

UNIVERSIDADE NOVE DE JULHO – UNINOVE

Tobias Gomes Santos RA: 914116275

José Nilton de Oliveira Silva RA: 914112464

**SISTEMA WEB DE GESTÃO DE USUÁRIOS DO SAMBA – GO-
SAMBA4**

SISTEMA WEB DE GESTÃO DE USUÁRIOS DO SOFTWARE SAMBA

SÃO PAULO

2018

Tobias Gomes Santos RA: 914116276
José Nilton de Oliveira Silva RA: 914112464

**SISTEMA WEB DE GESTÃO DE USUÁRIOS DO SAMBA – GO-
SAMBA4**

SISTEMA WEB DE GESTÃO DE USUÁRIOS DO SOFTWARE SAMBA

Projeto de Pesquisa apresentado ao Curso de Bacharelado em Ciência da Computação da Universidade Nove de Julho como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Linha de Pesquisa: Modelagem Computacional

Orientador: Prof. Dr. Leandro Zerbinatti

SÃO PAULO

2018

LISTA DE ILUSTRAÇÕES

Figura 1 – Lista da pasta app do projeto.....	4
Figura 2 – Lista da pasta backup do projeto antigo em Golang.....	4
Figura 3 - Lista de arquivos da pasta contribute.....	5
Figura 4 – Lista de arquivos da pasta dist.....	5
Figura 5 – Lista de arquivos da pasta docs, screens,ssl.....	6
Figura 6 – Instalação e compilação do Samba4 no servidor.....	6
Figura 7 – Instalando o framework Flask.....	6
Figura 8 – Tela inicial AdminLTE.....	25
Figura 9 – Tela de login.....	26
Figura 10 – Tela após login.....	26
Figura 11 – Tela de usuários cadastrados.....	26
Figura 12 – Tela cadastro de usuários.....	27
Figura 13 – Tela de grupos cadastrados.....	28
Figura 14 – Tela de cadastro de grupos.....	29

```
├── app
│   ├── __init__.py
│   ├── model
│   │   ├── auth
│   │   ├── __init__.py
│   │   └── users.py
│   ├── static
│   │   ├── bower_components
│   │   ├── dist
│   │   ├── images
│   │   ├── lib
│   │   ├── plugins
│   │   └── stylesheets
│   ├── templates
│   │   ├── 403.html
│   │   ├── 404.html
│   │   ├── 500.html
│   │   ├── 503.html
│   │   ├── base.html
│   │   ├── default
│   │   ├── faq.html
│   │   ├── groups
│   │   ├── help.html
│   │   ├── layout.html
│   │   ├── media.html
│   │   ├── privacy-policy.html
│   │   ├── reset-password.html
│   │   ├── terms-and-conditions.html
│   │   └── users
│   └── views
│       ├── default.py
│       ├── groups.py
│       ├── __init__.py
│       └── users.py
```

```
├── backup_beego
│   ├── conf
│   │   └── app.conf
│   ├── controllers
│   │   ├── default.go
│   │   └── login.go
│   ├── go-samba4
│   ├── libs
│   │   └── auth_user.go
│   ├── main.go
│   ├── routers
│   │   └── router.go
│   ├── static
│   │   ├── images
│   │   ├── lib
│   │   └── stylesheets
│   ├── tests
│   │   └── default_test.go
│   └── views
│       ├── 403.html
│       ├── 404.html
│       ├── 500.html
│       ├── 503.html
│       ├── faq.html
│       ├── help.html
│       ├── index.html
│       ├── login.html
│       ├── media.html
│       ├── privacy-policy.html
│       ├── reset-password.html
│       ├── terms-and-conditions.html
│       ├── user.html
│       └── users.html
```

```
├── contribute
│   ├── AdminLTE
│   │   ├── bower_components
│   │   ├── bower.json
│   │   ├── changelog.md
│   │   ├── composer.json
│   │   ├── dist
│   │   ├── documentation
│   │   ├── Gruntfile.js
│   │   ├── index2.html
│   │   ├── index.html
│   │   ├── LICENSE
│   │   ├── package.json
│   │   ├── package-lock.json
│   │   ├── pages
│   │   ├── plugins
│   │   ├── README.md
│   │   ├── starter.html
│   │   └── yarn.lock
│   └── README.md
```

```
├── dist
│   ├── go_samba4
│   ├── ssl
│   │   ├── server.crt
│   │   └── server.key
│   ├── static
│   │   ├── bower_components
│   │   ├── dist
│   │   ├── images
│   │   ├── lib
│   │   ├── plugins
│   │   └── stylesheets
│   └── templates
│       ├── 403.html
│       ├── 404.html
│       ├── 500.html
│       ├── 503.html
│       ├── base.html
│       ├── default
│       ├── faq.html
│       ├── groups
│       ├── help.html
│       ├── layout.html
│       ├── media.html
│       ├── privacy-policy.html
│       ├── reset-password.html
│       ├── terms-and-conditions.html
│       └── users
```

```

├── docs
│   └── TCC
│       ├── ARTIGO_Joelson.pdf
│       ├── Backup
│       ├── Exemplo_TCC-1_Projeto-Grupo_A.pdf
│       ├── Exemplo_TCC-2_Concluido-Grupo_A.pdf
│       ├── links_samba.txt
│       ├── Projeto_TCC_final.doc
│       ├── TCC_DESEN_WEB_Exemplo.pdf
│       ├── Template_Professor_TCC_02-05.doc
│       ├── Template_Professor_TCC_24-03.doc
│       ├── Template_Projeto_TCCI_1.doc
│       └── TPL_TCCI.doc
├── go_samba4.py
├── LICENSE
├── list_dir.txt
├── make_bin.py
├── README.md
├── screens
│   ├── add_group.png
│   ├── add_user.png
│   ├── AdminLTE.png
│   ├── dashboard.png
│   ├── grupos.png
│   ├── login.png
│   └── users.png
├── ssl
│   ├── server.crt
│   └── server.key
└── tree_.txt

```

```

## Instalando, Compilando e Configurando o Samba4 4.8.2

apt-get install -y libreoffice-dev git build-essential libattr1-dev libblkid-dev libpam0g-dev
apt-get install -y autoconf python-dev python-dnspython libacl1-dev gdb pkg-config libpopt-dev libldap2-dev
apt-get install -y dnstools acl attr libbsd-dev docbook-xsl libcups2-dev libgnutls28-dev

cd /usr/src
get_samba4=https://download.samba.org/pub/samba/stable/samba-4.8.2.tar.gz
wget -c $(get_samba4)
tar xvfz $(basename $(get_samba4))
cd $(basename $(get_samba4))sed 's/,tar.gz//)'
./configure --with-ads --with-shared-modules=idmap_ad --enable-debug --enable-selftest --with-systemd --prefix=/opt/samba4
make
make install

echo '[Unit]
Description=Samba4 AD Daemon
After=syslog.target network.target

[Service]
Type=forking
PIDFile=/opt/samba4/var/run/samba.pid
LimitNOFILE=16384
EnvironmentFile=/etc/sysconfig/samba4
ExecStart=/opt/samba4/sbin/samba $SAMBAOPTIONS
ExecReload=/usr/bin/kill -HUP $MAINPID

[Install]
WantedBy=multi-user.target' > /etc/systemd/system/samba4.service

### Add PATH
echo 'export PATH=/opt/samba4/bin:/opt/samba4/sbin' >> /etc/profile
source /etc/profile

### Create Domain Samba4 like AD
# samba-tool domain provision --server-role=dc --use-rfc2307 --dns-backend=SAMBA_INTERNAL --realm=LINUXPRO.NET --domain=LINUXPRO --adminpass=Linuxpro123456
# or
# samba-tool domain provision --server-role=dc --use-rfc2307 --function-level=2008_R2 --use-xattrs=yes --dns-backend=SAMBA_INTERNAL --realm=LINUXPRO.NET --domain=LINUXPRO --adminpass=Linuxpro123456

### Add start script on boot
# systemctl daemon-reload
# systemctl enable samba4.service
# systemctl start samba4.service

```

```

## Instalando o Framework Flask

apt-get install -y python-pip git-core dos2unix
apt-get install -y libsasl2-dev python-dev libldap2-dev libssl-dev
pip install flask pyinstaller event psutil python-ldap Flask-Caching

## Clonando o go-samba4 do GitHub

cd /opt/
git clone https://github.com/jniltinho/go-samba4.git

```

LISTA DE ABREVIATURAS E SIGLAS

Python	Python é uma linguagem de programação de alto nível, interpretada, de script, imperativa, orientada a objetos, funcional, de tipagem dinâmica e forte.
Samba	Samba é o pacote padrão de interoperabilidade do Windows para Linux e Unix.
Flask	Web structure written in Python and based on the WSGI library
WEB	World Wide Web.
Git	Sistema de controle de versão distribuído.
Github	Plataforma de hospedagem de código para controle de versão e colaboração.
BSD	Berkeley Software Distribution.
AD	Active Directory é uma implementação de serviço de diretório no protocolo LDAP.

