

# DEV OS: CRIAÇÃO DE UMA DISTRIBUIÇÃO LINUX PARA DESENVOLVIMENTO DE SOFTWARE

SCAVASSA, S. M.; VIANA, J. A. A. (orientador)  
e-mail: samuel040103@gmail.com, jose.viana@fatec.sp.gov.br

**Resumo:** Este trabalho visa facilitar a adoção do Linux como ambiente de desenvolvimento principal, oferecendo uma distribuição pré-configurada com ferramentas para desenvolvimento e um ambiente gráfico com bom desempenho. Para isso é preciso conhecimentos técnicos sólidos sobre o sistema operacional, processos de construção e distribuição. O desenvolvimento do sistema proposto foi dividido em fases, incluindo a escolha da base, do ambiente gráfico, dos pacotes e aplicativos, a personalização da distribuição, configuração do sistema, desenvolvimento da aplicação personalizada, testes e correção de bugs, e a distribuição final.

**Palavras-chave:** Ambiente de desenvolvimento, distribuição pré-configurada, ferramentas de desenvolvimento, ambiente gráfico, conhecimentos técnicos, sistema operacional.

**Abstract:** *This work aims to facilitate the adoption of Linux as the main development environment, offering a pre-configured distribution with development tools and a graphical environment with good performance. This requires solid technical knowledge about the operational system, construction processes and distribution. The development of the proposed system was divided into phases, including the choice of the base, the graphical environment, packages and applications, customization of the distribution, system configuration, development of the customized application, testing and bug fixing, and the final distribution.*

**Keywords:** *Development environment, pre-configured distribution, development tools, graphical environment, technical knowledge, operating system.*

## 1. Introdução

Segundo Campos (2006, p. 1) a expressão *software* livre foi “criada pela Free Software Foundation, é o *software* que pode ser usado, copiado, estudado, modificado e redistribuído sem restrição.”. Juntamente com o código, o *software* também carrega uma licença de uso chamada de General Public License (GPL) ou Berkeley Software Distribution (BSD). Esse tipo de *software*, defende o que chamamos de “quatro liberdades”, sendo elas: uso, cópia, modificações e redistribuição. Esse tipo de aplicação, foi amplamente adotado pela comunidade de desenvolvedores que promove uma disseminação do conhecimento e a melhoria das aplicações que liberam seu código fonte. Nesse contexto, o principal *software* conhecido dito como um *software* livre é o sistema operacional GNU/Linux ou apenas Linux.

O sistema operacional Linux, é o *kernel* ou núcleo do sistema desenvolvido por Linus Torvalds em 1991 e que hoje é mantido pela comunidade mundial de desenvolvedores incluindo grandes empresas como a Canonical, Red Hat, IBM além de programadores individuais. O *kernel* Linux é poderoso, atualizado com frequência e flexível, permitindo ao desenvolvedor modificá-lo à vontade. Dessa forma, ele é utilizado para a construção de diversos sistemas para diversos dispositivos como: servidores, *desktops* e celulares.

Para sua utilização em *desktops*, segundo Campos (2006, p. 1) o conjunto do *kernel* Linux, juntamente com um ambiente gráfico, gerenciador de janelas e as aplicações forma uma Distribuição Linux para uso em *desktops*. Atualmente, existem diversas distribuições feitas por diversas empresas cada uma com vantagens e desvantagens, mas dentre as mais famosas vale ressaltar: Debian, Fedora (mantido pela Red Hat), SUSE (mantido pela OpenSUSE) e Ubuntu (mantido pela Canonical).

No contexto das diversas distribuições Linux disponíveis, é importante ressaltar que cada uma delas oferece diferentes conjuntos de recursos e abordagens, o que amplia ainda mais a flexibilidade e a capacidade de personalização do sistema operacional. O Debian, por exemplo, é conhecido por sua estabilidade e foco em *software* livre, enquanto o Fedora, destaca-se por sua ênfase na inovação tecnológica e atualizações frequentes. Já o SUSE, oferece uma combinação de estabilidade e suporte empresarial, e o Ubuntu, busca proporcionar uma experiência amigável ao usuário. Essa variedade de opções permite que os usuários escolham uma distribuição que melhor atenda às suas necessidades específicas de desenvolvimento de *software*. Além disso, a comunidade Linux é altamente ativa e colaborativa, o que resulta em uma vasta quantidade de documentação, fóruns e suporte online disponíveis para auxiliar os desenvolvedores na utilização e personalização do sistema. Assim, ao optar pelo sistema operacional Linux, os desenvolvedores têm acesso a uma comunidade engajada, recursos abrangentes e a possibilidade de escolher uma distribuição que se alinhe com suas preferências e requisitos de desenvolvimento (Kriger, 2023).

Desta forma, o sistema operacional Linux é considerado ideal para o desenvolvimento de *software*, dentre suas principais características que atraem os desenvolvedores estão a alta disponibilidade de ferramentas e IDEs, a ampla gama de recursos disponíveis para o Sistema Operacional (SO) contempla quase todos os disponíveis no mercado, terminal e linha de comando poderoso e flexível, grande número de bibliotecas disponíveis e a flexibilidade e personalização do sistema que permite o ajuste de acordo com a necessidade do usuário (Conheça as vantagens do sistema Linux para programadores e desenvolvedores, 2023).

Porém, muitos usuários optam por outros sistemas devido à complexidade inicial do entendimento do funcionamento do sistema ou pela complexidade de configuração do SO para se adequar a necessidade. Dessa forma, muitos desenvolvedores por receio também optam por outros sistemas como o Windows devido a facilidade de instalação e pela inflexibilidade no quesito ambiente gráfico.

Nesse sentido, a criação de uma distribuição Linux que venha pré-configurada para o desenvolvimento de *software* é uma possível solução para este problema. Esta distribuição, deve conter no mínimo as linguagens populares do momento como: Python, Java, C#, Dart, e Rust, também deve vir com uma IDE pré-instalada, como o Visual Studio Code, e um ambiente gráfico leve, com uma curva de aprendizado baixa e parecido com outros sistemas como o XFCE ou Cinnamon.

Para isso, a base utilizada nesta nova distribuição tem que conter o *Kernel* atual, um grande repositório de pacotes e uma variedade de ambientes gráficos e versões. Com isso, a distribuição que se enquadra nesses requisitos é o Ubuntu. Com a base e o ambiente escolhidos, resta realizar as alterações necessárias como modificar as telas do sistema e os pacotes necessários, além de instalar os recursos que serão oferecidos (UZAYR, 2023).

## 2. Justificativa

Este trabalho pretende oferecer uma solução para que desenvolvedores adotem o sistema operacional Linux como ambiente e desenvolvimento principal. Trazendo diversos recursos pertinentes para o contexto, este trabalho pretende facilitar a migração de outros sistemas. Este também se propõe a ser uma ferramenta útil para os desenvolvedores pois irá contar com diversos recursos de desenvolvimento. Dessa forma, se espera utilizar conhecimentos em sistemas operacionais, programação de aplicações e *software* livre para sua construção. Em resumo, este trabalho pretende desenvolver uma ferramenta para desenvolvedores, utilizar os conhecimentos absorvidos na trajetória do curso e por fim, contribuir com a comunidade Open Source.

## 3. Objetivo

Este trabalho tem por objetivo fazer um levantamento dos requisitos essenciais de uma distribuição para o público desenvolvedor, realizar uma curadoria da distribuição que será utilizada como base, fazer um levantamento de pacotes essenciais e não essenciais para a proposta do sistema, realizar uma curadoria do ambiente gráfico e gerenciador de janelas que serão utilizados, desenvolvimento do sistema para Linux voltado para o público de desenvolvimento de *software*, realizar o desenvolvimento de uma aplicação de instalação de pacotes pertinentes ao contexto de desenvolvimento e distribuir o sistema através de um arquivo ISO.

#### 4. Fundamentação Teórica

A criação de uma distribuição Linux requer uma base sólida de conhecimento técnico, tanto do sistema operacional em si quanto dos processos envolvidos na construção e compartilhamento de um sistema Linux. Além disso, é fundamental ter compreensão dos requisitos técnicos e legais envolvidos no desenvolvimento e divulgação de um novo SO Linux.

