UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

TRABALHO FINAL DE DESENVOLVIMENTO FULL-STACK

O Ambiente de Consulta Web Para Nutricionistas: ANAMNEASY

GOIÂNIA - GO

2023

UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

TRABALHO FINAL DE DESENVOLVIMENTO FULL-STACK

O Ambiente de Consulta Web Para Nutricionistas: ANAMNEASY

Professor Dr. Jacson Rodrigues Barbosa

Fillipe Mendonça Albuquerque 202107677

Gabriel Aguiar de Castro Moura Rezende 202000510

GOIÂNIA - GO

2023

SUMÁRIO

PLANEJAMENTO

O trabalho final em Desenvolvimento Full-Stack exige a dedicação adicional de 64 horas (com 32h/discente), distribuídas em:

11/maio: Início da atividade. Organização do grupo: Fillipe Mendonça Albuquerque (202107677 fillipe.albuquerque@discente.ufg.br) e Gabriel Aguiar de Castro Moura Rezende (202000510 gabrielcmrezende@discente.ufg.br).

14/maio: Entendimento do escopo e análise da documentação de referência. Abertura do repositório. Definição dos tópicos do processo de consulta da nutricionista.

https://github.com/mourarezendecas/devfullstack/tree/master/anamnese-facilitada

16/maio: Uso do Vaadin em sala.

18/maio: Uso do Vaadin em sala.

23/maio: Não teve aula. Incentivo à continuação do projeto do grupo.

25/maio: Uso do Vaadin em sala. Início sobre back-end, Java Persistence API (JPA) e Hibernate.

30/maio: Definição do diagrama de classes e apresentação do escopo para o Jacson em aula. Foi apresentado o início do protótipo e partes dos códigos de autenticação do usuário usando CRN.

01/junho: Definição da estratégia para persistência dos dados dos pacientes, uso do jakarta e postgres.

15/junho: Desenvolvimento de código back-end.

20/junho: Apresentação sobre o status do projeto em sala.

29/junho: Desenvolvimento de código back-end.

06/julho: Apresentação sobre o status do projeto em sala.

20/julho: Desenvolvimento de código front-end.

27/julho: Desenvolvimento de código front-end.

03/agosto: Apresentação sobre o status do projeto em sala.

10/agosto: Apresentação final do projeto em sala.

Custo estimado

Caso este trabalho utilize como base um valor subsidiado de R$25,00/h a entrega final implicaria em um custo total de R$1600,00. No mercado é comum encontrar valores brasileiros que variam de R$100 a R$500 por hora técnica.

1. INTRODUÇÃO

Este capítulo apresenta o escopo do trabalho final da disciplina Desenvolvimento Full-Stack, com o desenvolvimento de um software de domínio e escopo definido pelo grupo de trabalho, com parte front-end e back-end, utilizando a plataforma Vaadin (disponível em: https://vaadin.com/).

A plataforma Vaadin foi sugerida para acelerar o desenvolvimento da solução full-stack, com programação em linguagem Java e demais tecnologias escolhidas pelo grupo ao longo do trabalho.

1.1. Objetivo

Este trabalho tem como objetivo o desenvolvimento de um software para consulta de nutricionistas, incluindo:

- Realizar o desenvolvimento de um um *Minimum Viable Product* (MVP), com front-end e back-end utilizando o framework de frameworks Vaadin;
- Desenvolver a persistência de dados dos pacientes;
- Funções de *create*, *read*, *update* e *delete* (CRUD) de paciente;
- Modelar funcionalidades utilizadas pela nutricionista em papel, como o preenchimento manual, para a plataforma web.

Este grupo de trabalho pretende, de acordo com a disponibilidade dos recursos, incluir no projeto:

- Segurança da informação, LGPD e dados sensíveis;
- Persistência das informações em caso de queda de conexão ou de energia;
- Tabela de refeições conforme preferências de pacientes;
- Testes unitários.

1.2. Negócio

A aplicação ANAMNEASY tem o objetivo de auxiliar a execução, organização, consultas, correlações e análise de dados por nutricionistas em plataforma web. Trata-se de um MVP, com utilização do framework Vaadin.

2. REQUISITOS, PROCESSOS E TECNOLOGIAS

Este capítulo apresenta inúmeras decisões relevantes ao projeto, como as tecnologias usadas pelo grupo de trabalho, personas, domínio, história de usuário, diagrama de classes e arquitetura relevante para implementação da aplicação.

Para o desenvolvimento do projeto ANAMNEASY foram utilizadas as tecnologias:

- Vaadin Flow para reaproveitamento do front-end e focar no desenvolvimento back-end;
- Trello e Github para gestão de atividades;
- WhatsApp e Discord para mensagens sobre o projeto;
- Figma para prototipação;
- Git e GitHub como ferramenta de versionamento e repositório do projeto;
- Google Docs para redação dos requisitos e decisões do projeto;
- Postgres como banco de dados relacional.
- Intellij Ultimate com conta de discente UFG;

Este projeto utiliza reuniões semanais (sprints) no modelo de gestão ágil, com trabalho colaborativo, sprint review, sprint planning e backlog.

