

Bonde do Godzilla 🦖

Descrição do produto:

Administração de estoque e financeiro de forma intuitiva e que seja possível visualizar através do mobile.

Oportunidades e justificativa:

- > Controle mais intuitivo do estoque;
- > Não existem softwares intuitivos e mobile para controle de estoque e lojas em geral.

Benefícios:

- > Organização dos produtos disponíveis para o uso;
- > Controle de quando vai comprar um produto que está em falta;
- > Comunicação entre as áreas da empresa: administrativo, produção, vendas.

Recursos:

- > Banco de dados SQL
- > REST API para backend
- > Frontend web e mobile

Equipe:

- Gabriel Carvalho
- Guilherme Reis
- Maria Carolina Florio
- Vitor Grechi Kuninari

Etapas:

Estudo planejamento e desenvolvimento (CBL)

Riscos:

Custo do produto e concorrentes.

Crítérios de qualidade:

Usabilidade, estabilidade e otimização

Clientes:

A parte administrativa e de produção de empresas voltadas para o comercio

UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática

ENGENHARIA DE SOFTWARE

Prof.: Renata Mendes de Araujo

TG2 - Levantamento de Requisitos

Nome:	TIA:
Gabriel Ferreira de Carvalho	41806107
Guilherme Santiago Reis	31920918
Maria Carolina de Moraes Florio	31988075
Vitor Grechi Kuninari	31906230

EXERCÍCIOS

Identifique, declare e documente os requisitos do software a serem desenvolvidos no projeto.

Se você quiser usar alguma ferramenta específica, ok. Se não, uma planilha Excel resolve!

Neste momento, concentre-se na qualidade da declaração dos requisitos.

Dicas:

1. Liste todos os usuários do software que será construído.

Funcionários, administradores.

2. O software terá que conversar com outros sistemas? Liste também.

Sim, sistemas de controle de finanças e vendas e banco de dados.

3. Para cada um dos usuários e/ou outros sistemas mencionados acima, liste tudo o que o software precisará oferecer para cada um deles.

- App mobile - visualização e controle de estoque.
- App desktop - para melhor controle do estoque.
- API para comunicação com controle financeiro e o banco de dados.

4. Descreva estas funcionalidades como requisitos de usuário.

Funcionário

- Alterar quantidade de itens;
- Visualizar item;
- Visualizar tabela de itens;
- Filtrar tabela de itens;
- Ordenar tabela de itens;
- Notificar administrador para pedir novos itens.

Administrador

- Adicionar novos itens;
- Alterar quantidade de itens;
- Apagar itens existentes;
- Visualizar item;
- Visualizar tabela de itens;
- Filtrar tabela de itens;
- Ordenar tabela de itens;
- Notificar produtores para fabricar novos itens.

5. Verifique se as descrições são claras, não ambíguas e completas. A lista contém tudo o que é necessário?

Sim, a lista contém tudo que é necessário.

Lembre-se que a lista não é elaborada para você, ela é elaborada para comunicar os requisitos para a equipe e para fechar um acordo de desenvolvimento com o cliente!

Neste processo, perceba que a demanda pode ser alterada, conforme a equipe discute os requisitos. Se isso acontecer, reveja a descrição da demanda!

UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática

ENGENHARIA DE SOFTWARE

Prof.: Renata Mendes de Araujo

TG3 – Priorização de Requisitos

Nome:

Gabriel Ferreira de Carvalho

Guilherme Santiago Reis

Maria Carolina de Moraes Florio

Vitor Grechi Kuninari

TIA:

41806107

31920918

31988075

31906230

EXERCÍCIOS

Considere a lista de requisitos levantada pela equipe para o projeto.

Revise-a, se necessário, aproveitando para classificar corretamente os requisitos identificados em funcionais e não funcionais, bem como aperfeiçoar sua descrição.

Priorize os requisitos de acordo com sua criticidade, esforço e risco.

Selecione os requisitos que farão parte do escopo do projeto.

