



RELATÓRIO – PROJETO INTEGRADO III MAISELF

Equipe:

Eric Rodrigues - 470296

Gustavo Colombo - 470168

Ítalo Lima - 470651

Marcos Gênesis - 471549

Professor:

Camilo Camilo Almendra

QUIXADÁ

Outubro, 2021

SUMÁRIO

1	INTRODUÇÃO	2
2	SISTEMAS RELACIONADOS	2-6
2.1	Evernote	2-3
2.2	7 Weeks	3-4
2.3	Notion	4-5
2.4	Mobilis	5-6
3	DESCRIÇÃO DO SISTEMA	6
3.1	Diferencial do Nosso Sistema	7
4	REQUISITOS	8-23
4.1	Histórias do Usuário	8-15
4.2	Requisitos Não Funcionais	16-24
5	PROTOTIPAÇÃO DO PROJETO	25
6	REPOSITÓRIOS DE CÓDIGO E TRACKER DE TASKS	26
6.1	Repositório do Back-End	26
6.2	Repositório do App Mobile	26
6.3	Tracker de Tasks	26

1 INTRODUÇÃO

Um tema que sempre cerca as pesquisas periféricas dos adolescentes e jovens adultos nos últimos anos é o mundo do desenvolvimento pessoal. Autores como Jordan Patterson, David Goggins e Tony Robbins inflamaram esse tema com seus livros lançados nos últimos 3 anos. Observando o mercado de aplicações mobile voltadas a esse tema, é possível notar uma situação curiosa: a grande maioria destes, tem como foco sempre uma subárea desse âmbito. Por exemplo, um subtema cujo existem bastantes opções no mercado de aplicativos é o de controle financeiro, com grandes nomes no mercado como Mobbils, Organizze, entre outros. Para o gerenciamento de hábitos também existem diversas opções, assim como para a organização de tarefas e acompanhamento rotineiro, como Todoist, Notion, Trello e etc. No entanto, é nítido o espaço existente para uma aplicação que centralize esses dados em uma só plataforma, assim como o Rappi fez para o mercado de delivery no Brasil e na América Latina. Observando essa oportunidade, decidimos então criar a Maiself - Alusão a palavra *myself* em inglês, que pode significar “eu”, com uma personalização para o nosso dialeto “cearensês”.

De acordo com James Clear, autor de Atomic Habits, um dos mais best sellers mais vendidos do New York Times, existem 5 tópicos que compõem uma estratégia para a criação de um novo hábito, são esses: Comece com algo incrivelmente pequeno; aumente seu hábito em pequenos passos; conforme à construção; divida seu hábito em partes, quando você falhar, volte ao normal rapidamente e seja paciente, mantenha um ritmo sustentável. Baseado nesses e em outros estudos científicos, nós, primeiramente, organizamos ideias que customizam o desenvolvimento pessoal, baseado nos principais componentes das nossas vidas, como se fossemos um projeto, em constante execução, constituído de involuções e evoluções, no entanto, personificando ainda mais o cenário de desenvolvimento pessoal moldado por analogias, aprimoramos essa ideia inicial e reformulamos o âmbito, devido ao fato de que projetos podem ser relacionados com prazos, burocracia, impedimentos, tarefas exaustivas e etc, o que não caracteriza a nossa intenção, baseando-se no propósito de que cada pessoa possui um DNA inteiro particular, e que mesmo gêmeos possuem material genético distinto, reconhecemos cada indivíduo como um universo à parte, que nasce como uma pequena estrela, mas que pode se tornar uma galáxia, ou se perder em meio confusão, como um buraco negro.

O projeto Maiself visa edificar uma base centralizada de informações sobre a vida de cada usuário, baseado nos registros realizados por ele, para que assim seja possível visualizar de maneira mais clara, objetiva e organizada o processo de desenvolvimento pessoal trilhado por cada um, respeitando o seu próprio tempo, visando uma vida mais organizada, produtiva e feliz.

2 SISTEMAS RELACIONADOS

Nesta seção serão descritos 3 sistemas que encontramos em pesquisa e disponíveis na internet, que se aproximam ao máximo do domínio do projeto com objetivos do sistema, possíveis funcionalidades e tecnologias.

1.1 Evernote

Sistema para anotação de tarefas/notas: Nesse sistema, o usuário pode escrever suas notas sem ficar restrito ao acesso desktop, pode baixar uma versão mobile da solução e acessar o que escreveu e ainda anotar o que desejar, fora navegar entre blogs, sites, guardar documentos.

Funcionalidades:

1. **CRUD notas/anotações.**
2. **Cadastro de lembretes dos itens anotados.**
3. **Cadastro de etiquetas dos itens anotados.**
4. **Criar um ‘caderno’ para criar notas - deixar mais organizado suas notas.**
5. **Upload de anexos a notas como imagens, vídeos, áudios.**
6. **Criar listas**

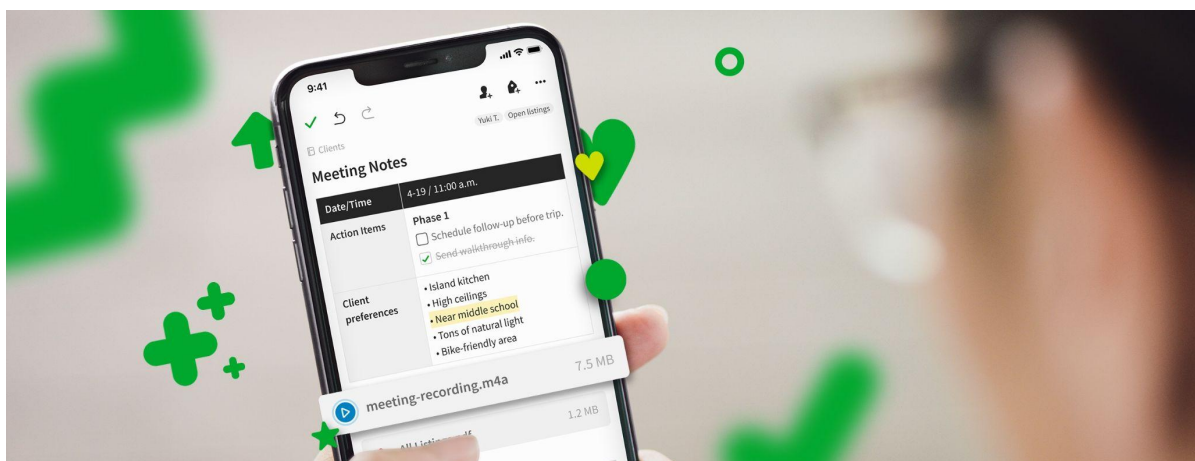


Figura 1 – Tela do Evernote
Fonte: [Link para o Evernote](#)

1.2 7 Weeks:

Aplicativo para gerenciamento/criação de novos hábitos: O aplicativo disponível de forma gratuita para Android é para a criação de novos hábitos, atos que são repetidos até que tornam-se regra na rotina da pessoa. Estudiosos dizem que para a criação de um novo hábito temos que incorporá-lo durante 49 dias (7 semanas) até que ele seja costume. O aplicativo consiste em criar um novo hábito, o motivo por que você está criando ele e quando pretende terminá-lo e partir disso o aplicativo o mantém informado do progresso.

