
**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE**
**CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA**
**CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**JOÃO PEDRO SANTINO ESPÍNDULA**

# SISTEMA DE FAILOVER AUTOMATIZADO COM MONITORAMENTO SINTÉTICO PARA O LDAP DO LABORATÓRIO DE SISTEMAS DISTRIBUÍDOS DA UFCG

**CAMPINA GRANDE - PB**

**2023**

**JOÃO PEDRO SANTINO ESPÍNDULA**

# SISTEMA DE FAILOVER AUTOMATIZADO COM MONITORAMENTO SINTÉTICO PARA O LDAP DO LABORATÓRIO DE SISTEMAS DISTRIBUÍDOS DA UFCG

**Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.**

**Orientadora: Professor Dr. Thiago Emmanuel Pereira.**

**CAMPINA GRANDE - PB**

**2023**

**JOÃO PEDRO SANTINO ESPÍNDULA**

# SISTEMA DE FAILOVER AUTOMATIZADO COM MONITORAMENTO SINTÉTICO PARA O LDAP DO LABORATÓRIO DE SISTEMAS DISTRIBUÍDOS DA UFCG

**Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.**

## BANCA EXAMINADORA:

**Professor Dr. Thiago Emmanuel Pereira**

**Orientador – UASC/CEEI/UFCG**

**Professora Dr. Rohit Gheyi**

**Examinador – UASC/CEEI/UFCG**

**Professor Tiago Lima Massoni**

**Professor da Disciplina TCC – UASC/CEEI/UFCG**

**Trabalho aprovado em: 14 de fevereiro de 2023.**

**CAMPINA GRANDE - PB**

# RESUMO (ABSTRACT)

Lightweight Directory Access Protocol (LDAP) is a protocol targeted to applications that provide interactive read-write access to a directory. This protocol is used by the Distributed Systems Laboratory (LSD) at the Federal University of Campina Grande (UFCG) in storing user data for internal systems and to provider their authentication. In cases of failure of this system — whether network failures, lack of resources or any other — several other laboratory systems are compromised, since the user needs to be authenticated to use them. This includes services in the lab's own data center as well as services in public clouds. In this work, we will plan and develop a solution for the implementation of a failover targeting LSD's LDAP. This failover will aim to guarantee access to LSD user data even if the LDAP system becomes unavailable. It will do this by monitoring that system and, in case of failure, automatically deploying a replica of it in a public cloud.

# Sistema de Failover automatizado com monitoramento sintético para o LDAP do Laboratório de Sistemas Distribuídos da UFCG

João Pedro Santino Espíndula
Departamento de Sistemas e Computação
Universidade Federal de Campina Grande
Campina Grande, Paraíba — Brasil
joao.espindula@ccc.ufcg.edu.br

Thiago Emmanuel Pereira
Departamento de Sistemas e Computação
Universidade Federal de Campina Grande
Campina Grande, Paraíba — Brasil
temmanuel@computacao.ufcg.edu.br

## Link do repositório

https://github.com/pedroespindula/ldap-failover

## Resumo

O Lightweight Directory Access Protocol (LDAP) é um protocolo direcionado a aplicativos de gerenciamento que fornecem acesso interativo de leitura e gravação a um diretório. Esse protocolo é utilizado pelo Laboratório de Sistemas Distribuídos (LSD) na Universidade Federal de Campina Grande (UFCG) no armazenamento de dados dos usuários para sistemas internos em conjunto com sua autenticação. Em casos de falha desse sistema — sejam falhas de rede, falta de recursos ou qualquer outro — vários outros sistemas do laboratório ficam comprometidos, uma vez que o usuário precisa estar autenticado para a utilização deles. Isso inclui serviços no próprio data center do laboratório, bem como serviços em nuvens públicas. Neste trabalho, iremos projetar e desenvolver uma solução para a implantação de um failover visando o LDAP do LSD. Esse failover terá como objetivo garantir o acesso aos dados dos usuários do LSD mesmo que o sistema do LDAP se torne indisponível. Ele fará isso através do monitoramento desse sistema, e, em caso de falha, a implantação automática de uma réplica dele em uma nuvem pública.

## Keywords

LDAP, Failover, nuvem, AWS, Infraestrutura.

## 1. INTRODUÇÃO

No laboratório de Sistemas Distribuídos (LSD) da Universidade Federal de Campina Grande (UFCG), toda a autenticação de usuários e parte do armazenamento dos seus dados é feito a partir do Lightweight Directory Access Protocol (LDAP) [2]. Aplicações críticas para os usuários — como o seu próprio acesso aos computadores do laboratório — dependem do LDAP e não funcionam corretamente caso ele esteja indisponível. Atualmente, este não possui nenhum mecanismo de recuperação caso ocorram falhas. Isso faz com que o LDAP se torne um ponto único de falha [3] para o data center do laboratório.

O LDAP foi um protocolo criado em 1993 e visa prover uma interface simples para aplicações que permitem o acesso a diretórios através da definição de políticas e permissões de acesso [4]. Ele é uma melhoria do protocolo Directory Access Protocol (DAP) através da adoção do protocolo TCP/IP como meio de comunicação entre seus pares [1].

Para o acesso de um diretório por um usuário na base LDAP, é necessário que aquele usuário apresente credenciais de acesso para sua autenticação e autorização. O servidor LDAP consultará a sua base de usuários e validará o acesso requisitado. Caso aquele usuário esteja com as permissões corretas para o acesso, o LDAP concederá o acesso.

Tendo em vista essa utilidade, houveram diversas implementações deste protocolo [5] que permitiram a utilização dessa funcionalidade em conjunto com outros serviços. Ou seja, ao invés de se ter como alvo um diretório, se tem como alvo um serviço. Nesse sentido, caso o LDAP valide as credenciais e as permissões do usuário, ele terá acesso ao serviço requisitado.

Especificamente no LSD, é utilizado o OpenLDAP (https://www.openldap.org/), sendo uma implementação OpenSource do protocolo. Este software está implantado no data center do LSD em uma máquina virtual, gerenciada pela equipe de suporte do laboratório. Atualmente, esse serviço possui apenas uma réplica, facilitando a ocorrência de indisponibilidade tanto do próprio sistema, como também de sistemas dependentes. Ou seja, não havia um mecanismo de recuperação de falhas para esse sistema.

Além disso, já houve casos onde o data center do LSD se tornou indisponível totalmente devido a falhas de energia da UFCG ou de falhas de rede advindas da Rede Nacional de Pesquisa (RNP — https://www.rnp.br/) — conexão da UFCG com a internet. Isso afetou sistemas externos como o Google Workspaces que dependiam do LDAP e que deveriam estar disponíveis para o usuário mesmo com alguma falha do data center. Nesse caso em específico, os usuários não conseguiam acessar seus e-mails através do Google Workspace, pois estes estavam vinculados a base de dados LDAP que não podia ser acessada.

Buscando evitar que tanto sistemas internos do data center, como também sistemas externos a ele sejam afetados pela indisponibilidade do LDAP, esse projeto propõe que seja estudado e planejado uma solução para haver um mecanismo de failover para esse sistema.

