

Pedro Henrique Oliveira Toscano Ximenes

**RELATÓRIO DE ESTÁGIO INTEGRADO: BIT
SERVICES INOVAÇÃO E TECNOLOGIA -
LABORATÓRIO DE SISTEMAS
EMBARCADOS E COMPUTAÇÃO
PERVASIVA (EMBEDDED)**

Campina Grande, Brasil

19 de março de 2021

Pedro Henrique Oliveira Toscano Ximenes

**RELATÓRIO DE ESTÁGIO INTEGRADO: BIT
SERVICES INOVAÇÃO E TECNOLOGIA -
LABORATÓRIO DE SISTEMAS EMBARCADOS E
COMPUTAÇÃO PERVASIVA (EMBEDDED)**

Relatório de estágio integrado submetido à
Unidade Acadêmica de Engenharia Elétrica
da Universidade Federal de Campina Grande
como parte dos requisitos necessários para a
obtenção do grau de Bacharel em Ciências no
Domínio da Engenharia Elétrica.

Universidade Federal de Campina Grande - UFCG

Centro de Engenharia Elétrica e Informática - CEEI

Departamento de Engenharia Elétrica - DEE

Orientador: Saulo Oliveira Dornellas Luiz, D. Sc.

Campina Grande, Brasil

19 de março de 2021

Pedro Henrique Oliveira Toscano Ximenes

**RELATÓRIO DE ESTÁGIO INTEGRADO: BIT
SERVICES INOVAÇÃO E TECNOLOGIA -
LABORATÓRIO DE SISTEMAS EMBARCADOS E
COMPUTAÇÃO PERVASIVA (EMBEDDED)**

Relatório de estágio integrado submetido à Unidade Acadêmica de Engenharia Elétrica da Universidade Federal de Campina Grande como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciências no Domínio da Engenharia Elétrica.

Trabalho aprovado em: / /

Saulo Oliveira Dornellas Luiz, D. Sc.
Orientador

Convidado

Campina Grande, Brasil
19 de março de 2021

*Dedico este trabalho aos meus pais, Rovênia Maria de Oliveira Toscano Ximenes e
Carlos Alberto Toscano Ximenes.*

Agradecimentos

A jornada desde a procura de estágios até sua finalização não foi fácil. Inúmeras coisas boas aconteceram durante esse período, mas também ocorreram vários problemas no meio do caminho e várias pessoas me apoiaram para que tudo ocorresse bem no final, por isso gostaria de agradecê-las.

Primeiro, à minha família. A meus pais por sempre me apoiarem, às minhas irmãs e meus cunhados por sempre servirem de inspiração para mim em todos os aspectos e aos meus primos Filipe e Sarah, pela hospitalidade e toda ajuda que me deram.

Agradeço a toda B2W Digital por ajudarem tanto no meu desenvolvimento. Ao pessoal do RH pelo trabalho, paciência e ajuda nas questões burocráticas (que foram muitas) e ao pessoal com que trabalhei diretamente durante esse tempo. Foram várias equipes e várias pessoas que gastaram com frequência seu tempo para me orientar, ajudar e ensinar. A isso sou muito grato.

Aos meus amigos que me apoiaram muito nessa etapa da minha vida, em especial a meu amigo Luan que me ajudou muito nessa etapa sempre que precisei.

Aos professores Saulo pela ajuda, orientação e paciência e Gutemberg pela enorme ajuda que me deu em todo o processo. Sou muito grato por isso.

Por fim, à minha namorada, Ana Paula, por estar comigo em todos os momentos, por sempre me apoiar e simplesmente por ser a pessoa que ela é. Muito obrigado.

Lista de ilustrações

Figura 1 – Estrutura hierárquia da companhia.	3
Figura 2 – Relacionamento Tópico-Assinatura no Pub/Sub.	5
Figura 3 – Ambiente de desenvolvimento do Power BI.	6
Figura 4 – Fluxo final da atividade de automação.	8
Figura 5 – Fluxo final da atividade.	11
Figura 6 – Fachada do Laboratório.	13
Figura 7 – Princípio básico do algoritmo de Aprendizado por Reforço.	15
Figura 8 – Posição do quadrotor nos eixos x,y, z.	18
Figura 9 – Velocidade do quadrotor nos eixos x,y, z.	19
Figura 10 – Rotações por segundo estabelecidas pelo agente em cada rotor.	19

Lista de abreviaturas e siglas

GCP	<i>Google Cloud Platform</i>
AWS	<i>Amazon Web Services</i>
BQ	<i>BigQuery</i>
BI	<i>Business Intelligence</i>
HTTP	<i>Hypertext Transfer Protocol</i>
SQL	<i>Structured Query Language</i>
ML	<i>Machine Learning</i>
DDPG	<i>Deep Deterministic Policy Gradients</i>

Sumário

1	INTRODUÇÃO	1
2	PRIMEIRA PARTE - B2W DIGITAL	2
2.1	BIT Services Inovação e Tecnologia	2
2.2	Ferramentas Utilizadas	3
2.2.1	Google Cloud Platform	3
2.2.2	Microsoft Power BI	5
2.2.3	<i>Frameworks</i> Utilizados	6
2.3	Atividades Desenvolvidas	7
2.3.1	Treinamentos	7
2.3.2	1ª Equipe	8
2.3.3	2ª Equipe	9
2.3.4	3ª Equipe	10
2.3.5	4ª Equipe	10
2.3.6	5ª Equipe	11
3	SEGUNDA PARTE - EMBEDDED	13
3.1	Laboratório de Sistemas Embarcados e Computação Pervasiva - Embedded	13
3.2	Fundamentação Teórica	14
3.2.1	Aprendizado por Reforço	14
3.3	Atividades Desenvolvidas	15
4	CONSIDERAÇÕES FINAIS	21
	REFERÊNCIAS	22

1 Introdução

Neste relatório são descritas as atividades realizadas durante a execução do Estágio Integrado do aluno Pedro Henrique Oliveira Toscano Ximenes, realizado no período de 27 de Maio de 2020 a 07 de Dezembro de 2020. Diferentemente do que normalmente é visto no curso de graduação em Engenharia Elétrica na Universidade Federal de Campina Grande, em consequência de adversidades originadas pela pandemia global de Covid-19, a total integralização do estágio se deu em duas partes, como descrito a seguir.

A primeira etapa foi realizada no período entre 27 de Maio de 2020 e 10 de Setembro de 2020 (totalizando 462 horas) na empresa Bit Services Inovação e Tecnologia Ltda., divisão da área de tecnologia e inovação da empresa B2W Digital. Essa etapa foi executada totalmente em *Home Office*, ou seja, as atividades não foram executadas nos domínios físicos da empresa.

Para integralização das 198 horas necessárias para conclusão das 660 horas totais, a segunda parte do estágio foi realizada no Laboratório de Sistemas Embarcados e Computação Pervasiva - Embedded da Universidade Federal de Campina Grande, no período de 15 de Outubro de 2020 a 09 de Dezembro de 2020.

