



# Computadores de baixo custo para o ensino de programação em escolas públicas utilizando TV Boxes

*B. G. Rodrigues      G. M. Adamo      M. V. R. Oliveira  
I. C. Garcia*

Relatório Técnico - IC-PFG-24-37  
Projeto Final de Graduação  
2024 - Dezembro

UNIVERSIDADE ESTADUAL DE CAMPINAS  
INSTITUTO DE COMPUTAÇÃO

The contents of this report are the sole responsibility of the authors.  
O conteúdo deste relatório é de única responsabilidade dos autores.

# Computadores de baixo custo para o ensino de programação em escolas públicas utilizando TV Boxes

Bruno Gonçalves Rodrigues      Giovanna Magario Adamo  
Maria Vitória Rodrigues Oliveira      Islene Calciolari Garcia

## Resumo

O acesso à educação no Brasil ainda não é democrático, assim como o acesso a tecnologia. Muitos estudantes não possuem a infraestrutura condizente com as necessidades de ensino atuais, especialmente em se tratando de estudantes da rede pública de ensino. O contato com tecnologias modernas e o acesso a conteúdos como o ensino de programação ainda é extremamente limitado. Neste mesmo cenário de escassez, a Receita Federal confiscou nos últimos anos um grande número de TV Boxes piratas, equipamentos utilizados para acesso a conteúdos de forma ilegal, e que seriam destruídos, gerando lixo eletrônico e desperdício de equipamentos, caso não fossem doados para instituições de ensino, como a Unicamp, a fim de possibilitar a realização de projetos que buscassem seu reaproveitamento de forma benéfica para a sociedade.

Assim, este projeto tem como objetivo a utilização destas TV Boxes como computadores de baixo custo, que possam ser usados em escolas públicas para o ensino de programação, por meio de iniciativas próprias ou de parceiros. Por meio da instalação de um novo Sistema Operacional e sua personalização, voltada para o uso educacional, é possível oferecer aos estudantes um ambiente que possibilita o contato com a aprendizagem de informática e programação.

## 1 Introdução

A falta de computadores nas escolas públicas do país é um problema já conhecido e que parece longe de ser resolvido. Mesmo que atualmente haja uma gama de dispositivos diferentes que possam ser usados no ensino (Notebooks, Chromebooks, tablets etc), a realidade nas escolas de ensino público ainda se encontra antiquada. Segundo um levantamento recente do Centro de Estudos e Pesquisas em Tecnologia de Redes e Operações, apenas 29% das escolas públicas brasileiras possuem qualquer tipo de equipamento para os alunos acessarem a internet (tablet, notebooks ou desktops) e, dentre as que possuem, a média é de 1 equipamento para 10 alunos.[1] Os dados também mostram a falta de conectividade nas escolas, que possuem planos de internet com velocidade abaixo da esperada.

Um dos principais fatores para essa baixa adesão das escolas a novas tecnologias é o custo elevado dos equipamentos, ainda mais dado os orçamentos limitados que as escolas possuem, como foi dito pelo presidente do Fórum Nacional de Conselhos Estaduais de Educação (Foncede) na audiência pública na Comissão de Educação (CE).[2] Neste cenário, a busca por equipamentos de baixo custo que executem as funções de um computador se faz necessária para a inclusão digital nas escolas. Um equipamento que se enquadra nesta classificação é a TV Box, um dispositivo eletrônico simples e barato que foi projetado para o acesso de conteúdos online, especialmente de streamings de vídeo, usadas principalmente em televisões que não possuem acesso à internet.

A TV Box possui portas USB para conectar mouse e teclado, além de uma entrada HDMI para conexão com monitores. Quando utilizada para acessar ilegalmente conteúdos e canais pagos, pode ser considerada uma forma de pirataria, o que leva à apreensão desses aparelhos pela Receita Federal. No entanto, esses dispositivos têm o potencial de se transformarem em computadores de baixo custo para serem utilizados nas escolas do país.

Em todo o país, temos diversos trabalhos e projetos que visam transformar as TV Boxes em computadores para serem utilizados na educação. Os projetos, como *Transformação de TV Boxes piratas em computadores de baixo custo suportados por nuvem computacional* (Ribeiro et al., 2023) e *Uma proposta de computadores de baixo custo utilizando tv boxes apreendidas* (Lopes et al., 2023), apesar de possibilitarem o acesso a dispositivos de baixo custo, não tiveram suas soluções testadas por estudantes para verificar se eles se adaptam ao sistema operacional e interface do novo dispositivo, além de sofrerem com dificuldades de performance.

Tendo os projetos supracitados como referência e os problemas que ainda persistiram no uso da TV Box como computador, esse trabalho tem como objetivo a transformação das TV Boxes em computadores de baixo custo com boa usabilidade para que alunos de escolas públicas possam utilizá-los sem dificuldades, especialmente no aprendizado de programação. Considerando os problemas de performance do sistema encontrados em outros trabalhos e as limitações do próprio dispositivo em termos de memória RAM, processador, entre outros, propomos uma nova interface gráfica para o Armbian. Após diversas tentativas com outros sistemas operacionais mais leves que detalharemos nas seções posteriores, a interface proposta apresenta melhor usabilidade do que as demais interfaces padrão do Armbian, além de performance superior em aplicações mais pesadas como navegadores, Integrated Development Environments (IDEs), etc.

## 2 Motivação

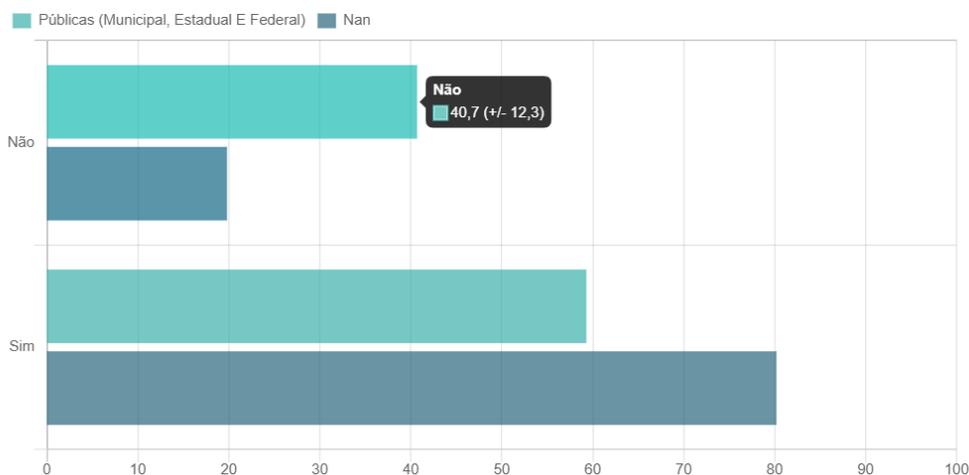
A Receita Federal realizou diversas operações de apreensão de dispositivos ilegais, parte do Plano de Ação de Combate à Pirataria. Nestas ações, comumente os dispositivos são destruídos e posteriormente descartados, o que resulta em geração de lixo eletrônico e desperdício de materiais

que poderiam ser utilizados para outros fins. A fim de evitar este desperdício, a Receita Federal estabeleceu parcerias com diversas instituições de ensino superior para adequação destes dispositivos para realizarem outras funções, principalmente no ramo da educação e da inclusão digital. A UNICAMP foi uma dessas instituições que recebeu as TV Boxes, recebendo cerca de 200 aparelhos em meados de 2023. Estes aparelhos vêm sendo aplicados em pesquisas de graduação e pós-graduação, bem como em disciplinas dos cursos do Instituto de Computação.

O acesso a computadores em escolas públicas para o ensino de alunos no Brasil é limitado, e fatores como acesso à internet e ensino de programação são ainda mais inacessíveis. De acordo com dados do Cetic (Centro Regional de Estudos para o Desenvolvimento da Sociedade da Informação), em 2022, mais de 40% das escolas públicas não possuíam computador disponível para uso dos alunos em atividades educacionais, valor que cai para apenas 19,8% se comparado com escolas particulares, como é possível observar no gráfico abaixo.

#### B1B - ESCOLAS QUE POSSUEM COMPUTADOR DISPONÍVEL PARA USO DOS ALUNOS EM ATIVIDADES EDUCACIONAIS

TOTAL DE ESCOLAS DE ENSINO FUNDAMENTAL E MÉDIO



Fonte: CGI.br/NIC.br, Centro Regional de Estudos para o Desenvolvimento da Sociedade da Informação (Cetic.br), Pesquisa sobre o uso das tecnologias de informação e comunicação nas escolas brasileiras - TIC Educação 2022

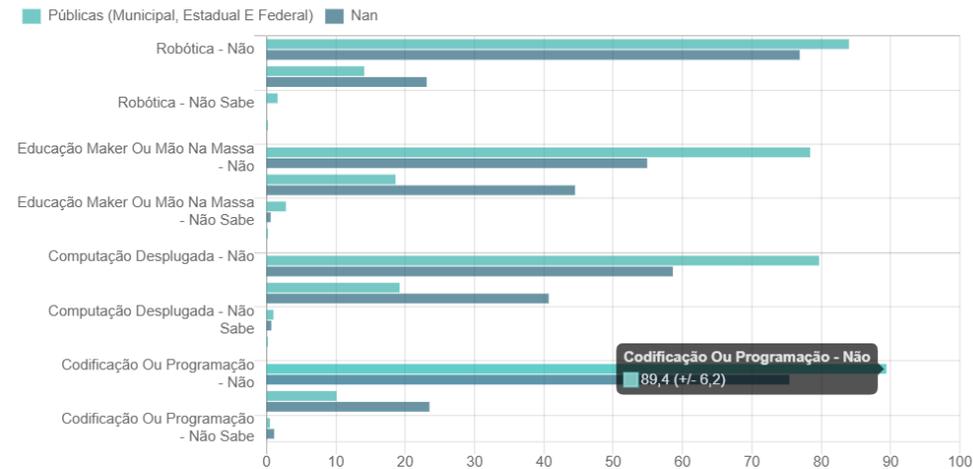
Por questões de arredondamento, a soma dos resultados pode não totalizar 100%.

