



Desenvolvimento de Aplicações OpenRAN

**Plataforma e  
Desenvolvimento de xApps e  
rApps**

M3 - A3.3

Programa OpenRAN@Brasil - Fase 2

# Sumário

<b>Lista de ilustrações</b> . . . . .	<b>3</b>
<b>1 Introdução</b> . . . . .	<b>4</b>
<b>2 Plataforma de suporte para desenvolvimento e execução de aplicações OpenRAN</b> . . . . .	<b>7</b>
<b>3 Desenvolvimento de xApps</b> . . . . .	<b>10</b>
3.1 <i>Handover</i> xApp . . . . .	10
3.2 <i>Monitoring</i> xApp . . . . .	11
3.3 Solução RIC-O . . . . .	12
3.4 <i>Intent</i> xApp . . . . .	13
3.5 <i>Slicing</i> xApp . . . . .	14
<b>4 Desenvolvimento de rApps</b> . . . . .	<b>15</b>
4.1 <i>Energy Savings</i> rApp . . . . .	15
4.2 <i>Intent</i> rApp . . . . .	16
<b>5 Treinamento sobre desenvolvimento e execução de aplicações OpenRAN</b>	<b>18</b>
<b>6 Conclusão</b> . . . . .	<b>21</b>
<b>7 Histórico de versões deste documento</b> . . . . .	<b>22</b>
<b>8 Execução e aprovação</b> . . . . .	<b>23</b>

## Lista de ilustrações

Figura 1 – Principais componentes da <i>blueprint</i> v0. . . . .	8
Figura 2 – Principais componentes da <i>blueprint</i> v1. . . . .	8
Figura 3 – Ambiente de desenvolvimento. . . . .	17

# 1 Introdução

Nos últimos anos, a evolução das redes de comunicação móvel tem sido impulsionada pela busca por maior flexibilidade, eficiência e redução de custos. Nesse contexto, a tecnologia OpenRAN (*Open Radio Access Network*) emergiu como um complemento importante aos esforços realizados para evoluir a RAN, introduzindo programabilidade, inteligência e interfaces abertas. OpenRAN oferece a possibilidade de desagregar os componentes da estação base, permitindo a interoperabilidade entre hardware e software de diferentes fornecedores. Isso não apenas promove a inovação, mas também cria um ecossistema mais dinâmico e competitivo.

Com o advento de OpenRAN, em especial a iniciativa O-RAN Alliance, surgem oportunidades significativas para o desenvolvimento de aplicações para atender demandas específicas de redes de acesso por rádio. Essas aplicações podem ser implementadas utilizando interfaces abertas e padronizadas, para melhorar a eficiência operacional, otimizar o desempenho da rede e proporcionar serviços diferenciados aos usuários finais. Este relatório aborda a plataforma de suporte e o desenvolvimento de xApps e rApps, componentes cruciais para a operação eficiente e otimizada das redes OpenRAN. Esses aplicativos desempenham um papel fundamental na orquestração e automação das funções de rede, proporcionando melhorias significativas em termos de desempenho, gestão de recursos e experiência do usuário. A capacidade de desenvolver xApps e rApps independentes do fornecedor de hardware permite uma personalização mais granular das redes móveis, adaptando-as às necessidades específicas de diferentes operadoras e cenários de implementação.

As xApps são projetadas para operar na camada de controle próximo a tempo

real, sendo implementadas no *Near-Real-Time RAN Intelligent Controller* (Near-RT RIC). Elas são responsáveis por tarefas críticas, como otimização de tráfego, alocação de espectro e gestão de interferências, respondendo rapidamente às mudanças nas condições de rede. Por outro lado, as rApps são executadas no *Non-Real-Time RAN Intelligent Controller* (Non-RT RIC) e focam em funções de otimização que não exigem respostas imediatas, como planejamento de rede a longo prazo e análise preditiva.

