

PROFISSA - PROGRAMMABLE FUTURE INTERNET FOR SECURE SOFTWARE ARCHITECTURES

Projeto de Pesquisa na modalidade Auxílio à Projeto Temático, chamada de Pesquisa Estratégica sobre a Internet, <http://www.fapesp.br/13757>, submetido à Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP).

Pesquisador Responsável: Lisandro Zambenedetti Granville, Dr.

Brasília, 8 de Abril de 2020

Informações Gerais do Projeto

- Título do projeto:

PROFISSA - Programmable Future Internet for Secure Software Architectures

- Pesquisador responsável:

Lisandro Zambenedetti Granville, Dr.

- Instituição sede do projeto:

Instituto de Informática da Universidade Federal do Rio Grande do Sul

- Período de vigência proposto:

1 de janeiro de 2021 a 31 de dezembro de 2026

General Information about the Project

- Title:

PROFISSA - Programmable Future Internet for Secure Software Architectures

- Researcher in charge:

Lisandro Zambenedetti Granville, Dr.

- Host institution:

Instituto de Informática at the Universidade Federal do Rio Grande do Sul

- Project proposed term:

1 January 2021 to 31 December 2026

Resumo

Com os recentes avanços em tecnologias voltadas às redes privadas seguras, a quinta geração de telecomunicação (5G) e a Internet das Coisas (IoT), as redes vem presenciando mudanças nos perfis de uso de forma inesperada com requisitos adversos, como baixa latência, segurança de fluxos ou alta resiliência a qualquer momento. Tais mudanças, quando aplicadas sobre redes programáveis podem envolver modificações no software desenvolvido para a rede (*e.g.*, P4). Técnicas de engenharia de software podem ser aplicadas para melhorar/agilizar/otimizar o desenvolvimento de software de rede, especialmente quando mudanças no perfil de rede ocorrem. Nesse caso, o projeto PROFISSA é proposto para investigar, mapear e avançar a utilização de técnicas de engenharia de software quando aplicadas às redes programáveis no intuito de melhorar a qualidade estrutural, funcional e do processo de desenvolvimento de programas de rede. O projeto proverá um arcabouço de códigos modulares e reutilizáveis de programas de rede desenvolvidos com boas práticas de desenvolvimento. Além disso, os programas de rede desenvolvidos no projeto serão executados em ambientes reais de redes programáveis implementadas sobre um testbed para desenvolvimento e análise de software de rede. Para tanto, busca-se a integração entre RNP e o projeto FABRIC, para criar ambientes de redes programáveis reais de última geração, levando em consideração, por exemplo, a utilização da programabilidade no plano de dados com a utilização de dispositivos que suportam P4.

Palavras-chaves: Redes programáveis, engenharia de software, SDN, P4, 5G, IoT, cibersegurança.

Abstract

Recent advances in technologies aimed at secure private networks, the fifth generation of telecommunication (5G) and the Internet of Things (IoT), computer networks have witnessed changes in usage profiles unexpectedly with adverse requirements, such as low latency, secure flow or high resilience at any time. Such changes, when applied over programmable networks, may involve modifications to the network software developed (*e.g.*, P4). Software engineering techniques can be applied to improve / enhance / optimize the development of network software, especially when changes in the network profile occur. In this case, the PROFISSA project is proposed to investigate, map and advance the use of software engineering techniques when applied to programmable networks in order to improve the structural and functional quality and the process of developing network programs. The project will provide a modular and reusable code framework for network programs developed with good development practices. In addition, the network programs developed in the project will be executed in real environments of programmable networks implemented on a testbed for the development and analysis of network software. To this end, the integration between RNP and the FABRIC project is sought, to create real programmable network environments of the latest generation, taking into account, for example, the use of programmability in the data plan with the use of devices that support P4.

Keywords: Programmable networks, testbed, software engineering, P4, 5G, IoT, cybersecurity.

Sumário

Informações Gerais do Projeto	i
General Information about the Project	ii
Resumo	iii
Abstract	iv
1 Enunciado do problema	1
1.1 Objetivos	2
2 Resultados esperados	5
2.1 Principais contribuições	5
2.2 Impacto do Projeto	7
3 Desafios científicos e tecnológicos e os meios e métodos para superá-los	9
3.1 Aplicação de engenharia de software em redes programáveis softwarizadas	9
3.1.1 Princípios de engenharia de software	9
3.1.2 Arquitetura conceitual do projeto PROFISSA	12
3.2 Casos de uso	13
3.2.1 Aplicações de Redes 5G/6G em Infraestrutura Programável	13
3.2.2 Aplicações de Segurança Distribuída em Infraestrutura Programável	16
3.2.3 Aplicações de Internet das Coisas sobre infraestrutura programável	19
3.3 Implantação de infraestrutura programável	22
4 Pesquisadores Principais e suas responsabilidades	26
4.1 Equipe	31
5 Cronograma	33
6 Disseminação e avaliação	35
7 Outros apoios	36
Referências bibliográficas	37

1 Enunciado do problema

As redes programáveis objetivam promover a implementação em software de partes da rede que tradicionalmente seriam implementadas através de dispositivos de hardware dedicados, também chamadas de redes definidas em software (SDN) (KELLERER et al., 2019). Tal mudança de paradigma permite uma maior flexibilidade no desenvolvimento de aplicações e protocolos de rede, reduz as barreiras de inovação e facilita a adaptação rápida da rede em resposta a mudanças flutuantes de demandas ou requisitos (FEAMSTER; REXFORD; ZEGURA, 2014). Isso permite, por exemplo, aumentar a capacidade de processamento de um dispositivo, incluir um novo mecanismo de segurança na rede, ou desenvolver um novo protocolo experimental puramente através da programabilidade e configuração de software na rede (COX et al., 2017). Concretamente, redes baseadas em software têm ganhado ampla aceitação, por exemplo em redes de telecomunicação, Internet das Coisas e redes privadas corporativas, que são ambientes heterogêneos de redes de borda e que possuem requisitos distintos, como priorização de fluxos, maior tolerância a atrasos e mitigação de tráfego malicioso (SACCO; ESPOSITO; MARCHETTO, 2020). Nesse contexto, as redes de borda precisam ser frequentemente reconfiguradas para se adaptar aos diferentes cenários existentes, bem como às modificações do perfil de uso de seus usuários. Para que este tipo de configuração seja realizada de forma eficaz e eficiente, redes programáveis definidas por software (Software-Defined Networks ou SDN) podem ser empregadas para facilitar a adaptação do comportamento das redes de borda (SARASWAT et al., 2019).

As redes programáveis são conceitualmente organizadas em quatro planos: (i) Plano de aplicação; (ii) Plano de controle; (iii) Plano de encaminhamento; e (iv) Plano de gerenciamento (MARROTTA et al., 2017). No plano de aplicação, aplicativos de camadas superiores são projetados por provedores de serviços e administradores para atender aos seus interesses e requisitos de seus clientes. Esses aplicativos eventualmente geram requisições envolvendo recursos de rede aos controladores do plano de controle. As requisições são interpretadas e convertidas em configurações refinadas pelos controladores para serem encaminhadas ao plano de encaminhamento. No plano de encaminhamento, os dispositivos de borda, como switches e rádios programáveis, recebem as configurações que ajustam seus comportamentos. Também, os comportamentos dos dispositivos de borda podem ser alterados por aplicativos de camadas inferiores instalados localmente. Por fim, no plano de gerenciamento, os componentes de uma SDN (por exemplo, aplicativos, controladores e dispositivos de encaminhamento) são monitorados e ajustados preservando a integridade de toda a rede para atender aos pilares do gerenciamento de rede, *i.e.*, gerenciamento de falhas, configuração, contabilidade, desempenho e segurança (FCAPS) (WICKBOLDT et al., 2015).

Os programas de rede se dividem entre (i) aplicativos de camadas superiores e (ii) aplicativos de camadas inferiores. Os aplicativos de camadas superiores são desenvolvidos em linguagens

de alto nível, utilizando principalmente a manipulação e reconfiguração de regras de encaminhamento de fluxos de pacotes utilizando uma visão global da rede (COX et al., 2017) (ALSAEEDI; MOHAMAD; AL-ROUBAIEY, 2019). Já os aplicativos das camadas inferiores do plano de encaminhamento são desenvolvidos para alterar o comportamento individual de dispositivos de encaminhamento, através da modificação do pipeline para processamento de pacotes, utilizando linguagens de baixo nível, como P4 (BOSSHART et al., 2014)(DING et al., 2020). Ainda que ambos os tipos de aplicativos possam ser desenvolvidos pelo mesmo desenvolvedor, estes não são necessariamente projetados para operar concomitantemente. Isto é, existe um problema ao se desenvolver programas de rede por não haver paradigmas, definições de interfaces, ou metodologias para guiar o desenvolvimento de aplicativos de rede, principalmente quando pertencentes a diferentes planos (KELLERER et al., 2019).

O processo atual de desenvolvimento de aplicações em redes programáveis tipicamente não contempla a adoção de práticas consolidadas de engenharia de software. Isso frequentemente leva à ocorrência de bugs e outros problemas de codificação nos programas desenvolvidos (FREIRE et al., 2018a) (LIU et al., 2018). Portanto, programas de rede em uma SDN precisam ser desenvolvidos a partir de um arcabouço de códigos reutilizáveis, modulares e que permitam a interoperabilidade em ambientes heterogêneos. Tais programas devem ser fundamentados nos princípios sólidos da engenharia de software, maximizando a utilização dos recursos e projetados para atender às demandas específicas de cada ambiente, sem comprometer a rede.

1.1 Objetivos

Ao passo que a programabilidade e softwarização de redes trouxe inúmeros benefícios (*e.g.*, facilidade de experimentação com novos protocolos, redução das barreiras de inovação), essa mudança de paradigma também pode comprometer o funcionamento correto da rede caso não haja a adoção de um rigoroso processo de desenvolvimento de aplicações em rede. Dessa forma, propõe-se o projeto Internet do Futuro Programável para Arquiteturas e Softwares Seguros (*Programmable Future Internet for Secure Software Architectures* - PROFISSA), cujo objetivo principal é estabelecer processos, métodos, interfaces, protocolos e modelos que facilitarão e tornarão mais confiável o desenvolvimento de software para redes programáveis.

Para se alcançar esse objetivo, destacam-se os subobjetivos e ações abaixo:

1. Investigar práticas de desenvolvimento de software de rede baseado em conceitos de engenharia de software;
2. Investigar e propor processos de gestão de ciclo de vida de desenvolvimento de programas de redes programáveis;
3. Investigar e propor metodologias e novas interfaces para interoperabilidade entre programas de rede de camadas superiores e de camadas inferiores sem comprometer o funciona-

mento dos planos de uma SDN;

4. Propor novos protocolos principalmente relacionados com a publicação de novos programas de rede, permitindo que demais programas já instalados possam coexistir;
5. Investigar e aprimorar o ciclo de vida do desenvolvimento de aplicativos de camadas inferiores de rede utilizando linguagens como, por exemplo, P4;
6. Estudar e aprimorar o ciclo de vida do desenvolvimento de aplicativos de camadas superiores de rede utilizando interfaces como, por exemplo, FlowRuleAPI, FlowObjectiveAPI e Intent API;
7. Investigar e propor boas práticas de gestão no desenvolvimento de softwares de rede;
8. Investigar, testar e propor novas técnicas de correção de erros e bug-fixers;
9. Investigar e propor novos modelos de negócio e fluxo de trabalho (*workflows*) para aprimorar o desenvolvimento de software de rede;
10. Investigar a eficácia e eficiência da metodologia sendo proposta, considerando diferentes casos de uso com diferentes perfis de rede e objetivos para teste e análise dos softwares de rede desenvolvidos;
11. Analisar os benchmarks e conjunto de dados existentes para testar e avaliar a metodologia empregada em ambiente simulado ou emulado dos softwares de rede desenvolvidos;
12. Projetar plataformas de desenvolvimento de software de rede baseadas em fundamentos de engenharia de software em infraestrutura real;
13. Desenvolver e analisar programas de rede desenvolvidos sobre infraestruturas reais providas pela RNP e FABRIC;
14. Definir casos de uso e perfis de rede, considerando cenários de IoT, redes sem-fio avançadas (5G/6G), redes corporativas privadas, entre outros para avaliar os métodos propostos;
15. Desenvolver e publicar programas de rede baseados em 5G/6G;
16. Desenvolver e publicar programas de rede baseados em cibersegurança;
17. Desenvolver e publicar programas de rede baseados em IoT;
18. Disponibilizar infraestrutura para experimentação aberta ao público de ensino e pesquisa;
19. Explorar e validar a infraestrutura integrada entre RNP e FABRIC de redes programáveis para experimentação e geração de provas de conceito.

A partir do atendimento das ações e objetivos supra-citados, o projeto PROFISSA irá estruturar e conceber boas práticas de gestão e de desenvolvimento de software voltados às redes programáveis. Uma lista dos resultados esperados no projeto é apresentada e descrita a seguir.

2 Resultados esperados

Nessa seção, apresentam-se as principais contribuições previstas para esse projeto. Para cada contribuição, são indicadas as etapas relacionadas à tal contribuição.

2.1 Principais contribuições

Como resultado das investigações propostas nesse projeto de pesquisa, almeja-se investigar, definir e avançar as técnicas de engenharia de software quando aplicadas ao desenvolvimento de programas de rede nas diferentes redes programáveis. A partir da aplicação dessas técnicas espera-se vencer desafios científicos e tecnológicos na provisão de infraestrutura para diferentes casos de uso de redes programáveis. Para tal, espera-se alcançar as seguintes contribuições:

Classificação do estado da arte quanto à aplicação de engenharia de software em redes programáveis, considerando tanto a programabilidade de regras de encaminhamento (aplicações de camada superior) como a programabilidade dos switches propriamente ditos (aplicações de camadas inferiores), e sistematizando as principais técnicas e estratégias comumente utilizadas. Essa contribuição será apresentada na forma de uma revisão de literatura agrupando as principais soluções existentes nessa área. Essa contribuição será obtida a partir da conclusão do marco 2 no cronograma.

