

# Documento de Engenharia de Software

## DIVIDEA.Í

Preparado por:

Augusto Perez de Andrade - 11611EMT009  
Gabriel Renato Oliveira Alves - 11621EMT007  
VÍctor Assunção Ávila - 11721EMT006  
Yuri Lima Almeida - 11621EMT022

*Faculdade de Computação*  
*Engenharia de Software*

30 de Março de 2022

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>2</b>
1.1	Objetivo deste documento . . . . .	2
1.2	Escopo do produto . . . . .	2
1.2.1	Nome do produto e principais componentes . . . . .	2
1.2.2	Missão do produto . . . . .	3
1.2.3	Limites do produto . . . . .	3
1.2.4	Restrições . . . . .	3
1.2.5	Benefícios esperados do produto . . . . .	3
1.3	Definições, Acrônimos e Abreviações . . . . .	3
1.4	Estrutura do texto . . . . .	4
<b>2</b>	<b>Organização dos trabalhos</b>	<b>5</b>
2.1	Reuniões . . . . .	5
2.2	Ferramenta colaborativa para gestão e controle do projeto . . . . .	5
<b>3</b>	<b>Especificação de requisitos do sistema</b>	<b>6</b>
3.1	Requisitos Funcionais . . . . .	6
3.1.1	Diagrama de Casos e Usos . . . . .	11
3.2	Requisitos não Funcionais . . . . .	12
3.2.1	Requisitos de Performance . . . . .	12
3.2.2	Requisitos de Segurança . . . . .	12
3.2.3	Requisitos de Usabilidade . . . . .	12
<b>4</b>	<b>Arquitetura do Sistema</b>	<b>13</b>
4.1	Definição do ferramental . . . . .	14
4.1.1	NodeJs . . . . .	14
4.1.2	Express . . . . .	14
4.1.3	MongoDB . . . . .	14
4.1.4	Flutter . . . . .	14
4.1.5	Docker . . . . .	15

# Capítulo 1

## Introdução

### 1.1 Objetivo deste documento

O objetivo deste artefato é descrever e especificar os requisitos que devem ser atendidos pelo produto **DIVIDEA.Í**, de forma a satisfazer as necessidades de seus clientes, bem como definir o produto a ser feito, para os desenvolvedores do grupo.

### 1.2 Escopo do produto

O **DIVIDEA.Í** tem por escopo permitir a *usuários* verificar informações sobre dívidas. Os *usuários* com as mesmas dívidas poderão se associar por grupos permitindo a transparência das informações entre os associados. As dívidas serão integradas ao sistema pelos próprios *usuários*.

#### 1.2.1 Nome do produto e principais componentes

O nome do produto será chamado de **DIVIDEA.Í** e possuirá os seguintes componentes para garantir o gerenciamento e segurança das informações:

- Módulo de Registro de Dívidas: responsável pelo recebimento e validação de registros de dívidas em formato PDF, bem como pelo envio de respostas de sucesso ou falha no carregamento do registro.
- Módulo de criação de grupos: responsável por disponibilizar a ferramenta de criar um grupo de compartilhamento e rateio de dívidas.
- Módulo de cadastro de *usuários*: responsável pelo cadastro e validação das informações de registro dos usuários no sistema.
- Módulo para o *usuário*: responsável por permitir a consulta *online* de algumas informações do registro de dívidas:
  1. Valor integral
  2. Valor parcial
  3. Data de Vencimento
  4. Código de Barras

O valor parcial será obtido em função da quantidade de *usuário* associados ao grupo que a dívida pertence e a ponderação por usuário. Por *default* a ponderação será definida como unitária para todos os *usuários*, isto é, distribuição igualitária.

### 1.2.2 Missão do produto

O **DIVIDEA.Í** tem por missão garantir a transparência e o controle de dívidas de grupos de *usuários*, assegurando a consulta das informações de forma simples e prática. Implementado com tecnologia do estado na arte na época do desenvolvimento.