Caso o LDAP fique indisponível, parte dos sistemas internos e externos do Laboratório de Sistemas Distribuídos também ficarão indisponíveis. Isso acontece, pois o LDAP é usado para autenticação e autorização nestes serviços. Ou seja, todos os outros sistemas o utilizam para identificar qual usuário está executando uma operação e se ele tem a permissão necessária para fazer isso.

Mesmo com a adição de réplicas no cluster LDAP para aumentar a disponibilidade, existem ainda falhas que podem comprometer todo o cluster. Um exemplo disso é a ocorrência de uma falha na rede que conecta o serviço LDAP à internet. Isso faria com que o sistema estivesse disponível para os sistemas internos (rede privada), mas indisponível para sistemas externos que o utilizam (rede externa).

Com esse trabalho, projetamos e desenvolvemos a solução necessária para que o LDAP consiga ser resiliente o suficiente para superar problemas de indisponibilidade no domínio da UFCG. Com isso, nesse trabalho, conseguimos contribuir com:

- Um **Módulo de Detecção de falhas** para identificação de indisponibilidade do serviço LDAP;
- Um **Módulo de Backup** para recuperação e armazenamento dos dados do serviço LDAP;
- A conteinerização do serviço LDAP em conjunto com os **Módulos de Backup** e **de Detecção de Falhas**;
- A infraestrutura necessária para a implantação de uma réplica do sistema em uma nuvem pública;
- A automatização do processo de implantação desse serviço nessa nuvem pública.

Nesse sentido, caso haja algum problema com o serviço, não haveria perda de disponibilidade de outros serviços dependentes, já que uma nova réplica do sistema será implantada de forma automática.

## 2. ARQUITETURA DA SOLUÇÃO

Essa seção visa prover uma visão geral sobre a arquitetura da solução e sobre as decisões tomadas para a criação dela.

### 2.1 VISÃO GERAL

Primeiramente, temos dois componentes, o **Sistema Primário** do LDAP (no LSD) e o **Sistema de Backup** (na AWS).

O **Sistema Primário** é o sistema que está em uso normalmente pelos usuários do LSD e é o sistema que está se monitorando para identificação de falhas.

O **Sistema de Backup** é uma cópia do **Sistema Primário** que será ativado em caso de falhas. Esse sistema deve utilizar um *snapshot* dos dados e ter o mesmo comportamento do **Sistema Primário**.

Para que o **Sistema de Backup** seja ativado corretamente, é necessário que ele tenha acesso aos dados do **Sistema Primário**. Nessa solução, isso foi feito através de um **Módulo de Backup** que consulta o **Sistema Primário** e armazena os seus dados em um **Serviço de Armazenamento** da AWS (S3).

Além disso, é necessário também a identificação de falhas do **Sistema Primário.** Para isso, é utilizado um **Módulo de Detecção de Falhas** que faz essa detecção e implanta automaticamente o **Sistema de Backup**.

O **Módulo de Backup** consulta o **Sistema Primário** segundo um período configurável (para esse projeto, diariamente) e armazena os dados atuais dele no **Serviço de Armazenamento**.

O **Módulo de Detecção de Falhas** consulta a cada o **Sistema Primário** segundo um período configurável (para esse projeto, 5 minutos) para identificar se ele está ativo. Em caso de falha no **Sistema Primário**, o **Módulo de Detecção de Falhas** implanta automaticamente o **Sistema de Backup**.

O **Sistema de Backup**, então, recupera os backups no **Serviço de Armazenamento** e fica disponível para ser utilizado pelos usuários.

Dessa forma, temos a seguinte interação entre os componentes e o usuário a partir do diagrama de contexto do modelo C4 [12]:

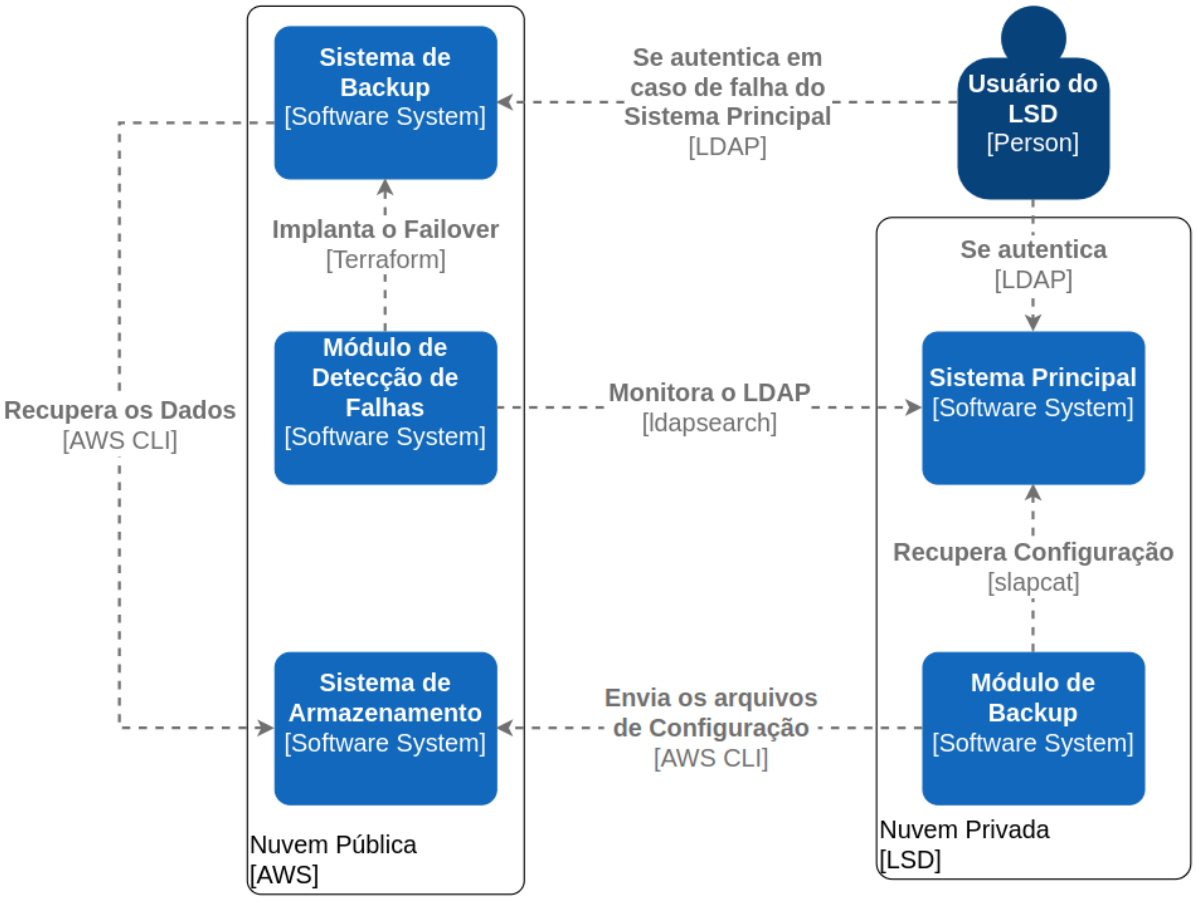


**Fig. 1**: Diagrama de contexto e de interação entre os componentes do Sistema de Failover.

## 2.2 VISÃO ESTRUTURAL

Essa solução é composta por cinco componentes básicos:

- **Módulo de Backup:** Responsável por recuperar os dados do **Sistema Primário** e armazená-los no **Sistema de Armazenamento**;
- **Módulo de Detecção de Falhas**: Responsável por identificar se o **Sistema Primário** está funcional ou não. Além disso, ele é o responsável por implantar de automaticamente o **Sistema de Backup** em caso de falha;
- **Sistema de Armazenamento**: Responsável por armazenar os dados do **Sistema Primário** em caso de falha. Além disso, ele é a fonte de dados do **Sistema de Backup** em sua inicialização;
- **Sistema Primário:** O Sistema Primário que está em funcionamento normalmente. Ele é o sistema que está sendo modificado para que haja tolerância a falhas;
- **Sistema de Backup:** Cópia do **Sistema Primário** que será ativado em caso de falhas.