SUMÁRIO

LISTA DE ILUSTRAÇÕES

LISTA DE ABREVIATURAS E SIGLAS.....	4
1 INTRODUÇÃO	7
1.1 DELIMITAÇÃO DO TEMA.....	7
1.2 PROBLEMA	7
1.3 HIPÓTESE(S)	7
1.4 OBJETIVOS	8
1.4.1 <i>Objetivos gerais</i>	8
1.4.2 <i>Objetivos específicos</i>	9
1.5 JUSTIFICATIVA	9
2 REVISÃO DA LITERATURA	10
2.1 PROJETO.....	11
3 METODOLOGIA	11
3.1 SOFTWARE UTILIZADOS.....	12
3.1.1 <i>Sistemas</i>	12
3.1.2 <i>Aplicação Web</i>	12
3.1.3 <i>Flask (Framework web)</i>	13
3.1.4 <i>AdminLTE</i>	14
3.1.5 <i>Bootstrap</i>	14
3.1.6 <i>JQuery</i>	15
3.1.7 <i>PyInstaller</i>	15
3.1.8 <i>Samba</i>	16
3.1.9 <i>Samba-tool</i>	16
3.1.10 <i>Debian</i>	16
3.1.11 <i>Oracle VM VirtualBox</i>	17
3.2 PROCOLOS.....	18
3.2.1 <i>LDPA (Lightweight Directory Access Protocol)</i>	17

3.2.2 <i>Active Directory</i>	18
3.2.3 <i>SMB/CIFS</i>	18
3.3 COMPILAÇÃO E INSTALAÇÃO DO SAMBA 4.8.2.....	19
3.4 INSTALAÇÃO DO GO-SAMBA.....	22
4 RESULTADO	24
4.1 SISTEMA PROPOSTO.....	24
4.2 TELAS DO SISTEMA.....	25
5 ANÁLISE DOS RESULTADOS	30
6 CONCLUSÃO	31
REFERÊNCIAS BIBLIOGRÁFICAS.....	32
FOLHA DE APROVAÇÃO DO PROJETO.....	33
ANEXOS.....	34

1 INTRODUÇÃO

1.1 DELIMITAÇÃO DO TEMA

Até a versão 3 do software Samba, existia uma interface web chamada SWAT para administrá-lo, com a chegada da nova versão 4.0 em dezembro de 2012, essa interface web de administração foi descontinuada para o Samba 4.0 e surgiu uma nova interface chamada SWAT2 que está sendo desenvolvida em Python, mas ainda não foi finalizada e não possui um passo a passo simples e funcional de como instalar e integrar essa ferramenta em um Servidor com o serviço do Samba 4 ativo.

Foram realizadas várias pesquisas na internet e não foi encontrada nenhuma interface de administração web de fácil instalação para gerenciar o Samba, nem ao menos algo simples que possa criar, editar e excluir usuários no Samba.

Podemos administrar o Samba pela ferramenta gráfica RSAT disponível no ambiente Microsoft Windows, porém isso implica em ter uma estação desktop Windows a partir da versão 7 e precisa ser um Windows que consiga integrar no ambiente Active Directory do Samba 4.

1.2 PROBLEMA

O projeto Go-Samba4 tem como solução a gestão de usuários e grupos no ambiente server do sistema Samba 4, com uma interface web simples, leve e intuitiva, desenvolvida para promover um ambiente controlado onde o administrador possa gerenciar um sistema que antes só podia ser gerenciado por linha de comando usando o samba-tool no ambiente Linux ou por uma interface Desktop Windows.

1.3 HIPÓTESE

Esse trabalho apresenta um projeto que vai criar uma solução web desenvolvida na linguagem Python usando o framework Flask para gestão de usuários no software Samba 4, preocupando-se em criar algo que seja simples e intuitivo, de fácil execução em um servidor com o software Samba na versão 4.7.0 ou superior instalado.

O intuito é que o projeto seja open source (aberto), com seu código disponível no GitHub no endereço <https://github.com/jniltinho/go-samba4>, possibilitando que outros desenvolvedores possam baixar, usar e contribuir de alguma forma na melhoria e crescimento do projeto.

1.4 OBJETIVOS

Criar uma interface fácil e útil para que o usuário final tenha uma maior compreensão na utilização do sistema proposto. Temos como objetivo facilitar o usuário desde a instalação até a configuração e administração para criar, excluir, alterar, bloquear usuário e grupos no servidor samba 4.

1.4.1 Objetivos gerais

Neste trabalho, apresenta-se um projeto de uma aplicação web que pode ser acessado a partir de qualquer navegador, seu desenvolvimento será baseado na linguagem Python usando o framework Flask, para o gerenciamento de usuários, grupos e seus respectivos acessos.

1.4.2 Objetivos específicos

- Levantar informações de usuários e grupos.
- Desenvolver as páginas e formulários web em html, css e jquery.
- Desenvolver os Layout e formulários do sistema.
- Cadastrar informações.

1.5 JUSTIFICATIVA

A necessidade do desenvolvimento do sistema foi identificada quando não havia uma solução para gerenciamento e abstração do Samba 4 e sua interface de linha de comando chamada samba-tool. O sistema proposto será desenvolvido em Python onde todo o desenvolvimento será open source, assim permitindo que outros possam interagir, melhorar e evoluir o projeto.

A ideia é criar um sistema executável que possua uma interface web auxiliando o usuário na execução das suas tarefas sem haver necessidade de conhecimento avançados em Linux e na linha de comando do samba-tool.

A utilização desse software para pequenas e médias empresas é recomendada, pois ajuda e facilita a administração na gestão de usuários e grupos no Samba de maneira muito rápida e simples.

2 REVISÃO DA LITERATURA

Este capítulo apresenta um projeto baseado no Samba 4.8.2 para administração remota de servidores Windows.

2.1. Projeto

Depois de pesquisar bastante foi constatado que não havia um sistema com uma interface de administração web de fácil instalação, para o gerenciamento do Samba 4 já desenvolvida.

Com um tema e ideia já definidos do projeto, foi dado início ao desenvolvimento, onde o objetivo é criar uma solução web desenvolvida em Python utilizando o framework Flask para a gestão de usuários no software Samba 4 podendo ser acessada em qualquer navegador.

Um dos nossos critérios para esse projeto é que ele seja de fácil entendimento para os usuários, mesmo aqueles que não tenham um conhecimento avançado na linha de comando Linux. O projeto trará uma interface de fácil compreensão para o usuário desde a instalação até a configuração do sistema.

3 METODOLOGIA

Este capítulo apresenta uma descrição dos softwares e sistemas utilizados para o desenvolvimento do projeto proposto, bem como dos processos envolvidos.

3.1 Software Utilizados

O sistema será desenvolvido com a linguagem Python utilizando o Framework Flask e o Samba 4.8.2, para o acesso do administrador no sistema web, utilizamos a autenticação no Samba, sendo que o usuário que efetuar o login não tiver perfil de administrador no Samba, não poderá administrar no sistema web.