### 1.2.3 Limites do produto

- A. Devido ao exíguo prazo definido para o desenvolvimento e entrega do sistema, não será contemplada a funcionalidade para pagamentos, cabendo aos *usuário* realizá-los.
- B. Sobre a mesma condição do item anterior, não será contemplada a funcionalidade para controle de crédito entre membros de um grupo, funcionalidade esta que permitiria o empréstimo entre os membros de um mesmo grupo.
- C. O sistema será operado inicialmente sobre os três módulos principais, sem prejuízo de um processo evolutivo, ante as eventuais necessidades que surgirão no curso de sua utilização.

### 1.2.4 Restrições

Nº	Restrição	Definição
1	Ambiente	O ambiente operacional a ser utilizado é o Android e iOS.
2	Expansibilidade	O produto deve ser desenvolvido levando-se em consideração que poderá ser expandido com novas funcionalidades.
3	Legal	O produto deverá estar de acordo com as leis e regulamentos vigentes na época da homologação.
4	Segurança	O produto deverá restringir o acesso através de senhas individuais para cada usuário.

Table 1.1: Restrições

### 1.2.5 Benefícios esperados do produto

Os benefícios esperados do produto são apresentados na Tabela 1.2.

Nº	Benefício	Valor para o cliente
1	Consulta transparente das dívidas associadas a um grupo	Essencial
2	Agilidade na identificação de dividendos	Essencial
3	Notificação via e-mail e pelo aplicativo sobre mudanças importantes	Essencial
4	Rateio das dívidas personalizado ou igualitário entre os membros de um grupo	Essencial
5	Banco de créditos entre os membros de um grupo	Opcional

Table 1.2: Benefícios esperados do produto

## 1.3 Definições, Acrônimos e Abreviações

A definição de termos específicos e siglas é listada na Tabela 1.3

Nº	Sigla	Definição
1	PDF	<i>Portable Document Format</i>
2	<i>default</i>	Padrão
3	RF-X	Requisito Funcional número X
4	RNF/PER-X	Requisito Não Funcional de Performance número X
5	RNF/SEG-X	Requisito Não Funcional de Segurança número X
6	RNF/USA-X	Requisito Não Funcional de Usabilidade número X
7	BD	Banco de dados
8	API	<i>Application Programming Interface</i>
9	Migrations	Comandos que executam declarações SQL
10	HTTPS	<i>Hyper Text Transfer Protocol Secure</i>
11	JSON	<i>JavaScript Object Notation</i>

Table 1.3: Definições, acrônimos e abreviações

## 1.4 Estrutura do texto

O restante do texto encontra-se organizado da maneira descrita a seguir.

- No Capítulo 2: Organização dos trabalhos.
- No Capítulo 3: Especificação de requisitos do sistema.
- No Capítulo 4: Arquitetura do sistema.
- No Capítulo 5: Modelos de sistema.
- No Capítulo 6: Evolução do sistema.

## Capítulo 2

# Organização dos trabalhos

### 2.1 Reuniões

Para manter o desenvolvimento das ideias em andamento, foram adotadas reuniões semanais entre os integrantes do grupo com o objetivo de dividir tarefas e organizar cronogramas. As reuniões foram programadas de forma remota por meio do *software* Discord, uma vez que as atividades foram conduzidas em época de pandemia. Desta forma, o grupo fixou o horário das *sprints* às 9h dos domingos.

### 2.2 Ferramenta colaborativa para gestão e controle do projeto

Como ferramenta de gestão do projeto, usamos o Trello. O Trello é um aplicativo disponível em Web para gerenciar e controlar as atividades dos participantes de um projeto.

No Projeto DIVIDEA.Í, usamos a metodologia organizacional Kanban para dividir as atividades e associá-las a membros específicos, enquanto usamos o Scrum como técnica ágil para definir a recorrência das entregas e reuniões.

