

**CENTRO DE EDUCAÇÃO TECNOLOGICA PAULA SOUZA  
ETEC DE TABOÃO DA SERRA**

**Curso Técnico de Desenvolvimento de Sistemas**

**Bruno Marques Ferreira**

**Caio César dos Santos Albino**

**Leonardo Pucci**

**Matheus de La Torre Farias**

**Matheus Ernesto dos Santos**

**Ricardo Gomes dos Santos**

**Tiago Alves Lessa Muniz**

**PEGA INFO**

**Taboão da Serra**

**2021**

**Bruno Marques Ferreira**

**Caio César dos Santos Albino**

**Leonardo Pucci**

**Matheus de La Torre Farias**

**Matheus Ernesto dos Santos**

**Ricardo Gomes dos Santos**

**Tiago Alves Lessa Muniz**

## **PEGA INFO**

Trabalho de Conclusão de Curso apresentado à Etec de Taboão da Serra, do Centro Estadual de Educação Tecnológica Paula Souza, como requisito parcial para a obtenção da habilitação profissional de Técnico de Nível Médio em Desenvolvimento de Sistemas sob a orientação do(s) Professor(es) Fabiano Jorge Oliveira Lopes e Rodolfo Votto Filho.

**Taboão da Serra**

**2021**

Aprovada em: \_\_\_\_\_ / \_\_\_\_\_ / \_\_\_\_\_

Conceito: \_\_\_\_\_

Banca de Validação:

---

Professor.....

Etec de Taboão da Serra

Orientador

---

Professor .....

Etec de Taboão da Serra

---

Professor .....

Etec de Taboão da Serra

**TABOÃO DA SERRA – SP**

**2021**

## **RESUMO**

Esta dissertação descreve a produção do aplicativo “Pega Info”, desenvolvido por alunos do curso de desenvolvimento de sistemas da Etec de Taboão da Serra. A ideia do trabalho surgiu após a observação de dificuldades enfrentadas pelos parentes dos membros do grupo na hora de adquirir produtos eletrônicos. Estas dificuldades variavam entre falta de conhecimento sobre peças de computador, dúvidas sobre quais celulares mais se encaixavam no que estavam procurando e necessidade de saber as lojas que ofereciam estes produtos pelos melhores preços. Estes obstáculos constatados serviram como impulso e motivação para o desenvolver de uma ideia sobre como sanar este problema, e posteriormente o desenvolver um software que viesse de fato a auxiliar o usuário em suas compras. O aplicativo tem como objetivo auxiliar os mais leigos no ramo de informática a adquirir seus eletrônicos, como computadores de mesa, notebooks, tablets e celulares. O software ajuda o usuário por meio de explicações sobre componentes vitais para o funcionamento dos dispositivos, além de mostrar as lojas em que o produto está disponível no momento e o preço atual dele. O Pega Info pode ser usado por pessoas de qualquer idade que desejam ter uma ajuda ou recomendação na hora de comprar seus dispositivos, com tudo, a interface do aplicativo foi desenvolvida para ser a mais intuitiva possível, para que os mais idosos possam utilizá-lo sem dificuldades.

Palavras-chave: Auxiliar, Eletrônicos, Intuitivo.

## **ABSTRACT**

This dissertation describes the production of the “Pega Info” application, developed by students from the systems development course at Etec in Taboão da Serra. The idea for the work came after observing the difficulties faced by the group members' relatives when purchasing electronic products. These difficulties ranged from lack of knowledge about computer parts, doubts about which cell phones best fit what they were looking for, and need to know the stores that offered these products at the best prices. These identified obstacles served as impetus and motivation to develop an idea on how to solve this problem, and later to develop software that would actually help the user in their purchases. The application aims to help lay people in the computer industry to acquire their electronics, such as desktop computers, notebooks, tablets and cell phones. The software helps the user by explaining vital components for the devices to work, in addition to showing the stores where the product is currently available and its current price. Pega Info can be used by people of any age who want to have a help or recommendation when buying their devices, however, the application's interface was developed to be as intuitive as possible, so that the elderly can use it without difficulties.

**Keywords:** Help, Electronic Components, Intuitive.

## **RESUMEN**

Esta disertación describe la producción de la aplicación "Pega Info", desarrollada por estudiantes del curso de desarrollo de sistemas de Etec de Taboão da Serra. La idea del trabajo surgió tras la observación de las dificultades a las que se enfrentan los familiares de los miembros del grupo a la hora de comprar productos electrónicos. Estas dificultades iban desde el desconocimiento de las piezas informáticas, las dudas sobre qué móviles se ajustaban mejor a lo que buscaban y la necesidad de saber qué tiendas ofrecían estos productos a los mejores precios. Estos obstáculos sirvieron de impulso y motivación para desarrollar una idea sobre cómo resolver este problema, y luego desarrollar un software que realmente ayudara al usuario en sus compras. La aplicación pretende ayudar a los profanos en informática a comprar sus dispositivos electrónicos, como computadores de escritorio, portátiles, tabletas y teléfonos móviles. El software ayuda al usuario mediante explicaciones sobre los componentes vitales para el funcionamiento de los dispositivos, además de mostrar las tiendas en las que está disponible el producto en ese momento y su precio actual. Pega Info puede ser utilizado por personas de cualquier edad que deseen tener una ayuda o recomendación a la hora de comprar sus dispositivos, con todo, la interfaz de la aplicación ha sido desarrollada para ser lo más intuitiva posible, de manera que las personas mayores puedan utilizarla sin dificultades.

Palabras clave: Ayuda, Electrónica, Intuitiva.

## **Lista de Imagens**

Imagen 1.....	13
Imagen 2.....	14
Imagen 3.....	15
Imagen 4.....	16
Imagen 5.....	18
Imagen 6.....	22
Imagen 7.....	24
Imagen 8.....	25
Imagen 9.....	26
Imagen 10.....	27
Imagen 11.....	28
Imagen 12.....	29
Imagen 13.....	30
Imagen 14.....	31
Imagen 15.....	32
Imagen 16.....	33
Imagen 17.....	34
Imagen 18.....	35
Imagen 19.....	36
Imagen 20.....	37
Imagen 21.....	38
Imagen 22.....	39
Imagen 23.....	40
Imagen 24.....	41
Imagen 25.....	42

Imagen 26.....	43
Imagen 27.....	44
Imagen 28.....	45
Imagen 29.....	45
Imagen 30.....	46
Imagen 31.....	47
Imagen 32.....	47
Imagen 33.....	48
Imagen 34.....	48
Imagen 35.....	49
Imagen 36.....	50
Imagen 37.....	50
Imagen 38.....	51
Imagen 39.....	51
Imagen 40.....	52
Imagen 41.....	52
Imagen 42.....	53
Imagen 43.....	54
Imagen 44.....	54
Imagen 45.....	55
Imagen 46.....	56
Imagen 47.....	56

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>9</b>
1.1 Problemática .....	9
1.2 Justificativa .....	10
1.1 Objetivo geral e objetivos específicos. ....	10
1.2 Metodologia. ....	11
1.3 Resultado dos formulários.....	11
<b>2 DESENVOLVIMENTO .....</b>	<b>16</b>
2.1 Requisitos funcionais e não funcionais do projeto.....	16
2.2 Diagramas de caso de uso do projeto.....	17
2.3 Descrição de todos os softwares que estão sendo utilizados na produção/registro da monografia .....	19
2.4 Descrição de todos os softwares que estão sendo utilizados na produção do programa/aplicativo do projeto .....	20
2.5 Canvas e modelo de negócios.....	21
2.5 Telas do projeto .....	23
2.6 Códigos .....	37
<b>3 CONCLUSÃO.....</b>	<b>54</b>
<b>REFERÊNCIAS.....</b>	<b>55</b>

## 1 INTRODUÇÃO

Com o avanço cada vez mais rápido da tecnologia, as pessoas estão tendo que se modernizar para se integrar as mudanças da sociedade, com tudo, é grande o número de pessoas que não possuem conhecimento suficiente nas áreas tecnológicas para adquirir o eletrônico que mais se adeque a sua necessidade e, em muitos casos, acabam pagando por algo que não as atende. O **Pega Info** foi pensado para resolver esse problema, auxiliando os mais leigos no assunto a comprar um dispositivo que venha a atender suas necessidades, mantendo uma relação de custo/benefício que o ajude a fazer parte do mundo digital.

### 1.1 Problemática

É notável a crescente demanda por peças de computador e eletrônicos em geral, demanda essa que aumentou 20% na pandemia no ano de 2020 (Fonte: Gazeta Digital), o que indica que as pessoas, embora algumas ainda tenham um pouco de resistência, estão tentando cada vez mais tentando fazer parte do mundo digital e das facilidades que ele nos proporciona.

Com tudo, que obstáculo é esse que muitas pessoas ainda enfrentam na hora de integrar-se ao mundo digital? A resposta é: a falta de conhecimento. Ainda é grande o número de pessoas que não sabem como comprar um computador ou um celular, e muitas dessas nem sempre tem alguém na família disponível a todo momento para ajudá-las nessa hora.

Por isso, uma ajuda digital que seja fácil e intuitiva é fundamental para auxiliar aqueles mais leigos nos vários termos tecnológicos a se nortear no mundo digital, e essa é a base do Pega Info, um programa para ajudar as pessoas a fazer a escolha certa na hora de adquirir um dispositivo que venha a suprir a sua necessidade.

## **1.2 Justificativa**

De acordo com pesquisas realizadas em nossa microrregião, é grande a necessidade de um auxílio aos mais leigos na hora comprar um eletrônico. De 30 pessoas, 63% disseram que sentem dificuldade em encontrar algo que atenda a sua demanda. 90% responderam que os sites de venda em geral poderiam ser mais explicativos, para ajudar o cliente a fazer a melhor compra, 90,3% das pessoas que respondeu que gostaria que os sites recomendassem uma opção de compra baseada na procura do usuário. Cerca de 22% das respostas eram de pessoas com mais de 50 anos, o que evidencia uma dificuldade por parte dos mais idosos. Essas pesquisas foram feitas utilizando a plataforma Google Forms, sendo compartilhados links de acesso para as pessoas responderem as perguntas.

Matérias publicadas recentemente comprovam o aumento da busca por eletrônicos em todo o país, só a busca por computadores aumentou em 26.1% do quarto bimestre de 2019 para 2020 (Fonte: IDC,2021), algo que alavancou o preço de vários produtos desta área, criando uma escassez de componentes para as fornecedoras. A grande quantidade e variedade de produtos disponíveis no mercado, a falta de especificações desses produtos em alguns sites, o crescimento de procura, o aumento dos preços, juntamente com o aumento da exigência dos compradores, servem para evidenciar a necessidade da criação de uma plataforma como Pega Info, que mostre ao usuário as opções de compras na internet, analisando a melhor relação custo/benefício.

### **1.1 Objetivo geral e objetivos específicos.**

**Geral** – criar um programa que ajude o usuário a adquirir o dispositivo eletrônico que melhor atenda a suas necessidades.

## **Específicos**

Etapa 1 - Certificar a dificuldade das pessoas em adquirir seu eletrônico.

Etapa 2 - Levantar a idade média das pessoas que responderam.

Etapa 3 - Tabelar os dados das pesquisas.

Etapa 4 - Organizar informações e explicações referentes aos eletrônicos.

Etapa 5 - Desenvolver o software.

Etapa 6 - Testar o software.

Etapa 7 - Divulgar o software.

Etapa 8 - Acompanhar o nível de eficácia e satisfação com o aplicativo.

## **1.2 – Metodologia.**

Para levantar as informações necessárias para o prosseguimento da criação do aplicativo, fizemos algumas pesquisas, sendo elas: pesquisas “boca a boca” com funcionários da Etec Taboão da Serra, e formulários do Google Forms compartilhados em redes sociais e aplicativos de mensagem, e enviados a familiares e conhecidos dos membros do grupo. Com uma média da faixa etária das respostas de cerca de 27 anos. Para obter os dados referentes ao mercado eletrônico no Brasil, foi utilizado a internet, consultando sites especializados em notícias sobre tecnologia e matérias sobre demanda de eletrônicos em jornais e outros sites.

## **1.3 Resultado dos formulários**

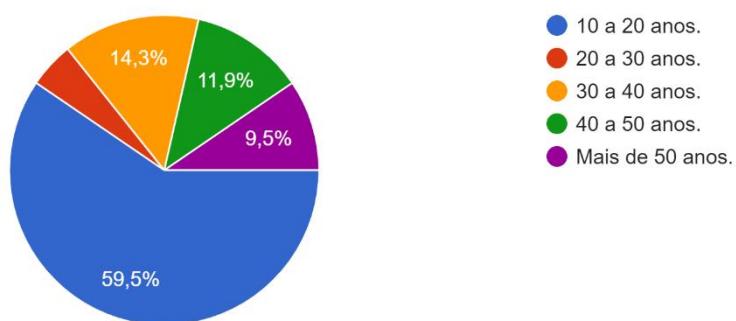
Em nosso formulário fizemos algumas perguntas, entre elas: a faixa etária daqueles entrevistados, se os mesmos possuem alguma dúvida ao comprar algum dispositivo eletrônico, se os sites de vendas poderiam ser mais explicativos e se gostariam de algum apoio para cessar suas dúvidas durante a compra de um novo produto envolvendo eletrônicos.

Conseguimos em torno de 91 respostas em nossa pesquisa online, além de outras que realizamos durante o tempo em que estivemos realizando o levantamento de dados. Os resultados dos formulários nos ajudaram a encontrar nosso público-alvo, e enfim descobrir como iríamos montar nosso aplicativo, qual interface seria mais atrativo para o público e etc.

### Imagen 1 – Primeira pergunta do formulário

Qual é a sua faixa etária?

