



Universidade Estadual de Maringá

Departamento de Informática

Disciplina: Construção de Software

Professor: Guilherme Corredato Guerino

Software para Aplicação de Uso em Unidades em Academias

Alunos:

Pedro Henrique de Melo Costa
Renan Augusto Leonel
Henrique Ribeiro Favaro
Danillo Dias Nascimento

RA:

112653
115138
115408
090561

Maringá
2022

Sumário

Backlog do Produto	3
2. Arquitetura Projetada para o Software	4
Arquitetura Cadastrar Perfil	4
Arquitetura Gerenciar Treinos	5
Arquitetura Gerenciar Saúde	5
Arquitetura Gerenciar Metas	6
Arquitetura Gerenciar IMC	7
Diagrama de Visão Global	7
2.1 Diagramas de Caso de Uso	8
Diagrama de Casos de Uso do Cadastro	8
Diagrama de Casos de Uso das Movimentações	8
Diagrama de Casos de Uso dos Relatórios e Consultas	9
2.2 Diagramas de Atividades	9
Diagrama de Atividade - H01	9
Diagrama de Atividade - H02	10
Diagrama de Atividade - H03	10
Diagrama de Atividade - H04	10
Diagrama de Atividade - H05	11
Diagrama de Atividade - H06	11
Diagrama de Atividade - H07	12
3. Backlog da Sprint	12
Planejamento 1 Sprint	13
Planejamento 2 Sprint	14
Planejamento 3 Sprint	15
Resultado Das Sprints	16
Sprint 1	16
Sprint 2	17
Sprint 3	18
Arquitetura	19
Linguagem	19
Frameworks	19
Banco de dados:	20
APIs	20
7. Referências	22

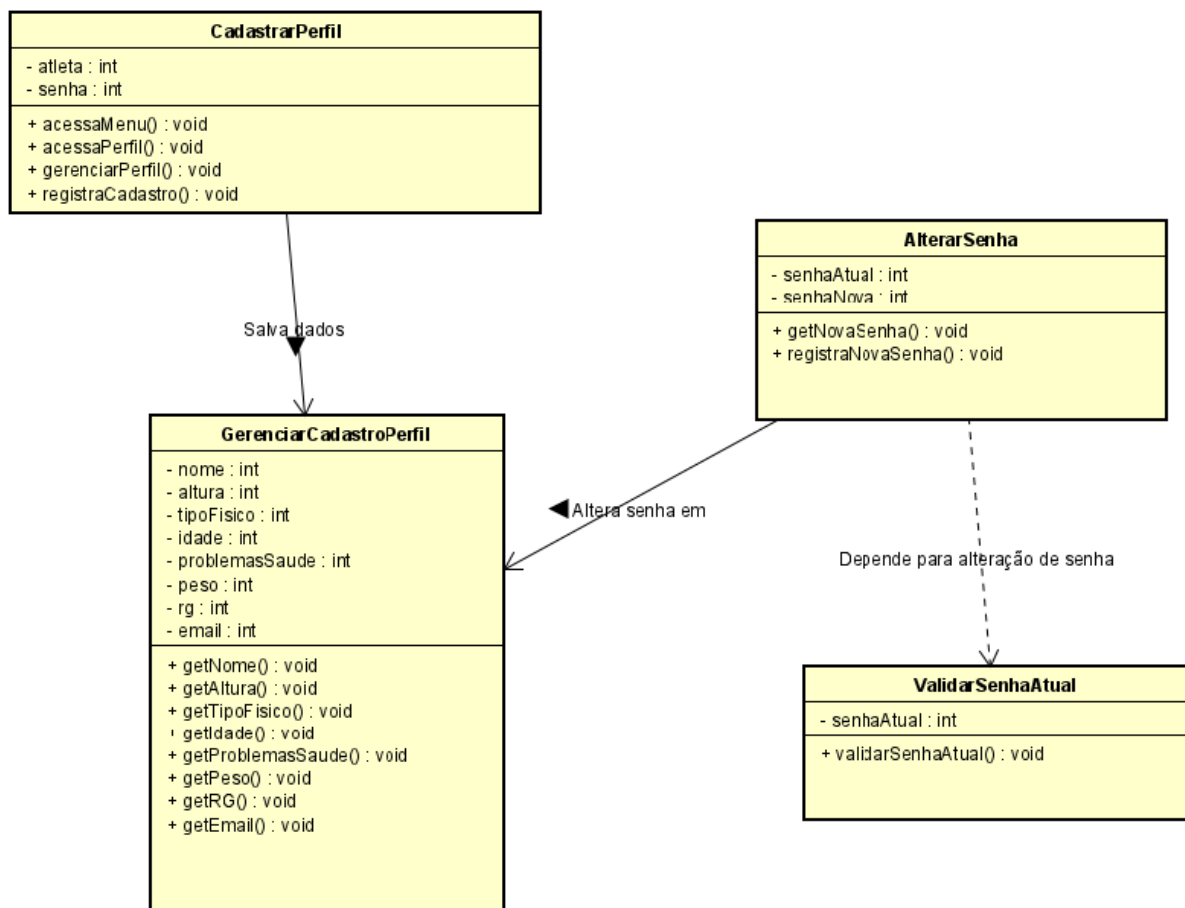
1. Backlog do Produto

Prioridade	História	Código
1	O usuário deve fazer um cadastro de seus dados pessoais (Nome completo, idade, tipo físico (ectomorfo, mesomorfo, endomorfo), doenças crônicas, peso e altura	H01
-	O usuário pode também adicionar uma foto de perfil, ou permanecer com a default do sistema.	Apêndice H01
2	O software deve disponibilizar a criação de perfis personalizados para cada aluno com seus respectivos objetivos de treinos (perder peso, ganhar peso, saúde/doença - exemplo: doença cardiovascular, definição muscular, etc).	H02
3	O software deve acompanhar em tempo real os exercícios concluídos e a serem concluídos, para isso, cada exercício contará com uma caixa para marcação pelo usuário assim que o exercício for finalizado.	H03
4	O software deve disponibilizar uma planilha integrada com alimentos selecionados para a dieta do usuário.	H04
5	O software deve permitir que o monitor poste vídeos para cada exercício, demonstrando a maneira correta de executá-los para que não ocorram lesões por mau uso dos equipamentos ou exercícios.	H05
6	O software deve oferecer uma calculadora para o cálculo de IMC e sua condição física (Magreza Grave, Saudável, Sobrepeso, entre outros).	H06
7	O software deve trazer diversos artigos científicos que mostrem a relação dos exercícios físicos com nosso bem estar físico e emocional.	H07
-	Os artigos serão categorizados por tema/objetivo, como por exemplo: emagrecimento, hipertrofia, benefícios para idosos, etc.	Apêndice H07
-	Nesta aba, também ficará registrado os artigos que o usuário já leu, de forma que o título fique com uma cor de destaque (greyed out).	Apêndice H07