2.1. Fluxo de trabalho no Vaadin

O Vaadin é um framework de frameworks e facilita o desenvolvimento full-stack por permitir o uso de anotações e abstrair trechos de códigos, modelar views e estilização através de código e integrar rapidamente a abordagem front-end com back-end.

Para as atividades em sala e para o projeto seguir o fluxo:

- Usar o Vaadin Flow em vez do Hilla. O Flow tem um foco no back-end e o Hilla no front-end;
- Definir as telas da aplicação;
- Desenvolver o front-end, com os componente do Vaadin, sem necessariamente utilizar CSS e HTML;
- Desenvolver o back-end, com as regras de negócio da aplicação;
- Utilizar os frameworks para banco de dados, como JPA e Hibernate. Vários frameworks foram desenvolvidos com base no JPA (EclipseLink, OpenJPA,

Hibernate e TopLink). Os frameworks realizam o mapeamento objeto-relacional, ou seja, entre os objetos e o banco de dados relacional.

2.2. Persona

Para este projeto o cliente final é denominado nutricionista. Trata-se de um profissional da área de saúde com formação em nutrição que pode conhecer, ou não, sobre desenvolvimento de software.

2.3. Anamnese e objetivos da nutricionista

O Anexo I apresenta um exemplo de ficha de anamnese utilizada pela nutricionista na primeira consulta. Trata-se de um levantamento detalhado denominado anamnese que, em geral, contém:

- O estilo de vida do paciente;
- Hábitos alimentares e preferências;
- Histórico médico;
- Preocupações sobre saúde, bem-estar e objetivos do paciente;
- Avaliação antropométrica e medidas primárias isoladas;
- Registros por imagem, vídeo e/ou simulações sobre a projeção do resultado do acompanhamento nutricional.

Essa avaliação individualizada auxilia na determinação de um plano alimentar, a definição de referências alimentares atingíveis de forma saudável conforme a necessidade do paciente, como:

- Redução de gordura;
- Aumento de massa muscular;
- Reeducação alimentar;
- Redução de celulite, acne e doenças na derme e epiderme;
- Controle de diabetes, colesterol, triglicérides, gastrite, pressão arterial, constipação intestinal, menopausa e outras complicações;
- Auxílio na nutrição na gestação, amamentação e desmame;
- Promoção da saúde, qualidade de vida, bem-estar e beleza do indivíduo através da alimentação.

2.4. Histórias de Usuário (HU)

No intuito de ampliar o entendimento sobre o software, foi feita uma entrevista com paciente de nutricionista, com o relato da primeira consulta. Trata-se de um paciente obeso que recebeu indicação de um primo médico. O paciente relatou dificuldade de dormir, dificuldade de respirar. No município em que ele morava, na época que começou a sentir os sintomas, havia uma clínica de nutricionistas, presente em todo o Brasil (franquia), com a primeira consulta gratuita. Nessa primeira consulta existe o roteiro:

- Na primeira consulta a nutricionista coletou hábitos alimentares, detalhes sobre todas as refeições e fez anotações em papel, no pc e no tablet.
- Ela mediu a massa, fez bioimpedância, mediu altura, mediu braço/cintura alta (acima do umbigo)/cintura baixa (abaixo do umbigo)/coxa/panturrilha.
- Ela indicou consumo de água com meta de 35 mL/kg/día, mínimo 2L.
- Ela repassou os manuais com vários tipos de comida. A clínica usa o padrão de cores (verde, amarela, laranja, vermelha) para categorizar as comidas e mostrar a evolução para o paciente. Trata-se de uma gradação conforme os resultados mês a mês. Cada tabela possui as frutas, castanhas, proteínas e demais tipos de alimentos permitidos.
- Ela indicou usar o app da franquia e fez a dieta (montou o cardápio). O paciente relatou falta de alarme como lembrete de água no app.
- O app indica receitas para cada tipo de manual (por cores), com todas as refeições do dia (café da manhã, almoço, lanche, janta, sobremesa).
- Ela indicou alimentação de 3 em 3h, com atenção às quantidades. Para esse paciente a meta era reduzir a quantidade de proteína dominante por refeição e a quantidade de alimento.
- Ela informou sobre alguns "alimentos de estimação" que afetam o estado emocional do paciente. Há permissão para consumo desses tipos de alimentos, com atenção à frequência e quantidade.
- O app mostra um gráfico de evolução dos indicadores a cada consulta.
- O app possui consulta de alimentos para ver qual categoria (cor) ele está. O paciente entende que seria importante mostrar a receita e as calorias médias desses alimentos nessa ferramenta.

