

Brasil 6G

Projeto Brasil 6G

Atividade 2.3 Definição e aquisição dos componentes de software da plataforma Brasil 6G



UNIVERSIDADE FEDERAL
DO RIO DE JANEIRO



Histórico de Atualizações:

Versão	Data	Autor(es)	Notas
1	20/07/22	Daniely Gomes Silva Mariana Baracat de Mello Eduardo Saia Lima Juliano Silveira Ferreira Sand Luz Correa Cristiano Bonato Both Felipe Augusto Pereira de Figueiredo Tibério Tavares Rezende	Elaboração de conteúdo
2	27/07/22	Luciano Leonel Mendes	Revisão de texto

Acrônimos

3GPP	<i>3rd Generation Partnership Project</i>
4G	<i>Quarta Geração</i>
5G	<i>Quinta Geração</i>
6G	<i>Sexta Geração</i>
AI	<i>Artificial Intelligence</i>
API	<i>Application Programming Interface</i>
AR	<i>Augmented Reality</i>
ARENA	<i>Augmented Reality Edge Networking Architecture</i>
CDN	<i>Content Delivery Network</i>
CNI	<i>Container Network Interface</i>
CoAP	<i>Constrained Application Protocol</i>
CPU	<i>Central Process Unit</i>
DLT	<i>Distributed Ledger Technology</i>
EDA	<i>Event-driven Architecture</i>
eMBMS	<i>Evolved Multimedia Broadcast Multicast Services</i>
eNodeB	<i>enhanced Node B</i>
EPC	<i>Evolved Packet Core</i>
ETSI	<i>European Telecommunications Standards Institute</i>
FIBRE	<i>Future Internet Brazilian Environment for Experimentation</i>
gNodeB	<i>next generation Node B</i>
GPP	<i>General Purpose Processors</i>
GPGPU	<i>General Purpose Graphics Processing Unit</i>
GPU	<i>Graphics Processing Unit</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IA	<i>Inteligência Artificial</i>
IMS	<i>IP multimedia subsystem</i>
Inatel	<i>Instituto Nacional de Telecomunicações</i>
IoT	<i>Internet of Things</i>
ISP	<i>Internet Services Providers</i>
JNI	<i>Java Native Interfaces</i>
LDPC	<i>Low-Density Parity Check</i>
LTE	<i>Long Term Evolution</i>
LTE-A	<i>Long Term Evolution Advanced</i>
LTE-M	<i>Long Term Evolution for Machines</i>

MANO *Management and Orchestration*
ML *Machine Learning*
MQTT *Message Queuing Telemetry Transport*
NAS *Non-Access Stratum*
NB-IoT *Narrowband IoT*
NFV *Network Function Virtualization*
NGAP *New Generation Application Protocol*
NR *New Radio*
OFDM *Orthogonal Frequency Division Multiplexing*
ONOS *Open Network Operating System*
OSA *OpenAirInterface Software Alliance*
OSM *Open Source Management and Orchestration*
PCIe *Peripheral Component Interconnect Express*
RAN *Radio Access Network*
RoF *radio-over-fiber*
SDN *Software Defined Network*
SDR *Software Defined Radio*
SVM *Support Vector Machines*
UFG *Universidade Federal de Goiás*
UHD *USRP Hardware Driver*
VIM *Virtualized Infrastructure Manager*
VNF *Virtual Network Functions*
VoLTE *voice over LTE*
WAN *Wide Area Network*
XR *Extended Reality*

Sumário

1	Introdução	1
2	<i>Softwares</i> para Aplicações	2
3	<i>Softwares</i> para Núcleo da Rede	6
4	<i>Softwares</i> para Plataforma de Virtualização	7
5	<i>Softwares</i> para Suporte a IA	9
6	<i>Softwares</i> para os Transceptores da Rede de Longo Alcance	12
7	Software para Implementação de Pilhas de Protocolos Utilizados na Estação Radiobase 5G	13
8	Conclusão	14

1 Introdução

A evolução das comunicações sem fio tem proporcionado um aumento exponencial do número de dispositivos conectados e estima-se que, em 2030, a Quinta Geração (5G) de Redes Móveis não seja capaz de atender à essa demanda massiva de usuários. Dessa forma, a implantação da Sexta Geração (6G) de Redes Móveis virá para garantir um aumento no desempenho e na qualidade de serviço para todos os usuários da rede, além de dar suporte a aplicações inovadoras. A complexidade desse cenário futuro requer um controle preciso dos recursos disponíveis no sistema e também estabelece requisitos para que essa futura rede possa entrar em operação. A definição desses requisitos, aliada ao uso de tecnologias habilitadoras, é considerada fundamental para viabilizar os casos de uso e aplicações que integrarão definitivamente os mundos digital, físico e biológico [1]. Dessa forma, os sistemas de comunicação vigentes terão que passar por uma série de transformações que os habilitarão a operar nas condições impostas pela rede 6G. Essas mudanças afetarão todos os níveis da comunicação, que se estendem desde a camada de aplicação até a camada física.

A arquitetura de um sistema de redes móveis define a forma como ele é organizado e abrange as funções e requisitos necessários para caracterizar os serviços oferecidos por esse sistema. Para alcançar as metas propostas dessa futura rede de comunicações, um novo conceito de arquitetura será implementado [2]. Nessa arquitetura, a Inteligência Artificial (IA) será utilizada em diferentes níveis de processamento, garantindo a conectividade inteligente entre os inúmeros usuários e dispositivos que estarão aptos a analisar e executar a alocação de recursos. Inovações no processo de implementação de soluções estão sendo propostas. Dentre essas inovações verifica-se o desenvolvimento de uma plataforma que compõe a rede 6G, abrangendo desde o núcleo da rede até as camadas de transporte, física e de enlace da *Radio Access Network* (RAN). Com base nesse contexto, esse relatório tem como objetivo apresentar os *softwares* que serão utilizados na Plataforma do Projeto Brasil 6G, trazendo uma breve descrição de quais estão sendo empregados nas aplicações previstas neste projeto, no desenvolvimento do núcleo da rede e nas plataformas de virtualização. Também serão apresentados os *softwares* aplicados no suporte à IA, na implementação dos transceptores da rede de longo alcance e nas pilhas de protocolos previstas para a operação de uma estação radiobase 5G, cuja evolução resultará na infraestrutura da próxima geração.

