

Engenharia Disciplinada de Softwares

Projeto de Bloco – TP 9

Professor Armenio Cardoso

Vinicius Dornelas Braz
18/09/2022

1. *Expansão do protótipo funcional do sistema:*

R. Link para repositório: <https://github.com/braz-vinicius/venturahr>.

Demonstração dos casos de uso em gif animado disponíveis na pasta: <https://github.com/braz-vinicius/venturahr/raw/master/docs/demos/>

2. Aponte alguma refatoração no código para utilização de 2 padrões de design. Aponte o porquê realizar refatoração em projetos ágeis.

R. A solução foi refatorada para utilizar os seguintes padrões de design:

- **Repository pattern:** Inicialmente o software utilizava acesso ao banco por meio das classes do Entity Framework. A solução foi refatorada para utilizar o padrão Repository, abstraindo as complexidades do banco e diminuindo o acoplamento da solução com o Entity Framework e com o próprio banco.
- **Chain of Responsibility:** Inicialmente as exceções geradas pelos controllers da API eram tratadas individualmente e manualmente. Com a utilização do padrão Chain of Responsibility foi possível criar uma classe de middleware (*ErrorHandlerMiddleware*) que permitiu um lugar único e centralizado para seu tratamento evitando código duplicados, aumentando a manutenibilidade da solução.

O processo de refatoração é essencial ao desenvolvimento ágil, pois permite a evolução técnica do software de forma iterativa e sustentável, ao mesmo tempo em que as funcionalidades já entregues não são comprometidas.

3. Aponte todos os padrões de projeto utilizados na implementação e destaque as responsabilidades

R. Podemos destacar a utilização dos seguintes padrões:

- **Layered (Padrão arquitetural):** O padrão Layered foi utilizado para garantir a separação das diversas camadas da solução (Api, Application, Domain e Infrastructure), permitindo uma maior coesão, menor acoplamento e maior reusabilidade do software desenvolvido.
- **Model-View-Controller (Padrão arquitetural):** O padrão MVC foi utilizado na camada da API, permitindo uma maior separação de responsabilidades e desacoplamento do software.
- **Repository (Padrão de projeto):** O padrão Repository foi utilizado na camada de acesso a dados da aplicação, fornecendo uma API agnóstica de acesso, facilitando a reusabilidade da solução e abstraindo as complexidades do banco e do Entity Framework.
- **Facade (Padrão de projeto):** O padrão Facade foi utilizado na classe JwtUtils para lidar com os cenários de autenticação via JWT, mascarando as complexidades de geração e validação de tokens pelas classes JwtSecurityHandler e JwtSecurityToken providas pelo framework ASP.NET Core.

4. Justifique a importância dos testes unitários realizados até aqui.

R. Os testes unitários foram importantes para garantir a consistência do software desenvolvido ao longo das sucessivas iterações, evitando que novas implementações afetassem o comportamento de outras funcionalidades previamente construídas, além de garantir o correto funcionamento das classes e dos métodos desenvolvidos.

5. Identifique possíveis pontos de evolução e variação no projeto.

R.

- Mudança do esquema de autenticação/autorização para OAuth.
- Evolução do SPA para suportar clientes móveis (Android e iOS).
- Utilização de pipelines de CI (Continuous Integration) para automatizar a execução dos testes e builds.
- Utilização de pipelines de CD (Continuous Delivery) para automatizar a distribuição de releases.

6. Descreva brevemente o histórico da evolução do projeto destacando a composição dos documentos, a implementação do código e mostrando se foi aderente as metodologias ágeis propostas.

R. O projeto se baseou na metodologia *Disciplined Agile Delivery* (DAD), com a utilização do ciclo de vida Ágil. Conforme o DAD o projeto foi fatiado conforme as seguintes fases:

- **Inception:** Nesta etapa foram identificados os objetivos do projeto, assim como a visão de requisitos, plano de projeto, diagramas de modelagem do sistema e análise da viabilidade técnica e de recursos.
- **Construction:** Nessa etapa todas as funcionalidades necessárias foram construídas, dividida em várias iterações de tempo delimitado. No final de cada iteração (no caso concreto, ao final de cada TP), um incremento demonstrável da solução potencialmente consumível foi produzido e disponibilizado no repositório Git. Além disso foi disponibilizada a demonstração dos casos de uso por meio de GIFs animados na pasta **docs/demos** do repositório.
- **Transition:** A etapa de transição concentra-se na entrega e demonstração do sistema em produção. Do ponto de vista ágil, a fase de transição termina quando os stakeholders realizam o sign-off e o sistema é totalmente implantado. No caso concreto deste trabalho a transição ocorrerá após a apresentação do *Projeto de Bloco* e obtenção do devido sign-off pelo Professor Armenio Cardoso, na condição de stakeholder máximo do projeto.

a. Relembre o ciclo de vida que foi utilizado por seu grupo, com a justificativa da escolha e compare com a execução.

R. Considerando as peculiaridades do projeto, foi escolhido o ciclo de vida *Agile*, tendo em vista a existência de um direcionamento dos que os usuários necessitam, o fato do time de desenvolvimento possuir um mindset ágil e os requisitos funcionais apresentaram baixa e/ou média complexidade, conforme o fluxograma de escolha de ciclo de vida sugerido pelo PMI (Project Management Institute), disponível abaixo:

