

André Igor Nóbrega da Silva

**Relatório de Estágio Supervisionado  
Aquarela Inovação Tecnológica**

Campina Grande, Paraíba

Outubro de 2021

André Igor Nóbrega da Silva

## **Relatório de Estágio Supervisionado** **Aquarela Inovação Tecnológica**

Relatório de Estágio Integrado submetido à Coordenação de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande, Campus 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: Danilo Freire de Souza Santos, D.Sc

Campina Grande, Paraíba  
Outubro de 2021

André Igor Nóbrega da Silva

## **Relatório de Estágio Supervisionado Aquarela Inovação Tecnológica**

Relatório de Estágio Integrado submetido à Coordenação de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande, Campus Campina Grande, como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciências no Domínio da Engenharia Elétrica.

Trabalho aprovado. Campina Grande, Paraíba, 19 de outubro de 2021:

---

**Danilo Freire de Souza Santos, D.Sc**  
Orientador

---

**Jaidilson Jó da Silva**  
Convidado

Campina Grande, Paraíba  
Outubro de 2021

# Agradecimentos

Agradeço, primeiramente, à minha família, Solange Nóbrega, José Artur e Arthur Victor por me trazerem até aqui. Vocês sempre estiveram lá por mim nos conselhos sábios, no entendimento e na torcida pelo sucesso. Agradeço à minha melhor amiga e namorada Camila Machado, pelo carinho, pelos ensinamentos, pelo amor e pelo apoio incondicional.

Agradeço ao professor Danilo Freire pela orientação e acompanhamento ao longo deste estágio. Agradeço, também, à empresa Aquarela pela oportunidade de crescimento. Em especial, ao Marcos Santos e Daniela Zanesco pelo acolhimento e ao Rodrigo Miranda pelos diversos ensinamentos. Agradeço, também, aos meus colegas João Gabriel, Lukas Correa, Renata Dias e Gabriel Dias pelo companheirismo nas dificuldades enfrentadas.

# Lista de ilustrações

Figura 1 – Diagrama do ETL . . . . .	10
Figura 2 – Representação do conceito de DAG . . . . .	10
Figura 3 – Representação da implementação de uma DAG . . . . .	11
Figura 4 – Representação da metodologia DCIM . . . . .	13
Figura 5 – Representação da metodologia Scrum . . . . .	14
Figura 6 – Diagrama com as principais etapas da análise exploratória dos dados . . . . .	15
Figura 7 – Representação da estratégia de escolha de <i>features</i> implementada . . . . .	16
Figura 8 – Representação da construção da variável aleatória do movimento Browniano . . . . .	18
Figura 9 – Gráfico com a simulação de Monte Carlo com movimento browniano . . . . .	18
Figura 10 – Gráfico com a fronteira eficiente de portfólios . . . . .	20

# Lista de abreviaturas e siglas

UFCG	Universidade Federal de Campina Grande
SC	Santa Catarina
ETL	<i>Extract, Transform and Load</i>
DAG	<i>Directed Acyclic Graph</i>
ACL	Ambiente de Comercialização Livre

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>7</b>
1.1	Objetivos	7
1.2	Estrutura do relatório	7
<b>2</b>	<b>FUNDAMENTAÇÃO TECNOLÓGICA</b>	<b>8</b>
2.1	Ferramentas Utilizadas	8
2.1.1	Python	8
2.1.2	<i>Pandas</i>	9
2.1.3	<i>Scikit-Learn</i>	9
2.1.4	<i>Airflow</i>	9
<b>3</b>	<b>ATIVIDADES REALIZADAS</b>	<b>12</b>
3.1	Ambiente de Trabalho	12
3.1.1	Metodologia	13
3.2	Desenvolvimento de Modelos de Previsão	14
3.3	Cálculo de Risco de Portfólios	16
3.4	Otimizações de Carteiras por Simulação de Monte Carlo	19
<b>4</b>	<b>CONCLUSÃO</b>	<b>22</b>
	<b>REFERÊNCIAS</b>	<b>23</b>

# 1 Introdução

Neste relatório são apresentadas e descritas as principais atividades desenvolvidas no estágio supervisionado pelo discente André Igor Nóbrega da Silva, do curso de graduação em Engenharia Elétrica na Universidade Federal de Campina Grande (UFCG). O estágio foi realizado na empresa Aquarela Inovação Tecnológica <sup>1</sup>, com sede em Florianópolis - SC, no período entre 01 de Junho de 2021 e 31 de Setembro de 2021, totalizando 522 horas. Em detrimento da pandemia atual do Coronavírus, o estágio foi realizado em formato de *home office*, ou seja, as atividades não foram realizadas dentro dos domínios físicos da empresa.

## 1.1 Objetivos

Além de ser uma exigência para a formação e obtenção de diploma no curso de Engenharia Elétrica, o estágio possui como principais objetivos proporcionar uma experiência profissional para o aluno, de modo que se possa colocar em prática conhecimentos adquiridos ao longo do curso, bem como servir de ponte para o mercado de trabalho. A disciplina de estágio se torna uma maneira de refinar ferramentas técnicas já conhecidas, além de capacitar o aluno em habilidades pessoais, tão importantes para o ambiente profissional.

Dentre as principais atividades desenvolvidas pelo estudante, destacam-se:

- Preparação e transformação de dados em Python
- Desenvolvimento de modelos de estimação baseados em dados
- Otimização por meio de simulações de Monte Carlo

## 1.2 Estrutura do relatório

Este documento está dividido em 4 capítulos. No primeiro realizamos uma introdução ao trabalho e apresentamos os objetivos. No segundo, descreve-se as principais ferramentas utilizadas para o desenvolvimento das atividades. Em seguida, discorre-se sobre as atividades desenvolvidas, assim como os conceitos que o embasaram. Por fim, o quarto capítulo apresenta a conclusão do trabalho.