Para construir uma distribuição Linux, é necessário ter conhecimento das principais componentes do sistema, como o *kernel*, ferramentas do sistema, bibliotecas e aplicativos de *desktop*, bem como habilidades em linguagens de programação como Shell Scripting e Flutter para o desenvolvimento da aplicação personalizada. Além disso, é importante ter compreensão sobre gerenciamento de pacotes e sistemas de arquivos para a criação de pacotes de *software* e para empacotamento de *softwares* em geral (LOVE, 2010).

Desta forma, com uma bagagem teórica sólida para realizar o desenvolvimento em si, será necessário ferramentas para atingir o objetivo proposto. Nesse sentido, o primeiro requisito é a utilização de um sistema operacional Linux como ambiente principal, para que seja possível realizar testes e realizar o desenvolvimento do aplicativo de instalação. O sistema escolhido como principal para desenvolver foi o Ubuntu 22.04 LTS (UBUNTU, 2022).

Para realizar as alterações na distribuição base, é necessário acessar um sistema com ela instalada. Para isso, um ambiente virtual permite a instalação da distribuição base de modo a não afetar o sistema em uso, além de permitir emulação de hardware que permite o teste do novo sistema em diferentes situações. Assim, para realizar a virtualização foi utilizado os *softwares* VirtualBox 7.0, VMWare Player e Boxes (ORACLE, 2022; VMWARE, 2022; GNOME, 2022).

A fim de desenvolver uma aplicação, é necessário a criação de código fonte que será compilado em uma aplicação. Para realizar a escrita do código foi utilizado a IDE Visual Studio Code juntamente com o *framework* Flutter (MICROSOFT, 2022; GOOGLE, 2022). Para o versionamento do código foi utilizado a ferramenta Git, juntamente com o site GitHub (CHACON; STRAUB, 2014).

Por fim, para realizar a compilação do sistema final em um arquivo ISO próprio para a instalação em outras máquinas, foram utilizadas ferramentas de construção de ISO como o mkisofs e o genisoimage (BLUM; BRESNAHAN, 2021).

## 5. Trabalhos Similares

Segundo Anderson S. André & et al. (2014) em seu trabalho de desenvolvimento de uma distribuição Linux para a terceira idade, foi realizada a divisão das distribuições em três camadas: “Camada Linux, formado pelo núcleo (*kernel*), por um gerenciador de boot, pelos processos de inicialização, pelo Login, pelo Shell e pelo servidor X; Gerenciador de Janelas, responsável pela interação de usuário por meio de interfaces gráficas baseadas em janelas; Aplicações.” (Anderson S. André & et al., 2014). Nesse sentido, o seu trabalho de desenvolvimento pode também ser dividido em três partes. A primeira foi a seleção da base do *kernel*, onde o Ubuntu Server 10.04 LTS foi o escolhido. A segunda, foi a seleção e instalação de um ambiente gráfico onde o vitorioso foi o gerenciador de janelas Fluxbox. E por fim, foi selecionado os pacotes e aplicações a serem adicionados. Após isso, a personalização estava pronta restando apenas a compilação para uma imagem ISO.

Este trabalho, apesar de muito bem elaborado com um passo a passo bem definido, comete alguns erros. Fica mais em evidência, é a escolha do gerenciador de janelas Fluxbox que apesar de seu excelente desempenho é muito simples, oferece poucos recursos e que para a proposta do projeto (uma distribuição voltada para o público idoso) aparenta não se encaixar muito bem.

Outro trabalho similar, desenvolvido por Cauê Aruã Zani de Souza (2009) é a criação de uma distribuição Linux voltada para a educação infantil. Neste trabalho, é exposto a importância da liberdade no aprendizado e como isso pode ser associado ao mundo Open Source. Desta forma, é proposto a criação de um SO que permita o aprendizado sendo de fácil manejo e que disponha de ferramentas que atendam e esta proposta. Para isso, o autor escolhe como base o Arch Linux, uma distribuição muito famosa pela sua leveza, alta disponibilidade de pacotes e principalmente pela sua flexibilidade, juntamente com o ambiente gráfico LXDE, ambiente este extremamente simples e com alto desempenho, e uma seleção de aplicações voltadas para a educação.

Nesta distribuição, a escolha do Arch Linux como base foi assertiva para a proposta escolhida, pois esta permite a instalação do mínimo necessário permitindo ao desenvolvedor inserir pacotes e aplicações necessários. A escolha do gerenciador de janelas LXDE, também foi certa pois este é simples e possui apenas funcionalidades básicas, porém cumpre sua proposta de forma excepcional (possui um ótimo desempenho). Em conclusão, este trabalho desenvolveu uma distribuição que atende os requisitos propostos.

O último trabalho estudado foi desenvolvido por TRES, Crineu & QUEIROZ, Ulisses Muniz (2003) onde os autores desenvolvem uma distribuição personalizada e de uso genérico. Neste, a base escolhida foi o Fedora Linux. Dentre os motivos que levaram esta escolha, é plausível de citação: “método fácil de fazer atualizações, com mínimas modificações em arquivos de configuração; adoção dos *softwares* mais populares e mais conhecidos, livrando o usuário de programas desconhecidos e criando pacotes padrões de *softwares*; promoção de perspectivas globais, com suporte a tantas línguas e localidades geográficas quantas forem possíveis, tornando o produto universal.” (TRES, Crineu & QUEIROZ, Ulisses Muniz, 2003). Após a escolha da base, foi escolhido pelos autores não modificar o gerenciador de tarefas padrão da distribuição, desta forma mantendo a interface GNOME. Por fim, alguns pacotes foram implementados e outros foram removidos, concluindo a customização.

Por fim, é possível concluir que este trabalho atingiu seu objetivo de ser um sistema de uso genérico devido ao que o sistema usado de base (Fedora Linux) já oferecia. A pouca customização também foi um fator importante para isso, pois ao não limitar o sistema a um público específico a distribuição pode atingir um maior número de usuários.

## 6. Metodologia

Este tópico tem como objetivo, demonstrar quais serão os processos metodológicos utilizados para realizar o desenvolvimento abordado, se apoiando principalmente em pesquisas técnicas e científicas, utilizando os métodos e procedimentos amplamente utilizados pela comunidade Open Source.

### 6.1. Pesquisa técnica e científica

Antes de iniciar o desenvolvimento, uma pesquisa técnica e científica foi realizada. Esta pesquisa teve como objetivo, analisar quais são os métodos, ferramentas e procedimentos são reconhecidos e utilizados para o desenvolvimento de *software* e desenvolvimento de uma distribuição Linux.

### 6.2. Desenvolvimento em ambientes virtuais

Para realizar o desenvolvimento do novo sistema, máquinas virtuais foram utilizadas. Sua utilização, permite a criação de um ambiente com recursos computacionais controlados, flexível, volátil, portátil e separado ao sistema utilizado para realizar o desenvolvimento.

### 6.3. Processo de melhoria contínua

Durante o desenvolvimento, foi aplicado um processo de melhoria contínua para o sistema, onde ao avançar em uma etapa do desenvolvimento o ambiente virtual em uso era descartado e um novo era feito, onde somente o que de fato foi concretizado era utilizado.

### 6.4. Laboratórios de testes

Para analisar a qualidade dos produtos desenvolvidos, ambientes de teste separados do ambiente de desenvolvimento foram utilizados, onde os foi analisado o comportamento do sistema e do aplicativo. Dessa forma, possíveis erros ou *bugs* foram encontrados e corrigidos.

## 7. Desenvolvimento

Para desenvolver o sistema proposto, o processo de desenvolvimento foi dividido em 5 fases. A primeira foi a escolha da base e do ambiente gráfico, a segunda a escolha dos pacotes e aplicativos, a terceira foi desenvolvimento da aplicação, a quarta foi a integração do sistema e o aplicativo e por fim, a distribuição do sistema no formato ISO.