#	Requisitos: 1, 3, 4, 2	Classificação	Dependências	Esforços	Relevância	Risco
1	Adicionar itens	RF	---	MÉDIO	ALTA	BAIXO
2	Alterar itens	RF	RF1	MÉDIO	MÉDIA	BAIXO
3	Visualizar produtos	RNF	RF1	FÁCIL	ALTA	BAIXO
4	Notificar produtores para fabricar novos itens	RF	RF1, RF2	MÉDIO	ALTA	MÉDIO

UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática

ENGENHARIA DE SOFTWARE

Prof.: Renata Mendes de Araujo

TG4 – Estimando o Projeto

Nome:	TIA:
Gabriel Ferreira de Carvalho	41806107
Guilherme Santiago Reis	31920918
Maria Carolina de Moraes Florio	31988075
Vitor Grechi Kuninari	31906230

EXERCÍCIOS

Estime o esforço e tempo para desenvolvimento do projeto.

- 1) Defina e organize as fases e atividades necessárias para o desenvolvimento do projeto.
- 2) Considere a lista de requisitos e as atividades principais definidas para o projeto, estime o esforço para desenvolver cada um deles.
- 3) Estime também o esforço necessário para realizar outras atividades do projeto que não envolvam diretamente o desenvolvimento do projeto.
- 3) Use uma técnica de estimativas escolhida pela equipe. Sugiro usarem uma estimativa baseada em julgamento de especialistas, ex. Scrum Poker.
- 4) Registre as estimativas na planilha de requisitos.
- 5) Identifique os principais marcos/entregas do projeto.

#	Requisitos	Classificação	Dependências	Esforços	Relevância	Risco	Estimativa de Esforço
1	O sistema deve permitir adicionar novos itens	RF	---	MÉDIO	ALTA	BAIXO	5 dias
2	O sistema deve permitir fazer alterações nos itens: quantidade, preço, nome etc.	RF	RF1	MÉDIO	MÉDIA	BAIXO	8 dias
3	O sistema deve exibir os produtos do estoque	RNF	RF1	BAIXO	ALTA	BAIXO	5 dias
4	O sistema deve notificar produtores para fabricar novos itens quando eles estiverem em poucas quantidades	RF	RF1, RF2	MÉDIO	ALTA	MÉDIO	5 dias
5	O sistema deve permitir excluir produtos do estoque	RF	RF1	BAIXO	ALTA	ALTO	3 dias
6	O sistema deve possuir uma integração com APIs do mercado da área de vendas	RF	---	MÉDIO	MÉDIA	MÉDIO	14 dias
7	O sistema deve permitir a criação de diferentes perfis de usuários	RF	---	ALTO	MÉDIA	ALTO	30 dias
8	O sistema deve permitir o acesso aos usuários cadastrados caso o login e senha estiverem corretos	RF	RF7	BAIXO	ALTO	ALTO	30 dias
9	O sistema deve notificar quando um produto esgotou	RF	RF1, RF4	BAIXO	ALTO	ALTO	8 dias
10	O usuário pode definir quando as notificações serão enviadas	RF	RF1, RF4	BAIXO	BAIXO	MÉDIO	4 dias
11	O sistema deve permitir o usuário filtrar os itens do estoque	RNF	R1	MÉDIO	ALTO	MÉDIO	14 dias
12	O sistema deve permitir o usuário ordenar os itens do estoque	RNF	R1	MÉDIO	ALTO	MÉDIO	14 dias

Historia de usuários



Personas

- João: Gerente de estoque
- Maria: Dona da loja



RF1 - O sistemas deve permitir adicionar novos itens

Como João, quero adicionar novos itens que chegam na loja, para que os novos itens da loja estejam sempre cadastrados.



RF2 - O sistema deve permitir fazer alterações nos itens: quantidade, preço, nome e etc.

Como João, quero alterar quantidades, preços e nomes para manter o estoque dos produtos atualizados.



RF4 - O sistema deve permitir excluir produtos do estoque

Como João, quero excluir produtos do estoque para manter apenas os produtos que desejamos vender dentro de nossa loja.

...



RNF1 - O sistema deve exibir os produtos do estoque.

Como Maria, quero ver os produtos que estão no estoque da minha loja para ter controle da minha empresa.