84 respostas

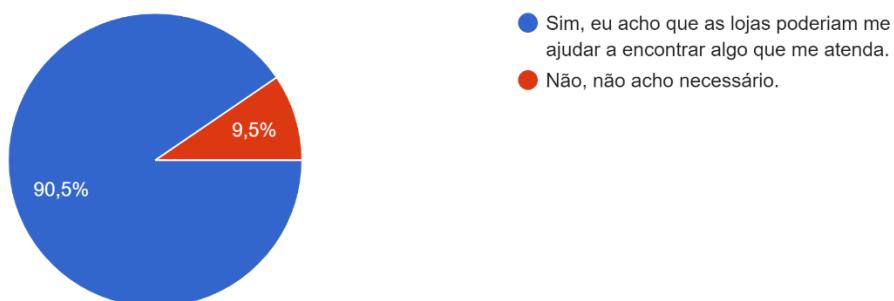


Dentro da nossa primeira pergunta, descobrimos a faixa etária do público que respondeu nosso questionário, isso nos ajudou a ter noção de quem estava respondendo. Com 59,5% das pessoas entre 10 a 20 anos de idade e 9,5% com mais de 50 anos de idade, dando um sentido maior para nossas próximas perguntas, já que sabemos que estamos lidando com esse público como maioria.

## Imagen 2 – Segunda pergunta do formulário

Você acha que os sites de venda em geral poderiam ser mais explicativos e te ajudar a entender o que você está comprando?

84 respostas



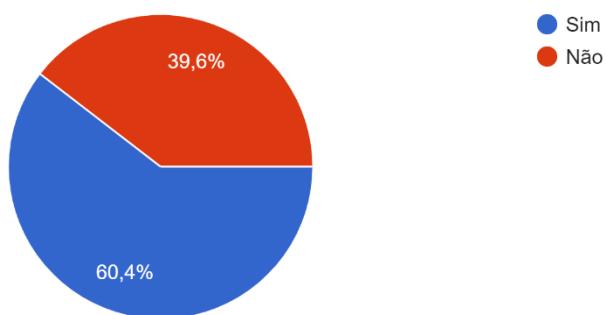
Já na segunda pergunta presente no nosso formulário, descobrimos o que as pessoas pensam sobre os sites de vendas, entendendo que nossos possíveis usuários iriam se interessar por sites mais explicativos e que buscassem dar aos nossos usuários mais auxílio durante sua busca ou compra.

Tal dado nos deu a ideia de que mais de 90% das pessoas entrevistadas, e nisso inclui nossos colegas de classe, que estão ligados a essa área, gostariam de um auxílio para encontrar o produto que mais se encaixaria na sua necessidade e no seu bolso, nos dando motivação pra seguir com esse projeto em específico, já que sabíamos que teríamos um público existente.

### Imagen 3 – Terceira pergunta do formulário

Na hora de comprar um novo eletrônico (como um celular ou computador/notebook), você sente dificuldade de pesquisar e encontrar um que atenda as suas necessidades ?

91 respostas



Na terceira pergunta do formulário, buscamos entender mais sobre o que as pessoas sentem quando estão procurando por um novo produto. Se elas sentem dificuldade ou não em encontrar aquilo que procura.

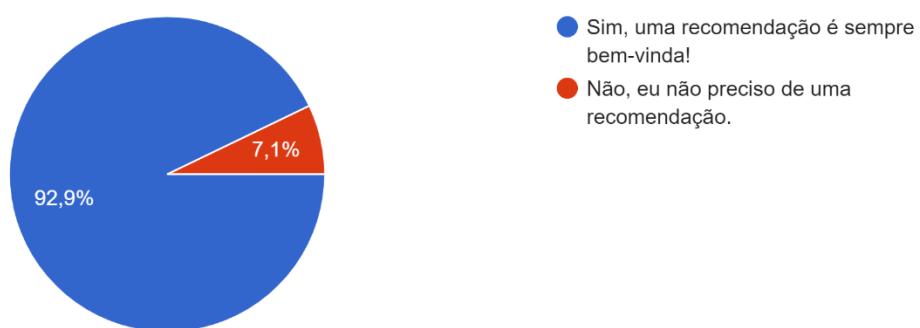
Nisso conseguimos descobrir que mais de 60% dos entrevistados sentiram ou sentiriam dificuldade em encontrar o produto ideal para eles, seja com celulares, tablets, computadores ou notebooks.

Podemos levar em consideração que talvez os entrevistados poderiam ter até encontrado um produto com um custo-benefício melhor para eles, porém não sabem disso, e assim acabaram gastando mais dinheiro com um produto, mas que também serviu para suas necessidades.

#### **Imagen 4 – Quarta pergunta do formulário**

Você acha que uma recomendação de compra, de acordo com aquilo que você procura, te ajudaria a fazer um bom negócio?

84 respostas



Por fim, dentro da nossa quarta e última pergunta, procuramos descobrir sobre o interesse dos nossos entrevistados sobre recomendações de compra para encontrar o produto ideal e com o melhor custo-benefício, que é a base do nosso aplicativo, solucionar as possíveis dúvidas de nossos usuários.

Com isso descobrimos que 92,9% dos entrevistados se interessariam por isso, nos dando a certeza de que esse seria um bom tema para nosso trabalho, e um aplicativo necessário e útil para ser usado no dia a dia, já que sempre teríamos um público ativo, pois sempre irão existir pessoas procurando precisando de auxílio na hora de adquirir novos dispositivos eletrônicos.

## **2 DESENVOLVIMENTO**

### **2.1 Requisitos funcionais e não funcionais do projeto**

Requisitos Funcionais:

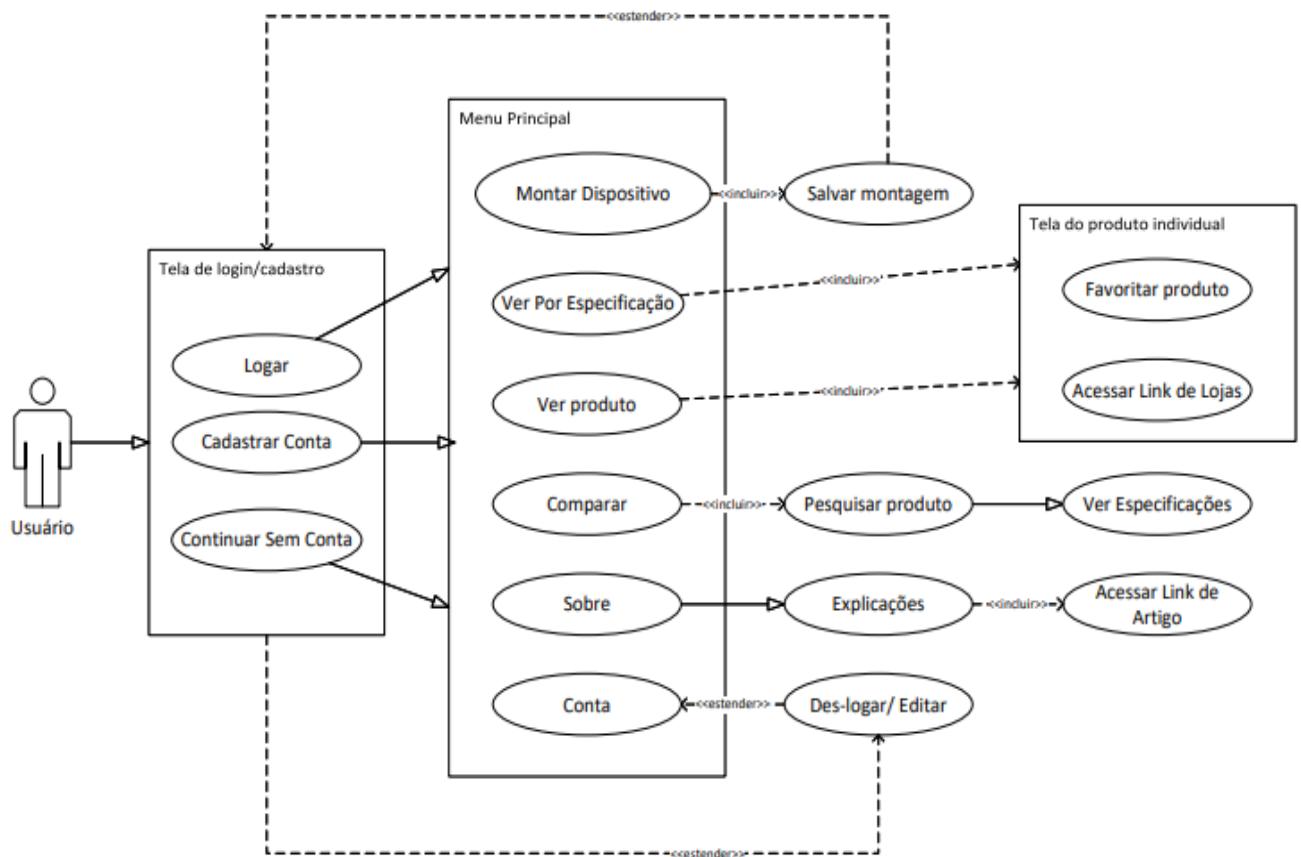
- Consultar Banco de Dados para requisitar dados dos componentes;
- Sistema de Cadastro/Login com a finalidade de salvar setups montados dentro da aplicação;
- Sistema de CRUD (Create, Read, Update, Remove) para a manipulação do banco de dados;
- Possibilidade de continuar sem login na aplicação, mas perdendo o recurso de salvar setups;
- Recurso de reset de senhas;
- Exibir as lojas que possuem o componente assim como o preço em cada uma delas;
- Comparar preços entre lojas e exibir o menor preço;
- Comparar peças e exibir qual é a melhor entre elas (benchmark);
- Recurso de Suporte ao Cliente;

- **Requisitos Não Funcionais:**

- Executável em Plataforma Android;
- Aplicativo desenvolvido em: Java e XML;
- O aplicativo irá operar somente com a execução do usuário;
- Necessário conexão com a internet.

## 2.2 Diagramas de caso de uso do projeto

**Imagen 5 – Diagrama de caso de uso**



A primeira tela é a de Login/Cadastro, por onde o usuário começa ao abrir o aplicativo. Nessa tela existem as seguintes funções: logar, registrar e continuar sem conta, onde todas as funções encaminham-no para o Home. Na tela Home podem ser vistas as funções:

- ✓ Montar dispositivo: que encaminha o usuário para uma outra tela, onde ele poderá estar realizando a montagem de seu computador de acordo com as peças que escolher. É necessário que o usuário tenha logado com sua conta anteriormente para que ele possa salvar sua montagem ou adicionar aos favoritos alguns setups pré-montados.
- ✓ Ver por especificação: Nessa função, será mostrado ao usuário alguns cards em sua tela, os quais, ao selecioná-los de acordo com aquilo que procura, serão exibidos dispositivos que atendem a especificação que ele selecionou.
- ✓ Ver produto: Após escolher o produto que deseja saber mais informações, o usuário pode estar clicando na imagem dele para ser levado a uma nova tela que mostrará todas as especificações do produto escolhido, como: marca, modelo, ano de lançamento, lojas disponíveis etc. Caso deseje adicionar o produto aos favoritos, é necessário que o usuário esteja logado com sua conta.
- ✓ Comparar: A tela comparar tem o objetivo de ajudar o usuário a fazer a melhor escolha ao analisar dois produtos diferentes. Essa função permite a comparação de até 3 produtos diferentes, que terão suas especificações colocadas lado a lado para facilitar a análise de suas características pelo usuário.
- ✓ Sobre: A tela sobre é fundamental para base do aplicativo, nela é realizada uma importante tarefa do app: explicar os componentes eletrônicos ao usuário. A tela exibe cards com as fotos dos componentes e, ao clique do usuário, mostra uma breve explicação sobre a função da peça para o bom funcionamento do computador, notebook, tablet ou celular. Caso o usuário deseje uma explicação mais detalhada sobre a função dos componentes, ele poderá clicar no botão “ver mais” que o levará para uma página na internet com as explicações completas.

- ✓ Conta: A função da tela de conta é exibir ao usuário as principais informações de sua conta, bem como a oferecer a possibilidade de excluí-la. Além de mostrar algumas informações técnicas do app, como o jurídico.

## **2.3 Descrição de todos os softwares que estão sendo utilizados na produção/registro da monografia**

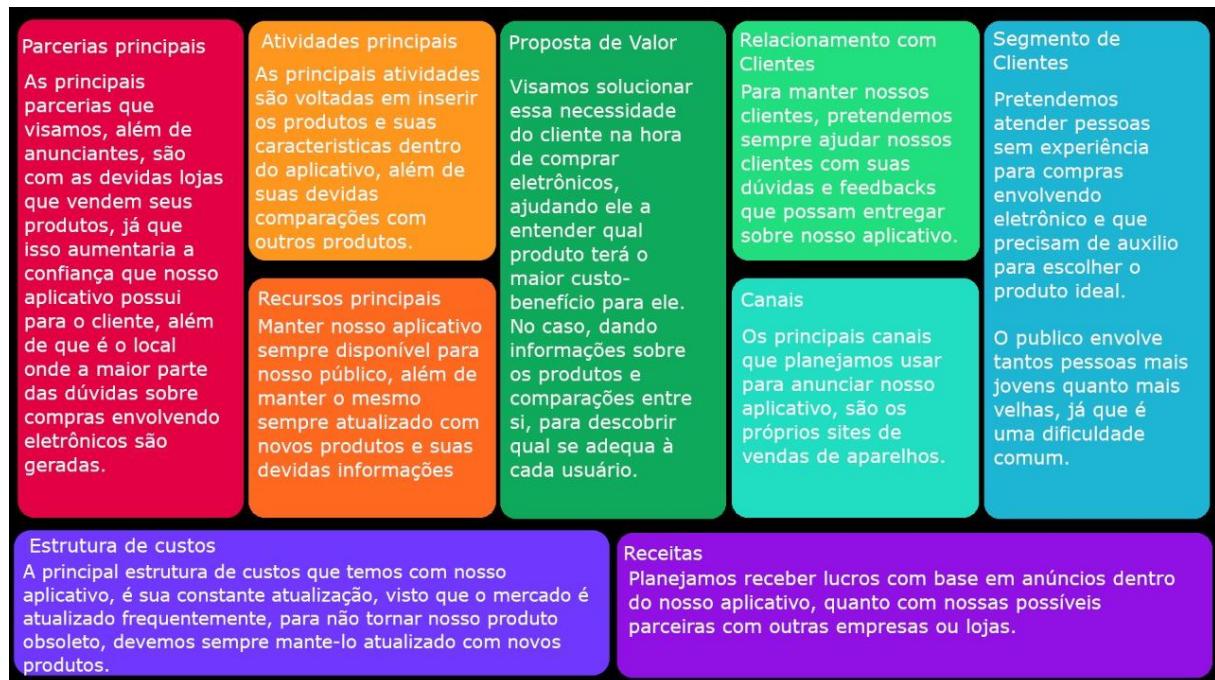
- Diário de bordo
  - ✓ Microsoft Word: usado pela facilidade que o programa proporciona e pela familiaridade que os membros do grupo possuem com o software.
- Cronograma de atividades
  - ✓ Microsoft Excel: usado pela familiaridade que os membros do grupo possuem com o software.
- Diagrama de casos de uso
  - ✓ Microsoft Visio: usado devido a atividades propostas anteriormente pelos professores nas quais o Visio foi utilizado para as desenvolver, gerando uma certa facilidade de uso pelos membros.

