UNIVERSIDADE FEDERAL DE SÃO JOÃO DEL-REI

Gabriel Rodrigues Chaves Carneiro

Wonderland: Orquestrando o país das maravilhas

São João Del Rei

2025

UNIVERSIDADE FEDERAL DE SÃO JOÃO DEL-REI

Gabriel Rodrigues Chaves Carneiro

Wonderland: Orquestrando o país das maravilhas

Dissertação apresentada como requisito para obtenção do título de Mestre em Ciência da Computação do curso de mestrado do Programa de Pós Graduação em Ciência da Computação (PPGCC) da Universidade Federal De São João Del-Rei (UFSJ).

Orientador: Flávio Luiz Schiavoni

Universidade Federal De São João Del-Rei

Mestrado em Ciência da Computação

São João Del Rei

2025

Ficha catalográfica elaborada pela Divisão de Biblioteca (DIBIB)
e Núcleo de Tecnologia da Informação (NTINF) da UFSJ,
com os dados fornecidos pelo(a) autor(a)

C289w

Carneiro, Gabriel.
Wonderland: Orquestrando o país das maravilhas / Gabriel Carneiro ; orientador Flávio Schiavoni. -- São João del-Rei, 2025.
101 p.

Dissertação (Mestrado - Programa de Pós-Graduação em Ciência da Computação) -- Universidade Federal de São João del-Rei, 2025.

1. software livre. 2. laboratório de pesquisa. 3. infraestrutura auto-hospedada. 4. computação distribuída. 5. colaboração. I. Schiavoni, Flávio, orient. II. Título.

Gabriel Rodrigues Chaves Carneiro

Wonderland: Orquestrando o país das maravilhas

Dissertação apresentada como requisito para obtenção do título de Mestre em Ciência da Computação do curso de mestrado do Programa de Pós Graduação em Ciência da Computação (PPGCC) da Universidade Federal De São João Del-Rei (UFSJ).

Trabalho aprovado. São João Del Rei, 11 de julho de 2025:

Dr. Flávio Luiz Schiavoni
Orientador

Dr. Rafael Sachetto Oliveira
Convidado 1

Dr. Marcelo Soares Pimenta
Convidado 2

São João Del Rei
2025

Agradecimentos

Este trabalho só pôde ser concluído devido ao apoio de diversas pessoas incríveis. Minha gratidão eterna à minha namorada e companheira Aretha, por me dar forças em meus momentos mais difíceis, e ser minha camarada e parceira em minhas aventuras. Sem você eu não teria entrado no mestrado, e nem sequer teria terminado de escrever este texto.

Agradeço principalmente à minha família por ter provido a base que permitiu que eu chegasse até aqui. Ao meu pai Waldir e à minha mãe Regina, obrigado por sempre acreditarem e investirem em minha educação. À minha irmã Rafaela, nosso espírito de competitividade me fez chegar onde estou hoje.

A todos os membros do laboratório de pesquisa ALICE, sem vocês este projeto nunca teria sequer sido cogitado. Obrigado por participarem de todas as atividades, reuniões e oficinas necessárias para a implantação de nossa infraestrutura. Este projeto é dedicado especialmente a vocês.

Agradeço também ao meu orientador Flávio por me orientar (e desorientar) nos momentos mais complicados. Suas ideias malucas tornaram esse projeto um verdadeiro país das maravilhas. Agradeço também à evolução tecnológica que permitiu a criação dos modelos de linguagem, pois este texto foi revisado, corrigido e reescrito utilizando inteligência artificial.

Agradeço a todas as amizades que fiz durante todos esses anos na UFSJ, jamais me esquecerei de vocês. A todos os camaradas do DCOMP, que compartilharam das mesmas dores e dificuldades durante essa jornada. A todos os docentes, técnicos e terceirizados que compõem a universidade, vocês são o que torna ela incrível.

E agradeço as agências de fomento CAPES, CNPQ, FAPEMIG e UFSJ por terem fomentado e possibilitado a execução deste e muitos outros projetos e pesquisas.

Este trabalho só chegou até aqui graças a todas essas pessoas maravilhosas — a elas, dedico esta dissertação.

Resumo

Esta dissertação apresenta o desenvolvimento de uma infraestrutura tecnológica auto-hospedada voltada para a criação artística e colaborativa no contexto do Laboratório ALICE, da Universidade Federal de São João del-Rei. O trabalho parte da observação de problemas práticos enfrentados no dia a dia do laboratório, como a fragmentação de ambientes, dependência de plataformas proprietárias, dificuldade de acesso remoto e ausência de controle centralizado. Em resposta, foi proposta e implementada a plataforma **WONDERLAND**, um sistema orquestrador que integra ferramentas de software livre para provisionamento automático de máquinas, sincronização de arquivos, gestão de artefatos, controle de acesso e colaboração remota. A solução foi projetada para ser flexível, sustentável e adaptável às demandas técnicas e artísticas do laboratório, oferecendo aos usuários um ambiente personalizado, seguro e sempre disponível.

Palavras-chaves: software livre; laboratório de pesquisa; infraestrutura auto-hospedada; computação distribuída; colaboração.

Abstract

This dissertation presents the development of a self-hosted technological infrastructure designed to support artistic and collaborative creation within the ALICE Laboratory at the Federal University of São João del-Rei. The work stems from the analysis of practical challenges faced in the lab's daily routines, such as fragmented environments, reliance on proprietary platforms, limited remote access, and lack of centralized control. In response, the WONDERLAND platform was proposed and implemented—an orchestration system that integrates free and open-source tools for automated provisioning, file synchronization, artifact management, access control, and remote collaboration. The solution was designed to be flexible, sustainable, and adaptable to both technical and artistic demands, providing users with a personalized, secure, and always-available environment.

Key-words: free software; research laboratory; self-hosted infrastructure; distributed computing; collaboration.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

APT	Advanced Packaging Tool
AUR	Arch User Repository
CI/CD	Continuous Integration/Continuous Delivery
CTAN	Campus Tancredo Neves
DAWs	Digital Audio Workstations
DCOMP	Departamento de Ciência da Computação
FFmpeg	Fast Forward MPEG
FOSS	Free and Open Source Software
LADSPA	Linux Audio Developer's Simple Plugin API
LDAP	Lightweight Directory Access Protocol
NFS	Network File System
NGinx	Engine X (servidor web)
OBS	Open Broadcast Studio
PAM	Pluggable Authentication Modules
PPAs	Personal Package Archives
PPGCC	Programa de Pós Graduação em Ciência da Computação
RTMP	Real-Time Messaging Protocol
SFTP	SSH File Transfer Protocol
SSHFS	SSH File System
SVN	Subversion

Sumário

1 Introdução

Este trabalho tem sua concepção no âmbito do Laboratório de Pesquisa ALICE. Localizado no Departamento de Ciência da Computação da Universidade Federal de São João del-Rei, é um ambiente multidisciplinar com pesquisas e atividades que abrangem computação, música, teatro e artes (SCHIAVONI et al., 2024), com alunos tanto da graduação, quanto pós graduação (SCHIAVONI et al., 2021).

A infraestrutura física é composta por algumas máquinas, instrumentos músicas analógicos e digitais, algumas cafeteiras, e um único servidor, e tudo isto para suportar uma vasta variedade de atividades realizadas no laboratório, cujo foco central é computação musical.

As diversas atividades realizadas abarcam múltiplas áreas do conhecimento, e acontecem de forma a incluir todos os interessados, mesmo para quem não está familiarizado com o ambiente. Diversos encontros semanais são realizados regularmente afim de discutir os projetos vigentes e futuros, além de discutirmos e nos organizarmos para eventos como congressos, apresentações artísticas.

Além de encontros regulares, algumas das pequisas realizadas pelos alunos buscam promover aulas, cursos e oficinas. Essas atividades tem como objetivo estimular a troca de conhecimentos em temas variados como produção musical, *beatmaking* (SOUSA; SCHIAVONI, 2023) (SOUSA; SOUSA; SCHIAVONI, 2023), mixagem (SOUZA; SCHIAVONI, 2023), uso do Linux, performances musicais ao vivo (SOUZA; SCHIAVONI, 2024), musicalização (JORDãO JOSIANE DE FATIMA RIBEIRO, 2023), performance teatral, peças musicais até mesmo criação de *websites*.

As pesquisas individuais e coletivas têm um forte enfoque em programação, que serve de base para a criação artística e científica. Mesmo com a computação musical como foco, os projetos não se limitam apenas a ela, tendo uma multitude de áreas de atuação.

1.1 O problema

Para compreendermos os reais objetivos deste projeto, iremos detalhar, descrever e analisar alguns cenários envolvendo as atividades realizadas no contexto do laboratório ALICE. Estes cenários incluirão atividades como oficinas, projetos de desenvolvimento de software, reuniões e pesquisas individuais, com a finalidade de identificar os desafios técnicos e organizacionais enfrentados, servindo como base para propormos soluções que aprimorem o fluxo de trabalho e gestão de recursos do laboratório.

Uso cotidiano do laboratório

A infraestrutura do laboratório é composta por algumas **máquinas independentes**, cada uma com sistemas e configurações **distintos**, fazendo com que os logins e os arquivos pessoais **não sejam sincronizados** entre elas, causando possíveis empecilhos para os usuários. Por exemplo, é comum que os usuários tenham uma máquina preferida para trabalhar, não sendo necessário transferir arquivos e configurações sempre que irão utilizar uma maquina não habitual. Essa situação evidencia a necessidade de uma solução centralizada para melhorar o gerenciamento do ambiente tecnológico.

Oficinas presenciais / síncronas

Neste cenários, podemos imaginar uma oficina de mixagem de áudio. Para o funcionamento correto da atividade, existem requisitos obrigatórios a serem cumpridos, como instalação de softwares nas máquinas como DAWs, hosts de plugins e servidores de áudio, como Mixxx, LDASPA e JACK. Além dos softwares, a oficina demanda uma série de **artefatos**: plugins, samples, músicas e documentos de apoio. Esses recursos podem vir acompanhados de referências como vídeos, livros e sites que auxiliem os alunos a entender as técnicas de mixagem e a aplicação prática dos conceitos aprendidos. Ao final da atividade, espera-se que os participantes consigam produzir artefatos – como músicas e sets – além de registrar dúvidas e desafios enfrentados durante o processo.

Do lado do organizador, há necessidades adicionais. Ele não só precisa coordenar a execução da oficina, mas também é responsável instalar e configurar em cada máquina individualmente o conjunto de software necessário pelos alunos para a realização do evento, além de gravar e transmitir ao vivo as sessões, utilizando plataformas de **videoconferência** como o **Google Meet**. Também cabe a ele documentar todo o processo e garantir que os artefatos – sejam os materiais de apoio distribuídos no início ou os produtos finais gerados pelos alunos – fiquem disponíveis para consulta posterior, geralmente sendo armazenados em plataformas como o **Google Drive**.

Esse cenário evidencia uma forte dependência de **plataformas proprietárias** tanto para a organização, quanto para a realização, como dito a respeito de plataformas de vídeo conferência e hospedagem de arquivos. Isto não é necessariamente um problema, já que estas plataformas garantem confiabilidade e funcionalidades extremamente úteis, porém, por se tratarem de plataformas de uso geral, a personalização de acordo com as necessidades do laboratório muitas vezes só é possível utilizando uma plataforma paga, ou uma plataforma gratuita que forneça planos pagos.

Oficinas assíncronas

Além de oficinas síncronas presenciais, outra possibilidade para transferência de conhecimento entre membros do nosso grupo de pesquisa pode utilizar oficinas assíncronas, na forma de videoaulas, por exemplo. Tal conteúdo costuma ser hospedado em plataformas de **compartilhamento de vídeo** como o **Youtube**, no entanto, tal plataforma traz diversas questões para nosso propósito. A primeira dela é a censura para o caso de conteúdo com direito autoral. É normal que em uma oficina de computação musical seja utilizado imagens, sons e vídeos de outras peças mas tal utilização pode acarretar na violação de direitos autorais e um vídeo com tais elementos pode vir a ser censurado na plataforma. A segunda questão é que tais oficinas podem carecer de outros artefatos para além do próprio vídeo. Materiais como slides, exemplos, códigos, arquivos de sons e outros precisam ser disponibilizados juntamente com o vídeo da aula em si mas tal condição não existe nas plataforma mais populares. Por fim, estas plataformas possuem a recomendação de outros vídeos, propagandas no meio do vídeo e diversos outros pontos que podem comprometer a atenção e o propósito dos vídeos gerados.

Projetos de Desenvolvimento de Software

Outro cenário importante envolve os projetos de desenvolvimento de software realizados no laboratório. Estes projetos necessitam que o ambiente de trabalho seja consistente, com suas dependências disponíveis e configuradas. Durante o ciclo de desenvolvimento deste software, algumas ferramentas são necessárias para escrever, editar e compilar este código, e para garantir a persistência dos arquivos, e também para acesso remoto colaborativo, são utilizadas ferramentas para **versionamento** como o **Git**. Portanto, o uso de plataformas como **GitHub** e **GitLab** para manter uma versão remota do código é elementar, e novamente nos vemos dependentes de ferramentas de terceiros para partes vitais de nosso fluxo de trabalho.

Os projetos geralmente contam com diversas dependências (bibliotecas, plugins e frameworks), e é imprescindível que estas estejam corretamente instaladas, em suas versões corretas. Devido ao fato de que as máquinas são usadas por múltiplas pessoas em diferentes momentos, pode ocorrer que diferentes projetos tenham dependências conflitantes, fazendo com que alguns alunos evitem usar as as mesmas máquinas por não entenderem o suficiente sobre. E novamente devido ao fato de que as máquinas são independentes, é comum que os alunos utilizem sempre as mesmas máquinas, a fim de não ter de reconfigurar o sistema de acordo com suas necessidades.

Produções Musicais

Sendo um laboratório cujo foco principal é computação musical, diversas atividades realizadas no ALICE envolvem diretamente a produção musical. Atividades como **Jam Sessions**, oficinas de **Beatmaking** e **Mixagem**, além de experimentações e performances que acarretam na produção de músicas. Por questões de direitos autorais, ficamos restritos e armazenar estas obras em plataformas como **Soundcloud** ou **Google Drive**, novamente nos restringindo a plataformas **proprietárias**.

Reuniões

No laboratório, as reuniões são parte essencial do processo de colaboração, mas frequentemente os participantes não podem estar presentes fisicamente simultaneamente. Essa situação exige o uso de **videoconferências** e ferramentas de comunicação remota como o *Google Meet* para garantir que as discussões ocorram de forma produtiva. Durante essas reuniões, são debatidos assuntos diversos – como a organização de atividades e eventos, o andamento de projetos individuais e coletivos, planejamento do cronograma para publicações em eventos e congressos e etc. Para isto utilizamos constantemente ferramentas colaborativas – como calendários compartilhados, editores de documentos e sistemas de compartilhamento de arquivos. E como citado anteriormente no caso da oficina de mixagem, a dependência do uso de ferramentas de terceiros muitas vezes faz com que adaptemos o nosso fluxo de trabalho à elas, e não o contrário.

Escrita de artigos, relatórios e monografias

Por se tratar de um grupo de pesquisa com alunos de Iniciação científica, graduação e mestrado, é comum que os discentes estejam constantemente utilizando ferramentas de escrita de textos acadêmicos. Para permitir a **colaboração** nestas escritas, **editores online** como o **Google Docs** ou o Overleaf acabam sendo utilizados de forma que mais de uma pessoa possa contribuir com a produção de conhecimento. Isso também gera uma dependência de ferramentas externas que permitem acesso gratuito mas que não permitem muita customização ou liberdade de configuração.

Documentação de processos

É comum, em nosso laboratório, que algum sistema demande nova configuração ou que um processo tenha alguns passos para ser feito. No entanto, não temos no momento nenhuma estrutura que permite a documentação destes processos e parte do conhecimento adquirido por discentes pesquisadores é perdida, seja ela o passo a passo para alcançar determinado objetivo, seja os artefatos utilizados para isso, como arquivos de configuração ou *scripts*.

Troca de arquivos e demais artefatos de software

Muitas vezes, na criação científica ou artística há a necessidade de compartilhar arquivos ou informações em geral como links, documentos online, textos, vídeos e outros. Este compartilhamento ocorre de diversas maneiras mas normalmente acaba sendo feito por meio de **ferramentas de comunicação** como o **Whatsapp** ou o **Telegram**. No entanto, tais ferramentas além de serem proprietárias foram feitas para uma comunicação efêmera e por isso não possuem organização de suas mensagens ou um histórico simples de ser acessado.

Análise dos problemas descritos

A partir dos problemas que observamos, iremos levantar os requisitos não funcionais para este projeto. Inicialmente, fica claro que a utilização de soluções fechadas ou proprietárias pode limitar a capacidade de adaptação, gerando amarras e dificultando a personalização. Conforme apresentada na Tabela 1.

Tabela 1 – Soluções existentes

Problema	**Soluções**
Reuniões	Google Meet, Zoom, Teams
Armazenamento Remoto	Google Drive, Dropbox, Onedrive
Controle de versionamento	GitHub, GitLab
Edição de documentos	Google Docs, Microsoft Office, Overleaf
Músicas	Spotify, Soundcloud
Vídeos	Youtube
Documentação	Github, Google Docs
Transmissões ao vivo	Youtube, Twitch
Chat	WhatsApp, Facebook Messenger, Telegram, Microsoft Teams

Sendo assim, optamos pela utilização de soluções **auto-hospedadas** e exclusivamente de **Software Livre**, efetivamente removendo as barreiras impostas pela utilizações proprietárias, permitindo a flexibilidade e extensibilidade que nos é limitada por fornecedores externos.

Um problema identificado previamente foi a dificuldade dos usuários em acessar seus ambientes virtuais de forma consistente por dependerem de um dispositivo específico. Em muitos casos, a experiência se torna fragmentada, pois o ambiente personalizado do usuário não se mantém igual, e o acesso aos documentos fica condicionado à máquina em que eles foram originalmente configurados. Essa situação pode ser frustrante e restringe a possibilidade de continuidade no trabalho em contextos onde a mobilidade é necessária. Sendo assim, garantir que os usuários poderão acessar seus ambientes em diferentes

máquinas, e também remotamente, se torna um requisito primordial para solucionar esta questão.

Um desafio evidente é a dificuldade em promover uma integração efetiva entre os usuários. O ambiente tecnológico disponível para os alunos não possui nenhum mecanismo pensado na colaboração, como compartilhamento de informações e arquivos, ou ferramentas colaborativas que pudessem possibilitar trabalho cooperativo remoto. Desta forma, implantar uma infraestrutura que possa mediar os processos criativos e acadêmicos independentemente de soluções proprietárias é um dos requisitos deste trabalho.

Podemos também identificar uma desorganização e até mesmo perda dos materiais produzidos no laboratório ao longo do tempo. Relatórios, códigos, documentos e diversas outras produção se dispersas em múltiplas plataformas, HDs e gavetas, ou até mesmo perdidos, comprometendo o acúmulo de conhecimento do grupo. Além disso, não existe uma maneira simples de se buscar estes artefatos, sendo necessário ir atrás das diversas fontes não documentadas que usamos para tal. Isso evidencia a necessidade de mecanismos e documentação de processos para garantir a persistências destas produções, assegurando armazenamento, acesso e buscas destes materiais.

Algo que também se vê necessário é discutir os aspectos relacionados à segurança e ao acesso aos recursos deste sistema, para minimizar o risco de interferências indesejadas ou acesso indevido aos serviços. Portanto será necessário estabelecer mecanismos rigorosos de controle de identidade e acesso, delimitando claramente quem e como pode ter acesso, garantindo mais segurança e confiabilidade no ambiente.

Um dos maiores desafios para uma implantação deste sistema é certificar de que o sistema tenha continuidade. De nada adianta um grande sistema que absolutamente ninguém entende o funcionamento, e que não existam pessoas capazes de realizar manutenção ou adaptação para novas necessidades. Portanto, faz-se necessário a criação de documentações e capacitações que assegurem o futuro do sistema e de sua integração com o fluxo de trabalho do laboratório a longo prazo.

Cada um desses pontos reflete problemas reais enfrentados atualmente, e a Tabela 2 sumariza os requisitos levantados. Quando mapeamos essas dificuldades, elas indicam os requisitos não funcionais que são necessários para transformar o ambiente. Dessa forma, a escolha dos requisitos – desde a adoção de soluções *open source* até a capacitação dos usuários – se fundamenta nas limitações e desafios observados, preparando o terreno para futuras discussões e decisões, sem, neste momento, apresentar uma solução final. Essa abordagem permite que os desafios sejam compreendidos de forma profunda, garantindo que as futuras escolhas tecnológicas estejam alinhadas com as reais necessidades e restrições do contexto analisado.

Tabela 2 – Requisitos Não Funcionais

Requisito	**Descrição**
Open Source	Usar soluções auto-hospedadas e exclusivamente open source.
Mobilidade de usuário	Permitir ao usuário acessar seu ambiente operacional personalizado e arquivos pessoais de maneira transparente, independente da máquina utilizada.
Colaboração	Possibilitar a colaboração entre os usuários, permitindo que trabalhem juntos de forma integrada e compartilhada.
Persistência dos artefatos	Garantir que todos os materiais, documentos, códigos e produções dos usuários estejam armazenados de forma permanente e acessível.
Disponibilidade e Acesso Remoto	Assegurar que o sistema esteja sempre disponível e operacional, permitindo o acesso contínuo aos serviços e arquivos compartilhados, inclusive de forma remota.
Controle de acesso	Controlar de forma rigorosa quem pode acessar o sistema e o que pode ser acessado.
Documentação	Disponibilizar documentação clara e acessível para facilitar a continuidade da gestão técnica e operacional mesmo após mudanças de pessoal.
Capacitação	Garantir facilidade na manutenção e atualização do sistema pelos próprios usuários.

1.2 Trabalhos relacionados

As ideias, problemas e soluções discutidas neste trabalho não são exclusivas do Laboratório Alice; diversas abordagens semelhantes já foram exploradas em outros contextos e instituições. Alguns exemplos de trabalhos relacionados são apresentados a seguir.

O artigo "Gerenciamento Remoto dos Computadores dos Laboratórios de Informática do IFSULDEMINAS, Usando a Ferramenta FOG Project"(RATIS; MOURA; ROSSI, 2023) investiga a utilização do FOG Project como ferramenta para facilitar a manutenção preventiva em ambientes acadêmicos. Os autores realizam uma comparação detalhada entre o FOG Project e o Clonezilla, considerando aspectos como tempo de execução, complexidade de configuração, incidência de erros e utilização de recursos. A pesquisa busca avaliar o potencial do FOG Project para tornar a manutenção mais eficiente, além de promover o compartilhamento de conhecimentos sobre gerenciamento remoto de computadores.

Outro trabalho relevante é a dissertação intitulada "Design Implications of an Online Collaborative Workspace Developed Using Open Source Software"(BOTHMA, 2006). Este estudo descreve o desenvolvimento de um espaço colaborativo online, utilizando exclusivamente software livre, voltado para atender os grupos de pesquisa DISSAnet e IKS.

Entre as funcionalidades desenvolvidas, destacam-se a criação de uma biblioteca digital, fórum de discussão, motor de hipermídia adaptativa e um banco de dados específico. O objetivo do estudo foi compreender os requisitos e especificações necessárias para a criação de ambientes colaborativos flexíveis, testando sua eficácia e usabilidade por meio de entrevistas e desenvolvimento iterativo do protótipo. A pesquisa concluiu que, apesar de atender às necessidades iniciais, algumas limitações precisam ser exploradas futuramente.

O trabalho "OSCAR meta-package system"(MUGLER; NAUGHTON; SCOTT, 2005) fala a respeito do OSCAR, uma sistema de provisionamento de *clusters* a partir de meta-pacotes chamados **Pacotes OSCAR**. Estes pacotes podem conter softwares, configurações, *scripts* e outras dependências necessárias para as máquinas. Mugler, Naughton e Scott (2005) faz uma análise da versão 4.0 do OSCAR, e mostra uma pesquisa realizada com a comunidade do projeto sobre algumas das propostas de melhoria do sistema.

Ao analisarmos os desafios e as dificuldades encontradas, percebemos que os requisitos não funcionais não surgiram de forma isolada, mas sim a partir de problemas concretos que, ao serem mapeados, nos guiam para as funcionalidades que desejamos implementar. Antes de discutirmos nossa proposta, é imprescindível discutirmos a literatura vigente, com trabalhos relacionados e soluções encontradas para problemas similares, e a partir disso propormos nosso próprio sistema.

Para ilustrar, vamos considerar o primeiro aspecto levantado: a escolha por soluções auto-hospedadas e exclusivamente open source. Esse requisito não funcional nasce de um cenário em que a dependência de tecnologias proprietárias impõe limitações à flexibilidade e à capacidade de customização do ambiente. Ao refletir sobre essa dificuldade, percebemos que é necessário estabelecer um ambiente padronizado e configurado, o que nos leva diretamente ao **Provisionamento e Padronização do Ambiente**, que será discutido na seção 2.1. Assim, a necessidade de liberdade e autonomia, identificada no requisito não funcional, encontra sua expressão funcional na criação de um laboratório onde as máquinas estejam preparadas com todas as ferramentas necessárias já instaladas e configuradas automaticamente.

Um ponto que merece destaque é a dificuldade dos usuários em acessar seus ambientes de trabalho de maneira consistente, especialmente quando estão vinculados a um único dispositivo. Esse problema, mapeado na necessidade de mobilidade, nos leva ao requisito funcional de **Mobilidade do Usuário e Sincronização de Arquivos**, que será discutido na seção 2.2. Aqui, a experiência fragmentada que os usuários enfrentam se transforma em um requisito que visa oferecer um ambiente personalizado e consistente, independente do dispositivo utilizado, garantindo uma continuidade no fluxo de trabalho e na experiência do usuário.

Além disso, a ausência de um mecanismo eficaz para que os usuários possam interagir e colaborar de forma integrada foi um desafio recorrente. Quando nos deparamos com

essa dificuldade, ficou claro que a falta de integração comprometia a troca de informações e o trabalho conjunto, aspectos essenciais para a evolução e o dinamismo do ambiente. Essa constatação impulsionou a definição do requisito funcional **Ferramentas Colaborativas e Acesso Remoto**, que será discutido na seção 2.3. Dessa forma, a necessidade de colaboração, identificada de forma não funcional, é convertida em uma funcionalidade que, ao ser implementada, permitirá que os usuários trabalhem juntos, mesmo que estejam geograficamente distantes, com acesso remoto aos recursos compartilhados.

Outra área crítica está relacionada à organização e à preservação dos materiais produzidos ao longo do tempo. A dispersão e a possível perda de documentos, códigos e demais produções evidenciam a urgência de se adotar um mecanismo robusto de armazenamento. Assim, o requisito não funcional de persistência dos artefatos se traduz na funcionalidade **Gestão e Persistência dos Artefatos**, que visa assegurar que todo o conhecimento e as produções sejam mantidos de forma permanente e organizada, evitando retrabalho e perdas que comprometam a continuidade do projeto, que será discutido na seção 2.4.

Também nos deparamos com questões de segurança e controle de acesso, onde a falta de um gerenciamento adequado pode levar a problemas de integridade e até mesmo a acessos indevidos. Essa preocupação, registrada como um requisito não funcional, direciona-se para o requisito funcional **Controle de Acesso Centralizado**, que será discutido na seção 2.5. Ao estabelecer mecanismos de autenticação e autorização rigorosos, garantimos que somente usuários autorizados possam acessar determinadas áreas, preservando a integridade do ambiente e das informações.

Por fim, a sustentabilidade do projeto a longo prazo depende fortemente da continuidade do conhecimento e da capacidade de manutenção do sistema. A necessidade de documentar de forma clara os processos e de capacitar os usuários para que possam, autonomamente, manter e evoluir o ambiente, é refletida no requisito funcional **Capacitação e Sustentabilidade**, que será discutido na seção 2.6. Assim, a clareza na documentação e a promoção de treinamentos não surgem por acaso, mas sim como resposta à dificuldade de manter a continuidade operacional diante de mudanças na equipe ou na gestão.

