

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Lucas Menezes de Souza

**Sistema de Análise de Risco de Logística de  
Transporte Baseado em Simulação de Monte  
Carlo**

**Uberlândia, Brasil**

**2020**

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Lucas Menezes de Souza

**Sistema de Análise de Risco de Logística de Transporte  
Baseado em Simulação de Monte Carlo**

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: Dr. Rivalino Matias Jr.

Universidade Federal de Uberlândia – UFU

Faculdade de Ciência da Computação

Bacharelado em Sistemas de Informação

Uberlândia, Brasil

2020

# Agradecimentos

Gostaria de agradecer primeiramente a Deus, aos meus pais e meu irmão por todo apoio até o presente momento, pois sem eles não teria chegado onde estou hoje.

Ao Prof. Dr. Rivalino Matias Jr, pela paciência, sugestões, apoio e orientação nessa longa jornada, pois sem ele este Trabalho de Conclusão de Curso não seria possível.

Ao Raif Cervený Passos e Pablo Exequiel Leyes Ontivero pela amizade o indiscutível suporte, motivação e orientação ao longo de todos meus anos na Universidade.

Ao Diego José Oliveira pela amizade e apoio ao longo dos últimos anos.

Ao Sr. Tomoki Tsujikawa pela oportunidade de aplicar os conhecimentos adquiridos ao longo do curso.

E a todas as outras pessoas que de forma direta contribuíram durante meu percurso no curso.

*“Aproveite os desvios da sua jornada. Aproveite muito. Pois, só através deles você encontrará as coisas que precisa para chegar ao final dela.” - Yoshihiro Togashi*

# Resumo

Este trabalho tem como objetivo realizar o desenvolvimento de uma prova de conceito de um sistema de análise de risco quantitativa na área de logística, mais especificamente, desenvolver um sistema capaz de auxiliar o usuário a realizar tomadas de decisões baseado em simulações de tempos de entregas realizadas.

Adotou-se o uso de tecnologias como *Ruby on Rails*, *React.js*, *Docker* e *Docker Compose* com a finalidade de desenvolver uma aplicação capaz de executar em diferentes sistemas operacionais sem que houvesse a necessidade de investir muito tempo nas configurações do ambiente.

O projeto foi desenvolvido em duas frentes principais, *Frontend* e *Backend*, sendo o FDD(*Feature-Driven Development*) a metodologia de desenvolvimento utilizada no decorrer do projeto além de metodologias de análise quantitativa, como simulação de Monte Carlo e análise estatística descritiva.

**Palavras-chave:** Análise, Risco, Quantitativo, Sistema, Web.

# Lista de ilustrações

Figura 1 – Exemplo de função densidade de probabilidade [Matias 2006] . . . . .	18
Figura 2 – Exemplo de Distribuição de Probabilidade [Matias 2006] . . . . .	19
Figura 3 – Exemplo do gráfico da distribuição normal [Barbetta 2010] . . . . .	20
Figura 4 – Transformação Distribuição Normal em evento de distribuição normal padrão [Barbetta 2010] . . . . .	21
Figura 5 – Ordem das etapas de desenvolvimento no FDD [Highsmith 2002] . . . . .	23
Figura 6 – Fluxo melhor detalhado do desenvolvimento no FDD [Retamal 2016] . . . . .	24
Figura 7 – Diagrama da estrutura do projeto em containers . . . . .	25
Figura 8 – Diagrama de caso de uso do projeto . . . . .	25
Figura 9 – Diagrama Entidade-Relacionamento do Banco de Dados . . . . .	26
Figura 10 – Diagrama de atividades de um cenário do sistema . . . . .	27
Figura 11 – Tela inicial do sistema . . . . .	28
Figura 12 – Tela da página de simulações . . . . .	29
Figura 13 – Tela da secção de simulações prévias . . . . .	30
Figura 14 – Tela da secção do Gráfico . . . . .	31
Figura 15 – Método responsável por calcular a média das etapas de entrega . . . . .	32
Figura 16 – Método responsável por calcular a média pela posição do array . . . . .	33
Figura 17 – Representação de parte do algoritmo média do tempo de entrega . . . . .	33
Figura 18 – Classe responsável por gerar números aleatórios baseado em uma dis- tribuição . . . . .	36
Figura 19 – Estrutura das pastas utilizando Docker Compose . . . . .	38
Figura 20 – Conteúdo do docker-compose . . . . .	39
Figura 21 – Conteúdo do dockerfile do backend . . . . .	40
Figura 22 – Conteúdo do dockerfile do frontend . . . . .	40
Figura 23 – Fluxograma para realização de uma simulação . . . . .	41
Figura 24 – Parâmetros utilizados no caso de uso da Entrega 1 . . . . .	42
Figura 25 – Resultado gráfico da simulação do caso de uso da Entrega 1 . . . . .	43
Figura 26 – Parâmetros utilizados no caso de uso da Entrega 2 . . . . .	43
Figura 27 – Resultado gráfico da simulação do caso de uso da Entrega 2 . . . . .	44

# Lista de abreviaturas e siglas

API	<i>Application Programming Interface</i>
FDD	<i>Feature-Driven Development</i>
GSL	<i>GNU Scientific Library</i>
IC	Intervalo de Confiança
JSON	<i>JavaScript Object Notation</i>
MCS	Monte Carlo <i>Simulation</i>
MVC	<i>Model-View-Controller</i>
PMBok	<i>Project Management Body of Knowledge</i>
PMI	<i>Project Management Institute</i>
PDF	<i>Probability Density Function</i>
SAQR	Sistema de Análise Quantitativa de Risco
TCC	Trabalho de Conclusão de Curso
TDD	<i>Test-Driven Development</i>
T.I.	Tecnologia da Informação

# Sumário

	<b>Lista de ilustrações</b> . . . . .	<b>5</b>
<b>1</b>	<b>INTRODUÇÃO</b> . . . . .	<b>9</b>
<b>1.1</b>	<b>Contextualização</b> . . . . .	<b>9</b>
1.1.1	Logística . . . . .	9
1.1.2	Gestão de Projetos e Gerenciamento de Risco . . . . .	10
<b>1.2</b>	<b>Justificativa</b> . . . . .	<b>12</b>
<b>1.3</b>	<b>Objetivos</b> . . . . .	<b>12</b>
1.3.1	Objetivo Geral . . . . .	12
1.3.2	Objetivo Específico . . . . .	12
<b>1.4</b>	<b>Metodologia</b> . . . . .	<b>13</b>
1.4.1	Estudo da literatura . . . . .	13
1.4.2	Acompanhamento do cotidiano . . . . .	13
1.4.3	Processo de desenvolvimento do software . . . . .	13
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA</b> . . . . .	<b>15</b>
<b>2.1</b>	<b>Trabalhos Relacionados</b> . . . . .	<b>15</b>
2.1.1	Análise de Risco . . . . .	15
2.1.2	Logística . . . . .	16
2.1.3	Conclusão sobre os trabalhos relacionados . . . . .	17
<b>2.2</b>	<b>Método Simulação de Monte Carlo (MCS)</b> . . . . .	<b>17</b>
2.2.1	Intervalo de Confiança (IC) . . . . .	19
2.2.2	Distribuição Normal . . . . .	20
2.2.3	Geração de Números Aleatórios . . . . .	21
<b>3</b>	<b>AMBIENTE, PROJETO E IMPLEMENTAÇÃO</b> . . . . .	<b>23</b>
<b>3.1</b>	<b>Ambiente</b> . . . . .	<b>23</b>
<b>3.2</b>	<b>Projeto</b> . . . . .	<b>23</b>
3.2.1	Modelagem . . . . .	24
3.2.2	Requisitos do Sistema . . . . .	28
<b>3.3</b>	<b>Descrição do sistema</b> . . . . .	<b>28</b>
3.3.1	Tela Inicial . . . . .	28
3.3.2	Secção de Simulação . . . . .	29
3.3.3	Secção de simulações prévias . . . . .	30
3.3.4	Secção do Gráfico . . . . .	31
3.3.5	Algoritmo da simulação . . . . .	31

<b>3.4</b>	<b>Implementação</b>	<b>34</b>
3.4.1	Requisitos do Sistema	34
3.4.2	Backend	35
3.4.3	Cálculos estatísticos	35
3.4.4	Frontend	36
3.4.5	Contêiner	37
<b>4</b>	<b>ESTUDO DE CASO</b>	<b>41</b>
4.1	Realização de uma simulação	41
4.2	Entrega 1	42
4.3	Entrega 2	43
<b>5</b>	<b>CONCLUSÃO</b>	<b>45</b>
5.1	Trabalhos Futuros	45
	<b>REFERÊNCIAS</b>	<b>47</b>

# 1 Introdução

## 1.1 Contextualização

Este trabalho está inserido no contexto de uma empresa de logística, e foi aplicado na área de planejamento e entrega de bens. Mais especificamente na estimativa de prazos de entregas de cargas em um processo logístico.

No presente capítulo irei contextualizar alguns dos principais tópicos abordados no trabalho que foi desenvolvido, como: Gestão de Projetos, Análise de Quantitativa de Risco e a área de logística.

### 1.1.1 Logística

De acordo com o artigo de [Silva 2017], logística pode ser definida como uma ferramenta de planejamento do transporte, controle, armazenamento e distribuição de itens, além de serviços que podem ser utilizados no processo produtivo. Tais processos envolvem a otimização de espaço, capacitação profissional, utilização dos recursos de forma eficiente e eficaz, através de sua disposição adequada e padronização de processos.

A área possui algumas atividades principais, tais como: Transporte, manutenção de estoques e processamento de pedidos. Além de atividades de apoio como por exemplo: Armazenagem, manuseio de materiais, embalagem de proteção, obtenção, programação do produto e manutenção de informação.

Por fim, o artigo também elenca diversos tipos de logística, alguns exemplos são: Logística de abastecimento e logística de distribuição.

A logística de abastecimento é a parte correspondente ao conjunto de operações que associam o fluxo de materiais e informações, administrando o transporte dos mesmos entre os fornecedores e a empresa, além da estruturação do processo de abastecimento, estocagem e recebimento.

