UcONTROL

IPCA



12607 – Joel Carvalho

12251 – Henrique Costa

12715 – Daniel Eleutério

**Computação Móvel ll**

Índice

[1 Introdução 3](#_Toc471737625)

[1.1 Enquadramento 3](#_Toc471737626)

[1.2 Apresentação do projeto 3](#_Toc471737627)

[1.3 Tecnologias utilizadas 3](#_Toc471737628)

[1.4 Organização do relatório 4](#_Toc471737629)

[2 Desenvolvimento 5](#_Toc471737630)

[2.1 Persistência de dados 5](#_Toc471737631)

[2.1.1 Dados Exemplo 5](#_Toc471737632)

[Tabela 1 Tabela Dados\_Utilizador 5](#_Toc471737633)

[Tabela 2 Tabela Unidade 6](#_Toc471737634)

[Tabela 3 Tabela Tipo\_Refeicao 6](#_Toc471737635)

[Tabela 4 Tabela Alimentos 6](#_Toc471737636)

[Tabela 5 Tabela Refeicoes 6](#_Toc471737637)

[Tabela 6 Tabela Consumos 6](#_Toc471737638)

[2.1.2 DER 7](#_Toc471737639)

[2.2 Implementação e processos 8](#_Toc471737640)

[3 Conclusões 9](#_Toc471737641)

[3.1 Objectivos realizados 9](#_Toc471737642)

[3.2 Limitações & trabalho futuro 10](#_Toc471737643)

[3.3 Apreciação final 11](#_Toc471737644)

[5 Anexos 12](#_Toc471737645)

# Introdução

## Enquadramento

O presente trabalho consiste numa aplicação *Android*, focalizada na área da domotica.

A aplicação é capaz de ajudar o utilizador a perder, ganhar ou manter o seu peso, através do cálculo da taxa metabólica basal (tmb), a aplicação sugere um número de calorias que o utilizador terá que comer diariamente.

Após isso, o utilizador insere todos os alimentos que ingeriu ao longo do dia.

Para a realização deste projeto foram utilizadas várias ferramentas de diferentes disciplinas.

## Apresentação do projeto

O projeto centralizar-se-á, no desenho e no desenvolvimento uma aplicação que permite ao utilizador registar todos os alimentos que come diariamente, conseguindo através do cálculo das calorias atingir um determinado objetivo.

Objetivos a cumprir nesta aplicação:

1. O utilizador deve conseguir introduzir alimentos;
2. O utilizador pode mudar e rever os alimentos inseridos;
3. O utilizador pode consultar o seu peso ao longo do tempo;
4. O utilizador pode escolher que plano deve seguir, se pretende perder, manter ou ganhar peso;
5. A interface deve ser simples e atrativa;
6. A utilização da aplicação deve ser rápida;

## Tecnologias utilizadas

Para a realização deste projeto prático foram utilizadas várias ferramentas, nomeadamente:

* No âmbito da disciplina de Interação de Dispositivos Móveis:

*NinjaMock* – Planear e criar *Wireframes*;

*Photoshop* – Criar os ecrãs para a aplicação;

*Flinto* – Tornar os ecrãs de Photoshop interativos;

* Computação Móvel:

*Android Studio* – Criar a aplicação;

*SQLiteStudio* – Analisar a base de dados;

* Análise e Projeto de Sistemas:

Visio – Criação de diagramas;

* Armazenamento e Acesso a Dados:

*SQLite* – Criação da base de dados;

## Organização do relatório

Inicialmente, começamos por tratar da análise do projeto, onde exibimos os diagramas, que descrevem o que o utilizador pode realizar com a aplicação, e os tipos de requisitos funcionais e não funcionais.

Posto isto, observamos o capítulo do desenvolvimento, onde contém, com descrição, tudo aquilo que foi realizado ao longo do projeto, até obter o produto final.

De seguida é observável, um capítulo dedicado à disciplina de Armazenamento e Acesso a Dados, mostrando o Diagrama de Entidade Relação, composto por relações e tabelas.

Relativamente à disciplina de Interação com Dispositivos Móveis, são descritos alguns detalhes do desenho em *wireframe*.

Para concluir, a parte da implementação do código e passar de toda esta teoria para a prática.

# Desenvolvimento

Este projeto, foi concebido por três alunos, sendo mais fácil a divisão de tarefas e partilha de conhecimentos.