Esses componentes assumem **Papéis IAM** (*Roles*) específicos para que eles possam efetuar as operações necessárias. Um **Papel IAM** é um conjunto de permissões dado a uma entidade para execução de ações dentro da AWS. Como, por exemplo, uma permissão de criação de novos recursos na AWS ou uma permissão de leitura de dados em um sistema de armazenamento [14].

Os **Papéis IAM** definidos para os componentes citados acima são:

- **Backuper**: Permissão de leitura e escrita no **Serviço de Armazenamento** para armazenamento dos arquivos do **Sistema Primário**. Papel assumido pelo **Módulo de Backup**.
- **Deployer**: Permissão de implantação e de criação de recursos na AWS para implantação do **Sistema de Backup**. Papel assumido pelo **Módulo de Detecção de Falhas**;
- **Execution**: Permissão de execução assumida pelo **Sistema de Backup.** Permite que a AWS gerencie a implantação do **Sistema de Backup.**
- **Task**: Permissões que o container da task possui quando está em execução**.** Permite que o **Sistema de Backup** acesse o **Serviço de Armazenamento**.

## 2.3 TECNOLOGIAS

As tecnologias utilizadas para a criação dessa solução, foram:

- **Terraform** [6]: Para a definição de infraestrutura como código;
- **Scripts Shell**: Para a criação do **Módulo de Detecção de Falhas** e do **Módulo de Backup** do LDAP;
- **Docker** [7]: Para conteinerização da aplicação do LDAP e dos scripts.
- **Amazon Web Services (AWS)** [8]: Para hospedar os componentes da solução;
    - **AWS Lambda** [9]: Para hospedar **Módulo de Backup e de Detecção de Falhas**;
    - **AWS ECS** [10]: Para hospedar o **Sistema de Backup**;
    - **AWS S3** [11]: Para hospedar os arquivos de configuração e de backup do LDAP (**Serviço de Armazenamento**);

## 2.4 DECISÕES ARQUITETURAIS

Esta seção visa descrever as decisões arquiteturais tomadas durante o desenvolvimento desse projeto.

### 2.4.1 UTILIZAÇÃO DAS TECNOLOGIAS

O **Terraform** foi utilizado para o gerenciamento facilitado do estado da infraestrutura e é uma ferramenta amplamente utilizada no mercado [13]. Além disso, ela possui suporte a gerenciamento da nuvem escolhida (AWS) nativamente.

Quanto ao **Shell**, essa linguagem foi escolhida para que se pudesse fazer chamadas nativas a executáveis como o **Netcat** e o **Slapcat**. Além disso, na arquitetura já existente da nuvem do LSD envolvendo o LDAP, já eram utilizados **Scripts Shell** para o gerenciamento do sistema (ex: processos de backup e restauração).

Sobre o **Docker**, ele foi utilizado para que não tivéssemos problema de configuração do LDAP em caso de mudança de sistema operacional, seja versão, seja de distribuição ou até de uma completa mudança do sistema. Dessa forma, conseguimos garantir a estabilidade de implantação da solução independente do sistema e do ambiente em que se está o executando.

A **AWS** foi escolhida tendo em vista a disponibilidade e os recursos providos por essa nuvem pública. Além disso, o autor da solução já possuía experiência com essa nuvem.

### 2.4.2 TEMPO DE RECUPERAÇÃO

Foi avaliado o tempo necessário para a recuperação do **Sistema Primário** do LDAP. Estabeleceu-se que o tempo máximo esperado para a recuperação do **Sistema Primário** é de 30 minutos. Dessa forma, temos um SLO de disponibilidade de aproximadamente 99.9% considerando que temos apenas uma falha no mês [15].

Esse tempo foi utilizado como base para a decisão de frequência de requisições pelo **Módulo de Detecção de Falhas** para o **Sistema Primário**. Com isso, foi adotado um intervalo de 5 minutos entre requisições. Consideramos também que seriam necessários menos de 5 minutos para a implantação do **Sistema de Backup** pelo **Módulo de Detecção de Falhas**.

### 2.4.3 IMPLANTAÇÃO DOS MÓDULOS

Tendo em vista que a solução apresentada visa a recuperação de falhas tanto pontuais (falha do sistema) como catastróficas (falha da nuvem privada), decidiu-se implantar o **Módulo de Detecção de Falhas** fora da nuvem privada do LSD. Com isso, mesmo em caso de falha catastrófica da nuvem do LSD, o **Módulo de Detecção de Falhas** continuará funcionando normalmente e se conseguirá recuperar a falha através do **Sistema de Backup**.

Para o **Módulo de Backup**, foi feita apenas uma adaptação do script que já existia na nuvem do LSD. Dessa forma, não alteramos sua implantação, apenas seu código-fonte. Esse **Módulo de Backup** foi implantado no LSD, pois era necessário que o slapcat (comando que faz o backup) fosse executado na mesma máquina que está sendo executado o servidor. Mesmo assim, isso não impactou a resiliência do failover, já que o script é executado apenas uma vez por dia e os dados de backup estão fora da nuvem do LSD.

## 3. METODOLOGIA

Essa seção visa explicitar quais foram os processos adotados para a construção dessa solução, quais foram os principais desafios encontrados e como esses desafios foram resolvidos.

## 3.1 PROCESSO ADOTADO

Foram utilizados princípios de desenvolvimento ágil, onde foram iterativamente entregues funcionalidades do software como:

- Conteinerização do LDAP;
- Implantação do LDAP na AWS;
- Módulo de Backup;
- Recuperação do backup pelo LDAP;
- Módulo de Detecção de Falhas.

Ocorreram também reuniões com o orientador do projeto para haver a validação do que estava sendo feito, esclarecimento de dúvidas, e discussão sobre possíveis modificações.

Além disso, foram consultados administradores de sistema que gerenciam o ambiente do LDAP no LSD para que estes possam esclarecer potenciais dúvidas e também validar o trabalho que está sendo desenvolvido.

Ao final dessa etapa, todo o ferramental necessário para a detecção de falhas no LDAP bem como o processo de implantação da réplica do serviço foram automatizados e estavam funcionando corretamente.

## 3.2 DESAFIOS E SOLUÇÕES

Durante todo o desenvolvimento desse trabalho, encontramos diversos desafios para a resolução do nosso problema. Nessa seção esclarecemos quais foram esses desafios e como eles foram resolvidos para o nosso caso de uso.

