



**UNIVERSIDADE  
FEDERAL DO CEARÁ**  
CAMPUS DE QUIXADÁ

**CURSO DE ENGENHARIA DE SOFTWARE**

**RELATÓRIO - ANÁLISE DO DOMÍNIO - PROJETO INTEGRADO I  
EzHealth**

**Equipe:**

Paulo Ravi Feijão Leal - 485363

Pedro Anderson Costa Martins - 479257

Aurislânia Pereira Batista - 398330

**Professora:**

Carla Ilane Moreira Bezerra

**Quixadá - CE**

**Dezembro, 2020**

## SUMÁRIO

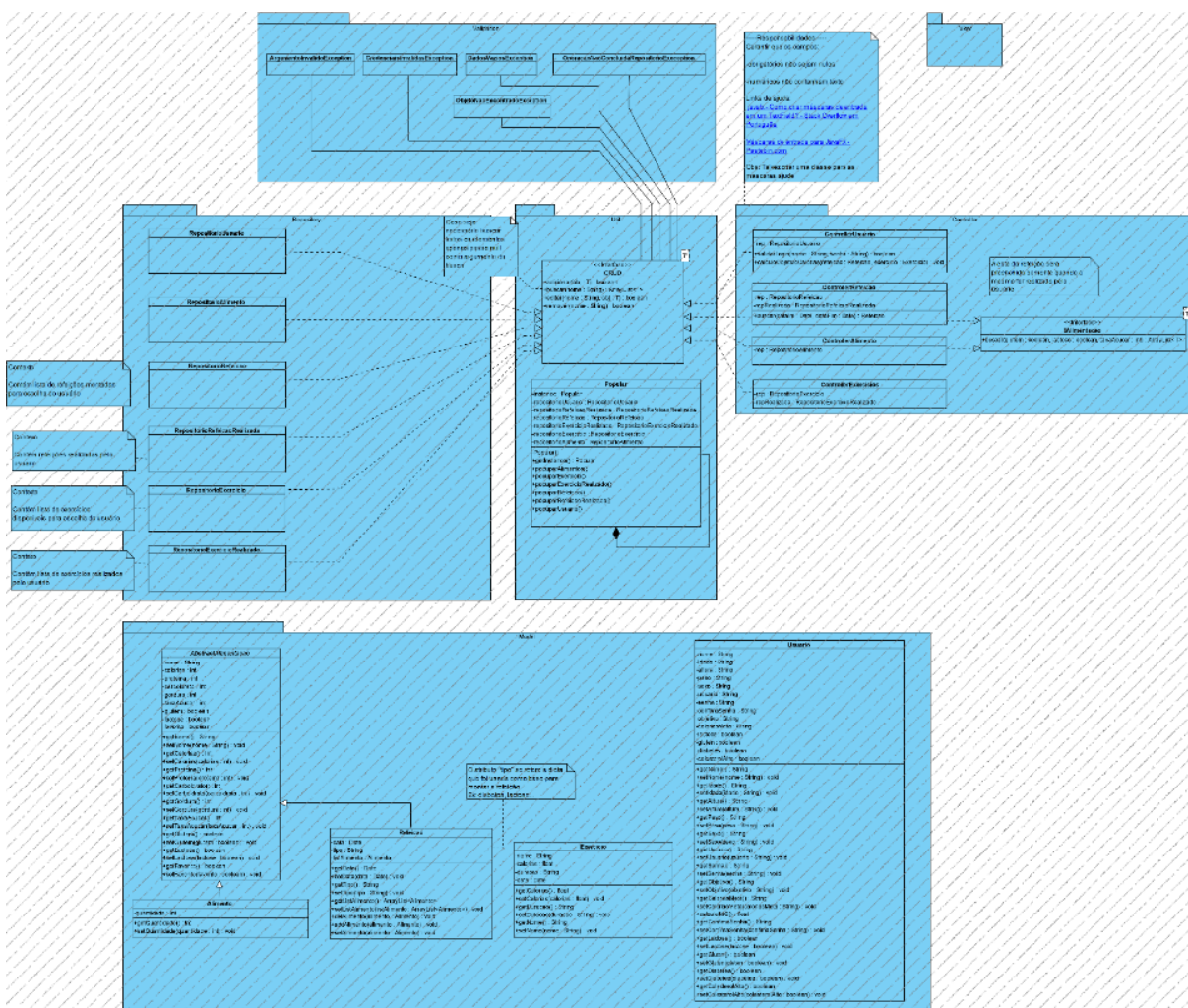
<b>MODELAGEM DO PROJETO</b>	<b>1</b>
Diagrama de Classes	1
Pacote: Model	2
Pacote: View	3
Pacote: Controller	3
Pacote: Util	3
Pacote: Repository	3
Diagramas de Sequência	4
Criar refeições	4
Realizar Exercícios	5

## 1. MODELAGEM DO PROJETO

### 1.1. Diagrama de Classes

O diagrama de Classes abaixo mostra todas as classes do sistema EzHealth, contendo também todos os seus relacionamentos, métodos e atributos.