Funcionalidades:

1. Cadastro do hábito.
2. Cadastro da justificativa do hábito.
3. Cadastro da data limite do hábito para incorporá-lo
4. Visualização da estatística do progresso no hábito, com porcentagem, a partir dos dias feitos, da data limite e dos dias que o usuário não fez
5. Emitir alarmes para realizar o ato do hábito
6. Marcar os dias concluídos no calendário de realização do hábito

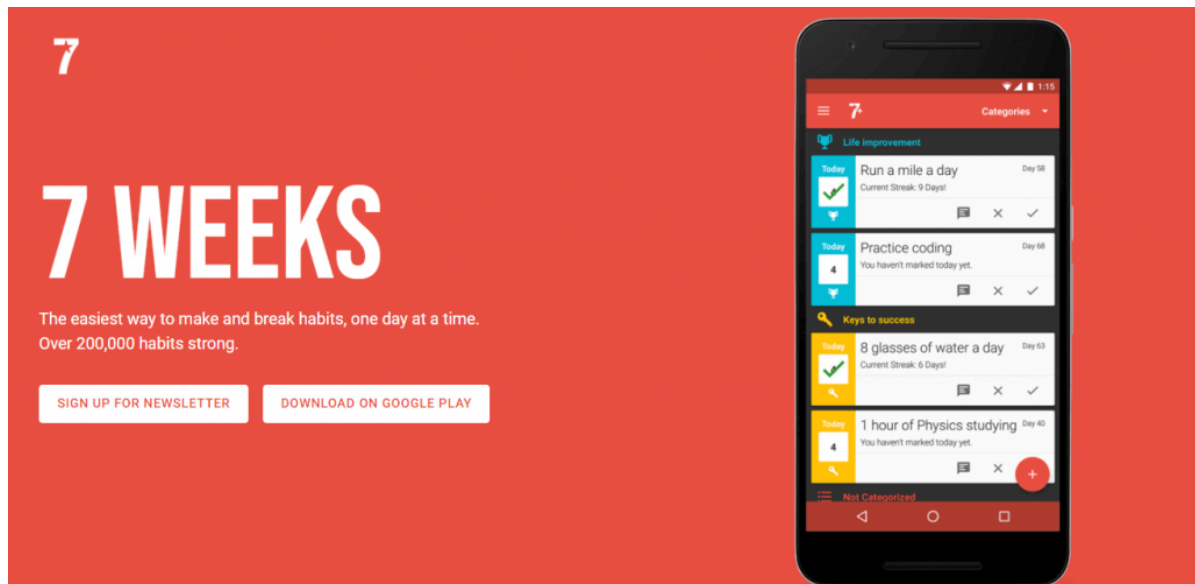


Figura 2 – Tela inicial do aplicativo 7 weeks.
Fonte: [Link para o 7 Weeks](#)

1.3 Notion

Sistema/aplicativo para anotações em geral: Esse sistema, bastante popular entre os programadores é disponível em forma de site, aplicativo móvel e aplicação desktop, ele serve para o gerenciamento e armazenamento de notas, textos em geral, basicamente tudo o que você anota está guardado pra sempre em infinitas páginas e em quantos workspaces quiser.

Uma das coisas que o torna tão incrível é a sua alta integração com diversos meios de escrita, onde o mesmo dispõe integração de escrita com o *Markdown*, *LáTeX* e etc, tudo isso dentro da própria ferramenta. A mesma também dispõe de integração com o calendário, quadros *Kanban* e etc.

Funcionalidades:

1. **CRUD de workspaces.**
2. **CRUD de páginas dentro dos workspaces**
3. **CRUD de páginas dentro das páginas.**
4. **Anotações, listas, to-list, bloco de notas e muito mais podem ser cadastrados nas páginas.**
5. **Anexo de imagens as páginas, menções a outras páginas.**
6. **Compartilhar páginas com a web ou com quem apenas quiser**

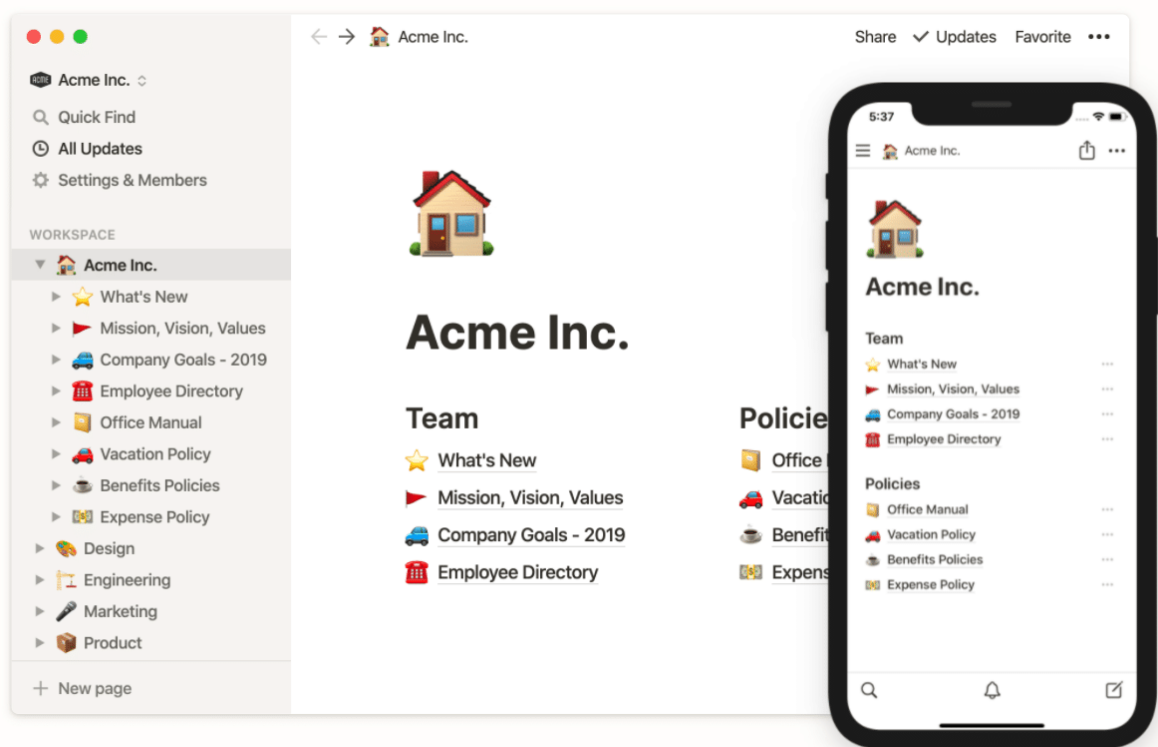


Figura 3 – Tela do Sistema e aplicativo Notion.

