

ENTREGA PARCIAL

Grupo: 3

Integrantes

João Pedro Castro da Rocha
Gustavo Freitas Santos
Anthony de Paula Cândido
Pedro Cremonezi Fiorini
Leandro Henrique Costa de Sousa

Tema: Diário Alimentar

Descrição: O sistema analisa a ingestão diária de alimentos do usuário e fornece um resumo detalhado de calorias e macronutrientes, baseando-se em algum banco de dados para encontrar essas informações.

User Stories

1) Como a pessoa vai encontrar o alimento que ela está procurando? E como ela vai informar as quantidades?

- Sistema de busca inteligente
- Busca pelo nome do alimento no banco de dados
- Exibe na tela todas as opções que correspondem ao nome
- Opção para escolher as quantidades de porção em unidades caseiras
- Conversão de cada unidade caseira para unidade padrão

Cartões CRC

1.1) Classe: BuscaAlimento

Atributos:

string _nomeAlimento

Operações:

BuscaAlimento(nome)
void formatarNomeAlimento()
bool procurarNomeAlimento () # Aciona a classe BancoDeDados
void exibirAlimentosEncontrados()
void selecionarAlimento()
void selecionarQuantidadePorcao() # Aciona a classe MedidasCaseiras
void cadastrarAlimentoUsuario() # Aciona a classe Alimento

Colaborações:

BancoDeDados
Alimento
MedidasCaseiras

1.2) Classe: BancoDeDados

Atributos:

```
string _idAlimentoRequerido  
string* _nomesAlimentosObtidos  
double* _nutrientesObtidos
```

Operações:

```
bool abrirBancoDeDados()  
bool fecharBancoDeDados()  
bool buscaIDAlimento()  
void getNomesAlimentos() # Nomes sao passados para classe BuscaAlimento  
void getNutrientesAlimento() # Nutrientes sao passados para classe Nutrientes
```

Colaborações: XXX

1.3) Classe: MedidasCaseiras

Atributos:

```
int quantidadePorcao  
string* nomesMedidasCaseiras  
int* valoresMedidasCaseiras
```

Operações:

```
int converteMedidaGramas()  
void cadastrarNovaMedida()
```

Colaborações: XXX

2) Como o usuario pode verificar quais os alimentos que ele consumiu naquele dia? E os que ele consumiu na última semana?

- Histórico de todos os alimentos que o usuario consumiu serão guardados localmente
- Sistema gera um arquivo com os alimentos e as quantidades da porção consumida
- Usuario pode acessar os alimentos de qualquer data anterior

Cartoes CRC

2.1) Classe: Usuario

Atributos

```
string _nomeUsuario  
ConsumoDiario* _AlimentosUsuario
```

Operações

```
void cadastrarUsuario()  
void criarHistorico() # Aciona a classe Historico
```

void editarNomeUsuario()

Colaborações:

BuscaAlimento

Historico

2.2) Classe: Historico

Atributos

Usuario usuario

Data _dataCriacaoHistorico

Operações

void criarArquivoHistorico()

void deletarArquivoHistorico()

bool abrirArquivoHistorico()

bool fecharArquivoHistorico()

void adicionarAlimentosHistorico()

void exibirHistorico()

Colaborações:

Data

Usuario

2.3) Classe: ConsumoDiario

Atributos:

Refeicao refeicoes[];

Operações:

void adicionarRefeicao()

void removerRefeicao()

Colaborações:

Refeicao

2.4) Classe: Refeicao

Atributos

Alimento alimentos[]

*string nomeRefeicao

Operações

void adicionarAlimento()

void removerAlimento()

Colaborações:

Alimento

2.5) Classe: Alimento

Atributos

Essas são as 3 informações disponíveis no banco de dados sobre o alimento

string _nomeAlimento

string _variedadeAlimento

string _tipoPreparo

int quantidadeGramas

Operações

Alimento(_nomeAlimento, quantidadeGramas) # Alimento é inicializado pela Classe BuscaAlimento

Alimento(_nomeAlimento, _variedadeAlimento, _tipoPreparo) # Alimento também é inicializado pela Classe CadastroAlimento

Colaborações:

BuscaAlimento

CadastroAlimento

2.7) Classe: Data

Atributos:

int dia, mes, ano

Operações

void lerData()

void exibirData()

Colaborações: XXX

3) Como o usuário visualizará as quantidades de nutrientes totais após consumir um ou mais alimentos?

