

UNIP Marquês

Análise e Desenvolvimento de Sistemas

Letícia Alves P. Ferreira Lopes - H320EC0

Miguel Batista de Paula - R861JH0

Isaque de Medeiros Pinto - R8189G2

PROJETO PARA DESENVOLVIMENTO DE PLATAFORMA PARA ESTUDOS

Projeto sobre desenvolvimento de plataforma para ensino de Python gratuito

São Paulo

2025

Letícia Alves P. Ferreira Lopes - R861JH0
Miguel Batista de Paula - H320EC0
Isaque de Medeiros Pinto - R8189G2

PROJETO PARA DESENVOLVIMENTO DE PLATAFORMA PARA ESTUDOS

Projeto sobre desenvolvimento de plataforma para ensino de Python gratuito

Projeto Integrado Multidisciplinar em ADS

Projeto Integrado Multidisciplinar
para a obtenção do título
de tecnólogo em ADS
, apresentado à Universidade
Paulista UNIP Campus
Marquês.

São Paulo

2025

Resumo

Este projeto tem como objetivo desenvolver uma plataforma educacional interativa, com foco no ensino de lógica computacional e programação básica, utilizando Python como linguagem. Usaremos recursos matemáticos e estatísticos para analisar o desempenho, com isso, permitindo assim a compreensão de padrões de aprendizagem para proporcionar um ensino eficaz. Já a Infraestrutura Computacional terá um funcionamento estável, visando que os usuários tenham eficiência, segurança e bom desempenho, conforme suas necessidade.

Ademais, seguiremos a Lei Geral de Proteção de Dados (LGPD), assegurando a privacidade dos usuários e evidenciando medidas de cibersegurança para evitar possíveis ataques e vazamentos. Por ultimo, incentivaremos valores éticos e sustentáveis, no uso da tecnologia, para promover de uma forma consciente e responsável.

Abstract

This project aims to develop an interactive educational platform focused on teaching computational logic and basic programming using the Python language. Mathematical and statistical tools will be used to analyze user performance, enabling the understanding of learning patterns to provide effective teaching. The computational infrastructure will ensure stable operation, aiming to deliver efficiency, security and good performance, suited to the needs of users. In addition, the project will be compliant with the General Data Protection Law (LGPD), guaranteeing user privacy and implementing cybersecurity measures to prevent potential attacks and data leaks. Finally, ethical and sustainable values will be encouraged in the use of technology to promote conscious and responsible practices.

Sumário

Resumo.....	3
Abstract	3
1. Introdução	1
2. Estrutura do Projeto em Python	1
2.1 Sistema de Login.....	1
2.2 Sistema de Registro dos Cursos	6
4. Infraestrutura de Rede	12
5. Requisitos de Hardware para Escalabilidade	17
6. Com relação as normas de segurança de dados LGPD Lei Geral de Proteção de Dados Pessoais (LGPD).....	18
7. CONCLUSÃO.....	22
8. Referencias	23

1. Introdução

A linguagem **Python** se consolidou como uma das mais utilizadas no mercado de tecnologia, sendo reconhecida por sua versatilidade e facilidade de uso. Criada por **Guido van Rossum** e lançada oficialmente em **1991**, Python se destaca pela sua sintaxe clara e objetiva, o que a torna acessível tanto para iniciantes quanto para profissionais da área. Sua aplicabilidade é bastante ampla, abrangendo desde desenvolvimento web e automação até áreas mais avançadas, como **ciência de dados, inteligência artificial e análise de dados**.

Por ser uma linguagem de **alto nível, interpretada e multiparadigma**, Python permite que os desenvolvedores adotem diferentes estilos de programação, como a **orientação a objetos, programação funcional e imperativa**. Além disso, sua grande comunidade de usuários e a vasta quantidade de bibliotecas disponíveis facilitam a criação de soluções robustas e eficientes.



Neste trabalho, desenvolvemos um projeto prático utilizando Python, que simula um sistema de **gerenciamento de usuários e cursos**. O sistema contempla funcionalidades como **cadastro, login e listagem de contas**, armazenando as informações em arquivos no formato **JSON**, o que permite a persistência dos dados de maneira simples e eficiente.

Além do desenvolvimento do sistema em Python, também incluímos uma **simulação de rede** utilizando o software **Packet Tracer**, com o objetivo de planejar e representar uma infraestrutura de rede adequada para o funcionamento do sistema. Dessa forma, buscamos abordar tanto os conceitos de desenvolvimento de software quanto aspectos fundamentais de infraestrutura e segurança de redes.

2. Estrutura do Projeto em Python

Nosso sistema foi desenvolvido em Python, utilizando uma codificação simples que permite armazenar e gerenciar dados em arquivos no formato JSON. Esse formato foi escolhido pela sua leveza, flexibilidade e fácil integração com a linguagem.

O projeto possui duas partes principais:

-  Sistema de autenticação (login e cadastro de usuários);
-  Sistema de gerenciamento dos cursos dos usuários.

2.1 Sistema de Login

A seguir, apresenta-se o código responsável pela autenticação e cadastro de usuários:

Próxima página...

import json – Importa da biblioteca o json

admin = False

ACCOUNTS_FILE = "JSONN.jsonl"

def loadAccounts(): - Função de puxar as contas registradas no arquivo.

try:

with open(ACCOUNTS_FILE, 'r') as file:

data = json.load(file)

return data

except FileNotFoundError:

print(f"Arquivo '{ACCOUNTS_FILE}' não encontrado. Criando um novo.")

return []

except json.JSONDecodeError:

print(f"Erro ao decodificar o arquivo JSON '{ACCOUNTS_FILE}'. Verifique a formatação.")

return []

def saveAccounts(accounts): - função de abrir contas registradas no arquivo.

try:

with open(ACCOUNTS_FILE, 'w') as file:

json.dump(accounts, file, indent=4)

print("Contas salvas com sucesso.")

except IOError:

print(f"Erro ao escrever no arquivo '{ACCOUNTS_FILE}'.")

def userFileExists(username): - Função que verifica o nome colocado no login.

accounts = loadAccounts()

for account in accounts:

if account.get("username") == username:

```

        return True
    return False

```

```

def authenticate(username, password): - Função que verifica a Senha colocada no login
    accounts = loadAccounts()
    for account in accounts:
        if account.get("username") == username and account.get("password") == password:
            print(f"Login bem-sucedido para o usuário: {username}")
            return True
    print("Nome de usuário ou senha incorretos.")
    return False

