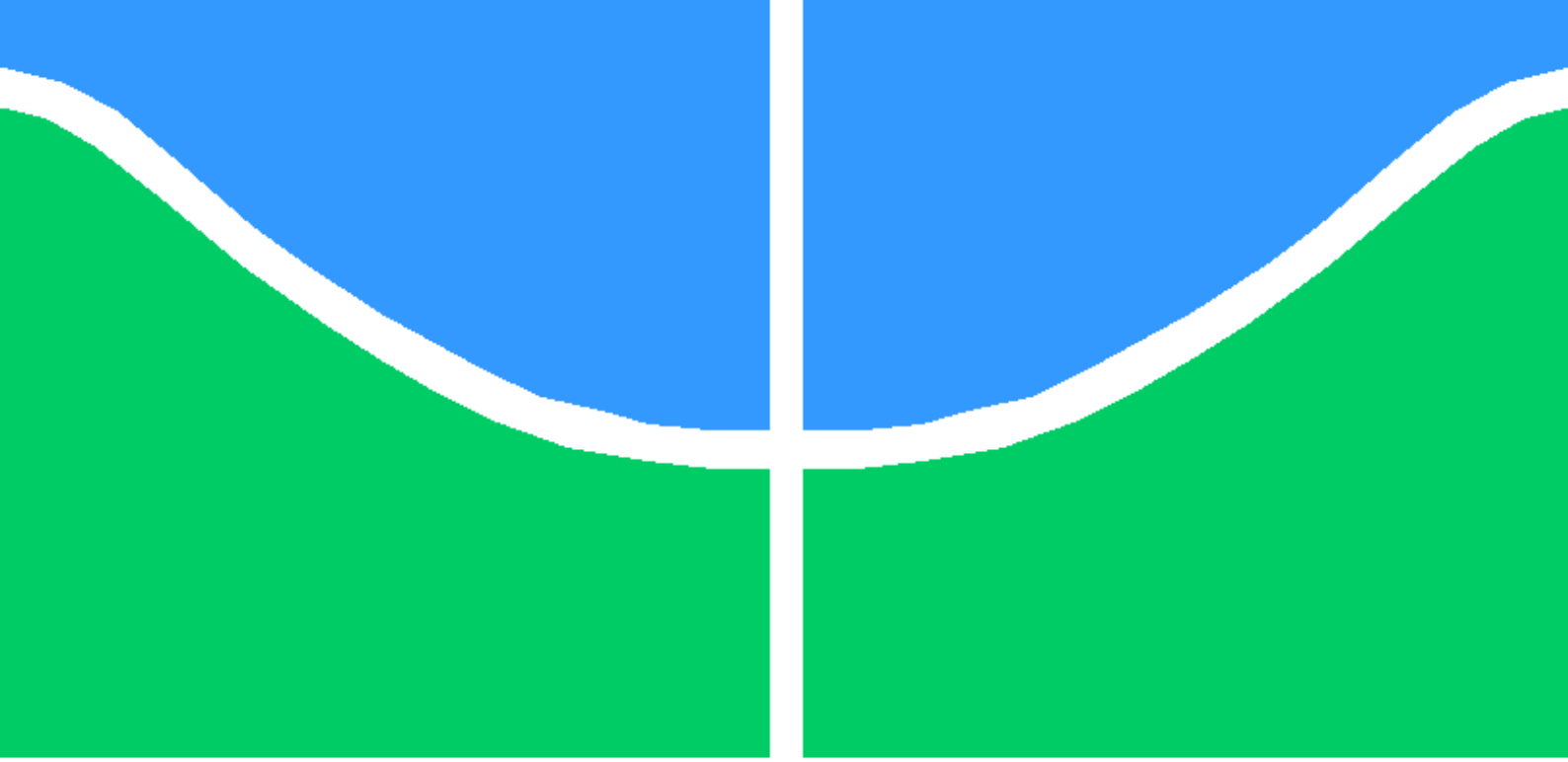

PROPOSTA DE ARQUITETURA DE UM PORTAL SEGURO DE PROVAS ELETRÔNICAS UTILIZANDO SDN COMO SOLUÇÃO DE GERENCIAMENTO DOS USUÁRIOS DA REDE

JOÃO PAULO PIMENTEL

DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

PROPOSTA DE ARQUITETURA DE UM PORTAL SEGURO DE PROVAS ELETRÔNICAS UTILIZANDO SDN COMO SOLUÇÃO DE GERENCIAMENTO DOS USUÁRIOS DA REDE

JOÃO PAULO PIMENTEL

Orientador: PROF. DR. GEORGES DANIEL AMVAME NZE, ENE/UNB

DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA

PUBLICAÇÃO PPEE.MP - 016/2022
BRASÍLIA-DF, 28 DE ABRIL DE 2022.

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

PROPOSTA DE ARQUITETURA DE UM PORTAL SEGURO DE PROVAS ELETRÔNICAS UTILIZANDO SDN COMO SOLUÇÃO DE GERENCIAMENTO DOS USUÁRIOS DA REDE

JOÃO PAULO PIMENTEL

DISSERTAÇÃO DE MESTRADO ACADÊMICO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA ELÉTRICA.

APROVADA POR:

Prof. Dr. Georges Daniel Amvame Nze, ENE/UnB
Orientador - Presidente

Prof. Dr. Rafael Rabelo Nunes, ENE/UnB
Examinador Interno

Prof. Dr. Artur Carlos de Morais, SEMOB
Examinador Externo

Prof. Dr. Robson de Oliveira Albuquerque, ENE/UnB
Suplente

BRASÍLIA, 28 DE ABRIL DE 2022.

FICHA CATALOGRÁFICA
JOÃO PAULO PIMENTEL
PROPOSTA DE ARQUITETURA DE UM PORTAL SEGUROS DE PROVAS ELETRÔNICAS UTILIZANDO SDN COMO SOLUÇÃO DE GERENCIAMENTO DOS USUÁRIOS DA REDE
2022xv, 75p., 201x297 mm
(ENE/FT/UnB, Mestre, Engenharia Elétrica, 2022)
Dissertação de Mestrado - Universidade de Brasília
Faculdade de Tecnologia - Departamento de Engenharia Elétrica

REFERÊNCIA BIBLIOGRÁFICA

JOÃO PAULO PIMENTEL (2022) PROPOSTA DE ARQUITETURA DE UM PORTAL SEGUROS DE PROVAS ELETRÔNICAS UTILIZANDO SDN COMO SOLUÇÃO DE GERENCIAMENTO DOS USUÁRIOS DA REDE. Dissertação de Mestrado em Engenharia Elétrica, Publicação 016/2022, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 75p.

CESSÃO DE DIREITOS

AUTOR: JOÃO PAULO PIMENTEL
TÍTULO: PROPOSTA DE ARQUITETURA DE UM PORTAL SEGUROS DE PROVAS ELETRÔNICAS UTILIZANDO SDN COMO SOLUÇÃO DE GERENCIAMENTO DOS USUÁRIOS DA REDE.
GRAU: Mestre ANO: 2022

JOÃO PAULO PIMENTEL
Depto. de Engenharia Elétrica (ENE) - FT Universidade de Brasília (UnB) Campus Darcy Ribeiro CEP 70919-970 - Brasília - DF - Brasil

Agradecimentos

Agradeço primeiramente a Deus por ter me dado força e saúde para concluir este trabalho com sucesso, principalmente em um período muito crítico em virtude da pandemia COVID-19.

Agradeço ao meu orientador, professor Dr. Georges Daniel Amvame Nze, que me orientou de forma profissional e amiga nas horas mais complicadas durante este trabalho e aturou tantas dúvidas e problemas relativos ao assunto e outros detalhes pertinentes à criação desta dissertação.

Ao meu Coorientador, professor Dr. Fábio Lúcio Lopes de Mendonça, que teve a paciência e que cobrou muito a conclusão deste trabalho, além disso, ajudou bastante no desenvolvimento da dissertação e publicação dos artigos.

À minha Mãe Darci, esposa Roselene e filhas (Caterine, Sofia e Emily), por acreditar na minha capacidade e ter paciência dando total apoio, incentivo e compreensão de minha ausência durante a elaboração deste trabalho.

Aos Professores do PPEE - Programa de Pós Graduação em Engenharia Elétrica da UnB, Rafael Timóteo de Sousa Júnior, Flávio Elias Gomes de Deus e Rafael Rabelo Nunes, pelas grandes dicas, constante apoio, incentivo e amizade, essenciais para o desenvolvimento deste trabalho. Agradeço também aos membros da banca.

Agradeço em especial ao Programa de Pós-Graduação Profissional em Engenharia Elétrica (PPEE) da UnB, às Agências brasileiras de pesquisa, desenvolvimento e inovação CNPq e CAPES, ao Laboratório LATITUDE/UnB, ao Projeto de Pesquisa "Pesquisa Aplicada ao Desenvolvimento de Sistemas de Softwares Seguros"do Gabinete de Segurança Institucional da Presidência da República/CEPESC e ao Projeto de Pesquisa aplicada à integração de inovações tecnológicas à arquitetura de sistemas de informação da secretaria nacional de assistência social (SNAS) do Ministério da Cidadania-MC. Projetos de Pesquisa que me deram subsídio para escolha do meu tema e que fui bolsista durante o período do meu trabalho.

PROPOSTA DE ARQUITETURA DE UM PORTAL DE PROVAS SEGURO UTILIZANDO SDN COMO SOLUÇÃO DE GERENCIAMENTO DOS USUÁRIOS DA REDE.

Autor: João Paulo Pimentel
Orientador: Georges Daniel Amvame Nze
Programa de Pós-graduação Profissional em Engenharia Elétrica - PPEE
Brasília, abril de 2022

O crescente processo de desenvolvimento de *softwares* vem enfrentando uma série de tecnologias e metodologias de segurança baseadas em aplicação e controle de *firewalls*, com dificuldades para atender às exigências previstas pelas normas e modelos de segurança de *software*. Neste contexto, este trabalho apresenta um modelo de arquitetura de técnica de segurança a ser implantado em um Portal Seguro de Provas Eletrônicas (PSPE) utilizando uma rede definida por *software* (SDN) como mecanismo de gerência dos usuários da Rede com autenticação por SSO e geração de um *Token* através de um serviço de *QR Code* gerado para a autenticação e validação destes usuários.

Palavras-chave: **Portal Seguro, SSO, *QR Code*, *Token*, SDN**.

ARCHITECTURAL PROPOSAL OF A SECURE EVIDENCE PORTAL USING SDN AS A NETWORK USER MANAGEMENT SOLUTION.

Author: João Paulo Pimentel
Supervisor: Georges Daniel Amvame Nze
Pot-graduate Program Professional on Electrical Engineeering - PPEE
Brasilia, april 2022

The growing process of software development has been facing a series of technologies, security techniques and methodologies based on application and firewall control, with difficulties to meet the requirements set forth by software security standards and models. In this context, this work presents a security technique architecture model to be implemented in a Secure Electronic Evidence Portal (PSPE) using a software-defined network (SDN) as a network user management mechanism with SSO authentication and generation of a Token through a QR Code service generated for the authentication and validation of these users.

Keywords: **Secure Portal, SSO, QR Code, Token, SDN.**.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE TERMOS E SIGLAS

AD	*Active Directory*
API	*Application Programming Interface*
CSS	*Cascading Style Sheets*
HTML	*HyperText Markup Language*
IoT	*Internet das Coisas*
jQuery	*JavaScript Query*
JSON	*JavaScript Object Notation*
ONOS	*Open Network Operating System*
OVF	*Open Virtualization Format*
PSPE	Portal Seguro de Provas Eletrônicas
QR Code	*Quick Response Code*
RP	*Relying Party*
SAAS	*Auth-as-a-Service*
SDN	Redes Definidas por Software
SSH	*Secure Shell*
SSL	*Secure Sockets Layer*
SSO	*Single Sign-On*
TLS	*Transport Layer Security*
UML	Linguagem de Modelagem Unificada
VM	*Máquina Virtual*
VPN	Rede Virtual Privada

Capítulo 1

Introdução

Quando o assunto é sobre falhas de segurança, é comum imaginar o autor deste ataque burlando mecanismos de proteção de segurança, tais como *firewalls*, ou outros dispositivos de segurança físicos ou mesmo por *software*, explorando falhas em sistemas operacionais ou aplicativos com problemas de segurança em busca de informações que vão satisfazer sua intenção do referido ataque. Para estas invasões são utilizados dois métodos principais, um deles é atacar uma vulnerabilidade até então desconhecida ou sem segurança implementada e o outro método é descobrir as vulnerabilidades do sistema e ataca-las com a utilização de códigos maliciosos e ferramentas desenvolvidas para este fim, pois estas falhas podem estar presente em ativos de uma companhia e um dos tipos de ativos mais vulneráveis nas empresas são os seus próprios sistemas com informações importantes da empresa e de seus clientes [2].

O conceito de segurança de *softwares* envolve plataformas de *hardware*, aplicações, ferramentas, protocolos de comunicação, sistemas operacionais potencialmente diferentes. Assim como, modelos de dados, tipos de aplicativos e assuntos relacionados a políticas de segurança, entre outros componentes. Dessa forma, não podemos nos referir a uma segurança de *software* de forma única, homogênea e coesa. Para isso, um conjunto de técnicas podem auxiliar em criação de um modelo de *software* mais seguro, envolvendo diversos mecanismos de controle, com novas tecnologias [2].

Considerando tal situação, existe a possibilidade, e mesmo a necessidade, de integração de tecnologias com componentes específicos (por exemplo, um sistema de autenticação, criptografias de alto nível baseadas em algoritmos de *Quick Response Code* (QR Code) e componentes de redes para auxiliar operações), levando a uma possível necessidade de mesclar tais tecnologias com o objetivo de ser um sistema seguro. Sem utilizar os moledos de arquiteturas convencionais que acaba tornando as aplicações lentas e complexas, com isso podemos criar soluções seguras, mas que são implementadas etapas baseadas em uma arquitetura distribuída [3].

Neste trabalho, será proposto um modelo de arquitetura de Portal Seguro de Provas Eletrônicas (PSPE), que a princípio irá utilizar uma série de tecnologias para dar subsidio a quesitos de segurança, como: mecanismo de cadastro componentes baseado em *Single Sign-On* (SSO), como

mecanismo de criptografia utilizará, sistemas de *QR Code* com geração de *Token* como sistema de autenticação para os usuários e tecnologia de Redes Definidas por *Software* (SDN) como solução de gerenciamento de usuários na rede. Neste Portal, o usuário terá a segurança de autenticação única e segura é feito justamente para que os acadêmicos possam realizar as suas avaliações de acordo com o seu agendamento, que após a escolha da referida disciplina, data e horário será gerado um *QR Code* e *Token* para validação e confirmação ao entrar no laboratório de informática - Labredes, ou somente um *Token* para acessar remotamente o referido laboratório que irá realizar a avaliação. Tal arquitetura do modelo proposto será melhor explicada no capitulo 3.

A ideia central do PSPE - Portal Seguro de Provas Eletrônicas é a implementar um mecanismo de validação e controle dos usuários através de um serviço de autenticação por SSO, que na sequência o acadêmico terá a opção de escolher qual tipo de avaliação irá realizar, gerando o *QR Code* no qual deverá apresenta-lo no laboratório de informática que irá realizar a prova, através de autenticação o mesmo terá a autorização de realizar sua avaliação de modo presencial. Além disso, o sistema também irá realizar um controle de sessão de usuários por máquina autenticando via SDN.

Com o crescente avanço da pandemia causada pelo COVID-19, houve um crescimento gradativamente de aulas *online* com a utilização de técnicas e ferramentas específicas. Entretanto, após pesquisas realizadas não foi possível identificar alguma ferramenta específica que pudesse obter um controle de segurança para realização de provas eletrônicas de forma *online*. Para isso, o PSPE seguindo o modelo de arquitetura de aplicação de provas de forma presenciais, houve a iniciativa de avançar a arquitetura para um mecanismo hibrido, em caso de escolha de um modo remoto, será informado ao acadêmico sobre a necessidade da instalação de um serviço de VPN - *Virtual Private Network*, do inglês (Rede Virtual Privada) no caso específico do laboratório Latitude a utilização do software *software OpenVPN Connect* para se conectar e realizar a sua avaliação de forma remota. Uma vez conectado o usuário terá acesso as máquinas do laboratório de informática da instituição, acessando máquinas específicas com a utilização de sistemas SDN, podendo assim, realizar um pré-agendamento através de instruções que serão encaminhadas para o e-mail do aluno. Para provas no modo presencial, após apresentar o *QR Code* no laboratório previamente agendado para a devida avaliação, este gerenciamento será resolvido com a utilização de SDN, o módulo de autenticação/validação deste Portal de Provas irá gerar um *Token* válido somente para a avaliação solicitada na máquina liberada. Com a utilização da matrícula e o *Token* gerado pelo Portal *Web*, o usuário irá logar no referido Portal de Provas e realizar a devida avaliação. Para as provas no modo remoto, o usuário irá utilizar matrícula e *Token* enviados para o *e-mail*.

Dessa forma, a arquitetura do sistema irá utilizar três pilares para controle de segurança baseados em: autenticação SSO, o QR *Code* com Geração de *Token* e SDN para controle de sessão dos usuários.

1.1 Motivação

Os *softwares* sempre foram alvos de especulações ou notícias que afirmem que o mesmo é inseguro perceberá que muitos potenciais clientes ou usuários, passam a fugir do produto ou da empresa, ou pior, as pessoas com algum ressentimento do produto passam a aumentar ainda mais o acúmulo de publicidade negativa, provando à outras pessoas que usar o produto é perigoso. Infelizmente as pessoas tendem a acompanhar somente as notícias ruins, o que causa um dano irreparável de tal modo que chegará o momento em que as pessoas procuraram o produto do concorrente, afundando de vez o produto inseguro [4].

Por isso, é preciso garantir a segurança do *software*, apesar das normas de segurança serem bem claras, as mesmas não disponibilizam de uma forma detalhada a forma de implantação nos processos, tornando difícil para as empresas saber exatamente como atendê-las em sua plenitude. Por este motivo, este estudo foi desenvolvido e se experimentou na prática a aplicação de algumas normas e práticas sugeridas de segurança [5].

Visando ampliar os estudos sobre segurança da informação e novos modelos de técnicas de segurança em *softwares* com a utilização de componentes e SDN, o centro de pesquisa "CEPESC" junto com o Laboratório de Tecnologia da Tomada de Decisão - Latitude da Universidade de Brasília (UNB) criaram uma parceria através de um projeto de pesquisa o que teve como meta inicial a criação de um *software* seguro o que incentivou a desenvolver este trabalho em soluções adquiridas no projeto.

Após alguns meses de trabalho junto à equipe do Projeto CEPESC e o laboratório de pesquisa do Programa de Pós Graduação Profissional em Engenharia Elétrica da UnB, analisando diversas técnicas de segurança, observando os diversos modelos de sistemas de provas eletrônicas existentes, foi o principal motivo que levou a acreditar na possibilidade da criação de uma arquitetura para elaboração de um mecanismo de reconhecimento automático de usuários para realização de provas de forma *online* e segura. O estudo inicial apontou para as possibilidades das redes SDN constituírem uma técnica básica de um sistema de reconhecimento de usuários baseadas em redes que permitisse autenticação e controle de sessão por usuários. Por tais razões, a pesquisa voltou-se à proposta de uma arquitetura de sistema para tal finalidade e a avaliação de sua efetividade.

Atualmente existem vários sistemas e portais de provas em escolas e universidades, entretanto nem todos estão preocupados em manter segurança e inteligência de autenticação única e com uma validação através de *QR Code* e através de um controle por usuário. Dessa forma, o PSPE utilizará desses artifícios para além de facilitar o controle de realização das avaliações nas instituições, também irá definir e colaborar com uma segurança tanto da parte da instituição quanto da parte do usuário de maneira centralizada e inteligente.

1.2 Objetivo

O principal objetivo deste trabalho é propor um modelo de arquitetura de um *software* seguro através de um Portal Seguro de Provas Eletrônicas (PSPE), de modo a permitir funcionalidades de interoperação de segurança tais como SSO para cadastro único dos usuários, *QR Code* com geração de *Token* como sistema de autenticação para os usuários e a tecnologia SDN como solução de gerenciamento de sessão dos usuários na rede. O embasamento do trabalho comtempla os estudos sobre arquitetura de *softwares* seguros, modelos de comunicação baseada em redes definidas por *softwares*, e outros temas pertinentes que são tratados no estudo de trabalhos relacionados.

1.3 Objetivos Específicos

Para alcançar o objetivo geral desta proposta, os seguintes objetivos específicos foram definidos:

- Estudar e especificar funcionalidades de SSO para cadastro único dos usuários.
- Propor a arquitetura de *software* seguro que atenda o PSPE.
- Apresentar um protótipo envolvendo as tecnologias e funcionalidades existentes no PSPE, oferecendo uma realização da arquitetura proposta para permitir a avaliação dos resultados.
- Realizar validação da proposta, por intermédio de baterias de testes das funcionalidades, seu desempenho e comparação clássica de *software* seguro.

1.4 Contribuição do Trabalho

O modelo aqui proposto articula-se em três módulos, em que a arquitetura do sistema irá utilizar três pilares para controle de segurança baseados em: autenticação SSO, o *QR Code* com a geração de *Token* para validação dos usuários e o SDN para controle de sessão dos usuários.

As demais contribuições são buscadas através da avaliação do modelo proposto, o que inclui o desenvolvimento do Portal Seguro de Provas Eletrônicas - PSPE para testar e validar a API fazendo esta comunicação através do Protocolo de comunicação *OpenFlow*.

Além disso, o trabalho irá contribuir para o Laboratório de Tecnologia da Tomada de Decisão - Latitude da Universidade de Brasília (UNB), na qual o projeto vem sendo executado como prova de conceito.

1.5 Publicações vinculadas ao Trabalho

- [6] Pimentel, João Paulo; Nze ,Georges Daniel Amvame; Mendonca, Fábio L. L.; De Sousa, Jr., Rafael T.; Canedo, Edna Dias; Ribeiro, Vanessa Coelho. Proposta de Arquitetura de um Portal de Provas Seguro Utilizando SDN como Solução de Gerenciamento de Redes. In: Conferências Ibero-Americanas WWW/Internet e Computação Aplicada 2020, Virtual 19-20 novembro 2020, Vilamoura. Lisboa: IADIS Press, 2020. v. 1. p. 237-240. ISBN: 978-989-8704-25-2
- [7] Pimentel, João Paulo; Monteiro, Matheus Santos; Francisco L., De Caldas Filho; Mendonca, Fábio L. L.; Fares, Awatef Ali Yousef R.;De Sousa, Jr., Rafael T.. Vulnerabilidades de Sistemas Ciber Físicos. In: Conferências Ibero-Americanas WWW/Internet e Computação Aplicada 2020, Virtual 19-20 novembro 2020, Vilamoura. Lisboa: IADIS Press, 2020. v. 1. p. 202-206. ISBN: 978-989-8704-25-2
- [8] DE MENDONÇA, Fabio Lucio Lopes; DA SILVA CAMÕES, Renato José; PIMENTEL, João Paulo. CRIAÇÃO DE UM PROJETO DE SOC DE PEQUENO PORTE, VIABILIZANDO MELHORAS DE MONITORAMENTO E SOLUÇÕES DE SEGURANÇA DA INFORMAÇÃO. TECNOLOGIAS EM PROJEÇÃO, v. 10, n. 2, p. 46-57, 2019.

1.6 Metodologia

No sentido de alcançar os objetivos desta dissertação, o projeto de pesquisa constitui-se das fases iniciais de identificação e caracterização do problema de softwares seguros, levantamento de uma hipótese de que uma solução de segurança distribuída, com modulos de autenticação e controle de sessão.

Trata-se de uma pesquisa do tipo exploratória sobre as tecnologias e ferramentas utilizadas para o desenvolvimento de um protótipo de Portal Seguro de Provas Eletrônicas, também será demostrado a interação deste Portal com a autenticação do usuário e a comunicação com o outro módulo do sistema aonde os estudantes irão realizar de fato as avaliações.

1.7 Organização do Trabalho

Esta dissertação é composta por cinco capítulos, incluindo este de introdução.