Dessa forma, um ambiente virtual foi preparado para realizar a instalação do sistema base, permitindo realizar a portabilidade desse sistema, criar um histórico de versões além de permitir o ajuste de recursos computacionais. Assim, para realizar essa virtualização foi escolhido o *software* VirtualBox em sua versão 7.0.12.

Figura 1: Definições ambiente virtual

<b>Geral</b>	
Nome:	TG
Sistema Operacional:	Ubuntu (64-bit)
Grupos:	TG
<b>Sistema</b>	
Memória Principal:	4096 MB
Processadores:	2
Ordem de Boot:	Óptico, Disco Rígido
Tipo de Chipset:	ICH9
Aceleração:	Paginação Aninhada, Paravirtualização KVM
<b>Tela</b>	
Memória de Vídeo:	16 MB
Controladora Gráfica:	VMSVGA
Servidor de Desktop Remoto:	Desabilitado
Gravação:	Desabilitado
<b>Armazenamento</b>	
Controladora:	IDE
Dispositivo IDE Secundário 0:	[Disco Óptico] Vazio
Controladora:	SATA
Porta SATA 0:	TG.vdi (Normal, 40,00 GB)

Fonte: Autor (2024)

## 7.1. Escolha da base e ambiente gráfico

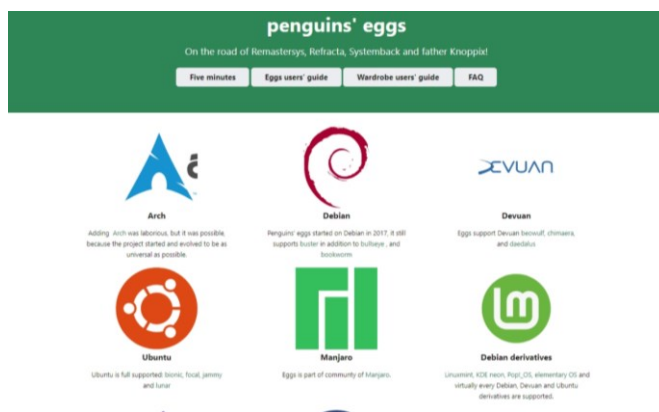
Durante a primeira etapa do desenvolvimento, foi necessária uma pesquisa sobre as distribuições Linux atuais para a escolha daquela que foi utilizada como base para o nosso sistema. Devido aos recursos oferecidos, como amplo repositório de aplicações, desempenho satisfatório, ampla documentação, comunidade extremamente ativa e suporte para 5 anos, a distribuição escolhida foi o Linux Mint 21.1 que tem como base o Ubuntu 22.04 LTS. Dessa forma, é possível obter todas as vantagens do Ubuntu, utilizando um sistema customizado coerente com a proposta do sistema.

A escolha do ambiente gráfico é a parte mais importante do presente trabalho, pois é necessário a escolha de um ambiente que cumpra os seguintes requisitos: Seja simples, parecido com o Windows, tenha um bom desempenho e seja personalizável. Com isso em mente, o ambiente escolhido foi o Cinnamon. Esse ambiente é o padrão da distribuição base e oferece recursos avançados de personalização, além de ser o ambiente recomendado pela comunidade para obter desempenho devido sua leveza e estabilidade.

Outro fator relevante para a escolha desta distro, é o suporte oferecido pelo projeto Penguin's Eggs. Este é um projeto Open Source que tem como objetivo facilitar a criação de novas distribuições Linux, oferecendo uma aplicação que permite criar uma nova distro a partir de outra. Nessa aplicação, é possível compilar um sistema para o formato de arquivo ISO, com um instalador (no caso, o padrão é o instalador Kalamaris), além de oferecer suporte a executar o SO sem necessidade de instalação.

Para utilizar essa ferramenta, ao acessar a página do projeto (<https://penguins-eggs.net/>), é possível notar quais distribuições tem o suporte atualmente para realizar instalação da ferramenta. Nesse sentido, o Linux Mint se encaixa nesse requisito.

Figura 2: Página inicial do projeto Penguin's Eggs



Fonte: Autor (2024)

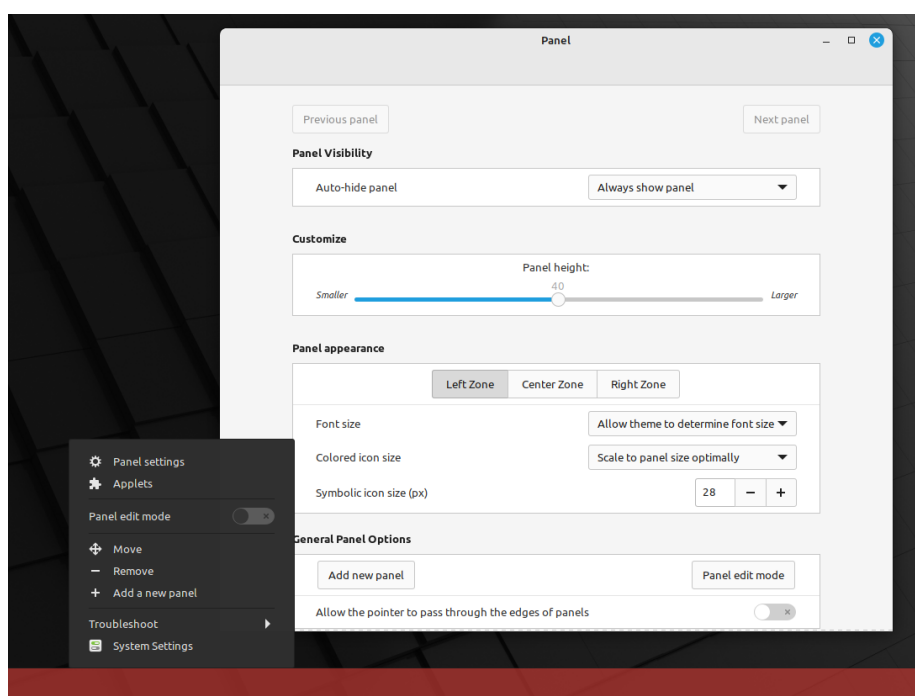


### 7.1.1. Personalização da base e ambiente gráfico

Para que a distribuição tenha uma identidade visual própria, a personalização de questões visuais é essencial. Dessa forma, o ambiente gráfico Cinnamon oferece uma gama de opções de personalização de forma nativa, sendo possível modificar sua interface de forma quase completa.

A primeira etapa da personalização, foi a modificação do painel (ou barra de tarefas), para isso ao clicar com botão direito no painel e posteriormente em “Settings” o tamanho do painel foi ajustado para 36 pixels.

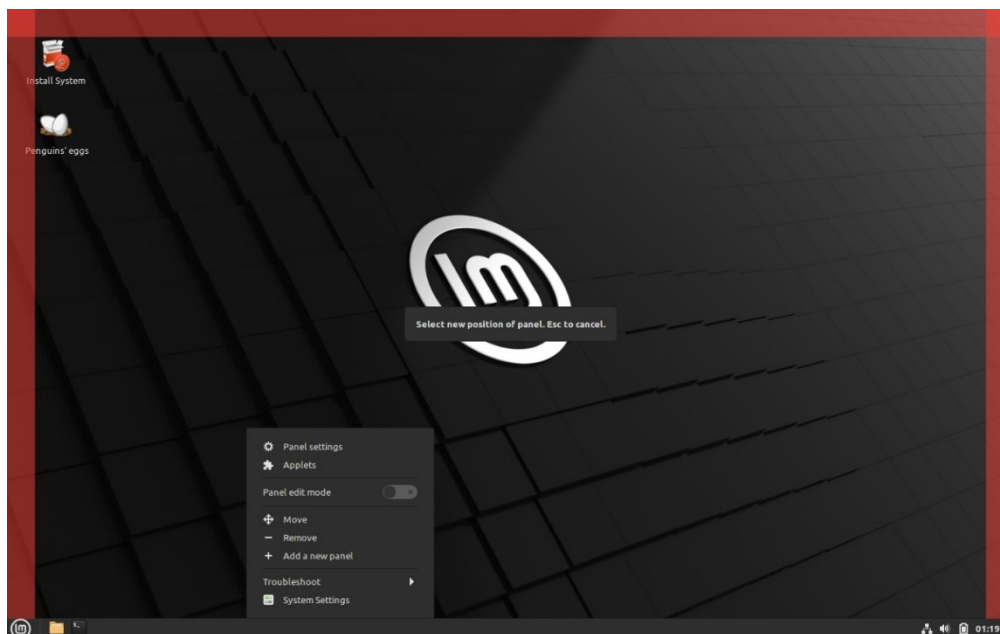
Figura 3: Diminuição do tamanho do painel



Fonte: Autor (2024)

Após alterar sua altura, sua posição na tela foi alterada para o topo. Para isso, ao clicar com o botão direito no painel e em “Move”, três faixas vermelhas apareceram e ao clicar em alguma delas, o painel será movido.