Figura 6 - Diagrama de Classes EzHealth



A imagem em alta resolução está disponível [neste link](#) da Wiki EzHealth

Todos os dados que necessitam de persistência de dados serão salvos através de uma classe de repositório que trabalhará na manipulação dos dados salvos em arquivos com extensão txt.

#### **1.1.1. Pacote: Model**

Neste pacote criamos todas as classes de modelo do sistema. Entre elas: Usuário, Exercício, Refeição, Alimento e AbstractAlimentacao. Nessa camada do sistema implementamos as regras de negócio referentes ao contexto em que o sistema está inserido. A maioria das classes deste pacote está como uma classe concreta para sua instanciação e possui apenas uma classe abstrata (AbstractAlimentacao) para representar de forma geral atributos que estão em comum entre classes de alimento e refeição.

#### **1.1.2. Pacote: View**

Para a classe de model criamos dois layouts principais. Esses layouts servem como modelos que podem ser estendidos para as demais classes que necessitam usá-los. Existem apenas dois layouts no sistema, o layout principal nomeado de `LayoutMain` e o layout de popus nomeado de `LayoutPopup`. Eles representam respectivamente, os modelos que devem servir para apresentação principal das telas, como a home, e para apresentação de telas de pop up, utilizando `JDialog`, como é o caso da tela pop up editar exercícios. As demais classes desse pacote são classificadas entre "Telas" e "Popups", com cada uma estendendo o `LayoutMain` para as "Telas" e o `LayoutPopup` para os "Popups".

#### **1.1.3. Pacote: Controller**

Para realizar a implementação dos controladores do projeto utilizamos a interface `CRUD` do pacote `Util`, tornando obrigatório a construção do escopo das operações essenciais. Além das operações necessárias para classes controller deverá conter os métodos necessários dependendo de seu contexto. Para realizar a validação de dados e tratamento de erros foram utilizadas exceções criadas no pacote `Validation`. As exceções serão "levantadas" de acordo com sua necessidade e deverão ser tratadas no pacote `View` apresentando para o usuário uma mensagem que auxilie ele a realizar o fluxo correto na aplicação.

#### **1.1.4. Pacote: Util**

O pacote Util foi criado com o intuito de criar classes e interfaces utilitárias para o resto do sistema. Seu conteúdo pode variar de acordo com a necessidade de expansão das classes utilitárias que forem necessárias. Essas decisões devem ser feitas para cada membro da equipe. Na versão atual do sistema este pacote possui quatro classes e uma interface, sendo essa interface, nomeada de `CRUD`, que foi utilizada no pacote `Controller` descrito anteriormente. Os demais itens são classes que auxiliam na implementação dos

viewers, como é o caso da classe `ViewUtils`, `DragListener` e "DatasFormatadas". Já a classe `Popular` auxilia as classes do pacote `Repository` que será descrito mais à frente.

#### **1.1.5. Pacote: Repository**

Esse pacote possui classes com a responsabilidade de armazenarem os dados do sistema localmente na memória. Assim como foi dito antes, essas classes utilizam o singleton Popubar que está no pacote Util para realizar a criação de objetos no sistema. Além disso, as classes desse pacote implementam as operações básicas descritas na interface "CRUD" do pacote Util. Essa classe trabalha em conjunto com as classes do pacote Model e pacote Controller para realizar buscar, cadastrar, atualizações e remoções dentro do sistema.

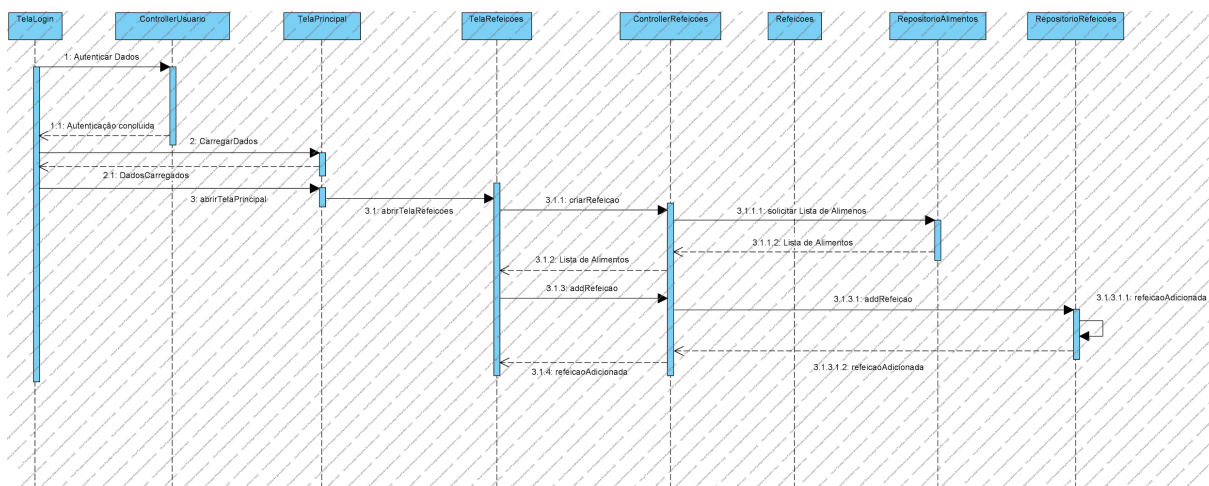
### **1.2. Diagramas de Sequência**

A seguir, são mostrados dois diagramas de sequência do sistema EzHealth. As funcionalidades escolhidas para a realização dos diagramas foram: criar refeições e realizar exercícios.

