

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
Bacharelado em Engenharia de Software

IHELPU

Matheus Santos Rosa Carneiro

Rafael Soares Botazini

Raíssa Carolina Vilela da Silva

Vitor Augusto Alves de Jesus

Histórico de Revisões			
Data	Autor	Descrição	Versão
21/05/2020	Raíssa Vilela	Desenvolvimento dos aspectos presentes na seção 1.	1.0
21/05/2020	Matheus Santos	Desenvolvimento dos aspectos presentes na seção 1.	1.1

SUMÁRIO

1	Apresentação	3
1.1	Problema	3
1.2	Objetivos do trabalho.....	4
1.3	Definições e Abreviações	4
2	Requisitos	5
2.1	Requisitos Funcionais	5
2.2	Requisitos Não-Funcionais.....	5
2.3	Restrições Arquiteturais	5
2.4	Mecanismos Arquiteturais	6
3	Modelagem e projeto arquitetural	6
3.1	Visão de Casos de Uso	7
3.2	Visão Lógica	8
3.3	Visão Física.....	Erro! Indicador não definido.
3.4	Modelo de dados (opcional)	9
4	Prova de conceito / protótipo arquitetural.....	Erro! Indicador não definido.
4.1	Implementação e implantação.....	Erro! Indicador não definido.
4.2	Interfaces.....	Erro! Indicador não definido.
5	Avaliação da Arquitetura.....	10
5.1	Cenários	10
5.2	Avaliação	11
6	REFERÊNCIAS	13
7	APÊNDICES	14

1 Apresentação

Este projeto abordará o desenvolvimento de uma aplicação que permite a comunicação entre pessoas que têm dificuldades ou dúvidas sobre matérias acadêmicas. Além disso, são pessoas que desejam aprender ou solucionar estas dúvidas em grupos com indivíduos que apresentam ter conhecimento prévio neste assunto. A proposta terá como funcionalidades: criar ou entrar em grupos categorizados por matérias, para que seja possível solucionar estas dúvidas em tempo real. A ideia foi concebida da necessidade observada por docentes que entenderam ser necessário o desenvolvimento de uma plataforma onde seja possível sanar suas dúvidas acadêmicas. Esta aplicação será desenvolvida utilizando o framework REACT, que permite o desenvolvimento de aplicações para diferentes plataformas, o que é um grande desafio no mercado atual, tendo em vista a capacidade de integrar um código-fonte para diferentes plataformas. Uma das principais contribuições deste trabalho é poder conectar tanto os estudantes que têm dúvidas quanto as pessoas que têm disponibilidade e conhecimento para saná-las. Essa comunicação entre usuários é feita através da criação de um grupo categorizado onde é possível que cada usuário envie uma mensagem e seja respondido em tempo real.

1.1 Problema

A iniciativa de realizar o desenvolvimento deste projeto, se deu devido a dificuldade de docentes para sanar suas dúvidas de forma rápida e eficiente, fazendo com que prejudique a otimização do tempo destes para o esclarecimento de dificuldades de disciplinas acadêmicas. Visto que normalmente, estas dúvidas são solucionadas dentro das salas de aulas junto aos professores ou através de pesquisas *online*.

1.2 Objetivos do trabalho

O objetivo geral deste projeto é tornar possível a comunicação entre pessoas que se encontram com dificuldades ou dúvidas em matérias acadêmicas e que estejam à procura de alguém que possa auxiliá-las na resolução destas dúvidas. O intuito é utilizar o *chat* como ferramenta colaborativa de aprendizagem, de forma que possibilite maior interação entre alunos e pessoas capazes de solucionar dúvidas acadêmicas.