2 Softwares para Aplicações

Daniely Gomes Silva, Mariana Baracat de Mello, Eduardo Saia Lima, Juliano Silveira
 Ferreira, Sand Luz Correa, Tibério Tavares Rezende
 daniely@inatel.br, mariana.mello@dtel.inatel.br, elima@get.inatel.br, silveira@inatel.br,
 sandluz@ufg.br, tiberio@mtel.inatel.br

Nesta seção será apresentada uma breve descrição dos *softwares* empregados para o desenvolvimento das aplicações propostas no Projeto Brasil 6G:

1. **Matplotlib** [3]: biblioteca de *software* de código aberto para criação de gráficos e visualizações de dados em geral, desenvolvida para a linguagem de programação *Python* e sua extensão de matemática NumPy. Foi projetada e desenvolvida para ter a mesma usabilidade do MATLAB, mas com a flexibilidade da linguagem *Python*, apresentando a vantagem de ser código aberto (*open source*) e totalmente gratuita. Esta biblioteca poderá ser utilizada para criar visualizações estáticas, animadas e interativa em *Python*.
2. **OptiSystem** [4]: possibilita projetar, testar e avaliar o desempenho de sistemas ópticos completos, viabilizando a obtenção de especificações técnicas dos componentes ópticos e elétricos que deverão ser utilizados na implementação prática das redes 6G. Especificamente, o *software* será utilizado em aplicações de transmissão de sinais *radio-over-fiber* (RoF), possibilitando avaliar de maneira completa a qualidade de enlaces ópticos, considerando todos os efeitos ópticos e elétricos que ocorrem nos sistemas reais. Dessa forma, o uso *software* possibilita uma avaliação sobre a viabilidade da implantação prática do sistema, bem como especificar as principais características dos componentes a serem adquiridos, tais como a potência e banda de modulação do laser, tipo de moduladores e fibras ópticas, banda de operação do fotodetector, além de ganho e figura de ruído dos amplificadores.
3. **Thingsboard** [5]: plataforma de *Internet of Things* (IoT) para coleta, processamento, visualização e gerenciamento de dados de dispositivos. Essa plataforma permite a criação e visualização de dados em tempo real e controle remoto de dispositivos através de *dashboards* configuráveis e personalizáveis. A plataforma em questão suporta os protocolos *Message Queuing Telemetry Transport* (MQTT), *Constrained Application Protocol* (CoAP) e *Hypertext Transfer Protocol* (HTTP). É um *software* a ser utilizado por aplicações, cujo principal uso seria para demonstrar a integração de dispositivos e aplicações de IoT com o transceptor da rede de longo alcance.
4. **ARENA** [6]: *Augmented Reality Edge Networking Architecture* (ARENA) é um arcabouço projetado para desenvolver aplicações de *Augmented Reality* (AR) de forma simplificada e escalável. Particularmente, o arcabouço facilita o desenvolvimento de aplicações de *Extended Reality* (XR) baseadas em web. Para atingir esse objetivo, o arcabouço inclui: (1) um serviço de pesquisa geoespacial, (2) um armazenamento de dados persistente, (3) um sistema de autenticação e controle de acesso, (4) um cliente web *front-end*, e (5) um mecanismo de execução que hospeda aplicações próximas a usuários e/ou fontes de dados. No projeto, desejamos usar o ARENA para criar e executar cenas de realidade estendida no caso de uso sobre *smart farm*. Mais especificamente, desejamos demonstrar a diferença na latência em dois cenários de interação: (1) quando o usuário interage com objetos da cena e estes estão armazenados próximo ao usuário (*edge*) e (2) quando o

usuário interage com objetos da cena e estes estão armazenados na nuvem. A cena ainda não está definida. Neste momento estamos estudando como criar, executar e posicionar cenas no ARENA.

