

CATÁLOGO DE PRODUTOS MULTIPLATAFORMA COM FLUTTER

Hugo Biazibetti Reis, Fabrício Gustavo Henrique

Faculdade de Tecnologia de Ribeirão Preto (FATEC)

Ribeirão Preto, SP – Brasil

hugo.reis3@fatec.sp.gov.br,
fabricao.henrique@fatec.sp.gov.br

Resumo. Após o início da pandemia, o custo da mídia impressa aumentou expressivamente, e na tentativa de minimizar esses custos, foi projetada a ideia de um catálogo virtual que permitiria unir a praticidade de obter a informação de maneira rápida e eficiente, com a economia e sustentabilidade do não uso do papel. Assim, o objetivo deste trabalho foi desenvolver um catálogo virtual, na forma de um aplicativo, com intuito de facilitar o acesso e busca de produtos que uma empresa possa comercializar. Os primeiros resultados mostram que a escolha do framework Flutter foi apropriada já que permitiu gerar um aplicativo compatível com três plataformas utilizando apenas um código. Inicialmente foram projetadas cinco telas do aplicativo, as quais se mostram satisfatórias em relação à resposta das consultas e atualização em tempo real das informações. O código encontra-se disponível em https://github.com/hugobiareis/catalogo_de_produtos.

Abstract. After the beginning of the pandemic, the cost of print media increased significantly, and to minimize these costs, the idea of a virtual catalog was designed that would allow to unite the practicality of obtaining information quickly and efficiently, with the economy and sustainability of not using paper. Thus, the objective of this work was to develop a virtual catalog, in the form of an application, to facilitate access and search for products that a company can sell. The first results show that the choice of the Flutter framework was appropriate as it allowed generating an application compatible with three platforms using only one code. Initially, five screens of the application were designed, which are satisfactory in relation to response to queries and real-time updating of information. The code is available at https://github.com/hugobiareis/catalogo_de_produtos.

1. Introdução

Atualmente os dispositivos moveis, celulares e tablets, tem alcançado cada dia mais as famílias de todas as classes sociais. Esta realidade foi causada pelos baixos custos alcançados na fabricação dos dispositivos provocados pela grande concorrência dos fabricantes. Esta grande concorrência trouxe ao mercado uma grande variedade de

tecnologias, muitas vezes incompatíveis entre si, como por exemplo os sistemas operacionais Android e iOS. Cada grande fabricante possui as chamadas lojas de aplicativos, que permite que criadores independentes disponibilizem os mais variados tipos de conteúdo para os usuários: educacional, financeiro, entretenimento, comunicação, entre outros. Com a necessidade de criar conteúdo para as diferentes tecnologias de forma otimizada surgiram os frameworks, dentre eles o *Flutter*, que permite com uma só programação, criar aplicativos voltados para Web, Android e iOS.

Este trabalho tem como objetivo desenvolver um catálogo virtual na forma de aplicativo, multiplataforma, utilizando o framework *Flutter*, através do *VS Code*. A ideia de um catálogo virtual é promissora, já que após o início da pandemia o custo da mídia impressa aumentou expressivamente, e na tentativa de minimizar esses custos, está ferramenta virtual permitiria unir a praticidade de obter a informação de maneira rápida e eficiente, com a economia e sustentabilidade do não uso do papel. Além disso, a criação de um catálogo virtual nos permitiria atingir um maior número de pessoas utilizando menos recursos, como por exemplo com a impressão dos catálogos e distribuição deles, seja presencialmente ou enviando pelos Correios ou similares.

2. Metodologia

2.1. Elicitação de requisitos

Os requisitos de *software* são as características, limitações, exigências e funcionalidades, que o cliente espera dele. Uma das técnicas utilizadas para realizar o levantamento dos requisitos é fazer uma entrevista estruturada com os usuários do sistema. Assim, para este trabalho foi realizada uma entrevista com um proprietário de uma fábrica de ferramentas voltadas para manutenção de motores para levantar os principais requisitos do *software*.

2.1.1 Requisitos funcionais

RF001 – O aplicativo deve exibir os itens marcados como lançamento na tela inicial, ordenados dos mais recentes para os mais antigos.

RF002 – O aplicativo deve permitir pesquisar os itens pela descrição resumida, descrição detalhada, código, classificação ou família.