Os objetivos específicos do projeto são:

- ✓ Apresentar a contextualização e facilitar a resolução de dúvidas;
- ✓ Utilizar o framework REACT para desenvolver a interface final para o usuário, *frontend*;
- ✓ Utilizar o framework NEST.JS e NODE.JS para estruturar o *backend* da aplicação;
- ✓ Utilizar um *web-socket* para desenvolvimento do *web-chat* em tempo real;
- ✓ Possibilitar a comunicação entre estudantes interessados em solucionar dúvidas em grupos categorizados;

1.3 Definições e Abreviaturas

- RN (*React Native*) – é uma estrutura *JavaScript* para escrever aplicativos móveis reais e nativos para iOS e Android.
- NEST (*Nest.js*) – é uma estrutura para criar aplicativos *Node.js* do lado do servidor eficientes e escalonáveis.
- NODE (*Node.js*) – uma plataforma de desenvolvimento de código aberto para a execução de código *JavaScript* no lado do servidor.
- Front (*Frontend*) – é tudo com o qual o usuário interage.
- Back (*Backend*) – refere-se a qualquer parte de um site ou programa de software que os usuários não veem.
- Mongo (*MongoDB*) – é um banco de dados de documentos com a escalabilidade e flexibilidade que você deseja com a consulta e indexação de que você precisa.

2 Requisitos

Esta seção descreve os requisitos contemplados nesta descrição arquitetural, divididos em dois grupos: funcionais e não funcionais.

2.1 Requisitos Funcionais

Enumere os requisitos funcionais previstos para a sua aplicação. Concentre-se nos requisitos funcionais que sejam críticos para a definição arquitetural. Lembre-se de listar todos os requisitos que são necessários para garantir cobertura arquitetural. Esta seção deve conter uma lista de requisitos ainda sem modelagem.

ID	Descrição	Prioridade
RF001		
RF002		

2.2 Requisitos Não-Funcionais

Enumere os requisitos não-funcionais previstos para a sua aplicação. Entre os requisitos não funcionais, inclua todos os requisitos que julgar importante do ponto de vista arquitetural ou seja os requisitos que terão impacto na definição da arquitetura.

ID	Descrição
RNF001	
RNF002	

Obs: acrescente mais linhas, se necessário.

2.3 Restrições Arquiteturais

Enumere as restrições arquiteturais. Lembre-se de que as restrições arquiteturais geralmente não são consideradas requisitos uma vez que limitam a solução candidata. Os requisitos não impõem restrição, mas precisam ser satisfeitos.

As restrições impostas ao projeto que afetam sua arquitetura são (por exemplo):

- O software deverá ser desenvolvido em Python/Django
- A comunicação da API deve seguir o padrão RESTful

2.4 Mecanismos Arquiteturais

Visão geral dos mecanismos que compõem a arquitetura do software baseando-se em três estados: (1) análise, (2) design e (3) implementação. Em termos de Análise devem ser listados os aspectos gerais que compõem a arquitetura do software como: persistência, integração com sistemas legados, geração de logs do sistema, ambiente de front end, tratamento de exceções, formato dos testes, formato de distribuição/implantação (deploy), entre outros. Em Design deve-se identificar o padrão tecnológico a seguir para cada mecanismo identificado na análise. Em Implementação, deve-se identificar o produto a ser utilizado na solução.

Ex: Análise (Persistência), Design (ORM), Implementação (Hibernate).

Análise	Design	Implementação
Persistência		
Front end		
Back end		
Integração		
Log do sistema		
Teste de Software		
Deploy		

3 Modelagem e projeto arquitetural

Apresente uma visão geral da solução proposta para o projeto e se necessário, inclua um diagrama de visão geral. Esse diagrama é equivalente ao “Diagrama de contexto”.



Figura 1 – Diagrama de contexto: Visão Geral da Solução

Obs: substitua esta imagem por outra, que seja adequada ao seu projeto.

3.1 Visão de Casos de Uso

O diagrama de casos de uso oferece uma visão global dos casos de uso e dos atores que dele participam. Inclua um diagrama para cada um dos atores ou um diagrama que apresente as funcionalidades gerais do projeto.