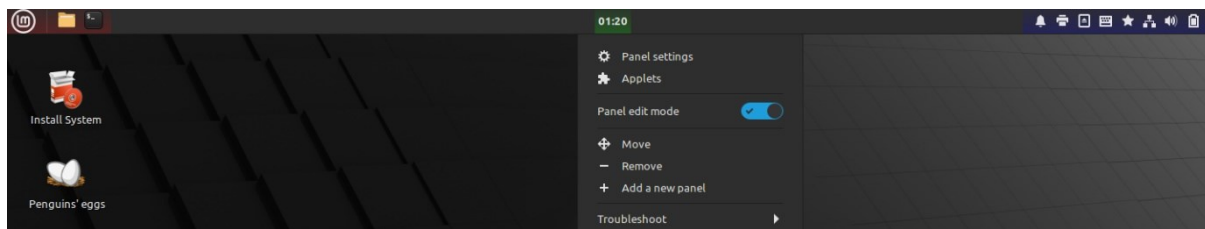
Figura 4: Mudança na disposição do painel na tela



Fonte: Autor (2024)

O próximo passo, foi a centralização do relógio. Ao clicar com o botão direito no painel e habilitar o modo de edição do painel, três zonas de edição irão aparecer no painel permitindo mover os elementos entre elas.

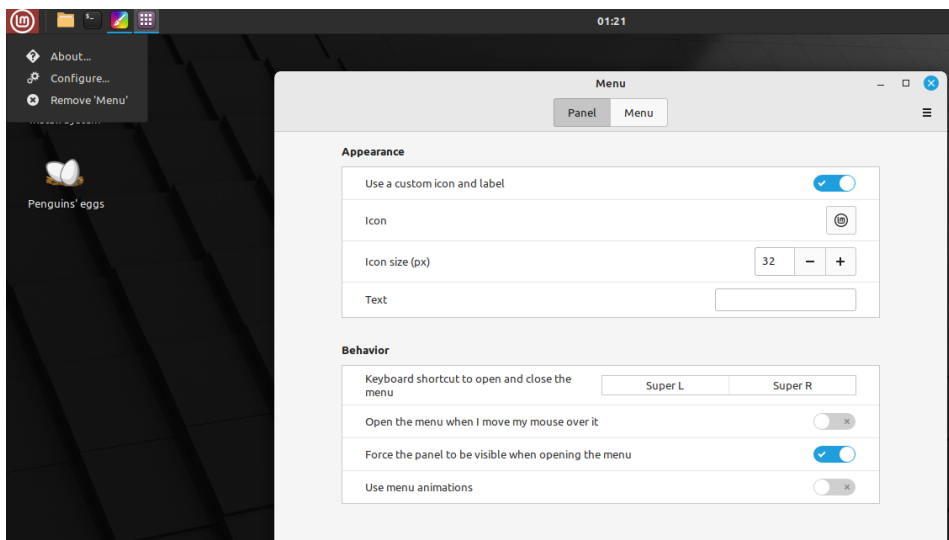
Figura 5: Centralização do relógio



Fonte: Autor (2024)

Para finalizar a edição do painel, o ícone e o texto do painel foram alterados. Ao clicar com o botão direito sobre o ícone do sistema e em configurações o menu de edição foi aberto, permitindo alterar o ícone padrão e o texto padrão.

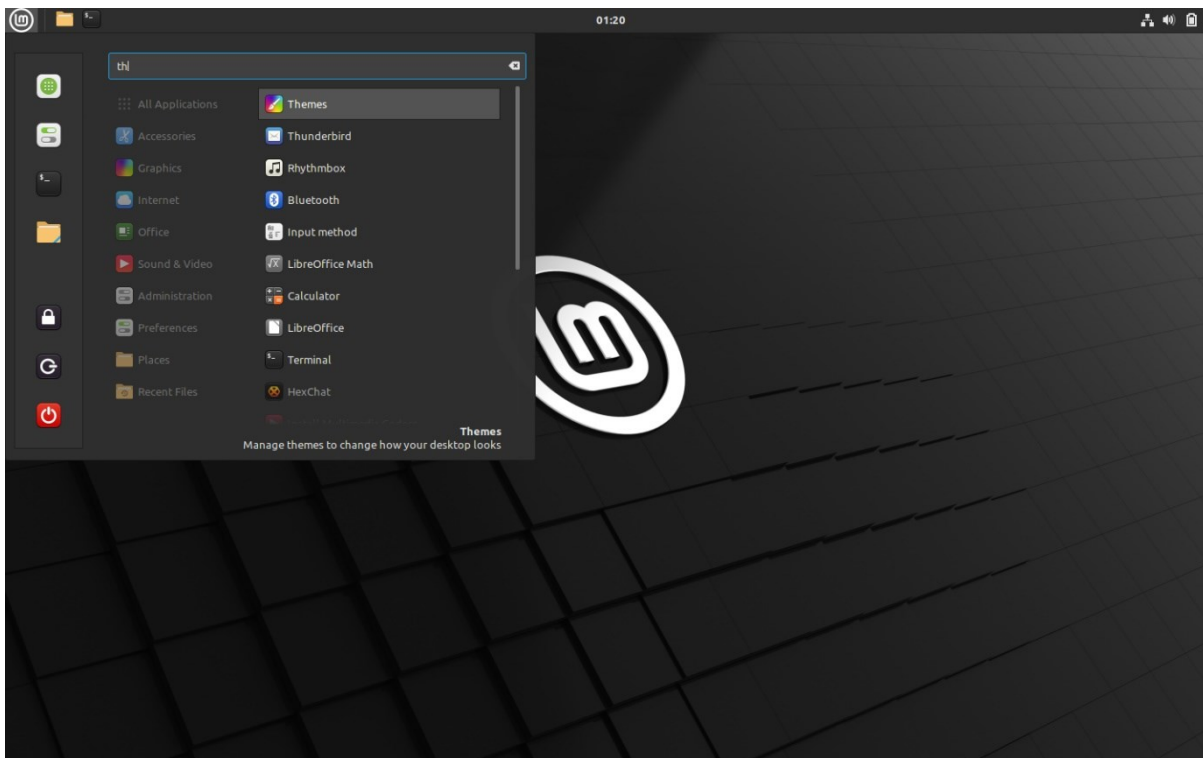
Figura 6: Alteração do ícone e texto do menu



Fonte: Autor (2024)

Com a finalização da edição do painel, o próximo passo foi a personalização do tema do sistema. Para isso, ao clicar na tecla “Super”/”Windows” basta pesquisar pela opção “Themes” no sistema.