---

<sup>1</sup> <<https://www.aquare.la/>>



## 2 Fundamentação Tecnológica

Neste capítulo apresentamos as principais ferramentas tecnológicas necessárias para o desenvolvimento dos trabalhos. Como maior destaque, devemos apontar a linguagem de programação Python e seus *frameworks* *pandas*, *numpy* e *scikit-learn*.

### 2.1 Ferramentas Utilizadas

Dentro do contexto do desenvolvimento de *softwares* que envolvem a aplicação de métodos de Inteligência Artificial e, mais especificamente, Aprendizagem de Máquina, a linguagem de programação Python <sup>1</sup> é bastante utilizada. Dessa maneira, de modo a compreender o contexto das atividades desenvolvidas, nesta seção discorre-se sobre os principais *frameworks* e serviços externos utilizados.

#### 2.1.1 Python

Python é uma linguagem de programação inicialmente criada em 1991 com uma filosofia baseada em legibilidade de código. Dentre diversas outras ferramentas para desenvolvimento de *software*, algumas vantagens do Python podem ser destacadas (MINDFIRE, 2017):

- Qualidade de *software*: a qualidade de código é uma das principais bases da construção do Python e da cultura "*Pythonica*". As estruturas da linguagem foram desenvolvidas de modo que possa-se criar códigos legíveis, reusáveis e fáceis de realizar a manutenção
- Produtividade do desenvolvedor: estudos mostram que um código em Python é geralmente de um terço a um quinto menor do que o equivalente em outras linguagens como C++ ou Java.
- Portabilidade: Os programas em Python geralmente podem ser executados sem alterações em todas as grandes plataformas computacionais

Além desses pontos, podemos apontar que o Python possui um grande suporte de bibliotecas padrão. Dentro dos últimos anos, o Python vêm se tornando uma das principais linguagens para aplicações em Ciência de Dados e em Aprendizagem de Máquina por meio da utilização de bibliotecas como *Pandas*, *Numpy*, *Scikit-Learn* dentre outras. Por essas razões, as atividades trabalhadas pelo discente ao longo da disciplina de estágio foram, em sua maioria, utilizando tal ferramenta.

---

<sup>1</sup> <<https://www.python.org/>>

### 2.1.2 *Pandas*

*Pandas*<sup>2</sup> é uma biblioteca *open source* desenvolvida para Python no final de 2009. Trata-se de uma ferramenta altamente utilizada para preparação e análise de dados. O *framework* foi construído a partir de um outro pacote também bastante utilizado para a computação científica, o *Numpy*<sup>3</sup>, essencial para realização de operações em *arrays* e matrizes.

O *Pandas* permite que possamos realizar de maneira fácil tarefas como limpeza de dados, imputação de dados, normalizações, uniões entre tabelas, visualização de dados, análises estatísticas, dentre outras.

### 2.1.3 *Scikit-Learn*

O *Scikit-Learn*<sup>4</sup> é outra biblioteca externa *open source* do Python que dispõe ferramentas simples e eficientes para a realização de análise de dados e construção de estimadores de Aprendizagem de máquina.

Dentre seus principais componentes, podemos apontar algoritmos para a realização de classificações, regressões, agrupamentos, redução de dimensionalidade, seleção de modelos e pré-processamento de dados. Algumas das suas implementações incluem *Support Vector Machines*, *K-Nearest Neighbours*, *Random Forests*, além de ajuda para a realização de técnicas como *GridSearch*, *Cross-Validation*.

Dessa maneira, a biblioteca foi largamente utilizada para a busca de modelos capazes de realizar previsões acerca do preço de ativos dentro do mercado de indústrias brasileiro.

### 2.1.4 *Airflow*

Além de lidar com exploração de dados e construção de estimadores, foi necessário, também, desenvolver uma Pipeline de extração, transformação e carregamento de dados (do inglês *Extract, Transform and Load* ou ETL), conforme ilustrada na Figura 1.

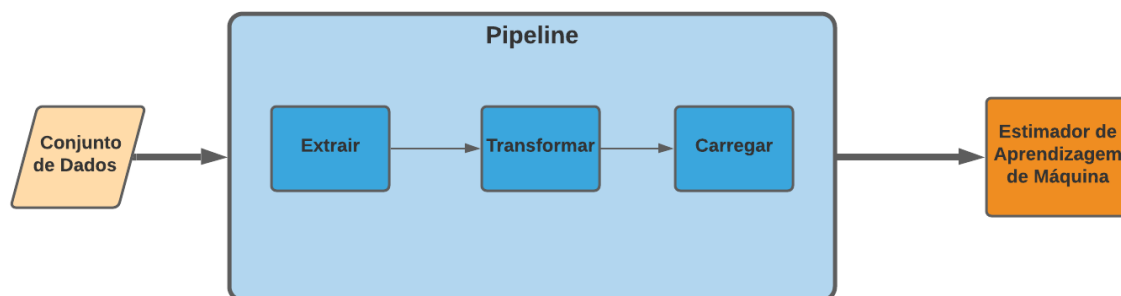
---

<sup>2</sup> <<https://pandas.pydata.org/>>

<sup>3</sup> <<https://numpy.org/>>

<sup>4</sup> <<https://scikit-learn.org/>>

Figura 1 – Diagrama do ETL

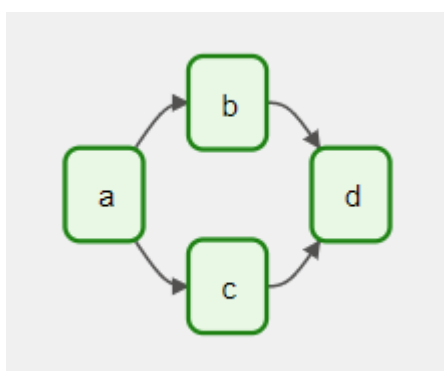


Fonte: Autoria Própria, 2021

Para a realização desta etapa, a ferramenta selecionada foi o *Apache Airflow*<sup>5</sup>, surgida em 2015 como um projeto *open source* da *AirBnB*. Hoje em dia a tecnologia é utilizada em empresas como a *Tesla*, *Square* e *Spotify*. O *AirFlow* é uma ferramenta que permite o processamento de dados em tempo real e de maneira online. Foi desenvolvido em Python, dessa forma, possui amplo suporte com a linguagem, além de uma interface gráfica de uso simplificado.