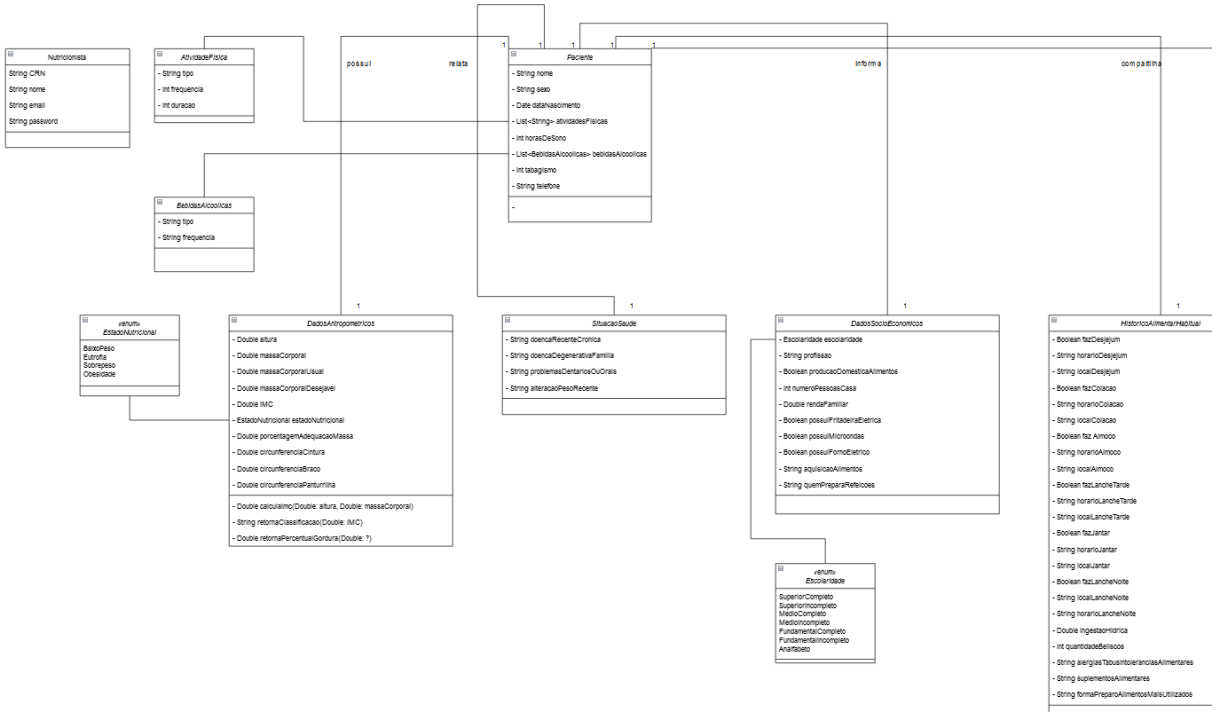
- O app faz registro de procedimentos, indica receitas diversas atualizadas todos os dias, indica produtos próprios da franquia (como barras de cereais), indica unidades franqueadas no município/estado/país (com endereço e localização GPS), indicam conteúdo para leitura e informação.
- O app possui sistema de pontuação para troca (gamificação) de alguns recursos.
- A duração da primeira consulta foi de 1h.
- A nutricionista entregou as tabelas de cores impressas, em papel de boa qualidade, e uma cartilha de propaganda da clínica, com todos os serviços estéticos.
- O agendamento pode ser feito por telefone, WhatsApp ou pelo app, de seg-sex 8-17h e sáb 8-12h, com tolerância de 15min para chegada do paciente ou haverá reagendamento. Há comunicação ativa sobre os agendamentos, com confirmações no dia anterior, sempre preciso. Acredito ser um bot inicialmente. Caso não seja confirmado há ligação.
- O paciente conversa com a nutricionista diariamente sobre a alimentação, com envio de fotos, massa dos pratos (paciente tem rotina de fazer refeições em restaurantes por kg). A nutricionista faz comentários e dá feedback na mesma hora. Ela indica os erros técnicos, quantidades e tipos de comidas permitidas, tudo pelo WhatsApp. Ela não responde sábado à tarde ou domingo o dia todo.

O Anexo I, fornecido por outra nutricionista, indica os pontos coletados na anamnese.

2.5. Diagrama de Classes

Para este projeto foi desenvolvido um diagrama de classes, focando o MVP para persistência dos dados do paciente. O documento, pelo número de classes e tamanho do documento, disponível em: https://drive.google.com/file/d/1zhgsL4wGXapPxy6Mo6riwPI6oYAzqzXX/view?usp=share_link. Foi utilizado a ferramenta web gratuita diagrams.net que permite a edição colaborativa do documento, como mostra a Figura 1.

Figura 1 - Amostra do diagrama de classes.



Fonte: próprios autores.