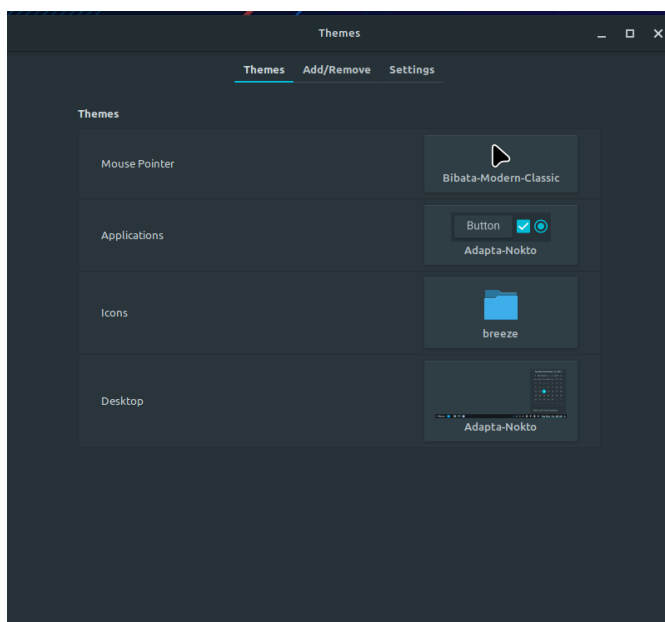
Figura 7: Buscando a opção Themes



Fonte: Autor (2024)

Ao clicar nessa opção, um painel de opções de temas será aberto. Com ele basta selecionar os temas do sistema desejado. Para este projeto, foi utilizado as seguintes configurações: Bibata-Modern-Classsic para o mouse, Adapta-Noko para aplicativos e *desktop* e por fim o tema breeze para ícones.

Figura 8: Seleção do tema do sistema



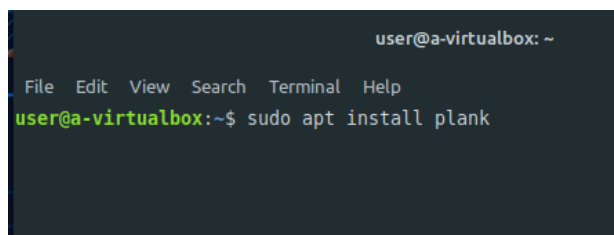
Fonte: Autor (2024)

Com o tema definido e aplicado, foi instalado o aplicativo Plank para emular uma barra de tarefas flutuante, retrátil e transparente. Para isso, utilizando o terminal os seguintes comandos foram executados:

1. `sudo apt-get update`
2. `sudo apt install plank`

Comandos para atualizar os repositórios de aplicativos do sistema e recuperar o pacote do repositório e instalá-lo respectivamente.

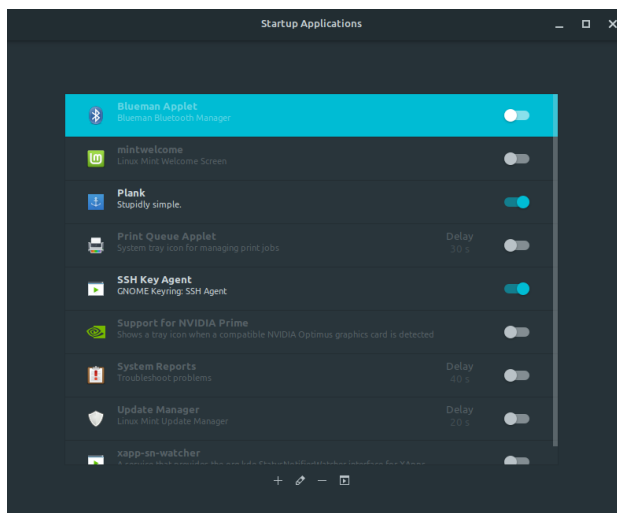
Figura 9: Instalação Plank



Fonte: Autor (2024)

Após instalar o Plank, é necessário configurar sua inicialização automática junto ao boot do sistema. Para isso, no painel “Startup Applications” basta selecionar o check-box referente a aplicação desejada.

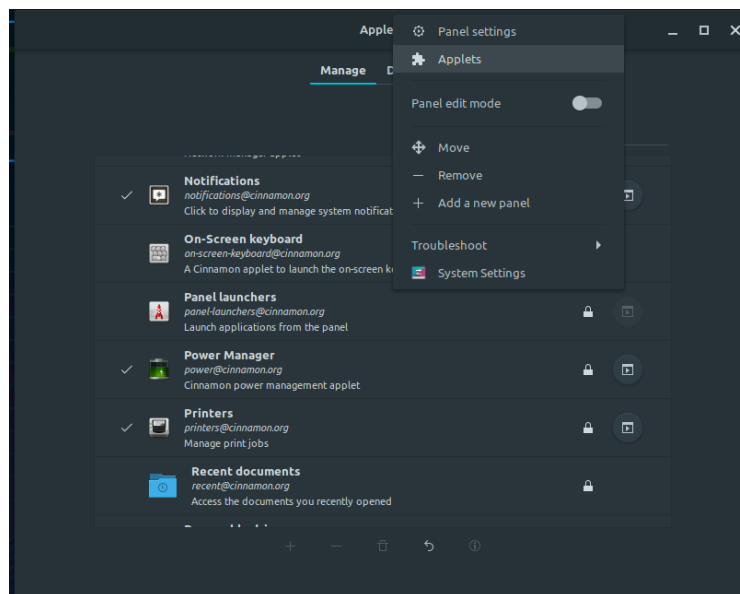
Figura 10: Configuração Plank



Fonte: Autor (2024)

Com o Plank devidamente configurado, é possível remover os ícones dos aplicativos em execução do painel para que o layout da tela tenha uma interface amigável sem duplicidade de informação contribuindo para seu minimalismo. Dessa forma, ao clicar com o botão direito no painel e selecionar a opção “Applets”, o painel de configuração dos Applets foi aberto. Assim, ao deselecionar a opção “Panel launchers” os ícones apareceram somente no aplicativo Plank.

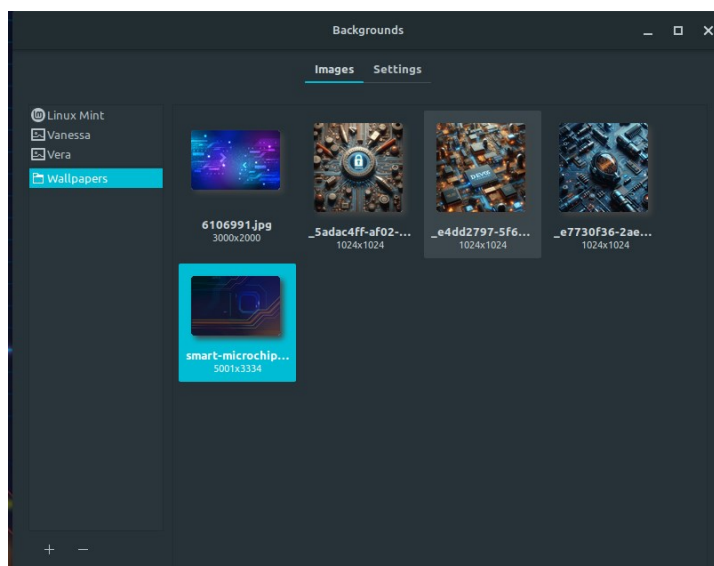
Figura 11: Removendo ícones do painel



Fonte: Autor (2024)

Por fim, para completar a personalização foi adicionado uma pasta com papéis de parede selecionados para o sistema. Para isso, ao clicar com o botão direito sobre a área de trabalho, ao selecionar a opção de “Backgrounds” o menu de opções de papéis de parede será aberto. Dessa forma, ao selecionar o botão “+” foi necessário encontrar a pasta no sistema de arquivos. Assim, os papéis de parede foram carregados.

Figura 12: Selecionado papéis de parede



Fonte: Autor (2024)

## 7.2. Escolha dos pacotes e aplicativos

No sistema, foram instaladas as linguagens Python (em sua versão 3.10.2) e Java (em sua versão JDK 11.0.20), o *framework* Node JS (em sua versão 12.22.9) e o IDE de desenvolvimento Visual Studio Code (VSCode), em virtude de sua destacada popularidade no cenário da programação contemporânea. Essa seleção baseia-se em pesquisas confiáveis da plataforma de versionamento de código mundialmente conhecida GitHub, conforme evidenciado por fontes respeitáveis. Segundo o levantamento disponibilizado no site Hostinger (Quais\_Sao\_as\_Linguagens\_de\_Programacao\_Mais\_Usadas\_em\_2023, 2023), essas linguagens emergem como líderes indiscutíveis no mundo da programação em 2023. Adicionalmente, dados da pesquisa realizada pela Código Fonte TV (Pesquisa de Programadores 2023, 2023) confirmam o amplo uso dessas tecnologias por programadores, consolidando sua relevância na atualidade.