O capítulo 2 trata de trabalhos relacionados e bases conceituais da dissertação, o que inclui uma revisão sobre as ferramentas e tecnologias adotadas para o desenvolvimento do Portal de Provas, tais como autenticação SSO, *QR Code*, *Token*, SDN, Protocolo *OpenFlow*, *Simulador Mininet* ,*Ryu*, *Microsoft Visual Studio*, Linguagem de Programação *.NET*, Serviços de nuvem do Microsoft *Azure*, objetos *JSON*, dentre outras tecnologias adotadas na elaboração desta proposta. Neste capítulo também são apresentados os trabalhos relacionados com esta Dissertação.

No capítulo 3, além da Estrutura Analítica do Projeto, aonde serão mostrados os levantamentos de requisitos, a análise e validação de requisitos, declaração de objetos, lista de eventos, especificação dos casos de uso e os diagramas de caso de uso e de classes, também será apresentada a Proposta de Arquitetura do PSPE, constituindo a contribuição central da dissertação.

O capítulo 4 destina-se à avaliação da proposta do Portal de Provas, descrevendo o protótipo desenvolvido, as configurações do ambiente, os cenários de validação das funcionalidades do referido Portal Seguro utilizando SDN como solução de Gerenciamento de Redes, bem como a escalabilidade, desempenho deste modelo que utiliza SDN.

O capítulo 5 apresenta a conclusão deste trabalho e propõe trabalhos futuros.

Capítulo 2

Revisão Bibliográfica e Trabalhos Relacionados

A autenticação SSO (*Single Sign-On*) na *Web*, como exemplo do *Login* do Google ou Facebook, são baseados no protocolo *OpenID Connect* [9]. Esse protocolo permite que através da confiança capitam a autenticação do usuário aos provedores de identidade, ou seja, é através dele que é passada a identificação do usuário para a devida autenticação e validação nas aplicações *Web*. No SSO, um serviço da *Web*, nesse contexto chamado RP (*Relying Party*), delega autenticação para o chamado provedor de identidade (IdP). A tarefa de um IdP no SSO da *Web* é autenticar usuários e atestar suas identidades aos RPs. Alguns serviços que os usuários usam diariamente como provedores de *e-mail* ou redes sociais, servem como IdPs [10].

Hoje, diversos usuários da *Web* utilizam suas contas do Facebook ou do Google para acessar sites de terceiros (*Relying Party*). Esse esquema de *login* único baseado na *Web* é ativado pelo *OAuth* 2.0, um protocolo de autorização de recursos da *Web* que foi adotado pelos principais provedores de serviços [11].

O *OpenID Connect* expande o protocolo de autorização do *OAuth* 2.0 a ser usado como um protocolo de autenticação que permite a execução de *login* único, ou seja, ele apresenta o conceito de *token* de ID, que admite ao cliente verificar a identidade e obter informações básicas de perfil sobre o referido usuário. Como ele é uma extensão do *OAuth* 2.0, permite que os aplicativos adquiram *tokens* de acessos com segurança. Estes *tokens* de acesso são utilizados para acessar os recursos protegidos por um servidor de autorização [12].

No *OpenID Connect*, o cliente é conhecido como Parte Confiante (RP) e o Servidor de Autorização é conhecido como Provedor *OpenID* (OP), conforme é mostrado na figura 2.1.

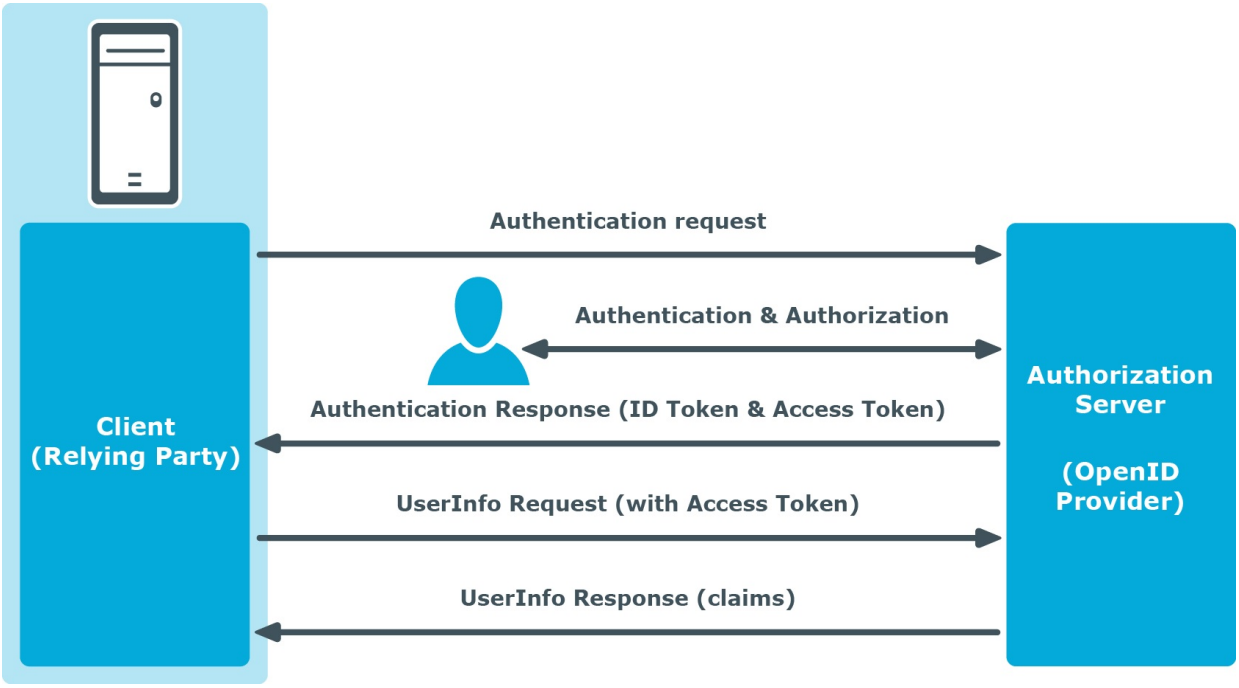

Figura 2.1: Visão conceitual do OpenID Connect. [1]

O *QR Code* (*Quick Response Code*) que é utilizado para resposta rápida [13] tem a vantagem nesta proposta do Portal, pois vem da facilidade em realizar a leitura durante a entrada no laboratório para realizar a devida prova. O *QR Code* é um símbolo bidimensional, elaborado em 1994 pela Toyota com o intuito de ser utilizado no controle de produção de preças automotivas [3], mas acabou sendo utilizado em vários outros seguimentos, tais como controle de passageiros de avião, para vendas de produtos, indicação de sites de vendas, propaganda e para validar mensagens enviadas por *WhatsApp* ou *Telegram*.

A imagem do *QR Code* consegue transmitir cerca de 7 mil caracteres, tudo isso compactado em uma matriz de pontos, e estes pontos podem ser lido rapidamente de qualquer ângulo por leitores decodificadores. Esta matriz, com tamanho mínimo de 21x21 células e máximo de 177x177 [3] vem sendo utilizado cada vez mais em todos os estabelecimentos, tais como academias para atividades físicas, entrada em shows ou jogos de futebol. Segue abaixo um exemplo de um *QR Code* com texto simples, mostrado na figura 2.2, nele contém o título desta Dissertação.

Figura 2.2: Exemplo de *QR Code* com o Título da Dissertação. Gerado através do *codegenerator*

Como já exposto na Seção 1, a presente dissertação tem a proposta de apresentar uma arquitetura de um Portal seguro utilizando SDN como Gerenciamento de Redes, desta forma, neste capítulo serão abordados alguns conceitos pertinentes a este trabalho nas questões relativas às tecnologias adotadas para a implementação deste Portal, assim como a comunicação entre a solução e o SDN que cuidará deste Gerenciamento de Redes através de *Software*.

2.1 SDN - Redes Definidas por *Software*

A sigla "SDN", que é referente a *Software Defined Networking* é simplesmente uma das motivações que surgiram para as Redes Definidas por *sofware* e esta ideia veio para que o administrador da Rede pudesse separar as regras de envio de pacotes, em equipamentos distintos, do envio propriamente dito [14].

Sobre SDN é importante destacar o que é citado por Menezes:

> *Software Defined Networking* (SDN) é um novo paradigma para as redes de computadores que muda a maneira de gerenciar, administrar e controlar a rede de computadores, quebrando as limitações impostas das redes atuais ao desacoplar o plano de controle, relacionado ao controle do tráfego de rede, do plano de dados, responsável por encaminhar o tráfego em acordo com o plano de controle (KREUTZ et al, 2015, apud MENEZES et al, 2018, p. 1) [15].

O gerenciamento baseado em SDN que será controlado por um elemento central, ficou conhecido por separar do controle da rede dos dispositivos de encaminhamento, tais como roteadores, *switches*, fazendo com esta lógica fosse definida por *software* de forma centralizada, provendo à rede através da utilização de aplicações customizadas para cada cenário definido [14] e estas

aplicações implementam a inteligência da rede e proveem abstração dos detalhes técnicos dos equipamentos e dos protocolos.

As Redes Definidas por *Software* são bem mais que atribuíções de atividades lógicas que são definidas pelo protocolo *OpenFlow*, este protocolo faz parte da aplicação lógica do conceito de SDN, juntamente com o desenvolvimento de aplicações apropriadas para as regras de negócio da empresa, manipulação da rede logicamente ao invés de fisicamente, tudo isso abrange seu conceito, não apenas o processamento lógico [16].

2.2 *Docker*

*Docke*r é uma plataforma aberta para desenvolvimento, envio e execução de aplicativos permitindo a separação dos seus aplicativos de sua infraestrutura para que possa entregar o *software* de uma forma mais ágil e consistente. Com o Docker, pode-se gerenciar a infraestrutura de rede da mesma forma que gerencia seus aplicativos e aproveitar da metodologia do Docker para enviar, testar e implantar os códigos da aplicação e enviá-los para produção [17].

A plataforma *Docker* oferece a capacidade de empacotar e executar um aplicativo em um ambiente isolado denominado contêiner. O isolamento e a segurança permitem que você execute vários contêineres simultaneamente em um determinado host. Os contêineres são leves e contêm tudo o que é necessário para executar o aplicativo, portanto, não é necessário depender do que está instalado atualmente no host, podendo até compartilhar contêineres facilmente enquanto trabalha e certificar de que todos com quem é compartilhado recebam o mesmo contêiner que funciona da mesma maneira [17].

2.3 Protocolo *openflow*

O *OpenFlow* é a principal interface de código aberto entre SDN controladores e comutadores ou outros caminhos de dados. Possui suporte para diversos aplicativos usando protocolos de planos de dados com a classificação de pacotes sendo executadas usando tabelas de correspondência e operações de processamento de pacotes chamadas de ações ou instruções [18]. O *OpenFlow* é um padrão definido pela Open Networking Foundation (ONF) para implementar SDN em equipamentos de rede e define a interface entre um controlador *OpenFlow* e um comutador *OpenFlow*, ou seja, ele permite que o *OpenFlow Controller* instrua o comutador *OpenFlow* sobre como lidar com pacotes de dados recebidos [19].

O protocolo *OpenFlow*, originalmente criado pela Universidade de Stanford, tinha a função de atender as novas arquiteturas e protocolos derivadas de equipamentos comerciais. Aprimorado em conceitos de fluxo, onde se tem a combinação de determinados campos do cabeçalho do pacote que virá a ser processado pelo equipamento, onde uma determinada ação será tomada [14].

Geralmente os *switches* trabalham com endereços de memória para realizar suas atividades,

tais como encaminhar pacotes e até resolver políticas de segurança, portanto, esse modelo rígido não se aplica a equipamentos que trabalham com o protocolo *OpenFlow*, onde se é possível realizar todas essas ações e ainda programar conforme a necessidade de negócio, [20] ou seja, um *switch* que não trabalha com *OpenFlow* tem tanto o encaminhamento de pacotes, que é chamado de *data path*, como as tomadas de decisões que é chamado de *control path*, sendo estas tarefas realizadas no próprio equipamento. Já os *switches* que trabalham com o protocolo *OpenFlow* não realizam o *control path*, apenas o *data path*. Uma controladora destes tipos de *swiches* é a encarregada por se comunicar com o *switch*, por meio de mensagens, a fim de escolher quais ações deverão tomar [20].

2.4 Simulador *Mininet*

A Ferramenta *Mininet* é utilizada para realizar uma simulação de SDN permitindo uma rápida prototipação para uma grande infraestrutura virtual de rede de computadores e para isso precisa somente de um computador. Também é possível criar protótipos de redes virtuais escaláveis baseados em *software*, tais como o *OpenFlow*, utilizando primitivas de virtualização de Sistema Operacional, criando, interagindo e até customizando protótipos de Redes Definidas por *Software* [16]. A plataforma de Emulação de Rede *Mininet* cria uma rede *virtual OpenFlow* - com controlador, *switches, hosts e links* - em uma única máquina real ou máquina virtual [20].

Mininet habilita a prototipagem rápida de SDN, testes desta topologia sem a necessidade de conectar a uma rede física, podendo ser utilizada por vários desenvolvedores ao mesmo tempo e trabalhando de forma independente na mesma topologia. Estas redes executam códigos reais, incluindo aplicativos de rede Unix/Linux padrão, assim como também executam o *Kernel* Linux e pilha de rede de uma forma real. [21]

Os códigos que foram desenvolvidos e testados utilizando *Mininet* para um controlador *OpenFlow*, ou seja, para um comutador modificado, ou estação de trabalho, pode passar para um sistema real com poucas alterações de códigos, que servirá para teste real, avaliação de desempenho e implementação. Isto significa que um projeto que trabalha em *Mininet* geralmente pode ir diretamente para os comutadores físicos [21].

Vários sistemas operacionais virtualizam recursos de computação utilizando abstração de processos. *Mininet* usa virtualização baseada em processos para executar diversas estações de trabalho e comutadores em um único *kernel* do sistema operacional. Com o *Mininet* pode criar comutadores OpenFlow de *kernel* ou de espaço do usuário, controladores para controlar os comutadores, e as estações de trabalho para se comunicar através da rede simulada. *Mininet* conecta comutadores e estações de trabalho usando pares *ethernet* [21].

2.5 ONOS

O Controlador ONOS (*Open Network Operating System*) é um controlador SDN de código aberto, distribuído sob licença da APACHE 2.0, para contruções de solucões de Gerenciamento de Redes por *Software* projetado para atender as necessidades ou simplesmente ajudar as empresas na economia do *hardware*, pois além de oferecer a flexibilidade para criar e implantar seviços de rede dinâmicas com interfaces com programação bem simples, também oferecem seus serviços de nuvem ONOS que inclui uma plataforma e vários aplicativos para gerência de redes atuando como controlador SDN extensível, modular e distribuído [22].

2.6 Oracle *VirtualBox*

O Oracle *VirtualBox* é um aplicativo de virtualização de plataforma que pode ser instalado em computadores Intel ou AMD, utilizando os Sistemas Opercionais Windows, macOS, Linux ou Oracle Solaris. Ele é utilizado para executar vários Sistemas Operacionais dentro de máquinas virtuais, ou seja, dentro do macOS poderá executar um Sistema Operacional Linux ou Windows. [23]

O *VirtualBox* também pode importar e exportar máquinas virtuais nos formatos OVF (*Open Virtualization Format*) e Formatos de serviço em nuvem. OVF é um padrão de plataforma suportado por muitos produtos de virtualização que permite a criação de máquinas virtuais prontas que podem então ser importadas. O Oracle *VM VirtualBox* torna a importação e exportação OVF fácil de fazer, usando a janela do Gerenciador do *VirtualBox* ou a interface de linha de comando. [23]

2.7 *Ryu*

Ryu é uma estrutura de SDN baseada em componentes que fornece componentes de software com APIs bem definidas que tornam mais fácil para os desenvolvedores criarem novos aplicativos de gerenciamento e controle de rede. *Ryu* suporta vários protocolos para gerenciar dispositivos de rede, como *OpenFlow*, *Netconf*, *OF-config*, dentre outros dispositivos [24]. Por ser uma distribuição gratuita e estar sob a licença da Apache 2.0, os códigos ficam disponíveis para baixar no git no seguinte endereço: https://github.com/faucetsdn/ryu [25].

No sentido de demonstrar sobre a instalação, configuração e utilização do *Ryu*, segue abaixo algumas instruções básicas para o Sistema Operacional Linux Ubuntu (16.04 LTS ou posterior):

- Instalando o *Ryu* a partir do *prompt* de comandos: % pip install ryu
- Instalando o *Ryu* a partir do código fonte disponível no *git*:
  - % git clone https://github.com/faucetsdn/ryu.git

  - % cd ryu; pip install .

- Após criar e salvar o seu aplicativo *Ryu*, basta digitar: % ryu-manager yourapp.py

Para os requisitos operacionais para as funções do *Ryu* serão necessários alguns pacotes extras:

- Lxml e ncclient
- NETCONF
- Console SSH
- O serviço de protocolo de banco de dados *SQLAlchemy*

Para instalar estes pacotes extras será necessário exeucutar o seguinte comando:

- % pip install -r tools/optional-requires

2.8 Microsoft *Visual Studio*

O Microsoft *Visual Studio* é uma IDE para criação de códigos, testagem de códigos, controle de versão do *Git* e do *GitHub*, análise, depuração, controle, colaboração e implantação, facilitando assim a integração com o Microsoft *Azure* que é uma plataforma utilizada para execução de aplicativos e serviços, baseada nos conceitos da computação em nuvem. [26].

2.8.1 Linguagem de Programação .NET

O desenvolvimento de aplicações com a Linguagem de Programação .NET (*Web e Desktop*) utilizando C# e *Visual Basic* que são linguagens projetadas para a criação de vários aplicativos para serem executados no .NET *Framework*. Essas linguagens são poderosas, fortemente tipadas e orientadas a objeto. Elas se baseiam no .NET utilizando o compilador *Roslyn*, que fornece APIs de análise de código avançadas disponíveis como software livre no GitHub [27].

2.8.2 ASP.NET Core

O ASP.NET *Core* é uma estrutura *open source* e multiplataforma para a criação de aplicativos atuais, conectados à Internet e baseados em nuvem, como serviços e aplicativos *Web*, aplicativos IoT e *back-ends* móveis. Os aplicativos desenvolvidos com ASP.NET *Core* podem ser executados nos *runtimes* do .NET *Core* ou do .NET *Framework*. Ele foi projetado para fornecer uma estrutura de desenvolvimento otimizado para aplicativos que são implantados na nuvem ou executados localmente e também consiste em componentes modulares com sobrecarga mínima, para manter a flexibilidade durante a construção de soluções inovadoras [28].

2.8.3 Serviços de nuvem do Microsoft *Azure*

O Serviço de Aplicativo do *Azure* permite que sejam criados e implantados aplicativos *Web*, *back-ends* de dispositivos móveis e *APIs RESTful* na linguagem de programação de sua escolha sem gerenciar a infraestrutura. Eles oferecem o dimensionamento automático e alta disponibilidade, compatível com Windows e Linux e permite implantações automatizadas do *GitHub*, *Azure DevOps* ou qualquer repositório *Git* [29].

O Serviço de Aplicativo não agrega apenas o poder do Microsoft *Azure* ao seu aplicativo, como segurança, balanceamento de carga, dimensionamento automático e gerenciamento automatizado, podendo aproveitar também seus recursos de *DevOps*, como implantação contínua desde o *Azure DevOps, GitHub, Docker Hub* e outras fontes, gerenciamento de pacotes, ambientes de preparo, domínio personalizado e certificados TLS/SSL [29].

2.9 Objetos JSON

JavaScript Object Notation (JSON) é um formato baseado em texto padrão para representar dados estruturados com base na sintaxe do objeto *JavaScript*. É comumente usado para transmitir dados em aplicativos da Web (por exemplo, enviar alguns dados do servidor para o cliente, para que possam ser exibidos em uma página da Web ou vice-versa) [30].

2.10 Bootstrap

Bootstrap é um *framework JavaScript, HTML e CSS* para desenvolvimento de *sites* e aplicações *web* responsivas e elaborada para trabalhar com a filosofia *mobile first*, tornando o desenvolvimento *front-end* bem ágil e com uma facilidade para integrar com todos os tipos de dispositivos e projetos de desenvolvimento [31].

2.11 jQuery

jQuery é uma biblioteca *JavaScript* para manipulação de documentos HTML, eventos, animações e interações com AJAX para desenvolvimento de aplicações web [32].

2.12 Git

Git é um sistema de controle de versão distribuído gratuito e de código aberto projetado para lidar com todos os tipos de projetos com velocidade e eficiência [33].

2.13 Rede Virtual GNS3

O GNS3 é um *software* de código aberto e gratuito utilizado para emular, configurar, testar e solucionar problemas de redes virtuais e reais. GNS3 permite que você execute uma pequena topologia que consiste em apenas alguns dispositivos em seu *Notebook* ou PC, para aqueles que têm muitos dispositivos hospedados em vários servidores ou até mesmo hospedados na nuvem. Originalmente emulando apenas dispositivos Cisco usando um software chamado *Dynamips*, o GNS3 agora evoluiu e oferece suporte a vários dispositivos de outros fornecedores de rede, incluindo *switches* virtuais Cisco, Cisco ASAs, *Brocade vRouters*, *switches Cumulus Linux*, instâncias *Docker*, HPE VSRs, dentre outros dispositivos Linux. [34]

2.14 Banco de Dados *NoSQL Firebase*

O *Firebase* é um banco de dados *NoSQL*, ou seja, ao contrário de um banco SQL, não existem tabelas e nem registros, mas sim uma árvore JSON hospedada na nuvem. Quando é adicionado os dados a esta árvore JSON, eles se tornam um nó na estrutura JSON com uma chave associada. [35]

2.15 Ferramenta UML *Lucidchart*

Lucidchart é uma plataforma baseada na *web* que permite aos usuários colaborarem no desenho, revisão e compartilhamento de gráficos e diagramas tais como o Diagrama de Caso de Uso, Diagrama de Classe, Diagrama de Atividade, dentro outros utilizados para documentação de *softwares* [36].

2.16 Trabalhos Relacionados

Este trabalho está relacionado com a Dissertação [37], onde é apresentada uma proposta de Identidade Eletrônica e Autenticação de Médicos no Brasil que facilite a atenticação e autorização com níveis de garantia em sistemas informáticos (SSO, *Token*), fornecendo com isso, uma identificação eletrônica para os médicos entrarem nos diversos sitemas dos Hospitais ou Clínicas que eles trabalham. Mas a proposta apresentada no PSPE, além destas autenticações e autorizações apresentadas, vamos ter o *QR Code* que irá validar/autenticar a entrada no Laboratório, ou realizar remotamente, a devida prova agendada pelo PSPE após a solicitação do usuário.

O proposta do PSPE também está relacionada com o trabalho [38], que teve como objetivo demonstrar, com base no conhecimento adquirido e numa prova de conceito original desenvolvida especificamente, que ainda existe evolução possível nesta área, integrando duas tecnologias atuais, o protocolo *OPenID Connect* (utilizando o SSO) e o Cartão de Cidadão Nacional que,

conjugadas, permitem criar um método de autenticação pessoal forte e seguro, que pode ser completamente gerido pelo utilizador. No PSPE também foi proposto a implementação de segurança com o SSO, mas utilizando também Token e SDN.

Este trabalho também relacionado ao artigo [6], onde é apresentada uma proposta incial da implementação deste Portal de Provas seguro utilizando SDN como solução de gerenciamento de redes. Neste artigo foi demonstrado como seria esta interação do usuário com o Portal de Provas realizando o seu agendamento, gerando o seu QR Code e indo até ao Laboratório realizar a sua avaliação agendada. Mas devido à Pandemia do COVID-19 para a conclusão deste trabalho foi adicionado a forma de acesso remoto via VPN para a realização da prova, e utilizando uma rede definida por software (SDN) como mecanismo de gerência de autenticação por SSO e geração de um *Token* através de um serviço de *QR Code* gerado para a autenticação e validação dos usuários.

