

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
02/09/2020	Vitor Augusto	Desenvolvimento da sessão 3.1.1. Descrição resumida dos casos de uso	1.2
02/09/2020	Matheus Santos; Raissa Vilela; Vi- tor Augusto	Desenvolvimento de diagramas sessão 3.	1.3

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	5
3	Modelagem e projeto arquitetural	6
3.1	Visão de Casos de Uso	7
3.2	Visão Lógica.....	9
3.3	Visão Física.....	Erro! Indicador não definido.
3.4	Modelo de dados (opcional)	11
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	12
5.1	Cenários.....	12
5.2	Avaliação.....	13
6	REFERÊNCIAS.....	15
7	APÊNDICES	16

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

2.1 Requisitos Funcionais

ID	Descrição	Prioridade
RF001	Registrar no sistema.	Alta
RF002	Fazer login.	Alta
RF003	Fazer logout.	Média
RF004	Criar sala de bate papo.	Alta
RF005	Alterar o nome da sala.	Baixa
RF006	Entrar na sala de bate papo.	Alta
RF007	Enviar mensagem.	Alta
RF008	Sair da sala.	Alta
RF009	Listar salas disponíveis.	Alta

2.2 Requisitos Não-Funcionais

ID	Descrição
RNF001	Performance: a aplicação deve registrar as mensagens em até 5 segundos.
RNF002	Segurança: a autenticação do usuário deve seguir algum padrão de criptografia.
RNF003	Performance: o usuário deve conseguir entrar em uma sala de bate-papo em até 5 segundos.
RNF004	Tolerância a falhas: as mensagens que não forem efetivadas devem ser reenviadas quando a conexão for reestabelecida.

2.3 Restrições Arquiteturais

- A aplicação não enviará nenhum tipo de anexo.
- Só serão utilizados softwares e facilitadores gratuitos.
- O software deverá ser desenvolvido em Node.js e React.
- O *deploy* da aplicação deverá ser feito em uma plataforma gratuita.

2.4 Mecanismos Arquiteturais

Análise	Design	Implementação
Persistência	Banco de dados não relacional.	MongoDB
Front end	Tecnologias Web	React, HTML5, CSS3

Back end	Plataforma de Desenvolvimento Orientada a Objetos	Node.js
Integração	Webservice REST	
Log do sistema		
Teste de Software		
Deploy	Ambiente de integração contínua	Heroku

3 Modelagem e projeto arquitetural

O diagrama de contexto é essencial para o desenvolvimento, devido que, mostra rapidamente o escopo e os limites de um sistema. Além disso, não necessita de nenhum conhecimento técnico para ser entendido e é fácil de se expandir pela adição de diferentes níveis.