Outras aplicações pertinentes ao uso do usuário final instalados no sistema, foram o navegador Open Source Chromium e o reprodutor de mídia Open Source VLC.

Ao analisar os aplicativos já instalados por padrão, foi notório sua vasta gama de opções. Porém, dentre essas opções algumas não se mostraram necessárias para o foco do sistema. Dentre elas, estão: Firefox, Hexchat, Thunderbird, LibreOffice-math, LibreOffice-base, Hypnotix, Celluloid, Rhythmbox e LibreOffice-draw.

Desta forma, com as aplicações essenciais para o sistema foi elaborado um shell script para realizar o processo de instalação e remoção dos aplicativos de forma automatizada.

Figura 13: Shell script instalação e remoção aplicativos

```

1 curl -d 'https://code.visualstudio.com/sha/download?
  build=stable&os=linux-deb-x64'
2
3 sudo dpkg --install code_1.89.1-1715060508_amd64.deb
4
5 sudo apt remove firefox hexchat thunderbird libreoffice-math
6 libreoffice-base hypnotix celluloid rhythmbox
7
8 sudo apt remove libreoffice-draw
9
10 sudo apt autoremove
11
12 sudo apt install vlc chromium
13
14 sudo apt install python3-pip python3-venv
15
16 sudo apt install nodejs
17
    
```

Fonte: Autor (2024)

### 7.3. Desenvolvimento aplicação

Atualmente, as distribuições Linux oferecem várias maneiras de instalar aplicativos, como usar gerenciadores de pacotes como apt, yum, pacman, entre outros, ou recorrer a lojas de aplicativos específicas de cada distribuição. Embora eficazes em suas próprias capacidades, esses métodos podem ser desafiadores para usuários iniciantes ou para aqueles que desejam uma experiência mais intuitiva.

Dessa forma, um aplicativo dedicado de instalação de aplicativos pode oferecer uma interface gráfica amigável, simplificando o processo de descoberta, instalação e atualização de aplicativos para usuários de todos os níveis de habilidade sendo coerente ao contexto do sistema proposto.

Para realizar o desenvolvimento desta aplicação, o *framework* Flutter em sua versão 3.21.0 foi escolhido devido sua funcionalidade de compilação da aplicação para Linux de forma nativa.

#### 7.3.1. Definição do escopo

A aplicação desenvolvida, tem como objetivo ajudar os novos usuários a se familiarizarem ao sistema, permitindo configurações rápidas como instalação de aplicativos e troca de senha. Também é de extrema importância contextualizar o usuário a cerca do sistema, uma vez que este será executado juntamente do iniciar do sistema.

Dessa forma a aplicação oferece as seguintes funcionalidades:

1. Contextualização do usuário sobre o sistema:
  - a. Contém uma tela de boas-vindas ao usuário, explicando um pouco sobre o sistema e apresentando de forma visual algumas funcionalidades.
2. Instalação de aplicações e ferramentas:
  - a. O aplicativo possui uma seção que permite a instalação de aplicativos e ferramentas para o desenvolvimento de *software* de forma simples e rápida.
3. Informações adicionais sobre o sistema e desenvolvedor:
  - a. Página que contém informações específicas sobre o sistema.
  - b. Página que contenha informações do autor do sistema.
4. Orientações sobre o sistema:

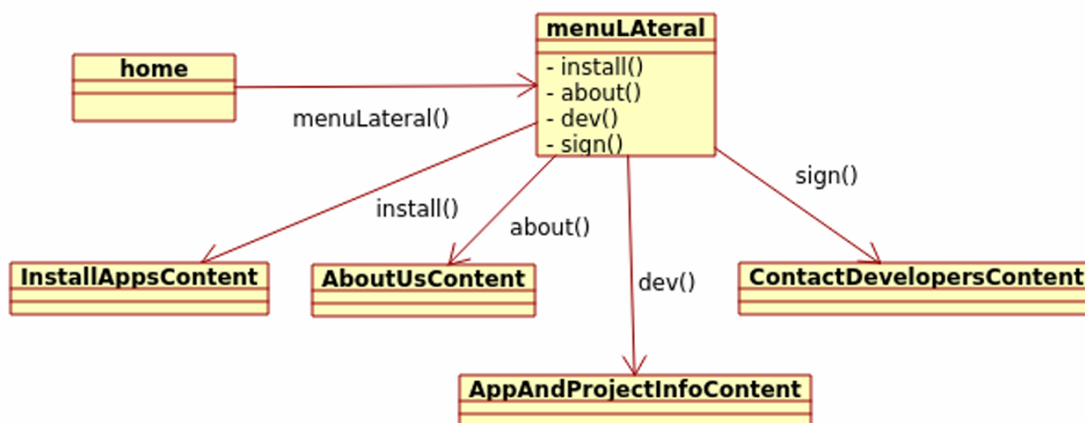


- a. Seção que contenha orientações de uso do sistema e permita a definição de uma nova senha para o usuário.

### 7.3.2. Diagrama de classe

O Diagrama de classe representativo utilizado para representar as principais classes da aplicação e como elas se relacionam (figura 14).

Figura 14: Diagrama de classe.



Fonte: Autor (2024)

### 7.3.3. Interfaces do aplicativo

#### I. Tela inicial

Ao iniciar o sistema ou a aplicação, a primeira tela será a tela Home/Boas-vindas. Esta tela contém um breve texto introdutório, juntamente com um carrossel de imagens do sistema (figura 15).

Figura 15: Tela inicial.

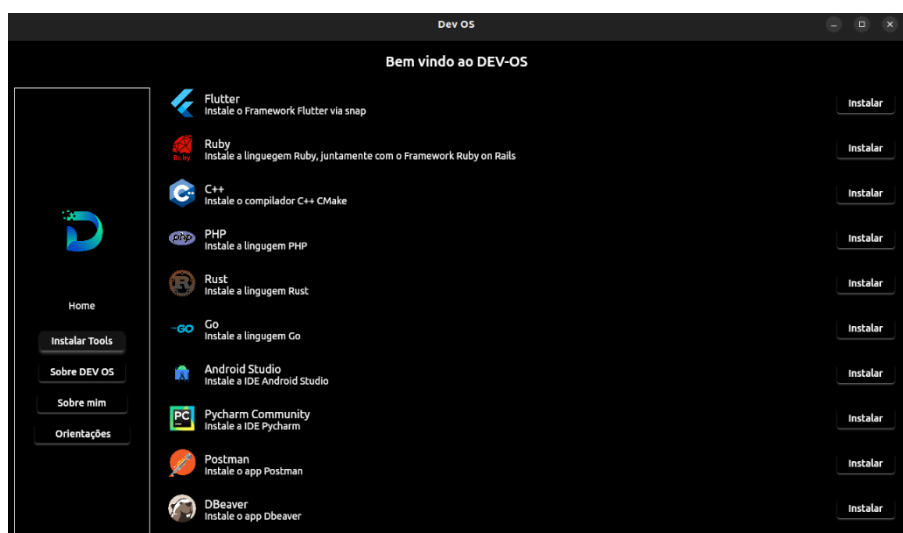


Fonte: Autor (2024)

## II. Tela de instalação de aplicações

Ao descer sobre o menu lateral esquerdo, a próxima tela disponível é a de instalação de ferramentas (figura 16). Esta, contém uma lista de aplicações que o usuário pode instalar clicando sobre o botão instalar, localizado a direita de cada opção.

Figura 16: Tela de instalação de aplicações



Fonte: Autor (2024)

## III. Tela sobre o sistema

Ainda percorrendo o menu lateral, a próxima opção disponível é a tela “Sobre DEV OS”. Esta página (figura 17) contém algumas informações sobre o sistema, como as aplicações que ele oferece de maneira pré-instalada.