3.1.1 Sistemas

Python

Python é uma linguagem de programação de alto nível, interpretada, de script, imperativa, orientada a objetos, funcional, de tipagem dinâmica e forte. Foi lançada por Guido Van Rossum em 1991. Atualmente possui um modelo de desenvolvimento comunitário, aberto e gerenciado pela organização sem fins lucrativos Python Software Foundation. Apesar de várias partes da linguagem possuírem padrões e especificações formais, a linguagem como um todo não é formalmente especificada. O padrão de facto é a implementação CPython.

A linguagem foi projetada com a filosofia de enfatizar a importância do esforço do programador sobre o esforço computacional. Prioriza a legibilidade do código sobre a velocidade ou expressividade. Combina uma sintaxe concisa e clara com os recursos poderosos de sua biblioteca padrão e por módulos e frameworks desenvolvidos por terceiros.

Python é uma linguagem de propósito geral de alto nível, multi paradigma, suporta o paradigma orientado a objetos, imperativo, funcional e procedural. Possui

tipagem dinâmica e uma de suas principais características é permitir a fácil leitura do código e exigir poucas linhas de código se comparado ao mesmo programa em outras linguagens. Devido às suas características, ela é principalmente utilizada para processamento de textos, dados científicos e criação de CGIs para páginas dinâmicas para a web. Foi considerada pelo público a 3ª linguagem "mais amada", de acordo com uma pesquisa conduzida pelo site Stack Overflow em 2018, e está entre as 5 linguagens mais populares, de acordo com uma pesquisa conduzida pela RedMonk.

O nome Python teve a sua origem no grupo humorístico britânico Monty Python, criador do programa Monty Python's Flying Circus, embora muitas pessoas façam associação com o réptil do mesmo nome (em português, píton ou pitão).

3.1.2 Aplicação Web

Em computação, aplicação web designa, de forma geral, sistemas de informática projetados para utilização por meio de um navegador, através da internet ou aplicativos desenvolvidos utilizando tecnologias web HTML, JavaScript e CSS. Pode ser executado a partir de um servidor HTTP (Web Host) ou localmente, no dispositivo do usuário.

Uma aplicação web também é definida em tudo que se é processado em algum servidor, exemplo: quando você entra em um e-commerce a página acessada antes de vir até seu navegador é processada em um computador ligado a internet que retorna o processamento das regras de negócio nele contido. Por isso se chama aplicação e não simplesmente site web.

A função do servidor web é receber uma solicitação (requisição) e devolver (resposta) algo para o cliente. O browser permite ao usuário solicitar um recurso e quando o servidor responde a uma solicitação são encontrados recursos como: páginas HTML, figuras e documento PDF que são exibidas depois para o usuário. Geralmente os servidores enviam instruções para o browser escritas em HTML. O HTML diz ao browser como apresentar conteúdo ao usuário web.

O servidor em si tem alguns recursos, mas por algumas deficiências não consegue processar tudo sozinho como: criações de páginas dinâmicas e o armazenamento de dados em um banco de dados.

Páginas dinâmicas – Quando a aplicação é executada no servidor, este disponibiliza somente páginas estáticas. Porém, para efetuar essa comunicação é necessário o auxílio de uma outra aplicação de ajuda que é passada através de Servlet.

Armazenar dados no servidor – Para efetuar essa ação o servidor precisa de uma aplicação de apoio (Servlet), fazendo com que o servidor envie esses parâmetros para o Servlet.

As falhas de segurança podem surgir em diferentes etapas, tais como: análise de requisito; especificação; implementação. Os riscos de aplicação na vulnerabilidade de uma empresa podem causar impactos.

O HTTP usa um modelo de solicitações e respostas. Uma solicitação ocorre quando o usuário faz uma solicitação HTTP e o servidor web devolve uma resposta HTTP, sendo que o browser verifica como tratar esse conteúdo. Se a resposta que vem do servidor for uma página HTML, então é inserido na resposta HTTP.

As diferenças entre as solicitações GET e POST são que enquanto o GET anexa dados do formulário no final da URL o POST inclui dados do formulário no corpo da solicitação.

3.1.3 Flask (Framework web)

É um framework web escrito em Python e baseado na biblioteca WSGI (Web Server Gateway Interface) “em português Interface de Porta de Entrada do Servidor Web: é uma especificação para uma interface simples e universal entre servidores web e aplicações web ou frameworks para a linguagem de programação Python.” e na biblioteca de Jinja2. Flask está disponível sob os termos da licença BSD.

Flask tem a flexibilidade da linguagem de programação Python e provê um modelo simples para desenvolvimento web. Uma vez importando no Python, Flask pode ser usado para economizar tempo construindo aplicações web. Um exemplo

de aplicação desenvolvida com Flask é a página da comunidade de desenvolvedores do framework.

É chamado de micro framework porque mantêm um núcleo simples, mas extensível. Não há uma camada de abstração do banco de dados, validação de formulários, ou qualquer outro componente onde bibliotecas de terceiros existem para prover a funcionalidade. Assim, Flask suporta extensões capazes de adicionar tais funcionalidades na aplicação final. Há uma vasta coleção de bibliotecas para resolver essas questões em Python, isso simplifica o framework e torna sua curva de aprendizado mais suave.

3.1.4 AdminLTE

É um template admin web html totalmente responsivo, baseado no framework Bootstrap 3. Altamente personalizável e fácil de usar, adapta-se a muitas resoluções de tela, de pequenos dispositivos móveis a grandes telas de Desktop.

3.1.5 Bootstrap

Bootstrap é um framework web com código-fonte aberto para desenvolvimento de componentes de interface e front-end para sites e aplicações web usando HTML, CSS e JavaScript, baseado em modelos de design para a tipografia, melhorando a experiência do usuário em um site amigável e responsivo.

O Bootstrap é um dos projetos mais bem avaliados no site GitHub, com mais de 111.600 estrelas e 51.500 forks.

3.1.6 jQuery

jQuery é uma biblioteca de funções JavaScript que interage com o HTML, desenvolvida para simplificar os scripts interpretados no navegador do cliente (client-side). Lançada em dezembro de 2006 no BarCamp, de Nova York, por John Resig. Usada por cerca de 77% dos 10 mil sites mais visitados do mundo, jQuery é a mais popular das bibliotecas JavaScript.

jQuery é uma biblioteca de código aberto e possui licença dual, fazendo uso da Licença MIT ou da GNU General Public License versão 2. A sintaxe do jQuery foi desenvolvida para tornar mais simples a navegação do documento HTML, a seleção de elementos DOM, criar animações, manipular eventos, desenvolver aplicações AJAX e criação de plugins sobre ela. Tais facilidades permitem aos desenvolvedores criarem camadas de abstração para interações de baixo nível de modo simplificado em aplicações web dinâmicas de grande complexidade.

3.1.7 PyInstaller

PyInstaller reúne um aplicativo Python e todas as suas dependências em um único pacote. O usuário pode executar o aplicativo empacotado sem instalar um interpretador Python ou qualquer módulo. O PyInstaller suporta Python 2.7 e Python 3.3+ e agrupa corretamente os principais pacotes do Python, como numpy, PyQt, Django, wxPython e outros.

O PyInstaller é testado no Windows, Mac OS X e Linux. No entanto, não é um compilador cruzado: para fazer um aplicativo do Windows, você executa o PyInstaller no Windows; para fazer um aplicativo Linux, você deve executar no Linux, etc.

3.1.8 Samba

Samba é uma reimplementação de software livre do protocolo de rede SMB/CIFS e foi originalmente desenvolvido por Andrew Tridgell. Ele fornece serviços de arquivo e impressão para vários clientes Microsoft Windows e pode se integrar a um domínio do Microsoft Windows Server, como um Controlador de Domínio (DC) ou como um membro do domínio. A partir da versão 4, ele suporta domínios do Active Directory e do Microsoft Windows NT.

Executado na maioria dos sistemas Unix, OpenVMS e Unix, como Linux, Solaris, AIX e variantes do BSD, incluindo macOS Server da Apple e o cliente macOS (Mac OS X 10.2 e superior). Samba é padrão em quase todas as distribuições do Linux e é comumente incluído como um serviço básico do sistema em outros sistemas operacionais baseados em Unix. Samba é liberado sob os termos da Licença Pública Geral GNU. O nome Samba vem do SMB (Server Message Block), o nome do protocolo padrão usado pelo sistema de arquivos de rede do Microsoft Windows.