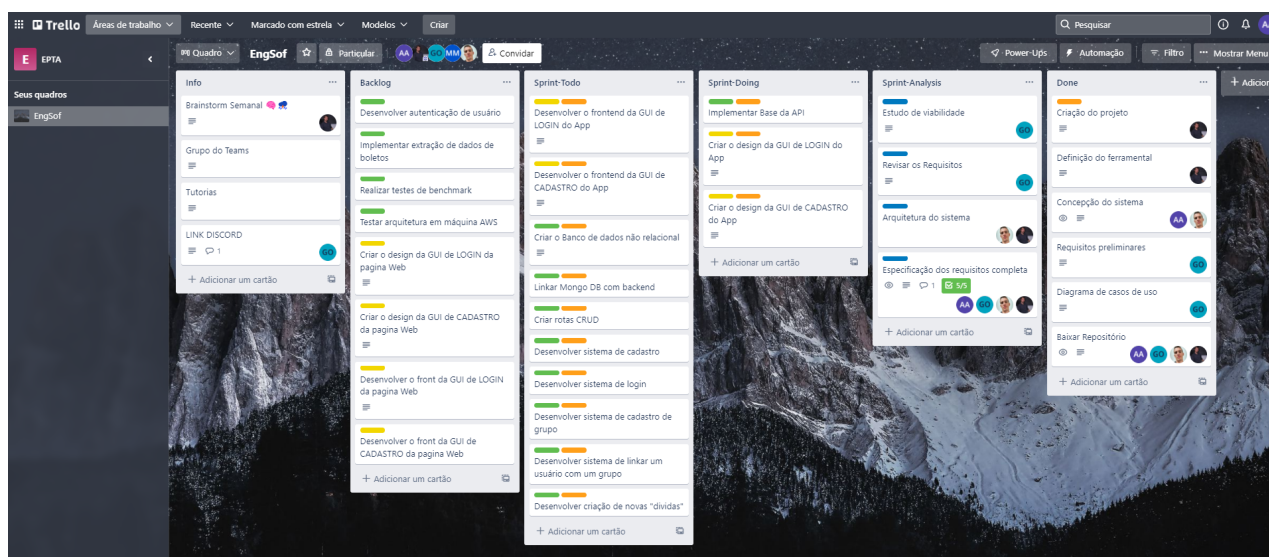


Figure 2.1: Captura de tela demonstrando o ambiente Trello trabalhado

## Capítulo 3

# Especificação de requisitos do sistema

### 3.1 Requisitos Funcionais

Table 3.1: Requisitos Funcionais

ID	Descrições dos Requisitos Funcionais	Prioridade
RF-01	Cadastrar <i>usuários</i> :	Essencial
	a. O sistema disponibilizará a opção de cadastro.	
	b. Ao acessar o cadastro, serão exibidos campos a serem preenchidos pelo <i>usuário</i> , inicialmente seriam,	
	<ul style="list-style-type: none"> <li>• Campo para preencher com o e-mail;</li> <li>• Campo para preencher com a senha;</li> <li>• Campo para preencher com a confirmação da senha.</li> </ul>	
	c. Ao confirmar cadastro o usuário estará registrado no sistema.	
RF-02	Efetuar <i>login</i> de <i>usuários</i> :	Essencial
	a. O sistema disponibilizará a opção de <i>login</i> .	
	b. Ao acessar o <i>login</i> , serão exibidos campos a serem preenchidos pelo <i>usuário</i> , que seriam,	
	<ul style="list-style-type: none"> <li>• Campo para preencher com o e-mail;</li> <li>• Campo para preencher com a senha;</li> <li>• Campo para confirmar para salvar informações de login.</li> </ul>	
	c. Ao preencher os campos com informações já cadastradas no sistema (RF-01) o <i>usuário</i> terá acesso as funcionalidades do sistema.	
	d. Se o campo salvar informações de login, inicialmente previsto para ser uma caixa de texto, for selecionado o <i>usuário</i> efetuará login automático quando acessar o sistema.	