### 3.2.1 CUSTOS

O primeiro desafio que encontramos foram os custos com as implantações. Como poderíamos implementar um failover que não gerasse custos enquanto ele não estivesse ativo (ambiente produtivo)? E ainda, como implementar e testar esse failover e toda a infraestrutura necessária para ele, sem que houvesse custos (ambiente de desenvolvimento)?

Quanto aos custos do ambiente produtivo, criamos um mecanismo para que o Sistema de Backup só fosse ativado quando o Sistema Primário do LDAP ficasse fora do ar. Isso fez com que tivéssemos apenas custos mínimos com a infraestrutura enquanto o Sistema Primário estivesse funcionando e só tivéssemos custos por completo quando ele estivesse fora do ar.

Além disso, mesmo o **Módulo de Detecção de Falhas** e de **Backup** sendo executados como rotinas, a baixa quantidade de execuções desses dois scripts não geram custos na AWS por mês.

Em relação ao ambiente de desenvolvimento, conseguimos não ter custos a partir da utilização de um ambiente de Sandbox provisionado pelo Cloud Guru. Esse ambiente permitiu que criássemos uma conta da AWS de forma temporária sem custos e, em conjunto com a utilização de infraestrutura como código, conseguimos desenvolver nossa infraestrutura. Mesmo que a conta da AWS fosse deletada devido ao seu caráter temporário, conseguiríamos manter nosso progresso de desenvolvimento devido à infraestrutura estar sendo criada via código.

Os custos relativos à assinatura do Cloud Guru foram arcados pela empresa do autor. Isso aconteceu, pois o autor utilizava dessa plataforma para cursos de formação necessários para o seu trabalho.

### 3.2.2 BACKUP NÃO DISPONÍVEL

No ambiente inicial, o backup do LDAP era efetuado na nuvem do LSD. Dessa forma, o LDAP de failover não tinha acesso direto a esse backup, quebrando a consistência de dados e configurações.

Para resolver isso, adaptamos o **Módulo de Backup** para que as cópias dos arquivos estivessem disponíveis no **Sistema de Armazenamento** (S3) e que o **Sistema de Backup** tivesse acesso.

As modificações necessárias foram:

- Adição de exportação para o **Sistema de Armazenamento** através do **Módulo de Backup**;
- Criação de *role* específica da AWS para o **Módulo de Backup** (Backuper);
- Criação da infraestrutura necessária para o **Sistema de Armazenamento** (Bucket em si e políticas de acesso);

Com isso, ao ser executado, o **Sistema de Backup** faz o download dos arquivos do **Sistema de Armazenamento** e os utiliza como sua base de dados dentro do contêiner.

### 3.2.3 MECANISMO DE MONITORAMENTO

Gostaríamos de ter um mecanismo de monitoramento que ele ativasse o nosso failover em caso de falha. No entanto, não se encontrou nenhum monitoramento que se encaixasse no cenário da solução apresentada. Isso aconteceu dado que:

- A solução contempla um protocolo de um sistema específico (LDAP);
- Deveria ser possível, a partir da identificação da falha, realizar uma implantação automática;
- A solução deveria utilizar algo que não gerasse custos.

Com isso, foi necessário desenvolver o próprio mecanismo de monitoramento em conjunto com a ativação da implantação automática.

# 4. RESULTADOS

Nessa seção, é explicitado como foi feita a avaliação e a validação da solução criada. Além disso, é explicitado quais são as limitações dessa solução.

## 4.1 AVALIAÇÃO DA SOLUÇÃO

As avaliações ocorreram por falhas simuladas em uma cópia local do sistema LDAP do LSD. Isso evitou que o sistema de produção do LDAP se tornasse inoperante. Essa cópia local do sistema teve como ênfase os dados e a configuração de implantação adotada na nuvem do LSD.

A primeira validação feita foi o funcionamento do **Módulo de Backup**. Validou-se se o **Módulo de Backup** conseguia recuperar os dados do **Sistema Primário** e armazená-los no **Serviço de Armazenamento**. A validação foi feita a partir do agendamento da execução do **Módulo de Backup** tendo em vista o **Sistema Primário**. Com esse agendamento, observou-se tanto os logs do **Módulo de Backup** (**Fig. 2**) para identificarmos se ele havia sido executado corretamente, como também observou-se o **Serviço de Armazenamento** para garantir que os arquivos haviam sido armazenados corretamente (**Fig. 3**).

```
root@b736c34eaebd:/ldap# date
Thu Jan 26 00:33:05 UTC 2023
root@b736c34eaebd:/ldap# crontab -l
34 0 * * * bash /ldap/backup-ldap.sh >> /ldap/backup-ldap.logs 2>&1
root@b736c34eaebd:/ldap# ls
backup-ldap.sh  conf  entrypoint.sh
root@b736c34eaebd:/ldap# ls
backup-ldap.logs  backup-ldap.sh  conf  entrypoint.sh
root@b736c34eaebd:/ldap# cat backup-ldap.logs
Recuperando os usuários do LDAP
Usuários recuperados com sucesso! Segue uma amostra do arquivo:
dn: dc=lsd,dc=ufcg,dc=edu,dc=br
o: lsd.ufcg.edu.br
dc: lsd
structuralObjectClass: organization
entryUUID: d2d421fc-70ea-1034-802e-332d1a8f60dd
creatorsName: cn=admin,dc=lsd,dc=ufcg,dc=edu,dc=br
createTimestamp: 20150406205412Z
objectClass: dcObject
objectClass: organization
entryCSN: 20150408151657.408719Z#000000#000#000000
Recuperando as configurações do LDAP
Configurações recuperadas com sucesso! Segue uma amostra do arquivo:
dn: cn=config
objectClass: olcGlobal
cn: config
olcArgsFile: /var/run/slapd/slapd.args
olcLogLevel: stats
olcPidFile: /var/run/slapd/slapd.pid
olcSizeLimit: 2000
olcToolThreads: 1
structuralObjectClass: olcGlobal
entryUUID: d2933a3e-70ea-1034-8d02-9f4748619e31
Enviando arquivos para o S3
upload: ../backup/config.ldif to s3://lsd-ldap-backup/config.ldif
upload: ../backup/users.ldif to s3://lsd-ldap-backup/users.ldif
root@b736c34eaebd:/ldap# date
Thu Jan 26 00:34:18 UTC 2023
root@b736c34eaebd:/ldap#
```

**Fig. 2:** Execução automática do **Módulo de Backup** com recuperação dos dados do LDAP e envio para o **Sistema de Armazenamento**