O principal componente do *Airflow* é chamado de DAG (do inglês *Directed Acyclic Graph*). Trata-se de uma coleção de tarefas a serem executadas de maneira paralela ou sequencial. É possível agendar tarefas a serem executadas, visualizar dependências entre tarefas ou até mesmo diagramas de Gantt. Uma visualização das DAGS pode ser observada nas Figuras 2 e 3.

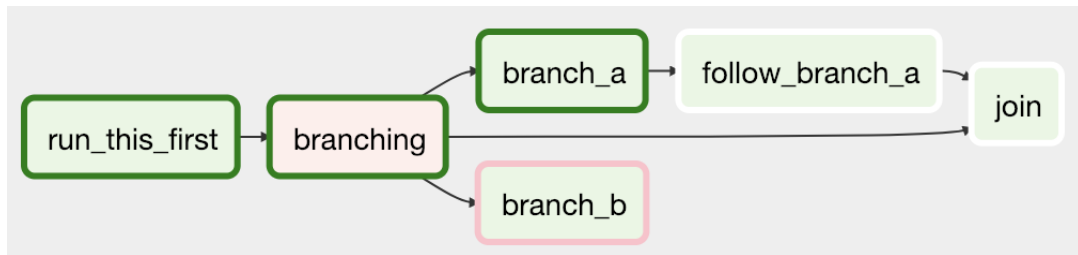
Figura 2 – Representação do conceito de DAG



Fonte: (AIRFLOW, 2021)

<sup>5</sup> <<https://airflow.apache.org/>>

Figura 3 – Representação da implementação de uma DAG



Fonte: (AIRFLOW, 2021)

No exemplo da Figura 2, temos uma DAG composta por 4 tarefas (a, b, c e d). Podemos perceber que existe uma dependência entre tarefas: b e c são dependentes de a; d é dependente de b e de c. As DAGs são implementadas de modo a executar um determinado *script* em Python. Essa estrutura permite que criemos um fluxo de dados ao longo do projeto. Por exemplo, em uma DAG que possui 3 tarefas (A, B e C), a tarefa A pode preparar dados para B analisar enquanto C envia um e-mail sobre os resultados. Ou ainda A pode monitorar uma posição para que B abra uma porta e C acenda luzes.

Seguindo esses conceitos e utilizando o *Airflow*, pode-se montar uma estrutura de extração, transformação e carregamento de dados que são dependentes entre si e que deve ser executada em determinada hora do dia. Utilizando essa estrutura de transformação de dados com o *Airflow*, no próximo capítulo discutimos sobre a construção de estimadores de aprendizagem de máquina, que consomem esses dados preparados. Discutimos, também, sobre os outros dois módulos principais do sistema desenvolvido: o cálculo de risco de ativos e a otimização de portfólios.

## 3 Atividades Realizadas

Neste capítulo discorreremos sobre as principais atividades realizadas pelo discente durante o período de estágio. Aqui destacamos os principais conceitos utilizados pelo aluno no desenvolvimento das tarefas propostas pelo supervisor da Aquarela.

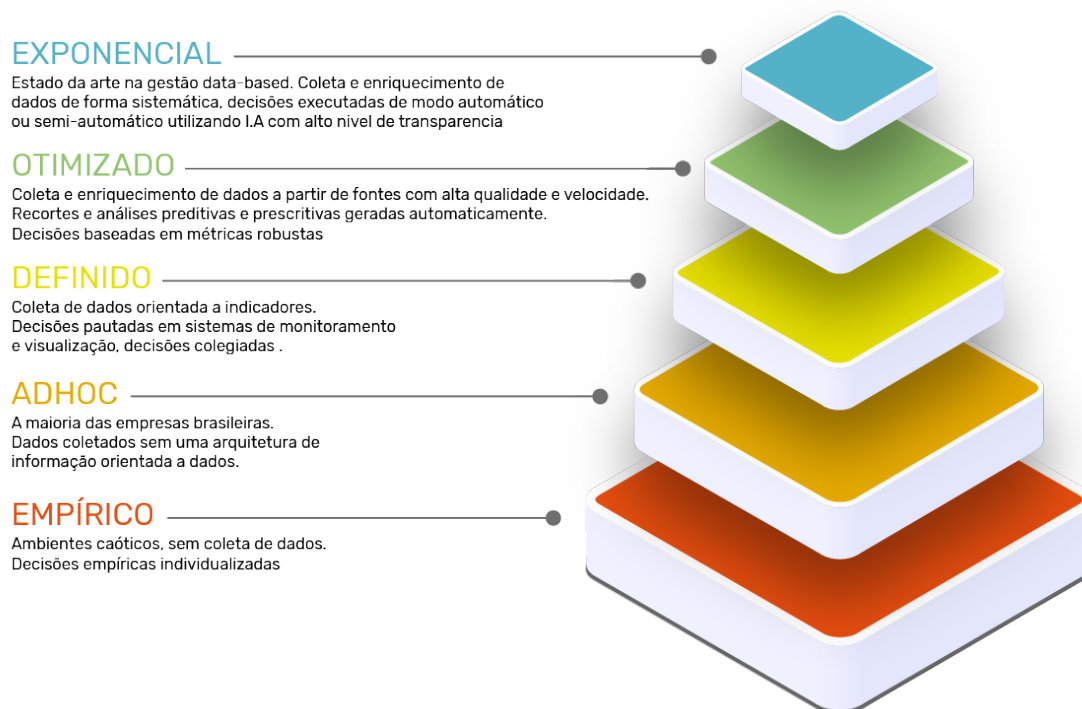
### 3.1 Ambiente de Trabalho

A Aquarela é uma empresa criada no ano de 2010 e atua principalmente na concepção, comercialização e distribuição de produtos e serviços centrados na área da Inteligência Artificial. Dentre as suas principais atividades, podemos destacar o auxílio às organizações no sentido de otimizar processos e operações, criação de novos negócios e aumento da maturidade de análise e cultura centrada em dados.