Em resumo, a proposta se baseia na ideia de que cada requisito funcional principal foi cuidadosamente derivado dos desafios reais identificados por meio dos requisitos não funcionais. Essa abordagem permite que o projeto não apenas responda às limitações atuais, mas também se prepare para futuras necessidades, garantindo que cada funcionalidade implementada esteja em perfeita sintonia com os problemas e as expectativas do ambiente. Dessa forma, estabelecemos um caminho claro para transformar os desafios em soluções, sem, neste momento, apresentar uma solução final, mas sim demonstrando como cada aspecto crítico se traduz em funcionalidades essenciais para o sucesso do projeto.

1.3 Estrutura da dissertação

Esta dissertação está organizada em 8 capítulos da seguinte forma: O Capítulo 1 é a introdução, onde foram apresentados o contexto, objetivos e a justificativa deste trabalho. O Capítulo 2 apresenta a proposta geral deste projeto, levantando requisitos para a infraestrutura baseado nos problemas postos na introdução. Os Capítulo 3 a 7 seguem uma mesma estrutura lógica: cada um deles aborda um dos pilares postos na proposta, e é dividido entre referencial teórico e implantação prática, permitindo que seja possível encapsular as discussões conceituais juntamente com a implantação. O Capítulo 8 apresenta e discute os resultados obtidos, reavendo os problemas levantados na introdução e analisando se eles foram resolvidos com a implantação da infraestrutura. Por fim traremos as considerações finais no Capítulo 9, avaliando os resultados obtidos, apontando problemas e limitações, e sugerindo possíveis caminhos de trabalhos futuros.

2 Proposta

Ao analisarmos os desafios e as dificuldades encontradas, percebemos que os requisitos não funcionais não surgiram de forma isolada, mas sim a partir de problemas concretos que, ao serem mapeados, nos guiam para as funcionalidades que desejamos implementar. Antes de discutirmos nossa proposta, é imprescindível discutirmos a literatura vigente, com trabalhos relacionados e soluções encontradas para problemas similares, e a partir disso propormos nosso próprio sistema.

Para ilustrar, vamos considerar o primeiro aspecto levantado: a escolha por soluções auto-hospedadas e exclusivamente Software Livre. Esse requisito não funcional nasce de um cenário em que a dependência de tecnologias proprietárias impõe limitações à flexibilidade e à capacidade de customização do ambiente. Ao refletir sobre essa dificuldade, percebemos que é necessário estabelecer um ambiente padronizado e configurado, o que nos leva diretamente ao **Provisionamento e Padronização do Ambiente**, que será discutido na seção 2.1. Assim, a necessidade de liberdade e autonomia, identificada no requisito não funcional, encontra sua expressão funcional na criação de um laboratório onde as máquinas estejam preparadas com todas as ferramentas necessárias já instaladas e configuradas automaticamente.

Um ponto que merece destaque é a dificuldade dos usuários em acessar seus ambientes de trabalho de maneira consistente, especialmente quando estão vinculados a um único dispositivo. Esse problema, mapeado na necessidade de mobilidade, nos leva ao requisito funcional de **Mobilidade do Usuário e Sincronização de Arquivos**, que será discutido na seção 2.2. Aqui, a experiência fragmentada que os usuários enfrentam se transforma em um requisito que visa oferecer um ambiente personalizado e consistente, independente do dispositivo utilizado, garantindo uma continuidade no fluxo de trabalho e na experiência do usuário.

Além disso, a ausência de um mecanismo eficaz que permitam aos usuários interagir e colaborar de forma integrada foi um desafio recorrente nos problemas levantados. Em retrospecto, a falta de integração comprometia a troca de informações e o trabalho conjunto, aspectos essenciais para a evolução e o dinamismo do ambiente. Isso nos levou a definição do requisito funcional **Ferramentas Colaborativas e Acesso Remoto**, que será discutido na seção 2.3. Dessa forma, a necessidade de colaboração, identificada de forma não funcional, é convertida em uma funcionalidade que, ao ser implementada, permitirá que os usuários trabalhem juntos, mesmo que estejam geograficamente distantes, com acesso remoto aos recursos compartilhados.

Outra área crítica está relacionada à organização e à preservação dos materiais

produzidos ao longo do tempo. A dispersão e a possível perda de documentos, códigos e demais produções evidenciam a urgência de se adotar um mecanismo robusto de armazenamento. Chegando assim no requisito funcional **Gestão e Persistência de artefatos**, visando assegurar que tudo o que for produzido no laboratório e por seus membros seja mantido de forma permanente e organizada, evitando perdas que possam comprometer a continuidade dos projetos, que será discutido na seção 2.4.

A falta de um gerenciamento adequado de permissões e acesso pode nos levar a questões de segurança. Este requisito não funcional nos direciona ao requisito funcional **Controle de Acesso Centralizado**, que será discutido na seção 2.5. Ao estabelecermos mecanismo rigorosos de autenticação e autorização, podemos garantir que apenas pessoas autorizadas possam acessar certos serviços ou documentos, preservando a integridade dos dados e de seus usuários.

A sustentabilidade do projeto e de toda a infraestrutura implantada depende a longo prazo da continuidade do conhecimento e da capacidade dos futuros usuários de realizar as devidas manutenções e adaptações necessárias. Documentações claras e objetivas sobre processos, aulas e capacitações são necessárias para assegurar um futuro para o sistema. Isso será discutido na seção 2.6.

Em resumo, a proposta se baseia na ideia de que cada requisito funcional principal foi cuidadosamente derivado dos desafios reais identificados por meio dos requisitos não funcionais. Essa abordagem permite que o projeto não apenas responda às limitações atuais, mas também se prepare para futuras necessidades, garantindo que cada funcionalidade implementada esteja em perfeita sintonia com os problemas e as expectativas do ambiente. Dessa forma, estabelecemos um caminho claro para transformar os desafios em soluções, sem, neste momento, apresentar uma solução final, mas sim demonstrando como cada aspecto crítico se traduz em funcionalidades essenciais para o sucesso do projeto.

2.1 Provisionamento e Padronização do Ambiente

Como relatado na seção de problemas na introdução, muitas questões se resumem no fato de que as máquinas do laboratório ALICE não possuem nenhum tipo de padronização em suas configurações, nem manuais nem automáticas. A partir disso, iniciaremos esta proposta discutindo o provisionamento e padronização deste ambiente.

Primeiramente, precisamos garantir que todas as máquinas tenham configurações idênticas, de forma que os usuários possam ter a liberdade usar qualquer uma sem nenhum problema. Para isto, é preciso discutir processos que permitam a preparação (instalação do SO e suas configuração) destas máquinas de forma que isto ocorra de forma automatizada com o mínimo de intervenções humanas, e que esteja pronta para uso no fim deste processo. Podemos também a possibilidade de que as máquinas realizem automaticamente

verificações para se assegurar que suas configurações estejam atualizadas, checando se o estado atual do sistema corresponde às especificações impostas, sem nenhum tipo de intervenção manual. Neste cenário todas as atualizações poderiam ser automaticamente buscadas e aplicadas periodicamente para garantir uniformidade entre as máquinas.

Indo além, uma proposta interessante é a possibilidade de que este provisionamento seja acompanhado de diferentes tipos de perfis, onde cada perfil teria configurações distintas para operações distintas. Um exemplo disto seria termos um perfil para "tocar músicas ao vivo", com configurações voltadas para baixa latência e processamento de áudio, ou um perfil voltado para desenvolvimento de software, que conta com as ferramentas e configurações utilizadas neste tipo de tarefa.

Aliado a tudo isto, poderíamos implantar mecanismos de verificação e autorrecuperação, onde as próprias máquinas monitoram sua integridade, juntamente com mecanismos de autorrecuperação em casos de falha. Não limitado a apenas isso, poderíamos também realizar o monitoramento continuo destas máquinas, verificando o hardware e software delas.

Resumindo: estamos propondo que o provisionamento e gerenciamento de ambiente seja automatizado, que possa se moldar a diferentes tarefas, que seja auto-recuperável e padronizado, para garantir um fluxo de trabalho transparente e independente de máquinas específicas, permitindo maior flexibilidade aos usuários. A tabela 3 sintetiza os requisitos levantados nessa seção.

Tabela 3 – Provisionamento e Padronização do Ambiente

Requisito	**Descrição**
Automatização do Provisionamento	Processo automatizado de instalação e configuração das estações, eliminando intervenções manuais.
Padronização do Ambiente	Manutenção de um ambiente uniforme com ferramentas e configurações predefinidas.
Bootstrapping e Verificação de Integridade	Rotinas de inicialização que identificam e corrigem inconsistências automaticamente.
Perfis Dinâmicos	Combinação de uma camada base padronizada com personalização adaptada ao usuário.
Autorrecuperação	Mecanismos proativos para monitorar e corrigir falhas antes que afetem os usuários.
Provisionamento Seletivo	Ativação de diferentes perfis ou configurações conforme a finalidade ou contexto de uso.

2.2 Mobilidade do Usuário e Sincronização de Arquivos

Depois de propormos que a infraestrutura tecnológica do laboratório fosse padronizada para garantir consistência entre as máquinas, precisamos também discutir a respeito

da mobilidade e sincronização de arquivos pessoais neste ambiente. Para isso podemos imaginar que o espaço pessoal dos usuários se "mova"juntamente com ele, de forma que todos os seus arquivos e configurações pessoais seja transportado para outras estações de trabalho. Discutiremos mais a respeito de tecnologias, protocolos e padrões usados nas seções 3 e 4 dos conceitos relacionados.

Assim como proposto na seção anterior quando falamos a respeito do uso de perfis para diferentes tarefas, seria interessante também a possibilidade de implantar o mesmo mecanismo para os usuários, onde poderiam definir diferentes configurações e aplicações específicos para cada tipo de tarefa, permitindo mais liberdade de customização.

Ampliando as possibilidades já discutidas, poderíamos também permitir que este ambiente pessoal seja acessado remotamente, fora da infraestrutura física do laboratório. Isso permitiria que muitas atividades pudessem ser continuadas mesmo quando haja a impossibilidade de se estar presencialmente no ambiente, ou até mesmo dentro do próprio laboratório mas utilizando dispositivos pessoais que se conectam neste ambiente.

Por fim, para certificar a integridade destes deste sistema a longo prazo, poderíamos implementar rotinas de backup automáticos, aliados a processos de recuperação automática em casos de falhas. Isso asseguraria que mesmo em casos extremos de perda ainda poderíamos recuperar todo o material.

Em resumo, propomos criar um ambiente personalizado que seja acessível de maneira transparente em qualquer máquina física de nosso laboratório, como também a possibilidade de acessar esse ambiente remotamente, e que tudo isso esteja assegurado através de rotinas de backup e auto-recuperação em casos de falha. A tabela 4 sintetiza os requisitos levantados nessa seção.

2.3 Ferramentas Colaborativas e Acesso Remoto

Após termos estabelecido um ambiente padronizado, com provisionamento automático e mobilidade plena dos usuários, precisamos dar o próximo passo, pensando agora em como as pessoas podem efetivamente trabalhar juntas de forma colaborativa. Nesse contexto, é importante considerar que diferentes tipos de documentos e materiais exigem diferentes tipos de colaboração, cada um com suas particularidades, o que demanda ferramentas específicas e mecanismos distintos para apoiar essa diversidade.

Inicialmente, podemos considerar a **criação e edição colaborativa** de documentos como um exemplo clássico. Aqui, torna-se essencial reconhecer que nem todo documento de texto é igual, e cada tipo pode exigir estratégias colaborativas específicas. Por exemplo, um documento acadêmico formatado em LaTeX possui características muito diferentes de um documento em formato tradicional, como um arquivo DOC, ou mesmo

Tabela 4 – Mobilidade do Usuário e Sincronização de Arquivos

Requisito	**Descrição**
Ambiente Personalizado	Carregamento automático do perfil do usuário, com configurações e preferências individuais.
Sincronização Automática	Atualização contínua dos arquivos pessoais, garantindo acesso consistente em qualquer dispositivo.
Continuidade de Sessão	Retomada do trabalho exatamente no ponto em que foi interrompido, com restauração de aplicações e janelas.
Histórico de Atividades	Registro detalhado das ações recentes, facilitando a retomada e o entendimento do contexto de trabalho.
Antecipação de Necessidades	Mecanismos que sugerem recursos com base nos hábitos e preferências dos usuários.
Acesso Remoto Integrado	Permite que o ambiente personalizado seja acessado de qualquer local, mantendo a experiência unificada.

de um documento criado em um editor online simples, voltado para anotações rápidas e edição simultânea em tempo real.

Enquanto a edição colaborativa de um texto acadêmico formatado em LaTeX normalmente requer mecanismos robustos para controle de versão, visualização e resolução eficiente de conflitos, outros tipos de documentos podem priorizar interações mais dinâmicas e instantâneas, permitindo edição simultânea por múltiplos usuários em tempo real, com alterações refletidas imediatamente para todos os participantes. Assim, é importante que as ferramentas propostas sejam capazes de atender a esses contextos distintos, oferecendo mecanismos apropriados para cada cenário específico de colaboração.

Além da edição simultânea (colaboração síncrona), onde os usuários podem acompanhar as mudanças ao mesmo tempo e participar ativamente juntos, também precisamos considerar a colaboração assíncrona. Neste caso, as pessoas trabalham em diferentes horários, locais ou ritmos, mas precisam que suas contribuições estejam registradas, organizadas e facilmente rastreáveis. Para isso, seria essencial a adoção de mecanismos eficientes para acompanhar o **histórico de edições**, mantendo um registro detalhado de quem fez cada alteração, quando e com que propósito. Assim, um usuário que acessa o documento após um intervalo de tempo pode rapidamente compreender o contexto das mudanças, garantindo que o trabalho permaneça coeso e organizado, independentemente do intervalo entre contribuições.

Uma dimensão fundamental dessa proposta envolve o uso de **controle de versões**. Nesse sentido, podemos pensar em mecanismos semelhantes aos oferecidos por sistemas como **Git**, adaptados à realidade e necessidades do laboratório, permitindo que diferen-

tes versões dos documentos, códigos ou outros materiais sejam facilmente gerenciadas. A ideia é possibilitar que os usuários não apenas rastreiem alterações individuais, mas também consigam recuperar facilmente versões anteriores, comparar modificações feitas por diferentes pessoas e resolver eventuais conflitos que possam surgir durante o processo colaborativo.

Esses mecanismos podem ir além da simples edição de textos ou códigos, abrangendo também documentos complexos como planilhas, apresentações e arquivos multimídia. Cada um desses tipos de arquivos traz consigo desafios específicos, e a proposta precisa considerar ferramentas que consigam oferecer uma colaboração fluida e intuitiva, independentemente do tipo de material sendo trabalhado.

Além disso, uma característica importante a ser pensada é o **acesso remoto** a essas ferramentas colaborativas. Para maximizar a flexibilidade do ambiente, precisamos garantir que os usuários possam participar ativamente dos processos colaborativos, mesmo quando estiverem fora das instalações físicas do laboratório. Isso significa que todas essas funcionalidades colaborativas devem ser acessíveis por meio de interfaces intuitivas e seguras, permitindo que as pessoas contribuam, comentem e discutam os materiais mesmo à distância, mantendo assim a produtividade e a fluidez do trabalho.

Nesse sentido, seria essencial propor um ambiente integrado para **videoconferências** e reuniões virtuais. Essa solução permitiria que os usuários realizassem encontros remotos com facilidade, compartilhando telas, documentos e apresentações em tempo real, promovendo discussões produtivas mesmo que fisicamente distantes. Seria interessante ainda que esse recurso fosse integrado ao sistema principal do laboratório, **permitindo a gravação** automática das reuniões e facilitando a revisão posterior, especialmente útil para aqueles que não puderam participar ao vivo.

Além disso, é importante levar em conta a necessidade de ferramentas para organização coletiva do tempo, como **calendários compartilhados**. Esses calendários poderiam permitir o agendamento simplificado de reuniões, o acompanhamento dos compromissos de toda a equipe e ainda fornecer alertas automáticos para eventos ou prazos importantes, garantindo que todos estejam alinhados e bem informados sobre as atividades planejadas.

Uma possibilidade interessante é um sistema colaborativo para **anotações rápidas**, onde usuários possam compartilhar ideias, apontamentos ou informações breves de forma imediata e prática. Essa ferramenta poderia funcionar como uma espécie de quadro compartilhado, onde todos possam adicionar conteúdos relevantes, organizar tarefas pendentes ou fazer pequenas reuniões rápidas para esclarecer dúvidas pontuais, otimizando a comunicação cotidiana.

Adicionalmente, podemos propor mecanismos para comunicação direta, como **chats**

integrados ou fóruns de discussão internos ao ambiente colaborativo. Esses recursos são úteis para que os usuários possam rapidamente trocar ideias, tirar dúvidas pontuais ou discutir sobre questões que surgem durante o trabalho diário, sem a necessidade de formalidades como e-mails ou reuniões mais longas.

Podemos ainda imaginar a possibilidade de mecanismos adicionais que complementem e enriqueçam a colaboração, como comentários contextuais integrados aos documentos, **ferramentas de comunicação** embutidas nos próprios ambientes colaborativos e notificações automáticas que avisem quando mudanças relevantes ocorrerem nos arquivos compartilhados. Esses elementos adicionais tornam o processo de colaboração mais rico e eficiente, incentivando os usuários a se manterem sempre conectados às atualizações e discussões importantes.

Por fim, também é válido considerar a ideia de um sistema centralizado que organize e facilite a **busca dos materiais** compartilhados. Esse sistema poderia oferecer recursos como categorização automática de documentos, pesquisa inteligente e capacidade de reunir diferentes tipos de artefatos relacionados a um mesmo projeto ou atividade, garantindo que os usuários possam localizar facilmente os materiais e informações necessários, evitando a perda de tempo ou retrabalho.

Dessa forma, ao propor ferramentas colaborativas avançadas e acessíveis, estaremos construindo um ambiente completo e versátil, que apoia diferentes estilos e formas de colaboração, valorizando tanto interações rápidas e dinâmicas quanto processos mais estruturados e detalhados, garantindo que a experiência colaborativa seja sempre produtiva, organizada e intuitiva para todos os usuários envolvidos.

A tabela 5 sintetiza os requisitos levantados nessa seção.

Tabela 5 – Ferramentas Colaborativas e Acesso Remoto

Requisito	**Descrição**
Edição Colaborativa de Documentos	Suporte para edição simultânea, com controle de versões adaptado a diferentes formatos.
Colaboração Síncrona e Assíncrona	Ferramentas que possibilitam trabalho em tempo real e em momentos distintos, mantendo a coesão do conteúdo.
Videoconferências Integradas	Reuniões virtuais com compartilhamento de tela e gravação, facilitando a comunicação à distância.
Calendários e Agendamento Compartilhado	Organização de compromissos e eventos de forma colaborativa e integrada.
Anotações Rápidas	Espaços para brainstorming e troca imediata de ideias, como quadros colaborativos.
Gerenciamento de Tarefas	Sistema para atribuição, monitoramento e feedback das atividades e projetos em grupo.
Comunicação Direta	Chats e fóruns internos para troca rápida de informações e discussões pontuais.

2.4 Gestão e Persistência dos Artefatos

Após estabelecer um ambiente plenamente colaborativo, que permite aos usuários trabalhar juntos de maneira integrada, surge uma nova necessidade fundamental: assegurar que tudo o que for produzido por esses usuários seja preservado, gerenciado e permaneça acessível de forma confiável ao longo do tempo.

Nesse ponto da proposta, é importante refletir sobre a diversidade dos materiais produzidos pelo laboratório. Artefatos podem variar desde documentos simples até projetos complexos, passando por códigos-fonte, relatórios, apresentações, gravações de reuniões, imagens, vídeos e até mesmo arquivos multimídia diversos. Cada tipo de material possui características próprias, exigindo **abordagens específicas** para garantir uma gestão eficaz e permanente.

Um dos desafios nesse sentido está na **organização eficiente** e intuitiva de tais artefatos. Precisamos imaginar sistemas capazes de **armazenar e categorizar** automaticamente os arquivos produzidos, utilizando metadados que facilitem uma busca ágil e precisa pelos usuários. Esse sistema poderia incluir funcionalidades inteligentes que **automaticamente identifiquem** o contexto dos documentos, sugerindo etiquetas, categorias ou conexões com outros materiais semelhantes, o que evitaria que os usuários perdessem tempo precioso procurando por arquivos específicos.

Adicionalmente, seria essencial propor mecanismos robustos para garantir a preservação contínua dos artefatos produzidos. Uma das ideias nesse sentido seria implementar uma **rotina automatizada** e periódica de **backups**, garantindo que cópias de segurança estejam disponíveis regularmente e protegidas contra acidentes ou falhas técnicas. Esses backups poderiam, inclusive, ser organizados de maneira incremental e histórica, permitindo que versões anteriores dos materiais sejam facilmente recuperadas, mesmo que tenham sofrido modificações ou exclusões acidentais.

A proposta também pode incluir um sistema detalhado de **versionamento** dos materiais, algo especialmente importante no contexto de documentos críticos, como textos acadêmicos, relatórios técnicos e código-fonte. Esse sistema poderia ser semelhante aos mecanismos utilizados em ferramentas como Git, permitindo que os usuários consigam acompanhar detalhadamente as mudanças feitas ao longo do tempo, comparar versões e até reverter alterações específicas, mantendo uma história completa e segura de cada artefato produzido.

Um ponto importante a ser considerado é a possibilidade de **auditoria** e controle da integridade dos arquivos. Um mecanismo automático poderia **monitorar** continuamente o estado dos documentos, códigos ou produções armazenadas, alertando imediatamente caso alguma inconsistência, corrupção ou perda seja detectada. Isso garantiria que o ambiente permanecesse sempre íntegro e confiável, sem depender exclusivamente de

verificações manuais ou periódicas feitas pelos usuários.

Além disso, seria valioso pensar em mecanismos de gestão de **permissões e acessos** específicos para os artefatos armazenados. Essa ideia consiste em oferecer aos usuários um controle detalhado sobre quem pode acessar ou modificar cada tipo de arquivo, definindo claramente responsabilidades, contribuindo para a segurança e evitando conflitos ou alterações indevidas.

Ainda pensando em gerenciamento avançado, podemos imaginar um sistema que permita o estabelecimento de **relações entre diferentes** artefatos, criando uma rede de conexões entre documentos, códigos, gravações de reuniões ou outras produções relacionadas. Essa rede poderia facilitar muito o entendimento dos projetos ou contextos de trabalho, fornecendo aos usuários uma visão global clara sobre o fluxo das atividades e suas respectivas produções.

Por fim, a ideia de **persistência dos artefatos** também envolve garantir que esses materiais permaneçam acessíveis não apenas agora, mas também no futuro, mesmo que tecnologias ou equipes sejam alteradas. Nesse sentido, é importante **propor padrões** claros e documentados sobre como os artefatos serão armazenados, identificados e acessados, prevendo a possibilidade de exportação ou migração futura para outras plataformas ou formatos, mantendo a independência do sistema atual e garantindo a continuidade histórica.

Dessa forma, com a proposta de gestão e persistência dos artefatos, estaremos garantindo não apenas que o conhecimento produzido seja armazenado com segurança, mas que também permaneça organizado, acessível e útil ao longo do tempo. Essa abordagem permite uma evolução segura e contínua do ambiente, protegendo o trabalho realizado e mantendo-o disponível de forma eficiente para futuras consultas ou reutilizações.

A tabela 6 sintetiza os requisitos levantados nessa seção.

2.5 Controle de Acesso Centralizado

Após estabelecer um ambiente que garante a preservação e organização de todo o conhecimento produzido, o próximo passo lógico é garantir que essas informações sejam protegidas adequadamente. Nesse contexto, precisamos propor mecanismos eficazes e detalhados para controlar o acesso às diversas áreas, ferramentas e artefatos disponíveis no sistema.

Primeiramente, é importante refletir sobre a diversidade de usuários que compõem o ambiente do laboratório. Cada pessoa ou grupo pode ter papéis, funções e níveis de **responsabilidade** diferentes, exigindo políticas específicas de acesso. Portanto, a proposta inclui a implantação de um sistema de autenticação e autorização centralizado, capaz de

Tabela 6 – Gestão e Persistência dos Artefatos

Requisito	**Descrição**
Organização Automática	Armazenamento e categorização dos artefatos com uso de metadados para facilitar buscas.
Backup Automático	Rotinas periódicas de backup, com versões incrementais para proteção dos dados.
Controle de Versões	Mecanismos para rastrear e comparar alterações, possibilitando a recuperação de versões anteriores.
Auditoria e Monitoramento	Verificação contínua da integridade dos arquivos, com alertas automáticos em caso de inconsistências.
Gestão de Permissões	Controle refinado sobre quem pode acessar ou modificar cada artefato armazenado.
Documentação	Padrões definidos para exportar ou migrar dados, garantindo a acessibilidade futura dos artefatos.

reconhecer rapidamente quem é o usuário, quais são suas permissões e qual é o seu nível de acesso aos diferentes recursos e conteúdos do sistema.

Para isso, podemos imaginar um ambiente que permita a criação flexível de **perfis e grupos** de usuários, possibilitando um gerenciamento simples e intuitivo de permissões e responsabilidades. Cada perfil poderia ser configurado detalhadamente, indicando quais áreas, documentos, ferramentas ou até mesmo tipos específicos de ações são permitidas ou negadas. Além disso, esses perfis poderiam ser facilmente alterados e ajustados ao longo do tempo, garantindo que o sistema se adapte rapidamente a mudanças organizacionais ou novas necessidades.

Um ponto relevante para a proposta de controle de acesso é a possibilidade de **auditoria** contínua e detalhada dos **acessos** realizados. Seria importante contar com mecanismos capazes de registrar em tempo real quem acessa quais recursos, quando isso ocorreu e quais ações foram realizadas durante aquele acesso. Esse registro detalhado permitiria que os administradores identificassem rapidamente possíveis acessos indevidos ou comportamento anormal, garantindo maior segurança e facilidade para investigar problemas quando necessário.

Podemos também imaginar mecanismos adicionais de **segurança adaptativa**, capazes de reagir dinamicamente diante de situações incomuns ou de tentativas suspeitas de acesso. Por exemplo, caso sejam detectadas tentativas repetidas ou incomuns de acesso não autorizado, o sistema poderia automaticamente restringir temporariamente determinados recursos, notificar os responsáveis ou até mesmo exigir níveis adicionais de autenticação para continuar permitindo o acesso.

Outro recurso interessante para complementar essa proposta é a adoção de **auten-**

ticação multifatorial, especialmente em situações mais críticas ou em acessos remotos. Nesse cenário, o sistema poderia exigir métodos adicionais de validação para garantir a identidade do usuário, diminuindo significativamente os riscos associados a vazamentos de senhas ou credenciais comprometidas.

Além disso, é fundamental considerar a flexibilidade na aplicação das permissões, permitindo que seja possível estabelecer **diferentes níveis de acesso** conforme contextos específicos. Por exemplo, um usuário pode ter permissões mais amplas em determinado projeto, enquanto em outro projeto suas permissões seriam mais restritas. Essa abordagem granular garantiria que os acessos estivessem sempre alinhados às reais necessidades operacionais e organizacionais.

Uma funcionalidade complementar que enriqueceria bastante o ambiente seria a possibilidade dos **próprios usuários** solicitarem **alterações ou ampliações** de suas **permissões**, de forma controlada e gerenciada por meio de fluxos automáticos de aprovação por administradores ou responsáveis específicos. Essa dinâmica simplificaria o gerenciamento e aumentaria a agilidade na adaptação dos níveis de acesso aos usuários conforme suas responsabilidades ou necessidades evoluem.

Também podemos propor um mecanismo de **integração** entre o **controle de acesso** e o sistema de **provisionamento inicial** discutido anteriormente, garantindo que novos usuários sejam automaticamente cadastrados com um conjunto básico de permissões e acessos padronizados, conforme a função ou posição que assumirem no laboratório. Dessa forma, novos usuários já começariam com o nível adequado de acesso sem necessidade de ajustes manuais constantes.

