

**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA
ETEC DA ZONA LESTE**

**Ensino Médio com Habilitação Profissional de Técnico em
Desenvolvimento de Sistemas**

**Luan Avila Fernandes
Luan Gonçalves Ribeiro
Thiago Teruel Teixeira**

Economic

Pré-projeto apresentado ao Curso do Ensino Médio com Habilitação Profissional de Técnico em Desenvolvimento de Sistemas, orientado pelo Prof. Jeferson Roberto de Lima, como requisito final para obtenção do título de Técnico em Desenvolvimento de Sistemas.

São Paulo

2021

Luan Avila Fernandes
Luan Gonçalves Ribeiro
Thiago Teruel Teixeira

Economic

Pré-projeto apresentado ao Curso do Ensino Médio com Habilitação Profissional de Técnico em Desenvolvimento de Sistemas, orientado pelo Prof. Jeferson Roberto de Lima, como requisito final para obtenção do título de Técnico em Desenvolvimento de Sistemas.

São Paulo
2021

DEDICATÓRIA

Dedicamos este trabalho a todos os professores que nos auxiliaram durante a construção e desenvolvimento, aos nossos familiares que nos apoiaram para a realização deste, aos nossos colegas de classe e de grupo que nos ajudaram e contribuíram para a finalização no nosso projeto.

AGRADECIMENTOS

Pela conclusão deste trabalho, gostaríamos de agradecer a Deus por ter nos dado tudo que sempre precisamos para alcançar nossos objetivos. Agradecemos de coração também a Etec da Zona Leste por toda a ajuda e pela educação dada. Temos este lugar como nosso segundo lar. Aos nossos orientadores e demais professores que tanto nos ajudaram a chegar na conclusão deste trabalho. Toda nossa gratidão para nossas famílias, amigos e colegas de curso. Vocês são o motivo do nosso empenho, esforços e dedicação. Por fim, agradecemos a todas as pessoas que de alguma forma estiveram envolvidas na realização deste trabalho.

“Talvez não tenha conseguido fazer o melhor, mas lutei para que o melhor fosse feito. Não sou o que deveria ser, mas Graças a Deus, não sou o que era antes”.

MARTHIN LUTHER KING

RESUMO

Este projeto tem como objetivo auxiliar na economia de gastos com produtos fornecidos por mercados, através de plataformas e aplicativos virtuais distribuídos para os clientes e comércios. Com estudos sobre modelagens de bancos, bancos de dados, construção de e desenvolvimento de softwares para plataformas web e mobile, o website e o aplicativo receberão dados disponibilizados pelos mercados e estes serão distribuídos para os usuários e clientes destas redes comerciais, ajudando-os a gerenciar as compras e auxiliando-os a escolher um local para obtenção destes.

Palavras-chave: Economia. Produtos fornecidos. Mercados. Gerenciar. Compras.

ABSTRACT

This project has the objective to assist in the economy of spending with products supplied by markets, through virtual platforms and applications spread for the clients and commerce. With study about data modeling, databases, construction and development of software for web and mobile platforms, the website and the app will receive data made available by the markets along with this will be distributed to the users and clients with this commercial networks, helping managements the purchase besides assisting choose the location for getting this.

Keywords: Economy. Products supplied. Markets. Managements. The purchase.

LISTA DE ILUSTRAÇÕES

Figura 1 - Logotipo do aplicativo AondeConvem	13
Figura 2 - Logotipo do aplicativo Facily	13
Figura 3 - Logotipo da marca Extra	14
Figura 4 - Variáveis no PHP	15
Figura 5 - Código básico de Dart rodando um for	16
Figura 6- Importações de biblioteca do Flutter e do Firebase	17
Figura 7 - Comandos para agilizar o desenvolvimento no Flutter	17
Figura 8 - Imagem das pastas relacionadas ao Android e ao IOS	17
Figura 9 - Classes Statefull e Stateless do flutter	18
Figura 10 - Programa em Dart demonstrando na prática os Widgets	19
Figura 11 - Tabela de um banco MySQL	20
Figura 12 - Código em Kotlin	22
Figura 13 - Diagrama de Atividade	23
Figura 14 - Diagrama de Sequência	23
Figura 15 - Diagrama de Caso de Uso	24
Figura 16 - Diagrama de caso de uso	25
Figura 17 - Diagrama de atividade	26
Figura 18 - Tela de Login	27
Figura 19 - Tela Perfil do Mercado	28
Figura 20 - Tela de configurações avançadas do mercado	28
Figura 21 - Tela para adicionar produtos ao destaque e a oferta	29
Figura 22 - Tela de visualização dos produtos do mercado	29
Figura 23 - Tela de cadastro de produtos	30
Figura 24 - Tela de Edição de produtos	30
Figura 25 - Tela de exclusão do produto	31
Figura 26 - Tela Mercado	31
Figura 27 - Tela Produtos do Mercado	32
Figura 28 - Tela Produtos	32
Figura 29 - Tela Pesquisa	33
Figura 30 - Tela Listagem Principal	33
Figura 31 - Tela Lista de Compras	34
Figura 32 - Tela Usuário	34

Figura 33 - Tela de Comparação	35
Figura 34 - Tela de escolha de Mercados	35
Figura 35 - Tela de cadastro de mercado.....	36
Figura 36 - Tela de cadastro de usuário.....	36

LISTA DE SIGLAS

Backend as a Service (BaaS)

Foreign Key (FK)

Java Virtual Machine (JVM)

Primary Key (PK)

Relational Database Management System (RDBMS)

Sistema gerenciador de Banco de Dados (SGBD)

Structured Query Language (SQL)

Software Development Kit (SDK)

SUMÁRIO

1	Introdução	12
2	Referencial Teórico	13
2.1	Aplicativos Semelhantes	13
2.2	PHP	14
2.3	DART	15
2.4	FLUTTER.....	16
2.5	Banco de Dados.....	19
2.5.1	MySQL	20
2.6	JAVA.....	20
2.7	Kotlin	21
2.8	Diagramas.....	22
3	Desenvolvimento	25
3.1	Diagrama de caso de uso	25
3.2	Diagrama de atividade	25
3.3	Telas	27
4	Conclusão	38
	Referências.....	39

1 INTRODUÇÃO

Muitas das vezes estamos fazendo determinada ação, e precisamos de um produto para terminá-la, seja um ovo para um bolo, ou um cadarço para o tênis. Vamos pegar a parte do bolo, você precisa de um produto, mas num momento difícil economicamente, não quer gastar muito para fazer um simples bolo. Temos a questão do transporte, e se aquele lugar não tiver o produto desejado? o que eu faço?