Inicialmente, o maior foco foi a idealização do projeto. Iniciamos pelo desenho da aplicação em papel, para discutirmos o posicionamento dos elementos a aparecer no ecrã. Sequentemente, passamos do papel para o *NinjaMock*, onde é possível colocar *links* entre as páginas e visualizar tudo com maior rigor. Posto isto, definimos cores e o tipo de letra da nossa aplicação, usando o Photoshop.

De seguida, cuidamos da análise do projeto, nomeando requisitos funcionais e requisitos não funcionais, estruturamos a base de dados (DER) e construímos os diagramas necessários para a construção da aplicação, com o auxílio do *Visio.*

Após a análise, passamos para o *AndroidStudio*, construindo contratos e o *OpenHelper*. Durante a construção da base de dados no *AndroidStudio*, fomos fazendo, em paralelo, a base de dados em *Sql Server*, para uma mais fácil e melhor construção da base de dados.

Para terminar, algumas das maiores dificuldades sentidas, foi na implementação do código, pois este é um processo demorado, que exige bastante esforço e dedicação.

## Persistência de dados

### Dados Exemplo

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Nome** | **Género** | **Altura** | **Peso** | **Idade** | **Grau\_Atividade** | **Meta\_Atingir** | **Calorias\_Consumir** |
| 1 | Manuel | Masculino | 180 | 75 | 30 | Ativo | Manter peso | 1855.20 |

Tabela 1 Tabela Dados\_Utilizador

|  |  |  |
| --- | --- | --- |
| **IDUnidade** | **Nome** | **Sigla** |
| 1 | Gramas | G |

Tabela 2 Tabela Unidade

|  |  |
| --- | --- |
| **IdTRefeicao** | **NomeRefeicao** |
| 1 | Almoço |
| 2 | Jantar |

Tabela 3 Tabela Tipo\_Refeicao

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **IDUnidade** | **Nome** | **QtdAconselhada** | **Unidade** | **Calorias** |
| 1 | Arroz | 100 | 1 | 120 |

Tabela 4 Tabela Alimentos

|  |  |  |
| --- | --- | --- |
| **IdRefeicao** | **Data** | **IdTRefeicao** |
| 1 | 2016-2-1 | 1 |

Tabela 5 Tabela Refeicoes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **IdAlimento** | **Quantidade** | **Calorias** | **IdRefeicao** |
| 1 | 1 | 100 | 100 | 1 |

Tabela 6 Tabela Consumos

### DER



Fig 4 DER da aplicação

O Diagrama de Entidade Relação, é composto por 6 tabelas, onde apenas, cinco estão relacionadas entre si.

A tabela “Dados\_Utilizador”, serve apenas para a recolha dos dados do utilizador, nos quatro primeiros passos da nossa aplicação.

A tabela “Alimentos”, é composta por 5 atributos, sendo um deles uma chave estrangeira (Unidade) e outra chave primária (IdAlimento).

A tabela “Unidade”, é composta por 3 atributos. Esta tabela é importante porque todos os alimentos têm um certo tipo de unidade, daí o relacionamento com a tabela “Alimentos”, para evitar a redundância de dados.

A tabela “Tipo\_Refeicao”, é composta por 2 atributos. Um que identifica a refeição e outro com o nome da refeição (Pequeno-Almoço, Almoço, Jantar e Outras-Refeições).

A tabela “Refeicoes”, é composta por 3 atributos, com a chave primária “IdRefeicao” e uma chave estrangeira, “IdTRefeicao”, relacionada com a tabela “Tipo\_Refeicao”. Esta tabela, identifica o tipo de refeição e a data em que foi consumida.

Finamente, a tabela “Consumos”, é composta por 5 atributos, e através dela o utilizador pode ver todos os alimentos que consumiu consoante a data do consumo e a respetiva refeição.

## Implementação e processos

Relativamente à programação, dividimos o projeto em *packages* para uma melhor organização do projeto.

Inicialmente, tratamos da elaboração da base de dados, criando tabelas, bem como, as suas relações (Ver anexo 2).Após isto, iniciamos com a construção de *layouts* e classes.

Primeiramente, cuidamos dos objetivos mais importantes, como por exemplo deixar o utilizador criar o seu perfil, guardando na base de dados, todos os seus dados.