Com todas essas funcionalidades, o controle de acesso centralizado proposto seria abrangente, robusto e suficientemente flexível para atender às demandas dinâmicas do laboratório, garantindo segurança e eficiência na gestão dos recursos, das informações e das atividades desenvolvidas pelos usuários.

A tabela 7 sintetiza os requisitos levantados nessa seção.

2.6 Capacitação, Sustentabilidade e Gestão de conhecimento

Após termos discutido e proposto uma infraestrutura automatizada, padronizada, colaborativa e segura, precisamos abordar um aspecto igualmente importante: garantir que todo esse ambiente seja sustentável ao longo do tempo. A sustentabilidade, nesse contexto, não se restringe apenas a questões técnicas, mas também envolve a manutenção e evolução contínua das soluções adotadas, especialmente diante de mudanças naturais, como a troca de membros da equipe ou novos desafios tecnológicos.

Nesse sentido, é essencial propor mecanismos claros e eficazes de capacitação para

Tabela 7 – Controle de Acesso Centralizado

Requisito	**Descrição**
Autenticação Centralizada	Sistema unificado para gerenciamento de identidades e validação dos usuários.
Gestão de Perfis	Criação e gerenciamento flexível de perfis e grupos, com definição de níveis de acesso diferenciados.
Auditoria de Acessos	Registro detalhado das atividades de acesso, permitindo monitoramento em tempo real.
Segurança Adaptativa	Mecanismos que respondem a comportamentos suspeitos com restrições ou autenticação adicional.
Autenticação Multifatorial	Implementação de múltiplos métodos de verificação para acesso a recursos críticos.
Permissões Granulares	Definição detalhada de acessos, permitindo controle refinado sobre os recursos.
Integração com Provisionamento	Cadastro automático de novos usuários com permissões pré-definidas, alinhado ao provisionamento do ambiente.

os próprios usuários do sistema. A ideia é que cada pessoa envolvida tenha autonomia e segurança para operar, manter e até mesmo aprimorar o ambiente ao longo do tempo. Para isso, seria ideal estabelecer estratégias de **treinamento** e atualização constante, garantindo que todos estejam sempre alinhados com as funcionalidades e processos estabelecidos.

Uma proposta interessante seria a criação de um **ambiente de aprendizado** integrado diretamente ao próprio sistema do laboratório. Nesse ambiente, seria possível oferecer tutoriais interativos, documentação detalhada e materiais didáticos relacionados diretamente às funcionalidades existentes. Os usuários poderiam acessar conteúdos específicos sobre como realizar tarefas comuns, solucionar problemas técnicos ou mesmo executar processos mais complexos dentro da plataforma.

Além disso, poderíamos propor sessões periódicas de **treinamento presencial ou remoto**, onde seriam discutidas as melhores práticas de utilização do sistema, apresentadas novas funcionalidades, ou compartilhados casos reais de uso bem-sucedidos entre os usuários. Essas sessões seriam uma oportunidade para a equipe do laboratório se manter atualizada, trocar experiências e garantir que todos estejam aproveitando ao máximo as ferramentas disponíveis.

Outro aspecto relevante é garantir a existência de uma **documentação detalhada**, atualizada e acessível. A documentação proposta deveria cobrir desde a estrutura geral do sistema até detalhes mais específicos sobre sua operação, instalação, configuração, gerenciamento e recuperação diante de problemas. Uma ideia seria organizar essa

documentação em uma plataforma centralizada, oferecendo recursos de busca inteligente e categorização clara, permitindo que usuários e administradores localizem rapidamente a informação necessária.

Além disso, seria útil que essa documentação fosse **produzida de forma colaborativa**, permitindo que os próprios usuários possam contribuir com novas orientações, correções ou sugestões de melhorias ao longo do tempo. Essa estratégia garante que o material permaneça sempre atualizado e refletindo as necessidades reais de quem utiliza a plataforma no dia a dia.

Uma questão central é assegurar que as soluções propostas sejam tecnicamente **independentes de tecnologias específicas** que possam ficar **rapidamente obsoletas**. Cabe garantir a independência técnica das soluções propostas, de modo que não dependam de tecnologias específicas que possam se tornar obsoletas rapidamente. Dessa forma, a capacitação poderia incluir uma **abordagem conceitual**, onde os usuários aprendem não apenas como usar uma tecnologia ou ferramenta específica, mas também compreendem os conceitos e princípios subjacentes. Com isso, diante de uma mudança futura na infraestrutura ou nas tecnologias utilizadas, os usuários terão muito mais facilidade em adaptar-se rapidamente, garantindo continuidade operacional sem dificuldades.

Complementarmente, podemos considerar ainda um mecanismo de gestão do conhecimento interno, como um **fórum** ou plataforma interna, onde usuários possam compartilhar dificuldades, soluções, boas práticas e aprendizados adquiridos no uso diário das ferramentas. Essa interação coletiva ajudaria a preservar o conhecimento dentro da própria equipe, reduzindo a dependência de usuários específicos ou especialistas externos para resolver problemas recorrentes ou dúvidas comuns.

Por fim, pensando em sustentabilidade técnica e administrativa, seria relevante propor um processo estruturado de **transição entre equipes** ou gerações de usuários. Esse processo poderia incluir estratégias claras sobre como realizar a transferência eficiente do conhecimento, documentação das experiências adquiridas e organização de oficinas ou reuniões específicas para alinhar novas equipes às práticas e rotinas estabelecidas anteriormente.

Com todas essas estratégias combinadas, a proposta para capacitação e sustentabilidade do ambiente se tornaria extremamente robusta e resiliente, garantindo que o laboratório permaneça funcional, atualizado e eficiente, independentemente das mudanças tecnológicas, organizacionais ou de equipe ao longo do tempo.

A tabela 8 sintetiza os requisitos levantados nessa seção.

Tabela 8 – Capacitação e Sustentabilidade

Requisito	**Descrição**
Ambiente de Aprendizado Integrado	Plataforma com tutoriais interativos, materiais didáticos e documentação detalhada.
Sessões de Treinamento	*Workshops* e treinamentos regulares, presenciais ou remotos, para atualização dos usuários.
Documentação Atualizada	Documentação centralizada, clara e organizada, que abranja desde a estrutura do sistema até processos operacionais.
Contribuição Colaborativa	Mecanismo para que os próprios usuários atualizem e melhorem a documentação continuamente.
Gestão do Conhecimento Interno	Fóruns e plataformas para troca de experiências e compartilhamento de soluções entre usuários.
Atualização Automatizada da Documentação	Rotinas que verificam e atualizam a documentação conforme mudanças no sistema.
Abordagem Conceitual	Capacitação que foca em ensinar os princípios subjacentes, permitindo adaptações futuras independentemente das ferramentas.
Transição de Equipe Estruturada	Procedimentos claros para transferência de conhecimento entre diferentes gerações de usuários.

3 Provisionamento e Gerenciamento de Configurações

Conceitos relacionados

Segundo Red Hat (2023b), o provisionamento de sistemas é o processo de preparação e configuração de computadores para que estejam prontos para uso, garantindo que o sistema operacional e os softwares necessários sejam instalados de maneira adequada. Esse processo pode ser realizado de forma manual ou automatizada, dependendo da escala e da complexidade da infraestrutura.

O provisionamento abrange desde a instalação do sistema operacional até a configuração de aplicações e recursos necessários para que a máquina possa desempenhar suas funções corretamente. Esse processo pode ter diferentes abordagens, como a instalação manual por meio de mídias físicas (pendrive, DVD) ou instalação remota via PXE Boot. Além disso, pode envolver o uso de imagens de sistema previamente configuradas ou a automação do processo de instalação por meio de ferramentas como Preseed, Kickstart e Autoinstall.

Enquanto o provisionamento tem como objetivo "inicializar"a máquina com um sistema operacional e suas configurações, o conceito de gerenciamento de configurações vai um pouco além. O gerenciamento de configurações tem como objetivo a manutenção e padronização do estado desejado dos sistemas já provisionados. Isso inclui a instalação e configuração de softwares, aplicação de políticas de segurança, gerenciamento de usuários e definição de permissões. Ferramentas como Ansible, Puppet, Chef e SaltStack permitem que essas configurações sejam aplicadas de forma automatizada e consistente (Red Hat, 2023a).

Métodos de Instalação de Sistemas Operacionais

Existem diferentes métodos para se instalar um sistema operacional, como por exemplo o uso de um instalador fornecido pelos desenvolvedores, ou por meio da cópia de uma imagem de um sistema pronto (GEEKSFORGEEKS, 2021).

Por instalador

As distribuições Linux geralmente contam com instaladores interativos que através de opções apresentadas, realiza a instalação do sistema operacional. Existem métodos para

se personalizar e automatizar este processo, mas iremos discutir sobre eles mais adiante. Existem alguns métodos diferentes para usar um instalador:

- Mídias físicas (Pendrive, DVD, etc.) Este método exige o uso de alguma mídia física como discos e pendrives conectados à maquina.
  - **Vantagens:** Simplicidade, independência de rede.
  - **Desvantagens:** Exige intervenção manual, não escalável, exige ter mídias físicas à disposição.
- PXE Boot (Network Booting)
  - **Vantagens:** Facilita a instalação remota e em larga escala.
  - **Desvantagens:** Requer infraestrutura de rede configurada (Oracle, 2017).

Instalação por Imagem

Método que usa uma cópia pronta do sistema para agilizar a implantação, geralmente usado em conjunto de ferramentas como Clonezilla, FOG Project e DRBL.

- **Vantagens:** Rápido para replicar sistemas idênticos.
- **Desvantagens:** Exige captura e manutenção de imagens, menos flexível para personalização.

Automação de Instalação e Configuração

Para automatizar o processo de instalação de uma distribuição Linux, existem diferentes abordagens que eliminam a necessidade de intervenção manual, tornando a implantação de sistemas mais rápida e padronizada. Entre essas opções, destacamos dois métodos: arquivos de resposta automatizada e a customização do instalador.

- **Arquivos de resposta automatizada:** esses arquivos contêm todas as informações necessárias para a instalação, como configurações de partição, pacotes a serem instalados e parâmetros do sistema, permitindo que a instalação ocorra sem interação do usuário. Preseed (Debian-based), Kickstart (Red Hat-based), Autoinstall (Ubuntu Server 20.04+) são exemplos de arquivos de resposta automatizada.
  - **Vantagens:** Simples, não requer manutenção do instalador ou de uma imagem.
  - **Desvantagens:** Ainda requer o uso de um instalador.

- **Customização do Instalador:** Outra abordagem é a customização dos instaladores, onde a imagem de instalação da distribuição é modificada para incluir pacotes específicos, configurações pré-definidas e até scripts personalizados. Isso permite criar uma mídia de instalação já adaptada às necessidades do ambiente, facilitando a implantação em larga escala.
  - **Vantagens:** Configuração padronizada, sem necessidade de interação manual.
  - **Desvantagens:** Requer manutenção da imagem.

Gerenciamento de configurações

O gerenciamento de configurações tem como objetivo a manutenção e padronização do estado desejado dos sistemas já provisionados, através de ferramentas (Red Hat, 2023a) como Ansible, Puppet, Chef, SaltStack e também pode ser realizado através de um gerenciador de pacotes em distribuições Linux.

Pacotes, Meta-Pacotes e Gerênciadores de Pacotes

Um gerenciador de pacotes em uma distribuição Linux é uma ferramenta que automatiza a instalação, atualização, remoção e gerenciamento de softwares em um sistema operacional. Ele facilita a manutenção do sistema ao lidar com dependências entre pacotes, garantindo que todos os componentes necessários sejam instalados corretamente. O gerenciador de pacotes durante a instalação irá se encarregar de colocar os arquivos nos locais corretos, e baixar e instalar os softwares e bibliotecas listados na lista de dependência do pacote (FALKO, 2007).

Um pacote em uma distribuição Linux é um arquivo que contém todos os elementos necessários para que um software específico seja instalado e funcione corretamente no sistema. Isso inclui os arquivos executáveis, bibliotecas, scripts de instalação, documentação e metadados que descrevem o pacote, como seu nome, versão e a lista de dependências.

Já os Meta-pacotes são pacotes que funcionam como uma coleção de aplicações que serão instaladas em conjunto. Eles são pacotes "vazios", ou seja, não contêm os softwares em si, mas uma lista de dependências. Isso permite que vários programas e bibliotecas que normalmente seriam instalados separadamente possam ser agrupados em uma única instalação. Esse agrupamento facilita a configuração de ambientes de software específicos, como um conjunto de ferramentas para desenvolvimento ou um pacote de software para edição de mídia, por exemplo. Além disso, esses meta pacotes podem incluir arquivos de configuração, o que ajuda a garantir que todos os softwares funcionem conforme esperado logo após a instalação. (AOKI, 2024) (Ubuntu Community, 2024)

APT

O APT (Advanced Packaging Tool) é o gerenciador de pacotes usado nas distribuições baseadas em *Debian* como o *Ubuntu* e o *PopOS*. Ele é um gerenciador de pacotes estável que lida com pacotes binários e gerencia múltiplas versões de cada pacote em seu repositório. (AOKI, 2024) Assim como outros, o APT possui o conceito de meta-pacotes, permitindo que aplicações sejam agrupadas em pacotes (Ubuntu Community, 2024).

Os PPAs, ou Personal Package Archives, são repositórios de software usados principalmente no Ubuntu para distribuir e atualizar pacotes que não estão disponíveis nos repositórios oficiais ou que são versões mais recentes (ou alternativas) dos pacotes que estão nos repositórios oficiais. Eles permitem que qualquer um publique suas próprias versões de pacotes, facilitando o acesso a softwares novos ou personalizados sem esperar pela integração no repositório principal do Ubuntu. Os PPAs não são suportados oficialmente pelo Debian, pelo fato de que foram criados pensando no ubuntu, que não necessáriamente utiliza as mesmas versões de pacote, causando problemas de incompatibilidade. (OVENS, 2018)

Pacman

O Pacman é o gerenciador de pacotes do Arch Linux. Tal como o APT, o Pacman também gerencia pacotes binários. No entanto, diferente do APT, o Pacman adota a abordagem rolling release, o que significa que mantém somente a versão mais recente de cada pacote em seus repositórios. Além do repositório oficial, também existe a AUR (Arch User Repository), que é um repositório gerenciado pela própria comunidade, no qual qualquer pessoa pode submeter um pacote. Além do repositório oficial, o Arch Linux também conta com a AUR (Arch User Repository), um repositório gerenciado por membros da comunidade onde qualquer usuário pode submeter um pacote. (MCRAE et al., 2024) (ARCHLINUXWIKI, 2024a)

Assim como o APT, o Pacman também suporta o conceito de meta pacotes. Esses pacotes não contêm arquivos de software propriamente ditos, mas são usados para agrupar várias dependências sob um único nome de pacote. Isso facilita a instalação e gerenciamento de grupos de pacotes relacionados, permitindo aos usuários instalar múltiplos softwares e suas dependências de forma simplificada e eficiente. (ARCHLINUXWIKI, 2024b)

Nix

Nix é um sistema que gerencia pacotes e configurações. Desenvolvido para o NixOS, porém pode ser usado em outras distribuições Linux. Ele cria ambientes isolados com configurações e dependências consistentes. Para isso, usa um arquivo que define as dependências e configurações de um pacote, criando um ambiente específico para ele. Cada

pacote é instalado em um local único, definido por um identificador próprio, o que evita conflitos de dependências e permite a reprodutibilidade. (TEAM, 2024a)

O Nix oferece soluções para a criação de meta pacotes, mas é ainda mais poderoso com perfis. Perfis no Nix são conjuntos de pacotes e configurações que podem ser gerenciados de forma independente e ativados ou desativados conforme necessário. Eles permitem manter múltiplas versões de um software instaladas simultaneamente e alternar entre diferentes conjuntos de software e configurações sem causar conflitos. Cada perfil possui seu próprio ambiente, facilitando a criação de ambientes de desenvolvimento ou produção consistentes e isolados. (TEAM, 2024b)

Implantação

Nesta seção, exploramos o processo de provisionamento e padronização de configuração do ambiente, que nasceu da necessidade de garantir uma base sólida e uniforme para todas as estações de trabalho. A partir dos desafios identificados foram levantados requisitos para automatizar e uniformizar a instalação dos sistemas. O primeiro passo na implantação deste projeto foi o provisionamento e padronização do ambiente do laboratório.

Utilizando o Debian Preseed, conseguimos definir previamente todas as configurações essenciais, como idioma, layout de teclado, particionamento do disco e instalação de pacotes específicos de nosso PPA. Essa técnica garante que cada máquina seja configurada de maneira uniforme, atendendo aos critérios de automatização e padronização, além de facilitar a manutenção do ambiente.

Além disso, para assegurar que todas as máquinas operem com os mesmos conjuntos de softwares essenciais, criamos um Personal Package Archive (PPA) com pacotes customizados, que incluem desde ferramentas de autenticação e montagem de diretórios até soluções para produção de áudio e desenvolvimento. Desta forma, para que o sistema esteja atualizado com os softwares e configurações necessárias, basta realizar uma atualização tradicional do sistema, que o próprio gerenciador de pacotes será responsável por aplicar as novas modificações. Com isso cumprimos parcialmente o requisito de *Bootstraping* e Verificação de integridade, já que esse papel agora é responsabilidade do gerenciador de pacotes.

Dessa forma, esta seção detalha como cada ferramenta e método foi empregado para transformar um ambiente disperso em uma infraestrutura sólida, confiável e pronta para suportar as demandas do laboratório. A tabela 9 mostra quais requisitos foram cumpridos na implantação.

Tabela 9 – Requisitos de Provisionamento e Padronização e seu cumprimento na implementação

Requisito	**Cumprido?**
Automatização do Provisionamento	Sim
Padronização do Ambiente	Sim
Bootstrapping e Verificação de Integridade	Sim
Perfis Dinâmicos	Não
Autorrecuperação	Parcial
Provisionamento Seletivo	Não

Automatização do Provisionamento

Para evitar a necessidade de instalação manual do sistema operacional em cada máquina, foi utilizada a abordagem baseada no *Debian Preseed*, um método que permite automatizar completamente a instalação da distribuição Debian. Com essa técnica, foi possível definir todas as configurações essenciais previamente, eliminando a necessidade de intervenção do usuário durante o processo de instalação.

O arquivo Preseed [1] utilizado contém definições para as seguintes configurações:

- Idioma, layout de teclado e fuso horário.
- Usuário padrão.
- Particionamento de disco e configuração do sistema de arquivos.
- Instalação automática de pacotes do PPA do projeto.

Essa abordagem permitiu que a instalação e configuração de uma máquina do laboratório pudesse ser concluída rapidamente, garantindo que todas as máquinas operassem sob um ambiente uniforme e previsível.

Padronização do Ambiente

Além da instalação do sistema operacional, era necessário garantir que todas as máquinas possuíssem as mesmas configurações e os mesmos conjuntos de softwares essenciais para suas respectivas funções. Entre as opções discutidas no capítulo anterior, a possibilidade de se utilizar o próprio gerenciador de pacotes do sistema para realizar o gerenciamento de configurações se pareceu a mais atrativa. Criamos então um *Personal Package Archive* [2] (PPA), com pacotes e meta-pacotes com configurações personalizadas para nosso contexto. Desta forma conseguimos garantir que todas as máquinas que utilizem nossos pacotes terão uma configuração identica gerenciada pelo próprio APT.

[1] Disponível em <https://git.alice.ufsj.edu.br/alice/preseed/>.
[2] Disponível em <https://alice.ufsj.edu.br/ppa/>.

O PPA contém pacotes específicos para diferentes propósitos, agrupados em meta-pacotes que simplificam sua instalação. Entre os pacotes disponibilizados, destacam-se:

- **alice-auth**: Responsável por configurar a autenticação via LDAP e a montagem automática de diretórios remotos via NFS.
- **alice-wol**: Implementa um serviço *systemd* para ativar o *Wake-on-LAN*, permitindo a inicialização remota das máquinas.
- **alice-hostname**: Define dinamicamente o *hostname* das máquinas com base no endereço IP atribuído.
- **alice-system-core**: Conjunto de pacotes essenciais para o funcionamento básico das máquinas do laboratório.
- **alice-audio-production**: Contém ferramentas voltadas para produção e edição de áudio, como o LMMS, Mixxx, Jack e Ardour.
- **alice-dev-core**: Inclui compiladores, interpretadores e ferramentas básicas para desenvolvimento de software.
- **alice-dev**: Conjunto expandido de ferramentas de desenvolvimento, incluindo IDEs e frameworks específicos.
- **alice-full**: Pacote que agrega todos os pacotes anteriores, fornecendo um ambiente completo para qualquer atividade no laboratório.

Para garantir que usuários menos técnicos fossem capazes de realizar a manutenção do sistema, a fontes de todos os pacotes foi armazenada e versionada com o Gitea, e foi desenvolvido um fluxo de trabalho para a sua construção e publicação automática. Isso também permite que novos pacotes sejam criados publicados automaticamente no PPA. Dessa forma, a adição de novas ferramentas ao ambiente do laboratório pode ser realizada sem a necessidade de intervenção manual no servidor. A figura 1 mostra uma captura de tela da organização que possui os pacotes no Gitea [3].

Gerenciamento de serviços do servidor

Para garantir uma administração eficiente dos serviços e facilitar a escalabilidade da infraestrutura, optou-se por executar a maior parte das aplicações do servidor dentro de contêineres Docker. Essa abordagem permite que cada serviço seja executado de forma isolada, garantindo que atualizações, reconfigurações e manutenções possam ser feitas sem impactar o restante do sistema.

[3] Disponível em <https://git.alice.ufsj.edu.br/alice-meta-packages/>.

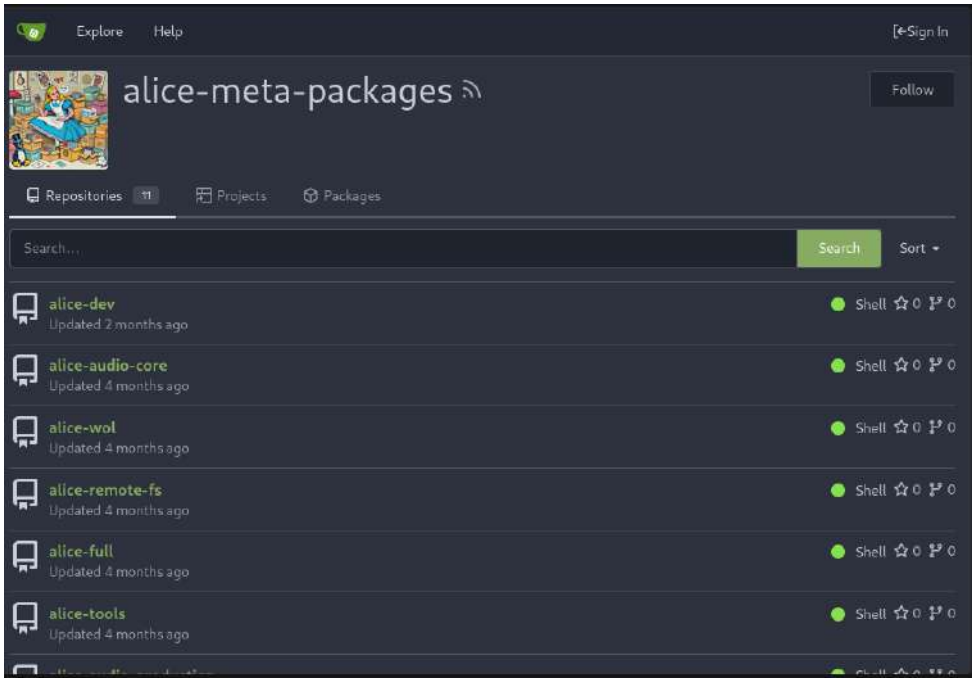

Figura 1 – Captura de tela do Gitea mostrando os pacotes do PPA

A organização dos contêineres é feita utilizando o Docker Compose, permitindo definir toda a configuração dos serviços em arquivos declarativos. Esses arquivos *docker-compose.yml* são armazenados e versionados no Gitea [4], garantindo que qualquer alteração na infraestrutura seja registrada e possa ser revertida, se necessário. Isso possibilita uma implantação replicável, onde qualquer nova instância do servidor pode ser rapidamente provisionada com as mesmas configurações. Para a visualização de estatísticas e informações sobre os contêineres utilizamos o Portainer [5]. A figura 2 mostra a interface do Portainer com os serviços sendo executados.

Dentre os serviços que foram implantados em contêineres, destacam-se:

- **Infraestrutura de autenticação e gerenciamento**: LDAP, Authelia, phpLDAPadmin e Portainer.
- **Ferramentas colaborativas**: Nextcloud, Etherpad, Gitea, Funkwhale e Jitsi Meet.
- **Serviços de proxy e segurança**: Traefik, que gerencia a comunicação entre os serviços e a internet, garantindo roteamento eficiente e provisionamento automatizado de certificados SSL.
- **Soluções de armazenamento e versionamento**: Filestash e os serviços relacionados à sincronização de arquivos.

[4] Disponível em <https://git.alice.ufsj.edu.br/alice-docker/>.
[5] Disponível em <https://docker.alice.ufsj.edu.br/>.

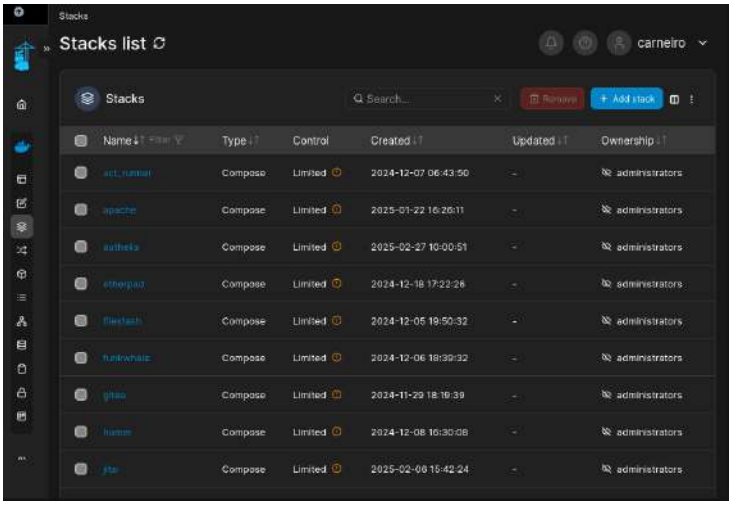

Figura 2 – Captura de tela da aplicação Portainer, nela podemos ver uma listagem dos projetos compose em execução

A escolha do Traefik como proxy reverso permitiu um controle eficiente do tráfego dentro da infraestrutura, garantindo que cada aplicação pudesse ser acessada de maneira segura e sem necessidade de configuração manual de portas. O Traefik também foi integrado ao Authelia, permitindo que determinados serviços exijam autenticação LDAP para acesso, reforçando a segurança da infraestrutura. A figura 4 demonstra como ficou a arquitetura de proxy reverso. A figura 3 mostra uma captura de tela da ferramenta Traefik dash [6], que disponibiliza informações sobre o reverse proxy do sistema.