#### **1.2.1. Criar refeições**

No diagrama de sequência a seguir, está descrito o processo de criação de uma refeição personalizada pelo usuário. Ao realizar login a classe Tela Login vai autenticar os dados e caso tudo esteja correto ele retorna que a autenticação foi concluída e a partir daí o controlador manda para a classe Tela Inicial os dados carregados e ela retorna para o usuário essa tela. O usuário requisita a abertura da tela de Refeições, e a Tela principal envia essa requisição para a classe Tela Refeições. A partir daí o usuário pede para criar uma refeição nova, e a Tela de refeições manda essa requisição para o Controlador de Refeições, que pede ao Repositório de Refeições a lista de alimentos pré cadastrada. o Repositório então retorna ao Controlador de Refeições essa lista, que a retornar para a Tela de Refeições, assim o usuário a vê. Após selecionar a refeição da lista e adicionar em adicionar, o Controlador manda para o Repositório a refeição adicionada, que retorna para o Controlador uma mensagem de confirmação, e ele retorna para a Tela de Refeições essa mensagem, que chega ao usuário.

Figura 7 - Diagrama de Sequências 1 EzHealth

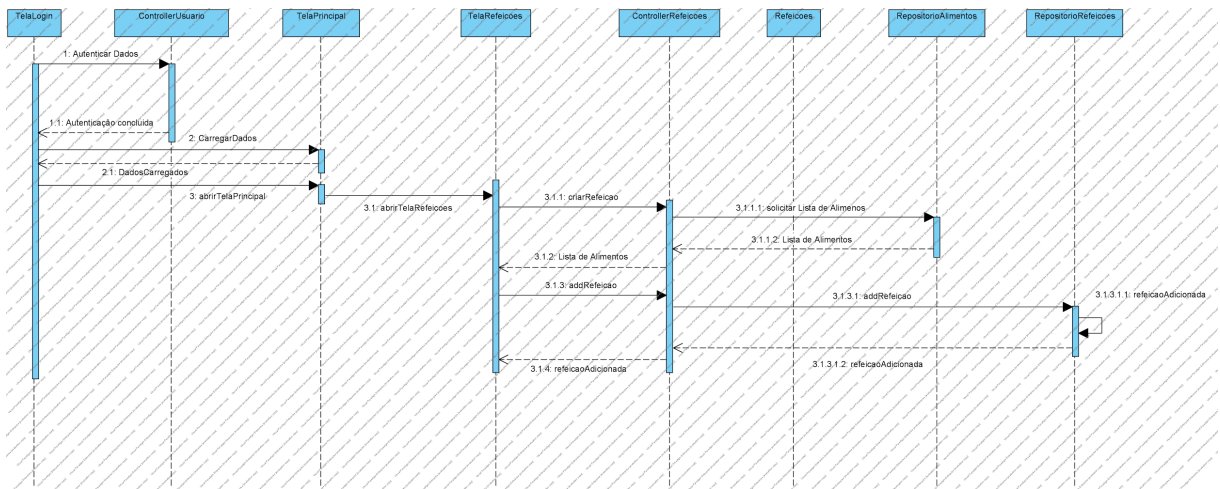


A imagem em alta resolução está disponível [neste link](#) da Wiki EzHealth.

### 1.2.2. Realizar Exercícios

No diagrama de sequência a seguir, está descrito o processo de adição de um exercício pelo usuário. Ao realizar login a classe Tela Login vai autenticar os dados e caso tudo esteja correto ele retorna que a autenticação foi concluída e a partir daí o controlador manda para a classe Tela Inicial os dados carregados e ela retorna para o usuário essa tela. Quando o usuário pede para adicionar exercícios, a Tela de Exercícios passa essa requisição para o Controlador de Exercícios e ele solicita ao Repositório a lista de Exercícios cadastrados para que o usuário selecione a realizada. A partir daí o Repositório retorna essa lista para o Controlador, e ele retorna para a Tela de exercícios que mostra ao usuário a lista de exercícios cadastrada no banco. O usuário diz que quer adicionar o exercício x e a Tela de Exercícios passa essa requisição para o Controlador, que a passa para o Repositório de Exercícios, onde ele ficará guardado. Ele retorna que o exercício foi cadastrado e a partir daí a Tela de Exercícios recalcula o objetivo de calorias diário do usuário, passando para a classe Usuário o quanto de calorias foi perdida durante o exercício.

Figura 8 - Diagrama de Sequências 2 EzHealth



A imagem em alta resolução está disponível [neste link](#) da Wiki EzHealth.