## Editar cadastro:

- RF-03
- a. Efetuar *login* (RF-02);
  - b. Escolher dentre as opções a de configuração do aplicativo;
  - c. Na configuração do aplicativo entre as opções disponíveis escolher a opção de editar informações de cadastro;
  - d. O sistema exibirá a lista de informações de cadastro (RF-01), cabendo ao usuário selecionar qual gostaria de alterar;
  - e. Ao selecionar uma informação três campos serão exibidos,
    - Campo para ser preenchido com a informação atual
    - Campo para ser preenchido com a nova informação
    - Campo para confirmar a alteração
  - f. Ao confirmar a alteração a informação atual será substituída pela nova informação.
- Essencial

## Excluir cadastro:

- RF-04
- a. Efetuar *login* (RF-02);
  - b. Escolher dentre as opções a de configuração do aplicativo;
  - c. Na configuração do aplicativo entre as opções disponíveis escolher a opção de excluir o cadastro;
  - d. O sistema exibirá dois campos,
    - Campo para ser preenchido com a senha atual
    - Campo para confirmar a ação
  - e. Ao confirmar, o *usuário* deletará seu cadastro do sistema.
- Essencial

## Criar grupo:

- RF-05
- a. Efetuar *login* (RF-02);
  - b. Escolher dentre as opções a de criar grupo;
  - c. O sistema exibirá dois campos,
    - Campo para preencher com o nome do grupo.
    - Campo para preencher com breve descrição sobre o grupo (opcional).
  - d. Ao confirmar, o *usuário* criará um grupo.
- Essencial



Editar grupo:

- |       |   |           |
|-------|---|-----------|
| RF-06 | <ul style="list-style-type: none"> <li>a. Efetuar <i>login</i> (RF-02);</li> <li>b. Escolher dentre as opções a de mostrar grupos;</li> <li>c. Escolher um grupo alvo;</li> <li>d. No grupo, selecionar a opção de configuração do grupo;</li> <li>e. Dentre as opções da configuração escolher a de modificar grupo;</li> <li>f. O sistema exibirá a lista de informações fornecidas na criação de grupo (RF-05), cabendo ao usuário selecionar qual gostaria de alterar;</li> <li>g. Ao selecionar uma informação dois campos serão exibidos,               <ul style="list-style-type: none"> <li>• Campo para ser preenchido com a nova informação</li> <li>• Campo para confirmar a alteração</li> </ul> </li> <li>h. Ao confirmar a alteração a informação alvo será substituída pela nova informação.</li> </ul> | Essencial |
|-------|---|-----------|

Excluir grupo:

- |       |  |           |
|-------|--|-----------|
| RF-07 | <ul style="list-style-type: none"> <li>a. Efetuar <i>login</i> (RF-02);</li> <li>b. Escolher dentre as opções a de mostrar grupos;</li> <li>c. Escolher um grupo alvo;</li> <li>d. No grupo, selecionar a opção de configuração do grupo;</li> <li>e. Dentre as opções da configuração escolher a de deletar grupo;</li> <li>f. Desde que o usuário seja dono do grupo, o sistema exibirá dois campos,               <ul style="list-style-type: none"> <li>• Campo para preencher com a senha;</li> <li>• Campo para confirmar a ação</li> </ul> </li> <li>g. Ao confirmar, o grupo será deletado.</li> </ul> | Essencial |
|-------|--|-----------|

Convidar para grupo:

- |       |   |           |
|-------|---|-----------|
| RF-08 | <ul style="list-style-type: none"> <li>a. Efetuar <i>login</i> (RF-02);</li> <li>b. Escolher dentre as opções a de mostrar grupos;</li> <li>c. Escolher um grupo alvo;</li> <li>d. No grupo, selecionar a opção de configuração do grupo;</li> <li>e. Dentre as opções da configuração escolher a de convidar para o grupo;</li> <li>f. O sistema exibirá dois campos,               <ul style="list-style-type: none"> <li>• Campo para ser preenchido com o e-mail de cadastro do <i>usuário</i> (RF-01) a ser convidado</li> <li>• Campo para confirmar o convite</li> </ul> </li> <li>g. Ao confirmar, um convite será enviado ao <i>usuário</i> alvo.</li> </ul> | Essencial |
|-------|---|-----------|