O objetivo principal da empresa é o de contribuir com o nível de inteligência utilizado no mercado brasileiro, de modo a fomentar uma cultura de tomadas de decisões baseadas em dados. Para isso, a Aquarela atua, principalmente, no setor da saúde, no setor da indústria de alimentos e no setor da energia utilizando uma metodologia própria chamada de DCIM.

Dentre o conteúdo intelectual da empresa, podemos destacar a metodologia *DCIM (Data Culture Introduction Methodology)* e a plataforma Vortex que possuem como objetivo criar sistemas e soluções inteligentes que gerem impacto positivo nas operações dos clientes. A metodologia *DCIM* utiliza uma escala de 5 níveis para entender a maturidade em relação aos dados dos clientes, propondo métodos de ação específicos para cada nível, conforme pode ser observado na Figura 4.

Figura 4 – Representação da metodologia DCIM

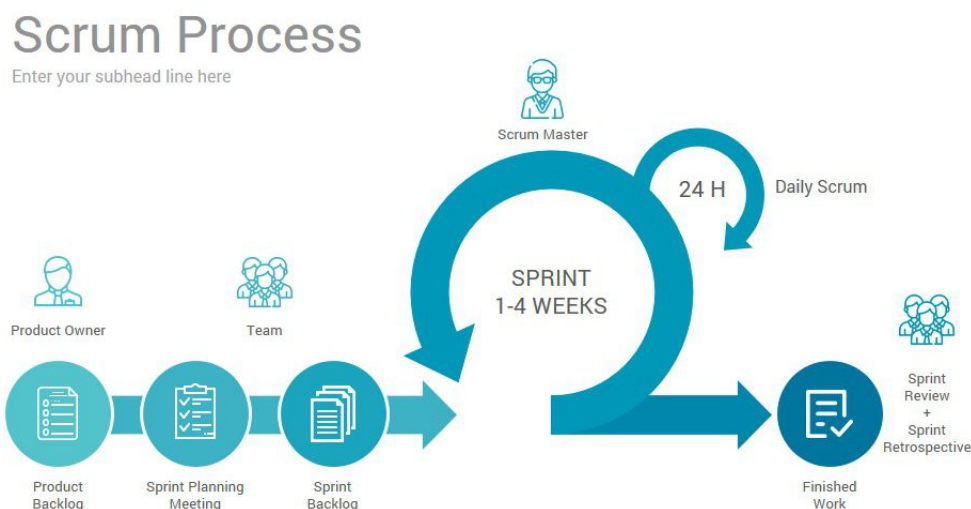


Fonte: <https://www.aquare.la/dcim/>. Acessado em 27 set. 2021

### 3.1.1 Metodologia

O estagiário atuou em um dos projetos da divisão de Operações da empresa, focado no desenvolvimento de projetos para clientes externos. A área de Operações possui uma estrutura organizacional dividida em *Squads*, em que cada *Squad* é responsável por um determinado projeto e utiliza a metodologia ágil *Scrum* (Figura 5) associada ao *Kanban* para gestão e planejamento de atividades. Especificamente na empresa, as *sprints* possuem a duração de uma semana e inicia-se com a *planning*, em que a equipe se reúne e atribui importância e dificuldade às atividades propostas pelo representante do cliente, o *Product Owner*. A gestão das reuniões é de responsabilidade do *Scrum Master*, que é a pessoa também responsável por mapear as dificuldades dos desenvolvedores ao longo das semanas. Ao final da semana, o ciclo se encerra com as atividades de *review* e *retrospective*. Na primeira, a equipe analisa a *sprint* decorrida e observa se as atividades foram concluídas. Na segunda, ocorre uma crítica do grupo acerca dos pontos positivos e negativos percebidos ao longo da semana.

Figura 5 – Representação da metodologia Scrum



Fonte: (SERVICES, 2021)

O projeto de desenvolvimento ao qual o aluno foi alocado possui como objetivo desenvolver um sistema de previsão e alocação de ativos dentro da indústria brasileira utilizando ferramentas de análise de dados. Dentre as principais atividades trabalhadas pelo discente, podemos destacar a análise, preparação e transformação de dados utilizando *Python*, a construção de estimadores de Aprendizagem de Máquina e a otimização de funções baseado em simulações de Monte Carlo.

O aluno atuou sobre orientação de um dos desenvolvedores *seniors* da empresa, que possui experiência e formação na área de estatística e análise de dados. O trabalho desenvolvido pelo aluno e apresentado nas próximas seções passou por uma avaliação da equipe em que o discente foi alocado além do citado orientador.