## **2.4 Descrição de todos os softwares que estão sendo utilizados na produção do programa/aplicativo do projeto**

- Android Studio;
  - ✓ Programação do aplicativo: Usado pela familiaridade que os membros do grupo têm com o software.
- MySQL;
  - ✓ Manutenção do banco de dados do aplicativo: Usado devido a atividades propostas anteriormente pelos professores nas quais o MySQL foi utilizado para as desenvolver, gerando uma certa facilidade de uso pelos membros.
- Visual Studio Code;
  - ✓ Programação do aplicativo: Também usado devido a atividades propostas anteriormente pelos professores nas quais o Visual Studio Code foi usado para as desenvolver, gerando uma maior facilidade de uso pelos membros.
- Gimp;
  - ✓ Design e prototipagem de telas: Assim como os demais softwares acima, o GIMP também foi usado devido a atividades propostas anteriormente pelos professores nas quais o mesmo foi usado para as desenvolver, além de ser um software gratuito e mais leve que os demais programas disponíveis na internet.
- ibisPaint X;
  - ✓ Design e desenvolvimento do logo: utilizado devido a sua disponibilidade em aparelhos móveis, o que facilitou a contribuição de membros do grupo que não tem acesso a computadores ou a softwares de edição para com o design.

## 2.5 Canvas e modelo de negócios.

### Imagen 6 - Canvas



Com o uso do Canvas, que concluímos durante a montagem do projeto no 3º Bimestre, tivemos uma noção maior de como cada etapa do nosso projeto funcionaria, desde partes envolvendo os custos e receitas até as parcerias e recursos principais para nosso aplicativo funcionar.

Dentro do nosso segmento de clientes, pretendemos atender pessoas sem experiência para compras envolvendo os eletrônicos. Envolvendo tanto pessoas mais jovens quanto pessoas mais velhas, visto que é uma dificuldade comum entre as pessoas, como foi mostrado durante nossas pesquisas.

No nosso relacionamento com clientes, pretendemos manter os mesmos visando em sempre ajudá-los, seja com nosso aplicativo, ou com dúvidas diretas enviadas para nossa equipe, caso nosso aplicativo não seja o suficiente.

Nossos principais canais pensados para promover nosso aplicativo, seriam os próprios sites de lojas que vendem eletrônicos, já que seria o primeiro local onde alguém poderia encontrar alguma dúvida envolvendo isso.

Já na nossa proposta de valor, seriam as próprias informações dispostas para o cliente, sendo esse nosso diferencial, possuir tudo que nosso cliente possivelmente precisa saber para escolher seu produto, de forma detalhada e organizada.

As principais atividades que nossa equipe possui com nosso aplicativo são voltadas em inserir os produtos e suas características nele, mantendo o mesmo sempre atualizado, além das comparações presentes no app envolvendo cada produto.

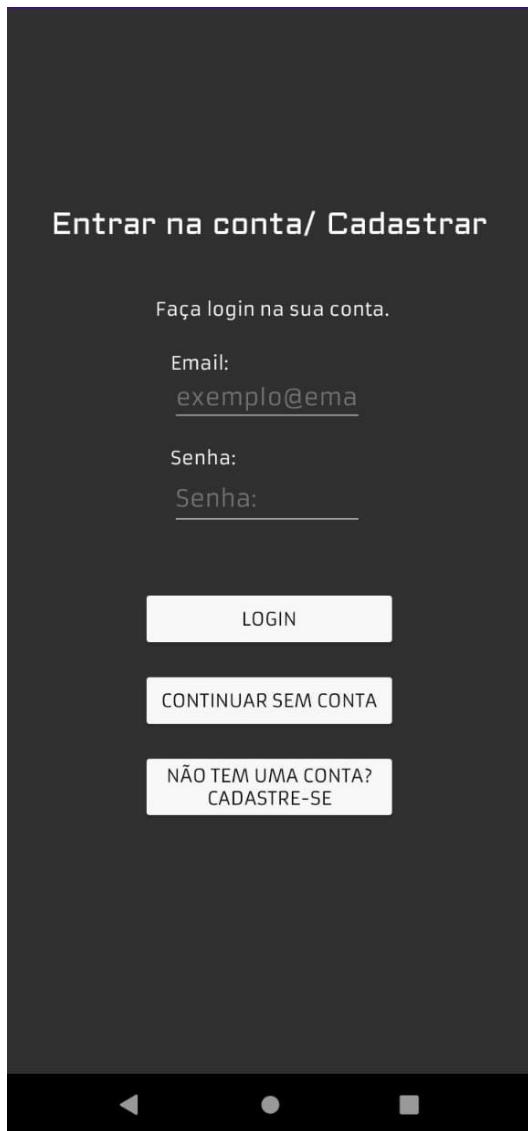
Os principais recursos que precisamos para manter nosso aplicativo funcionando, são os próprios dados de produtos, que precisam sempre estar atualizados, visto que o mercado atualiza rapidamente com novos produtos, além de manter nosso aplicativo sem erros e funcionando.

As possíveis parcerias que visamos com nosso aplicativo, são com as próprias lojas de produtos, onde podemos por exemplo, destacar produtos de lojas específicas nas telas iniciais do nosso aplicativo, enquanto isso, os nossos parceiros poderiam colocar nosso aplicativo em anúncios dentro do próprio site, já que nos daria uma credibilidade maior com nossos clientes.

Por fim, nossa estrutura de custos é voltada em manter nosso aplicativo em funcionamento e atualizado, enquanto nossa receita é gerada por anúncios e pelas possíveis parcerias.

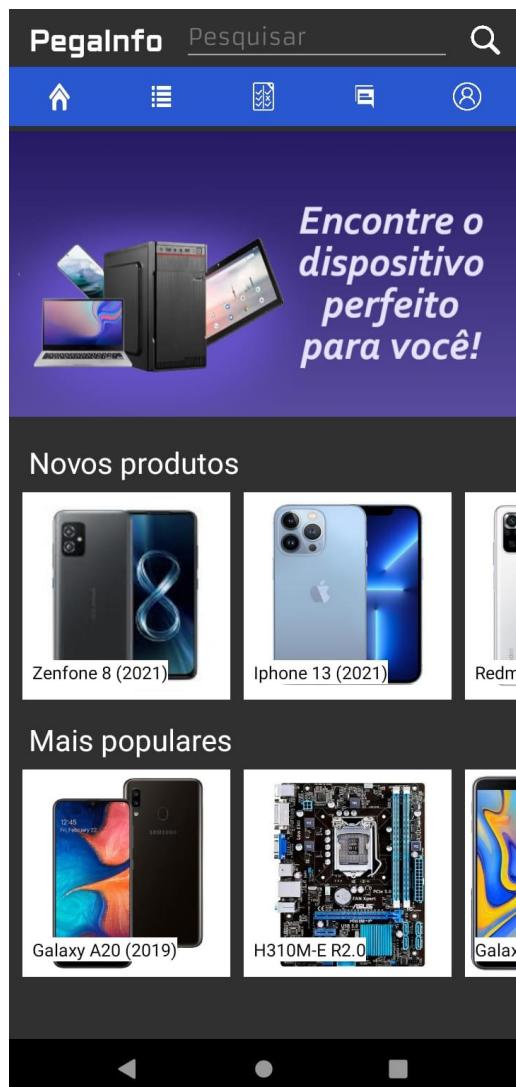
## 2.5 Telas do projeto

Imagen 7 - Tela – Login/Cadastro



Tela de Login/Cadastro, a primeira tela do aplicativo. Nessa tela existem as seguintes funções: logar, registrar e continuar sem conta, onde todas a funções encaminham-no para o Home.

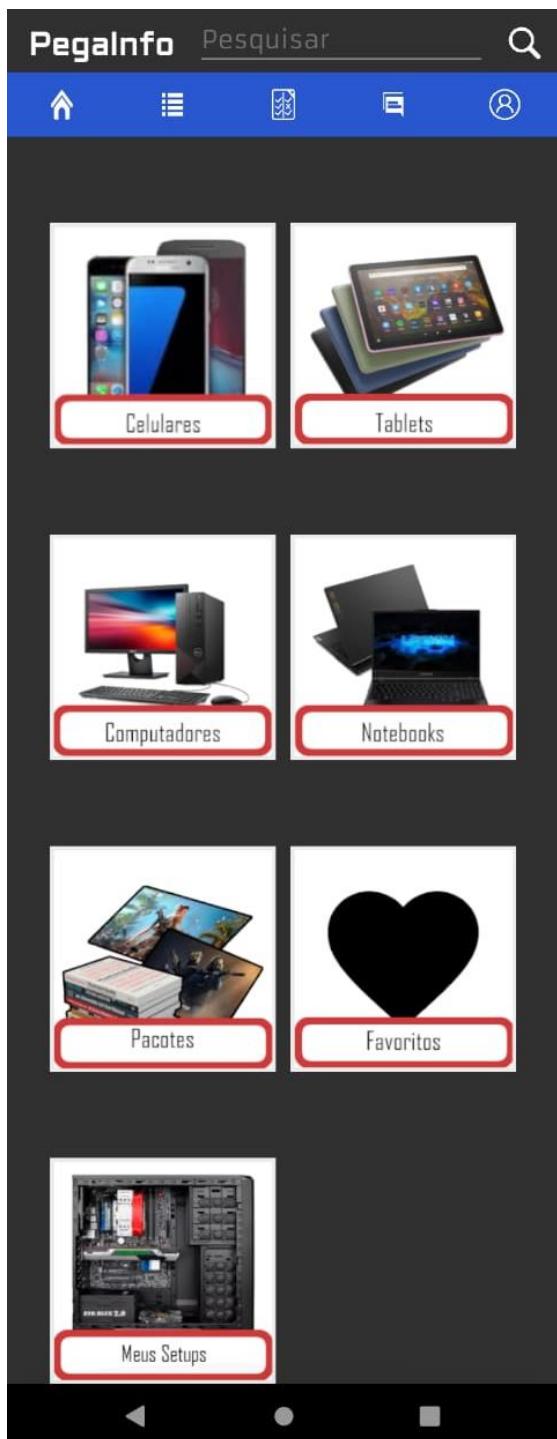
## Imagen 8 - Tela - Home



A tela Home está dividida em 3 subtópicos:

- Encontrar dispositivo: leva o usuário a tela de procura por especificação, na qual ele pode selecionar a qual demanda ele quer que o dispositivo procurado atenda.
- Novos produtos: Como o próprio título já diz, essa sessão mostra os principais novos lançamentos do mercado tecnológico.
- Mais populares: essa função agrupa os dispositivos mais procurados e adquiridos do mercado, separando-os para o usuário poder ter a ciência de quais produtos estão em alta.

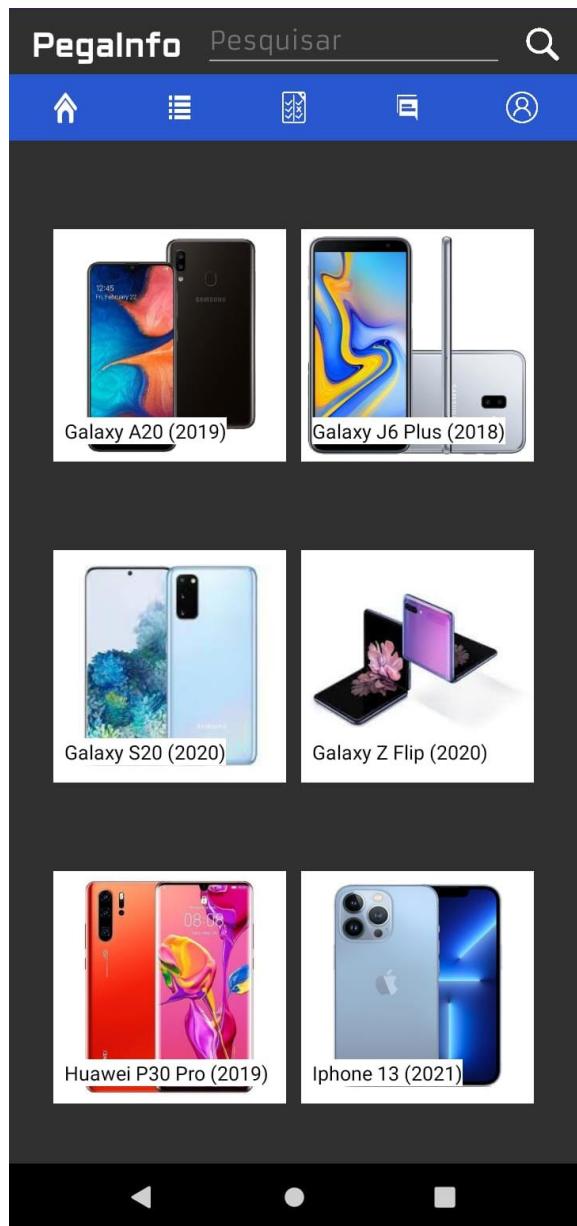
## Imagen 9 - Tela – Menu



No menu de produtos, o usuário pode encontrar de forma mais prática seu produto desejado, além de poder procurar por pacotes de produtos e setups já prontos.