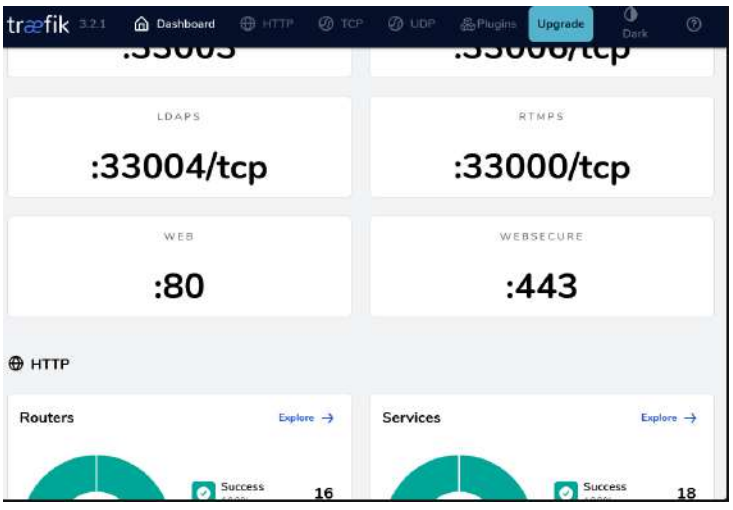

Figura 3 – Captura de tela da aplicação Traefik Dash, nela podemos ver informações sorbe o reverse proxy

Para garantir um controle eficiente da infraestrutura, toda a configuração do servidor foi versionada e documentada. Os arquivos de configuração dos contêineres e scripts auxiliares foram armazenados no Gitea, permitindo um rastreamento detalhado das alte-

[6] Disponível em <https://traefik.alice.ufsj.edu.br/>.

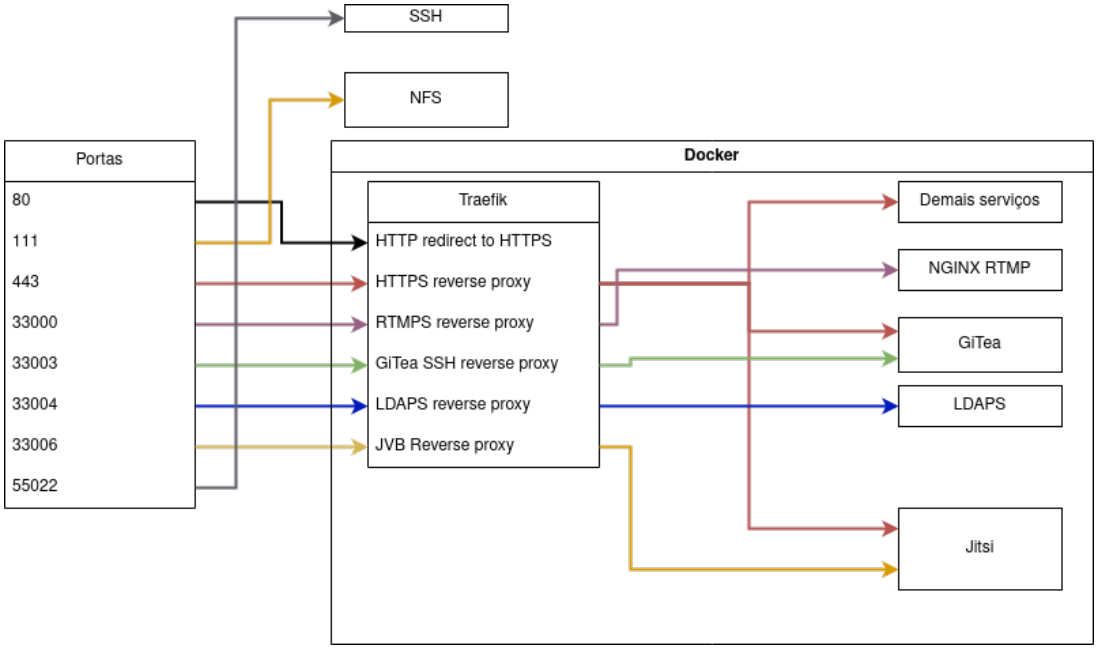

Figura 4 – Arquitetura de gerenciamento de portas e serviços no servidor. As portas de SSH e NFS são redirecionadas diretamente para serviços rodando no próprio sistema, fora do ambiente Docker. As demais portas são gerenciadas por um reverse proxy Traefik, que distribui o tráfego externo para os containers Docker correspondentes. Entre esses serviços estão o servidor de videoconferência Jitsi, o Gitea para controle de versões, um servidor LDAPS para autenticação, um NGINX RTMP para streaming e outros serviços web. Essa configuração permite centralizar o gerenciamento de certificados, roteamento e acesso, mantendo o servidor organizado e seguro

rações realizadas. Essa abordagem facilita a recuperação do ambiente em caso de falhas, além de permitir que novas máquinas possam ser provisionadas rapidamente com a mesma configuração.

O uso de contêineres, aliado ao versionamento das configurações e à autenticação centralizada, garantiu um ambiente modular, seguro e facilmente replicável. Com essa abordagem, é possível não apenas manter a infraestrutura do laboratório funcionando de maneira eficiente, mas também permitir que futuras expansões e ajustes sejam realizados sem comprometer a estabilidade do sistema.

4 Mobilidade de usuários e compartilhamento de arquivos

Conceitos relacionados

Compartilhar arquivos é tornar acessíveis dados digitais para outras pessoas ou dispositivos. Isso significa que um usuário pode disponibilizar documentos, fotos, vídeos e outros conteúdos de forma que outros possam acessar, baixar ou até mesmo modificar esses arquivos (JOHN, 2014). Essa ação pode ser feita de modos simples – como enviar por e-mail ou usar um pendrive – ou por meios mais elaborados, como serviços na nuvem, protocolos de transferência de arquivos e sistemas de arquivos distribuídos, que atendem a necessidades técnicas específicas. Em resumo, o compartilhamento de arquivos é a prática de distribuir ou facilitar o acesso a conteúdos digitais, adaptando a forma de acordo com a situação ou ambiente de trabalho.

Alguns protocolos foram criados para realizar transferência de arquivos, como FTP e o SFTP. O FTP é um protocolo utilizado para a transferência de arquivos entre um cliente e um servidor em uma rede. É conhecido por sua simplicidade e ampla compatibilidade, mas tem a desvantagem de não ser seguro, pois transmite dados em texto claro, incluindo senhas. Por isso, é mais adequado para redes seguras ou situações onde a segurança não é uma preocupação principal (PRESS, 2008). Existe também o SFTP, que é uma versão do FTP que utiliza do protocolo SSH (GASSER; HOLZ; CARLE, 2014). Diversos outros protocolos existem para este proposito, como o WebDAV que opera sob o HTTP (VILLANUEVA, 2024), mas não são tão relevantes no contexto deste trabalho.

Sistemas de Arquivos Distribuídos

Um Sistema de Arquivos Distribuído (DFS) permite que usuários de computadores fisicamente separados compartilhem dados e recursos de armazenamento através de uma rede comum (SILBERSCHATZ; GALVIN; GAGNE, 2006). Ele é integrado ao sistema operacional de cada computador conectado, possibilitando a utilização de um sistema de arquivos único e compartilhado (LEVY; SILBERSCHATZ, 1990).

Segundo LEVY; SILBERSCHATZ, nos sistemas de arquivos distribuídos (DFS), há dois métodos principais para acessar arquivos armazenados remotamente: o método de serviço remoto e o método de cache. No serviço remoto, a troca de arquivos entre cliente e servidor é constante, qualquer modificação ou leitura dispara uma troca entre eles, isso em uma rede lenta isso pode se tornar uma experiência horrível. No cache, a

troca de arquivos é mediada por um sistema que faz uma cópia local deles, a fim de evitar a comunicação constante com o servidor, mas por isso existe um risco de inconsistência quando dois clientes estão acessando os arquivos simultaneamente. Alguns dos sistemas de arquivos remotos que existem são NFS (NOVECK; HAYNES, 2015) (Network File System) e SSHFS (SSHFS, 2024) (SSH File System).

O NFS (Network File System) é um protocolo de sistema de arquivos distribuído que permite que diferentes computadores acessem arquivos uns dos outros como se fossem locais. Foi desenvolvido pela Sun Microsystems em 1984 com o objetivo de facilitar o compartilhamento de arquivos entre sistemas Unix(SANDBERG, 1986).

Seu funcionamento baseia-se no uso do protocolo de rede TCP/IP para transmitir dados. Ele utiliza um servidor NFS para armazenar e gerenciar os arquivos, enquanto os clientes NFS se conectam a esse servidor para acessar os arquivos (SHEPLER et al., 2000). Isso permite a centralização de dados, facilitando o gerenciamento e acesso. Ele pode apresentar problemas de desempenho em redes lentas e questões de segurança, uma vez que a transmissão de dados pode não ser criptografada (SANDBERG, 1986).

Já o SSHFS (SSH File System) permite montar sistemas de arquivos remotos usando o protocolo SSH. Utilizando criptografia, SSHFS transmite os dados em segurança. No entanto, pode ser mais lento em redes de alta latência e requer uma conexão estável. (Bartosz Fenski, 2011)

Existem também soluções completas para realizar o compartilhamento de arquivos. Essas soluções geralmente tem diversas funcionalidades além de apenas transferir arquivos, como interfaces próprias, sistemas avançados de permissão, serviços de busca e indexação. Alguns exemplos são Google Drive, Dropbox, iCloud, dentro muitas outras plataformas proprietárias. Plataformas de software livre auto-hospedaveis também existem, como Nextcloud, Owncloud, e Filestash.

Implantação

O segundo passo na implantação deste projeto é garantir a mobilidade dos usuários dentro do laboratório, permitindo que os usuários consigam trabalhar em qualquer máquina, tendo acesso aos seus arquivos pessoais e ambiente de forma transparente. Esta seção detalha como foi garantida a mobilidade dos usuários e a sincronização de seus arquivos pessoais no laboratório. Diante da necessidade de oferecer um ambiente personalizado e consistente, independentemente da máquina utilizada, adotamos uma abordagem que integra autenticação centralizada com acesso dinâmico aos dados dos usuários.

Através do servidor LDAP e do cliente SSSD nas máquinas, os usuários podem acessar qualquer dispositivo do laboratório com suas credenciais únicas, cumprindo o pri-

meiro passo de permitir que utilizem múltiplas máquinas configuradas igualmente usando as mesmas credenciais. Complementando essa integração, implantamos um diretório NFS no servidor para que os diretórios pessoais sejam montados automaticamente ao efetuar login, usando informações sobre o ponto de montagem que são armazenadas diretório. Ao adotarmos diretórios pessoais no NFS, garantimos que tanto os arquivos quanto as configurações individuais dos usuários sejam mantidos de forma consistente em todas as máquinas. Dessa forma, independentemente do dispositivo utilizado, o usuário encontra seu ambiente configurado conforme suas preferências, atendendo ao requisito de oferecer um ambiente personalizado e de sincronização automática.

Além disso, soluções como o clientes SFTP e o cliente web Filestash permitem o acesso remoto aos arquivos mesmo fora na rede da Universidade, garantindo que os dados estejam sempre disponíveis, cumprindo o requisito de acesso remoto.

Esta estratégia não apenas cria um ambiente unificado e flexível, mas também facilita a retomada do trabalho exatamente de onde o usuário parou, contribuindo para uma experiência de uso mais fluida e eficiente. A tabela 10 mostra quais requisitos foram cumpridos na implantação.

Tabela 10 – Requisitos de Mobilidade do Usuário e Sincronização de Arquivos

Requisito	**Cumprido?**
Ambiente Personalizado	Sim
Sincronização Automática	Sim
Continuidade de Sessão	Não
Histórico de Atividades	Não
Acesso Remoto Integrado	Sim

Integração com o Sistema de Autenticação

Para garantir que os usuários possam acessar qualquer máquina do laboratório sem necessidade de criar contas locais separadas, todas as máquinas foram integradas ao servidor LDAP. A autenticação dos usuários e a atribuição de permissões são gerenciadas centralmente, permitindo que cada membro do laboratório utilize suas credenciais únicas para acessar qualquer dispositivo.

Essa integração foi realizada utilizando o SSSD (*System Security Services Daemon*), um serviço que permite que máquinas clientes autentiquem usuários diretamente em um servidor LDAP. Além da autenticação de usuários, o SSSD também foi configurado para gerenciar informações como:

- Grupos.
- Hosts, permitindo identificação dinâmica das máquinas.

- Pontos de montagem do NFS para a montagem automática de diretórios pessoais.
- Regras de *sudo* armazenadas no LDAP, permitindo gerenciamento centralizado de permissões administrativas.

O diagrama da Figura 5 ilustra essa abordagem.

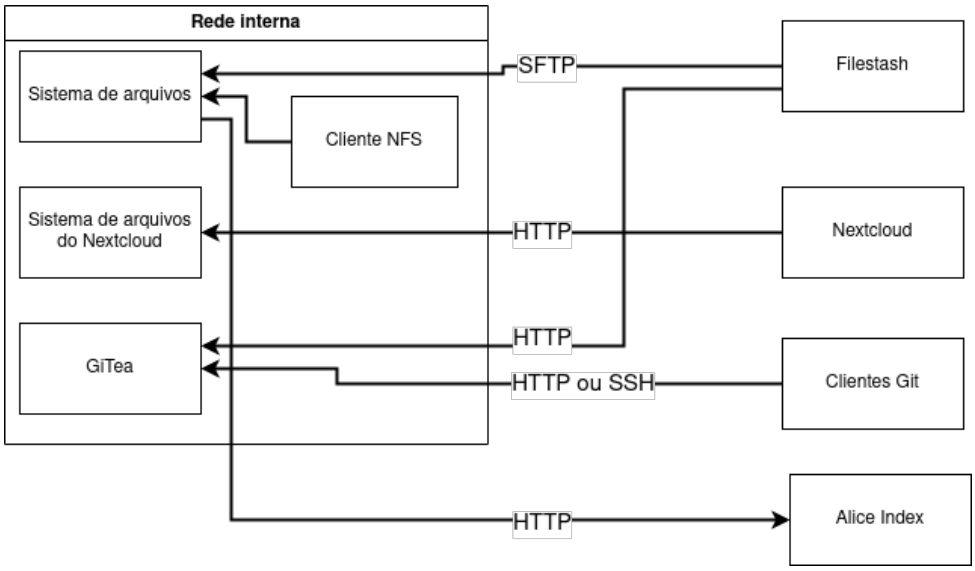

Figura 5 – Arquitetura de compartilhamento e acesso a arquivos e repositórios no laboratório. A imagem representa a infraestrutura de serviços internos utilizados para o armazenamento e distribuição de arquivos no ambiente do laboratório. À esquerda, encontram-se os sistemas de arquivos locais e serviços autohospedados, acessíveis apenas dentro da rede interna. O armazenamento é exposto via diferentes protocolos de rede, de acordo com a aplicação: o sistema de arquivos geral pode ser acessado via SFTP através do FileStash, ou via NFS por clientes internos. O sistema de arquivos do Nextcloud é exposto via HTTP e acessado diretamente pelo próprio Nextcloud. O serviço Gitea serve repositórios Git que podem ser acessados via HTTP ou SSH por clientes Git, além de ser acessado por navegadores via HTTP. Por fim, o Alice Index acessa conteúdos por meio de requisições HTTP

Além disso, para evitar conflitos de nomenclatura entre as máquinas, foi implementado um *script* executado na inicialização do sistema que define automaticamente o nome da máquina com base em seu endereço IP, consultando o banco de dados do LDAP. Dessa forma, evita-se a necessidade de configurar manualmente os nomes de host de cada máquina.

Sincronização de Diretórios Pessoais

Uma das necessidades identificadas foi garantir que os usuários tivessem acesso aos seus arquivos pessoais em qualquer máquina do laboratório. Para isso, optou-se por

utilizar o NFS (*Network File System*), permitindo que os diretórios `/nethome` dos usuários fossem montados automaticamente em qualquer máquina na qual realizassem login.

Essa montagem foi configurada para ocorrer dinamicamente utilizando o *autofs*, que detecta a presença de um usuário autenticado e monta automaticamente seu diretório pessoal a partir do servidor NFS. Com essa solução, cada usuário pode acessar seus arquivos de qualquer máquina do laboratório sem a necessidade de transferências manuais.

Além do NFS, foi disponibilizado o acesso remoto aos arquivos pessoais via SFTP no servidor principal, permitindo que usuários acessem seus dados mesmo fora do laboratório. Para facilitar esse acesso, foi implantado o Filestash [1], um cliente Web para SFTP que permite navegar pelos arquivos diretamente de um navegador, sem necessidade de instalar software adicional. A figura 6 mostra a tela de login com os diferentes clientes pre-configurados, enquanto a figura 7 mostra a tela após a conexão SFTP.

Figura 6 – Captura de tela do Filestash na tela de seleção de clientes

[1] Disponível em <https://files.alice.ufsj.edu.br/>.

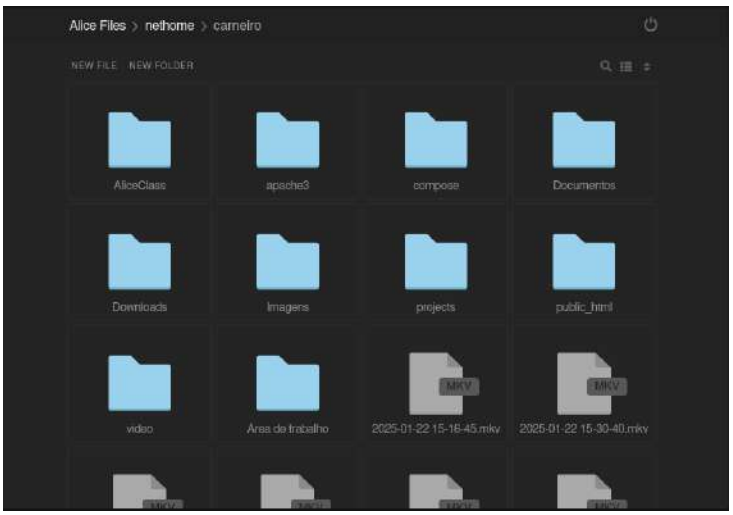

Figura 7 – Captura de tela do Filestash, a imagem mostra um usuário conectado visualizando sua pasta pessoal

5 Ferramentas Colaborativas

Conceitos relacionados

Sistemas de Versionamento de Código

Sistemas de controle de versão são usados para gerenciar mudanças em documentos, programas, e outros tipos de arquivos. Eles mantêm um registro das alterações feitas, mostrando quem mudou algo, o que foi mudado e quando. Isso ajuda a evitar problemas quando várias pessoas trabalham no mesmo arquivo ao mesmo tempo. Eles também permitem que você veja versões antigas de um arquivo. (ZOLKIFLI; NGAH; DERAMAN, 2018) (STEFAN, 2009)

Sistemas de controle de versão podem ser centralizados ou descentralizados. Nos sistemas centralizados, como o Subversion (SVN), todos os arquivos e históricos de revisão são armazenados em um servidor central. Isso significa que para acessar ou modificar os arquivos, os usuários precisam se conectar a esse servidor. Esse modelo pode simplificar a administração mas depende muito da disponibilidade do servidor central.

Já nos sistemas descentralizados, como o Git e o Mercurial, cada usuário tem uma cópia completa do repositório, incluindo todo o histórico de alterações. Isso permite que os usuários trabalhem de forma independente e só sincronizem suas alterações com outros quando necessário. Esse modelo oferece maior flexibilidade e robustez, especialmente em ambientes onde os colaboradores não estão sempre conectados à mesma rede. (ZOLKIFLI; NGAH; DERAMAN, 2018)

O Git é uma ferramenta usada em projetos de desenvolvimento de software para controle de versão distribuído. Permite que os desenvolvedores gerenciem e registrem alterações nos arquivos de um projeto ao longo do tempo, facilitando a colaboração e a manutenção do código-fonte. Com sua estrutura distribuída, possibilita que vários desenvolvedores trabalhem simultaneamente em diferentes partes do código, mantendo um histórico detalhado das mudanças realizadas. Além disso, sua integração com plataformas de hospedagem de código, como GitHub, GitLab e Bitbucket, simplifica a colaboração remota e a revisão de código. Vamos agora entender em detalhes os conceitos fundamentais que envolvem o Git:

- **Repositório:** No Git, um repositório é o local onde todos os arquivos e o histórico de um projeto são armazenados. Ele pode ser local, mantido no computador do desenvolvedor, ou remoto, hospedado em um servidor. Enquanto o repositório local contém todos os arquivos do projeto e seu histórico completo de alterações, o

repositório remoto é uma cópia acessível por outros membros da equipe.

- **Commits:** Cada commit no Git representa uma alteração específica feita em um ou mais arquivos do projeto. Acompanhado por uma mensagem descritiva, um commit registra o progresso do projeto ao longo do tempo e fornece um histórico detalhado das modificações realizadas.

- **Branches:** As branches no Git são ramificações do código principal do projeto. Elas permitem que os desenvolvedores trabalhem em novas funcionalidades ou correções sem afetar o código principal. Cada branch representa uma linha de desenvolvimento separada, o que facilita o trabalho paralelo e a organização do trabalho em equipe.

- **Merges:** O merge é o processo de combinar as alterações de uma branch em outra. Quando uma funcionalidade ou correção é concluída em uma branch, as alterações podem ser mescladas de volta ao ramo principal do projeto. Isso permite a integração organizada do trabalho desenvolvido em diferentes branches.

- **Rebase:** O rebase é uma operação no Git que permite reorganizar o histórico de commits de uma branch, baseando-a em uma nova base. Essa operação é frequentemente usada para "limpar"o histórico de commits, removendo commits desnecessários ou reorganizando-os de forma mais lógica. Ao reorganizar os commits, o rebase pode ajudar a manter um histórico de alterações mais claro e coeso, facilitando a revisão e o entendimento do desenvolvimento do projeto ao longo do tempo.

- **Blame:** O blame, ou "responsabilização", é uma funcionalidade do Git que permite identificar quem fez uma alteração específica em um arquivo. Com o comando git blame, é possível visualizar o autor de cada linha de um arquivo, juntamente com o commit em que a alteração foi feita. Essa funcionalidade é útil para rastrear a origem de um código específico, entender o contexto de uma mudança e atribuir responsabilidades aos colaboradores. O blame pode ser uma ferramenta valiosa para a revisão de código, investigação de bugs e comunicação eficaz entre os membros da equipe.

Plataformas de Comunicação e Videoconferência

Jitsi é um conjunto de ferramentas de código aberto, projetado para fornecer soluções eficazes de comunicação via internet, com um foco especial em videoconferências. Dentre os componentes notáveis do Jitsi, destacam-se o Jitsi Meet, o Jitsi Videobridge e o Jicofo, cada um desempenhando um papel vital na otimização da comunicação online.

- Meet(Jitsi, 2003c) é uma plataforma de videoconferência acessível diretamente de qualquer navegador, eliminando a necessidade de instalações adicionais. Ele facilita

a realização de reuniões online, permitindo recursos como compartilhamento de tela e salas de conferência protegidas por senha. A interface do Jitsi Meet é projetada para ser intuitiva, permitindo aos usuários iniciar e configurar suas videoconferências com facilidade e rapidez.

- Jicofo(Jitsi, 2003b) atua como o cérebro por trás das operações, gerenciando a lógica de controle para as reuniões. Esse componente assegura a alocação eficiente dos recursos e a correta formação das chamadas entre os participantes, garantindo assim a fluidez das videoconferências.

- Jibri(Jitsi, 2003a), parte do ecossistema Jitsi, é uma solução técnica desenvolvida para gravar e transmitir videoconferências do Jitsi Meet. Ele opera automação de navegador via Selenium (Selenium Project, 2015) para ingressar em reuniões como um participante não interativo, capturando áudio e vídeo. Para a captura e codificação desses fluxos, o Jibri se vale do FFmpeg (ORG, 2020), uma ferramenta versátil de processamento de mídia. Além disso, utiliza dispositivos de loopback de áudio configurados via Alsa para a captura de áudio sem necessidade de hardware externo.

Implantação

Após a implantação de uma infraestrutura que permite Mobilidade do usuário e consistência no uso, o próximo passo na implantação do sistema é a hospedagem de ferramentas colaborativas. A implantação das ferramentas colaborativas foi realizada para criar um ambiente integrado e dinâmico, onde a colaboração entre os membros do laboratório acontece de forma fluida e prática. Essa estratégia atendeu a diversos requisitos, como a edição colaborativa de documentos, permitindo que vários usuários trabalhem simultaneamente, a colaboração tanto síncrona quanto assíncrona, por meio de videoconferências, quadros digitais e anotações rápidas.

Cada ferramenta escolhida cumpre um papel específico: o *Jitsi Meet* viabiliza reuniões virtuais com recursos como gravação e integração com o *Etherpad* e o *Excalidraw*; o *Etherpad* facilita a edição conjunta de textos com um histórico detalhado de mudanças; o *Overelaf* proporciona edição colaborativa de documentos *Latex*; o *Open Gist* permite a criação de notas rápidas; o *Filestash* permite edição colaborativa de documentos *Office*; o *Excalidraw* oferece um quadro branco digital para esboços e diagramas em tempo real. o *Nextcloud* oferece calendários compartilhados que podem se integrar facilmente ao Linux e Android, além de uma poderosa ferramenta de *Chat*. Além disso, o Gitea foi adotado para colaboração, versionamento de código e automação de processos.

Embora o gerenciamento de tarefas não tenha sido implementado nesta fase, os demais requisitos foram integralmente atendidos, proporcionando uma plataforma cola-

borativa robusta e alinhada com as necessidades de comunicação e organização do laboratório.

A tabela 11 mostra quais requisitos foram cumpridos na implantação.

Tabela 11 – Ferramentas Colaborativas e Acesso Remoto

Requisito	**Descrição**
Edição Colaborativa de Documentos	Sim
Colaboração Síncrona e Assíncrona	Sim
Videoconferências Integradas	Sim
Calendários e Agendamento Compartilhado	Sim
Anotações Rápidas	Sim
Gerenciamento de Tarefas	Não
Comunicação Direta	Sim

A seguir, detalha-se cada uma dessas categorias e as ferramentas associadas.

A colaboração entre os membros do laboratório pode ocorrer tanto presencialmente quanto remotamente. Para permitir a edição simultânea de documentos, quadros interativos e notas de reuniões, foram implantadas as seguintes ferramentas:

Jitsi Meet: Videoconferências e Integração com Outras Ferramentas

O Jitsi Meet [1] foi implantado como a solução principal para videoconferências no laboratório. Ele permite reuniões online entre os membros do ALICE, sendo especialmente útil para discussões de projetos, apresentações e eventos híbridos. Além da funcionalidade de videoconferência, o Jitsi possibilita:

- **Gravação e transmissão** de reuniões para referência futura.
- **Integração com o Etherpad**, permitindo edição colaborativa de textos durante as chamadas.
- **Integração com o Excalidraw**, oferecendo um quadro branco digital para anotações e ilustrações em tempo real.

Um dos principais diferenciais do Jitsi é a sua integração com outras ferramentas colaborativas, como:

- **Etherpad**: Permite a edição simultânea de textos durante as chamadas, garantindo que reuniões sejam documentadas de maneira eficaz.
- **Excalidraw**: Funciona como um quadro branco digital, possibilitando a ilustração de conceitos e a elaboração de diagramas em tempo real.

[1] Disponível em <https://meet.alice.ufsj.edu.br/>.

- **Alice Class**: Plataforma desenvolvida no próprio laboratório, voltada para gestão de atividades e transmissão de conhecimento.

A figura 8 mostra uma captura de tela da página inicial da instância de Jitsi Meet do laboratório.

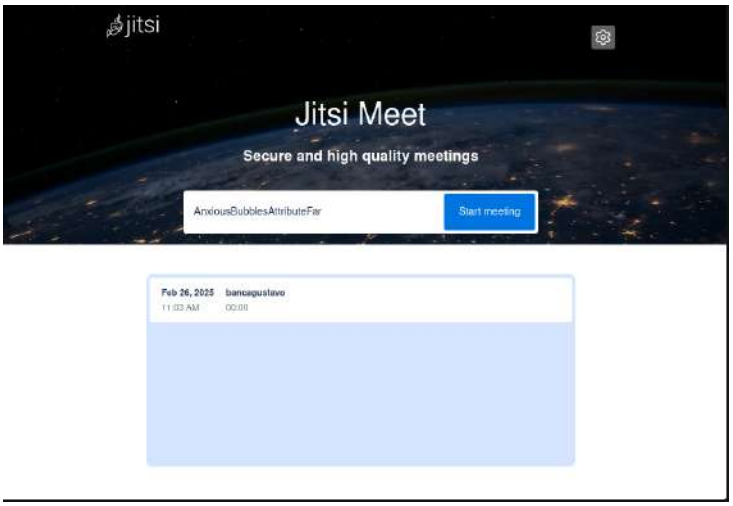

Figura 8 – Captura de tela da página inicial da aplicação Jitsi Meet

Etherpad: Edição Colaborativa de Textos

O Etherpad [2] foi escolhido como a principal ferramenta para edição coletiva de textos. Ele permite que múltiplos usuários editem um mesmo documento simultaneamente, sendo útil para a criação de anotações, atas de reuniões, rascunhos de projetos e outros tipos de registros escritos. O uso do Etherpad pode beneficiar o laboratório ao:

- Facilitar a criação de anotações compartilhadas durante reuniões e oficinas.
- Permitir a escrita colaborativa de documentos técnicos e tutoriais.
- Oferecer um histórico detalhado de edições para rastrear mudanças realizadas por diferentes usuários.