Outro trabalho relacionado com esta Dissertação é o artigo [7] visando a melhoria da segurança eletrônica mais especificamente o motor dos portões eletrônicos, este artigo tem como objetivo apresentar um protótipo de dispositivo de segurança capaz de aproveitar da vulnerabilidade dos sistemas de portão eletrônico, como interceptar o código de um portão por um dispositivo próximo, possibilitando que um invasor abra o portão sem autorização do proprietário da casa. Com o advento dos dispositivos e tecnologias de IoT (*Internet of Things*), essas vulnerabilidades podem ser exploradas ainda mais longe, permitindo que o código do portão seja enviado para terceiros pela internet, além de comandar a abertura do portão a distância. Já no PSPE, o módulo de segurança terá que impedir que outras pessoas consigam *Tokens* válidos para realizarem avaliações sem a devida autorização, verificar o IP da máquina que solicitou o agendamento, com a máquina que irá realizar o acesso remoto para realizar a avaliação.

Capítulo 3

Estrutura Analítica do Projeto e Proposta de Arquitetura do PSPE

3.1 Estrutura Analítica do Projeto

Na Figura 3.1, é possível acompanhar o desenvolvimento de todo o projeto do Portal Seguro de Provas Eletrônicas - PSPE, podendo analisar a ordem cronológica de cada atividade.

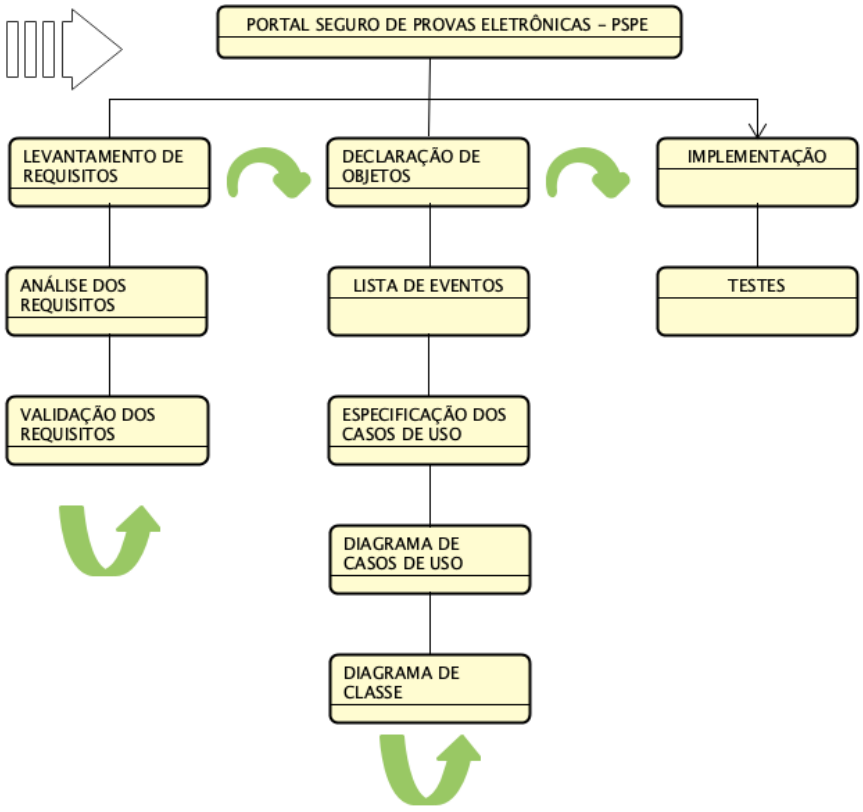

Figura 3.1: Estrutura Analítica do Projeto

3.2 Especificação dos Requisitos

3.2.1 Contexto do Projeto

Este projeto deverá auxiliar na integração entre professores e acadêmicos durante a escolha da sua avaliação a ser realizada de forma remota ou presencial. O objetivo principal será notificar o acadêmico para todo tipo de atividade que seja de seu interesse, desde uma simples notificação de quais agendamentos de disciplinas foram realizados pelos acadêmicos e até avisos gerais de cancelamentos de avaliações agendadas, mudanças de local de avaliação (no caso de avaliação presencial), dentre outros avisos necessários para uma excelente interação entre o acadêmico e devida avaliação previamente agendada no PSPE.

3.2.2 Requisitos Não Funcionais do PSPE

Estão disciminadas nesta seção os aspectos relacionados à segurança do PSPE:

RNF1: Somente usuários cadastrados ou utilizando login do Google terão acessos ao Sistema.

RNF2: Todos os campos do cadastro/login são de preechimentos obrigatórios.

RNF3: Todos os campos para a solicitação de Agendamento de Prova são de preenchimentos obrigatórios.

RNF4: Para a utilização do PSPE via VPN, todas as credenciais de acesso deverão ser validadas e preenchidas conforme informe recebidos no e-mail do usuário.

3.2.3 Requisitos Funcionais - Professor

Estão discriminadas nesta seção os itens levantados e posteriormente analisados. Onde será possível gerenciar os dados de comunicação ao acadêmico, sendo assim os requisitos são:

RF1: O professor terá acesso ao envio de avisos no PSPE da seguinte forma: poderá inserir um aviso escolhendo a quem deseja comunicar, aos acadêmicos somente da disciplina que foi agendada, do curso no qual ele ministra as disciplinas, da unidade ou de toda Universidade. Também será possível excluir e editar um aviso que já esteja em circulação. Os mesmos serão vistos em uma tabela na forma de relatórios, para que os professores tenham controle dos avisos enviados. Para o envio de cada aviso, será necessário o título, texto e a quem deseja comunicar.

RF2: O professor poderá manipular/enviar as avaliações que lecionam, da seguinte forma: inserir, editar e excluir uma avaliação. Para inserir uma avaliação, o professor precisará informar o nome da disciplina, pontuação da diciplina, quantidades de questões e sua pontuação, e o turno da diciplina, sendo vespertino, matutino ou noturno, e as questões que irão compor a referida avaliação. As disciplinas serão mostradas na forma de tabela, para que os professores possam alterar dados assim que necessário.

RF3: O professor poderá visualizar os agendamentos de avaliações das suas disciplinas. Alterar o início de cada período da avaliação. Nesta mesma tela, o professor poderá acompanhar na forma de relatório quais são os acadêmicos que tem agendamento válido para as suas diciplinas.

RF4: Após o cadastro das avaliações nas devidas disciplinas, o professor poderá informar as notas de cada um de acordo com as correções, pois podem ter questões subjetivas. Para isso, o professor precisará informar qual disciplina deseja liberar as notas, para aqueles que lecionam mais de uma, posteriormente um relatório de todos acadêmico cadastrados estará disponível na tela. A cada nota inserida, o acadêmico receberá uma notificação que sua nota já está disponível. As notas a serem lançadas, será baseada na quantidade de questões e pontuação/peso da referida avaliação que o professor informou quando cadastrou a avaliação. O professor poderá inserir, editar e excluir uma nota.

3.2.4 Requisitos Funcionais - Acadêmico

Nesta sessão, estão discriminados os itens que cumpriram os requisitos para compor o gerenciamento dos dados pelos acadêmicos. Sendo assim, os itens são:

RF1: Após o acadêmico acessar o PSPE, ele irá informar alguns dados para que seu registro seja efetuado ou utilizar o login do Google para se registrar, mas deverá completar o cadastro informando a matrícula.

RF2: Feito o login no PSPE, o acadêmico irá confirmar seus dados, nome, email e contatos.

RF3: Após a conclusão dos requisitos RF1 e RF2, os acadêmicos estarão prontos para utilizar o PSPE, nele poderá acompanhar e visualizar todos os dados necessários e que competem. Poderá agendar as suas avaliações e verificar as notas que tiraram nas devidas avaliações.

RF4: A cada ação realizada pelo professor no PSPE, o acadêmico estará sujeito a receber notificações de acordo com a atividade que o professor executou. Se o professor cadastrar um aviso, os respectivos acadêmicos a quem ele direcionar este aviso, irá receber uma notificação.

RF5: O acadêmico poderá acompanhar algumas informações referente ao curso no menu de Avisos.

3.3 Visão de Caso de Uso

Um caso de uso representa uma funcionalidade requirida pelo usuário. Portanto a visão de casos de uso, que inclui a descrição dos professores e acadêmicos, o diagrama de caso de uso e as especificações dos casos de uso, representa a interação entre os usuários e as funcionalidades que o PSPE fornecerá.

3.3.1 Modelo de Casos de Uso

Um modelo de acaso de uso é um modelo que descreve como diferentes tipos de usuários interagem com o PSPE. Tem o objetivo de auxiliar no processo de desenvolvimento do PSPE.

3.3.2 Diagrama de Casos de Uso

Na figura 3.2 é representado o Diagrama de Caso de Uso do PSPE, onde temos o Sistema Acadêmico sendo um Ator fundamental para as validações destes usuários do Sistema (Professor/Acadêmico) e listagem das disciplinas que o acadêmico está matriculado. O Ator Acadêmico além de visualizar os aviso cadastrados, editados e excluídos pelos Professores, também realiza os agendamentos das avaliações, realizam as avaliações e também visualizam as notas que foram lançadas/liberadas pelos Professores. Já o Ator Professor fica responsável por Gerenciar os Avisos que são enviados aos Acadêmicos, Gerenciar as Avaliações, Agendamentos e Notas para serem liberadas ou lancçadas referentes às avaliaçoões realizadas. Demais detalhamentos estão na Especificação de Casos de Uso.

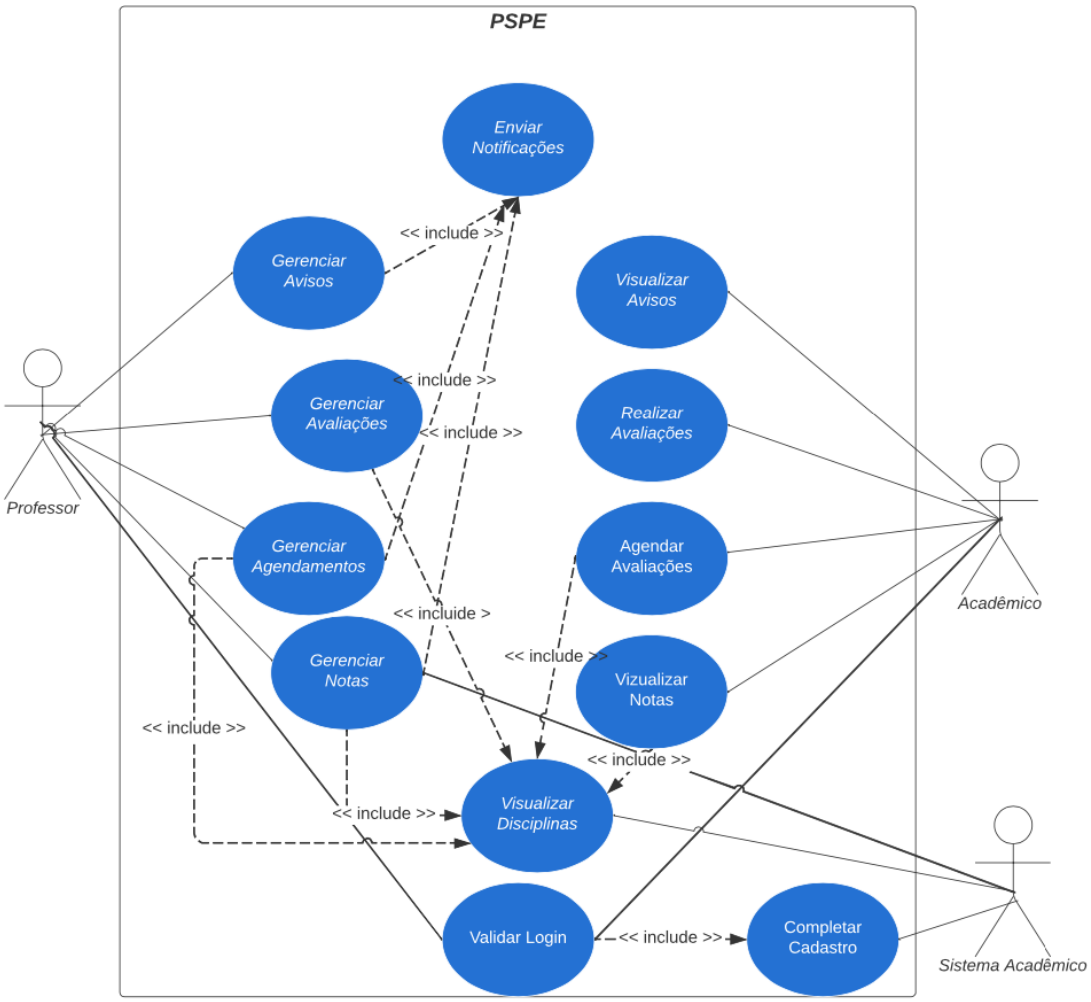

Figura 3.2: Diagrama de Caso de Uso - PSPE

3.3.3 Especificacão de Casos de Uso

- **Caso de Uso - Gerenciar Avisos**
- Descrição: permite que o Professor possa cadastrar, visualizar, editar e excluir um aviso.
- Ator: Professor.
- Fluxo de Eventos: o caso de uso começa quando o Professor seleciona o sub-menu Avisos no menu principal.
- Fluxo Básico:
    1. Professor cadastra novo aviso para os acadêmicos.
    2. Professor salva os dados do aviso.
    3. Acadêmico recebe notificação de um novo aviso.
    4. Sistema emite mensagens de dados do aviso salvo com sucesso.
- Fluxos Alternativos

    A.1 - Editar Aviso

    A.1.1 - Seleciona aviso que deseja editar.

    A.1.2 - Altera os dados necessários.

    A.1.3 - Professor salva os dados do aviso.

    A.1.4 - Acadêmico recebe notificação de alterações.

    A.1.5 - Sistema emite mensagem que os dados foram atualizado com sucesso.

    A.2 - Excluir Aviso

    A.2.1 - Seleciona aviso que deseja excluir.

    A.2.2 - Aviso é excluído.

    A.2.3 - Acadêmico recebe notificação de alterações.

    A.2.4 - Sistema emite mensagem que o Aviso foi excluído com sucesso.

- **Caso de Uso - Gerenciar Avaliações**
- Descrição: permite que o Professor possa cadastrar, visualizar, editar e excluir uma avaliação.
- Ator: Professor.
- Fluxo de Eventos: o caso de uso começa quando o Professor seleciona o sub-menu Avaliações no menu principal.

- Fluxo Básico:
    1. Professor cadastra nova Avaliação para os acadêmicos.
    2. Professor salva os dados da Avaliação.
    3. Sistema emite mensagens de dados da Avaliação salva com sucesso.
- Fluxos Alternativos

    A.1 - Editar Avaliação

    A.1.1 - Seleciona a Avaliação que deseja editar.

    A.1.2 - Altera os dados necessários.

    A.1.3 - Professor salva os dados da Avaliação.

    A.1.4 - Sistema emite mensagem que os dados foram atualizado com sucesso.

    A.2 - Excluir Avaliação

    A.2.1 - Seleciona a Avaliação que deseja excluir.

    A.2.2 - Avaliação é excluída.

    A.2.3 - Sistema emite mensagem que a Avaliação foi excluída com sucesso.

- **Caso de Uso - Gerenciar Agendamentos**
- Descrição: permite que o Professor possa visualizar, editar e excluir um Agendamento.
- Ator: Professor.
- Fluxo de Eventos: o caso de uso começa quando o Professor seleciona o sub-menu Agendamento no menu principal.
- Fluxo Básico:
    1. Professor visualiza os Agendamentos realizados pelos acadêmicos.
- Fluxos Alternativos

    A.1 - Editar Agendamento

    A.1.1 - Seleciona Agendamento que deseja editar.

    A.1.2 - Altera os dados necessários.

    A.1.3 - Professor salva os dados do Agendamento.

    A.1.4 - Acadêmico recebe notificação de alterações.

    A.1.5 - Sistema emite mensagem que os dados foram atualizado com sucesso.

    A.2 - Excluir Agendamento

    A.2.1 - Seleciona Agendamento que deseja excluir.

    A.2.2 - Agendamento é excluído.

A.2.3 - Acadêmico recebe notificação de alterações.

A.2.4 - Sistema emite mensagem que o Agendamento foi excluído com sucesso.

- **Caso de Uso - Gerenciar Notas**
- Descrição: permite que o Professor possa liberar, cadastrar, visualizar, editar e excluir uma Nota referente a uma disciplina.
- Ator: Professor.
- Fluxo de Eventos: o caso de uso começa quando o Professor seleciona o sub-menu Notas no menu principal e na sequência seleciona uma disciplina.
- Fluxo Básico:
    1. Professor libera/cadastra nova Nota para os acadêmicos.
    2. Professor salva os dados da Nota.
    3. Acadêmico recebe notificação de um nova Nota liberada/cadastrada.
    4. Sistema emite mensagens de dados da Nota liberada/cadastrada com sucesso.
- Fluxos Alternativos

    A.1 - Editar Nota

    A.1.1 - Seleciona Nota que deseja editar.

    A.1.2 - Altera os dados necessários.

    A.1.3 - Professor salva os dados da Nota.

    A.1.4 - Acadêmico recebe notificação de alterações.

    A.1.5 - Sistema emite mensagem que os dados foram atualizado com sucesso.

    A.2 - Excluir Nota

    A.2.1 - Seleciona Nota que deseja excluir.

    A.2.2 - Nota é excluído.

    A.2.3 - Acadêmico recebe notificação de alterações.

    A.2.4 - Sistema emite mensagem que a Nota foi excluída com sucesso.

- **Caso de Uso - Enviar Notificações**
- Descrição: permite que o PSPE possa Enviar Notificações ao Acadêmicos.
- Atores: Professor, Sistema PSPE.
- Caso de uso envolvidos: Gerenciar Avisos, Gerenciar Agendamentos e Gerenciar Notas.

- Fluxo de Eventos: o caso de uso começa quando o Professor seleciona o sub-menu Avisos no menu principal e faz um cadastro, altera ou exlcui um novo aviso, altera ou exclui um agendamento ou libera, cadastra, altera ou exclui uma Nota.

- Fluxo Básico:

    1. Professor cadastra novo aviso ou libera/cadastra Nota para os acadêmicos.
    2. Professor salva os dados.
    3. Acadêmico recebe notificação de um novo aviso, ou nota liberada/cadastrada.
    4. Sistema emite mensagens de dados do aviso saldo, nota salva/liberada com sucesso.

- Fluxos Alternativos

    A.1 - Editar Aviso, Agendamento ou Nota.

    A.1.1 - Seleciona aviso/nota que deseja editar.

    A.1.2 - Altera os dados necessários.

    A.1.3 - Professor salva os dados do aviso/nota.

    A.1.4 - Acadêmico recebe notificação de alterações.

    A.1.5 - Sistema emite mensagem que os dados foram atualizado com sucesso.

    A.2 - Excluir Aviso, Agendamento ou Nota.

    A.2.1 - Seleciona aviso, ou Agendamento ou nota que deseja excluir.

    A.2.2 - Aviso é excluído.

    A.2.3 - Acadêmico recebe notificação de alterações.

    A.2.4 - Sistema emite mensagem que o Aviso, Agendamento ou Nota foi excluída com sucesso.

- **Caso de Uso - Visualizar Disciplinas**

- Descrição: permite que o Professor possa visualizar as Disciplinas que contém Acadêmicos com avaliações agendadas e Acadêmicos selecionar as diciplinas que estão cursando para realizar os agendamentos das avaliações.

- Atores: Professor, Acadêmico, Sistema Acadêmico.

- Caso de uso envolvidos: Gerenciar Agendamentos, Gerenciar Avaliações, Agendar Avaliações, Visualizar Notas e Gerenciar Notas.

- Fluxo de Eventos: o caso de uso começa quando o Professor seleciona o sub-menu Avaliações, Agendamentos ou Notas no menu principal ou Acadêmico realiza um Agendamento de Avaliação.

- Fluxo Básico:

    1. Professor cadastra/altera/exlclui uma avaliação no PSPE, Professor altera/exclui um Agendamento e o Acadêmico realiza um agendamento de avaliação.

    2. Professor/Acadêmico salva os dados.

    3. Professor/Acadêmico recebe notificação.

    4. Sistema emite mensagens de dados salvo com sucesso.

- Fluxos Alternativos: N/A

- **Caso de Uso - Validar Login**

- Descrição: permite que Professores e Acadêmicos validem os seus acessos ao PSPE.

- Atores: Professor, Acadêmicos.

- Caso de uso envolvidos: Completar Cadastro.

- Fluxo de Eventos: o caso de uso começa quando o Professor/Acadêmico realiza um login no PSPE e o sistema solicita matrícula para Validar os seus respectivos logins chamando o caso de uso Completar Cadastro.

- Fluxo Básico:

    1. Professor/Acadêmico realiza login no PSPE.

    2. Caso de Uso Completar Cadastro solicita a matrícula.

    3. Professor/Acadêmico informa matrícula.

    4. Professor/Acadêmico recebe notificação que cadastro foi completado com sucesso.

- Fluxos Alternativos

    A.1 - Professor/Acadêmico informa matrícula errada.

    A.1.1 - PSPE informa que matrícula é inexistente.

    A.2 - Professor/Acadêmico informa matrícula e não existe disciplina vinculada a matrícula informada.

    A.2.1 - PSPE informa que matrícula não tem disciplina vinculada.

- **Caso de Uso - Completar Cadastro**

- Descrição: permite que Professores e Acadêmicos completem os seus cadastros para realizar Login no PSPE.

- Atores: Professor, Acadêmicos.

- Caso de uso envolvidos: Validar Login.

- Fluxo de Eventos: o caso de uso começa quando o Professor/Acadêmico realiza um login no PSPE e o sistema solicita matrícula para completar os cadastros para realizarem os logins no PSPE.
- Fluxo Básico:
    1. Professor/Acadêmico realiza login no PSPE.
    2. Caso de Uso Completar Cadastro solicita a matrícula.
    3. Professor/Acadêmico informa matrícula.
    4. Professor/Acadêmico recebe notificação que cadastro foi completado com sucesso.
- Fluxos Alternativos

    A.1 - Professor/Acadêmico informa matrícula errada.

    A.1.1 - PSPE informa que matrícula é inexistente.

    A.2 - Professor/Acadêmico informa matrícula e não existe disciplina vinculada a matrícula informada.

    A.2.1 - PSPE informa que matrícula não tem disciplina vinculada.

- **Caso de Uso - Visualizar Avisos**
- Descrição: permite que o Acadêmico possa visualizar os avisos.
- Ator: Acadêmico.
- Fluxo de Eventos: o caso de uso começa quando o Acadêmico seleciona o sub-menu Avisos no menu principal.
- Fluxo Básico:
    1. Acadêmico visualiza o aviso.
- Fluxo Alternativo

    A.1 - Não existe aviso ao Acadêmico.

    A.1.1 - Sistema emite mensagem que não existe Aviso para o Acadêmico.

- **Caso de Uso - Agendar Avaliações**
- Descrição: permite que o Acadêmico possa Agendar e visualizar uma avaliação agendada.
- Ator: Acadêmico.
- Casos de Uso envolvidos: Visualizar Disciplinas.
- Fluxo de Eventos: o caso de uso começa quando o Acadêmico seleciona o sub-menu Agendar Avaliações no menu principal.

- Fluxo Básico:

    1. Acadêmico seleciona uma disciplina para agendar nova Avaliação.

    2. Acadêmico marca o Modo de Realização da Prova (Presencial)

    3. Acadêmico Selecione a data para agendamento da Avaliação.

    4. Acadêmico Selecione o Horário disponível para agendamento da Avaliação.

    5. Acadêmico clica em Agendar Avaliação.

    6. PSPE emite mensagem informando que o Agendamento da Avaliação realizada com sucesso e gera o QR CODE para a Avaliação agendada.

    7. PSPE envia o email com o QR CODE e TOKEN para a Avaliação agendada, sala e horário.

- Fluxos Alternativos

    A.1 - Acadêmico marca o Modo de Realização da Prova (Remoto)

    A.1.1 - Acadêmico Selecione a data para agendamento da Avaliação.

    A.1.2 - Acadêmico Selecione o Horário disponível para agendamento da Avaliação.

    A.1.3 - Acadêmico clica em Agendar Avaliação.

    A.1.4 - PSPE emite mensagem informando que o Agendamento da Avaliação realizada com sucesso e gera o TOKEN para a Avaliação agendada.

    A.2 - PSPE informa que não existe avaliação para a diciplina selecionada.

    A.2.1 - Sistema emite mensagem que não existe avaliação para a diciplina selecionada.