```

```

def login(): - Função que coleta os dados referente ao nome e senha.
    print("\n--- Login ---")
    username = input("Digite seu nome de usuário: ")
    password = input("Digite sua senha: ")
    authenticate(username, password)

```

```

def listAccounts(): - Função que mostra a lista de contas registradas.
    print("\n--- Lista de Contas ---")
    accounts = loadAccounts()
    if not accounts:
        print("Nenhuma conta registrada.")
        return
    for account in accounts:
        print(f"Nome de Usuário: {account.get('username')}")

```

```

def addAccount(): - Função que adiciona uma conta no arquivo
    print("\n--- Adicionar Nova Conta ---")
    username = input("Digite o novo nome de usuário: ")

```

```
if userFileExists(username):
    print("Este nome de usuário já existe. Escolha outro.")
    return

password = input("Digite a senha para o novo usuário: ")
accounts = loadAccounts()
accounts.append({"username": username, "password": password})
saveAccounts(accounts)
print(f'Conta para o usuário '{username}' criada com sucesso.')

def firstMenu(): - Função do menu.
    while True:
        print("\n--- Menu Principal ---")
        print("1. Login")
        print("2. Listar Contas")
        print("3. Adicionar Conta")
        print("4. Sair")

        choice = input("Escolha uma opção: ")

        if choice == '1':
            login()
        elif choice == '2':
            listAccounts()
        elif choice == '3':
            addAccount()
        elif choice == '4':
            print("Saindo do programa.")
            break
        else:
            print("Opção inválida. Tente novamente.")
```



```
def main():
```

```
    firstMenu()
```

```
if __name__ == "__main__":
```

```
    main()
```

Foi feito o uso de “try...excepty” para possíveis erros que pode ocorrer no decorrer da execução do programa, como por exemplo a entrada de dados do usuário, se ocorrer um erro de formatação e causar um possível erro, o programa vai avisar qual é o erro para o usuário e prosseguir sem interrupção do código.

Abaixo mostra o funcionamento do código utilizando o IDLE 3.12 do python.

```
==== RESTART: C:\Users\Isaque\Desktop\ONG trabalho\Estrutura trabalho pim.py ===
```

```
--- Menu Principal ---
```

```
1. Login
```

```
2. Listar Contas
```

```
3. Adicionar Conta
```

```
4. Sair
```

```
Escolha uma opção: |
```

```
--- Menu Principal ---
```

```
1. Login
```

```
2. Listar Contas
```

```
3. Adicionar Conta
```

```
4. Sair
```

```
Escolha uma opção: 1
```

```
--- Login ---
```

```
Digite seu nome de usuário: Rodrigo
```

```
Digite sua senha: 12345
```

```
Nome de usuário ou senha incorretos.
```

```
--- Menu Principal ---
```

```
1. Login
```

```
2. Listar Contas
```

```
3. Adicionar Conta
```

```
4. Sair
```

```
Escolha uma opção: 2
```

```
--- Lista de Contas ---
```

```
Nome de Usuário: isaque
```

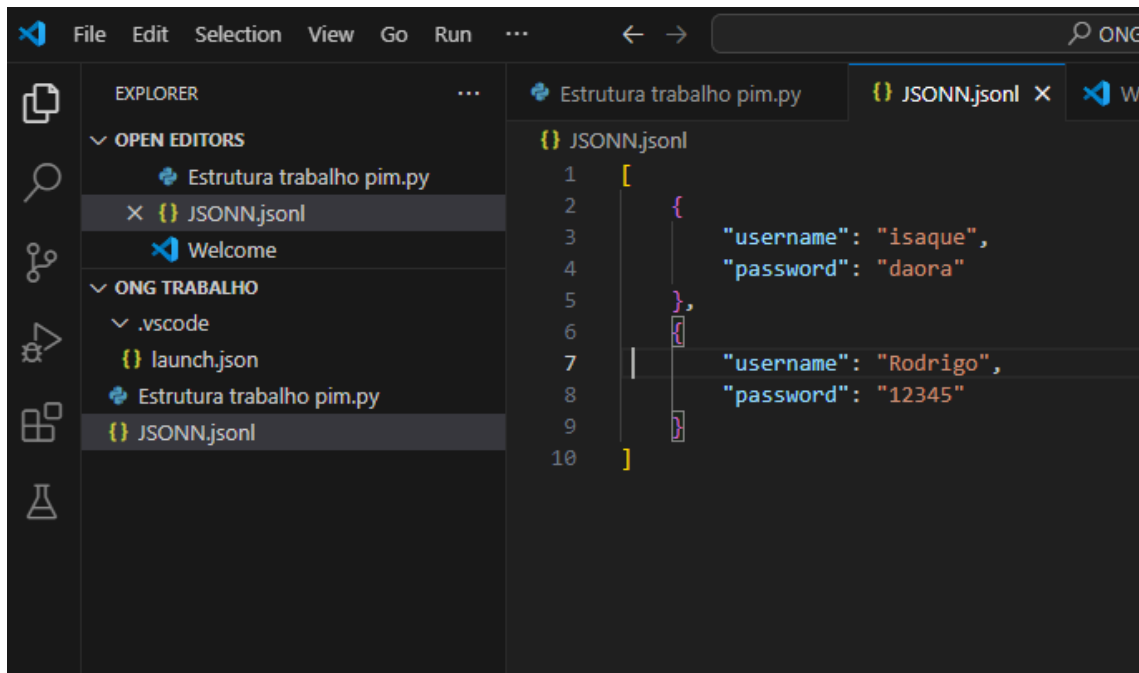
```

--- Menu Principal ---
1. Login
2. Listar Contas
3. Adicionar Conta
4. Sair
Escolha uma opção: 3

--- Adicionar Nova Conta ---
Digite o novo nome de usuário: Rodrigo
Digite a senha para o novo usuário: 12345
Contas salvas com sucesso.
Conta para o usuário 'Rodrigo' criada com sucesso.

--- Login ---
Digite seu nome de usuário: Rodrigo
Digite sua senha: 12345
Login bem-sucedido para o usuário: Rodrigo

```



Como é possível observar subiu de forma automática.

2.2 Sistema de Registro dos Cursos

O segundo módulo do projeto permite que os usuários escolham e armazenem seus cursos, garantindo que esses dados sejam persistentes.