O desenvolvimento desses aplicativos envolve uma combinação de técnicas avançadas de programação e Inteligência Artificial/Aprendizado de Máquina (do inglês, *Machine Learning/Artificial Intelligence*, além de um profundo entendimento das especificidades das redes de telecomunicações. Este relatório se refere à **Atividade 3.3 - Desenvolver xApps e rApps**, a qual foi dividida nas seguintes ações:

- Criação de plataforma de suporte para desenvolvimento e execução de aplicações OpenRAN.
- Desenvolvimento de xApps.
- Desenvolvimento de rApps.
- Treinamento sobre desenvolvimento e execução de aplicações OpenRAN.

Este relatório corresponde a um dos entregáveis do Projeto Open-RAN@Brasil desenvolvido em parceria entre a Rede Nacional de Ensino e Pesquisa (RNP), Universidade Federal de Goiás (UFG), Universidade Federal do Pará (UFPA) e Universidade do Vale do Rio dos Sinos (UNISINOS).

## Objetivos do Relatório

O objetivo principal deste relatório é apresentar os resultados relacionados à Atividade 3.3, os quais envolvem aplicativos alinhados com a padronização O-RAN para xApps e que funcionem como provas de conceito do funcionamento *near real-time*

e *non real-time* do RIC. Vale destacar que esta atividade ainda está em andamento e, portanto, o presente relatório apresenta os resultados em seu estado atual, o qual terá evoluções até a conclusão.

Ao longo da realização da Atividade 3.3 foram identificadas algumas demandas que não estavam claras durante o planejamento inicial. Isso era previsto, dado que até mesmo os padrões da O-RAN estão em desenvolvimento e o software relacionado passam por atualizações regulares. Assim, foi identificada a necessidade de uma versão estável e de fácil implantação de uma plataforma de suporte para desenvolvimento e execução de aplicações OpenRAN. Outra necessidade identificada foi de um material introdutório para treinamento sobre desenvolvimento e execução de aplicações OpenRAN. Abordar essas necessidades são consideradas ações importantes e objetivos secundários relevantes.

## 2 Plataforma de suporte para desenvolvimento e execução de aplicações OpenRAN

O desenvolvimento de aplicações OpenRAN depende da instalação e configuração de diversas ferramentas que incluem orquestrador de carga de trabalho (i.e., Kubernetes), repositório de contêineres (i.e., *Docker Registry*), arcabouços de desenvolvimento (e.g., *Framework for Python xApp development*), dentre outras. Apesar da ampla documentação, a preparação do ambiente consome tempo significativo, especialmente de usuários iniciantes. Nesse contexto, foi desenvolvida uma solução que encapsula os recursos necessários para desenvolvimento e execução de aplicações OpenRAN, permitindo uma implantação simples, rápida e menos sujeita a erros.

A abordagem adotada foi desenvolver a solução no formato de uma *blueprint*, que no nosso contexto se refere a imagens pré-construídas de máquinas virtuais que contêm todo o ambiente de desenvolvimento e execução pronto para ser utilizado. A primeira *blueprint* é baseada no *Release H* do *Near-RT RIC*, que na época do lançamento era a mais recente da *O-RAN Software Community* (OSC), oferecendo as ferramentas necessárias para a criação de xApps, conforme ilustrado na Figura 1.

A versão inicial, *blueprint* v0, fornece um ambiente básico para o desenvolvimento e testes de xApps. No entanto, essa *blueprint* não permite o envio de mensagens de controle de rádio. Neste contexto, foi criada a *blueprint* v1, a qual inclui um simulador da interface E2 para habilitar a comunicação entre xApps e rádio. Além disso, foram