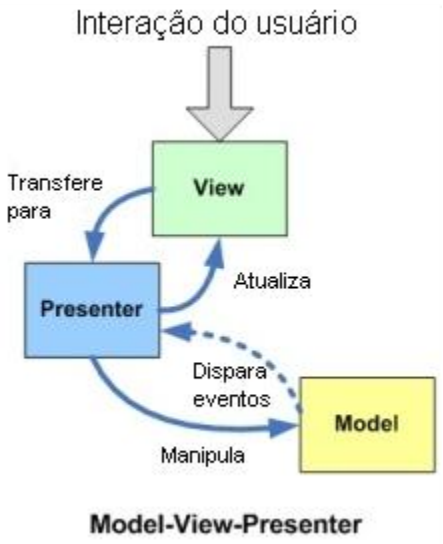
2.6. Arquitetura

Não é escopo deste trabalho avançar profundamente na análise dos atributos de qualidade relevantes para a escolha dos estilos e padrões arquiteturais. Por se tratar de um MVP foca-se aqui em definir uma arquitetura capaz de cobrir as seguintes demandas:

- Simplicidade na aplicação por ser uma solução local;
- Organizar claramente as responsabilidades no software;
- Facilitar inspeções;
- Facilitar testes;
- Facilitar detecção de falhas;
- Facilitar manutenção;
- Incluir persistência de dados;

Uma derivação da arquitetura em camadas inclui o Model-View-Presenter (MVP), como mostra a Figura.

Figura 2 - Esquema do MVP.



Model-View-Presenter

Fonte: DEVMEDIA, consultado em 13/06/2023, disponível em: <https://www.devmedia.com.br/o-padrao-mvp-model-view-presenter/3043>.

O uso de camadas permite a divisão clara de responsabilidades, fácil entendimento das funções, aplicação de regras próprias de segurança por camada e simplifica os testes, pois cada camada pode ser testada de forma independente, facilitando a identificação e correção de problemas. Os módulos independentes facilitam o desenvolvimento, manutenção e evolução de cada camada de forma separada. Isso facilita a reutilização de componentes em diferentes partes do sistema.

O uso de camadas busca trabalhar o desacoplamento das camadas de acordo com o arranjo:

- Camada de apresentação: rotas e controladores.
- Camada de negócio: serviços, regras de negócio e interface de desacoplamento com a camada de persistência.
- Camada de persistência: Jakarta e abstração das *queries*.
- Camada do banco de dados: SGBD em PostgreSQL.

3. PROTÓTIPO

Este capítulo apresenta o modelo preliminar de protótipo desenvolvido em Figma, disponível em: https://www.figma.com/file/gzZYUquBBMn2mYu97ZzFwX/Prototipa%C3%A7%C3%A3o-DFS?type=design&node-id=0-1&mode=design, como mostra a Figura 4.

Figura 4 - Protótipo do cadastro do usuário.

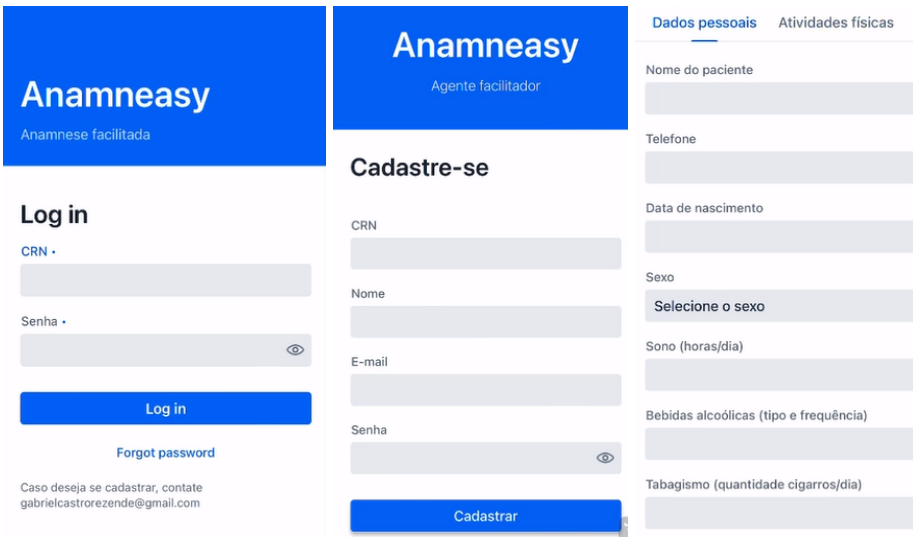
Fonte: próprios autores.

Por se tratar de um MVP, a partir das definições no protótipo e referência do front-end do próprio Vaadin, o grupo de trabalho iniciou o desenvolvimento, deixando o modelo Figma incompleto. A facilidade de desenvolvimento das views e transições, por exemplo, permitiu essa passagem direta, com iteração no desenvolvimento e visualização das telas, sem a necessidade de primeiro fazer o protótipo e depois desenvolver o código.