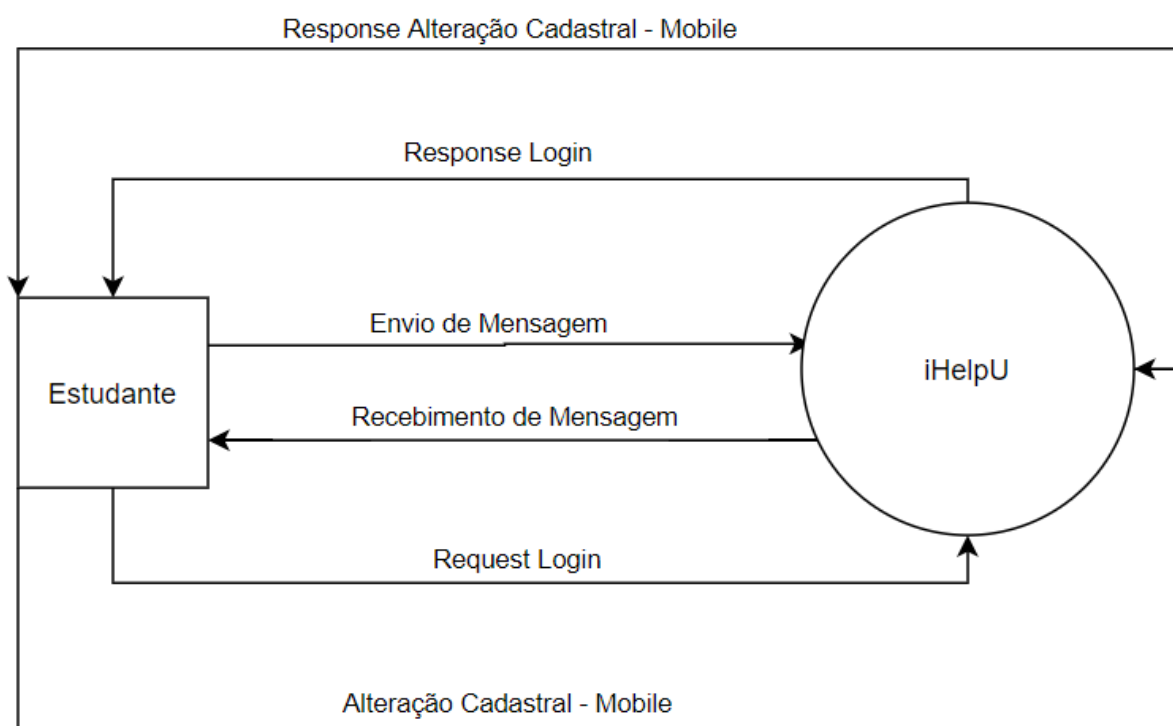


Figura 1 – Diagrama de contexto: Visão Geral da Solução

3.1 Visão de Casos de Uso

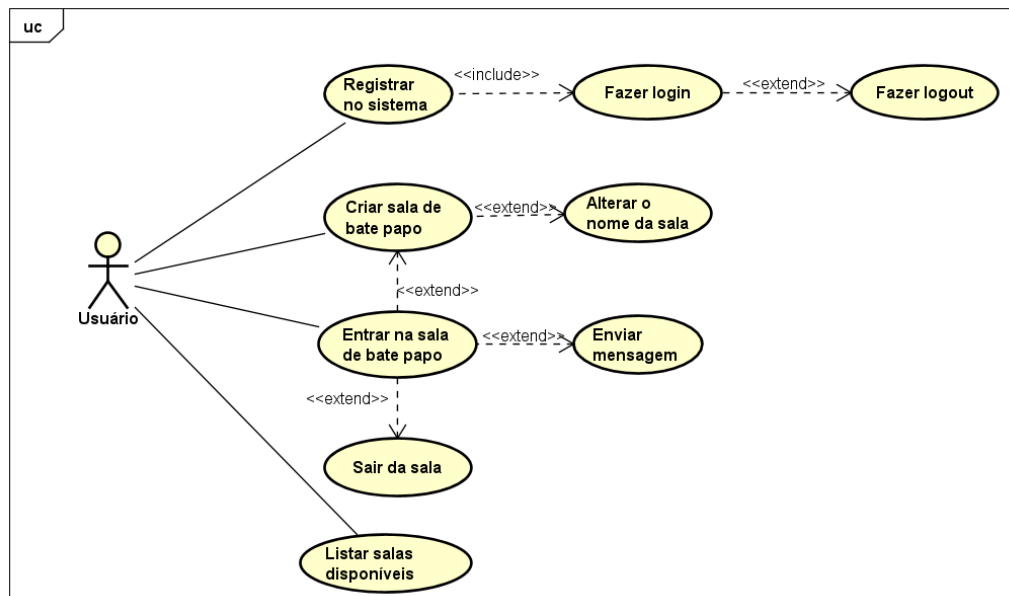


Figura 2 – Diagrama de Caso de Uso - Usuário

3.1.1 Descrição resumida dos casos de uso

UC01 – REALIZAR O CADASTRO NO SISTEMA	
Descrição	Este caso de uso é direcionado para o usuário que deseja se registrar no sistema.
Atores	Estudante
Prioridade	1
Requisitos associados	RF001
Fluxo Principal	Usuário insere Nome, Universidade, E-mail, Senha e Telefone para realizar cadastro no sistema e clica no botão “Registrar”.

UC02 – REALIZAR LOGIN NO SISTEMA	
Descrição	O caso de uso realizar login no sistema descreve a situação que o usuário deseja entrar no sistema.
Atores	Estudante
Prioridade	1
Requisitos associados	RF002
Fluxo Principal	Usuário acessa a URL para acessar ao sistema, insere seu login e sua senha e clica no botão de login para acessar a aplicação.