3.1.9 Samba-tool

É a ferramenta principal de linha de comando para administração do Samba, que foi desenvolvida em Python 2.7 com seus módulos Python que estão na pasta `lib/python2.7/site-packages` ou `lib64/python2.7/site-packages` do projeto Samba após a sua compilação e instalação no servidor, os módulos fazem um bind com as bibliotecas em C na qual é desenvolvida o projeto Samba.

3.1.10 Debian

O Debian é uma organização exclusivamente de voluntários dedicada ao desenvolvimento de software livre e a promover os ideais da comunidade de Software Livre. O Debian Project começou em 1993, quando Ian Murdock lançou um

convite aberto a criadores de software para contribuírem para uma distribuição de software completa e coerente baseada no relativamente novo kernel Linux. Esse relativamente pequeno grupo de dedicados entusiastas, originalmente com fundos da Free Software Foundation e influenciados pela filosofia GNU, cresceu com o passar dos anos para uma organização com cerca de 1062 Debian Developers.

Os Debian Developers estão envolvidos numa série de atividades, incluindo a administração do site WEB e do arquivo FTP, design gráfico, análise legal de licenças de software, escrever documentação, e, claro, manter pacotes de software.

3.1.11 Oracle VM VirtualBox

Oracle VM VirtualBox é um software de virtualização desenvolvido pela empresa Innotek depois comprado pela Sun Microsystems que posteriormente foi comprada pela Oracle que, como o VMware Workstation, visa criar ambientes para instalação de sistemas distintos. Ele permite a instalação e utilização de um sistema operacional dentro de outro, assim como seus respectivos softwares, como dois ou mais computadores independentes, mas compartilhando fisicamente o mesmo hardware, geralmente usado por desenvolvedores de sistemas, ou profissionais de TI.

3.2 Protocolos:

3.2.1 LDAP (Lightweight Directory Access Protocol)

LDAP é um protocolo de rede que roda sobre o TCP/IP que permite organizar os recursos de rede de forma hierárquica, como uma árvore de diretório, onde temos primeiramente o diretório raiz, em seguida a rede da empresa, o departamento e por fim o computador do funcionário e os recursos de rede (arquivos, impressoras, etc.) compartilhados por ele. A árvore de diretório pode ser criada de acordo com a necessidade.

O LDAP oferece uma grande escalabilidade. É possível replicar servidores (para backup ou balanceamento de carga) e incluir novos servidores de uma forma hierárquica, interligando departamentos e filiais de uma grande multinacional por exemplo. A organização dos servidores neste caso é similar ao DNS: é especificado um servidor raiz e a partir daí é possível ter vários níveis de sub-servidores, além de mirrors do servidor principal.

3.2.2 Active Directory

O Active Directory (AD) é um serviço de diretório baseado no protocolo LDAP, nativo dos sistemas Windows Server oferecido desde a versão Windows Server 2000. O AD armazena informações de usuários, impressoras, servidores, grupos de usuários, computadores e políticas de segurança.

O AD pode ser utilizado tanto em redes de pequenas organizações, como em grandes corporações. Uma rede onde o AD está instalado pode conter um ou mais domínios. Com o uso do AD, um usuário precisa ser cadastrado em apenas um dos domínios, podendo receber permissões para usar recursos em qualquer um dos domínios. A compatibilidade entre os domínios é possível graças a relação de confiança entre eles.

3.2.3 SMB/CIFS

O SMB/CIFS (Server Message Block/Common Internet File System) é um protocolo de redes cujo o uso mais comum como foi dito anteriormente é o compartilhamento de arquivos em uma LAN. Este protocolo permite que o cliente manipule arquivos como se estes estivessem em sua máquina local. Operações como leitura, escrita, criação, apagamento e renomeação são suportadas, sendo a única diferença de que os arquivos manipulados não estão no computador local e sim em um servidor remoto. O protocolo SMB/CIFS funciona enviando pacotes do cliente para o servidor.

Cada pacote é tipicamente baseado em uma requisição de algum tipo, como a abertura ou leitura de um arquivo. O servidor então recebe este pacote, verifica-o para ver se a requisição é válida, ou seja, verifica se o cliente possui as permissões apropriadas para efetuar a requisição e finalmente executa a requisição e retorna um pacote de resposta ao cliente. O cliente então analisa o pacote de resposta para determinar se a requisição inicial foi completada com sucesso.

O SMB/CIFS é razoavelmente um protocolo de rede de alto nível. No modelo OSI ele é provavelmente melhor descrito como uma camada de Aplicação/Apresentação. Isto significa que o SMB/CIFS depende de outros protocolos para o transporte. O protocolo mais utilizado para transporte confiável é o NetBIOS sobre TCP.

Apesar do compartilhamento de arquivos ser a principal proposta do SMB/CIFS existem outras funções associadas a ele. A maioria das implementações de SMB/CIFS são capazes de determinar outros servidores SMB/CIFS na rede (browsing), compartilhar impressoras e até mesmo fornecer complicadas técnicas de autenticação.

3.3 Compilação e Instalação do Samba 4.8.2

Para compilar e instalar o software samba na versão 4.8.2, você vai precisar baixar e instalar uma distribuição Linux atual de 64 bits, desenvolvemos o go-samba4 com a distribuição Debian 9 64 bits.

Você precisa instalar o Debian 9, modo servidor, apenas os pacotes básicos e de acesso sem nenhuma interface gráfica, após instalado, configurado e atualizado o servidor, execute os passos abaixo na sequência linha por linha.

Primeiramente efetuamos a instalação dos pacotes para compilador o software samba, usando o programa apt-get para isso, lembrando que todo o processo deve ser executado com o usuário root.

```
apt-get install -y libreadline-dev git build-essential libattr1-dev
```

```
apt-get install -y libblkid-dev libpam0g-dev
```

```
apt-get install -y autoconf python-dev python-dnspython libacl1-dev
```

```
apt-get install -y gdb pkg-config libpopt-dev libldap2-dev
```

```
apt-get install -y dnsutils acl attr libbsd-dev docbook-xsl libcups2-dev libgnutls28-dev
```

Após a instalação dos pacotes acima, iremos baixar o software samba na versão 4.8.2 diretos do ftp do projeto samba com o comando wget e logo em seguida efetuamos a compilação, lembrando que se houver um erro em qualquer um desses passos repita o procedimento que deu errado, só avance se não houver erro.

```
cd /usr/src
```

```
get_samba4=https://download.samba.org/pub/samba/stable/samba-4.8.2.tar.gz
```

```
wget -c ${get_samba4}
```

```
tar xvfz $(basename ${get_samba4})
```

```
cd $(basename ${get_samba4}|sed "s/.tar.gz//")
```

```
./configure --with-ads --with-shared-modules=idmap_ad --enable-debug --enable-selftest --with-systemd --prefix=/opt/samba4
```

```
make
```

```
make install
```

Após compilado e instalado o samba 4.8.2 precisamos criar um arquivo para iniciar o daemon do samba no sistema, para isso criaremos um arquivo chamado /etc/systemd/system/samba4.service com o conteúdo abaixo:

```
[Unit]
```

```
Description=Samba4 AD Daemon
```

```
After=syslog.target network.target
```

```
[Service]
```

```
Type=forking
```

```

PIDFile=/opt/samba4/var/run/samba.pid
LimitNOFILE=16384
EnvironmentFile=/etc/sysconfig/samba4
ExecStart=/opt/samba4/sbin/samba $SAMBAOPTIONS
ExecReload=/usr/bin/kill -HUP $MAINPID
[Install]
WantedBy=multi-user.target