Fonte: [Link para o Notion](#)

Vídeo demonstrativo: [Demonstração Notion](#)

1.4 Mobills

Sistema/aplicativo para gerenciamento e organização financeira: Mobills é o aplicativo de gerenciamento de custos pessoais mais bem avaliado do país, mas com também disponibilidade no sistema web. O aplicativo é muito útil para quem deseja ter um maior controle de gastos, para quem deseja estabelecer teto de gastos ou apenas controlar seus

gastos, com o montante que entra subdividido em categorias e o valor de cada entrada e o montante que sai subdividido em categorias também e com valores resultando no valor total da carteira.

Funcionalidades:

1. Cadastrar Transações.
2. Classificar Transações como entrada ou saída
3. Classificar Transações de entrada/saída em categorias pré-prontas.
4. Nomear categorias para Transações.
5. Dar valor inicial a carteira todo mês para ter o balanço no final do mês.
6. Criar objetivos para alcançar com a economia de dinheiro.



Figura 4 – Tela do Sistema Mobills.

Fonte: [Link para a Mobills](#)

3 DESCRIÇÃO DO SISTEMA

O aplicativo proposto pela nossa equipe é denominado de **Maiself** e propõe como solução principal a integração de funcionalidades divididas em distintos módulos e presente em variadas aplicações no mercado, ou seja, a proposta é fornecer ao usuário final um aplicativo móvel que o proporcione o total controle das principais áreas da sua vida. através do gerenciamento dos aspectos que as compõem, como a organização financeira para o domínio das finanças, o acompanhamento dos hábitos para a rotina

individual, a utilização de ferramentas para a automatização e apoio nas atividades do seu dia-a-dia, registros referentes à evolução e cuidado com a sua saúde e auxiliares para memorização, controle e documentação de tarefas, compromissos e lembretes, através de um histórico completo que funciona como um diagnóstico de atividades, no qual pode ser visto em diferentes perspectivas, textuais e gráficas, tudo isso garante uma integridade dos “dados” resultantes de tudo que você faz, todos os dias, desde aquela pedalada na terça-feira pela manhã, passando pelo check-list de atividades diárias que você tem a cumprir no trabalho, até o acompanhamento do seu IMC no último período de 6 meses, que pode indicar que você perdeu peso, quando decidiu economizar nos lanches noturnos que costumava pedir na sua lanchonete preferida, todas as nossas ações têm como resultado (direto ou indireto) a produção de dados e ter o versionamento, a visualização, o registro e a aquisição deles pode ser crucial para aumentar a eficiência na busca pelo controle e equilíbrio da própria existência.

4 Nosso Diferencial

Como anteriormente citado, nosso objetivo tem como unir diferentes módulos de organização pessoal em uma única aplicação web, esse é um dos nossos diferenciais dos sistemas apresentados anteriormente também. Outro diferencial é que para soluções web não há sistemas de controle de hábitos ou de controle de finanças(não de graça pelo menos), então estamos trazendo dois segmentos de controle pessoal que quase toda pessoa hoje em dia no mundo da tecnologia visa, podendo dentro do sistema navegar entre esses módulos e usar qual quiser totalmente de graça.

5 REQUISITOS

Requisitos de software podem ser descritos brevemente como a interseção entre uma necessidade real de alguém e a documentação estratégica que possibilita a implementação da funcionalidade. O processo começa sempre com um problema, uma necessidade, uma ideia ou até mesmo uma melhoria necessária, alguém tem isso definido no campo da ideia e um dia, “magicamente” tal pensamento vira parte de artefatos produzido pelo processo de desenvolvimento de software, seja de forma documental ou em uma linguagem de programação, por isso é tão importante entender e buscar realizar com assertividade as operações referentes à requisitos.

Aqui estão dispostos e documentados os Requisitos do Maiself, sendo eles divididos entre funcionais (seguindo o formato supracitado) e não-funcionais, com a expectativa de estarem todos idealizados e descritos da forma mais concisa possível, visando agregar o maior valor possível as partes interessadas.

5.1 Histórias de Usuário

Em desenvolvimento de software e gerenciamento de produto, uma História de Usuário (User Stories) é uma especificação de uma ou mais sentenças na linguagem de negócio ou cotidiana do usuário final ou usuário do sistema que captura o que um usuário faz ou necessita fazer como parte de sua função de trabalho. Histórias de usuário

são usadas com metodologias ágeis de desenvolvimento de software como a base para definir o escopo de um projeto de software. É uma técnica de análise de requisitos. Ela captura o "quem", "o quê" e "por quê" de um requisito em uma forma concisa e simples, geralmente limitada em detalhes, de forma que possa ser escrita a mão em um pequeno cartão de notas de papel.

[US01] - O usuário poderá realizar seu cadastro no aplicativo.

Cartão	Eu como usuário comum, quero me cadastrar no sistema passando meus dados pessoais para que possa utilizar os recursos do <i>app</i> .
Tag	Usuário
Plano de Usuário	Freemium, Premium

[US02] - O usuário poderá efetuar login no aplicativo, utilizando os dados de Email e Senha.

Cartão	Eu como usuário comum, quero realizar login no app utilizando email e senha, para que possa entrar no aplicativo.
Tag	Usuário
Plano de Usuário	Freemium, Premium

[US03] - O usuário poderá editar suas informações pessoais no sistema.

Cartão	Eu como usuário comum, quero editar minhas informações pessoais no meu perfil, para que os dados fiquem atualizados/integros.
Tag	Usuário
Plano de Usuário	Freemium, Premium

[US04] - O usuário poderá excluir sua conta a partir do momento que não estiver mais satisfeito com o site ou desejar apenas excluir a conta.

Cartão	Eu como usuário comum, quero poder excluir minha conta, para que meus dados não existam mais na base de dados.
Tag	Usuário
Plano de Usuário	Freemium, Premium

[US05] - O usuário deverá ser capaz de cadastrar os seus hábitos no aplicativo.

Cartão	Eu como usuário comum, quero definir e cadastrar meus hábitos no aplicativo, para que possa ter acompanhamento visual e gradativo da minha evolução.
Tag	Hábitos
Plano de Usuário	Freemium, Premium

[US06] - O usuário deverá ser capaz de visualizar a listagem de todos os seus hábitos definidos no aplicativo.