UC03 – REALIZAR LOGOUT NO SISTEMA	
Descrição	O caso de uso é realizar logout no sistema descreve a situação que o usuário deseja sair do sistema.
Atores	Estudante
Prioridade	1

Requisitos associados	RF002
Fluxo Principal	Estando logado no sistema o usuário clica no sistema logout e será direcionado para a área não logada do sistema

UC04 – REALIZAR CRIAÇÃO DA SALA DE BATE PAPO

Descrição	O caso de uso criação da sala de bate papo descreve a criação das salas de bate papo pelos usuários.
Atores	Estudante
Prioridade	1
Requisitos associados	RF003
Fluxo Principal	Estando logado no sistema o usuário clica no botão “Criar nova sala” e um modal é aberto e nele o usuário insere o nome da sala e seleciona o assunto a ser discutido

UC05 – REALIZAR ALTERAÇÃO DO NOME DA SALA

Descrição	O caso de uso alterar o nome da sala descreve a alteração do nome das salas de bate papo pelos usuários.
Atores	Estudante
Prioridade	2
Requisitos associados	RF005
Fluxo Principal	O usuário que criou a sala poderá renomear clicando em um botão de edição na da sala.

UC06 – ENTRAR NA SALA DE BATE PAPO

Descrição	O caso de uso entrar na sala de bate papo descreve o fluxo de entrada na sala de bate papo pelos usuários.
Atores	Estudante
Prioridade	1
Requisitos associados	RF006 e RF008
Fluxo Principal	A partir da barra de buscas, o usuário pesquisa a sala que deseja ingressar e clicar no botão para fazer parte do grupo.

UC07 – SAIR DA SALA DE BATE PAPO

Descrição	O caso de uso sair da sala de bate papo é acionado quando um usuário deseja sair da sala de bate papo está participando.
Atores	Estudante
Prioridade	1
Requisitos associados	RF006
Fluxo Principal	Estando em algum grupo, o usuário pode clicar no botão de “sair do grupo” para deixar a sala.

UC08 – ENVIAR MENSAGEM	
Descrição	O caso de uso enviar mensagem descreve o fluxo de envio de mensagens para os grupos participantes.
Atores	Estudante
Prioridade	1
Requisitos associados	RF007
Fluxo Principal	Estando em algum grupo previamente criado, o usuário poderá digitar sua mensagem para que todos os participantes consigam vê-las.

UC09 – SAIR DA SALA	
Descrição	O caso de uso sair da sala é acionado quando algum dos integrantes de algum grupo deseja sair do chat que está participando.
Atores	Estudante
Prioridade	1
Requisitos associados	RF008
Fluxo Principal	Estando em um dos chats o usuário clica no botão “sair” e será direcionado a tela logada do sistema lista todas as salas que o usuário está participando.

UC10 – REALIZAR LISTAGEM DE SALAS DISPONÍVEIS	
Descrição	O caso de uso realizar listagem de salas disponíveis é acionado quando algum usuário deseja pesquisar algum assunto que deseja esclarecer suas dúvidas.
Atores	Estudante
Prioridade	3
Requisitos associados	RF009
Fluxo Principal	Estando logado no sistema o usuário poderá listar as salas disponíveis de acordo com uma barra de pesquisa para encontrar o assunto desejado.

3.2 Visão Lógica

O diagrama de classes é fundamental pois mapeiam de forma clara a estrutura do sistema ao modelarmos classes, atributos, operações e relações entre objetos. O modelo de componentes descreve a estrutura física do sistema e como esses componentes do sistema e como eles se relacionam. Além disso, ele enfatiza o comportamento do serviço quanto à sua interface.