Prospecção e avanço de técnicas para maximizar a qualidade estrutural de módulos que compõem programas de camadas inferiores e de camadas superiores. As principais técnicas identificadas na literatura serão mapeadas e aprimoradas para que permitam melhorar a qualidade do código escrito e o quão bem estruturado ele é ao ser desenvolvido para redes programáveis. Por exemplo, aplicações de teste de trecho de código, bem como checadores léxicos e de sintaxe serão desenvolvidos voltados para o desenvolvimento de programas de rede. Essa contribuição será obtida a partir da conclusão dos marcos 4, 5 e 6 do cronograma, considerando os desafios de engenharia de software e casos de uso envolvendo redes privadas seguras, IoT e 5G/6G.

Prospecção e avanço de técnicas para maximizar a qualidade funcional de módulos que compõem programas de camadas inferiores e de camadas superiores. As técnicas serão baseadas no levantamento do estado da arte, considerando-se as limitações das soluções já existentes. Técnicas que permitam a verificação automática da integridade e identificação de potenciais conflitos, inconsistências ou dependências de regras em planos de dados programáveis serão desenvolvidas. Por exemplo, esse desenvolvimento resultará na definição concreta de testes unitários, de carga e de segurança voltados para programas de rede. Essa contribuição será obtida a partir da conclusão dos marcos 3, 4 e 6 do cronograma, considerando os desafios de engenharia de software e casos de uso envolvendo redes privadas seguras, IoT e 5G/6G.

Prospecção e avanço de técnicas para maximizar a qualidade de processo de desenvolvimento de software para redes programáveis. Após o levantamento bibliográfico, uma definição das técnicas e estratégias a serem adotadas que tornem o processo de implementação de programas de rede o mais equilibrado possível. Nesse aspecto, clientes e participantes do projeto são os mais afetados, justamente pelo fato de que essa qualidade tem um impacto direto nos prazos e requisitos do produto. Por exemplo, técnicas de planejamento e gestão de projetos, como scrum, waterfall, scrumterfall, entre outras, serão personalizadas para o desenvolvimento de programas de rede, garantindo melhores resultados e entregas dos mesmos. Essa contribuição será obtida a partir da conclusão dos marcos 2, 3, 4, 5 e 6 do cronograma, considerando os desafios de engenharia de software e casos de uso envolvendo redes privadas seguras, IoT e 5G/6G.

Sistematização do processo de correção de erros que devem ser garantidas em redes programáveis, juntamente a um conjunto de estudos de caso que ilustrem essas situações. Essa contribuição será apresentada na forma de uma taxonomia ou de um survey cobrindo as principais técnicas de projeto para eliminação de erros. A partir da sistematização do processo de correção de erros será possível definir um escopo para a abordagem dos aspectos mais críticos relacionados à correção de erros em redes programáveis e quais boas práticas devem ser utilizadas para evitar a ocorrência dos mesmos. Essa contribuição será obtida a partir da conclusão dos marcos 3, 5, 6 e 7 do cronograma.

Desenvolvimento de um arcabouço de código reutilizável, que permita a aceleração do processo de desenvolvimento de programas de rede, validação automática de regras a serem inseridas em switches, desenvolvimento de módulos P4. O arcabouço deve apresentar garantias em relação a consistência das funcionalidades de rede para sua utilização em ambientes reais. O mesmo será avaliado de forma analítica e experimental, de modo a observar a robustez dos códigos propostos que irão compor o arcabouço. Essa contribuição será obtida a partir da conclusão dos marcos 5, 6 e 7 do cronograma.

Transposição das técnicas de engenharia de software para casos de uso de redes programáveis. Essas técnicas envolvem a qualidade no processo de desenvolvimento, a correção de erros e a reutilização de código, que devem ser aplicadas em três casos de uso em áreas emergentes com potencial de obtenção de benefícios significativos: redes 5G/6G, segurança distribuída e sistemas IoT. O processo de desenvolvimento das técnicas mencionadas será baseado nos requisitos, características e restrições dos casos de uso, para aproximar a área de aplicação com o plano de dados programável da rede em cenários reais.

Avaliação quantitativa e qualitativa do processo de pavimentação da lacuna entre o desenvolvimento de serviços de rede de alto nível (5g/6G, segurança e IoT) em código para o plano de dados programável em redes de testbed em laboratórios, assim como na infraestrutura programável disponibilizada pela RNP em colaboração com o FABRIC. A avaliação de desempenho (quantitativa) fornecerá subsídios para compreender os tradeoffs existentes no uso de diferentes aplicações distribuídas sobre uma infraestrutura programável, com relação à métricas de QoS de rede e QoE do usuário. A avaliação funcional (qualitativa) fornecerá subsídios para compreen-

der os impactos das decisões sobre o processo de desenvolvimento de software na infraestrutura programável de rede. Ambas avaliações serão realizadas durante a execução do projeto e gerarão importantes feedbacks para o aprimoramento constante do processo de desenvolvimento de código para redes programáveis.

Por fim, no que se refere aos **insumos de pesquisa e disseminação de conhecimento** ao longo da execução desse projeto, os seguintes periódicos serão objetivados para publicação de artigos científicos: Elsevier Computer Networks, IEEE Communications Magazine, IEEE Surveys & Tutorials, IEEE Networks, Springer Journal of Network and Systems Management, Wiley International Journal of Network Management, IEEE Transactions on Network and Service Management, entre outros. Além disso, o projeto terá as seguintes conferências como objetivos para publicações: International Conference on emerging Networking EXperiments and Technologies (CoNEXT), ACM Symposium on SDN Research (SOSR), IEEE International Conference on Computer Communications (INFOCOM), IEEE Global Communications Conference (GLOBECOM), IEEE International Conference on Communications (ICC), IEEE/IFIP Network Operations and Management Symposium (NOMS), International Conference on Network and Service Management (CNSM), IFIP/IEEE International Symposium on Integrated Network Management (IM), IEEE World Forum on Internet of Things (WF-IoT), Global IoT Summit (GIoTS), Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg), entre outros. Além de artigos científicos, prevê-se a criação de cursos, Webinars, além de publicação em canais abertos de material de educação e divulgação de software livre para disseminar o conhecimento gerado no projeto. Essas contribuições serão o foco das **atividades 8 e 9** do cronograma.

2.2 Impacto do Projeto

Redes de computadores e a Internet são essenciais para o funcionamento dos mais diversos setores da sociedade atual. Com a introdução de programabilidade nessas redes, vários benefícios podem ser elencados para os administradores de rede, como maior manutenibilidade dos dispositivos e oportunidade de inovação em curto período tempo sem a necessidade de aguardar atualizações lançadas pelos fabricantes de equipamentos. Todavia, a popularização do desenvolvimento de programas para redes traz vários riscos, principalmente por não haver uma definição clara de como o ciclo de desenvolvimento dos programas de rede deva ser realizado. Além disso, a ausência de definições acerca de como e quais técnicas de engenharia de software devam ser utilizadas, levam a uma maior chance de se introduzir inconsistências, erros e falhas de projeto no desenvolvimento de programas de rede. Dessa forma, torna-se necessário garantir que o processo de desenvolvimento de software seja conduzido de forma adequada para evitar problemas ao se implantar novos programas de rede.

Ao se elaborar guias, definições, modelos e boas práticas para o desenvolvimento de software

de rede, o projeto PROFISSA auxiliará desenvolvedores a evitar a inserção de inconsistências, erros e problemas. Além disso, PROFISSA beneficiará não só os desenvolvedores de programas de rede pertencentes a grande empresas, mas também proverá guias de desenvolvimento para desenvolvedores autônomos e pesquisadores de redes de computadores. Os desenvolvedores poderão criar novos protocolos de rede e de controle de acesso ao meio, algoritmos de análise profunda de pacotes, firewalls, entre outros. PROFISSA irá prover diretrizes aos desenvolvedores levando em consideração técnicas de engenharia de software que ajudarão a depurar os códigos sendo criados, utilizando testes unitários e técnicas de planejamento de projetos, bem como prover um arcabouço de códigos reutilizáveis, voltados para programas de rede. PROFISSA também pode ser útil para fornecer atualizações consistentes de códigos para redes programáveis, utilizando diferentes abordagens e linguagens de desenvolvimento, *e.g.*, P4, e regras de encaminhamento, *e.g.*, FlowRuleAPI. Além disso, a integração do projeto em questão à infraestrutura do projeto FABRIC propiciará não apenas um testbed para desenvolvimento e análise de software de rede no mundo real, mas também o intercâmbio de conhecimento entre os diversos parceiros do projeto. Um resultado secundário do projeto é promover a participação da comunidade brasileira da área de redes de computadores no projeto FABRIC. Postula-se, portanto, que a presente proposta tem grande potencial para avançar de forma significativa a fronteira do conhecimento referente ao desenvolvimento de software em redes programáveis, bem como a inexplorada área de engenharia de software aplicada às redes programáveis.

3 Desafios científicos e tecnológicos e os meios e métodos para superá-los

Os desafios a serem atacados no projeto PROFISSA podem ser organizados em: (i) desafios relacionados com a aplicação de engenharia de software em redes programáveis; e (ii) desafios e oportunidades relacionados com a utilização da engenharia de software em diferentes casos de uso. Os casos de uso podem ser divididos em três seções (a) aplicações para tomada de decisões em 5G; (b) aplicações para segurança distribuída de redes; e (c) aplicações para infraestrutura programável para a Internet das Coisas. Por fim, os meios tecnológicos e a infraestrutura a ser utilizada no projeto PROFISSA são declarados na Seção 3.3.

3.1 Aplicação de engenharia de software em redes programáveis softwarizadas

A popularização das redes programáveis definidas em software tem proporcionado a criação de um grande volume de código por desenvolvedores independentes (LOPES et al., 2016), assim como acontece em ambientes de código livre (GURBANI; GARVERT; HERBSLEB, 2006). Isso traz a necessidade da adoção de práticas de Engenharia de Software, para garantir que o software produzido seja de qualidade tanto do ponto de vista funcional como de código fonte. Ter código manutenível é fundamental para a redução da probabilidade de introdução de bugs, bem como a redução dos custos e tempo na evolução de software (MENS; DEMEYER, 2008) e as mesmas necessidades se destacam nas redes programáveis (SARASWAT et al., 2019).

3.1.1 Princípios de engenharia de software

Artefatos de código desenvolvidos em linguagens focadas na programabilidade de redes podem ser explorados para identificar boas e más práticas que estão sendo adotadas pela comunidade. Isso permitirá observar desafios sendo enfrentados pelos desenvolvedores. Dentre as técnicas de Engenharia de Software que podem ser empregadas nesse contexto, pode-se destacar:

1. Derivação de elementos de software e boas práticas que possam ser reutilizadas em diferentes tipos de aplicações (COX et al., 2017);
2. Avaliação de como princípios de desenvolvimento de software são seguidos, tais como modularização (PARNAS, 1972) e separação de interesses (PESSOA et al., 2017);

3. Identificação de problemas de projeto e más práticas de desenvolvimento na forma de *code smells* (REFACTORING:..., 1999) que aparecem no código produzido (TUFANO et al., 2017);
4. Identificação de contextos onde bugs no código aparecem com maior frequência e de formas de evitá-los (ZHANG et al., 2020).

Assim, técnicas e ferramentas de Engenharia de Software que dão suporte para resolver problemas nessas direções podem ser utilizadas, adaptadas e propostas para o contexto de SDN. Exemplos são linguagens específicas de domínio (FOWLER, 2010) (LOPES et al., 2016) tanto no nível de modelagem como de implementação, métricas de código (LANZA; MARINESCU, 2010), identificação de code smells de forma automatizada (PAIVA et al., 2017), e preditores de bugs (D'AMBROS; LANZA; ROBBES, 2012).

Em particular, as abstrações disponíveis em uma SDN possibilitam implantar qualquer caso de uso, a maioria deles por meio de componentes de software no controlador. No entanto, esse processo envolve uma variedade de técnicas e desafios. Como a SDN é uma mudança de paradigma da arquitetura de redes na qual a rede é definida por meio de um conjunto de camadas de software, ela requer o uso de linguagens de programação formais, especificações e técnicas de engenharia de software para superar os desafios comuns na indústria de software.

Dessa forma, o software pode ser criado usando uma linguagem específica de domínio (DSL) e / ou uma linguagem de uso geral (GPL). A primeira é apropriada para um domínio de aplicativo específico, enquanto a última é aplicável entre diferentes domínios. Ou seja, uma DSL tende a oferecer construções mais abstratas e fáceis de usar do que uma GPL. A vantagem das linguagens SDN baseadas em DSL é abstrair sua complexidade para os desenvolvedores, geralmente através do uso de muitos recursos e métodos criados para um domínio específico (LOPES et al., 2016).

Vale ressaltar que uma DSL pode ser textual ou visual (também chamada de Linguagens de Modelagem Específicas de Domínio - DSML) e, em ambos os casos, a engenharia usada para criar uma DSL ou mesmo uma DSML deve definir: (1) uma sintaxe abstrata (isto é, gramática), (2) uma sintaxe concreta (isto é, representação), (3) mapeamentos entre sintaxe abstrata e concreta e (4) uma descrição da semântica. Considerando que linguagens de modelagem tendem a fornecer abstrações de nível superior às textuais, as DSMLs tendem a ser mais fáceis do que as DSLs, especialmente porque a notação visual de uma DSML visa abstrair a representação textual de uma DSL, ocultando detalhes de implementação.

Além disso, as DSMLs são apoiadas por ferramentas que podem: i) detectar erros sintáticos e semânticos antecipadamente, ou até impedir que eles aconteçam, ii) orientar a equipe de engenharia da aplicação para a adoção de padrões de projeto apropriados; iii) verificar a integridade estrutural informando sobre elementos ausentes; iv) reduzir o trabalho de modelagem aplicando convenções ou boas práticas de programação, v) dar suporte à geração completa de código e vi) manter as especificações em diferentes níveis de abstrações consistentes. No escopo de DSML, Model-Driven Engineering (MDE) é definido como um paradigma de desenvolvimento de software que usa modelos como artefatos principais em todas as fases do desenvolvimento. Portanto,

no MDE, os modelos são mais do que itens de documentação; eles são objetos executáveis. Em outras palavras, uma vez criados os modelos, o código de destino pode ser gerado e compilado ou interpretado para execução.