- **Caso de Uso - Realizar Avaliações [Presencial]**

- Descrição: permite que o Acadêmico possa realizar as avaliações agendadas no Modo Presencial.

- Ator: Acadêmico.

- Fluxo de Eventos: o caso de uso começa quando o Acadêmico seleciona o sub-menu Realizar Avaliações no menu principal no Laboratório Agendado para a realização da avaliação.

- Fluxo Básico [Presencial]:

    1. Acadêmico apresenta o QR CODE na entrada do Laboratório para realizar a Avaliação no Laboratório agendado pelo PSPE.

    2. Acadêmico entra com matrícula e TOKEN que recebeu no e-mail para acessar o PSPE no módulo de Realizar Avaliação na máquina agendada.

    3. Acadêmico realiza avalicação.

- Fluxos Alternativos

  A.1 - Acadêmico tenta acessar o sub-menu Realizar Avaliações em outra máquina que não seja do Laboratório agendado pelo PSPE.

  A.1.1 - Sistema emite mensagem que não existe avaliação agendada nesta máquina/laboratório, favor conferir e-mail recebido do PSPE.

  A.2 - Acadêmico tenta acessar o sub-menu Realizar Avaliações no Laboratório e máquina agendada e digita matrícula/token errado.

  A.2.1 - Sistema emite mensagem que matrícula/token está errado, favor conferir e-mail recebido do PSPE.

- **Caso de Uso - Realizar Avaliações [Remoto]**

- Descrição: permite que o Acadêmico possa realizar as avaliações agendadas no Modo Remoto.

- Ator: Acadêmico.

- Fluxo de Eventos: o caso de uso começa quando o Acadêmico seleciona o sub-menu Realizar Avaliações no menu principal utilizando as credenciais enviadas via e-mail para acessar o Laboratório agendado via VPN.

- Fluxo Básico [Remoto]:

  1. Acadêmico entra com matrícula e TOKEN que recebeu no e-mail para acessar o PSPE no módulo de Realizar Avaliação na máquina agendada acessando pela VPN.

  2. Acadêmico realiza avalicação.

- Fluxos Alternativos

  A.1 - Acadêmico tenta acessar o sub-menu Realizar Avaliações em outra máquina que não seja do Laboratório agendado pelo PSPE.

  A.1.1 - Sistema emite mensagem que não existe avaliação agendada nesta máquina/laboratório, favor conferir e-mail recebido do PSPE.

  A.2 - Acadêmico tenta acessar o sub-menu Realizar Avaliações no Laboratório e máquina agendada via VPN e digita matrícula/token errado.

  A.2.1 - Sistema emite mensagem que matrícula/token está errado, favor conferir e-mail recebido do PSPE.

- **Caso de Uso - Visualizar Notas**

- Descrição: permite que o Acadêmico possa visualizar as Notas das avaliações realizadas no PSPE e lançadas/liberadas pelo Professor.

- Atores: Acadêmico.

- Caso de uso envolvidos: Gerenciar Avaliações, Agendar Avaliações e Gerenciar Notas.
- Fluxo de Eventos: o caso de uso começa quando o Acadêmico seleciona o sub-menu Notas do PSPE.
- Fluxo Básico:
    1. Acadêmico visualiza notas lançadas/liberadas.
    2. Sistema mostra as notas lançadas/liberadas pelo Professor.
- Fluxo Alternativo:

    A.1 - Acadêmico tenta acessar o sub-menu Notas e não tem nenhuma nota lançadas/liberadas pelo Professor.

    A.1.1 - Sistema emite mensagem que não existe notas lançadas/liberadas para nenhuma disciplina.

3.3.4 Diagrama de Classes

Um Diagrama de Classes é um diagrama utilizado nas modelagens de sistemas Orientados a Objeto (OO). Neste Diagrama é possível visualizar um conjunto de informações como, conjunto de classes principais que atua diretamente no sistema, conjunto de interfaces e colaboração e seus relacionamentos.

Na figura 3.3 será mostrado as principais classes para a construção do PSPE referente ao módulo de Agendamento das Avaliações:

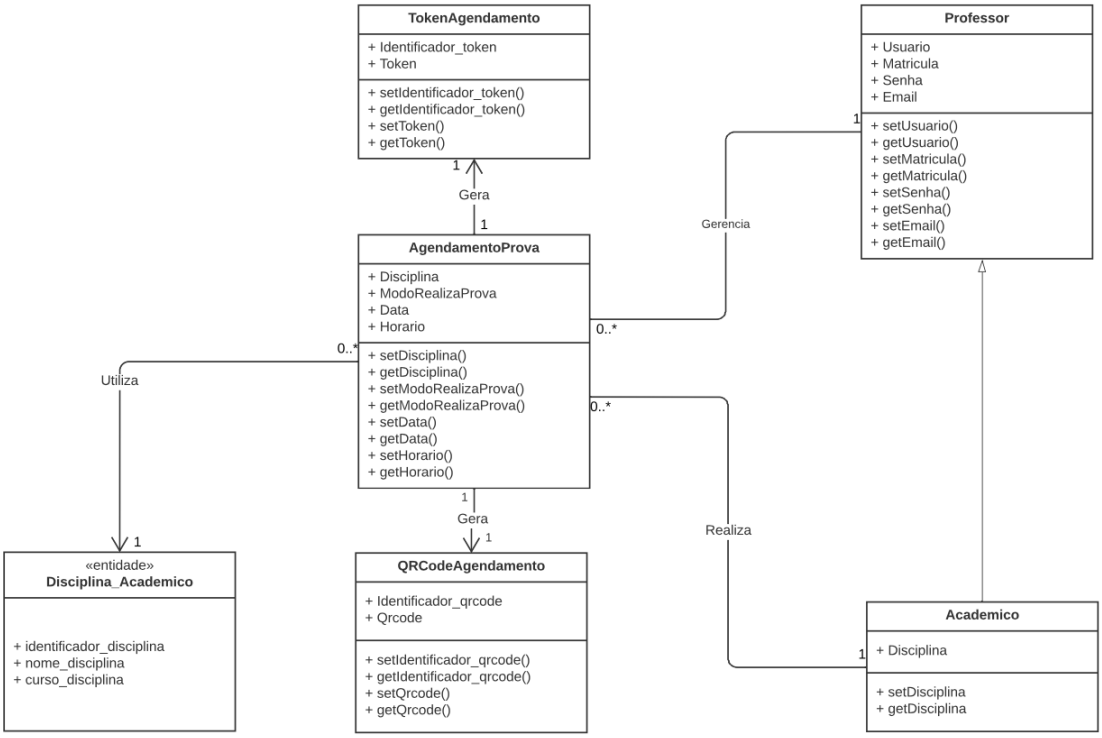

Figura 3.3: Diagrama de Caso de Uso - PSPE

3.4 Proposta de Arquitetura do PSPE

A arquitetura proposta para este Portal Seguro de Provas será de autenticação SSO, geração de um *Token* através de um *QR Code* gerado para a autenticação e validação de um determinado usuário (neste caso, o Acadêmico). Na sequência o Acadêmico irá acessar o Portal de Provas a fim de realizar o agendamento da sua devida avaliação. A ideia central do Portal Seguro de Provas é a autenticação/validação dos usuários através de SSO, que na sequência terá a opção do Acadêmico escolher qual avaliação irá fazer selecionando alguma disciplina que está matriculado e com isso será gerado um *QR Code* no qual o Acadêmico deverá apresentá-lo no laboratório que será realizada a devida avaliação, ou acessar remotamente uma máquina específica para realizar a prova de forma remota. No caso da prova ser realizada de forma presencial, após apresentar o *QR Code* no laboratório previamente agendado para a devida avaliação, este gerenciamento de usuário será resolvido com a utilização de SDN, o módulo de autenticação/validação deste Portal de Provas irá gerar um *Token* válido somente para a avaliação solicitada na máquina liberada. Com a utilização da matrícula e o *Token* gerado pelo PSPE e enviado por email para o Acadêmico, ele irá logar no referido Portal de Provas e realizar a devida avaliação. No cenário da prova ser realizada de forma remota, o Acadêmico irá receber as instruções de instalação e configuração do *OpenVPN Connect* para realizar a referida avaliação de forma remota.

Nesta arquitetura proposta, para ficar mais seguro, o *Token* enviado para o Acadêmico que será validado somente em uma máquina específica do laboratório, terá um prazo para expirar de 20 minutos após o início do horário agendado, ou seja, se o Acadêmico não realizar o *login* no referido Portal para realizar a sua avaliação nestes 20 minutos de tolerância, ele terá que solicitar

novamente um novo agendamento que poderá mudar de estação de trabalho para a realização da referida avaliação.

Sobre o gerenciamento via SDN, este módulo deverá ter uma configuração para enviar automaticamente as requisições dos Acadêmicos para os destinos (máquinas nos laboratórios) através do Plano de Controle de um controlador SDN com sua Tabela de Roteamento (Camada de Controle), utilizando como protocolo de comunicação o *OpenFlow*. Este controlador SDN enviará as informações para os *switches* (Camada de Infraestrutura) que comunicarão com as máquinas definidas para a realização de cada avaliação e na hora da realização da Prova, ao digitar a matrícula e *token* recebido através do email, serão confirmados na tabela do Plano de Controle SDN utilizando API (*Application Programming Interface*) para comunicar com o Portal (Camada de Aplicação) conforme é mostrado na figura 3.4. Na Camada de Aplicação ainda temos os módulos de Autenticação SSO, de Geração do *QR Code* e *Token*.

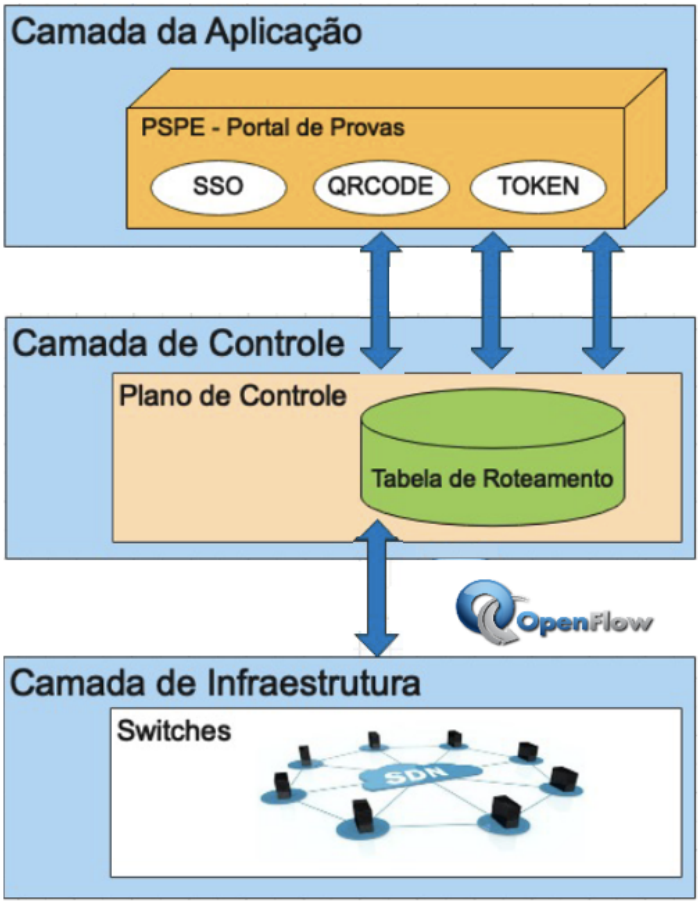

Figura 3.4: Arquitetura Proposta para o Portal de Provas

Capítulo 4

Validação e Discussão dos Resultados

Para realizar a validação dos resultados, serão demonstradas as configurações necessárias para o funcionamento do PSPE.

4.1 Configurando a API ASP.NET *Core Identity*

Para o gerenciamento de usuários, senhas, dados de perfil, funções e *tokens* do Portal de Provas foi utilizado a API ASP.NET *Core Identity*. Os usuários podem criar uma conta com as informações de logon armazenadas no Identity ou podem usar um provedor de logon externo (Google ou Facebook).

A plataforma Microsoft *Identity* é uma evolução da plataforma de desenvolvedor do *Azure Active Directory* (AD do *Azure*), e para o Portal de Provas estamos utilizando para uma autenticação e autorização mais segura, a API *Web IdentityServer4*, que é um *Framework* .NET baseado no *OpenID Connect* e *OAuth* 2.0 para ASP.NET *Core*. O *IdentityServer4* habilita os seguintes recursos de segurança:

- Autenticação como um serviço;
- *Logon* único/*logoff* (SSO) em vários tipos de aplicativos;
- Controle de acesso para APIs;
- Emissão de *tokens* de identidade;
- Validação de *tokens*;

4.2 Configurando o Banco de Dados *NoSQL Firebase* para o Portal de Provas

Para salvar estes dados dos usuários foi configurado o armazenamento de Tabelas do Banco de Dados *NoSQL Firebase*. Foram criadas a tabela usuario com os campos nome, matricula e email e a tabela agendamento com os campos disciplina, modorealizacao e datahora conforme apresentado nas figuras 4.1 e 4.2.

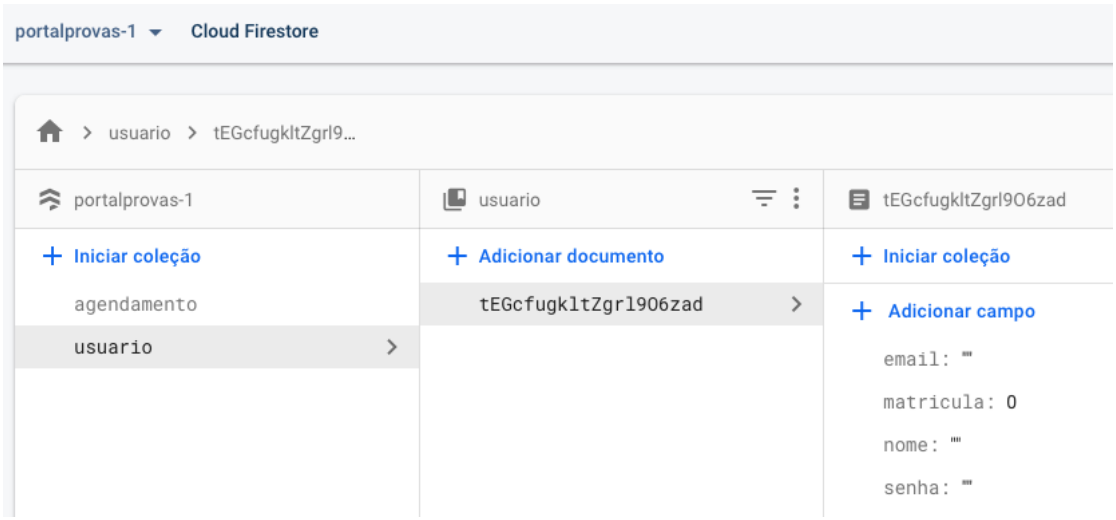

Figura 4.1: Tabela de Usuários do Portal de Provas

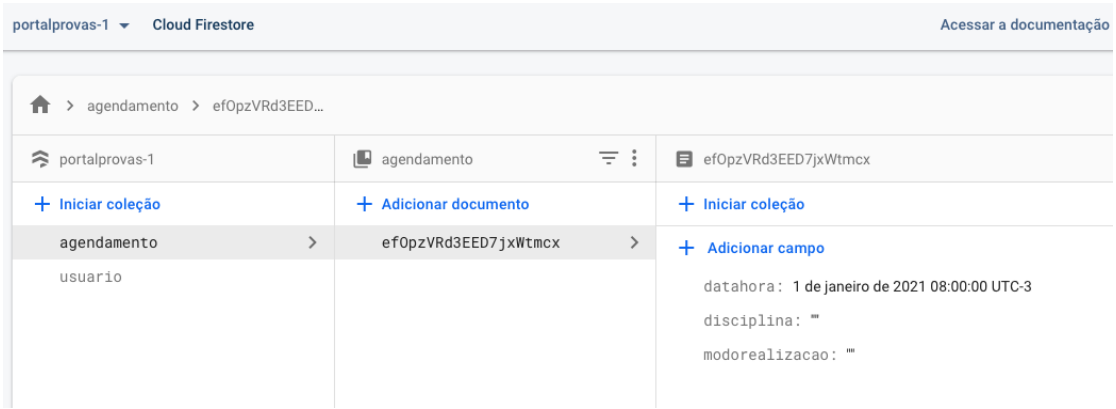

Figura 4.2: Tabela de Agendamentos do Portal de Provas

4.3 Configurando o Login Externo utilizando o Google para acessar o Portal de Provas

Para utilizar os serviços do Google em nome de um usuário quando o usuário está *offline*, temos que utilizar um fluxo do lado do servidor híbrido em que um usuário autoriza seu aplicativo no lado do cliente usando o cliente da API *JavaScript* e será enviado um código de autorização

especial único para o servidor. O servidor troca este código de uso único para adquirir seu próprio acesso e atualizar *tokens* do Google para que o servidor possa fazer suas próprias chamadas de API, o que pode ser feito enquanto o usuário está *offline*. Esse fluxo de código único tem vantagens de segurança em relação a um fluxo puro do lado do servidor e ao envio de tokens de acesso ao servidor.

4.3.1 *Login* do Google para aplicativos do lado do servidor

Para a implementação do fluxo de código único, no caso o botão de *login* do Google, este botão tem que fornecer um *token* de acesso e um código de autorização. O código deverá ser o único que o servidor pode trocar com os servidores do Google por um *token* de acesso. Para isso foram configuradas nas seguintes etapas 4.3:

- Criado um ID e um segredo do cliente no Console de API do Google;
- Incluído a biblioteca da plataforma do Google página de *login* do Portal de Provas;
- Inicializado o objeto *GoogleAuth*;
- Adicionado o botão de *login* à página de *login* do Portal de Provas;

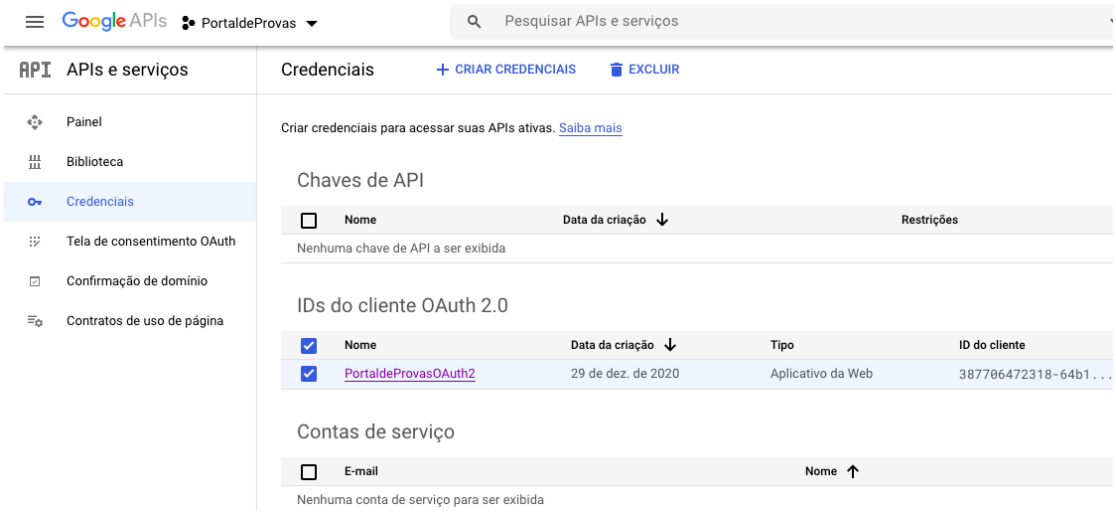

Figura 4.3: Configurando a API do Google para autenticar o Login do Portal de Provas

4.3.2 Configurando as autenticações permitidas para o Portal no Microsoft *Azure*

No menu Autenticação/Autorização do Microsoft *Azure* do Portal de Provas, serão configuradas as autenticações permitidas para acessar o referido Portal. Para o Portal de Provas foi incluído o *login* válido do Google para a devida autenticação e validação dos usuários do Portal de Provas, conforme é mostrado na figura 4.4.

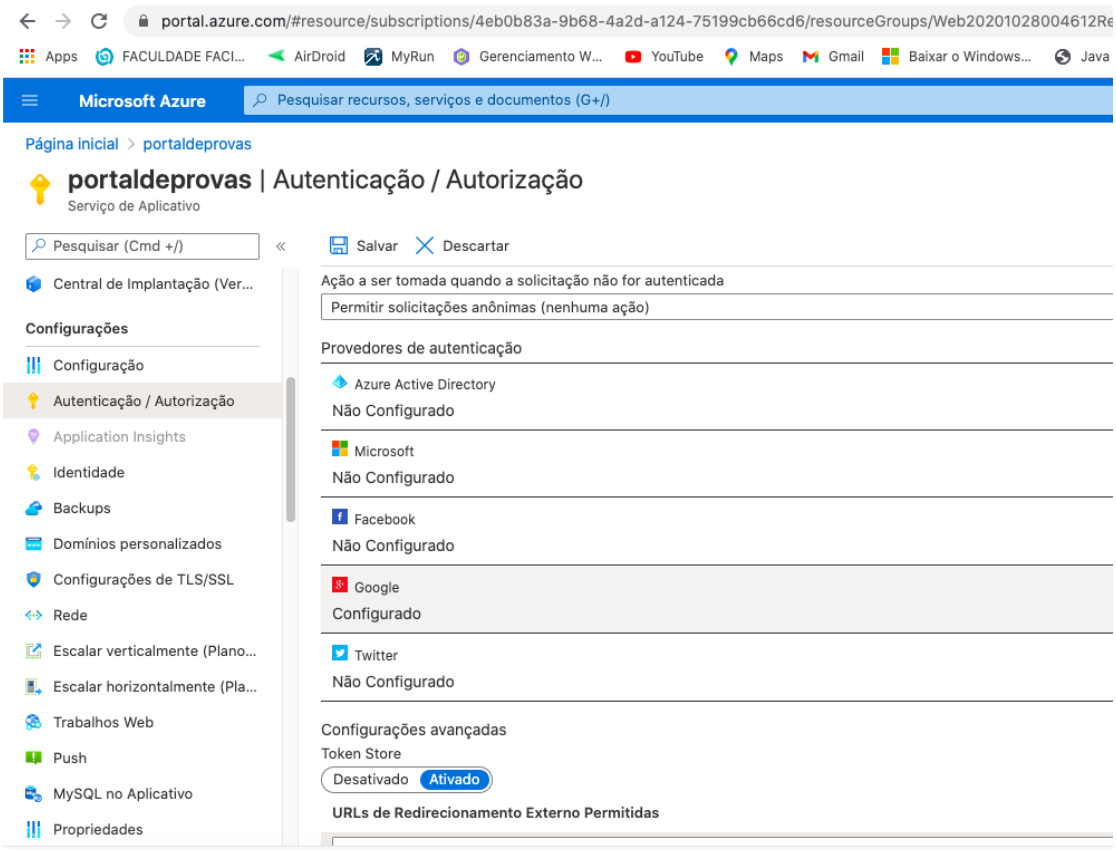

Figura 4.4: Configurando o Google para autenticar o Login do Portal de Provas

4.4 Tela de Cadastro de *Login* para o Portal de Provas

Na arquitetura apresentada, a Tela de Cadastro de *Login* desenvolvida é apresentada na figura 4.5, que tem a função de cadastrar os usuários para o Portal com as devidas validações de *email*, senha e a confirmação da senha. Também será validada nesta tela se as senhas informadas são iguais 4.6. A matrícula é necessária para ser utilizada após o login na listagem das diciplinas que o Acadêmico está matriculado e em caso de ser um Professor, irá listar as disciplinas ministradas.