Diagrama de classe

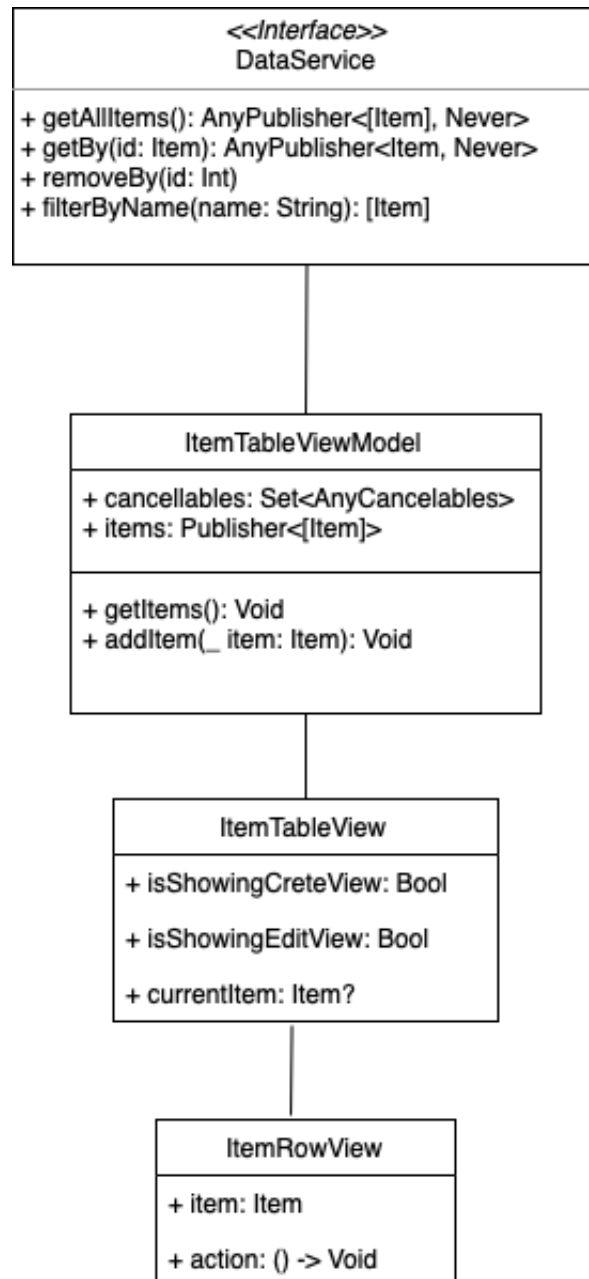


Diagrama de módulo

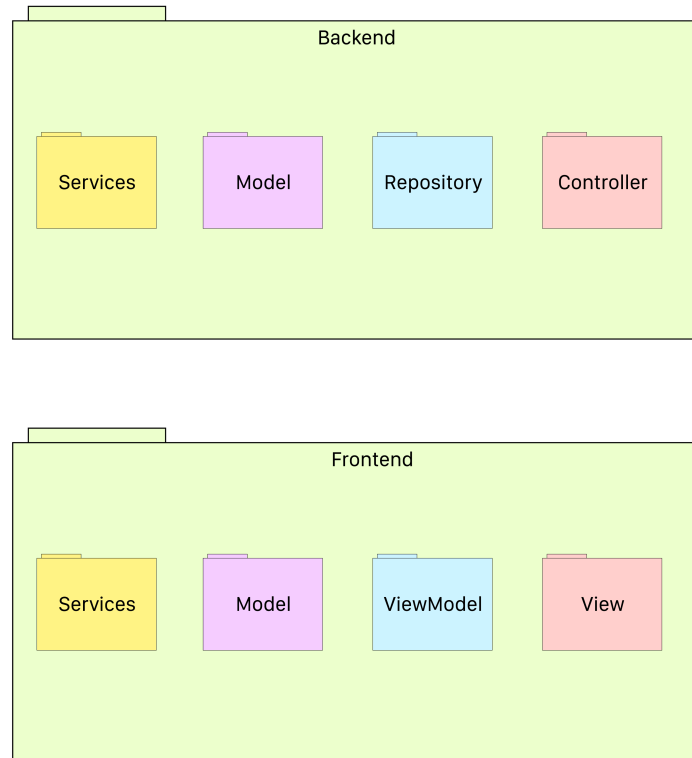
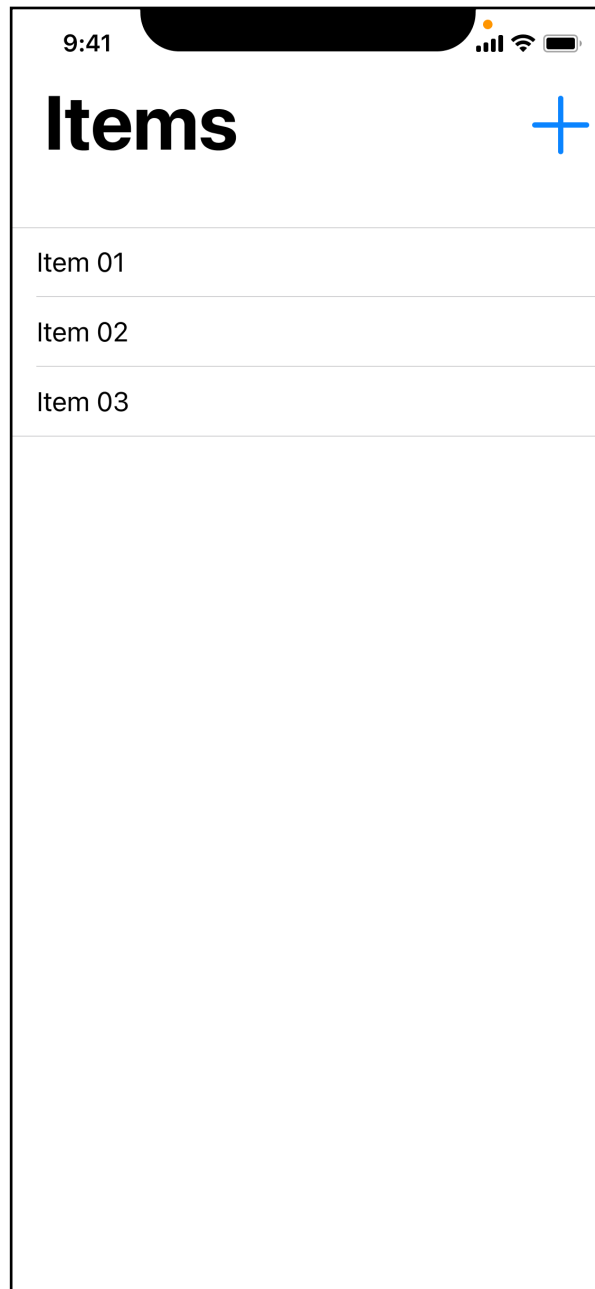