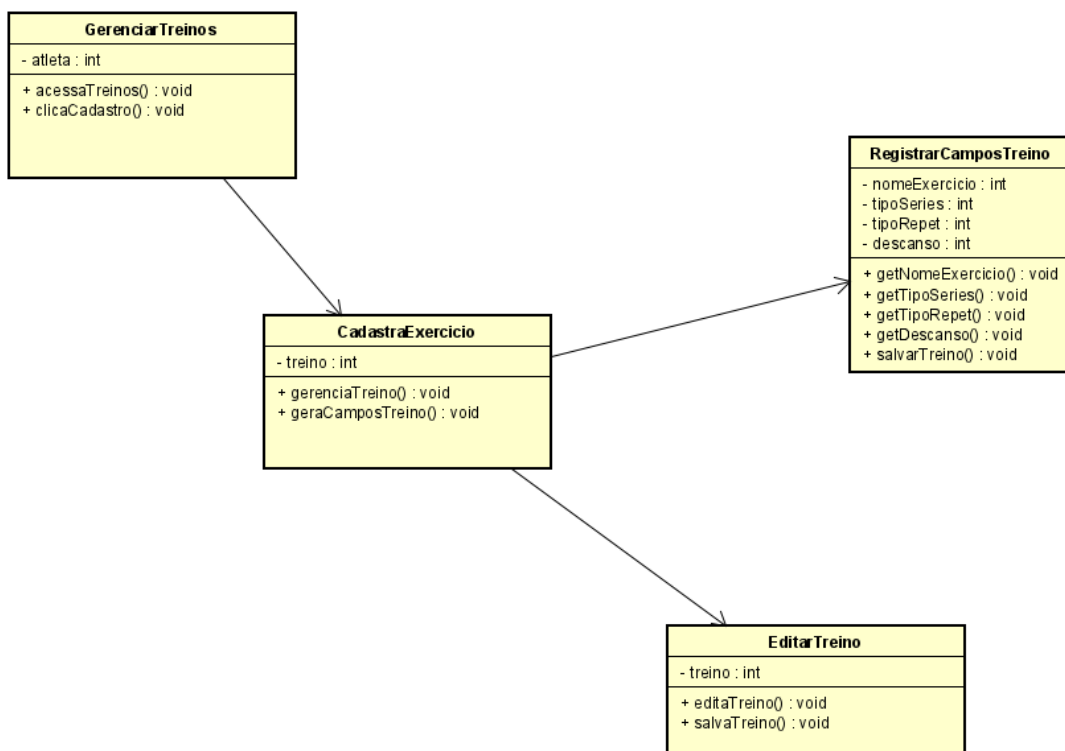
2. Arquitetura Projetada para o Software

A arquitetura escolhida para o projeto foi a arquitetura MVC (Model - View - Control), por ser comumente utilizada no desenvolvimento de aplicativos mobile. Essa estrutura ajuda o aplicativo a se concentrar em funções e tarefas separadas simultaneamente com a ajuda dessas camadas isoladas. Dessa forma, podemos refatorar o código com maior facilidade, além de adicionar novos recursos, simplificando o processo de manutenção. Além disso, a arquitetura MVC é amplamente utilizada nos sistemas iOS, o que nos incentiva a aderir o uso dessa arquitetura no software.