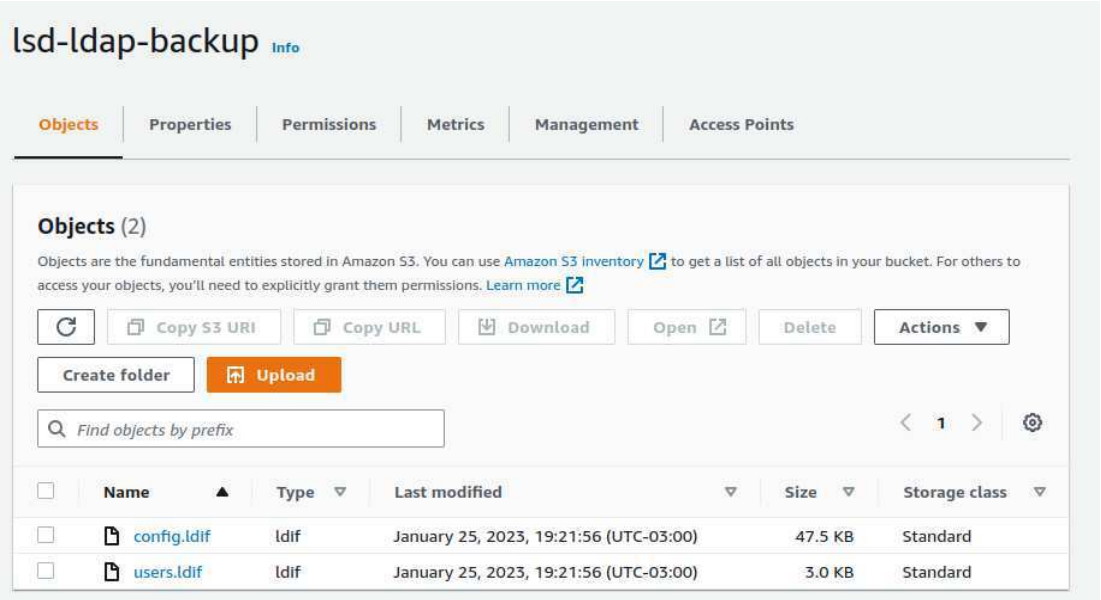


**Fig. 3:** Arquivos de configuração do LDAP no **Sistema de Armazenamento**.

Validou-se também que o **Módulo de Detecção de Falhas** conseguia identificar que o sistema estava saudável. Para isso, foi feito o apontamento do **Módulo de Detecção de Falhas** para a cópia local que estava sendo executada normalmente. Ele conseguiu identificar que a cópia local estava saudável e não efetuou nenhuma operação (**Fig. 4**).

Na sequência, foi feita a validação de que o **Módulo de Detecção de Falhas** conseguia identificar que o sistema **não** estava saudável e conseguia implantar automaticamente o **Sistema de Backup**. Foi simulado uma falha na cópia local (**Fig. 6**), identificado se o **Módulo de Detecção de Falhas** conseguiu identificar corretamente a falha (**Fig. 7**) e identificado se o **Sistema de Backup** (**Fig. 8**) tinha sido implantado corretamente.

Por fim, foi feita a validação do funcionamento do **Sistema de Backup**. Houve a identificação de logs atestando o seu funcionamento através do login com um usuário específico configurado na base de dados do **Sistema Primário** (**Fig. 9**, **10** e **11**).

Com essas validações feitas, conseguiu-se validar que o sistema de recuperação de falhas estava funcionando como esperado.

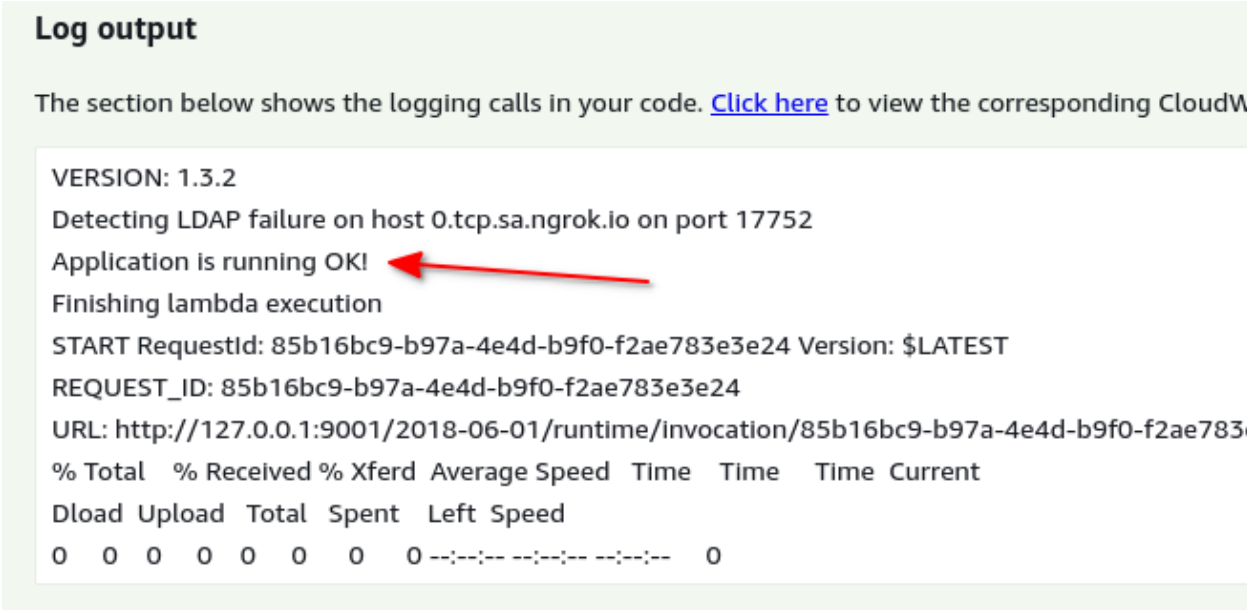


**Fig. 4:** Consulta do **Módulo de Detecção de Falhas** ao **Sistema Primário** sem a execução de nenhuma operação adicional.

```
local-ldap-1  | 63d1aae2 @(#) $OpenLDAP: slapd  (Ubuntu) (May 12 2022 13:52:38) $
local-ldap-1  |         Debian OpenLDAP Maintainers <pkg-openldap-devel@lists.alioth.debian.org>
local-ldap-1  | 63d1aae2 slapd starting
local-ldap-1  | 63d1ac3b conn=1000 fd=14 ACCEPT from IP=172.18.0.1:37698 (IP=0.0.0.0:389)
local-ldap-1  | 63d1ac3b conn=1000 op=0 BIND dn="" method=128
local-ldap-1  | 63d1ac3b conn=1000 op=0 RESULT tag=97 err=0 text=
local-ldap-1  | 63d1ac3c conn=1000 op=1 SRCH base="ou=users,dc=lsd,dc=ufcg,dc=edu,dc=br" scope=2 deref=0
 filter="(objectClass=*)"
local-ldap-1  | 63d1ac3c conn=1000 op=1 ENTRY dn="ou=users,dc=lsd,dc=ufcg,dc=edu,dc=br"
local-ldap-1  | 63d1ac3c conn=1000 op=1 ENTRY dn="cn=joao pedro santino espindula,ou=users,dc=lsd,dc=ufc
g,dc=edu,dc=br"
local-ldap-1  | 63d1ac3c conn=1000 op=1 SEARCH RESULT tag=101 err=0 nentries=2 text=
local-ldap-1  | 63d1ac3c conn=1000 op=2 UNBIND
local-ldap-1  | 63d1ac3c conn=1000 fd=14 closed
local-ldap-1  | 63d1ac3c conn=1001 fd=14 ACCEPT from IP=172.18.0.1:37710 (IP=0.0.0.0:389)
local-ldap-1  | 63d1ac3c conn=1001 op=0 BIND dn="" method=128
local-ldap-1  | 63d1ac3c conn=1001 op=0 RESULT tag=97 err=0 text=
local-ldap-1  | 63d1ac3d conn=1001 op=1 SRCH base="ou=users,dc=lsd,dc=ufcg,dc=edu,dc=br" scope=2 deref=0
 filter="(objectClass=*)"
local-ldap-1  | 63d1ac3d conn=1001 op=1 ENTRY dn="ou=users,dc=lsd,dc=ufcg,dc=edu,dc=br"
local-ldap-1  | 63d1ac3d conn=1001 op=1 ENTRY dn="cn=joao pedro santino espindula,ou=users,dc=lsd,dc=ufc
g,dc=edu,dc=br"
local-ldap-1  | 63d1ac3d conn=1001 op=1 SEARCH RESULT tag=101 err=0 nentries=2 text=
local-ldap-1  | 63d1ac3d conn=1001 op=2 UNBIND
local-ldap-1  | 63d1ac3d conn=1001 fd=14 closed
```