Este relatório contém, portanto, duas partes, de forma a tratar separadamente os tópicos: “Descrição do Local de Estágio”, “Ferramentas Utilizadas”, "Fundamentação Teórica" e “Atividades Desenvolvidas” com foco nas individualidades de cada etapa de estágio.

2 Primeira Parte - B2W Digital

Esse capítulo é dedicado à demonstração das particularidades da primeira experiência do Estágio Integrado, bem como à explicação das tecnologias utilizadas no decorrer do processo.

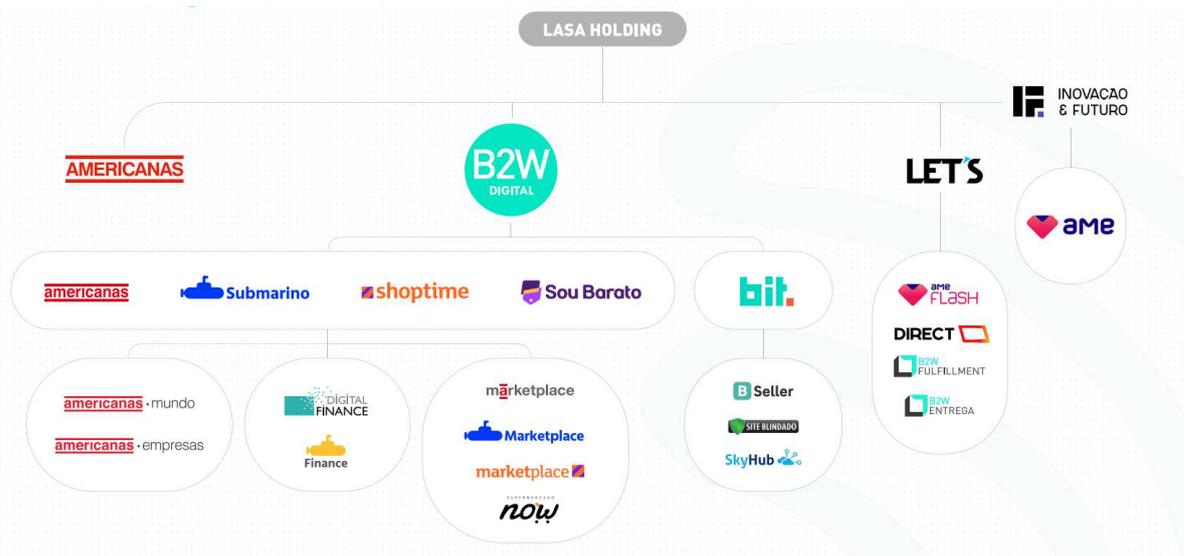
2.1 BIT Services Inovação e Tecnologia

A BIT Services Inovação e Tecnologia é uma empresa responsável, como o próprio nome sugere, pelas áreas de Inovação e Tecnologia e que faz parte do grupo B2W Digital, empresa que atua no ramo de *e-commerce* no Brasil sendo mais conhecida por ser a dona das marcas Americanas.com, Submarino, Shoptime e SouBarato.

Fundada em 2006 com a fusão da Americanas.com e Submarino, a B2W Digital vem, desde então, sendo uma das mais influentes empresas no setor de comércio eletrônico em todo o território nacional, sendo considerada a maior companhia digital da América Latina. Com sede no Rio de Janeiro, RJ, a empresa possui 21 centros de distribuição espalhadas por todo país e, no caso do BIT especificamente, possui unidades no Rio de Janeiro, RJ, em São Paulo, SP, e no Recife, PE.

Da mesma forma que a BIT Services é uma empresa pertencente ao grupo B2W Digital, a própria B2W, por sua vez, tem cerca de 60% do seu capital social possuído pelo grupo LASA Holding (Lojas Americanas S.A) e, dessa forma, possuindo controle acionário. Na Figura 1 é apresentada a estrutura da empresa de forma visual, assim como outras empresas associadas.

Figura 1 – Estrutura hierárquica da companhia.



Fonte: B2W Digital.

2.2 Ferramentas Utilizadas

Primeiramente, será apresentado um resumo sobre ferramentas e sistemas *third-party* que foram bastante utilizados nas atividades desempenhadas. São eles, principalmente, o *Google Cloud Platform*¹ (GCP) e alguns de seus recursos (*BigQuery*², *Pub/Sub*³, *Dataflow*⁴, *Cloud Functions*⁵ e *Cloud Storage*) e o Microsoft Power BI⁶, além de alguns *frameworks* específicos.

2.2.1 Google Cloud Platform

Trata-se de uma plataforma de computação em nuvem oferecida pelo Google. Nele, diversos recursos em nuvem estão disponíveis em diversas áreas como Big Data, *networking*, infraestrutura, armazenamento e aprendizado de máquina. Dentre os recursos disponíveis, os utilizados durante a execução do estágio foram os seguintes:

¹ Google Cloud Overview. Acessado em: 19 dez. 2020. Disponível em: <<https://cloud.google.com/docs/overview>>.

² What is BigQuery? | Google Cloud. Acessado em: 19 dez. 2020. Disponível em: <<https://cloud.google.com/bigquery/docs/introduction>>.

³ What Is Pub/Sub? | Cloud Pub/Sub Documentation | Google Cloud. Acessado em: 19 dez. 2020. Disponível em: <<https://cloud.google.com/pubsub/docs/overview>>.

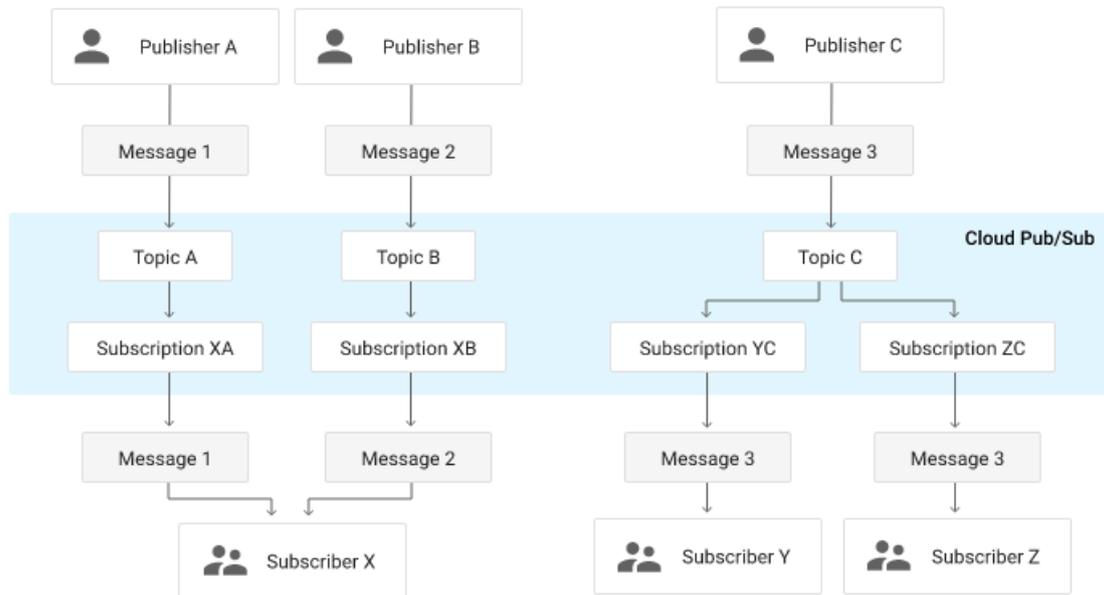
⁴ Dataflow documentation | Google Cloud | Google Cloud. Acessado em: 20 dez. 2020. Disponível em: <<https://cloud.google.com/dataflow/docs>>.