Além disso, sua integração com o Jitsi possibilitam a anotação de pontos importantes diretamente na plataforma. A figura 9 mostra uma captura de tela de um documento sendo editado no etherpad por duas pessoas.

[2] Disponível em <https://pad.alice.ufsj.edu.br/>.

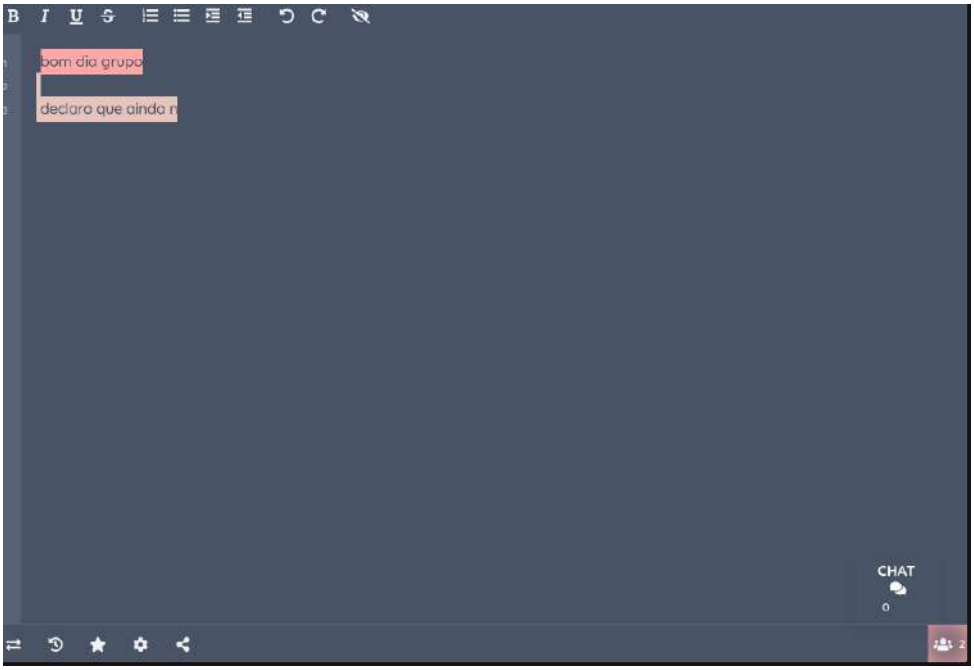

Figura 9 – Captura de tela do etherpad, no qual duas pessoas editam um documento

Excalidraw: Quadro Branco Digital

Para atividades que exigem representações visuais, diagramas e esboços, existe a possibilidade da utilização do Excalidraw, um quadro branco digital colaborativo. Esse recurso pode ser útil para:

- Elaborar diagramas técnicos e fluxogramas durante discussões.
- Criar representações gráficas de conceitos musicais e computacionais.
- Trabalhar em conjunto em rascunhos visuais durante reuniões e apresentações.

Para atividades que envolvem esquemas visuais, diagramas e representações gráficas, foi implantado o Excalidraw. Assim como o Etherpad, o Excalidraw pode ser integrado ao Jitsi, permitindo que os usuários acessem um quadro branco compartilhado dentro das videoconferências. Essa ferramenta permite a criação de ilustrações colaborativas, tornando-se um recurso valioso para oficinas, apresentações e planejamento de projetos. A figura 10 mostra a aplicação em execução dentro de uma chamada do Jitsi.

Overleaf: edição colaborativa de documentos Latex

Para a edição colaborativa de documentos Latex, hospedamos umas instância personalizada do Overleaf [3]. Essa instância usa um fork que implementa a integração com

[3] Disponível em <https://overleaf.alice.ufsj.edu.br/>.

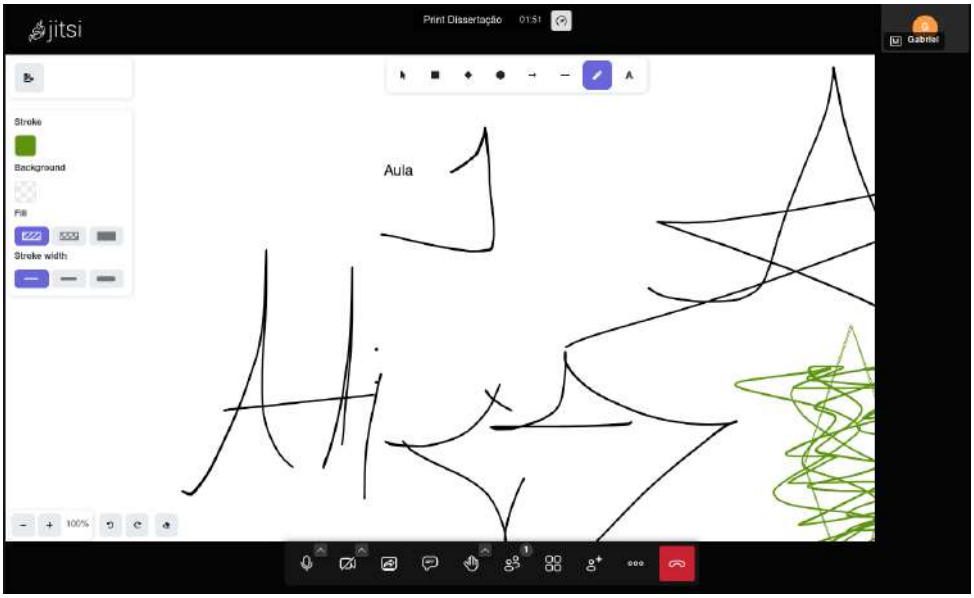

Figura 10 – Captura de tela do Jitsi meet com o quadro branco do Excalidraw aberto

um servidor LDAP, já que a versão gratuita do projeto não possui esta funcionalidade. A figura 11 mostra uma captura de tela de nossa instância do overleaf.

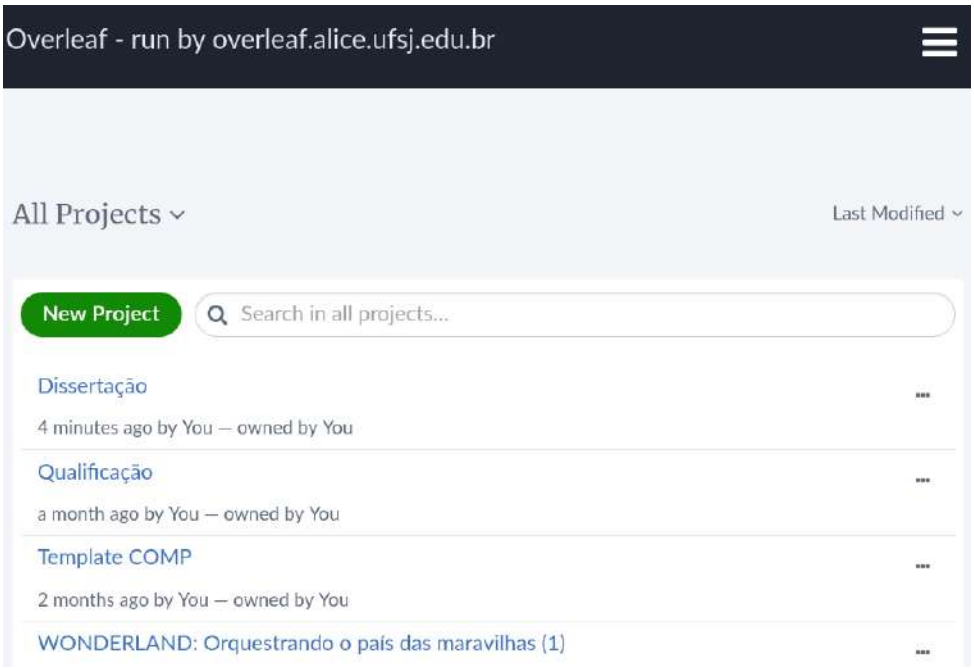

Figura 11 – Captura de tela do overleaf

Doku Wiki: documentação colaborativa

Para garantir que as documentações, tutoriais, e informações sobre projetos e atividades fosse armazenada, implantamos uma instância da DokuWiki [4]. Ela é uma pla-

[4] Disponível em <https://wiki.alice.ufsj.edu.br/>.

taforma leve e auto-hospedável para Wikis, que utilizamos para armazenar guias técnicos e tutoriais sobre as ferramentas utilizadas, documentação dos projetos, documentação das ferramentas do Wonderland, e tutoriais sobre outros projetos e ferramentas.

Ela serve como um repositório central de conhecimento, garantindo que informações importantes sejam documentadas e acessíveis para todos os usuários. A figura 12 mostra uma captura de tela da página inicial da Wiki.

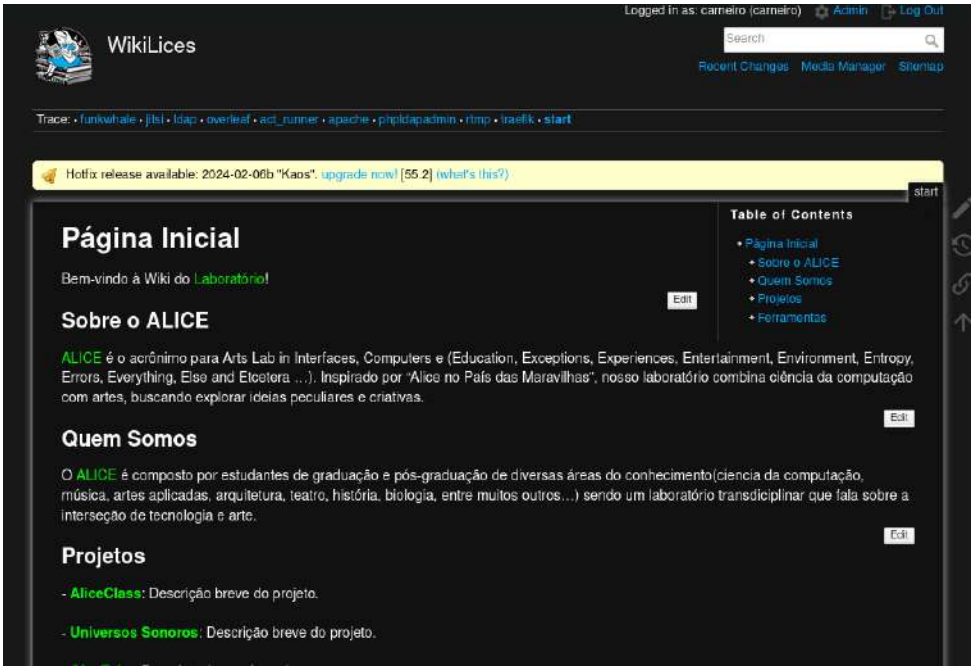

Figura 12 – Captura de tela da página inicial da Wiki

Gitea: Versionamento de Código e Automação

O versionamento de código e a automação de processos garantem a consistência dos projetos desenvolvidos no laboratório. Para esse fim, propõe-se a adoção do *Gitea* [5] como plataforma para hospedagem de repositórios Git.

O Gitea foi escolhido como a plataforma principal para controle de versão de código no laboratório. Ele permite que os projetos de software sejam organizados em repositórios Git, garantindo que todas as modificações sejam registradas e documentadas de forma estruturada.

Além do versionamento, o Gitea também foi configurado para automatizar diversos processos dentro do laboratório, incluindo:

- **Publicação de pacotes no PPA**: O fluxo de trabalho [6] foi configurado para

[5] Disponível em <https://git.alice.ufsj.edu.br/>.
[6] Disponível em <https://git.alice.ufsj.edu.br/alice-meta-packages/deb-deploy-action>.

que pacotes de software desenvolvidos no laboratório possam ser automaticamente compilados e publicados.

- **Deploy de sites e serviços**: Alterações em repositórios específicos disparam ações automatizadas para atualização de páginas web e sistemas internos.

Dessa forma, o Gitea não apenas facilita a colaboração entre desenvolvedores, mas também contribui para a manutenção da infraestrutura do laboratório. A figura 13 mostra a página de exploração de repositório de nossa instância do Gitea.

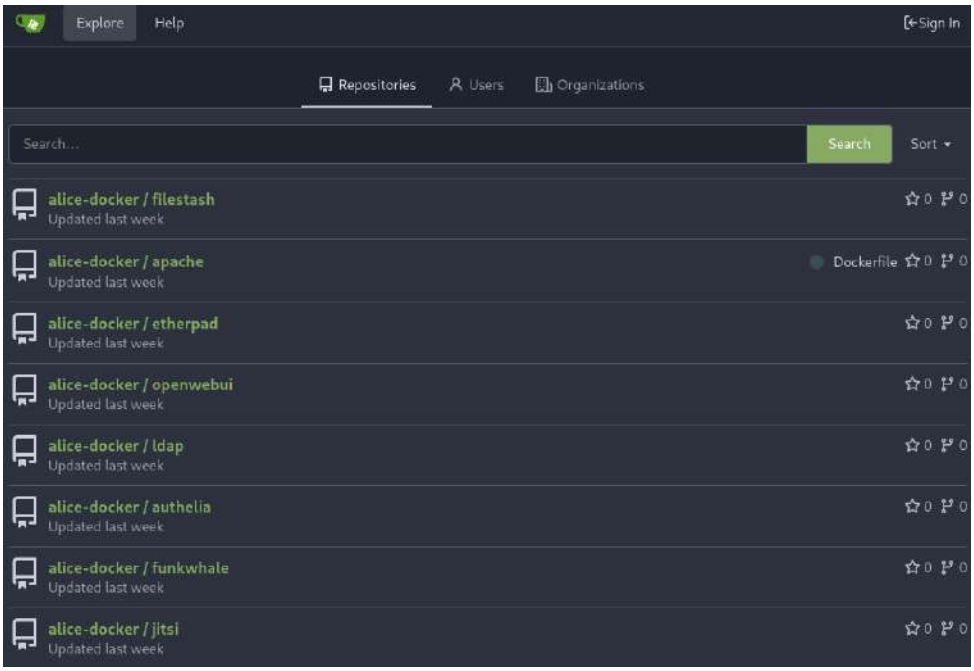

Figura 13 – Captura de tela de explorar repositórios na instância de Gieta do laboratório

6 Gestão e Persistência dos Artefatos

Implantação

Com a infraestrutura do laboratório estabelecida, garantindo a mobilidade dos usuários e a disponibilização de ferramentas colaborativas, surge a necessidade de um sistema eficiente para armazenar, organizar e preservar os artefatos produzidos. Sem um mecanismo adequado de gestão desses arquivos, o ambiente correria o risco de fragmentação da informação, dificultando o acesso e a continuidade dos projetos desenvolvidos no laboratório.

A diversidade de artefatos gerados – como documentos, códigos-fonte, produções musicais, gravações de reuniões e outros tipos de material multimídia – exige uma solução flexível e integrada. Dessa forma, foram implantadas diferentes soluções para contextos distintos. Essa abordagem atende ao requisito de **Persistência dos Artefatos**, assegurando que o conhecimento produzido seja mantido ao longo do tempo, sem risco de perda ou dispersão.

Para o armazenamento e versionamento de código-fonte, foi implantado o **Gitea**, um serviço de repositório Git auto-hospedado. Além de possibilitar o rastreamento das alterações em projetos de desenvolvimento, o Gitea também facilita a colaboração entre os membros do laboratório ao fornecer um ambiente estruturado para controle de versões e revisão de código. Essa escolha atende ao requisito de **Controle de Versões**, garantindo que as modificações feitas nos artefatos possam ser gerenciadas de forma eficiente e segura.

No contexto de arquivos multimídia, diferentes ferramentas foram adotadas para lidar com formatos específicos. Para armazenamento e compartilhamento de arquivos de áudio e músicas, utilizou-se o **Funkwhale**, uma plataforma descentralizada voltada para a organização e distribuição de conteúdo sonoro. Com isso, os usuários podem armazenar e acessar suas produções musicais diretamente na plataforma, sem depender de soluções externas. Já para documentos e registros de conhecimento, foi utilizada uma **Wiki**, servindo como repositório de documentação, tutoriais e registros das atividades realizadas no laboratório. Dessa forma, a Wiki atua como um ponto de referência para consultas e contribuições contínuas, garantindo a persistência do conhecimento gerado ao longo do tempo.

Além dessas soluções, o **Nextcloud** foi utilizado para o armazenamento de arquivos genéricos, oferecendo uma interface acessível para sincronização e compartilhamento de documentos entre os usuários. Essa ferramenta possibilita a organização de arquivos diversos, permitindo que os membros do laboratório armazenem seus materiais de forma

segura e acessível. Por fim, para textos acadêmicos e produção colaborativa de artigos, o **Overleaf** foi utilizado como editor LaTeX baseado na web, facilitando a escrita científica e garantindo um fluxo de trabalho contínuo entre diferentes autores.

Além das soluções especializadas para diferentes tipos de artefatos, os usuários do laboratório também possuem um espaço pessoal para armazenamento de arquivos acessíveis remotamente. Cada integrante pode utilizar a pasta `public_html` em seu diretório pessoal, disponibilizando arquivos através do servidor web Apache. Essa abordagem permite que os usuários compartilhem seus próprios artefatos diretamente, sem a necessidade de plataformas adicionais. O acesso a esses arquivos pode ser feito tanto via navegador quanto por meio de ferramentas como o **Filestash**, que fornece uma interface web para explorar e gerenciar os diretórios pessoais. Além disso, outros usuários podem acessar os arquivos compartilhados de maneira direta, seja via HTTP ou utilizando protocolos como SFTP, garantindo flexibilidade no compartilhamento de conteúdos dentro do laboratório. Essa solução complementa as demais ferramentas adotadas, permitindo um fluxo livre de troca de arquivos entre os integrantes, sem restrições rígidas de formato ou categoria.

Para aprimorar o uso do `public_html` como meio de compartilhamento de artefatos, foi desenvolvida uma solução própria chamada **Alice Index**. Essa ferramenta realiza varreduras periódicas nos diretórios `public_html` dos usuários, indexando automaticamente os arquivos encontrados com base em seu tipo. Com isso, é possível organizar e facilitar o acesso aos conteúdos compartilhados através de uma página web, permitindo que usuários encontrem rapidamente arquivos de interesse sem precisar navegar manualmente pelos diretórios. Essa abordagem aproveita a infraestrutura existente, integrando o espaço pessoal de cada usuário em um sistema unificado de indexação, sem a necessidade de configurações adicionais ou plataformas externas.

Outro aspecto importante foi a implementação de um **sistema de backup automatizado** de todo o sistema, que embora não seja específico dos artefatos, garante que os arquivos armazenados estejam protegidos contra falhas, exclusões acidentais ou problemas de integridade. Com esse mecanismo, o ambiente assegura que os artefatos possam ser recuperados em caso de incidentes, atendendo ao requisito de **Backup Automático**.

Além disso, foram estabelecidas políticas de **gestão de permissões**, garantindo que apenas usuários autorizados pudessem modificar ou acessar determinados arquivos, conforme suas responsabilidades no laboratório e a ferramenta utilizada (por exemplo nem todo usuário tem permissão de utilizar o Git, e mesmo os que tem, não podem alterar qualquer repositório). Essa funcionalidade atende ao requisito de **Gestão de Permissões**, assegurando que os dados sejam protegidos contra acessos indevidos.

Apesar dessas implementações, algumas funcionalidades previstas não foram exploradas nesta fase do projeto. O requisito de **Auditoria e Monitoramento**, que permitiria uma verificação contínua da integridade dos arquivos e alertas automáticos em caso de

inconsistências, não foi implementado devido à ausência de uma demanda crítica para esse nível de controle. Além disso, embora o ambiente permita a exportação e migração dos artefatos para outras plataformas, a criação de uma documentação formalizada para esse processo ainda não foi realizada, deixando o requisito de **Documentação** apenas parcialmente atendido.

Dessa forma, a solução implantada garante um modelo robusto de armazenamento e persistência dos artefatos, permitindo que os usuários mantenham seus arquivos organizados e protegidos ao longo do tempo. A seguir, são apresentados os detalhes técnicos da implementação e das ferramentas utilizadas. A tabela 12 mostra quais requisitos foram cumpridos na implantação.

Tabela 12 – Gestão e Persistência dos Artefatos

Requisito	**Descrição**
Organização Automática	Sim
Backup Automático	Pacial
Controle de Versões	Sim
Auditoria e Monitoramento	Parcial
Gestão de Permissões	Sim
Documentação	Parcial

Nextcloud e Filestash: Compartilhamento de Arquivos

Para armazenamento e compartilhamento de documentos, foram adotadas duas ferramentas principais: Nextcloud e Filestash.

O **Nextcloud** [1] foi implantado como a solução principal para armazenamento de arquivos, permitindo que documentos sejam organizados em pastas compartilhadas e acessíveis de qualquer dispositivo. Além do compartilhamento de arquivos, o Nextcloud oferece:

- **Edição colaborativa de documentos**: Utilizando o OnlyOffice, é possível editar arquivos de texto, planilhas e apresentações diretamente na plataforma.
- **Controle granular de permissões**: Usuários podem definir quem pode visualizar e editar cada arquivo.
- **Sincronização automática**: Arquivos podem ser sincronizados entre diferentes dispositivos, garantindo acesso contínuo aos materiais.

Em nossa instâncias customizamos o tema padrão da aplicação, usando a mesma cor que alguns websites do laboratório, como é possível ver na figura 14.

[1] Disponível em <https://drive.alice.ufsj.edu.br/>.

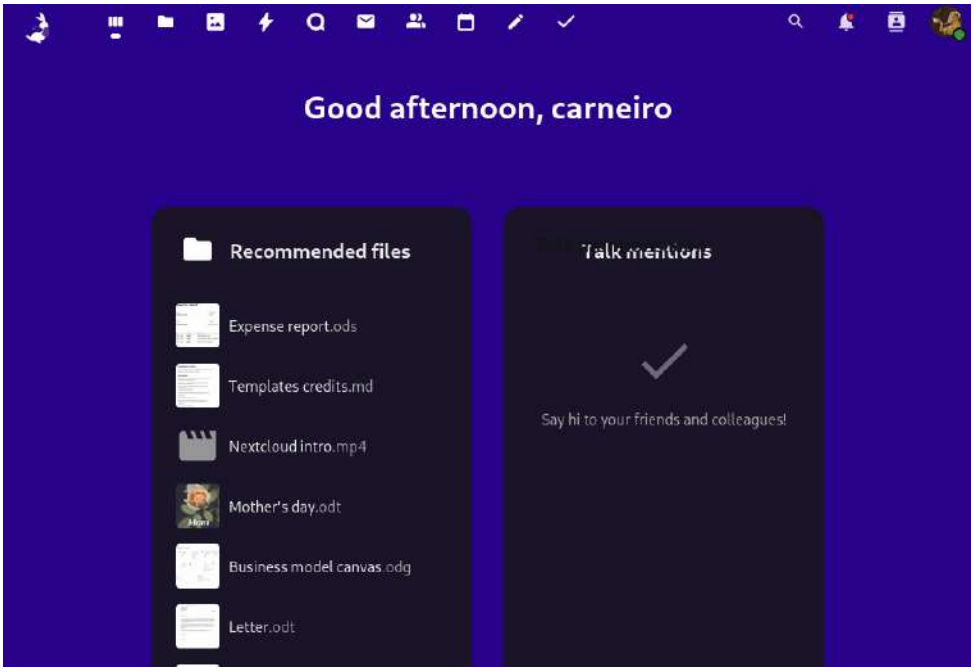

Figura 14 – Captura de tela do Nextcloud com um tema customizado

O **Filestash**, por sua vez, foi implantado como uma interface web para o acesso remoto a arquivos via SFTP. Ele permite que os usuários naveguem pelos seus diretórios diretamente pelo navegador, facilitando a gestão de arquivos sem necessidade de um cliente dedicado. Ele também permite que usuário compartilhem arquivos com usuários não conectados na plataforma.

Funkwhale: Organização de Produções Musicais

Para armazenar e compartilhar músicas produzidas no laboratório, foi implantado o Funkwhale [2], uma plataforma de streaming de áudio que permite que os usuários façam *upload* de faixas e organizem suas produções de forma acessível. A figura 15 mostra uma captura de tela da aplicação, mostrando um usuário conectado e algumas músicas publicadas.

Entre suas funcionalidades, destacam-se:

- Organização de faixas por álbuns e artistas.
- Reprodução em *streaming* diretamente da plataforma.
- Controle de acesso baseado em grupos LDAP.

[2] Disponível em <https://audio.alice.ufsj.edu.br/>.

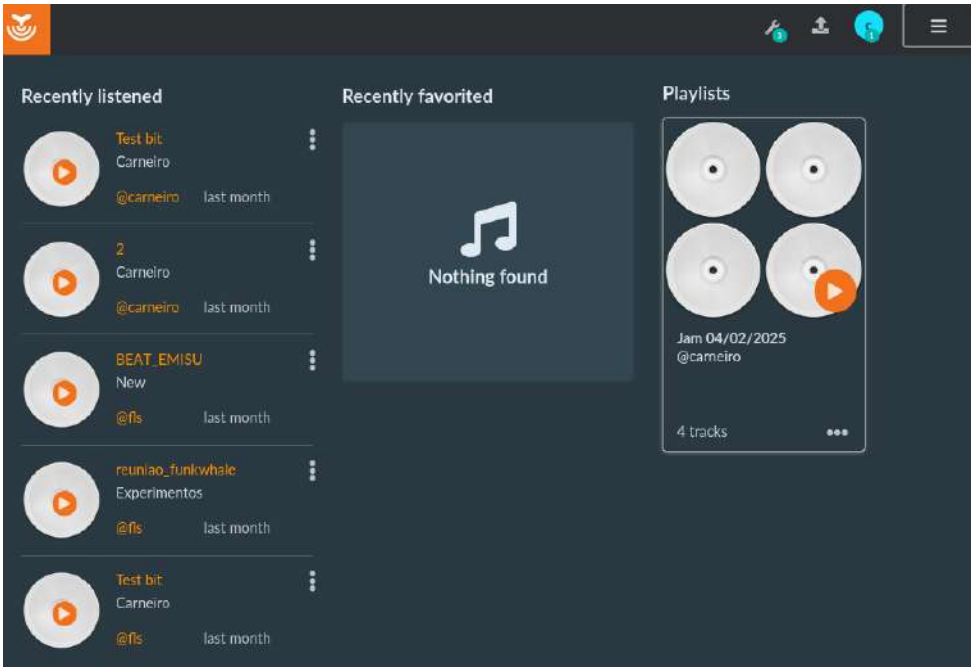

Figura 15 – Captura de tela do Funkwhale, mostrando um usuário conectado e algumas músicas publicadas

Gitea: Versionamento de Código e Automação

Além de mediar a colaboração na produção de código, o Gitea desempenha um papel fundamental ao funcionar também como um gerenciador de artefatos. Isso porque ele armazena e versiona código, garantindo um controle eficiente das alterações realizadas. Além disso, o processo de automação de *deploy* de pacotes do PPA pode ser lido como uma forma de geração de novos artefatos que são então armazenados no PPA. Dessa forma, sua utilização se torna essencial para manter a integridade e a rastreabilidade dos desenvolvimentos no ambiente do laboratório.