Simple, com o uso do aplicativo, *economic*, você ficará sabendo os melhores preços, e o lugar mais acessível para sua compra. É algo simples, porém muito útil, nos dias de hoje, não havia nenhum tipo de programa que fizesse esse tipo de ação, então viemos suprir essa necessidade.

De acordo com Mohsin (2020), hoje em dia, as vendas de produtos pessoais, está em um declínio imenso. Uma pesquisa realizada pelo NZN *Intelligence* em 2019 mostrou que 74% dos consumidores brasileiros preferem comprar online. Por esse fato, vários aplicativos de venda e entrega estão sendo criado atualmente.

MCC-ENET (2021) diz que o E-commerce brasileiro cresce 73,88% em 2020, dados de vendas online, revela índice MCC-ENET. O e-commerce segue em expansão no Brasil. No mês de dezembro de 2020, registraram alta de 53,83%, em relação ao mesmo período de 2019.

Utilizaremos livros, sites, dados, diagnósticos, PHP, DART, FLUTTER, BANCO DE DADOS, SQL, MYSQL. Também usaremos dados de estabelecimentos pessoais, como mercados, na inserção dos produtos.

2 REFERENCIAL TEÓRICO

Neste tópico, veremos as tecnologias utilizadas para a construção do nosso projeto e referenciais que ajudarão a entender a parte técnica deste.

2.1 Aplicativos Semelhantes

Neste tópico será apresentado alguns aplicativos com objetivos similares ao nosso projeto.

De acordo com Beggiora (2021), um dos aplicativos de e-commerce mais utilizados é o AondeConvem, e está disponível como site ou app para Android e iOS, já ultrapassando os dez milhões de downloads. Nele, você confere promoções da sua cidade por meio de encartes e pode escolher as ofertas que mais te interessam para salvar para mais tarde. Este aplicativo gerencia a escolha do cliente, e os informa, podendo guardar no carrinho ou comprar na hora.

Na Figura 1 veremos o primeiro item apresentado, mostrando a logo do AondeConvem.

Figura 1 - Logotipo do aplicativo AondeConvem



Fonte: (OBS Virtual, 2020)

A partir de pesquisas feitas pela Beggiora (2020) o Facity é um aplicativo gratuito para celulares Android e IOS que permite a compra de produtos com descontos de até 70%. Para isso, o usuário deve criar grupos de compra e compartilhar com seus amigos. No aplicativo, estão disponíveis eletrônicos, acessórios para celular, mercado, bebidas e até mesmo hortifruti. Os produtos são entregues em todo o Brasil.

O logo do aplicativo apresentado está localizado na Figura 2.

Figura 2 - Logotipo do aplicativo Facity



Fonte: (FACILY, 2021)

Salles (2020), afirma, com certeza, que você já ouviu falar de uma das maiores marcas

de mercados e varejos do mundo, ela é lembrada em todo o Brasil quando citado algo referenciado a mercado.

O Extra possui o Aplicativo Clube Extra, que oferece ao participante do programa acesso a benefícios exclusivos, como ofertas personalizadas.

Trouxemos a logo do Extra, apresentada na Figura 3

Figura 3 - Logotipo da marca Extra



Fonte: (EXTRA Divulgação, 2021)

2.2 PHP

Dall (2018) *Personal Home Page Tools* (PHP), criada no ano de 1994 pelo programador Rasmus Lerdorf. Foi criada por um conjunto de scripts escritos em linguagem C. Ao longo da expansão da linguagem, Rasmus adicionou o uso do banco de dados na linguagem, sendo chamada pelos programadores da época de *Forms Interpreter* (FI). Hoje sua sigla significa: *Hypertext Preprocessor*, que trazendo para o português fica pré – processador de hipertexto.

Niederauer (2017) afirma que a linguagem PHP, é uma das mais utilizadas no quesito programação web, sendo usado em milhões de sites espalhados pela internet. Uma característica que diferencia essa linguagem das demais é sua afinidade em interagir com o mundo web. Ele aponta pontos positivos do uso do PHP, sendo eles: o fato da linguagem ser gratuita, instalando seu pacote pelo site principal da linguagem e sua estrutura “embutida” no HTML, sendo misto o uso do HTML e PHP no meio do programa. PHP aceita vários tipos de banco de dados, alguns dos aceitos são, MySQL, PostgreSQL, Oracle, SQL Server, dentre outros. O PHP pode ser executável em, Linux, Windows, ou variantes Unix, como Solaris e OpenBSD.

Moraz (2005) cita alguns tipos de dados suportados em PHP:

- *int*: é um tipo de variável que suporta alguns tipos de números, por exemplo, positivos em base decimal, negativos em base decimal, inteiro na base octal, no caso 0, e inteiro na base hexadecimal.
- *float*: é o valor que aceita o ponto flutuante em sua composição, por exemplo, 1.752 e 530.000.
- *string*: Uma *string* é uma série de caracteres, onde um caractere é o mesmo que um byte. Isso significa que o PHP possui suporte a um conjunto de apenas 256 caracteres.
- *array*: *Array* pode ser definido como mapeamento ou vetores indexados. De uma forma mais clara, um valor de tipo *array* é um dicionário em que os índices são as chaves de acesso. Sendo válido ressaltar que, podem ser qualquer tipo de valor, não somente inteiros.
- *object*: Um *object* é uma instanciação de uma classe.

Os tipos de dados são declarados de acordo coma Figura 4.

Figura 4 - Variáveis no PHP

```

1  <?php
2      $variavel1 = 1;
3      $variavel2 = 1.5;
4      $variavel3 = 'Economic';
5      $array = ['Luan Avila', 'Luan Gonçalves', 'Thiago Teruel'];
6
7  ?>

```

Fonte: Aatoria Própria, 2021.

2.3 DART

Dart é uma linguagem de programação desenvolvida pela google utilizando como base as linguagens C e *JavaScript*. Ela possibilita a programação de sistemas e aplicativos multiplataformas, assim, podendo realizar a construção de um aplicativo por exemplo, para usuários de IOS e ANDROID (SILVA, 2019).

O *Dart* tem capacidade de interagir com a *thread* nativa dos aparelhos em que estão rodando, e por interpretar o *JAVA*, *KOTLIN* ou *SWIFT*, ele identifica qual o tipo de aparelho e a partir de um único código desenvolvido, ele realiza desempenhos similares para essas diversas plataformas (FRANKLIN e SAMUEL FILHO 2020).

Conforme disse Fernandes (2020), está linguagem foi desenvolvida dando destaque a construção da interface de usuário, como o aperfeiçoamento para IU, uma melhor

produtividade no desenvolvimento e alta velocidade de processamento nas mais diversas plataformas.

Dart em si, é uma linguagem de *back-end*, e o *Framework Flutter* como *Front-end*, mas assim como *JavaScript* podem ter implementações com a linguagem *server side node* (FRANKLIN e SAMUEL FILHO 2020).