## Ingressar em grupo:

- |       |   |           |
|-------|---|-----------|
| RF-09 | <ul style="list-style-type: none"><li>a. Efetuar <i>login</i> (RF-02);</li><li>b. Escolher dentre as opções a de convites;</li><li>c. Escolher um convite;</li><li>d. O sistema exibirá dois campos,<ul style="list-style-type: none"><li>• Campo destinado a aceitar o convite</li><li>• Campo destinado a rejeitar o convite</li></ul></li><li>e. Ao aceitar, o <i>usuário</i> ingressará no grupo.</li><li>f. Ao rejeitar, o <i>usuário</i> não ingressará no grupo.</li></ul> | Essencial |
|-------|---|-----------|

## Agregar dívida:

- |       |  |           |
|-------|--|-----------|
| RF-10 | <ul style="list-style-type: none"><li>a. Efetuar <i>login</i> (RF-02);</li><li>b. Escolher dentre as opções a de mostrar grupos;</li><li>c. Escolher um grupo alvo;</li><li>d. No grupo, selecionar a opção de integrar dívida;</li><li>e. O sistema solicitará para que o <i>usuário</i> escolha a dívida em formato PDF;</li><li>f. Agregada a dívida ao grupo as seguintes informações estarão disponíveis,<ul style="list-style-type: none"><li>• Valor;</li><li>• Data de vencimento;</li><li>• Código de barras.</li></ul></li></ul> | Essencial |
|-------|--|-----------|

## Visualizar dívida:

- |       |   |           |
|-------|---|-----------|
| RF-11 | <ul style="list-style-type: none"><li>a. Efetuar <i>login</i> (RF-02);</li><li>b. Escolher dentre as opções a de mostrar grupos;</li><li>c. Escolher um grupo alvo;</li><li>d. No grupo, selecionar a opção de visualizar dívidas;</li><li>e. O sistema irá expor todas as dívidas associadas a aquele grupo.</li><li>f. O usuário, para cada dívida, poderá ver informações como<ul style="list-style-type: none"><li>• Valor;</li><li>• Data de vencimento;</li><li>• Código de barras.</li></ul></li></ul> | Essencial |
|-------|---|-----------|

Alterar estado de dívida:

- |       |   |           |
|-------|---|-----------|
| RF-12 | <ul style="list-style-type: none"><li>a. Efetuar <i>login</i> (RF-02);</li><li>b. Escolher dentre as opções a de mostrar grupos;</li><li>c. Escolher um grupo alvo;</li><li>d. No grupo, selecionar a opção de visualizar dívidas;</li><li>e. O sistema irá expor todas as dívidas associadas a aquele grupo.</li><li>f. Escolher uma dívida alvo;</li><li>g. Dentre as opções escolher a de alterar estado da dívida;</li><li>h. O sistema apresentará os possíveis estados, cabendo ao usuário selecionar para qual deseja alterar.</li></ul> | Essencial |
|-------|---|-----------|

Notificar *usuários*

- |       |  |           |
|-------|--|-----------|
| RF-13 | <ul style="list-style-type: none"><li>a. As notificações serão realizadas via e-mail e pelo próprio aplicativo.</li><li>b. Os itens:<ul style="list-style-type: none"><li>• RF-01</li><li>• RF-02</li><li>• RF-03</li><li>• RF-04</li></ul></li><li>geram notificação individual, ou seja, apenas para o <i>usuário</i> atuador.</li><li>c. Os itens:<ul style="list-style-type: none"><li>• RF-06</li><li>• RF-07</li><li>• RF-08</li><li>• RF-09</li><li>• RF-10</li><li>• RF-12</li></ul></li><li>geram notificação para todos <i>usuários</i> comuns a um grupo.</li></ul> | Essencial |
|-------|--|-----------|