Presente na próxima página...

```
import json
```

```
import os
```

```
lista_de_cursos = (  
    "Curso de Python Inicial",  
    "Curso de Python Intermediário",  
    "Curso de Python Final"  
)
```

```
meus_cursos = []
```

```
ARQUIVO = "JSONNC.jsonl"
```

```
def carregar_cursos():  
    global meus_cursos  
    if os.path.exists(ARQUIVO):  
        try:  
            with open(ARQUIVO, 'r') as arquivo:  
                conteudo = arquivo.read().strip()  
                if conteudo:  
                    meus_cursos = json.loads(conteudo)  
                else:  
                    meus_cursos = []  
        except (json.JSONDecodeError, FileNotFoundError):  
            print("Arquivo corrompido ou inválido. Iniciando com lista vazia.")
```

```

        meus_cursos = []

    else:

        meus_cursos = []

def salvar_cursos():

    with open(ARQUIVO, 'w') as arquivo:

        json.dump(meus_cursos, arquivo, indent=4)

def verCursos():

    print("\n--- Cursos Disponíveis ---")

    for i, curso in enumerate(lista_de_cursos):

        print(f"{i + 1}. {curso}")

    print("-----")

def meusCursos():

    print("\n--- Meus Cursos ---")

    if not meus_cursos:

        print("Você ainda não possui nenhum curso.")

    else:

        for i, curso in enumerate(meus_cursos):

            print(f"{i + 1}. {curso}")

        print("-----")

def addCurso():

```

```
print("\n--- Adicionar Curso ---")

verCursos()

try:

    escolha = input("Digite o número do curso que deseja adicionar (ou '0' para cancelar): ")

    escolha = int(escolha)

    if escolha == 0:

        print("Operação cancelada.")

        return

    indice_curso = escolha - 1

    if 0 <= indice_curso < len(lista_de_cursos):

        curso_selecionado = lista_de_cursos[indice_curso]

        if curso_selecionado in meus_cursos:

            print(f"{curso_selecionado}' já está nos seus cursos!")

        else:

            meus_cursos.append(curso_selecionado)

            salvar_cursos()

            print(f"{curso_selecionado}' foi adicionado com sucesso!")

        else:

            print("Número inválido. Tente novamente.")

except ValueError:
```

```
        print("Por favor, digite um número válido.")

except Exception as e:

    print(f'Ocorreu um erro: {e}')


# --- Programa Principal ---


carregar_cursos()


x = input("Digite 'x' para iniciar: ")

if x.lower() == "x":

    print("\nBem-vindo(a) à plataforma de cursos!")


while True:

    print("\n--- Menu ---")

    print("1. Ver Cursos Disponíveis")

    print("2. Ver Meus Cursos")

    print("3. Adicionar Curso")

    print("4. Sair")

    print("-----")

    opcao = input("Escolha uma opção: ")

    if opcao == "1":

        verCursos()

    elif opcao == "2":
```

```

        meusCursos()

    elif opcao == "3":

        addCurso()

    elif opcao == "4":

        print("Obrigado por usar! Até logo.")

        break

    else:

        print("Opção inválida. Tente novamente.")

else:

    print("Entrada incorreta. Programa encerrado.")

```

3. Execução do Código

```

--- Menu ---
1. Ver Cursos Disponíveis
2. Ver Meus Cursos
3. Adicionar Curso
4. Sair
-----
Escolha uma opção: 2

--- Meus Cursos ---
Você ainda não possui nenhum curso.
-----

Digite 'x' para iniciar: x

Bem-vindo(a) à plataforma de cursos!

--- Menu ---
--- Menu ---
1. Ver Cursos Disponíveis
2. Ver Meus Cursos
3. Adicionar Curso
4. Sair
-----
Escolha uma opção: 1

--- Cursos Disponíveis ---
1. Curso de Python Inicial
2. Curso de Python Intermediário
3. Curso de Python Final
-----

```

O código foi executado na versão **IDLE 3.12 do Python**, funcionando corretamente, armazenando os dados dos usuários e dos cursos no formato JSON de forma local. A execução é feita de forma simples, diretamente pelo terminal do Python ou pelo próprio IDLE.

```

--- Menu ---
1. Ver Cursos Disponíveis
2. Ver Meus Cursos
3. Adicionar Curso
4. Sair
-----
Escolha uma opção: 3

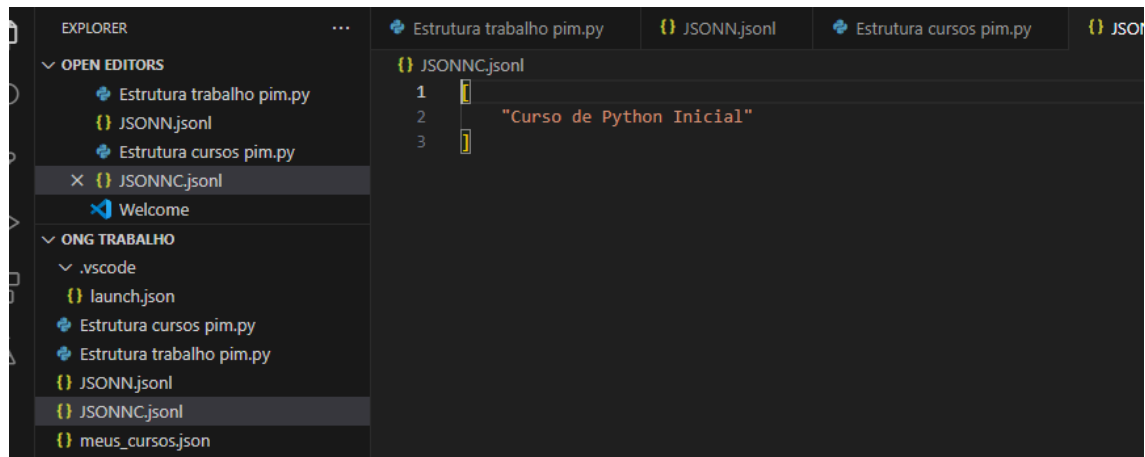
--- Adicionar Curso ---

--- Cursos Disponíveis ---
1. Curso de Python Inicial
2. Curso de Python Intermediário
3. Curso de Python Final
-----
Digite o número do curso que deseja adicionar (ou '0' para cancelar): 1
'Curso de Python Inicial' foi adicionado com sucesso!

--- Menu ---
1. Ver Cursos Disponíveis
2. Ver Meus Cursos
3. Adicionar Curso
4. Sair
-----
Escolha uma opção: 2

--- Meus Cursos ---
1. Curso de Python Inicial
-----

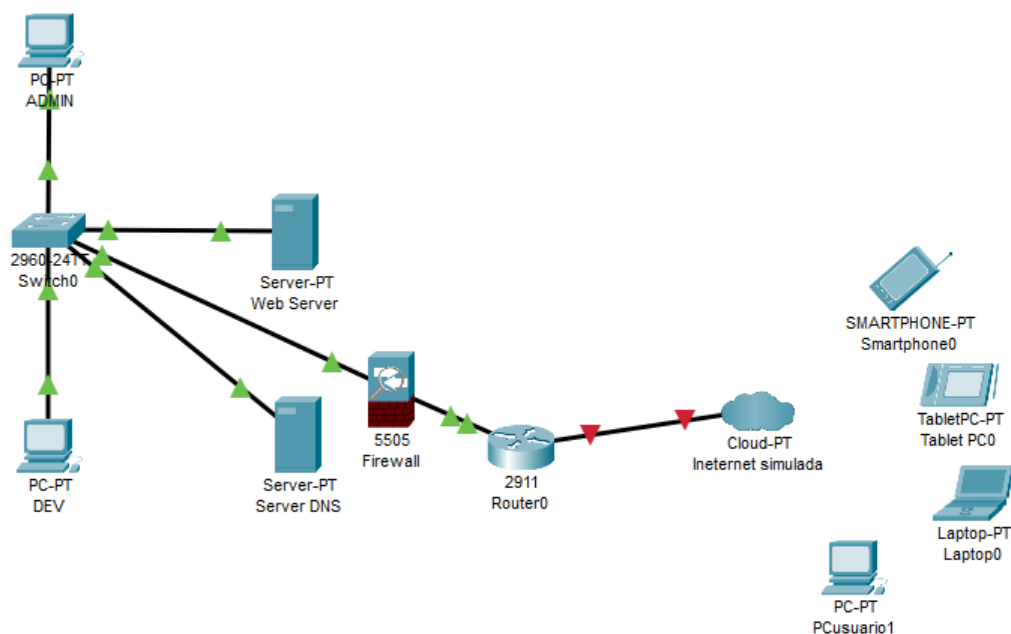
```