## 3.2 Desenvolvimento de Modelos de Previsão

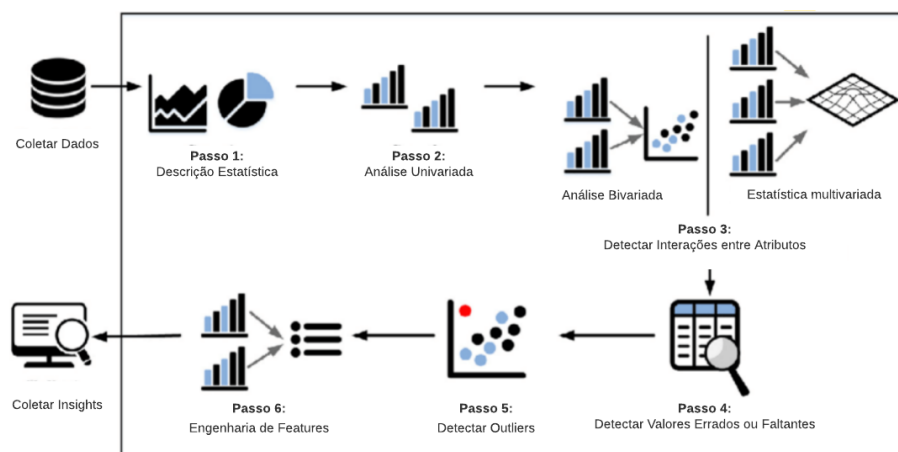
Dentre as atividades desenvolvidas pelo discente em seu período de estágio, devemos destacar o desenvolvimento de um modelo de previsão do preço de ativos comercializados na indústria brasileira. Tal modelo foi construído em cima de um paradigma de dados, ou seja, utilizando Aprendizagem de Máquina. A Aprendizagem de Máquina possui duas grandes divisões principais: a aprendizagem supervisionada e a aprendizagem não supervisionada. A aprendizagem supervisionada, que é o foco deste trabalho, se refere ao problema de encontrar uma função que mapeia entradas e saídas a partir de um conjunto de dado empírico de treinamento. Ou seja, dado um conjunto  $D = \{(x_i, y_i) | i = 1, 2, \dots, n\}$ , em que  $x$  é a variável de entrada e  $y$  é a saída (alvo da predição), queremos encontrar uma função  $f$  tal que  $f(x) \approx y$ . A partir desse processo, queremos utilizar a função  $f$  para realizar predições em dados de entrada não antes vistos  $x_*$ , obtendo  $y_*$ . Como estamos saindo

de um conjunto  $D$  limitado e utilizando a função aprendida para novos dados, podemos afirmar que o treinamento é um processo de indução (RASMUSSEN; WILLIAMS, 2006).

É preciso uma boa compreensão acerca dos dados utilizados para o treinamento que, nesse caso, foram fornecidos pelo cliente do projeto da Aquarela. As seguintes atividades foram realizadas para a construção do estimador:

1. **Análise exploratória dos dados:** nessa primeira fase, o objetivo é encontrar as principais características das variáveis fornecidas para o problema. Dessa forma, foi analisado a descrição estatística dos dados, gráficos de distribuição das variáveis (gráficos de linha, histogramas) e gráficos de interação entre variáveis (correlação cruzada entre as *features* e entre as *features* e a variável alvo). Além disso, observa-se dados faltantes, necessidade de limpeza dos dados e *outliers*. A Figura 6 ilustra as principais etapas deste processo.

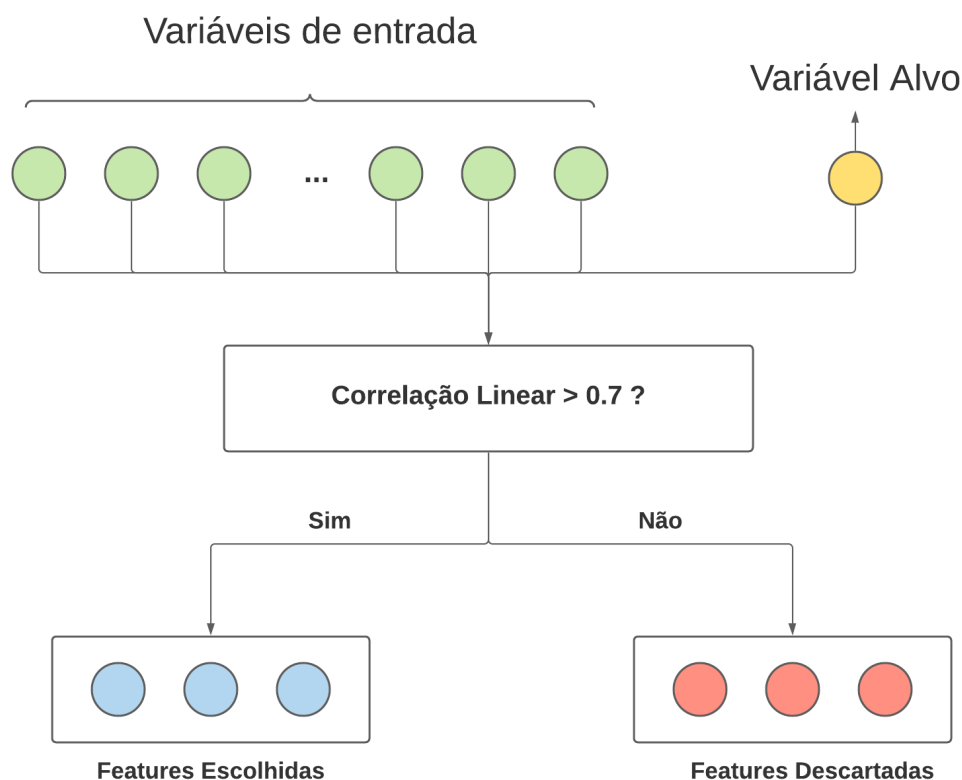
Figura 6 – Diagrama com as principais etapas da análise exploratória dos dados



Fonte: (GHOSH et al., 2018)

2. **Definição de uma estratégia de seleção de *features*:** em um problema real de aprendizagem de máquina, é importante definir uma estratégia de variáveis (entradas e saídas) que se adeque ao problema do cliente. Nem sempre o cliente possui esse mapeamento de função e é uma das tarefas do Cientista de Dados ajudá-lo nesse processo. Nas atividades desenvolvidas no estágio, foi necessário definir um modo de escolha de *features* dentre mais de 60 variáveis para a previsão do preço dos ativos. Implementou-se, então, uma seleção baseada na correlação linear entre as possíveis entradas e a variável alvo, de modo que apenas as *features* que apresentaram uma correlação maior que o *threshold* foram escolhidas. Esse valor do *threshold* se configurou como um dos hiperparâmetros para o treinamento do modelo. A Figura 7 ilustra esse processo.