Figura 1: Dados de escolas que possuem computador disponível para uso dos alunos em atividades educacionais em 2022.[3]

Ao observar dados de acesso à internet, verificamos que mais de 46% das escolas públicas não possuem computador e acesso à internet para uso dos alunos, comparado a apenas 26,5% das escolas particulares. Com relação ao ensino de programação, quase 90% dos coordenadores pedagógicos de escolas públicas afirmaram que não ofertavam aulas de codificação ou programação, como é possível observar no gráfico abaixo.

### G1 - COORDENADORES PEDAGÓGICOS, POR OFERTA DE AULAS OU ATIVIDADES SOBRE ROBÓTICA E PROGRAMAÇÃO NAS ESCOLAS

TOTAL DE COORDENADORES PEDAGÓGICOS DE ESCOLAS DE ENSINO FUNDAMENTAL E MÉDIO



Fonte: CGI.br/NIC.br, Centro Regional de Estudos para o Desenvolvimento da Sociedade da Informação (Cetic.br), Pesquisa sobre o uso das tecnologias de informação e comunicação nas escolas brasileiras - TIC Educação 2022

Por questões de arredondamento, a soma dos resultados pode não totalizar 100%.

Figura 2: Dados de coordenadores pedagógicos, por oferta de aulas ou atividades sobre robótica e programação nas escolas em 2022.[4]

Considerando o problema de escassez de recursos tecnológicos e acesso à aprendizagem de programação nas escolas públicas, o objetivo deste trabalho é transformar as TV Boxes em computadores de baixo custo para que sejam usados no ensino em escolas públicas, especialmente no ensino de programação. Sabemos que a área da tecnologia é cada vez mais procurada e demandada no mercado de trabalho, o que gera a necessidade da modernização dos currículos escolares para que se adequem às necessidades dos alunos, como por exemplo a inclusão de disciplinas relacionadas a Computação como a programação. Além disso, há evidências que indicam que o aprendizado de programação ainda no ensino básico gera efeitos positivos no desenvolvimento do raciocínio lógico e capacidade de abstração, habilidades úteis em outras disciplinas e áreas.[5][6] Por estes motivos, decidimos focar também em tornar a TV Box um computador possível de ser utilizado no ensino de programação, adicionando ferramentas que auxiliem no aprendizado de programação.

## 3 Objetivos

### 3.1 Objetivo Geral

O objetivo geral deste projeto é instalar e adaptar um sistema operacional livre nas Tv Boxes ilegais apreendidas pela Receita Federal e doadas para a Unicamp, para serem utilizadas como computadores de baixo custo no ensino de programação em escolas públicas, permitindo o desenvolvimento de algoritmos simples de forma fluida e intuitiva para os alunos do ensino fundamental e médio.

### 3.2 Objetivos Específicos

Os objetivos específicos listados abaixo serão implementados para atingir o objetivo geral.

- Identificar sistemas operacionais open-source que podem ser instalados em TV Boxes;
- Instalar e configurar as opções de sistemas operacionais nas TV Boxes;
- Analisar qual sistema operacional melhor se adequa ao dispositivo;
- Realizar testes práticos com alunos de escolas públicas para avaliar a usabilidade do sistema operacional;

## 4 Trabalhos Relacionados

O projeto de Ribeiro et al. [7] foi desenvolvido com o objetivo de utilizar TV Boxes para a realização de atividades de programação da disciplina de Algoritmos e Programação de Computadores (MC102), que é uma disciplina introdutória de programação presente no currículo de diversos cursos de graduação da UNICAMP. Os métodos explorados para transformar TV Boxes em computadores de baixo custo incluíram a tentativa de uso de ROMs personalizadas para o sistema Android, o uso do sistema Armbian e, como solução final, uma arquitetura remota, utilizando a TV Box com Armbian como ponto de acesso para VMs (máquinas virtuais) dos serviços da Amazon Web Services (AWS), mais especificamente o serviço o EC2 (Elastic Computing). Por fim, o projeto concluiu que o desempenho das TV Boxes nos métodos abordados foi insatisfatório, com lentidão e dificuldade de uso dos softwares propostos.

O trabalho de Lopes et al. [8] tem como objetivo transformar TV Boxes em computadores de baixo custo usando soluções open-source. Realizou-se a triagem de diferentes modelos de TV Boxes e definição de métodos para instalação de um novo sistema operacional nestes modelos. O sistema escolhido foi o Armbian, e os testes foram realizados com a imagem do SO com legacy kernel, já incluindo a interface gráfica xfce. Por fim, os testes foram realizados com diversos programas, como editor de texto, IDE, navegador, editor de imagens e terminal. Os resultados de desempenho foram comparados aos de computadores pessoais, e foi concluído que as TV Boxes podem ser reaproveitadas para uso pessoal e/ou acadêmico. Porém, identificou-se desempenho insatisfatório em aplicações pesadas, em especial aquelas com maior uso gráfico, devido às limitações do hardware.

O projeto de Rodrigues et al.[9], nomeado educabox, tem como objetivo democratizar o acesso ao conhecimento, especialmente em regiões onde a infraestrutura tecnológica pode ser limitada, por meio de TV Boxes. O projeto visa possibilitar a instalação de imagens Android e Armbian personalizadas em TV Boxes, com aparência e aplicativos pensados para fins educacionais. Seu repositório reúne imagens para diferentes dispositivos, além de tutoriais de instalação, muito úteis no desenvolvimento de projetos semelhantes, inclusive o presente. Os objetivos futuros do projeto incluem a criação de uma ROM própria e o desenvolvimento de processos e repositório completo para comunidade.

## 5 Referencial Teórico

### 5.1 Sistema Operacional Linux e suas distribuições

Linux é um termo abrangente que se refere aos sistemas operacionais que utilizam o núcleo (kernel) Linux. O núcleo Linux foi desenvolvido pelo programador finlandês Linus Torvalds, inspirado no sistema operacional Minix. O Linux é conhecido por ser open source (código aberto), ou seja, o código-fonte está disponível para que qualquer pessoa possa utilizá-lo, modificá-lo e distribuí-lo de acordo com os termos da licença da GNU GPL versão 2.[10][11][12]

Atualmente, existem diversas distribuições Linux que servem para propósitos distintos, desde sistemas embarcados (DD-WRT) até computadores pessoais (Ubuntu, Fedora etc). Uma distribuição Linux é definida como um sistema operacional que utiliza o núcleo (kernel) Linux com diversos softwares e utilidades que fazem funcionar o sistema operacional e o fazem ter uma usabilidade melhor. Neste trabalho, utilizou-se as distribuições Linux: Armbian, Mobian e PostmarketOS.

O Armbian é uma distribuição Linux baseada na distribuição Debian ou na distribuição Ubuntu com foco em processadores com arquitetura ARM. Nesse trabalho, utilizou-se a versão do Armbian que é baseada na distribuição Debian. O objetivo do Armbian é tornar a experiência de computadores com processador ARM com o Linux mais uniforme, mantendo a performance com features leves para o hardware.[13] Existe uma comunidade do Armbian com foco em suporte para TV Boxes, que inclui uma discussão para modelos de TV Boxes com processadores Rockchip, que é o caso da TV Box usada. Entretanto, no site, reitera-se que não há suporte oficial do Armbian para TV Boxes.[14] Existe apenas um suporte extremamente limitado na comunidade por meio de discussões.

O Mobian é um projeto em desenvolvimento que tem como objetivo portar a distribuição Debian para dispositivos móveis, como smartphones e tablets. O projeto surgiu em 2020 e é mantido pelos times do Mobian e do DebianOnMobile. O Mobian é responsável pela manutenção de pacotes que não estão inclusos no Debian e geração da imagem e distribuição, enquanto o DebianOnMobile é responsável pela manutenção dos pacotes já inclusos no Debian.

O Mobian utiliza a interface gráfica Phosh, desenvolvida pela Purism e baseada na Gnome, que possui foco na usabilidade para dispositivos móveis. [15] Os dispositivos suportados oficialmente pelo Mobian são Pine64 Pinephone, Pine64 Pinephone Pro, Pine64 PineTab 1 e 2, Purism Librem 5, que são dispositivos móveis feitos para Linux, e de dispositivos Android temos o OnePlus 6/6T e o Pocophone F1. Há alguns outros dispositivos que estão em fase de teste e também há a versão do Mobian para dispositivos com arquitetura x86-64.