Por fim, a logística de distribuição consiste na administração dos centros de distribuições, no transporte de cargas, localização das unidades de movimentação controle das rotas de transporte e de expedição.

O presente trabalho tem como intuito auxiliar nas tomadas de decisões em relação ao processo de entrega de produtos, no qual para tal, foram elencadas cinco etapas principais do processo de transporte a fim de realizar a entrega do projeto de conclusão de curso no tempo delimitado.

As etapas são:

- Tempo de seleção do(s) produto(s). (Tempo desde o pedido ser processado até o momento que o(s) produto(s) é(são) separado(s) no galpão para ser(em) carregado(s).)
- Tempo de carregamento do(s) produto(s).
- Tempo de transporte da carga.
- Tempo despendido no recebimento do(s) produtos(s).
- Tempo utilizado para o descarregamento e armazenamento do(s) produto(s).

### 1.1.2 Gestão de Projetos e Gerenciamento de Risco

Um projeto pode ser definido como esforço de trabalho realizado por um determinado período de tempo, que possui prazo de início e fim.

De acordo com o PMBoK [PMI 2013], a gerência de projetos consiste em aplicar diversos tipos de ferramentas, habilidades, conhecimento e técnicas durante as atividades do projeto com intuito de atender seus requisitos. Tal gerenciamento é realizado através da implementação correta de diversos processos. No modelo de gerenciamento de projetos definido pelo PMBoK, existem 5 grupos de processos:

- Iniciação,
- Planejamento,
- Execução,
- Monitoramento e controle, e
- Encerramento.

Estes grupos categorizam cerca de 47 processos.

Este trabalho está relacionado com a área de Gerência de Projetos em Sistemas de Informação.

A Gerência de Projetos é de fundamental importância em diversas áreas, em especial na área de Sistemas de Informação, pois, através de sua aplicação busca-se assegurar a implementação das melhores práticas para o provimento da solução computacional definida para a organização.

Além disso, a sistematização implementada pela Gerência de Projetos visa otimizar a utilização dos diferentes recursos (ex. humanos, financeiros, materiais, etc.), a fim de evitar problemas tais como perda de prazos do projeto, prejuízos financeiros por erros de

estimativas de custos do projeto, não mitigação de riscos variados, dentre outros aspectos recorrentemente citados na literatura que justificam sua aplicação.

Atualmente, na área de Gerência de Projetos existem diversas abordagens e padrões, como por exemplo PRINCE 2 (*Projects In Controlled Environments*) [AXELOS 2016], ISO 10006 (*International Organization for Standardization 10006*) [ISO 2016], IPMA (*International Project Management Association*) [IPMA 2016] e PMBoK [PMI 2016].

Com base nos trabalhos “*Comparing Prince2 with PMBoK*” [Wideman 2002], “*Comparação da ISO 10006 com o PMBoK, Qualidade de Software*” [Gondim 2005] e “*Comparative Analysis of the ISO 10006:2003 against other Standards and Guidelines related to Software Project Management*” [Flores-Rios, Rodríguez-Elias e Olguín-Espinoza 2009], verificou-se que o PMBoK atende todas as necessidades do trabalho, além do interesse particular do autor desse trabalho em se aprofundar nesta abordagem. Portanto, optou-se por usar a abordagem do PMBoK neste trabalho.

A abordagem preconizada pelo PMBoK concentra-se em um conjunto de melhores práticas para Gerência de Projetos, as quais estão organizadas em dez áreas de conhecimento, que são:

- Gestão de integração,
- Gestão do escopo,
- Gestão de tempo,
- Gestão de custos,
- Gestão de qualidade,
- Gestão de recursos humanos,
- Gestão das comunicações,
- Gestão de riscos,
- Gestão de aquisições e
- Gestão de envolvidos (*stakeholder*) do projeto.

Dentre estas áreas de conhecimento, escolheu-se a área de Gestão de Risco para a realização deste estudo, que será melhor detalhado posteriormente.

Baseado no PMBoK [PMI 2013], o gerenciamento de riscos inclui diversos processos, como planejamento do gerenciamento dos riscos, identificação dos riscos, realização

das análises qualitativa e quantitativa dos riscos, planejamento das respostas aos riscos e o controle dos riscos.

A análise de riscos divide-se em análise qualitativa, que consiste em elencar os riscos com base em suas prioridades para analisá-los ou atuar após serem avaliados a probabilidade e o impacto das suas ocorrências. Já a análise quantitativa, tem como objetivo analisar numericamente o impacto em que os riscos identificados têm, direta ou indiretamente, sobre o projeto.

Um risco de projeto se caracteriza como um evento ou condição que, caso aconteça, influenciará o projeto, mas não necessariamente de forma negativa; um risco pode ter efeitos tanto positivos quanto negativos.

O [capítulo 2.1.1](#) discorrerá mais sobre o tema de análise de risco.

## 1.2 Justificativa

O gerenciamento dos riscos tem como objetivo aumentar a probabilidade e o impacto positivo dos eventos, além de reduzir a probabilidade de impactos negativos que ocorrerem ao longo de um projeto [PMI 2013], ou seja, a redução de incertezas.

Como consequência da implantação do gerenciamento de risco em uma empresa, esta se beneficia obtendo subsídios obtidos de forma sistematizada para seu processo de tomada de decisão, otimização de custos e gerenciamento mais adequado dos seus prazos.

## 1.3 Objetivos

### 1.3.1 Objetivo Geral

O objetivo geral do trabalho foi realizar um estudo de análise quantitativa de riscos baseando-se na abordagem do PMBoK, e desenvolver uma prova de conceito de um sistema de análise quantitativa de risco para uma empresa de logística de transporte.

A prova de conceito desenvolvida tem como finalidade ser um sistema inicial para criar simulações de tempos de entrega para uma empresa de logística a partir de informações providas pelo usuário.

### 1.3.2 Objetivo Específico

Em prol do cumprimento do objetivo geral foram considerados os seguintes objetivos específicos:

- Definir o processo de análise de risco para as etapas de entrega de uma empresa de logística.
- Identificar as principais atividades de logística sujeitas ao processo de análise de risco.
- Criar um gráfico de risco para melhor avaliar os riscos envolvidos no projeto (Objetivo do Software).

## 1.4 Metodologia

Esta seção tem como intuito apresentar as etapas realizadas a fim de alcançar os objetivos especificados na seção 2 do capítulo 1. As etapas a seguir elencam e detalham um pouco melhor as principais fases.

### 1.4.1 Estudo da literatura

A primeira etapa do trabalho consistiu em estudar os temas de gerenciamento de risco de acordo com o PMBoK [PMI 2013], mais especificamente análise de risco quantitativo, a área de logística, *Feature-Driven Development* [Highsmith 2002] e complementar o estudo de *Ruby on Rails* [Rails 2020] e *React.js* [Facebook 2020].

O propósito de estudar esses assuntos foi para adquirir os conhecimentos necessários para a realização das demais etapas do projeto.

### 1.4.2 Acompanhamento do cotidiano

Além da aquisição dos conhecimentos técnicos necessários, outra etapa importante do projeto foi realizar o trabalho de acompanhamento *in loco* nos vários setores da empresa a fim de vivenciar as necessidades e adquirir os conhecimentos de logística exigidos para esse trabalho.

Ao acompanhar os setores da empresa foi possível perceber que o setor de transporte de produtos e mercadorias era uma opção para realizar a aplicação de análise de risco quantitativa.

Com o acompanhamento foi possível elencar as principais etapas do processo de entrega, como foi citado no [capítulo 1.1.1](#)

### 1.4.3 Processo de desenvolvimento do software

Durante o projeto de desenvolvimento do software proposto nesse trabalho foi utilizada a metodologia ágil *Feature-Driven Development* [Highsmith 2002], sendo que o

desenvolvimento do mesmo foi realizado na linguagem de programação e *framework Ruby* [Ruby 2016] *on Rails* [Rails 2020], tendo como banco de dados o *PostgreSQL* [PostgreSQL 2020]. Além disso, *RSpec* [Team 2016] foi utilizado como ferramenta para auxiliar com *Test-Driven Development* [Aniche 2014], que foi aplicado em conjunto com o FDD (*Feature Driven Development*).

O desenvolvimento do trabalho foi realizado em quatro principais etapas. Primeiramente houve a decisão do tema e a busca pela oportunidade de implementar o trabalho, sendo que algumas empresas de logística na cidade de Uberlândia foram abordadas.

Em seguida deu-se início ao estudo sobre gerenciamento de risco, análise quantitativa de risco e logística, o qual foi realizada através do estudo da literatura, e o acompanhamento do cotidiano inicialmente.

Após tais etapas terem sido realizadas, iniciou-se o levantamento de dados, análise de requisito do sistema para que finalmente partisse para à última etapa principal do projeto que foi a do estudo complementar para o desenvolvimento do sistema, implementação e testes do mesmo.

## 2 Revisão Bibliográfica

### 2.1 Trabalhos Relacionados

#### 2.1.1 Análise de Risco

Esta seção trata trabalhos relacionados com a mesma temática ou abordagens utilizadas nesta monografia.

O artigo “Análise Quantitativa de Risco Baseada no Método de Monte Carlo: Abordagem PMBoK” [Matias 2006] tem como objetivo descrever sobre o método de Monte Carlo referenciado pelo PMBoK [PMI 2004], seus principais componentes e distribuições de probabilidades.

Além de abordar as bases da análise quantitativa de risco baseando-se no Método de Simulação de Monte Carlo(MCS). Por fim, o artigo apresenta um exemplo de aplicação da análise quantitativa de risco com o método MCS com duas abordagens, uma baseada em dados coletados através de um banco de dados e outra através das informações repassadas por um profissional especialista.

O artigo relata o fato da crescente demanda da análise quantitativa de risco nos projetos de diferentes áreas e demonstra que um dos desafios de utilizar MCS é uma maior dificuldade de modelar as variáveis aleatórias a partir da lista de riscos identificados, que é algo que deve ser levado em consideração no momento da decisão para aplicação do método.