Alice Index: Indexação Automatizada de Artefatos

Com o crescimento da quantidade de artefatos compartilhados pelos usuários do laboratório, tornou-se necessário um mecanismo para facilitar a descoberta e organização desses arquivos. Embora a estrutura do `public_html` ofereça um meio simples para publicação de conteúdos acessíveis via web, sua navegação pode ser limitada, especialmente à medida que os diretórios crescem e contêm uma diversidade de tipos de arquivos. Para resolver esse problema, foi desenvolvida a ferramenta **Alice Index** [3], um sistema que automatiza a indexação dos arquivos compartilhados nos diretórios `public_html` dos usuários.

O funcionamento do Alice Index baseia-se em varreduras periódicas, nas quais

[3] Disponível em <https://alice.ufsj.edu.br/s/alice-index/>.

o sistema percorre os diretórios `public_html` de cada usuário e identifica os arquivos armazenados. A indexação não se limita apenas à listagem dos arquivos; ela também classifica os conteúdos com base em seu tipo, permitindo que os usuários filtrem e encontrem rapidamente documentos, códigos-fonte, imagens, arquivos multimídia e outros tipos de artefatos sem precisar navegar manualmente pelos diretórios. Além disso, o sistema não apenas identifica os arquivos, mas também oferece diferentes formas de visualização para determinados tipos específicos, ampliando a utilidade da ferramenta.

Atualmente, o Alice Index indexa três categorias principais de arquivos: **arquivos de áudio**, **arquivos do Pure Data** e **arquivos do LMMS**. Para arquivos de áudio, o sistema fornece uma interface simplificada que permite a pré-escuta diretamente pelo navegador, facilitando a navegação e organização das produções sonoras do laboratório. Já para arquivos do **Pure Data** e do **LMMS**, o Alice Index utiliza funcionalidades provenientes de outros projetos do próprio laboratório: **PdSearch** e **LmmsSearch**. Esses projetos, desenvolvidos originalmente como parte de outro trabalho de mestrado, foram parcialmente portados para o Alice Index, permitindo que informações detalhadas desses arquivos sejam extraídas e apresentadas de forma visualmente estruturada.

O **PdSearch** analisa os arquivos do Pure Data e extrai metadados, incluindo informações sobre objetos utilizados no patch, conexões internas e propriedades relevantes, oferecendo uma representação detalhada do fluxo de sinal do arquivo. Da mesma forma, o **LmmsSearch** processa arquivos do LMMS e fornece informações sobre as trilhas, plugins utilizados e estrutura do projeto, permitindo que os usuários tenham um panorama mais completo dos arquivos sem precisar abrir o software original. Essa integração entre os sistemas expande as possibilidades do Alice Index, permitindo não apenas a descoberta dos arquivos, mas também um entendimento mais aprofundado de seu conteúdo antes mesmo da abertura em um software dedicado.

A interface gerada pelo Alice Index apresenta uma listagem estruturada dos arquivos disponíveis, podendo ser acessada via navegador. Os usuários podem visualizar seus próprios conteúdos indexados, bem como os arquivos disponibilizados por outros integrantes, além da possibilidade de busca, garantindo um meio eficiente de compartilhamento de informações. Além disso, como o sistema funciona de maneira automatizada, ele mantém a indexação sempre atualizada sem a necessidade de intervenção manual, refletindo alterações nos diretórios `public_html` de forma dinâmica.

Por fim, o Alice Index se mostra uma solução eficiente para potencializar o uso do `public_html`, permitindo que o laboratório aproveite ao máximo a infraestrutura existente sem a necessidade de plataformas externas complexas. Com sua implementação, os usuários ganham um mecanismo simples, mas poderoso, para organizar e acessar os artefatos produzidos, tornando o fluxo de trabalho mais ágil e colaborativo.

Alice Class: Plataforma para Compartilhamento de Aulas e Oficinas

O **Alice Class** [4] (COSTA; SCHIAVONI, 2024) é uma plataforma desenvolvida para o compartilhamento de aulas, oficinas e cursos de maneira aberta e acessível. Criado por Costa e Schiavoni (2024), membros do laboratório, e com colaborações periódicas para a integração com o *Wonderland* por parte do autor desta dissertação, o projeto foi concebido para permitir a disponibilização de conteúdos educativos de forma simples e eficiente, utilizando diversas ferramentas já presentes na infraestrutura do laboratório para garantir sua funcionalidade. Dessa forma, o Alice Class integra autenticação centralizada, controle de versionamento e suporte a transmissões ao vivo, proporcionando um ambiente unificado para a disseminação de conhecimento.

A plataforma adota um modelo **estático**, utilizando o **Jekyll** para a geração das páginas. Isso garante um desempenho otimizado e uma arquitetura de fácil manutenção, eliminando a necessidade de servidores dinâmicos para exibição do conteúdo. Para gerenciar as versões do site e adicionar novos conteúdos, o **Gitea** foi incorporado ao fluxo de desenvolvimento da plataforma. Sempre que uma nova aula ou oficina é adicionada, o Alice Class realiza um **commit** no repositório correspondente, acionando automaticamente o processo de **build** e atualização do site estático. Essa abordagem baseada em versionamento garante que todas as modificações sejam registradas e possam ser revertidas, se necessário, proporcionando um histórico completo das publicações.

Para manter a integridade e organização dos conteúdos, o Alice Class implementa um sistema de **autenticação de usuários** baseado no diretório centralizado **LDAP**. Embora a plataforma seja de acesso livre para visualização dos materiais, apenas usuários autenticados podem realizar **upload** de novos conteúdos. Esse mecanismo impede modificações não autorizadas e garante que apenas membros do laboratório possam contribuir com novas aulas e oficinas.

O fluxo de upload de novos vídeos no Alice Class é estruturado da seguinte forma: após autenticar-se no sistema, o usuário realiza o envio do arquivo, que é armazenado localmente no backend da plataforma. Uma vez concluído o upload, o sistema atualiza automaticamente o banco de dados local e sincroniza as modificações com o repositório do **Gitea**. Esse processo dispara a reconstrução do site, garantindo que o novo conteúdo seja incorporado à plataforma sem necessidade de intervenção manual. Essa integração direta com o Gitea permite um controle eficiente das versões e mantém o ambiente sempre atualizado com as últimas publicações.

Além do compartilhamento de vídeos sob demanda, o Alice Class também suporta **transmissões ao vivo** por meio da integração com um **servidor RTMP**. Esse recurso permite que o Alice Class atue como um **player web** para transmissões em tempo real,

[4] Disponível em <https://class.alice.ufsj.edu.br/>.

proporcionando uma interface acessível para que usuários acompanhem eventos ao vivo diretamente pelo navegador. A integração com o RTMP torna possível conectar o sistema a ferramentas populares de transmissão, como o **OBS Studio** e o **Jitsi Meet**, viabilizando a exibição de aulas em tempo real com alta qualidade e sem a necessidade de plataformas externas de streaming.

Com essas funcionalidades, o Alice Class se consolida como uma solução prática e bem integrada ao ecossistema do laboratório, garantindo um fluxo contínuo e organizado para a publicação e transmissão de conteúdos educacionais. Ao utilizar tecnologias existentes dentro da infraestrutura do laboratório, o sistema minimiza dependências externas e maximiza a flexibilidade para personalizações futuras, assegurando um ambiente de ensino acessível e eficiente.

DokuWiki: Registro e Documentação

Nossa wiki vai muito além de ser apenas um repositório de informações. Ela funciona também como um gerenciador de artefatos de documentação, organizando todos os registros produzidos no laboratório. Além disso, ela pode agir como um indexador de artefatos, o que facilita a localização e o acesso aos conteúdos. Dessa forma, os usuários conseguem encontrar rapidamente o que precisam, tornando o acesso ao conhecimento mais simples e eficiente.

7 Gerenciamento de Identidade e Acesso

Conceitos relacionados

Segundo Bertino e Takahashi (2010), o Gerenciamento de Identidade e Acesso controla quem pode acessar sistemas e informações, garantindo que usuários sejam corretamente identificados e tenham permissões adequadas dentro de um ambiente digital. Esse gerenciamento envolve três aspectos principais: autenticação, autorização e provisionamento de usuários.

A autenticação verifica se um usuário é realmente quem diz ser ao tentar acessar um sistema. Esse processo pode envolver senhas, autenticação multi-fator ou outros métodos. Já a autorização define o que um usuário autenticado pode acessar e quais ações ele pode realizar. Além disso, há o provisionamento de usuários (ou gerenciamento de identidades), que lida com a criação, atualização e remoção de contas, garantindo que cada usuário tenha acesso apenas enquanto for necessário.

Autenticação

Segundo Lopez, Oppliger e Pernul (2004), a autenticação é definida como o processo de verificar a identidade de um objeto ou sujeito. É a prova de que uma entidade é quem ela afirma ser. Por exemplo, em uma transação de comércio eletrônico, a autenticação garante que a identidade do cliente é validada antes de prosseguir.

Tipos de Autenticação:

- Autenticação Baseada em Senha – O método mais comum, mas vulnerável a ataques como phishing e força bruta.
- Autenticação Multifator (MFA) – Exige mais de um fator para autenticação, como senha + biometria ou senha + token.
- Autenticação Biométrica – Usa características físicas como impressão digital ou reconhecimento facial.
- Autenticação Baseada em Certificados – Usa certificados digitais para validar a identidade do usuário.

Em ambientes onde diversos sistemas exigem autenticação, um problema comum é a necessidade de gerenciar múltiplas credenciais. Cada serviço pode manter seu próprio

banco de usuários, obrigando administradores a criar, atualizar e remover contas separadamente. Para os usuários, isso significa ter que lembrar diferentes combinações de login e senha. Além disso, em casos onde uma conta precisa ser desativada, a falta de um controle unificado pode resultar em acessos indevidos a sistemas que não foram devidamente atualizados.

Por conta dos problemas provenientes do gerenciamento de múltiplas identidades, existe o conceito de Autenticação Centralizada, no qual todas as credenciais dos usuários são armazenadas e gerenciadas por um único serviço, como um servidor LDAP ou Active Directory. Em vez de cada sistema manter um banco de usuários próprio, eles consultam essa base centralizada sempre que precisam autenticar alguém. Esse modelo permite que administradores tenham um ponto único para gerenciar contas e permissões, tornando o controle de acessos mais eficiente e simplificando a remoção de usuários quando necessário.

Outra solução complementar é o Single Sign-On (SSO), que vai além da centralização das credenciais ao permitir que um usuário, após um único login, tenha acesso a vários serviços sem precisar inserir suas credenciais repetidamente. Enquanto a autenticação centralizada garante que todos os sistemas utilizem as mesmas credenciais, o SSO elimina a necessidade de novos logins dentro da mesma sessão, proporcionando uma experiência mais fluida. Para implementar SSO, diferentes protocolos podem ser utilizados, dependendo do ambiente e dos serviços envolvidos. Os principais protocolos incluem:

- Kerberos – Protocolo baseado em *tickets*, amplamente utilizado em redes corporativas e sistemas operacionais para autenticação única de usuários em máquinas e serviços.
- SAML (Security Assertion Markup Language) – Protocolo baseado em XML, usado para autenticação federada entre organizações e provedores de serviços.
- OAuth2 – Protocolo de autorização que permite que usuários concedam acesso a serviços sem compartilhar suas credenciais diretamente.
- OpenID Connect (OIDC) – Protocolo de autenticação baseado no OAuth2, que permite que serviços confiem na identidade de um usuário autenticado por um provedor externo.

Cada um desses protocolos tem seu uso específico. O Kerberos é mais adequado para SSO em redes internas, enquanto OAuth2 e OpenID Connect são mais utilizados para SSO em aplicações web e serviços na nuvem.

Embora autenticação centralizada e Single Sign-On (SSO) sejam conceitos distintos, eles se interligam em muitos aspectos, pois ambos buscam reduzir a complexidade no gerenciamento de credenciais e melhorar a experiência do usuário. A autenticação

centralizada garante que todos os sistemas utilizem um único repositório de identidades, enquanto o SSO permite que, após um único login, o usuário acesse diversos serviços sem precisar autenticar-se novamente. Por exemplo, um servidor LDAP pode ser utilizado como a base central de autenticação, enquanto ferramentas como Authelia ou Kerberos implementam o SSO sobre essa estrutura, permitindo logins unificados sem exigir novas credenciais para cada serviço.

Autorização e controle de acesso

A autorização é o processo de conceder permissões com base na identidade autenticada. Trata-se de determinar se uma entidade autenticada tem permissão para acessar ou executar certas ações em um recurso. No exemplo do comércio eletrônico, após a autenticação do cliente, a autorização decide se o cliente tem permissão para realizar uma compra ou acessar informações específicas. Dentro os modelos de controle de acesso, iremos discutir dois: o Role-Based Access Control (RBAC), e o Attribute-Based Access Control (ABAC).

Segundo Ferrari (1992), o Role-Based Access Control (RBAC) é um modelo de controle de acesso que define permissões com base em funções atribuídas aos usuários. Em vez de conceder permissões diretamente a cada usuário, o RBAC agrupa permissões em funções (roles), e os usuários recebem essas funções de acordo com suas responsabilidades dentro da organização. Isso simplifica o gerenciamento de acessos, garantindo que usuários tenham apenas as permissões necessárias para suas atividades, reduzindo riscos de segurança e erros administrativos. Esse modelo é amplamente utilizado em sistemas corporativos, sendo frequentemente implementado em diretórios LDAP, onde grupos representam funções específicas e os usuários são atribuídos a esses grupos para herdar automaticamente as permissões associadas.

Segundo Hu et al. (2015), Controle de Acesso Baseado em Atributos (ABAC) é um modelo flexível de controle de acesso no qual as permissões são concedidas com base em atributos associados aos usuários, recursos e contexto da requisição. Diferente do RBAC (Role-Based Access Control), que restringe o acesso com base em funções predefinidas, o ABAC utiliza uma abordagem mais dinâmica, considerando múltiplas variáveis para determinar se um usuário pode ou não acessar um determinado recurso. Esses atributos podem incluir informações como cargo, departamento, localização, dispositivo utilizado e até mesmo horário da solicitação. Essa granularidade adicional permite um controle mais refinado, sendo amplamente adotado em ambientes que exigem políticas de segurança mais detalhadas e adaptáveis. Além disso, o ABAC facilita a implementação de regras complexas sem a necessidade de criar e gerenciar um grande número de funções, como ocorre no modelo RBAC.

Provisionamento de usuários

Segundo phoenixNAP (2024), o **provisionamento de usuários** é o processo de criação, gerenciamento e manutenção de contas de usuários e direitos de acesso nos sistemas de TI de uma organização. Envolve tarefas como a criação, modificação e exclusão de contas, além do gerenciamento de permissões e privilégios de usuários com base em suas funções e responsabilidades. Este processo visa alinhar o acesso dos usuários aos requisitos organizacionais, garantindo que funcionários, prestadores de serviços, parceiros e outras partes interessadas tenham o nível apropriado de acesso para desempenhar suas funções de maneira eficaz, ao mesmo tempo que protege dados e recursos confidenciais contra acesso não autorizado ou uso indevido.

Existem diversos métodos de provisionamento de usuários, cada um com características e casos de uso específicos:

- **Provisionamento Manual:** Neste método, os administradores criam, modificam ou excluem manualmente contas de usuário e direitos de acesso utilizando ferramentas ou interfaces nativas fornecidas por aplicativos ou sistemas individuais. Embora forneça controle refinado, pode ser demorado, propenso a erros e carece de escalabilidade.

- **Provisionamento Automatizado:** Utiliza ferramentas de software ou sistemas de gerenciamento de identidade para agilizar e automatizar os processos de criação, modificação e exclusão de contas de usuários. Essas ferramentas integram-se a vários sistemas e aplicativos de TI, permitindo que os administradores definam fluxos de trabalho de provisionamento, políticas de acesso e controles de acesso baseados em funções.

- **Provisionamento de Autoatendimento:** Permite que os usuários gerenciem seus próprios direitos de acesso e solicitações de provisionamento por meio de interfaces ou portais fáceis de usar. Os usuários podem solicitar novas contas, modificar privilégios de acesso existentes ou redefinir senhas sem intervenção direta dos administradores de TI.

- **Provisionamento Baseado em Função:** Gira em torno da atribuição de direitos de acesso e permissões com base em funções predefinidas ou funções de trabalho dentro de uma organização. Os usuários são atribuídos a funções específicas, e os privilégios de acesso são concedidos ou revogados automaticamente com base nas atribuições de funções.

- **Provisionamento Baseado em Atributos:** Considera atributos ou características adicionais do usuário, como departamento, local ou afiliação ao projeto, ao

conceder direitos de acesso, permitindo um controle mais granular sobre os processos de provisionamento de acesso.

LDAP

O LDAP (Lightweight Directory Access Protocol)(HOWES; SMITH, 1997) é um protocolo usado para acessar e gerenciar diretórios de informações em redes. Esses diretórios armazenam dados sobre usuários, grupos, dispositivos e outros recursos, organizados de forma hierárquica. Com LDAP, é possível centralizar e administrar essas informações, permitindo que várias aplicações e serviços acessem os dados de maneira consistente e segura. O LDAP é amplamente utilizado para autenticação e autorização (HOWES; SMITH; GOOD, 2003), facilitando a administração de credenciais e permissões de acesso. Dentre as principais implementações do protocolo LDAP, destacam-se:

- Microsoft Active Directory (MICROSOFT, 1999) – Solução proprietária da *Microsoft*, amplamente utilizada em redes corporativas.
- OpenLDAP (OPENLDAP, 1998) – Implementação de código aberto, frequentemente usada em ambientes Linux.
- Apache Directory Server (APACHE, 2006) – Outra alternativa de código aberto, projetada para alta escalabilidade e integração com aplicações empresariais.

O LDAP armazena informações em uma estrutura hierárquica, onde os dados são representados em um formato de árvore chamado DIT (Directory Information Tree). Cada entrada dentro do LDAP é identificada por um Distinguished Name (DN), que serve como um caminho único para acessar a informação dentro da árvore.

As principais unidades organizacionais do LDAP incluem:

- Usuários: Representam contas individuais dentro do diretório.
- Grupos: Conjuntos de usuários organizados para facilitar o controle de permissões.
- Computadores: Dispositivos autenticados dentro da rede.
- Serviços: Registros que armazenam informações sobre aplicações que utilizam o LDAP.
- Pontos de montagem de sistemas de arquivos remotos: Registros que armazenam informações sobre pontos de montagem remotos como NFS e Samba.
- Regras de *sudo* no *Linux*: Armazena regras de permissões de usuários de um sistema Linux, similar ao arquivo sudoers.

Essa organização permite a gestão centralizada de credenciais e a unificação do acesso a diversos sistemas, reduzindo a necessidade de múltiplos logins e aumentando a segurança administrativa. Estas unidades organizacionais são definidas através dos chamados *schemas*.

Um dos usos do LDAP é a autenticação centralizada. A autenticação baseada em LDAP permite a centralização no gerenciamento e autenticação de usuários, eliminando a necessidade de contas locais separadas para cada sistema ou serviço.

Além de autenticação, o serviço pode ser usado também para autorização. Através do uso de grupos e atributos de um usuário é possível definir dentro do próprio diretório e em aplicações que utilizem o protocolo, regras de controle de acesso baseadas em funções (Role-based Access Control - RBAC) e atributos (Attribute-based Access Control - ABAC). Isso possibilita um controle granular sobre quem pode acessar quais recursos, garantindo um ambiente mais seguro e organizado.

Para definir quais tipos de usuários e grupos podem editar ou visualizar recursos dentro do diretório, é possível se utilizar da lista de controle de acesso (ACL) do LDAP para definir quem pode acessar quais recursos dentro do diretório. Essas regras de acesso podem ser configuradas para restringir ou conceder permissões de leitura, escrita e modificação de entradas.

Outros exemplos de uso de grupos no LDAP incluem:

- Controle de acesso a máquinas e serviços – Usuários podem ser agrupados em categorias como "usuários padrão", "administradores"e "ex-alunos", cada um com permissões diferenciadas.
- Permissões específicas dentro de aplicações – Algumas ferramentas permitem autenticação via LDAP e utilizam grupos para definir privilégios administrativos.
- Definição de políticas de segurança – Grupos LDAP podem ser configurados para restringir o acesso a determinados arquivos ou sistemas dentro da rede.

Nem todos os serviços oferecem suporte direto à autenticação via LDAP. Para resolver essa limitação, podem ser utilizados *middlewares* de autenticação, que atuam como intermediários entre o LDAP e aplicações que precisam de autenticação centralizada.

Um exemplo de *middleware* é o Authelia, que funciona como uma camada de autenticação para aplicações web. Ele permite autenticação LDAP e pode ser integrado a um proxy reverso (como Traefik ou Nginx) para restringir o acesso a páginas específicas com base em permissões definidas no LDAP.

O uso de proxies reversos com autenticação LDAP possibilita:

- Bloqueio de acesso a serviços administrativos, garantindo que apenas usuários autenticados possam acessá-los.
- Autenticação unificada para múltiplas aplicações, reduzindo a necessidade de logins repetidos.
- Compatibilidade com aplicações que não suportam LDAP nativamente, permitindo a aplicação de regras de controle de acesso de forma transparente.

Além do Authelia, servidores web como Apache também podem ser configurados para autenticação via LDAP, possibilitando o controle de acesso baseado em diretivas definidas no `.htaccess`.

A administração de um diretório LDAP envolve o provisionamento de usuários (a criação, modificação e remoção de usuários, grupos) e demais objetos armazenados no servidor, como ACLs, schemas e configurações. Embora seja possível gerenciar o LDAP via linha de comando utilizando ferramentas como `ldapadd`, `ldapmodify` e `ldapsearch`, a complexidade dessa abordagem torna recomendável o uso de interfaces gráficas para facilitar a administração.

Dentre as ferramentas disponíveis para essa função, destacam-se:

- LDAP Account Manager (LAM) – Interface web popular para administração de LDAP, voltada para ambientes Unix.
- phpLDAPadmin (PLA) – Interface baseada em PHP que permite gerenciamento avançado de LDAP com suporte a templates personalizados.
- Apache Directory Studio – Ferramenta baseada em Java para administração de diretórios LDAP, mais integrada ao Apache Directory Server.

As três ferramentas oferecem funcionalidades básicas para a administração do LDAP, mas possuem diferenças importantes quanto ao suporte a schemas e à flexibilidade de personalização.

LDAP Account Manager (LAM) O LAM é uma solução web amplamente utilizada para gerenciar diretórios LDAP, fornecendo uma interface amigável para a criação e administração de usuários e grupos. Entretanto, sua versão gratuita apresenta a limitação de suportar apenas o schema RFC2307, sem permitir a personalização de schemas. Isso o torna inadequado para cenários que requerem estruturas mais flexíveis, como o uso de RFC2307bis e groupOfEntries.

phpLDAPadmin (PLA) O phpLDAPadmin oferece maior flexibilidade e permite a criação de templates personalizados, possibilitando a administração de objetos

conforme schemas específicos. Dessa forma, é possível definir como usuários, grupos e permissões são representados dentro do diretório LDAP, garantindo compatibilidade com diferentes configurações.

Apache Directory Studio O Apache Directory Studio é uma ferramenta mais avançada, baseada em Java, que fornece suporte completo para gerenciar diretórios LDAP. Ele permite edição direta dos objetos do LDAP, visualização hierárquica e suporte a diversas operações administrativas. No entanto, essa ferramenta é mais integrada ao Apache Directory Server, e pode exigir configurações adicionais para ser utilizada eficientemente com servidores como OpenLDAP. Embora seja uma opção poderosa, seu foco em uma solução específica limita sua aplicabilidade para ambientes que utilizam outras implementações do LDAP.

Principais diferenças entre versão gratuita do LAM, phpLDAPadmin e Apache Directory Studio

Característica	**LAM**	**PLA**	**ADS**
Templates	Não	Sim	Sim
RFC2307	Sim	Sim	Sim
RFC2307bis	Não	Sim	Sim
groupOfEntries	Não	Sim	Sim
Interface web	Sim	Sim	Não (aplicação desktop)
Foco principal	OpenLDAP	Agnóstico	Apache Directory Server

Tabela 13 – Comparação entre LDAP Account Manager (LAM), phpLDAPadmin (PLA) e Apache Directory Studio (ADS)

A escolha da ferramenta de administração do LDAP depende das necessidades do ambiente e da infraestrutura utilizada.

- Para um ambiente simples, baseado em RFC2307 e focado em Unix/Linux, o LDAP Account Manager pode ser suficiente.
- Para um ambiente que exige suporte a schemas personalizados (RFC2307bis, groupOfEntries), o phpLDAPadmin é mais adequado, pois permite a customização da interface.
- Para quem utiliza o Apache Directory Server, o Apache Directory Studio é uma escolha natural, pois oferece integração nativa com essa tecnologia.

Embora o LDAP seja frequentemente utilizado como base para soluções de SSO, ele não implementa essa funcionalidade diretamente. Para isso, é necessário utilizar protocolos adicionais, como:

- Kerberos – Protocolo de autenticação que pode ser integrado ao LDAP para oferecer suporte ao SSO.
- OAuth e OpenID Connect – Protocolos modernos utilizados para autenticação web unificada.
- SAML (Security Assertion Markup Language) – Frequentemente usado para SSO em ambientes corporativos.

A integração do LDAP com SSO pode simplificar ainda mais o acesso a aplicações dentro do laboratório, reduzindo a necessidade de múltiplos logins e melhorando a experiência dos usuários.

O uso do LDAP como sistema de autenticação centralizada apresenta diversas vantagens, tais como:

- Gestão centralizada de usuários – Todas as credenciais são administradas em um único local, facilitando a manutenção e controle.
- Consistência nos acessos – Os usuários podem utilizar as mesmas credenciais para acessar diferentes serviços.
- Maior segurança – O LDAP permite a aplicação de políticas de autenticação e controle de acesso mais rigorosas.

Entretanto, a adoção do LDAP também pode apresentar desafios, incluindo:

- Complexidade na configuração inicial – A estruturação do diretório e a definição das permissões exigem planejamento cuidadoso.
- Dependência do servidor LDAP – Em caso de falha do servidor, os usuários podem ficar impedidos de acessar os serviços.
- Compatibilidade com aplicações legadas – Algumas ferramentas podem não suportar autenticação LDAP, exigindo o uso de middlewares.

Mesmo com esses desafios, a autenticação baseada em LDAP continua sendo uma solução eficaz para ambientes que exigem gerenciamento centralizado de usuários e permissões.

Implantação

À medida que o ambiente do laboratório se torna mais estruturado, garantindo a disponibilidade de recursos e a colaboração eficiente entre os usuários, surge a necessidade de um controle mais rigoroso sobre a autenticação e a autorização de acessos. Com múltiplos serviços implantados e usuários acessando máquinas e aplicações de diferentes pontos, o gerenciamento de credenciais descentralizado rapidamente se tornaria um gargalo tanto para a administração quanto para a segurança do ambiente.

Dessa forma, a implantação de um sistema de **autenticação centralizada** surge como uma solução para garantir um fluxo de acesso seguro e eficiente. O objetivo principal desse requisito é permitir que todas as máquinas e serviços utilizem um único ponto de autenticação, eliminando a necessidade de múltiplos cadastros e reduzindo problemas como senhas duplicadas ou esquecidas. Além disso, a centralização viabiliza uma administração mais simplificada e segura, garantindo que a remoção de um usuário reflita automaticamente em todo o sistema.

Para atender a essa necessidade, foi implementado um **sistema unificado de autenticação** baseado em **LDAP (Lightweight Directory Access Protocol)**, garantindo que todas as máquinas e serviços utilizem um único ponto de validação de identidade, cumprindo assim o requisito de **Autenticação Centralizada**. A figura 16 mostra um diagrama representando o sistema de autenticação implantado.