5. **Digitanimal** [7]: o *software* de monitoramento remoto da empresa *Digitanimal* permite que as informações de monitoramento de coleiras de rebanho de gado, coletadas localmente, sejam transferidas e processadas pelos servidores remotos da empresa e sejam disponibilizadas através de uma plataforma web em um aplicativo móvel. Como as informações coletadas a partir de coleiras de gado e disponibilizadas em tais plataformas pode-se destacar a localização, temperatura e informações comportamentais do gado. Esse *software* irá viabilizar uma importante aplicação prevista na demonstração do Projeto Brasil 6G, que é o monitoramento remoto de gado. Na demonstração prevista, as informações coletadas em uma fazenda são transferidas ao servidor da empresa *Digitanimal* fazendo uso do enlace de comunicação disponibilizado pelo transceptor do Projeto Brasil 6G. As informações acessíveis na plataforma web e aplicativo móvel, viabilizadas graças a aquisição da renovação de licença em questão, permitem comprovar que uma importante aplicação do agronegócio é viabilizada pela conectividade provida pelo transceptor 6G.
6. **OSM** [8]: o *Open Source Management and Orchestration* (OSM) é uma plataforma de código aberto hospedada no *European Telecommunications Standards Institute* (ETSI) que fornece uma pilha para *Management and Orchestration* (MANO) para aplicações de *Network Function Virtualization* (NFV). O OSM é capaz de trabalhar com modelos de informações publicados abertamente, adequados para gerenciamento de *Virtual Network Functions* (VNF), e que pode interagir com diferentes *Virtualized Infrastructure Manager* (VIM), dependendo da aplicação a ser construída. Em relação aos VIMs compatíveis com OSM, está sendo considerada a utilização do *OpenStack* ou *Eclipse fog05*. Isso dependerá de qual VIM atenda melhor as demandas do projeto. No contexto do Projeto Brasil 6G, este software será utilizado para criação de aplicações NFV fora do núcleo 5G (que já possui funções virtualizadas com NFV) para testes de cenários que abordam redes terrestre-satélite. É importante ressaltar que essa aplicação precisa de interoperabilidade direta com um VIM (*OpenStack* ou *Eclipse fog05*).
7. **ONOS** [9]: *Open Network Operating System* (ONOS) é um controlador *Software Defined Network* (SDN), projetado para grandes redes, baseado em um *cluster* de controladores auto-coordenados. O ONOS é um projeto de código aberto desenvolvido pelo ON.Lab e mantido pela *Linux Foundation*, apoiado por uma grande comunidade de operadores de rede e fornecedores. ONOS foi projetado e é utilizado especificamente para lidar com problemas de confiabilidade e escalabilidade que surgem em grandes redes *Internet Services Providers* (ISP) e *Wide Area Network* (WAN). O sistema suporta nativamente uma versão distribuída do controlador, rodando em um *cluster* de servidores. Para o contexto dos Projeto Brasil 6G, este aplicativo será utilizado para testes de escalabilidade de redes terrestre-satélite funcionando com SDN. Um *testbed* experimental possível é a utilização de um *cluster* de controladores ONOS para avaliação da quantidade de tráfego trocado entre os mesmos. Como o tráfego depende das atualizações específicas confirmadas na estrutura de dados compartilhada, o problema pode ser abordado analisando o impacto de cada atualização para todas as estruturas de dados compartilhadas (ou seja, topologia, fluxo e armazenamentos de *host*) que gerenciam o estado da rede.

8. **OpenStack** [10]: o *software OpenStack* é uma plataforma *open source* que usa recursos virtuais agrupados para criar e gerenciar nuvens dos tipos privadas e públicas. As ferramentas que abrangem a plataforma *OpenStack* lidam com os serviços essenciais de *Cloud Computing*. Entre estes serviços é possível citar: computação, topologia e configuração de rede, armazenamento, identidade e imagens de sistemas operacionais. No contexto do Projeto Brasil 6G, esse *software* será utilizado como VIM em interoperabilidade com o OSM.
9. **Eclipse fog05** [11]: *Eclipse fog05* é um projeto *open source* que visa fornecer uma infraestrutura descentralizada para provisionar e gerenciar recursos de computação, armazenamento, comunicação e dispositivos de entrada e saída disponíveis em qualquer lugar da rede. O gerenciamento e controle pode ser levado mais próximo dos usuários por aplicações criadas pela junção de técnicas de *Cloud Computing* e IoT. O *Eclipse fog05* aborda sistemas altamente heterogêneos, mesmo aqueles com nós extremamente limitados pelos recursos solicitados pelos usuários, e pode alavancar a infraestrutura em nuvem ou funcionar igualmente sem ela. No contexto do Projeto Brasil 6G, este software será utilizado como VIM em interoperabilidade com o OSM.
10. **Shell Script (GNU/Linux)** [12]: *shell script* é o nome dado a um arquivo codificado que será interpretado por algum programa tipo *shell* em *GNU/Linux*. Atualmente existem vários programas do tipo *shell*. Além dos principais *sh* e *bash*, existem também *ksh*, *zsh*, *csh* e *tsh*. Um arquivo escrito e definido como *shell script* necessita basicamente do interpretador *shell* nativo em sistemas operacionais baseados em *Unix*. Para o contexto do Projeto Brasil 6G, este software será utilizado para automatização de processos e integração dos demais *softwares* e plataformas selecionados para o projeto.
11. **Nginx** [13]: *Nginx* (ou *Engine-x*) é uma ferramenta utilizada como servidor *web* que trabalha com os protocolos HTTP/HTTPS e IMAP/POP3/Proxy. O *Nginx* é semelhante ao servidor *Apache* e foi gerado a partir da necessidade de desenvolver uma solução capaz de suportar a crescente demanda de solicitações na Internet desde 2002. Esse servidor funciona com o método *Event-driven Architecture* (EDA), que é preparado para lidar com uma maior demanda de conexões se comparado ao modo convencional, como é o caso do *Apache*, no qual o processamento é baseado em um único segmento para cada usuário. Pode ser utilizado no contexto do Projeto Brasil 6G como opção de servidor *web* para aproveitar os recursos de *proxy* e armazenamento em *cache*. Para isso, é necessário compilar o *Nginx* e alavancar o módulo *proxy*. Este módulo permite armazenar dados em *cache* nos discos locais dos pontos de presença remotos *Content Delivery Network* (CDN). Precisa de interoperabilidade com *Varnish cache* para funcionar como um servidor CDN.
12. **Imagem IOTA** [14]: *IOTA* é uma criptomoeda que usa uma tecnologia diferente de *blockchain*. Essa criptomoeda surgiu da necessidade de soluções financeiras seguras e descentralizadas voltadas para IoT. *IOTA* foi descrita em 2016, porém os primeiros conceitos surgiram em 2014. No contexto do Projeto Brasil 6G será utilizada como tecnologia de *Distributed Ledger Technology* (DLT) para testes com aplicação de IoT.
13. **Hyperledger** [15]: trata-se de um projeto colaborativo envolvendo várias indústrias que foi iniciado em dezembro de 2015 pela *Linux Foundation*. O *Hyperledger* possui o objetivo de avançar a tecnologia de registro distribuído *blockchain* em múltiplos segmentos da

indústria. Para o Projeto Brasil 6G será utilizado como tecnologia de DLT para testes com aplicação de IoT e comparação com a imagem *IOTA*.