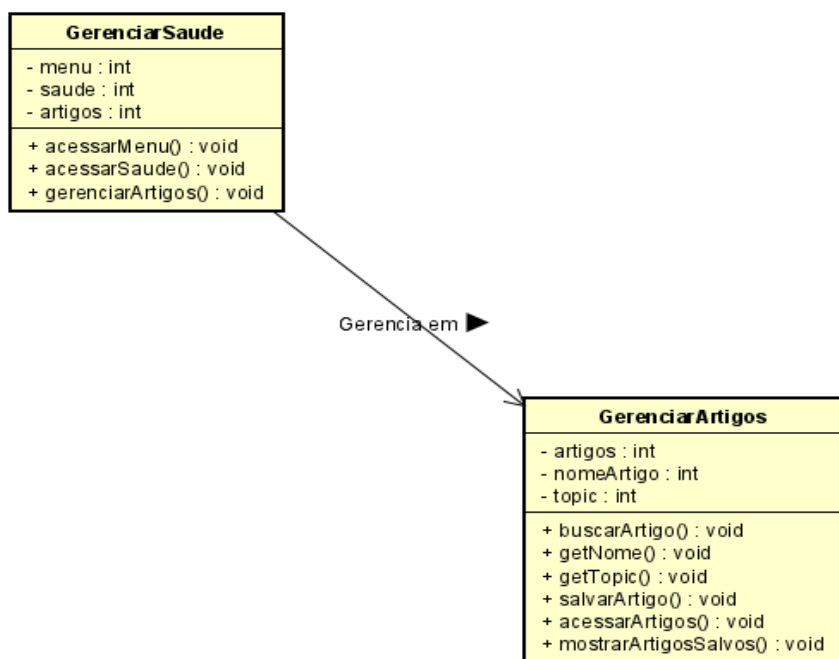
Arquitetura Cadastrar Perfil

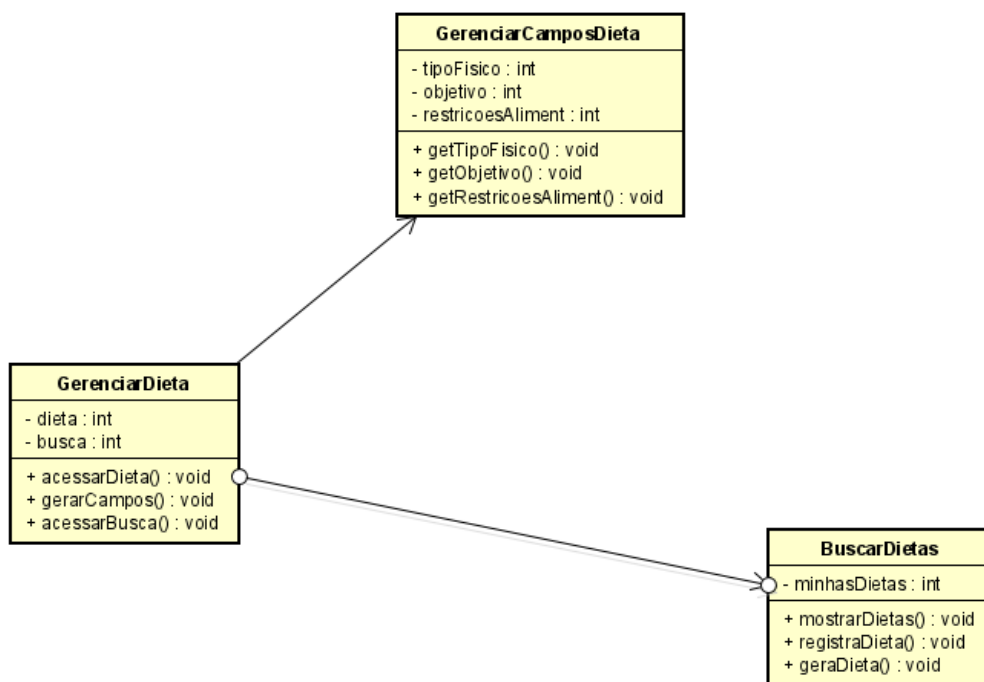


Arquitetura Gerenciar Treinos

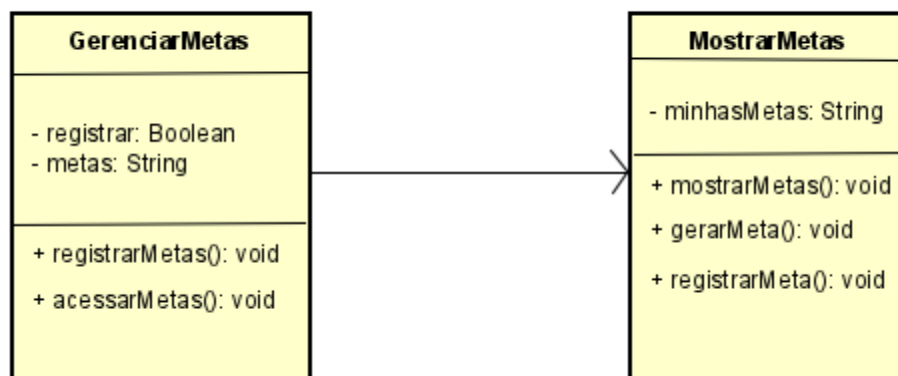


Arquitetura Gerenciar Saúde





Arquitetura Gerenciar Metas



Arquitetura Gerenciar IMC

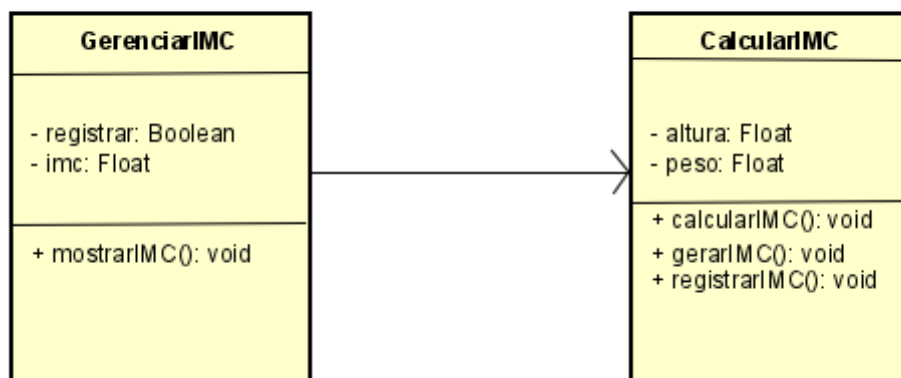
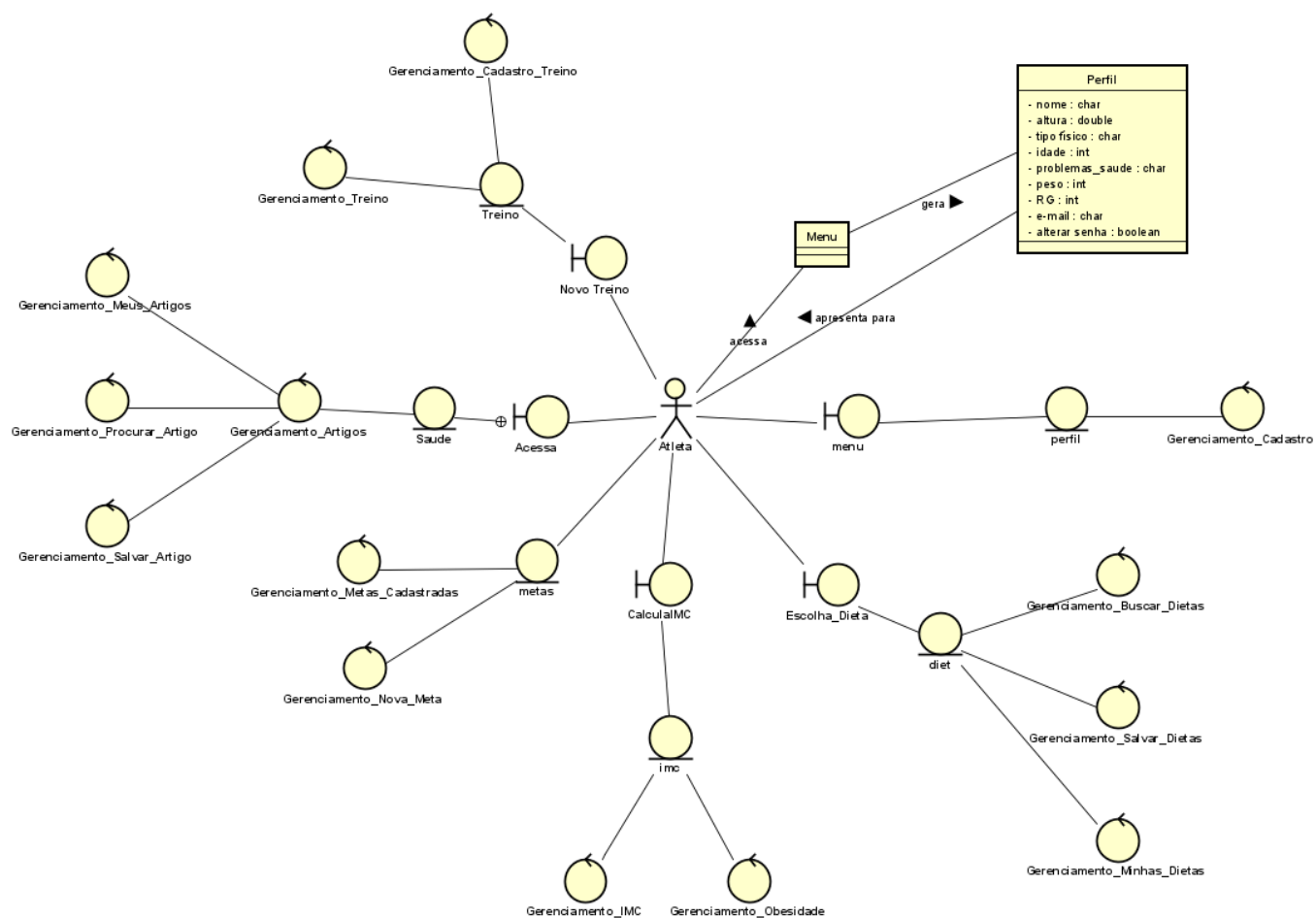