Além da autenticação, foi necessário criar um **sistema flexível de gestão de perfis e permissões**, atendendo ao requisito de **Gestão de Perfis**. Com isso, foi possível definir grupos de usuários com diferentes níveis de acesso, garantindo que cada pessoa tivesse acesso apenas aos serviços e recursos compatíveis com seu papel no laboratório. Para reforçar esse controle, também foram estabelecidas **Permissões Granulares**, permitindo um refinamento detalhado sobre quais operações cada usuário poderia realizar dentro dos serviços. Sem essa camada adicional de controle, a autenticação unificada, por si só, não impediria acessos indevidos a serviços sensíveis.

Para facilitar o gerenciamento de identidades e a criação de usuários dentro do sistema LDAP, foi utilizado o **PHPLdapAdmin** [1], que permite a administração visual dos registros do diretório. Esse sistema conta com suporte a **templates de criação de objetos**, possibilitando a definição de diferentes perfis de usuários já configurados com parâmetros específicos, alinhando-se ao requisito de **Integração com o Provisionamento**. Dessa forma, seria possível criar automaticamente contas padronizadas para diferentes tipos de usuários, como professores, alunos ou administradores, com permissões predefinidas. No entanto, no contexto do laboratório, essa funcionalidade não se mostrou essencial, uma vez que a diferenciação entre usuários foi feita através de **grupos**, dado o

[1] Disponível em <https://ldap.alice.ufsj.edu.br/>.

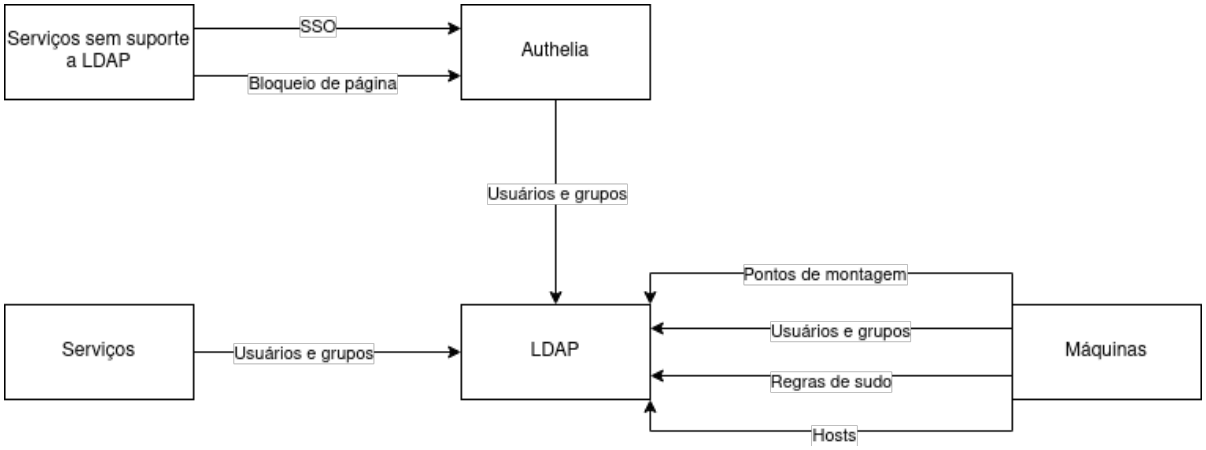

Figura 16 – Integração centralizada de autenticação, autorização e configuração via LDAP. A imagem mostra a arquitetura de autenticação e gerenciamento centralizado de usuários no laboratório, baseada em um servidor LDAP. No centro da estrutura, o servidor LDAP fornece dados de usuários, grupos, regras de sudo, hosts e pontos de montagem para os serviços e máquinas do ambiente. Serviços compatíveis utilizam diretamente o LDAP para autenticação e controle de acesso. Já serviços sem suporte nativo a LDAP são protegidos por um sistema de SSO intermediado pelo Authelia, que atua como camada de autenticação e autorização, fazendo a mediação com o LDAP. As máquinas clientes também consultam o LDAP para obter informações de login e montagem de diretórios, permitindo uma administração centralizada e coerente em todo o sistema

número reduzido de pessoas utilizando o ambiente. Além disso, como o cadastro de novos usuários não é uma tarefa recorrente, optou-se por manter o provisionamento manual, sem a necessidade de um mecanismo automatizado para essa função.

Outro ponto aspecto importante é a **auditoria de acessos**, que foi parcialmente implantada. Com um histórico detalhado de quem acessou quais serviços e quando, é possível rastrear atividades suspeitas, identificar possíveis falhas de segurança e aplicar políticas adaptativas caso tentativas de acesso indevido sejam detectadas. E todas as ferramentas usadas possuem certo nível de auditoria de acessos ao consultar os *logs*, mas como isso não foi realizado de forma centralizada, consideramos que implantação desse requisito foi parcial.

Por fim, o requisito de **Segurança Adaptativa**, que previa a implementação de mecanismos automáticos para detectar e reagir a acessos suspeitos, não foi abordado nesta fase do projeto. Essa funcionalidade demandaria um esforço adicional para integrar soluções mais avançadas de monitoramento e resposta a incidentes.

Para atender a essas necessidades, a solução adotada no laboratório envolveu a implementação do LDAP (Lightweight Directory Access Protocol) como diretório centralizado de identidades, complementado pelo Authelia para oferecer suporte à autenticação

multifator e integração com serviços que não possuem suporte nativo ao LDAP. Além disso, foram configuradas regras específicas para segmentação de usuários, definição de permissões com base em grupos e restrição de acesso a determinados serviços.

Dessa forma, a solução implantada atendeu a maioria dos requisitos levantados, como mostra a tabela 14, garantindo um controle eficiente de autenticação e autorização no laboratório. A seguir, serão detalhados os aspectos técnicos da implementação e as ferramentas utilizadas para viabilizar esse modelo.

Tabela 14 – Controle de Acesso Centralizado

Requisito	**Descrição**
Autenticação Centralizada	Sim
Gestão de Perfis	Sim
Auditoria de Acessos	Parcial
Segurança Adaptativa	Não
Autenticação Multifatorial	Parcial
Permissões Granulares	Sim
Integração com Provisionamento	Sim

A autenticação e o controle de acesso no ambiente do laboratório foram implementados utilizando o **LDAP** como diretório centralizado. A proposta inicial previa a utilização do LDAP para unificar o gerenciamento de credenciais, permissões e configuração de serviços. Durante a implementação, a estrutura planejada foi validada e aprimorada para garantir melhor integração com os serviços utilizados pelo laboratório.

A seguir, detalham-se os aspectos técnicos da autenticação centralizada, as soluções utilizadas para controle de permissões e as mudanças realizadas em relação à proposta original.

O LDAP foi configurado como repositório principal de usuários e grupos, permitindo que a autenticação nas máquinas e serviços do laboratório fosse realizada de maneira unificada. Além do armazenamento de credenciais, o diretório LDAP também foi utilizado para definir permissões de acesso e armazenar informações auxiliares sobre os usuários.

Dentre as principais configurações aplicadas ao servidor LDAP, destacam-se:

- Utilização do esquema **RFC2307bis** para permitir grupos aninhados e facilitar a associação entre usuários e permissões.
- Definição de regras de **ACL** (*Access Control List*) para controlar o acesso às informações armazenadas no diretório.
- Integração com o **SSSD** (*System Security Services Daemon*) para autenticação e autorização nas máquinas do laboratório.

- Armazenamento de informações adicionais, como regras de *sudo*, pontos de montagem NFS e mapeamento de hosts.

Essa estrutura garantiu maior organização na gestão de permissões e possibilitou um controle mais granular sobre os recursos disponíveis.

Execução do Servidor LDAP em Contêiner Docker

Para garantir maior flexibilidade, segurança e facilidade de manutenção, o servidor LDAP foi implantado dentro de um contêiner Docker. Essa abordagem permite o isolamento da aplicação em relação ao sistema operacional base, reduzindo o risco de conflitos com outros serviços e facilitando atualizações. Além disso, o uso de contêineres proporciona maior portabilidade, permitindo a replicação da configuração em diferentes servidores sem a necessidade de ajustes manuais.

A configuração do LDAP foi feita utilizando a imagem osixia/docker-openldap, que contém alguns scripts para definir alguams variavéis através de secrets do docker. A definição do contêiner foi estruturada através de um arquivo *docker-compose.yml*, armazenado e versionado no Gitea, o que permite rastrear modificações e recuperar configurações anteriores sempre que necessário. Esse arquivo especifica não apenas a imagem utilizada, mas também as configurações essenciais, como a persistência dos dados do diretório LDAP em volumes, a definição do domínio base, além das portas expostas para comunicação com outros serviços da infraestrutura do laboratório.

A implementação do LDAP dentro de um contêiner trouxe benefícios significativos para o ambiente do laboratório. Em primeiro lugar, simplificou a instalação e configuração do serviço, tornando possível a replicação do sistema em poucos comandos, caso seja necessário reinstalar ou migrar o diretório para outro servidor. A segurança do sistema também foi aprimorada por meio da utilização de redes internas do Docker, restringindo o acesso direto ao serviço LDAP e permitindo que apenas aplicações autorizadas, como o phpLDAPadmin e o middleware de autenticação Authelia, se comuniquem com o diretório. Além disto, utilizamos a função de proxy reverso TCP do Traefik para garantir SSL na comunicação, sem a necessidade de configurar diretamente na aplicação.

Definição de Grupos e Controle de Acesso

A organização das permissões foi realizada por meio da definição de grupos no LDAP, permitindo o controle de acessos baseado em funções e responsabilidades. A estrutura de grupos foi planejada para garantir que cada usuário tivesse os privilégios necessários para suas atividades, sem comprometer a segurança do ambiente.

Os principais grupos definidos e suas respectivas permissões foram:

- **ldap_admin**: Administração completa do LDAP, permitindo criar, modificar e excluir usuários e grupos.
- **git_admin**: Controle administrativo sobre repositórios e configurações do *Gitea*.
- **alicer**: Grupo padrão para usuários ativos no laboratório, garantindo acesso a serviços gerais.
- **lab_sudo**: Permissões administrativas nas máquinas do laboratório e no servidor.
- **ssh**: Acesso remoto ao servidor principal via SSH.
- **egresso**: Grupo para ex-membros do laboratório, garantindo acesso limitado a serviços específicos.

Além disso, a utilização do atributo `memberof` no esquema RFC2307bis permitiu que aplicações verificassem facilmente a quais grupos um usuário pertence, simplificando a autenticação em serviços externos.

Configuração de ACLs no LDAP

Para garantir a segurança das informações armazenadas no LDAP, foram implementadas regras de ACL que controlam quais usuários e serviços podem acessar determinados dados.

As principais regras definidas foram:

- **Acesso irrestrito ao root** via socket Unix para permitir operações administrativas locais.
- **Permissão total para o usuário admin**, garantindo controle total sobre o diretório LDAP.
- **Acesso completo ao grupo ldap_admin** para administração de usuários e permissões.
- **Autenticação anônima permitida** apenas para a tentativa inicial de login, garantindo que usuários possam iniciar a autenticação sem acesso prévio.
- **Acesso de leitura concedido a um usuário específico para consultas por aplicações** como SSSD, Authelia e serviços integrados ao LDAP.

Essa estrutura garantiu um modelo seguro e eficiente para o gerenciamento das credenciais do laboratório.

Administração do Diretório LDAP com phpLDAPadmin

Para gerenciar os usuários, grupos e demais objetos dentro do diretório LDAP, foi implantada a ferramenta **phpLDAPadmin**. Essa aplicação web fornece uma interface gráfica para administração do LDAP, facilitando operações comoc riação, modificação e remoção de usuários, grupos e outras entradas, e a execução de consultas e filtros para visualizar rapidamente informações armazenadas no diretório.

O phpLDAPadmin foi escolhido por ser uma solução leve, amplamente utilizada na administração de servidores LDAP baseados em OpenLDAP, e permite customização de *templates* [2] para acomodar diferentes tipos de *schemas*, como pode ser visto na figura 17. A interface simplificada permite que administradores realizem ajustes sem necessidade de manipulação direta dos arquivos de configuração via terminal.

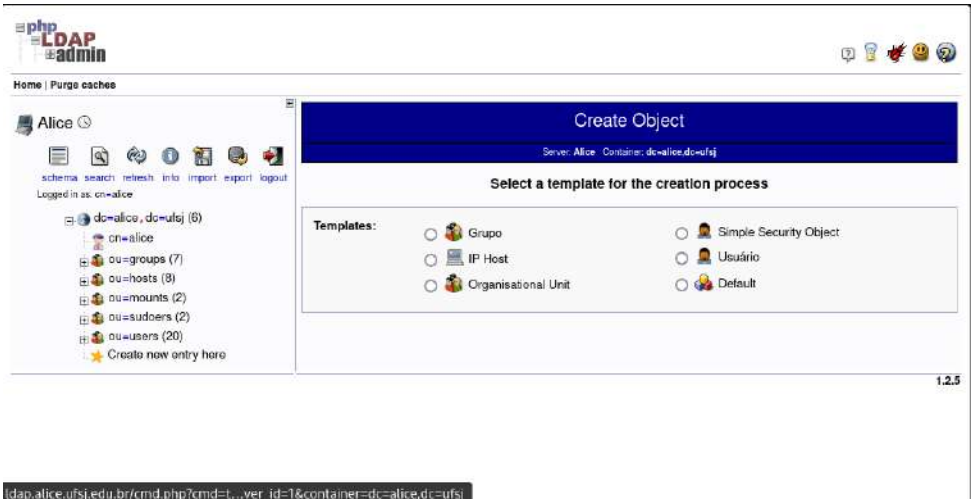

Figura 17 – Captura de tela do PHPLdapAdmin mostrando os templates de criação de objeto

Configuração e Integração com o Ambiente do Laboratório

O phpLDAPadmin foi configurado como um serviço hospedado dentro de um contêiner Docker, com o proxy reverso configurado através do *Traefik*, garantindo maior segurança e facilidade de manutenção. Além disso, foram criados *templates* personalizados do phpLDAPadmin para facilitar o gerenciamento de informações no diretório, visto que adotamos um modelo não usual. A adição dessa ferramenta complementa a infraestrutura de autenticação e autorização centralizada, garantindo que a administração do LDAP seja acessível e eficiente para os responsáveis pelo sistema.

[2] Disponível em <https://git.alice.ufsj.edu.br/alice-docker/phpldapadmin/src/branch/main/templates/creation>.

Integração com Serviços e Máquinas

Para garantir que o LDAP fosse utilizado de forma ampla dentro do laboratório, foi necessário integrá-lo a diversos serviços e máquinas. A autenticação dos usuários foi configurada para funcionar tanto nas estações de trabalho do laboratório quanto em aplicações hospedadas no servidor.

As integrações realizadas foram:

- **SSSD** para autenticação, montagem do NFS, regras de sudo e hosts nas máquinas do laboratório e o servidor, como pode ser visto na figura 18.
- **Gitea**, **Jitsi**, **Nextcloud** e outras ferramentas configuradas para autenticação via LDAP.
- **Authelia** como *middleware* de autenticação, protegendo serviços web que não possuem suporte nativo ao LDAP.

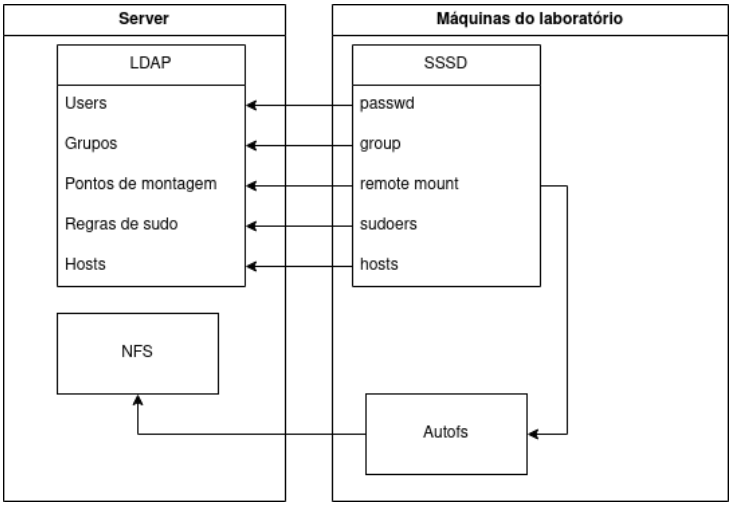

Figura 18 – Esta imagem representa a arquitetura de autenticação e gerenciamento de diretórios no laboratório. No lado esquerdo está o servidor, que oferece dois serviços principais: LDAP (para autenticação e informações dos usuários) e NFS (para compartilhamento de arquivos). No lado direito estão as máquinas do laboratório, que se conectam ao servidor. Elas usam o SSSD para buscar informações de autenticação no LDAP, permitindo que os usuários façam login com suas credenciais centralizadas. Além disso, utilizam Autofs para montar automaticamente os diretórios pessoais dos usuários, que estão armazenados no servidor via NFS. Dessa forma, toda vez que alguém acessa uma máquina do laboratório, suas credenciais e seus arquivos pessoais são puxados diretamente do servidor, garantindo um ambiente unificado e consistente em todas as máquinas.

Para garantir que os usuários possam acessar qualquer máquina do laboratório sem necessidade de criar contas locais separadas, todas as máquinas foram integradas ao

servidor LDAP. A autenticação dos usuários e a atribuição de permissões são gerenciadas centralmente, permitindo que cada membro do laboratório utilize suas credenciais únicas para acessar qualquer dispositivo. Essa integração foi realizada utilizando o SSSD (*System Security Services Daemon*), um serviço que permite que máquinas clientes autentiquem usuários diretamente em um servidor LDAP. Além da autenticação de usuários, o SSSD também foi configurado para gerenciar informações como:

- Grupos.
- Hosts, permitindo identificação dinâmica das máquinas.
- Pontos de montagem do NFS para a montagem automática de diretórios pessoais.
- Regras de *sudo* armazenadas no LDAP, permitindo gerenciamento centralizado de permissões administrativas.

Além disso, para evitar conflitos de nomenclatura entre as máquinas, foi implementado um *script* executado na inicialização do sistema que define automaticamente o nome da máquina com base em seu endereço IP, consultando o banco de dados do LDAP. Dessa forma, evita-se a necessidade de configurar manualmente os nomes de *host* de cada máquina.

Para garantir que aplicações sem suporte nativo ao LDAP pudessem utilizar a autenticação centralizada, foi adotado o **Authelia** [3], que atua como intermediário entre o LDAP e os serviços web, como pode ser visto na figura 19. O Authelia foi configurado para autenticar usuários via LDAP e fornecer autenticação unificada para aplicações protegidas.

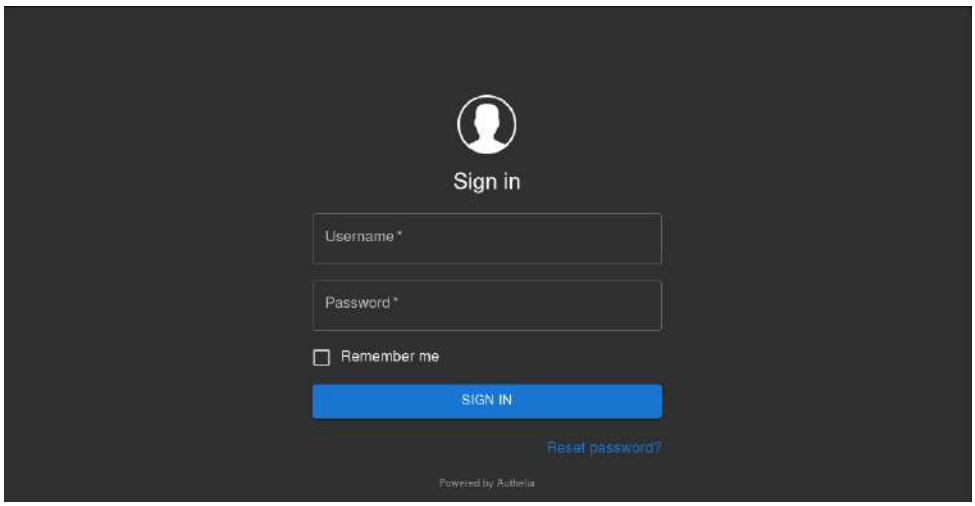

Figura 19 – Captura de tela do Authelia bloqueando um serviço

[3] Disponível em <https://authelia.alice.ufsj.edu.br/>.

Dentre os recursos implementados com o Authelia, destacam-se:

- Autenticação protegida para serviços como **Linux Dash** e **Traefik Dash**, que não possuem controle de acesso nativo.
- Restrição de acesso a páginas e serviços específicos com base nos grupos LDAP.
- Integração com o Traefik, permitindo autenticação centralizada sem necessidade de alterações nos serviços individuais.

8 Resultados

A proposta deste trabalho teve como objetivo imaginar um sistema abstrato que seria capaz de resolver uma série de problemas que ocorrem durante as atividades dos membros do laboratório Alice, sendo intitulado de "Wonderland"(país das maravilhas). Ganhou essa alcunha devido ao fato de que desde sua concepção tínhamos a noção de que a implantação de todos os requisitos levantados seria impossível, tanto por questões técnicas relacionadas a viabilidade, quanto por questões de que este projeto tem um tempo finito para ser concluído. O termo país das maravilhas remete perfeitamente a proposta: um lugar incrível, porém fantasioso.

Por termos a consciência de que seria impossível cumprir todos os requisitos levantados, decidimos que uma abordagem que se adapta às tecnologias existentes seria a mais plausível. A implantação se deu de maneira que todos os requisitos principais foram cumpridos, de forma a termos uma infraestrutura integrada em diversos níveis diferentes.

Para dissecarmos os resultados iremos retomar agora os problemas listados na introdução do trabalho, e discutiremos como cada um dos problemas foi atacado e resolvido.

8.1 Uso cotidiano do laboratório

Foi relatado que a infraestrutura do laboratório era composta por algumas **máquinas independentes**, cada uma com sistemas e configurações **distintos**, fazendo com que os logins e os arquivos pessoais **não fossem sincronizados** entre elas, causando possíveis empecilhos para os usuários.

- Máquinas provisionadas automaticamente com **Debian e Preseed**, garantindo:
  - Configurações padronizadas
  - Instalação automática de pacotes via **PPA próprio**
  - Usuários centralizados no **LDAP**
- Diretórios pessoais sincronizados com **NFS** e acessado remotamente com o **SFTP e Filestash**, permitindo que usuários utilizem qualquer máquina

8.2 Oficinas síncronas

Como dito na introdução, para o funcionamento correto das oficinas são necessários softwares específicos a serem utilizados, bem como artefatos como documentações e arquivos auxiliares, além do uso de plataformas proprietárias para transmissão.

- Máquinas já configuradas previamente com ferramentas necessárias através do PPA
- Videoconferências realizadas através do **Jitsi Meet**, com gravação e possibilidade de transmissão para o **Alice Class**
- Documentação colaborativa em tempo real com **Etherpad** e armazenamento no **Nextcloud** e **Filestash**

8.3 Oficinas assíncronas

Nas oficinas assíncronas, foi colocado como um grande problema a dependência de plataformas proprietárias para vídeo conferência e transferência das dependências necessários. Para lidar com estas questões, utilizamos:

- Videoaulas hospedadas através do **Alice Class**
- Conteúdos complementares associados às aulas (exemplos, documentos, arquivos) organizados no mesmo ambiente, e armazenados usando **Nextcloud**, **Filestash**, **Etherpad** e **Doku Wiki**
- Evita restrições e problemas com plataformas externas como **YouTube**

8.4 Desenvolvimento de software

No cenário de desenvolvimento de software descrito nos problemas envolvidos foi dito que existia uma forte dependência do uso de plataformas como GitHub para a hospedagem do código fonte de trabalhos de desenvolvimento de software. Para isto implantamos uma instância do GiTea, e a utilizamos para a integração com nosso PAA:

- Repositórios hospedados em instância própria do **Gitea**
- Padronização de ambientes de desenvolvimento via gerenciamento de configurações com meta-pacotes e **PPA**
- Automação e integração contínua (CI/CD) feita via **Gitea Actions**
- Independência de plataformas como **GitHub**

8.5 Produções musicais

Sendo um laboratório cujo foco principal é computação musical, diversas atividades realizadas no ALICE envolvem diretamente a produção musical. Atividades como **Jam**

Sessions, oficinas de **Beatmaking** e **Mixagem**, além de experimentações e performances que acarretam na produção de músicas. Novamente foi levantada a questão de dependência de plataformas proprietárias, o que nos levou a:

- Organizar e postar musicais internamente na plataforma **Funkwhale**
- Músicas agrupadas por oficinas, jams ou álbuns, acessíveis remotamente
- Independência de plataformas externas (Google Drive, Soundcloud) e suas limitações

8.6 Reuniões

No laboratório, as reuniões são parte essencial do processo de colaboração, mas frequentemente os participantes não podem estar presentes fisicamente simultaneamente. Essa situação exige o uso de **videoconferências** e ferramentas de comunicação remota proprietárias para garantir que as discussões ocorram de forma produtiva, além do uso de ferramentas para anotações e calendários.

Para solucionar isso, implantamos:

- Videoconferências realizadas internamente pelo **Jitsi Meet**
- Pautas e atas colaborativas feitas em tempo real no **Etherpad** e documentos armazenados no **Nextcloud** e **Filestash**
- Independência total de serviços externos (Google Meet, Google Docs)

8.7 Escrita de artigos, relatórios e monografias

Por se tratar de um grupo de pesquisa com alunos de Iniciação científica, graduação e mestrado, é comum que os discentes estejam constantemente utilizando ferramentas de escrita de textos acadêmicos. Para permitir a **colaboração** nestas escritas, **editores online** como o **Google Docs** ou o Overleaf acabam sendo utilizados de forma que mais de uma pessoa possa contribuir com a produção de conhecimento. Para isto implantamos soluções abertas equivalentes:

- Edição colaborativa com ferramentas próprias:
  - Documentos gerais editados e armazenados no **Nextcloud** e **Filestash**
  - Documentos LaTeX colaborativos via **Overleaf (instância própria)**
  - Arquivos acessíveis remotamente via **Filestash** (interface web via SFTP)

- Eliminação da dependência de Google Docs ou plataformas externas

8.8 Documentações de processos

É comum, em nosso laboratório, que algum sistema demande nova configuração ou que um processo tenha alguns passos para ser feito. Para isto passamos a utilizar uma Wiki interna:

- Plataforma de documentação técnica interna: **DokuWiki**
- Documentação de processos de como utilizar as ferramentas
- Documentação de projetos

8.9 Troca de arquivos e demais artefatos e software

Muitas vezes, na criação científica ou artística há a necessidade de compartilhar arquivos ou informações em geral como links, documentos online, textos, vídeos e outros. Este compartilhamento ocorre de diversas maneiras mas normalmente acaba sendo feito por meio de **ferramentas de comunicação** como o **Whatsapp** ou o **Telegram**.

Para solucionar este problema implantamos diversas soluções abertas:

- Todos os artefatos organizados internamente no **Nextcloud** com controle de acesso
- Interface alternativa com acesso remoto via SFTP no **Filestash**
- Uso de ferramentas colaborativas como forma de armazenamento:
  - Overleaf: Documentos LaTeX
  - Nextcloud: Arquivos
  - Filestash: Arquivos
  - Doku Wiki: Documentação
  - Funkwhale: Músicas
  - Gitea: Códigos
  - Open Gist: Anotações
  - Alice Class: Vídeo aulas
  - Alice Index: Indexação de arquivos públicos
- Independência de serviços externos (Google Drive, Onedrive, etc.)

8.10 Capacitação e Sustentabilidade

Para garantir a continuidade do sistema, foi criado um plano de capacitação técnica para os membros do laboratório. O foco foi facilitar o uso da infraestrutura implantada, promover mais autonomia entre os usuários e reduzir a dependência de pessoas específicas para a manutenção e extensão do sistema. Para isto foram realizadas oficinas de capacitação, mentorias e documentações colaborativas.