14. **Metamask** [16]: *Metamask* é uma carteira para cripto-moedas e também um *gateway* para aplicações *blockchain*. Pode ser instalado como uma extensão no navegador *web* e como um aplicativo *mobile*. *Metamask* auxilia na criação de contas dos atores (chaves privada e pública). Essa carteira digital é um *gateway* flexível que permite a conexão com diferentes redes, inclusive redes privadas em ambiente de teste local ou público. No contexto do Projeto Brasil 6G será utilizado para simular a carteira de um ator nas aplicações que utilizam DLT (horta inteligente com *IOTA*, monitoramento e rastreamento de vacinas e cadeia de alimentos frescos com gêmeos digitais e IoT).
15. **Cooja** [17]: é um simulador do sistema operacional *Contiki* para IoT. Um *Contiki mote* simulado no *Cooja* é um sistema *Contiki* real compilado e em execução. O sistema operacional *Contiki* é um sistema *open source* para IoT que permite ligar à Internet microcontroladores de baixo custo e também de baixo consumo energético. Isso é realizado compilando o *Contiki* para a plataforma nativa como uma biblioteca compartilhada e carregando a biblioteca em *Java* usando *Java Native Interfaces* (JNI). Várias bibliotecas *Contiki* diferentes podem ser compiladas e carregadas na mesma simulação *Cooja*, representando diferentes tipos de nós sensores ou redes heterogêneas. O *Cooja* controla e analisa um sistema *Contiki* através de algumas funções. No contexto do Projeto Brasil 6G será utilizado para testar escalabilidade de uma aplicação IoT.

3 *Softwares* para Núcleo da Rede

Daniely Gomes Silva, Cristiano Bonato Both
 daniely@inatel.br, cbboth@unisinis.br

Nesta seção será apresentada uma breve descrição dos *softwares* empregados na implementação do núcleo da rede proposto no Projeto Brasil 6G:

1. **Free5GC** [18]: projeto de código aberto para as redes móveis de 5G. O objetivo principal desse projeto é implementar a rede principal 5G definida nas *Releases* 15 e 16 lançadas pelo *3rd Generation Partnership Project* (3GPP). Esse projeto poderá ser utilizado como base inicial na integração dos componentes de *software*, integração das redes do Instituto Nacional de Telecomunicações (Inatel), integração com as demais ilhas *Future Internet Brazilian Environment for Experimentation* (FIBRE), integração com a nuvem da Universidade Federal de Goiás (UFG) e testes e validação das integrações.
2. **Open Air Interface** [19]: *OpenAirInterface Software Alliance* (OSA) é um consórcio sem fins lucrativos para desenvolver um ecossistema de *software* e *hardware* de código aberto para redes celulares 3GPP. A aliança suporta *OpenAirInterface* para o desenvolvimento da pilha de protocolos das redes celulares em *hardware* comercial de baixo custo. Nesse contexto, *OpenAirInterface* poderá ser utilizado na integração dos componentes de *hardware* e *software*, integração das redes do Inatel, integração com as demais ilhas FIBRE, integração com a nuvem da UFG e testes e validação das integrações.

4 *Softwares* para Plataforma de Virtualização

Daniely Gomes Silva, Cristiano Bonato Both, Tibério Tavares Rezende
 daniely@inatel.br, cbboth@unisinios.br, tiberio@mtel.inatel.br

Nesta seção será apresentada uma breve descrição dos *softwares* empregados nas plataformas de virtualização (*cloud-native*) propostas no Projeto Brasil 6G. Esses *softwares* podem ser usados no núcleo da rede, mas também podem ser usados em outras partes da rede, como por exemplo RAN e transporte.

1. **Kubernetes** [20]: projeto de código aberto utilizado para automatizar a implantação, o dimensionamento e o gerenciamento de aplicativos em uma unidade padrão de *software* (*container*) que empacota o código e todas as suas dependências para que o aplicativo seja executado de forma rápida e confiável de um ambiente de computação para outro. *Kubernetes* poderá ser utilizado como plataforma de virtualização na integração dos componentes de *software*, integração das redes do Inatel, integração com as demais ilhas FIBRE, integração com a nuvem da UFG e testes e validação das integrações.
2. **Helm** [21]: ferramenta para gerenciar pacotes de recursos pré-configurados do *Kubernetes*, ou seja, *Helm* é uma ferramenta que simplifica a instalação e o gerenciamento de aplicativos *Kubernetes*. Dessa forma, *Helm* poderá ser utilizado para gerenciar a instalação na integração dos componentes de *software*, integração das redes do Inatel, integração com as demais ilhas FIBRE, integração com a nuvem da UFG e testes e validação das integrações.
3. **Ansible** [22] : plataforma base para construir e operar a automação de *software*. A plataforma inclui todas as ferramentas necessárias para implementar a automação em toda a infraestrutura de *software*. Nesse contexto, *Ansible* poderá ser utilizado para gerenciar a automação da instalação de *software* na integração dos componentes de *software*, integração das redes do Inatel, integração com as demais ilhas FIBRE, integração com a nuvem da UFG e testes e validação das integrações.
4. **Graphana** [23]: aplicação web de análise de código aberto multiplataforma e visualização interativa da web. Essa aplicação fornece tabelas, gráficos e alertas para a web quando conectado a fontes de dados suportadas. Além disso, é expansível através de um sistema de *plug-in*. *Grafana* poderá ser utilizado para as visualizações das redes do Inatel, ilhas FIBRE, a nuvem da UFG e testes e validação das integrações de redes.
5. **Prometheus** [24]: projeto baseado em computação em nuvem para um sistema de monitoramento de sistemas e serviços. O sistema coleta métricas de destinos configurados em determinados intervalos, avalia expressões de regras, exibe os resultados e pode acionar alertas quando condições especificadas são observadas. Nesse contexto, *Prometheus* poderá ser utilizado no monitoramento das redes do Inatel, ilhas FIBRE, nuvem da UFG e validação das integrações dessas redes.
6. **Docker** [25]: *Docker* é um conjunto de aplicações de plataforma como serviço que usam virtualização de nível de sistema operacional para entregar *software* em pacotes chamados contêineres. Os contêineres são isolados uns dos outros e agrupam seus próprios *softwares*, bibliotecas e arquivos de configuração de rede e de operação. Nesse contexto, será utilizado