Figura 7 – Representação da estratégia de escolha de *features* implementada

Fonte: Autoria própria, 2021

3. **Busca em grade por modelos de regressão:** após a escolha das variáveis, foi utilizada uma técnica de busca automática de modelos e de parâmetros, conhecida como a busca em grade (ou do inglês, *grid search*). A busca em grade nos ajuda calcular os valores ideais de hiperparâmetros de modelos. É uma pesquisa exaustiva que é realizada em valores de parâmetros específicos de um modelo. O melhor modelo foi escolhido de acordo com a métrica de avaliação do  $r^2$ .

### 3.3 Cálculo de Risco de Portfólios

Juntamente com previsões acerca do preço dos ativos, o sistema desenvolvido durante o estágio também precisou realizar uma análise de risco acerca dos produtos recomendados pela Inteligência Artificial. Na próxima seção, discutiremos como esse cálculo de risco de investimento foi combinado com as previsões para montar uma carteira de investimento ótima, baseada no índice de *sharpe*.

Um dos principais indicadores atuais para mensuração de risco é chamado de VaR, do inglês *Value at Risk* (valor em risco). o VaR é um indicador estatístico da possível quantidade perda financeira sobre um determinado período de tempo. É bastante utilizado

por bancos, corretoras de investimento e grandes comercializadoras para medir suas possíveis perdas (KENTON; ESTEVEZ, 2021).

Matematicamente, o VaR representa uma probabilidade de perda. Por exemplo, um banco pode determinar que um investimento possui um VaR(95%) de R\$ 100.000,00 reais sobre um período de um mês. Isso significa que há uma chance de 5% do dado investimento apresentar uma perda de R\$100.000,00 em um mês. Existem diversas metodologias para o cálculo do VaR, dentre as quais podemos destacar o cálculo pela matriz de variância e covariância ou o cálculo por meio de simulações de Monte Carlo.

No projeto, implementou-se a técnica de Monte Carlo para o cálculo do VaR utilizando o movimento geométrico Browniano:

- Por meio do retorno histórico dos ativos considerados para o portfólio, constrói-se uma matriz de variância e covariância. Essa matriz representa como a variação no preço de um ativo impacta os demais;
- Define-se uma janela de tempo a qual deseja-se calcular o VaR;
- Adquire-se o preço dos ativos na data atual;
- Por meio do movimento browniano e das matrizes de correlação, constroem-se  $n$  cenários de simulação para a evolução do preço dos ativos ao longo dos próximos períodos de tempo, conforme pode ser visto na Figura 9;
- Por fim, observa-se o preço dos ativos no final do período de simulação e pega-se o preço que representa o percentil de 5% dessa distribuição. A diferença entre o preço inicial e o preço final representa o VaR de 5%, ou seja, existe uma probabilidade de 5% do investimento apresentar essa queda.

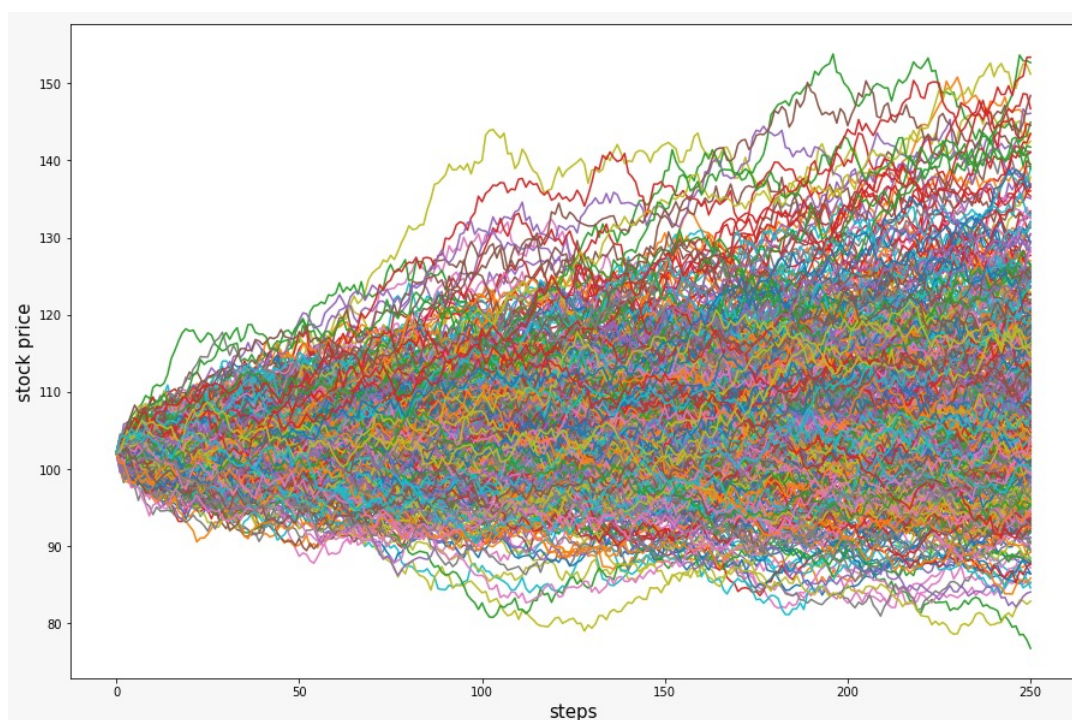
Na Figura 9, cada linha colorida representa uma evolução aleatória do preço de um ativo ao longo do tempo. No eixo das ordenadas está representado o preço do ativo e no eixo das abcissas, o tempo. Intuitivamente, podemos imaginar que o movimento Browniano representa uma caminhada aleatória em que as variáveis aleatórias dos passos são advindas de uma curva gaussiana com média e variância determinada pelos retornos históricos dos ativos. Na Figura 8, ilustramos esse processo: a partir da série de retornos históricos, calculamos a média e a variância. Essas duas variáveis são suficientes para determinar uma curva gaussiana. Na Figura 9, cada linha colorida, para cada linha colorida, a evolução do preço é dado a partir de uma amostragem aleatória dessa gaussiana. Ao final das  $n$  simulações, utilizamos o percentil dos 5% piores preços para determinar o VaR.