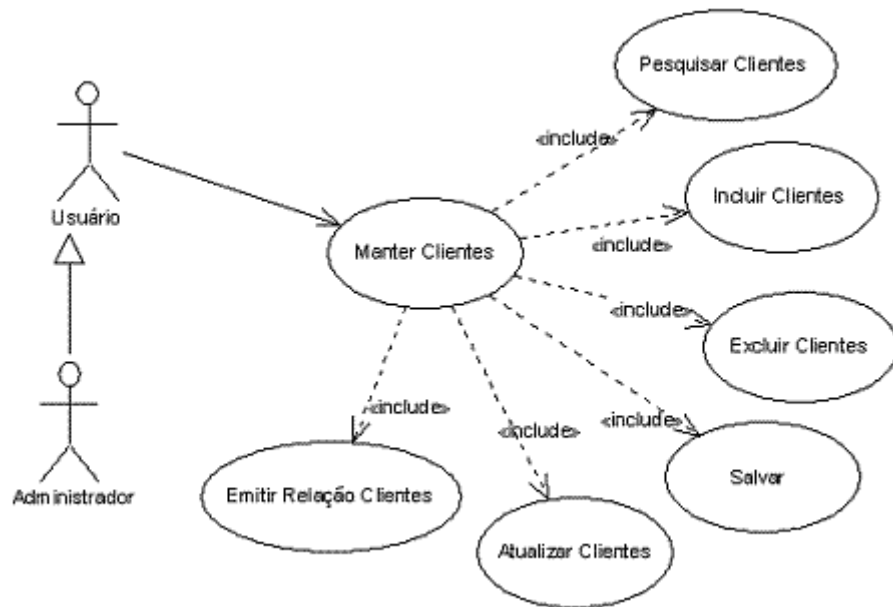


Figura 2 – Diagrama de Caso de Uso - Usuário

Obs: Acrescente uma breve descrição sobre o diagrama apresentado na Figura 2.

3.1.1 Descrição resumida dos casos de uso

Nesta seção, os casos de uso devem ser resumidos. Esse detalhamento pode ser na forma de um texto sintético ou, alternativamente, você pode optar por descrever histórias de usuários seguindo os métodos ágeis. Neste caso a seção deve chamar “Estórias de usuários”. Lembre-se das características de qualidade das estórias de usuários, ou seja, o que é preciso para descrever boas estórias de usuários.

UC01 – NOME DO CASO DE USO 01	
Descrição	
Atores	
Prioridade	
Requisitos associados	
Fluxo Principal	

UC02 – NOME DO CASO DE USO 02	
Descrição	
Atores	
Prioridade	
Requisitos associados	
Fluxo Principal	

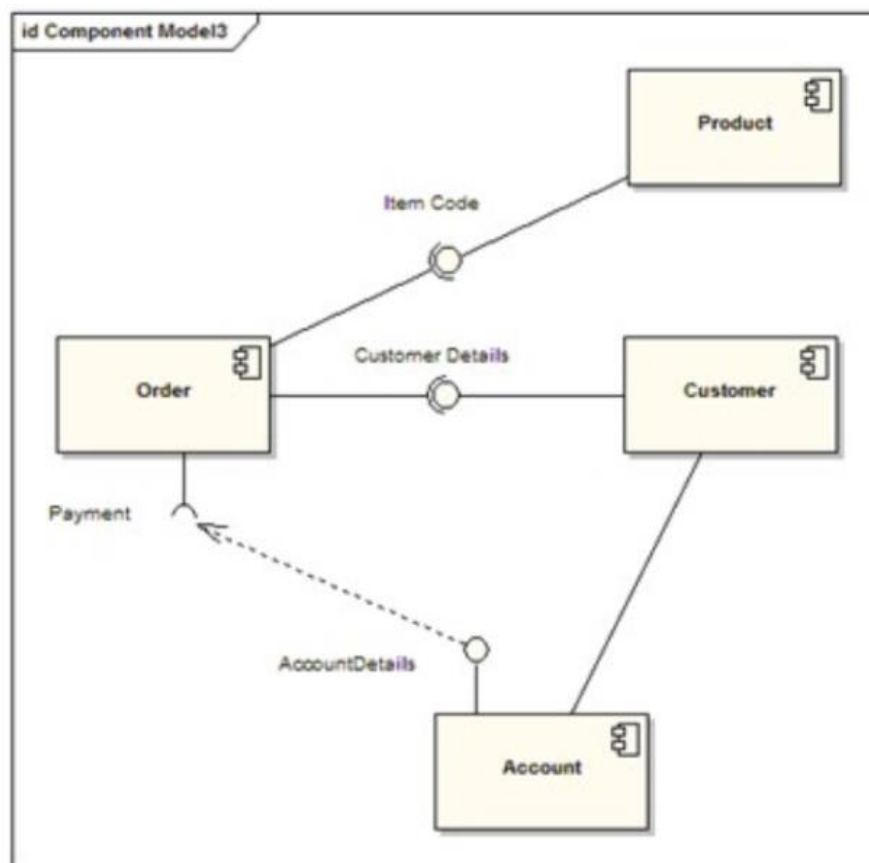


Figura 4 – Diagrama de Componentes

Apresente uma descrição detalhada dos artefatos que constituem o diagrama de implantação.