Cartão	Eu como usuário comum, quero acessar a listagem de todos os meus hábitos cadastrados no aplicativo, para que possa acompanhar se e como estou.
Tag	Hábitos
Plano de Usuário	Freemium, Premium

[US07] - O usuário deverá ser capaz de editar os seus hábitos cadastrados na plataforma.

Cartão	Eu como usuário comum, quero acessar um hábito que tenho cadastrado e editar as informações dele para que as informações possam ser atualizadas ou o escopo seja atualizado.
Tag	Hábitos
Plano de Usuário	Freemium, Premium

[US08] - O usuário poderá priorizar os seus hábitos definidos.

Cartão	Eu como usuário comum, quero estabelecer uma relação de prioridade entre os hábitos que criei, para que possa dar mais importância a aqueles que realmente mais necessitam.
Tag	Hábitos
Plano de Usuário	Freemium, Premium

[US09] - O usuário poderá cadastrar lembretes para os hábitos que definiu no aplicativo.

Cartão	Eu como usuário comum, quero definir lembretes dentro do aplicativo para que o mesmo me alertar acerca de realizá-los.
Tag	Hábitos
Plano de Usuário	Freemium, Premium

[US10] - O usuário poderá cadastrar informações referentes ao seu corpo.

Cartão	Eu como usuário comum, quero realizar o cadastro de informações do meu corpo, para que haja cálculos em determinados hábitos de determinado módulo, o de Hábitos.
Tag	Hábitos
Plano de Usuário	Freemium, Premium

[US11] - O usuário poderá verificar o cálculo do IMC com os seus dados registrados de Peso e Altura.

Cartão	Eu como usuário comum, quero realizar o cálculo de IMC no módulo de Hábitos seguindo as normas da OMS, para que possa acompanhar como está meu corpo em relação ao cumprimento de hábitos
Tag	Hábitos
Plano de Usuário	Freemium, Premium

[US12] - O usuário de acordo com sua assiduidade na realização dos hábitos deverá poder visualizar um gráfico que ilustre a sua estabilidade, de acordo com a fórmula definida pelas regras de negócio.

Cartão	Eu como usuário comum, quero obter um <i>heatmap</i> e gráfico de frequência de de assiduidade dado a medida que eu realizo as tarefas de hábito, para que possa acompanhar minha evolução.
Tag	Hábitos
Plano de Usuário	Freemium, Premium

[US13] - O usuário poderá ser notificado diariamente sobre o cumprimento de seus hábitos.

Cartão	Eu como usuário comum, quero que o aplicativo me notifique diariamente a respeito dos meus hábitos, para que eu possa me policiar melhor sobre o cumprimento de cada um.
Tag	Hábitos
Plano de Usuário	Freemium, Premium

[US14] - O usuário poderá cadastrar montantes de dinheiro na categoria de Entrada

Cartão	Eu como usuário comum, quero cadastrar montantes de dinheiro na categoria entrada no módulo de Financeiro, para que possa ter controle do dinheiro que entra na minha conta.
Tag	Financeiro
Plano de Usuário	Premium

[US15] - O usuário poderá cadastrar montantes de dinheiro na categoria de Saída

Cartão	Eu como usuário comum, quero cadastrar montantes de dinheiro na categoria saída no módulo de Financeiro, para que possa ter controle do dinheiro que sai na minha conta.
Tag	Financeiro
Plano de Usuário	Premium

[US16] - O usuário poderá editar o valor das Entradas financeiras cadastradas

Cartão	Eu como usuário comum, quero editar o valor de uma entrada financeira no módulo financeiro para que o valor fique atualizado ou correto
Tag	Financeiro
Plano de Usuário	Premium

[US17] - O usuário poderá editar o valor das Saídas financeiras cadastradas

Cartão	Eu como usuário comum, quero editar o valor de uma saída financeira no módulo financeiro para que o valor fique atualizado ou correto
Tag	Financeiro
Plano de Usuário	Premium

[US18] - O usuário poderá remover uma Entrada financeira cadastrada

Cartão	Eu como usuário comum, quero remover uma entrada financeira cadastrada para que o valor do balanço final seja atualizado
Tag	Financeiro
Plano de Usuário	Premium

[US19] - O usuário poderá remover uma Saída financeira cadastrada

Cartão	Eu como usuário comum, quero remover uma saída financeira cadastrada para que o valor do balanço final seja atualizado
Tag	Financeiro
Plano de Usuário	Premium

[US20] - O usuário poderá verificar o balanço de suas finanças

Cartão	Eu como usuário comum, quero obter o valor do balanço das finanças subtraindo as entradas das saídas para que possa ter o balanço final seja mensal, trimestral ou semestral
Tag	Financeiro
Plano de Usuário	Premium

[US21] - O usuário poderá ser notificado ao chegar em um ambiente que remeta a construção de um hábito seu.

Cartão	Eu como usuário comum, quero que o aplicativo me notifique toda vez que eu passar por um local que me remeta a construção de um hábito meu, para que eu me lembre da construção do meu hábito. Exemplo: quando ele passa pela praça onde caminha.
Tag	Mobilidade
Plano de Usuário	Freemium, Premium

[US22] - O usuário poderá atualizar a foto do seu perfil utilizando a câmera do próprio aparelho.

Cartão	Eu como usuário comum, quero que quando eu for atualizar meu perfil, tenha a opção de atualizar a foto utilizando a câmera do meu próprio aparelho, para que assim possa pôr uma foto atual no cadastrado.
Tag	Mobilidade
Plano de Usuário	Freemium, Premium

[US23] - O usuário poderá cadastrar junto com um hábito, uma localização que fará parte do hábito.

Cartão	Eu como usuário comum, desejo que quando eu for cadastrar um novo hábito, possa associar ao mesmo, uma localização de prática para o hábito, que servirá como uma caracterização do ambiente da prática.
Tag	Mobilidade

Plano de Usuário	Freemium, Premium
-------------------------	-------------------

[US24] - O usuário poderá adquirir o plano premium fazendo utilização de cartão de crédito.

Cartão	Eu como usuário comum, desejo que quando eu for adquirir o plano premium, possa inserir meu cartão de crédito como forma e meio de pagamento da assinatura.
Tag	Pagamentos
Plano de Usuário	Freemium, Premium

[US25] - O usuário poderá adquirir o plano premium através de assinaturas que englobam períodos de tempo.

Cartão	Eu como usuário comum, desejo que quando eu for adquirir o plano premium, possa fazer a aquisição por período de tempo, exemplo: assinar o plano premium por 2 meses, 3 meses, 6 meses, 1 ano, etc...
Tag	Pagamentos
Plano de Usuário	Freemium, Premium

[US26] - O usuário poderá realizar a rescisão assinatura premium que ele adquiriu, antes do término do período.