Figura 8 – Representação da construção da variável aleatória do movimento Browniano



Fonte: Autoria própria

Figura 9 – Gráfico com a simulação de Monte Carlo com movimento browniano



Fonte: (BROWNIAN..., 2021)

O Movimento Geométrico Browniano (MGB) é um processo estocástico que satisfaz a seguinte equação diferencial (LEWINSON, 2020):

$$dS = \mu dt + \sigma S dW_t \quad (3.1)$$

em que:

- S: preço do ativo

- $\mu$ : o retorno médio do ativo ao longo de um período de tempo
- $\sigma$ : coeficiente de difusão, ou volatilidade dos retornos
- $W_t$ : movimento browniano

O preço do ativo em um dado instante de tempo é dado por:

$$S(t) = S(0)exp\left(\mu - \frac{1}{2}\sigma^2\right)t + \sigma W_t \quad (3.2)$$

Para cada ativo definido no escopo do projeto pelo cliente e para cada portfólio recomendado pelo sistema de IA, foi necessário calcular o risco do investimento por meio da simulação de monte carlo.

### 3.4 Otimizações de Carteiras por Simulação de Monte Carlo

A partir do cálculo de previsão de preços de ativos, e de uma metodologia para o cálculo do risco de um ativo, utilizamos a teoria do portfólio moderno, proposta inicialmente por Harry Markowitz em 1952 (INSTITUTE, 2021) para otimização de portfólios.

A variável que desejamos otimizar é a proporção de investimento em cada um dos produtos. Dados os 10 ativos, uma carteira possível, por exemplo, é o investimento uniforme, ou seja, 10% do dinheiro disponível entre cada um. A partir das previsões obtidas pelo estimador de aprendizagem de máquina e do preço do ativo no instante atual, foi possível calcular uma variável de retorno esperado. Trata-se da expectativa do sistema em relação ao lucro que se pode obter com o investimento. Para fins de ilustração, podemos supor que o preço previsto pelo ativo A é de R\$350,00. O preço atual do mesmo ativo pode ter o valor de R\$300,00, indicando que o sistema acredita que o ativo está sendo negociado a um preço barato. Isso implica em uma oportunidade, já que é possível comprar uma determinada quantidade e, caso as expectativas se concretizem, vender a um preço mais alto no futuro. O valor esperado de retorno é dado por:

$$RetornoEsperado = \frac{P_e(R\$)}{P_m(R\$)} \quad (3.3)$$

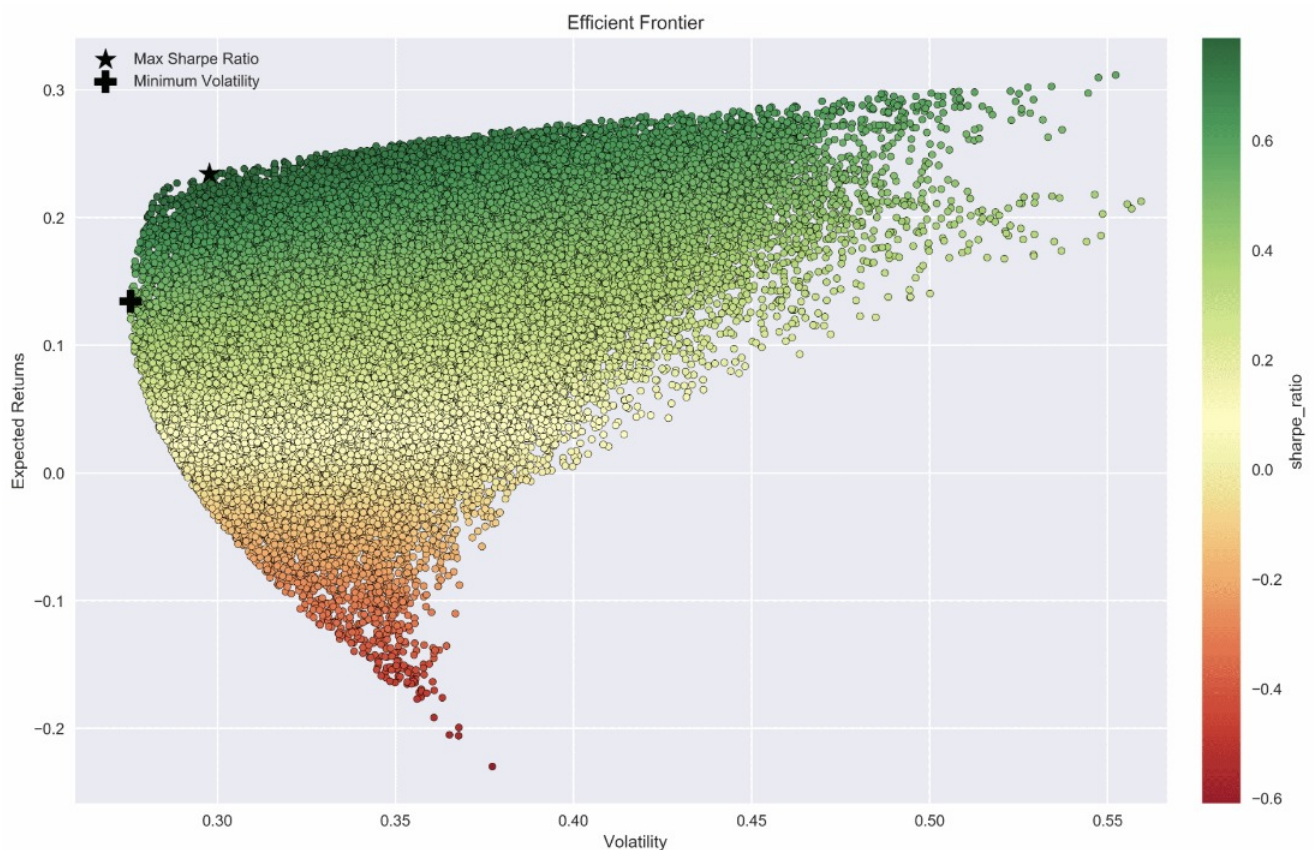
em que  $P_e$  é o preço em reais da previsão do estimador e  $P_m$  é o preço do mercado atual do ativo.