```

Agora precisamos configurar provisionamento do domínio com a ferramenta de linha de comando chamada samba-tool, com o comando abaixo, lembrando que você precisa trocar o nome MYDOMAIN pelo seu domínio real.

```

/opt/samba4/bin/samba-tool domain provision --server-role=dc --use-rfc2307 --
function-level=2008_R2 --use-xattrs=yes --dns-backend=SAMBA_INTERNAL --
realm=MYDOMAIN.NET --domain=MYDOMAIN --adminpass=Mypass123456

```

Depois de configurado corretamente o provisionamento do domínio você precisa iniciar o daemon do samba para o serviço ficar ativo no servidor e o serviço subir quando reiniciar o servidor, para isso execute o procedimento abaixo:

```

systemctl daemon-reload
systemctl enable samba4.service
systemctl start samba4.service

```

Após efetuar os passos acima, quando você executar o comando `ps -aux` sua saída será essa na imagem abaixo:


```

nilton@linuxpro: ~
Arquivo Editar Abas Ajuda
root 422 0.0 0.2 69944 5352 ? Ss 11:08 0:00 /usr/sbin/sshd -D
root 426 0.0 2.0 536136 43032 ? Ss 11:08 0:00 samba: root process
root 487 0.0 1.2 536136 24696 ? S 11:08 0:00 samba: task[s3fs_parent]
root 488 0.0 1.9 542860 39408 ? S 11:08 0:12 samba: task[dcesrv]
root 489 0.0 1.3 536136 27680 ? S 11:08 0:00 samba: task[nbtd]
root 490 0.0 1.1 536136 24504 ? S 11:08 0:00 samba: task[lwrepl]
root 491 0.0 1.5 536136 31856 ? S 11:08 0:00 samba: task[ldapsrv]
root 492 0.0 1.2 536136 25000 ? S 11:08 0:00 samba: task[cldapd]
root 493 0.0 1.7 542432 35276 ? S 11:08 0:00 samba: task[kdc]
root 494 0.0 1.3 536136 28144 ? S 11:08 0:18 samba: task[dreplsrv]
root 495 0.0 1.2 536136 24632 ? S 11:08 0:00 samba: task[winbindd_parent]
root 496 0.0 1.1 536136 24416 ? S 11:08 0:00 samba: task[ntpd_signd]
root 497 0.0 1.7 536136 35812 ? S 11:08 0:00 samba: task[kccsrv]
root 498 0.0 1.3 536136 26752 ? S 11:08 0:01 samba: task[dnssupdate]
root 501 0.0 1.0 536140 22500 ? S 11:08 0:00 samba: tfork waiter process
root 502 0.0 1.0 536140 22500 ? S 11:08 0:00 samba: tfork waiter process
root 504 0.0 2.4 519720 50332 ? Ss 11:08 0:00 /opt/samba4/sbin/smbd -D --option=server role check:inhibit=y
root 505 0.0 2.1 453248 43728 ? Ss 11:08 0:00 /opt/samba4/sbin/winbindd -D --option=server role check:inhib
root 508 0.0 1.7 542852 35568 ? S 11:08 0:00 samba: task[dnss]
root 519 0.0 0.1 257684 2984 ? Sl 11:08 0:06 /usr/sbin/VBoxService --pidfile /var/run/vboxadd-service.sh
root 530 0.0 1.2 510876 24908 ? S 11:08 0:00 /opt/samba4/sbin/smbd -D --option=server role check:inhibit=y
root 531 0.0 1.1 510876 23332 ? S 11:08 0:00 /opt/samba4/sbin/smbd -D --option=server role check:inhibit=y
root 532 0.0 1.2 519720 25012 ? S 11:08 0:00 /opt/samba4/sbin/smbd -D --option=server role check:inhibit=y
root 544 0.0 0.0 20468 1808 ? Ss 11:08 0:00 /sbin/dhclient -4 -v -pf /run/dhclient.enp0s3.pid -lf /var/li
root 563 0.0 0.3 95204 6792 ? Rs 11:09 0:00 sshd: root@pts/0
root 565 0.0 0.2 56424 5860 ? Ss 11:10 0:00 /lib/systemd/systemd --user
root 566 0.0 0.0 82404 1520 ? S 11:10 0:00 (sd-pam)
root 572 0.0 0.2 21208 5084 pts/0 Ss 11:10 0:00 -bash
root 1095 0.0 0.0 0 0 ? S 16:04 0:05 [kworker/0:2]
root 1254 0.0 0.0 0 0 ? S 17:53 0:00 [kworker/0:1]
root 1268 0.0 0.0 0 0 ? S 17:58 0:00 [kworker/0:0]
root 1280 0.0 0.1 38304 3268 pts/0 R+ 18:01 0:00 ps -aux
root@ad-server:~#

```

Vai aparecer alguns processos no Linux com o nome de samba, se a imagem for parecida com essa a instalação e configuração foi efetuada com sucesso, podendo assim continuar com a instalação do projeto go-samba4.

3.4 Instalação do Go-Samba

```
apt-get install -y git-core
```

```
cd /opt/
```

```
git clone https://github.com/jniltonho/go-samba4.git
```

```
mv go-samba4/dist /opt/
```

```
rm -rf go-samba4 && mv dist go-samba4
```

```
chmod +x go-samba4/go_samba4 && cd /opt/go-samba4/
```

Após executar os comandos acima você precisa iniciar o go-samba4 criando o seu iniciador, crie um arquivo chamado /etc/systemd/system/go_samba4.service com o conteúdo abaixo:

[Unit]

Description=Go-Samba4 Daemon

After=syslog.target network.target

[Service]

WorkingDirectory=/opt/go-samba4

ExecStart=/opt/go-samba4/go_samba4 --server-prod

NonBlocking=true

[Install]

WantedBy=multi-user.target

Para iniciar execute os passos abaixo:

systemctl daemon-reload

systemctl enable go_samba4.service && systemctl start go_samba4.service

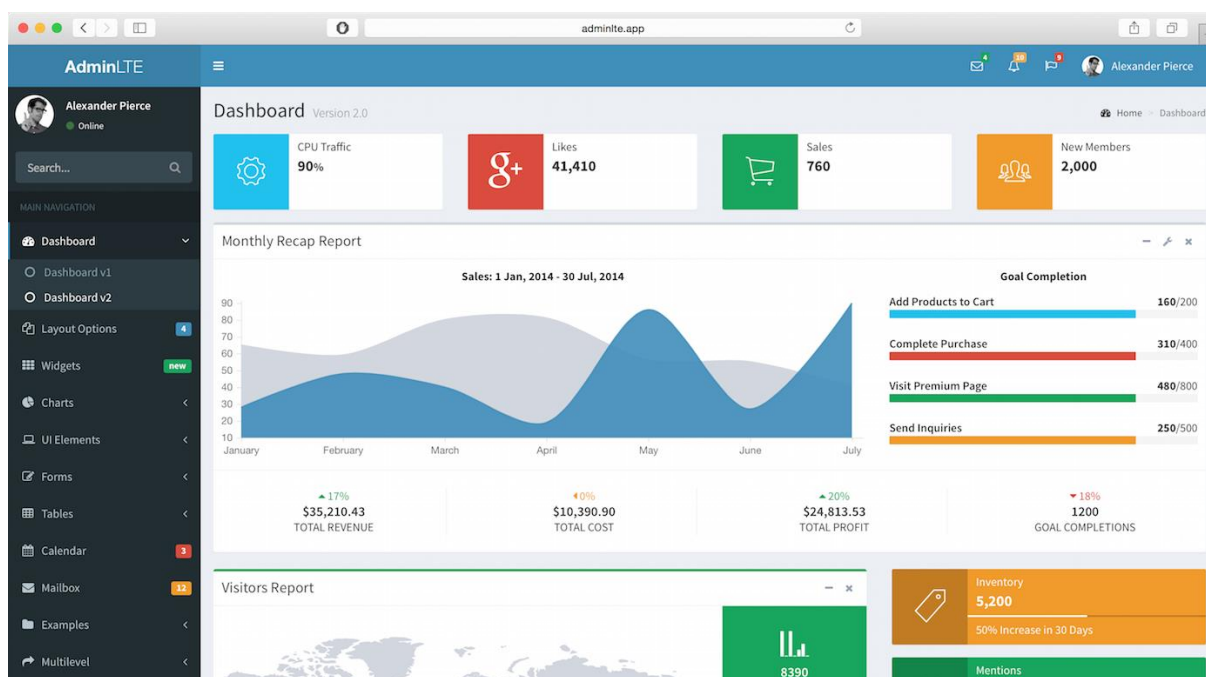
4 RESULTADOS

Este capítulo apresenta uma abordagem dos resultados que são esperados a partir do desenvolvimento do projeto.