Defendemos que uma DSML é de suma importância como tópico de pesquisa para criar aplicativos SDN em um alto nível de abstração. Por exemplo, uma DSML para SDN pode modelar todas as políticas que compõem o comportamento de rede desejado por um operador ou uma organização (LOPES et al., 2016). Além disso, um modelo de política baseado em DSML pode gerar código para qualquer linguagem ou controlador SDN, aumentando a compatibilidade de políticas em diferentes cenários. Portanto, a especificação de uma DSML para SDN pode facilitar o gerenciamento da rede, devido à sua capacidade de permitir a implementação gráfica de alto nível do comportamento e aplicativos da rede. Além disso, uma especificação DSML avalia vários conceitos e relacionamentos no domínio SDN, fornecendo uma plataforma robusta na qual os operadores ou desenvolvedores de rede podem confiar.

Além disso, a correteza de um aplicativo ou de um ambiente SDN está relacionada ao comportamento da rede, que deve estar de acordo com os requisitos definidos pelo operador de rede. Para conseguir isso, um determinado aplicativo em execução no controlador precisa verificar e validar todas as regras que podem ser aplicadas à rede. Essa verificação e validação pode ser fornecida como recurso por uma linguagem de programação SDN ou mesmo por uma DSML que forneça as abstrações necessárias para antecipar uma análise quantitativa da qualidade oferecida pela rede. Por exemplo, se o operador de rede quiser restringir certo intervalo de endereços IP, o controlador precisará de um aplicativo que identifique se existe alguma outra regra que será afetada por essa restrição. Além disso, o aplicativo (ou seu controlador subjacente) precisa decidir se esse operador tem privilégios para implantar a regra. Dessa forma, como a arquitetura SDN permite a programação da rede, questões relacionadas à escalabilidade, desempenho, confiabilidade e segurança precisam ser tratadas adequadamente. Em particular, atributos de dependabilidade (por exemplo, disponibilidade e confiabilidade). A alta disponibilidade e a capacidade de recuperar rapidamente de falhas são essenciais para a adoção de SDNs em larga escala.

Ao se expandir o conceito de MDE para otimizar o atendimento de objetivos específicos, *Goal-Oriented Requirements Engineering* (GORE) oferece meios comprovados para decompor requisitos técnicos e não técnicos em entidades bem definidas (objetivos) e raciocinar sobre as alternativas para atendê-los. Por isso, tem sido usado como um meio de modelar e raciocinar sobre a capacidade do sistema de se adaptar às mudanças em ambientes dinâmicos.

Rodrigues et al. (RODRIGUES et al., 2019) propõe um arcabouço fim-a-fim para o planejamento de deployment em ambientes dinâmicos e heterogêneos, onde são aplicados conceitos de MDE, a partir do modelo de objetivos. Adicionalmente, para reconhecer e gerenciar incertezas no processo de garantia de qualidade de serviço desde o início do ciclo de vida do sistema, Solano et al. (FELIX SOLANO et al., 2019) propõem uma abordagem e um arcabouço orientados a objetivos de ponta a ponta seguindo a idéia de *control feedback* (ASTRÖM; MURRAY, 2010; HELLERSTEIN et al., 2004), onde o controlador (i.e. sistema de gerenciamento) monitora continuamente os sta-

tus de custos e de confiabilidade do sistema gerenciado, bem como as condições de contexto para resolver as incertezas parametrizadas nas funções de custo e de confiabilidade. Em seguida, funções de qualidade são utilizadas em tempo de execução para (i) avaliar a confiabilidade e o custo do sistema e (ii) avaliar as ações de políticas que devem ser acionadas para atingir os objetivos, orientando assim as decisões de adaptação do sistema gerenciado.

No projeto PROFISSA propõe-se a utilização dos vários conceitos supra citados e boas práticas de engenharia de software para beneficiar a programabilidade da rede, na busca de otimizar o desenvolvimento de programas de rede. Além disso, pretende-se conceber meios para integrar as camadas da SDN com o gerenciamento da programabilidade da rede em si por meio das interfaces de gerenciamento utilizando as técnicas de MDE e GORE.

3.1.2 Arquitetura conceitual do projeto PROFISSA

Na Figura 3.1, uma arquitetura conceitual do projeto PROFISSA é apresentada. Os quatro planos de uma rede programável baseada em SDN, bem como seus componentes, são apresentados em tons de cinza (componentes utilizados no projeto, mas não necessariamente desenvolvidos no mesmo). Os aplicativos de camadas superiores são apresentados em tons de vermelho. Já, os aplicativos de camadas inferiores são apresentados em tons de verde. Ao inserir práticas de engenharia de software e controle do ciclo de vida de desenvolvimento de programas de redes programáveis, um quinto plano é introduzido em tons de amarelo, o plano de programabilidade, sendo este o plano principal a ser contemplado pelo projeto PROFISSA.

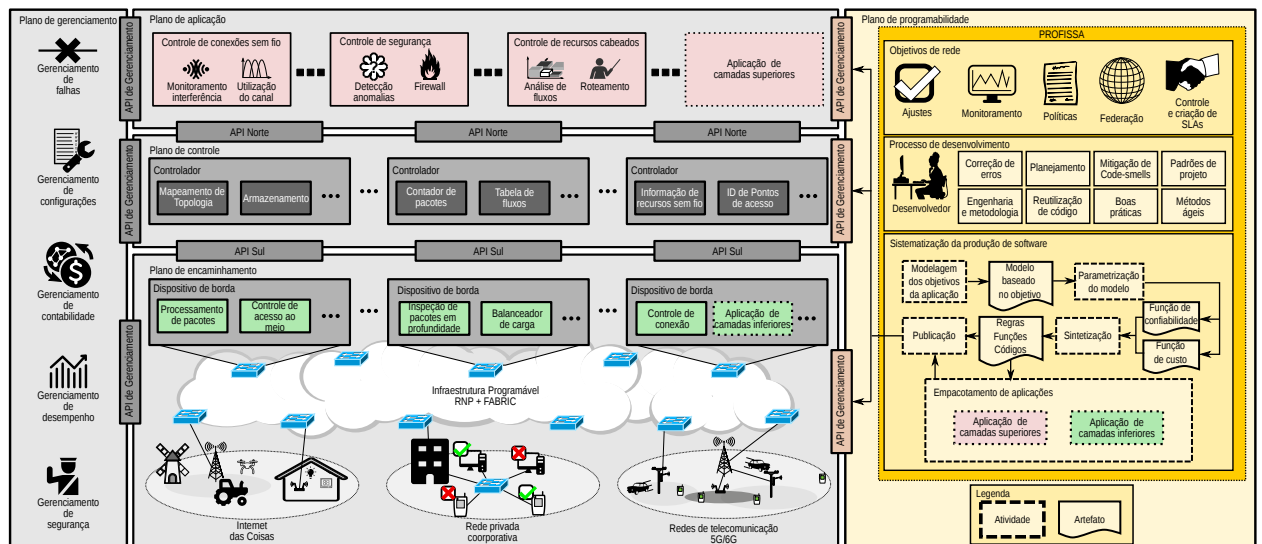


Figura 3.1: Aplicação de processos de desenvolvimento em redes programáveis.

O plano de programabilidade engloba todo o desenvolvimento de aplicações de rede contemplando seus principais objetivos: (a) reajuste dinâmico, (b) monitoramento contínuo, (c) aplicação de políticas, (d) criação de uma rede federada, (e) controle de qualidade, e (f) garantias de *Service-Level Agreements*(SLA). Considerando os objetivos de rede, inicia-se o processo de desen-

volvimento de softwares para redes programáveis. Esse processo deve ser realizado contemplando vários pilares de engenharia de software, como correção de erros (bug-fixers), utilização de padrões de projeto (*design patterns*), boas práticas de desenvolvimento e etc. O processo de desenvolvimento pode ser sistematizado ao se aplicar técnicas de GDE ou MDE. Por exemplo, cria-se um modelo baseado no objetivo de rede, que deverá ser parametrizado, gerando funções objetivo de confiabilidade e custo. Por fim, sintetizam-se os algoritmos, criam-se regras e/ou códigos, que deverão ser empacotados em aplicações a serem publicadas nas redes programáveis. Esse processo, ainda que simplificado, já permite uma visão inicial do controle do ciclo de vida do desenvolvimento de programas de rede. Mas, para que tal sistematização, bem como sua aplicação, utilização, refinamento e potencial utilização precisa ser cautelosamente definida ao longo do projeto PROFISSA. Em resumo, existe uma grande variedade de contribuições feitas no contexto de Engenharia de Software que merecem ser exploradas para permitir que o software produzido no contexto de redes programáveis tenha uma melhor qualidade, custo e tempo de evolução.

Para melhor investigar os benefícios da Engenharia de Software aplicada às redes programáveis e atacar os desafios destacados nessa proposta serão elaborados três casos de uso no projeto PROFISSA considerando (i) redes de telecomunicação da próxima geração 5G e 6G; (ii) redes privadas seguras e (iii) redes programáveis da Internet das Coisas (IoT). As investigações, protótipos, arcabouço de software e artigos científicos gerados durante a pesquisa e desenvolvimento de cada um desses casos de uso, servirão como insumo e prova de conceitos desse projeto. Para que o projeto PROFISSA possa ser executado, propõe-se, também, a implantação e utilização de uma infraestrutura programável a partir da colaboração entre RNP, FABRIC e UFRGS como meio para se experimentar a introdução do plano de programabilidade em redes programáveis no projeto PROFISSA. Esses aspectos serão discutidos nas próximas seções.

3.2 Casos de uso

No projeto PROFISSA, a engenharia de software aplicada para a criação de programas de redes programáveis é indispensável para a solução de problemas em diferentes áreas de pesquisa. Dessa forma, como prova de conceitos da utilização dessa engenharia, destacam-se três casos de usos de redes programáveis: (i) aplicações para tomada de decisões em 5G/6G; (ii) aplicações para segurança distribuída de redes; e (iii) aplicações para infraestrutura programável para a Internet das Coisas. Cada um dos casos de uso propostos é motivado, descrito e apresentado a seguir.

3.2.1 Aplicações de Redes 5G/6G em Infraestrutura Programável

Com o lançamento recente do primeiro padrão para redes da quinta geração 5G (3RD GENERATION PARTNERSHIP PROJECT (3GPP), 2015), a comunidade científica iniciou a pesquisa para o desenvolvimento da sexta geração de telecomunicação (6G). A infraestrutura de comunicação 5G/6G vem gradualmente incorporando o conceito de redes programáveis e computação de borda

levando a criação das redes *Fog Radio Access Networks* (F-RAN) (JIANG et al., 2019). Uma F-RAN é o resultado da aplicação do paradigma de computação em nevoeiro em uma Cloud-RAN (C-RAN) (YOUSEFPOUR et al., 2019). Esse tipo de rede herda componentes e aspectos da arquitetura centralizada da C-RAN, como o repositório de unidades de processamento de banda base (BBU pool) que processa a carga de trabalho das antenas, *Remote Radio Heads* (RRH), geograficamente distribuídas, que, por sua vez, comunicam-se com os equipamentos dos usuários (UE). Seguindo um paradigma de computação em nevoeiro, a F-RAN distribui as principais funções, como computação, armazenamento, comunicação e controle, estendendo o processamento até a borda da rede usando *Micro Datacenters* (MDC) conectados através de redes programáveis, transformando uma RRH típica em uma Fog-RRH. A arquitetura de rede F-RAN pode ser vista como uma evolução da C-RAN, buscando alcançar um equilíbrio entre centralização e distribuição de recursos computacionais.

F-RAN pode otimizar o desempenho da rede dinamicamente através da programabilidade em rede, aproveitando-se do poder de processamento próximo à borda, quando disponível. Nesse caso, os provedores de aplicações, como Netflix, Amazon, Twitter e Facebook, podem participar das decisões dinâmicas a serem tomadas na F-RAN para otimizar a prestação de serviço aos usuários, como mostrado na Figura 3.7. Cada tomada de decisão na F-RAN exige conhecimento atualizado das condições da rede e disponibilidade de recursos resultando em vários desafios em aberto, incluindo a tomada de decisões sobre o cache de borda (JIANG et al., 2018), alocação de máquinas virtuais de BBU (YU et al., 2016) e minimização do consumo de energia por meio do gerenciamento de recursos (CHIEN; LAI; CHAO, 2019).

Cada uma das decisões envolve a comunicação entre vários nodos da rede de encaminhamento. Por exemplo, a utilização de cache só pode ser aplicado eficientemente para o caso em que os nodos da rede possuem aplicações capazes de executar a troca de conteúdo dinamicamente através da rede. Já para a alocação de máquinas virtuais e migração das mesmas, a rede deve estar programada para aplicar redundância de conteúdo numa escala de tempo de segundos. Por fim, o consumo de energia só poderá ser minimizado de forma eficiente se a rede estiver programada para distribuir a carga de trabalho entre antenas para que algumas delas possam ser desligadas. Para cada um dos exemplos citados, o comportamento da rede 5G/6G deve ser radicalmente alterado através de aplicativos de rede em operação.

As mudanças radicais do comportamento dos dispositivos de uma F-RAN podem ocorrer em paralelo dada a execução de outros programas de rede ao mesmo tempo, gerando conflitos. Para evitar tais conflitos, durante a fase de planejamento dos programas de rede, os desenvolvedores podem, por exemplo, utilizar métodos ágeis no desenvolvimento de software. Nesses métodos, existe a prática comum de se convidar o cliente para testar um produto mínimo viável (MVP) a cada fase de desenvolvimento. A partir dessa prática, os desenvolvedores de programas de rede diferentes podem se tornar o cliente dos demais e vice-versa participando do desenvolvimento dos programas de rede a serem instalados em uma F-RAN. Além disso, outras técnicas de engenharia de software e gestão de projetos, e.g., *Busines Model Canvas*, podem ser utilizadas para alterar o

- A partir de uma metodologia padrão, criar um arcabouço de software para incentivar as práticas de reuso e desenvolvimento de programa de rede voltados a troca de conteúdo dinâmica, considerando a necessidade de redundância de conteúdo numa escala de tempo de segundos.
- A partir de uma metodologia padrão, criar um arcabouço de software para incentivar as práticas de reuso e desenvolvimento de programa de rede capaz de distribuir as cargas de trabalho das antenas, minimizando a sobrecarga na rede e o consumo de energia.