**Fig. 5:** Logs da consulta sendo executada no **Sistema Primário**.

```
local-ldap-1  | 63d1adb2 @(#) $OpenLDAP: slapd  (Ubuntu) (May 12 2022 13:52:38) $
local-ldap-1  |         Debian OpenLDAP Maintainers <pkg-openldap-devel@lists.alioth.debian.org>
local-ldap-1  | 63d1adb2 slapd starting
local-ldap-1  | 63d1adb8 conn=1000 fd=14 ACCEPT from IP=172.18.0.1:56684 (IP=0.0.0.0:389)
local-ldap-1  | 63d1adb8 conn=1000 op=0 BIND dn="" method=128
local-ldap-1  | 63d1adb8 conn=1000 op=0 RESULT tag=97 err=0 text=
local-ldap-1  | 63d1adb8 conn=1000 op=1 SRCH base="ou=users,dc=lsd,dc=ufcg,dc=edu,dc=br" scope=2 deref=0
 filter="(objectClass=*)"
local-ldap-1  | 63d1adb8 conn=1000 op=1 ENTRY dn="ou=users,dc=lsd,dc=ufcg,dc=edu,dc=br"
local-ldap-1  | 63d1adb8 conn=1000 op=1 ENTRY dn="cn=joao pedro santino espindula,ou=users,dc=lsd,dc=ufc
g,dc=edu,dc=br"
local-ldap-1  | 63d1adb8 conn=1000 op=1 SEARCH RESULT tag=101 err=0 nentries=2 text=
local-ldap-1  | 63d1adb8 conn=1000 op=2 UNBIND
local-ldap-1  | 63d1adb8 conn=1000 fd=14 closed
local-ldap-1  | 63d1adb8 conn=1001 fd=14 ACCEPT from IP=172.18.0.1:56696 (IP=0.0.0.0:389)
local-ldap-1  | 63d1adb8 conn=1001 op=0 BIND dn="" method=128
local-ldap-1  | 63d1adb8 conn=1001 op=0 RESULT tag=97 err=0 text=
local-ldap-1  | 63d1adb8 conn=1001 op=1 SRCH base="ou=users,dc=lsd,dc=ufcg,dc=edu,dc=br" scope=2 deref=0
 filter="(objectClass=*)"
local-ldap-1  | 63d1adb8 conn=1001 op=1 ENTRY dn="ou=users,dc=lsd,dc=ufcg,dc=edu,dc=br"
local-ldap-1  | 63d1adb8 conn=1001 op=1 ENTRY dn="cn=joao pedro santino espindula,ou=users,dc=lsd,dc=ufc
g,dc=edu,dc=br"
local-ldap-1  | 63d1adb8 conn=1001 op=1 SEARCH RESULT tag=101 err=0 nentries=2 text=
local-ldap-1  | 63d1adb9 conn=1001 op=2 UNBIND
local-ldap-1  | 63d1adb9 conn=1001 fd=14 closed
^CGracefully stopping... (press Ctrl+C again to force)
[+] Running 1/1
 ⠿ Container local-ldap-1  Stopped                                                    0.3s
canceled
[pedro@pagarme local]$ echo "LDAP PARADO COM KEYBOARD INTERRUPT"
LDAP PARADO COM KEYBOARD INTERRUPT
[pedro@pagarme local]$
```

**Fig. 6:** Interrupção da execução do **Sistema Primário** através de uma interrupção do container docker.

```
2023-01-25T19:32:23.765-03:00  VERSION: 1.3.2
2023-01-25T19:32:23.765-03:00  Detecting LDAP failure on host 0.tcp.sa.ngrok.io on port 17752
2023-01-25T19:32:24.089-03:00  Failure detected!
2023-01-25T19:32:24.089-03:00  Creating the failover...
2023-01-25T19:33:22.125-03:00  Initializing modules...
2023-01-25T19:33:22.140-03:00  Initializing the backend...
2023-01-25T19:33:24.135-03:00  Initializing provider plugins...
2023-01-25T19:33:24.135-03:00  - terraform.io/builtin/terraform is built in to Terraform
2023-01-25T19:33:24.135-03:00  - Reusing previous version of hashicorp/aws from the dependency lock file
2023-01-25T19:33:25.943-03:00  - Using previously-installed hashicorp/aws v4.51.0
2023-01-25T19:33:25.943-03:00  Terraform has been successfully initialized!
2023-01-25T19:33:25.943-03:00  You may now begin working with Terraform. Try running "terraform plan" to see
2023-01-25T19:33:25.943-03:00  any changes that are required for your infrastructure. All Terraform commands
2023-01-25T19:33:25.943-03:00  should now work.
2023-01-25T19:33:25.943-03:00  If you ever set or change modules or backend configuration for Terraform,
2023-01-25T19:33:25.943-03:00  rerun this command to reinitialize your working directory. If you forget, other
2023-01-25T19:33:25.943-03:00  commands will detect it and remind you to do so if necessary.
2023-01-25T19:33:33.722-03:00  Terraform used the selected providers to generate the following execution
2023-01-25T19:33:33.722-03:00  plan. Resource actions are indicated with the following symbols:
2023-01-25T19:33:33.722-03:00  + create
2023-01-25T19:33:33.722-03:00  Terraform will perform the following actions:
2023-01-25T19:33:33.722-03:00  # aws_iam_policy.allow-log-creation will be created
2023-01-25T19:33:33.722-03:00  + resource "aws_iam_policy" "allow-log-creation" {
2023-01-25T19:33:33.722-03:00  + arn = (known after apply)
2023-01-25T19:33:33.722-03:00  + description = "Allow log creation on CloudWatch"
2023-01-25T19:33:33.722-03:00  + id = (known after apply)
```

```
2023-01-25T19:36:47.190-03:00  Apply complete! Resources: 15 added, 0 changed, 0 destroyed.
2023-01-25T19:36:47.191-03:00  Outputs:
2023-01-25T19:36:47.191-03:00  elb_url = "lsd-ldap-load-balancer-3ebd35d516b9a091.elb.us-east-1.amazonaws.com"
2023-01-25T19:36:47.196-03:00  Creation finished!
2023-01-25T19:36:47.196-03:00  Finishing lambda execution
```