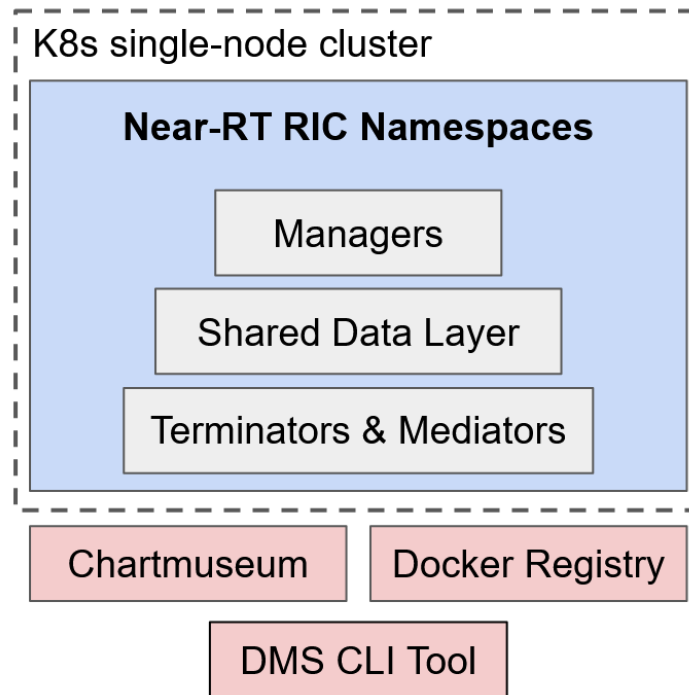


Figura 1 – Principais componentes da *blueprint* v0.

disponibilizadas xApps de exemplo que interagem com o rádio e o novo *Release I* do *Near-RT RIC* foi introduzido nessa *blueprint*. Os principais componentes da *blueprint* v1 são apresentados na Figura 2.

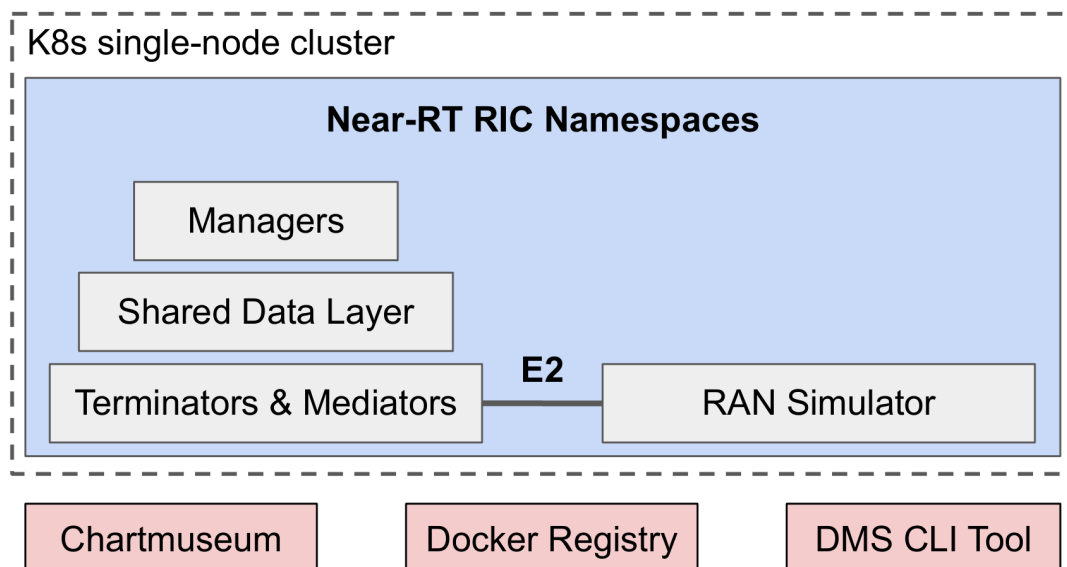


Figura 2 – Principais componentes da *blueprint* v1.

Ambas as versões das *blueprints* podem ser acessadas a partir do seguinte repositório público, com máquinas virtuais disponíveis para os sistemas de virtualização Xen e VirtualBox: <<https://github.com/LABORA-INF-UFG/openran-br-blueprint>>. Novas versões de *blueprints* com novos recursos e atualizações de software estão sendo planejadas e serão divulgadas ao longo do projeto, uma vez que foram observados benefícios tanto para a equipe do projeto como para a comunidade envolvida com OpenRAN no Brasil.