Segundo Silva (2019), o *Dart*, como todas as outras linguagens de programação, tem estruturas como *for* e *while* semelhantes a *Java*, *C++*, *C* e outras. Podemos visualizar um exemplo de uma dessas estruturas na Figura 5.

Figura 5 - Código básico de Dart rodando um for



```

1 void main() {
2   List integrantesEconomic = [
3     'Luan Avila',
4     'Luan Gonçalves',
5     'Thiago Teruel'
6   ];
7   for (int i = 0; i < integrantesEconomic.length;
8     i++) {
9     print('Integrante ${i}:' +
10    integrantesEconomic[i]);
  
```

Console

```

Integrante 0:Luan Avila
Integrante 1:Luan Gonçalves
Integrante 2:Thiago Teruel
  
```

Fonte: Autoria Própria, 2021.

2.4 FLUTTER

Flutter é um *framework* da Linguagem *Dart* de código aberto, desenvolvido em *C*, *C++* e *Dart* pela *Google*, com ele podemos desenvolver para *IOS*, *Android*, *Desktop* e *Web*. Esta ferramenta possui muitas bibliotecas e *casses on-line*, podendo estas serem utilizadas por todo e qualquer desenvolvedor (SILVA, 2019).

Marinho (2020) diz que para começarmos a desenvolver um aplicativo com *Flutter*, precisamos utilizar as bibliotecas padrões chamadas de *material.dart* e *cupertino.dart*, respectivamente para *Android* e *IOS*. Com elas importadas, podemos usar de *Widgets* já criados pelo *Flutter* e customizá-los. Também existem bibliotecas externas, produzidas por outros desenvolvedores, tendo elas suas funcionalidades específicas e sendo disponibilizadas para todos, como as do *firebase*. Utilizamos essas bibliotecas a partir de comandos *import* e seus caminhos de pacote, apresentado na figura 6.

Figura 6- Importações de biblioteca do Flutter e do Firebase

```
1 import 'package:flutter/material.dart';
2 import 'package:firebase_core/firebase_core.dart';
3 import 'package:cloud_firestore/cloud_firestore.dart';
```

Fonte: Autoria Própria, 2021.

Para Fernandes (2020) esta é uma *Software Development Kit* (SDK) para a construção de um projeto multiplataforma a partir de um único código, possibilitando o desenvolvedor a se dedicar em uma interface de alta performance para sua aplicação.

Silva (2019) citou algumas características principais do *Flutter*, como um rápido desenvolvimento, tendo uma ferramenta para carregar o código completo sem precisar reiniciá-lo com a utilização de recursos do teclado, “R” maiúscula para reinicialização e o “r” minúscula para recarregar só as modificações, uma ótima performance na atuação de multiplataformas e uma criação de interfaces de usuário, demonstrada na figura 7.

Figura 7 - Comandos para agilizar o desenvolvimento no Flutter

```
Running with sound null safety
To hot restart changes while running, press "r" or "R".
For a more detailed help message, press "h". To quit, press "q".
```

Fonte: Autoria Própria, 2021.

A compilação do código do *Flutter* ocorre a partir do ARM, um método onde se compila o projeto para cada linguagem, sendo elas no KOTLIN e JAVA para Android ou SWIFT para IOS, trazendo um melhor desempenho em cada dispositivo (FERNANDES, 2020).

As compilações de cada sistema operacional ficam em suas respectivas pastas, como mostrado na figura 8.

Figura 8 - Imagem das pastas relacionadas ao Android e ao IOS

```
> android
> build
> ios
```

Fonte: Autoria Própria, 2021.

No desenvolvimento de telas pelo *Flutter*, podemos optar por construí-la de forma mais estática ou dinâmica, através de classes chamadas de *StatefulWidget*, capaz de dinamizar a visualização da tela, e a *StatelessWidget*, que interpreta a tela de uma forma mais estática, com elementos que não mudarão seu estado (MARINHO, 2020), como na figura 9.

Figura 9 - Classes Statefull e Stateless do flutter

```

12 > class ReferenciaTCC extends StatelessWidget { ...
35
36 class ReferenciaTCC2 extends StatefulWidget {
37   const ReferenciaTCC2({ Key? key }) : super(key: key);
38
39   @override
40   _ReferenciaTCC2State createState() => _ReferenciaTCC2State();
41 }
42
43 > class _ReferenciaTCC2State extends State<ReferenciaTCC2> { ...

```

Fonte: Autoria Própria, 2021.

A programação *Flutter* é organizada em *Widget* diz Marinho (2020), este podendo ser um botão, um campo de texto ou até um simples quadrado colorido. Estes *Widgets* ao serem interpretados, geram componentes para as plataformas ou sistemas em que eles estão sendo compilados.

- Scaffold: O “esqueleto” do layout da aplicação, ele estrutura corretamente os *Widgets* inseridos.
- Container: Este é um espaço “livre”, onde podemos moldá-lo e editá-lo, como sua cor, seu tamanho entre outros.
- Text: Nos permite inserir qualquer tipo de *string*.
- Center: Centraliza os *Widgets*.
- child: É o elemento responsável por referenciar um *Widget* dentro de outro.
- ListView: Gera Listas, podendo ser automática ou “manual”.

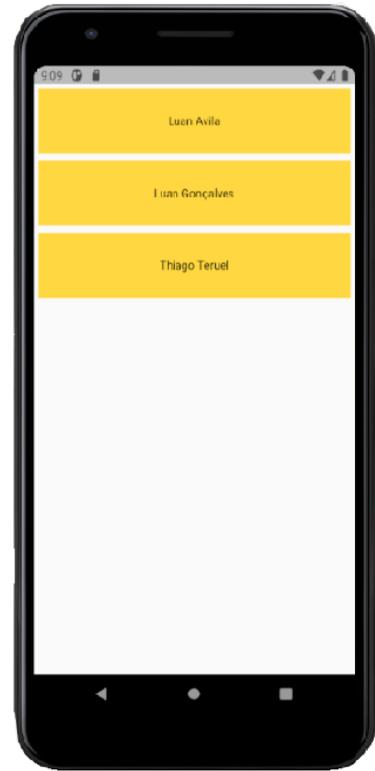
Conforme visto na figura 10.