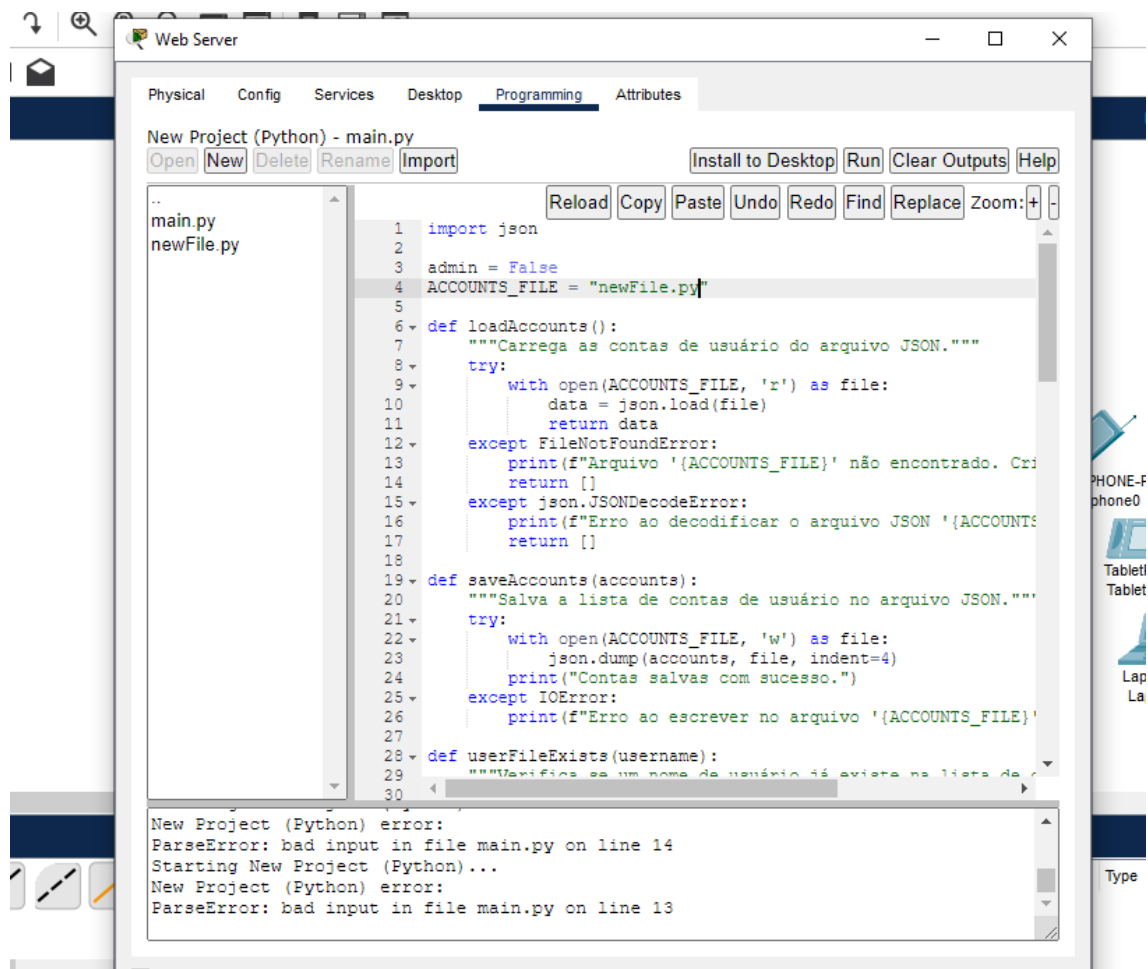
4. Infraestrutura de Rede

Assim como é importante desenvolver um software robusto, é fundamental planejar uma infraestrutura física e lógica de rede eficiente. Essa estrutura garante estabilidade, segurança, disponibilidade e escalabilidade do serviço.

