Padrão
de Refatoração

Francisco Rafael - francisco.rafaell.pereira@gmail.com - CPF: 068.778.653-32

Padrões de Refatoração

- Refatoração é o ato de melhorar o projeto sem alterar seu comportamento
- Como desenvolvedores, se existe algo que não podemos fugir é de **refatorações**
- A realidade é que na indústria não se fala muito disso: quando encontramos código ruim, somos ágeis em colocar a culpa no desenvolvedor que passou anteriormente noprojeto
- Mas será que estamos fazendo algo para melhorar isso, ou somente aumentando o problema, sendo irresponsáveis?

Padrões de Refatoração

- Por isso, é essencial entender melhor sobre as mecânicas de alteração de software
- Segundo o livro "Trabalho Eficaz com Código Legado", existem 4 razões para se alterar um software:
  - Inclusão de uma funcionalidade
  - Correção de um bug
  - Melhoria do projeto (se refere a estrutura)
  - Otimização de uso de recursos (processamento ou memória, por exemplo)

Padrões de Refatoração

- Código legado é algo que a grande maioria dos desenvolvedores temem. Mas como alterá-los com segurança?
- A resposta está em conhecer PADRÕES de refatoração de código
- Tentar alterar códigos críticos sem conhecer boas práticas e padrões vai levar a um código de difícil manutenção, e baixa legibilidade, resultando em maiores custos de manutenção e maiores taxas de bugs

Padrões de Refatoração

- Para aplicar o padrão Repository, precisamos primeiro identificar onde estão os pontos de alteração, ou seja, onde está o código de acesso a dados
- Em seguida, precisamos eliminar as dependências ao acesso a dados. Isso é feito exatamente se utilizando do padrão Repository
- Para isso, podemos utilizar o seguinte fluxo:
  - **Extrair Método:** migrar o código de acesso para um método focado nisso
  - **Extrair Classe:** migrar o código do método extraído para uma classe dedicada ao acesso a dados
  - **Extrair Interface:** a partir dessa classe, criar uma interface representando o contrato do repositório
  - **Substituição:** passar a interface via injeção de dependência nas classes que necessitam do acesso a dados

Até a próxima aula!

Francisco Rafael - francisco.rafaell.pereira@gmail.com - CPF: 068.778.653-32

Padrão Repository

Francisco Rafael - francisco.rafaell.pereira@gmail.com - CPF: 068.778.653-32

Padrão Repository

- Padrão utilizado para abstrair o acesso a dados, permitindo a abstração dos detalhes de implementação
  - Ao se utilizar também uma interface, é também realizado o desacoplamento
- Componente de domínio do Domain-Driven Design, responsável pelo acesso aos objetos armazenados, mas sem especificar a respeito de sua fonte

Padrão Repository

- Com o seu uso, classes da camada Application como serviços de aplicação ou commands/queries não precisam mais ter dependência direta em bibliotecas de acesso a dados como Entity Framework Core e Dapper
- Na Arquitetura Limpa, é composto por:
  - **Interface:** contém o contrato dos métodos do repositório, ficando localizada na camada Core
  - **Implementação:** contém a implementação dos métodos definidos na interface, podendo utilizar Entity Framework Core, Dapper, ADO.NET, ou outras ferramentas de acesso a dados

Até a próxima aula!

Francisco Rafael - francisco.rafaell.pereira@gmail.com - CPF: 068.778.653-32