O uso das referência do Vaadin e a forma como ele encapsula o desenvolvimento do front-end facilitaram o desenvolvimento das *views* e conexões com as regras de negócio do projeto. O arquivo de inicialização do projeto permite configurar a seleção de temas que integram informações aplicáveis em todo o projeto, como padrão de cores, fontes e outros recursos.

Mesmo com o uso de código em Java para desenvolver as views, o Vaadin ainda exige a utilização de estilização em CSS, como utilizado para o layout principal da aplicação, como mostra a Figura 5.

Figura 5 - Exemplos de telas.



Fonte: próprios autores.

As Views desenvolvidas no projeto incluem componentes do Vaadin, como Atividades-Fisicas, Dados-Antropometricos, Dados-Pessoais, Dados-Socioeconomicos, Frequencia-Consumo-Alimentar, Historia-Alimentar-Habitual e Situacao-Saude, conforme os model previamente definidos. A Figura mostra o componente Vaadin para Atividades-Fisicas. Da mesma forma, as Views incluem as páginas da aplicação, como Anamnese, CadastroPaciente, LoginNutricionista, MainLayout, MainPageNutricionista e RegistroNutricionista. Essas páginas definem todas as rotas da aplicação.

Figura 6 - Model-Cadastro-Nutricionista.

```java
package com.dfs.views.components;

import com.vaadin.flow.component.Component;
import com.vaadin.flow.component.formlayout.FormLayout;
import com.vaadin.flow.component.textfield.TextField;

public class AtividadesFisicasComponents {
    public static Component generateAtividadeFisica(){
        TextField tipoAtividade = new TextField("Tipo");
        TextField frequenciaAtividade = new TextField("Frequência");
        TextField duracaoAtividade = new TextField("Duração");

        FormLayout formLayout = new FormLayout();

        formLayout.add(
                tipoAtividade,
                frequenciaAtividade,
                duracaoAtividade
        );

        formLayout.setResponsiveSteps(
                new FormLayout.ResponsiveStep("0", 1),
                new FormLayout.ResponsiveStep("500px", 3));

        return formLayout;
    }
}
```

Fonte: próprios autores.

Para o desenvolvimento do back-end, com o uso do Vaadin fica evidente o nível de abstração do código e uso de anotações para abstrair métodos, conexões e persistência de dados. A Figura 6 mostra como ficou o código para o cadastro de um nutricionista. Por ser uma relação 1:N, ou seja, um nutricionista atende N pacientes, as principais regras de negócio e parte mais robusta do código do back-end está voltado para o paciente, como o Model-AtividadeFisica, Model-Bebidas-Alcoolicas, Model-DadosAntropometricos, Model-Dados-Socieconomicos, Model-Historico-Alimentar-Habitual, Model-Paciente, Model-Planejamento-Dietetico, Model-Situacao-Saude. Três ENUM foram utilizados no projeto, como Escolaridade, Estado Nutricional e Sexo do paciente.

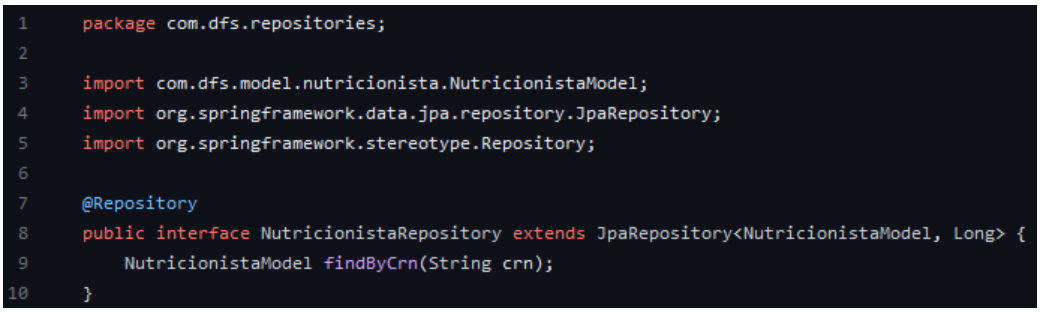
Figura 6 - Model-Cadastro-Nutricionista.

```java
package com.dfs.model.nutricionista;

import jakarta.persistence.*;
import lombok.*;

@Entity(name="Nutricionistas")
@Table(name="Nutricionistas")
@Getter
@Setter
@NoArgsConstructor
@AllArgsConstructor
@EqualsAndHashCode(of = "id")
public class NutricionistaModel {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String crn;
    private String nome;
    private String email;
    private String senha;
}
```

Fonte: próprios autores.

Conforme o modelo MVP, também foi criada uma classe NutricionistaRepository, como mostra a Figura 7.