O PostmarketOS é uma distribuição Linux com foco em dispositivos móveis baseado na distribuição Alpine Linux, lançada em 2017. Possui uma gama de diferentes interfaces gráficas disponíveis, como a Phosh, a Gnome, a Xfce4, a Plasma Mobile entre outras. Também possui uma quantidade significativa de dispositivos suportados na categoria Main e Community, que vão desde smartphones até placas de desenvolvimento. Há uma quantidade ainda maior de dispositivos que estão em fase de teste, inclusive uma TV Box. Possui uma documentação vasta que pode receber contribuições de qualquer pessoa, já que o projeto se classifica como um FOSS (free and open-source project) e que possui informações sobre dispositivos distintos e problemas enfrentados pelos usuários.[16]

## 5.2 Sistema Operacional Android

O Android é um sistema operacional baseado no núcleo Linux, projetado principalmente para dispositivos móveis, como smartphones e tablets. Sua primeira versão comercial foi lançada em 2008 e, desde então, o Android se tornou o sistema operacional móvel mais utilizado no mundo. Segundo a própria documentação oficial do Android Open Source Project (AOSP), o sistema é um projeto de código aberto que permite a fabricantes, desenvolvedores e comunidades criarem experiências customizadas a partir de sua base.[17]

O Android é construído sobre o núcleo Linux, mas possui modificações para atender às necessidades de dispositivos móveis, incluindo suporte a diversas arquiteturas de processadores, como ARM, x86 e RISC-V.[18][19] Uma das principais características do Android é sua flexibilidade e capacidade de personalização. Fabricantes de hardware, como Samsung, Xiaomi e OnePlus, utilizam o AOSP como base para criarem suas próprias interfaces de usuário, como One UI, MIUI e

OxygenOS, respectivamente. Essa customização inclui alterações no design, na inclusão de funcionalidades específicas e de aplicativos proprietários.

Além de dispositivos móveis, o Android também é amplamente utilizado em dispositivos embarcados, como smart TVs, sistemas automotivos, e TV Boxes, como é o caso do projeto, por meio de soluções como Android TV e Android Automotive OS. Essa versatilidade é possível pelo fato de o Android ser modular, permitindo que diferentes camadas do sistema sejam adaptadas a propósitos específicos.

## 6 Modificando o Sistema Operacional da TV Box

O trabalho foi desenvolvido com as TV Boxes fornecidas pela Receita Federal à Unicamp, do modelo TX2 com chipset Rockchip RK 322 e as seguintes especificações:

- **Chipset:** Rockchip RK3066 1.46 Ghz, Quad-core ARM Cortex-A7
- **Memória RAM:** 2GB DDR3
- **Memória ROM:** 16GB eMMC
- **Sistema:** Android 7.1.2

Os projetos envolvendo TV Boxes já citados, de Ribeiro et al. (2023) e Lopes et al. (2023), apresentaram desafios para encontrar sistemas operacionais compatíveis com a arquitetura ARM do processador do equipamento disponível na Unicamp. Além da compatibilidade de arquitetura, o sistema operacional escolhido deve cumprir com as expectativas de usabilidade para ensino de programação. Usando como referência projetos como o educabox (Rodrigues et al), optou-se inicialmente por instalar uma imagem Armbian compatível com o hardware da TV Box, sistema baseado em Linux (Debian).

Para realizar a instalação do sistema seguiu-se o passo-a-passo de acordo com o projeto de Ribeiro et al. (2023) baseado no tutorial criado pela equipe do Smart Campus UNICAMP[20], seguindo os passos indicados na seção *Metodologia de instalação nas TV Boxes*.

Com o sistema instalado, a configuração do Armbian foi utilizada, rodando o comando `sudo armbian-config`, selecionando a opção Software, depois Default e então xfce para instalar uma interface gráfica. Optou-se pela xfce por ter um bom custo-benefício entre desempenho e usabilidade. Essa interface apresenta um sistema desktop amigável, apesar de apresentar problemas como ícones mal posicionados ou sobrepostos e falta de intuitividade. Porém, realizando testes com o Armbian e a interface xfce para tarefas simples e direcionadas à nossa proposta, como criação de arquivo, uso do terminal e navegação na internet, o sistema apresentou baixo desempenho, levando muito tempo para abrir aplicações e responder a comandos e quebrando a aplicação em execução esporadicamente, como ao usar o navegador.

Devido aos problemas encontrados com o Armbian, realizou-se uma nova tentativa, mantendo-se o sistema Android nativo da TV Box. Além de apresentar desempenho satisfatório, o Android tem compatibilidade com o hardware e muitas opções de aplicativos para atender diferentes demandas, como navegador, editor de texto, IDEs com compilação para diferentes linguagens de programação, dentre outros.

## 6.1 Mantendo o sistema Android nativo

Para atender aos objetivos mantendo-se o sistema Android nativo da TV Box, a opção escolhida foi a de instalar um launcher no sistema que emulasse a interface gráfica de um computador desktop, com menu iniciar, janelas de aplicativos redimensionáveis, gerenciador de arquivos semelhante aos do Linux e Windows e boa usabilidade com periféricos (teclado e mouse). Esta opção pareceu promissora dada a simplicidade do processo, mantendo-se a performance observada no sistema Android e a possibilidade de criação de uma ROM para o dispositivo utilizando este launcher como padrão no caso de um resultado satisfatório.

O launcher escolhido foi o Sentio Desktop, que apresenta uma interface intuitiva semelhante a de sistemas operacionais desktop. O launcher tem ferramentas próprias, como gerenciador de arquivos e menu iniciar, e permite o uso de múltiplas janelas redimensionáveis.

O launcher foi utilizado juntamente com navegador, editor de texto e aplicativos de ensino de programação e terminal python para simular o cenário proposto pelo objetivo do trabalho. Estes aplicativos foram baixados pela Play Store e utilizados simultaneamente nos testes, que consistiam de realizar uma atividade proposta pelo app de ensino de programação no terminal python, realizando pesquisas no navegador e documentando o progresso no editor de texto.

É possível ver na Figura 3 uma imagem do Sentio sendo usado em um dispositivo Android, com múltiplas janelas redimensionáveis, menu iniciar e barra de tarefas.

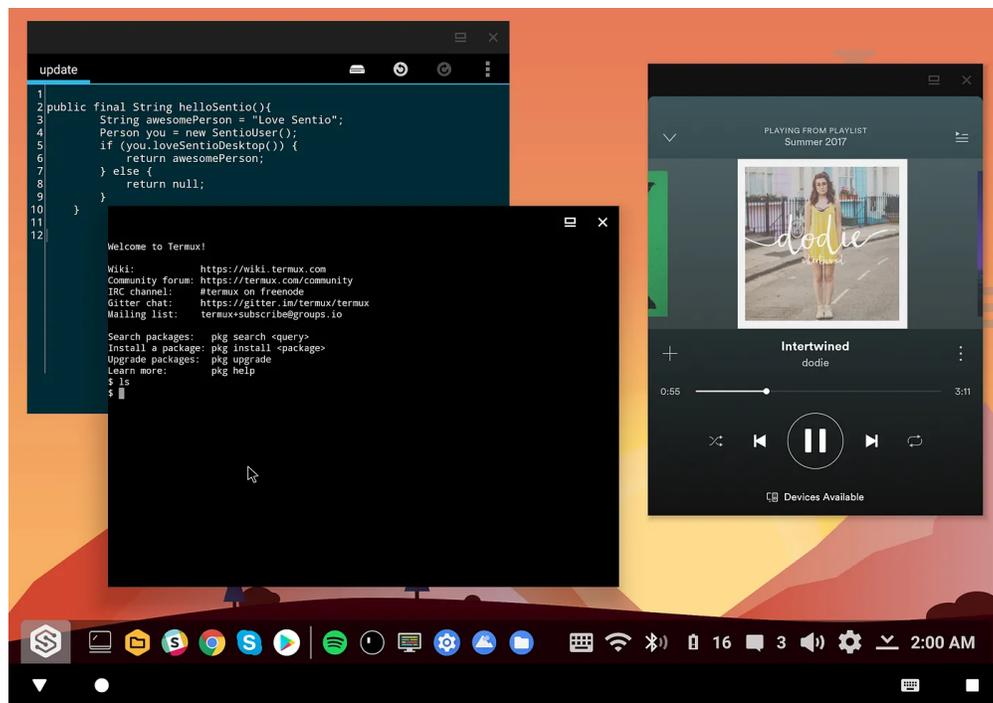


Figura 3: Sentio Desktop em dispositivo Android.[21]

A Figura 3 demonstra o sistema, que foi apresentado na disciplina de MC853 - Projeto em Sistemas de Programação da Unicamp no segundo semestre de 2023 como proposta de uso da TV Box. Além disso, o projeto foi apresentado na 19ª edição do programa Unicamp de Portas Abertas (UPA), em 2024, na qual visitantes da Unicamp puderam utilizar o sistema e se familiarizar com sua implementação e objetivos futuros do projeto.

O teste se mostrou possível de ser realizado, com todos os aplicativos funcionando. No entanto,

o desempenho não foi satisfatório. A navegação entre apps foi lenta, sites pesados não finalizaram o carregamento no navegador e a experiência é limitada pela falta de um terminal que possibilite configurações avançadas e uma experiência desktop nativa. Além disso, o uso de uma aplicação proprietária se mostrou limitado e pouco escalável, uma vez que a distribuição é de propriedade privada. Possíveis melhorias e personalização ficam a cargo da empresa proprietária do software e não de uma comunidade livre.

Estes fatores combinados ajudaram a determinar que o melhor direcionamento para o trabalho seria a utilização de um sistema operacional open-source compatível com a TV Box e com possibilidade proporcionar uma experiência desktop, tanto no que diz respeito a possibilidades de programação, quanto em interface gráfica e experiência do usuário. Assim, buscou-se novamente sistemas baseados em Linux, como o Armbian, que pudessem nos oferecer essas vantagens juntamente com um melhor desempenho de usabilidade.