Conforme diagrama apresentado na Figura X, as entidades participantes da solução são:

- **Product** - Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras nunc magna, accumsan eget porta a, tincidunt sed mauris. Suspendisse orci nulla, sagittis a lorem laoreet, tincidunt imperdiet ipsum. Morbi malesuada pretium suscipit.
- **Order** - Praesent nec nisi hendrerit, ullamcorper tortor non, rutrum sem. In non lectus tortor. Nulla vel tincidunt eros.
- **Account** - Nulla eget viverra mi, eu malesuada erat. Maecenas molestie turpis sem, quis consectetur lorem feugiat vitae. Aenean malesuada ipsum eu luctus consequat.

3.3 Modelo de dados (opcional)

Caso julgue necessário para explicar a arquitetura, apresente o diagrama de classes ou diagrama de Entidade/Relacionamentos ou tabelas do banco de dados. Este modelo pode ser essencial caso a arquitetura utilize uma solução de banco de dados distribuídos ou um banco NoSQL.

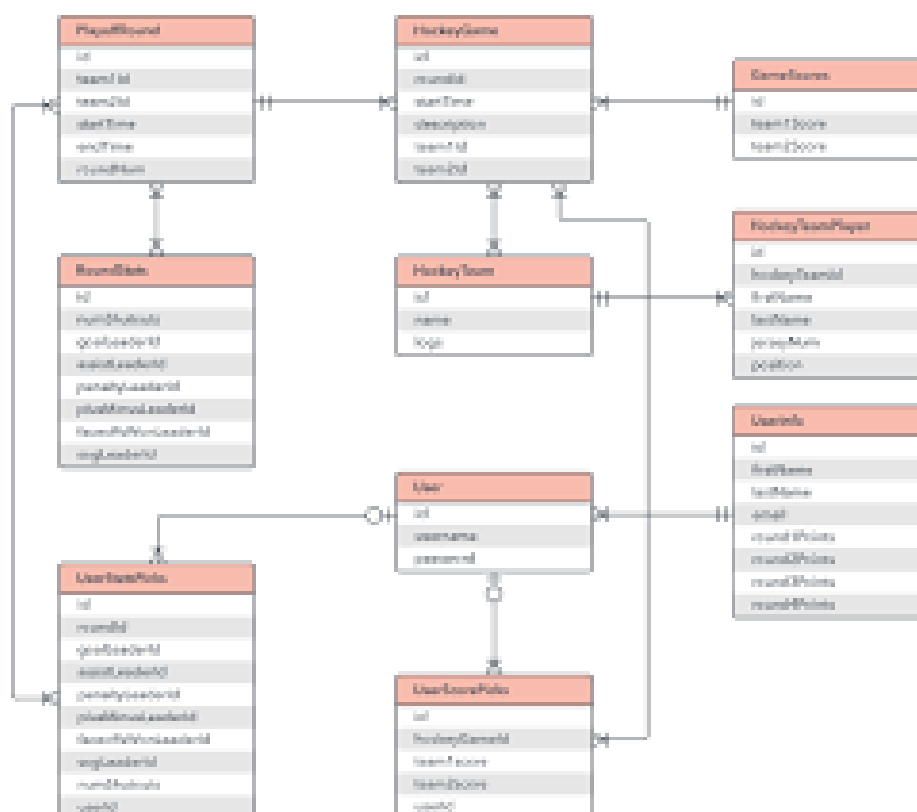


Figura 5 – Diagrama de Entidade Relacionamento (ER)

Obs: Acrescente uma breve descrição sobre o diagrama apresentado na Figura 3.

4 Avaliação da Arquitetura

Esta seção descreve a avaliação da arquitetura apresentada, baseada no método ATAM.

4.1 Cenários

Apresente os cenários de testes utilizados na realização dos testes da sua aplicação. Escolha cenários de testes que demonstrem os requisitos não funcionais sendo satisfeitos.