Figura 7 - NutricionistaRepository.



```java
package com.dfs.repositories;

import com.dfs.model.nutricionista.NutricionistaModel;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

@Repository
public interface NutricionistaRepository extends JpaRepository<NutricionistaModel, Long> {
    NutricionistaModel findByCrn(String crn);
}
```

Fonte: próprios autores.

4.TESTES

Este capítulo apresenta as estratégias utilizadas para realizar testes unitários. Foi utilizado o Groovy, uma sintaxe Java compatível que permite o agrupamento de testes e fácil legibilidade do código.

Figura 4 - Exemplo dos testes em Groovy.

```groovy
import com.dfs.model.paciente.enums.EstadoNutricional
import com.dfs.model.paciente.DadosAntropometricos
import spock.lang.Specification

class DadosAntropometricosSpec extends Specification{
    void 'testa calculo de imc'(){
        when:
        double resultadoObtido = DadosAntropometricos.calculaIMC(altura, massaCorporal)
        then:
        Math.abs(resultadoEsperado - resultadoObtido) < 0.0001
        where:
        altura | massaCorporal | resultadoEsperado
        1.77   | 77            | 77/(1.77*1.77)
    }

    void 'testa retorno da classificacao'(){
        when:
        EstadoNutricional resultadoObtido = DadosAntropometricos.retornaEstadoNutricional(imc)
        then:
        resultadoEsperado == resultadoObtido
        where:
        imc  | resultadoEsperado
        18   | EstadoNutricional.BAIXO_PESO
        18.5 | EstadoNutricional.EUTROFIA
        25   | EstadoNutricional.SOBREPESO
        30   | EstadoNutricional.OBESIDADE
    }
}
```

Fonte: próprios autores.

5.PROJETOS FUTUROS

Por ser voltada para profissionais da área da saúde sugere-se o desenvolvimento de uma interface para usuários utilizando o padrão FHIR (Fast Healthcare Interoperability Resources). O FHIR tem inúmeros pontos positivos e foi indicado como padrão para a área de saúde no Brasil (Fonte: https://rnds-guia.prod.saude.gov.br/docs/rnds/tecnologias/).

As vantagens gerais do uso do FHIR são:

- Combina os recursos do HL7 com os padrões da web mais recentes;
- Aumentar a interoperabilidade no sistema de saúde, com troca de informações com mais segurança em multiplataformas;
- Padronizar e simplificar o método pelo qual os dados de saúde são trocados e auxiliar gestores de saúde, prestadores de serviço e consumidores no compartilhamento de informações de forma descomplicada por qualquer tipo de software;
- Ampliar o custo-benefício no desenvolvimento e implantação das aplicações da área de saúde e permitir o uso de *Application Programming Interfaces* (API) para simplificar o módulo de integração dos sistemas.

Não é expectativa deste trabalho o domínio sobre o padrão FHIR, uma vez que a documentação é extensa (fontes: https://hl7.org/fhir/ e http://hl7.org/fhirpath/N1/) e completamente customizada para cada contexto da área da saúde.

Outro padrão que pode ser avaliado para a aplicação é o TISS para Saúde Suplementar no Brasil.

REFERÊNCIAS

BARBOSA, J. R. Desenvolvimento Full-Stack: Notas de Aula. Instituto de Informática. Universidade Federal de Goiás. 2023-1.

ANEXO I

FICHA DE ANAMNESE NUTRICIONAL

1 DADOS PESSOAIS

Nome:

Sexo: ( ) F ( ) M

Data de nascimento:

Idade:

Atividade física (tipo, frequência e duração):

Sono (horas/dia):

Bebidas alcoólicas (tipo e frequência):

Tabagismo (quantidade cigarros/dia):

Telefone:

2 DADOS ANTROPOMÉTRICOS

Estatura (m):

Peso corporal atual (kg):

IMC (kg/m²):

Classificação:

Peso corporal usual (kg):

Peso corporal desejável (kg):

% de adequação de peso:

Circunferências:

Cintura: _____, Braço: _____, Panturrilha: _____

% de gordura:

3 SITUAÇÃO DE SAÚDE

Doença recente ou crônica:

Presença de doença crônica degenerativa na família:

Problemas dentários ou orais que interferem no consumo de alimentos:

Alteração de peso recente:

4 DADOS SOCIOECONÔMICOS

Escolaridade:

Profissão:

Produção doméstica de alimentos:

Número de pessoas na casa:

Renda familiar:

Eletrodomésticos básicos: ( ) fritadeira elétrica, ( ) microondas, ( ) forno elétrico

Aquisição de alimentos:

Quem prepara as refeições:

5 HISTÓRIA ALIMENTAR HABITUAL

( ) Desjejum

Horário: Local:

( ) Colação

Horário: Local:

( ) Almoço

Horário: Local:

( ) Lanche da tarde

Horário: Local:

( ) Jantar

Horário: Local:

( ) Lanche noturno

Horário: Local:

Ingestão hídrica diária:

“Beliscos” durante o dia:

Alergias, tabus ou intolerâncias alimentares:

Suplementos alimentares:

Alimentos que não consome:

Formas de preparo dos alimentos mais utilizados:

6 FREQUÊNCIA DE CONSUMO ALIMENTAR HABITUAL

Leite e derivados

Vegetais (quais)

Pães

Açúcar (onde)

Quitandas

Doces/balas

Cereais/massas

Suco/tipo

Carnes

Refrigerante

Ovos

Café (com o que)

Embutidos

Frituras

Leguminosas

Fast foods

Frutas

Consumo de óleo/mês:

Consumo de sal/mês:

Consumo de açúcar/mês:

Temperos utilizados em casa:

PLANEJAMENTO DIETÉTICO

1 NECESSIDADE ENERGÉTICA:

Taxa metabólica basal:

Necessidade energética:

2 MACRONUTRIENTES:

Proteínas (%):

Lipídios (%):

Carboidratos (%):

3 REFEIÇÕES:

Desjejum (%):

Colação (%):

Almoço (%):

Lanche da tarde (%):

Jantar (%):

Lanche da noite (%):

4 MICRONUTRIENTES:

Vitamina C (mg):

Sódio (mg):

Cálcio (mg):

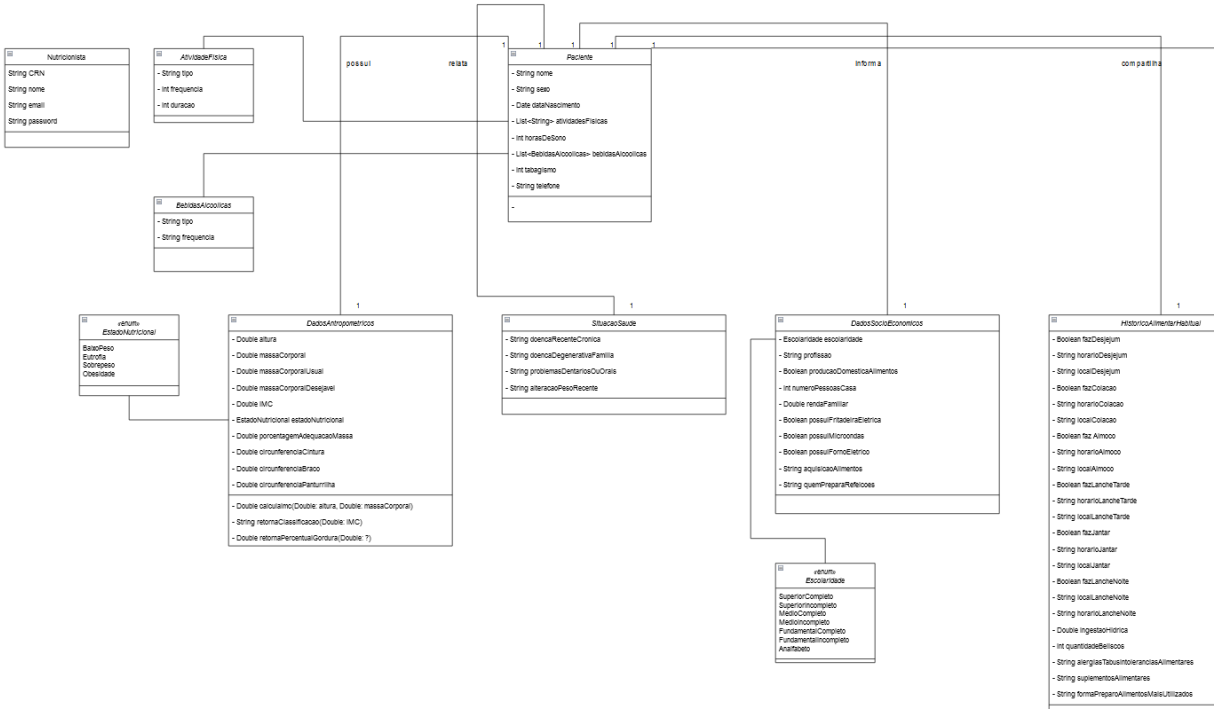
Ferro (mg):