Cartão	Eu como usuário comum, após ter adquirido o plano premium desejo poder realizar a rescisão do plano.
Tag	Pagamentos
Plano de Usuário	Freemium, Premium

[US27] - Em caso de rescisão por parte do usuário, antes do término do período referente ao seu plano premium, o mesmo deverá pagar uma multa correspondente.

Cartão	Eu como usuário comum, em caso de rescisão do meu plano antes do término do período referente ao seu plano premium, devo pagar a devida multa de quebra de termos da assinatura, que se caracteriza como um valor proporcional de 25% do restante do período da assinatura.
Tag	Pagamentos
Plano de Usuário	Freemium, Premium

[US28] - O usuário poderá realizar a rescisão assinatura premium que ele adquiriu antes sem multa alguma com no máximo 1 mês após a adesão.

Cartão	Eu, como usuário comum, após ter adquirido o plano premium desejo realizar a rescisão do plano, sem pagar qualquer tipo de multa por quebra de termos de assinatura, no período que compreende no máximo 1 mês a aquisição do plano.
Tag	Pagamentos
Plano de Usuário	Freemium, Premium

[US29] - O usuário poderá chacoalhar seu smartphone, para o aplicativo voltar para a home.

Cartão	Eu, como usuário comum, estando em qualquer ambiente do aplicativo, desejo chacoalhar o aparelho celular para fazer o aplicativo me redirecionar a tela principal do meu celular.
Tag	Mobilidade
Plano de Usuário	Freemium, Premium

4.2 Requisitos Não Funcionais

Um requisito não-funcional é uma descrição ou característica que um sistema deve exibir, ou mesmo uma restrição que deve ser respeitada pelo mesmo.

Requisitos não-funcionais são atributos de qualidade para um software. Estes atributos podem ser classificados de diferentes formas. A abordagem deste trabalho foca em um exemplo de classificação binária, onde os atributos podem ser de qualidade externa ou interna.

“Atributos de qualidade diferenciam os sistemas que simplesmente fazem o suposto daqueles que encantam os seus usuários”

- Os atributos de qualidade externa são mais voltados para o usuário, e influenciam a experiência e percepções do mesmo, sobre a qualidade de um sistema.
 - São normalmente avaliados quando o sistema está em execução.
- Os atributos de qualidade interna são voltados para os desenvolvedores e mantenedores do sistema.
 - São normalmente percebidos nas etapas de design e codificação.

Atributos de qualidade interna impactam os de qualidade externa, por isso, apesar das diferenças, os dois tipos estão conectados e são cruciais para o sucesso de um software, algo que vai além de simplesmente fazer o que o cliente pediu.

GLOSSÁRIO

- **Definição dos atributos**
 - **Requisitos não-funcionais derivados**

Requisitos de Usabilidade

- **É a capacidade de quão facilmente o sistema pode ser compreendido, aprendido e atraente para quem usa, no caso o usuário. Esse RNF não se restringe a interfaces visuais mas a qualquer sistema que tenha comunicação ou interação com o usuário como um terminal do linux por exemplo e se subdivide em outras categorias.**
- **Requisitos Não-Funcionais Derivados:**

[RNF 01] - Os usuários deverão se familiarizar com a interface em pouco tempo (Média de 2 execuções de fluxo).

Identificador	[RNF 01] - Os usuários deverão se familiarizar com a interface em pouco tempo (Média de 2 execuções de fluxo).
Descrição	Os usuários que irão usar o aplicativo Maiself deverão em pouco tempo de uso aprender/compreender o fluxo de execução e como medida de tempo usamos a média de 2 vezes cada execução de fluxo dentro do sistema.

Prioridade	Essencial
-------------------	-----------

Tabela 01 – Tabela do Requisito não Funcional 01.

[RNF 02] - Os usuários não precisarão de treinamento prévio para compreender o fluxo de execução do aplicativo.

Identificador	[RNF 02] - Os usuários não precisarão de treinamento prévio para compreender o fluxo de execução do aplicativo.
Descrição	O objetivo da interface e do sistema é ser o mais simples e descritivo possível, com isso não precisaremos obrigar o usuário a ter conhecimento aprofundado de tecnologia para usar o sistema, apenas seguir um fluxo de execução simples e descritivo.
Prioridade	Essencial

Tabela 02 – Tabela do Requisito não Funcional 02.

[RNF03] - A interface do aplicativo deve possuir design simples e *Flat* para melhor familiarização dos usuários.

Identificador	[RNF03] - A interface do aplicativo deve possuir design simples e <i>Flat</i> para melhor familiarização dos usuários
Descrição	Para que o fluxo de atividades e interface sejam simples e descritivos, por convenção, usaremos o design Flat que busca o design minimalista, priorizando mais a informação em tela e mantendo a simplicidade evitando a poluição visual.
Prioridade	Essencial

Tabela 03 – Tabela do Requisito não Funcional 03.

[RNF 04] - Menus existentes dentro do aplicativo não devem possuir mais que um nível de ramificação interna.

Identificador	[RNF 04] - Menus existentes dentro aplicativo não devem possuir mais que um nível de ramificação interna.
Descrição	Para que não haja poluição visual nem confusão de informações para o usuário, definimos que caso haja menus eles não podem ter mais de um nível de ramificação interna para não poluir a tela de informações

	nem armazenar muitas informações em um só lugar.
Prioridade	Importante

Tabela 04 – Tabela do Requisito não Funcional 04.

Requisitos de Eficiência

- Esse tipo de RNF se refere ao desempenho do sistema desenvolvido, se o tempo de execução/resposta e recursos são compatíveis com o que foi desenvolvido, este também se divide em categorias;
- **Requisitos Não-Funcionais Derivados:**

[RNF05] - O aplicativo deverá tratar dados como Altura e Peso de usuários em ponto flutuante com maior precisão possível de no máximo 2 casas decimais após a vírgula.

Identificador	[RNF05] - O aplicativo deverá tratar dados como Altura e Peso de usuários em ponto flutuante com maior precisão possível de no máximo 2 casas decimais após a vírgula.
Descrição	Os cálculos dentro do aplicativo ou armazenamento de informações que sabe-se que são números não inteiros devem ser guardados do tipo numérico float com precisão de 2 casas decimais.
Prioridade	Essencial

Tabela 05 – Tabela do Requisito não Funcional 05.

[RNF 06] - Os cálculos de IMC de cada usuário deverão ser feitos com base na fórmula disponibilizada pela Organização Mundial da Saúde (OMS).