Um segundo trabalho relacionado à área de gerenciamento de risco é o livro “*Risk Analysis - A Quantitative Guide*” [Vose 2008], especificamente no capítulo 14, “*Eliciting from expert opinion*”, seu objetivo é oferecer uma diretriz sobre como reunir e modelar as informações adquiridas dos profissionais especialistas da área que o gerenciamento de risco foi implementado.

O capítulo se baseia em abordar sobre fontes de viés e erros que podem ser encontrados durante as estimações subjetivas e, posteriormente, são apresentadas e exemplificadas várias técnicas de modelagem e de distribuição usadas na modelagem baseado em informações de especialistas.

Por último, são apresentados métodos de *brainstorming* e reuniões individuais com os especialistas a fim de conseguir coletar informações mais relevantes para serem utilizadas no desenvolvimento da análise de risco. Observa-se que nesse estudo de TCC, todas essas técnicas foram importantes para a obtenção de dados para o desenvolvimento desse trabalho.

Outro trabalho usado como referência para esse estudo é o capítulo 13, *Decision Theory*, do livro “*Probabilistic Risk Analysis: Foundations and Methods*” [Cooke 2001], o qual aborda algumas das ideias mais importantes sobre teoria de decisões, principalmente sobre a importância da análise de decisões na análise quantitativa de risco, que cada vez mais é usada como *input* nos processos de tomada de decisão.

De acordo com o livro, as teorias são divididas em duas: Normativa e prescritiva. As teorias normativas demonstram na prática qual é a melhor decisão, e são preferidas pelos tomadores de decisões que acreditam que essa seja a decisão racional. Enquanto as teorias prescritivas derivam de modelos matemáticos e geralmente utilizam de métodos os softwares para auxiliar os tomadores de decisões a realizarem suas escolhas. Ao longo do capítulo são apresentadas teorias e exemplos sobre as duas ênfases.

Como conclusão deste capítulo do livro, os autores defendem a ideia de que grande parte dos tomadores de decisão já estão preparados para aceitarem prematuramente a ideia base sobre as decisões feitas racionalmente. No entanto, os modelos normativos não são tão triviais para serem usados da melhor maneira, dado que na maioria das vezes os tomadores de decisões provavelmente não sabem com antecedência quais são os *trade-offs* que eles realmente acreditam.

Na análise de risco, além dos *trade-offs* serem de extrema importância e requererem uma abordagem bem elaborada é também necessário um alto grau de entendimento sobre o modelo de tomada de decisão. Para que sejam realmente bem entendidos, para isso acontecer é necessário um alto nível de uso interativo entre o modelo e o tomador de decisão.

### 2.1.2 Logística

Na área de logística, o artigo “A Importância de Sistemas de Informação para a Competitividade Logística” [Nazário 1999] tem como objetivo abordar o grande diferencial que a área de T.I. agrega às diversas áreas nas quais ela é aplicada, seja na parte de comunicação e vendas ou até mesmo na logística de entrega de produtos.

O artigo cita exemplos de algumas empresas de diferentes setores que trabalharam com Tecnologia de Informação e o impacto positivo que é gerado, focando principalmente no papel de Tecnologia da Informação na área de logística. Abordando sobre diversos níveis dos sistemas de informação na área de logística, desde o funcionamento básico dos sistemas de informação até mesmo sobre ERPs (*Enterprise Resource Planning*), planejamento estratégico, controle gerencial, apoio à tomada de decisões, entre outros.

Por fim, o artigo conclui que é cada vez maior a necessidade das empresas terem sistemas integrados devido ao alto benefício que os mesmos trazem para as organizações, e que com tais sistemas integrados podem prover um grande benefício para o alavancamento

das operações de logística.

O segundo artigo sobre logística chamado “Gerenciamento de risco e rastreadores apoiando no processo de gestão da logística: Um estudo de caso.” [Picinin João Luiz Kowaleski 2016] tem como objetivo realizar uma análise de algumas ferramentas de T.I. focadas em gestão de logística e elencar vantagens e desvantagens da utilização dessas ferramentas, do impacto de gerenciamento de risco e da importância das ferramentas que auxiliam o processo de gestão e tomadas de decisão para uma empresa da área de transporte rodoviário de cargas.

O artigo aborda sobre alguns desafios das operações de logística no Brasil, sobre as ferramentas de T.I. aplicadas à área, além do gerenciamento de risco logístico.

De acordo com o trabalho, tem como principal finalidade conseguir prever e/ou calcular os riscos envolvidos na situação em questão, e quais as estratégias devem ser tomadas para que o gerenciamento possa oferecer soluções eficazes para os riscos que foram reconhecidos.

Por fim, o artigo cita os benefícios da utilização das ferramentas de T.I. utilizando um estudo de caso, concluindo que o setor de logística possui diversos processos que exigem um alto nível de aplicação de serviços, que coincidentemente ao artigo anterior, afirma que tais processos e serviços devem conter a menor quantidade de erros possíveis a fim de impedir que os terceiros envolvidos no processo não sejam afetados.

### 2.1.3 Conclusão sobre os trabalhos relacionados

Diferente dos trabalhos citados anteriormente, este trabalho visa unir as áreas de conhecimento de análise quantitativa de risco e logística a fim de desenvolver um *software* baseada nas principais etapas de entrega de uma empresa de logística, trabalhando mais profundamente com a parte estatística de mensuração das probabilidades de eventos da área de logística ocorrerem.

Tais mensurações de probabilidade têm como finalidade garantir um maior controle sobre os acontecimentos provendo as informações necessárias para que o usuário possa criar planos de mitigação dos riscos.

## 2.2 Método Simulação de Monte Carlo (MCS)

De acordo com o artigo “Análise Quantitativa de Risco Baseada no Método de Monte Carlo: Abordagem PMBoK” [Matias 2006], o método de Monte Carlo é um método de solução matemático probabilístico que produz seus resultados com base nas simulações realizadas obtidas através de dados amostrais.

Devido aos avanços tecnológicos, atualmente o método MCS pode realizar milhares de simulações em pouco tempo. Além disso, ele difere dos outros métodos de “discretização” numéricas convencionais, que geralmente são descritos através de equações diferenciais, parciais ou ordinárias descrevendo sistemas físicos ou matemáticos. Já o método de Monte Carlo na maioria das vezes é aplicado através de sua simulação direta, não necessitando de descrever seus comportamentos por meio de equações, porém, para que isso possa acontecer, é necessário que o sistema ou processo possa ser descrito através de distribuições de probabilidade.

Em complemento, de acordo com o livro “Random Number Generation and Monte Carlo Methods” [Gentle 2003], na maioria dos casos, quando expressões matemáticas são avaliadas, é preferido as aproximações padrão de análise numérica, porém o método de Monte Carlo provê uma alternativa que as vezes é a única abordagem tratável. Além disso, MCS frequentemente é preferido para avaliar integrais em domínios que possuem uma alta dimensão.

O método MCS pode ser simplificado em basicamente duas estruturas:

- Função densidade de probabilidade, e
- Gerador de números aleatórios.

A função densidade de probabilidade é um modelo probabilístico construído a partir de valores seguindo uma certa distribuição de probabilidade, definida utilizando dois grupos de valores: as hipóteses dos resultados plausíveis e uma regra que define a probabilidade de cada resultado (ou grupos de resultados) acontecer.

A *Probability Density Function (PDF)* pode ser representada através de dois eixos,  $X$  e  $Y$ , no qual  $X$  são os intervalos dos valores que podem ocorrer enquanto o eixo  $Y$  representa a probabilidade dos intervalos de valores acontecerem, como pode ser visualizado na [Figura 1](#) a seguir, onde o eixo  $X$  seria o eixo de milhões de reais, enquanto os valores de  $Y$  representa a probabilidade em valores decimais.

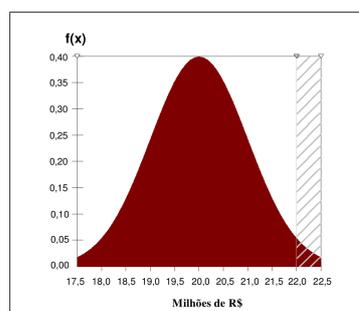


Figura 1 – Exemplo de função densidade de probabilidade [Matias 2006]

Além disso, de acordo com o artigo [Matias 2006], as probabilidades de ocorrência podem ser analisadas através do **intervalo de confiança (IC)**.

Na **Figura 2** é possível observar o conceito de intervalo de confiança sendo aplicado, na qual cada *PDF* tem seu intervalo de confiança delimitando e correspondendo a um nível de 90% de confiança.

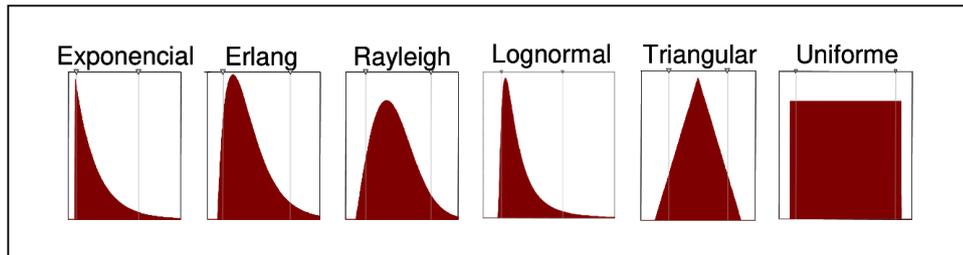


Figura 2 – Exemplo de Distribuição de Probabilidade [Matias 2006]

Supondo que os gráficos da **Figura 2** sejam um exemplo de estimativas do tempo de duração do transporte de uma mercadoria o qual são representados pelos valores do eixo *X*. Então com 90% de confiança, os intervalos demarcados conteriam a provável duração do transporte da mercadoria.

Após ser definido qual distribuição de probabilidade melhor se adéqua ao comportamento da variável aleatória estudada (no caso tempo de transporte de uma mercadoria) é necessário gerar valores aleatório de acordo com esta mesma distribuição. Os valores gerados farão parte da amostra que será utilizada para construir uma outra *PDF* que permitirá a realização de diversas análises, como por exemplo, a quantificação dos riscos e os efeitos do mesmo em relação ao projeto. Para que isso seja possível, o método de Monte Carlo requer a utilização de geradores de números aleatórios capazes de gerar os valores de acordo com as distribuições de probabilidade escolhidas.