Diagrama de Visão Global



2.1 Diagramas de Caso de Uso

Diagrama de Casos de Uso do Cadastro

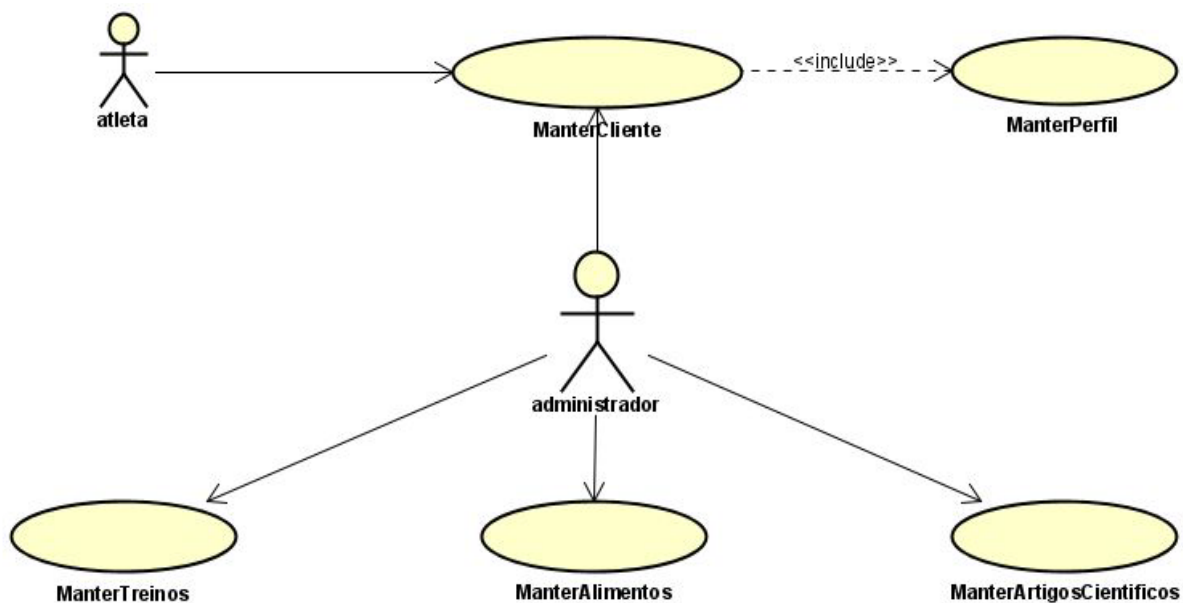


Diagrama de Casos de Uso das Movimentações

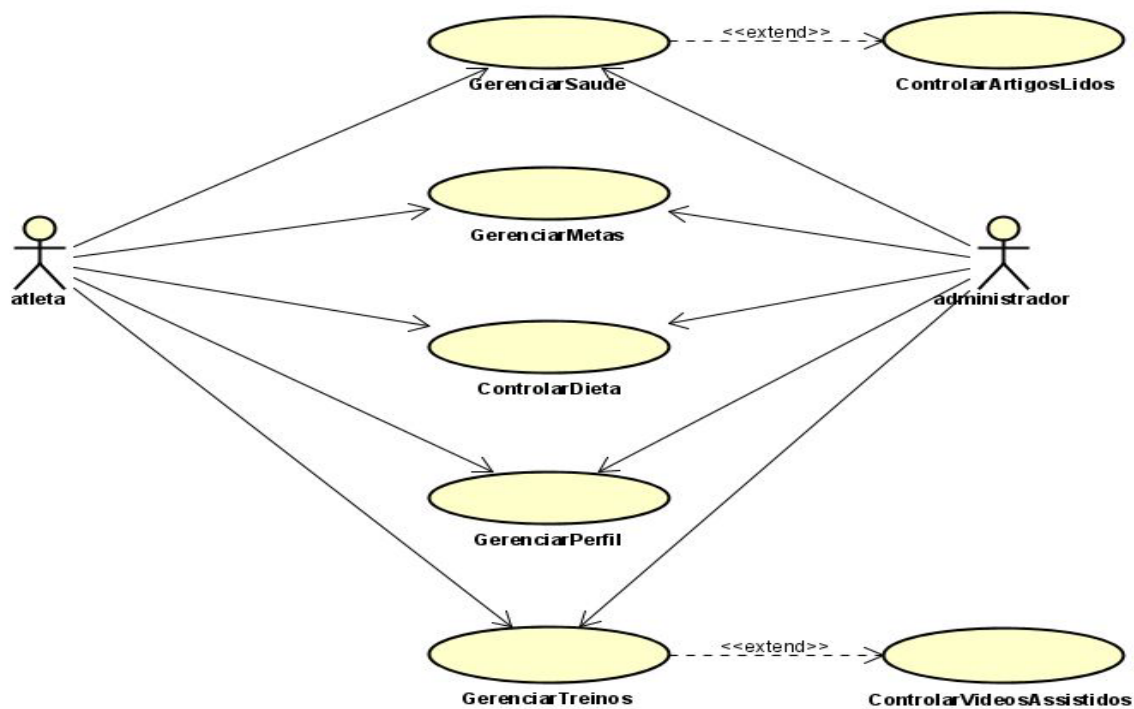
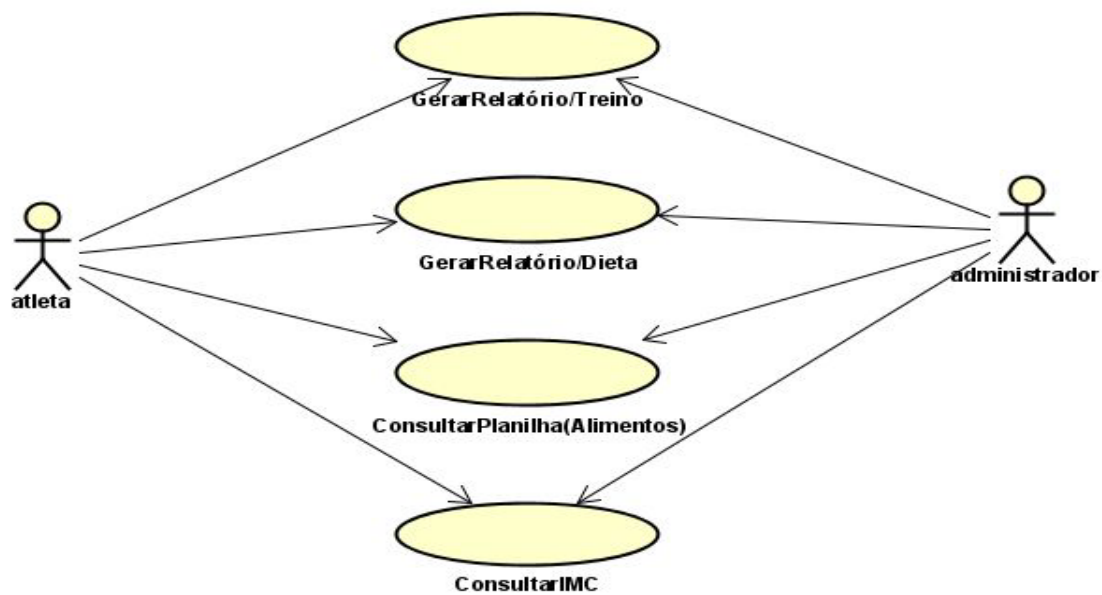