Figura 17: Tela “Sobre DEV OS”

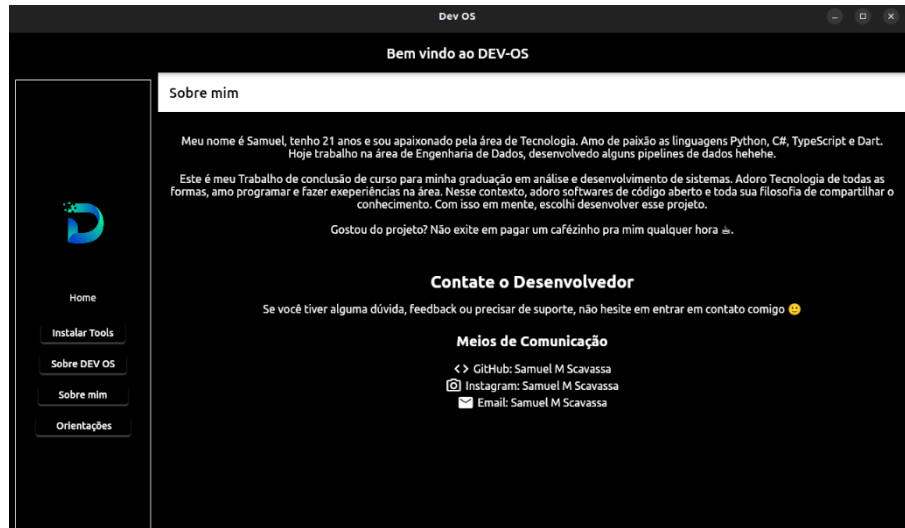


Fonte: Autor (2024)

#### IV. Tela sobre o desenvolvedor

Ainda percorrendo o menu lateral, a próxima opção disponível é a tela “Sobre mim”. Esta página (figura 18) contém algumas informações sobre o autor do sistema.

Figura 18: Página “Sobre mim”

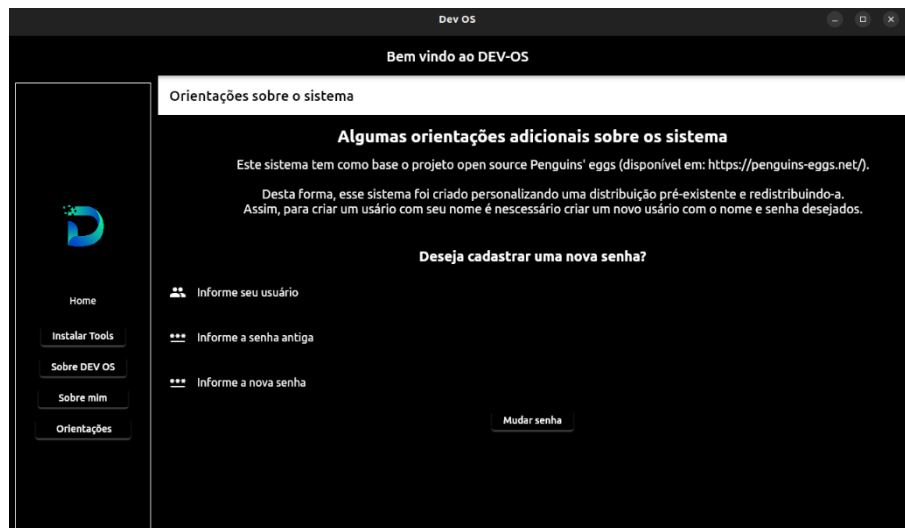


Fonte: Autor (2024)

#### V. Tela orientações

Por fim, a última opção disponível é a tela “Orientações”. Esta página (figura 19) contém algumas orientações sobre o sistema, e a funcionalidade de troca de senha do usuário.

Figura 19: Tela “Orientações”



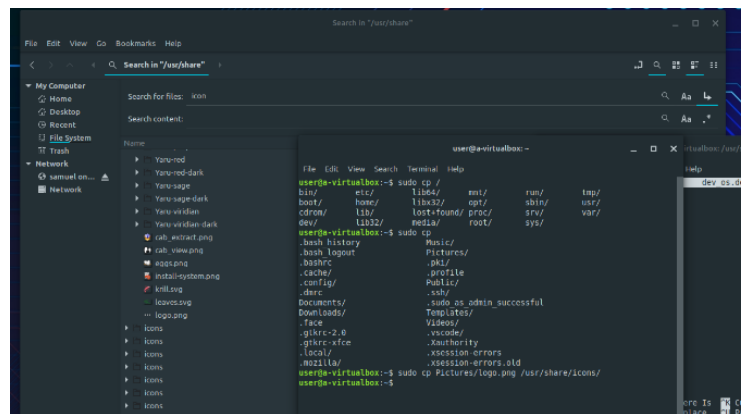
Fonte: Autor (2024)

## 7.4. Integração sistema e aplicativo

Para realizar a integração entre a aplicação e o sistema desenvolvidos, primeiramente o arquivo gerado pela compilação do código fonte tem que estar disponível no diretório “/usr/share/” do sistema para que esta aplicação fique disponível para todos os usuários. Uma vez com o arquivo compilado disponível no diretório “~/home/user/Documents/” basta copiá-lo com o comando “sudo cp -r Documents/dev\_os /usr/share/”. Assim, a pasta dev\_os que contém o aplicativo compilado estará em um diretório compartilhado a todos os usuários.

O próximo passo, é copiar o ícone que será utilizado para representar a aplicação no sistema. Uma vez com o arquivo disponível no diretório “~/home/user/Pictures/” basta copiá-lo para o diretório “usr/share/icons/”, para que este também fique disponível para todos os usuários, como mostra a figura 20.

Figura 20: Copiando ícone.



Fonte: Autor (2024)

Para que a aplicação possa ser executada e reconhecida pelo sistema, um arquivo .desktop deve ser criado no diretório “/usr/share/applications/”. Dessa forma, utilizando o comando “sudo nano” no diretório acima o editor de texto nano será aberto com permissões de super usuário, com isso é possível criar o arquivo neste diretório como mostra a figura 21.

Figura 21: Criação do arquivo .desktop

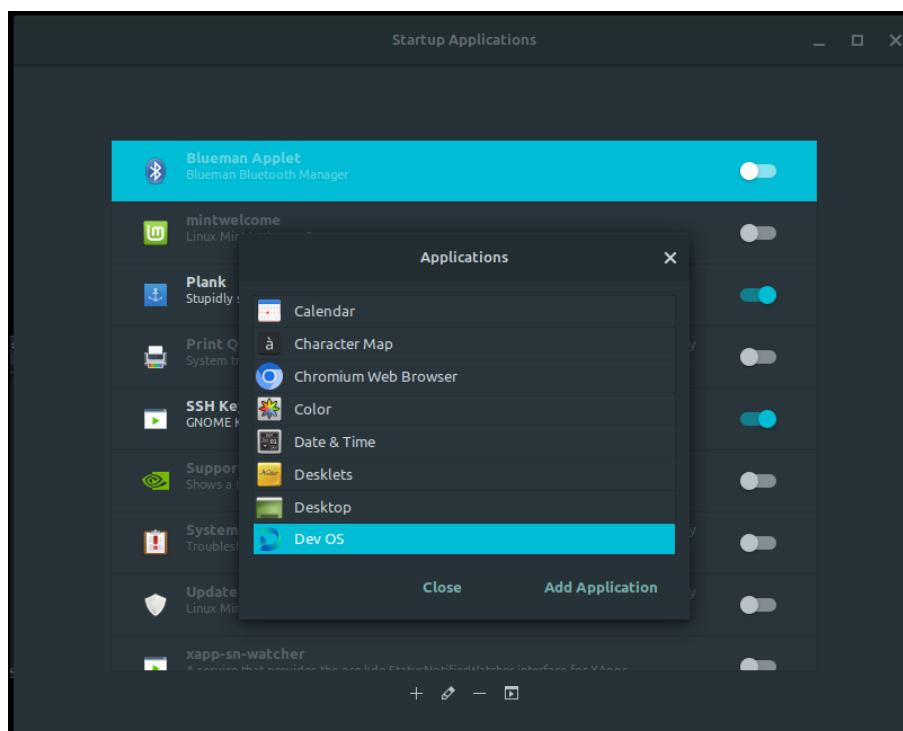
```

user@a-virtualbox: /usr/share/applications
File Edit View Search Terminal Help
GNU nano 6.2 dev_os.desktop
[Desktop Entry]
Version=1.0
Name=Dev OS
Comment=Utilitário do Dev OS
Exec=/usr/share/dev_os/dev_os
Icon=/usr/share/icons/logo.png
Terminal=false
Type=Application
Categories=Utility;Application;
    
```

Fonte: Autor (2024)

Por fim, foi configurado a inicialização automática da aplicação junto ao iniciar do sistema. Para isso, o gerenciador de aplicações de inicialização foi acessado e nele foi adicionado a aplicação personalizada (figura 22).