Figura 10 - Programa em Dart demonstrando na prática os Widgets

```

1  import 'package:flutter/material.dart';
2
3  List integrantesEconomic = ['Luan Avila', 'Luan Gonçalves', 'Thiago Teruel'];
4
5  void main() => runApp(
6    MaterialApp(
7      debugShowCheckedModeBanner: false,
8      home: ReferenciaTCC(),
9    ), // MaterialApp
10 );
11
12 class ReferenciaTCC extends StatelessWidget {
13   const ReferenciaTCC({Key? key}) : super(key: key);
14
15   @override
16   Widget build(BuildContext context) {
17     return Scaffold(
18       body: ListView.builder(
19         itemCount: integrantesEconomic.length,
20         itemBuilder: (BuildContext context, int index) {
21           return Container(
22             margin: EdgeInsets.all(5),
23             width: double.infinity,
24             height: 80,
25             color: Colors.amberAccent,
26             child: Center(
27               child: Text(integrantesEconomic[index]),
28             ), // Center
29           ); // Container
30         },
31       ), // ListView.builder
32     ); // Scaffold
33   }
34 }

```



Fonte: Autoria Própria, 2021.

2.5 Banco de Dados

Um banco de um conjunto organizado de dados persistentes, que são na verdade dados que podem ser manipulados, e usados por uma aplicação de um determinado projeto, empresa ou sistema que seja capaz de consumi-la. *Structured Query Language* (SQL) é um banco de dados relacional, *relational database management system* (RDBMS), ele é separado por tabelas e colunas que podem interagir umas com as outras, através de “novas” tabelas temporárias geradas através de tabelas “antigas” (DATE, 2004).

Um Sistema gerenciador de Banco de Dados (SGBD), é um sistema capaz de gerenciar bancos de dados, como exemplo, o phpmyadmin para o SQL, também agregando características como segurança, recuperação de falhas, armazenamento de dados e propriedades ACID, Atomicidade, Consistência, Isolamento e Durabilidade (LÓSCIO, 2016).

2.5.1 MySQL

O banco de dados MySQL utiliza do SQL como linguagem e é baseado no armazenamento de dados a partir de tabelas e colunas que, em cada linha, guardam dados, semelhante a uma planilha (MILANI, 2007). Exemplo na figura 11.

Figura 11 - Tabela de um banco MySQL

CODPRODUTO	NOMEPRODUTO	TIPOPRODUTO	QTDPRODUTO	VALORPRODUTO
1	VIDEOGAME	ELETRÔNICO	100	400.00
2	IMPRESORA	INFORMÁTICA	150	367.00
3	TELEVISOR	ELETRÔNICO	400	600.00
4	CELULAR	TELEFONE	300	250.00
5	CÂMERA DIGITAL	DIGITAIS	200	400.00
6	PROCESSADOR	INFORMÁTICA	100	346.00
7	FILMADORA DIGITAL	DIGITAIS	200	700.00
8	MONITOR	INFORMÁTICA	100	231.00

Fonte: Autoria Própria, 2021.

Por ser um banco de dados relacional todas as colunas são predefinidas com os tipos de dados que serão inseridos nela, relações entre as tabelas do banco entre outras possibilidades, como o *int* para valores numéricos do tipo inteiro e o *varchar* para valores do tipo texto, a ferramenta *Primary Key* (PK) e a *Foreign Key* (FK) para colunas de identificação e colunas que herdaram informações de outras tabelas (OLIVEIRA e MADRUGA, 2019).

2.6 JAVA

Heinzelmann (2010) explica que a linguagem começou a surgir em 1991 na Sun Microsystems. Inicialmente estava fazendo parte de outro projeto, chamado Green Project, que tinha como objetivo possibilitar a convergência entre computador, equipamentos eletrônicos e eletrodomésticos. Inicialmente chamada de Oak, traduzido para o português ficava Carvalho, pelo simples fato de ser a árvore que James Gosling avistava na janela da sede do projeto. Um dos principais trabalhos de James era adaptar a linguagem Oak para a internet, surgindo assim, em 1995, a plataforma Java. Ele cita que uma das principais diferenças de Java e outras linguagens de programação existentes na época era que, o Java era executado sobre uma *Java Virtual Machine* (JVM), ou seja, qualquer plataforma de hardware ou equipamento eletrônico que possa executar uma máquina virtual conseguirá executar java. Fazendo assim, jus ao dilema de Java, “write once, run anywhere”, ou em português, “escreva uma vez, rode em qualquer lugar”.

Goodrich (2013) explica que, para incentivar o bom uso de comentários em bloco e a produção automática de documentação, o ambiente de programação Java vem com um programa para a geração de documentos chamado de javadoc. O programa examina uma coleção de arquivos fontes Java que tenham sido comentados usando certas palavras reservadas, chamadas tags, e produzem uma série de documentos HTML que descrevem as classes, métodos, variáveis e constantes contidas nesse arquivo. O autor menciona em seu texto o chamado pseudocódigo, que significa criar um código de uma maneira mais compreensível á olhos humano, em vez de escrever um código real. É uma mistura de língua natural com estruturas de programação de alto nível que descrevem as ideias principais que estão por tras da implementação de uma estrutura de dados ou algoritmos. Porém não existe uma definição precisa para a linguagem em pseudocódigo, em razão de sua dependência da língua natural.

Andl (2021) cita os construtores, os quais são métodos especiais destinados á inicialização e ao preparo de novos objetos durante sua criação, ou seja, durante sua instanciação, podendo igualmente aos métodos comuns, os construtores, podem também receber parâmetros, o que permite caracterizar um objeto durante sua criação, no entanto, só podem ser acionados por meio do operador new, responsável pela criação de novos objetos.

2.7 Kotlin

Kotlin desenvolvida pela *JetBrains* em 2011, é uma linguagem de programação multipataforma, orientada a objetos e funcional, concisa e estaticamente tipada. Esta linguagem pode ser compilada para a Máquina virtual Java e que também pode ser traduzida para código nativo. Foi anunciada em 2017 pela Google como linguagem oficial do sistema Android (LECHETA, 2018).

Kotlin tem a linguagem idêntica ao Java, porém Kotlin tem interoperabilidade, sintaxe, *null safety*, *coroutines*, e suporte no Android Jetpack e outras bibliotecas. Essas são algumas das diferenças que deixam o Kotlin a frente do Java. Temos dois tipos de variáveis em Kotlin val e var:

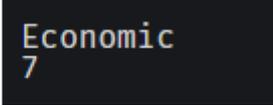
- val: para declarar constantes, e é a forma mais recomendada para declaração de variáveis.
- var: declara qualquer variável cujo valor pode ser alterado.

Assim como no Java o Kotlin possui tipos como *long*, *int*, *short*, *boolean*, classes, funções, expressões, *arrays*, *loops*, *while*, *for* (GLAUBER, 2019).

Um exemplo básico de Kotlin encontra-se na figura 12.

Figura 12 - Código em Kotlin

```
1 fun main(args : Array<String>) {
2     val nome = "Economic"
3     var numero = 7
4     println(nome)
5     println(numero)
6 }
```

The image shows a dark rectangular box containing the output of the Kotlin code. The text "Economic" is on the top line and "7" is on the bottom line, both in a light-colored monospace font.