A seguir estão os principais elementos projetados no **Packet Tracer**, simulando a estrutura de rede para o servidor deste sistema:

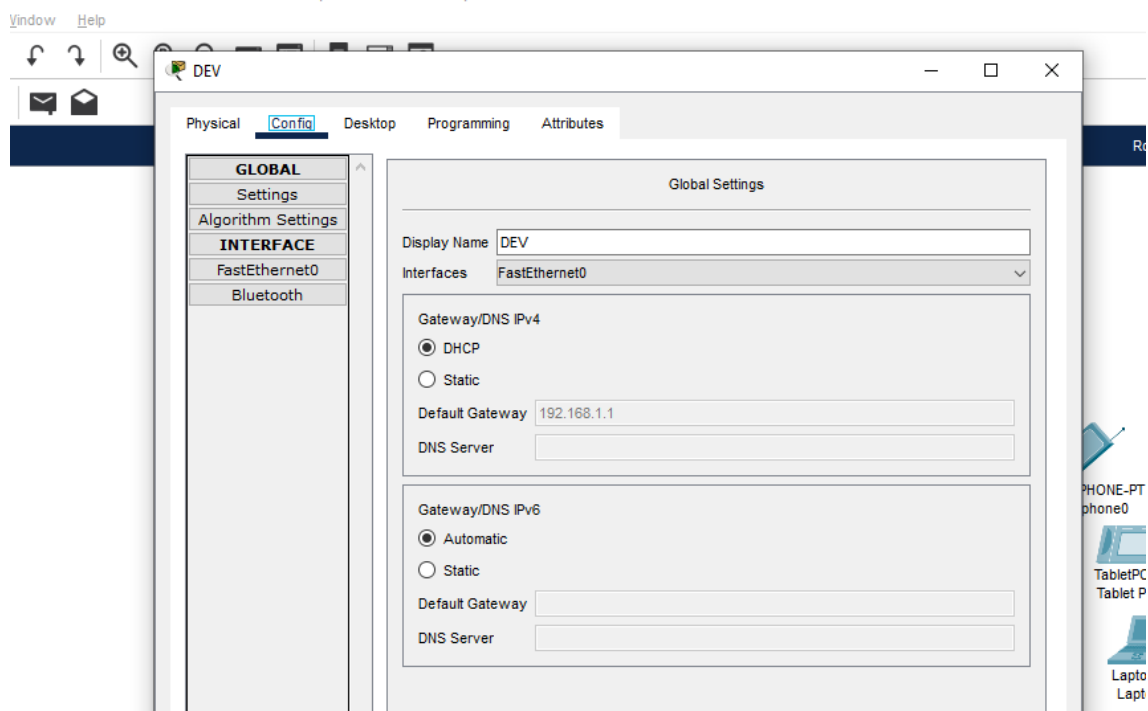


Tendo em mente um servidor seguro tanto para navegação dos clientes quanto para a segurança do site, espera-se que seja implantado um Firewall, entre o roteador e o switch que é conectado com o web server e os demais PC.

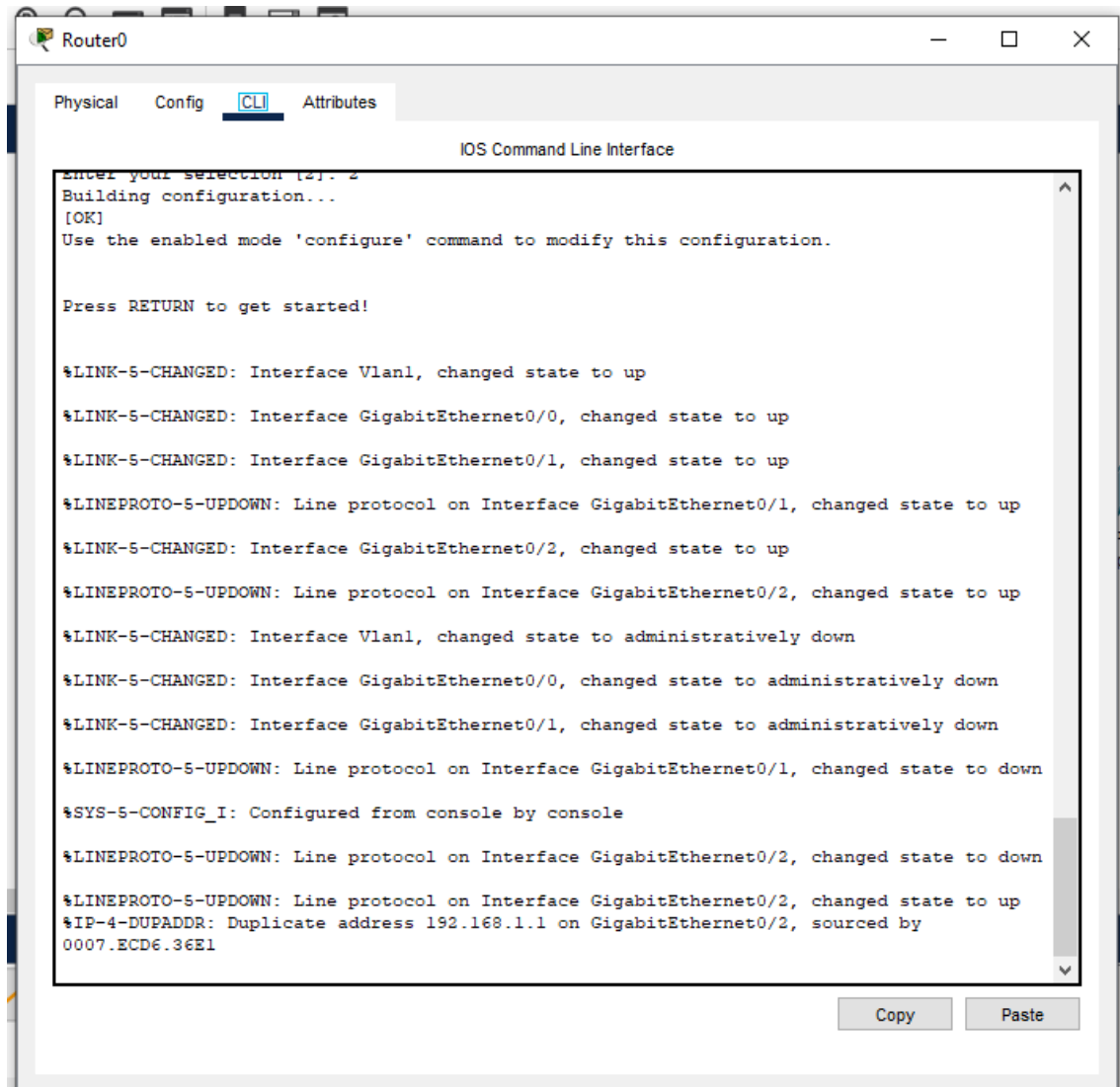


Configuração do site no web server

Switch Central: distribui a conexão para o servidor web, servidor DNS e os PCs da rede.