Figura 22: Adicionando aplicação ao iniciar do sistema.



Fonte: Autor (2024)

## 7.5. Compilação para ISO

Uma vez com a aplicação Penguin's Eggs instaladas conforme a documentação oficial (Eggs users' guide, 2024), o seguinte shell script foi elaborado (figura 23), para configurar a aplicação conforme o sistema em uso (instalação de dependências e libs necessárias), realizar a instalação do instalador calamares e iniciar a compilação.

Figura 23: Shell script para compilação

```

1
2 # Comando para configurar eggs para o sistema em uso
3 sudo eggs dad -d to re-configure eggs
4
5 # Comando para instalar o instalador calamares
6 sudo eggs calamares
7
8 # Comando para iniciar a compilação do sistema para ISO
9 sudo eggs produce -c
10

```

Fonte: Autor (2024)

Após o término da execução da aplicação o arquivo .iso estará disponível no diretório “~/home/eggs/.mnt/\* .iso”.

## 8. Resultados e Discussões

Ao iniciar o desenvolvimento, o propósito do sistema era ser uma opção Linux para desenvolvedores sendo um sistema moderno e com um consumo de recursos computacionais baixo. Esse objetivo foi plenamente atingido.

O sistema, nomeado de Dev OS, apresentou um consumo computacional coerente com seu objetivo, em média 800 megabytes de memória RAM em estado ocioso além de apresentar uma boa performance em um ambiente virtual com um processador de apenas dois núcleos e 2 gigabytes de memória RAM.

Seu aspecto visual também cumpriu seu objetivo inicial. A interface gráfica Cinnamon permite que o usuário, uma vez com conhecimentos técnicos necessários, realize modificações profundas em seu visual. A escolha do tema da interface, ícones e papéis de parede também contribuiu para a identidade visual do sistema.

Durante o desenvolvimento do sistema, a ideia de que o sistema iria ter todas as ferramentas de desenvolvimento populares se mostrou inviável uma vez que isso aumentaria no tamanho do sistema final, aumentaria o tempo de instalação devido ao fato de ter mais arquivos e nem todas as ferramentas seriam utilizadas pela maioria dos usuários. Para contornar essa situação, foi proposto o desenvolvimento de uma aplicação que permitiria a instalação das ferramentas que o sistema não iria fornecer de maneira pré-instalada. Com isso, a ideia da aplicação foi estendida para uma aplicação inicial do sistema.

Portanto, é possível concluir que o sistema atingiu seu objetivo. Também se espera com esse sistema contribuir com a comunidade Open Source e ser uma opção para desenvolvedores adotarem o Linux como ambiente de desenvolvimento principal.

## 9. Conclusão

Durante o desenvolvimento deste trabalho, conhecimentos no sistema operacional Linux foram amplamente utilizados como: conhecimento do sistema de diretórios, comandos, serviços, processos, programação em shell script e personalização de interfaces.

Além disso, conhecimentos em programação foram aprofundados como a manipulação de recursos do sistema através de uma aplicação, conceitos de programação orientada a objetos, código limpo e o *framework* Flutter.

Portanto, é possível concluir que o presente trabalho teve contribuição direta para ampliação de conhecimento na área não somente de sistemas operacionais, mas também na área de desenvolvimento. Espera-se que os resultados obtidos possam servir de base para futuros projetos Open Source, fortalecendo essa comunidade contribuindo com a sociedade oferecendo aplicações e ferramentas de qualidade e um custo acessível.

## 10. Referências

- ANDRÉ, Anderson S. et al. *Desktop Linux Customizado Para Simplicidade—Uso na Terceira Idade*. Anais SULCOMP, v. 7, 2014.
- BLUM, Richard; BRESNAHAN, Christine. *Linux Command Line and Shell Scripting Bible*. 4. ed. Hoboken: Wiley, 2021.
- CAMPOS, Augusto. O que é Linux. BR-Linux. Florianópolis, março de, 2006.
- CAMPOS, Augusto. O que é *software* livre. BR-Linux. Florianópolis, março de, 2006.
- CAMPOS, Augusto. O que é uma distribuição Linux. BR-Linux. Florianópolis, 2006.
- CHACON, Scott; STRAUB, Ben. *Pro Git*. 2. ed. New York: Apress, 2014.
- Código Fonte TV. Pesquisa de Programadores 2023. Disponível em: <<https://pesquisa.codigo-fonte.com.br/2023>>. Acesso em: 10 out. 2023.
- DE SOUZA, Cauê Aruã Zani. DESENVOLVIMENTO DE UMA DISTRIBUIÇÃO LINUX VOLTADA A EDUCAÇÃO INFANTIL. 2009.
- Eggs users' guide. Disponível em: < <https://penguins-eggs.net/docs/Tutorial/eggs-users-guide> >. Acesso em: 12 abr. 2024.
- GOOGLE. Flutter Documentation. 2022. Disponível em: <https://flutter.dev/docs>. Acesso em: 15 jun. 2024.
- GNOME. GNOME Boxes Documentation. 2022. Disponível em: <https://help.gnome.org/users/gnome-boxes/stable/>. Acesso em: 15 jun. 2024.

Hostinger. Linguagens de Programação Mais Usadas em 2023. Disponível em: <[https://www.hostinger.com.br/tutoriais/linguagens-de-programacao-mais-usadas#Quais\\_Sao\\_as\\_Linguagens\\_de\\_Programacao\\_Mais\\_Usadas\\_em\\_2023](https://www.hostinger.com.br/tutoriais/linguagens-de-programacao-mais-usadas#Quais_Sao_as_Linguagens_de_Programacao_Mais_Usadas_em_2023)>. Acesso em: 10 out. 2023.

Linux From Scratch - Crie sua própria distro Linux do zero! Disponível em: <<https://plus.diolinux.com.br/t/linux-from-scratch-crie-sua-propria-distro-linux-do-zero/51936>>. Acesso em: 14 jun. 2023.

LOVE, Robert. *Linux Kernel Development*. 3. ed. Upper Saddle River: Addison-Wesley, 2010.

MICROSOFT. Visual Studio Code Documentation. 2022. Disponível em: <https://code.visualstudio.com/docs>. Acesso em: 15 jun. 2024.

ORACLE. VirtualBox User Manual. 2022. Disponível em: <https://www.virtualbox.org/manual/UserManual.html>. Acesso em: 15 jun. 2024.

Pensando em criar uma distro Linux? É bom ler isso... Disponível em: <<https://plus.diolinux.com.br/t/pensando-em-criar-uma-distro-linux-e-bom-ler-isso/27511>>. Acesso em: 14 jun. 2023.

SILVEIRA, Sergio Amadeu da. *Software livre*. 2018.

SOUSA, Fiterlinge Martins. Processo de customização de uma distribuição GNU/Linux. In: VII CONNEPI-Congresso Norte Nordeste de Pesquisa e Inovação. 2012.

TRES, Crineu; QUEIROZ, Ulisses Muniz de. Estudo de implementação de uma distribuição LINUX. 2003.

UBUNTU. Documentação do Ubuntu 22.04 LTS. 2022. Disponível em: <https://ubuntu.com/download/desktop>. Acesso em: 15 jun. 2024.

Viva o Linux - A maior comunidade GNU/Linux da América Latina! Disponível em: <<https://www.vivaolinux.com.br/>>.

VMWARE. VMware Workstation Player Documentation. 2022. Disponível em: <https://www.vmware.com/products/workstation-player/workstation-player-evaluation.html>. Acesso em: 15 jun. 2024.