De seguida, criamos o *layout* principal, que contém os três botões principais da aplicação, que permite ao utilizador atualizar o seu peso, verificar os alimentos ingeridos e adicionar alimentos.

Na classe inserir peso, o utilizador pode atualizar o peso frequentemente.

Na classe ver alimentos ingeridos, permite ao utilizador remover e visualizar todos os alimentos, que ele ingeriu ao longo do dia, consoante o tipo de refeição.

Para concluir, na classe adicionar alimentos, possibilita ao utilizador introduzir os alimentos que consumiu, pesquisar todos os alimentos existentes na base de dados, bem como, eliminar alimentos da base de dados (Ver anexo 3).

# Conclusões

Em suma, este trabalho foi realizado com satisfação, onde os objetivos mais relevantes, foram realizados, embora tivéssemos alguns altos e baixos.

Em programação foi onde sentimos mais dificuldades. Começando por alguns gráficos que foram inicialmente idealizados, mas acabamos por não os realizar devido ao facto de ser uma tarefa algo trabalhosa e de um certo grau de dificuldade elevado.

Outra das dificuldades , foi na inserção de dados nas tabelas, ou seja, a ideia pretendida era que numa determinada tabela aparecessem dados, mas essa tabela continha “Id’s”, de tabelas relacionas.

Mais tarde surgiu outra dificuldade, que era calcular a idade do utilizador, fundamental para calcular a taxa metabólica basal.

Por fim, outra adversidade foi em colocar a data no *select* verificar os alimentos que ingeriu num determinado dia.

Este trabalho foi muito enriquecedor, porque nos fez refletir e trabalhar em conjunto, partilhando ideias entre nós. Deparamo-nos com uma situação bastante semelhante àquelas que existem no mundo de trabalho e deixou-nos mais contextualizado com o que iremos enfrentar no mundo de trabalho.

Por sua vez, comprometemo-nos a elaborar estas adversidades com o acumular de experiencia e conhecimento, para que seja assim possível melhorar a aplicação.

Relativamente, à relação entre os elementos do grupo foi a melhor, porque nos entendemos bastante bem e já fizemos outros projetos em conjunto. Para concluir, reiteramos que foi um grande prazer realizar este trabalho.

## Objectivos realizados

Os objetivos que foram realizados são:

1. O utilizador pode introduzir alimentos;
2. O utilizador pode mudar e rever os alimentos inseridos;
3. O utilizador pode escolher que plano deve seguir, se pretende perder, manter ou ganhar peso;
4. O utilizador pode ver calorias consumidas;
5. O utilizador pode atualizar o seu peso;
6. O utilizador pode criar novos alimentos, ou elimina-los.

## Limitações & trabalho futuro

A aplicação tem algumas limitações, como os gráficos relativos ao peso durante a semana e ao das calorias que o utilizador consumiu durante um determinado período de tempo.

Outra das limitações é não calcular a idade precisa do utilizador. Na realidade ela calcula a idade, apenas entre o ano. Ou seja, se o utilizador nasceu na data “02-08-1993”, a aplicação vai dizer-lhe que ele tem 23 anos, o que não está correto porque o utilizador ainda não celebrou o seu vigésimo terceiro aniversário, tendo ainda 22 anos.

A última limitação encontrada foi colocar a data para que o utilizador conseguisse verificar que alimentos foram ingeridos num determinado dia. Acabamos mesmo por ter que inserir a data manualmente nos *selects*, para ocultar essa falha até conseguirmos resolver o problema.

Com tudo, estes estas lacunas são para resolver, porque são partes fundamentais e imprescindíveis na aplicação. Com o ganhar de experiencia e conhecimento pretendemos corrigir todos estes problemas e tornar a aplicação utilizável de forma segura sem qualquer tipo de erro ou falha.

## Apreciação final

Este trabalho, foi dos mais trabalhosos e benéficos, que alguma vez elaboramos, porém foi também aquele que nos deu maior prazer, porque na verdade sentimos que podíamos fazer algo consistente, alguma coisa que tivesse futuro.

Embora tenhamos consciência que é difícil conseguir alcançar sucesso com a nossa aplicação, devido à competitividade do mercado, mas esperamos atingir determinado patamar devido ao esforço e dedicação que tivemos.

A nossa aplicação nasceu com o propósito de satisfazer certas necessidades das pessoas, porque aplicações destas existem muitas, mas a nossa ideia era tentar melhorar de alguma forma.

5 Anexos