**Fig. 7:** Logs de identificação da falha e implantação do **Failover**.

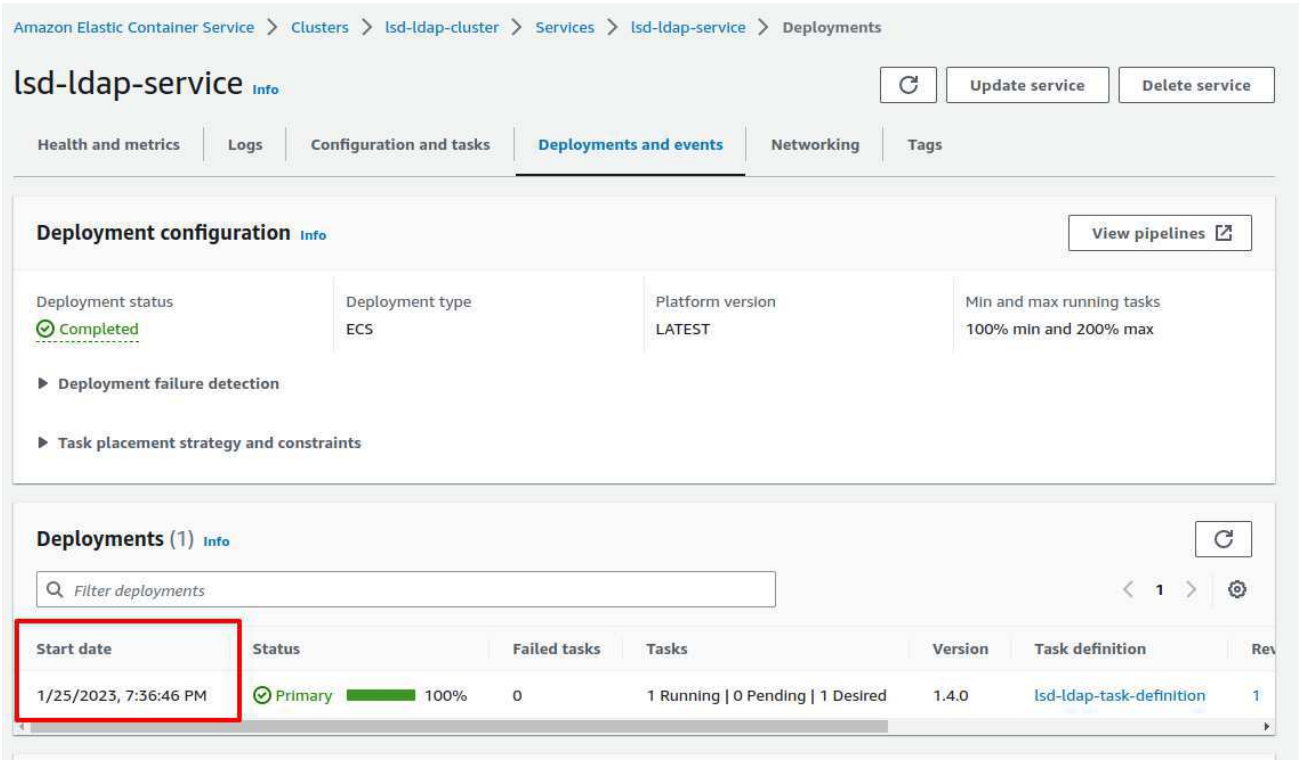


**Fig. 8:** Evidência da implantação do **Sistema de Backup**.

```
[pedro@pagarme ufcg]$ ldapsearch -x -H "ldap://lsd-ldap-load-balancer-3ebd35d516b9a091.elb.us-east-1.ama
zonaws.com:389" -b "ou=users,dc=lsd,dc=ufcg,dc=edu,dc=br" -LLL
dn: ou=users,dc=lsd,dc=ufcg,dc=edu,dc=br
objectClass: organizationalUnit
objectClass: top
ou: users

dn: cn=Joao Pedro Santino Espindula,ou=users,dc=lsd,dc=ufcg,dc=edu,dc=br
cn: Joao Pedro Santino Espindula
uid: joao.espindula
sn: Espindula
givenName: Joao Pedro Santino Espindula
mail: joao.espindula@lsd.ufcg.edu.br
uidNumber: 1690
gidNumber: 1082
loginShell: /bin/bash
homeDirectory: /home/joao.espindula
description: Projeto: TCC Manel, Solicitante: Thiago Emmanuel
metadata: CPF: 98677898409
title: True
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: top
objectClass: lsdUser

[pedro@pagarme ufcg]$ date
qua 25 jan 2023 19:40:48 -03
```

**Fig. 9:** Operação de pesquisa executada no **Sistema de Backup** via CLI.

| Timestamp | Task | Message |
|---|---|---|
| 1/25/2023, 7:41:21 PM | 343f889832414edbbb0e59872aa2fe02 | 63d1b011 conn=1041 op=2 UNBIND |
| 1/25/2023, 7:41:21 PM | 343f889832414edbbb0e59872aa2fe02 | 63d1b011 conn=1041 fd=14 closed |
| 1/25/2023, 7:41:21 PM | 343f889832414edbbb0e59872aa2fe02 | 63d1b011 conn=1041 op=1 SRCH base="ou=users,dc=lsd,dc=ufcg,dc=edu,dc=br" scope=2 deref=0 filter="(objectClass=*)" |
| 1/25/2023, 7:41:21 PM | 343f889832414edbbb0e59872aa2fe02 | 63d1b011 conn=1041 op=1 ENTRY dn="ou=users,dc=lsd,dc=ufcg,dc=edu,dc=br" |
| 1/25/2023, 7:41:21 PM | 343f889832414edbbb0e59872aa2fe02 | 63d1b011 conn=1041 op=1 ENTRY dn="cn=joao pedro santino espindula,ou=users,dc=lsd,dc=ufcg,dc=edu,dc=br" |
| 1/25/2023, 7:41:21 PM | 343f889832414edbbb0e59872aa2fe02 | 63d1b011 conn=1041 op=1 SEARCH RESULT tag=101 err=0 nentries=2 text= |
| 1/25/2023, 7:41:21 PM | 343f889832414edbbb0e59872aa2fe02 | 63d1b011 conn=1041 op=0 BIND dn="" method=128 |
| 1/25/2023, 7:41:21 PM | 343f889832414edbbb0e59872aa2fe02 | 63d1b011 conn=1041 op=0 RESULT tag=97 err=0 text= |
| 1/25/2023, 7:41:21 PM | 343f889832414edbbb0e59872aa2fe02 | 63d1b011 conn=1041 fd=14 ACCEPT from IP=172.31.4.241:6750 (IP=0.0.0.0:389) |