⁵ Cloud Functions documentation | Cloud Functions Documentation. Acessado em: 20 dez. 2020. Disponível em: <<https://cloud.google.com/functions/docs>>.

⁶ What is Power BI? - Power BI. Acessado em: 20 dez. 2020. Disponível em: <<https://docs.microsoft.com/en-us/power-bi/fundamentals/power-bi-overview>>.

- **BigQuery:** o *Google Cloud Platform* oferece a infraestrutura necessária para armazenamento e consulta de massivos blocos de dados, deixando o usuário livre de qualquer preocupação com infraestrutura, assim como boa parte dos recursos no GCP. É, portanto, o ambiente em que os dados são armazenados em tabelas relacionais e consultados via SQL.
- **Google Cloud Storage:** assim como o *BigQuery*, o *Storage* é um recurso voltado para armazenamento. Porém, diferente do anterior que armazena dados em forma de tabelas, o *Storage* armazena arquivos.
- **Pub/Sub:** serviço de mensageria assíncrona que oferece armazenamento durável de mensagens e serviço de entrega em tempo real. Funciona utilizando o conceito de tópicos e assinaturas, na qual os tópicos são responsáveis por reter mensagens enviadas por um *publisher* e as assinaturas representam o serviço de *streaming* da mensagem entre um único tópico e as aplicações “assinantes”, que receberão a mensagem. Na Figura 2 é representado o relacionamento tópico-assinatura.
- **Dataflow:** recurso que oferece uma solução unificada para processamento de dados em lote (*batch*) e *streaming*.
- **Cloud Functions:** esse recurso se enquadra na categoria FaaS (sigla para *Function as a Service*) e oferece toda a infraestrutura operacional necessária para executar códigos escritos em diferentes ambientes, como Python 3.8, Java 11, Go 1.13 e Node.js 12 (no momento da escrita deste relatório). Além disso, as *functions* podem ser gatilhadas por algum evento específico. Por exemplo, uma das possibilidades de gatilho é o *Cloud Storage*, de modo a executar o código contido na *Cloud Function* sempre que algum arquivo é inserido em determinada pasta.

Figura 2 – Relacionamento Tópico-Assinatura no Pub/Sub.

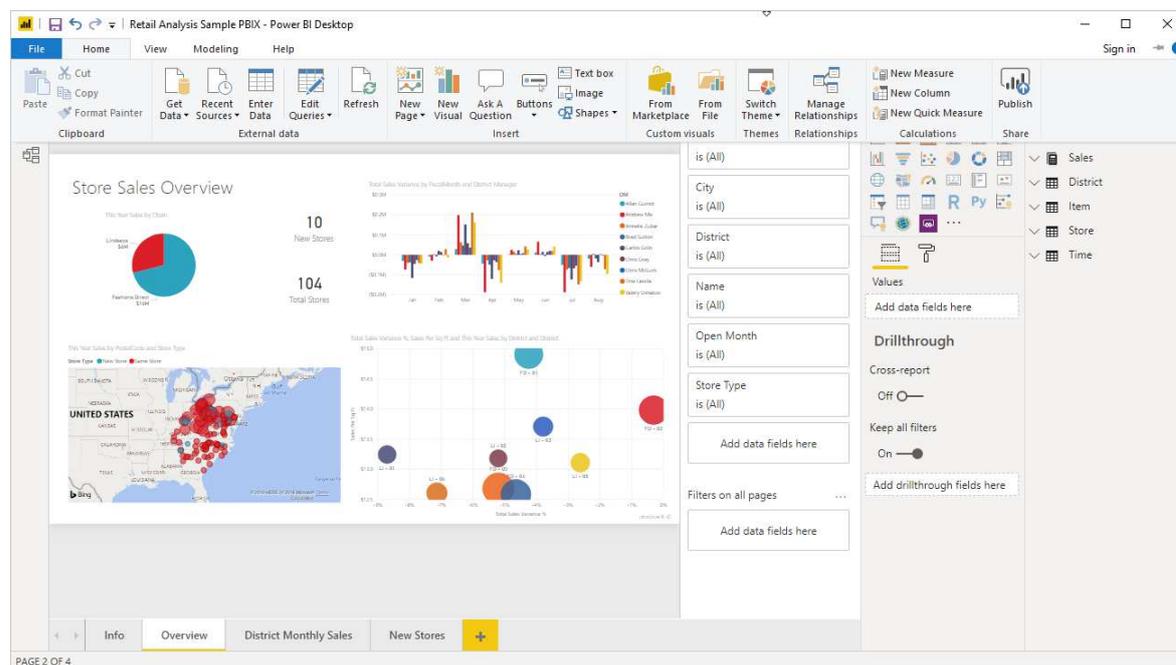


Fonte: cloud.google.com/pubsub/docs/overview

2.2.2 Microsoft Power BI

Como o próprio nome sugere, o serviço da Microsoft oferece recursos voltados para a área de *Business Intelligence* (BI). O *Power BI* é, atualmente, a ferramenta mais utilizada no mercado para o desenvolvimento de *dashboards* (visualizações interativas) e relatórios de BI, que são desenvolvidos via interface simples. Além disso, os dados utilizados nos relatórios podem ser carregados de forma *offline* ou até mesmo em tempo real. Na Figura 3 é apresentada a interface do Power BI, com algumas de suas funcionalidades e um *dashboard* em construção.

Figura 3 – Ambiente de desenvolvimento do Power BI.



Fonte: <docs.microsoft.com/pt-br/power-bi/fundamentals/desktop-what-is-desktop>. Acessado em: 10 mar. 2021.

2.2.3 Frameworks Utilizados

Frameworks, de forma simples, são conjuntos de bibliotecas cuja funcionalidade é apresentar uma solução simples, com funcionalidade genérica e que permitem reusabilidade de código, para uma determinada linguagem de programação. No decorrer do estágio, foram utilizados três *frameworks* que serão apresentados a seguir (em parênteses, está destacada a linguagem de programação à qual pertence):

- *Flask* (Python)⁷
- *Spring Boot* (Java)⁸
- *Django* (Python)⁹

Todos os três são *frameworks* voltados para desenvolvimento Web, ou seja, possuem soluções práticas para criação de rotas, requisições e diversas funcionalidades exigidas nesse tipo de desenvolvimento.