Diagrama de Casos de Uso dos Relatórios e Consultas



2.2 Diagramas de Atividades

Diagrama de Atividade - H01

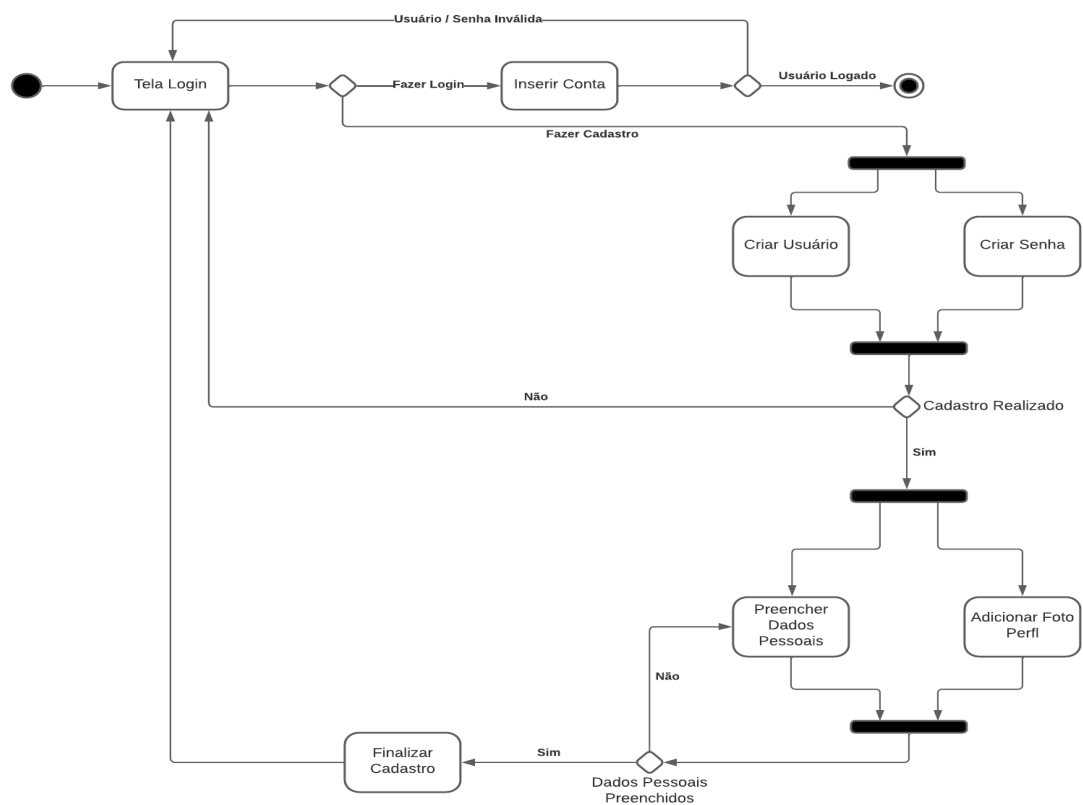


Diagrama de Atividade - H02

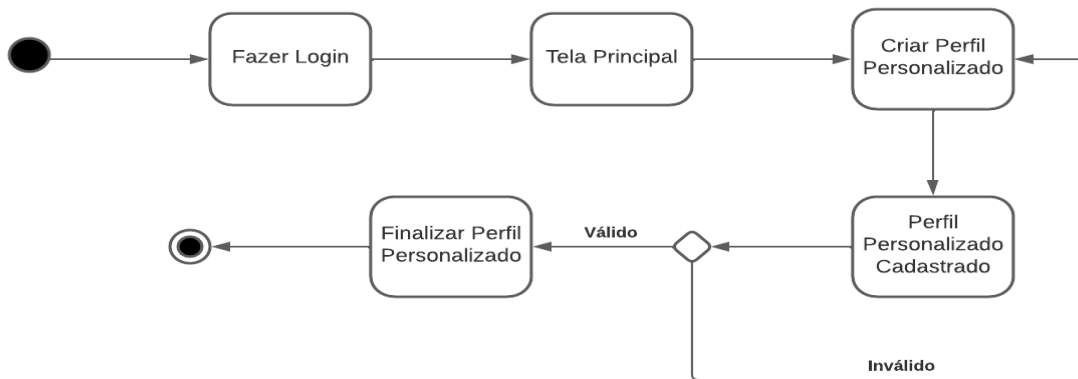


Diagrama de Atividade - H03

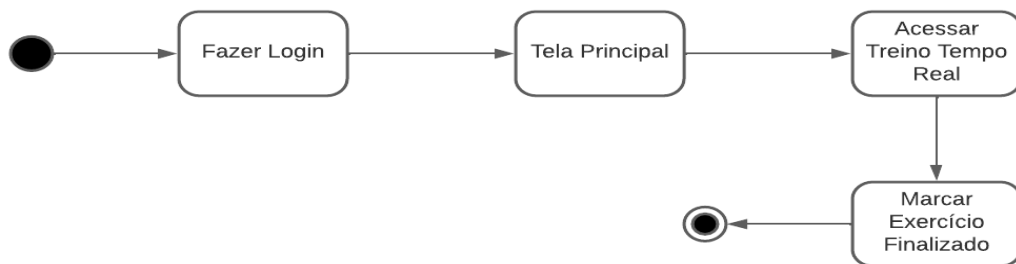


Diagrama de Atividade - H04

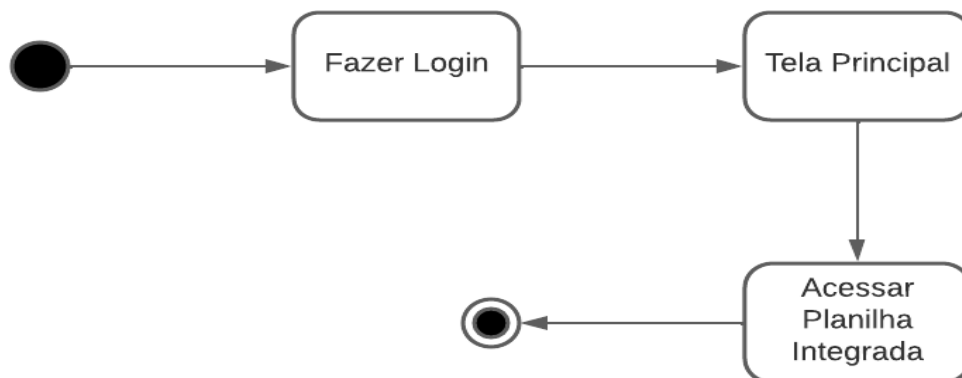


Diagrama de Atividade - H05

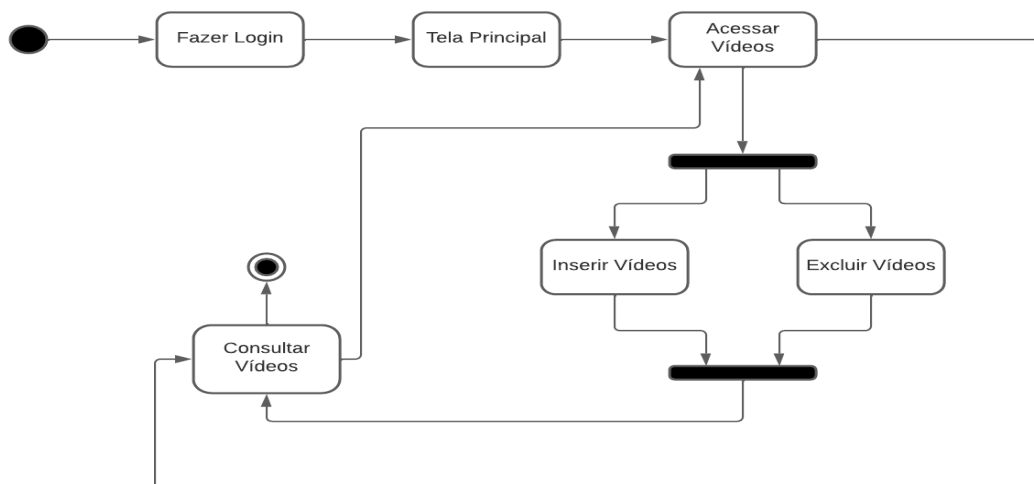


Diagrama de Atividade - H06

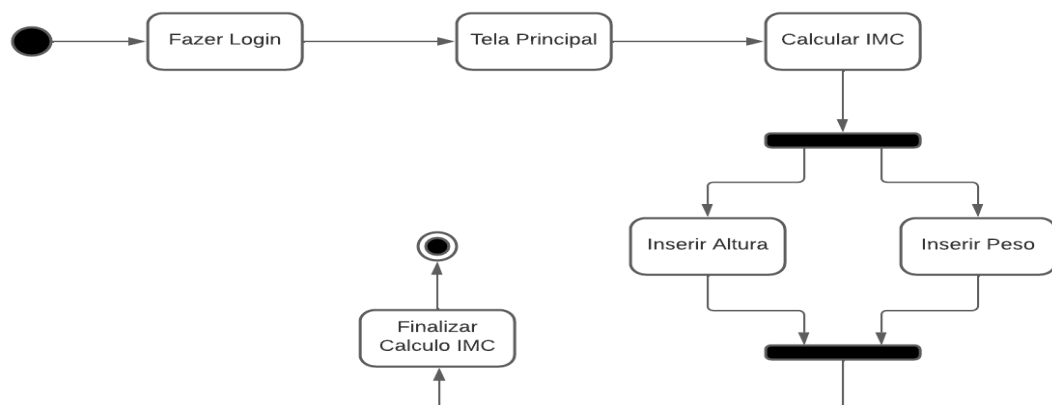
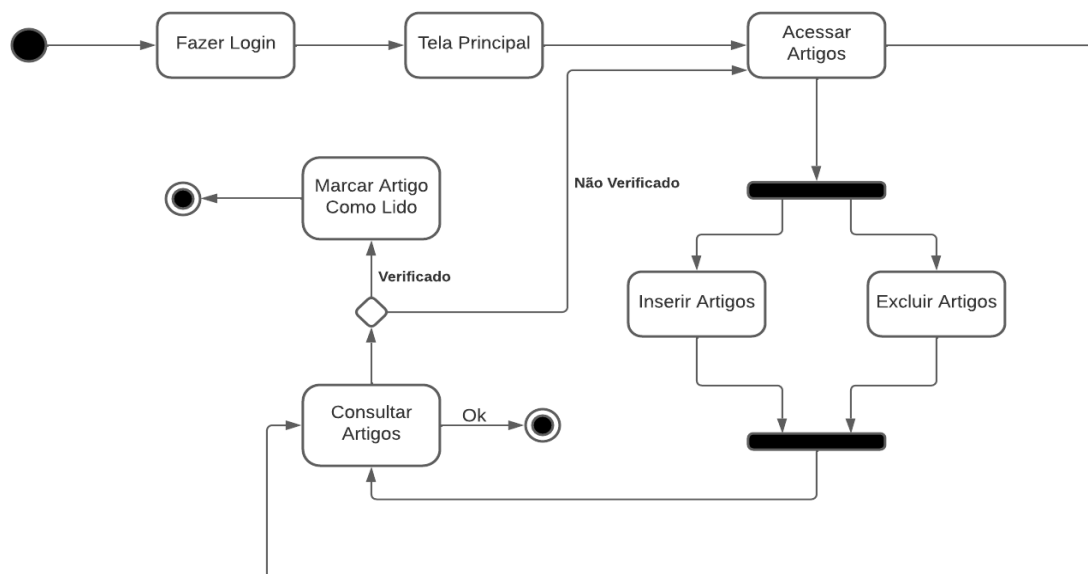


Diagrama de Atividade - H07



3. Backlog da Sprint

Para o planejamento de cada sprint foi criada uma tabela onde contém as histórias que serão desenvolvidas com time-boxed de 2 semanas cada. As tarefas estão organizadas com indicadores únicos, e indicando seu responsável pela implementação. Depois foi realizada outras tabelas, onde contém o resultado das sprints, no qual foram realizadas modificações em determinadas histórias e tarefas. Abaixo encontra-se as tabelas:

Planejamento 1 Sprint

ID	Histórias	Tarefas ID	Tarefas	Responsável	Status