## 6.2 Sistema open-source

A busca por alternativas de sistemas operacionais open-source compatíveis com o processador ARM da TV Box que trouxessem ao usuário uma experiência desktop apresentou algumas possibilidades. Dentre elas, os sistemas Ubuntu Touch, Mobian e PostmarketOS se mostraram os mais promissores para os objetivos do projeto. Os três sistemas são baseados em Linux e focados em dispositivos ARM, como smartphones, oferecendo uma experiência leve e fluida, com o máximo de compatibilidade.

Inicialmente, o Ubuntu Touch foi escolhido por apresentar um sistema mais polido do que as outras opções, além de uma vasta documentação e possibilidades interessantes para o projeto. O sistema, apesar de projetado para smartphones, tem a possibilidade de uso na versão desktop, que redimensiona o sistema e habilita menu iniciar, janelas redimensionáveis, dentre outras funcionalidades. Além disso, é possível rodar comandos e instalar aplicativos, assim como no sistema Ubuntu, por meio das configurações.

Como a quantidade de imagens disponíveis do Ubuntu Touch é limitada e somente para alguns aparelhos específicos, a forma de criar uma imagem compatível com a TV Box é por meio do processo de portabilidade, que permite a criação de uma imagem compatível com o hardware desejado. Como especificado pelo guia de portabilidade do Ubuntu Touch, para a criação da imagem, dadas as especificações do hardware, o método é o de *Full system image method*, que necessita do uso de uma imagem preexistente (LineageOS) compatível com o hardware. Para a TV Box não foi possível encontrar esta imagem e como o processo de portabilidade é longo e incerto, essa possibilidade foi descartada e os outros sistemas citados foram considerados.

Os principais desafios na criação de imagens personalizadas geralmente envolvem compatibilidade de drivers (em caso de smartphones, geralmente câmeras, antenas, sensores etc) e compatibilidade do kernel. Como a TV Box tem interfaces limitadas, sem telas, câmeras ou sensores, o principal desafio do projeto foi com o kernel. Neste sentido, para seguir com as demais opções de sistemas, uma das estratégias foi a de instalar novamente o Armbian, sistema completamente compatível com a TV Box (drivers e kernel) e criar imagens sobrescrevendo seu sistema de arquivos, de forma a aproveitar essa compatibilidade. Desta forma, seguiu-se para o Mobian e PostmarketOS, respectivamente.

Para testar o Mobian, o primeiro caminho escolhido foi realizar a instalação do sistema em um cartão SD utilizando o Armbian. Para isso, realizou-se o seguinte passo-a-passo:

1. Inserção do cartão na TV Box (com o Armbian já instalado).
2. Verificação se o cartão SD foi detectado (comando `lsblk`).

3. Criação de uma nova partição e formatação.
4. Montagem da partição criada do cartão SD.
5. Download de uma imagem do Mobian na TV Box diretamente do site de imagens do Mobian: <https://images.mobian.org/>. O arquivo .gz da imagem mais recente dentro da pasta “rockchip” foi escolhido, já que o processador é um Rockchip. Na primeira tentativa de instalação, utilizou-se o arquivo que é installer (mobian-installer-pinephonepro-phosh-12.0.img.gz). No entanto, este arquivo serve somente para instalar o Mobian diretamente em um dispositivo, não sendo adequado para a finalidade desejada.
6. Descompactação e cópia da imagem para a partição montada do cartão SD.
7. Sincronização do cartão SD com o comando sync para garantir que os dados haviam sido copiados corretamente.
8. Desmonte da partição do cartão SD e reinicialização da TV Box para que o dispositivo ligasse e fizesse o boot a partir do cartão SD.

Após a etapa 8 a TV Box inicia no sistema do Armbian, ao invés de iniciar no sistema Mobian instalado no cartão SD. Outra abordagem foi escolhida após algumas tentativas tentando iniciar a TV Box por meio do cartão SD. Uma das tentativas foi a alteração direta no boot.cmd do Armbian para que iniciasse utilizando o cartão SD, o que ocasionou um erro em que o processo de inicialização não conseguia inicializar no sistema de arquivos (erro: init not found).

A outra abordagem, ainda tentando utilizar o Mobian, foi a de sobrescrever o sistema de arquivos do Armbian com o sistema de arquivos do Mobian, mantendo o kernel e drivers do Armbian, já que funcionam perfeitamente. Esta abordagem surgiu devido ao problema de incompatibilidade do sistema operacional Mobian com a TV Box, visto que não havia uma porta direta para o dispositivo. Ao instalar diretamente o Mobian, provavelmente seriam encontrados diversos problemas, especialmente com o kernel e drivers essenciais como áudio, rede, dentre outros. Por isso, acreditou-se que faria sentido realizar a sobrescrita do sistema de arquivos, para que fosse possível utilizar de recursos gráficos do Mobian, com o kernel e drivers do Armbian na camada abaixo. Este foi o passo-a-passo realizado:

1. Download da imagem do Mobian do mesmo modo descrito na abordagem anterior, tendo atenção aos detalhes supracitados.
2. Montagem da partição da imagem correspondente a raiz do Linux. Na imagem, havia duas partições: uma referente ao boot e outra referente ao sistema de arquivos Linux, que era a de interesse.
3. Verificação da montagem, avaliando se foi bem sucedida, verificando se havia arquivos essenciais como fstab e pastas como /usr/bin, típicas de sistemas operacionais baseados em Linux.
4. Montagem da partição raiz do Armbian e utilização do comando rsync para copiar os arquivos do Mobian para o Armbian, tomando o cuidado de manter as mesmas permissões.

Após o último comando, observou-se alguns erros durante a cópia e, ao reiniciar a TV Box, o sistema não iniciou novamente. Houve mais algumas tentativas e também foi realizada a tentativa de instalar o Mobian diretamente pelo Multitool, e então verificou-se que o problema era que a

arquitetura do processador Rockchip para o qual a versão utilizada do Mobian foi feita (PinePhone Pro) era aarch64 (64-bit), enquanto a arquitetura do Rockchip da TV Box era armv7 (32-bit), o que gerou uma incompatibilidade ao copiar o Mobian sobre o Armbian.[22] Como não havia imagem do Mobian compatível com a arquitetura armv7, decidiu-se tentar outro sistema operacional para realizar a operação de sobrescrita, que ainda parecia promissora, dado que os problemas enfrentados eram relacionados ao Mobian e não ao processo de sobrescrita.

O próximo sistema operacional escolhido foi o PostmarketOS, que parecia promissor dada a sua extensa documentação e compatibilidade com diversos dispositivos. Infelizmente, este sistema operacional também não tinha uma porta diretamente para o modelo de TV Box utilizado, nem para o processador Rockchip, então buscou-se a imagem genérica que poderia se adequar melhor ao modelo.

Pela página do PostmarketOS é possível ver quais dispositivos tem imagens disponíveis e se elas já possuem um bom suporte (categorias main e community) e as que ainda estão em fase de teste (categoria testing). Também, pelo mesmo site, é possível checar as especificações técnicas de cada dispositivo, como arquitetura, processador etc.[23] A primeira opção que foi considerada foi a porta da TV Box genérica da AmLogic, porém, ao checar as especificações, a arquitetura dela é aarch64, o que geraria o mesmo problema que ocorreu com o Mobian.[24] Então, buscou-se imagens genéricas de dispositivos que possuíam um processador Rockchip e que tivessem arquitetura armv7, priorizando dispositivos que oferecessem um suporte melhor, ou seja, que não estivessem em fase de teste. A segunda opção que foi considerada, levando em conta os requisitos listados, foi a imagem genérica para o Chromebook Veyron, que possui o Rockchip RK3288 e arquitetura armv7, e, como não parecia haver incompatibilidades até o momento, esta imagem foi a escolhida para se seguir com o processo.[25]

Para realizar o processo de sobrescrita, seguiu-se, primeiramente, a sugestão do guia de instalação do PostmarketOS e a imagem foi gerada por meio do pmbootstrap, uma ferramenta própria de build do PostmarketOS. O processo foi inteiramente feito na TV Box. Para instalar o pmbootstrap no dispositivo foi necessário clonar o repositório, pois a versão do pmbootstrap disponível para instalação direta no Armbian é muito antiga e não pode mais ser utilizada. O passo-a-passo para geração da imagem foi seguido conforme o tutorial do PostmarketOS, disponível neste link: [https://wiki.postmarketos.org/wiki/Installation/Using\\_pmbootstrap](https://wiki.postmarketos.org/wiki/Installation/Using_pmbootstrap). Durante a geração da imagem, deixou-se tudo da forma padrão e sugerida pelo tutorial, e escolheu-se a imagem sem interface gráfica (console) para evitar maiores complicações, dado que a interface gráfica poderia ser instalada posteriormente.

Com a imagem já gerada no próprio dispositivo, seguiu-se para a montagem da partição da imagem correspondente a raiz do Linux, como feito anteriormente com o Mobian, porém, descobriu-se que a imagem do Google Veyron Chromebook não possui uma partição raiz do Linux, somente do Chrome OS. Pensou-se que poderia ser por conta do pmbootstrap e, então, houve a tentativa de baixar a imagem pré-buildada disponível no site do PostmarketOS, porém esta imagem também apresentou as mesmas partições. Assim, houve a necessidade de escolher outra imagem, visto que não seria possível realizar o processo de sobrescrita entre sistema de arquivos de dois sistemas operacionais distintos.