Devido às características do MCS, e relativa simplicidade em aplicá-la, a mesma foi escolhida para ser a ferramenta estatística base para o desenvolvimento do sistema.

### 2.2.1 Intervalo de Confiança (IC)

Em estatística, o IC é um tipo de estimativa por intervalo de um parâmetro populacional desconhecido. No qual, o mesmo é um intervalo entre dois valores, o limite inferior e o limite superior. Ambos valores são afetados pelo nível de confiança, que de acordo com [Barbetta 2010], quanto maior o nível de confiança exigido, maior será a magnitude do intervalo de confiança, ou seja, quanto menor for a amplitude do intervalo de confiança mais precisa é a estimativa.

O nível de confiança é estabelecido pelos usuários responsáveis e não segue nenhum padrão. É importante ressaltar que, quanto maior o intervalo de confiança, mais

abrangente será a estimativa.

Em adição a isso, com base no livro “A Modern Introduction to Probability and Statistics: Understanding Why and How” [Dekking C. Kraaikamp 2005], o intervalo de confiança para a média, é a forma mais comum de se referir aos intervalos de confiança feitos para o valor esperado de um certo modelo de distribuição.

## 2.2.2 Distribuição Normal

De acordo com [Barbetta 2010], a distribuição normal é considerada a distribuição de probabilidade mais importante, pois a mesma permite modelar diversos fenômenos naturais e, além disso, possibilita realizar aproximações a fim de calcular probabilidades de muitas variáveis aleatórias que têm outras distribuições, além de ser importante na inferência estatística.

Ela é caracterizada por uma função de probabilidade, na qual o gráfico descreve uma curva em forma de sino. Tal forma de distribuição evidencia que existe maior probabilidade de a variável aleatória assumir valores próximos do centro do gráfico.

A Figura 3, representa a função densidade probabilidade normal, e indica seus dois parâmetros  $\mu$  (média) e o  $\sigma$  (desvio padrão)

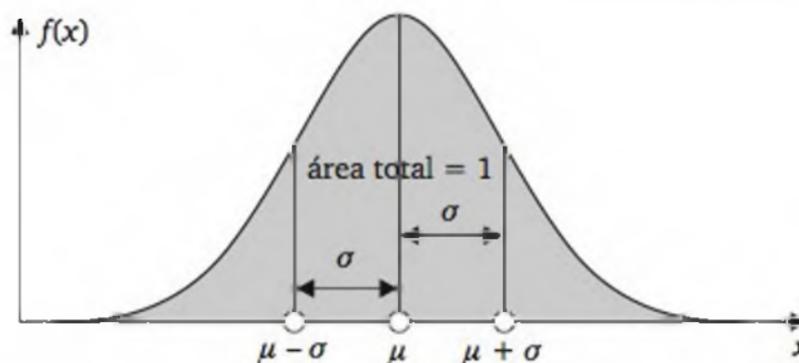


Figura 3 – Exemplo do gráfico da distribuição normal [Barbetta 2010]

[Barbetta 2010] ainda comenta que o modelo de distribuição normal possuem algumas características, como:

- A curva é simétrica e torno de  $\mu$ , conseqüentemente, os valores da média e mediana são iguais.
- A área total sob a curva é igual a 1
- Em teoria, a curva prolonga-se de  $-\infty$  a  $+\infty$

- Qualquer combinação linear de variáveis aleatórias normais é também uma variável aleatória normal.
- O afastamentos da média, em unidades de desvio padrão, preservam a mesma área sob a curva, independentemente dos valores de  $\mu$  e  $\sigma$ .

Há também a distribuição normal com média zero e desvio padrão unitário, em outras palavras  $Z : N(0, 1)$ , que também é chamada de distribuição normal padrão. Na qual, toda e qualquer probabilidade(área na curva) sob a densidade da variável  $X$  em questão, pode ser avaliada sob a densidade de  $Z$ , como demonstrado na [Figura 4](#)

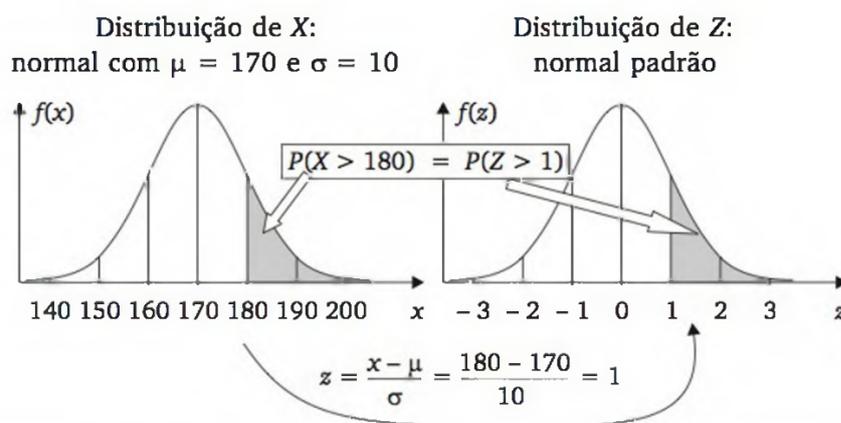


Figura 4 – Transformação Distribuição Normal em evento de distribuição normal padrão [[Barbetta 2010](#)]

As probabilidades de uma variável com distribuição normal podem ser representadas por áreas sob a curva da distribuição normal padrão. Por sua vez, existem algumas tabelas utilizadas, contendo valores pré-calculados dessas áreas, uma delas é comumente chamada de tabela  $Z$ , sendo que uma de suas variantes relaciona os valores positivos de  $Z$  com a área sob a curva superior da curva.

### 2.2.3 Geração de Números Aleatórios

Como abordado no livro “Simulation Modeling and Analysis” [[Law 2014](#)], não é o mais correto dizer geração de variáveis aleatórias, vez que tais variáveis, na teoria da probabilidade matemática, são definidas como funções matemáticas que satisfazem certas condições.

Existe um abrangente campo que tem como objetivo estudar sobre a área de geração de variáveis aleatórias, dentre elas, diferentes metodologias para a geração de números aleatórios.

Ainda, de acordo com [[Law 2014](#)], cita que a aleatoriedade estatística pode até não ser desejável, e que outras propriedades de números gerados, como por exemplo a

'uniformidade' da distribuição dos pontos são mais importantes em certas aplicações, como por exemplo a integração com Monte Carlo.

Para que um "bom" gerador números aleatórios aritmético seja criado, é preciso que o mesmo tenha algumas propriedades, tais como:

1. Os números devem parecer distribuídos uniformemente entre  $[0, 1]$ ;
2. Não devem apresentar correlação entre si;
3. De um âmbito prático, que o gerador seja naturalmente rápido e utilize o mínimo de armazenamento possível;
4. Passível de reproduzir exatamente um determinado fluxo de números aleatórios;
5. Deve haver uma forma do gerador produzir facilmente diferentes saídas/fluxos de números aleatórios;
6. O gerador deve ser portátil, de forma que seja possível de ser reutilizado em diferentes computadores e compiladores padrões.

Por fim, o autor comenta que a maioria do geradores geralmente utilizados atendem os pontos, 2, 3, 4, 5 e 6, mas que alguns falham no critério 1, que, de acordo com ele é absolutamente necessário para obter um resultado correto de uma simulação.

No presente trabalho não há a implementação de um sistema de gerador de números aleatórios, mas a utilização de uma biblioteca que implementa a *GNU Scientific Library* [FSF 2020].

## 3 Ambiente, Projeto e Implementação

O capítulo apresenta informações sobre o ambiente de desenvolvimento, tais como ferramentas, tecnologias e técnicas de desenvolvimento utilizadas. Além disso, são descritas as decisões da arquitetura do projeto, assim como seus respectivos módulos.

### 3.1 Ambiente

A aplicação proposta foi desenvolvida em duas partes, *Frontend* e *Backend*, nas quais cada uma das partes utilizou de diferentes tecnologias.

O *Frontend* foi desenvolvido utilizando Javascript em conjunto com a biblioteca React [Facebook 2020], além das linguagens HTML e CSS .

Já o *Backend* foi desenvolvido como REST API [Masse 2011], utilizando a linguagem de programação Ruby com o *framework* de aplicações *Web*, Rails [Rails 2020].

Para o armazenamento dos dados utilizados e gerados em todo projeto foi utilizado o banco de dados PostgreSQL [PostgreSQL 2020].

Além das tecnologias supracitadas, também foi utilizada a tecnologia de *containers*, sendo elas Docker e Docker Compose, vez que os mesmos possibilitam portabilidade e reprodutibilidade dos processos em *containers*, garantindo efetivamente que a aplicação funcione da mesma maneira em qualquer ambiente [Docker 2020].

### 3.2 Projeto

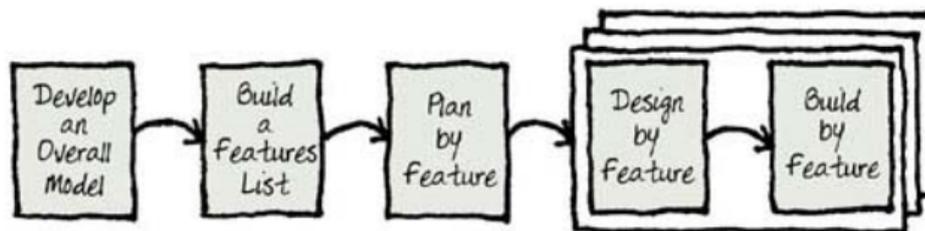


Figura 5 – Ordem das etapas de desenvolvimento no FDD [Highsmith 2002]

O SAQR (Sistema de Análise Quantitativa de Risco) tem como objetivo prover informações com intuito de auxiliar tomadas de decisões relacionadas ao tempo de entrega de produtos para uma empresa de logística.