### 3.1.1 Diagrama de Casos e Usos

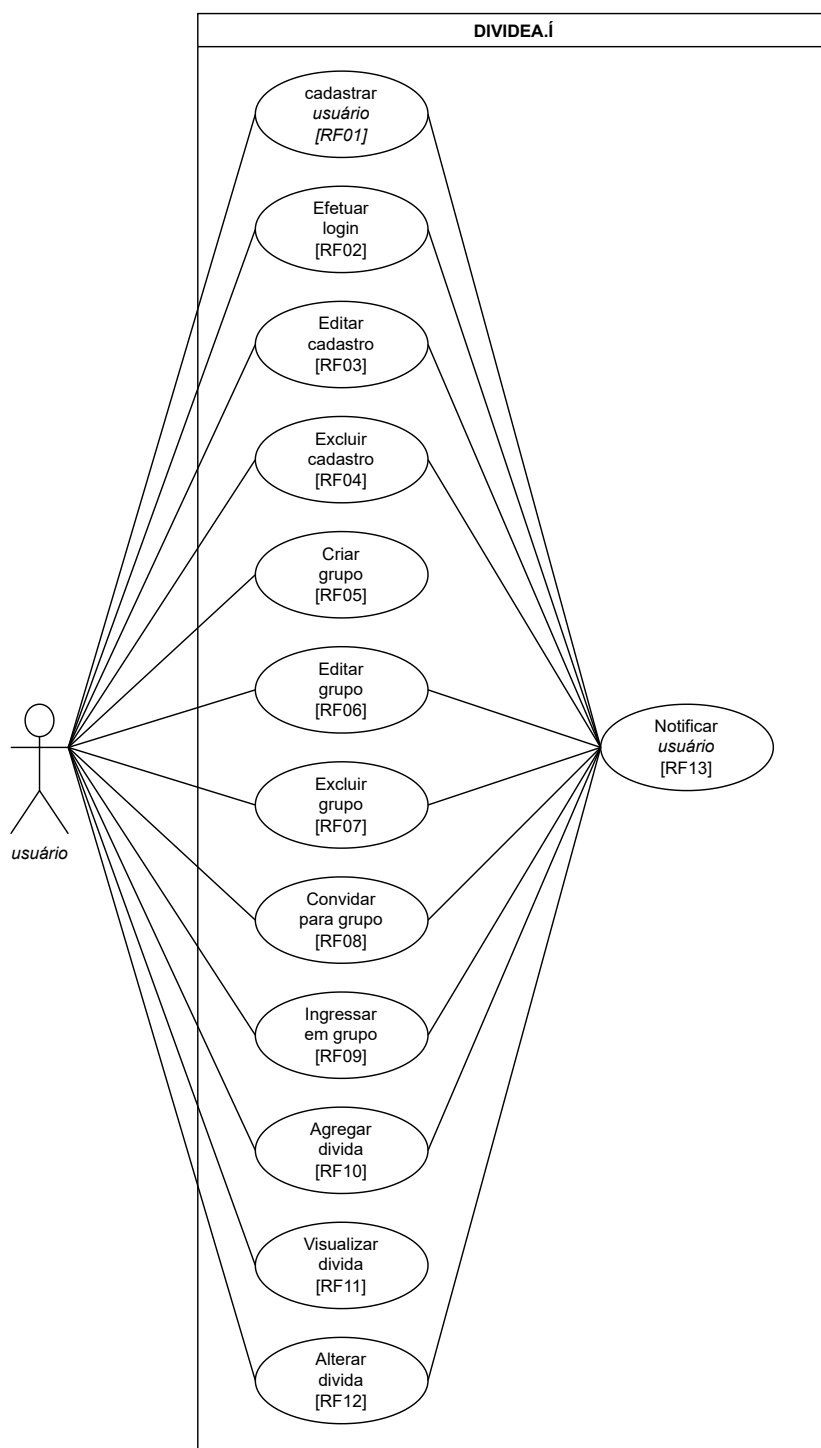


Figure 3.1: Diagrama de casos e usos

## 3.2 Requisitos não Funcionais

### 3.2.1 Requisitos de Performance

Table 3.2: Requisitos de Performance

ID	Descrição
RNF/PER-01	O sistema deve usar os recursos disponibilizados pelos aparelhos móveis de forma exata às suas necessidades, não desperdiçando a utilização de nenhum destes recursos, evitando o desperdício de informações e aumentando a autonomia da bateria do aparelho.
RNF/PER-02	O sistema deve ter um limite na obtenção das respostas a serem fornecidas ao usuário dependendo da operação requisitada, de forma que o usuário não fique esperando infinitamente por uma opção e seja corretamente informado quando alguma operação está demorando mais do que o previsto ou estourou seu tempo de limite.
RNF/PER-03	A aplicação deve tratar erros inerentes ao sistema em que está rodando, como problemas na conexão, eventual fechamento da aplicação por falta de memória, falta de espaço local para armazenamento de dados, entre outros, de forma que seu funcionamento não seja prejudicado ou, em última instância, que o usuário seja avisado do porquê do não funcionamento da aplicação.
RNF/PER-04	A aplicação deve utilizar somente o necessário da conexão de dados provida pelo aparelho móvel, de forma a não ocupar desnecessariamente recursos compartilhados com outras aplicações.
RNF/PER-05	O usuário deve ter retorno visual claro da demora de suas operações, de modo que saiba estimar quanto tempo cada uma delas demorará a atender as suas necessidades.