Considerando as etapas destacadas, os pesquisadores da Universidade Federal do Rio de Janeiro (UFRJ) coordenarão o esforço para investigar potenciais soluções para o desenvolvimento de programas de rede para uma 5G/6G programável. A Universidade de Brasília (UnB) e a Universidade Federal do Rio Grande do Sul (UFRGS) irão colaborar na investigação e desenvolvimento desse caso de uso. No primeiro ano de projeto, prevê-se a investigação de padrões, modelos e arcabouços de software para desenvolver programas de rede que possam influenciar as decisões sendo tomadas na rede programável F-RAN da 5G/6G. Em paralelo, estudos e propostas de mecanismos para coexistência de programas de rede considerando diferentes operadoras e provedores de aplicação serão investigados. Já, no segundo ano de projeto, prevê-se o desenvolvimento e teste de protótipos de aplicação de rede voltados para a otimização de cache e migração de cargas de trabalho de antenas. Posteriormente, no terceiro ano, prevê-se a prototipação de programas de rede voltados a redundância de tráfego em rede para migração de imagens virtuais na escala de segundos. A partir dos protótipos desenvolvidos, no quarto ano, estima-se que a coexistência das aplicações em uma mesma infraestrutura possa ser testada e resultados servirão de prova de conceitos em relação às metodologias, padrões e modelos propostos. Por fim, no quinto ano, prevê-se o teste, processo de depuração, refinamento e divulgação das metodologias, métodos e programas desenvolvidos durante a execução desse caso de uso, bem como a mensuração do desempenho otimizado fim-a-fim de uma rede programável F-RAN da 5G/6G sobre a infraestrutura de testbeds da RNP+FABRIC.

3.2.2 Aplicações de Segurança Distribuída em Infraestrutura Programável

O Relatório da Symantec de ameaças à segurança da Internet (SYMANTEC, 2019) indica um aumento de 56% no número de ataques na web em 2019. Atividades maliciosas como ataques por verificações de rede, negação de serviço e de força bruta estão entre as ameaças mais comuns identificadas. Tais atividades maliciosas ameaçam não apenas indivíduos, mas também algumas organizações coletivas, como instituições de saúde pública, financeiras e governamentais. Nesse contexto, os sistemas de detecção de intrusão de rede (NIDS) desempenham um papel importante como ferramentas para identificar ameaças em potencial.

Existem duas abordagens diferentes para NIDS em relação ao tipo de dados analisados: baseado em pacotes e baseado em fluxo. No primeiro, a inspeção profunda de pacotes (DPI) é realizada levando em consideração as cargas individuais de pacotes, bem como as informações do cabeçalho. No segundo, os fluxos, como coleções de pacotes, são analisados em relação às suas propriedades, por exemplo, duração, número de pacotes, número de bytes e porta de origem/destino. Para executar a classificação de tráfego em tempo real, é necessário analisar um grande volume de dados, impedindo a utilização de mecanismos mais precisos e complexos, como DPIs baseados em redes neurais. Como as abordagens baseadas em fluxo podem classificar todo o tráfego que inspeciona um equivalente a 0,1% do volume total (SPEROTTO et al., 2010), NIDS com base na análise de fluxo surgem como boas soluções para classificação de tráfego em tempo real.

Nos últimos anos, diferentes classificadores baseados em fluxo foram propostos com base em Inteligência Artificial, como aprendizado superficial e profundo. De acordo com o relatório em (Umer2017), os melhores classificadores baseados em fluxo atingem cerca de 99% de precisão. Embora bastante precisos, os classificadores de fluxo atuam como uma primeira barreira para detecção de ataques, sendo que todo o tráfego classificado como malicioso ainda precisa passar por um DPI para análise profunda. Dessa forma, cria-se um sistema distribuído de segurança, onde classificadores de fluxo devem ser associados a DPI instalados nos nós da rede. O posicionamento de ambos, classificadores baseados em fluxos e DPIs, pode ser otimizado quando colocados em pontos estratégicos da rede. Entretanto, baseado no comportamento e tráfego da rede, o posicionamento ótimo pode ser alterado devendo ser recalculado. Dessa forma, cria-se a necessidade de se desenvolver um sistema de segurança distribuído, onde os classificadores baseados em fluxo podem ser dinamicamente instanciados na topologia da rede e conectados a DPIs.

A programabilidade de rede aliada a possibilidade de se instanciar e remover dinamicamente os aplicativos de rede, permite a criação de sistemas de segurança distribuídos, como apresentado na Figura 3.3. Um sistema de gestão de redes programáveis pode fazer a implantação de um controlador para tomada de decisão sobre a distribuição das aplicações que irão classificar fluxos ou realizar inspeção de pacotes DPI. O controlador ciente da topologia de rede ao detectar comportamentos diferentes no tráfego de rede, sendo de rajadas ou de média e alta vazão, poderá decidir quais os nós de rede são mais interessantes para acomodar um classificador de fluxos ou uma aplicação de análise de pacotes DPI. O classificador de fluxos pode ser instanciado a partir de protocolos como o Openflow. Já, uma aplicação de análise de pacotes pode ser desenvolvida utilizando tecnologias P4. Ainda que seja possível desenvolver um sistema de segurança distribuído, vários desafios de pesquisa e desenvolvimento precisam ser atacados, como seguem.

- A partir de uma metodologia padrão, criar um arcabouço de software para otimizar e planejar o desenvolvimento um classificador de fluxos para redes programáveis minimizando a sobrecarga de comunicação com o controlador.
- A partir de uma metodologia padrão, criar um arcabouço de software para otimizar e plane-

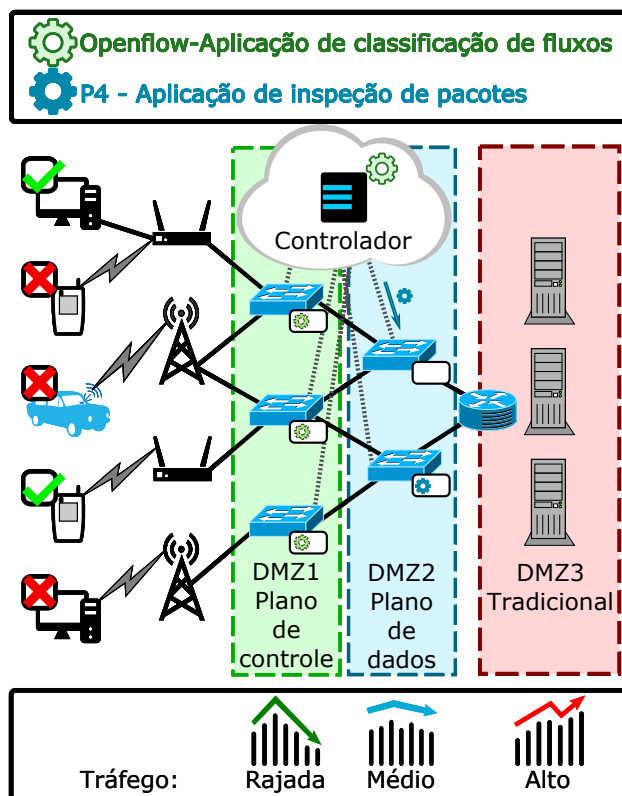


Figura 3.3: Cenário voltado a aplicações de segurança distribuída

jar o desenvolvimento uma aplicação P4 para DPI sem sobrecarregar a memória disponível em um switch programável.

- Otimizar a decisão de posicionamento das aplicações na topologia da rede programável.
- Mensurar os ganhos fim-a-fim ao se utilizar um sistema distribuído de segurança.

A partir da utilização de um ambiente programável RNP + FABRIC sendo considerado nessa proposta de projeto, os pesquisadores da Universidade de Brasília coordenarão o esforço para atacar cada um dos desafios elencados no caso de uso apresentado. No primeiro ano de projeto, prevê-se o desenvolvimento e teste de protótipo do classificador de fluxos para redes programáveis utilizando o protocolo Openflow e a tecnologia P4 se necessário. Já, no segundo ano, prevê-se o desenvolvimento e teste de um protótipo da aplicação DPI utilizando P4. Posteriormente, no terceiro ano, prevê-se a pesquisa e desenvolvimento da ligação entre os aplicativos de classificação de fluxos e DPI em redes programáveis reais. No quarto ano, o desenvolvimento de mecanismos de otimização na tomada de decisão sobre o posicionamento das aplicações dentro de diferentes topologias de rede programáveis. Por fim, no quinto ano, prevê-se o teste e a mensuração do desempenho fim-a-fim do sistema otimizado de segurança distribuída criado.

3.2.3 Aplicações de Internet das Coisas sobre infraestrutura programável

Seguindo uma tendência tecnológica na qual dispositivos possam comunicar entre si, a utilização de uma infraestrutura programável surge como uma importante aliada em pesquisas sobre a Internet das Coisas (IoT). Pretende-se atender às necessidades de pesquisa em IoT utilizando diferentes conceitos de redes programáveis, tais como SDN e funções de redes virtualizadas (NFV). Além das especificidades dos protocolos de IoT, estes dispositivos apresentam desafios como o consumo de energia e a sua capacidade de processamento disponível (PATEL, K. K.; PATEL, S. M. et al., 2016). Outro grande desafio das aplicações de IoT está relacionada ao número de dispositivos conectados – de acordo com a Ericsson (ERICSSON, 2016), a expectativa é de que no ano de 2025 haverá aproximadamente 5 bilhões de dispositivos de IoT conectados à Internet.

O gerenciamento de uma solução de IoT ainda não apresenta consenso na indústria e academia, tanto no processo de implantação inicial (deployment) como no gerenciamento dinâmico dessas infraestruturas, já que as escolhas fim a fim em um caso de uso de IoT são determinantes para o desempenho obtido - e em muitos casos necessário - para diferentes tipos de aplicações de IoT, como cidades inteligentes, agricultura inteligente e assistência médica.

Em geral, a infraestrutura computacional distribuída necessária para executar aplicações de IoT é composta não apenas por dispositivos (sensores e atuadores), servidores de nuvem e estações móveis. Existem várias motivações para utilização de infraestrutura computacional mais para a borda da rede, ou seja, infraestrutura intermediária e mais próxima dos dispositivos que realiza comunicação, armazenamento e processamento de dados. Essa infraestrutura é comumente chamada de computação na borda ou computação em névoa, que pode abranger plataformas com vários níveis de poder computacional. O IoT Computing Continuum (IoTInuum) foi proposto para facilitar a compreensão e possibilitar a automatização dos processos de implantação (deployment) e gerenciamento dinâmico de componentes de software (ZYRIANOFF, I. et al., 2020). A existência de níveis diferentes de elementos da infraestrutura computacional distribuída varia de acordo com características, requisitos e restrições dos casos de uso. Por exemplo, em um cenário agrícola, algumas fazendas podem ter conectividade intermitente com a Internet, levando à necessidade de infraestrutura computacional na própria sede. Por outro lado, fazendas situadas em locais com boa conectividade podem desejar que não haja nenhum tipo de infraestrutura interna. Outras ainda podem desejar a existência de um mini datacenter com grande capacidade computacional na própria fazenda.

O IoTInuum (Figura 3.4) identifica alguns estágios definidos pela sua localização geográfica (mais perto ou mais longe dos dispositivos) e pela sua capacidade computacional (maior ou menor). O conceito de IoTInuum permite a formalização desta infraestrutura altamente distribuída e facilita a criação de diferentes planos de deploy para facilitar o mapeamento dos componentes da arquitetura, distribuídos em cinco estágios: S1-Thing, S2-Mist, S3-Fog, S4-Cloud e S5-Terminal.

- S1-Thing: representa o estágio implementado pelo dispositivo de hardware, como sensores e atuadores, que convertem sinais analógicos para digitais e desempenham transformações simples e específicas do dispositivo, como por exemplo o processo de calibração;
- S2-Mist: nós de Mist estão instalados em campo, como por exemplo os gateways de rádio no contexto das LPWAN ou outra tecnologia similar, que fornecem o suporte à comunicação de dados e agregação de dados. Os nós de Mist estão próximos dos dispositivos (Things) que os suporta e geralmente possuem recursos computacionais modestos, como por exemplo um Raspberry Pi;
- S3-Fog: os nós de névoa, ou fog, são em geral instalados em locais abrigados, com fornecimento estável de energia, representados por equipamentos como laptops, desktops ou pequenos servidores, que conferem confiabilidade, robustez, resiliência e baixa latência para aplicações sensíveis ao tempo de resposta;
- S4-Cloud: nuvens públicas ou privadas em datacenters com servidores físicos com muitos recursos para fornecer máquinas virtuais. Há um incremento considerável da capacidade de processamento entre o estágio S1 e S4;
- S5-Terminal: é o estágio onde o usuário final interage com a aplicação, em geral conectado ao estágio S4. Ressalta-se que em algumas configurações o estágio S5 pode também estar conectado ao estágio S3.