⁷ Welcome to Flask. Acessado em: 20 dez. 2020. Disponível em: <<https://flask.palletsprojects.com/en/1.1.x/>>

⁸ Spring Boot. Acessado em: 20 dez. 2020. Disponível em: <<https://spring.io/projects/spring-boot>>

⁹ Django Overview. Acessado em: 20 dez. 2020. Disponível em: <<https://www.djangoproject.com/start/overview/>>

2.3 Atividades Desenvolvidas

O principal objetivo do estágio foi de introduzir ao estagiário noções importantes, tanto técnicas quanto comportamentais, para o desenvolvimento pessoal e profissional. Para isso, o estagiário participou de uma série de treinamentos e, em sequência, foram escolhidas cinco áreas da companhia, uma por vez, para a realização das atividades, permanecendo por cerca de três semanas em cada e tendo a oportunidade de lidar com diferentes desafios, pessoas e aprendizados, que serão melhor detalhados em seguida.

2.3.1 Treinamentos

A primeira etapa realizada do estágio foi a participação em uma série de treinamentos, cujo objetivo foi a introdução do estagiário em diversos tópicos da tecnologia que são muito utilizados atualmente. Os assuntos tratados foram os seguintes:

- Linux Básico
- Introdução ao Java
- Introdução ao Python
- Controle de Versão (Git)
- *Amazon Web Services* (AWS)
- Introdução ao Javascript
- Conceitos de Estruturas de dados
- *Machine Learning*
- *Business Intelligence* (BI)
- Banco de Dados
- Introdução ao Docker
- Metodologia Scrum

Todos os treinamentos foram ministrados por profissionais da empresa bastante capacitados no conteúdo específico oferecido. Muitos desses conteúdos foram utilizados de forma prática no restante do estágio, como Python, Java, Linux, Git e Bancos de Dados.

2.3.2 1ª Equipe

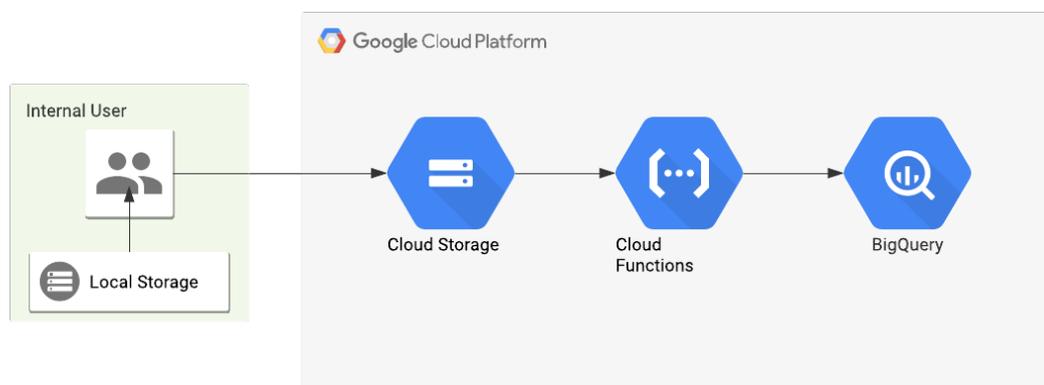
No comércio eletrônico moderno, é muito comum que as empresas possuam um sistema conhecido como 3P (do inglês *third-party relationship*) ou *marketplace*, ou seja, é possível que parceiros utilizem a plataforma da companhia, o Submarino.com por exemplo, para realizar suas vendas. As pessoas que usufruem dessa modalidade para realização de vendas são conhecidas internamente como *sellers*.

A primeira equipe visitada tem por objetivo realizar análises referentes a essa modalidade 3P, gerando valiosos dados para a companhia. Nela, foi possível ter um primeiro contato com o Google Cloud Platform, ferramenta de computação em nuvem do Google, que seria muito utilizada também nas outras áreas do estágio, além de explorar também a programação na linguagem Python.

A principal atividade realizada, de forma geral, consistiu na automação de um processo que anteriormente era feito de forma manual e consistia em diversas etapas repetitivas e geravam desgaste e perda de tempo para quem fosse realizá-las. De tempos em tempos, uma outra área da companhia envia uma tabela em formato .csv com informações que devem ser incluídos nos bancos de dados da equipe. Para isso, uma série de passos eram manualmente realizados.

A atividade consistiu em desenvolver um *script* em Python que fosse capaz de importar as informações do arquivo .csv, tratá-las para que ficassem no formato desejado e exportá-las para uma tabela no BigQuery. O *script* foi então incorporado a uma Cloud Function que é gatilhada sempre que o arquivo que contém as informações é colocado em uma pasta específica (*bucket*) no Google Cloud Storage. Na Figura 4 é representado o fluxo final desenvolvido. Os maiores desafios dessa atividade consistiram em desenvolver o *script* em Python e aprender a utilizar os recursos citados do Google Cloud Platform.

Figura 4 – Fluxo final da atividade de automação.



Fonte: Autoria própria.

2.3.3 2ª Equipe

Por se tratar de uma grande companhia com diversas áreas focadas nos mais diversos assuntos, nem todas as equipes estão diretamente relacionadas com tecnologia da informação, como equipes de marketing ou de estoque. Contudo, muitas delas precisam ter acesso às informações que podem ser difíceis de conseguir, seja por falta do conhecimento técnico específico ou por não ser uma prioridade em sua alocação de esforços. Por isso, essa equipe existe com finalidade de realizar consultas, criar *dashboards* ou desenvolver atividades que sejam solicitadas por essas diversas outras equipes da companhia.

Uma das atribuições da equipe é criar consultas programadas (ou seja, *queries* que são realizadas automaticamente de tempos em tempos), chamadas de *jobs*, para diversas outras áreas da companhia. As consultas realizadas no BigQuery (assim como todos os outros recursos do Google Cloud Platform) não são gratuitas e, por isso, o custo para a companhia gerado por essa equipe pode ser muito alto. Pensando nisso, a atividade realizada consistiu em criar um método para rastrear as informações de cada *job* (equipe e departamento que solicitou essa consulta) e, assim, poder juntar com outras informações já existentes (como gasto total da consulta em determinado período de tempo) e montar um relatório interativo do tipo *dashboard* utilizando o Microsoft Power BI.

A primeira ideia pensada foi a de utilizar os rótulos (*labels*), ferramenta disponibilizada para a maioria dos recursos no GCP, para guardar essas informações. Contudo, após uma série de testes, não foi possível criar consultas programadas com rótulos. Para tentar entender o motivo, a equipe de suporte do Google Cloud Platform foi acionada e, após realizarem testes, perceberam que de fato não existe suporte para criação de consultas programadas com rótulos e afirmaram que iriam trabalhar para adicionar essa funcionalidade.