### 3.2.2 Requisitos de Segurança

Table 3.3: Requisitos de Segurança

ID	Descrição
RNF/SEG-01	O sistema deverá oferecer segurança e privacidade, de modo que só usuários autorizados possam acessar as informações.

### 3.2.3 Requisitos de Usabilidade

Table 3.4: Requisitos de Usabilidade

ID	Descrição
RNF/USA-01	O sistema deve prover suas funcionalidades ao usuário da maneira mais usual possível, de modo que em poucas interações e de forma intuitiva o usuário consiga encontrar as informações que deseja.

## Capítulo 4

# Arquitetura do Sistema

O diagrama de componentes da Figura 4.1, representa a arquitetura do sistema. O projeto foi dividido em 3 grandes componentes o *front-end*, o *back-end* e o *database*. O *database* utilizará de um banco não relacional chamado MongoDB®, este conterá o cadastro dos usuários, dos grupos e das contas de cada grupo. O *back-end* será responsável por todos os recursos de funcionamento da aplicação tais:

- Gestão dos usuários na aplicação
- Gestão dos grupos criados pelos usuários
- Gestão do banco de dados através das *Migrations*
- Gestão API à qual o *front-end* vai se comunicar
- Gestão da funcionalidades extras, tal o sistema de visão computacional capaz de ler e processar boletos

O *front-end* por sua vez será responsável por renderizar as telas e páginas utilizando o *Flutter*. O protocolo de comunicação entre o *front-end*, o *back-end* e o *database* será o HTTPS.