4.1 Sistema proposto

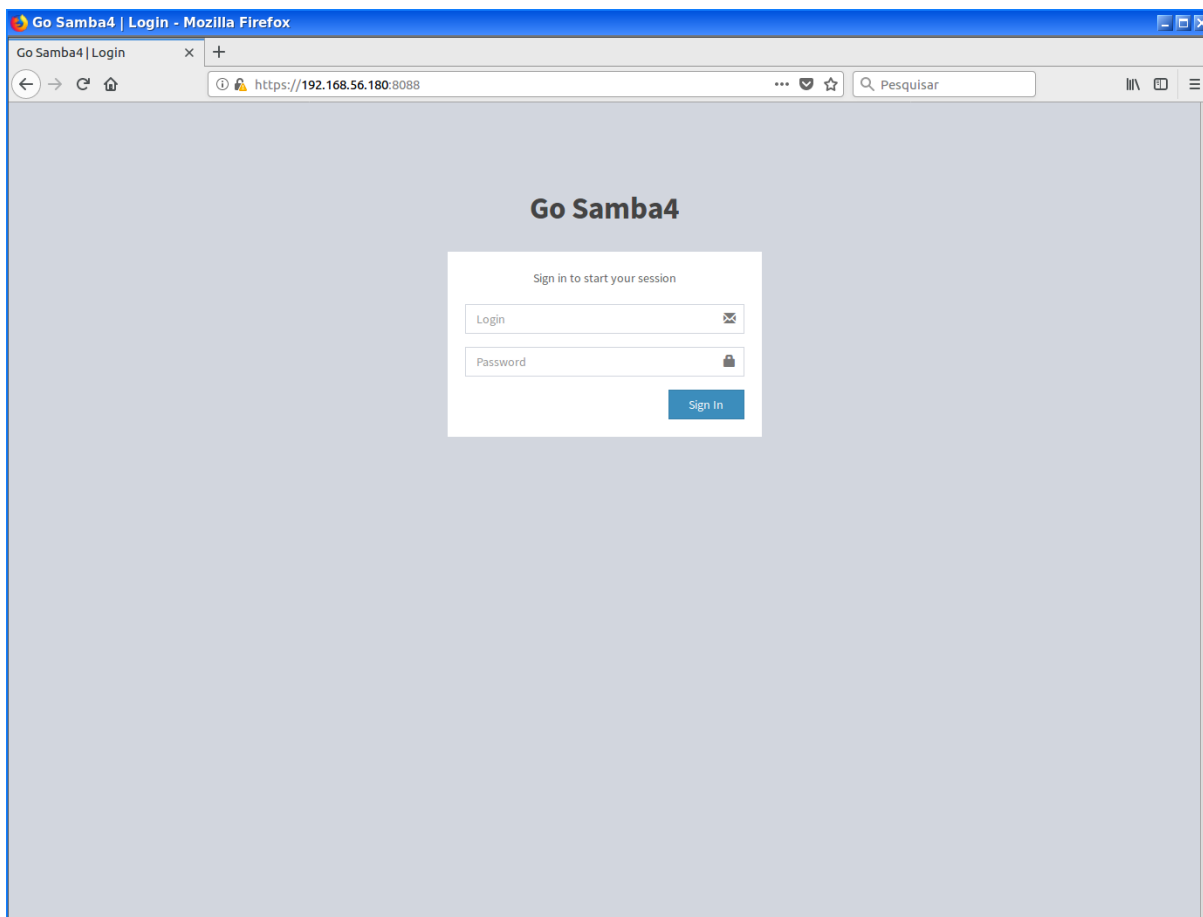
O projeto, previamente, concluído não foi de difícil desenvolvimento, pois tivemos como escolha utilizar sistemas baseados em Python que possui uma linguagem fácil e compatibilidade com outros softwares utilizado no desenvolvimento do projeto.

Com tudo pronto e configurado deu-se início ao desenvolvimento, houve complicações no desenvolvimento do layout onde pôr fim foi decidido que iríamos utilizar um pronto com isso o sistema já possui a tela de login, as regras de acesso dos devidos usuários. Foi utilizado um projeto de template free feito com Bootstrap chamado AdminLTE que está disponível no GITHUB no seguinte endereço <https://github.com/almasaeed2010/AdminLTE>.



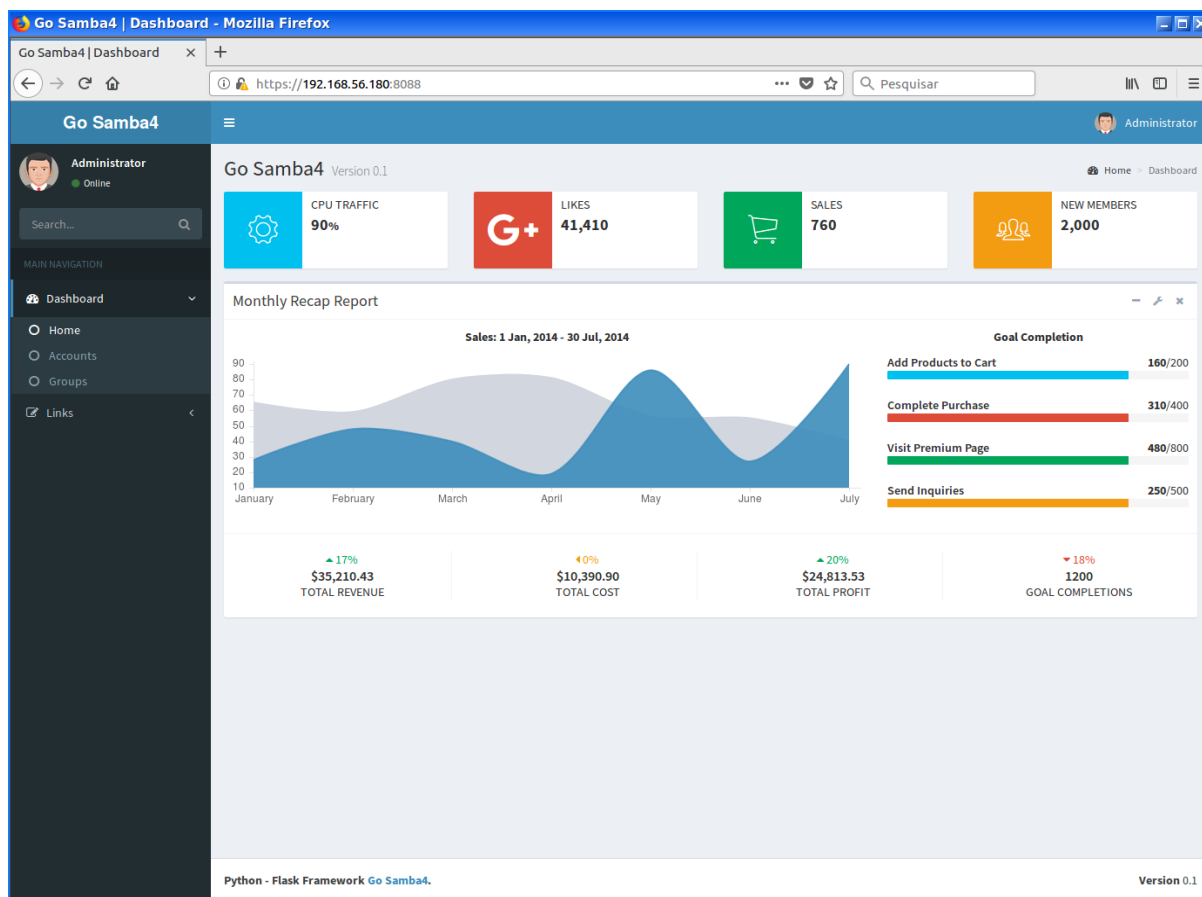
4.2 Telas do Sistema

Tela de login, nessa tela é necessário colocar o login e senha do usuário administrador do Samba4.