Figura 4.5: Criando um *login* para autenticar no Portal de Provas

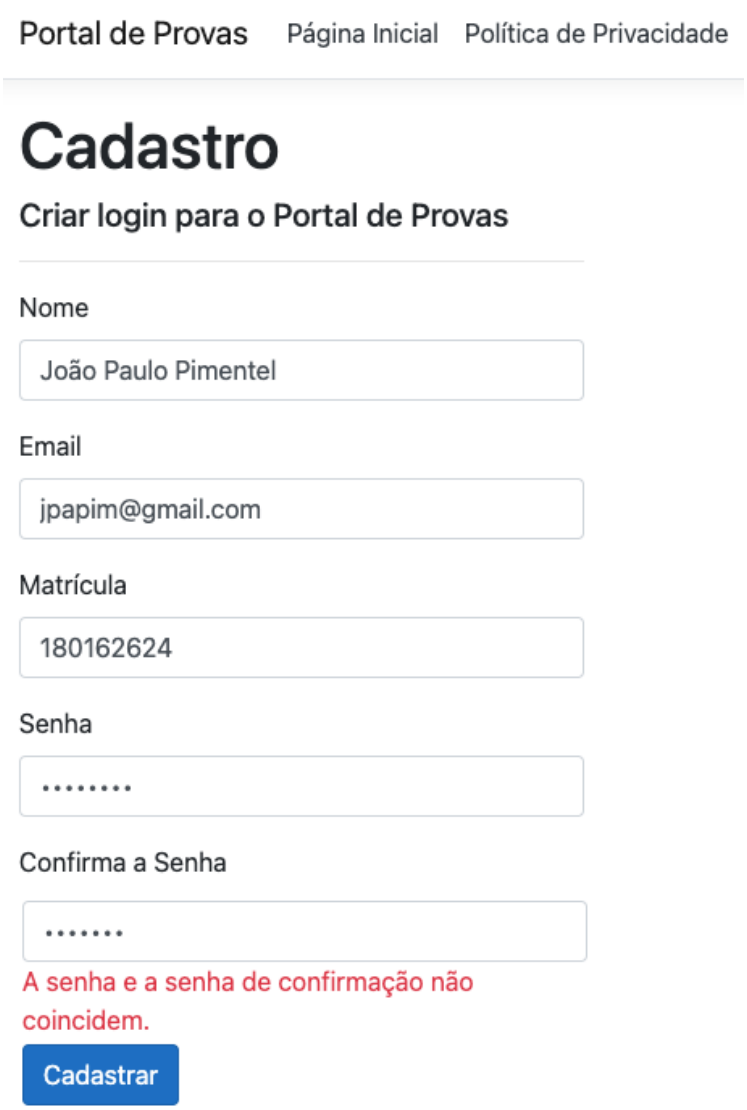

Figura 4.6: Exemplo de tentativa de criação de login para autenticar no Portal de Provas com senhas diferentes

Se as senhas estiverem iguais (Senha e Confirma a Senha) o cadastro é realizado com sucesso 4.7 ou se o Portal de Provas já estiver cadastrado o referido email, enviará a mensagem que o *login* já está em uso.

Figura 4.7: Exemplo de tentativa de criação de *login* para autenticar no Portal de Provas com senhas iguais

4.5 Tela de *Login* do Portal de Provas

A Tela de *Login* é apresentada na figura 4.8 aonde é solicitado o email e senha cadastrados.

Figura 4.8: Tela de *login* para autenticar no Portal de Prova

Outra opção para logar no Portal de Provas é utilizando o seu *login* do Google, conforme é mostrado na figura 4.9, com isso, o usuário é autenticado e automaticamente direcionado para a página de validação do login, no qual é solicitado a Matrícula do Usuário para realizar a vinculação do mesmo com as disciplinas que estão cadastradas no Sistema Acadêmico que o PSPE acessará através de uma API. Para usuários do Tipo Acadêmico, esta API devolverá as disciplinas que estão vinculadas ao Acadêmico, e para usuários do Tipo Professor terá outra API que listará as diciplinas que o Professor ministra naquele semestre. Com isso, o usuário é direcionado para a tela principal do PSPE.

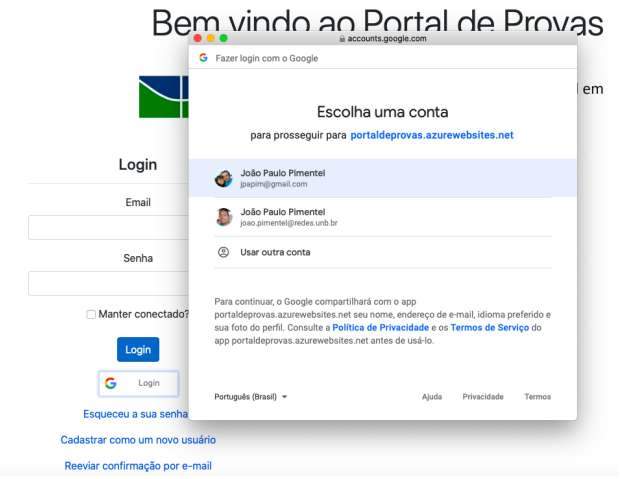

Figura 4.9: Login com o e-mail do Google

Se o usuário logado é do tipo Acadêmico, será direcionado para a página principal mostrada na figura 4.10

Figura 4.10: Tela Principal PSPE - Acadêmico

O agendamento é demostrado na figura 4.11, e após o devido preeenchimento dos dados para agendamento da avaliação (Disciplina, Data do Agendamento e Horário disponível) e clicado no botão "Agendar Avaliação"é gerado um *QR Code* que será utilizado para entrar no Laboratório pré agendado, conforme é mostrado na Figura 4.12.

Figura 4.11: Tela de Agendamento de Avaliação

Este *QR Code* também é enviado para o e-mail do usuário juntamente com o *Token* válido para este agendamento de avaliação.

Figura 4.12: Tela do *QR Code* da Avaliação Agendada com o Laboratório Previsto

Segue exemplo na figura 4.13 do email que o Acadêmico irá receber com o *Token* válido para o agendamento da avaliação solicitada no PSPE.

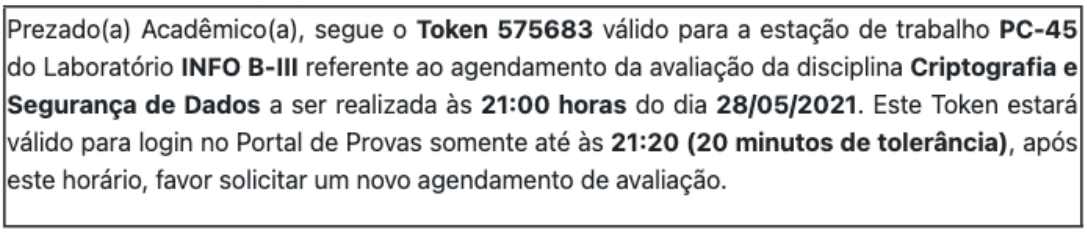

Figura 4.13: Tela do *Token* enviado ao Acadêmico por email

Com o *Token* para realizar a avaliação, será necessário que o Acadêmico vá ao Laboratório e Estação de Trabalho Agendados e entre com as credenciais de Matrícula e Token recebido, conforme é mostrado na figura 4.14

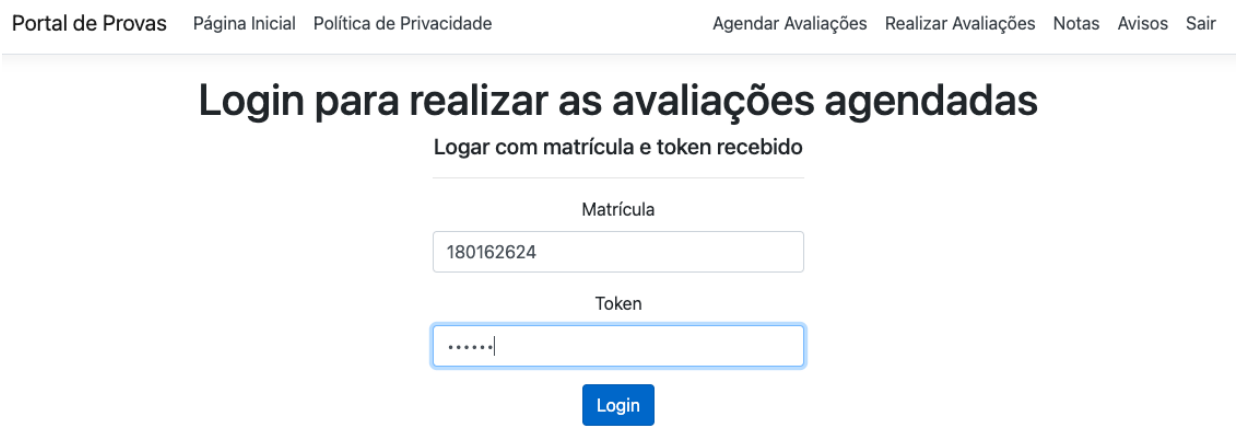

Figura 4.14: Tela de Login para a realização das avaliações agendadas

Se o usuário logado é do tipo Professor, será direcionado para a página principal mostrada na figura 4.15

Figura 4.15: Tela Principal PSPE - Professor

Após o Professor logar no PSPE ele poderá visualizar, editar e excluir os agendamentos realizados pelos Acadêmico clicando no menu "Agendamentos". Nesta opção temos um filtro para pesquisar estes agendamentos por matrícula, nome do aluno, disciplina, modo de realização da avaliação, data, horário, laboratório e máquina, conforme é mostrado na figura 4.16.

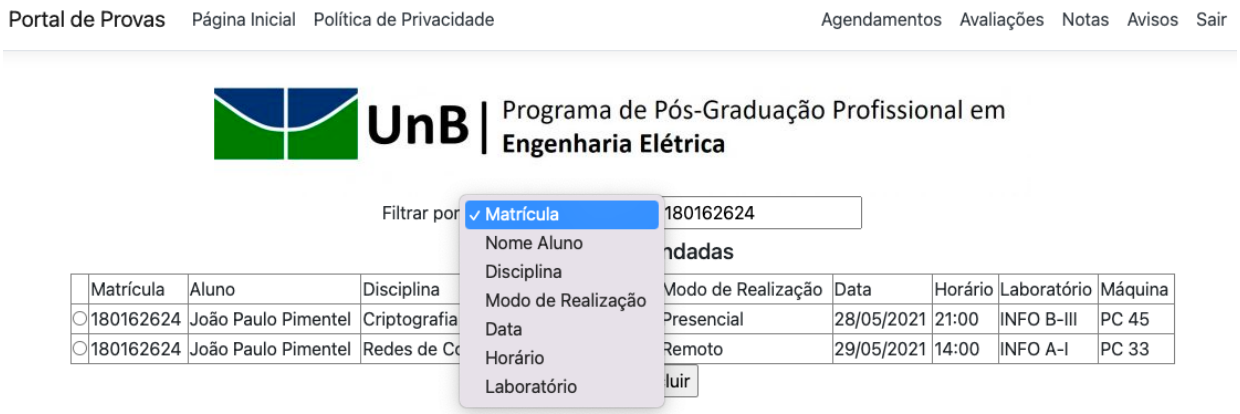

Figura 4.16: Tela de Gerenciamento dos Agendamentos das Avaliações - Professor

Após realizar a seleção do filtro, o Professor pesquisa no campo de texto ao lado. Voltando os resultados, para editar ou excluir, deverá clicar na opção ao lado da Matrícula e acionar o botão correspondente a sua ação, conforme mostrado na figura 4.17.

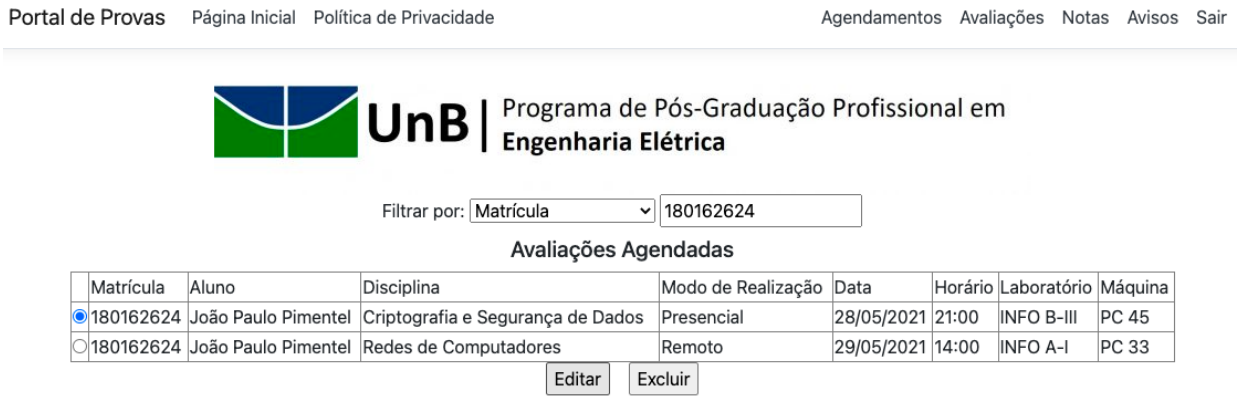

Figura 4.17: Tela de Gerenciamento dos Agendamentos das Avaliações após o filtro

4.6 Configurando o *OpenVPN Connect* para realização de Provas Remotas

A VPN (Rede Virtual Privada) é uma rede de computadores privada virtual que as informações de uma rede local são trafegadas por um túnel através de uma rede pública qualquer, mas são protegidos neste túnel com a utilização de criptografia. No caso do *OpenVPN* que vamos utilizar para os Acadêmicos que escolherem a forma remota de realização da Prova agendada, ele utiliza protocolos criptográficos como o SSL e TLS (Camada 2 e 3 do Modelo OSI). Os dados trafegados neste túnel virtual serão gerenciados e também cuidarão da seguraça da rede por alguns dos protocolos, que podem ser o IPSec (*Internet Protocol Security*), o L2TP (*Layer 2 Tunneling Protocol*), o PPTP (*Point-to-Point Tunneling Protocol*) ou o L2F (*Layer 2 Forwarding*).

O *OpenVPN* utiliza chaves secretas compartilhadas, ou seja, a autenticação de usuários e realizada com senha e a autenticação utilizando certificados. A vantagem na utilização do *OpenVPN* é como os pacotes de dados são transmitidos, podendo utilizar o protocolo UDP ou o TCP. O protocolo UDP é o mais adequado, pois transmite os pacotes de uma forma direta, ou seja, sem repetição, o que é melhor para o desempenho e instabilidade da rede.

Já pensando em garantir uma maior segurança, será instalado junto ao *OpenVPN*, o *software OpenSSL*, para trabalhar com os protocolos SSL e TLS, e têm a função de criptografar. Estes protocolos promovem a integridade e privacidade dos dados entre a comunicação das redes, permitindo assim autenticação das duas partes envolvidas.

Para esta configuração iremos utilizar uma máquina para ser a Matriz (uma máquina de exemplo para ser acessada pelo *OpenVPN)* com o Linux Debian 11 *bullseye* e outra cliente com o Sistema Operacional MAC OS Catalina.

- Instalação do *OpenVPN* e *OpenSSL* na máquina Matriz:
- Comandos:

  apt-get install openvpn openssl

  mkdir /etc/openvpn/easy-rsa (criação de um diretório que receberá os arquivos de configurações que estão em uma diretório padrão)

  cp -rp /usr/share/doc/openvpn/examples/easy-rsa/2.0/. /etc/openvpn/easy-rsa/

  vim /etc/openvpn/easy-rsa/vars (editando o arquivo vars)

  Conteúdo do arquivo:

  export KEY COUNTRY=”BR”

  export KEY PROVINCE=”DF”

  export KEY CITY=”Brasília”

  export KEY ORG=”UnB”

  export KEY EMAIL=joao.pimentel@redes.unb.br

- Salve o arquivo vars e vamos acessar o diretório easy-rsa:
- Comandos:

  cd /etc/openvpn/easy-rsa

  /etc/openvpn/easy-rsa’ source vars (Ativando as modificações)

  /etc/openvpn/easy-rsa ./clean-all (Ativar e criar o diretório /etc/openvpn/easy-rsa/keys para guardar as chaves e certificados do servidor de VPN)

  ./build-ca (Criando o certificado)

  ./build-key-server servidorvpn (Criando a chave)

./build-key jpapim (Chaves de acesso para usuários)

./build-dh

cd /etc/openvpn/easy-rsa/Keys (Acessando diretório das chaves)

openvpn –genkey –secret chave.key (Criação da chave)

mkdir /etc/openvpn/Keys (Criando o diretório keys)

cp -a /etc/openvpn/easy-rsa/keys/chave.key dh1024.pem ca.crt servervpn.crt server.key /etc/openvpn/Keys/ (Copiando as chaves para o diretório Keys)

- Editar o arquivo principal da VPN que fica no diretório /etc/openvpn:
- Comandos:

vim /etc/openvpn/server.conf (Arquivo de configuração da VPN - chamamos ele de server.conf)

Arquivo de configuração do OpenVPN:

proto udp

port 1194

dev tun

server 10.0.0.0 255.255.255.0

push "route 192.168.0.0 255.255.255.0"

push "dhcp-option DNS 192.168.0.1"(IP da sua rede)

route 10.0.0.0 255.255.255.0

comp-lzo

keepalive 10 120

float

ifconfig-pool-persist /etc/openvpn/ipp.txt

max-clients 10

persist-key

persist-tun

log-append /var/log/openvpn.log

verb 6

tls-server

dh /etc/openvpn/keys/dh1024.pem

ca /etc/openvpn/keys/ca.crt

cert /etc/openvpn/keys/servervpn.crt

key /etc/openvpn/keys/servervpn.key

tls-auth /etc/openvpn/keys/chave.key

status /var/log/openvpn.stats

- Após digitar estas linhas, salvar e sair deste arquivo, reiniciar o serviço da OpenVPN:
- Comandos:

  /etc/init.d/openvpn restart

Após reiniciar o serviço da OpenVPN, vamos observar se foi iniciado corretamente, se ocorrer algum erro, vamos verificar os passos anteriores e na sequência utilizar o comando "ifconfig"para verificarmos se está ativa a interface virtual tun0.

Para que as informações trafeguem com segurança pelo tunelamento de forma compactada, facilitando a entrega dos pacotes e ser ágil o fluxo de dados pelo túnel virtual, deve-se instalar o pacote lzop cconforme comando abaixo:

- Instalando o Izop
- Comando:

  apt-get install lzop

Após a conclusão das configurações do Servidor OpenVPN, vamos agora configurar a máquina Cliente que será no Sistema Operacional MacOS Catalina e seguiremos os seguintes passos:

- Configurando a máquina Cliente:
- Comando:

  Baixar o software da OpenVPN e instalar;

  Criar em ./OpenVPN/config/ o diretório “keys” para colocar os certificados;

  Copiar os arquivos que estão no servidor no diretório /etc/openvpn/easy-rsa/Keys, são eles: dh1024.pem, ca.crt, usuariovpn.crt, usuariovpn.key e chave.key. Após isso, colar estes arquivos no diretório que criamos anteriormente ./OpenVPN/config/keys;

  Dentro de ./OpenVPN/config, devemos criar o arquivo em um editor de texto, chamado cliente;

- O arquivo cliente deve conter as seguintes linhas:

  dev tun

  proto udp

remote 200.200.200.200

port 1194

pull

comp-lzo

keepalive 10 120

float

tls-client

persist-tun

persist-key

remote-cert-tls server

dh keys/dh1024.pem

ca keys/ca.crt

cert keys/seuusuario.crt

key keys/seuusuario.key

tls-auth keys/chave.key

route-method exe

route-delay 2

Após escrever a configuração, salvar o arquivo na extensão ".ovpn". Depois de salvar nessa extensão, o arquivo ficará com o logotipo da OpenVPN.

Depois de executar o OpenVPN.GUI aparecerá um ícone de computadores de tela de cor vermelha na barra de ferramentas. Clicando com o botão direito, confirme "Conectar", irá abrir uma página de conexão. Se o arquivo de configuração estiver certo, o ícone ficará verde ou continuará vermelho, então deve verificar a configuração.

Após todas estas configurações e explicações sobre a definição de uma VPN e de como configurá-la, a nossa VPN para ser utilizada para os alunos realizarem as avaliações de forma remota estará totalmente configurada e funcionando. E com a configuração correta e os certificados protegidos, a VPN pode ser usada com segurança e suas informações entre as redes distantes serão passadas com proteção do tunelamento da VPN.

4.7 A API de comunicação entre as Camadas

As interfaces dos programas de aplicativos definidos por *software* (as SDN APIs) são *APIs RESTful* SDN utilizadas para se comunicar entre o controlador SDN e os serviços e aplicativos em execução na rede. Essas APIs são utilizadas para facilitar a orquestração e automação eficiente da

rede para se alinhar às necessidades de diferentes aplicativos por meio da programação de rede SDN.

A API de comunicação escolhida para ser utilizada entre o Portal de Provas (Módulo de Agendamento) com o SDN foi a *Northbound* API, desenvolvida utilizando o protocolo *Openflow*, ela é a ponte entre o aplicativo desenvolvido (no caso o módulo de autenticação e validacão do usuário do Portal) e o controlador SDN. Os aplicativos podem dizer à rede o que eles precisam (dados, armazenamento , largura de banda e assim por diante) e a rede pode fornecer esses recursos ou comunicar o que tem.

4.8 Preparando o Controlador SDN

O controlador utilizado para esta configuração da rede SND foi o ONOS (*Open Network Operating System*), sendo implementado como um container utilizando o software *Docker* no sistema operacional MAC OS Catalina. A instalação, ativação e verificação do *Docker* no host do Controlador SDN foram realizadas com os seguintes comandos:

- Atualização do repositório, ativação do HTTPS e Certificados:
- Comandos:

  apt-get update

  apt-get install apt-transport-https ca-certificates

  apt-get install docker-engine

  service docker start

  docker version

- Para baixar e executar o container do ONOS:
- Comandos:

  docker pull onosproject/onos

  docker images

  docker run -td -p 8111:8111 -p 80:8181 -p 6644:6644 –name ONOS- Controller onosproject/onos

  docker ps -a

4.9 Preparando o simulador *Mininet* VM no macOS Catalina

Para executar o simulador *Mininet* no macOS, conforme mostrado na Figura 4.18, utilizamos o *VirtualBox* de acordo com as instruções abaixo:

- Primeiramente baixamos a imagem do *Mininet* VM no repositório do Git:

  https://github.com/mininet/mininet/releases

  Para o nosso Projeto foi utilizado a versão 2.2.2 do Mininet.

- Para a nossa simulação foi baixado o VirtualBox para macOS:

  https://www.virtualbox.org/wiki/Downloads

  Para o nosso Projeto foi utilizado a versão 6.1.18 do Oracle *VirtualBox*.

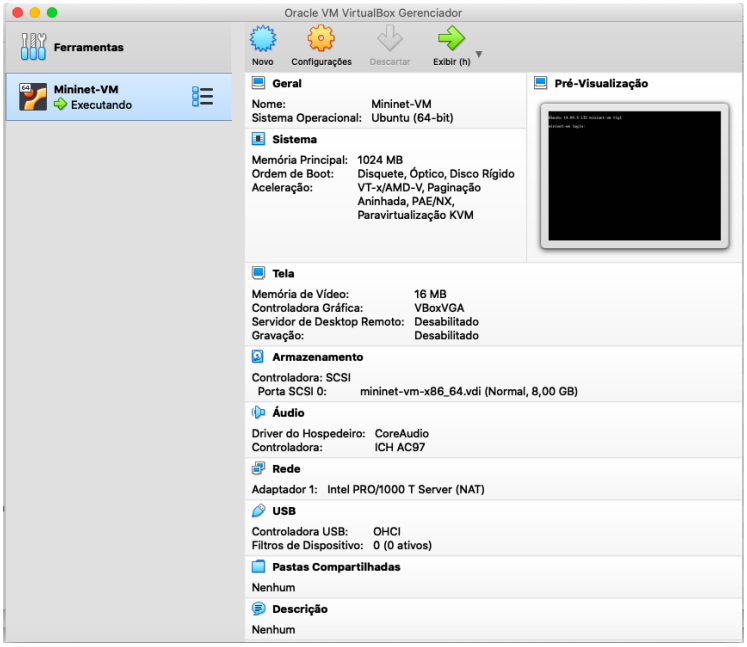

Figura 4.18: Tela do Mininet executando no *Virtula Box*

- Para esta simulação foi realizado o login na VM, usando o seguinte login e senha:

  mininet-vm login: mininet

  password: mininet

- Primeiro, vamos encontrar o endereço IP da VM, que para o VirtualBox provavelmente está no intervalo 192.168.xy no console da VM:

  ifconfig eth0

- Para exibir opções de inicialização com as opções de ajuda do Mininet foi necessário digitar o seguinte comando:

  sudo mn -h

- Interagindo com hosts e switches

  Iniciando uma topologia mínima com o seguinte comando (4.19):

  sudo mn

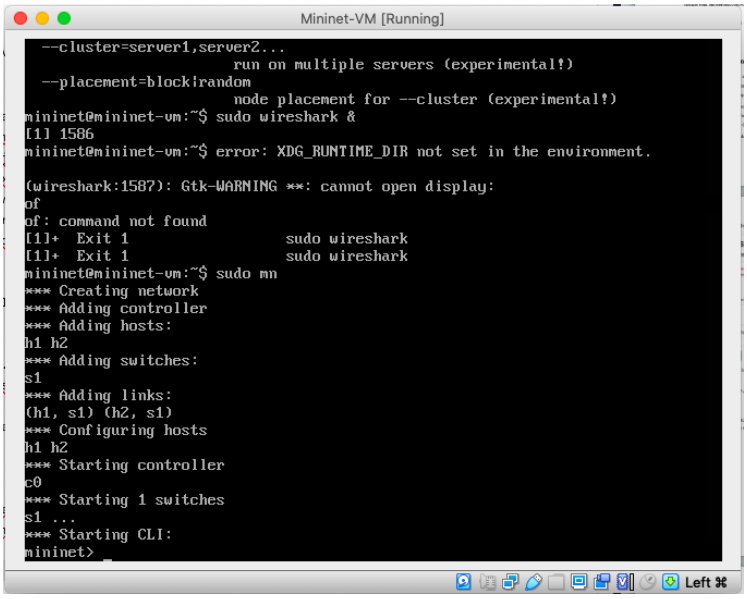

Figura 4.19: Tela do Mininet executando SUDO MN no Virtual Box

A topologia padrão que já vem configurada no *Mininet*, é a minimal, que inclui um *switch kernel OpenFlow* conectado a dois *hosts*, além do controlador de referência *OpenFlow*. Essa topologia também pode ser especificada na linha de comando com o seguinte comando: –topo=minimal. Todos os quatro recursos (2 processos de host, 1 processo de switch, 1 controlador básico) agora estão em execução na VM.