## 3 Desenvolvimento de xApps

### 3.1 *Handover* xApp

O *Handover* xApp foi projetado para aumentar a eficiência energética na rede. Este xApp opera sob a orientação de políticas específicas de economia de energia da *Energy Savings* rApp. A principal função da *Handover* xApp é ajustar dinamicamente o estado operacional da rede em resposta a essas políticas. Quando a política determina a ativação de novas gNBs, a *Handover* xApp facilita esse processo ativando os E2Nodes necessários. Esta ativação não é apenas uma questão de ligar nós adicionais, mas também envolve balanceamento de carga de rede. A *Handover* xApp consegue essa funcionalidade, gerenciando o número de usuários ativos por E2Nodes, garantindo que cada um opere dentro de sua capacidade ideal. Este equilíbrio é mantido através de mecanismos precisos de controle de transferência e admissão, distribuindo eficientemente os *User Equipments* (UEs) pelos E2Nodes disponíveis, otimizando o uso da rede e a qualidade de serviço do usuário e melhorando a eficiência energética.

Quando a política de economia de energia exige a desativação de determinados E2Nodes, a *Handover* xApp é fundamental no gerenciamento dessa transição. Essa xApp garante uma experiência de usuário, movendo UEs dos E2Nodes programados para desativação para aqueles que permanecem ativos. Este processo é meticulosamente gerenciado por meio de protocolos de controle de *handover* e admissão, garantindo que a redistribuição dos UEs não comprometa o desempenho da rede ou a experiência do usuário. A desativação dos E2Nodes, orientada pela política de economia de energia, é um passo significativo para a redução do consumo de energia na rede, contribuindo para

a sustentabilidade e eficiência global da infraestrutura RAN.

Esta xApp é um componente crítico para o fluxo de *Energy Savings*, que começa com a geração de políticas pela *Energy Savings* rApp. Essas políticas, detalhando os estados dos E2Nodes e a eficiência energética, são gerenciadas pelo Sistema de Gerenciamento de Políticas A1 no *Service Management and Orchestration* (SMO) e comunicadas ao RIC Near-RT através da interface A1. A *Handover* xApp implementa essas políticas ativando ou desativando E2Nodes. O código fonte dessa xApp será disponibilizado para a comunidade assim que o artigo científico que está sendo escrito for publicado.

## 3.2 *Monitoring* xApp

A *Monitoring* xApp é um componente crítico no complexo ecossistema RAN, com foco na coleta e análise de várias métricas de desempenho. Esta função de monitoramento é essencial para avaliar a eficiência operacional e a capacidade de resposta da RAN em tempo real. A função principal da *Monitoring* xApp envolve a captura de dados sobre as conexões de UEs, como a potência de rádio das antenas conectadas a UE e das antenas vizinhas do UE. Especificamente, a *Monitoring* xApp captura métricas importantes, como potência recebida do sinal de referência (RSRP), qualidade recebida do sinal de referência (RSRQ), relação sinal-interferência mais ruído (SINR) e indicador de intensidade do sinal recebido (RSSI) para cada gNB conectado e UEs vizinhas. Estas métricas são cruciais para avaliar a qualidade do link de rádio e o desempenho geral da rede.

Além da medição do UE, a *Monitoring* xApp também abrange a coleta de dados relacionados à eficiência energética, rendimento da rede e potência de rádio para cada E2Node. Esta coleta abrangente de dados é vital para operar a *Energy Savings* rApp dentro da estrutura Non-RT RIC. A integração e o processamento dessas informações pelo Near-RT RIC e sua transferência através da interface O1/VES para o Non-RT RIC

equipam a *Energy Savings* rApp com informações essenciais para otimizar o consumo de energia da rede e promover a operação sustentável e eficiente da rede.

A *Monitoring* xApp também estende seus recursos para a interface O-Cloud, que monitora a utilização de recursos de computação, incluindo uso de CPU e memória, bem como a latência entre nós de computação. Esse aspecto de monitoramento é cada vez mais importante nas arquiteturas modernas de RAN baseadas em nuvem, onde o gerenciamento eficiente de recursos computacionais impacta diretamente o desempenho da rede e a experiência do usuário. Os dados coletados pela *Monitoring* xApp são, portanto, essenciais para manter as operações ideais da infraestrutura em nuvem, garantindo que o sistema RAN permaneça robusto e adaptável às diversas demandas e condições da rede. O código fonte dessa xApp será disponibilizado para a comunidade assim que o artigo científico que está sendo escrito for publicado.