- Usuário pode pedir as informações nutricionais do alimento, da refeição ou de tudo que consumiu no dia

- Sistema mostra os valores totais de carboidratos, proteínas, lipídios, minerais e vitaminas

Cartões CRC

3.1) Classe: Nutrientes # Classe-filha de Alimento

Atributos:

double* _vetorNutrientes # armazena a quantidade de todos os nutrientes

string _nomeAlimento # da classe Alimento

int quantidadeGramas # da classe Alimento

Operações:

```
void procurarNutrientesAlimento() # Aciona a classe BancoDeDados
void calcularNutrientesPorcao()
void exibirNutrientesAlimento() # Aciona a classe Tabela
void selecionarNutrientesAlimento()
void exibirClassesNutrientes()
```

Colaborações:

BancoDeDados

3.2) Classe: Tabela # Classe para mostrar tabelas

Atributos:

```
string* _cabecalho
string** _matrizElementos
int _numLinha
int _numColunas
```

Operações:

```
void exibirTabela()
void ordenarTabela()
```

Colaborações: XXX

4) Como o usuario pode adicionar um alimento que nao está no banco de dados?

- Usuario poderá cadastrar o alimento e as suas informações nutricionais
- O Banco de Dados será atualizado localmente
- Uma nova linha será adicionado ao banco de dados com o nome, variedade e tipo de preparo do alimento em questão

Cartões CRC

4.1) Classe: CadastroAlimento

Atributos:

```
Alimentos _infoAlimento
Nutrientes _infoNutrientes
```

Operações

```
void cadastrarNomeAlimento() # Aciona a Classe Alimentos
void cadastrarNutrientes() # Aciona a Classe Nutrientes
```

Colaborações:

Alimentos
Nutrientes

4.2) BancoDeDados (mesma classe do 1.2 mas com outras operacoes)

Atributos:

```
...
static int num_linhas
static int num_colunas
```

Operações:

```
...
void adicionarLinha()
void adicionarColuna()
void removerLinha(int pos_linha)
void removerColuna(int pos_coluna)
void escreverCelula(int pos_linha, int pos_coluna)
void limparCelula(int pos_linha, int pos_coluna)
```

Colaborações: XXX

5) Como o usuario pode definir as suas metas diárias de calorias e outros nutrientes?

- Opção de “Definir Metas Diárias” no sistema
- O usuario pode especificar limites minimos e maximos para cada nutriente
- Outra opção de “Avaliar Metas Diárias”
- Ao selecionar, será mostrado as porcentagem de quanto falta para atingir a meta ou o quanto já passou
- Sistema tbm vai gerar graficos de barra para mostrar o quão proximos das metas o usuario está

5.1) Classe: MetasDiarias # precisa se comunicar com a Classe Historico

Atributos:

```
Nutrientes _nutrientesAnalisados
double* metasNutrientes
double* porcentagensMetas
```

Operações:

```
void lerMetasNutrientes()
void acessarHistorico() # Aciona a classe Historico
void calcularPorcentagens()
void exibirPorcentagens()
```

Colaborações:

```
Nutrientes
Historico
```

5.2) Classe: Grafico

Atributos:

Janela _janelaGrafico
Alimento _alimentoExibido
Nutrientes _nutrienteExibidos
MetasDiarias metasExibidas

Operações:

void criarJanelaGrafico() # Aciona a classe Janela
void exibirGraficoPorcentagem()
void exibirGraficoValorAbsouto()

Colaborações:

Janela
Alimento
Nutrientes
MetasDiarias

5.3) Janela

Atributo:

int _largura
int _altura

Operações:

void abrirJanela()
void minimizarJanela()
void fecharJanela()

Colaborações:

Window (classe externa – da biblioteca SFML)