A terceira e última opção foi a imagem genérica da Nvidia Tegra, que, apesar de não ter um processador Rockchip, era a única outra porta genérica com suporte da comunidade que possuía arquitetura armv7.[26] Para facilitar o processo, como não foi visualizado haver distinção entre as imagens pré-buildadas e as imagens geradas pelo pmbootstrap, seguiu-se pelo caminho de baixar a imagem pré-buildada diretamente do site do PostmarketOS: <https://images.postmarketos.org/bpo/>. Utilizou-se a versão estável 24.06 sem interface gráfica (console). O passo-a-passo de sobrescrita realizado foi idêntico ao descrito anteriormente, tomando o cuidado, ao realizar a cópia

do sistema de arquivos, para não sobrescrever a pasta de inicialização (boot) do Armbian, já que não havia a intenção de mudar o kernel e os drivers.

Após a primeira tentativa de cópia, ao reiniciar o sistema, ele iniciou no PostmarketOS, utilizando o kernel do Armbian, com alguns problemas referentes ao clock, à interface sem fio e à nftables. Porém, o principal problema nesta primeira tentativa, foi que, ao tentar executar qualquer comando, ocorre o erro que o sistema de arquivos é somente de leitura (read-only filesystem). Após a investigação do problema, descobriu-se que está relacionado ao arquivo fstab, arquivo de configuração responsável por listar todos os discos e partições e indicar como eles devem ser inicializados e/ou integrados ao sistema de arquivos. O arquivo fstab deve ser mantido original do Armbian para evitar este erro de permissão.

Nas próximas tentativas, o foco foi conectar a TV Box em uma rede sem fio (Wi-fi) para que fosse possível realizar a instalação de uma interface gráfica. Apesar de não haver nenhum problema com o Network Manager, responsável por gerenciar as conexões de rede do sistema, ele não conseguia encontrar a interface de rede sem fio, o que o fazia não encontrar nenhuma rede sem fio disponível e, portanto, não ter acesso à internet. Houve diversas tentativas para manter a interface de rede sem fio do Armbian, mantendo diversos arquivos originais do Armbian referentes às pastas de firmware e network, e até mesmo tentando manter o Network Manager do Armbian, mas todas falharam em resolver o problema de conexão à rede sem fio do PostmarketOS.

Após ter falhado com a imagem pré-buildada sem interface gráfica (console), houve a tentativa de realizar a operação de sobrescrita com uma imagem pré-buildada que já possuísse interface gráfica, para tentar contornar o problema de não ser possível a instalação da interface gráfica e na esperança de que o problema de conexão à rede sem fio não ocorresse nessa versão do PostmarketOS. A primeira tentativa de sobrescrita foi com o Armbian sem ter interface gráfica já instalada, tomando todos os cuidados supracitados, porém, com essa configuração, o sistema não inicia. Pensou-se que a razão poderia ser a falta de uma interface gráfica instalada no Armbian, então, repetiu-se o processo com o Armbian com uma interface gráfica instalada (xfce). Porém, obteve-se o mesmo resultado, em que o sistema não iniciou.

Pensando que estes problemas poderiam ter relação com o fato de a imagem ser pré-buildada, realizou-se a geração de 2 imagens, uma com interface gráfica xfce e a outra sem. Entretanto, ao realizar o processo com a imagem com interface gráfica, houve o mesmo erro anterior, e ao realizar o processo com a imagem sem interface gráfica, voltou-se a ter o erro de sistema de arquivos de leitura (read-only) mesmo com a manutenção do arquivo fstab original do Armbian. Houve uma segunda tentativa sobrescrevendo o fstab do Armbian com o do PostmarketOS, mas obteve-se o mesmo erro. Após todas estas tentativas, ficou claro que este caminho de sobrescrita não funcionaria, pois, mesmo havendo compatibilidade entre esta imagem do PostmarketOS e o Armbian, os problemas que surgiram não são podem ser solucionados de forma trivial ou ignorados. Portanto, a partir destas tentativas falhas, voltou-se a estudar o Armbian e alternativas para otimizá-lo.

### 6.3 Uso do Armbian

Por fim, o Armbian foi instalado novamente, seguindo os passos citados anteriormente, mas sem a instalação de uma interface gráfica default do armbian-config. Ao invés disso, optou-se por buscar interfaces gráficas que atendessem aos objetivos do projeto com desempenho superior na utilização de aplicações, sem comprometer a usabilidade. Esta busca apontou para a interface LXDE, extremamente leve, criada com o objetivo de performar em uma vasta quantidade de hardwares, apresentando possibilidades de personalização e uma experiência amigável ao usuário.[27]

Para realizar a instalação da LXDE, os seguintes passos foram seguidos:

1. Garantir que o sistema está atualizado, utilizando o comando `sudo apt-get update`
2. Instalar a interface LXDE, com o comando `sudo apt-get install -y xorg lxde lightdm policykit-1`
3. Reiniciar o sistema, usando o comando `sudo reboot`

Com a interface gráfica instalada, foram realizados testes para verificar se tudo estava funcionando corretamente e entender as possibilidades de personalização do sistema. Foi necessário realizar o ajuste do teclado para o padrão pt-BR, já que por padrão, a configuração do teclado é en-US. Para isso, executou-se os seguintes passos:

1. Reconfiguração do layout do teclado com o comando: `sudo dpkg-reconfigure keyboard-configuration`
  - (a) Escolher o modelo apropriado do teclado, no Brasil, o padrão é: teclado de 105 teclas (opção: Generic 105-key PC).
  - (b) Selecionar o layout pt-BR (Portuguese-Brazilian).
  - (c) Confirmar as alterações.
2. Atualizar as configurações com o comando: `sudo setupcon`
3. Verificar se o arquivo `/etc/default/keyboard` tenha as seguintes linhas:
  - `XKBMODEL="pc105"`
  - `XKBLAYOUT="br"`
  - `XKBVARIANT=`
  - `XKBOPTIONS=`
4. Reiniciar o serviço de teclado com o comando: `sudo service keyboard-setup restart`
5. Reiniciar o sistema.

Percebeu-se também que uma performance superior em relação a velocidade das aplicações era obtida ao usar a resolução de 1024x768 pixels, com taxa de atualização de 60 Hz. Ademais, foi fácil personalizar a barra de tarefas e área de trabalho por meio das configurações.

Para atender aos objetivos do projeto, foram buscados navegadores e IDEs para programação em python. As principais alternativas de navegadores foram o Midori e o Falkon, que são leves e compatíveis com o sistema. Após instalar os navegadores, concluiu-se que o Falkon apresentava melhor usabilidade. Além disso, avaliou-se as IDEs Thonny e Geany e optou-se pelo uso do Thonny por ser mais intuitivo, apresentar terminal próprio para execução e bom desempenho.

A Figura 4 apresenta a interface LXDE no sistema Armbian da TV Box, com múltiplas aplicações abertas.

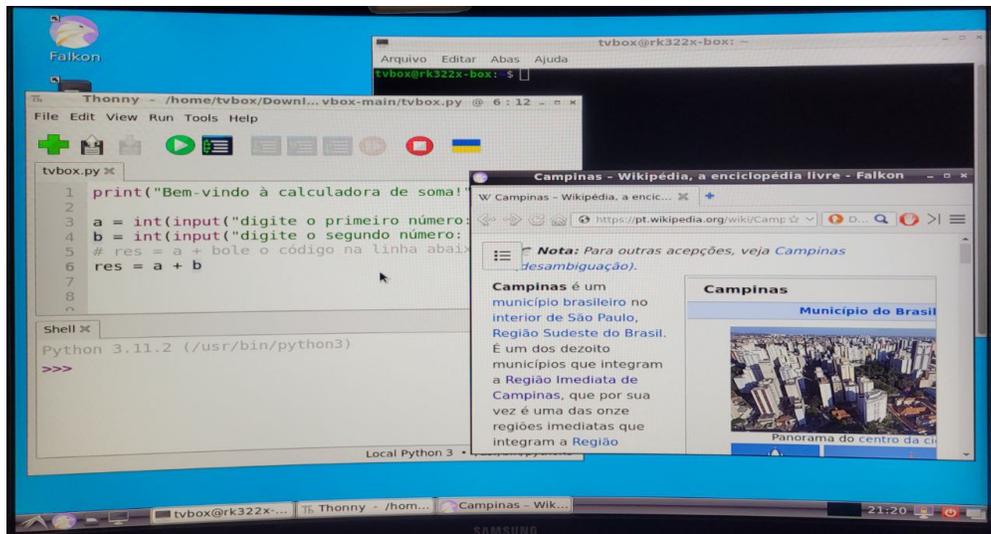


Figura 4: TV Box com sistema Armbian e interface LXDE

Por fim, utilizando as ferramentas instaladas juntamente com as do sistema, como terminal, gerenciador de arquivos e editor de texto, o desempenho foi superior a todas as outras opções instaladas. Todos os aplicativos responderam bem e foi possível realizar atividades de programação sem muitas dificuldades, fazendo pesquisas e rodando o código, tanto via IDE quanto via terminal. Assim, seguiu-se para a realização de testes com nosso público alvo para validar a solução.