H01	O usuário deve fazer um cadastro de seus dados pessoais (Nome completo, idade, tipo físico (ectomorfo, mesomorfo, endomorfo), doenças crônicas, peso e altura	T01.1	Formular o template inicial do app, como cores principais de background, fontes, ideia de layout, navegação e estilização.	Henrique/Pedro	To do
		T01.2	Criação de tela inicial, apresentando uma tela de bem-vindo e botão para se cadastrar	Pedro/Henrique	To do
		T01.3	Na tela de cadastro, apresentar os campos de nome, idade, tipo físico, doenças crônicas, peso e altura, para que o usuário possa inserir suas informações	Renan/Danillo	To do
		T01.4	Após o cadastro, apresentar uma tela de resposta mostrando que o cadastro foi efetuado com sucesso, e redirecionar para a criação de perfil	Danillo/Pedro	To do
H02	O software deve disponibilizar a criação de perfis personalizados para cada aluno com seus respectivos objetivos de treinos (perder peso, ganhar peso, saúde/doença - exemplo: doença cardiovascular, definição muscular, etc).	T02.1	Criar tela semelhante à de cadastro, oferecendo as opções de objetivos de treino	Renan	To do
		T02.2	Após a seleção de objetivo, perguntar em outra tela se o usuário possui alguma doença/problema de saúde	Pedro	To do