Considerando a impossibilidade de implementar a primeira ideia, foi proposta uma segunda ideia: adicionar um comentário em cada uma das consultas, informando o cliente e projeto que requisitou essa consulta, além da tabela do BigQuery na qual o resultado é guardado. Posteriormente, foi desenvolvido um *script* em JavaScript responsável por ler esse comentário em cada *job* e separar as informações de modo que pudessem ser salvas separadamente em uma coluna de uma tabela. Assim, esse *script* foi incorporado em uma consulta programada que, uma vez por dia, monitora as atividades das outras consultas existentes, obtendo informações de custo. Agora, é possível discriminar esse custo de acordo com o cliente. Esses dados são atualizados em uma tabela do BigQuery. A etapa final consistiu em criar um *dashboard* utilizando o Microsoft Power BI para melhor visualização dos dados.

2.3.4 3ª Equipe

De forma simples, essa equipe pode ser considerada algo como o RH Digital, não por realizar tarefas típicas de Recursos Humanos, mas sim por gerar informações valiosas para a equipe de RH com intuito de melhorar diversos aspectos humanos da empresa, como o processo de contratação ou a experiência interna dos colaboradores.

As atividades desenvolvidas nessa equipe foram diretamente relacionadas a um projeto iniciado pelos estagiários anteriores que passaram pela equipe. O projeto consiste em desenvolver uma plataforma (por meio de uma API Rest) para a qual qualquer funcionário da empresa tenha acesso à árvore de cargos da companhia. Nessa árvore, é possível verificar qualquer cargo, quais são as competências básicas esperadas para alguém nessa posição, quais são as movimentações possíveis a partir desse cargo e diversas outras informações.

Anteriormente, o projeto já continha um banco de dados estruturado no qual as informações eram carregadas, boa parte dos *endpoints* e um protótipo do *front-end* já desenvolvidos. Contudo, os *endpoints* desenvolvidos possuíam implementação apenas para o método de requisição do tipo *get*, ou seja, o usuário, até o momento, conseguiria fazer apenas operações de consulta das informações.

Pensando nisso, a principal atividade consistiu na implementação de outros métodos de requisição para o projeto. Foram eles: *post* (permite a inserção de novas informações), *put* (permite a modificação de informações já existentes) e *delete* (permite a remoção de informações). Além disso, também foram desenvolvidos testes unitários para o método *get* (previamente implementado).

Essa atividade consistiu em um grande desafio pelo fato de tudo ter sido desenvolvido utilizando a linguagem de programação Java, para a qual não se possuía qualquer experiência. Além disso, para o desenvolvimento dos métodos, foi utilizado o *Spring Boot framework* para Java que oferece diversos recursos para um desenvolvimento simples de aplicações web.

2.3.5 4ª Equipe

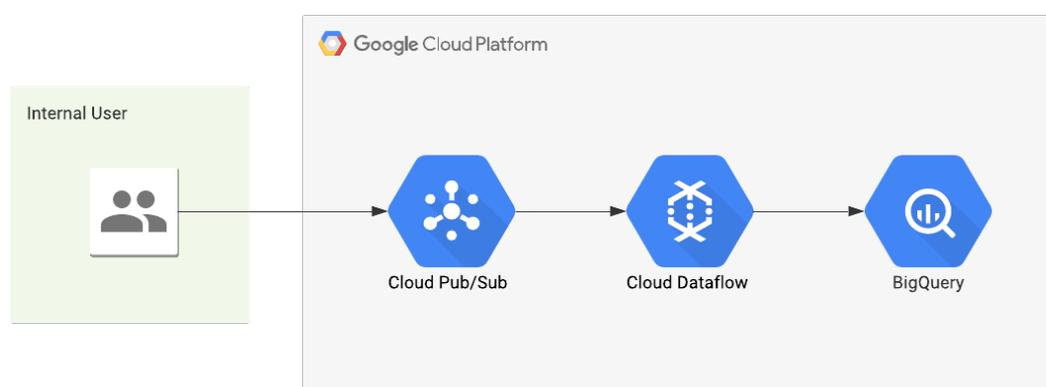
Equipe de *BigData* da companhia, a BEE Data Platform é responsável por coletar, armazenar, analisar e desempenhar as mais diversas operações com enormes quantidades de dados das mais variadas fontes, gerando dados que são consumidos por diversas outras áreas da companhia. Seu objetivo principal é impulsionar a cultura de dados na companhia com uma plataforma disponível e confiável, que permita intensificar o uso de dados através de uma experiência colaborativa.

Diferentemente das outras 4 equipes visitadas, essa etapa do estágio não foi focada em de fato realizar alguma atividade ou projeto para a equipe, mas sim em aprender a

usar determinadas ferramentas e recursos. Quatro pessoas da equipe se revezaram para mentorear o aprendizado de três recursos do *Google Cloud Platform* (*Pub/Sub*, *Dataflow* e *Data Studio*), além do *framework* em Python, *Flask*.

Assim, foi desenvolvido um *script* em Python para criar um tópico e uma assinatura no *Pub/Sub*, um *pipeline* no *Dataflow* e uma tabela no *BigQuery*. Além disso, um outro *script* é responsável por enviar a mensagem para o tópico no *Pub/Sub*, que por sua vez é percebido pela assinatura. O *pipeline* no *Dataflow* é responsável por realizar o *streaming* da mensagem entregue pela assinatura, salvando as informações em uma tabela no *BigQuery*. Na Figura 5, é representado graficamente o fluxo explicado.

Figura 5 – Fluxo final da atividade.



Fonte: Autoria própria.

Por fim, foi utilizado o *Data Studio* (serviço do GCP que se assemelha ao *Microsoft Power BI*) para criação de um *dashboard* a partir de um conjunto de dados com diversas informações, em que ficou livre para o estagiário a escolha dos dados que queria mostrar. Assim, foram escolhidos dados sobre incidentes ocorridos nas marcas nos últimos dois anos, apresentando diversas informações interativas como quantidade total de incidentes distribuída em uma série temporal, além de um gráfico de rosca com a quantidade de incidentes por marca e uma série de outras informações. Por se tratar de informações sensíveis da companhia, o *dashboard* não pode ser anexado a este relatório.

2.3.6 5ª Equipe

Esta equipe é responsável pelo algoritmo, sistemas e dados relacionados ao mecanismo de busca nos sites de *e-commerce* da companhia (*Americanas.com*, *Submarino*, *Shoptime* e *SouBarato*), porém, é apenas focada no *backend*.

Como uma outra equipe é responsável pelo *frontend*, eles desenvolveram uma aplicação que imita um *endpoint* ainda não finalizado. Dessa forma, o pessoal responsável pelo *frontend* não precisa esperar o fim do desenvolvimento do *backend* para iniciar suas atividades. A aplicação funciona da seguinte forma: é possível cadastrar um determinado *endpoint* na aplicação e inserir um exemplo do formato do conteúdo que será utilizado no verdadeiro *endpoint*. Por exemplo, o *endpoint localhost::8080/home/exemplo* recebe o seguinte conteúdo:

```
{
    "chave1": "valor1",
    "chave2": {
        "chave3": "valor3"
    }
}
```

Contudo, não seria mais possível acessar a versão antiga caso alguém atualize esse *endpoint* para receber, por exemplo:

```
{
    "chave1": "valor1",
    "chave2": "valor2",
    "chave3": "valor2"
}
```

Para resolver esse problema, a atividade desenvolvida consistiu em desenvolver um sistema de versões para os *endpoints* da aplicação. Para isso, utilizou-se a linguagem de programação Python e o *framework* Django, que tem o mesmo propósito que o *Flask* (Python) e o *Spring Boot* (Java), mas que, para a linguagem em questão, é mais robusto. Dessa forma, após o desenvolvimento, é possível acessar qualquer versão, desde que ela seja informada no cabeçalho da requisição; caso nenhuma versão seja informada, será mostrada a mais recente. O versionamento é automático, de forma que o usuário não precisa se preocupar em explicitar a versão que ele está registrando.