### 3.3 Solução RIC-O

Nós exploramos a desagregação do Near-RT RIC em componentes que atendem aos requisitos rigorosos de latência e, ao mesmo tempo, apresentam uma solução economicamente viável. Por exemplo, a proteção contra tempestades de sinalização em uma rede O-RAN exige que o Near-RT RIC suporte loops de controle fim-a-fim tão baixas quanto 10 ms. Neste contexto, nós propomos um novo RIC *Orchestrator* (RIC-O) que otimiza a implantação dos componentes Near-RT RIC em toda a rede envolvendo o paradigma nuvem-borda. Os nós de computação de borda geralmente apresentam recursos limitados e são caros em comparação com a nuvem. Esses componentes críticos para o desempenho do Near-RT RIC e de determinadas xApps devem ser executados na borda, enquanto outros componentes podem ser executados na nuvem. Além disso, o RIC-O emprega uma eficiente estratégia para reagir as mudanças repentinas e reimplantar componentes dinamicamente. O RIC-O é avaliado analiticamente e por meio de experimentos do mundo real em uma implantação estendida do Kubernetes, implementando o RIC-O e o NearRT RIC desagregado.

O RIC-O foi publicado no periódico IEEE Journal on Selected Areas in Communications com o título "RIC-O: Efficient placement of a disaggregated and distributed RAN Intelligent Controller with dynamic clustering of radio nodes"<sup>1</sup>

### 3.4 *Intent* xApp

A *Intent* xApp está sendo projetada para, em combinação com outras xApps, executar a coordenação de ações para a implementação das intenções do operador de rede na RAN. Esta xApp recebe as intenções da *Intent* rApp, expressas em metas de *key performance indicator* (KPI) e dados observáveis da RAN, e decide quais ações devem ser tomadas para atingir esses objetivos, além de definir eventos (temporais e espaciais) que determinam a ativação dessas intenções. Exemplos de ações possíveis incluem o *handover* dos UEs para outras antenas, o controle da potência das antenas e a alocação de recursos de banda.

Para definir as ações a serem executadas, a *Intent* xApp monitora os KPIs e dados relevantes, observando os desvios em relação aos objetivos estabelecidos. Quando ações são necessárias, elas são comunicadas as xApps pertinentes: a xApp de *Handover* para mover os UEs, a *Power Control* xApp para gerenciar a potência das antenas, e a *Resource Manager* xApp para a alocação de recursos. A execução dessas ações deve ser realizada de maneira a garantir a estabilidade da RAN, assegurando que a experiência do usuário seja adequada. A estabilidade da RAN é garantida por meio da coordenação eficaz das ações e pelo monitoramento contínuo das mesmas, utilizando protocolos de controle apropriados. Dessa forma, a *Intent* xApp permite a definição de metas de alto nível para a rede, coordenando a operação da RAN para atingir os resultados desejados. O código fonte dessa xApp será disponibilizado para a comunidade assim que o artigo científico que está sendo escrito for publicado.

