



UNIVERSIDADE FEDERAL DO CEARÁ

Code smells em frameworks front-end

Disciplina: Qualidade de Software

Equipe: Douglas Eduardo e Júlio Moraes

Projeto: <https://github.com/mui/material-ui>

Fork: <https://github.com/douglasessousa/material-ui>

Eu estou atualmente trabalhando na refatoração do seguinte code smell:
NNA (Non-Null Assertion), que ocorre quando o operador '!' é utilizado para forçar o TypeScript a tratar valores possivelmente null ou undefined como sempre existentes. Esse smell aparece em diferentes partes do projeto, principalmente no acesso a propriedades de objetos e estados que não possuem garantia explícita de inicialização.

Minhas principais dificuldades na remoção do code smell são:
Garantir que os valores envolvidos realmente estejam definidos em todos os cenários de uso. Além disso, em alguns casos, essas propriedades dependem de props opcionais ou de comportamentos dinâmicos do componente, o que exige um cuidado maior para evitar erros em tempo de execução.

Eu estou usando os seguintes métodos de refatoração para remover o code smell:
Estou substituindo o operador ! por optional chaining (?.), verificações condicionais e operadores de coalescência nula (??), garantindo que o acesso às propriedades só ocorra quando os valores existirem ou que valores padrão sejam utilizados.

Eu estou atualmente trabalhando na refatoração do seguinte code smell:
Uso do tipo **any** que remove toda a type safety do TypeScript, permitindo qualquer valor sem verificação.

Minhas principais dificuldades na remoção do code smell são:
Garantir tipos adequados para props complexas, e lidar com cenários dinâmicos onde tipos são desconhecidos

Eu estou usando os seguintes métodos de refatoração para remover o code smell:
Type Assertion Específica (as TipoConcreto em vez de as any); Generic Constraints para manter flexibilidade com type safety; Union Types para valores com possibilidades conhecidas; Interface/Type Definitions para estruturas complexas; Satisfies Operator para validação em tempo de desenvolvimento

Eu estou atualmente trabalhando na refatoração do seguinte code smell:
Overly Flexible Props (OFP): Uso de tipagens frouxas e excessivamente genéricas (ex: [key: string]: any;, object) nas propriedades dos componentes.

Minhas principais dificuldades na remoção do code smell são:



UNIVERSIDADE FEDERAL DO CEARÁ

Garantir que ao remover o `any` ou `[key: string]: any;`, o componente ainda aceitasse todas as props HTML padrão sem quebrar a funcionalidade. Também definir interfaces explícitas para objetos complexos que estavam passando despercebidos como `object` ou `any` em componentes de contexto

Eu estou usando os seguintes métodos de refatoração para remover o code smell:
Estender tipos conhecidos do React e MUI (`ComponentPropsWithoutRef`, `TabsOwnProps`, `HTMLAttributes`) para obter tipagem completa e segura, eliminando a necessidade de index signatures genéricos. Também estou criando interfaces explícitas para tipar props de objetos ou contextos que antes usavam `any` ou `object`

Eu estou atualmente trabalhando na refatoração do seguinte code smell:
Missing Union Type Abstraction (MUT): Tipos de união de strings literais

Minhas principais dificuldades na remoção do code smell são:
O código ser muito extenso dificultou a localização de todas as ocorrências do tipo de união/intersecção para garantir sua completa abstração. Tipar corretamente genéricos repetitivos em arquivos de especificação de tipos (specs), exigindo a criação de tipos genéricos auxiliares.

Eu estou usando os seguintes métodos de refatoração para remover o code smell:
Criar um type nomeado para a união de strings literais ou intersecções complexas