Além disso, foi desenvolvido um *endpoint* específico que mostra ao requisitante todas as versões existentes e também foi desenvolvido o tratamento de erros para que, por exemplo, o usuário seja corretamente informado caso tente fazer uma requisição equivocada, como explicitar uma versão inexistente.

3 Segunda Parte - Embedded

As informações presentes nesse capítulo relatam as atividades desenvolvidas durante a segunda parte do Estágio Integrado, realizado no Laboratório Embedded, além de explicar com maiores detalhes os algoritmos estudados.

3.1 Laboratório de Sistemas Embarcados e Computação Pervasiva - Embedded

O Laboratório de Sistemas Embarcados e Computação Pervasiva (Embedded) foi fundado em dezembro de 2005 e integra o grupo de laboratórios do Centro de Engenharia Elétrica e Informática (CEEI) da Universidade Federal de Campina Grande (UFCG), em Campina Grande, Paraíba. Na Figura 6 é apresentada uma fotografia da fachada do laboratório.

O Embedded é credenciado no Comitê da Área de Tecnologia de Informação (CATI) para receber recursos da Lei de Informática, tendo o Parque Tecnológico da Paraíba como interveniente financeiro também credenciado no CATI. Diante disso, possui um histórico de parcerias com grandes empresas, em diversos projetos relacionados com sua área de atuação.

Atualmente, o laboratório possui linhas de atuação nas áreas de: Internet das Coisas (IoT), *Software-Hardware Co-Design*, Microeletrônica e outras áreas voltadas para tecnologia de ponta. A equipe do laboratório é formada por professores e pesquisadores do CEEI, e alunos de doutorado, mestrado e graduação.

Figura 6 – Fachada do Laboratório.



Fonte: <www.embedded.ufcg.edu.br>. Acesso em: 10 mar. 2021.

3.2 Fundamentação Teórica

3.2.1 Aprendizado por Reforço

Aprendizado por reforço ou *reinforcement learning*[1] é uma área do Aprendizado de Máquina (*Machine Learning* ou simplesmente ML). Diferentemente de outras principais técnicas de ML, como o Aprendizado Supervisionado (*Supervised Learning*) e o Aprendizado Não Supervisionado (*Unsupervised Learning*), o aprendizado por reforço não parte, no geral, da utilização de dados já existentes para o treinamento de seu agente, ou seja, a máquina deve aprender por si só como desempenhar determinada função.

Essa técnica é bastante utilizada em situações em que o próprio programador desconhece a melhor saída para o problema ou situações em que simplesmente seja muito difícil coletar dados para o treinamento, por exemplo. Contudo, assim como a maioria das técnicas de aprendizado de máquina, a utilização de determinado algoritmo depende bastante do problema e da criatividade daquele que programa para apresentar uma solução.

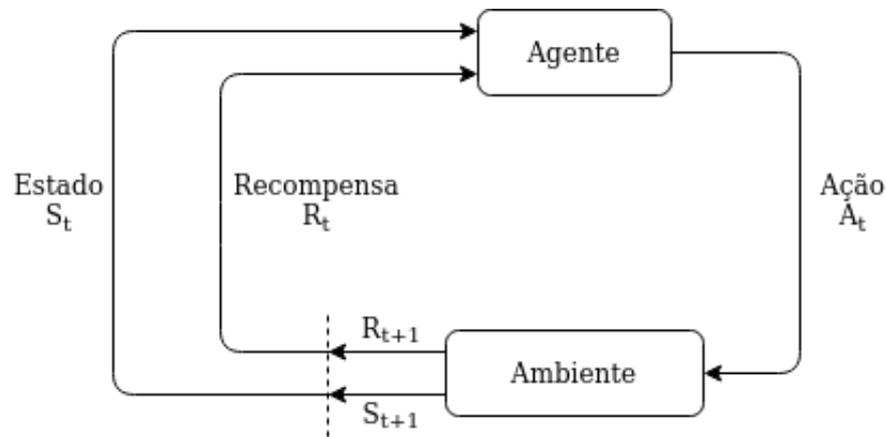
Um exemplo bastante comum é o caso de “robôs” desenvolvidos para jogar xadrez. Existem virtualmente infinitas possibilidades de desfecho em um jogo de xadrez e cada jogada é única em sua situação, já que é dependente de uma série de fatores, e gerará inúmeras novas possibilidades após ser jogada. Presumindo que o objetivo seja treinar uma máquina que seja uma boa jogadora e que saiba se adaptar às mais diversas situações em que se encontrará, percebe-se que não faz sentido utilizar métodos como aprendizado supervisionado em seu treinamento. Deseja-se treinar um agente que saiba analisar a situação em que se encontra e entregar o melhor movimento possível. Outro exemplo bastante comum é o de carros autônomos. É impossível ensinar a um carro que deve dirigir por conta própria todas as decisões que ele deve tomar, já que o ambiente extremamente volátil do trânsito requer constantes tomadas de decisão para situações muitas vezes imprevisíveis.

O algoritmo, portanto, funciona da seguinte maneira: o agente (aquele que toma decisões) se encontra em um determinado ambiente em uma determinada situação (estado). O agente não sabe que ações ele deve tomar para alcançar o resultado final desejado e, por isso, realiza alguma ação aleatória, que vai mudar o estado do ambiente em que se encontra e vai então tomar uma nova ação, repetindo o processo por um número determinado de vezes. A cada ação tomada pelo agente, uma recompensa é gerada de forma que, quanto melhor for a decisão do agente, maior será sua recompensa e quanto pior for sua decisão, menor será a recompensa.

Por exemplo, digamos que o agente em questão é um robô e seu objetivo final é caminhar para um ponto no plano diretamente à sua frente. Caso o robô decida dar um passo para trás ele receberá uma recompensa negativa e, caso decida dar um passo para frente, receberá uma recompensa positiva. Desta forma, na próxima vez que este

robô se encontrar na mesma ou em semelhante situação, ele saberá que a melhor ação a ser tomada é dar um passo para frente e não para trás. Na Figura 7 é apresentado esse processo.

Figura 7 – Princípio básico do algoritmo de Aprendizado por Reforço.