<sup>1</sup> <https://doi.org/10.1109/JSAC.2023.3336159>

### 3.5 Slicing xApp

Em redes de acesso com código aberto para um ambiente experimental, ainda não existem implementações fim-a-fim da tecnologia de fatiamento em O-RAN. Por exemplo, no projeto Open Air Interface<sup>2</sup>, existe uma aplicação disponível para *Slicing* xApp que opera com Near-RT RIC, denominada FlexRIC, mas esta não segue estritamente as especificações da aliança O-RAN. No projeto de código aberto srsRAN<sup>3</sup>, recentemente foi iniciada a implementação de partes do *Service Model de RAN Control* para a interface E2. Entre as partes implementadas estão o *Service Style 2*, chamado *Radio Resource Allocation Control*, e o *Service Style 6*, conhecido como *Slice-level Physical Resource Block* (PRB) quota, que permite alterar a porcentagem de PRBs alocada para uma fatia específica da rede. Infelizmente, a implementação ainda está incompleta. Além disso, no projeto srsRAN não há disponibilidade de uma xApp ou Near-RT RIC. Estamos trabalhando no desenvolvimento de um sistema de fatiamento fim-a-fim na RAN, de acordo com as especificações da O-RAN. Utilizamos a *Slicing* xApp com o FlexRIC do projeto Open Air Interface, integrada ao projeto srsRAN, que possui os *Service Styles* necessários para uma implementação experimental em um rádio *Universal Software Radio Peripheral* (USRP). Para assegurar a conformidade entre os dois projetos, ajustamos a numeração dos parâmetros das mensagens do FlexRIC e corrigimos a maneira como as mensagens são interpretadas para configurar a quantidade de PRBs que serão alteradas em todas os UEs registrados em uma fatia de rede específica, ao invés de configurar apenas o UE identificado no cabeçalho da mensagem de controle. Atualmente, nossa implementação ainda não está plenamente em conformidade com as especificações da aliança O-RAN, uma vez que o escalonador do projeto srsRAN não considera a fatia à qual um UE pertence. Estamos trabalhando para tornar o escalonador consciente do fatiamento, o que representa uma modificação estrutural significativa no projeto.

---

<sup>2</sup> <https://openairinterface.org/>

<sup>3</sup> <https://www.srsran.com/>

## 4 Desenvolvimento de rApps

Para o desenvolvimento de rApps, nós criamos um ambiente baseado em qemu/libvirt. Inicialmente, verificamos e gerenciamos as redes virtuais para a criação de uma Máquina Virtual com configurações específicas de CPU, memória e disco. O processo de instalação do sistema operacional Ubuntu Server foi realizado incluindo configurações de rede, particionamento de disco e criação de usuário. Além disso, nós ajustamos as configurações do sistema, das interfaces de rede e preparamos a Máquina Virtual para uso com clusters Kubernetes. Posteriormente, realizamos a instalação de componentes Near-RT RIC e Non-RT RIC e configuramos o armazenamento e monitoramento de dados. Todas as descrições sobre a instalação dos pacotes necessários para criação e gerenciamento da Máquina Virtual, seguida da configuração de uma rede interna utilizando um arquivo XML específico, foram apresentadas no PlugFest Spring 2024 da O-RAN Alliance <<https://github.com/LABORA-INF-UFG/PlugFest-O-RAN-2024/blob/main/README.md>>.

### 4.1 *Energy Savings* rApp

Nós criamos uma *Energy Savings* rApp integrando SMO mínimo, Near-RT RIC, Non-RT RIC, xApps e rApps. Esta integração garante que as políticas de economia de energia sejam formuladas com base em uma análise abrangente de dados e implementadas de forma eficaz, promovendo o desempenho da rede e minimizando o consumo de energia. Figura 3 mostra a sequência de interações entre os componentes utilizados para a solução de economia de energia, considerando três fluxos, (i) Fluxo de Monitoramento, (ii) Fluxo de Potência do Sinal do E2Node e (iii) Fluxo de Política de *handover*.

O processo começa com uma solicitação de conexão do UE encaminhado pelo E2Node. Após a conexão bem-sucedida, a *Monitoring* xApp exporta das UEs os seus status de conexão e métricas de desempenho para o Sistema de Monitoramento. O Sistema de Monitoramento transmite esses dados através da interface O1/VES. O componente *Data River* envia essas métricas de desempenho aos consumidores inscritos. Simultaneamente, a *Energy Savings* rApp calcula a solução de otimização de energia e emite a configuração de energia do rádio do E2Node através da interface O1. Esta configuração é seguida pela transmissão da política de *handover* do UE através da interface A1, que ocorre em paralelo. O procedimento de *handover* do UE é iniciado utilizando uma máquina de estado, que aciona o processo de *handover* para UEs utilizando o protocolo *E2 Application Protocol* (E2AP). Uma solicitação de transferência é enviada somente mediante um acionador específico e, após a conclusão, as sequências de desconexão e reconexão do UE são iniciadas. Por fim, o E2Node ajusta a configuração da potência do rádio pós-*handover*, completando o processo. O código fonte dessa rApp será disponibilizado para a comunidade assim que o artigo científico que está sendo escrito for publicado.