para criação de contêineres das aplicações criadas para os Projetos SINDISAT C e Brasil 6G visando a automatização no processo de testes, tanto dentro quanto fora da rede FIBRE e suas ilhas dentro do cenário de redes terrestre-satélite. O *Docker* é gerenciado pelo *Kubernetes* (em interoperabilidade e gerenciamento) e possui vasta documentação disponível para análise.

7. **Multus** [26]: é um *plug-in* que permite anexar várias interfaces de rede aos *Pods* do *Kubernetes*. Tradicionalmente, múltiplas interfaces de rede são empregadas por funções de rede para fornecer separação de planos de rede de controle, gerenciamento e dados/usuário. Eles também são usados para suportar diferentes protocolos ou pilhas de *software* e diferentes requisitos de ajuste e configuração. A Intel ajudou a facilitar os casos de uso de NFV em ambientes de contêiner, contribuindo com o *plug-in* de dispositivo *Multus Container Network Interface* (CNI). Esse *plug-in* que pode ser usado para criar várias interfaces de rede para *Pods* no *Kubernetes*. Isso é feito pelo *Multus* agindo como um *meta-plugin*, um *plug-in* CNI que pode chamar vários outros *plug-ins* CNI. No contexto do Projeto Brasil 6G, o *plug-in* vai ser usado para gerenciar redes e subredes necessárias para a comunicação interna nos protocolos *New Generation Application Protocol* (NGAP) e *Non-Access Stratum* (NAS) com o núcleo 5G, particularmente no projeto *free5GC* desenvolvido por professores da UNISINOS e UFG.

5 Softwares para Suporte a IA

Daniely Gomes Silva, Mariana Baracat de Mello, Eduardo Saia Lima, Felipe Augusto Pereira de Figueiredo

daniely@inatel.br, mariana.mello@dtel.inatel.br, elima@get.inatel.br,
felipe.figueiredo@inatel.br

Nesta seção será apresentada uma breve descrição dos *softwares* empregados para suporte a *Artificial Intelligence* (AI) utilizados no Projeto Brasil 6G:

1. **Matlab** [27]: *software* interativo de alta performance voltado para o cálculo numérico. O *Matlab* integra análise numérica, cálculo com matrizes, processamento de sinais e construção de gráficos em ambiente fácil de usar onde problemas e soluções são expressos somente como eles são escritos matematicamente. Esse *software* poderá ser usado para simular sistemas de comunicação sem fio, incluindo transmissores, canais e receptores. Também poderá ser utilizado para a geração de dados de treinamento e de validação para redes neurais.
2. **Jupyter Notebook** [28]: interface gráfica que permite a edição de *notebooks* em um navegador *web*. Poderá ser usado no projeto de redes neurais para desenvolver, documentar e executar o código, assim como informar os resultados encontrados.
3. **Numpy** [29]: biblioteca de código aberto com funções para se trabalhar com computação numérica em *Python*, que suporta o processamento de grandes matrizes multidimensionais, juntamente com uma grande coleção de funções matemáticas de alto nível para operar sobre essas matrizes.
4. **Matplotlib** [3]: biblioteca de *software* de código aberto para criação de gráficos e visualizações de dados em geral, desenvolvida para a linguagem de programação *Python* e sua extensão de matemática NumPy. Foi projetada e desenvolvida para ter a mesma usabilidade do *Matlab*, mas com a flexibilidade da linguagem *Python*, apresentando a vantagem de ser código aberto (*open source*) e totalmente gratuita. Essa biblioteca poderá ser utilizada para criar visualizações estáticas, animadas e interativa em *Python*.
5. **Tensorflow** [30]: biblioteca de código aberto para *Machine Learning* (ML) aplicável a uma ampla variedade de tarefas. É um *framework* para criação e treinamento de redes neurais para detectar e decifrar padrões e correlações. Além disso, o *TensorFlow* permite executar outras bibliotecas de rede neural de alto nível, como a *Application Programming Interface* (API) funcional *Keras*. O *TensorFlow* pode ser executado em dispositivos individuais e também em múltiplas *Central Process Units* (CPUs) e *Graphics Processing Units* (GPUs). Portanto, essa biblioteca poderá ser utilizada para a implementação de redes neurais e outras abordagens de ML.
6. **Keras** [31]: biblioteca utilizada em redes neurais, com código aberto e escrita em *Python*. É capaz de rodar em sobre *TensorFlow*, Microsoft *Cognitive Toolkit*, *R*, *Theano* ou *PlaidML*. Esse *software* contém várias funções para construir partes importantes de redes neurais, como camadas, funções de perda, funções de ativação, otimizadores, entre outras. Além das redes neurais padrão, o *Keras* tem suporte para redes neurais convolucionais e recorrentes. Suporta camadas de *drop-out*, normalização em *batch* e *pooling*. Também

pode ser executado em múltiplas CPUs e GPUs. Devido à sua facilidade de uso, essa biblioteca poderá ser utilizada para implementação de redes neurais por oferecer APIs simples e consistentes.