O sistema contém duas partes principais que trabalham de maneira conjunta, foi concebido levando em consideração as etapas da metodologia de desenvolvimento ágil

*FDD* [Highsmith 2002] em ambas as frentes de desenvolvimento (*Frontend* e *Backend*) enquanto o *TDD* [Corbucci 2015] também foi aplicado durante o desenvolvimento do *Backend*. Porém as metodologias foram adaptadas conforme a necessidade do projeto, vez que o sistema foi desenvolvido individualmente.

Como pode ser visualizado na Figura 6, a primeira etapa do processo consiste em desenvolver um modelo geral da extensão do domínio do problema [Highsmith 2002].

Já a segunda etapa, consiste em criar uma lista das funcionalidades desejadas, para em seguida, como descrito na etapa seguinte, realizar o planejamento por funcionalidade. Que, por fim, foram projetadas e desenvolvidas.

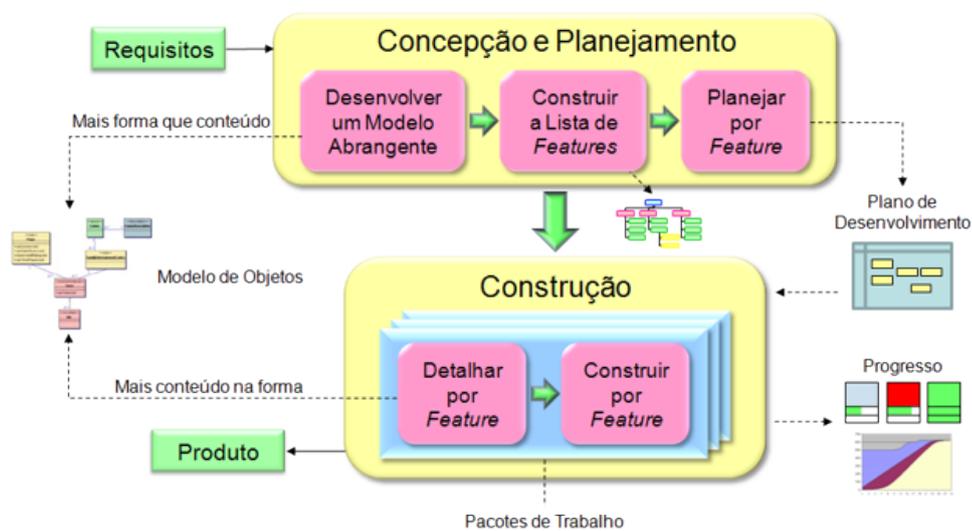


Figura 6 – Fluxo melhor detalhado do desenvolvimento no FDD [Retamal 2016]

### 3.2.1 Modelagem

Como supracitado, o projeto foi modelado com intuito de ser dividido em duas frentes, porém, para cada uma das partes do projeto, como banco de dados, aplicação do *Backend* e a aplicação *Frontend*, foram criados *containers* que se comunicam. Tal estrutura está representada na Figura 7.

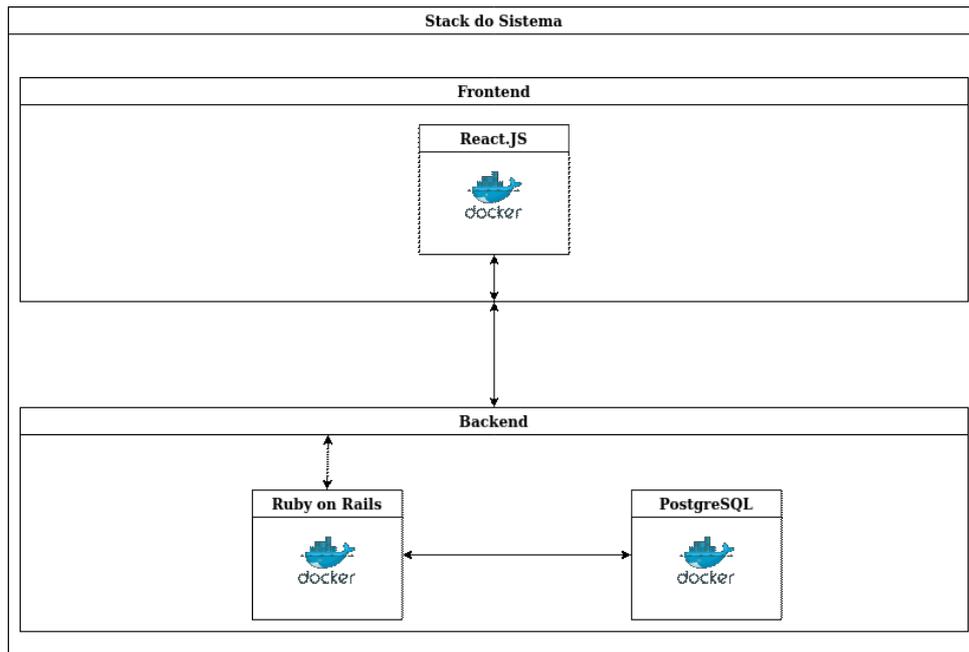


Figura 7 – Diagrama da estrutura do projeto em containers

Além disso, a modelagem foi feita de forma com que o usuário pudesse realizar 3 principais atividades no sistema, criar simulações, visualizar simulações previamente feitas e visualizar um gráfico com as informações de uma determinada simulação, como pode ser observado nos diagramas da [Figura 8.](#) e [Figura 10.](#)

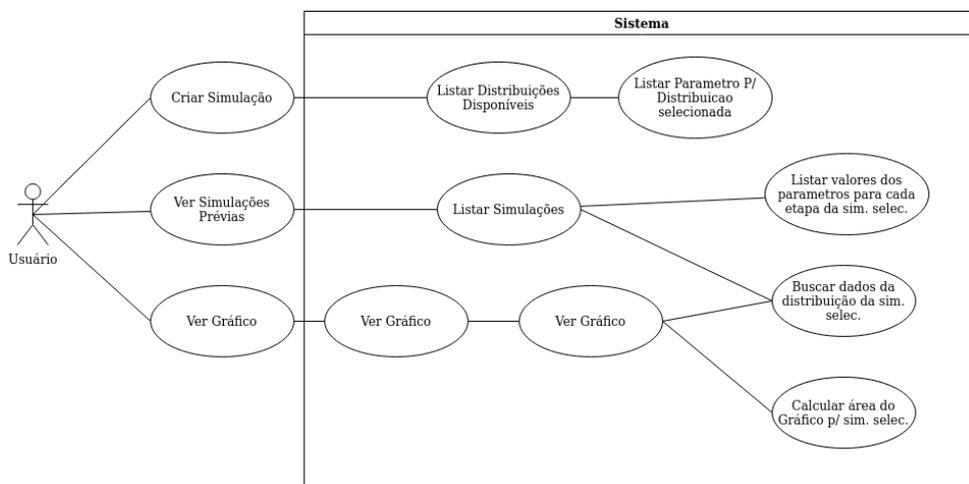


Figura 8 – Diagrama de caso de uso do projeto

Inicialmente o projeto havia sido planejado e modelado com o intuito de ser adequado para atender certa demanda de uma empresa específica, porém devido a algumas circunstâncias, durante a modelagem do mesmo, o escopo foi alterado. Contudo, mesmo algumas das tabelas não sendo utilizadas, a modelagem do banco de dados foi mantida, como foi demonstrado na [Figura 9.](#)

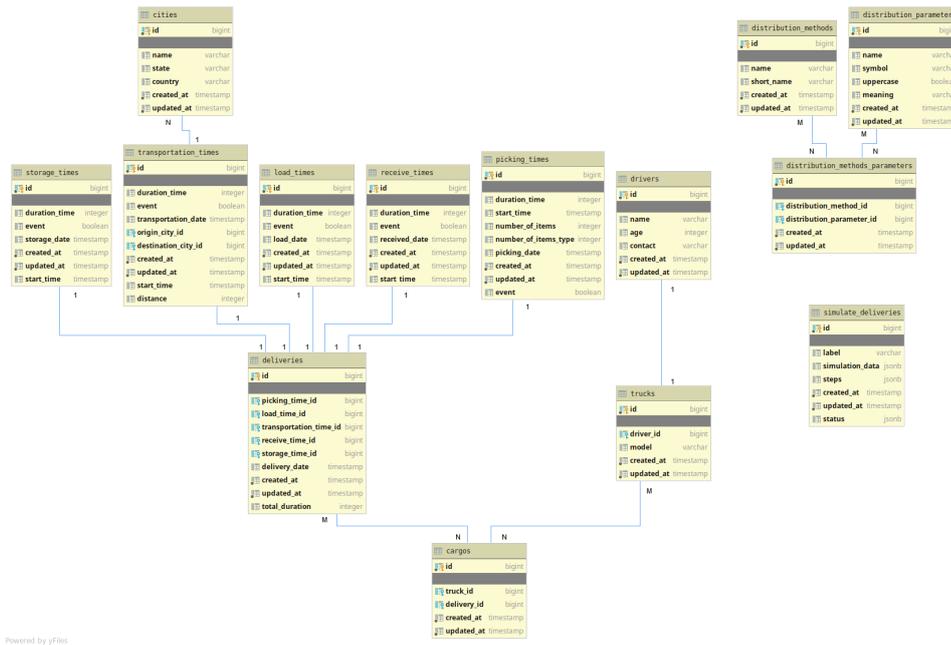


Figura 9 – Diagrama Entidade-Relacionamento do Banco de Dados

Na versão desenvolvida, as tabelas de fato utilizadas são: *"distribution\_methods"*, *"distribution\_parameters"*, *"distribution\_methods\_parameters"* e *"simulate\_deliveries"*.