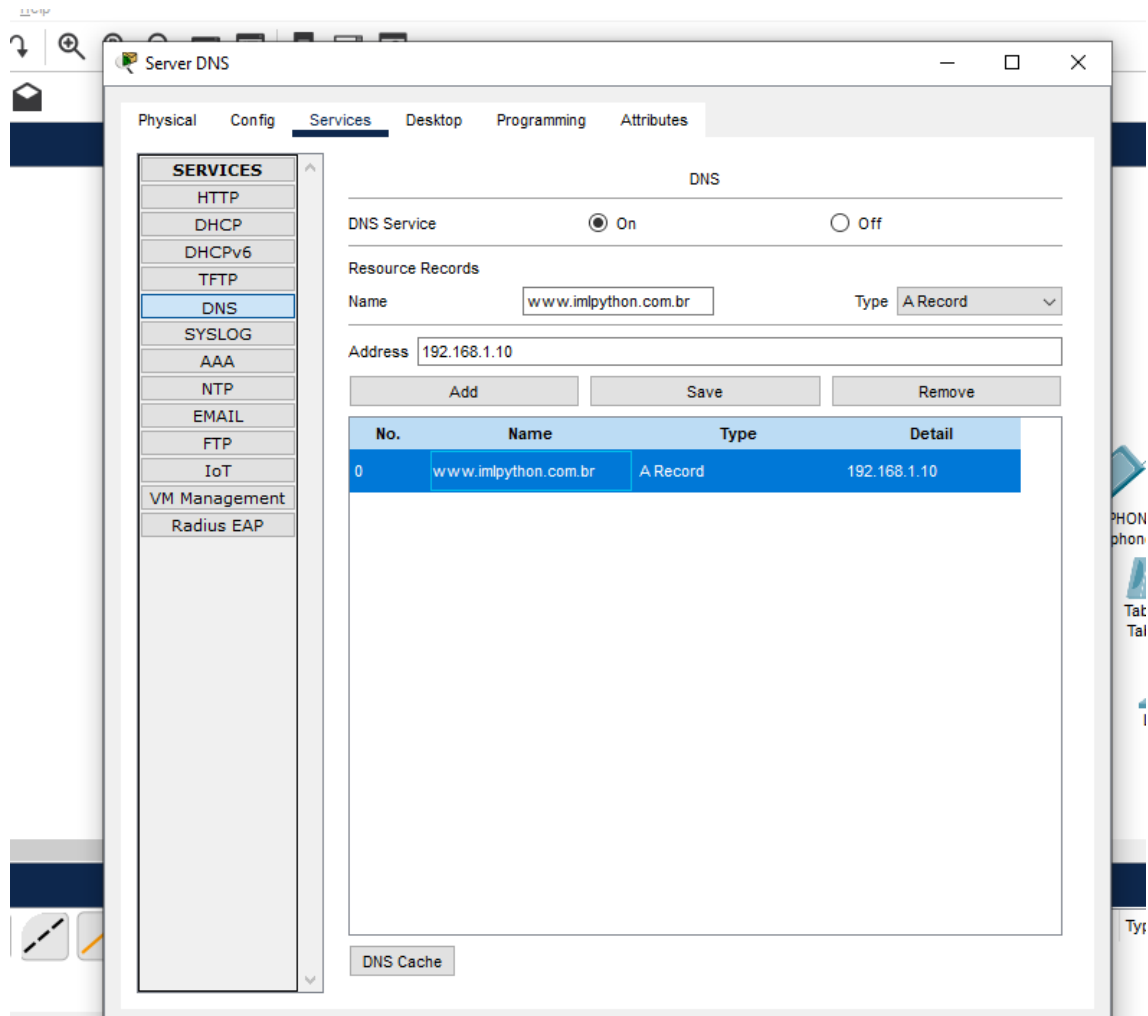


Pc DEV com a configuração do DHCP



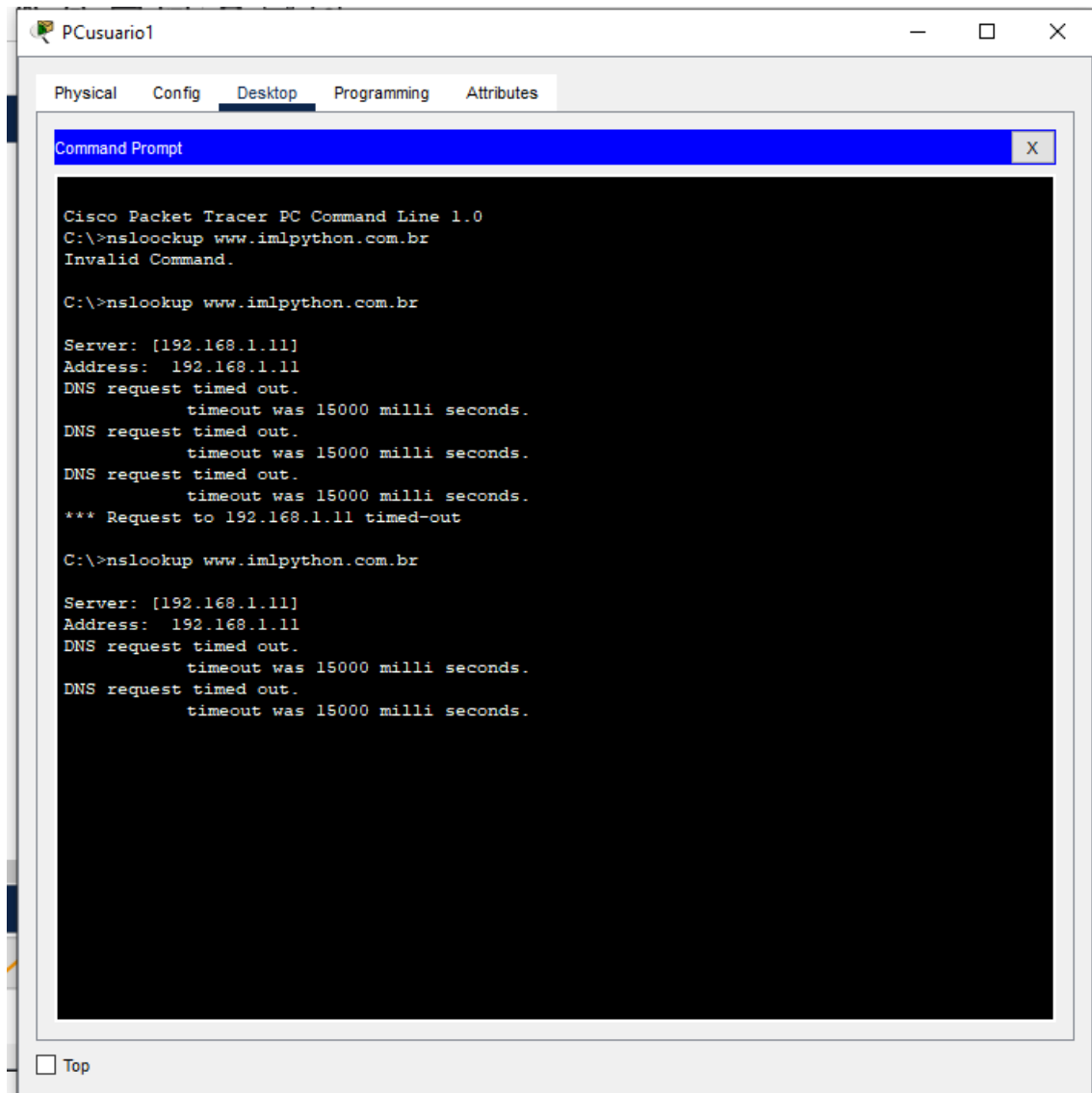
Configuração do roteador.