Identificador	[RNF 06] - Os cálculos de IMC de cada usuário deverão ser feitos com base na fórmula disponibilizada pela Organização Mundial da Saúde (OMS).
Descrição	O cálculo de IMC que irá haver dentro do Maiself no módulo de hábitos por exemplo seguirá a fórmula da OMS que é IMC = peso/(altura * altura) .
Prioridade	Essencial

Tabela 06 – Tabela do Requisito não Funcional 06.

[RNF 07] - As faixas indicativas de Índice de Massa Corpórea devem também seguir as medidas da OMS.

Identificador	[RNF 07] - As faixas indicativas de Índice de Massa Corpórea devem também seguir as medidas da OMS.
Descrição	Quando houver cálculo de IMC para os usuários, eles serão alocados em faixas que os representarão, de acordo com sua situação corpórea: Baixo peso muito grave = abaixo de 16 kg/m ² . Baixo peso grave = entre 16 e 16,99 kg/m ² . Baixo peso = entre 17 e 18,49 kg/m ² . Peso normal = entre 18,50 e 24,99 kg/m ² . Sobrepeso = entre 25 e 29,99 kg/m ² . Obesidade grau I = entre 30 e 34,99 kg/m ² . Obesidade grau II = entre 35 e 39,99 kg/m ² . Obesidade grau III (obesidade mórbida) = maior que 40 kg/m ² .
Prioridade	Essencial

Tabela 07 – Tabela do Requisito não Funcional 07.

[RNF 08] - O aplicativo deve ser capaz de tratar chamadas simultâneas a rotas diferentes em sua API.

Identificador	[RNF 08] - O aplicativo deve ser capaz de tratar chamadas simultâneas a rotas diferentes em sua API.
Descrição	O aplicativo Maiself deverá se integrar com o back-end em formato API REST e deve garantir que as requisições que serão feitas ao sistema deverão ser executadas simultaneamente com outros usuários que também fizeram requisições.
Prioridade	Essencial

Tabela 08 – Tabela do Requisito não Funcional 08 .

Requisitos de Confiabilidade

- O Requisito de Confiabilidade refere-se a se manter em um nível de desempenho específico quando posto em situações de desempenho específicas, suas subdivisões são: Tolerância a Falhas, Recuperabilidade, Conformidade e outros;
- Requisitos Não-Funcionais Derivados:

[RNF 09] - O aplicativo deve estar disponível 97% do tempo, 24 horas por dia , 7 dias por semana.

Identificador	[RNF 09] - O aplicativo deve estar disponível 97% do tempo referente às 24 horas por dia, 7 dias por semana.
Descrição	O sistema deve ser capaz de ficar no ar em média 97% do tempo de um calendário mensal, exceto para os períodos de manutenção e melhorias, isso são 24 horas por dia, nos dias que estiver no ar e 7 dias por essas referentes semanas, ou seja, o quanto ele “para” de forma programada resulta no percentual de 3%.
Prioridade	Importante

Tabela 09 – Tabela do Requisito não Funcional 09.

[RNF 10] - O aplicativo deve entrar em manutenção nos períodos de noite/madrugada. Precisamente das 00:00 AM às 03:30 AM todo final de mês entre dia 26 e 28.

Identificador	[RNF 10] - O aplicativo deve entrar em manutenção nos períodos de noite/madrugada. Precisamente das 00:00 AM às 03:30 AM.
Descrição	As manutenções do sistema devem ocorrer perto do final de mês com a finalidade de correção de bugs pequenos, caso não tenham sido descoberto bugs críticos. Caso as duas alternativas sejam inexistentes, será usado o tempo para implementação de melhorias no sistema. Esse período é o que estimamos que a minoria dos usuários seriam prejudicados com a parada do sistema.
Prioridade	Essencial

Tabela 10 – Tabela do Requisito não Funcional 10.

Requisitos de Portabilidade

- É a facilidade que um sistema tem de ser transferido para outro ambiente computacional sem sofrer alterações, ou seja, poder rodar em ambientes distintos.
- **Requisitos Não-Funcionais Derivados:**

[RNF 11] - O aplicativo deve funcionar em celulares que possuem o sistema operacional Android OU iOS.

Identificador	[RNF 11] - O aplicativo deve funcionar em celulares que possuem o sistema operacional Android OU iOS.
----------------------	---

Descrição	O aplicativo deve funcionar em celulares que possuem o sistema operacional Android OU iOS, desde que atenda às versões especificadas no [RNF 12].
Prioridade	Importante

Tabela 11 – Tabela do Requisito não Funcional 11.

[RNF 12] - O aplicativo deve funcionar em todos os dispositivos que possuam as versões mais recentes, tanto para o ambiente nativo Android quanto iOS.

Identificador	[RNF 12] - O aplicativo deve funcionar em todos os dispositivos que possuam as versões mais recentes, tanto para o ambiente nativo Android quanto iOS.
Descrição	O aplicativo deve executar em todo e qualquer dispositivo da plataforma Android, que esteja alinhado com as mais recentes atualizações da plataforma e com as mais novas features que foram implementadas também, também deve executar em todo e qualquer dispositivo da plataforma IOS, que esteja alinhado com as mais recentes atualizações e features da plataforma. Para Android, dispositivos acima do android 7.0, já para iOS, apenas modelos com o iOS 9 ou posterior instalado.
Prioridade	Essencial

Tabela 12 – Tabela do Requisito não Funcional 12.

Requisitos de Implementação

- São os requisitos que delimitam o escopo quanto à questão da implementação, seja através da especificação de tecnologias, definição do banco de dados, frameworks, etc.
- **Requisitos Não-Funcionais Derivados:**

[RNF 14] - A documentação da API da Maiself deverá seguir os padrões e ser documentada em Swagger.

Identificador	[RNF 14] - A documentação da Maiself deverá seguir os padrões e ser documentada em Swagger.
Descrição	A API da Maiself após concluída deverá ser documentada na plataforma Swagger, onde será documentado cada endpoint consumível e disponível da API e o corpo da Requisição.

Prioridade	Importante
-------------------	------------

Tabela 14 – Tabela do Requisito não Funcional 14.

[RNF 17] - O aplicativo deverá ser implementado sobre ambiente React Native em conjunto com a ferramenta Expo, apoiada sobre a linguagem/superset Javascript o Typescript.

Identificador	[RNF 17] - O aplicativo deverá ser implementado sobre ambiente React Native em conjunto com a ferramenta Expo, apoiada sobre a linguagem/superset Javascript o Typescript.
Descrição	O aplicativo deve ser implementado sobre o framework React Native, com o apoio da ferramenta Expo, utilizando o superset de Javascript o Typescript como linguagem/ferramenta base para a implementação.
Prioridade	Importante

Tabela 17 – Tabela do Requisito não Funcional 17.

Requisitos de Padrões

- Esses requisitos determinam quais padrões serão utilizados no desenvolvimento do sistema e na definição do projeto. Então, referem-se às formas utilizadas de acordo com a metodologia adotada, a forma de programação, etc.
- Requisito Não-Funcional.