- Comandos do Mininet:

  help (para ajuda nos comandos do Mininet)

  nodes (para exibir os nós de exibição)

  net (exibir os links)

  dump (informações sobre todos os nós)

4.10 Validação dos Resultados

Com o objetivo de validar a proposta deste Portal de Provas descrita nesta dissertação e de gerar parâmetros de avaliação para cada módulo (Autenticação SSO, Geração de *QR Code* e *Toke*) e camada (Aplicação, Controlador SDN e de Infraestrutura) apresentada do Portal, foi desenvolvido a camada de apresentação ao usuário, local este para o referido login do Acadêmico utilizando o Google como autenticação ou através do cadastro pelo usuário para demonstrar as funcionalidades e obter métricas de funcionamento sobre o agendamento das avaliações em cenários reais e utilizando o emulador para representar a comunicação com o Gerenciamento de Redes através de SDN.

Para realizar estes testes e levantamento de dados para resultados desta aplicação foram realizados testes no Laboratório LATITUDE da UnB - Universidade de Brasília primeiramente com 3

máquinas realizando o agendamento das avaliações, conforme mostrado na figura 4.20, o Portal de Provas gerou o QR Code para ser apresentado na entrada do laboratório que foi pré agendado. Este agendamento poderia ser realizado até pelo smartphone do usuário, computador pessoal ou em qualquer lugar, pois este módulo é responsável apenas para o agendamento das disciplinas nos laboratórios de acordo com as datas e horários disponíveis, não sendo obrigatório o agendamento nos Laboratórios da UnB.

Figura 4.20: Testes em 3 Máquinas no Laboratório Latitude da UnB

Este módulo da aplicação demonstrou que a segurança implementada no *login* SSO, conforme mostrado na Figura 4.21, tanto utilizando a plataforma Microsoft *Identity*, assim como utilizando o *login* do Google, demostra que somente uma pessoa irá realizar o login no referido Portal de Provas para realizar o agendamento e terá somente um *QR Code* gerado para cada disciplina por Acadêmico, pois está configurado a matrícula/email logado.

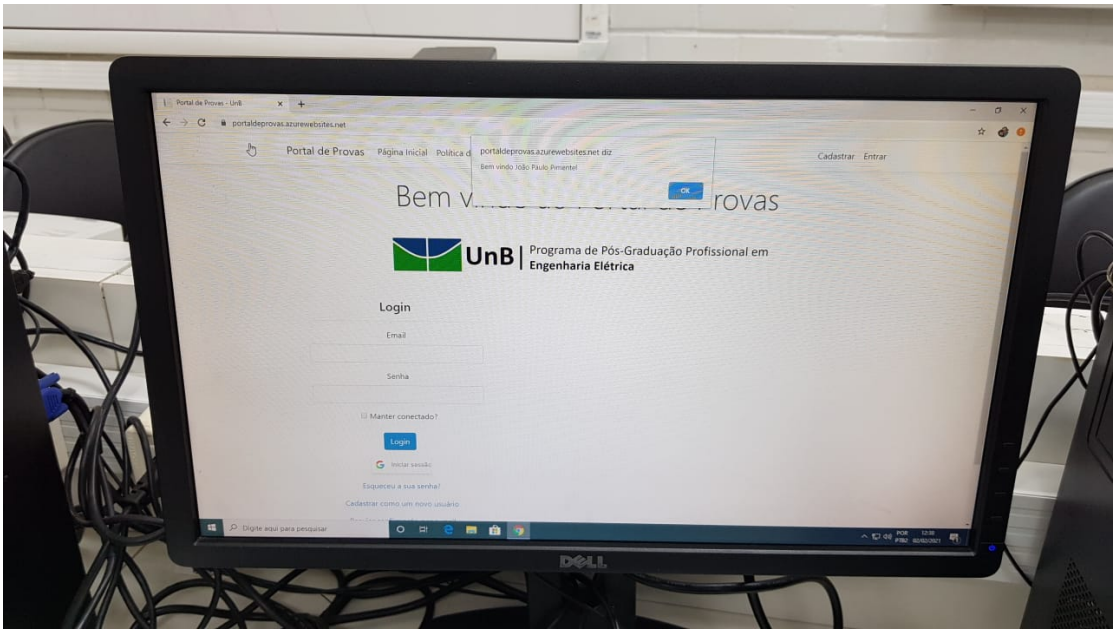

Figura 4.21: Testes Login do Google no Laboratório Latitude da UnB

Já o módulo da validação do *QR Code*, que é gerado no Portal de Provas, conforme mostrado na figura 4.22, apresentado pelo Acadêmico ao entrar no Laboratório, aonde deverá ter uma leitora de *QR Code* para validar este *QR Code* gerado para a referida avaliação.

O módulo SDN, foi testado utilizando o *Mininet* que fornece um ambiente de teste e desenvolvimento virtual para Redes Defininas por *Sofware*. O *Mininet* foi instalado no *VirtualBox* para o Sistema Operacional macOS Catalina, aonde respondeu de acordo com o Plano de Controle e Plano de Dados estabelecidos para o controle de tráfego da rede, pois ao receber uma solicitação o Gerenciamento de Usuários por SDN respondeu para cada solicitação de agendamento a uma porta dos *switches* configurados para esta rede SDN, com isso, também demonstrou a segurança através do IP da máquina que foi autorizada a realização da avaliação, ou seja, se o usuário tentar realizar a avaliação em qualquer outra máquina que não seja a máquina agendada, o *Token* recebido no *email* não será validado para o referido *login*.

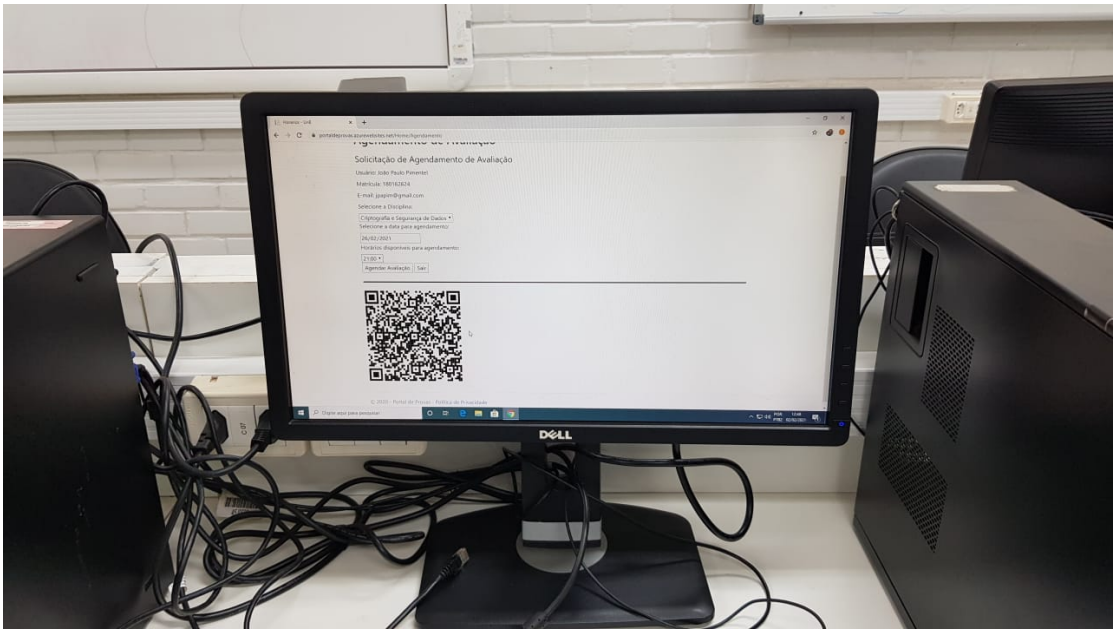

Figura 4.22: Testes Agendamento da Avaliação e Geração de QR Code no Laboratório Latitude da UnB

Devido a pandemia do Coronavírus não foi possível implementar nos *switches* fisicamente no Laboratório, pois iríamos precisar do apoio da equipe do Laboratório e não estamos com efetivo presencial na UnB, somente trabalhando de forma remota, no qual não daria para implementarmos em definitivo, mas após a volta das equipes para o presencial, este módulo será implementado e já teremos uma bateria de massa de testes que foi utilizada nos emuladores para serem utilizados fisicamente no laboratório da UnB.

Assim, para efeito dos testes de validação, o módulo desenvolvido para autenticar o usuário se mostrou seguro, que é o objetivo desta proposta de Portal de Provas, pois foram realizados testes na tentativa de logar com o *Token* digitado de forma errada, também foram realizados testes na tentativa de utilizar outra máquina que não seja a máquina previamente agendada (esta validação foi feita pelo IP) e também informando matrícula errada do usuário.

Este capítulo também descreve o ferramental usado na construção do Portal e o módulo de controle SDN e os cenários de validação, discutindo os resultados de testes em cada cenário.

Foi necessário o desenvolvimento e a implementação de um módulo para realizar cenários de validação em que testes específicos resultassem em obtenção de métricas de funcionamento do modelo proposto. Para tanto, é necessário que o protótipo implementado seja representativo do funcionamento dos dispositivos que foram emulados com o parte do Portal de Provas que foi implementado e hospedado no Microsoft *Azure*.

O Portal de Provas foi desenvolvido utilizando a linguagem de programação C# com a utilização do Microsoft *Visual Studio* para MAC OS, com auxilio de alguns programas como o *QR Code Generate*.

A metodologia de desenvolvimento do Portal e o módulo SDN seguiram a metodologia e

padrões da engenharia de *software* convencional, como a abstração e organização, utilizando-se ainda de programação Orientada a Objetos. Após a descrição da arquitetura lógica, será mostrada a descrição do ambiente de desenvolvimento do módulo SDN, mostrando as ferramentas utilizadas, como o Protocolo *Open Flow*, a Plataforma *Ryu*, além disso, descreveremos os cenários de testes e o desempenho da arquitetura funcionando na comunicação entre os módulos e o Portal propriamente dito.

A implementação do módulo principal descrito na arquitetura em questão foi realizado com ferramentas de desenvolvimento de código aberto ou de uso livre e distribuído gratuitamente na Internet, sendo todas multiplataforma. Os tópicos seguintes tratam em específico de cada uma das ferramentas utilizadas para a implementação do Portal de Provas, bem como as interações através das APIs entre as camadas da SDN.

4.11 Desenvolvimento do Portal de Provas

Para o desenvolvimento do Portal de Provas aqui proposto, seguiu-se o modelo de arquitetura apresentada no capítulo 3, utilizando todas as ferramentas descritas no capítulo 2, de maneira que o mesmo tem como característica simular o funcionamento de um Portal de Provas se comunicando com o módulo SND.

4.12 Descrição do Ambiente de Teste

Os testes foram implementados utilizando uma tabela definida para o Plano de Controle do container executando no *Docker* para simular a comunicação entre o Portal e o Gerenciamento através do SDN.

4.12.1 Descrição do *Hardware* Utilizado

O ambiente utilizado para as simulações do trabalho é composto por 1 computador do tipo MacBook com Processador Intel Dual Core-I5 1.8 Ghz, com 8GB de memória RAM 1600 MHz DDR3, com um espaço em disco de 120GB SSD e com uma placa de vídeo do tipo Off Board Intel HD Graphics com 1536 MB 4.23, responsável pelo ambiente de codificação e implementação do módulo SDN.

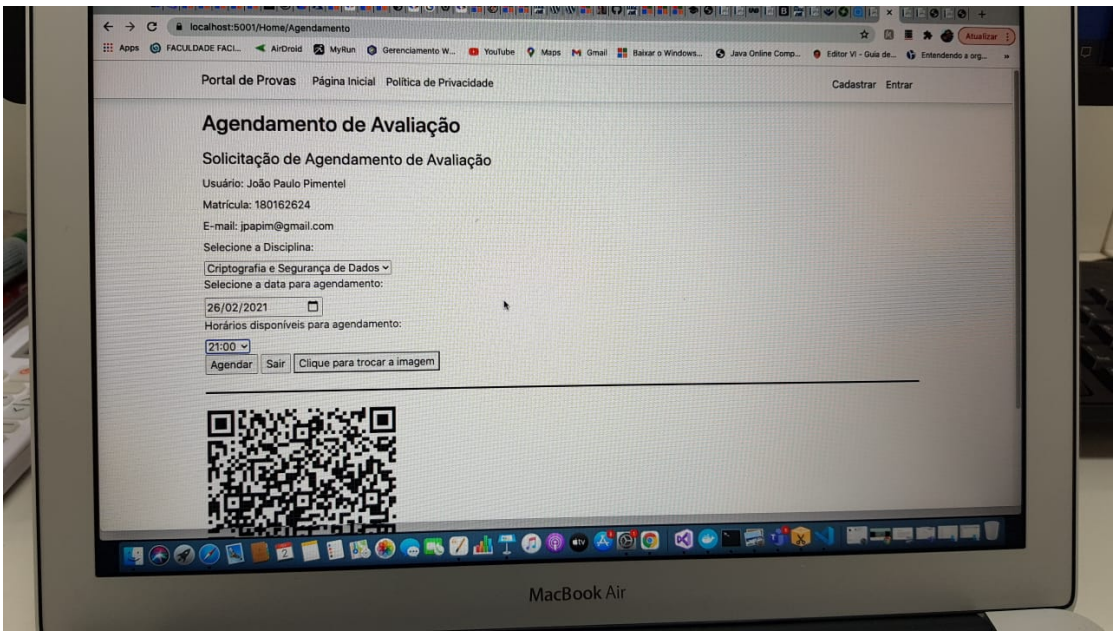

Figura 4.23: MacBook com o Mac OS Catalina para o Desenvolvimento

4.12.2 Descrição do Ambiente de *Software*

Com relação à parte de *softwares*, foi utilizado como sistema operacional do ambiente de desenvolvimento o Mac OS *Catalina* e *Docker* versão 3.01. Como linguagem de programação o C# utilizando o Microsoft *Visual Studio* para Mac OS na versão 8.8.4 4.24, implementando utilizando a tecnologia .NET Core SDK versão 3.1.406.

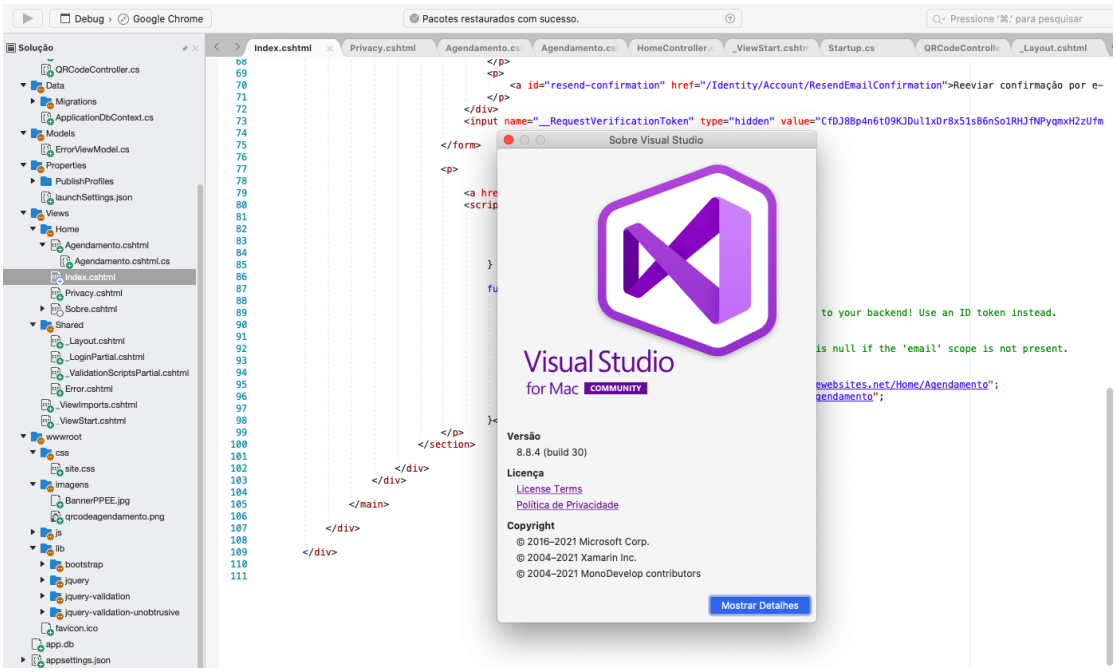

Figura 4.24: Visual Studio para Mac OS - Versão 8.8.4

Capítulo 5

Conclusão e Trabalhos Futuros

Esta Dissertação propôs uma arquitetura para um Portal de Provas Seguro para os Acadêmicos realizarem os seus agendamentos das Avaliações com segurança utilizando SSO para autenticação no Portal, *QR Code* para gerar um *Token* que será validado por um gerenciamento de usuários de Rede por controle SDN, com isso, garante que o Acadêmicos estará realizando a avaliação no laboratório que foi definido pelo gerenciamento SDN, ou remotamente utilizando VPN com o software *OpenVPN*.

Neste trabalho foi apresentado um breve histórico de desenvolvimento e algumas características dos usuários que irão utilizar o PSPE. Além de apresentar a Estrutura Analítica do Projeto PSPE, as especificações dos Requisitos referentes ao desenvolvimento da solução, foram elaborados alguns diagramas com base nos requisitos propostos pela UML, tais com: caso de uso e de classe. Para a elaboração estes diagramas foi utilizada a ferramenta *Web Lucidchart*.

Os desafios para este Portal de Provas só estão começando, pois a integração deste Portal com a base de dados de um Sistema Acadêmico deverá ser bem definida através de comunicação via *API REST*, ou seja, o PSPE deverá consumir este serviço de forma *online* para ter sempre a sua lista de disciplinas atualizadas em relação às disciplinas que os Acadêmicos estão matriculados, pois serão fundamentais para a realização dos agendamentos destas avaliações pelos Acadêmicos e o gerenciamento pelos Professores, assim como para os Acadêmicos visualizarem as notas lançadas/liberadas pelos Professores.

Já sobre a comunicação do Portal com o SDN com envio desta mensagem com o *Token* válido é uma inovação que garante que esta proposta seja aceita e utilizada no desenvolvimento de vários Portais de Provas em várias instituições de ensino.

5.1 Trabalhos Futuros

Para trabalhos futuros fica a ideia de desenvolver um aplicativo *mobile* para comunicar com este Portal e os Acadêmicos além de registrarem suas presenças com o *QR Code* gerado para a aula, validando este *QR Code* ao entrar no laboratório ou sala de aula, como também realizar os

agendamentos das avaliações e até quem sabe realizar as avaliações no aplicativo *mobile*, pois é a tendência de quase todas as aplicações *Web*, mas sempre lembrando de implementar uma segurança bem definida para que não ocorra falhas de segurança nestas validações/autenticações que o PSPE propôs.

Outra ideia a ser implementada neste Portal de Provas é a limitação do *browser* impedindo que o Acadêmico acesse outros sites externos ou documentos/*softwares* durante a realização da avaliação agendada.

Referências Bibliográficas

[1] ONEIDENTITY. *Cloud Access Manager*. 2020. Acessado em: 06 nov. 2020. Disponível em: <https://support.oneidentity.com/es-es/technical-documents/cloud-access-manager/8.1.4/how-to-develop-openid-connect-apps/2>.

[2] AL-SANEA M. S.; AL-DARAISEH, A. A. Security evaluation of saudi arabia's websites using open source tools. *Anti-Cybercrime (ICACC), 2015 First International Conference on [S.l.]. p.1-5*, 2015.

[3] T.J.SOON. Qr code. *Synthesis Journal*, 2008.

[4] HOWARD, M.; LEBLANC, D. *Escrevendo Codigo Seguro*. [S.l.]: Bookman Companhia, 2005.

[5] BRAZ, F. A. *Instrumentalização da Análise de Projeto de Software Seguro Baseada em Ameaças e Padrões*. 2009. Acessado em: 12 set. 2021. Disponível em: <https://repositorio.unb.br/bitstream/10482/4089/1/2009$_FabricioAtaidesBraz.pdf$>.

[6] PIMENTEL JOãO PAULO; MONTEIRO, M. S. F. L. D. C. F. M. F. L. L. F. A. A. Y. R. S. J. R. T. Proposta de arquitetura de um portal de provas seguro utilizando sdn como solução de gerenciamento de redes. *Conferências Ibero-Americanas WWW/Internet e Computação Aplicada 2020, Vilamoura. Lisboa: IADIS Press*, 2020.

[7] PIMENTEL JOãO PAULO; NZE, G. D. A. M. F. L. L. D. S. J. R. T. C. E. D. R. V. C. Vulnerabilidades de sistemas ciber físicos. *Conferências Ibero-Americanas WWW/Internet e Computação Aplicada 2020, Vilamoura. Lisboa: IADIS Press*, 2020.

[8] MENDONÇA, F. L. L. de; CAMÕES, R. J. da S.; PIMENTEL, J. P. Criação de um projeto de soc de pequeno porte, viabilizando melhoras de monitoramento e soluções de segurança da informação. *TECNOLOGIAS EM PROJEÇÃO*, v. 10, n. 2, p. 46–57, 2019.

[9] MICULAN M.; URBAN, C. Formal analysis of facebook connect single sign-on authentication protocol. *In: Proc. SofSem 2011, OKAT, pp. 99–116*, 2011.

[10] BARNES, R. Use cases and requirements for dns-based authentication of named entities (dane). rfc 6394 (informational). rfc. fremont, ca, usa: Rfc editor. *DOI: 10.17487/RFC6394. URL: https://www.rfc-editor.org/ rfc/rfc6394.txt*, 2011.

[11] PATTERSON, P. *Digging deeper into OAuth 2.0 on force.com*. 2014. Acessado em: 02 set. 2020. Disponível em: <https://developer.salesforce.com/page/Digging$_{D}eeper_{i}nto_{O}Auth_{2}.0_{o}n_{F}orce.com$>.

[12] MICROSOFT. *Entrada na Web com o OpenID Connect no Azure Active Directory B2C*. 2020. Acessado em: 06 nov. 2020. Disponível em: <https://docs.microsoft.com/pt-br/azure/active-directory-b2c/openid-connect>.