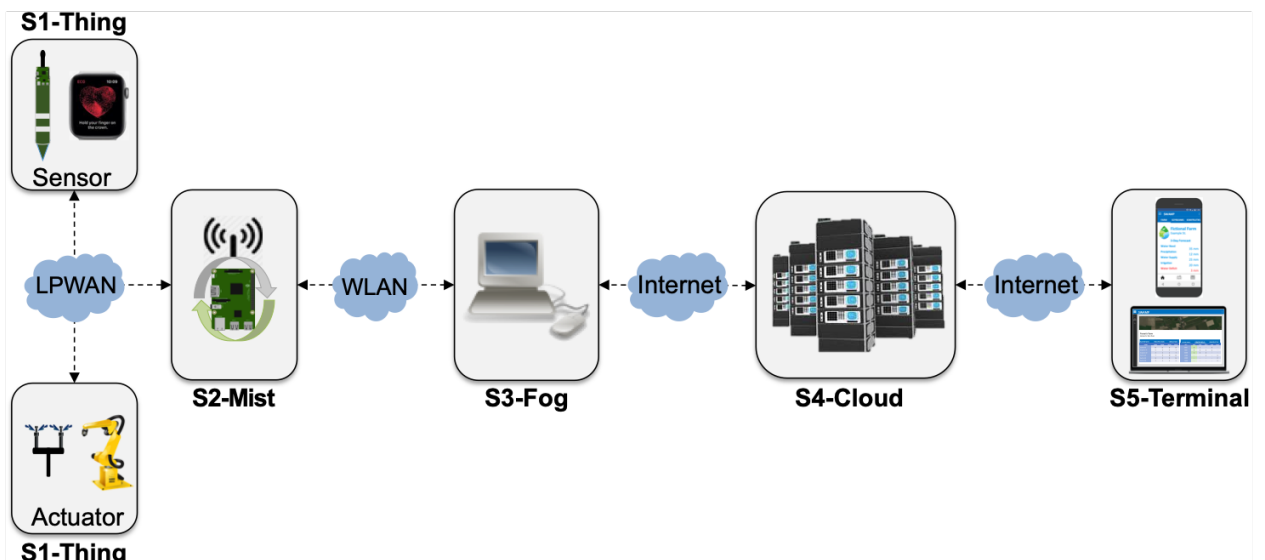


Figura 3.4: IoTinum (IoT Computing Continuum): implantação de aplicações IoT inteligentes em uma infraestrutura distribuída

A implantação (deployment) dos componentes de software numa infraestrutura computacional distribuída é influenciada por vários fatores. Além disso, o gerenciamento dinâmico desses componentes pode identificar a necessidade de alterar o local da sua execução, ou seja, o estágio

do IoTInuum. Sistemas automatizados tem a capacidade de decidir em tempo de implantação ou em tempo de execução o local mais apropriado onde um componente deve ser executado (por exemplo, no núcleo ou na borda da rede). Esses sistemas também podem realizar a implantação e reimplantação dos componentes. Este cenário é altamente aderente ao conceito das redes programáveis e virtualização de funções de rede, delegando ao software de orquestração a responsabilidade de otimizar o mapeamento entre os requisitos e o substrato disponível.

Uma iniciativa para formalizar este processo de orquestração é o conceito de Fog-Aware Computing (ZYRIANOFF, I. D. R., 2019), que propõe que o processo de implantação e mudança na infraestrutura ocorra dinamicamente, se adaptando às novas condições da infraestrutura, como desconexões, defeitos em nós de computação ou variação na carga de trabalho. O Fog-Aware Computing atua principalmente em três frentes: na implantação e/ou migração do sistema para uma nova infraestrutura; na adaptação do sistema em ambientes dinâmicos; e finalmente na seleção de diferentes implementações de software para atender aos requisitos do sistema como um todo.

A implementação do conceito de Fog-Aware Computing pode ser dividido em três aspectos:

1. Consciência (Awareness): este aspecto é a habilidade de estar consciente das mudanças na infraestrutura e na carga de trabalho - a capacidade de observar o estado do sistema em tempo de execução, como por exemplo a variação nos recursos dos nós de computação, desconexão de enlaces, incremento ou decremento na demanda (carga de trabalho), entre outros;
2. Inteligência: a partir de uma descrição da infraestrutura, a descrição da aplicação e um algoritmo de distribuição (placement), a inteligência aplica a estratégia do algoritmo para gerar um plano de deployment da aplicação na infraestrutura fornecida;
3. Execução: uma vez que o plano de deploy foi gerado pela inteligência, a execução é responsável por executar o plano, ou seja, instanciar os diferentes softwares (Linux Containers por exemplo), nos nós de computação escolhidos pela inteligência.

Um exemplo do funcionamento do Fog-Aware Computing é mostrado na Figura 3.5, na qual temos um paralelo entre a infraestrutura real e o modelo gerado pelo arcabouço. A infraestrutura é mapeada como um grafo, sendo que os pesos das arestas modelam a qualidade da conexão entre os nós. Utilizando algum algoritmo de distribuição determinado pelo o usuário, é realizada a implantação dinâmica das funções que compõem o sistema, sejam elas funções de rede - denotadas como círculos lilases - ou aplicações - pentágonos verdes.

Do ponto de vista de sistemas distribuídos, a reconfiguração pode ser tratada por sistemas de gerenciamento inteligentes, como o Fog-Aware Computing. Um problema que persiste, porém, é a reconfiguração da rede. Portanto, uma rede programável se torna de vital importância nesse cenário para que a reconfiguração do sistema distribuído seja realizada rapidamente e sem costuras (seamless).

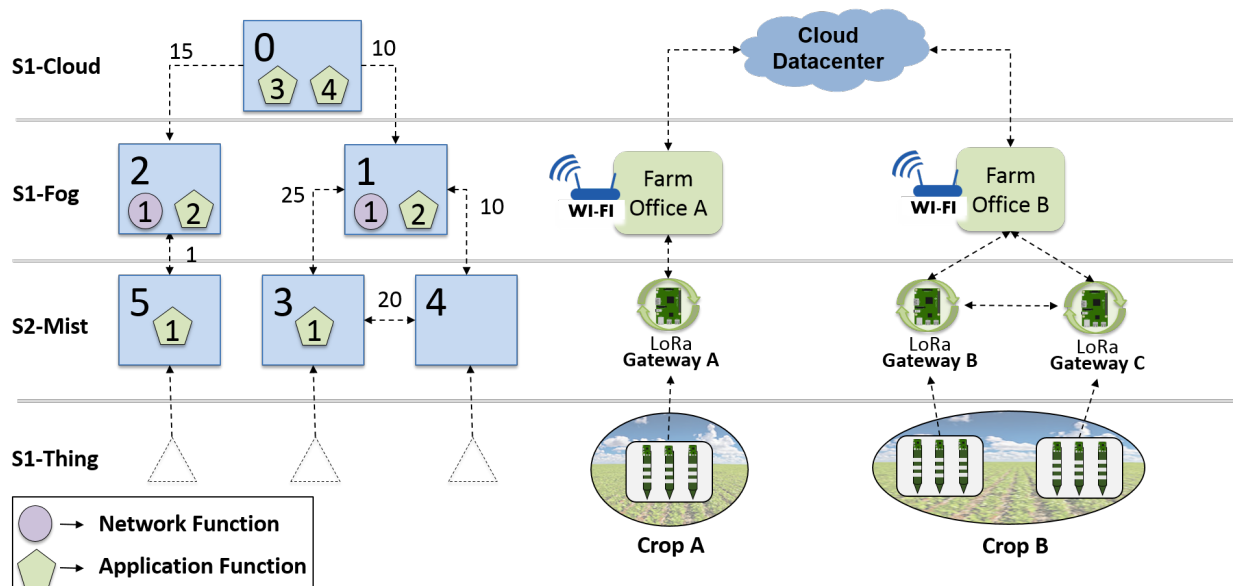


Figura 3.5: Fog-Aware Computing: Mapeamento da infraestrutura abstrata A no ambiente de IoT

3.3 Implantação de infraestrutura programável

(RNP + FABRIC + UFRGS)

Para conduzir experimentos controlados, considerando as técnicas de engenharia de software propostas no projeto PROFISSA, bem como desenvolver e analisar programas de rede é essencial contar com uma infraestrutura capaz de proporcionar a implantação do software desenvolvido, além de propiciar a elaboração de diferentes cenários de rede. Para a implantação da infraestrutura programável, o projeto conta com o apoio de dois parceiros fundamentais.

O primeiro parceiro é o projeto FABRIC¹, Estabelecido em outubro de 2019, o projeto FABRIC visa o provisionamento de uma infraestrutura exclusiva de pesquisa que permite pesquisas de ponta e exploratórias em redes, segurança cibernética, sistemas distribuídos de computação e armazenamento, aprendizado de máquina, entre outros tópicos (ilustrada na Figura 3.6). A infraestrutura programável almejada pelo FABRIC será acessível globalmente, composto por novos elementos de rede extensíveis, equipados com grandes quantidades de computação e armazenamento, interconectados por links ópticos dedicados de alta velocidade. Além disso, o FABRIC conectará várias bases de testes especializadas para pesquisa em nuvem (tais como os bases de dados de teste NSF Cloud CloudLab e Chameleon), para pesquisa em técnicas sem-fio como 5G, 5GB e 6G, bem como instalações de computação de alto desempenho e instrumentos científicos para criar um tecido rico para uma ampla variedade de atividades experimentais. Como principais vantagens, o FABRIC proporcionará:

- Desenvolvimento de novos aplicativos de Internet e ciência - Arquiteturas de rede *stateful* e aplicativos distribuídos para programação da rede;

¹<https://fabric-testbed.net/about/overview>

Anticipated FABRIC Topology



Figura 3.6: Topologia inicial almejada pelo testbed FABRIC

- Aplicações de segurança cibernética - Pesquisa realista em escala, facilitada pelo *peering* com redes de produção;
- Integração entre HPC, Wireless e IoT - Um ambiente diversificado que conectará os testbeds do programa PAWR (Platforms for Advanced Wireless Research)², nuvens NSF e centros HPC;
- Integração com Aprendizado de Máquina e Inteligência Artificial - Suporte para análise e controle de dados acelerados por GPU em rede;
- Auxiliar no treinamento da próxima geração de pesquisadores de ciência da computação.

O segundo parceiro do projeto é a Rede Nacional de Ensino e Pesquisa (RNP)³, uma rede avançada de alcance nacional para educação superior, pesquisa e inovação. que trabalha para promover o uso inovador de Tecnologias da Informação e Comunicação. Além de disponibilizar os mais diversos serviços de rede para instituições de ensino, a RNP também provê conectividade de alta velocidade e qualidade para as instituições parceiras através da Rede Ipê, ilustrada na Figura 3.7. A Rede Ipê é a rede acadêmica brasileira, presente em todo o território nacional oferecendo não apenas acesso à internet de qualidade, mas também suporte a transmissão de grandes volumes de dados para projetos científicos e desenvolvimento de novas tecnologias. A infraestrutura da rede Ipê engloba 27 Pontos de Presença (PoPs), um em cada unidade da federa-

²<https://advancedwireless.org/platforms/>

³<https://www.rnp.br/>

ção, além de ramificações para atender 1522 campi e unidades de instituições de ensino, pesquisa e saúde em todo o país, beneficiando mais de 3,5 milhões de usuários.

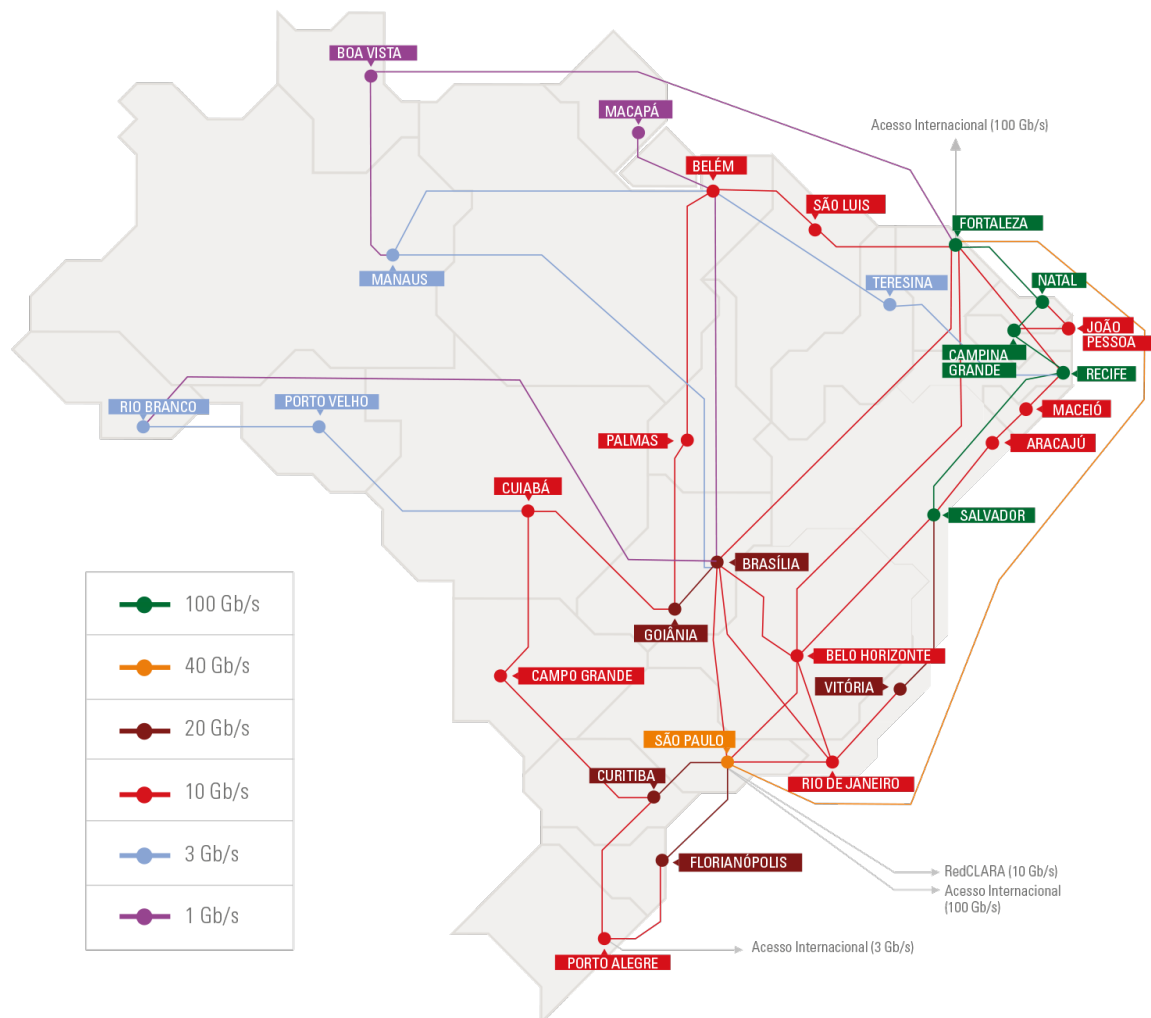


Figura 3.7: Rede Ipê da RNP

Dessa forma, um dos principais objetivos almejados por esse projeto é de interligar a infraestrutura programável para desenvolvimento de software de rede oferecida pelo FABRIC através da rede de ensino e pesquisa científica de alta velocidade fornecida pela RNP. Com isso, este projeto providenciará o primeiro testbed de infraestrutura programável de alta velocidade ligando nós FABRIC em diferentes países, *i.e.*, EUA e Brasil.