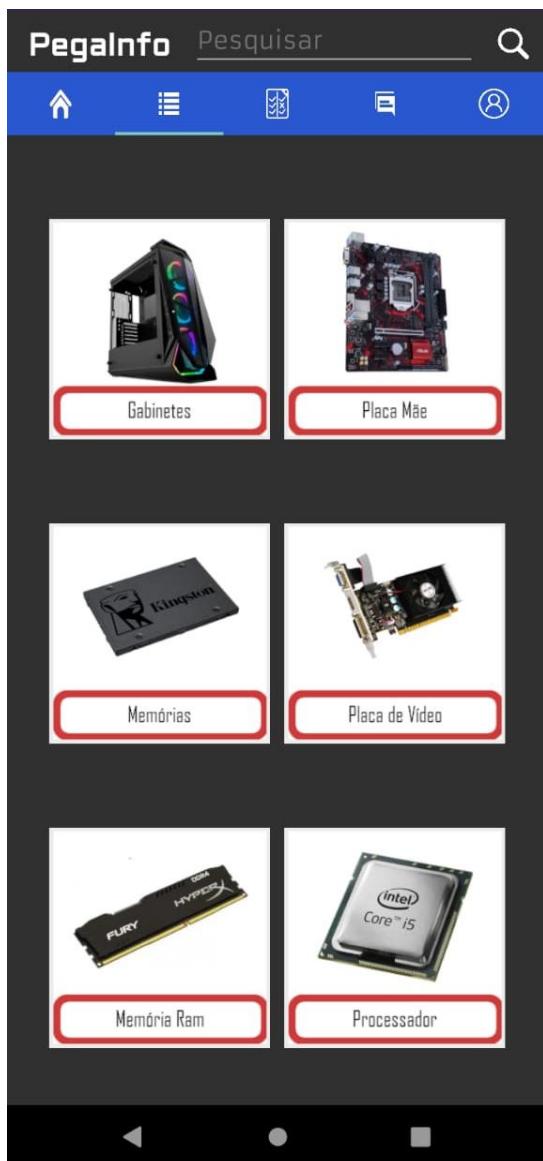
Nessa tela estão listados todos os produtos cadastrados no aplicativo, os quais o usuário, com um clique pode saber mais sobre suas especificações.

## Imagen 10 – Tela\_Celulares



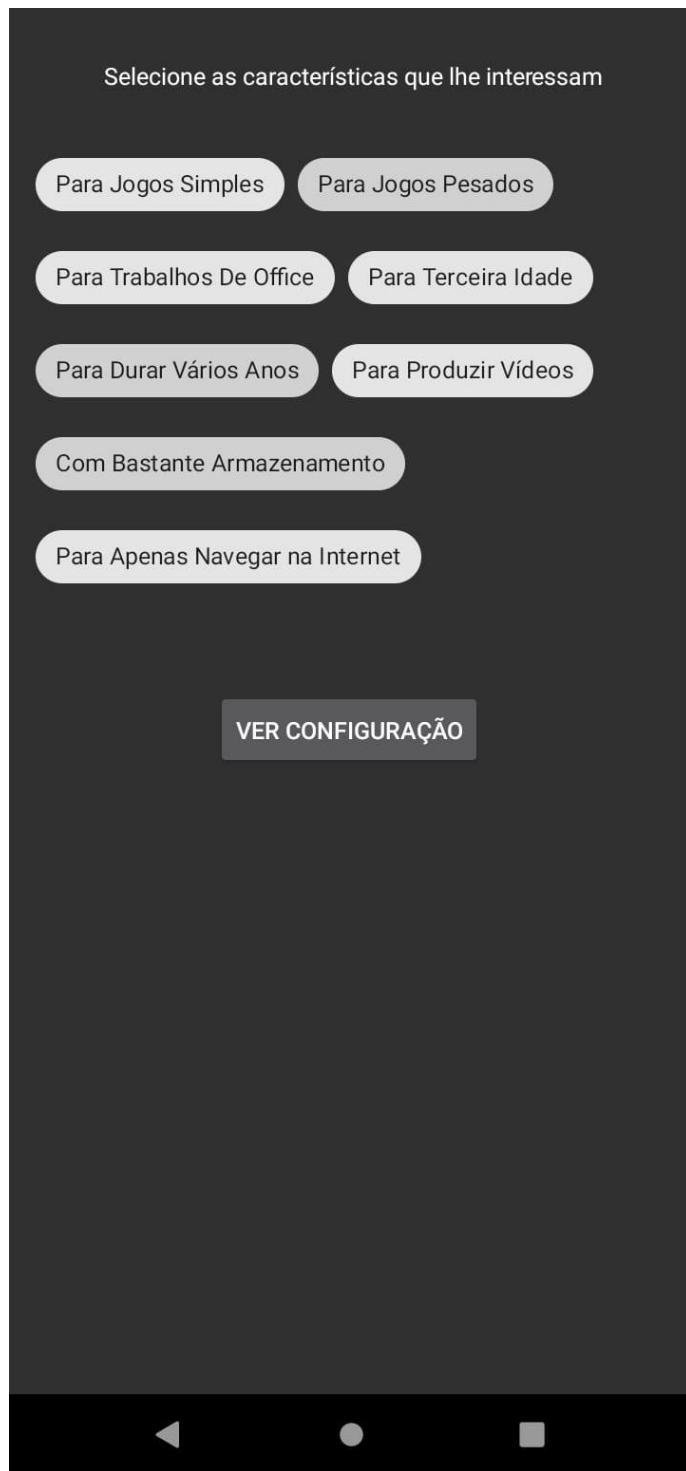
Nesta tela, o usuário tem uma lista da categoria que foi selecionada na tela de Menu, neste caso, foi selecionado a categoria de celulares, e então é listado todos os celulares cadastrados. Assim funciona para as outras categorias e também para as categorias do Menu PC's, apenas variando o que será listado.

### Imagen 11 – Menu PC's



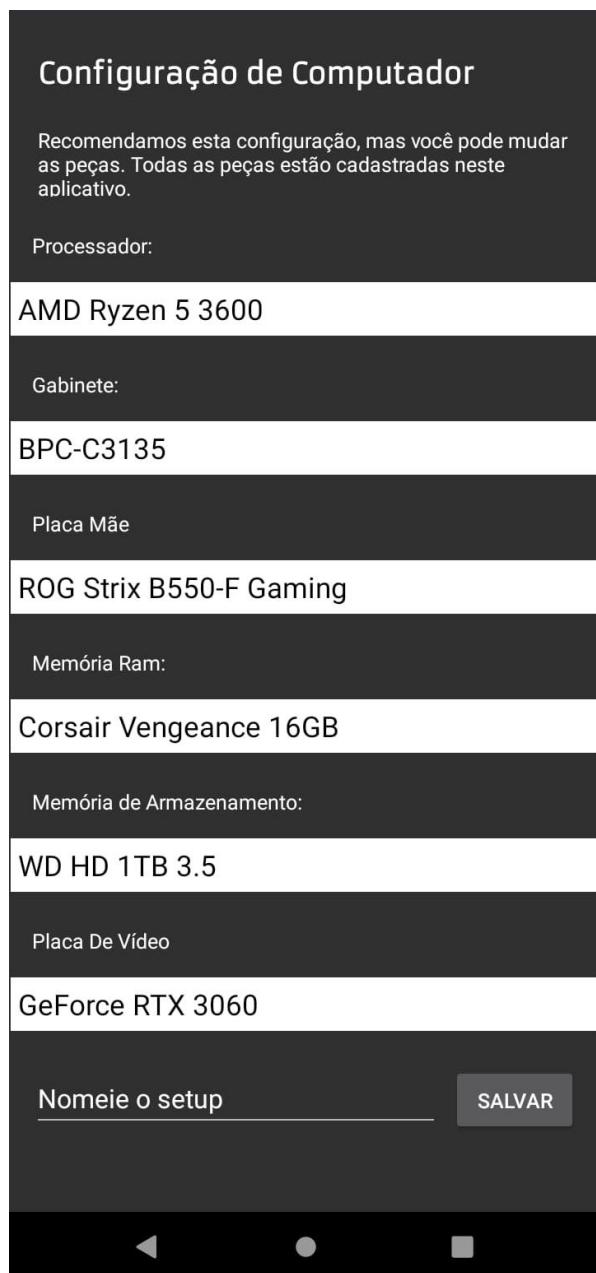
Ainda dentro da sessão de menu há uma tela específica para as peças de computador, todas separadas de acordo com seus tipos e finalidades.

## Imagen 12 – Tela\_Specs\_PC's



Nesta tela, o usuário escolhe as características que lhe interessam para o seu computador, assim que escolher, ele poderá continuar para a tela Montagem, pelo botão Ver configuração.

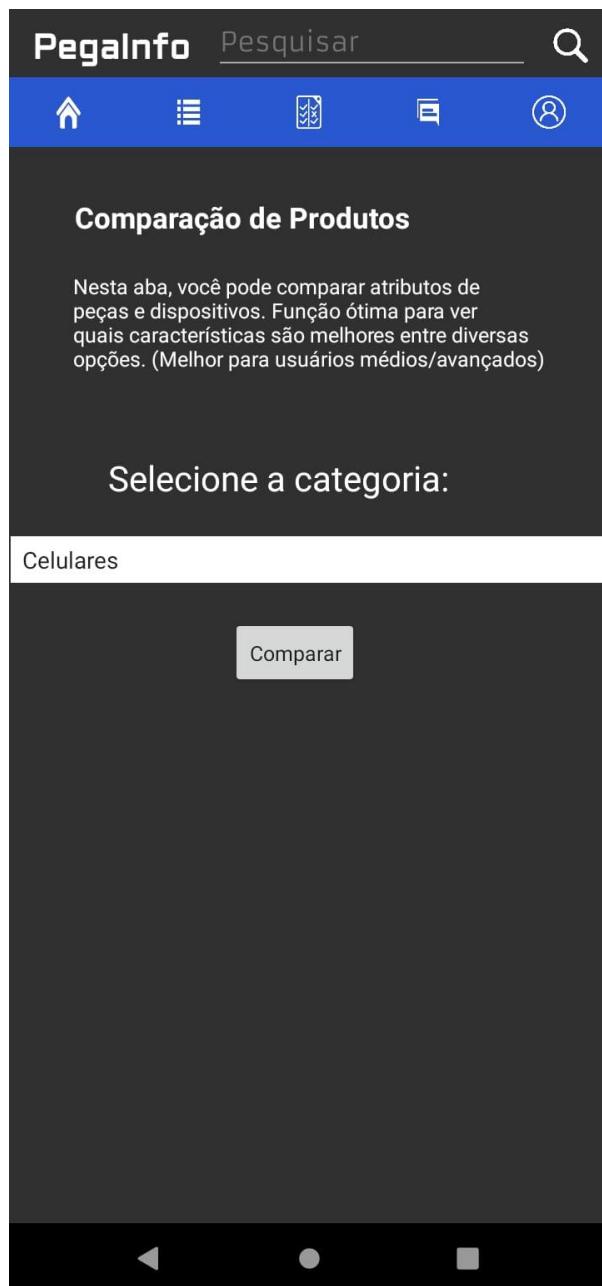
### Imagen 13 - Montagem



Nessa tela poderá estar realizando a montagem de seu computador de acordo com as peças que escolher. É necessário que o usuário tenha logado com sua conta anteriormente para que ele possa salvar sua montagem ou adicionar aos favoritos alguns setups pré-montados.

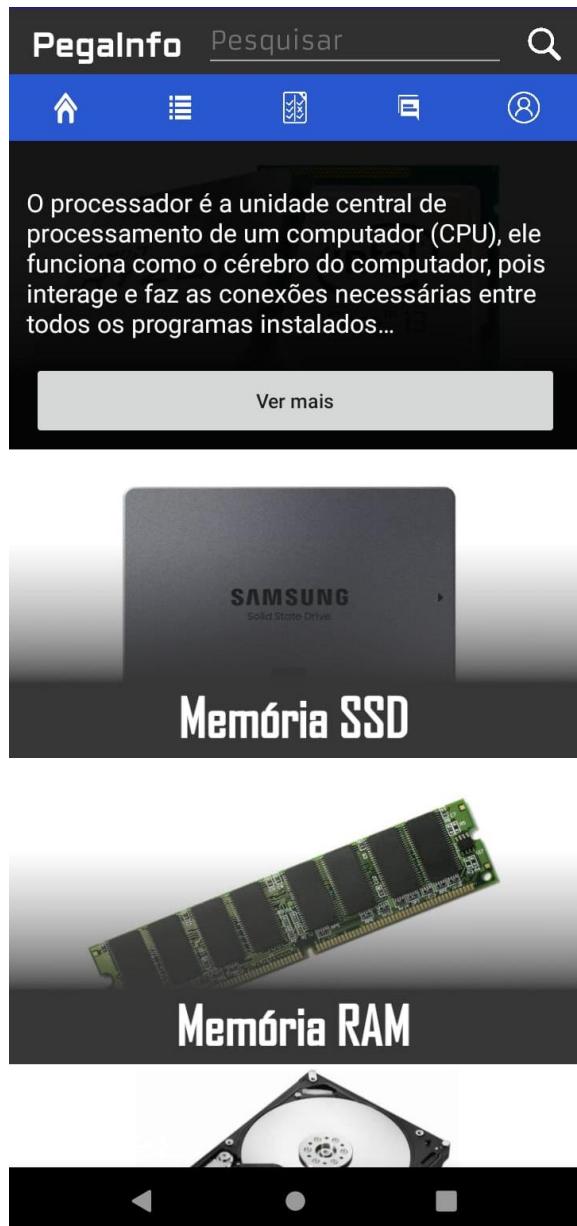
Além de poder nomear os setups, o usuário poderá salvá-los e adicioná-los aos favoritos.

## Imagen 14 - Tela – Comparação



A tela comparar tem o objetivo de ajudar o usuário a fazer a melhor escolha ao analisar dois produtos diferentes. Essa função permite a comparação de até 3 produtos diferentes, que terão suas especificações colocadas lado a lado para facilitar a análise de suas características pelo usuário.