[RNF 20] - O aplicativo deverá focar sua arquitetura na componentização dos elementos da interface.

Identificador	[RNF 20] - O aplicativo deverá focar sua arquitetura na componentização dos elementos da interface.
Descrição	A estruturação do aplicativo, seguirá um padrão que foca na reutilização de código para cenários semelhantes por meio da componentização da interface, assim gerando componentes que podem ser replicados em mais de um local da interface sem que haja repetição de código.
Prioridade	Essencial

Tabela 20 – Tabela do Requisito não Funcional 20.

Requisitos de Interoperabilidade

- Esse requisito é referente a necessidade da aplicação se conectar com outra aplicação, aqui deve ser especificado necessidades do sistema de implementação de webservices ou outros tipos de interfaces com sistemas externos.
- Requisitos Não-Funcionais.

[RNF 21] - O aplicativo deve ser modelado para consumir outras APIs caso necessário.

Identificador	[RNF 21] - O aplicativo deve ser modelado para consumir outras APIs caso necessário.
Descrição	O aplicativo que fará a comunicação com o back-end em formato API deve ser construído de forma que possa facilitar a integração e a realização de requisições com outras APIs não se prendendo apenas a nossa, um exemplo definido será a do uso da API do Google, para serviços de localização.
Prioridade	Essencial

Tabela 21 – Tabela do Requisito não Funcional 21.

[RNF 22] - O aplicativo reutilizará a mesma API desenvolvida para o sistema Web.

Identificador	[RNF 22] - O aplicativo reutilizará a mesma API desenvolvida para o sistema Web.
Descrição	O aplicativo reutilizará a mesma API desenvolvida para o sistema Web, utilizando um conceito bastante aplicado no mercado atualmente.
Prioridade	Essencial

Tabela 22 – Tabela do Requisito não Funcional 22.

[RNF 23] - A Maiself terá um único aplicativo que irá rodar na plataforma tanto Android quanto IOS.

Identificador	[RNF 23] - A Maiself terá um único aplicativo que executa tanto em plataforma Android quanto IOS.
Descrição	A Maiself terá um único aplicativo programado e implementado, que o mesmo irá executar sobre as principais plataformas móveis ativas hoje no mercado, sendo as mesmas Android e IOS, isso será possível graças ao expo, que gera um bundle “universal”.
Prioridade	Essencial

Tabela 23 – Tabela do Requisito não Funcional 23.

Requisitos de Ética

- Adicionamos esses requisitos extras com o intuito somente de documentar decisões que estão relacionadas com a visão e com os aspectos burocráticos do sistema.
- Requisitos Não-Funcionais Derivados:

[RNF 24] - A Maiself não divulgará nem negociará quaisquer dados ou estatísticas relacionadas ao usuário.

Identificador	[RNF 24] - A Maiself não divulgará nem negociará quaisquer dados ou estatísticas relacionadas ao usuário.
Descrição	Visando a integridade ética e a segurança dos nossos usuários é essencial que não iremos divulgar/vender, nenhum dado do usuário ou relacionado a ele.
Prioridade	Essencial

Tabela 24 – Tabela do Requisito não Funcional 24.

6 Prototipação

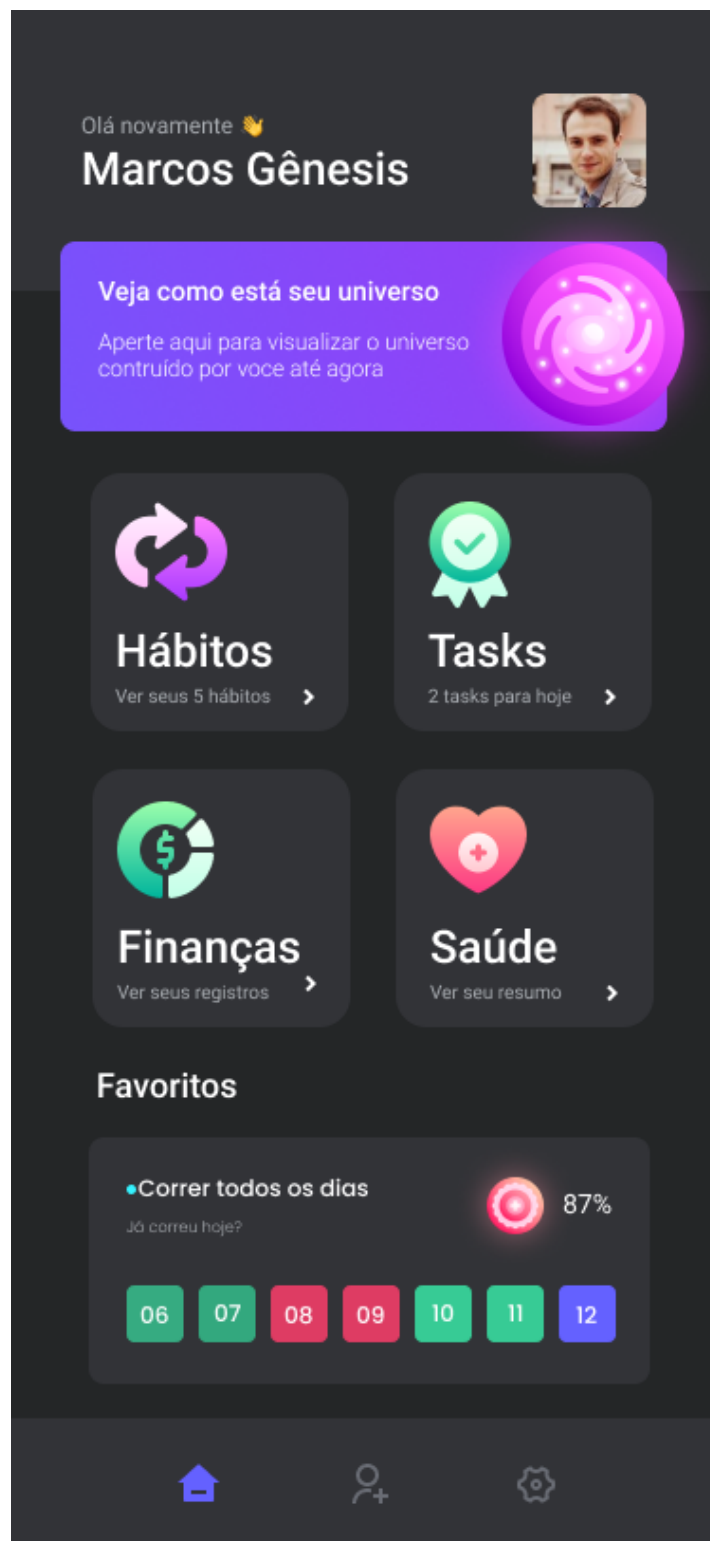


Figura 5 – Protótipo da tela home.