Planejamento 2 Sprint

ID	Histórias	Tarefas ID	Tarefas	Responsável	Status
H03	O software deve acompanhar em tempo real os exercícios concluídos e a serem concluídos, para isso, cada exercício contará com uma caixa para marcação pelo usuário assim que o exercício for finalizado.	T03.1	Criação da tela de treinos, onde o usuário terá acesso aos seus treinos.	Renan	To do
		T03.2	Ao acessar a um treino específico, cada exercício contido no treino, contará com uma caixa de marcação, podendo esta ser marcada após a execução do exercício.	-	To do
		T03.3	O usuário poderá criar um novo treino específico que desejar.	Pedro/Henrique/ Renan/Danillo	To do
H04	O software deve disponibilizar uma planilha integrada com alimentos selecionados para a dieta do usuário.	T04.1	Criar tela que permitirá ao usuário inserir seu tipo físico, objetivo de dieta, e suas restrições alimentares.	Pedro/Renan	To do
		T04.2	Criação de tela no qual o usuário poderá visualizar e acessar suas dietas.	Danillo/Henrique/ Pedro	To do
H05	O software deve permitir que o monitor poste vídeos para cada exercício, demonstrando a maneira correta de executá-los para que não ocorram lesões por mau uso dos equipamentos ou exercícios.	T05.1	Criar seção para upload de vídeos em cada exercício cadastrado no treino do atleta	-	To do
		T05.2	Ao clicar para postar vídeo, pedir autenticação de admin (login e senha), para reproduzir manter sem autenticação.	-	To do

Planejamento 3 Sprint

ID	Histórias	Tarefas ID	Tarefas	Responsável	Status
H06	O software deve oferecer uma calculadora para o cálculo de IMC, onde apontará sua condição física (Magreza Grave, Saudável, Sobrepeso, entre outros).	T06.1	Criação da tela de IMC, onde o usuário colocará sua altura e peso, para obter sua condição física.	Danillo/Henrique/Renan	To do
H07	O software deve trazer diversos artigos científicos que mostrem a relação dos exercícios físicos com o estar físico e emocional.	T07.1	Criar tela no qual terá diversos artigos científicos, no qual serão categorizados por tema/objetivo, como por exemplo: emagrecimento, hipertrofia, entre outros.	Renan	To do
		T07.2	Nesta tela ficará registrado os artigos que o usuário já leu, de forma que o título fique com uma cor de destaque (greyed out).	-	To do

Resultado Das Sprints

Sprint 1

ID	Histórias	Tarefas ID	Tarefas	Responsável	Status
H01	O usuário deve fazer um cadastro de seus dados pessoais (Nome completo, idade, tipo físico (ectomorfo, mesomorfo, endomorfo), doenças crônicas, peso e altura	T01.1	Formular o template inicial do app, como cores principais de background, fontes, ideia de layout, navegação e estilização.	Henrique/Pedro	Done
		T01.2	Criação de tela inicial, apresentando uma tela de bem-vindo e botão para se cadastrar	Pedro/Henrique	Done
		T01.3	Na tela de cadastro, apresentar os campos de nome, idade, tipo físico, doenças crônicas, peso e altura, para que o usuário possa inserir suas informações	Renan/Danillo	Done
T1	TAREFA ADICIONADA	TA1	Criar página para completar o cadastro com informações pessoais	Henrique/Pedro	Done
T2	TAREFA ADICIONADA	TA2	Integrar campos de Login e Cadastro com uma verificação BackEnd para implementação do banco de dados	Henrique	Done
H03	TAREFA PUXADA DA SPRINT 2	T03.1	Criação da tela de treinos, onde o usuário terá acesso aos seus treinos.	Renan	Done

T01.4 foi removida do planejamento inicial.

T02.1 foi alterada para sprint 2.

T02.2 foi removida

Sprint 2

ID	Histórias	Tarefas ID	Tarefas	Responsável	Status
H04	O software deve disponibilizar uma planilha integrada com alimentos selecionados para a dieta do usuário.	T04.1	Criar tela que permitirá ao usuário inserir seu tipo físico, objetivo de dieta, e suas restrições alimentares.	Pedro/Renan	Done
T3	TAREFA ADICIONADA	TA3	Criar tela para recuperação de senha, onde o usuário recebe um código de validação via e-mail para redefinir sua senha	Henrique	Done
H02	TAREFA PUXADA DA SPRINT 1	T02.1	Criar tela semelhante à de cadastro, oferecendo as opções de objetivos de treino	Renan	Done
H06	TAREFA PUXADA DA SPRINT 3	T06.1	Criação da tela de IMC, onde o usuário colocará sua altura e peso, para obter sua condição física.	Danillo/Henrique /Renan	Done

T03.1 foi puxada pela sprint 1.
 T03.2 foi removida.
 T03.3 foi passada para sprint 3.
 T04.2 foi passada para sprint 3.
 H05 foi removida.

Sprint 3

ID	Histórias	Tarefas ID	Tarefas	Responsável	Status
H07	O software deve trazer diversos artigos científicos que mostrem a relação dos exercícios físicos com o estar físico e emocional.	T07.1	Criar tela no qual terá diversos artigos científicos, no qual serão categorizados por tema/objetivo, como por exemplo: emagrecimento, hipertrofia, entre outros.	Renan	Done
		T07.2*	Nesta tela ficará registrado os artigos que o usuário já leu, de forma que o título fique com uma cor de destaque (greyed out).	Renan	Done
H03	TAREFA PUXADA DA SPRINT 2	T03.3	O usuário poderá criar um novo treino específico que desejar.	Danillo/Henrique/ Pedro/Renan	Done
H04	TAREFA PUXADA DA SPRINT 2	T04.2	Criação de tela no qual o usuário poderá visualizar e acessar suas dietas.	Danillo	Done
T4	TAREFA ADICIONADA	TA4	Tela de Perfil com todas as informações pessoais, foto de perfil e dados da conta	Henrique/Pedro/ Renan	Done
T5	TAREFA ADICIONADA	TA5	O usuário deve fazer um cadastro de seus dados pessoais (Nome completo, idade, tipo físico (ectomorfo, mesomorfo, endomorfo), doenças crônicas, peso e altura	Henrique	Done
T6	TAREFA ADICIONADA	TA6	Tela de Metas com um campo a ser adicionado, pelo usuário, com texto.	Henrique/Pedro/ Renan	Done

T06.1 foi puxada pela sprint 2.

* alteração na lógica do funcionamento da tarefa

5. Descrição do Software

Arquitetura

- **MVC:** O MVC é uma sigla do termo em inglês Model (modelo) View (visão) e Controller (Controle) que facilita a troca de informações entre a interface do usuário aos dados no banco, fazendo com que as respostas sejam mais rápidas e dinâmicas.
- **Model:** Essa classe também é conhecida como Business Object Model (objeto modelo de negócio). Sua responsabilidade é gerenciar e controlar a forma como os dados se comportam por meio das funções, lógica e regras de negócios estabelecidas.
- **Controller:** Responsável por intermediar as requisições enviadas pelo View com as respostas fornecidas pelo Model, processando os dados que o usuário informou e repassando para outras camadas.
- **View:** É a camada responsável por apresentar as informações de forma visual ao usuário. Em seu desenvolvimento devem ser aplicados apenas recursos ligados a aparência como mensagens, botões ou telas.