Fonte: Autoria Própria, 2021.

2.8 Diagramas

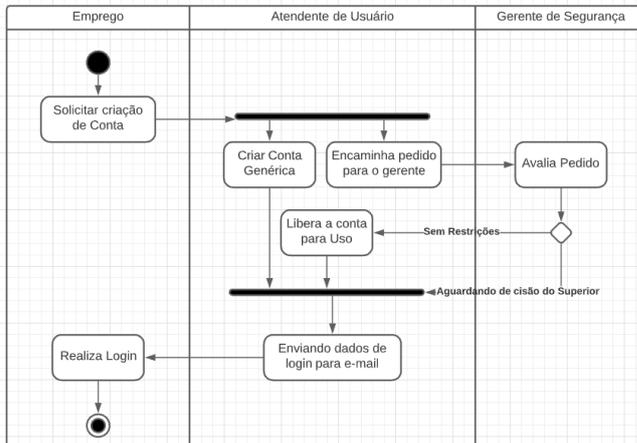
Booch (2006) diz que um diagrama é a apresentação gráfica de um conjunto de elementos, geralmente representadas como gráficos de vértices, itens, e arcos, relacionamentos. São desenhados para permitir a visualização de um sistema sobre diferentes perspectivas, sendo assim, um diagrama constitui uma projeção de um determinado sistema. É listado 13 tipos de diagramas, sendo eles: diagrama de classes, diagrama de objetos, diagrama de componentes, diagrama de estruturas compostas, diagrama de caso de uso, diagrama de sequências, diagrama de comunicações, diagrama de gráficos de estados, diagrama de atividades, diagrama de implantação, diagrama de pacote, diagrama de temporização, diagrama de visão geral da interação.

Manara (2014) afirma que quando iniciamos o desenvolvimento de um software, devemos passar por uma lista de execuções, no caso, tópicos, para que aquilo que desejemos fazer seja feito com excelência. Cita que dentre eles estão, definir as atividades que serão executadas ao longo do projeto, determinar quando, como e por quem serão executadas as atividades, delimitar os artefatos que serão utilizados e qual é o melhor momento para aplicá-las, determinar as ferramentas que auxiliarão na construção do produto de software e fornecer um guia para as atividades realizadas a todos os envolvidos com o desenvolvimento do produto. Sem um desses tópicos a serem executados ou a falta de um deles, pode ser crucial para o sucesso do software ou aquilo que está sendo desenvolvido.

Booch (2006) explica os 3 diagramas mais importantes, sendo eles diagrama de atividade, sequência e caso de uso. Diagrama de atividade exhibe a estrutura de um

processo, como o fluxo de controle e os dados de cada etapa de uma computação. Abrange a visão dinâmica do sistema e é importante principalmente para a modelagem da função de um sistema, como mostra na imagem 13.

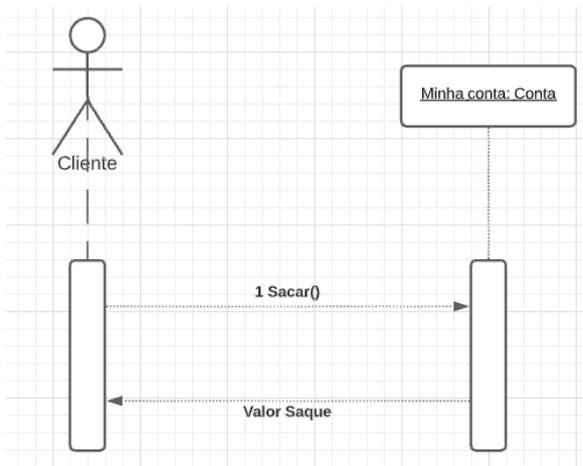
Figura 13 - Diagrama de Atividade



Fonte: Autoria Própria, 2021.

O diagrama de sequência, é um diagrama de interação cuja ênfase está na ordenação temporal das mensagens, ele representa conceitos básicos, enfatizando a ordem temporal, exemplo de diagrama de sequência na imagem 14 (BOOCH, 2006).

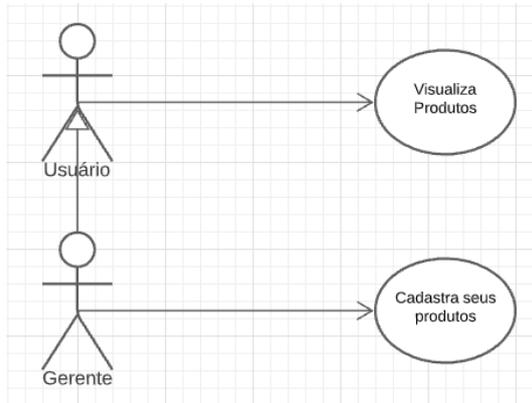
Figura 14 - Diagrama de Sequência



Fonte: Autoria Própria, 2021.

Booch (2006) por final explica o diagrama de caso de uso, o diagrama exibe um conjunto de uso e atores, um tipo especial de classe, e seus relacionamentos. Diagrama de caso de uso abrange a visão estática de casos de uso de sistema, como vemos na imagem 15.

Figura 15 - Diagrama de Caso de Uso



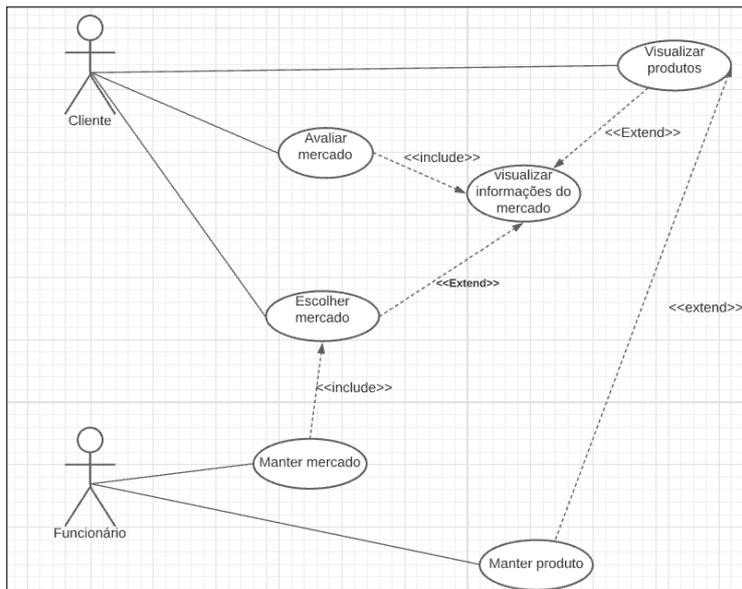
Fonte: Autoria Própria, 2021.