RF003 – O aplicativo deve permitir o usuário visitante criar um cadastro solicitando nome, e-mail e telefone celular.

RF004 – O aplicativo deve permitir o visitante se identificar através de login.

RF005 – O aplicativo deve permitir que o usuário logado adicione itens de interesse em uma lista.

RF006 – O aplicativo deve ter um botão para enviar a lista de produtos junto com os dados do cliente para um vendedor elaborar uma proposta comercial.

RF007 – O aplicativo deve fornecer opções para deslogar e alterar dados cadastrais.

2.1.2 Requisitos não funcionais

RNF001 – Deve ser multiplataforma.

RNF002 – Banco de dados deve armazenar fotos.

RNF003 – Banco de dados em nuvem.

2.1.3 Restrições

- Usuário tem que estar logado para solicitar cotação.
- Banco de dados em nuvem, necessário acesso com a internet.

2.2. Modelagem

A figura 1 mostra a estrutura de utilização do catálogo mediante um Diagrama de uso, em que o visitante poderá visualizar e pesquisar produtos, não sendo obrigatório o cadastramento e identificação. Porém será necessário se cadastrar e efetuar login caso queira solicitar um orçamento dos itens diretamente pelo aplicativo. Existe também a opção de efetuar login como administrador para efetuar a manutenção dos itens.

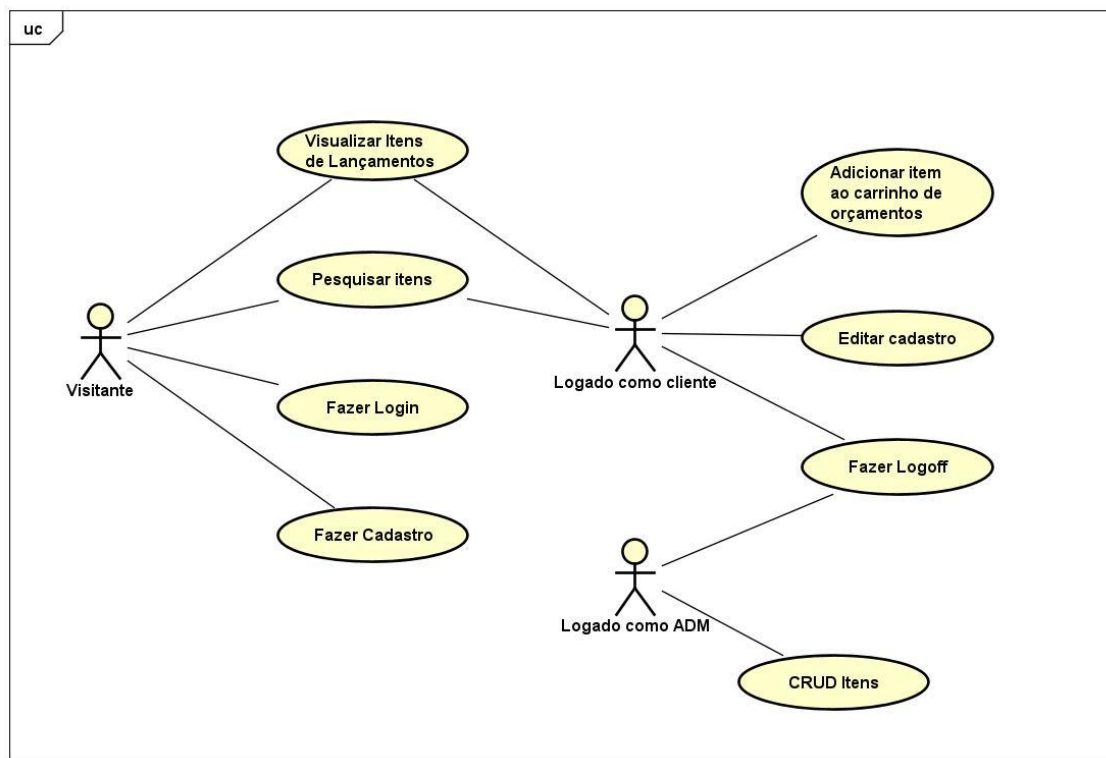


Figura 1. Diagrama de uso

Fonte: (Autoria própria, 2022)

2.3. Framework e banco de dados

O framework *Flutter*, que nas suas primeiras versões era conhecida como “Sky”, foi desenvolvido na forma de um kit de desenvolvimento de interface de usuário. É de código aberto e foi desenvolvido pela Google. Em 2015 foi apresentado na cúpula de desenvolvedores *Dart*, pois é baseado nesta linguagem. O *Flutter* facilita o

desenvolvimento permitindo criar aplicativos nativos Android, Web e iOS.