Roteador: responsável por fazer a comunicação entre redes externas (internet) e a rede local.



Configuração do DNS server, atribuindo um IP adress e um endereçamento.

Servidor DNS: realiza a resolução de nomes, atribuindo um IP fixo e um domínio personalizado para o site.



Foi realizado um teste de conectividade com comandos de **ping** para verificar o acesso do PC do usuário ao servidor, confirmando a funcionalidade da rede simulada.

5. Requisitos de Hardware para Escalabilidade

Considerando que o site deve suportar aproximadamente **100 mil usuários ativos**, é necessário um servidor robusto com os seguintes requisitos:

- **Processador (CPU):** 8 a 32 núcleos (Ex.: Intel Xeon ou AMD EPYC, com suporte a multi-thread);
- **Memória RAM:** de 64 GB a 256 GB;
- **Armazenamento:** SSD NVMe de alta velocidade, com no mínimo 1 TB, podendo usar RAID para redundância e segurança dos dados;
- **Rede:** uplink de 10 Gbps, com redundância para garantir disponibilidade;

- **Sistema Operacional:** Linux (preferencialmente **Ubuntu Server**) devido à sua estabilidade, segurança e comunidade ativa.

Por que escolhemos o Ubuntu Server?

O Ubuntu Server é uma escolha muito vantajosa quando se pensa em desempenho e custo-benefício. Ele foi desenvolvido com foco em eficiência, e por isso é consideravelmente mais leve do que o Windows Server, especialmente por não trazer uma interface gráfica pesada, o que o torna ideal para servidores.

Esse sistema operacional é amplamente utilizado em servidores web, o que o torna ainda mais atrativo. Como é comum nesse tipo de ambiente, muitos recursos e ferramentas já estão otimizados para o Ubuntu, como é o caso do NGINX o qual está no planejamento para ser usado no nosso site, um dos servidores web mais populares, que tradicionalmente funciona melhor em sistemas Linux.

Outro grande diferencial é a comunidade ativa de usuários Linux. Isso garante uma ampla variedade de tutoriais, fóruns e conteúdos de apoio facilmente acessíveis, além de constantes atualizações e inovações na área de segurança. Com isso, o Ubuntu Server se torna uma alternativa sólida e eficiente, especialmente para quem busca um sistema confiável, moderno e com excelente custo-benefício.

6. Com relação as normas de segurança de dados LGPD

Lei Geral de Proteção de Dados Pessoais (LGPD)

A Lei Geral de Proteção de Dados Pessoais (LGPD), Lei nº 13.709/2018, foi promulgada para proteger os direitos fundamentais de liberdade, privacidade e a livre formação da personalidade de cada indivíduo. A Lei trata do tratamento de dados pessoais, dispostos em meio físico ou digital, feito por pessoa física ou jurídica, de direito público ou privado, englobando um amplo conjunto de operações que podem ocorrer em meios manuais ou digitais.

Com base na lei Lei nº 13.709/2018, escolhemos implementar a criptografia como sistema de segurança, um sistema que garante confidencialidade conforme exigido pela LGPD.

Essa tecnologia assegura que informações pessoais sejam codificadas impedindo que invasores e pessoas não autorizadas tenham acesso indesejado mesmo que os dados vazem ou sejam interceptados.

A criptografia ajuda a proteger os dados das pessoas e torna a segurança da organização mais eficiente.

Ética, Cidadania e Sustentabilidade – Economia de Energia e Descarte Correto

Para promovermos o uso consciente e sustentável na tecnologia devemos colocar em pauta a educação, a inovação tecnológica e consumo sustentável e consciente.

Acreditamos que com essas práticas e com tópicos sugeridos abaixo, teríamos um baixo impacto ambiental e um futuro sustentável.

Sugestões de economia de energia e uso consciente de equipamentos:

- Desligar equipamentos eletrônicos, como monitores e computadores em caso de final de expediente e os mesmos não estarem em uso.

- Ativação do modo de economia de energia.

- Por meio de possíveis aplicativos e plataformas digitais, as pessoas poderiam se informar sobre práticas sustentáveis, aprender sobre o impacto de suas ações no meio ambiente.

- Trocar monitores antigos de tubo por modelos LED, que consomem menos energia.

- Substituir lâmpadas incandescentes por LED.

Estratégias para descarte de equipamentos sem uso, com defeito e obsoletos:

O lixo eletrônico é formado por equipamentos obsoletos e dispositivos eletrônicos com defeito que já não têm mais utilidade para a empresa. Geralmente, esses lixos precisam de atenção devido aos seus componentes mais danosos ao meio ambiente, como plástico, vidro e metais pesados.