A força de trabalho para implantação da infraestrutura programável providenciada pelo FABRIC e RNP, bem como a interconexão entre os nós da infraestrutura provida pela RNP será liderada pela UFRGS. A UFRGS, através do Grupo de Redes de Computadores do Instituto de Informática tem uma vasta expertise na área de redes programáveis, contando com inúmeros projetos, artigos científicos, plataformas experimentais, dissertações de mestrado e teses de doutorado na área. Assim, contando com a liderança e experiência da UFRGS, a implantação da infraestrutura se dará de maneira gradual, por meio de X etapas:

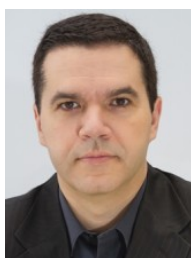
1. Definição do nó FABRIC a ser implantado;
2. Configuração do nó FABRIC isolado;
3. Testes e validação de infraestrutura do nó FABRIC;
4. Testes de conectividade interna do nó FABRIC ;
5. Definição junto à RNP do link de conectividade par ao nó FABRIC;
6. Configuração da conexão do nó FABRIC local com os demais nós da federação;
7. Testes de conectividade externa do nó FABRIC;
8. Testes de conectividade junto a federação FABRIC (multi-nó);
9. Disponibilização da infraestrutura para as demais instituições parceiras do projeto;
10. Implantação, teste e análise dos primeiros casos de uso elaborados pelo projeto.

A partir da elaboração da infraestrutura a ser utilizada, a equipe de execução do PROFISSA poderá desenvolver os estudos de caso sobre uma rede programável real e realizar testes com os insumos de software gerados no projeto. A seguir a equipe de execução do PROFISSA é descrita, bem como suas respectivas atividades a serem desenvolvidas no projeto.

4 Pesquisadores Principais e suas responsabilidades

Nesse capítulo, o pesquisador responsável e os pesquisadores principais são apresentados. Para cada um desses pesquisadores, suas atividades no projeto PROFISSA são descritas. Por fim, os expertises e atividades desenvolvidas por cada instituição parceira no projeto PROFISSA são brevemente descritas.

Lisandro Zambenedetti Granville - Pesquisador Responsável



O Pesquisador Responsável, Lisandro Zambenedetti Granville é Diretor do departamento Centro de Pesquisa e Desenvolvimento de Tecnologias Digitais de Informação e Comunicação (CTIC), da Rede Nacional de Ensino e Pesquisa (RNP). Além disso, Lisandro é professor Titular de Ciência da Computação no Instituto de Informática da Universidade Federal do Rio Grande do Sul (UFRGS), Brasil. Possui Ph.D. (2001) e M.Sc. (1998) em Ciência da Computação, ambos recebidos pela UFRGS. De setembro de 2007 a agosto de 2008, foi pesquisador visitante da Universidade de Twente, na Holanda, com o grupo Design and Analysis of Communication Systems. Também é membro do Grupo de Redes de Computadores da UFRGS, onde desenvolve projetos de pesquisa em gerenciamento de redes e serviços. Como professor titular, também está envolvido com atividades de supervisão e educação em cursos de graduação e pós-graduação em Ciência da Computação e Engenharia de Computação. Atualmente, atua como coordenador dos projetos Network Functions Virtualization ManageENT & ORchestration (NFV-Mentor - CNPq) e Green Cloud (FAPERGS), além de ser vice-coordenador do projeto Management Environment for Inter-domain Circuits in Advanced Networks (MEICAN - RNP).

Lisandro também ocupa os cargos de Presidente executivo da IEEE Global Communications Conference (GLOBECOM 2022), Diretor da Região da Sociedade de Comunicações da IEEE na América Latina - 2018-2019, Membro do conselho consultivo da Sociedade Brasileira de Computação (SBC), Editor-chefe do Journal of Internet Services and Applications - JISA (Springer), Editor-chefe associado do International Journal of Network Management - IJNM (Wiley), Membro do Conselho Editorial do Journal of Networks and Systems Management - JNSM (Springer) e IEEE Transactions on Network and Systems Management - TNSM (IEEE), Co-Presidente Geral da Conferência Internacional IEEE sobre Cloud Networking (CloudNet 2019), juntamente com Paulo Simões

Publicou mais de 140 artigos publicados em áreas relacionadas a redes de computadores, in-

cluindo publicações em prestigiados veículos, tais como IEEE Communication Mazine, IEEE Surveys & Tutorials, ACM SIGCOMM, IEEE IM, IEE NOMS e IFIP Networking. Orientou 16 dissertações de mestrado e 7 teses de doutorado.

No projeto, Lisandro atuará na coordenação dos esforços de pesquisa de maneira geral, nor-teando as atividades a serem desenvolvidas em equipe. Além disso, Lisandro irá contribuir ativa-mente para as definições relacionadas às áreas de gerencia de redes, SDN, NFV e programabilidade de rede envolvidas no desenvolvimento de programas de rede no projeto. Durante a elaboração do projeto, Lisandro trabalhará auxiliando no estabelecimento da equipe do projeto e será responsá-vel direto pela coordenação da equipe, seja através de reuniões de planejamento ou comunicação direta entre os pesquisadores. Lisandro também contribuirá para os refinamentos, melhorias e otimizações das ferramentas desenvolvidas, bem como nas avaliações científicas e atividades de divulgação do projeto. Por fim, como representante da RNP, Lisandro estará envolvido nas ati-vidades relacionadas ao estabelecimento da infraestrutura e conectividade do projeto, bem como seus desafios científicos e tecnológicos, todos diretamente ligados às atividades que envolvem a RNP.

Alberto E. Schaeffer-Filho - Pesquisador Principal - UFRGS



Prof. Dr. Alberto Egon Schaeffer Filho - Professor adjunto do Departamento de Informática Aplicada da Universidade Federal do Rio Grande do Sul, desde 2012. Possui B.Sc. (2002) em Ciência da Computação pela Universidade Federal do Rio Grande do Sul (UFRGS), M.Sc. (2005) em Ciência da Computação também pela UFRGS. Em 2009, obteve o título de Doutor em Computação pelo Imperial College London, no Reino Unido. Entre 2009 e 2012, fez pós-doutorado na Lancaster University, no Reino Unido. Atua nas áreas de programabilidade de redes e resiliência de redes. Possui ampla participação em projetos correlatos: coor-

denador do projeto “Resiliência Multinível em Infraestruturas de Rede baseadas em SDN e NFV [2017-2020]”, com financiamento da Chamada Universal - MCTI/CNPq No 01/2016, coordenador adjunto do projeto “GT-FENDE - Ecosistema Federado para Oferta, Distribuição e Execução de Funções Virtualizadas de Rede [2017-2019]”, grupo de trabalho apoiado pela Rede Nacional de Pesquisa (RNP), e membro do projeto “P4Sec - Securing Networks in the Programmable Data Plane Era”, financiado pela NSF/RNP. Possui 15 artigos em periódicos (incluindo IEEE Surveys & Tutorials, IEEE Communications Magazine e Proceedings of the IEEE), e mais de 60 publica-ções em conferências qualificadas (incluindo ACM CoNeXT, IEEE INFOCOM, IEE Globecom e IEEE ICC). Atuou no comitê de organização de importantes eventos científicos, incluindo coor-denador geral do SBRC’19 juntamente com Weverton Cordeiro, co-chair do IEEE ICC’18 CQRM symposium, e demo co-chair do IFIP/IEEE IM’17. Atua regularmente no comitê de programa de importantes conferências na área, incluindo NetSoft, IM, NOMS, SBRC, SBSeg, CNSM, etc, e

como membro do corpo editorial do International Journal of Network Management (Wiley) e do Journal of Network and Systems Management (Springer). Atualmente é bolsista PQ-2 do CNPq.

No projeto PROFISSA, o Prof. Alberto se valerá de experiência prévia em redes SDN/OpenFlow e em linguagem P4 para principalmente empregar esforços na análise de código fonte, identificação de más práticas e predição de bugs em aplicações de rede. Algumas das linhas de pesquisa a serem abordadas pelo projeto complementam trabalhos passados realizados pelo professor na área de identificação de bugs em código P4 (FREIRE et al., 2018a)(FREIRE et al., 2018b) e também na área de aplicação de design patterns em aplicações de rede (SCHAEFFER-FILHO et al., 2014). Dessa forma, junto com a equipe na UFRGS, o Prof. Alberto irá conduzir a investigação de como práticas de engenharia de software como reuso de artefatos, design patterns, princípios de modularização e separação de interesses, identificação de code smells e predição de bugs podem ser aplicadas no contexto de redes programáveis.

Carlos Alberto Kamienski - Pesquisador Principal - UFABC



Carlos Alberto Kamienski é Professor Titular em Ciência da Computação na Universidade Federal do ABC (UFABC) e doutor em Ciência da Computação pela Universidade Federal de Pernambuco (UFPE) em 2003. Tem experiência com as quatro principais atividades acadêmicas: ensino, pesquisa, extensão e gestão. Tem 24 anos de experiência em ensino superior em vários assuntos, principalmente em redes de computadores e sistemas distribuídos. Orientou 26 alunos de mestrado e dois alunos de mestrado que completaram seus estudos. Atualmente, orienta dois alunos de mestrado e sete de doutorado. Os seus interesses atuais de pesquisa incluem Internet das Coisas (IoT), agricultura inteligente, cidades inteligentes, computação em névoa, softwarização de redes (SDN, NFV) e Internet do Futuro. Publicou mais de 100 artigos científicos em importantes periódicos e congressos. Atualmente é bolsista de produtividade em pesquisa do CNPq nível 2. É o coordenador brasileiro do projeto SWAMP (swamp-project.org), uma colaboração entre Brasil e Europa que desenvolve métodos e abordagens baseados em IoT para a gestão inteligente de água na irrigação de precisão. Liderou a equipe da UFABC no projeto IMPReSS, uma colaboração entre Brasil e Europa para a gestão inteligente de energia baseada em IoT em prédios públicos. Desde 2013 é coordenador do Núcleo Estratégico NUVEM (nuvem.ufabc.edu.br), um grupo interdisciplinar de 20 professores e seus alunos apoiado pela reitoria da UFABC para atuar em ensino, pesquisa e extensão em cinco grandes áreas: sociedades inteligentes, sensações virtuais, mobilidade conectada, computação extrema e universos integrados. Coordenou e participou do comitê de programa de vários eventos científicos nacionais e internacionais. Atuou por mais de 10 anos em cargos de gestão acadêmica e administrativa na UFABC, como coordenador de curso de pós-graduação, Pró-reitor de Pós-Graduação e Assessor de Relações Internacionais. Também teve experiência como desenvolvedor e administrador de sistemas em diversas empresas

e inclusive foi sócio de uma empresa nos anos 1990.

No projeto PROFISSA, o Prof. Carlos atuará principalmente na liderança do caso de uso de IoT em redes programáveis, dada a sua experiência pregressa nessas duas áreas de pesquisa. Existem vários desafios científicos e tecnológicos a serem vencidos, descritos na seção 3.2.3, como: a) Desenvolvimento de um modelo conceitual para o mapeamento de funções e alto nível de reimplantação de componentes de software numa infraestrutura distribuída de IoT em código executável para a rede com plano de dados programável; b) Adequação desse modelo conceitual aos conceitos de engenharia de software para desenvolvimento de código de rede programável eficiente e seguro; c) Automatização (ou semi-automatização) do processo de geração de código de rede programável com base na necessidade de reconfiguração da rede de IoT subjacente como resultado da reimplantação de componentes de software; d) Instalação, testes e avaliação de desempenho da solução desenvolvida no projeto; e) Avaliação qualitativa das soluções de engenharia de software com base nos resultados práticos do caso de uso de IoT em redes programáveis.

Genáina Nunes Rodrigues - Pesquisadora Principal - UNB



Genáina N. Rodrigues é professora associada do Departamento de Ciência da Computação da Universidade de Brasília. Ela recebeu seu Ph.D. em Ciência da Computação pela University College London (2008) no grupo de Engenharia de Software. Anteriormente, obteve seu diploma de bacharel em Ciência da Computação pela Universidade de Brasília em 1999 e seu mestrado em Ciência da Computação pela Universidade Federal de Pernambuco (2002) no grupo Sistemas Distribuídos. Seus interesses de pesquisa englobam análise e modelagem de dependabilidade de sistemas de software, sistemas auto-adaptativos, engenharia de requisitos orientada a objetivos, desenvolvimento dirigido a modelos para requisitos de qualidade, verificação de modelos probabilísticos, assim como plataformas distribuídas. Em 2013, co-organizou o Congresso de Engenharia de Software do Brasil (CBSOFT) da Sociedade Brasileira de Computação (SBC). Atualmente, ela ocupa o cargo de coordenadora do Programa de Pós-Graduação em Informática da Universidade de Brasília. Genáina tem coordenado projetos financiados pelo CNPq, pela CAPES e pela FAPDF. Além disso, Genáina é membro de três outros projetos, incluindo dois internacionais. Dentre esses projetos internacionais, um deles faz parte do projeto Assuring Autonomy International Program (AAIP) entre a University of York e Lloyds Foundation. O outro projeto internacional é financiado pela Newton Fund e FAPDF. Entre as premiações, Genáina foi agraciada com as seguintes: Bolsa CAPES-Humboldt (2020-2022) como *experienced researcher* para pesquisadores considerados de excelência internacional, Bolsa de Produtividade do CNPq (PQ) (2019-2022), Professora visitante na Universidade UT2J-França (2018), Professora Homenageada no curso de Ciência da Computação da UnB (2014 – 2017), Bolsa de doutorado pleno no exterior pela CAPES (2002-2006). Genáina recebeu convites internacionais para pales-

trar nas seguintes Universidades/Intituições: INRIT/Toulouse (2018), University of Bournemouth (2013 and 2018), Chalmers University of Technology (2015), University of Namur (2014), University of Stuttgart (2013), Oxford University (2012). No trabalho acadêmico, Genáina tem atuado na formação de novos pesquisadores por meio dos alunos orientados. Dentre as orientações de alunos, ela orientou 14 alunos de Iniciação Científica e mais de 30 alunos de graduação, 14 dissertações de mestrado e 2 teses de doutorado. Atualmente, ela orienta 2 alunos de doutorado e 3 alunos de mestrado e 7 alunos de Iniciação Científica nos temas na área do tema de pesquisa proposto. Sua inserção na comunidade internacional pode ser evidenciada também pelo convite a participar como revisora de prestigiados periódicos como: IEEE Transactions on Software Engineering, ACM Transactions on Autonomous and Adaptive Systems, Journal of Systems and Software, Software and Systems Modeling, Information and Software Technology, entre outros. Além disso, Genáina tem participado como membro do comitê de programa de conferências internacionais como: International Conference on Conceptual Modeling (ER), ICSE New Ideas and Emerging Results (NIER), International Symposium on Adaptive Managing Systems (SEAMS), ACM Symposium on the Foundations of Software Engineering (FSE), entre outras. Genáina já publicou em 14 periódicos internacionais reconhecidos e mais de 28 artigos em conferências nacionais e internacionais.