A arquitetura do *framework* inclui a Biblioteca Foundation, a *Flutter Engine*, Linguagem de Programação *Dart* e Implementações de Widgets para Android (Material Data) e iOS (Cupertino).

O banco de dados que será utilizado é o *Firestore* da *Firebase*, que é uma plataforma para criar aplicativos web e móveis. Era independente de 2011 até 2014 quando foi adquirida pela Google. O *Firestore* é um banco *NoSQL* flexível e escalonável focado em aplicativos para web e mobile. Conta com atualizações em tempo real o que significa que qualquer atualização dos dados na base faz com que todas as aplicações atualizem imediatamente.

3. Desenvolvimento

A proposta para o desenvolvimento do aplicativo será que ele seja amigável e fácil de utilizar. Para isso foi utilizado o pacote *PageView* que permite navegar entre telas rolando lateralmente, este pacote exibe uma barra inferior com ícones que são os atalhos para as telas desejadas, caso o usuário prefira ir diretamente ao invés de navegar rolando lateralmente.

Para o carregamento do aplicativo será utilizado o pacote *SplashScreen* que aguarda a confirmação com o banco de dados e direciona para a tela de lançamentos. Este pacote permite exibir uma tela enquanto aguarda um tempo determinado ou então uma ação ser concluída. Nesta primeira versão do aplicativo foi determinado um tempo fixo.

Dentre as telas desenvolvidas para serem navegadas no *PageView*, a primeira será a tela de lançamentos. Esta tela exibe os produtos que foram marcados no banco como lançamento, então uma rotina de consulta ao banco trás todos os produtos assim marcados. Para a exibição destes produtos, mais uma vez pensando na facilidade de uso do aplicativo, foi utilizado o pacote *GridView*, que utiliza o resultado da consulta ao banco para gerar uma grid de cartões. Utilizando o pacote *Cards* juntamente com o *GestureDetector* e configurando o gesto *OnTap*, que faz com que ao tocar no *card* os produtos sejam exibidos com uma imagem e um título. Para realizar este detalhamento foi utilizada uma caixa de diálogo evitando com que o usuário saia da página. Ao evitar que o usuário saia da página conseguimos proporcionar uma melhor navegação pois evita uma nova consulta ao banco de dados que exigiria consumo de dados e tempo de espera.

A segunda tela será reservada para consulta de produtos utilizando palavras-chave. Da mesma forma que a primeira tela, os resultados são apresentados em forma de *cards* que permitem visualizar o detalhamento sem precisar navegar para outra página.

Na terceira tela será possível visualizar os itens que o usuário selecionará para formar uma lista e solicitar um orçamento. Essa seleção só será possível caso o usuário tenha feito o *Login* no aplicativo.

A quarta tela será destinada para que o usuário encontre as opções para realizar o login ou cadastramento. Caso o usuário já possua seu cadastro pronto, ele irá precisar de fornecer o nome de usuário e senha. Neste momento utilizamos mais uma facilidade do *Flutter* integrado com o *Firebase*, que é o pacote de Autenticação, em que de forma fácil conseguimos implementar uma consulta de usuário e senha ao banco de forma segura e

rápida, utilizando toda a infraestrutura e segurança do *Firebase*. O caso usuário necessite recuperar sua senha, o *Firebase* também oferece essa facilidade, um pacote de implementação de recuperação de senha.

4. Resultados

Os códigos desenvolvidos no projeto podem ser acessados no seguinte link: https://github.com/hugobiareis/catalogo_de_produtos.

Durante a inicialização do aplicativo será visualizado o logotipo da empresa, uma mensagem de benvinda e uma animação girando em círculo (Figura 2A), configurado no pacote *SplashScreen*. O resultado do uso dos *Cards*, alinhados pelo *GridView* pode ser visualizado na figura 2B.