É possível encontrar empresas especializadas em descarte, como a Ecobraz Emigre, que abrange diversos equipamentos tecnológicos e no site, explicam como funciona todo o processo.

Esses equipamentos ultrapassados são divididos em quatro categorias: linha branca; linha marrom; linha azul; e linha verde. Cada uma agrega diferentes tipos de equipamento e contam com formas diferentes de descarte.

Abaixo, veja o que cada categoria inclui

- Linha branca: refrigeradores, congeladores, fogões, máquinas de lavar roupas, lava-louças, secadores e condicionadores de ar;

- Linha marrom: TVs, monitores, aparelhos de DVD e VHS, câmeras de filmagem, equipamentos de áudio e outros eletrônicos de tubo, plasma, LCD e LED;

- Linha azul: liquidificadores, ferros de passar roupa, furadeiras, secadores, aspiradores de pó, batedeiras, entre outros eletrodomésticos de pequeno porte;

- Linha verde: PCs, MACs, notebooks, tablets, dispositivos móveis, entre outros acessórios de informática e de escritório.

Entre todas as categorias, a principal é a linha verde, que frequentemente está presente nas pequenas, médias e grandes empresas. Felizmente, atualmente o descarte destes itens é algo bem fácil.

É possível fazer o descarte com o mínimo de esforço a partir de pontos de coleta disponibilizados pelos próprios fabricantes dos dispositivos eletrônicos.

A seguir, iremos compartilhar estratégias para descarte adequado:

- Jamais descartar no lixo comum, pois muito desses equipamentos contêm plástico, vidro e metais pesados.

- Encaminhar os equipamentos para postos de coleta, como ponto de coleta de lixo eletrônico.

Conscientizar e educar a todos sobre o impacto ambiental do lixo eletrônico é essencial para iniciarmos a construção de futuro sustentável.

Cibersegurança- Boas Práticas de Segurança Digital

Para garantirmos segurança digital, aplicamos práticas que ajudam a nos proteger de ataques cibernéticos.

Tópicos sugeridos:

- Senhas fortes, utilizando letras minúsculas e maiúsculas, números e símbolos.

Observação: Vale lembrar que alguns sites ou aplicativos já utilizam essa medida protetiva como sugestão.

- Manter softwares atualizados, incluindo aplicativos e sistemas operacionais.
- Evitar links e anexos duvidosos.
- Ter hábito de fazer backups de arquivos importantes.
- Utilizar antivírus atualizado e confiável, além de manter outros sistemas seguros.
- Utilize um armazenador/gerenciador de senhas para gerar senhas seguras.
- Atento a todos os riscos da internet.
- É essencial identificar sinais de autenticidade nos sites visitados.

Algumas dessas práticas e sugestões podem ajudar a reduzir riscos, como phishing, e proteger informações pessoais.

Políticas de proteção de dados para garantir a privacidade e os direitos dos titulares

A organização tem como objetivo garantir que dados pessoais sejam visados e tratados de acordo com princípio de privacidade legislação vigente, como a Lei Geral de Proteção de Dados Pessoais.

São conjuntos de regras que empresas, instituições e demais organizações, para garantir que os dados pessoais de titulares fiquem em segurança e tenham privacidade.

Há três pilares fundamentais na proteção de dados: confidencialidade, integridade e disponibilidade (CID).

A política citada protege a privacidade, garante os direitos de titulares, evita vazamentos pessoais indesejáveis. Além disso, é essencial para manter a confiança de todos, protegendo-os de diversas ameaças digitais.

7. CONCLUSÃO

Para finalizar, este estudo nos permitiu assimilar e usar ideias chave de criação de programas e de arquitetura de redes. Com a linguagem Python, criamos um sistema de gestão de usuários direto, mas útil, que inclui funções como criar contas, validar acessos e ver listas de usuários. Usamos arquivos JSON para guardar os dados de um jeito eficaz.

Ademais, com a ajuda da simulação no Packet Tracer, pudemos idealizar um modelo de rede essencial, focado em assegurar que o sistema comunique bem e opere sem problemas, pensando em pontos como organização, proteção e acesso constante.

Elaborar este projeto nos deu uma experiência real e completa do que aprendemos durante o semestre, mostrando como é crucial juntar programação, raciocínio, proteção e redes nos sistemas de hoje. Por isso, entendemos que usar ferramentas fáceis de obter, como Python e Packet Tracer, é crucial para formar profissionais prontos para os desafios do setor de tecnologia.

Referências

<https://www.gov.br/previdencia/pt-br/aceso-a-informacao/lei-geral-de-protecao-de-dados-pessoais-lgpd>

Livros e autores renomados sobre LGPD:

BAPTISTA, Camila Monteiro; CARDOSO, Alexandre Magno. *LGPD: Lei Geral de Proteção de Dados Pessoais Comentada*. São Paulo: Revista dos Tribunais, 2020.

CISCO NETWORKING ACADEMY. Packet Tracer – Ferramenta de simulação de redes. Disponível em: <https://www.netacad.com/courses/packet-tracer>.

Cibersegurança- Boas Práticas de Segurança Digital:

<https://nic.br/noticia/na-midia/as-5-melhores-praticas-de-seguranca-digital-para-proteger-seus-dados/>

Instituições especializadas:

CERT.br – Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil. Disponível em: <https://cert.br/>

Estratégias para descarte de equipamentos sem uso, com defeito e obsoletos:

<https://wapssolutions.com.br/o-que-fazer-com-equipamentos-obsoletos/>

Sobre descarte correto de lixo eletrônico:

ABINEE (Associação Brasileira da Indústria Elétrica e Eletrônica). *Guia para descarte de resíduos eletroeletrônicos*. Disponível no site da ABINEE.

Ética, Cidadania e Sustentabilidade – Economia de Energia e Descarte Correto:

<https://ecobraz.org/public/>

Políticas de proteção de dados para garantir a privacidade e os direitos dos titulares:

<https://www.gov.br/previdencia/pt-br/aceso-a-informacao/lei-geral-de-protecao-de-dados-pessoais-lgpd>

PYTHON SOFTWARE FOUNDATION. Documentação oficial do Python. Disponível em:

<https://docs.python.org/pt-br/3/>.

UBUNTU. Documentação oficial do Ubuntu. Canonical Ltd., 2025. Disponível em:

<https://ubuntu.com/tutorials>.

Link do GITHUB com os códigos para serem analisados: <https://github.com/Isaque-Medeiros/PIM-1-Semestre>.