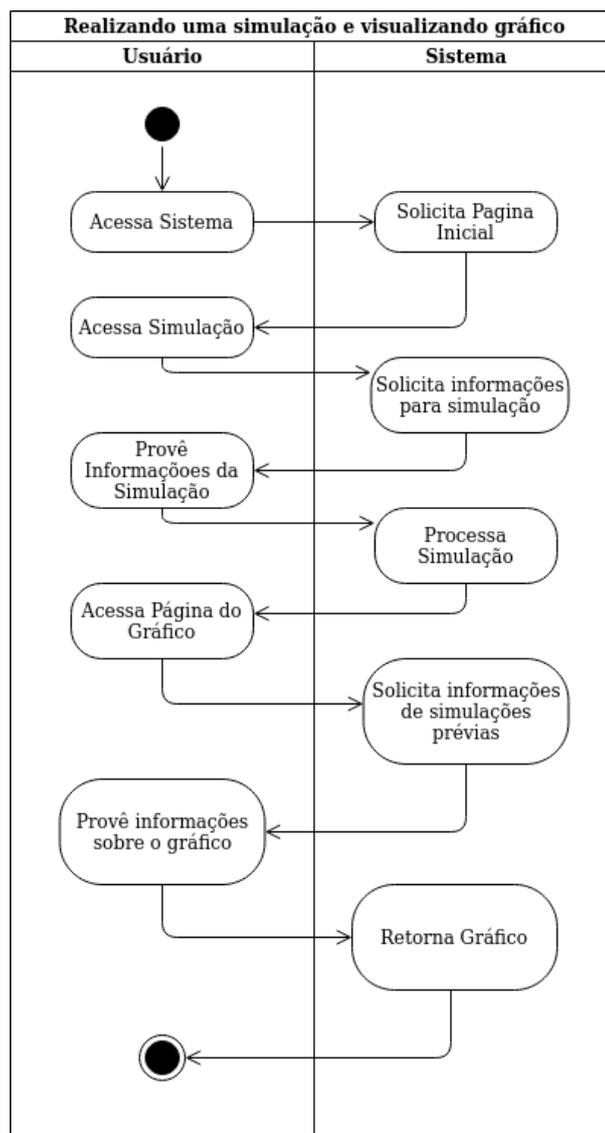


Figura 10 – Diagrama de atividades de um cenário do sistema

## 3.2.2 Requisitos do Sistema

## 3.3 Descrição do sistema

O SAQR (Sistema de Análise Quantitativa de Risco) é um sistema que pode auxiliar na tomada de decisão em relação à estimativa de tempo gasto na entrega de produtos. Ele possibilita a criação de simulações a partir de parâmetros conhecidos para cada etapa do processo de entrega, disponibilizando como resultado um gráfico de exposição ao risco de atrasos, dentre outras informações para auxiliar o usuário em sua decisão.

O projeto tem quatro seções principais:

1. *Home* - Tela Inicial
2. *Simulate* - Secção de simulação
3. *Previous Simulation* - Secção de simulações prévias
4. *Graph* - Secção do gráfico

Estas seções serão melhores detalhadas a seguir.

### 3.3.1 Tela Inicial

A tela inicial ('*Home*'), cujo modelo pode ser visualizado na [Figura 11](#), tem-se as barra explicação (em inglês e português) de cada seção presente no mesmo.

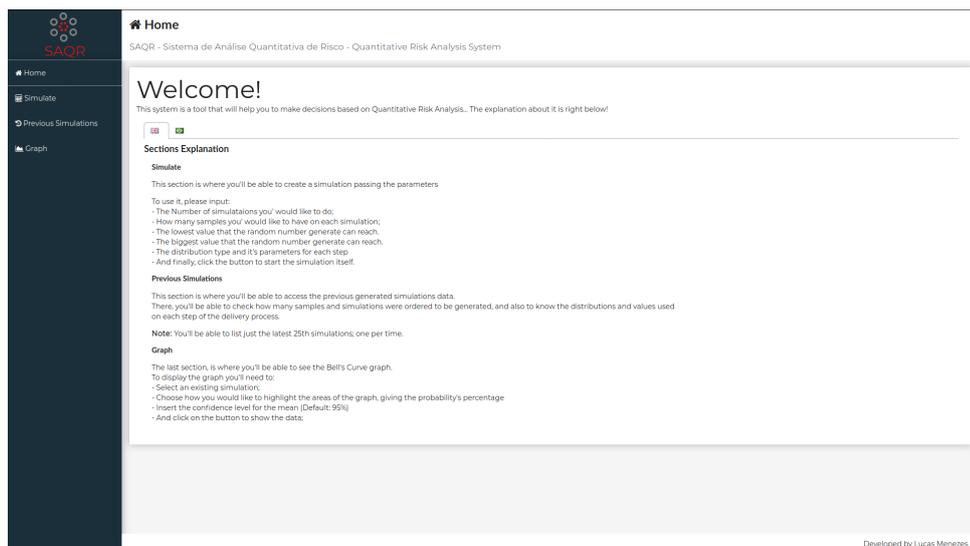


Figura 11 – Tela inicial do sistema

### 3.3.2 Secção de Simulação

Já a secção de simulação (*'Simulate'*) do sistema é onde é possível gerar novas simulações de entrega de produtos conforme apresentado na [Figura 12](#).

O tempo de entrega de produtos foi dividida em cinco etapas principais:

- *Picking Time* - Que agrupa desde o tempo do pedido ser processado até o momento que o(s) produto(s) é(são) separado(s) no galpão e está(ão) preparado(s) para ser(em) carregado(s).
- *Loading Time* - O tempo utilizado no carregamento do(s) produto(s).
- *Transportation Time* - O tempo gasto no transporte da carga.
- *Receiving Time* - O tempo despendido no recebimento do(s) produto(s).
- *Storing Time* - O tempo utilizado no descarregamento e armazenamento do(s) produto(s).

The screenshot shows a web interface for a simulation tool. On the left is a dark sidebar with navigation links: Home, Simulate, Previous Simulations, and Graph. The main content area is titled 'Simulate' and contains a form for setting simulation parameters. At the top, there are four input fields: 'Number of Simulations' (100), 'Number of samples for each simulation' (1000), 'Min. value for the seed' (100), and 'Max. value for the seed' (700). A checkbox 'Redirect to Graph Page?' is checked. Below are five sections, each for a different step: 'Step: Picking Time' (Exponential distribution, parameter  $\lambda$ ), 'Step: Loading Time' (Lognormal distribution, parameters  $\mu$  and  $\sigma$ ), 'Step: Transportation Time' (Lognormal distribution, parameters  $\mu$  and  $\sigma$ ), 'Step: Receiving Time' (Gamma distribution, parameters  $\alpha$  and  $\beta$ ), and 'Step: Storing Time' (Exponential distribution, parameter  $\lambda$ ). A 'Start Simulation' button is located at the bottom right of the form. The footer of the page reads 'Developed by Lucas Meneses'.

Figura 12 – Tela da página de simulações

Existem dois tipos de valores que devem ser inseridos antes de realizar uma simulação de estimativa de tempo de entrega: Os valores globais – são informações utilizadas para a simulação da entrega como um todo. E os valores por etapa – são informações que serão utilizados somente para gerar números aleatórios em cada uma das etapas.

Os dados globais que devem ser inseridos previamente são:

- *Number of Simulations* - Que consiste em informar quantas simulações você gostaria de realizar para a estimativa.

- *Number of samples for each simulation* - Quantas amostras cada simulação irá gerar.
- *Min.value for seed* - Qual valor mínimo a semente do gerador de número aleatório deverá utilizar.
- *Max.value for seed* - Qual valor máximo a semente do gerador de número aleatório deverá utilizar.

Já os dados específicos para etapa podem variar, pois primeiro é necessário informar o tipo de distribuição desejado e posteriormente os valores para seus respectivos parâmetros.

As distribuições disponíveis para gerar os números aleatórios são:

- Exponencial
- Lognormal
- Gamma
- Weibull

### 3.3.3 Secção de simulações prévias

Na secção de simulações prévias (*'Previous Simulation'*) existe uma caixa de seleção que você pode selecionar uma simulação da lista que contem as últimas 25 simulações que foram realizadas previamente. Assim será possível ver os parâmetros utilizados para gerar tal simulação, tais como os dados gerais como os dados específicos de cada passo.

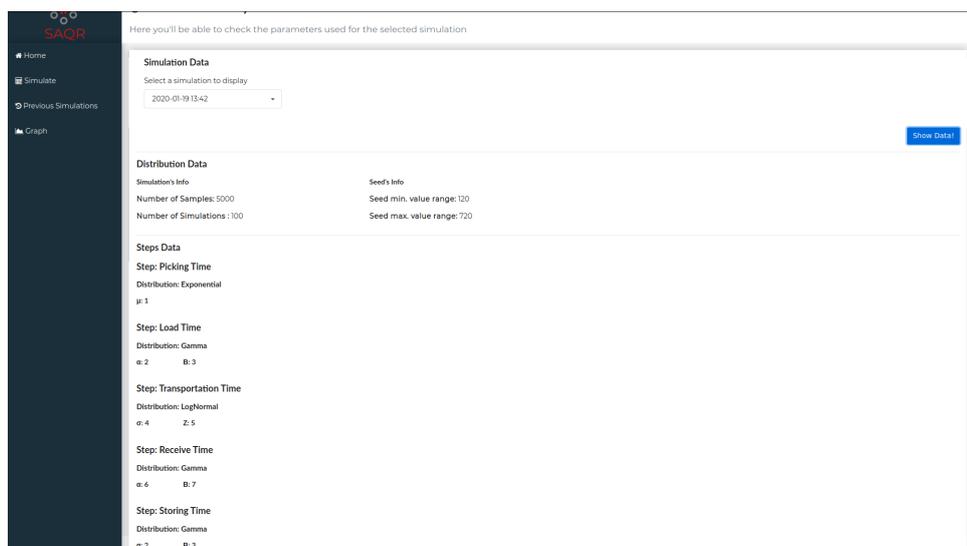


Figura 13 – Tela da secção de simulações prévias

### 3.3.4 Secção do Gráfico

A última secção do sistema é a secção do gráfico ('*Graph*'), a qual permite visualizar graficamente a simulação realizada. Para que isso seja feito, primeiramente, é necessário informar a simulação desejada, qual intervalo deverá ser destacado no gráfico e qual intervalo de confiança é o desejado para a média.

Existem duas opções de destaque para o gráfico: '*Until*' e '*Between*'.