No projeto PROFISSA, a Profa. Genáina atuará principalmente na liderança do caso de uso de segurança distribuída em redes programáveis, dada a sua experiência pregressa na parte de otimização e planejamento no contexto de sistemas adaptativos e em verificação de atributos de dependabilidade, que tem uma relação bastante estreita com segurança. Existem vários desafios científicos e tecnológicos a serem superados, descritos na Seção 3.2.2. Dentre eles, o desenvolvimento de um arcabouço de código reutilizável com foco na otimização e planejamento dos requisitos de segurança. Além disso, a Profa. Genáina, juntamente com sua equipe, irá trabalhar ativamente para contribuir na transposição de técnicas de engenharia de software envolvendo, por exemplo, técnicas de MDE e GORE para o especificação de APIs que integram o gerenciamento da SDN com o desenvolvimento dos aplicativos.

José Ferreira de Rezende - Pesquisador Principal - UFRJ



José F. de Rezende possui graduação em Engenharia Eletrônica (1988) e mestrado em Engenharia Elétrica (1992), ambos pela Universidade Federal do Rio de Janeiro, e doutorado em Ciência da Computação (1997) pela Université Pierre et Marie Curie, França. Ele é Professor do Programa de Engenharia de Sistemas e Computação da COPPE/UFRJ e, desde 2016, ele é assessor especial em P&D da RNP. Tem experiência na área de Teleinformática, atuando principalmente nos seguintes temas de pesquisa: QoS na Internet, Internet do Futuro, redes de alta velocidade, redes móveis, redes de sensores sem fio, redes definidas por soft-

ware, rádios cognitivos, experimentação remota e avaliação de desempenho. Atualmente, ele é pesquisador nível 1D do CNPq.

No projeto PROFISSA, o Prof. Rezende atuará principalmente na liderança do caso de uso de 5G em redes programáveis, dada a sua experiência nessas áreas. Os desafios científicos a serem vencidos nessa área consistem na concepção de mecanismos e técnicas capazes de determinar, de forma inteligente e na escala de tempo apropriada, a melhor maneira de gerenciar os recursos das redes F-RAN. Esse gerenciamento consiste em determinar, a cada instante, a distribuição do tráfego, o posicionamento dos caches de conteúdo na borda da rede e a carga em cada uma das antenas. Tudo isso de forma a atender aos requisitos de desempenho, escalabilidade e confiabilidade impostos pelas aplicações dos usuários, minimizar o consumo de energia e otimizar o uso de recursos. Em seguida, pelos métodos e técnicas de engenharia de software definidas nesse projeto, um outro desafio se impõe que é facilitar o desenvolvimento de programas/aplicações de rede que influenciem na tomada de decisões nesse cenário, permitindo a coexistência de múltiplas aplicações em uma mesma F-RAN. Assim, a equipe de pesquisadores do professor Rezende trabalhará em cooperação com os pesquisadores da UnB e da UFRGS para vencer tais desafios.

4.1 Equipe

Além dos membros da Instituição sede Rede Nacional de Ensino e Pesquisa (RNP), o projeto congrega equipes de 4 universidades, (i) Universidade Federal do Rio de Janeiro (UFRJ), Universidade de Brasília (UnB), Universidade Federal do ABC (UFABC) e Universidade Federal do Rio Grande do Sul (UFRGS). A RNP ficará responsável por prover conectividade à infraestrutura do projeto, além de outras ciberinfraestruturas. A infraestrutura do projeto contará com o apoio da FIU pelo projeto FABRIC, o qual provê os equipamentos necessários para adesão da infraestrutura em seu testbed. Dessa forma, a integração entre FABRIC e RNP se torna fundamental para o projeto, dada a infraestrutura providenciada pelo FABRIC e a conectividade garantida pela RNP.

Tabela 4.1: “Expertise” das instituições parceiras nos temas do projeto.

Universidade	Eng. de Software	Redes Programáveis/Testbed	Infraestrutura e Conectividade	Redes sem-fio	Cibersegurança	IoT
RNP		X	X			
UFRGS	X	X			X	
UFRJ				X		
UFABC						X
UnB	X				X	
FIU		X	X			

Pesquisadores da Florida International University/FIU pertencentes ao projeto FABRIC, atuarão nos temas de pesquisa do projeto e possuem histórico de colaboração, participação em projetos e publicação conjunta com os membros da equipe. Todos os pesquisadores da equipe do projeto PROFISSA atuam nas áreas dos temas dessa proposta e possuem histórico de publicação e

de orientação de pós-graduados, o que confere a equipe a capacidade de investigar os problemas aqui propostos. A Tabela 4.1 relaciona as áreas de especialização dos pesquisadores das diferentes instituições e os temas do projeto de pesquisa.

Abaixo, segue uma relação de todos os componentes da equipe do projeto PROFISSA.

Pesquisador Responsável

- **Rede Nacional de Ensino e Pesquisa:** Lisandro Zambenedetti Granville, Diretor do CTIC, RNP e Professor Titular da Universidade Federal do Rio Grande do Sul, PQ-1C

Pesquisadores Principais

- **Universidade Federal do Rio Grande do Sul:** Alberto Egon Schaeffer Filho, Professor Associado, UFRGS PQ-2
- **Universidade Federal do ABC de São Paulo:** Carlos Alberto Kamienski, Professor Titular, UFABC PQ-2
- **Universidade de Brasília:** Genáina Nunes Rodrigues, Professora Associada, UnB; PQ-2
- **Universidade Federal do Rio de Janeiro:** José Ferreira de Rezende, Professor Associado, UFRJ, PQ-1D

Pesquisadores Associados

- **Rede Nacional de Ensino e Pesquisa:** Carolina Howard Felicíssimo, Coordenadora de P&D, RNP; Lucas Bondan, Coordenador de P&D, RNP.
- **Universidade Federal do Rio Grande do Sul:** Ingrid Nunes, Professora Adjunto, UFRGS.
- **Universidade de Brasília:** Marcelo Antonio Marotta, Professor Adjunto, UnB; João Gondin - Professor Associado.

Apoio técnico

- **Rede Nacional de Ensino e Pesquisa:** Iara Machado, Diretora do Departamento de P&D, RNP; Leandro Neumann Ciuffo, Vice-Diretor do Departamento de P&D, RNP; Alex Soares de Moura, Gerente de P&D, RNP; Marcos Felipe Schwarz, Coordenador de P&D, RNP.
- **Florida International University/FIU:** Julio Ibarra, Vice presidente de Tecnologia Aumentada, FIU.

5 Cronograma

O projeto PROFISSA será desenvolvido com data inicial em janeiro de 2021 e com previsão de duração de cinco anos, com data final em janeiro de 2026. Durante a execução do projeto as seguintes atividades e marcos deverão ser executadas para sua completude.

1. **Reuniões de planejamento inicial:** Desenvolvimento de reuniões de planejamento para organização da equipe e início das atividades investigativas e de desenvolvimento do projeto.
2. **Pesquisa bibliográfica:** Atividade investigativa onde os principais artigos científicos, padrões, patentes e documentos auxiliares serão levantados para fundamentar o projeto de pesquisa PROFISSA. Além disso, estudos e planejamento para a governança e gestão dos dados de pesquisa a serem utilizados e alcançados durante o projeto serão conduzidos nessa atividade.
3. **Aplicação de técnicas para desenvolvimento de protótipos:** A partir da fundamentação teórica, serão elencadas as principais técnicas de engenharia de software a serem consideradas no desenvolvimento do projeto PROFISSA. As técnicas serão subdivididas entre cada um dos casos de uso abordados no projeto.
4. **Implementação inicial de ferramentas e protótipos para os casos de uso:** Ao se aplicar as técnicas de desenvolvimento, serão implementadas, testadas e ajustadas as ferramentas e os protótipos de programas de redes. Para cada programa de rede, códigos modularizados serão publicados em um arcabouço de software para redes programáveis para incentivar seu reuso.
5. **Avaliação com experimentos e simulações:** Cada um dos insumos da implementação, como programas, funções codificadas e módulos de software, serão executados sobre simulações para que possam mais tarde ser experimentados em redes programáveis reais pertencentes a infraestrutura da RNP+FABRIC.
6. **Refinamentos, melhorias e otimização das ferramentas e protótipos:** Ajustes e redesigno das ferramentas e softwares desenvolvidos serão realizados para otimizar seu desempenho, legibilidade e generalização para compor o arcabouço de software a ser criado no projeto PROFISSA.
7. **Avaliação científica quantitativa e qualitativa:** A partir do refinamento das ferramentas e protótipos, experimentos controlados serão conduzidos para quantificar métricas de desempenho de qualidade de software gerados no projeto PROFISSA. Além disso, prevê-se

a avaliação em cenário controlado com voluntários da área de desenvolvimento de software para redes programáveis para quantificar métricas qualitativas em relação aos programas desenvolvidos no projeto PROFISSA.

8. **Divulgação (software livre, cursos, Webinars, material educacional aberto):** Todo o conteúdo gerado de cunho livre será divulgado pela RNP em seu acervo público de códigos. Além disso, Webinars, cursos e materiais educacionais abertos serão divulgados em parceria a Escola Superior de Redes (ESR).
9. **Transferência de tecnologia:** Ao haver a geração de software de interesse da comunidade pertencente a RNP, os mesmo poderão ser publicados em iniciativas de divulgação da própria RNP, como o projeto NasNuvens.
10. **Oficinas de projeto online:** Os dados gerados, as metodologias e lições aprendidas, bem como as atividades e técnicas necessárias para se replicar as pesquisas realizadas no PROFISSA serão repassadas e divulgadas a partir da execução de oficinas (workshops) online. Além disso, os workshops servirão também para incentivar a participação da comunidade científica no projeto PROFISSA. As oficinas permitirão a coleta de novas opiniões, abrindo caminhos para novas pesquisa, além de ter a validação da pesquisa já realizada no projeto PROFISSA.

O cronograma a seguir auxilia a visualizar os períodos destinados a execução das atividades e marcos a serem concluídos durante a execução do projeto PROFISSA.

Tabela 5.1: Cronograma de execução

Ação / Período	2021-2022	2022-2023	2023-2024	2024-2025	2025-2026
1 Reuniões de planejamento inicial	■				
2 Pesquisa bibliográfica	■	■			
3 Aplicação de técnicas para desenvolvimento de protótipos		■	■		
4 Implementação inicial de ferramentas e protótipos		■	■	■	
5 Avaliação com experimentos e simulações		■	■	■	
6 Refinamentos, melhorias e otimização das ferramentas e protótipos			■	■	■
7 Avaliação científica quantitativa e qualitativa			■	■	■
8 Divulgação (software livre, cursos, material educacional aberto)	■	■	■	■	■
9 Transferência de tecnologia			■	■	
10 Oficinas de projeto online		■		■	

6 Disseminação e avaliação

Resultados derivados do projeto serão submetidos na forma de artigos científicos a periódicos e conferências de relevância da área de redes de computadores, tais como SBRC, ACM SIGCOMM, IEEE INFOCOM, IEEE GLOBECOM e IEEE CloudNET, e de engenharia de software, tais como ICSE, ESEC/FSE, SBES, IEEE Transactions on Software Engineering, e ACM Transactions on Software Engineering and Methodology. A avaliação do impacto dos resultados será realizada por meio da avaliação de pares desses veículos. Espera-se também que o trabalho dos bolsistas alocados no projeto derivem teses, dissertações e trabalhos de conclusão de curso.

Além de publicações, os resultados derivados do projeto também serão disseminados através de seminários e workshops, como por exemplo o WPEIF (Workshop de Pesquisa Experimental em Internet do Futuro) e WRNP (Workshop da RNP), ambos inseridos no SBRC (Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos), a reunião internacional GEFI (Global Experimentation for Future Internet) e workshops da comunidade do projeto FABRIC (FABRIC Community Workshops).

Adicionalmente, desde 2016 a RNP organiza um evento satélite no congresso anual da SBC (CSBC), voltado para a realização de tutoriais de uso de ambientes de experimentação em redes de computadores. No caso desse evento satélite continuar sendo aprovado pela organização do CSBC, os resultados do projeto também poderão ser disseminados nesse espaço.

Reuniões periódicas, sejam presenciais ou virtuais, serão realizadas para discussão de questões científicas, disseminação de resultados e avaliação interna. Sugere-se pelo menos a realização de uma reunião presencial ao ano, preferencialmente co-locada no evento SBRC. As reuniões virtuais poderão ocorrer sempre que necessário usando o serviço de conferência web fornecido pela RNP em sala virtual personalizada para o projeto.

Por fim, será criado um site web e páginas em mídias sociais para a divulgação do projeto e seus resultados. Parte do conteúdo do site web será destinado a monitorar:

1. Teses, dissertações e artigos científicos publicados;
2. Apresentações realizadas em conferências, workshops e seminários abertos ao público;

7 Outros apoios

O projeto proposto neste documento contará com o apoio direto da Rede Nacional de Ensino e Pesquisa (RNP), que além de oferecer acesso à equipamentos programáveis distribuídos em seus pontos de presença (PoPs) e datacenters, também mobilizará esforços junto à coordenação do projeto FABRIC para habilitar um nó do testbed FABRIC no Brasil, de forma a facilitar o acesso de pesquisadores brasileiros às plataformas de experimentação disponíveis nos EUA.