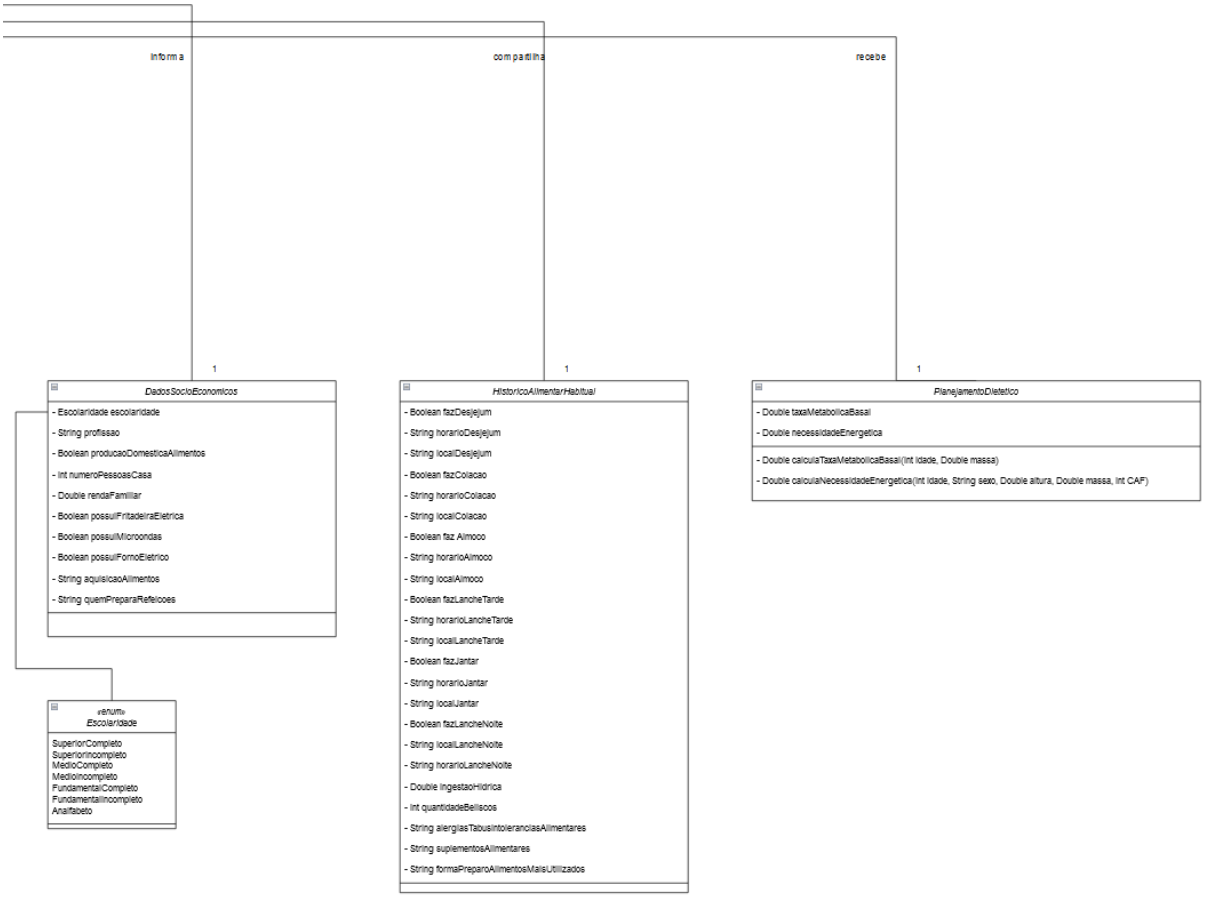
Folato (mcg):

Vitamina B12 (mcg):

APÊNDICE A

Diagrama de classes.




informa
compartilha
recebe
1
1
1
DadosSocioEconomicos
- Escolaridade escolaridade
- String profissao
- Boolean producaoDomesticaAlimentos
- Int numeroPessoasCasa
- Double rendaFamiliar
- Boolean possuiFritadeiraEletrica
- Boolean possuiMicroondas
- Boolean possuiFornoEletrico
- String aquisicaoAlimentos
- String quemPreparaRefeicoes
«enum»
Escolaridade
SuperiorCompleto
SuperiorIncompleto
MedioCompleto
MedioIncompleto
FundamentalCompleto
FundamentalIncompleto
Analfabeto
HistoricoAlimentarHabitual
- Boolean fazDesjejum
- String horarioDesjejum
- String localDesjejum
- Boolean fazColacao
- String horarioColacao
- String localColacao
- Boolean faz Almoco
- String horarioAlmoco
- String localAlmoco
- Boolean fazLancheTarde
- String horarioLancheTarde
- String localLancheTarde
- Boolean fazJantar
- String horarioJantar
- String localJantar
- Boolean fazLancheNoite
- String localLancheNoite
- String horarioLancheNoite
- Double IngestaoHidrica
- Int quantidadeBeliscos
- String alergiasTabusIntoleranciasAlimentares
- String suplementosAlimentares
- String formaPreparoAlimentosMaisUtilizados
PlanejamentoDietetico
- Double taxaMetabolicaBasal
- Double necessidadeEnergetica
- Double calculaTaxaMetabolicaBasal(Int Idade, Double massa)
- Double calculaNecessidadeEnergetica(Int Idade, String sexo, Double altura, Double massa, Int CAF)