3 DESENVOLVIMENTO

3.1 Diagrama de caso de uso

O diagrama de caso de uso, é um diagrama usado para descrever graficamente um subconjunto do modelo para simplificar a comunicação. Ele serve como um unificador em todo o desenvolvimento do sistema. Visto na imagem 16, o diagrama de caso de uso usado no trabalho.

Figura 16 - Diagrama de caso de uso

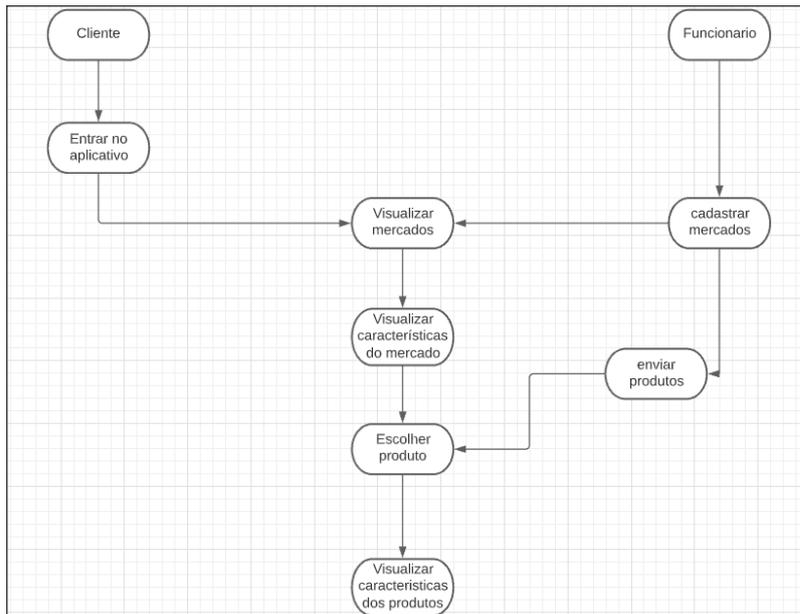


Fonte: Autoria Própria, 2021.

3.2 Diagrama de atividade

O diagrama de atividade é essencialmente um gráfico de fluxo, mostrando o fluxo de controle de uma atividade para outra e como serão empregados para fazer a modelagem de aspectos dinâmicos do sistema. O objetivo do diagrama de atividades é mostrar o fluxo de atividades em um único processo. O diagrama mostra como cada atividade depende uma da outra. Será visto na imagem 17, o diagrama de atividade usado no referenciado trabalho.

Figura 17 - Diagrama de atividade



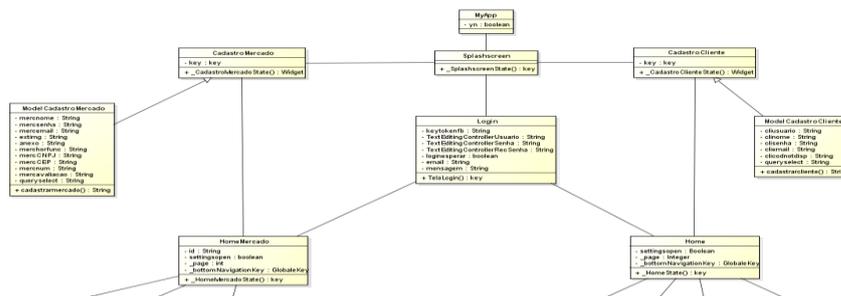
Fonte: Autoria Própria, 2021.

3.3 Diagrama de Classes

Em programação, um diagrama de classes é uma representação da estrutura e relações das classes que servem de modelo para objetos.

O mesmo é usado na representação estática na área da programação para descrever a estrutura de um sistema, apresentando suas classes, atributos, operações e as relações entre os objetos, como vemos nos exemplos da imagem 18 e 19.

Figura 18 - Diagrama de Classes



Fonte: Autoria Própria, 2021.

Figura 21 - Tela Perfil do Mercado



Fonte: Autoria Própria, 2021.

A tela de perfil do mercado fornece as informações do próprio mercado, como seu nome, sua logo, horários de funcionamento, localização e configurações avançadas, contendo a senha e e-mail da conta do mercado. As informações são editáveis, exceto o CNPJ. Tela de perfil do mercado ilustrada na figura 21.

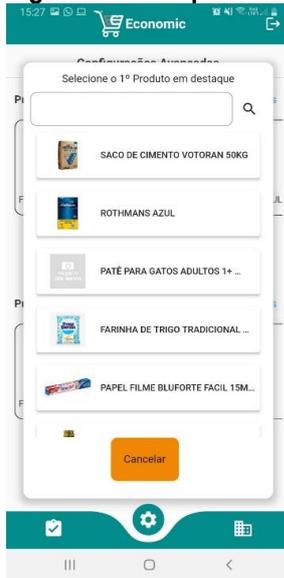
Figura 22 - Tela de configurações avançadas do mercado



Fonte: Autoria Própria, 2021.

A configuração avançada contém opções como adicionar em destaque e em oferta os produtos cadastrados no mercado, informar aos usuários se os produtos estão próximos a validade ou com redução de preços. Tela de configurações avançadas mostrada na figura 22.

Figura 23 - Tela para adicionar produtos ao destaque e a oferta



Fonte: Autoria Própria, 2021.

Esta tela é usada para pesquisar os produtos que o mercado queira adicionar ao destaque, contendo um campo de busca para facilitar o encontro dos produtos sem a necessidade de rolar a lista de produtos. Tela ilustrada na figura 23.

Figura 24 - Tela de visualização dos produtos do mercado



Fonte: Autoria Própria, 2021.

A tela de visualização de produtos, mostrada na figura 24, serve para encontrar produtos cadastrados pelo mercado, tendo os campos de pesquisa para encontrar os produtos específicos, lista de tipos de produtos para listá-los de acordo com seus tipos e um botão para cadastrar novos produtos.

Figura 25 - Tela de cadastro de produtos

15:28 Novo Produto

EAN: 7898902332236

Nome: PAPEL HIG LITOPEL SUL C/16 NEUTRO

Upload da imagem do produto, não obrigatório

Descrição

PAPEL HIG LITOPEL SUL C/16 NEUTRO

Marca: LITOPEL SUL

Tipo: PERF/HIGIENE

Preço

Varejo: Atacado: 1x

Cancelar Criar

Fonte: Autoria Própria, 2021.

O cadastro de produtos contém vários campos para inserir as informações, entre eles, o EAN do produto, nome do produto, imagem do produto, descrição, tipo do produto, preço de varejo e atacado, e a quantidade do atacado.

Após inserir um EAN já cadastrado, todos os outros campos serão completados automaticamente, levando praticidade na hora do cadastro. Tela mostrada na figura 25.