## 7 Resultados

A abordagem que obteve melhores resultados foi o uso do Armbian com uma nova interface mais leve, a LXDE. Utilizando esta abordagem, chegou-se à configuração final da TV Box. Essa configuração é voltada para o ensino de programação em escolas, com a instalação de aplicações como IDEs e navegadores. Assim, realizaram-se testes de usabilidade para entender se a configuração da TV Box alcança resultados satisfatórios e pode ser replicada futuramente em projetos sociais e nas escolas municipais e estaduais.

O teste de usabilidade foi desenvolvido a partir do que poderia ser usado em aulas de programação em geral e também, especificamente, nas aulas de programação de um projeto social da Unicamp chamado Olá Mundos.

O Olá Mundos é um projeto criado pela entidade estudantil Conpec, empresa júnior de Computação da Unicamp. O objetivo do projeto é democratizar o ensino de programação, com foco em jovens de baixa renda. Atualmente, o Olá Mundos conta com aulas de programação em escolas públicas de Campinas, em que os alunos de graduação dos cursos de Engenharia e Ciência da Computação que fazem parte da Conpec ministram aulas introdutórias de programação em Python. Infelizmente, o projeto acaba sofrendo alguns impactos devido à falta de estrutura das escolas e ao custo elevado de computadores e notebooks, não possuindo dispositivos próprios para emprestar aos alunos e limitando-se a estrutura da escola em que lecionam. Por isso, decidiu-se testar se as TV Boxes poderiam funcionar como computadores de baixo custo para o projeto, para que no futuro estes dispositivos possam ser utilizados pelo Olá Mundos e em projetos semelhantes.[28]

O teste foi composto por duas etapas: a primeira sugere a execução de tarefas gerais e a segunda de tarefas relacionadas à programação na TV Box. As tarefas de programação foram criadas visando possibilitar que qualquer estudante de Ensino Médio fosse capaz de executá-las

sem grandes dificuldades, independentemente de ter tido contato com programação anteriormente. A primeira etapa do teste consiste em três tarefas: encontrar a calculadora no sistema, abrir uma imagem localizada na pasta denominada “Downloads”, realizar uma pesquisa no navegador e, a partir do resultado, abrir a Wikipedia. Esta fase possui instruções mais gerais, com o objetivo de que os voluntários se familiarizem com o sistema e possam explorá-lo. A segunda etapa consiste em duas tarefas: a primeira é encontrar o arquivo “hello\_world.py”, abrir o terminal e executar o código através do comando “python3 hello\_world.py” e a segunda é acessar um repositório do GitHub, realizando o download do arquivo zipado e extraíndo-o, após a extração, abrir o arquivo “tvbox.py” e, após aberto, abrir novamente o navegador para copiar uma linha de código do arquivo “README.md” e colar na linha indicada no arquivo “tvbox.py”. Nesta etapa, há instruções bem específicas e detalhadas pois, como este de fato é o teste de usabilidade, não deveria haver interferências externas, pois o objetivo é entender se os usuários são capazes de realizar as tarefas sem auxílio. Por último, os voluntários tinham um tempo extra para explorar o sistema.

O arquivo com as instruções para cada etapa foi disponibilizado aos voluntários nas próprias TV Boxes durante todo o período de realização do teste. Com o arquivo já aberto no dispositivo, o objetivo era que o aluno voluntário fosse capaz de seguir todas as instruções, realizando todas as tarefas e, ao final, preencher um formulário de feedback sobre o sistema.

O formulário é anônimo e composto por 3 seções: a primeira referente às tarefas gerais, a segunda às tarefas de programação, a terceira referente ao feedback geral sobre o sistema, além de uma seção de perguntas sobre o participante. As perguntas que compõe a primeira seção do formulário são as seguintes:

- Em uma escala de 0 a 5, sendo 0 muito fácil e 5 muito difícil, qual seu nível de dificuldade em encontrar a calculadora?
- O que poderia ser melhor na busca?
- Em uma escala de 0 a 5, sendo 0 muito fácil e 5 muito difícil, qual seu nível de dificuldade em encontrar a pasta e o arquivo (maca.jpg)?
- O que você achou da organização das pastas?
- Em uma escala de 0 a 5, sendo 0 muito fácil e 5 muito difícil, qual seu nível de dificuldade em realizar a pesquisa e abrir o site?
- O que achou da barra de pesquisa e da organização do navegador durante a tarefa?

Esta seção de perguntas tem como objetivo avaliar, em geral, a execução das tarefas gerais, que eram mais simples, coletando um feedback quantitativo e qualitativo, para entender de forma completa se houve dificuldades e quais foram.

A segunda seção do formulário é composta pelas seguintes perguntas:

- Quais etapas do primeiro teste (hello\_world) você conseguiu executar?
  - Opções de resposta (multi-seleção): Abrir o Gerenciador de arquivos, Identificar o arquivo “hello\_world.py”, Abrir o terminal, Rodar o comando, Todas acima;
- Em uma escala de 0 a 5, sendo 0 muito fácil e 5 muito difícil, qual seu nível de dificuldade para realizar o primeiro teste (hello\_world)?
- Quais dificuldades teve ao realizar o primeiro teste (hello\_world)?

- Quais etapas do segundo teste (calculadora de soma) você conseguiu executar?
  - Opções de resposta (multi-seleção): Visitar o endereço no navegador, Baixar o arquivo zipado, Extrair o arquivo zipado, Abrir o arquivo "tvbox.py", Colar o comando do site no arquivo, Salvar o arquivo, Rodar o programa, Testar a calculadora, Todas acima;
- Em uma escala de 0 a 5, sendo 0 muito fácil e 5 muito difícil, qual seu nível de dificuldade para realizar o segundo teste (calculadora de soma)?
- Quais dificuldades teve ao realizar o segundo teste (calculadora de soma)?

O foco desta seção é entender quais foram as dificuldades encontradas ao realizar tarefas simples de programação, buscando medir quantos voluntários conseguiram realizar o teste de forma completa.

A última seção de perguntas referentes ao feedback do teste pede feedbacks gerais sobre o sistema, e é composta pelas seguintes perguntas:

- O que você achou da aparência geral da interface (ícones, cores, menus)?
- Foi fácil encontrar as coisas que você precisava? Alguma coisa foi difícil de entender?
- Se você pudesse mudar algo na interface, o que seria?
- Comparado ao notebook que você usa normalmente na escola, qual sistema você prefere? Por quê?
- Se você tivesse que usar esse sistema para estudar, acha que seria fácil?
- Você gostaria de usar esse sistema para estudar, fazer tarefas de programação?

Estas perguntas buscam avaliar como os voluntários se sentiriam se fossem usar este sistema na escola para estudos e tarefas de programação, entendendo a opinião geral sobre a aparência e usabilidade do sistema.

Por último, foram realizadas as seguintes perguntas para entender o nível de familiaridade dos participantes com dispositivos eletrônicos e com programação:

- Você já tinha acessado um computador antes?
  - Opções de resposta (escolha única): Sim, na escola.; Sim, tenho computador em casa.; Sim, na casa de alguém.; Sim, em uma biblioteca ou universidade.; Sim, em outro lugar.; Não.; Prefiro não responder;
- Quais dispositivos você tem acesso em casa?
  - Opções de resposta (multi-seleção): Computador; Celular; Tablet; Smart TV; Outros.; Prefiro não responder;
- Você tem celular?
  - Opções de resposta (escolha única): Sim, é o meu dispositivo pessoal.; Sim, mas eu divido com irmãos ou pais.; Não.; Prefiro não responder;
- Você tem acesso à internet em casa?

- Opções de resposta (escolha única): Sim; Não.; Prefiro não responder;
- Qual seu nível de conhecimento em programação?
  - Opções de resposta (escolha única): Não sei o que é programação.; Nunca programei.; Já programei coisas simples.; Tenho muita experiência com programação.;
- Qual sua idade?
- Qual seu gênero?

Com as respostas do formulário, obteve-se os gráficos relacionados ao nível de dificuldade em executar as tarefas gerais, atribuindo um valor em uma escala de 0 a 5. A Figura 5 refere-se aos níveis de dificuldade dos estudantes em encontrar o software de calculadora no sistema operacional.

Em uma escala de 0 a 5, sendo 0 muito fácil e 5 muito difícil, qual seu nível de dificuldade em encontrar a calculadora?

5 respostas

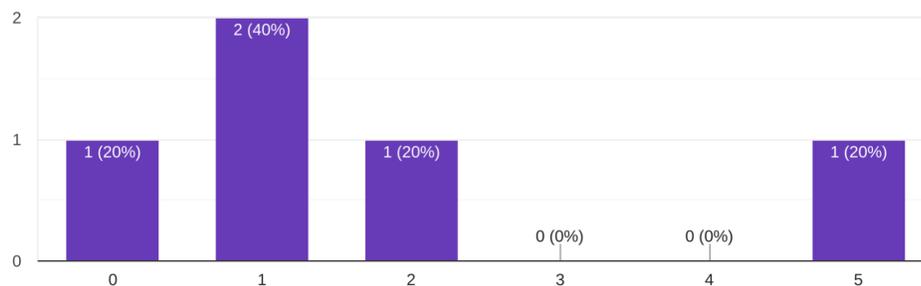


Figura 5: Gráfico de distribuição de respostas sobre dificuldade de encontrar aplicação de calculadora

Na Figura 5, visualiza-se que a maioria dos participantes teve facilidade com a tarefa de encontrar a calculadora, atribuindo níveis 0, 1 e 2. De acordo com as respostas na pergunta dissertativa, as dificuldades se concentraram na localização da aplicação, que poderia ter uma visibilidade maior em um atalho na área de trabalho, por exemplo. Uma pessoa também mencionou, como uma dificuldade, a nomenclatura do aplicativo da calculadora, que nesse sistema se chama Galculator, o que é diferente de outros sistemas operacionais.