Diagrama de banco de datos

Item
<ul style="list-style-type: none">• id: Int• name: String• stock: Int• supplier: String• category: String• price: Float

Protótipo de Tela



Implementação

<https://github.com/gbrlCM/Trabalho-Engenharia-Software/tree/main/Source/Shared>

Plano de testes

- O desenvolvimento do produto será feito através do método TDD (Test Driven Development), onde os teste serão feitos antes da própria *feature*.
- Em caso de alterações significativas em módulos, serão feitos testes de integração para garantir o funcionamento.
- Em cada **major release** serão feitos teste alpha e beta além de um **release candidate** para garantir o funcionamento das partes mais importantes.

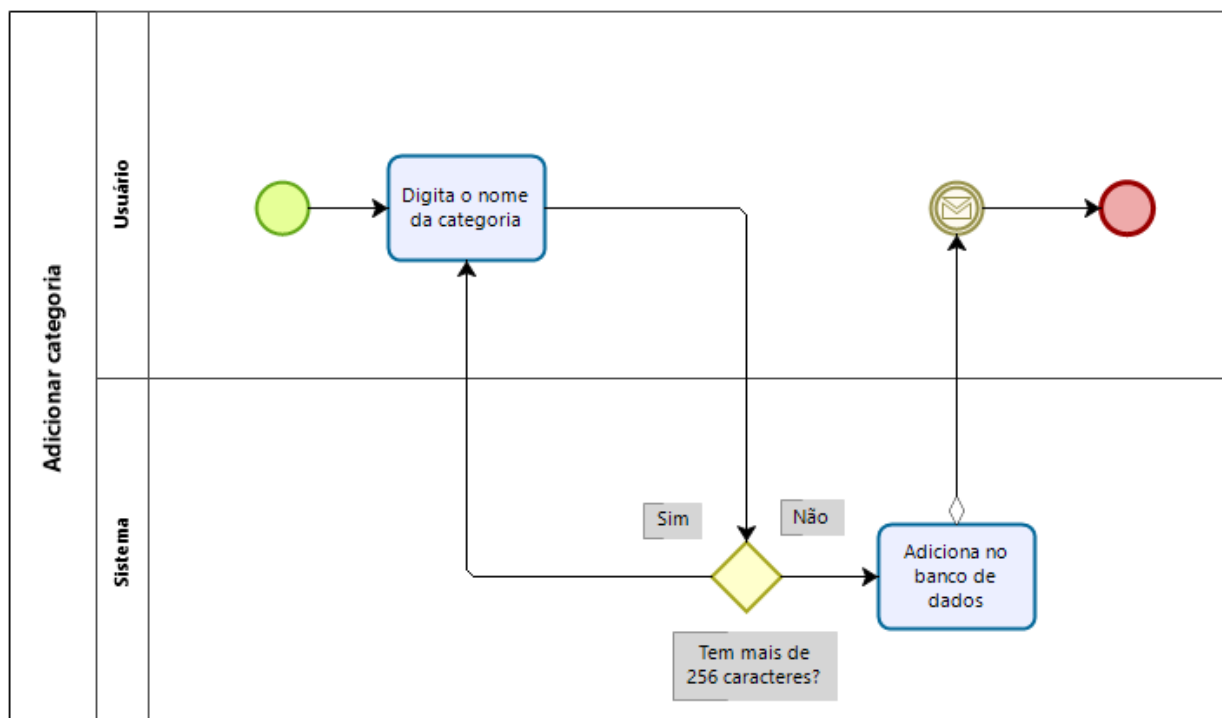
Caso de Teste Unitário | Classe itemsViewModel