7. **Tensorboard** [32]: ferramenta que faz parte da biblioteca *Tensorflow*, oferecendo a visualização e as ferramentas necessárias para a realização de experimentos e depuração de modelos de ML.
8. **SciPy** [33]: biblioteca de código aberto feita para a linguagem *Python* que fornece funções para álgebra linear, transformadas de *Fourier*, projeto de filtros e geração de números aleatórios.
9. **Scikit-Learn** [34]: biblioteca de ML de código aberto para a linguagem de programação *Python*. Ela inclui vários algoritmos de classificação, regressão e agrupamento incluindo florestas aleatórias, *Support Vector Machines* (SVM), *k-means* e árvores de decisão. Foi projetada para interagir com as bibliotecas *Python NumPy*, *SciPy* e *Matplotlib*.
10. **Seaborn** [35]: biblioteca de código aberto para visualização de dados *Python* baseada na biblioteca *Matplotlib*. Ela fornece uma interface de alto nível para desenhar gráficos estatísticos atraentes e informativos.
11. **Anaconda** [36]: distribuição das linguagens de programação *Python* e *R* para computação científica (ciência de dados, aplicativos de aprendizado de máquina, processamento de dados em larga escala, análise preditiva e outros), que visa simplificar o gerenciamento e a implantação de pacotes. A distribuição inclui pacotes de ciência de dados adequados para *Windows*, *Linux* e *macOS*. Portanto, o *Anaconda* é um gerenciador de bibliotecas e já vem com alguns pacotes úteis por padrão, como o *Jupyter Notebook*, *Spyder* e outros comumente usados para computação científica e ciência de dados.
12. **Spyder** [37]: ambiente de desenvolvimento integrado de código aberto para programação científica na linguagem *Python*. Assim, o *Spyder* é uma interface gráfica, semelhante ao *Matlab*, que permite a utilização de *Python* num ambiente interativo, facilitando a edição de *scripts*, teste, *debugging* e visualização gráfica. Essa ferramenta poderá ser útil para a elaboração e edição de códigos na linguagem *Python*.
13. **Pandas** [38]: biblioteca de *software* de código aberto criada para a linguagem *Python* para manipulação e análise de dados. Em particular, ela oferece estruturas e operações para manipular tabelas numéricas e séries temporais. *Pandas* é usada principalmente em ML, pois permite a importação de diferentes formatos de arquivo, como csv e excel, para a leitura em *dataframes*. Além disso, permite diversas operações de álgebra relacional, como projeção, junção e concatenação. Também executa funções de limpeza, como por exemplo o preenchimento, substituição ou inserção de valores nulos (*null*). Dessa forma, poderá ser usado para facilitar a manipulação, importação e a análise de dados para aplicação em redes neurais profundas.
14. **ModulationPy** [39]: módulo *Python* que permite substituir funções e objetos relacionados a modulações lineares digitais implementadas no *Matlab/Octave* relacionadas à banda base (M-PSK e M-QAM). Esse projeto é inspirado no projeto de código aberto *CommPy*. A biblioteca *ModulationPy* poderá ser utilizada em projetos de comunicação sem fio e sobre fibra para simular as modulações digitais lineares em banda base.

15. **Sionna** [40]: biblioteca de código aberto baseada no *TensorFlow* para simular a camada física de sistemas de comunicação sem fio. A modelagem rápida de sistemas de comunicação complexos de ponta a ponta é realizada de maneira simples através de uma API de alto nível. O *Sionna* foi desenvolvido e é continuamente estendido para impulsionar pesquisas em redes de comunicações 5G e 6G. Ele suporta configurações de simulação de nível de *link* MU-MIMO com códigos compatíveis com as redes 5G, como os códigos *Low-Density Parity Check* (LDPC) e Polar, os modelos de canal 3GPP, *Orthogonal Frequency Division Multiplexing* (OFDM), estimativa de canal, equalização e mapeamento suave. Portanto, o *Sionna* poderá facilitar a modelagem e adaptação de sistemas de comunicação sem fio baseadas nos padrões 5G a partes da pesquisa, além de servir como referência para comparar os algoritmos desenvolvidos com o estado da arte.
16. **OptiSystem** [4]: ferramenta que possibilita planejar, testar e simular quase todos os tipos de enlaces ópticos na camada de transporte dos sistemas de comunicações móveis. O *software* possui papel fundamental no Projeto Brasil 6G, especialmente relacionado ao uso de AI em redes 6G. Nesse caso, o *software* pode ser empregado para simular sistemas RoF que irão compor a camada de transporte das futuras redes 6G. Em tais sistemas, efeitos lineares a não-lineares precisam ser compensados, uma vez que afetam o desempenho da rede e, conseqüentemente, a qualidade de serviço para o usuário final. Os impactos de tais efeitos em sistemas RoF podem ser simulados utilizando o *Optisystem*, habilitando o desenvolvimento de modelos de pre-distorção e equalização, baseados em ML, que poderão ser aplicados nas futuras redes 6G.
17. **OpenCV** [41]: biblioteca de código aberto multiplataforma, totalmente livre para o uso acadêmico e comercial, para o desenvolvimento de aplicações de visão computacional, processamento de imagens e ML.
18. **CUDA** [42]: API destinada a computação paralela, *General Purpose Graphics Processing Unit* (GPGPU) e computação heterogênea, criada pela Nvidia. Esse *software* é destinado a placas gráficas que suportem a API, normalmente placas gráficas com *chipset* da Nvidia. A plataforma *CUDA* dá acesso ao conjunto de instruções virtuais da GPU e a elementos de computação paralela para a execução de núcleos de computação.
19. **Google Colaboratory Pro+** [43]: serviço de nuvem gratuito hospedado pelo *Google* para incentivar a pesquisa de ML e AI. É uma ferramenta colaborativa que permite que se misture código fonte em *Python* e texto rico (geralmente em *markdown*) com imagens e o resultado desse código.