Pela teoria de Markowitz, podemos variar os valores das proporções das alocações, o que implica em variações do retorno esperado da carteira e do risco (VaR) da carteira. Podemos, então, encontrar a melhor carteira observando e maximizando o índice de *sharpe*, dado por:

$$sharpe = \frac{RetornoEsperado}{Risco} \quad (3.4)$$

Utilizamos uma simulação de Monte Carlo para a geração de  $k$  portfólios com proporções aleatórias para cada um dos produtos considerados. Para cada um desses possíveis portfólios, calculamos o retorno esperado e o VaR. Em seguida, selecionamos o portfólio com o maior valor de *sharpe*. Pode-se mostrar que existe uma fronteira eficiente de portfólios possíveis de serem escolhidos. Na Figura 10, cada ponto é uma possível carteira, com diferentes alocações. No eixo das ordenadas, temos o retorno esperado do portfólio e no eixo das abcissas a volatilidade, ou VaR. A cor dos pontos indica a razão entre essas variáveis, ou seja, o índice *sharpe*. Cada um dos portfólios na fronteira externa (à esquerda e acima) é um portfólio ótimo sob alguma métrica. O índice *sharpe* foi escolhido em acordo com o cliente.

Figura 10 – Gráfico com a fronteira eficiente de portfólios



Fonte: (LEWINSON, 2020)

Após todo esse processo, essa recomendação é repassada para o cliente por meio de uma plataforma gráfica. O acompanhamento dessas recomendações e desses investimentos ainda está em andamento. Na seção seguinte, apresentamos a conclusão deste trabalho,

destacando a importância da metodologia utilizada, bem como o impacto das disciplinas cursadas ao longo da graduação em Engenharia Elétrica para a realização destas atividades.

## 4 Conclusão

Neste relatório, descreveu-se as principais atividades desempenhadas pelo discente em seu período de estágio, bem como as ferramentas e conceitos utilizados no processo. O projeto desenvolvido na empresa ainda está em andamento e possui boas perspectivas de escalabilidade futura.

Durante a realização das tarefas, devemos destacar a importância e a flexibilidade dos métodos de desenvolvimentos ágeis. Em projetos que envolvem um grande grau de complexidade, além de multidisciplinaridade, é essencial que haja uma troca constante de conhecimento entre os desenvolvedores e os especialistas na área. Nesse projeto específico, durante *sprints* semanais havia sempre um momento de discussão com a equipe técnica do cliente, enriquecendo o arcabouço intelectual utilizado. Essa conversa constante se converteu em ajustes incrementais ao longo do projeto, de acordo com as necessidades do cliente.

Para o discente, o projeto foi particularmente enriquecedor, já que envolveu diversos assuntos correlatos da Engenharia Elétrica. Uma grande base matemática foi necessária para acompanhar as atividades passadas, algo que foi bastante trabalhado ao longo da graduação nas disciplinas de base. Além disso, disciplinas envolvendo programação se mostraram fundamentais - Introdução à Programação, Técnicas de Programação e Informática Industrial.

Como principais dificuldades encontradas, podemos citar a utilização de conhecimentos complementares ao curso de Engenharia Elétrica. Em particular, a utilização de padrões de desenvolvimento de *software*, algo que foi necessário aplicar quase diariamente. Além disso, devemos apontar a base teórica envolvendo simulações de Monte Carlo e a Teoria Moderna de Portfólios.

# Referências

AIRFLOW, A. *Concepts*. 2021. Disponível em: <<https://airflow.apache.org/docs/apache-airflow/1.10.2/concepts.html>>. Citado 2 vezes nas páginas 10 e 11.

BROWNIAN Motion. 2021. Disponível em: <<https://brainstudy.info/images/brownian-motion-equation-finance>>. Citado na página 18.

GHOSH, A. et al. A comprehensive review of tools for exploratory analysis of tabular industrial datasets. *Visual Informatics*, v. 2, 12 2018. Citado na página 15.

INSTITUTE, F. *Modern Portfolio Theory*. 2021. Disponível em: <<https://corporatefinanceinstitute.com/resources/knowledge/trading-investing/modern-portfolio-theory-mpt/>>. Citado na página 19.

KENTON, W.; ESTEVEZ, E. *Value at Risk (VaR)*. 2021. Disponível em: <<https://www.investopedia.com/terms/v/var.asp>>. Citado na página 17.

LEWINSON, E. *Python for Finance Cookbook*. [S.l.]: Packt Publishing, 2020. Citado 2 vezes nas páginas 18 e 20.

MINDFIRE, S. *Python: 7 Important Reasons Why You Should Use Python*. 2017. Disponível em: <<https://medium.com/@mindfiresolutions.usa/python-7-important-reasons-why-you-should-use-python-5801a98a0d0b>>. Citado na página 8.

RASMUSSEN, C.; WILLIAMS, C. *Gaussian Processes for Machine Learning*. [S.l.]: MIT Press, 2006. Citado na página 15.

SERVICES, T. E. *Curso de Scrum. Por que estudar?* 2021. Disponível em: <<https://blog.training.com.br/projetos/curso-de-scrum/>>. Citado na página 14.