Lista de testes:

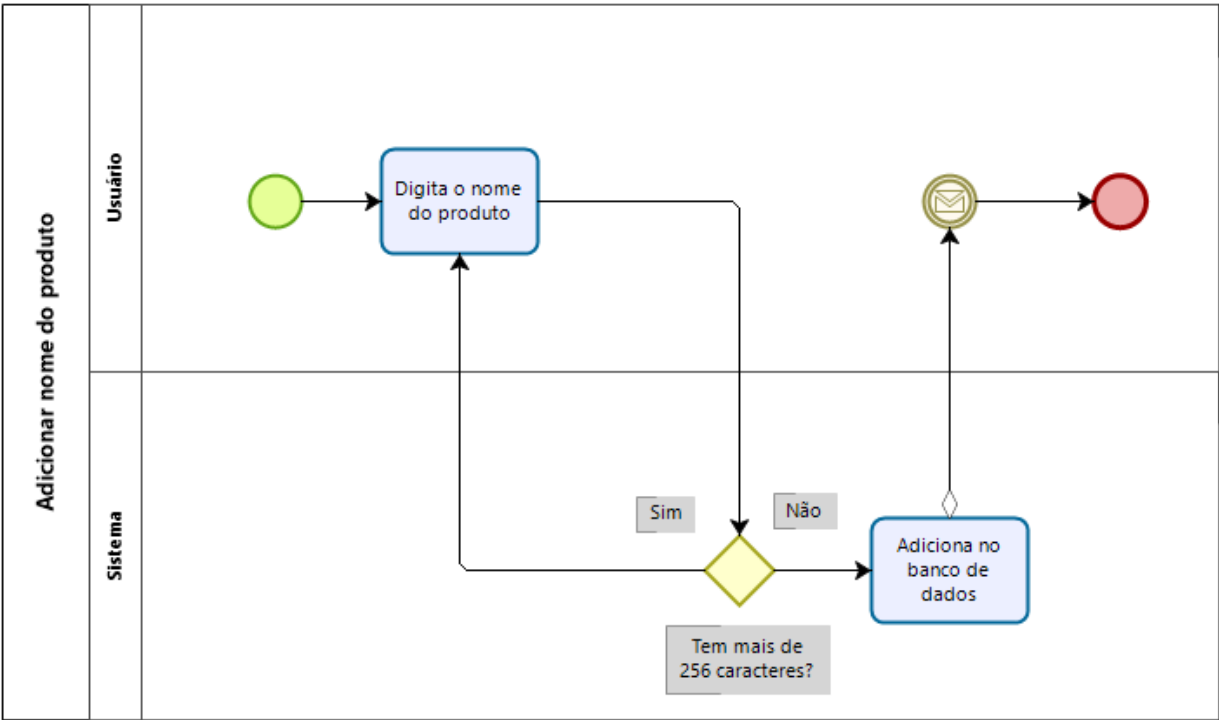
- getItem():
 - Um objeto da classe DSMock para passar dados "mocados" para a classe;
 - É esperado um retorno de uma lista específica.
- addItem():
 - É esperado um retorno da lista "mocada" mais o item adicionado.