Fonte: autoria própria

3.3 Atividades Desenvolvidas

A primeira atividade dessa etapa do estágio foi o estudo de Aprendizado por Reforço e Aprendizado de Máquina de modo geral, com ênfase no primeiro. Antes de qualquer etapa prática, foi necessário entender bem qual a premissa e se poderia ser aplicado no problema proposto: controlar um quadrotor por meio de técnicas de aprendizado de máquina. De forma simples, um quadrotor é um veículo aéreo não tripulado cujo sistema de propulsão é composto por quatro motores conectados a hélices (podendo ter mais, dependendo do modelo), dois girando em sentido horário e dois em sentido anti-horário. Ao modificar a velocidade de rotação dos rotores, é possível alterar o empuxo produzido por cada propulsor, alterando os momentos em cada eixo do quadrotor, e o empuxo resultante.

Assim como qualquer técnica de aprendizado de máquina, como o aprendizado supervisionado, existem diversos algoritmos que podem ser aplicados, cada um sendo melhor para determinadas situações. Sabendo disso, pensou-se em estudar um algoritmo de aprendizado por reforço relativamente simples, mesmo que não fosse o ideal para o problema final, de forma a ajudar no entendimento geral.

Q-Learning

Dessa forma, o algoritmo escolhido foi o *Q-Learning*[2], tomando como base um código exemplo¹ focado em ensinar um “carro” a buscar passageiros em determinada posição do plano e deixá-lo em outra, previamente selecionada. O algoritmo, em uma explicação simplificada, utiliza uma tabela, *Q-table*, para armazenar informações de experiências anteriores e, assim, escolher a melhor ação a ser tomada naquela situação. Com o passar dos episódios, ou seja, de cada experiência completa do treinamento, a tabela é atualizada, melhorando assim o conhecimento das consequências de suas ações futuras. Os valores armazenados na tabela são chamados de valores-Q (*Q-values*), que são mapeados para um par estado-ação, cujo valor para uma combinação específica representam a “qualidade” de uma ação tomada para aquele estado. Os valores-Q são calculados pela estimação de uma função-Q. Melhores valores-Q implicam em melhores chances de conseguir recompensas maiores.

Essa atividade consistiu, além do estudo teórico do algoritmo, na execução e análise do código em questão e foi essencial para complementar o estudo puramente teórico realizado anteriormente, de forma a consolidar os conceitos de agente, ambiente, ação e recompensa.

Percebe-se que, por se tratar de um algoritmo que utiliza uma tabela de dimensões finitas para “prever” a melhor ação a ser tomada, o *Q-Learning* não deve se encaixar muito bem em situações em que as possíveis ações não estão no domínio discreto, o que é o caso do objetivo deste trabalho, o quadrotor. Por se tratar de um objeto com três estados de posição angular e três estados de posição translacional que assumem valores contínuos, fica claro que o *Q-Learning* não seria a melhor escolha. Por isso, surgiu a necessidade de investigar um algoritmo que melhor se encaixasse à situação.

Deep Deterministic Policy Gradients

O próximo algoritmo escolhido para estudo foi o *Deep Deterministic Policy Gradients*[3], conhecido como DDPG. O algoritmo foi publicado em 2015 como uma adaptação do Q-Learning para espaços de ações com domínio contínuo, o que se mostra como uma evolução natural da pesquisa, contornando o problema apresentado pelo algoritmo anterior.

A política (*policy*) pode ser definida como uma densidade de probabilidade cujo agente utilizará para escolher suas ações. Utiliza-se, portanto, redes neurais profundas para estimar a política do agente, que toma informações do ambiente como entrada e retorna ações. De forma semelhante ao Q-learning, as ações, estados e recompensas são

¹ MARTIN, B. Reinforcement Q-Learning from Scratch in Python with OpenAI Gym. Disponível em: <<https://www.learndatasci.com/tutorials/reinforcement-q-learning-scratch-python-openai-gym/>>. Acesso em 20 dez. 2020.

monitoradas e utilizadas para geração de “pesos” dos parâmetros da rede, que são atualizados conforme o passar dos episódios. Como em todo algoritmo de aprendizado por reforço, a ideia é melhorar o desempenho do agente com o tempo e experiências.

O desempenho do agente é caracterizado por uma função dos pesos da rede neural em questão. A função pode ser considerada como “gradiente ascendente”, também utilizada em diversos outros algoritmos de aprendizado de máquina. Com o passar dos episódios, é esperado que a função retorne valores que aumentem a probabilidade de conseguir recompensas maiores ao tomar ações. Esse processo é chamado de algoritmo de reforço.

O DDPG utiliza quatro redes neurais em sua implementação:

- rede Q (*Q network*);
- rede de política determinística (*deterministic policy network*);
- rede Q alvo;
- rede de política determinística alvo.

Por meio da rede de política determinística, também conhecida como rede atuadora, o atuador mapeia diretamente estados em ações. A rede Q, ou rede crítica (*critic network*), tem duas entradas, a observação do ambiente e sua ação correspondente e, assim, gera um valor-Q, que traduz o quão boa é aquela ação. Além disso, são utilizadas duas redes alvo, que são cópias atrasadas das redes originais para que o sistema tenha mais tempo para considerar ações que tenham sido tomadas recentemente. Dessa forma, a rede pode achar um modelo mais robusto antes de começar a usar os valores atuais. A existência dessas redes aumenta bastante a estabilidade no treinamento.

O algoritmo utiliza, também, algo chamado de *experience replay buffer* ou simplesmente *replay buffer*, que armazena uma lista de tuplas com informações (estado, ação, recompensa e próximo estado) de forma que o aprendizado não é feito apenas com experiências recentes, mas com toda a experiência acumulada até o momento. Por fim, por se tratar de um domínio contínuo de ações, a exploração é feita ao adicionar ruído à própria ação, garantindo uma tomada de decisão estocástica. Com o passar do tempo, as ações vão se aproximando a seleções determinísticas, de forma que ações com maiores probabilidades de receberem melhor recompensa são selecionadas mais frequentemente.

O código utilizado nessa etapa² contém um modelo de um quadrotor com a aplicação do algoritmo DDPG. Utilizando a explicação mais detalhada do autor³, foi possível

² <https://github.com/bonn0062/quadcopter2>

³ BONNER, A. How to Train Your Quadcopter. Disponível em: <<https://towardsdatascience.com/how-to-train-your-quadcopter-adventures-in-machine-learning-algorithms-e6ee5033fd61>>. Acesso em 20 dez. 2020.