A Figura 6 apresenta a escala de dificuldade dos estudantes em encontrar a pasta Downloads e o arquivo de imagem.

Em uma escala de 0 a 5, sendo 0 muito fácil e 5 muito difícil, qual seu nível de dificuldade em encontrar a pasta e o arquivo (maca.jpg)?

5 respostas

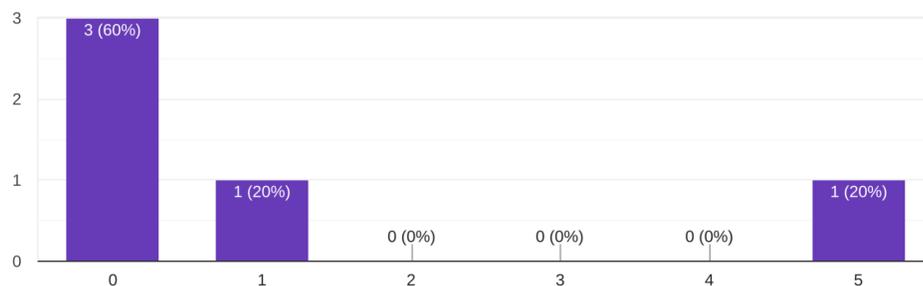


Figura 6: Gráfico de distribuição de respostas sobre dificuldade de encontrar pasta e arquivo .jpg

Os resultados da Figura 6 demonstram que, em relação à tarefa de encontrar a imagem no gerenciador de arquivos, os alunos atribuíram níveis de dificuldade próximos de zero. Em suas respostas, os alunos mostraram gostar da organização das pastas, um deles mencionou que seguia o mesmo padrão de outros sistemas operacionais. Um dos alunos mencionou que teve dificuldade somente na escolha da aplicação para abrir o arquivo da imagem, provavelmente pela falta de familiaridade com as aplicações do Linux voltadas à visualização de imagens.

Na Figura 7, é possível visualizar a escala de dificuldade dos estudantes em realizar uma pesquisa no navegador e abrir um resultado.

Em uma escala de 0 a 5, sendo 0 muito fácil e 5 muito difícil, qual seu nível de dificuldade em realizar a pesquisa e abrir o site?

5 respostas

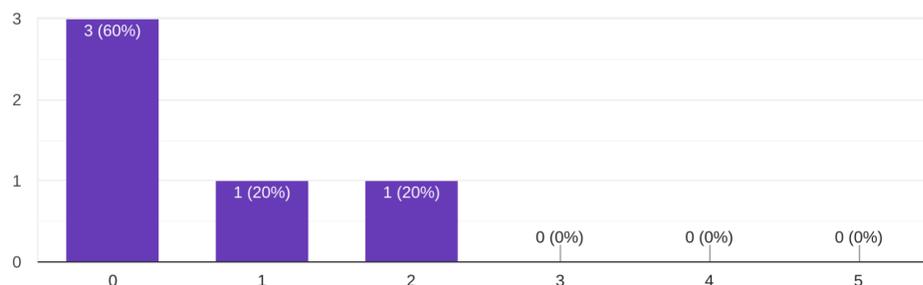


Figura 7: Gráfico de distribuição de respostas sobre dificuldade em realizar pesquisa no navegador

A Figura 7 revela que na tarefa envolvendo o navegador, os voluntários novamente não tiveram dificuldades e atribuíram níveis 0, 1 e 2 de dificuldade. Além disso, verifica-se que os estudantes acharam o navegador Falkon similar aos demais navegadores por eles utilizados. Uma das respostas discursivas mencionou a lentidão do navegador como um empecilho para realizar a tarefa rapidamente, o que era esperado pois o navegador é a aplicação mais pesada da TV Box. Apesar de o

Falkon ser um navegador mais leve, comparado a outros como Firefox e Chrome, ainda assim é um pouco pesado devido à memória RAM e CPU limitados do hardware do dispositivo.

O gráfico da Figura 8 refere-se à dificuldade dos estudantes em rodar o programa “hello\_world.py” na tarefa de programação.

Em uma escala de 0 a 5, sendo 0 muito fácil e 5 muito difícil, qual seu nível de dificuldade para realizar o primeiro teste (hello\_world)?

5 respostas

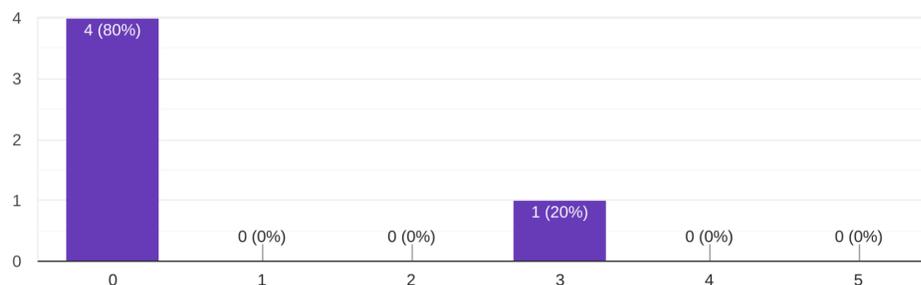


Figura 8: Gráfico de distribuição de respostas sobre dificuldade em rodar programa "hello\_world.py" @

Figura 8: Gráfico de distribuição de respostas sobre dificuldade em rodar programa "hello\_world.py" @

O gráfico da Figura 8 indica que a maioria dos alunos voluntários teve facilidade na tarefa de identificar e executar o código no terminal. Somente um dos alunos apresentou uma dificuldade maior nesta primeira tarefa, não conseguindo executar as tarefas de abrir o terminal e rodar o comando para executar o código.

Na Figura 9, é possível visualizar a distribuição de dificuldade dos estudantes em editar e rodar o programa "tvbox.py".

Em uma escala de 0 a 5, sendo 0 muito fácil e 5 muito difícil, qual seu nível de dificuldade para realizar o segundo teste (calculadora de soma)?

5 respostas

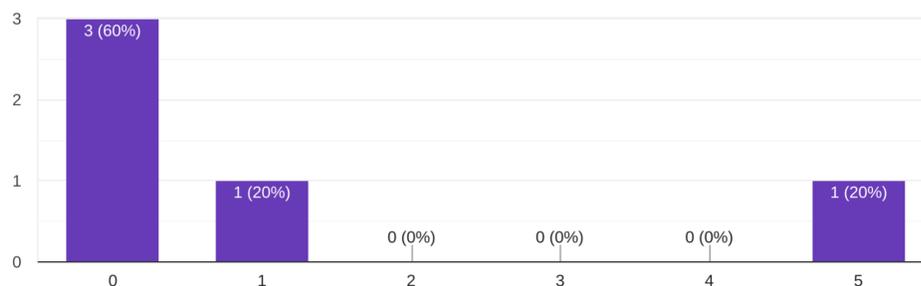


Figura 9: Gráfico de distribuição de respostas sobre dificuldade em editar e rodar o programa "tvbox.py"

Os resultados observados na Figura 9 demonstram que os participantes tiveram facilidade em realizar a tarefa, mesmo sendo uma tarefa um pouco mais complexa. Nesta tarefa, 1 dos 5 alunos não conseguiu completar a tarefa, conseguindo executar apenas o passo de abrir o arquivo de código.

Realizando a análise das respostas, observou-se que o voluntário que teve maior dificuldade nas tarefas de programação, não conseguindo finalizá-las, era o único que nunca havia programado, o que indica que as instruções poderiam estar mais claras e que seria melhor ter explicado melhor o teste, a fim de garantir o entendimento também de pessoas que não tiveram contato com a programação.

No quesito de feedbacks gerais sobre a interface, os usuários do teste demonstraram gostar da aparência geral do sistema e sugeriram mudanças em alguns detalhes, como barra de tarefas, deixar os ícones mais aparentes no sistema, por exemplo, adicionando à área de trabalho e mudar os nomes dos aplicativos. Com exceção do último, todas as mudanças são praticáveis no sistema, principalmente as relacionadas à barra de tarefas, que é inteiramente customizável e de fácil configuração.

Sobre a usabilidade, os voluntários, em geral, relataram poucas dificuldades, sendo a dificuldade maior encontrar a calculadora, devido à falta de familiaridade com o sistema. Um dos voluntários relatou que foi bastante difícil e que poderia ter mais instruções, o que parece ser uma consequência maior da forma com que as instruções foram apresentadas do que uma dificuldade com o sistema em si.

Nas perguntas finais referentes a entender se o uso do sistema seria adequado à escola, para a finalidade de estudo, os estudantes, em geral, acreditam que não seria muito fácil utilizar o sistema para este fim. Dois estudantes responderam que achariam fácil, um deles mencionou que seria fácil somente com a melhora da velocidade do navegador. Entretanto, para a finalidade de estudo de programação em específico, os alunos, de modo geral, demonstraram que gostariam de usar este sistema. Somente uma pessoa afirmou que para quem não tem familiaridade com programação, não seria ideal e que só conseguiu executar as tarefas com o auxílio das instruções. Este feedback mostra que o ideal ao utilizar a TV Box como computador seria realizar um tutorial breve e experimentos para que os alunos tenham familiaridade com o sistema antes de utilizá-lo em aula.