Sobre as opções de intervalo, caso a opção '*Until*' seja escolhida, é necessário informar o valor, na qual, a área que será destacada, que será do ponto mais à esquerda do gráfico até o número inserido. Este valor em questão é tempo médio de uma entrega, por exemplo, caso o usuário inserir um número que na curva está localizado aproximadamente no meio do gráfico, então ele ficará destacado do ponto mais à esquerda até a metade do gráfico onde o valor está representado.

A mesma regra também se aplica à opção '*Between*', porém a diferença nesta opção é que devem ser inserido dois valores, o intervalo de início e fim.

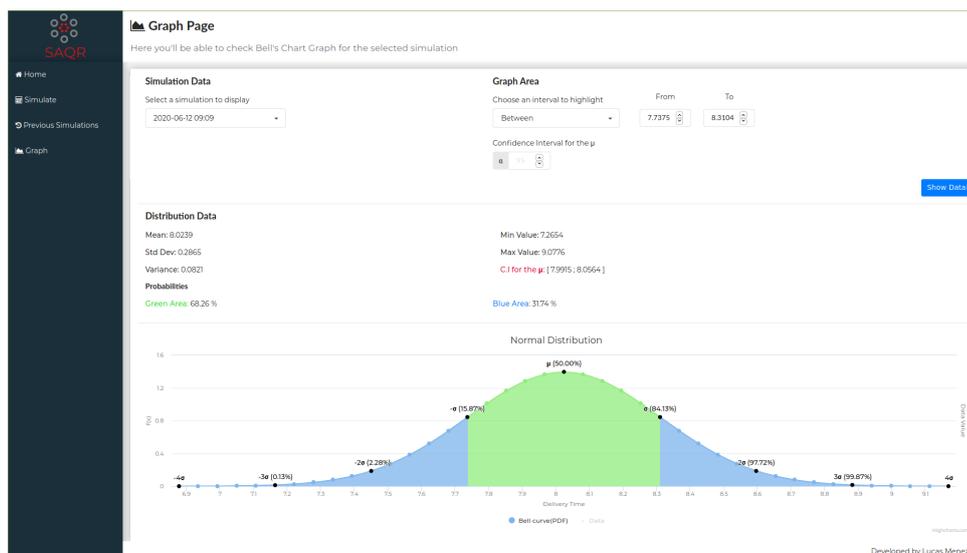


Figura 14 – Tela da secção do Gráfico

### 3.3.5 Algoritmo da simulação

Para que uma simulação seja realizada é necessário saber previamente quais são as *PDFs* que melhor se adequam para cada uma das etapas do processo de entrega, além de seus devidos parâmetros.

Além disso existem algumas outras informações que devem ser fornecidas para a realização da simulação, como:

- *Distribution Name*: Para cada uma das etapas do processo de entrega deve-se selecionar a *PDF* que melhor se adéqua aos dados e seus devidos parâmetros.
- *Number Of Simulations*: Quantas simulações de entrega deseja-se fazer.
- *Number of Sample For Each Simulation*: Quantas amostras cada etapa da simulação terá. Sendo que as etapas são: *Picking Time*, *Loading Time*, *Transportation Time*, *Receiving Time* e *Storing Time*.
- *Min value for seed*: O valor mínimo que a "semente" (*seed*) que o gerador de números aleatórios utilizará.
- *Max value for seed*: O valor máximo que a "semente" (*seed*) que o gerador de números aleatórios utilizará.

Após os serem corretamente informados, a simulação poderá ser realizada.

Para isso, inicialmente é calculado a média de cada uma das etapas de uma entrega (*Picking Time*, *Loading Time*, *Transportation Time*, *Receiving Time*, *Storing Time*). Em seguida as médias de cada uma das etapas são somadas, resultando no tempo médio de uma entrega, com isso foi realizado a simulação de uma entrega.

Estes passos são repetidos de acordo com a quantidade de vezes informadas no campo *Number Of Simulations*, e como o resultado final é uma lista contendo *N* números de simulações de tempos de entrega, que é um *array* de médias que utilizado para a construção do gráfico de distribuição normal, vez que uma população de médias tende a se aproximar de uma distribuição normal [Barbetta 2010].

As figuras [Figura 15](#) e [Figura 16](#) ilustram o código responsável pela simulação, enquanto a figura [Figura 17](#) representa este processo como um todo.

```
def self.calculate_mean_for_steps(number_of_simulations, number_of_samples, steps, star_seed_value, end_seed_value)
  final_arrays = []
  my_rng = RandomNumbersGenerator.new(star_seed_value, end_seed_value)

  steps.each do |step|
    distribution_data = step[:distribution_method]
    distribution_name = distribution_data[:name]
    distribution_parameters = distribution_data[:parameters]
    parameters_to_rng = {}

    distribution_parameters.each do |param|
      parameters_to_rng[param[:name].to_sym] = param[:value].to_f
    end

    steps_mean = []
    number_of_simulations.times do
      generated_samples = my_rng.distribution_based_generation(number_of_samples, distribution_name, parameters_to_rng)

      sumup = generated_samples.sum * 1.0
      mean = sumup / number_of_samples.to_f
      steps_mean << mean
    end

    final_arrays << steps_mean
  end

  calculate_mean_of_arrays_per_position(final_arrays)
end
```

Figura 15 – Método responsável por calcular a média das etapas de entrega

```
def self.calculate_mean_of_arrays_per_position(array_of_arrays)
  arrays_size = array_of_arrays.first.size

  array_of_arrays.each do |current_array|
    raise StandardError 'The arrays have different sizes' if current_array.size != arrays_size
  end

  # Group the array's elements per position
  transposed_array = array_of_arrays.transpose

  mean_of_arrays = transposed_array.map do |elem|
    (elem.sum / elem.size)
  end

  mean_of_arrays
end
```

Figura 16 – Método responsável por calcular a média pela posição do array

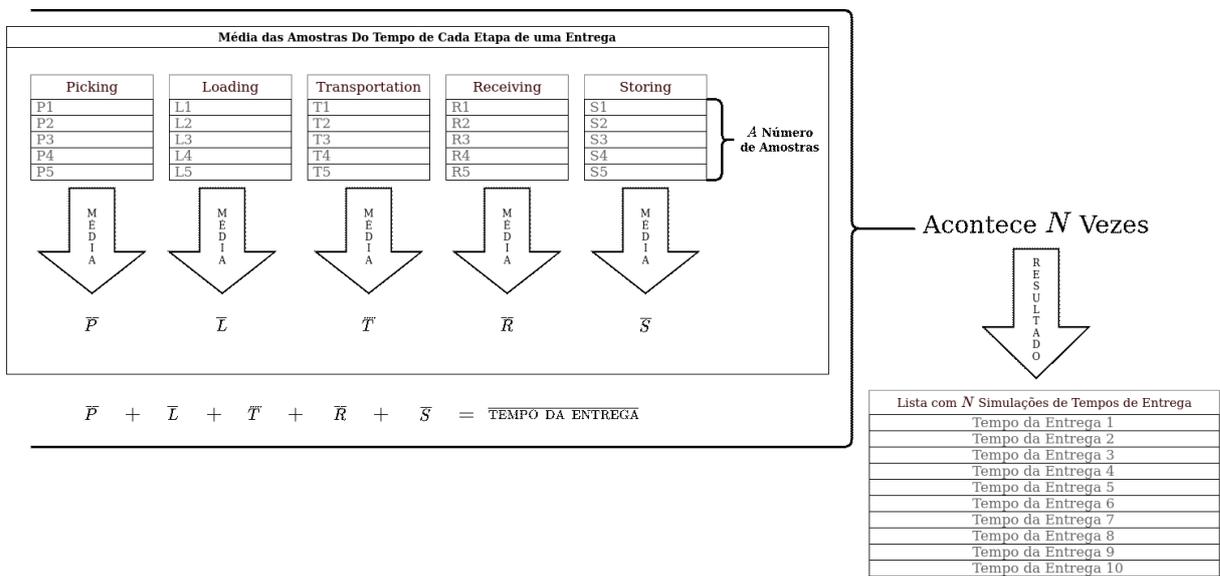


Figura 17 – Representação de parte do algoritmo média do tempo de entrega

## 3.4 Implementação

### 3.4.1 Requisitos do Sistema

Para que a prova de conceito sistema fosse passível de ser entregue, foram levantados alguns requisitos funcionais e não funcionais.

A seguir segue uma lista de alguns dos requisitos funcionais e não funcionais que foram implementados.

Lista de requisitos funcionais desenvolvidos:

- Criar página para realização da simulação
- Criar página para visualização dos parâmetros utilizados em dada simulação.
- Criar página para exibição gráfica de dada simulação
- Gerar números aleatórios baseado nas funções densidade probabilidade :
  - Gamma
  - Exponencial
  - Lognormal
  - Weibull
- Interface para usuário (*Frontend*)
- Opção de redirecionamento para página do gráfico após realização da simulação
- Opção de exibição do gráfico com diferentes formas de realce
- Realizar cálculo da média das médias dos valores gerados para cada etapa do processo de entrega

Lista de requisitos não funcionais entregues:

- *Backend* como API (para ser integrado com outras ferramentas)
- Containerização da aplicação

Porém, existem alguns requisitos, que não eram essenciais para a prova de conceito, e por tal motivo foram elencados como parte dos [trabalhos futuros](#).

### 3.4.2 Backend

O projeto do *Backend* foi desenvolvido em *Ruby on Rails* e foi implementado como uma JSON API. Como padrão das aplicações que utilizam o *framework Rails*, a arquitetura do projeto é MVC.

O banco de dados utilizado, por sua vez, foi o PostgreSQL, porém fazendo uma modificação nas configurações do *framework Rails* para que fosse possível a utilização da tecnologia *jsonb* do PostgreSQL.

Como foi apresentado anteriormente na [Figura 8](#), algumas das colunas da tabela "*simulate\_deliveries*" utilizado do tipo *jsonb*. Tal escolha foi feita devido à praticidade que a mesma provê ao trabalhar apenas com as operações de escrita e leitura somente, que é o caso destas colunas.

