



UNIVERSIDADE FEDERAL DO CEARÁ

Code smells em frameworks front-end

Disciplina: Qualidade de Software

Equipe: Paulo Victor, Tiago Tito

Projeto: Ngx-Chart

Fork: <https://github.com/VictorFCos/ngx-charts---Trabalho-Qualidade-de-software>

Eu estou atualmente trabalhando na refatoração do seguinte code smell: Overusing Any Type

Minhas principais dificuldades na remoção do code smell são: A principal dificuldade foi identificar o tipo exato que deveria substituir o any. Em alguns casos, como nos formatadores (tickFormatting, valueFormatting), foi necessário deduzir a assinatura da função baseada em como ela é usada no template, ou em outros lugares do código.

Eu estou usando os seguintes métodos de refatoração para remover o code smell: Realizei a especificação de tipos substituindo o tipo genérico any por tipos primitivos como number e string ou interfaces específicas como DataItem onde o dado era conhecido, além de ter implementado a definição de assinaturas de função para propriedades que aceitam funções, como a xAxisTickFormatting, definindo a assinatura completa em vez de manter o any para melhorar a segurança de tipos e a documentação, e também fiz a refatoração de objetos complexos como no caso do data: { children: any }, passando a estruturar o objeto esperado em vez de aceitar qualquer valor, o que serve como um passo inicial para a criação de uma interface no futuro.

Eu estou atualmente trabalhando na refatoração do seguinte code smell: Large Files

Minhas principais dificuldades na remoção do code smell são: Ao transpor manualmente o código html das strings inline para arquivos separados acabei incorrendo em erros de atenção como deixar a propriedade template juntamente com a templateUrl ou omitir vírgulas o que quebrou a compilação de formas difíceis de diagnosticar e gerou mensagens de erro obscuras do angular como o ng8001 sugerindo problemas de módulo quando na verdade a causa raiz eram falhas de sintaxe no typescript o que tornou o processo de depuração frustrante pois tratava-se apenas de pontuação ou propriedades duplicadas ademais garantir a integridade do módulo verificando metadados como styleUrls e dependências foi desafiador visto que qualquer inconsistência provocava erros em cascata

Eu estou usando os seguintes métodos de refatoração para remover o code smell: Em relação à extração de template estou movendo o conteúdo html que estava hardcoded na



UNIVERSIDADE FEDERAL DO CEARÁ

propriedade template do decorator component para arquivos html externos dedicados o que reduz drasticamente o tamanho dos arquivos typescript e separa a lógica da apresentação concomitantemente ao isolar o template html estou aplicando o princípio de separação de responsabilidades facilitando a leitura e manutenção tanto da lógica do componente via typescript quanto da sua estrutura visual em html

Eu estou atualmente trabalhando na refatoração do seguinte code smell: Too Many Inputs

Minhas principais dificuldades na remoção do code smell são: Foi uma refatoração bem tediosa, alguns componentes tinham mais de 20 inputs que tinham que ser mapeadas para uma interface, também tive que achar todos os usos desse componente para atualizar para o uso da interface. Além disso eu não era muito familiarizado com o Angular

Eu estou usando os seguintes métodos de refatoração para remover o code smell: Na maioria dos casos foi só implementar uma interface “Config” contendo tudo que era feito com “@Input()” antes e atualizar o “@Input()” para usar essa interface. Então fiz a verificação se o smell foi removido, e se o código ainda compilava e passava nos testes

Eu estou atualmente trabalhando na refatoração do seguinte code smell: Large Components

Minhas principais dificuldades na remoção do code smell são: Decidir o que deveria ser movido para o helper, Alguns componente muito grandes seriam muito difíceis de refatorar, então tive que procurar os smells mais viáveis de se refatorar

Eu estou usando os seguintes métodos de refatoração para remover o code smell: Busquei componentes que realizavam cálculos complexos ou muitas formatações de dados e extrai essas complexidades para uma função helper. Então fiz a verificação se o smell foi removido, e se o código ainda compilava e passava nos testes