## 4.2 *Intent* rApp

O uso de redes baseadas em intenções visa a coordenação eficiente de redes de grande porte, onde a gestão manual através de interfaces de comando ou políticas se torna impraticável, devido ao grande número de elementos a serem coordenados. A abordagem baseada em intenções fornece a automação necessária para a coordenação dessas redes complexas. A *Intent* rApp é responsável por transmitir as intenções para as xApps. As intenções definem objetivos de alto nível para a rede, que são expressos por meio de KPIs e outros dados observáveis da RAN. Esses objetivos são recebidos de uma aplicação *front-end*, que pode ser uma aplicação de monitoramento da rede usando *Machine Learning*, uma aplicação de processamento de linguagem natural, uma interface gráfica de usuário (GUI), ou uma interface de usuário textual (TUI). Essas aplicações convertem suas entradas para um formato *JavaScript Object Notation* (JSON), que é

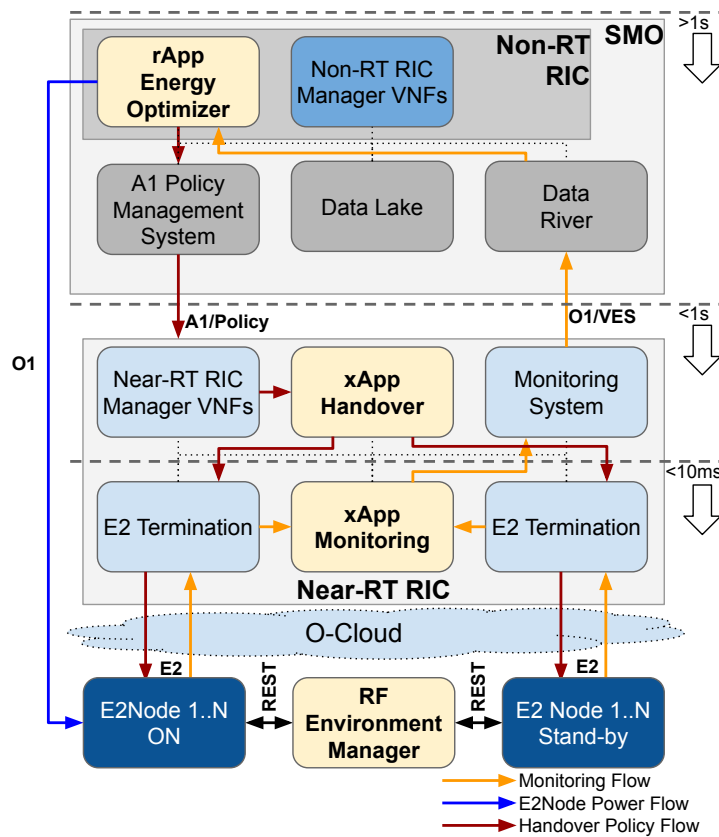


Figura 3 – Ambiente de desenvolvimento.

utilizado pelo rApp.

Além das métricas a serem otimizadas, a *Intent* rApp também pode conter informações adicionais que definem a aplicabilidade da intenção, tais como data e horário de aplicação, duração da aplicabilidade, localidade, ou eventos específicos da RAN que tornam uma intenção ativa. Localidades referem-se a locais físicos, como a orla do rio Guaíba ou a praça da Sé, enquanto eventos da RAN podem incluir a falha de nodos específicos da rede ou a congestão de pilhas de rede. O código fonte dessa rApp será disponibilizado para a comunidade assim que o artigo científico que está sendo escrito for publicado.

## 5 Treinamento sobre desenvolvimento e execução de aplicações OpenRAN

A iniciativa O-RAN Alliance oferece ampla documentação no formato de padrões ou descrições sobre instalação e uso através da O-RAN OSC. No entanto, esse tipo de documentação apresenta uma barreira de entrada significativa para os desenvolvedores interessados em criar aplicações OpenRAN. Um dos motivos é a necessidade de conhecer não apenas fundamentos de desenvolvimento de software, mas também de telecomunicações, mais especificamente, relacionados à RAN. A maior parte dos documentos assumem que os leitores são experientes em ambas as áreas, além de estar atualizados com diversas tecnologias como virtualização de funções de rede, redes definidas por software, aprendizado de máquina, orquestradores de carga de trabalho, Redes 5G, dentre outras. Além disso, a documentação dos padrões não tem nenhum compromisso em ser didática e estruturada para iniciantes. Enquanto a documentação oferecida pela OSC é bastante limitada e não tem acompanhado a evolução dos software, exigindo um esforço significativo até mesmo para dar os primeiros passos no desenvolvimento e execução de aplicações OpenRAN.