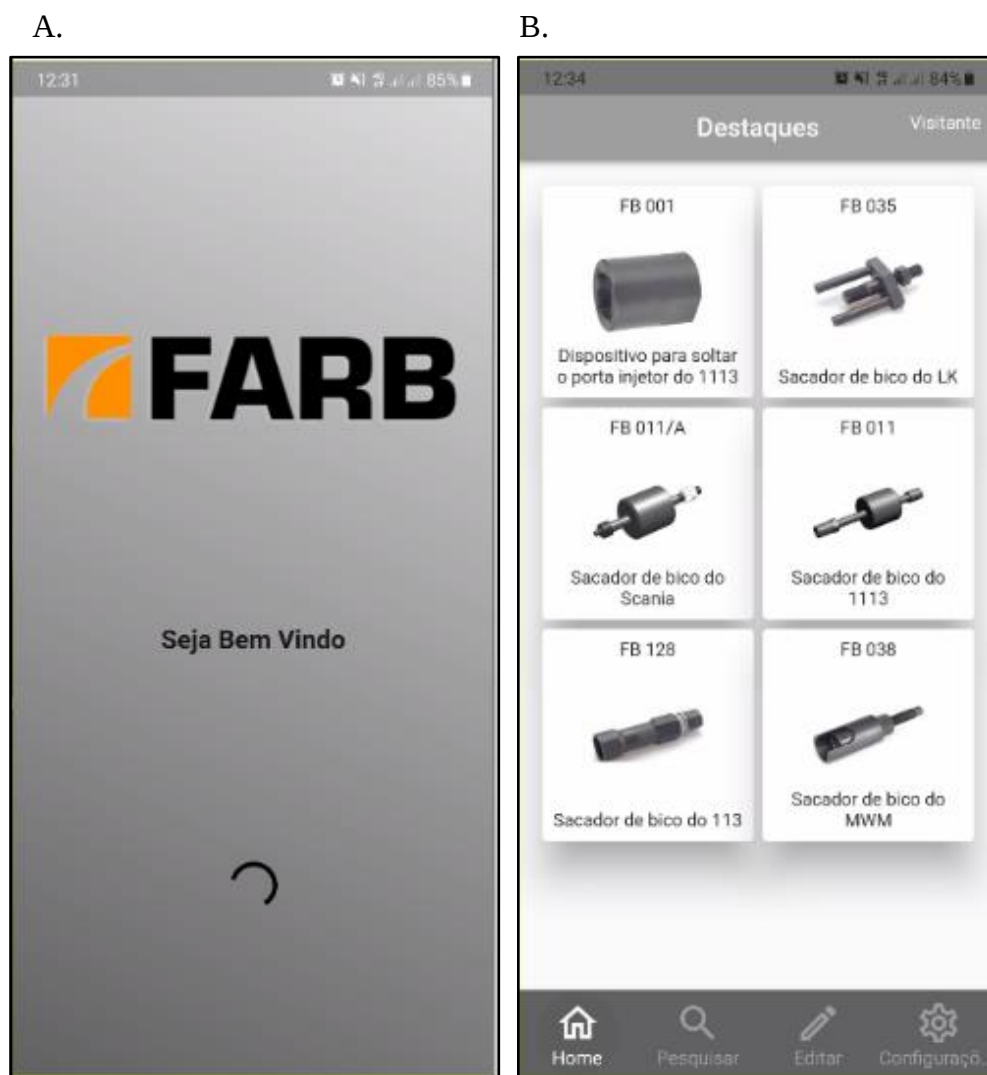


Figura 2. A) Tela inicial e B) Tela de lançamentos

Fonte: (Autoria própria, 2022)

Na figura 3, pode-se observar o menu de navegação do *PageView* e na parte inferior da página, o ícone fica em destaque em relação aos outros ícones, quando a página for selecionada.

Na tela de pesquisa o ícone em destaque foi a lupa (Figura 3A), na tela de editar o ícone em destaque será o lápis (Figura 3B). A tela configurações exibe outras funcionalidades como as notificações e os sons.