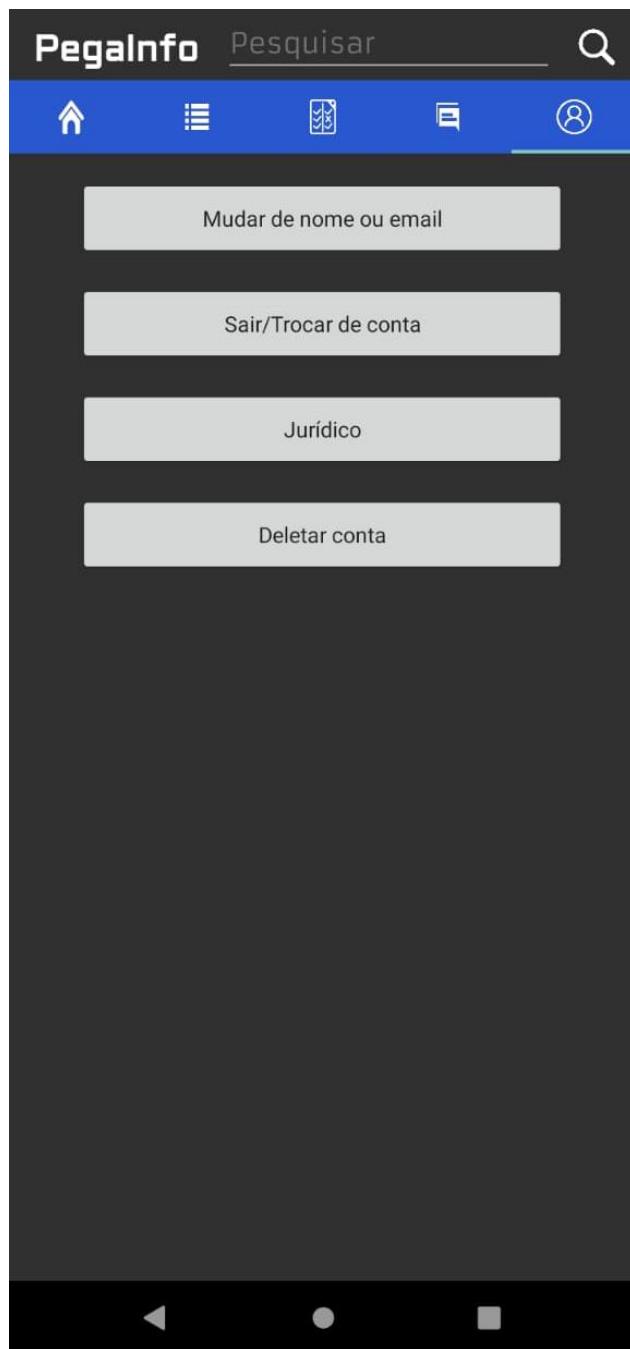
## Imagen 15 – Tela - Sobre



A tela sobre é fundamental para base do aplicativo, nela é realizada uma importante tarefa do app: explicar os componentes eletrônicos ao usuário. A tela exibe cards com as fotos dos componentes e, ao clique do usuário, mostra uma breve explicação sobre a função da peça para o bom funcionamento do computador, notebook, tablet ou celular.

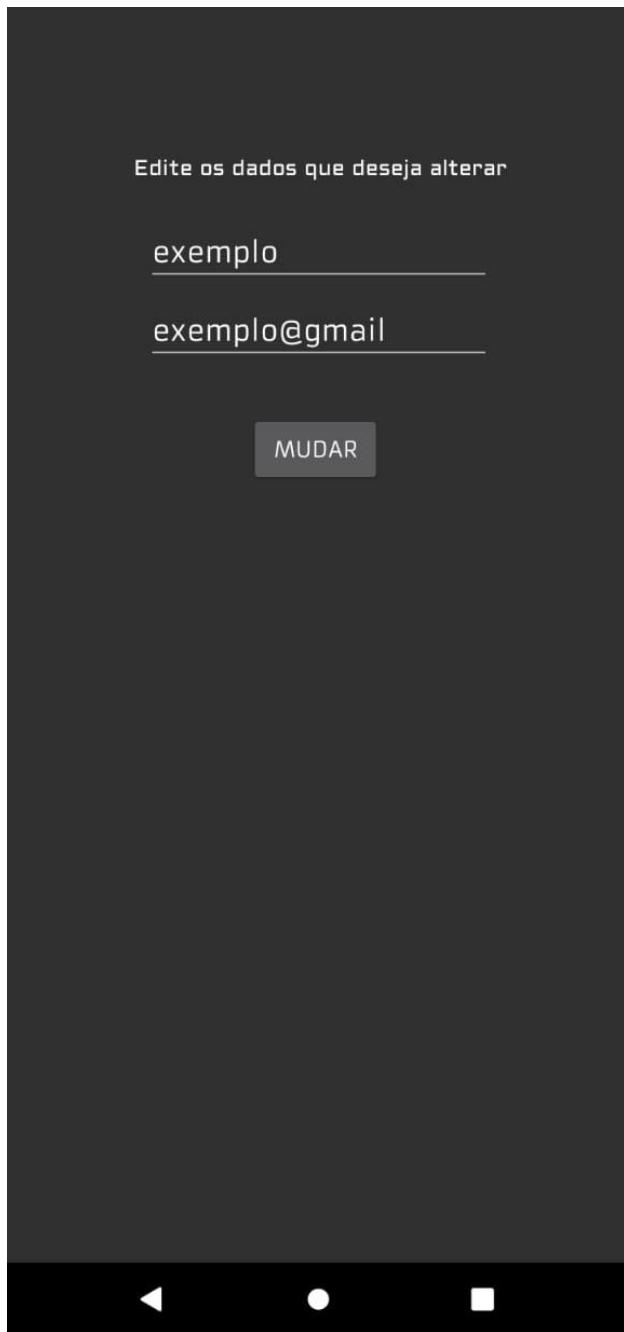
Caso o usuário deseje uma explicação mais detalhada sobre a função dos componentes, ele poderá clicar no botão “ver mais” que o levará para uma página na internet com as explicações completas.

**Imagen 16 - Tela – Conta**



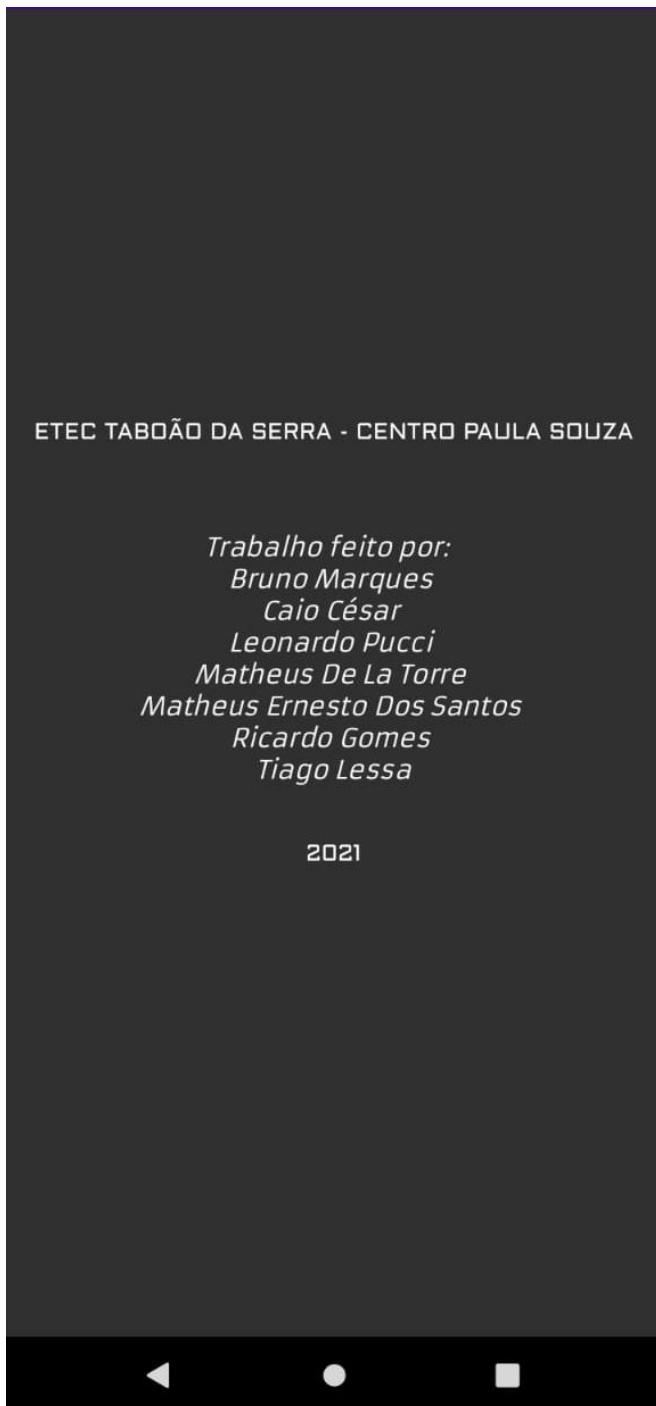
A função da tela de conta é exibir ao usuário as principais informações de sua conta, bem como a oferecer a possibilidade de excluí-la. Além de mostrar algumas informações técnicas do app, como o jurídico.

**Imagen 17 - Tela – Alterar dados**



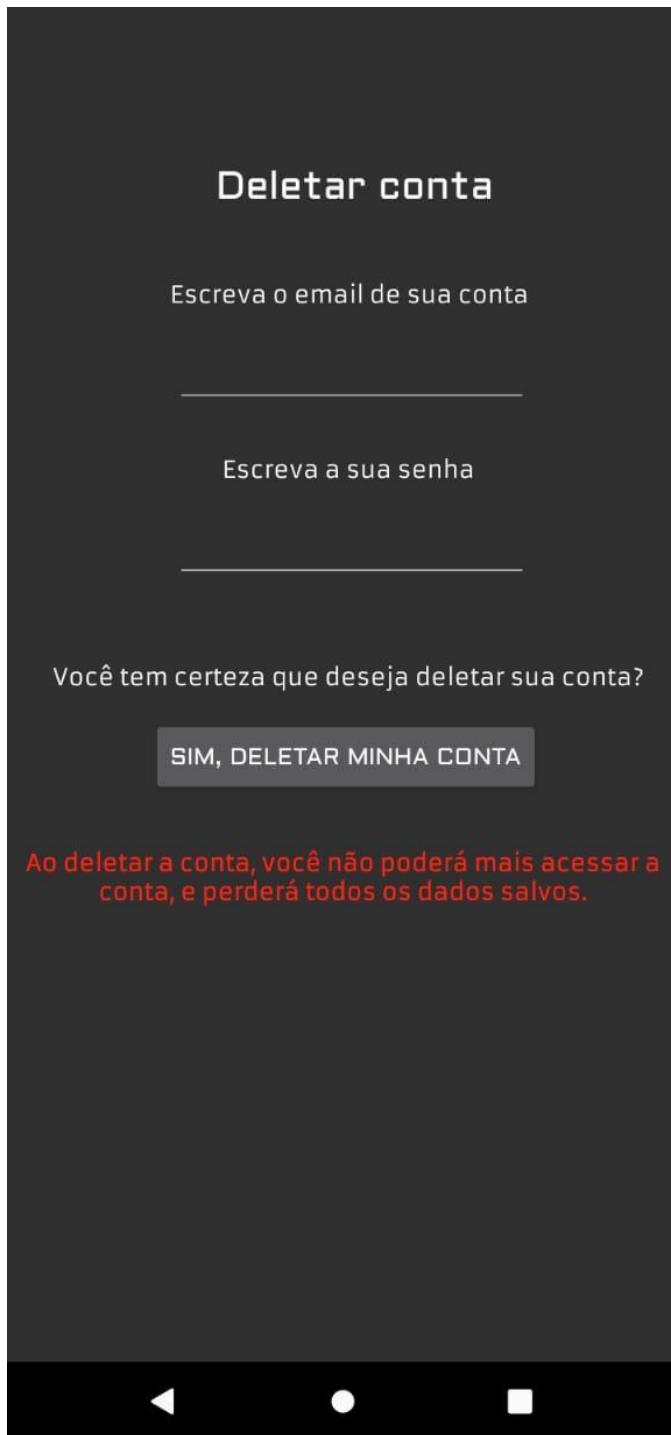
Na tela de alterar dados é possível, como o próprio nome já deixa explícito, alterar as informações da sua conta, como o e-mail, a senha e o nome de usuário.

### Imagen 18 - Tela – Jurídico



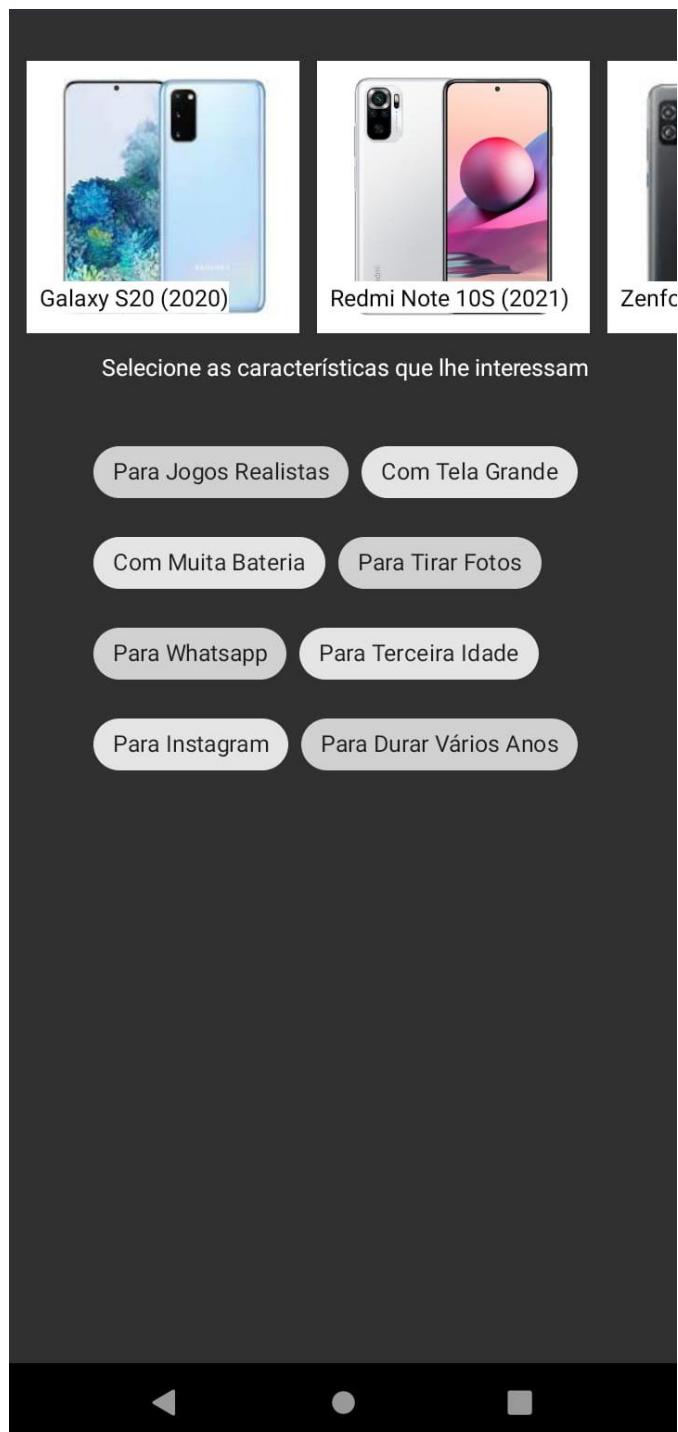
Essa tela mostra os responsáveis pelo desenvolvimento do aplicativo e sua apresentação. E no futuro contará com os meios de contato de cada membro, seja e-mail, WhatsApp ou telefone.