Nesse contexto, com o intuito de oferecer à equipe do projeto uma formação básica homogênea e de qualidade, foi desenvolvido um material para iniciantes no desenvolvimento e testes de aplicações OpenRAN, o qual inclui *slides*, exercícios e aplicações simples de exemplo. O material foi apresentado também em um formato *workshop*, em três encontros virtuais, os quais foram gravados e serão disponibilizados futuramente, após edição.

Todo o material está disponível no seguinte repositório público: <<https://github.com/LABORA-INF-UFG/xapp-workshop>>. A solução baseada em *blueprints*, apresentada no Capítulo 2, se mostrou bastante útil, permitindo que os participantes do *workshop* pudessem ter acesso rápido e fácil a uma plataforma pronta para desenvolvimento e testes de aplicações. O *workshop* foi organizado da seguinte forma:

- Primeiro encontro: Gerenciando, verificando e configurando xApps.
- Segundo encontro: Visão geral de uma xApp, geração de *logs*, *Shared Data Layer* (SDL) e *Representational State Transfer* (REST).
- Terceiro encontro: Comunicação *RIC Message Router* (RMR) e assinatura de nós E2.

Ao final do primeiro encontro, os participantes são capazes de:

- Integrar, instalar e desinstalar xApps.
- Criar e enviar imagens xApp.
- Verificar *pods*, serviços e registros de xApp.
- Verificar se as xApps estão cadastrados no *Application Manager*.
- Gerenciar os principais aspectos de um descritor xApp (arquivo de configuração).

Ao final do segundo encontro, os participantes são capazes de:

- Entender a estrutura de um xApp escrito usando o *Python xApp Framework* da OSC.
- Registrar mensagens do xApp usando a biblioteca *mdclogpy* da OSC.
- Criar, recuperar, atualizar e excluir dados usando a API da SDL.
- Desenvolver interação REST (como servidor e cliente) para uma xApp.

Ao final do terceiro encontro, os participantes são capazes de:

- Definir tabelas de rotas estáticas para mensagens RMR.
- Enviar/receber mensagens RMR de/para componentes *Near-RT RIC* (incluindo outras xApps).
- Assinar e cancelar a assinatura de nós E2.

## 6 Conclusão

O desenvolvimento de aplicações OpenRAN é uma oportunidade estratégica para a comunidade brasileira, pois permite que o país participe de maneira efetiva dos esforços mundiais para evolução das redes de acesso sem fio. A absorção desse tipo de conhecimento permite que a indústria nacional de software crie soluções que podem ser adotadas globalmente, gerando negócios, empregos e renda a longo prazo. Além disso, contribui para a redução da dependência tecnológica na área estratégica crítica de telecomunicações.

Neste relatório, foram apresentadas as ações que vem sendo realizadas no contexto da Atividade 3.3, as quais contribuem para o desenvolvimento de xApps e rApps. Essas ações já geraram evidências de sua execução como documentação, código de software, material de treinamento, artigos científicos, dentre outros. Como essa atividade ainda está em execução, suas contribuições ainda passarão por evolução até a conclusão.

## 7 Histórico de versões deste documento

<u>Data de Emissão</u>	<u>Versão</u>	<u>Descrição das Alterações Realizadas</u>
10/07/2024	1.0	Construção inicial do relatório
DD/MM/2024	1.?	Revisão de todo documento

## 8 Execução e aprovação

**Executado por:**

\_\_\_\_\_  
Luan Rios

**Revisado por:**

\_\_\_\_\_  
Fernando Farias

**Aprovado por:**

\_\_\_\_\_  
Fernando Farias

**Data da emissão:** DD/MM/2024