6 *Softwares* para os Transceptores da Rede de Longo Alcance

Daniely Gomes Silva, Juliano Silveira Ferreira
 daniely@inatel.br, silveira@inatel.br

Nesta seção será apresentada uma breve descrição dos *softwares* empregados na implementação do transceptor da rede de longo alcance proposto no Projeto Brasil 6G. O *Software Defined Radio* (SDR) foi adotado como paradigma para a implementação desse transceptor. Os componentes principais de *software* empregados em sua implementação são:

1. **GNU Radio** [44]: *framework open source* para implementação e testes das etapas de processamento digital de sinais e que é executado sobre o sistema operacional *Ubuntu/Linux*. As etapas de processamento de dados são implementadas na forma de blocos de processamento que são então importados e combinados em um projeto do *GNU Radio*. As linguagens de programação principais empregadas para a implementação dos blocos de processamento são *C++* e *Python*.
2. **UHD** [45]: *USRP Hardware Driver* (UHD) é uma interface executada em *General Purpose Processors* (GPP) e utilizada para conectar, enviar e receber dados entre o *software GNU Radio* e o SDR empregado na implementação do transceptor 6G.

7 Software para Implementação de Pilhas de Protocolos Utilizados na Estação Radiobase 5G

Daniely Gomes Silva
daniely@inatel.br

Nesta seção será apresentada uma breve descrição do *software* empregado na implementação das pilhas de protocolos previstas no desenvolvimento de uma estação radiobase 5G. A evolução dessa estação radiobase, devido à adição de novas funcionalidades, resultará na estação radiobase 6G proposta pelo Projeto Brasil 6G.

1. **Amarisoft** [46]: a pilha de protocolos da plataforma da *Amarisoft* atua como um *enhanced Node B* (eNodeB), *next generation Node B* (gNodeB), *Evolved Packet Core* (EPC) e núcleos Quarta Geração (4G) e 5G compatíveis com a regulamentação proposta pelo 3GPP, permitindo testes funcionais e de desempenho de dispositivos *New Radio* (NR), *Long Term Evolution* (LTE), *Long Term Evolution Advanced* (LTE-A), *Long Term Evolution for Machines* (LTE-M) e *Narrowband IoT* (NB-IoT). A plataforma possui um servidor *IP multimedia subsystem* (IMS) integrado e também um *gateway Evolved Multimedia Broadcast Multicast Services* (eMBMS) para testes de *voice over LTE* (VoLTE) e *multicast*. O *software* é executado em computadores com processador x86 acoplados a um *front-end* de rádio e é compatível com *AMARI Peripheral Component Interconnect Express* (PCIe) SDR 2x2 e outros *front-ends*.

8 Conclusão

A criação de uma plataforma apta para demonstrar as principais funcionalidades da Rede 6G, que também permita a avaliação de desempenho dos algoritmos propostos pelos pesquisadores do projeto Brasil 6G em condições reais de operação, requer uma grande quantidade de *softwares* e aplicativos . Esse relatório apresentou os principais programas que serão necessários para a construção dessa plataforma, destacando como cada um será empregado, passando desde o suporte à aplicações de AI até o uso de ferramentas de programação para a implementação do transceptor empregando SDR.

Referências

- [1] A. Oliveira Jr., C. F. M. Silva, F. d. O. Silva, K. V. Cardoso, L. L. Mendes, R. S. Silva, e S. L. Corrêa, “Projeto BRASIL6G - Casos de Uso e Requisitos para as Redes 6G,” Tech. Rep., Aug. 2021.
- [2] A. Alberti et al., “Arquiteturas para a Rede 6G,” Projeto Brasil 6G, Report, 2021.
- [3] “Matplotlib: Visualization with Python,” <https://matplotlib.org/>, 2022, [Online; acessado em 13 de junho de 2022].
- [4] “OptiSystem Overview,” <https://optiwave.com/optisystem-overview/>, 2022, [Online; acessado em 13 de junho de 2022].
- [5] “ThingsBoard Open-source IoT Platform,” <https://thingsboard.io/>, 2022, [Online; acessado em 13 de junho de 2022].
- [6] “The Augmented Reality Edge Network Architecture,” <https://arenaxr.org/>, 2022, [Online; acessado em 13 de junho de 2022].
- [7] “Digitanimal - Os melhores GPS para animais,” <https://osm.etsi.org/>, 2022, [Online; acessado em 20 de junho de 2022].
- [8] “What is OSM?” <https://digitanimal.com/>, 2022, [Online; acessado em 15 de junho de 2022].
- [9] “ONF ONOS,” <https://opennetworking.org/onos/>, 2022, [Online; acessado em 20 de junho de 2022].
- [10] “An OpenInfra Project OpenStack,” <https://www.openstack.org/>, 2022, [Online; acessado em 20 de junho de 2022].
- [11] “Eclipse fog05, The End-to-End Compute, Storage and Networking Virtualisation solution,” <https://fog05.io/>, 2022, [Online; acessado em 20 de junho de 2022].
- [12] “The Linux Foundation,” <https://linuxfoundation.org/>, 2022, [Online; acessado em 20 de junho de 2022].
- [13] “Nginx,” <https://www.nginx.com/>, 2022, [Online; acessado em 20 de junho de 2022].
- [14] “IOTA,” <https://www.iota.org/>, 2022, [Online; acessado em 20 de junho de 2022].
- [15] “Hyperledger,” <https://www.hyperledger.org/>, 2022, [Online; acessado em 20 de junho de 2022].
- [16] “A crypto wallet & gateway to blockchain apps,” <https://metamask.io/>, 2022, [Online; acessado em 20 de junho de 2022].
- [17] “<https://github.com/contiki-os/contiki/wiki/An-Introduction-to-Cooja>,” <https://metamask.io/>, 2022, [Online; acessado em 20 de junho de 2022].
- [18] “What is free5GC?” <https://www.free5gc.org/>, 2022, [Online; acessado em 15 de junho de 2022].