**Imagen 19 - Tela – Deletar conta**



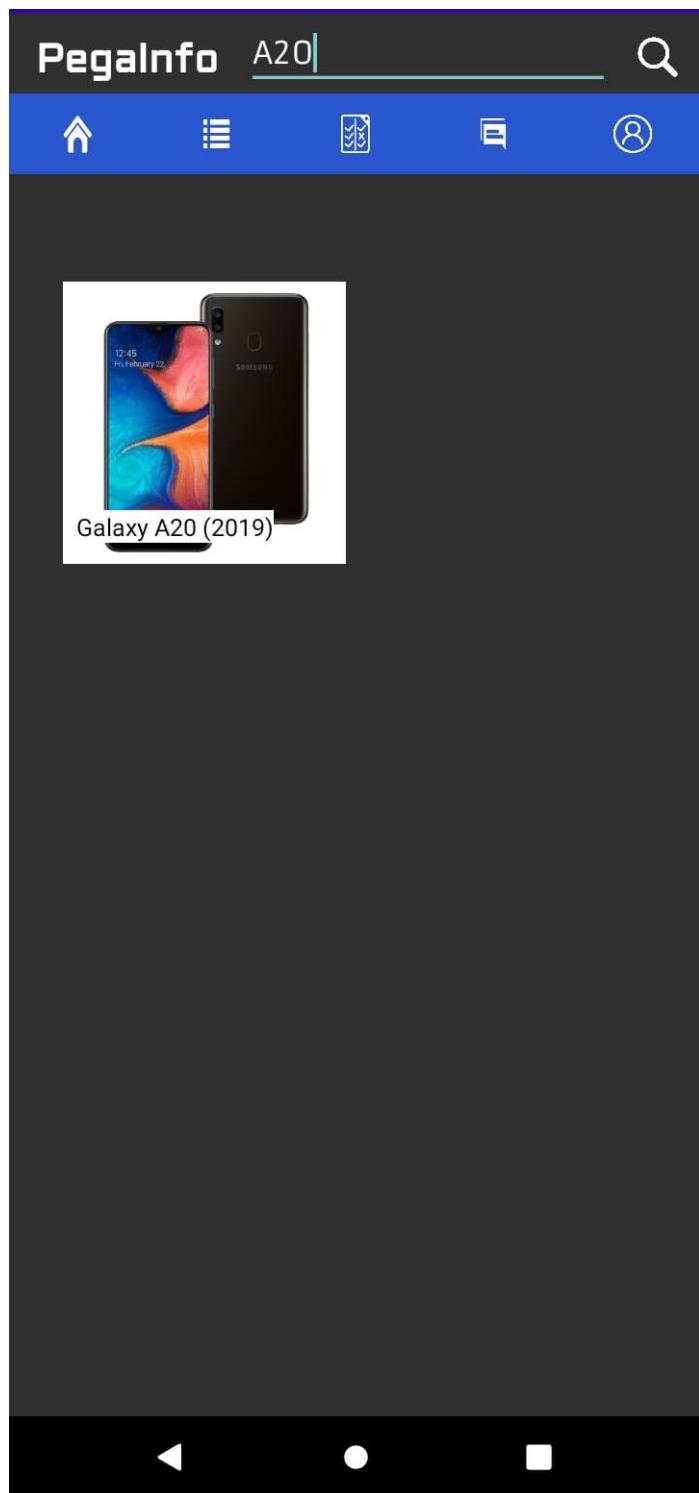
A tela de deletar conta, como o próprio nome já explicita sua função, serve para o usuário deletar sua conta se não a quiser mais.

## Imagen 20 – Tela\_Specs\_Celulares



Nessa tela, é mostrado alguns cards ao usuário, os quais, ao selecioná-los de acordo com aquilo que procura, serão exibidos dispositivos que atendem a especificação que ele selecionou.

### Imagen 21 - Tela – Pesquisa\_Produto



Essa tela mostra a função de pesquisar do aplicativo em funcionamento, exibindo o produto de acordo com aquilo que o usuário escreve na barra de pesquisa.

## Imagen 22 – Especs\_Produto

The screenshot shows a mobile application interface for a Samsung Galaxy A20 (2019). At the top, there is a navigation bar with icons for home, search, and account. Below the bar, the product name "Galaxy A20 (2019)" is displayed next to a small image of the phone. The date "2019-04-19" is also visible. A heart icon indicates the item can be favorited. The main content area contains a detailed description of the phone's features and a table of technical specifications.

**Descrição:** O Galaxy A20 é um modelo intermediário de entrada, mas que tem o grande diferencial de trazer uma tela com a tecnologia Super AMOLED da Samsung, ainda que com uma resolução mais baixa (HD+). Mesmo assim, vem com uma qualidade bem acima da média...

**Informações técnicas do produto**

Marca:	Samsung
Quantidade de RAM:	3gb
Processador:	SAMSUNG Exynos 7 Octa 7884
Tipo de tela:	Super AMOLED
Tamanho (Polegadas):	6.40"
Câmeras:	Câmera Principal (13MP) e Ultra Wide (5MP)
Sistema Operacional:	Android 10
Quantidade de Memória:	32gb
Bateria:	4000mAh
Resolução de Tela:	720x1560

**Lojas:**

Amazon	TrocaFone (Usado)	Casas Bahia
--------	-------------------	-------------

**ABRIR**   **ABRIR**   **ABRIR**

Após escolher o produto que deseja saber mais informações, o usuário pode estar clicando na imagem dele para ser levado a uma nova tela que mostrará todas as especificações do produto escolhido, como: marca, modelo, ano de lançamento, lojas disponíveis etc. Caso deseje adicionar o produto aos favoritos, é necessário que o usuário esteja logado com sua conta.

## 2.6 Códigos do projeto

### Função do botão de voltar

#### Imagen 23 - Códigos

```
@Override  
public void onBackPressed(){  
    switch (bckComando){  
        case "home":  
            TelaPrincipal();  
            break;  
        case "comparar":  
            TelaPrincipalDef();  
            util.trocaJanela(findViewById(R.id.lay_janela),R.layout.tela_principal_comparar);  
            PrincipalComparar();  
            break;  
        case "menu":  
            TelaPrincipalDef();  
            util.trocaJanela(findViewById(R.id.lay_janela),R.layout.tela_principal_menu);  
            PrincipalMenu();  
            break;  
        case "conta":  
            TelaPrincipalDef();  
            util.trocaJanela(findViewById(R.id.lay_janela),R.layout.tela_principal_conta);  
            PrincipalConta();  
            break;  
        default:  
            super.onBackPressed();  
            break;  
  
    }  
}
```

Esta função intercepta o evento do botão Voltar, e com uma variável global, recepta um comando pedido pelo próprio código e age de acordo com o nome.

Por exemplo, se o comando for “home”, o evento irá executar a função que carrega a tela principal juntamente com a exibição da tela de Home. O comando “Comparar” carrega a tela principal, e exibe a aba de Comparar, assim, termina carregando os comandos de comparar.

A variável bckPressed serve para, durante a execução do código, o comportamento do botão Voltar seja modificado. Por exemplo, ao MenuSelect ser executado (A Tela de Menu de peças de computadores), o bkcPressed muda para “menu”, para que o botão Voltar volte a tela de Menu.

## Validação de e-mail e senha antes do POST

### Imagen 24

```
152     if(email.equals("") && senha.equals(""))
153     {
154         alerta.Popup("Preencha o formulário antes de entrar",false);
155     }
156     else if(email.equals(""))
157     {
158         alerta.Popup("Preencha o email antes entrar",false);
159     }
160     else if(senha.equals(""))
161     {
162         alerta.Popup("Preencha o senha antes entrar",false);
163     }
164     else if(!email.contains("@") && !email.contains(".")) && email.length() <= 4)
165     {
166         alerta.Popup("Utilize um email válido", false);
167     }
168     else
169     {
170         if(v == findViewById(R.id.btn_xml_Cadastrar)){
171             conectador.cadastrar(resposta,null,email,senha);
172         } else {
173             conectador.logar(resposta,null,email,senha);
174         }
175     }
176 }
177 
```

Esta parte da função Login() serve para verificar se o e-mail e senha estão preenchidos corretamente para ser enviado, tendo pelo menos, um '@' no e-mail, ter mais caracteres ou igual a 4 e a senha não estar vazia, assim, é passado para o servidor validar o cadastro em verificar se existe a conta, se a senha está certa ou se é valido.

## Conexão com POST

### Imagen 25

```
43 public void conectarPOST(Response.Listener<String> resposta, Map<String, String> dados)
44 {
45
46     StringRequest requisicao = new StringRequest(Request.Method.POST, link, resposta, falha_padrao){
47         @Override
48         protected Map<String, String> getParams() throws AuthFailureError{
49             return dados;
50         }
51     };
52     conexao.add(requisicao);
53 }
54
55 public void conectarPOST(Response.Listener<String> resposta, Response.ErrorListener erro, Map<String, String> dados)
56 {
57     StringRequest requisicao = new StringRequest(Request.Method.POST, link, resposta, erro){
58         @Override
59         protected Map<String, String> getParams() {
60             return dados;
61         }
62     };
63     conexao.add(requisicao);
64 }
```

O Aplicativo PegalInfo utiliza o Volley, uma API do Google criada para o celular conseguir enviar dados de formulário PHP de maneira fácil e sem necessidade de criar threads separadas do aplicativo. Com o conectarPOST, o código o invoca utilizando um Response.Listener, que é um trecho de código que esperará até a resposta da internet para ser executado, além disso, utiliza um ErrorListener, que serve para capturar falha de conexão ou de envio pela internet. O Map de String “dados” é um array com Keys e Values de String que serve como \$\_POST para enviar ao servidor PHP como um <form>.

## Cancelamento de conexão e Links

### Imagen 26

```
77 public void cancelarConexoes(int tipo){  
78     switch (tipo){  
79         case 1:  
80             break;  
81         default:  
82             conexao.cancelAll(new RequestQueue.RequestFilter() {  
83                 @Override  
84                 public boolean apply(Request<?> request) {  
85                     return true;  
86                 }  
87             });  
88             break;  
89     }  
90 }  
91 }  
92  
93 public void abrirLINK(String URL){  
94     Intent browserIntent = new Intent(Intent.ACTION_VIEW, Uri.parse(URL));  
95     act.startActivity(browserIntent);  
96 }
```

O Cancelamento de conexão cancela conexões do Volley, para que o aplicativo consiga diminuir o consumo de dados quando algo não é mais preciso, por exemplo, ao alternar rapidamente entre a tela de Home e a de Comparar, o aplicativo cancela as conexões chamadas da tela de Home de baixar imagens de produtos, e assim também com outras telas que utilizam internet.

A função de abrirLink() permite o aplicativo executar um navegador de escolha do usuário para abrir um URL, que normalmente é um artigo sobre componentes ou produto em uma loja. Esta função também aceita que o link seja aberto no aplicativo da loja, por exemplo, se o link aberto for da Kabum, será perguntado se o usuário gostaria de abrir no Aplicativo KaBuM (se estiver instalado).

## TestarConexao() – Android Studio

### Imagen 27

```
public void testarConexao(){
    setContentView(R.layout.tela_carregando);

    alerta = new Testador(this);
    conector = new WebConn(this);
    util = new Utilidades(this);

    sharedPref = getPreferences(MODE_PRIVATE);
    SPEditor = sharedPref.edit();
    SPEditor.apply();

    Response.Listener resposta = response -> {
        if(response.toString().equals("Sucesso")){
            Login();
        }else {
            alerta.Popup("Erro: Servidor Desconfigurado",true);
        }
    };

    Response.ErrorListener erro = error -> {
        setContentView(R.layout.tela_sem_conexao);
        findViewById(R.id.btn_Recarregar).setOnClickListener(v -> testarConexao());
    };
    conector.testar(resposta, erro);
}
```

Nesta função, o aplicativo exibe na tela o logo e o título do aplicativo. Assim, testa a conexão do aplicativo, enviando um POST ao servidor, que verificará se o mySql está ligado corretamente e, se tiver, retornará uma mensagem ao aplicativo de sucesso. Se o aplicativo receber uma mensagem “Sucesso”, então quer dizer que tudo está ocorrendo corretamente e pode prosseguir com o código. Se não tiver o mySql funcionando, mas o localhost estiver ligado, exibirá uma mensagem de “Erro: Servidor Desconfigurado.”. Porém, se não haver internet ou resposta do servidor, o aplicativo mostrará uma tela de “Sem Internet”, com o botão de “Reconectar”, que tentará executar mais uma vez ao servidor, utilizando a mesma função.