Cenário 1 - Acessibilidade: Suspendisse consequat consectetur velit. Sed sem risus, dictum dictum facilisis vitae, commodo quis leo. Vivamus nulla sem, cursus a mollis quis, interdum at nulla. Nullam dictum congue mauris. Praesent nec nisi hendrerit, ullamcorper tortor non, rutrum sem. In non lectus tortor. Nulla vel tincidunt eros.

Cenário 2 - Interoperabilidade: Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Fusce ut accumsan erat. Pellentesque in enim tempus, iaculis sem in, semper arcu.

Cenário 3 - Manutenibilidade: Phasellus magna tellus, consectetur quis scelerisque eget, ultricies eu ligula. Sed rhoncus fermentum nisi, a ullamcorper leo fringilla id. Nulla

lacinia sem vel magna ornare, non tincidunt ipsum rhoncus. Nam euismod semper ante id tristique. Mauris vel elit augue.

Cenário 4 - Segurança: Suspendisse consectetur porta tortor non convallis. Sed lobortis erat sed dignissim dignissim. Nunc eleifend elit et aliquet imperdiet. Ut eu quam at lacus tincidunt fringilla eget maximus metus. Praesent finibus, sapien eget molestie porta, neque turpis congue risus, vel porttitor sapien tortor ac nulla. Aliquam erat volutpat.

4.2 Avaliação

Apresente as medidas registradas na coleta de dados. O que não for possível quantificar apresente uma justificativa baseada em evidências qualitativas que suportam o atendimento do requisito não-funcional.

Apresente uma avaliação geral da arquitetura indicando os pontos fortes e as limitações da arquitetura proposta.

Atributo de Qualidade:	Requisito de Qualidade
Segurança	Acesso deve ser controlado aos recursos restritos
Preocupação:	
Os acessos de usuários devem ser controlados de forma que cada um tenha acesso apenas aos recursos condizentes as suas credenciais.	
Cenários(s):	
Cenário 4	
Ambiente:	
Sistema em operação normal	
Estímulo:	
Acesso do administrador do sistema as funcionalidades de cadastro de novos produtos e exclusão de produtos.	
Mecanismo:	
O servidor de aplicação (Rails) gera um <i>token</i> de acesso para o usuário que se autentica no sistema. Este <i>token</i> é transferido para a camada de visualização (Angular) após a autenticação e o tratamento visual das funcionalidades podem ser tratados neste nível.	
Medida de Resposta:	
As áreas restritas do sistema devem ser disponibilizadas apenas quando há o acesso de usuários credenciados.	
Considerações sobre a arquitetura:	
Riscos:	Não existe
Pontos de Sensibilidade:	Não existe
Tradeoff:	Não existe

Evidências dos testes realizados

Apresente imagens, descreva os testes de tal forma que se comprove a realização da avaliação.

5 REFERÊNCIAS

Como um projeto da arquitetura de uma aplicação não requer revisão bibliográfica, a inclusão das referências não é obrigatória. No entanto, caso você deseje incluir referências relacionadas às tecnologias, padrões, ou metodologias que serão usadas no seu trabalho, relacione-as de acordo com o modelo a seguir.

SOBRENOME DO AUTOR, Nome do autor. **Título do livro ou artigo.** Cidade: Editora, ano.

SOBRENOME DO AUTOR, Nome do autor. **Título do livro ou artigo.** Cidade: Editora, ano.

SOBRENOME DO AUTOR, Nome do autor. **Título do livro ou artigo.** Cidade: Editora, ano.

SOBRENOME DO AUTOR, Nome do autor. **Título do livro ou artigo.** Cidade: Editora, ano.

SOBRENOME DO AUTOR, Nome do autor. **Título do livro ou artigo.** Cidade: Editora, ano.

SOBRENOME DO AUTOR, Nome do autor. **Título do livro ou artigo.** Cidade: Editora, ano.

6 APÊNDICES

Inclua o URL do repositório (Github, Bitbucket, etc) onde você armazenou o código da sua prova de conceito/protótipo arquitetural da aplicação como anexos. A inclusão da URL desse repositório de código servirá como base para garantir a autenticidade dos trabalhos.