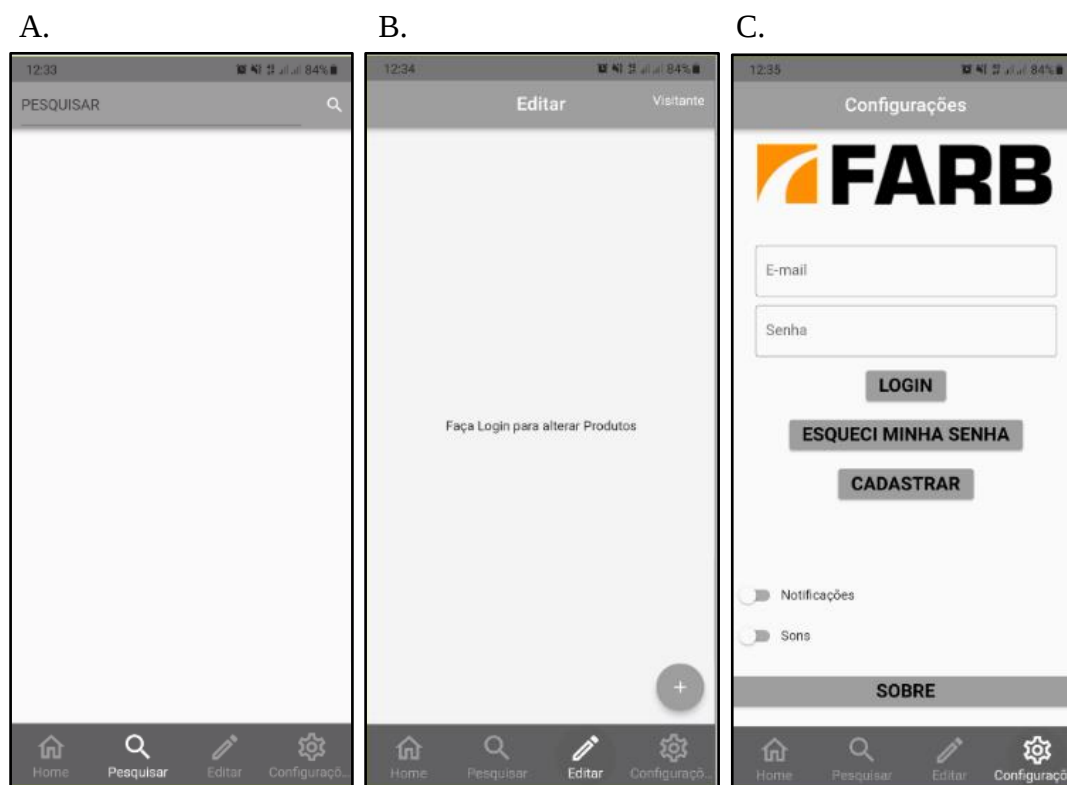


Figura 3. A) Tela de Pesquisa; B) Tela Produtos Seleccionados; C) Tela de Configurações

Fonte: (Autoria própria, 2022)

Quando o usuário está logado com o seu cadastro ou como visitante, o detalhamento é exibido de forma diferente, como pode ser visto na figura 4.