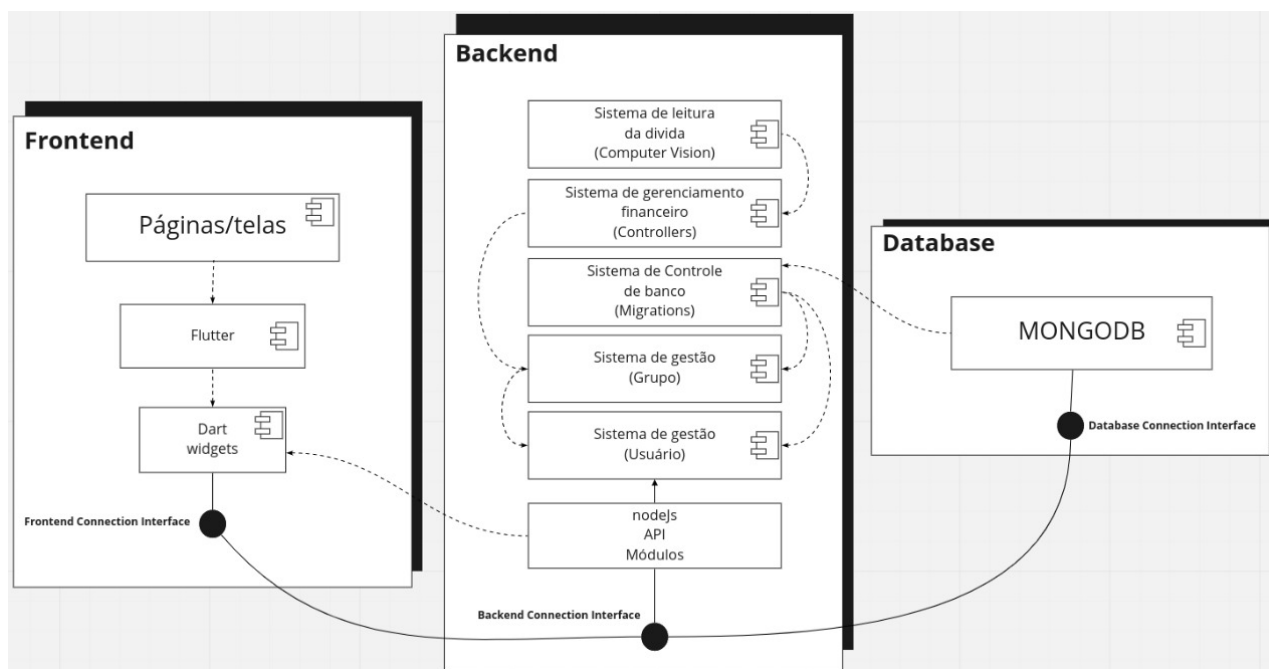


Figure 4.1: Diagrama de Componentes

## 4.1 Definição do ferramental

### 4.1.1 NodeJs

O NodeJs surgiu como um interpretador do JavaScript sem necessidade de um *browser*. O NodeJs é uma excelente ferramenta para aplicações que buscam escalabilidade e estabilidade, além disso devido aos seus *frameworks* se torna algo poderoso para desenvolvimento de APIs. Segue as principais vantagens do uso do Node no *back-end*:

- Fácil integração com bancos de dados (Relacionais e não relacionais);
- Comunicação com APIs e *webservices*;
- Adequado para Criptografia e Segurança;
- Facilidade em realizar Autenticação de usuário;
- Comunicação via json com com *front-end* e outros serviços.

Grandes *players* que usam NodeJs:

- Uber;
- Netflix;
- LinkedIn;
- Nasa;
- Walmart.

### 4.1.2 Express

O NodeJs por si só daria conta de gerir uma API para a aplicação desejada, entretanto, existem *frameworks* que facilitam neste desafio que é o caso do *microframework* chamado Express.

O Express é usado para se ter uma aplicação *web* fornecendo ferramentas e utilitários HTTP e *middlewares*. Existem outras ferramentas parecidas no mercado mas o Express conta com a maior documentação e uma ampla comunidade de apoiadores.

### 4.1.3 MongoDB

Para este projeto foi pensado em um banco de dados não relacional justamente pela flexibilidade de armazenar os dados sem usar uma estrutura específica, mas usando o formato JSON. Partindo do ponto que para a aplicação que será desenvolvida serão armazenados os dados de usuários, grupos, dívidas entre outras coisas então fica mais simples estruturar esses modelos com a junção NodeJs + MongoDB.

### 4.1.4 Flutter

Flutter é um *framework* para o desenvolvimento de aplicativos mobile para Android e iOS. Diversas são as características do Flutter. Dentre elas podemos citar:

- Multiplataforma - Podemos desenvolver aplicações com Flutter em qualquer sistema operacional (Windows, Linux e MacOS);

- Criação de aplicações nativas a partir de um único código base - Com o Flutter é possível desenvolvermos aplicações nativas para Android e iOS;
- Acesso direto aos recursos nativos do sistema - Uma aplicação criada com Flutter possui acesso nativo aos recursos do dispositivo (câmera, wifi, memória, etc);;
- Maior desempenho - As aplicações criadas com Flutter possuem um maior desempenho quando comparadas ao React Native, por exemplo, pois todo seu código-fonte é transformado em código nativo.

Grandes empresas vêm apostando no Flutter, como: Google, Alibaba, Groupon, Nubank, Google entre outras. Um dos grandes motivos para esta aposta é justamente a forma com que o Flutter funciona para criação de aplicações.

#### 4.1.5 Docker

O Docker é uma ferramenta que faz uso de contêineres para construir, gerenciar, executar e distribuir aplicações. Nós optamos por utilizá-lo para que assim fosse possível executar nosso aplicativo (e, caso fosse um aplicativo que fosse comercializado, seria útil para distribuí-lo) e suas dependências em um ambiente isolado, garantindo assim uma maior estabilidade.