Figura 26 - Tela de Edição de produtos

15:27 Editar Produto

EAN: 4534530100016

Nome: SACO DE CIMENTO VOTORAN 50KG

Descrição

CIMENTO DE ALTA QUALIDADE, QUE ATENDE AOS REQUISITOS TÉCNICOS DAS NORMAS ABNT, SENDO INDICADA PARA A PREPARAÇÃO DE DIVERSOS TIPOS DE OBRAS, REBOCO,

Marca: VOTORAN

Tipo: MAT CONSTRUCAO

Preço

Varejo: R\$ 33,40 Atacado: R\$ 0,00 1x

Voltar Editar

Fonte: Autoria Própria, 2021.

A tela de edição do produto, ilustrada na figura 26, informa os dados atuais cadastrados e a possibilidade de editar os preços e quantidade de atacado.

Figura 27 - Tela de exclusão do produto



Fonte: Autoria Própria, 2021.

A tela de exclusão de produtos contém informações como o nome deste, seus valores em atacado e varejo, sua descrição e outras informações. Também há um botão para excluir este do cadastro de produtos do próprio mercado. Tela mostrada na figura 27.

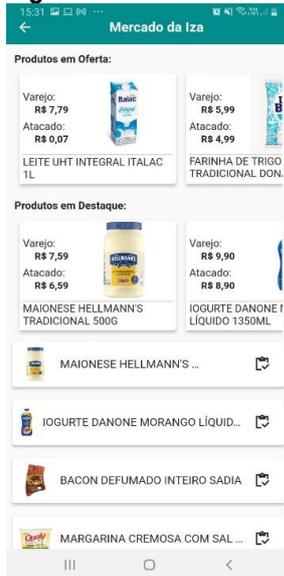
Figura 28 - Tela Mercado



Fonte: Autoria Própria, 2021.

A tela Mercado, ilustrada na figura 28, é utilizada para os usuários visualizarem todas as informações disponíveis do mercado tais como, localização, horário e dia de funcionamento. Ela possui um botão para ir para tela de Produtos do Mercado e um campo para os usuários avaliarem o mercado.

Figura 29 - Tela Produtos do Mercado



Fonte: Autoria Própria, 2021.

A tela Produtos do Mercado é utilizada para os usuários visualizarem todos os produtos de um mercado específico, visualizar os produtos em oferta, e em destaque.

A tela Produtos do Mercado é ilustrada na figura 29.

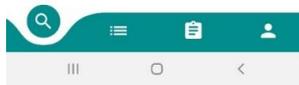
Figura 30 - Tela Produtos



Fonte: Autoria Própria, 2021.

Na tela Produtos o usuário, ilustrada na figura acima, consegue visualizar todas as informações disponíveis do produto tais como, nome, imagem, descrição, ean, preços de atacado e de varejo, o mercado fornecedor e a possibilidade de adicionar este produto a lista de compras através de um botão no final da tela.

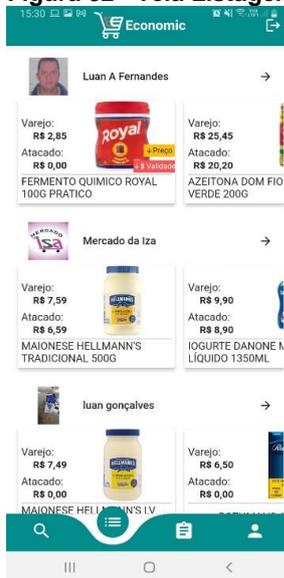
Figura 31 - Tela Pesquisa



Fonte: Autoria Própria, 2021.

Na tela Pesquisa, mostrada na figura 31, o usuário consegue pesquisar os produtos conforme o nome do produto desejado, basta digitar na barra de pesquisa, e clicar na lupa ao lado para executá-la, nesta tela o usuário poderá adicionar o produto a lista de compras.

Figura 32 - Tela Listagem Principal



Fonte: Autoria Própria, 2021.

Na tela Listagem Principal, ilustrada na figura 32, o usuário consegue visualizar os mercados mais próximos a ele que contém produtos em destaque, e visualizar um breve catálogo do mercado.

Figura 33 - Tela Lista de Compras



Fonte: Autoria Própria, 2021.

Na tela Lista de Compras podemos visualizar os produtos adicionados pelo usuário, e com eles, comparar seus preços em diferentes mercados através de um botão localizado ao final da tela, o usuário pode retirar o produto da lista de desejo.

A tela Lista de Compras ilustrada na figura 33.

Figura 34 - Tela Usuário



Fonte: Autoria Própria, 2021.

Na tela Usuário, mostrada na figura 34, o usuário pode visualizar suas credenciais, e alterá-las se achar necessário.

Figura 35 - Tela de Comparação

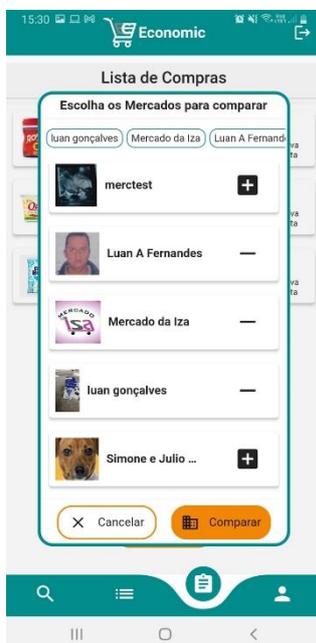


Produtos	Mercado da Iza	Luan A Fernandes
FARINHA DE TRIGO	R\$ 5,99	R\$ 10,00
FERMENTO QUÍMICO	-	R\$ 2,85
MARGARINA CREMOSA COM	R\$ 7,99	-

Fonte: Autoria Própria, 2021.

Na tela Comparação, mostrada na figura 35, o usuário poderá visualizar os produtos em sua lista e os preços deles nos mercados de sua escolha.

Figura 36 - Tela de escolha de Mercados



Fonte: Autoria Própria, 2021.

A tela de escolha de Mercados, ilustrada na figura 36, é usada para escolher os mercados que o usuário usará para comparar os preços dos produtos que estão em sua lista.

Figura 37 - Tela de cadastro de mercado

The image displays two sequential screenshots of a mobile application interface for market registration. The first screenshot, titled 'Mercado', shows a form with the following fields: 'Nome do Mercado:' (text input), 'Upload da logo do mercado' (button with an upload icon), 'E-mail vinculado ao Mercado:' (text input), and 'Senha:' (password input with a maximum of 15 characters and a visibility toggle). An orange 'Avançar' button is at the bottom. The second screenshot, also titled 'Mercado', shows the 'CNPJ:' field, a 'Localização do Mercado' button with a location pin icon, and a 'Horário de funcionamento' section. This section includes checkboxes for 'Segunda à Sexta', 'Sábado', 'Domingo', and 'Feriado', each followed by time selection fields and an edit icon. Orange 'Voltar' and 'Cadastrar' buttons are at the bottom.