A Florida International University (FIU) representada pelo Dr. Julio Ibarra é responsável pela Rede Acadêmica AmLight, que irá colaborar com as configurações de rede necessárias para conectar a infraestrutura de experimentação disponibilizada pela RNP à infraestrutura do testbed FABRIC. A FIU também hospeda um nó do testbed FABRIC. Através da AmLight os pesquisadores brasileiros poderão ter acesso ao FABRIC node da FIU enquanto não é concluída a instalação de um nó FABRIC no Brasil.

Ambos, FIU e a coordenação do projeto FABRIC, apoiarão na federação de recursos de sua infraestrutura ao nó da RNP para a oferta de um ambiente de experimentação ampliado para os pesquisadores e alunos do projeto PROFISSA, bem como para uso de demais pesquisadores brasileiros.

Referências bibliográficas

- 3RD GENERATION PARTNERSHIP PROJECT (3GPP). Specification - Release 16. **Technical Report - 21.915**, 3GPP, p. 1–95, 2015. Disponível em: <<https://www.3gpp.org/release-15%7D>>.
- ALSAEEDI, M.; MOHAMAD, M. M.; AL-ROUBAIEY, A. A. Toward Adaptive and Scalable OpenFlow-SDN Flow Control: A Survey. **IEEE Access**, v. 7, p. 107346–107379, 2019.
- ASTRÖM, K. J.; MURRAY, R. M. **Feedback systems: an introduction for scientists and engineers**. [S.l.]: Princeton university press, 2010.
- BOSSHART, P. et al. P4: Programming Protocol-Independent Packet Processors. **SIGCOMM Comput. Commun. Rev.**, Association for Computing Machinery, New York, NY, USA, v. 44, n. 3, p. 87–95, jul. 2014. ISSN 0146-4833. DOI: 10.1145/2656877.2656890. Disponível em: <<https://doi.org/10.1145/2656877.2656890>>.
- CHIEN, W.; LAI, C.; CHAO, H. Dynamic Resource Prediction and Allocation in C-RAN With Edge Artificial Intelligence. **IEEE Transactions on Industrial Informatics**, v. 15, n. 7, p. 4306–4314, abr. 2019.
- COX, J. H. et al. Advancing Software-Defined Networks: A Survey. **IEEE Access**, v. 5, p. 25487–25526, 2017.
- D’AMBROS, M.; LANZA, M.; ROBBES, R. Evaluating Defect Prediction Approaches: A Benchmark and an Extensive Comparison. **Empirical Softw. Engg.**, Kluwer Academic Publishers, USA, v. 17, 4–5, p. 531–577, ago. 2012. ISSN 1382-3256. DOI: 10.1007/s10664-011-9173-9. Disponível em: <<https://doi.org/10.1007/s10664-011-9173-9>>.
- DING, D.; SAVI, M.; ANTICHI, G.; SIRACUSA, D. An Incrementally-Deployable P4-Enabled Architecture for Network-Wide Heavy-Hitter Detection. **IEEE Transactions on Network and Service Management**, v. 17, n. 1, p. 75–88, 2020.
- ERICSSON, A. Cellular networks for massive IoT—Enabling low power wide area applications. **no. January**, p. 1–13, 2016.
- FEAMSTER, N.; REXFORD, J.; ZEGURA, E. The Road to SDN: An Intellectual History of Programmable Networks. **SIGCOMM Comput. Commun. Rev.**, Association for Computing Machinery, New York, NY, USA, v. 44, n. 2, p. 87–98, abr. 2014. ISSN 0146-4833. DOI: 10.1145/2602204.2602219. Disponível em: <<https://doi.org/10.1145/2602204.2602219>>.

FELIX SOLANO, G. et al. Taming Uncertainty in the Assurance Process of Self-Adaptive Systems: a Goal-Oriented Approach. In: 2019 IEEE/ACM 14th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS). [S.l.]: IEEE, mai. 2019. 2019-May, p. 89–99.

FOWLER, M. **Domain Specific Languages**. 1st. [S.l.]: Addison-Wesley Professional, 2010. ISBN 0321712943.

FREIRE, L. et al. Uncovering Bugs in P4 Programs with Assertion-Based Verification. In: PROCEEDINGS of the Symposium on SDN Research. Los Angeles, CA, USA: Association for Computing Machinery, 2018. (SOSR '18). ISBN 9781450356640. DOI: 10 . 1145 / 3185467 . 3185499. Disponível em: <<https://doi.org/10.1145/3185467.3185499>>.

FREIRE, L. et al. Uncovering Bugs in P4 Programs with Assertion-Based Verification. In: PROCEEDINGS of the Symposium on SDN Research. Los Angeles, CA, USA: Association for Computing Machinery, 2018. (SOSR '18). ISBN 9781450356640. DOI: 10 . 1145 / 3185467 . 3185499. Disponível em: <<https://doi.org/10.1145/3185467.3185499>>.

GURBANI, V. K.; GARVERT, A.; HERBSLEB, J. D. A Case Study of a Corporate Open Source Development Model. In: PROCEEDINGS of the 28th International Conference on Software Engineering. Shanghai, China: Association for Computing Machinery, 2006. (ICSE '06), p. 472–481. ISBN 1595933751. DOI: 10 . 1145 / 1134285 . 1134352. Disponível em: <<https://doi.org/10.1145/1134285.1134352>>.

HELLERSTEIN, J. L.; DIAO, Y.; PAREKH, S.; TILBURY, D. M. **Feedback control of computing systems**. [S.l.]: John Wiley & Sons, 2004.

JIANG, Y. et al. User Preference Learning-Based Edge Caching for Fog Radio Access Network. **IEEE Transactions on Communications**, v. 67, n. 2, p. 1268–1283, fev. 2019.

JIANG, Y. et al. User Preference Learning Based Edge Caching for Fog Radio Access Network. **IEEE Transactions on Communications**, v. 67, p. 1268–1283, nov. 2018.

KELLERER, W. et al. Adaptable and Data-Driven Softwarized Networks: Review, Opportunities, and Challenges. **Proceedings of the IEEE**, v. 107, n. 4, p. 711–731, abr. 2019.

LANZA, M.; MARINESCU, R. **Object-Oriented Metrics in Practice: Using Software Metrics to Characterize, Evaluate, and Improve the Design of Object-Oriented Systems**. 1st. [S.l.]: Springer Publishing Company, Incorporated, 2010. ISBN 3642063748.

LIU, J. et al. P4v: Practical Verification for Programmable Data Planes. In: PROCEEDINGS of the 2018 Conference of the ACM Special Interest Group on Data Communication. Budapest, Hungary: Association for Computing Machinery, 2018. (SIGCOMM '18), p. 490–503. ISBN 9781450355674. DOI: 10 . 1145 / 3230543 . 3230582. Disponível em: <<https://doi.org/10.1145/3230543.3230582>>.

LOPES, F. A.; SANTOS, M.; FIDALGO, R.; FERNANDES, S. A Software Engineering Perspective on SDN Programmability. **IEEE Communications Surveys Tutorials**, v. 18, n. 2, p. 1255–1272, Secondquarter 2016. ISSN 2373-745X.

MAROTTA, M. A. et al. Design considerations for software-defined wireless networking in heterogeneous cloud radio access networks. **Journal of Internet Services and Applications**, v. 8, n. 1, p. 18, dez. 2017.

MENS, T.; DEMEYER, S. **Software Evolution**. 1. ed. [S.l.]: Springer Publishing Company, Incorporated, 2008. ISBN 3540764399.

PAIVA, T.; DAMASCENO, A.; FIGUEIREDO, E.; SANT'ANNA, C. On the evaluation of code smells and detection tools. **Journal of Software Engineering Research and Development**, v. 5, 7:1–7:28, 2017. ISSN 2195-1721. Disponível em: <<https://sol.sbc.org.br/journals/index.php/jserd/article/view/433>>.

PARNAS, D. L. On the Criteria to Be Used in Decomposing Systems into Modules. **Commun. ACM**, Association for Computing Machinery, New York, NY, USA, v. 15, n. 12, p. 1053–1058, dez. 1972. ISSN 0001-0782. DOI: 10.1145/361598.361623. Disponível em: <<https://doi.org/10.1145/361598.361623>>.

PATEL, K. K.; PATEL, S. M. et al. Internet of Things - IoT: definition, characteristics, architecture, enabling technologies, application & future challenges. **International journal of engineering science and computing**, v. 6, n. 5, 2016.

PESSOA, L. et al. Building reliable and maintainable Dynamic Software Product Lines: An investigation in the Body Sensor Network domain. **Information and Software Technology**, Elsevier B.V., v. 86, p. 54–70, jun. 2017.

REFACTORING: Improving the Design of Existing Code. USA: Addison-Wesley Longman Publishing Co., Inc., 1999. ISBN 0201485672.

RODRIGUES, G. S. et al. GoalD: A Goal-Driven deployment framework for dynamic and heterogeneous computing environments. **Information and Software Technology**, v. 111, p. 159–176, jul. 2019.

SACCO, A.; ESPOSITO, F.; MARCHETTO, G. RoPE: An Architecture for Adaptive Data-Driven Routing Prediction at the Edge. **IEEE Transactions on Network and Service Management**, p. 1–1, 2020.

SARASWAT, S. et al. Challenges and solutions in Software Defined Networking: A survey. **Journal of Network and Computer Applications**, Elsevier Ltd, v. 141, March, p. 23–58, set. 2019.

SCHAEFFER-FILHO, A.; SMITH, P.; MAUTHE, A.; HUTCHISON, D. Network resilience with reusable management patterns. **IEEE Communications Magazine**, v. 52, n. 7, p. 105–115, 2014.

SPEROTTO, A. et al. An overview of IP flow-based intrusion detection. **IEEE Communications Surveys and Tutorials**, v. 12, n. 3, p. 343–356, 2010.

- SYMANTEC. **Internet Security Threat Report (ISTR) 2019** | Symantec. [S.l.: s.n.], abr. 2019. Disponível em: <<https://www.symantec.com/security-center/threat-report>>. Acesso em: 16 mai. 2019.
- TUFANO, M. et al. When and Why Your Code Starts to Smell Bad (and Whether the Smells Go Away). **IEEE Transactions on Software Engineering**, v. 43, n. 11, p. 1063–1088, 2017.
- WICKBOLDT, J. et al. Software-defined networking: management requirements and challenges. **IEEE Communications Magazine**, v. 53, n. 1, p. 278–285, jan. 2015.
- YOUSEFPOUR, A. et al. All one needs to know about fog computing and related edge computing paradigms: A complete survey. **Journal of Systems Architecture**, v. 98, p. 289–330, set. 2019.
- YU, Y. et al. Integrating Clustering and Learning for Improved Workload Prediction in the Cloud. In: IEEE 9th International Conference on Cloud Computing (CLOUD). [S.l.: s.n.], jun. 2016. p. 876–879.
- ZHANG, Y. et al. Secure Inter-Domain Forwarding Loop Test in Software Defined Networks. **IEEE Transactions on Dependable and Secure Computing**, v. 17, n. 1, p. 162–178, 2020.
- ZYRIANOFF, I. D. R. **Fog-Aware Computing: Automated Fog Computing Support for the Internet of Things**. 2019. f. 102. Diss. (Mestrado) – Universidade Federal do ABC (UFABC), Santo André.
- ZYRIANOFF, I. et al. Architecting and Deploying IoT Smart Applications: A Performance–Oriented Approach. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 20, n. 1, p. 84, 2020.

Plano de utilização

Na tabela 7.1, o plano de utilização orçamentário do projeto PROFISSA é apresentado. Esse plano contempla as principais rubricas para execução dos cinco anos de projeto de acordo com as definições exigidas pela FAPESP.

Tabela 7.1: Plano de utilização orçamentária

Orçamento				
Benefícios			Valor (R\$)	Valor (US\$)
Capital				
Material Permanente			96.300,00	0,00
Custeio				
Despesas de Transporte			96.000,00	0,00
Diárias			0,00	24.000,00
Infraestrutura			0,00	0,00
Material de Consumo			0,00	0,00
Serviços de Terceiros			0,00	0,00
Reserva Técnica - Benefícios Complementares			300.000,00	0,00
Reserva Técnica - Custo de Infraestrutura Direta do Projeto			64.620,00	0,00
Provisão para Importação			0,00	0,00
Total			R\$ 556.920,00	US\$ 24.000,00

Quotas de Bolsa				
Modalidade / Nível	Carga Horária	Duração (Meses)	Quantidade	Total
TT-1	15	24	2	R\$21,100.80
TT-3	40	24	4	R\$117,926.40
TT-4	40	24	3	R\$223,545.60
TT-5	40	24	5	R\$884,688.00
Bolsas como Item Orçamentário				R\$ 1.247.260,80
Reserva Técnica Institucional				R\$ 64.620,00
Custo Total da Proposta (em R\$) *				R\$ 1.999.600,80

Como pode ser observado, o projeto PROFISSA prevê a utilização da rubrica de material permanente no valor de R\$96.300,00 para a compra de notebooks para desenvolvimento do projeto (orçamentos seguem anexos a proposta). A implantação da infraestrutura de rede programável descrita da seção 3.3 será oferecida como contrapartida do projeto pela RNP. Além disso, as rubricas de diárias e despesas de transporte serão utilizadas para 16 visitas técnicas/missões a sede do projeto FABRIC na *Florida International University*/FIU ou outras visitas/missões que sejam pertinentes ao desenvolvimento do PROFISSA. As reservas técnicas são apresentadas conforme calculo previsto na chamada. Já, para execução do projeto, prevê-se a necessidade de bolsas para contratação de pessoal/mão de obra nas modalidade de Treinamento Técnico (TT-1,TT-3,TT-4 e TT-5). Cada uma das atividades a serem desenvolvidas nas bolsas de TT-* são melhores descritas nos documentos de “Atividades de bolsas” anexados a submissão dessa proposta de projeto. Por fim, o custo total do projeto de R\$ 1.999.600,80 é apresentado a partir da soma de todas as demais rubricas e reservas técnicas mencionadas.