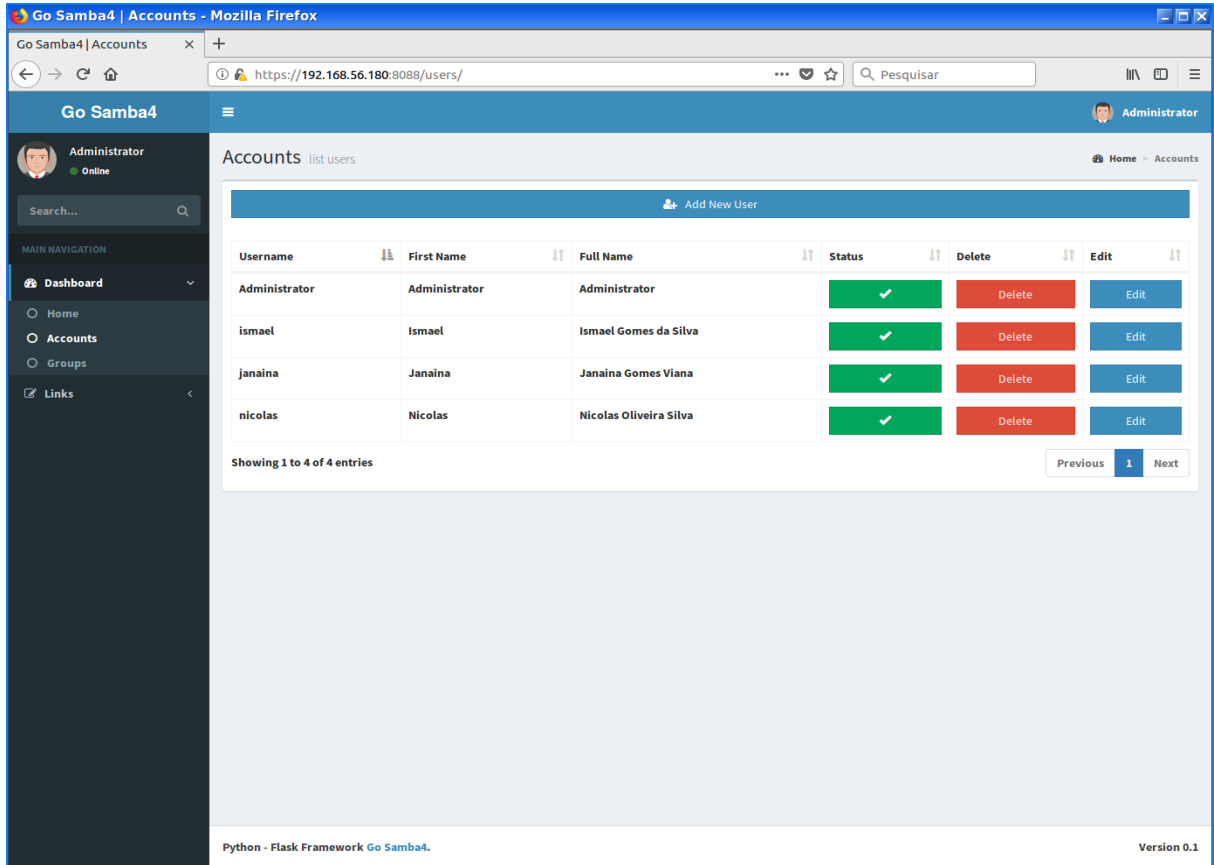
Tela de login

Após ter feito o login no sistema abre-se a tela inicial que ainda está em desenvolvimento. Nela será apresentado os gráficos e status do Servidor.



Tela inicial após login

Tela que lista os usuários cadastrado no samba4, nessa tela há a opção de adicionar e excluir usuários no samba4.



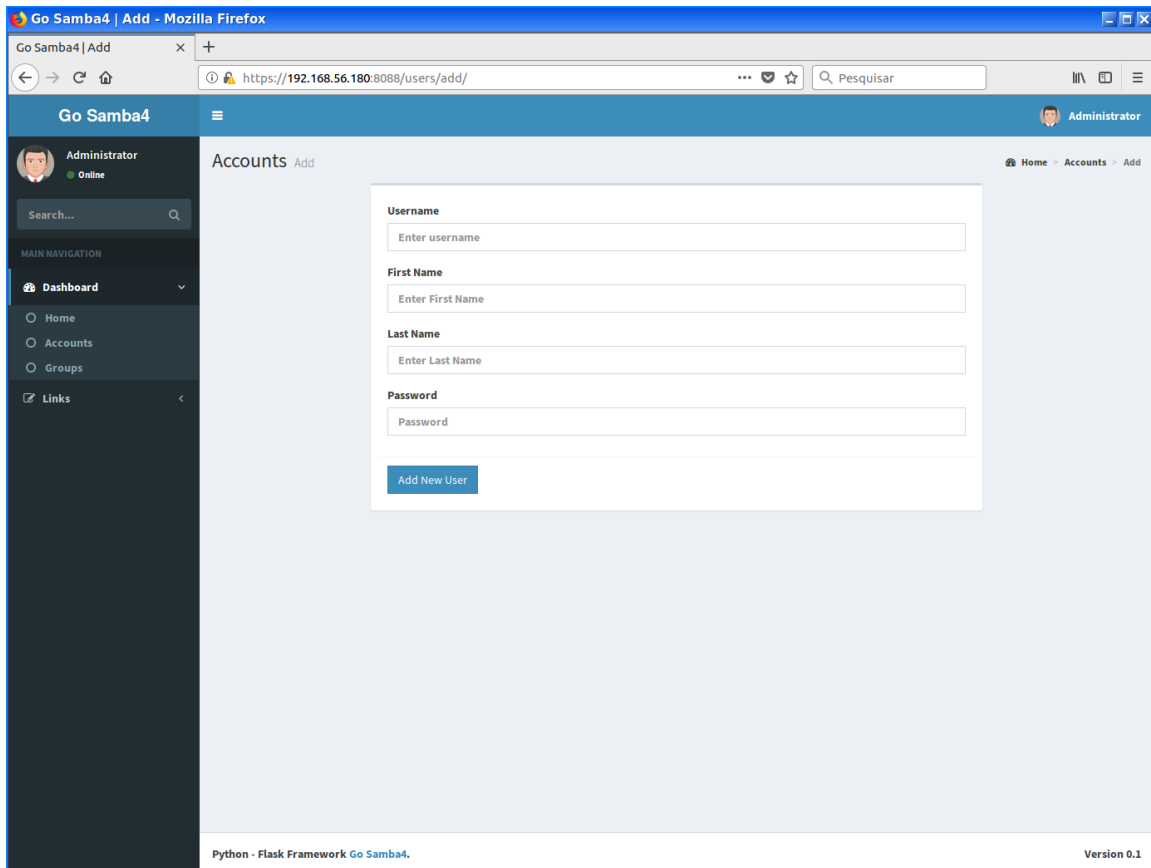
The screenshot shows the 'Go Samba4 | Accounts' page in a Mozilla Firefox browser. The page displays a list of users with the following data:

Username	First Name	Full Name	Status	Delete	Edit
Administrator	Administrator	Administrator	✓	Delete	Edit
ismael	Ismael	Ismael Gomes da Silva	✓	Delete	Edit
janaina	Janaina	Janaina Gomes Viana	✓	Delete	Edit
nicolas	Nicolas	Nicolas Oliveira Silva	✓	Delete	Edit

Showing 1 to 4 of 4 entries

Python - Flask Framework Go Samba4. Version 0.1

Tela de usuários cadastrados



The screenshot shows the 'Go Samba4 | Add' page in a Mozilla Firefox browser. The page displays a form for adding a new user with the following fields:

- Username: Enter username
- First Name: Enter First Name
- Last Name: Enter Last Name
- Password: Password

There is an 'Add New User' button at the bottom of the form.