[13] SRINOUNPAN, B. et al. The application of qr code technology to create the value-added products for the baan klong peek neur beehive community enterprise group at tambon suankhan, nakhon si thammarat province. *Systematic Reviews in Pharmacy*, v. 11, p. 519–528, 07 2020.

[14] MCKEOWN, N. e. a. Openflow: Enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review, v.38, no.2, pp.69–74*, 2008.

[15] MENEZES, P. M. et al. Estudo catalográfico sobre redes definidas por software (sdn) no âmbito brasileiro. In: . [S.l.: s.n.], 2018.

[16] COSTA, L. R. *OpenFlow e o paradigma de redes definidas por software*. 2013. Acessado em: 06 ago. 2020. Disponível em: <http://monografias.cic.unb.br/dspace/bitstream/123456789/391/1/Monografia$_{V}esao_{L}eitura_{e}m_{P}C.pdf$>.

[17] DOCS, D. *Visão geral do Docker*. 2020. Acessado em: 11 jun. 2020. Disponível em: <https://docs.docker.com/get-started/overview/>.

[18] FOUNDATION, O. N. *Software-Defined Networking: ONF SDN Evolution*. 2016. Acessado em: 12 jun. 2020. Disponível em: <http://www.opennetworking.org/wp-content/uploads/2013/05/TR-535$_{O}NF_{S}DN_{E}volution.pdf$>.

[19] NOVIFLOW. *The OpenFlow Protocol*. 2020. Acessado em: 11 jun. 2020. Disponível em: <https://noviflow.com/the-basics-of-sdn-and-the-openflow-network-architecture/>.

[20] AMORIM, R. *O uso do protocolo OpenFlow em redes definidas pro software*. 2016. Acessado em: 19 out. 2020. Disponível em: <http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/1831/1/CT$_{G}ESER_{I}I_{2}012_{0}9.pdf$>.

[21] FOUNDATION, O. N. *Mininet*. 2020. Acessado em: 05 dez. 2020. Disponível em: <https://opennetworking.org/mininet/>.

[22] FOUNDATION, O. N. *Open Network Operating System: ONOS*. 2021. Acessado em: 11 fev. 2021. Disponível em: <https://opennetworking.org/onos/>.

[23] ORACLE. *Manual do Oracle VirtualBox*. 2021. Acessado em: 23 fev. 2021. Disponível em: <https://www.virtualbox.org/manual/ch01.htmlovf-about>.

[24] TELEGRAPH, N.; CORPORATION, T. *O que é Ryu.* 2020. Acessado em: 22 nov. 2020. Disponível em: <https://ryu.readthedocs.io/en/latest/getting$_started.html what - s - ryu$>.

[25] COMMUNITY, R. S. F. *O QUE É RYU?* 2020. Acessado em: 04 dez. 2020. Disponível em: <https://ryu-sdn.org/>.

[26] MICROSOFT. *Visual Studio - As melhores ferramentas para qualquer desenvolvedor*. 2020. Acessado em: 05 dez. 2020. Disponível em: <https://visualstudio.microsoft.com/pt-br/>.

[27] MICROSOFT. *Desenvolva aplicativos .NET*. 2020. Acessado em: 05 dez. 2020. Disponível em: <https://visualstudio.microsoft.com/pt-br/vs/features/net-development/>.

[28] MICROSOFT. *Como criar aplicativos ASP.NET Core no Visual Studio para Mac.* 2019. Acessado em: 07 dez. 2020. Disponível em: <https://docs.microsoft.com/pt-br/visualstudio/mac/building-asp-net-core?view=vsmac-2019>.

[29] MICROSOFT. *Documentação do Serviço de Aplicativo.* 2020. Acessado em: 22 nov. 2020. Disponível em: <https://docs.microsoft.com/pt-br/azure/app-service/>.

[30] MOZILLA. *Trabalhando com JSON*. 2020. Acessado em: 05 dez. 2020. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Aprender/JavaScript/Objetos/JSON>.

[31] SILVA, M. S. *Aprenda a usar o framework Bootstrap para criar layouts CSS complexos e responsivos*. [S.l.]: Novatec, 2015.

[32] BALDUINO, P. *Dominando JavaScript com jQuery*. [S.l.]: Casa do Código, 2012.

[33] MOZILLA. *Gitv - tudo-é-local*. 2020. Acessado em: 05 dez. 2020. Disponível em: <https://git-scm.com/>.

[34] GNS3. *Introdução ao GNS3*. 2021. Acessado em: 27 set. 2021. Disponível em: <https://docs.gns3.com/docs/>.

[35] GOOGLE. *Sobre o Firebase*. 2021. Acessado em: 03 nov. 2021. Disponível em: <https://firebase.google.com/?hl=pt-br>.

[36] SOFTWARE, L. *O que é o Lucidchart*. 2021. Acessado em: 23 nov. 2021. Disponível em: <https://www.lucidchart.com/br>.

[37] VALE, C. B. A. d. A. et al. Identidade eletrônica e autenticação de médicos no brasil. 2021.

[38] FARIAS, É. C. et al. *Gestão de identidade pessoal com opendid connect e cartão de cidadão*. Dissertação (Mestrado), 2018.

Apêndice A

Artigo 1 Publicado e Apresentado no CIACA 2020

O artigo abaixo foi aceito e apresentado remotamente durante a 7a conferência Ibero-Americana de Computação Aplicada 2020 realizado em Lisboa, Portugal, no dia 20 de novembro de 2020:

PROPOSTA DE ARQUITETURA DE UM PROTAL DE PROVAS SEGURO UTILIZANDO SDN COMO SOLUÇÃO DE GERENCIAMENTO DE REDES

João Paulo Pimentel, Georges Daniel Amvame Nze, Fábio Lúcio Lopes de Mendonça, Robson de Oliveira Albuquerque, Vanessa Coelho Ribeiro, Rafael Timóteo de Sousa Júnior
joao.pimentel, georges.amvame, fabio.mendonca, robson, vanessa.ribeiro, desousa @redes.unb.br Programa de Pós-Graduação Profissional em Engenharia Elétrica (PPEE), Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, Brasil, Zipcode 70910-900

RESUMO

Este artigo apresenta a proposta de um Portal Seguro de Provas com autenticação única utilizando SSO (Single Sign-On) e geração de token através de um código de barras bidimensional (QR Code) gerado para a validação do usuário que deseja acessar o referido Portal com o intuito de realizar as suas avaliações previamente agendadas. Como solução de Gerenciamento de Redes, a proposta é de utilizar Redes definidas por software, ou seja, o SDN (Software Defined Networks).

PALAVAS CHAVES Portal, SSO, QR Code, Token, SDN.

1. INTRODUÇÃO A proposta deste artigo é de apresentar uma arquitetura de Portal de Provas seguro com autenticação SSO (Single Sign-On), utilizando QR Code (Quick Response Code) com geração de Token para os usuários do sistema e utilizando SDN (Software Defined Networking) como solução de gerenciamento de redes. Neste Portal o usuário terá a segurança de

autenticação única e segura é feito justamente para os alunos irem realizar as suas avaliações de acordo com o seu agendamento no Portal que após a escolha da referida disciplina, data e horário será gerado um QR Code para ser validado e confirmado ao entrar no laboratório que irá realizar a referida avaliação.

A autenticação SSO que é para validar e confirmar que é você mesmo que está utilizando o sistema Web, tem a ideia de exigir uma mesma senha para todos os acessos de forma segura e transparente, evitando com isso, que tenha que anotar ou tenha que gerar uma senha para cada tipo de aplicativo que estiver utilizando, ou seja, como a autenticação SSO, o usuário digita apenas uma vez o login e a senha no primeiro acesso e depois vai abrindo os demais aplicativos sem a necessidade de digitar novamente a senha. Um bom exemplo desta autenticação é a utilização da autenticação do seu e-mail que validará que é você mesmo que está utilizando o sistema Web, então deixaria você acessar o Portal para agendar a sua avaliação.

Outra validação de segurança do SSO seria o envio de código de validação para o seu número de celular ou e-mail para confirmar se é você mesmo que está utilizando o referido sistema Web, com isso, aumentaria a confiança sobre esta utilização.

O QR Code com Geração de Token neste Portal seria enviado para o e-mail do usuário cadastrado e seria utilizado no momento da entrada do aluno no destino da realização de sua avaliação, ou seja, no laboratório agendado. Ao apresentar o QR Code seria gerado um Token para ser utilizado no login no Portal na estação de trabalho agendada para o determinado login, ou seja, o Token somente será válido na máquina que foi liberada no momento do agendamento pelo aluno.

Este agendamento em máquinas específicas será realizado utilizando SDN como Solução de Gerenciamento de Redes e o OpenFlow como protocolo de comunicação, ou seja, um controlador SDN localizado na central de monitoramento com um plano definido fica responsável por gerenciar os switches da rede, liberando o acesso pelo Token informado somente naquela porta de rede que encontra a máquina que o aluno vai realizar a referida avaliação.

2. REFERENCIAL TEÓRICO E ARQUITETURA PROPOSTA

O gerenciamento baseado em SDN que será controlado por um elemento central, ficou conhecido por separar do controle da rede dos dispositivos de encaminhamento, tais como roteadores, switches, fazendo com esta lógica fosse definida por software de forma centralizada, provendo à rede através da utilização de aplicações customizadas para cada cenário definido (Mckeown et al., 2008) e estas aplicações implementam a inteligência da rede e proveem abstração dos detalhes técnicos dos equipamentos e dos protocolos.

O OpenFlow é a principal interface de código aberto entre SDN controladores e comutadores ou outros caminhos de dados. Possui suporte para diversos aplicativos usando protocolos de planos de dados com a classificação de pacotes sendo executadas usando tabelas de correspondência e operações de processamento de pacotes chamadas de ações ou instruções (Software-Defined Networking, 2016). O OpenFlow é um padrão definido pela Open Networking Foundation (ONF) para implementar SDN em equipamentos de rede e define a interface entre um controlador OpenFlow e um comutador OpenFlow, ou seja, ele permite que o OpenFlow Controller instrua o co-

mutador OpenFlow sobre como lidar com pacotes de dados recebidos (Noviflow, 2020).

A autenticação SSO na Web, como exemplo do Login do Google ou Facebook, são baseados no protocolo OpenID Connect (Miculan, M., Urban, C, 2011). Esse protocolo permite que através da confiança capitam a autenticação do usuário aos provedores de identidade, ou seja, é através dele que é passada a identificação do usuário para a devida autenticação e validação nas aplicações Web. No SSO, um serviço da Web, nesse contexto chamado RP (Relying Party), delega autenticação para o chamado provedor de identidade (IdP). A tarefa de um IdP no SSO da Web é autenticar usuários e atestar suas identidades aos RPs. Alguns serviços que os usuários usam diariamente como provedores de e-mail ou redes sociais, servem como IdPs (R. Barnes, 2011).

Hoje, diversos usuários da Web utilizam suas contas do Facebook ou do Google para acessar sites de terceiros (Relying Party). Esse esquema de login único baseado na Web é ativado pelo OAuth 2.0, um protocolo de autorização de recursos da Web que foi adotado pelos principais provedores de serviços (Patterson, 2014). O QR Code (Quick Response Code) que é utilizado para resposta rápida (Pillai, A. E., Prakash, D., Al-Marhoobi, N. A., & Shrivastava, M., 2017) tem a vantagem nesta proposta do Portal, pois vem da facilidade em realizar a leitura durante a entrada no laboratório para realizar a devida prova.

A arquitetura proposta neste trabalho será de pôr autenticação SSO, geração de um Token através de um QR Code gerado para a autenticação e validação de um determinado usuário (o aluno). Na sequência o aluno irá acessar o Portal a fim de realizar o agendamento da sua devida avaliação. A ideia central do do sistema é a autenticação/validação dos usuários através de SSO, que na sequência terá a opção do aluno escolher qual avaliação irá fazer e com isso será gerado um QR Code no qual o aluno deverá apresenta-lo no laboratório que será realizada a devida avaliação. Após apresentar o QR Code no laboratório previamente agendado para a devida avaliação, este gerenciamento será resolvido com a utilização de SDN, o módulo de autenticação/validação deste Portal irá gerar um Token válido somente para a avaliação solicitada na máquina liberada. Com a utilização da matrícula e o Token gerado pelo Portal Web, o aluno irá logar no referido Portal de Provas e realizar a devida avaliação.

Nesta arquitetura proposta, para ficar mais seguro, o Token enviado para o aluno que será validado somente em uma máquina específica do laboratório, terá um prazo para expirar de 20 minutos, ou seja, se o aluno não realizar o login no referido Portal para realizar a sua avaliação nestes 20 minutos, ele terá que solicitar novamente um novo código que poderá mudar até de estação de trabalho para a realização da referida avaliação.

Sobre o gerenciamento via SDN, este módulo deverá ter uma configuração para enviar automaticamente as requisições dos alunos para os destinos (máquinas nos laboratórios) através do plano de controle de um controlador SDN, utilizando como protocolo de comunicação o OpenFlow. Este controlador SDN enviará as informações para os switches que comunicarão com as máquinas definidas para a realização de cada avaliação e na hora da realização da Prova, ao digitar a matrícula e token recebido após a validação do agendamento, serão confirmados na tabela

do plano de controle SDN utilizando uma Interface para comunicar com o Portal.

Para o desenvolvimento deste Portal, será utilizada o Framework Flutter que é um kit de ferramentas de interface do usuário desenvolvido pelo Google para criar aplicativos que são compilados de forma nativa para dispositivos móveis, Web e desktop a partir de uma única base de código fonte (Flutter, 2020). A linguagem de programação utilizada neste Portal, será a linguagem Dart, que foi desenhada pelo Google para aplicações Web modernas e capacitadas para ambientes de alta performance (Dart, 2020). A base de dados para este Portal será o Firebase Realtime Database, que além de base de dados, é uma plataforma de desenvolvimento de fácil integração na Web. Os dados são armazenados como objetos JSON (JavaScript Object Notation) na nuvem, ou seja, ao contrário de um banco de dados SQL, não há tabelas e nem registros, mas sim uma árvore JSON (Firebase, 2020).

Este Portal terá três tipos de perfis de acessos, um de aluno que somente agenda e realiza a sua avaliação, outro de Professor para alimentar o Banco de Questões do referido Portal e liberação das provas e o de Coordenador que revisa e valida questões incluídas pelos Professores. A partir da validação das questões pelo Coordenador, estas questões serão liberadas para os Professores montarem suas provas e liberarem para serem realizadas pelos alunos.

2.APRESENTAÇÃO DOS RESULTADOS INICIAIS

Os resultados iniciais deste Portal, de acordo com a arquitetura proposta de desenvolvimento, apresentam uma tela de login na Figura A.1 aonde o usuário tem a opção de realizar o login se já tiver cadastrado, ou cadastrar um login de sua preferência, ou simplesmente continuar utilizando a sua conta do Facebook ou da Google.

Figura A.1: Tela deLogin

Já a tela de agendamento de Provas, ilustrada na Figura A.2 , demostra como será este agendamento do usuário, aonde após logado serão apresentadas as informações do usuário logado, tais como nome, matrícula, curso que está matriculado. Nesta tela o usuário seleciona a disciplina que ele quer realizar o agendamento da Prova, nesta lista somente vão aparecer as disciplinas que o usuário está matriculado, pois este Portal deverá receber estas informações do Portal de Matrículas, na sequência escolhe uma data no calendário e automaticamente irá aparecer uma lista de horários disponíveis para o referido agendamento, escolhido o horário e clicado no botão agendar, irá gerar um QR Code para apresentação no dia da Prova no Laboratório agendado.

Com o QR Code gerado o usuário vai apresentá-lo no laboratório agendado, ao apresentar o referido QR Code ele receberá um Token no e-mail cadastrado e/ou número de celular cadastrado com validade de 20 minutos para realizar o login no Portal na máquina agendada juntamente com sua matrícula. A Figura A.2 ilustra o modelo de mensagem que poderá chegar por e-mail ou SMS (Short Message Service).

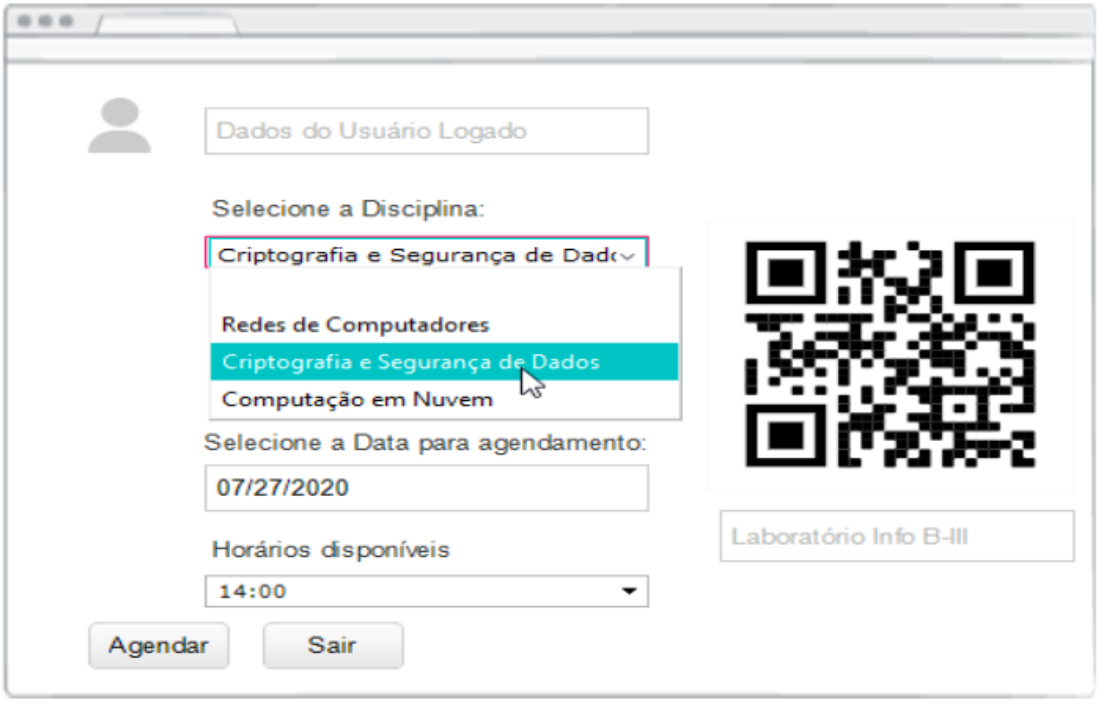

Figura A.2: Tela de Agendamento

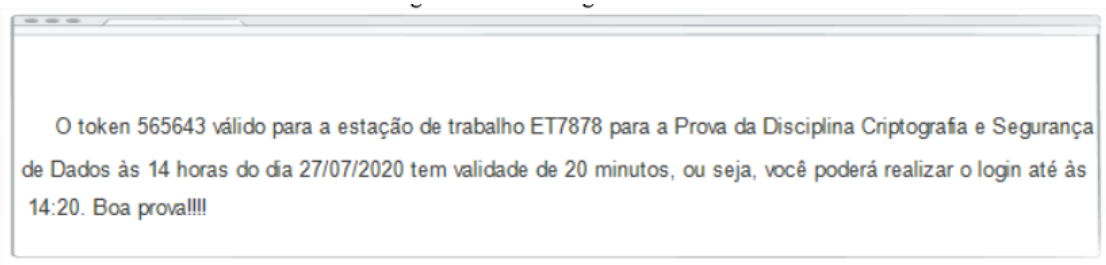

Figura A.3: Tela do Token enviado ao usuário

Com o Token o usuário irá efetuar o login no Portal através do módulo de realização de Provas.

4. CONCLUSÃO E TRABALHOS FUTUROS

Este artigo propôs uma arquitetura para um Portal de Provas Seguro para alunos realizarem os seus agendamentos das provas com segurança utilizando SSO para autenticação no Portal, QR Code para gerar um Token que será validado por um gerenciamento de Redes por controle SDN, com isso, garante que o aluno estará realizando a avaliação no laboratório que foi definido pelo gerenciamento SDN.

Os desafios para este Portal de Prova só estão começando, pois a integração deste Portal com a base de dados de um Portal de Matrícula deverá ser bem definida, a comunicação do Portal com o SDN com envio desta mensagem com o Token válido é uma inovação que garante que esta proposta seja aceita e utilizada no desenvolvimento de vários Portais de Provas em várias instituições de ensino.

Para trabalhos futuros fica a ideia de desenvolver um aplicativo mobile para comunicar com este Portal e os alunos além de registrarem suas presenças com o QR Code gerado para a aula, validando este QR Code ao entrar no laboratório ou sala de aula, como também realizar os agendamentos das avaliações e até quem sabe realizar as avaliações no aplicativo mobile, pois é a tendência de quase todas as aplicações Web, mas sempre lembrando de implementar uma segurança bem definida.

AGRADECIMENTO

Os autores agradecem o apoio do Programa de Pós-Graduação Profissional em Engenharia Elétrica (PPEE) da UnB Agências brasileiras de pesquisa, desenvolvimento e inovação CNPq e CAPES, bem como o suporte do Laboratório LATITUDE/UnB a a cooperação o Gabinete de Segurança Institucional da Presidência da República / CEPESC (TED 002/2017).

REFERÊNCIAS

Firebase. Disponível em: <https://firebase.google.com/docs/database/Web/structure-data?hl=pt-br>.Acessado em: 12 jul. 2020.

Flutter. Disponível em: <https://flutter.dev/>. Acessado em 10 jul. 2020.

Dart. Disponível em: <https://dart.dev/>. Acessado em 10 jul. 2020.

Mckeown, N. et al. OpenFlow: enabling innovation in campus networks. SIGCOMM Comput. Commun. Rev., ACM, New York, NY, USA, v. 38, n. 2, p. 69–74, 2008.

Miculan, M., Urban, C.: Formal analysis of Facebook Connect Single Sign-On authentication protocol. In: Proc. SofSem 2011, OKAT, pp. 99–116 (2011)

Noviflow. The OpenFlow Protocol. Disponível em: <https://noviflow.com/the-basics-of-sdn-and-the-openflow-network-architecture/>. Acessado em: 11 jun. 2020.

Patterson, P. Digging deeper into OAuth 2.0 on force.com (2014). Disponível em <https://developer.salesforce.com/page/Digging$_D eeper_i nto_O Auth_2.0_o n_F orce.com$ > $.Acessadoem12jun.2020$.

Pillai, A. E., Prakash, D., Al-Marhoobi, N. A., Shrivastava, M. (2017). Application of QR codes in tourism industry: A review of literature. International Journal of Computer Technology Applications, 8(6), 678–687.

R. Barnes. Use Cases and Requirements for DNS-Based Authentication of Named Entities (DANE). RFC 6394 (Informational). RFC. Fremont, CA, USA: RFC Editor, Oct. 2011. DOI: 10.17487/RFC6394. URL: https://www.rfc-editor.org/ rfc/rfc6394.txt.

Software-Defined Networking: ONF SDN Evolution. Open Networking Foundation, p. 8, 2016. Disponível em: <http://www.opennetworking.org/wp-content/uploads/2013/05/TR-535$_O NF_S DN_E volution.pdf > .Acessadoem : 12jun.2020.$

Apêndice

Apêndice B

Artigo 2 Publicado e Apresentado no CIACA 2020

VULNERABILIDADES DE SISTEMAS CIBER FÍSICOS

João Paulo Pimentel, Matheus Santos Monteiro, Francisco Lopes de Caldas Filho, Fábio L. L. Mendonça, Awatef Ali Yousef R Fares, Rafael T. de Sousa Jr. *Programa de Pós-Graduação Profissional em Engenharia Elétrica (PPEE), Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, Brasil, Zipcode 70910-900*

RESUMO

A primeira transmissão de rádio frequência foi no ano de 1892 e desde então essa tecnologia vem evoluindo com o passar dos anos. Uma das áreas que usam o RF (Radiofrequência) é na parte de segurança privada com o dispositivo emissor/receptor 433 MHz, que é usado em portões eletrônicos de residências particulares e empresas. No entanto, esta tecnologia apresenta falha de segurança de dados no envio do seu sinal, desta forma, deixando esta tecnologia vulnerável a ataques. Este artigo visa produzir um dispositivo IoT que explore essa vulnerabilidade, a fim de demonstrar a sua insegurança.