Modelo de classes

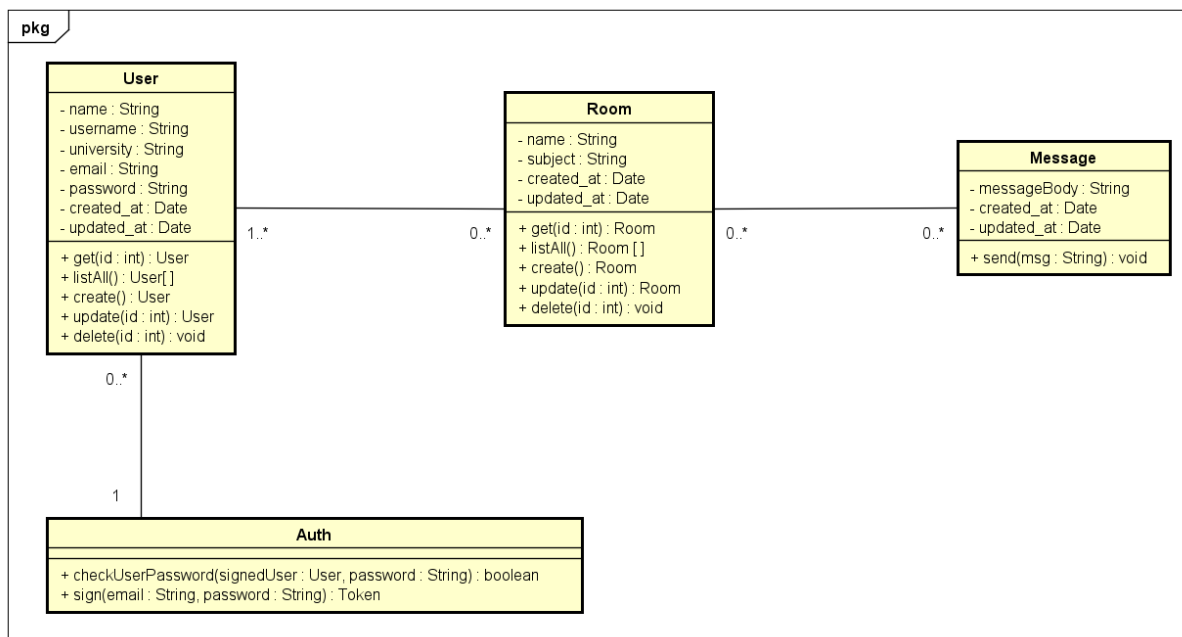


Figura 3 – Diagrama de classes

O diagrama de classes acima apresenta quatro principais classes da aplicação. A classe *user* é responsável por armazenar as informações dos usuários que acessam ao sistema. A classe *room* é responsável por garantir a consistência e informações referentes às salas criadas. A classe *Message* é responsável por instanciar as mensagens que serão trafegadas na sala e, por fim, a classe *Auth* é responsável pelo controle de sessão de cada usuário.

3.2.1 Modelo de componentes

Apresente o diagrama de componentes da aplicação, indicando, os elementos da arquitetura e as interfaces entre eles. Liste os estilos/padrões arquiteturais utilizados e faça uma descrição sucinta dos componentes indicando o papel de cada um deles dentro da arquitetura/estilo/padrão arquitetural. Indique também quais componentes serão reutilizados (navegadores, SGBDs, middlewares, etc), quais componentes serão adquiridos por serem proprietários e quais componentes precisam ser desenvolvidos.

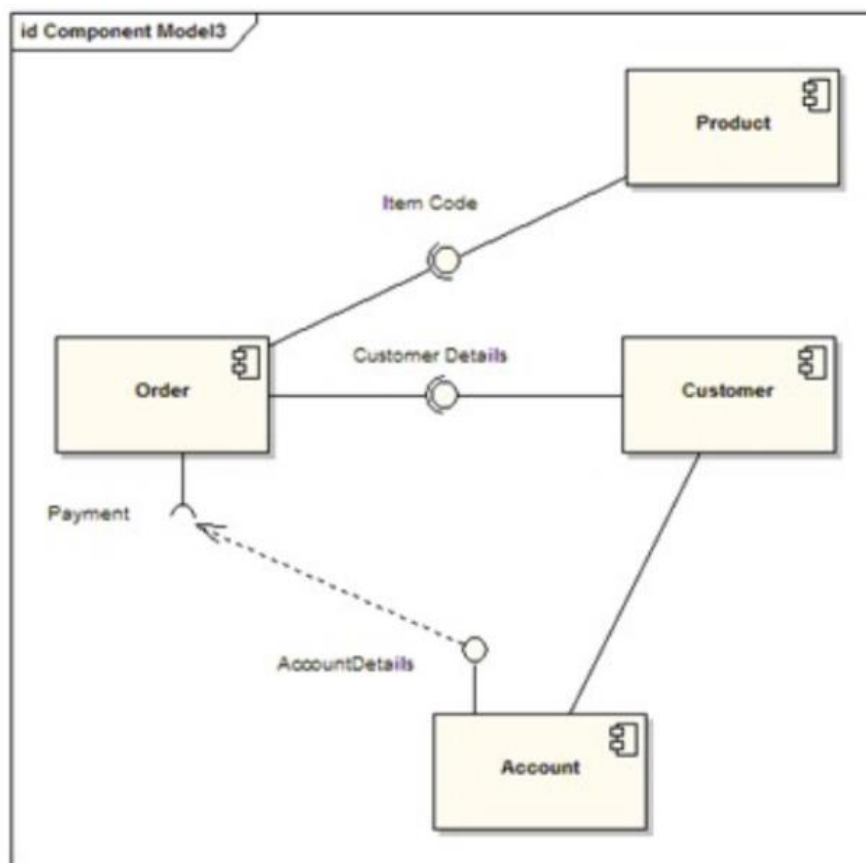


Figura 4 – Diagrama de Componentes

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.

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.