Por fim, é importante salientar que o teste foi realizado com 5 alunos somente, todos entre 16 e 17 anos e de ambos os gêneros, e que todos possuem internet, computador e smart TV em casa, além de um celular próprio. Pode-se dizer, portanto, que o teste foi realizado com estudantes que estão familiarizados com tecnologias e tem acesso a mais de um tipo de dispositivo eletrônico e internet em sua própria casa, o que pode influenciar nos resultados, não sendo possível garanti-los sob outras circunstâncias, como, por exemplo, um menor nível de conhecimento tecnológico.

## 8 Conclusão

A TV Box com sistema operacional Armbian e interface gráfica LXDE obteve um resultado satisfatório, dado que a finalidade pretendida é o ensino de programação em escolas públicas, sendo uma alternativa viável aos computadores e notebooks. Com o resultado obtido pelo teste de usabilidade, seria um caminho possível realizar um projeto piloto para integrar as TV Boxes nas escolas, executando este projeto com poucas escolas inicialmente para ver se há uma boa receptividade de uma quantidade maior de estudantes, já que o teste foi realizado com uma quantidade pequena de alunos.

O maior desafio ainda é a performance da TV Box no acesso à internet em navegadores, devido às aplicações web estarem se tornando cada vez mais complexas, e assim, necessitando de mais recursos para serem acessadas. Para os trabalhos futuros, seria interessante uma investigação profunda sobre as sobrecargas do sistema causadas pelo navegador e alternativas para mitigar ou

solucionar esta questão.

Acredita-se que as soluções propostas possibilitam o uso de TV Boxes em escolas públicas como computadores de baixo custo e que permitem o acesso ao aprendizado de programação por meio de iniciativas das escolas e parceiros. Projetos futuros podem se beneficiar destes resultados para seguir buscando soluções com melhor desempenho e experiência para o usuário, explorando outros sistemas operacionais e aproveitando ao máximo o hardware disponível para possibilitar acesso a tecnologia aos estudantes.

## Referências

- [1] CEPTR0.br. Panorama sobre conectividade nas escolas públicas brasileiras. Disponível em: <https://medicoes.nic.br/media/Publicacao-internet-escolas-2024.pdf>. Acesso em: 13 dez. 2024.
- [2] Agência Senado. Especialistas apontam desafios para uso de tecnologias digitais nas escolas. Disponível em: <https://www12.senado.leg.br/noticias/materias/2024/11/12/especialistas-apontam-desafios-para-uso-de-tecnologias-digitais-nas-escolas>. Acesso em: 13 dez. 2024.
- [3] CETIC.br. Dados sobre Escolas. Disponível em: [https://data.cetic.br/explore/?pesquisa\\_id=7&unidade=Escolas](https://data.cetic.br/explore/?pesquisa_id=7&unidade=Escolas). Acesso em: 13 dez. 2024.
- [4] CETIC.br. Dados sobre Coordenadores de Escolas. Disponível em: [https://data.cetic.br/explore/?pesquisa\\_id=7&unidade=Coordenadores](https://data.cetic.br/explore/?pesquisa_id=7&unidade=Coordenadores). Acesso em: 13 dez. 2024.
- [5] SHIMASAKI, Rodrigo; PRADO, Maria Elisabette Brisola Brito. O Ensino da Programação e o Desenvolvimento do Pensamento Lógico: uma Revisão Sistemática de Literatura. *Revista de Ensino, Educação e Ciências Humanas*, 22(2):197–205, 2021.
- [6] Cassiana Silveira Lucas, Rafael Silveira Ferreira, Raquel de Miranda Barbosa, and Tiago Guimarães Moraes. Introdução à programação no ensino fundamental. *Revista Viver IFRS*, 8(8), ago. 2020.
- [7] Alan F Ribeiro and Douglas D K Yoshioka and Guilherme Ramirez and Juliana B Silva and Marcio dos Santos and Mariana A Souza and Matheus A S Cândido and Matheus O Rodrigues and Vitor R Pietrobon and Juliana F Borin. Transformação de TV Boxes piratas em computadores de baixo custo suportados por nuvem computacional. Disponível em: <https://www.ic.unicamp.br/~reltech/PFG/2024/PFG-24-43.pdf>, 2023. Acesso em: 13 dez. 2024.
- [8] LOPES, Vinícius Amorim Sales. Uma proposta de computadores de baixo custo utilizando tv boxes apreendidas. Disponível em: <http://repositorio.ufc.br/handle/riufc/76488>, 2023. 44 f. Trabalho de Conclusão de Curso (graduação) – Curso de Engenharia de Software.
- [9] RODRIGUES, L. S.; CANTERO, S. V. A. B.; FALLEIROS, E. L. S e SANTOS, J. A. Educabox - Repositório no GitHub. Disponível em: <https://github.com/educabox>, 2024. Acesso em: 13 dez. 2024.
- [10] Operating System. History of the Linux Kernel. Disponível em: [https://www.operating-system.org/betriebssystem/\\_english/bs-linux.htm](https://www.operating-system.org/betriebssystem/_english/bs-linux.htm), 2023. Acesso em: 13 dez. 2024.
- [11] Linux Kernel Archives. About Linux Kernel. Disponível em: <https://www.kernel.org/linux.html>, 2024. Acesso em: 13 dez. 2024.
- [12] The Linux Foundation. What is Linux? Disponível em: <https://www.linux.com/what-is-linux/>, 2024. Acesso em: 13 dez. 2024.
- [13] ARMBian Documentation. ARMBian Documentation. Disponível em: <https://docs.armbian.com/>, 2024. Acesso em: 13 dez. 2024.

- [14] ARMBian Forum. TV Boxes - Armbian Community Forums. Disponível em: <https://forum.armbian.com/forum/24-tv-boxes/>, 2024. Acesso em: 13 dez. 2024.
- [15] Phosh. About Phosh. Disponível em: <https://phosh.mobi/about/>, 2024. Acesso em: 13 dez. 2024.
- [16] postmarketOS Contributors. postmarketOS. Disponível em: <https://postmarketos.org/>, 2024. Acesso em: 13 dez. 2024.
- [17] Android Open Source Project. Android Open Source Project. Disponível em: <https://source.android.com/?hl=pt-br>, 2024. Acesso em: 13 dez. 2024.
- [18] Android x86. Android x86. Disponível em: <https://www.android-x86.org/>, 2024. Acesso em: 13 dez. 2024.
- [19] RISC-V. The Android System on RISC-V has come! Disponível em: <https://riscv.org/ecosystem-news/2021/01/the-android-system-on-risc-v-has-come%ef%bc%81-open-chip-community/>, 2021. Acesso em: 13 dez. 2024.
- [20] Projeto Smart Campus UNICAMP. Tutorial de instalação de Linux para TV Box - modelo TX2. Disponível em: [https://smartcampus.prefeitura.unicamp.br/pub/artigos\\_relatorios/Smart\\_Campus\\_Tutorial\\_TV\\_Box\\_TX2.pdf](https://smartcampus.prefeitura.unicamp.br/pub/artigos_relatorios/Smart_Campus_Tutorial_TV_Box_TX2.pdf), 2023. Acesso em: 13 dez. 2024.
- [21] Sentio Superbook. Enabling resolution change (DPI) in Sentio Desktop. Disponível em: <https://medium.com/sentio-superbook/enabling-sentio-desktop-dpi-resolution-change-d1a0b40e2c84>, 2017. Acesso em: 13 dez. 2024.
- [22] postmarketOS Community. PINE64 PinePhone Pro (pine64-pinephonepro). Disponível em: [https://wiki.postmarketos.org/wiki/PINE64\\_PinePhone\\_Pro\\_\(pine64-pinephonepro\)](https://wiki.postmarketos.org/wiki/PINE64_PinePhone_Pro_(pine64-pinephonepro)), 2024. Acesso em: 13 dez. 2024.
- [23] postmarketOS Community. Devices - postmarketOS Wiki. Disponível em: <https://wiki.postmarketos.org/wiki/Devices>, 2024. Acesso em: 13 dez. 2024.
- [24] postmarketOS Community. Amlogic Generic TV Box (amlogic-tvbox). Disponível em: [https://wiki.postmarketos.org/wiki/Amlogic\\_Generic\\_TV\\_Box\\_\(amlogic-tvbox\)](https://wiki.postmarketos.org/wiki/Amlogic_Generic_TV_Box_(amlogic-tvbox)), 2024. Acesso em: 13 dez. 2024.
- [25] postmarketOS Community. Google Veyron Chromebook (google-veyron). Disponível em: [https://wiki.postmarketos.org/wiki/Google\\_Veyron\\_Chromebook\\_\(google-veyron\)](https://wiki.postmarketos.org/wiki/Google_Veyron_Chromebook_(google-veyron)), 2024. Acesso em: 13 dez. 2024.
- [26] postmarketOS Community. Nvidia Tegra armv7 (nvidia-tegra-armv7). Disponível em: [https://wiki.postmarketos.org/wiki/Nvidia\\_Tegra\\_armv7\\_\(nvidia-tegra-armv7\)](https://wiki.postmarketos.org/wiki/Nvidia_Tegra_armv7_(nvidia-tegra-armv7)), 2024. Acesso em: 13 dez. 2024.
- [27] LXDE Contributors. LXDE. Disponível em: <https://www.lxde.org/>, 2024. Acesso em: 13 dez. 2024.
- [28] Olá Mundos! Instagram - Olá, Mundos! Disponível em: <https://www.instagram.com/olamundos/>, 2024. Acesso em: 13 dez. 2024.