Figura 6 – Protótipo da tela de detalhamento do hábito.



Figura 7 – Protótipo da tela de listagem dos hábitos.



Figura 8 – Protótipo da tela de detalhamento completo do hábito.

6.1 Repositórios de Código e Tracker de Tasks

6.1.1 Back-End

<https://github.com/maiselfy/backend>

6.1.2 Mobile

<https://github.com/maiselfy/appmaiself>

6.1.3 Tracker de Tasks

<https://app.clickup.com/3115497/v/l/t/3115497>

6.1.4 Link para documentação do projeto

<https://github.com/italolima04/maiself-documentacao-pi-III>

7. Descrição de frameworks e linguagens utilizadas

Nesta seção será descrito os frameworks e linguagens de programação utilizados no nosso projeto tanto no front-end, que nesse disciplina de Projeto Integrado é a parte visual do aplicativo mobile que recebe os dados do usuário como entrada, manipula e mostra os dados do back-end que esse por sua vez é responsável por definir, modelar e armazenar os dados que vem do front-end.

- Linguagens/Frameworks Front-end
 - **React-Native:** React-Native é um framework javascript utilizado no desenvolvimento de aplicações mobile tanto para Android quanto IOS. Uma das vantagens de desenvolver em React-Native é que não é preciso desenvolver dois projetos diferentes para cada sistema operacional, mas é possível ter estilizações específicas para cada sistema operacional dependendo do projeto. Também pode utilizar o superset TypeScript que serve para tipar as propriedades, variáveis, para que seja mais descritivo o nosso código na passagem de valores que é o nosso caso.
 - **Expo:** O Expo é uma ferramenta que auxilia no desenvolvimento mobile com React-Native sem precisar instalar dependências nativas dos sistemas operacionais Android e IOS como SDK do Android ou XCode do Mac para

começar a desenvolver, isso por que o Expo(projetos criados com Expo) já possui um app que contém todo código nativo necessário pelo React-Native, então a única alteração vai ser no código javascript. Podemos criar um projeto que serve como template para termos como base o nosso desenvolvimento apenas com um comando em vez de desenvolver do zero.

- **Linguagens/Frameworks Back-end:**
 - **NodeJs:** Ele pode ser definido como um ambiente de execução de javascript server-side (runtime environment) e não uma linguagem de programação, onde uma base de código específica pode ser executada em tempo real. No nosso caso o runtime environment utilizado é o Node onde não precisamos necessariamente do navegador para executar o código javascript.
 - **NestJs:** É um framework javascript que por padrão utiliza o superset de tipagem de dados TypeScript e estrutura de testes Javascript Jest. Também é um framework orientado a desenvolvimento modular, ou seja, desenvolvemos baseado em módulos e não em entidades, mas o que seriam módulos? Seriam partes que poderiam ser isoladas/excluídas de outros módulos e estes ainda conseguiriam existir com suas regras de negócio. No caso da nossa aplicação podemos citar exemplos de módulos como o Usuário, Financeiro, Hábitos.

8. Banco de Dados

Toda aplicação tem a necessidade de armazenar dados, sejam estes dos mais diversos tipos, independente do âmbito em que ela está inserida, na nossa plataforma, existem dados de usuário, dados relacionados aos hábitos desses usuários, dados produzidos pelo relacionamento entre diferentes usuários, dados sobre as finanças do mesmo, etc... Para persistir esses dados, na era caracterizada pela imersão de dados, devemos tomar decisões baseado nos critérios que consideramos prioritários, e isso pode significar a co-utilização de diferentes tipos de bancos de dados ou até mesmo o uso de um banco de cache (redis), em caso de evolução. No entanto, essa opção fica para análise em momento futuro, na atual situação, o banco utilizado por nós é o PostgreSQL, devido a sua grande visibilidade no mercado, considerando um momento de alta da tecnologia, além disso há uma maior facilidade para lidar com os relacionamentos entre o módulo e outras entidades (no nosso cenário), visto que estes podem acontecer de maneira implícita ou explícita. Assim, como supracitado, focando na aplicação para o momento, aderimos ao modelo relacional em sua totalidade desenvolvida, onde os dados são modelados através de tabelas, via controle de versão do banco de dados, através de migrations. Isso nos permite proporcionar uma maior consistência e segurança para os mesmos.

Quanto a escolha do tipo do banco (relacional X não relacional), nos baseamos no fato de que nossas entidades estão “ligadas” através de relacionamentos e optamos pela opção SQL. Ademais, devido à circunstância de possuímos entidades bem definidas e distintas, é preferível realizar esse armazenamento em tabelas, onde cada tabela guarda as informações referentes a uma entidade, o que não acontece no NOSQL.

8.1 Outras opções

Um questionamento a ser feito em toda tomada de decisão é sobre as outras opções disponíveis para a escolha, assim, além de analisar os pontos fortes e fracos da opção principal, é aconselhável que seja realizada uma comparação entre tal e suas concorrentes. Falando em tecnologia, isso não é diferente, pelo contrário, devido a multiplicidade de opções, tecnologias, frameworks, o âmbito acaba se tornando um universo de opções-escolhas. Para banco de dados, comparamos o PostgreSQL (nossa escolha) com o MySQL, uma outra forte opção no mercado. Abaixo serão realçados alguns fatores que foram determinantes para a decisão.

PostgreSQL X MySQL

Velocidade X Robustez

O MySQL é o mais utilizado no desenvolvimento de aplicações onde a velocidade é importante (porém isto tem mudado com versões mais recentes do PostgreSQL), enquanto que o PostgreSQL se destaca por ser mais robusto e possuir muito mais recursos que o MySQL.

Custo

Ambos SGBDS são gratuitos.

Atualizações

Nas últimas versões do MySQL, os desenvolvedores acrescentaram diversos recursos que já existiam no PostgreSQL como transações (commit e rollback), triggers, stored procedures, views, lock line e constraints.

Armazenamento

O PostgreSQL suporta quantidade ilimitada de linhas, capacidade de armazenamento de bancos e tabelas de até 16 TB.

Comparativo Geral

As principais vantagens do PostgreSQL são: maior capacidade de armazenamento, maior robustez, segurança e escalabilidade, presença das funções mais modernas, que garantem o versionamento de mudanças (e a possibilidade de voltar atrás em determinados cenários), o PostgreSQL permite uma maior quantidade de sub-consultas, ou queries agrupadas, além de possuir um maior número de tipos de dados, garantindo o escopo de aplicações dos mais variados tipos, desde contextos genéricos, até à nichos mais específicos.

8.2 Diagrama visual