Diagramas de teste

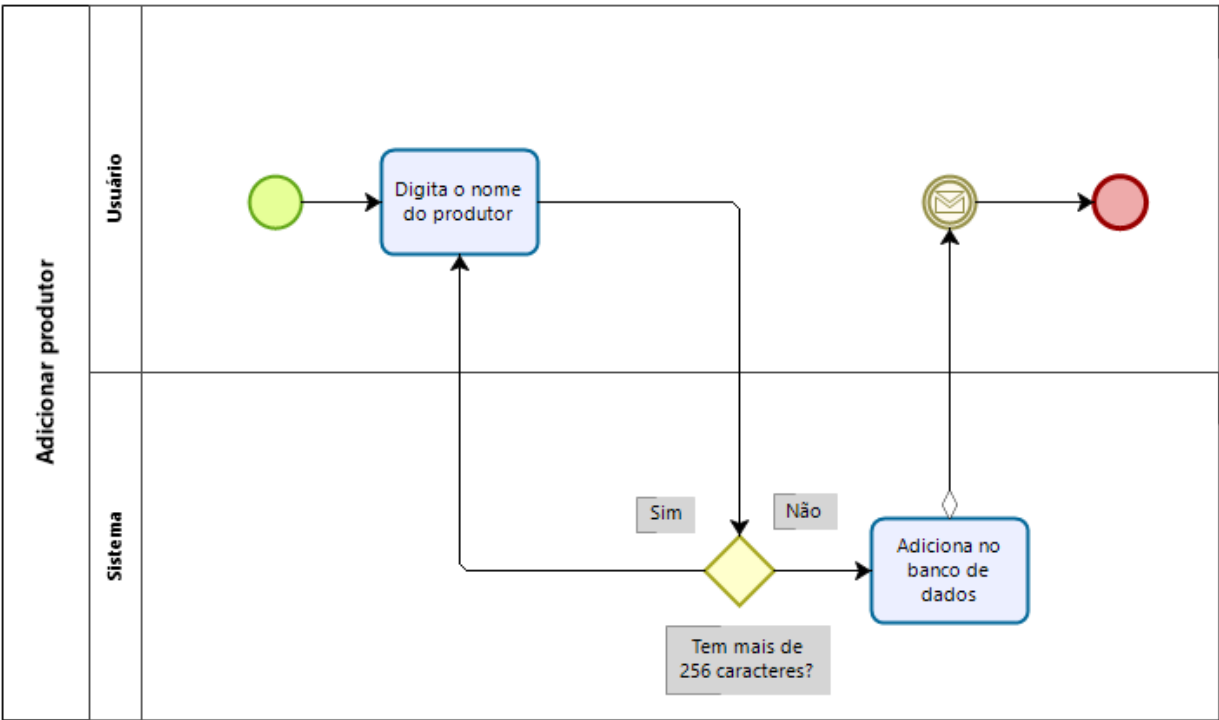
TS001 - Adicionar categoria



TS002 - Adicionar produto



TS003 - Adicionar produtor



Implantação de software

- As atividades necessárias para garantir sua distribuição e operação

Para publicar um app na appStore é necessário mandar para submissão na Apple, isso leva em torno de 24hs. Para a API é necessário colocar na nuvem em algum lugar como a AWS ou no Heroku.

- O tempo estimado para realizar estas atividades

Depois de enviar para Review um App leva 24hs para ter o resultado do Review, caso aprovado está automaticamente na loja, no caso de uma API leva alguns minutos para subir uma nova versão na nuvem.

- Os recursos (humanos e materiais) necessários para realizar estas atividades

Para um App é necessário uma conta de desenvolvedor da Apple que custa 100\$ por ano e para uma agilidade maior seria necessário algum serviço como o bitrice que é um serviço de CI/CD que custa para pequenas empresas 90\$ por mês.

Para uma API é necessário de algum serviço de cloud como a AWS ou Heroku.

No mundo atual é interessante que um dos desenvolvedores com mais experiência seja o responsável para submeter o app ou subir a app.

- Os eventuais riscos de sua realização.

Os maiores riscos são mudanças da política de revisão de apps pela Apple, e isso podem gerar possíveis atrasos para subir um app.