Python - Flask Framework Go Samba4. Version 0.1

Tela de cadastro de usuário

Tela que lista os grupos cadastrado no samba4, nessa tela você tem a opção de adicionar e excluir grupos no samba4.

Go Samba4 | Groups - Mozilla Firefox

Go Samba4 | Groups x +

https://192.168.56.180:8088/groups/ Pesquisar

Go Samba4 Administrator Online

Search...

MAIN NAVIGATION

- Dashboard
- Home
- Accounts
- Groups
- Links

Groups list groups Home > Groups

Add New Group

ID	Group Name	Delete	Edit
#ID	Account Operators	Delete	Edit
#ID	Administrators	Delete	Edit
#ID	Allowed RODC Password Replication Group	Delete	Edit
#ID	Backup Operators	Delete	Edit
#ID	Cert Publishers	Delete	Edit

Showing 1 to 5 of 37 entries

Previous 1 2 3 4 5 ... 8 Next

Python - Flask Framework Go Samba4. Version 0.1

Tela de grupos cadastrados

The screenshot shows a web browser window titled "Go Samba4 | Add - Mozilla Firefox". The address bar displays the URL "https://192.168.56.180:8088/groups/add/". The page features a dark blue sidebar on the left with a user profile for "Administrator" (Online) and a "MAIN NAVIGATION" menu containing "Dashboard", "Home", "Accounts", "Groups", and "Links". The main content area is titled "Group Add" and contains a form with a "Groupname" label, a text input field with the placeholder "Enter groupname", and a blue "Add New Group" button. A breadcrumb trail at the top right shows "Home > Groups > Add". The footer includes the text "Python - Flask Framework Go Samba4." on the left and "Version 0.1" on the right.

Go Samba4 | Add - Mozilla Firefox

Go Samba4 | Add

https://192.168.56.180:8088/groups/add/

Pesquisar

Go Samba4

Administrator
Online

Search...

MAIN NAVIGATION

Dashboard

Home

Accounts

Groups

Links

Group Add

Home > Groups > Add

Groupname

Enter groupname

Add New Group

Python - Flask Framework Go Samba4.

Version 0.1

Tela para cadastrar grupo

5 ANÁLISE DOS RESULTADOS

Com o objetivo de fazer um sistema web intuitivo onde o usuário final não tenha a necessidade de digitar linhas de comando no software samba-tool, foi criada uma interface web onde todos os softwares utilizados iriam rodar por trás do sistema sem que o usuário saiba.

O desenvolvimento da aplicação web, utilizando o Samba na versão 4.8.2 e a ferramenta framework flask, deixou o gerenciamento gestão dos usuários e grupos ainda mais fáceis com a implementação do template AdminLTE que é totalmente responsivo. Visto que não havia nenhuma aplicação como essa na internet pronta ou em desenvolvimento até a publicação desse projeto. Vimos nessa linha de pesquisa uma oportunidade de desenvolver algo novo e que até então revolucionário.

Esse projeto está sendo reconhecido pelos usuários e desenvolvedores do Github tanto nacional como também internacionalmente. Toda linha de pesquisa atualizações do projeto, códigos fontes e documentação estão sendo hospedados no Github.

Por fim alguns benefícios que o projeto possui, é a utilização do PyInstaller que oferece a possibilidade de gerar um binário executável sem haver a necessidade de instalação de bibliotecas Python ou quaisquer bibliotecas que estão no projeto, facilitando até mesmo aqueles usuários com menos experiência na instalação de aplicações no Linux.

Pela parte Web buscamos utilizar ferramentas compatíveis com a linguagem Python onde tivemos uma ampla exploração no desenvolvimento, nos trazendo possibilidades nas quais ainda não havíamos pensado e futuramente aplicá-las formando assim um projeto complexo sem a necessidade de “gambiarras” para um sistema fluido.

6 CONCLUSÃO

O desenvolvimento do projeto nos proporcionou a oportunidade de conhecer um ambiente novo, onde foi aplicado os conhecimentos adquiridos durante o curso de Ciência da Computação, contribuindo para o aprimoramento das atividades futuras.

Os estudos realizados durante todo o projeto revelaram que o sistema proposto quando implementado agregará benefícios substanciais para a administração de rede. O uso do Active Directory, principal serviço disponibilizado pelo Samba funciona centralizando as informações de usuários, grupos, computadores, impressoras e políticas de acesso e segurança.

O framework Flask nos proporcionou um ágil desenvolvimento devido as propriedades de linguagem Python.

O sistema de template AdminLTE feito com bootstrap 3 facilitou a criação das telas web, usando tal sistema focamos no desenvolvimento das funções e módulos para gestão dos usuários e grupos no backend do projeto.

Esse projeto foi iniciado no ano de 2013 com o nome de smb4manager, seu código fonte está disponível no Github nesse link <https://github.com/jniltinho/smb4manager>, desenvolvido em Python, juntamente com as libs Python e do Samba 4.

O projeto feito para o TCC foi renomeado para go-samba4. No início do TCC, decidimos reescrever todo o projeto em outra linguagem de programação (Golang), por causa da portabilidade da linguagem, fazendo com que um único binário seja usado para quase todas as distribuições Linux.

Em virtude da dificuldade na reescrita das funções e métodos para Golang, voltamos atrás e optamos por manter o projeto em Python, mas agora usando um empacotador chamado Pyinstaller para gerar um único binário, facilitando a instalação do software no Servidor.

REFERÊNCIAS BIBLIOGRÁFICAS

Joélson E. de Almeida, Alexssandro C. Antunes, Jéferson M. de Limas - Instituto Federal De Educação, Ciência e Tecnologia Catarinense

http://sombrio.ifc.edu.br/download/redes/TCC_2013/ARTIGO_Joelson.pdf

https://wiki.samba.org/index.php/User_Documentation

https://www.gta.ufrj.br/grad/01_2/samba/smbcifs.htm

[https://en.wikipedia.org/wiki/Samba_\(software\)](https://en.wikipedia.org/wiki/Samba_(software))

https://sambaxp.org/archive_data/SambaXP2013-DATA/wed/Michael_Adam_Stefan_Metzmacher-Samba4.pdf

https://wiki.samba.org/index.php/Package_Dependencies_Required_to_Build_Samba

https://wiki.samba.org/index.php/Setting_up_Samba_as_an_Active_Directory_Domain_Controller

https://wiki.samba.org/index.php/Adding_users_with_samba_tool

<https://pt.wikipedia.org/wiki/Python>

[https://pt.wikipedia.org/wiki/Flask_\(framework_web\)](https://pt.wikipedia.org/wiki/Flask_(framework_web))

<https://pyinstaller.readthedocs.io/en/v3.3.1/>

<https://www.pyinstaller.org>

https://pt.wikipedia.org/wiki/Aplica%C3%A7%C3%A3o_web

<https://adminlte.io/>

<https://github.com/almasaeed2010/AdminLTE>

<https://getbootstrap.com/docs/4.1/getting-started/introduction/>

<http://getbootstrap.com/>

<https://pt.wikipedia.org/wiki/JQuery>

<https://en.wikipedia.org/wiki/VirtualBox>

<https://pt.wikipedia.org/wiki/VirtualBox>

FOLHA DE APROVAÇÃO DO PROJETO

Tobias Gomes Santos Ra: 914116276

José Nilton de Oliveira Silva Ra: 914112464

SISTEMA WEB DE GESTÃO DE USUÁRIOS DO SAMBA4 – GO-SAMBA4

SISTEMA WEB DE GESTÃO DE USUÁRIOS DO SOFTWARE SAMBA4

Projeto de Pesquisa e Desenvolvimento apresentado ao curso de Bacharelado em Ciência da Computação da Universidade Nove de Julho como requisito parcial para obtenção do grau de Bacharel em 2018, sob a orientação do Prof. Dr. Leandro Zerbinatti

Data: ____/____/____

Assinatura do professor orientador

OBSERVAÇÕES: _____

ANEXOS