8.10.1 Oficinas

As oficinas foram realizadas no laboratório Alice e transmitidas ao vivo utilizando a plataforma Jitsi Meet, e tiveram como objetivo trabalhar fundamentos básicos, apresentar uma visão geral do sistema, abordando decisões de design e futuras manutenções e extensões.

Para isto elas foram divididas em 4 módulos:

- **Módulo 1**: Fundamentos de Linux, redes e serviços
- **Módulo 2**: Uso das ferramentas e fluxo de trabalho (LDAP, Nextcloud, Gitea, Filestash)
- **Módulo 3**: Infraestrutura e fundamentos técnicos (Docker, Traefik, NFS, ACLs)
- **Módulo 4**: Administração prática da infraestrutura

No primeiro módulo, os participantes receberam uma base sólida em sistemas Linux e redes, essenciais para a compreensão da infraestrutura implantada. Foram abordados os seguintes tópicos:

- Uso básico do terminal Linux, manipulação de arquivos, permissões e execução de comandos administrativos.
- Conceitos fundamentais de redes, incluindo endereços IP, portas e protocolos de comunicação.
- Introdução a sockets TCP e sua aplicação em protocolos como HTTP, RTMP e SSH.
- Noções sobre servidores de aplicação e a configuração de serviços na infraestrutura do laboratório.

Ao término deste módulo, os participantes passaram a compreender os princípios necessários para interagir com a infraestrutura implantada e administrar serviços básicos no ambiente Linux.

No segundo módulo, os participantes experimentaram de forma prática os serviços implantados, utilizando as ferramentas e compreendendo como elas se integravam ao fluxo de trabalho do laboratório. Foram trabalhados os seguintes pontos:

- Criação e gerenciamento de usuários no LDAP, simulando a adição de novos membros ao sistema.
- Uso de ferramentas colaborativas, explorando suas possibilidades no contexto do laboratório.
- Acesso remoto a arquivos via SFTP e utilização do Filestash como alternativa web.
- Sincronização e versionamento de código utilizando o Gitea, explorando repositórios, permissões e automação de processos.

Esse módulo familiarizou os participantes com o uso prático dos serviços disponíveis e os incentivou a incorporar essas ferramentas ao seu fluxo de trabalho.

No terceiro módulo, os participantes aprofundaram os conceitos e tecnologias fundamentais discutidos ao longo do processo. O foco foi direcionado aos seguintes aspectos:

- Funcionamento e estrutura do LDAP, incluindo esquemas, ACLs e autenticação centralizada.
- Sistemas de arquivos distribuídos, como NFS, e sua integração com o LDAP.
- Introdução ao Docker e à conteinerização, destacando seu uso na infraestrutura do laboratório.
- Arquitetura de redes para comunicação entre os serviços e a aplicação de proxy reverso com Traefik.

Esse módulo assegurou que os participantes compreendessem os fundamentos da infraestrutura, permitindo-lhes tomar decisões informadas ao realizar manutenções e expansões no sistema.

No último módulo, houve um aprofundamento prático na administração do sistema, capacitando os participantes a entender detalhadamente como os serviços foram configurados, além de aprender a mantê-los e expandi-los. Entre os tópicos abordados, destacaram-se:

- Gerenciamento dos contêineres Docker utilizados na infraestrutura.
- Configuração do LDAP e ajustes de permissões através de ACLs.
- Administração do provisionamento das máquinas do laboratório via Debian Preseed.
- Estratégias para integração de novos serviços ao LDAP, Authelia e Traefik.

Ao final desse módulo, os participantes estavam aptos a gerenciar a infraestrutura de forma independente, garantindo sua continuidade e evolução.

8.10.2 Documentação e sustentabilidade

Além das oficinas, foi necessário a criação de uma documentação extensiva que fosse capaz de descrever o sistema, explicar decisões de design, e prover tutoriais e instruções necessárias para a utilização, manutenção e extensão do projeto. Tudo isto foi hospedado em nossa Wiki, que foi criada através de um esforço coletivo de todos os membros do laboratório, que contribuíram escrevendo sobre os assuntos que possuem mais afinidade, além de escreverem seções sobre seus próprios trabalhos.

- A **Wiki do laboratório** centraliza a documentação técnica e colaborativa.
- Inclui tutoriais, registros de aulas, e manuais práticos.
- Participantes foram incentivados a contribuir com a documentação.
- A documentação se tornou um repositório vivo de conhecimento e facilitou a integração de novos membros.

8.10.3 Mentorias específicas

Alguns alunos possuíam interesses em áreas específicas receberem acompanhamento adicional. Por exemplo, um participante com maior interesse em provisionamento de sistemas recebeu uma mentoria focada no uso do Debian Preseed para instalação automatizada. Da mesma forma, outro participante, interessado na administração de servidores, participou de um treinamento para gestão de contêineres e extensão dos serviços.

9 Considerações Finais

O presente trabalho teve como principal objetivo a implementação de um sistema capaz de fornecer ao Laboratório Alice, da UFSJ, uma infraestrutura integrada que abrangesse autenticação, sincronização, colaboração e gerenciamento de sistemas. A proposta central foi oferecer uma alternativa baseada em software livre e de código aberto, eliminando a dependência de soluções proprietárias e proporcionando maior autonomia e flexibilidade ao laboratório. Dentro desse escopo, o objetivo inicial foi alcançado com sucesso, garantindo que as ferramentas e serviços disponibilizados atendessem às necessidades fundamentais da infraestrutura do laboratório.

Entretanto, devido à grande abrangência do projeto, o aprofundamento em cada uma das aplicações e ferramentas hospedadas foi limitado. O foco esteve na implementação e integração dos serviços, deixando espaço para futuras customizações e extensões que poderiam otimizar ainda mais o sistema. Cada aplicação possui um vasto potencial de configuração e adaptação, o que significa que há margem para melhorias e ajustes que possam tornar o ambiente ainda mais eficiente e alinhado às demandas do laboratório.

Além da implementação do sistema, foram ministradas aulas com o intuito de capacitar os alunos do laboratório a compreenderem a infraestrutura, permitindo que eles tivessem condições de realizar manutenções básicas quando necessário. Esse processo de ensino-aprendizagem também visou incentivar a extensão do sistema, capacitando os alunos a expandirem suas funcionalidades conforme novas necessidades surgissem. O impacto dessas aulas foi positivo, pois os participantes demonstraram interesse em entender o funcionamento do sistema e adquirir autonomia no gerenciamento da infraestrutura.

O projeto e as aulas tiveram uma repercussão além do esperado, despertando o interesse de outros laboratórios, departamentos e até mesmo empresas. Essas instituições viram na infraestrutura desenvolvida um modelo viável para suas próprias necessidades, levando à implementação de soluções similares adaptadas a diferentes contextos. Um exemplo notável dessa influência foi o grupo Cacel, que utilizou este trabalho como referência para desenvolver sua própria versão do sistema. A documentação detalhada produzida por meio da Wiki e as aulas ministradas serviram como base para a implantação da solução no grupo, demonstrando a aplicabilidade e relevância deste projeto em diferentes cenários.

Dessa forma, o impacto do presente trabalho vai além do Laboratório Alice, mostrando que a implementação de infraestruturas baseadas em software livre pode ser uma alternativa viável e replicável em diversas instituições. A estrutura desenvolvida abre caminhos para futuras pesquisas, melhorias e expansões, consolidando-se como um modelo

de gestão de sistemas baseado na colaboração, transparência e autonomia tecnológica.

9.1 Dificuldades Encontradas

Durante o desenvolvimento deste trabalho, diversos desafios foram encontrados, abrangendo desde a configuração do LDAP até questões estruturais do laboratório e da infraestrutura de rede da UFSJ. A seguir, são detalhados os principais problemas enfrentados ao longo do processo.

Problemas e desafios no uso e configuração do LDAP

- **Falta de documentação acessível**
  A modificação do LDAP para suportar esquemas personalizados se mostrou uma tarefa difícil devido à escassez de documentação acessível e clara. Muitos dos esquemas essenciais ainda estão em fase de discussão como RFC, como é o caso do *group of Entries*, que possibilita a criação de grupos vazios.

  Em contrapartida, o esquema *group of Names* não permite a existência de grupos sem membros, tornando-se parcialmente incompatível com a estrutura de grupos Unix. Isso ocorre porque, no Unix, o grupo principal do usuário é definido dentro da própria entrada do usuário, e não diretamente no grupo. Como resultado, um grupo pode parecer vazio, mesmo que, na prática, pertença a um usuário. Para contornar essa limitação, foi adotado um *workaround*, onde o próprio usuário é adicionado manualmente ao grupo para garantir a funcionalidade esperada.

- **Esquemas pré-configurados dificultam personalizações**
  Outra grande dificuldade encontrada foi o fato de que muitas distribuições do LDAP trazem esquemas pré-configurados por padrão, tornando as modificações praticamente inviáveis sem a necessidade de reconfiguração completa.

  Um exemplo disso ocorre no Ubuntu e no Debian, que vêm com o esquema NIS ativado por padrão. Esse esquema não possui suporte à interface *memberOf*, que é essencial para listar os grupos aos quais um usuário pertence. Dessa forma, qualquer necessidade de personalização exigiu a remoção da configuração inicial e a reimplementação do serviço do zero, aumentando a complexidade da implementação.

- **Dificuldade de acesso a informações específicas**
  Apesar da grande quantidade de materiais sobre LDAP disponíveis na internet, grande parte dessas informações é voltada para profissionais altamente especializados.

  Isso gerou dificuldades no momento da configuração avançada do serviço, pois, embora a instalação inicial do LDAP seja relativamente simples, o seu ajuste detalhado

para um ambiente específico exige um conhecimento mais aprofundado. A escassez de guias práticos dificultou a execução de personalizações importantes dentro do sistema.

- **Poucos tutoriais sobre integração com plataformas**
  A integração do LDAP com outras plataformas apresentou obstáculos significativos. Não existe um padrão universal de configuração, fazendo com que cada aplicação tenha suas próprias regras e métodos para interagir com o serviço de autenticação.

  Em muitos casos, a documentação dessas plataformas sobre LDAP era insuficiente ou excessivamente técnica, tornando o processo de integração um desafio. Isso demandou extensivos testes e ajustes para garantir que diferentes ferramentas pudessem autenticar usuários corretamente utilizando o servidor LDAP.

- **Falta de suporte ao LDAP em algumas aplicações**
  Algumas aplicações que seriam extremamente úteis dentro do laboratório simplesmente não possuem suporte nativo ao LDAP, tornando sua adoção menos viável em comparação a soluções que já oferecem integração com esse serviço de autenticação.

  Para contornar essa limitação, optou-se pelo uso do Authelia, que implementa um sistema de *Single Sign-On* (SSO) compatível com LDAP. Apesar de ser uma alternativa funcional, essa escolha gerou uma perda importante: a impossibilidade de utilizar os grupos do LDAP para controle de acesso granular. Embora seja possível configurar regras dentro do próprio Authelia para determinar permissões com base em grupos, essas regras ainda são limitadas ao nível de acesso e não oferecem um sistema de papéis mais sofisticado.

Problemas e desafios na integração de aplicações

- **Baixa integração entre ferramentas**
  Um dos principais desafios enfrentados foi a falta de compatibilidade entre as diversas ferramentas utilizadas na infraestrutura.

  Muitas das aplicações de código aberto disponíveis não possuem mecanismos fáceis de integração com outras ferramentas. Algumas sequer oferecem APIs, tornando a comunicação entre serviços uma tarefa complicada e, em alguns casos, inviável.

- **APIs extensas, mas de difícil implementação**
  Algumas aplicações, como o Nextcloud, oferecem APIs extensas que possibilitam uma integração mais profunda com outros serviços.

  No entanto, utilizar essas APIs exige um nível elevado de conhecimento técnico e uma compreensão detalhada da estrutura interna da ferramenta. Isso representa um

obstáculo para um projeto cujo objetivo era disponibilizar soluções prontas para uso, pois demandaria um tempo considerável de estudo e configuração.

- **Limitações em integrações existentes**
  Algumas aplicações permitem integração apenas dentro de um escopo muito restrito.

  O Filestash, por exemplo, só permite interações no *frontend*, sem oferecer mecanismos para que outras aplicações utilizem sua estrutura de gerenciamento de arquivos. Isso reduz significativamente as possibilidades de personalização e impede que a ferramenta seja aproveitada de maneira mais versátil dentro da infraestrutura.

Problema e desafio no laboratório e no uso do sistema

- **Fluxo constante de alunos**
  Alunos se formam, saem do curso, trocam a área de atuação, e com a troca de alunos no laboratório, tornar o projeto ainda mais especĩfico se torna um desafio.

- **Problemas futuros**
  A implementação do sistema não é algo trivial, mesmo com as aulas e a documentação disponível na wiki. Além disso, o fluxo constante de pessoas no laboratório torna a manutenção e a extensão do sistema ainda mais complicadas, especialmente considerando que a pesquisa principal de muitos não está diretamente relacionada a isso.

- **Internet do CTAN**
  A conexão de internet no Campus Tancredo Neves da UFSJ apresenta instabilidades recorrentes, o que impacta diretamente a disponibilidade dos serviços oferecidos. Em determinados momentos, a conexão pode falhar ou apresentar lentidão significativa, dificultando o acesso.

- **Restrições da rede da UFSJ**
  A rede interna da UFSJ possui regras para garantir a segurança e a integridade de seus sistemas eletrônicos. Uma dessas regras estabelece restrições de acesso, impedindo que alguns laboratórios tenham conexão direta com as VLANs[1] onde estão localizados os servidores da universidade. Essa restrição tem um papel importante na segurança da infraestrutura, dificultando o acesso de atores externos aos servidores. No entanto, essa política também gerou desafios em situações específicas, como na realização de cursos ministrados em outros laboratórios da universidade. Como os computadores desses laboratórios estavam em VLANs isoladas, o firewall da instituição bloqueava completamente o acesso aos servidores, impedindo que os

[1] As VLANs (Virtual Local Area Networks) são redes lógicas virtuais que permitem segmentar máquinas dentro da mesma infraestrutura de rede, garantindo maior controle e isolamento entre diferentes setores

participantes acessassem serviços hospedados localmente. Isso representou um obstáculo significativo para atividades que dependiam desses serviços, exigindo soluções alternativas para contornar essa limitação.

9.2 Trabalhos futuros

Com a conclusão deste trabalho, surge uma preocupação legítima em relação à continuidade e manutenção do sistema a longo prazo, especialmente considerando que, com o encerramento desta pesquisa, não estarei mais vinculado ao laboratório. As tecnologias utilizadas estão em constante evolução, assim como os padrões e boas práticas adotados na área. Isso pode gerar desafios futuros relacionados à compatibilidade, atualização e adaptação do sistema às novas demandas e inovações tecnológicas. Dessa forma, torna-se essencial que pesquisas subsequentes sejam conduzidas para garantir que o sistema permaneça funcional, eficiente e alinhado às necessidades do laboratório.

Uma possível linha de investigação futura seria a realização de estudos mais aprofundados sobre fragmentos específicos do sistema. Como este trabalho abordou um escopo amplo, algumas funcionalidades e aplicações não puderam ser exploradas com a profundidade desejada. Ao direcionar pesquisas futuras para aspectos mais específicos, seria possível desenvolver melhorias incrementais, otimizações de desempenho e até mesmo novas funcionalidades que poderiam agregar ainda mais valor ao sistema.

Além disso, também se pode considerar a realização de pesquisas com o objetivo de expandir o escopo deste estudo. O sistema desenvolvido foi concebido com foco no contexto do Laboratório Alice, atendendo às necessidades e particularidades desse ambiente específico. No entanto, há grande potencial para ampliar sua aplicação para outros laboratórios, ou até mesmo para departamentos inteiros dentro da universidade. Trabalhos futuros poderiam explorar adaptações e generalizações do sistema para que ele atenda a uma gama mais ampla de usuários e cenários, garantindo maior abrangência e impacto institucional.

Dessa forma, a continuidade da pesquisa pode seguir tanto no aprofundamento de aspectos específicos do sistema quanto na sua expansão para novos contextos. A realização desses estudos complementares e expansivos pode assegurar que o sistema continue evoluindo e se mantendo útil e relevante ao longo do tempo, mesmo diante das mudanças tecnológicas e institucionais que possam surgir.

Referências

AOKI, O. *Debian Reference*. [S.l.], 2024. Disponível em: <https://www.debian.org/doc/manuals/debian-reference/ch02.en.html>. Acesso em: 25 abr. 2024. Citado nas páginas 40 e 41.

APACHE, S. F. *Apache Directory Server*. 2006. <https://directory.apache.org/>. Servidor de Diretório LDAP. Citado na página 75.

ARCHLINUXWIKI. *Meta package and package group*. 2024. Disponível em: <https://wiki.archlinux.org/title/pacman>. Acesso em: 25 abr. 2024. Citado na página 41.

ARCHLINUXWIKI. *Meta package and package group*. 2024. Disponível em: <https://wiki.archlinux.org/index.php?title=Meta_package_and_package_group&action=history>. Acesso em: 25 abr. 2024. Citado na página 41.

Bartosz Fenski. *SSHFS (1) - Linux man page*. [S.l.], 2011. Disponível em: <https://linux.die.net/man/1/sshfs>. Acesso em: 7 mai. 2024. Citado na página 49.

BERTINO, E.; TAKAHASHI, K. *Identity management: Concepts, technologies, and systems*. [S.l.]: Artech House, 2010. Citado na página 71.

BOTHMA, P. *Design implications of an online collaborative workspace developed using open source software*. [S.l.]: University of Pretoria (South Africa), 2006. Citado na página 20.

COSTA, E. J. S.; SCHIAVONI, F. L. Alicecast: A orquestra do saber colaborativo. In: *Anais do 1o SIPAUS - Seminário Interdisciplinar de Pesquisa em Artes, Urbanidades e Sustentabilidade*. São João del-Rei / MG: EDUFSJ / UFSJ, 2024. p. 612–621. ISBN 978-65-88228-36-4. Citado na página 69 (2 ocorrências).

FALKO, A. Package manager: The core of a gnu/linux distribution. *Simon's Rock College*, Citeseer, 2007. Citado na página 40.

FERRARI, E. Role-based access control. In: *Access Control in Data Management Systems*. [S.l.]: Springer, 1992. p. 61–75. Citado na página 73.

GASSER, O.; HOLZ, R.; CARLE, G. A deeper understanding of ssh: Results from internet-wide scans. In: IEEE. *2014 IEEE Network Operations and Management Symposium (NOMS)*. [S.l.], 2014. p. 1–9. Citado na página 48.

GEEKSFORGEEKS. *Different methods of Operating System Installation*. 2021. Disponível em: <https://www.geeksforgeeks.org/tips-and-tricks-on-operating-system-installations/>. Acesso em: 14 jan. 2024. Citado na página 38.

HOWES, T.; SMITH, M. *LDAP: Programming directory-enabled applications with lightweight directory access protocol*. [S.l.]: Sams Publishing, 1997. Citado na página 75.

HOWES, T.; SMITH, M.; GOOD, G. S. *Understanding and deploying LDAP directory services*. [S.l.]: Addison-Wesley Professional, 2003. Citado na página 75.

HU, V. C. et al. Attribute-based access control. *Computer*, IEEE, v. 48, n. 2, p. 85–88, 2015. Citado na página 73.

Jitsi. *Jitsi Jibri*. 2003. Disponível em: <https://github.com/jitsi/jibri>. Acesso em: 25 jan. 2024. Citado na página 56.

Jitsi. *Jitsi Jicofo*. 2003. Disponível em: <https://github.com/jitsi/jicofo>. Acesso em: 25 jan. 2024. Citado na página 56.

Jitsi. *Jitsi Meet*. 2003. Disponível em: <https://jitsi.org/jitsi-meet/>. Acesso em: 25 jan. 2024. Citado na página 55.

JOHN, N. A. File sharing and the history of computing: Or, why file sharing is called "file sharing". *Critical Studies in Media Communication*, Taylor & Francis, v. 31, n. 3, p. 198–211, 2014. Citado na página 48.

JORDãO JOSIANE DE FATIMA RIBEIRO, F. L. S. M. D. B. R. Musicalização com o computador. In: *Anais do II Workshop de Computação Teórica e Aplicada*. São João del Rei - MG: DCOMP / UFSJ, 2023. p. 15–15. Disponível em: <https://dcomp.ufsj.edu.br/wocta/images/wocta2023-anais-min.pdf>. Citado na página 14.

LEVY, E.; SILBERSCHATZ, A. Distributed file systems: concepts and examples. *ACM Comput. Surv.*, Association for Computing Machinery, New York, NY, USA, v. 22, n. 4, p. 321–374, dec 1990. ISSN 0360-0300. Disponível em: <https://doi.org/10.1145/98163.98169>. Citado na página 48 (2 ocorrências).

LOPEZ, J.; OPPLIGER, R.; PERNUL, G. Authentication and authorization infrastructures (aais): a comparative survey. *Computers & Security*, Elsevier, v. 23, n. 7, p. 578–590, 2004. Citado na página 71.

MCRAE, A. et al. *Pacman Manual Page*. [S.l.], 2024. Disponível em: <https://man.archlinux.org/man/pacman.8.en>. Acesso em: 25 abr. 2024. Citado na página 41.

MICROSOFT, C. *Active Directory*. 1999. <https://docs.microsoft.com/en-us/windows-server/identity/active-directory>. Serviço de Diretório. Citado na página 75.

MUGLER, J.; NAUGHTON, T.; SCOTT, S. L. Oscar meta-package system. In: IEEE. *19th International Symposium on High Performance Computing Systems and Applications (HPCS'05)*. [S.l.], 2005. p. 353–360. Citado na página 21 (2 ocorrências).

NOVECK, D.; HAYNES, T. *Network File System (NFS) Version 4 Protocol*. [S.l.], 2015. Disponível em: <https://www.rfc-editor.org/rfc/rfc7530>. Citado na página 49.

OPENLDAP, P. *OpenLDAP*. 1998. <https://www.openldap.org/>. Serviço de Diretório LDAP. Citado na página 75.

Oracle. *Install an Operating System Using PXE Network Boot*. [S.l.], 2017. Disponível em: <https://docs.oracle.com/en/servers/x86/x7-2/os-installation-guide/gqsnx.html>. Acesso em: 14 jan. 2024. Citado na página 39.

ORG, F. ffmpeg documentation. 2020. Disponível em: <https://ffmpeg.org/documentation.html>. Acesso em: 25 jan. 2024. Citado na página 56.

OVENS, S. *The evolution of package managers*. 2018. Disponível em: <https://opensource.com/article/18/7/evolution-package-managers>. Acesso em: 25 abr. 2024. Citado na página 41.

phoenixNAP. *O que é Provisionamento de Usuários?* 2024. Disponível em: <https://phoenixnap.pt/gloss%C3%A1rio/provisionamento-de-usu%C3%A1rios>. Acesso em: 25 ago. 2024. Citado na página 74.

PRESS, N. *Nortel Data Networking Technology*. [S.l.]: Nortel Press, 2008. Citado na página 48.

RATIS, L. C.; MOURA, J. G. de; ROSSI, J. Gerenciamento remoto dos computadores dos laboratórios de informática do ifsuldeminas, usando a ferramenta fog project. *15º JORNADA CIENTÍFICA E TECNOLÓGICA E 12 º SIMPÓSIO DE PÓS-GRADUAÇÃO DO IFSULDEMINAS*, v. 15, n. 3, 2023. Citado na página 20.

Red Hat. *Gerenciamento de configuração*. 2023. Disponível em: <https://www.redhat.com/pt-br/topics/automation/what-is-configuration-management>. Acesso em: 25 ago. 2024. Citado nas páginas 38 e 40.

Red Hat. *O que é provisionamento?* 2023. Disponível em: <https://www.redhat.com/pt-br/topics/automation/what-is-provisioning>. Acesso em: 25 ago. 2024. Citado na página 38.

SANDBERG, R. The sun network file system: Design, implementation and experience. In: *in Proceedings of the Summer 1986 USENIX Technical Conference and Exhibition*. [S.l.: s.n.], 1986. Citado na página 49 (2 ocorrências).

SCHIAVONI, F. L. et al. Alice no país da pandemia (2021). In: *Proceedings of the 18th Brazilian Symposium on Computer Music*. Recife - PE: [s.n.], 2021. p. 266–271. ISSN 2175-6759. Citado na página 14.

SCHIAVONI, F. L. et al. Alice. In: *Anais do 1o SIPAUS - Seminário Interdisciplinar de Pesquisa em Artes, Urbanidades e Sustentabilidade*. São João del-Rei / MG: EDUFSJ / UFSJ, 2024. p. 483–504. ISBN 978-65-88228-36-4. Citado na página 14.

Selenium Project. *Selenium: Ferramenta de automação para navegadores*. 2015. Disponível em: <https://www.selenium.dev/>. Acesso em: 25 jan. 2024. Citado na página 56.

SHEPLER, S. et al. *NFS version 4 protocol*. [S.l.], 2000. Citado na página 49.

SILBERSCHATZ, A.; GALVIN, P. B.; GAGNE, G. *Operating System Concepts, Windows XP update*. [S.l.]: John Wiley & Sons, 2006. 705-725 p. Citado na página 48.

SOUSA, J. C.; SCHIAVONI, F. L. Depurando o underground: artistas independentes capacitando-se em produção musical com software livre. In: *Anais do XXXIII Congresso da ANPPOM*. [S.l.]: ANPPOM, 2023. p. 1–16. ISSN 1983-5973. Citado na página 14.

SOUSA, J. C.; SOUSA, E. S.; SCHIAVONI, F. L. Cultura underground e software livre. In: *Anais do 8º Congresso Internacional de Arte, Ciência e Tecnologia e Seminário de Artes Digitais 2023*. Labfront/UEMG, 2023. p. 1–8. ISSN 2674-7847. Disponível em: <https://doi.org/10.5281/zenodo.10413450>. Citado na página 14.

SOUZA, C. E. O.; SCHIAVONI, F. L. Mixxx e suas possibilidades de mixagem. In: *Anais do 8º Congresso Internacional de Arte, Ciência e Tecnologia e Seminário de Artes Digitais 2023*. Labfront/UEMG, 2023. p. 1–10. ISSN 2674-7847. Disponível em: <https://doi.org/10.5281/zenodo.10413492>. Citado na página 14.

SOUZA, C. E. O. de; SCHIAVONI, F. L. CriaÇÃo coletiva em orquestras de celulares. In: *Anais do XXI Congresso de Produção Científica e Acadêmica da UFSJ / XXX SIC - Seminário de Iniciação Científica*. São João del-Rei / MG: PROPE / UFSJ, 2024. p. 1–12. Citado na página 14.

SSHFS, D. *SSHFS: SSH File System*. [S.l.], 2024. Disponível em: <https://github.com/libfuse/sshfs>. Acesso em: 25 maio 2024. Citado na página 49.

STEFAN, O. Version control systems. *Computer Systems and Telematics pp*, p. 11–13, 2009. Citado na página 54.

TEAM, N. *Nix 2.22 Manual*. [S.l.], 2024. Disponível em: <https://nix.dev/manual/nix/2.22>. Acesso em: 24 abr. 2024. Citado na página 42.

TEAM, N. *Nix 2.22 Manual, profiles*. [S.l.], 2024. Disponível em: <https://nix.dev/manual/nix/2.22/package-management/profiles>. Acesso em: 24 abr. 2024. Citado na página 42.

Ubuntu Community. *MetaPackages*. 2024. Disponível em: <https://help.ubuntu.com/community/MetaPackages>. Acesso em: 25 abr. 2024. Citado nas páginas 40 e 41.

VILLANUEVA, J. C. *What is WebDAV?* 2024. Disponível em: <https://www.jscape.com/blog/what-is-webdav>. Acesso em: 9 jan. 2025. Citado na página 48.

ZOLKIFLI, N. N.; NGAH, A.; DERAMAN, A. Version control system: A review. *Procedia Computer Science*, Elsevier, v. 135, p. 408–415, 2018. Citado na página 54 (2 ocorrências).