## Login() – Android Studio

### Imagen 28

```
public void Login(){
    if(!sharedPref.getString("SAVED_EMAIL","",).equals(""))
        && !sharedPref.getString("SAVED_SENHA","",).equals("")){
        TelaPrincipal();
    }
    else {
        setContentView(R.layout.tela_login);

        findViewById(R.id.btn_xml_EstarSemConta).setOnClickListener(v -> TelaPrincipal());
        View.OnClickListener Clicado_ComConta_Cadastrar = v -> {
            EditText edt_email = findViewById(R.id.edt_xml_Email);
            EditText edt_senha = findViewById(R.id.edt_xml_Senha);
            String email = edt_email.getText().toString();
            String senha = edt_senha.getText().toString();

            Response.Listener resposta = response -> {
                try {
                    String status_result = new
                        JSONObject(response.toString()).getString("status");
                    String nome_result = new
                        JSONObject(response.toString()).getString("nome");

                    switch (status_result) {
                        case "Logado":
                            alerta.Popup("Seja bem-vindo, " + nome_result,
                                false);
                            SPEDitor.putString("SAVED_EMAIL",email);
                            SPEDitor.putString("SAVED_SENHA",senha);
                            SPEDitor.apply();
                            TelaPrincipal();
                            break;
                        case "Tem conta, senha incorreta":
                            alerta.Popup("Conta existente, porém senha incorreta",
                                false);
                            .
                            .
                            .
                    }
                } catch (JSONException e) {
                    alerta.Popup(e.getMessage(),false);
                }
            };
        };
    }
}
```

### Imagen 29

```
SPEDitor.putString("SAVED_EMAIL",email);
SPEDitor.putString("SAVED_SENHA",senha);
SPEDitor.apply();
```

---

```
TelaPrincipal();
break;
default:
    alerta.Popup("Aviso: " + status_result, false);
    break;
}
} catch (JSONException e) {
    alerta.Popup(e.getMessage(),false);
}
};

if(email.equals("") && senha.equals(""))
{
    alerta.Popup("Preencha o formulário antes de entrar",false);
}
else if(email.equals(""))
{
    alerta.Popup("Preencha o email antes entrar",false);
}
else if(senha.equals(""))
{
    alerta.Popup("Preencha a senha antes entrar",false);
}
```

### Imagen 30

```
        else if(!email.contains("@") && !email.contains(".")) && email.length() <= 4)
    {
        alerta.Popup("Utilize um email válido", false);
    }
else
{
    if(v == findViewById(R.id.btn_xml_Cadastrar)){
        conector.cadastrar(resposta,null,email,senha);
    } else {
        conector.logar(resposta,null,email,senha);
    }
}

findViewById(R.id.btn_xml_EstarComConta).setOnClickListener(Clicado_ComConta_Cadastrar);
findViewById(R.id.btn_xml_Cadastrar).setOnClickListener(Clicado_ComConta_Cadastrar);
}
}
```

Nesta função, é checado inicialmente se o usuário já logou antes no aplicativo, se já, o usuário será redirecionado para tela principal. Caso não tenha logado antes, será redirecionado para a tela de Login, onde poderá inserir um e-mail e senha em seus respectivos campos de texto, e quando o botão de Logar for pressionado, o aplicativo checará se existe tal registro e se as informações coincidem, assim que Logar, será coletado o nome do usuário do banco e exibido uma mensagem de Boas-Vindas com seu nome. Caso o botão pressionado seja de cadastrar, o e-mail e senha inseridos anteriormente serão cadastro no banco de dados. Caso ainda o botão de Continuar Sem Conta seja pressionado, o aplicativo continuará sem a função de Editar Cadastro e Deletar Conta na Tela de Conta, além de não poder favoritar produtos e também não poder salvar Setups Montados de Computador. Além disso, todos os botões levam a TelaPrincipal(), onde é exibido a tela de Home.

## TelaPrincipal() – Android Studio

### Imagen 31

```
public void TelaPrincipal(){
    conector.cancelarConexoes(0);
    setContentView(R.layout.tela_principal);

    util.trocaJanela(findViewById(R.id.lay_janela),R.layout.tela_principal_home);

    TabLayout tabsLayt = findViewById(R.id.tab_menus);
    tabsLayt.addOnTabSelectedListener(new TabLayout.OnTabSelectedListener() {
        @Override
        public void onTabSelected(TabLayout.Tab tab) {
            switch(tab.getPosition()){
            {
                case 1:
                    util.trocaJanela(findViewById(R.id.lay_janela),R.layout.tela_principal_menu);
                    PrincipalMenu();
                    break;
                case 2:
                    util.trocaJanela(findViewById(R.id.lay_janela),R.layout.tela_principal_comparar);
                    PrincipalComparar();
                    break;
                case 3:
                    util.trocaJanela(findViewById(R.id.lay_janela),R.layout.tela_principal_sobre);
                    PrincipalSobre();
                    break;
                case 4:
                    util.trocaJanela(findViewById(R.id.lay_janela),R.layout.tela_principal_conta);
            }
        }
    });
}
```

### Imagen 32

```
    PrincipalConta();
    break;
default:
    util.trocaJanela(findViewById(R.id.lay_janela),R.layout.tela_principal_home);
    PrincipalHome();
    break;
}
}
@Override
public void onTabUnselected(TabLayout.Tab tab) {

}

@Override
public void onTabReselected(TabLayout.Tab tab) {
    if(tab.getPosition() == 0){
        util.trocaJanela(findViewById(R.id.lay_janela),R.layout.tela_principal_home);
        PrincipalHome();
    }else if(tab.getPosition() == 1){
        util.trocaJanela(findViewById(R.id.lay_janela),R.layout.tela_principal_menu);
        PrincipalMenu();
    }else if(tab.getPosition() == 2){
        util.trocaJanela(findViewById(R.id.lay_janela),R.layout.tela_principal_comparar);
        PrincipalComparar();
    }else if(tab.getPosition() == 3){
        util.trocaJanela(findViewById(R.id.lay_janela),R.layout.tela_principal_sobre);
        PrincipalSobre();
    }else if(tab.getPosition() == 4){
        util.trocaJanela(findViewById(R.id.lay_janela),R.layout.tela_principal_conta);
        PrincipalConta();
    }
}
```

### Imagen 33

```
});  
PrincipalHome();  
TelaPesquisar();  
}
```

No início da TelaPrincipal, é cancelado conexões de Internet que estejam baixando imagens ou coletando informações (isto cabe apenas as requisições que iriam tentar utilizar objetos já deletados). Assim, é criada uma janela de telas, que fica abaixo das abas azuis, e lá é exibido as telas de conta, comparar, home, menu, sobre e suas subtelhas (Exceto telas cheias, como a tela de Deletar Conta, que faz parte da sessão Sobre). Após a criação de uma janela e ser adicionado a esta a tela de Home, é atribuído a cada Aba a sua sessão, que troca de janela e executa sua função quando é selecionada. Ao ser novamente selecionada a mesma aba, ela recarrega a mesma aba, se não, ela troca para a janela e a função da sessão clicada.

Após tudo isto, a TelaPrincipal executa a função de TelaPesquisar, que toma conta de tornar a barra de busca e a lupa funcionais para buscar produtos, e também toma conta de carregar a função de PrincipalHome, que cuida das funções e execuções da tela de Home.

### PrincipalHome() – Android Studio

### Imagen 34

```
public void PrincipalHome(){  
    bckComando = "";  
    conector.cancelarConexoes(0);  
    Response.Listener<String> resposta = response -> {  
        if (response.toString().startsWith("sqlError")){  
            alerta.Popup("Erro:" + response.toString(),true);  
        }else {  
            util.listaJanela(response,  
                findViewById(R.id.scrLay_Novos_Produtos), 1, v -> {  
                    ImageView modeloImagem = findViewById(v.getId());  
                    String modeloTexto = modeloImagem.getContentDescription().toString();  
                    bckComando = "home";  
                    util.produtoUnico(modeloTexto,findViewById(R.id.lay_janela),conector);  
                });  
        };  
        conector.pesquisar(resposta,null,"novos");  
  
        Response.Listener<String> resposta_2 = response -> {  
            if (response.toString().startsWith("sqlError")){  
                alerta.Popup("Erro:" + response.toString(),true);  
            }else {  
                util.listaJanela(response,  
                    findViewById(R.id.scrLay_Mais_Populares), 1, v -> {  
                        ImageView modeloImagem = findViewById(v.getId());  
                        String modeloTexto = modeloImagem.getContentDescription().toString();  
                        bckComando = "home";  
                        util.produtoUnico(modeloTexto,findViewById(R.id.lay_janela),conector);  
                );  
            };  
        };  
    };  
}
```

### Imagen 35

```
conector.pesquisar(resposta_2,null, "populares");
```

```
    findViewById(R.id.imv_Banner_Home).setOnClickListener(v -> {
        PrincipalMontar();
    });
}
```

Na tela de Home, é cancelado qualquer download sendo realizado pelo aplicativo e dado o comando de fechar o aplicativo ao apertar o Botão de Voltar. E então, é criado duas listas de produtos, a primeira contendo novos celulares, tablets e processadores, listados na ordem de mais novos de data de lançamento para mais antigos. Na segunda lista, é mostrado todos os produtos, na ordem de mais vistos até menos vistos (com base nas visualizações dos usuários do aplicativo).

Para preencher as listas, é utilizado o pesquisar() e o listaJanela(), sendo o primeiro da classe WebConn e a segunda da classe Utilidades. Assim, o pesquisar() carrega uma resposta com os produtos de cada tópico organizados em JSON, assim, o listaJanela() organiza este JSON em formato de lista para cada tópico.

Ao algum produto ser pressionado, é carregado a sua tela de produto, e também é mudada a função do botão Voltar para retornar a tela de Home (Depois de entrar em algum produto).

## PrincipalMenu() – AndroidStudio

### Imagen 36

```
public void PrincipalMenu(){
    conector.cancelarConexoes(0);
    ImageView.OnClickListener menuSelect = v -> {
        bckComando = "menu";
        String texto = "";
        switch (v.getId()){
            case R.id.img_xml_Celulares:
                texto = "d_celulares";
                break;
            case R.id.img_xml_Tablets:
                texto = "d_tablets";
                break;
            case R.id.img_xml_Notebooks:
                texto = "d_notebooks";
                break;
            case R.id.img_xml_Pacotes:
                texto = "d_desktops";
                break;
            default:
                texto = "d_celulares";
                break;
        }
        Response.Listener resposta = response -> {
            if (response.toString().startsWith("sqlError")){
                alerta.Popup("Erro:" + response.toString(),true);
            }else {
                util.listaJanela(response, findViewById(R.id.lay_janela), 0, v1 -> {
                    ImageView modelolImagen = findViewById(v1.getId());
                    String modeloTexto = modelolImagen.getContentDescription().toString();
                    util.produtoUnico(modeloTexto,findViewById(R.id.lay_janela),conector);
                });
            }
        };
        conector.pesquisar(resposta,null,texto);
    };
}
```

### Imagen 37

```
findViewById(R.id.img_xml_Celulares).setOnClickListener(menuSelect);
findViewById(R.id.img_xml_Tablets).setOnClickListener(menuSelect);
findViewById(R.id.img_xml_Notebooks).setOnClickListener(menuSelect);
findViewById(R.id.img_xml_Pacotes).setOnClickListener(menuSelect);
findViewById(R.id.img_xml_Favoritos).setOnClickListener(v -> {
    Response.Listener resposta = response -> {
        if (response.toString().startsWith("sqlError")){
            alerta.Popup("Erro:" + response.toString(),true);
        }else {
            util.listaJanela(response, findViewById(R.id.lay_janela), 0, v1 -> {
                ImageView modelolImagen = findViewById(v1.getId());
                String modeloTexto = modelolImagen.getContentDescription().toString();
                util.produtoUnico(modeloTexto,findViewById(R.id.lay_janela),conector);
            });
        }
    };
    Map<String, String> dados = new HashMap<>();
    dados.put("comando","verFavoritos");
    dados.put("email",sharedPref.getString("SAVED_EMAIL",""));
    conector.conectarPOST(resposta, dados);
});
findViewById(R.id.img_xml_Setups).setOnClickListener(v -> {
    conector.cancelarConexoes(0);
    Response.Listener respostaX = response -> {
        if (response.toString().startsWith("sqlError")){
            alerta.Popup("Erro:" + response.toString(),true);
        }
    };
});
```

## Imagen 38

```
 }else {
    bckComando = "menu";
    util.listaJanela(response, findViewById(R.id.lay_janela), 2, v1 -> {
        ImageView modeloImagem = findViewById(v1.getId());
        String modeloTexto = modeloImagem.getContentDescription().toString();
        util.trocaJanela(findViewById(R.id.lay_janela),R.layout.tela_produto);
        Response.Listener resposta2 = response2 -> {
            if (response2.toString().startsWith("sqlError")){
                alerta.Dialogo_Simples("Erro de Servidor");
            } else {
                try {
                    JSONObject texto_json = new JSONArray(
                        response2.toString()).getJSONObject(0);
                    TextView txv_modeloTexto = findViewById(R.id.txv_TituloModelo);
                    txv_modeloTexto.setText(texto_json.getString("Modelo"));
                    TextView anoTexto = findViewById(R.id.txv_Ano);
                    anoTexto.setText(texto_json.getString("Ano"));
                    ConstraintLayout lay_dados = findViewById(R.id.lay_Especificacao);

                    findViewById(R.id.img_favoritar).setVisibility(View.GONE);

                    conector.baixarImagem("Setup",
                        findViewById(R.id.img_produtoUnico));

                    //CRIAR KEYS E VALUES
                    JSONArray iter = new JSONArray(response2.toString());
                    int cont = 1;
                    int margem_altura = util.DPparaPixel(8);

                    while (cont < iter.length()) {
                        JSONObject linha = iter.getJSONObject(cont);
                        ConstraintSet lay_listaset = new ConstraintSet();
                        int altura;

```

## Imagen 39

```
                        if(linha.has("dado") || !linha.isNull("dado")){
                            altura = (
                                linha.getString("dado").length()
                                >= 25) ? 64 : 48;
                        }else {
                            altura = 48;
                        }

                        //KEY
                        TextView produtoKey = new TextView(this);
                        produtoKey.setText(linha.getString("coluna") + ':');
                        produtoKey.setHeight(util.DPparaPixel(altura));
                        produtoKey.setWidth(util.DPparaPixel(200));
                        produtoKey.setTextColor(getColor(R.color.tema));
                        produtoKey.setId(View.generateViewId());
                        util.bordaTXV(produtoKey);
                        produtoKey.setTextAlignment(View.TEXT_ALIGNMENT_CENTER);
                        produtoKey.setGravity(Gravity.CENTER);
                        lay_dados.addView(produtoKey);
                    //VALUE
                    TextView produtoValue = new TextView(this);
                    if(linha.has("dado") || !linha.isNull("dado")){
                        produtoValue.setText(linha.getString("dado"));
                    }else{
                        produtoValue.setText("--");
                    }
                    produtoValue.setHeight(util.DPparaPixel(altura));
                    produtoValue.setWidth(util.DPparaPixel(200));
                    produtoValue.setTextColor(getColor(R.color.tema));
                    produtoValue.setId(View.generateViewId());
                    util.bordaTXV(produtoValue);
                    produtoValue.setTextAlignment(View.TEXT_ALIGNMENT_CENTER);
                    produtoValue.setGravity(Gravity.CENTER);
                    lay_dados.addView(produtoValue);