PALAVRAS CHAVES

433 MHz , IoT, RF, Vulnerabilidade, Segurança.

1. INTRODUÇÃO

As comunicações sem fios é uma tecnologia que a cada dia mais é utilizada para envio e recebimento de informação (Gupta, et al., 2000). A sua criação se deu no ano de 1892 com o pesquisador e cientista padre Roberto Landell de Moura com a primeira transmissão de RF (Radiofrequência) com o alcance de 8 Km (Alencar, et al., 2004). Assim, com o passar dos anos os seus métodos e protocolos de envio a partir de RF evoluíram até chegar nos tempos de hoje com diferentes uma variedade de tecnologias como: Wi-fi, Bluetooth, BTLE, ZigBee, Z-Wave e entre outros.

Segundo (Waslon TA, et al., 2016) RF é qualquer frequência de ondas eletromagnéticas na faixa de 3 KHz a 300 GHz que são geradas por oscilações elétricas. Assim, para enviar e receber informações são necessários dois dispositivos um transmissor e outro receptor. O receptor codifica e envia o sinal em determinada frequência e o receptor recebe esse sinal e o decodifica.

Um dos modelos de dispositivos RF, mais usados na parte de segurança particular, é o emissor e receptor de RF 433 MHz, essa tecnologia sem fio é utilizada em motores portões eletrônicos domiciliares e empresariais. O módulo transmissor/receptor RF 433 tem uma frequência de 433 MHz que a sua transmissão ocorre com uma taxa de 1 Kbps a 10 Kbps (Kuno, et al., 1999). No entanto, a transmissão de sinal RF 433 MHz apresenta vulnerabilidade de segurança (Kahil, et al., 2017), assim comprometendo a proteção dos locais onde utilizam essa tecnologia.

Segundo a (Anon., s.d.) Eletrônica o mercado de segurança eletrônica brasileiro faturou 7,17 bilhões de reais no ano de 2019 e para o ano de 2020 tem uma previsão de um crescimento de 12%.

Desta Forma, mais pessoas físicas e jurídicas vão utilizar motor de portão eletrônico em suas casas, empresas e condomínios. Por garantir comodidade e permitir a abertura e fechamento remoto, reduzindo o tempo de exposição do motorista fora dos limites da residência. Este dispositivo, entretanto, apresenta falhas de segurança que se utilizadas, podem permitir que pessoas não autorizadas tenham acesso ao ambiente que deveria ser restrito apenas a pessoas que tenham posse do controle corretamente codificado do portão.

Visando a melhoria da segurança eletrônica mais especificamente o motor dos portões eletrônicos, este artigo tem como objetivo apresentar um protótipo de dispositivo de segurança capaz de aproveitar da vulnerabilidade dos sistemas de portão eletrônico, como interceptar o código de um portão por um dispositivo próximo, possibilitando que um invasor abra o portão sem autorização do proprietário da casa. Com o advento dos dispositivos e tecnologias de IoT (Internet of Things), essas vulnerabilidades podem ser exploradas ainda mais longe, permitindo que o código do portão seja enviado para terceiros pela internet, além de comandar a abertura do portão a distância.

2. TRABALHOS RELACIONADOS

Muitos métodos de provar a vulnerabilidade do RF 433 MHz já foram propostos, tendo por objetivo de explicitar a insegurança dessa tecnologia.

Em (Azevedo, 2019) foi proposto um sistema que explora vulnerabilidades de transmissores e receptores que operam frequência de 433 MHz. Utilizando dois microcontroladores, um transmissor e um receptor 433 MHz e um computador. O sinal codificado enviados pelo controle do portão eletrônico é copiado pelo receptor 433 MHz e em seguida mostrado os seus parâmetros - frequência, largura de pulso e protocolo - na serial da IDE do Arduino. Posteriormente, usando esses parâmetros, um emissor 433 MHz e uma microcontrolador é simulado o sinal de RF para abrir o portão. No entanto, essa solução é pouco prática e requer um notebook para poder ver os resultados obtidos, diferentemente proposto nesse artigo que visa fazer isso tudo em apenas um dispositivo IoT sem a utilização de um computador.

Enquanto no artigo (Hung, et al., 2019) foi proposto um sistema que utiliza apenas um microcontrolador, o transmissor RF 433 MHz, receptor RF 433 MHz e led's. Com o intuito de poder captar qualquer sinal de de transmissão de 433 MHz, no entanto a solução não apresenta controle sem fio, fazendo apenas a citação que possui autonomia mas não demonstra no artigo, deste modo será necessário o uso de um computador para analisar o sinal interceptado e enviado o dado, desta forma limitando o uso do dispositivo. No entanto, este artigo visa construir um dispositivo autônomo sem a utilização de um computador para analisar os dados obtidos.

Porém no artigo (Fernández & Rodrigo, 2018) foi feito um sistema com dois dispositivos uma para a recebimento do sinal RF 433 MHz utilizando um microcontrolador e um receptor RF 433 MHz e outro para enviar o sinal que utiliza um microcontrolador e um emissor RF 433 MHz, a fim de melhorar a segurança do sinal RF 433 MHz a partir de criptografia, assim protegendo os sinais enviados, mas o artigo proposto neste trabalho visa aproveitar a vulnerabilidade de dispositivo com um dispositivo único e autônomo, sem a necessidade de dois microcontroladores.

De forma similar, o artigo (Hussein, et al., 2017) tem como fim melhorar a segurança das tags e leitores de RFID, utilizando um dispositivo que possui um sistema com um microcontrolador e receptor e emissor de sinal RF 433 MHz. Assim, utilizando um protocolo de autenticação mútua entre os leitores de RFID e as tags. Desta forma, mantendo as tags e os RFID's protegidos de qualquer ataque de invasores, no entanto este artigo proposto tem como objetivo explorar essa vulnerabilidade do sinal RF 433 MHz.

Por outro lado, o artigo (Rasmussen, et al., 2007) utilizando o sensor Chipcon 1000 rádio, 433MHz, este demonstra que um receptor pode criar impressões digitais de rádio do dispositivo e depois identificar a proveniência das mensagens enviadas de um dispositivo ao outro , mesmo que as mensagens estejam ocultas. No entanto, o método de captura do sinal é analógico, então para converter esse sinal para um sinal digital é necessário um osciloscópio. Desta forma deixando essa solução complicada e com muitos processos. Em contra partida, a solução proposta nesse artigo tem como objetivo realizar receptações utilizando um dispositivo microcontrolado e de fácil manuseio, assim facilitando a replicação dessa solução.

3. ARQUITETURA PROPOSTA

Tomando como base os projetos mencionados nos trabalhos relacionados, foi desenvolvido, nesse artigo, um dispositivo que intercepta o sinal RF 433 MHz, não criptografado, de controles de portão eletrônico. Em seguida, esse sinal é gravado na memória do microcontrolador e depois é enviado os parâmetros do sinal capturado para um Bot do Telegram utilizando redes sem fio 802.11n. A figura B.1 demonstra a arquitetura proposta desse Projeto IoT.

3.1 HARDWARE

A figura B.2 demonstra o hardware da arquitetura proposta utilizando o software Freezing.

O dispositivo proposto possui um microcontrolador Wemos D1 Wifi ESP8266 que é o módulo mais utilizado para aplicações em IoT por ser de baixo custo e baixo consumo de energia, pois a sua potência é de 3.3 Volts e o seu módulo Wi-Fi possui o protocolo TCP/IP integrado, assim

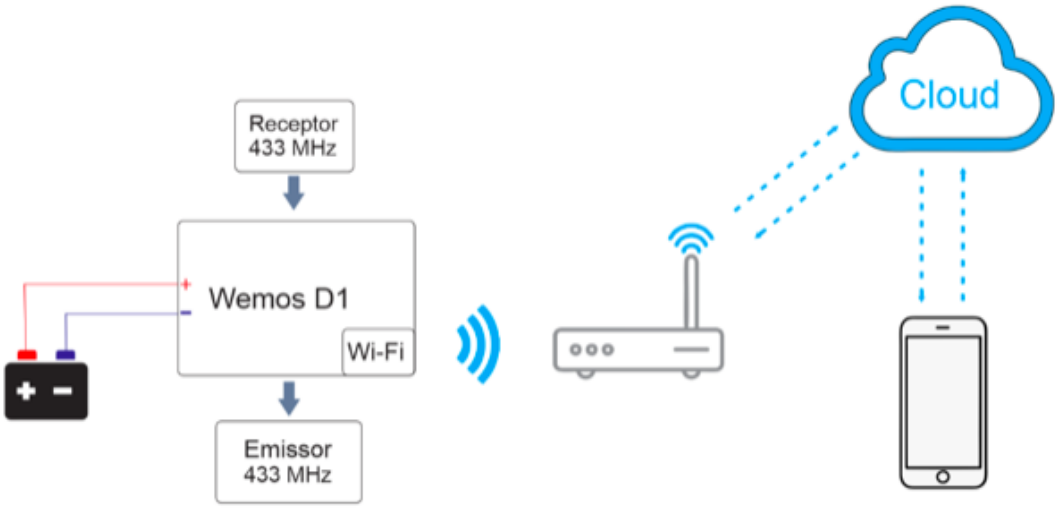

Figura B.1: Arquitetura proposta.

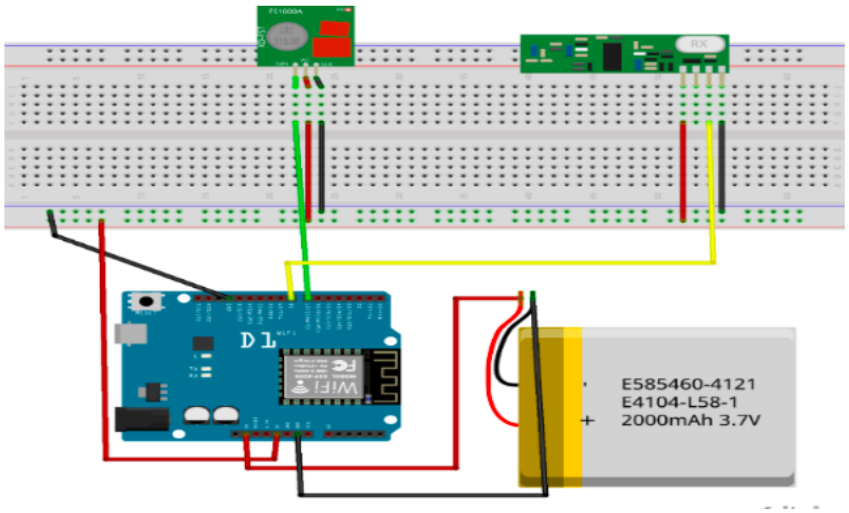

Figura B.2: Hardware da Arquitetura Proposta.

pode-se usar aplicações webs como o chatbot do Telegram (Kodali, et al., 2016). Também foi utilizado uma protoboard, o módulos RF 433 MHz que é formado por dois segmentos: o receptor, que é incumbido de captar os sinais de frequência 433 MHz, ele trabalha com uma tensão de 5 Volts e corrente de 4 mA com uma sensibilidade de -105 dB, e o transmissor, que é encarregado de emitir o sinal de frequência 433 MHz, ele trabalha com a tensão de 3.5 a 12 volts, com um raio de alcance de 20 a 200 metros e transmite dados com uma velocidade de 4 Kbps (Marcos & Lucas, 2018). Uma bateria, de preferência, 3,7v para alimentar o circuito e dar mobilidade ao dispositivo e por último uma antena para o emissor RF 433 MHz para poder melhorar a distância do envio do sinal (Alves, 2018).

Utilizando a IDE do Arduíno, que é um Ambiente de Desenvolvimento Integrado que serve para editar, compilar e fazer upload de códigos em microcontroladores (Fezari, et al., 2018), e as bibliotecas (Özgür, 2019), (Stefano, s.d.) e (Arduino, s.d.), foi criado uma API que captura os parâmetros de frequência, comprimento do pulso e protocolo do controle interceptado que se-

gundo (Özgür, 2019) pode ser 5 tipos, posteriormente envia os parâmetros para Telegram, que é um aplicativo de mensagens para smartphones que enviar e receber texto e mensagens multimídias (de, et al., 2016). no qual possui uma função chatbot que é um usuário controlado por um microcontrolador que a partir de um programa pode-se simula um usuário humano (Org, s.d.), essa função foi utilizada para diminuir o uso de periféricos no hardware e ter uma automação digital IoT.

E por fim, o sinal capturado pode ser simulado pelo microcontrolador a partir de um comando "/Play"do Telegram Bot assinado pelo smartphone como mostrada na figura B.3.

Ao ser acionado, o microcontrolador busca na memória EEPROM, que consiste na memória não volátil do dispositivo (Harari, et al., 1994), os parâmetros do controle copiado a fim de utilizar esse sinal para poder abrir o portão eletrônico referente ao controle com o sinal copiado.

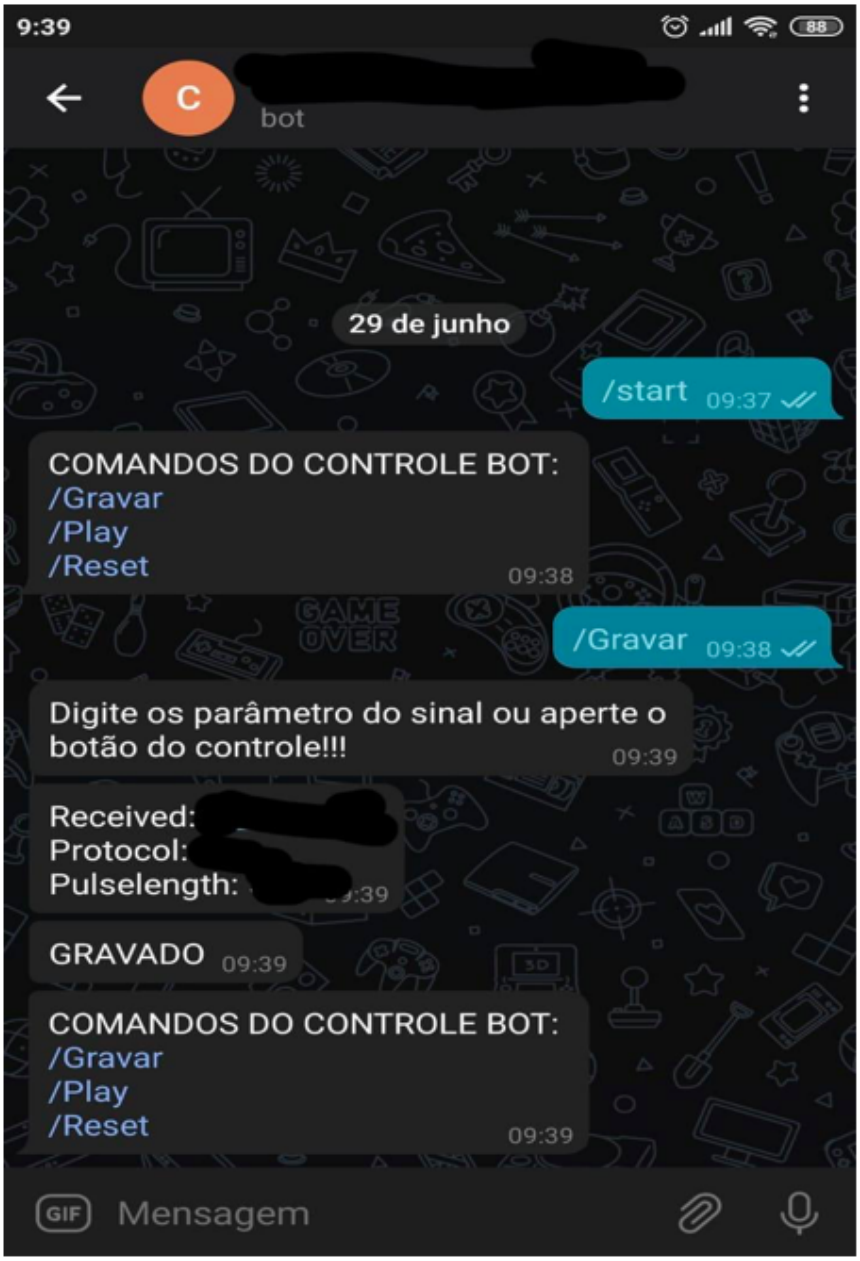

Figura B.3: Interface do Bot do Telegram.

4. RESULTADOS

O objetivo deste artigo é mostrar vulnerabilidade dos emissores de RF 433 MHz a partir de um dispositivo IoT via comandos Telegram bot. Desta forma, foram realizados dois tipos de testes, primeiro foi quanto à distância máxima de envio e recepção de dados e o segundo quanto à eficácia na clonagem dos sinais de RF 433 Mhz em ambientes controlados e com a autorização dos proprietários das residências. Primeiramente foi realizado testes quanto à distância máxima que receptor 433 MHz intercepta o sinal do controle do portão eletrônico. Foram utilizados 3 modelos de averiguação, no modelo 1 o receptor não possui antena, no modelo 2, o receptor possui uma antena senoidal de tamanho 2,5 cm e por último, no modelo 3, o receptor possui uma antena de tamanho 40 cm. Pode-se observa que o receptor 433 MHz consegue interceptar a uma distancia de 7,66 metros sem a antena, 11,4 metros com a antena senoidal de 2,5 cm e 18,5 metros com a antena de 40 cm sem anteparos. A tabela B.1 demonstra os resultados dos testes de cada modelo.

Modelos	Distância máxima (M)
modelo 1	**7,66**
modelo 2	**11,4**
modelo 3	**18,5**

Tabela B.1: Resultados dos testes de distância de interceptação de sinal do receptor 433 MHz.

Em seguida, foi feito a distância máxima do envio do sinal copiado com o emissor 433 MHz. Foi usado apenas um modelo com o emissor com a antena senoidal de 2,5 cm e nos testes deu que a distância máxima foi de +/- 20 metros sem anteparo.

Por último foram realizados testes amostrais com 15 controles eletrônicos, 5 motores de portões eletrônicos de 4 casas diferentes para mostrar a vulnerabilidade desses sistemas, onde foi possível copiar o sinal de 5 controles para verificar a eficácia do dispositivo. Para os controles que foram possíveis fazer a cópia, conseguiu abrir o portão de duas casas. No entanto, não foi possível copiar o sinal de 11 controles, pelo o fato desses controles ser criptografados com a tecnologia Rolling code.

A criptografia Rolling code consiste em um sistema de transmissão segura de RF, com um código fixo e um código rotativo de bit trinário intercalado (Farris, et al., 2000).

5. CONLCUSÃO E TRABALHOS FUTUROS

Em suma, foi construído um dispositivo IoT que consegue copiar, à distância, um sinal de controle remoto que trabalham com tecnologia de RF 433MHz para abrir portão eletrônico, depois esse dispositivo grava na memória e enviar via Telegram bot para um celular previamente cadastrado, sendo que os códigos estão sob sigilo, para evitar quebra da propriedade intelectual. Deste modo este disposto aproveita a vulnerabilidade dessa tecnologia tão comum em residências e empresas.

Os testes foram realizados em ambientes controlados e em residências autorizadas pelo os seus proprietários, desta forma com base nos testes e nos dados obtidos, que não foram mostrados

pelo o fato de ser dados sigilosos, é possível afirmar que o projeto alcançou a maioria dos objetivos propostos, dado que usando o Telegram bot e o dispositivo criado, foi possível capturar, gravar na memória e enviar no Telegram o sinal emitido por alguns controles eletrônicos e com este mesmo sinal, foi possível simular uma controle eletrônico e abrir portões eletrônicos de duas residências, dessa maneira demonstrando a vulnerabilidade de alguns dispositivos que utiliza sinal RF 433 MHz, porém alguns controles eletrônicos não foi possível de capturar o sinal, devido ao sistema de criptografia Rolling code.

Como Trabalhos futuros, pode ser realizado um sistema digital de portão eletrônico a partir de chatbot Telegram para melhorar o sistema de segurança dos portões eletrônicos.

AGRADECIMENTOS

Os autores agradecem o apoio das Agências brasileiras de pesquisa, desenvolvimento e inovação CNPq (Projetos INCT SegCiber 465741/2014-2, PQ-2 312180/2019-5 e LargEWiN BRICS2017-591), CAPES (Projetos FORTE 23038.007604/2014-69 e PROBRAL 88887.144009/2017-00) e FAPDF (Projetos UIoT 0193.001366/2016 e SSDDC 0193. 001365/2016), bem como o suporte do Laboratório LATITUDE/UnB (Projeto SDN 23106. 099441/2016-43), a cooperação com o Ministério da Economia (TEDs DIPLA 005/2016 e ENAP 083/2016), o Gabinete de Segurança Institucional da Presidência da República (TED 002/2017), a Advocacia-Geral da União (TED 697.935/2019) e o Conselho Administrativo de Defesa Econômica (TED 08700.000047/2019-14).

REFERÊNCIAS

Alves, S. I., 2018. Semáforo inteligente para veículos de emergência.. s.l.:s.n.

A., M. S. a. A., T. T. a. L. W. T., 2004. What Father Landell de Moura Used to Do in His Spare Time. Proceeding of the IEEE Transactions on the History of Electronics.

Anon., s.d. Mercado de segurança eletrônica no Brasil faturou R$ 7,17 bilhões em 2019. [Online] Available at: https://revistasegurancaeletronica.com.br/mercado-de-seguranca-eletronica-no-brasil-faturou-r-717-bilhoes-em-2019-previsao-e-de-crescimento-de-12-em-2020/[Acesso em 10 Julho 2020].

Arduino, s.d. EEPROM Library. [Online] Available at: https://www.arduino.cc/en/Reference/EEPROM

Azevedo, R. O. S. d., 2019. Sistema de gerenciamento remoto de dispositivos de Internet das Coisas protegidos por Rede Virtual Privada. s.l.:s.n.

D. O., J. C. a. S., D. H. a. N. M. P., 2016. Chatting with arduino platform through telegram bot. Em: 2016 IEEE International Symposium on Consumer Electronics (ISCE). s.l.:s.n., pp. 131-132.

F., B. L. a. F. J. J., 2000. Rolling code security system. Estados Unidos da América, Patente Nº 6,154,544.

F., M. a. A. D. A., 2018. Integrated Development Environment “IDE” For Arduino. WSN

applications, pp. 1-12.

F. R. J., 2018. Securización de Sistema de Mando a Distancia con RF Rodrigo. s.l., s.n.

Gupta, P. a. K. P. R., 2000. The capacity of wireless networks. IEEE Transactions on information theory, 46(2), pp. 388-404. H., E. a. N., R. D. a. M. S., 1994. Flash eeprom system. s.l. Patente Nº 5,297,148.

H., I. A. a. A., R. S. a. J. B. H., 2017. Design and Implementation of RFID Active Tags and Mutual Authentication Protocol with Ownership Transfer Stage. Iraqi Journal for Electrical And Electronic Engineering, 13(1), pp. 83-103.

H., P. D. a. V. B. T., 2019. Vulnerabilities in IoT Devices with Software-defined radio. Em: 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS). s.l.:IEEE, pp. 664-668.

K. et al., 2017. Innovation of a secured transmitter/receiver chain by creating a new encryption algorithm. Em: 2017 Sensors Networks Smart and Emerging Technologies (SENSET). s.l.:s.n., pp. 1-3.

K., R. K. a. M. K. S., 2016 . A low cost implementation of MQTT using ESP8266. Em: 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I). s.l.:s.n., pp. 404-408.

K. et al., 1999. Robotic wheelchair with three control modes. Em: Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C). s.l.:s.n., pp. 2590-2595.

M. V. C. d. L. O. L. H., 2018. Sistema de comunicação de dados utilizando Arduino e Módulo RF 433 MHZ. s.l.:s.n. Org, T., s.d. Telegram Bot. [Online] Available at: https://telegram.org/blog/bot-revolution

Özgür, S., 2019. Arduino libary for remote control outlet switches. [Online] Available at: https://github.com/sui77/rc-switch/

R., K. B. e. C. S., 2007. Implications of radio fingerprinting on the security of sensor networks. Em: 2007 Third International Conference on Security and Privacy in Communications Networks and the Workshops-SecureComm 2007. s.l.:s.n., pp. 331-340.

S. L., s.d. Simple Arduino Telegram BOT library for ESP8266. [Online] Available at: https://github.com/shurillu/CTBot

W. T. et al., 2016. 433 MHz (Wireless RF) communication between two Arduino UNO. American Journal of Engineering Research (AJER), 5(10), pp. 358-362.