/src/app/pages/component-sidenav/[component-sidenav.spec.ts](#)

../docs/src/app/pages/component-viewer/component-viewer.ts

../docs/src/app/pages/guide-list/guide-list.spec.ts

../docs/src/app/shared/analytics/analytics.ts

../docs/src/app/shared/carousel/carousel.spec.ts

../docs/src/app/shared/carousel/[carousel.ts](#)

./src/cdk/drag-drop/directives/test-utils.spec.ts

./src/cdk/drag-drop/drag-events.ts

./src/cdk/drag-drop/[drag-utils.ts](#)

[13 arquivos refatorados para completar os 40 code smells, mas ainda os 3 de DOM]

Minhas principais dificuldades na remoção do code smell são:

A grande recorrência do code smells em um único arquivo.

Exigiu o entendimento do domínio do dado.

Força leitura de:

- interfaces existentes
- tipos genéricos
- contratos internos da biblioteca

Precisou da criação de alguns novos tipos.

Eu estou usando os seguintes métodos de refatoração para remover o code smell:

A refatoração desse smell envolveu substituir any por tipos específicos ou interfaces mais apropriadas, melhorando a legibilidade, a robustez do código, além de aproveitar os benefícios de tipagem do TypeScript.

Responsável: Daniel Christopher

Eu estou atualmente trabalhando na refatoração do seguinte code smell:



UNIVERSIDADE FEDERAL DO CEARÁ

Large File, nos arquivos:

```
../docs/src/app/shared/doc-viewer/doc-viewer.ts
../docs/src/app/shared/documentation-items/documentation-items.ts
../docs/src/app/shared/example-viewer/example-viewer.ts
../src/material/sidenav/drawer.ts
../src/material/radio/radio.ts
../src/material/menu/menu.ts
../src/aria/private/menu/menu.ts
../src/aria/private/tabs/tabs.ts
../src/aria/private/tree/tree.ts
../src/material/progress-bar/progress-bar.ts
```

Minhas principais dificuldades na remoção do code smell são:

Arquivos muito grandes, os quais exigiam um bom conhecimento do seu funcionamento. Além de não estar acostumado com Angular, precisei entender arquivos com mais de 700 linhas de código.

Eu estou usando os seguintes métodos de refatoração para remover o code smell:

Apliquei o Extract Class, extraíndo as diversas classes que ficavam no mesmo arquivo. Além disso, separei métodos reutilizáveis de uma classe grande para deixá-la mais coesa. Também apliquei o Module Decomposition que consiste em separar o arquivo em camadas que destaquem a sua responsabilidade melhorando, assim, a sua manutenção.

Responsável: Daniel Christopher

Eu estou atualmente trabalhando na refatoração do seguinte code smell:

Large File, nos arquivos:

```
../docs/src/app/shared/doc-viewer/doc-viewer.ts
../docs/src/app/shared/documentation-items/documentation-items.ts
../docs/src/app/shared/example-viewer/example-viewer.ts
../src/material/sidenav/drawer.ts
../src/material/radio/radio.ts
../src/material/menu/menu.ts
../src/aria/private/menu/menu.ts
../src/aria/private/tabs/tabs.ts
../src/aria/private/tree/tree.ts
../src/material/progress-bar/progress-bar.ts
```

Minhas principais dificuldades na remoção do code smell são:

Componentes muito grandes, os quais exigiam um bom conhecimento do seu funcionamento e costume com angular. Além de terem outros code smells no meio, precisei ter um bom conhecimento de cada componente para dividir as suas responsabilidades.



UNIVERSIDADE FEDERAL DO CEARÁ

Eu estou usando os seguintes métodos de refatoração para remover o code smell:

Aplicando o ***composition pattern*** nos componentes, que consiste em dividir um componente grande em componentes menores que o compõem, dividindo a lógica em vários componentes ao invés de concentrá-la em um só.

Também dividi o componente em vários arquivos de forma que não gere o code smell "Large File".

Responsável: Miquéias Bento

Eu estou atualmente trabalhando na refatoração do seguinte code smell:

Too Many Inputs, nos arquivos:

./src/material/checkbox/checkbox.ts

./src/material/datepicker/calendar-body.ts

./src/material/select/select.ts

./src/material/slide-toggle/slide-toggle.ts

./src/material/stepper/[step-header.ts](#)

./src/material/tabs/tab-group.ts

./src/youtube-player/[youtube-player.ts](#)

Minhas principais dificuldades na remoção do code smell são:

O principal desafio foi garantir que o comportamento original fosse mantido, já que algumas propriedades acionavam lógicas internas quando eram alteradas. Por isso, foi necessário mapear com mais cuidado esses valores e preservar as ações já existentes.

Eu estou usando os seguintes métodos de refatoração para remover o code smell:

A refatoração consistiu na análise das propriedades decoradas com `@Input()` e na identificação de dados conceitualmente relacionados. Esses dados foram agrupados em objetos de configuração tipados, reduzindo a quantidade de inputs expostos pelo componente e tornando sua interface pública mais coesa.

Quando necessário, propriedades com comportamentos específicos ou que participavam de comunicação em duas direções foram mantidas separadamente para preservar a funcionalidade. As alterações foram realizadas de forma localizada, garantindo a manutenção do comportamento original e a eliminação do code smell identificado pela ferramenta.

Responsável: Miquéias Bento



UNIVERSIDADE FEDERAL DO CEARÁ

Eu estou atualmente trabalhando na refatoração do seguinte code smell:

Direct DOM Manipulation, nos arquivos:

../docs/scenes/src/app/scene-viewer/scene-viewer.ts

../docs/src/app/material-docs-app.ts

../docs/src/app/shared/analytics/analytics.ts

Minhas principais dificuldades na remoção do code smell são:

A principal dificuldade foi o forte acoplamento com o DOM, tendo que modificar trechos de outros arquivos HTML, e a recorrência em vários pontos do mesmo arquivo.

Eu estou usando os seguintes métodos de refatoração para remover o code smell:

A refatoração buscou substituir essas manipulações por data binding, controle de estado e APIs recomendadas pelo próprio framework, como:

- data binding ([class], [style], ngIf, ngFor);
- @ViewChild quando necessário;
- Renderer2 para casos mais específicos;