Linguagem

- **Javascript:** JavaScript é uma linguagem leve, interpretada e baseada em objetos com funções de primeira classe, mais conhecida como a linguagem de script para páginas Web, mas usada também em vários outros ambientes sem browser, tais como node.js, Apache CouchDB e Adobe Acrobat. O JavaScript é uma linguagem baseada em protótipos, multi-paradigma e dinâmica, suportando estilos de orientação a objetos, imperativos e declarativos (como por exemplo a programação funcional).

Frameworks

- **React Native:** O React Native é um Framework para desenvolvimento de aplicativos móveis multiplataforma. Baseado no React, framework JS para desenvolvimento web, o React Native possibilita a criação de aplicações móvel multiplataforma (Android e iOS) utilizando apenas Javascript. Porém, diferente de outros frameworks com esta mesma finalidade (Cordova, por exemplo), todo o código desenvolvido com o React Native é convertido para linguagem nativa do sistema operacional, o que torna o app muito mais fluido.
- **Expo:** O Expo é uma ferramenta utilizada no desenvolvimento mobile com React Native que permite o fácil acesso às API's nativas do dispositivo sem precisar instalar qualquer dependência ou alterar código nativo. Quando iniciamos no desenvolvimento mobile percebemos que o número de API's e recursos nativos que podemos controlar através da nossa aplicação é gigante, e muitas vezes não nos recordamos de todas opções que temos disponíveis. O Expo, por sua vez, oferece grande parte desses recursos de forma nativa e integrada e, por exemplo, você tem

acesso à recursos como câmera, microfone, player de música, entre outros, de forma muito simples utilizando essa ferramenta.

Banco de dados:

- **MongoDB:** O MongoDB é um banco de dados orientado a documentos que possui código aberto (open source) e foi projetado para armazenar uma grande escala de dados, além de permitir que se trabalhe de forma eficiente com grandes volumes. Ele é categorizado no banco de dados NoSQL (not only SQL) pois o armazenamento e a recuperação de dados no MongoDB não são feitas no formato de tabelas. O banco de dados também fornece suporte oficial de driver para todas as linguagens populares como C, C ++, C # e .Net, Go, Java, Node.js, Perl, PHP, Python, Motor, Ruby, Scala, Swift, Mongoid. Assim, pode-se criar um aplicativo usando qualquer uma dessas linguagens.

APIs

- **Cors:** O CORS (Cross-origin Resource Sharing) é um mecanismo utilizado pelos navegadores para compartilhar recursos entre diferentes origens. O CORS é uma especificação do W3C e faz uso de headers do HTTP para informar aos navegadores se determinado recurso pode ser ou não acessado.
- **Body-Parser:** Para tudo no Node.js existe um módulo. O body-parser é um módulo capaz de converter o body da requisição para vários formatos. Um desses formatos é o json que é basicamente um formato leve de troca de informações/dados entre sistemas, além de ser um formato leve para troca de dados é também muito simples de ler.
- **NodeMailer** (elasticemail): O nodemailer como o próprio site diz é um módulo para aplicações Node.js que permite enviar um e-mail facilmente.
- API própria [**Server.js**] para tratamentos Backend: Conceitualmente server é uma função que aceita opções e outras funções. O trabalho pesado já está implementado para que você possa se concentrar em seu projeto.
- **API MongoDB:** O MongoDB fornece persistência de dados de alto desempenho. A API de consulta do MongoDB suporta operações de leitura e gravação (CRUD) , bem como: Agregação de dados, pesquisa de texto e consultas geoespaciais. Além disso, o MongoDB fornece uma API de mecanismo de armazenamento conectável que permite que terceiros desenvolvam mecanismos de armazenamento para o MongoDB.

6. README

Para utilizar o software, é necessário a instalação do nodeJS, assim como npm ou yarn. Também é necessário a instalação do app Expo via AppStore ou PlayStore. Execute o comando “npm install” para instalar todas as dependências do projeto, e em seguida o comando “npm start” para inicializar a conexão com o mongoDB e executar o app no expo. Após a inicialização do app, será mostrado no terminal o seu endereço IP em que estará executando. Copie e cole este endereço (apenas os dígitos antes dos ‘ : ’) nos locais em que a variável “response” é inicializada dos arquivos index.js das pastas a seguir:

- Components
 - profilePic.js
- Cadastro
 - index.js
- CompletarCadastro
 - index.js
- Perfil
 - index.js
- RecuperarSenha
 - index.js
- RecuperarSenha2
 - index.js
- Signin
 - index.js
- Treinos
 - index.js

Links Úteis:

Trello utilizado: <https://trello.com/b/sGotRZ5L/quadro-scrum>

Github utilizado: <https://github.com/pedrohmelo/Gym-App>

7. Referências

Vitor Zucher. **O que é padrão MVC? Entenda arquitetura de softwares!**

. Le Wagon. Disponível em: <<https://www.lewagon.com/pt-BR>>. Acesso em: 23 de Abr. de 2022.

JavaScript. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>>. Acesso em: 23 de Abr. de 2022.

Ana Paula de Andrade. **O que é o React Native?**. TreinaWeb. Disponível em:

<<https://www.treinaweb.com.br/blog/o-que-e-o-react-native>>. Acesso em: 23 de Abr. de 2022.

Diego Fernandes. **Expo: o que é, para que serve e quando utilizar?**. Blog da Rocketseat. Disponível em: <<https://blog.rocketseat.com.br/expo-react-native/>>. Acesso em: 24 de Abr. de 2022.

Leandro Kovacs. **O que é e para que serve o MongoDB?**. Tecnblog. Disponível em: <<https://tecnoblog.net/responde/o-que-e-e-para-que-serve-o-mongodb/>>. Acesso em: 24 de Abr. de 2022.

Gabriel Machado. **O que é CORS e como resolver os principais erros**. TreinaWeb. Disponível em:

<<https://www.treinaweb.com.br/blog/o-que-e-cors-e-como-resolver-os-principais-erros>>. Acesso em: 24 de Abr. de 2022.

Felipe Batista. **Como converter os dados de uma requisição com o body-parser**.

Disponível em:

<<https://medium.com/@febatista107/como-converter-os-dados-de-uma-requisi%C3%A7%C3%A3o-com-o-body-parser-2b5b93100f00#:~:text=O%20body%2Dparser%20%C3%A9%20um,json%2C%20exatamente%20o%20que%20queremos.>>>. Acesso em: 24 de Abr. de 2022.

Bognar Junior. **NODEJS – ENVIANDO EMAIL NO NODE.JS UTILIZANDO O NODemailer**. Disponível em:

<<https://bognarjunior.wordpress.com/2017/05/21/nodejs-enviando-email-no-node-js-utilizando-o-nodemailer/#:~:text=O%20nodemailer%20como%20o%20pr%C3%B3prio,usu%C3%A1rios%20mais%20utilizam%20por%20padr%C3%A3o.>>>. Acesso em: 24 de Abr. de 2022.

Server.Js Server.Js. Disponível em: <<https://serverjs.io/documentation/>>. Acesso em: 25 de Abr. de 2022.