- [19] “OpenAirInterface - The fastest Growing Community and Software Assets in 5G Wireless,” <https://openairinterface.org/>, 2022, [*Online*; acessado em 15 de junho de 2022].
- [20] “Kubernetes - Production-Grade Container Orchestration,” <https://kubernetes.io/>, 2022, [*Online*; acessado em 15 de junho de 2022].
- [21] “Helm - The package manager for Kubernetes,” <https://helm.sh/>, 2022, [*Online*; acessado em 15 de junho de 2022].
- [22] “Red Hat Ansible Automation Platform on Microsoft Azure,” <https://www.ansible.com/>, 2022, [*Online*; acessado em 15 de junho de 2022].
- [23] “Grafana - Operational dashboards for your data here, there, or anywhere,” <https://grafana.com/>, 2022, [*Online*; acessado em 15 de junho de 2022].
- [24] “Prometheus - For Beginners, For Investors and For Passive Income Seekers,” <https://prometheusintelligence.com/>, 2022, [*Online*; acessado em 15 de junho de 2022].
- [25] “Developers Love Docker, Businesses Trust It,” <https://www.docker.com/>, 2022, [*Online*; acessado em 20 de junho de 2022].
- [26] “Kubernetes Multus-CNI,” <https://github.com/k8snetworkplumbingwg/multus-cni/wiki/multus>, 2022, [*Online*; acessado em 20 de junho de 2022].
- [27] “Matlab - Math, Graphics, Programming,” <https://www.mathworks.com/>, 2022, [*Online*; acessado em 20 de junho de 2022].
- [28] “Jupyter - Free software, open standards, and web services for interactive computing across all programming languages,” <https://jupyter.org/>, 2022, [*Online*; acessado em 20 de junho de 2022].
- [29] “NumPy - The fundamental package for scientific computing with Python,” <https://numpy.org/>, 2022, [*Online*; acessado em 20 de junho de 2022].
- [30] “Tensorflow - Uma plataforma completa de código aberto para machine learning,” <https://www.tensorflow.org/>, 2022, [*Online*; acessado em 20 de junho de 2022].
- [31] “Keras - Simple, Flexible, Powerful,” <https://keras.io/>, 2022, [*Online*; acessado em 20 de junho de 2022].
- [32] “TensorBoard: kit de ferramentas de visualização do TensorFlow,” <https://www.tensorflow.org/tensorboard?hl=pt-br>, 2022, [*Online*; acessado em 20 de junho de 2022].
- [33] “SciPy - Fundamental algorithms for scientific computing in Python,” <https://scipy.org/>, 2022, [*Online*; acessado em 20 de junho de 2022].
- [34] “Scikit-Learn - Machine Learning in Python,” <https://scikit-learn.org/stable/>, 2022, [*Online*; acessado em 20 de junho de 2022].
- [35] “Seaborn - Statistical Data Disualization,” <https://seaborn.pydata.org/>, 2022, [*Online*; acessado em 20 de junho de 2022].

- [36] “Anaconda - The World’s Most Popular Data Science Platform,” <https://www.anaconda.com/>, 2022, [*Online*; acessado em 23 de junho de 2022].
- [37] “Spyder - The Scientific Python Development Environment,” <https://www.spyder-ide.org/>, 2022, [*Online*; acessado em 23 de junho de 2022].
- [38] “Pandas - Python Data Analysis Library,” <https://pandas.pydata.org/>, 2022, [*Online*; acessado em 23 de junho de 2022].
- [39] “ModulationPy - Linear Digital Modems,” <https://pypi.org/project/ModulationPy/>, 2022, [*Online*; acessado em 23 de junho de 2022].
- [40] “Sionna - An Open-Source Library for Next-Generation Physical Layer Research,” <https://github.com/NVlabs/sionna>, 2022, [*Online*; acessado em 23 de junho de 2022].
- [41] “OpenCV AI Courses,” <https://opencv.org/>, 2022, [*Online*; acessado em 23 de junho de 2022].
- [42] “CUDA - NVIDIA Developer,” <https://developer.nvidia.com/cuda-zone>, 2022, [*Online*; acessado em 23 de junho de 2022].
- [43] “Google Colab - Choose the Colab plan that’s right for you,” <https://colab.research.google.com/signup>, 2022, [*Online*; acessado em 23 de junho de 2022].
- [44] “GNU Radio: The Free and Open Software Radio Ecosystem,” <https://www.gnuradio.org/>, 2022, [*Online*; acessado em 12 de julho de 2022].
- [45] “GNU Radio Manual and C++ API Reference - UHD Interface,” https://www.gnuradio.org/doc/doxygen/page_uhd.html, 2022, [*Online*; acessado em 12 de julho de 2022].
- [46] “Amarisoft - Your partner from the lab to the field,” <https://www.amarisoft.com/>, 2022, [*Online*; acessado em 20 de junho de 2022].