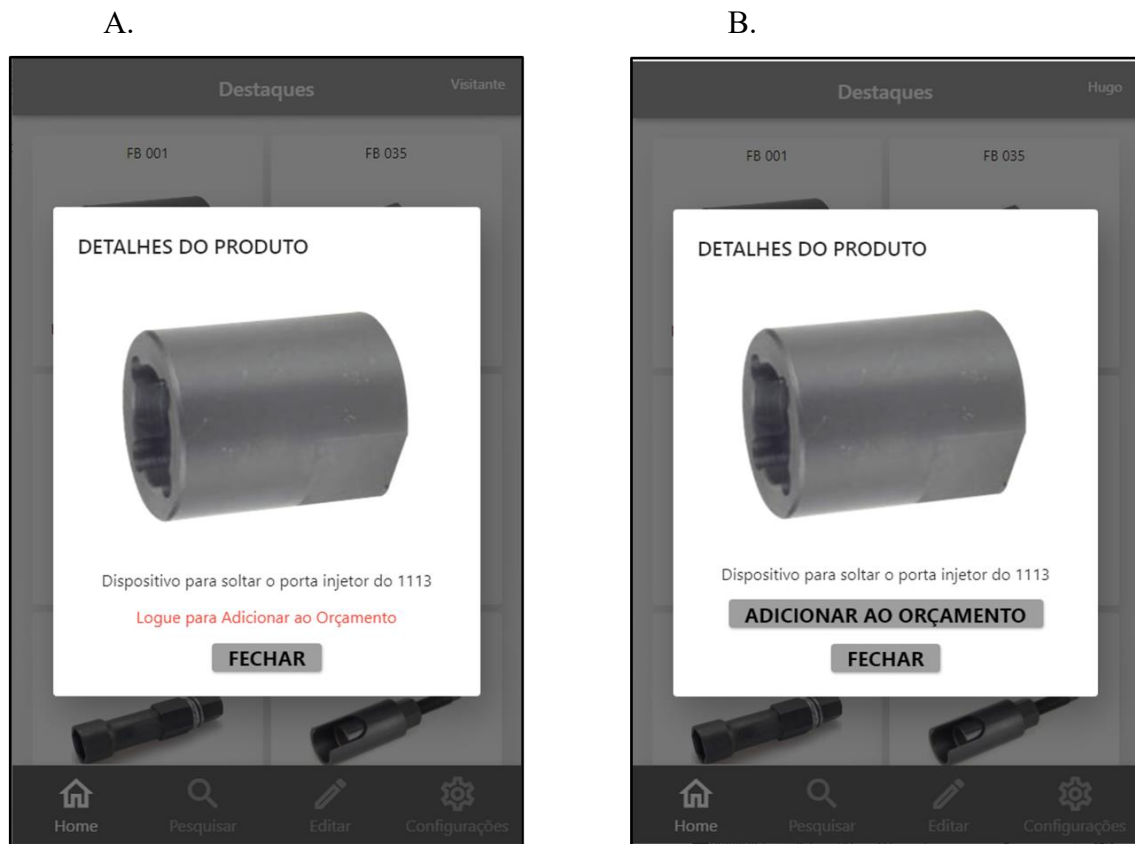


Figura 4. A) Detalhamento visitante B) Detalhamento logado

Fonte: (Autoria própria, 2022)

5. Considerações finais e implementações futuras

O aplicativo foi projetado para ser o mais simples possível, trazendo o detalhamento dos produtos em um *pop-up* ao invés de navegar para uma página dedicada a isto, evitando o carregamento total de uma nova página e, ao retornar, recarregar todas as informações, evitando o excesso de troca de dados e melhorando a experiência em redes de baixa capacidade.

O usuário não precisaria realizar login para ter acesso a todas as informações, então, ele pode escolher entre solicitar um orçamento via aplicativo ou entrar em contato com a empresa através de outros meios.

A escolha do *Flutter* para desenvolvimento mostrou-se uma escolha assertiva pela facilidade de gerar aplicações para três plataformas diferente com um único código e juntamente com o *Firebase* respondeu satisfatoriamente em relação à resposta das consultas e atualização em tempo real das informações.

Os requisitos obtidos junto ao stakeholder foram atendidos satisfatoriamente utilizando toda praticidade e versatilidade das ferramentas adotadas para o desenvolvimento do projeto.

Como trabalho futuro será necessário implementar as notificações do aplicativo, para que o usuário possa receber avisos de novos lançamentos e promoções.

6. Referências

- Flutter. (2022) Disponível em: <<https://docs.flutter.dev>>. Acesso em: 17 de novembro de 2022.
- DART. (2022). Disponível em: <<https://dart.dev/guides>>. Acesso em: 17 de novembro de 2022.
- Firebase. (2022). Disponível em: <<https://firebase.google.com/docs/guides?hl=pt>>. Acesso em: 17 de novembro de 2022
- Astah. (2022). Disponível em: <<https://astah.net/support/>>. Acesso em: 17 de novembro de 2022
- SOMMERVILLE, I. (2011) Software engineering 9th Edition. ISBN-10, v. 137035152
- COMURB. A EXPANSÃO DOS DISPOSITIVOS MÓVEIS E OS IMPACTOS PARA A VIDA NAS CIDADES. (2022) Disponível em: < <https://comurb.com.br/a-expansao-dos-dispositivos-moveis-e-os-impactos-para-a-vida-nas-cidades/> >. Acesso em: 18 de novembro de 2022.
- BUTCHER, I. (2020) Até 2023, 5,3 bilhões de pessoas estarão conectadas à Internet, diz Cisco. Mobile Time Disponível em: < <https://www.mobiletime.com.br/noticias/18/02/2020/ate-2023-53-bilhoes-de-pessoas-estarao-conectadas-a-internet-diz-cisco/> >. Acessado em: 18 de novembro de 2022.