```

## Imagen 40

```
lay_listaSet.clone(lay_dados);
//KEYS

lay_listaSet.connect(produtoKey.getId(), ConstraintSet.TOP,
    lay_dados.getId(), ConstraintSet.TOP, margem_altura);
lay_listaSet.connect(produtoKey.getId(), ConstraintSet.LEFT,
    lay_dados.getId(), ConstraintSet.LEFT,
    util.DPparaPixel(0));
//VALUES
lay_listaSet.connect(produtoValue.getId(), ConstraintSet.TOP,
    lay_dados.getId(), ConstraintSet.TOP, margem_altura);
lay_listaSet.connect(produtoValue.getId(), ConstraintSet.LEFT,
    lay_dados.getId(), ConstraintSet.LEFT,
    util.DPparaPixel(200));
//
lay_listaSet.applyTo(lay_dados);
margem_altura += util.DPparaPixel(altura);

cont++;
}

} catch (JSONException e) {
    alerta.Popup(e.getMessage(), false);
}
}
};

conector.verSetupUnico(resposta2,null,modeloTexto);
});
}
};

conector.verSetup(respostaX,null,sharedPref.getString("SAVED_EMAIL",""));
});
```

## Imagen 41

```
findViewByld(R.id.img_xml_Computadores).setOnClickListener(v -> {
    bckComando = "menu";
    util.trocaJanela(findViewByld(R.id.lay_janela),R.layout.tela_principal_menu_computadores);
    PrincipalComputadores();
});
```

Ao selecionar a aba Menu, a tela de Menu é exibida, e todos os downloads parados. Assim, é atribuída à cada imagem a função de carregar a lista de produtos de cada tipo, exceto às imagens de Computadores, Favoritos e de Setups. Nesta função de carregar uma lista de produtos, é exibida uma lista de dois produtos por linha onde cada produto leva a tela de suas especificações, a de TelaProduto. E também é definido, ao clicar em uma imagem, que o botão de Voltar carregará a tela de Menu.

Além disso, a imagem de Favoritos exibe os produtos que foram selecionados como favoritos pelo usuário, e a imagem de Setups mostrará os Setups montados pelos usuários. Porém, se não houver um login feito no aplicativo, aparecerá “Sessão Vazia”. Na imagem de Computadores, e carregada a Tela de MenuSelect (Menu de peças de computador), é executado a função de PrincipalComputadores() e também, como os outros, o botão de Voltar carrega a tela de Menu.

## PrincipalComparar() e PrincipalCompararSelect() – Android Studio

### Imagen 42

```
public void PrincipalComparar(){
    bckComando = "";
    conector.cancelarConexoes(0);
    findViewById(R.id.btn_Comparar).setOnClickListener(v -> PrincipalCompararSelect());
}

public void PrincipalCompararSelect(){
    bckComando = "comparar";
    conector.cancelarConexoes(0);
    Spinner spinCategorias = findViewById(R.id.spin_categorias);
    String categoria = spinCategorias.getSelectedItem().toString();
    String tabela = "";

    switch (categoria){
        case "Celulares":
            tabela = "d_celulares";
            break;
        case "Tablets":
            tabela = "d_tablets";
            break;
        case "Gabinetes":
            tabela = "p_gabinete";
            break;
        case "Memórias de Armazenamento":
            tabela = "p_memoria_armaz";
            break;
        case "Placas-Mãe":
            tabela = "p_placa_mae";
            break;
        case "Processador":
            tabela = "p_processador";
            break;
        case "Memórias RAM":
            tabela = "p_memoria_ram";
            break;
    }
}
```

## Imagen 43

```
case "Placas de vídeo":
    tabela = "p_placa_de_video";
    break;
case "Notebooks":
    tabela = "d_notebooks";
    break;
default:
    tabela = "d_celulares";
    break;
}

util.trocaJanela(findViewById(R.id.lay_janela),R.layout.tela_principal_comparar_select);

Spinner spinner = findViewById(R.id.spin_produtos);
String[] modelosCompararPOST = {"","",""};
```

---

```
String finalTabela = tabela;
Button btnAtualizar = findViewById(R.id.btn_Plus);
```

```
View.OnClickListener removerColuna = v -> {
    int x = Character.getNumericValue(v.getContentDescription().charAt(7));
    if(x == 1){
        ArrayList<String> arrayList = new ArrayList<>();
        for (int i = 0; i < spinner.getCount(); i++) {
            arrayList.add(spinner.getItemAtPosition(i).toString());
        }
    }
}
```

## Imagen 44

```
arrayList.add(modelosCompararPOST[0]);
ArrayAdapter<String> arrayAdapter = new ArrayAdapter<>(getApplicationContext(),
    R.layout.spinner_item, arrayList);
arrayAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
spinner.setAdapter(arrayAdapter);

modelosCompararPOST[0] = "";
util.compararProdutos(modelosCompararPOST, finalTabela,
    findViewById(R.id.lay_tabela), conector);
}else if(x == 2){
    ArrayList<String> arrayList = new ArrayList<>();
    for (int i = 0; i < spinner.getCount(); i++) {
        arrayList.add(spinner.getItemAtPosition(i).toString());
    }
    arrayList.add(modelosCompararPOST[1]);
    ArrayAdapter<String> arrayAdapter = new ArrayAdapter<>(getApplicationContext(),
        R.layout.spinner_item, arrayList);
    arrayAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    spinner.setAdapter(arrayAdapter);

    modelosCompararPOST[1] = "";
    util.compararProdutos(modelosCompararPOST, finalTabela,
        findViewById(R.id.lay_tabela), conector);
}else{
    ArrayList<String> arrayList = new ArrayList<>();
    for (int i = 0; i < spinner.getCount(); i++) {
        arrayList.add(spinner.getItemAtPosition(i).toString());
    }
}
```

## Imagen 45

```
arrayList.add(modelosCompararPOST[2]);
ArrayAdapter<String> arrayAdapter = new ArrayAdapter<>(getApplicationContext(),
    R.layout.spinner_item, arrayList);
arrayAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
spinner.setAdapter(arrayAdapter);

modelosCompararPOST[2] = "";
util.compararProdutos(modelosCompararPOST, finalTabela,
    findViewById(R.id.lay_tabela), conector);
}

if(modelosCompararPOST[0].equals("")){
    findViewById(R.id.img_remover1).setVisibility(View.INVISIBLE);
}if(modelosCompararPOST[1].equals("")){
    findViewById(R.id.img_remover2).setVisibility(View.INVISIBLE);
}if(modelosCompararPOST[2].equals("")){
    findViewById(R.id.img_remover3).setVisibility(View.INVISIBLE);
}
};

findViewById(R.id.img_remover1).setOnClickListener(removerColuna);
findViewById(R.id.img_remover1).setVisibility(View.INVISIBLE);
findViewById(R.id.img_remover2).setOnClickListener(removerColuna);
findViewById(R.id.img_remover2).setVisibility(View.INVISIBLE);
findViewById(R.id.img_remover3).setOnClickListener(removerColuna);
findViewById(R.id.img_remover3).setVisibility(View.INVISIBLE);

btnAtualizar.setOnClickListener(v -> {
    if(spinner.getSelectedItemPosition()
        spinner.getSelectedItemPosition().toString().equals("<-Selecione->")){
            alerta.Popup("Selecione um produto antes",true);
        }else {
            if (modelosCompararPOST[0].toString().length() <= 2) {
```

## Imagen 46

```
modelosCompararPOST[0] = spinner.getSelectedItem().toString();
findViewById(R.id.img_remover1).setVisibility(View.VISIBLE);
} else if (modelosCompararPOST[1].toString().length() <= 2) {
    modelosCompararPOST[1] = spinner.getSelectedItem().toString();
    findViewById(R.id.img_remover2).setVisibility(View.VISIBLE);
    findViewById(R.id.img_remover3).setVisibility(View.INVISIBLE);
} else if (modelosCompararPOST[2].toString().length() <= 2) {
    modelosCompararPOST[2] = spinner.getSelectedItem().toString();
    findViewById(R.id.img_remover3).setVisibility(View.VISIBLE);
} else {
    alerta.Popup("Retire antes algum modelo (Máximo de 3 modelos para comparar)", false);
    return;
}
ArrayList<String> arrayList = new ArrayList<>();
for (int i = 0; i < spinner.getCount(); i++) {
    if (i == spinner.getSelectedItemPosition()) {
        continue;
    }
    arrayList.add(spinner.getItemAtPosition(i).toString());
}
ArrayAdapter<String> arrayAdapter = new ArrayAdapter<>(getApplicationContext(),
    R.layout.spinner_item, arrayList);
arrayAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);

spinner.setAdapter(arrayAdapter);
util.compararProdutos(modelosCompararPOST, finalTabela,
    findViewById(R.id.lay_tabela), conector);
};

Response.Listener resposta = response -> {
```

## Imagen 47

```
Response.Listener resposta = response -> {
    try{
        if(response.toString().length() <= 5){
            alerta.Popup("Sem produtos para comparar",true);
            ArrayList<String> arrayList = new ArrayList<>();
            arrayList.add("Sem produtos");
            ArrayAdapter<String> arrayAdapter = new ArrayAdapter<>(this,
                R.layout.spinner_item, arrayList);
            arrayAdapter.setDropDownViewResource(
                android.R.layout.simple_spinner_dropdown_item);
            spinner.setAdapter(arrayAdapter);
            btnAtualizar.setOnClickListener(v ->
                alerta.Popup("Sem produtos para buscar",true));
        }else {
            JSONArray lista = new JSONArray(response.toString());
            ArrayList<String> arrayList = new ArrayList<>();
            arrayList.add("<-Selecione->");
            for(int i = 0;i < lista.length();i++){
                arrayList.add(lista.getJSONObject(i).getString("modelo"));
            }
            ArrayAdapter<String> arrayAdapter = new ArrayAdapter<>(this,
                R.layout.spinner_item, arrayList);
            arrayAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
            spinner.setAdapter(arrayAdapter);
        }
    } catch(JSONException e){
        alerta.Popup(e.getMessage(),false);
    }
};
conector.pesquisar(resposta,null,tabela);
}
```

Na tela de comparar, é exibido opções de escolha de tipo de produto a ser comparado, como celulares, tablets, peças e outros, e adicionado ao botão o evento PrincipalCompararSelect(), onde tratará de organizar, exibir e escolher os produtos a serem selecionados.

Na função PrincipalCompararSelect(), é carregado todos os produtos do tipo selecionado, assim são exibidos em uma lista branca em cima, e ao usuário apertar em adicionar o produto, se tiver sido selecionado algum, será exibido especificações em uma coluna e os seus valores em outra. Assim, pode-se ser comparado até três produtos simultaneamente, e então pode ser removido por uma imagem de Excluir acima de cada coluna. Nesta tela, o botão de Voltar tem função de retornar a primeira tela de Comparar, a que exibe os tipos de produto a serem selecionados.

### **3 CONCLUSÃO**

Conclui-se que o aplicativo Pega Info contribui significativamente para auxiliar as pessoas na escolha do eletrônico que melhor as atenda. Ajudando-as a montar o computador do jeito que acharem melhor, achando um notebook ou tablet que caiba no bolso, ou escolhendo o celular que mais se adeque a sua necessidade, o Pega Info é ideal para ajudar qualquer um a fazer a melhor escolha numa relação custo/benefício.

Além de poder desenvolver um software que vem a ajudar as pessoas, a produção do aplicativo também foi de grande contribuição para nós, exercitando não só a cooperação em grupo, como também nos ensinando ainda mais justamente sobre a área em que o app está situado, a área tecnológica, nos agregando ainda mais conhecimento sobre componentes eletrônicos e desenvolvimento de projetos.

## REFERÊNCIAS BIBLIOGRÁFICAS

ARRUDA, Karina. Na pandemia, cresce busca por aparelhos eletrônicos e demanda aumenta em 20%. **Gazeta Digital**, 15, dez. de 2020. Disponível em: <<https://www.gazetadigital.com.br/editorias/cidades/na-pandemia-cresce-busca-por-aparelhos-eletronicos-e-demanda-aumenta-em-20/638676>>. Acesso em: 5, mar. de 2021.

LEONARDO, André. O que é um processador? **Tecnoblog**, 20, dez. de 2020. Disponível em: <<https://tecnoblog.net/384499/o-que-e-um-processador>>. Acesso em: 24, nov. de 2021.

SALLES, Filipe. O que é SSD e como ele funciona? Saiba tudo sobre esse componente. **Zoom**, São Paulo, 02, ago. de 2019. Disponível em: <<https://www.zoom.com.br/notebook/deumzoom/o-que-e-ssd>>. Acesso em: 24, nov. de 2021.

BRITO, Felipe. O que é HD? **Zoom**, 02, ago. de 2019. Disponível em: <<https://www.zoom.com.br/pc-computador/deumzoom/o-que-e-um-hd>>. Acesso em: 24, nov. de 2021.

KARASINSKI, Vinicius. O que é uma fonte de energia e como ela é por dentro. **Zoom**, 23, mar. de 2012. Disponível em: <<https://www.tecmundo.com.br/fonte/20987-o-que-e-uma-fonte-de-energia-e-como-ela-e-por-dentro.htm>>. Acesso em: 24, nov. de 2021.

O que é a placa mãe e como funciona? **Kazuk** 11, dez. 2020. Disponível em: <<https://kazuk.com.br/blog/o-que-e-a-placa-mae/>>. Acesso em: 24, nov. de 2021.

MONTEIRO, Julio. O que é memória RAM e qual é sua função? **CanalTech**, 27, fev. de 2012. Disponível em: <<https://www.techtudo.com.br/noticias/2012/02/o-que-e-memoria-ram-e-qual-sua-funcao.ghtml>>. Acesso em: 24, nov. de 2021.