Fonte: Autoria Própria, 2021.

A tela de cadastro de mercado, na figura 37, contém campos para realizar o cadastro de mercado.

Figura 38 - Tela de cadastro de usuário

The image shows a single screenshot of a mobile application interface for user registration, titled 'Cliente'. It features a vertical list of input fields: 'Nome:', 'Email:', 'Usuário:', 'Senha:', and 'Confirmar Senha:'. Each field has a corresponding eye icon to toggle password visibility. An orange 'Cadastrar' button is positioned at the bottom of the form.

Fonte: Autoria Própria, 2021.

A tela de cadastro de usuário, ilustrada na figura 38, contém campos para a realização do cadastro de usuários.

4 CONCLUSÃO

O presente trabalho teve como ideia central, auxiliar clientes de mercados a encontrar os seus produtos desejáveis nos mercados selecionados, comparando-os para assim, escolher o que mais lhe agrada.

Este aplicativo consegue realizar informações em duas ramificações, o lado cliente e o lado mercado. O lado mercado consegue realizar tarefas como cadastrar produtos, classificá-los em oferta ou em destaque e informar aos clientes seu local e horário de funcionamento. O lado cliente oferece informações como os mercados mais próximos a ele, possibilidade de pesquisa de produtos para adicionar em sua lista, visualização de produtos de um único mercado e a construção de uma lista de desejos com a opção de comparação entre diferentes mercados.

REFERÊNCIAS

AONDECONVEM (ITÁLIA) (org.). Aplicativos economizam tempo e dinheiro na hora das compras. 2020. Disponível em: <http://grupocasa.com.br/page/11/?author=0> Acesso em: 2021.

ANDL, Peter. Java Guia do Programador - 4a Edição: atualizado para java 16. São Paulo: Novatec, 2021. 552 p.

BEGGIORA, Helito. Como usar o app Facily para obter descontos em compras compartilhadas: aplicativo permite comprar produtos com descontos de até 70%. Aplicativo permite comprar produtos com descontos de até 70%. 2020. Disponível em: <https://www.techtudo.com.br/dicas-e-tutoriais/2019/08/como-usar-o-app-facily-para-obter-descontos-em-compras-compartilhadas.ghtml>. Acesso em: 04 jun. 2021.

BEGGIORA, Helito. Economize! Cinco aplicativos de celular que buscam produtos com desconto. 2021. Disponível em: <https://www.techtudo.com.br/listas/noticia/2016/12/economize-cinco-aplicativos-de-celular-que-buscam-produtos-com-desconto.html>. Acesso em: 04 jun. 2021.

BOOCH, Grady. UML: guia do usuário. Rio de Janeiro: Elsevier Brasil, 2006. 474 p.

DATE, C. J. Introdução a sistemas de bancos de dados. 8. ed. Brasil: Elsevier, 2004. 865 p.

EXTRA (Brasil) (org.). EXTRA INOVA CONCEITO DE HIPERMERCADO NO PAÍS E REINAUGURA TRÊS LOJAS EM FORTALEZA. 2015. Disponível em: <https://www.divulgaacao.com.br/extra-inova-conceito-de-hipermercado-no-pais-e-reinaugura-tres-lojas-em-fortaleza/> Acesso em: 4 jun. 2021.

FACILY (Brasil) (org.). COMPRE JUNTO. 2021. Disponível em: <https://faci.ly/>. Acesso em: 04 jun. 2021.

FERNANDES, Warley Alves. DESENVOLVIMENTO DE UM APLICATIVO PARA AUXILIAR NA PREPARAÇÃO DE ALUNOS DE ENGENHARIA DE COMPUTAÇÃO PARA O ENADE. 2020. 61 f. TCC (Graduação) - Curso de Engenharia de Computação, Universidade, Pontifícia Universidade Católica de Goiás, Goiânia, 2020.

FRANKLIN, Matheus Maião; SAMUEL FILHO, Ronaldo Aparecido. Desenvolvimento de um Sistema de Gestão Escolar com o uso da Linguagem Dart com Framework Flutter. Revista Eletrônica E-F@Tec, Garça, São Paulo, v. 10, n. 1, p. 17-27, out. 2020. Anual. Disponível em: <https://fatecgarca.edu.br/ojs/index.php/efatec/issue/view/11>. Acesso em: 30 maio 2021.

GLAUBER, Nelson. Dominando o Android com Kotlin. São Paulo: Novatec Editora Ltda, 2019.

GOODRICH, Michael. Estruturas de Dados & Algoritmos em Java - 5ed. Porto Alegre: Bookman, 2013.

HEINZELMANN, Décio. Programação Java para a Web. São Paulo: Novatec, 2010. 640 p.

LECHETA, Ricardo R. Android Essencial com Kotlin. São Paulo: Novatec Editora Ltda, 2018.

MARINHO, Leonardo H. Iniciando com Flutter Framework: desenvolva aplicações móveis no dart side!. Brasil: Casa do Código, 2020. 187 p.

MILANI, André. MySQL - Guia do Programador. São Paulo, Sp: Novatec Editora, 2007. 400 p.

OLIVEIRA, Jefferson Dantas de; MADRUGA, Zulma Elizabete de Freitas. DESENVOLVIMENTO DE UM BANCO DE DADOS PARA ARMAZENAMENTO DE PRODUÇÕES SOBRE MODELAGEM MATEMÁTICA E CÁLCULO DIFERENCIAL E INTEGRAL. Revista de Educação Matemática e Tecnológica Iberoamericana, Garça, São Paulo, v. 10, n. 2, p. 1-17, 2019. Quadrimestral. Disponível em: <https://periodicos.ufpe.br/revistas/emteia/article/view/236193>. Acesso em: 23 setembro 2021. SALLES, Elis. Tudo sobre o site Extra. Disponível em: <https://apps.apple.com/br/app/clube-extra/id1186355193>. Acesso em: 04 jun. 2021.

SALLES, Elis. Extra: História de Sucesso. 2020. Disponível em: https://www.3sjundiai.com.br/pt/3522/extra:-historia-de-sucesso?mtid=3522&mturl_amigavel=extra:-historia-de-sucesso. Acesso em: 04 jun. 2021.

SILVA, João Pedro Paccola da. APLICATIVO ANDROID PARA COMPARTILHAMENTO DE TRANSPARÊNCIAS EM TEMPO REAL. 2019. 52 f. TCC (Graduação) - Curso de Ciência da Computação, Faculdade, Universidade Estadual Paulista, Bauru, 2019.