**Fig. 10:** Logs de conexão do **Sistema de Backup** para operação de pesquisa.

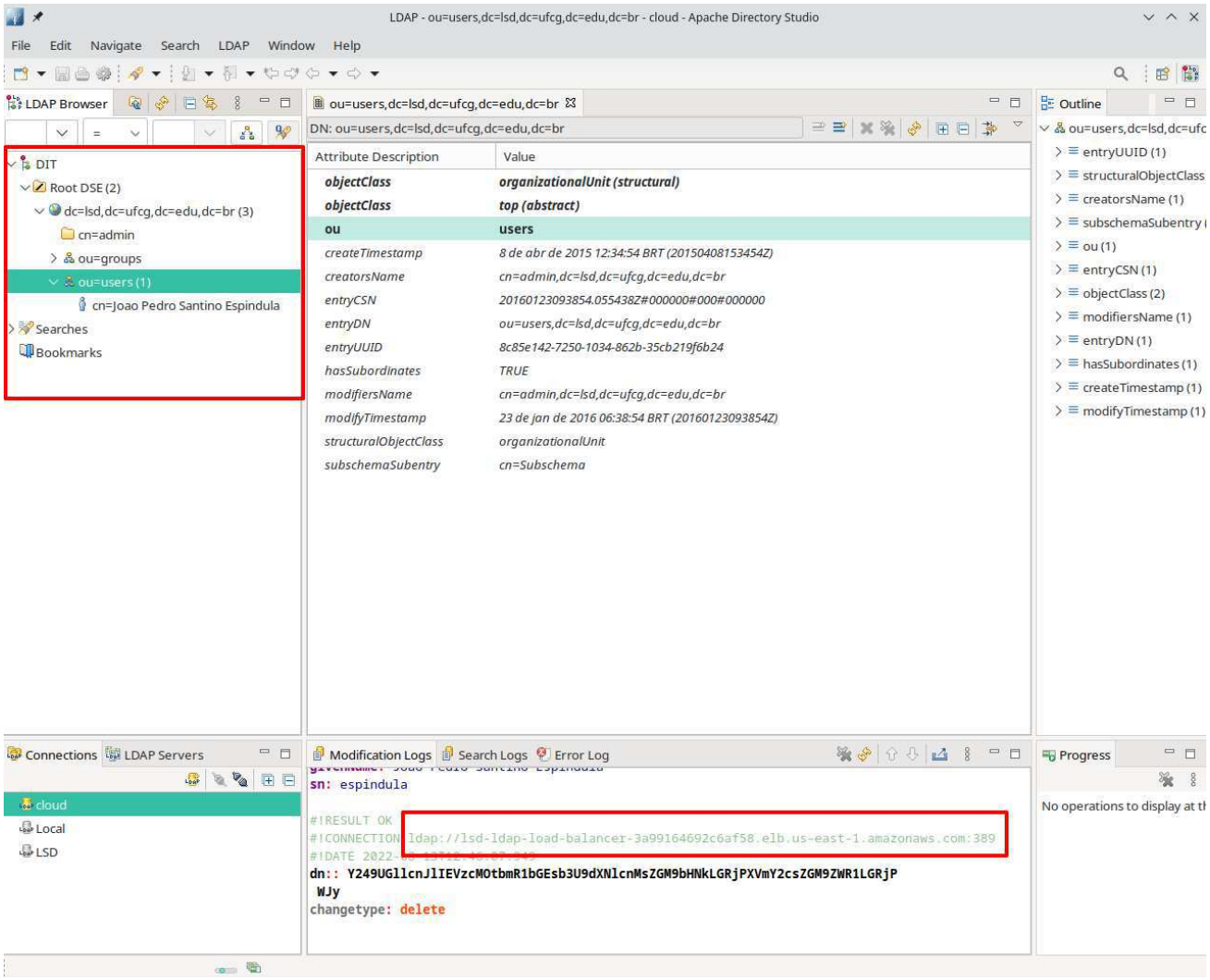


**Fig. 11:** Login no LDAP através do Apache Directory Studio.

## 4.2 LIMITAÇÕES

Nessa seção, procura-se identificar quais são as principais limitações do sistema e por que elas acontecem.

### 4.2.1 DETECÇÃO AUTOMÁTICA DE RECUPERAÇÃO

Essa solução não contempla a detecção automática de recuperação do **Sistema Primário**. Dessa forma, mesmo que o **Sistema Primário** seja recuperado, o **Sistema de Backup** ainda será utilizado. Para que ele deixe de ser utilizado em prol do **Sistema Primário**, são necessárias ações manuais.

Isso aconteceu, pois o objetivo principal dessa solução era viabilizar uma tolerância a falhas do LDAP.

### 4.2.2 TOPOLOGIA DE REDE

Essa solução não contempla  a adequação a topologias de rede privadas da nuvem do LSD. Isso inclui a comunicação entre a nuvem privada do LSD e a nuvem pública da AWS.

Além disso, não são consideradas alterações de apontamentos de DNS e interações com servidores DNS para essa alteração.

## 5. TRABALHOS FUTUROS

Como sugestão de trabalhos futuros, pode-se implantar essa arquitetura de Failover no sistema em produção do LDAP na nuvem do LSD. Esse sistema foi utilizado como exemplo para criação dessa arquitetura, mas, o sistema do LDAP se beneficiaria disso através do ganho de resiliência que essa arquitetura proporciona.

Além disso, pode-se tanto melhorar a solução apresentada atacando suas limitações, como também pode-se utilizar como base a arquitetura da solução para a implantação dela em outro sistema diferente do LDAP.

## 6. REFERÊNCIAS


[1] Yeong, W., Howes, T., & Kille, S. (1993, July). X.500 Lightweight Directory Access Protocol. Internet Engineering Task Force. Retrieved March 10, 2022, from https://www.ietf.org/rfc/rfc1487.txt

[2] Normas de Segurança. (2017, June 6). Wiki LSD. https://wiki.lsd.ufcg.edu.br/index.php/Normas_de_Seguran%C3%A7a

[3] Wikipedia contributors. (2022, January 12). Single point of failure. Wikipedia. https://en.wikipedia.org/wiki/Single_point_of_failure

[4] Willeke, J. (2019, June 20). Ldapwiki: History of LDAP. LDAP Wiki. https://ldapwiki.com/wiki/History%20of%20LDAP

[5] Blanton, S. (2021, September 16). What is LDAP Authentication? JumpCloud. https://jumpcloud.com/blog/what-is-ldap-authentication

[6] Docker: Accelerated, Containerized Application Development. (2023, January 23). Docker. https://www.docker.com/

[7] Hashicorp. (n.d.). Terraform by hashicorp. https://www.terraform.io

[8] Cloud Computing Services - Amazon Web Services (AWS). (n.d.). Amazon Web Services, Inc. https://aws.amazon.com/

[9] Amazon Lambda - AWS. (n.d.). Amazon Web Services, Inc. https://aws.amazon.com/pt/lambda/

[10] Introduction to Amazon ECS (1:37). (n.d.). Amazon Web Services, Inc. https://aws.amazon.com/pt/ecs/

[11] Armazenamento S3 - Simple Storage Service - Amazon Web Services. (n.d.). Amazon Web Services, Inc. https://aws.amazon.com/pt/s3/

[12] The C4 model for visualising software architecture. (n.d.). https://c4model.com/

[13] Hashicorp Terraform commands 9.84% market share in IT Management Software. (n.d.). https://enlyft.com/tech/products/hashicorp-terraform

[14] Perfis do IAM - AWS Identity and Access Management. (n.d.). https://docs.aws.amazon.com/pt_br/IAM/latest/UserGuide/id_roles.html

[15] Men, L. (2022). Site Reliability Engineering. Amazon Digital Services LLC - Kdp.