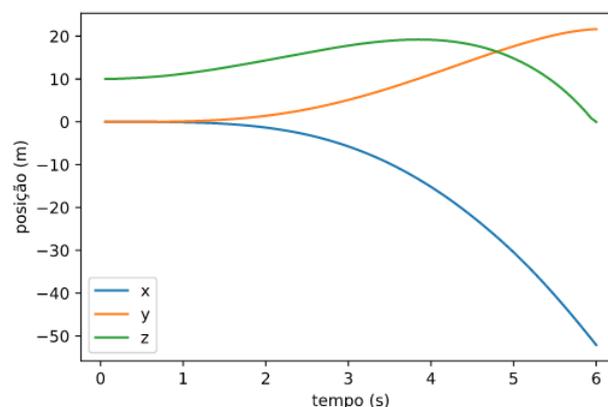
realizar as modificações no código necessárias para para o correto funcionamento do projeto. Além disso, alguns pacotes se mostraram incompatíveis e a execução não foi possível ao instalar os pacotes apresentados no arquivo *requirements.txt*, no qual se espera ter todas as versões compatíveis para uma adequada execução, que se mostrou possível com as seguintes versões:

- *tensorflow 2.3.1*
- *numpy 1.18.5*
- *matplotlib 2.0.0*
- *pandas 1.1.4*

Algumas versões desses pacotes se mostraram incompatíveis com a execução do código e, por isso, a execução de fato da atividade foi atrasada em alguns dias até que todos as versões corretas fossem encontradas e determinados pontos do código fossem alterados para a devida adequação.

A execução é feita por meio de um *notebook* Python já fornecido pelo autor e segue o seguinte fluxo: inicialmente é utilizado um agente muito simples para o controle do quadrotor que seleciona valores aleatórios para as ações de cada um dos quatro rotores. A posição e a velocidade do quadrotor nos eixos x, y, e z são apresentadas nas Figuras 8 e 9 respectivamente. As velocidade angulares comandadas para cada rotor são apresentadas na Figura 10.

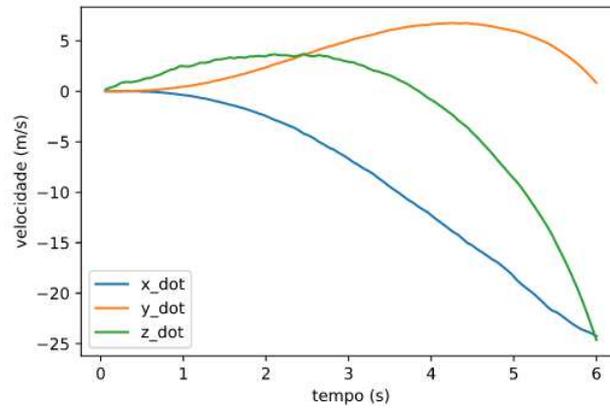
Figura 8 – Posição do quadrotor nos eixos x,y, z.



Fonte: autoria própria.

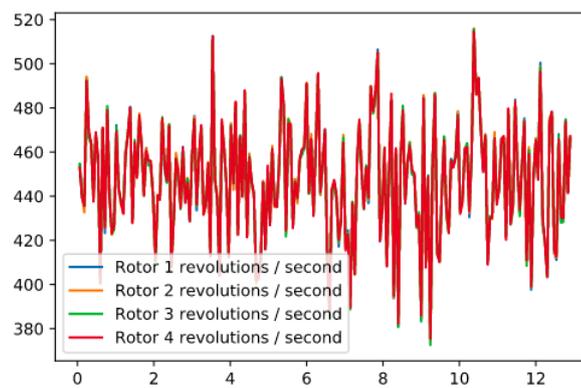
É então feita a troca do agente por um mais robusto e, assim, realizado de fato o treinamento. No exemplo a seguir, foram executados 500 episódios e, a cada execução,

Figura 9 – Velocidade do quadrotor nos eixos x,y, z.



Fonte: autoria própria.

Figura 10 – Rotações por segundo estabelecidas pelo agente em cada rotor.



Fonte: autoria própria.

uma recompensa entre -100 e 100. Além disso, o objetivo é fazer com que inicie na posição (0,0,0) e termine em (0,0,10). Após uma execução, foi possível observar os seguintes resultados:

- Episódios: 500
- Última recompensa: 83,07;
- Melhor recompensa: 93,201;
- Última posição: (-28.3, 5.5, 0.0);
- Melhor posição: (-24.3, 7.2, 0.0);
- Desempenho final (média dos 10 últimos episódios): 80,74.

Apesar de a melhor recompensa ter sido apontada como 93,201, é possível perceber que a melhor posição $(-24,3; 7,2; 0)$ ainda é consideravelmente distante do ponto final esperado $(0;0;10)$, apresentando de forma geral um desempenho abaixo do desejado. É importante ressaltar que essa atividade possuiu um caráter mais exploratório que de implementação. Porém, se torna necessário uma maior exploração do algoritmo para ser possível determinar se o algoritmo é viável ou não para o problema proposto, uma vez que o resultado da simulação não foi satisfatório. Nesse sentido, os trabalhos futuros devem contemplar: *(i)* a alteração de parâmetros do algoritmo para que se alcance um desempenho satisfatório; *(ii)* a inclusão de dinâmicas de voo não modeladas no código atual; *(iii)* uma melhor visualização das grandezas analisadas e *(iv)* a construção de um código mais robusto.

4 Considerações Finais

Este trabalho detalhou as atividades relacionadas à execução curricular do estágio integrado concluído em duas etapas. Foram desenvolvidas tarefas que podem ser relacionadas, principalmente, às áreas de Programação e Automação, apresentando diversos conceitos de computação em nuvem, desenvolvimento de *software* e ciência de dados (análise de dados e aprendizado de máquina).

É possível, então, perceber que boa parte desses conteúdos podem ser encontrados em variados níveis de profundidade, em diversas disciplinas que compõe a grade curricular do curso de Engenharia Elétrica, como: Informática Industrial, Sistemas em Tempo Real, Redes de Computadores, Introdução à Programação e Técnicas de Programação.

A experiência trouxe ao aluno um extremo desenvolvimento tanto no âmbito profissional quanto no âmbito pessoal. A possibilidade de trabalhar em uma das maiores companhias digitais da América Latina proporcionou contato direto com diversas ferramentas e linguagens de programação que estão muito presentes no mercado de trabalho atual. Além disso, a experiência de passar por diversas equipes da companhia e conhecer diversas pessoas muito experientes proporcionou uma visão de mundo e de mercado que não é possível alcançar apenas com a formação dentro da Universidade.

Já na segunda parte do estágio, também foi possível ter contato com a área de pesquisa acadêmica, dentro de um laboratório de ponta e reconhecido nacionalmente, o que leva, também a uma maior maturidade profissional e o maior conhecimento de atuação profissional. Finalmente, conclui-se que a soma das duas experiências foi muito importante para a formação profissional do aluno ao concluir os objetivos do estágio.

Referências

- [1] Kaelbling, Leslie P.; Littman, Michael L.; Moore, Andrew W. (1996). "Reinforcement Learning: A Survey". *Journal of Artificial Intelligence Research*. 4: 237–285. Disponível em: <https://arxiv.org/abs/9605103>.
- [2] van Hasselt, Hado; Guez, Arthur; Silver, David (2015). "Deep reinforcement learning with double Q-learning". *AAAI Conference on Artificial Intelligence*: 2094–2100. Disponível em: <https://arxiv.org/abs/1509.06461>
- [3] LILLICRAP, T. P., et al. "Continuous Control With Deep Reinforcement Learning", 2015. Disponível em: <https://arxiv.org/abs/1509.02971>