Por exemplo, os dados na coluna *steps* são salvos exatamente como são recebidos do *frontend*, ou seja, com o mesmo tipo de dados e na mesma formatação. O mesmo acontece quando o dado é recuperado do banco de dados, fazendo com que os dados não precisem ser tratados antes de serem retornados como resultado da requisição.

Caso fosse uma coluna do tipo *string* ou *blob*, seria preciso realizar a conversão de *json* para *string* para persistir no banco de dados e a operação inversa para enviá-la para o *frontend*.

O caso é similar na coluna "*simulation\_data*", que é a coluna que persiste os dados de uma simulação gerada, o qual deveria ser salvo em um campo *blob* por poder conter uma grande quantidade de valores gerados pela simulação.

### 3.4.3 Cálculos estatísticos

Para realizar parte dos cálculos estatísticos das simulações utilizou-se uma biblioteca chamada *Numo::GSL* [[Tanaka 2020](#)] que é uma interface para *Ruby* da biblioteca *GSL*, e uma classe que foi a base para a busca dos valores dentro da Tabela Z da distribuição normal [[Lang 2020](#)].

Os tipos de parâmetros que são utilizados em cada uma das distribuições disponíveis no sistema foram baseados na documentação da própria biblioteca [[Tanaka 2020](#)].

A tabela da distribuição normal padrão (também conhecida como Tabela Z), de acordo com [[Barbetta 2010](#)], relaciona os valores de *Z* com a área sob a curva, em outras palavras, "qualquer área (probabilidade) sob a densidade de *X* pode ser avaliada sob a densidade de *Z*." [[Barbetta 2010](#)].

A biblioteca *GSL* (*GNU Scientific Library*) [[FSF 2020](#)] é uma biblioteca numérica para desenvolvedores *C* e *C++*. Ela provê diversas funções e rotinas matemáticas como, por exemplo, geradores de números aleatórios, funções especiais entre outras.

Para o desenvolvimento dos cálculos estatísticos, foram utilizados os conceitos do livro Estatística para o curso de engenharia e informática [Barbetta 2010], tais como os conceitos de intervalo e nível de confiança, os conceitos sobre distribuição normal e tabela da distribuição normal padrão.

```

# Module to Generate Numbers based on a distribution
class RandomNumbersGenerator
  def initialize(start_value = 120, end_value = 720)
    @start_value = start_value
    @end_value = end_value
  end

  def secure_random_number
    SecureRandom.random_number(@start_value..@end_value)
  end

  def random_generator(distribution_name, distribution_parameters)
    gsl_random_generator = Numo::GSL::Rng::Rand.new(secure_random_number)

    case distribution_name
    when 'exp'
      mu = distribution_parameters[:mu].to_f
      final_val = gsl_random_generator.exponential(mu)
    when 'lognorm'
      zeta = distribution_parameters[:zeta].to_f
      sigma = distribution_parameters[:sigma].to_f
      final_val = gsl_random_generator.lognormal(zeta, sigma)
    when 'gamma'
      alpha = distribution_parameters[:alpha].to_f
      beta = distribution_parameters[:beta].to_f
      final_val = gsl_random_generator.gamma(alpha, beta)
    when 'weibull'
      alpha = distribution_parameters[:alpha].to_f
      beta = distribution_parameters[:beta].to_f
      final_val = gsl_random_generator.weibull(alpha, beta)
    else
      raise NoMethodError, "The #{distribution_name} method does not exist"
    end

    raise TypeError, "The number generated for #{distribution_name} was Null, check your parameters value" if final_val.nil?

    final_val
  end

  def distribution_based_generation(number_of_samples, distribution_name, distribution_parameters)
    number_of_samples = number_of_samples.to_i
    raise TypeError, 'number_of_samples is not a integer' unless number_of_samples.is_a? Integer

    arr = []
    number_of_samples.times do
      arr << random_generator(distribution_name, distribution_parameters)
    end
    arr
  end
end
end

```

Figura 18 – Classe responsável por gerar números aleatórios baseado em uma distribuição

### 3.4.4 Frontend

O *frontend*, por sua vez, foi desenvolvido utilizando principalmente a biblioteca *React.js* [Facebook 2020] e a biblioteca *Highcharts* [Highcharts 2020] para a criação do gráfico. Existiam diversas opções, como por exemplo *Angular*, *Vue.js*, *Ember.js* entre outros [Duomly 2020], porém *React.js* foi escolhido por ser uma biblioteca *javascript* baseada em componentes e ser bastante popular no mercado de tecnologia [Duomly 2020]. Sua principal função é a auxiliar na construção de interfaces de usuário.

Já a biblioteca *Highcharts* foi escolhida devido ter uma boa documentação de sua API, e sua utilização ser mais simples comparado a outras soluções como a biblioteca

*D3.js* [D3 2020], por exemplo. A principal funcionalidade utilizada no contexto do projeto foi principalmente a de exibição do gráfico da distribuição normal.

A arquitetura utilizada foi a padrão do *React*, utilizando de componentes para criar as páginas e pelo fato da complexidade do mesmo não ser muito elevada. Não foi necessário utilizar *Redux*, que é um contêiner de estado previsível para aplicações *javascript*, o mesmo ajuda a manter o comportamento consistente da aplicação [Redux 2020].

### 3.4.5 Contêiner

Com o intuito de facilitar a portabilidade e reprodutibilidade [Docker 2020] do sistema desenvolvido, foram criados arquivos de configuração de forma que fosse possível criar as imagens e posteriormente executar os devidos *containers* para que a *stack* da aplicação pudesse ser executada em qualquer sistema(Linux, Windows ou MacOS) com o mínimo de configurações possíveis.

Para tal foi necessário criar três arquivos de configuração diferentes, dois *Dockerfile*, um para o projeto do *frontend*, o segundo o *backend*, dentro de suas respectivas pastas, além do *Docker-compose.yml*, como representado na Figura 19.

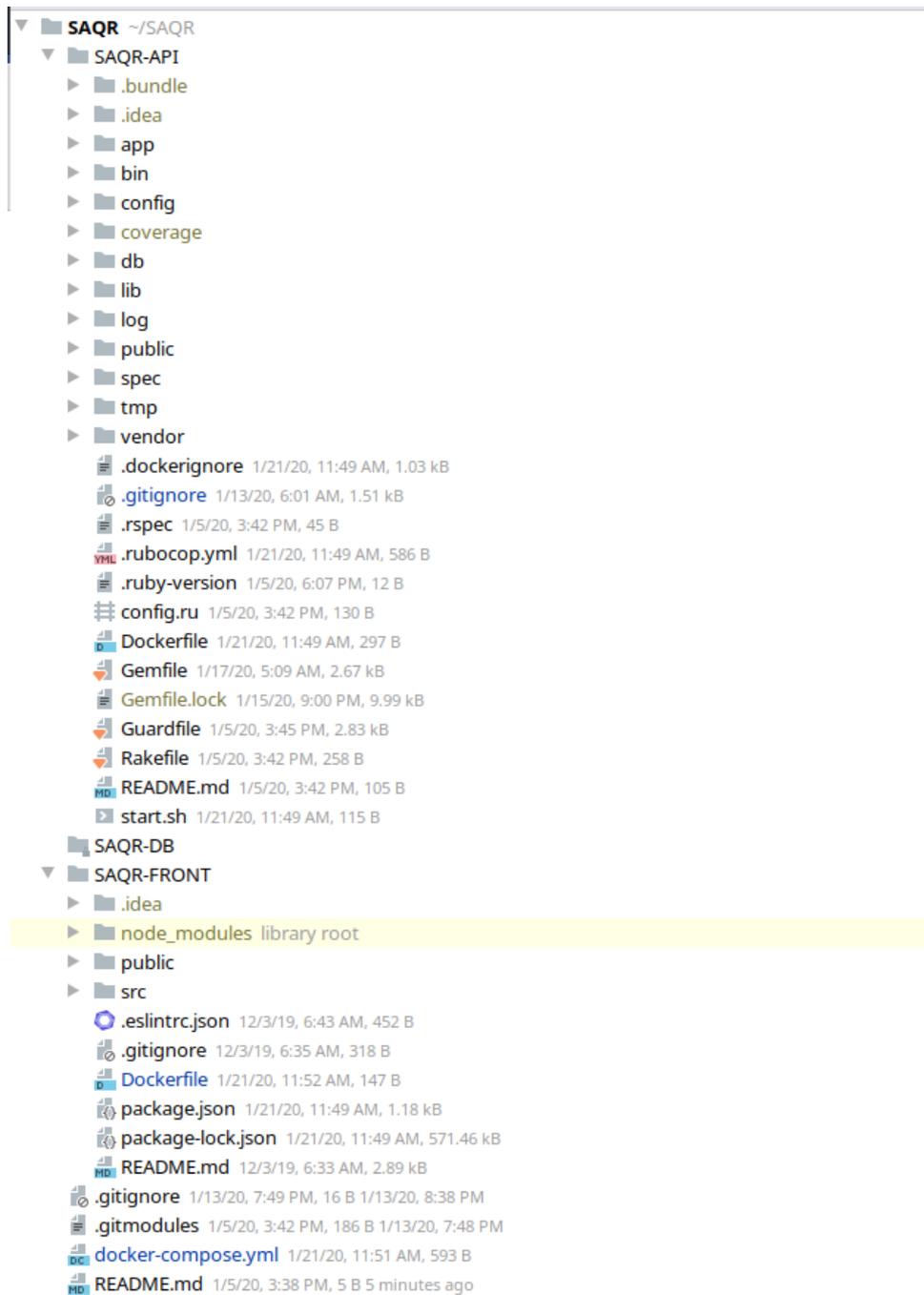


Figura 19 – Estrutura das pastas utilizando Docker Compose

```
version: '3'
services:
  saqr-db:
    image: postgres:9.5.12
    ports:
      - "5432:5432"
    volumes:
      - ~/SAQR/SAQR-DB:/var/lib/postgresql/data
    networks:
      - saqr_network

  saqr-api:
    build:
      context: ./SAQR-API
      dockerfile: Dockerfile
    ports:
      - "3000:3000"
    depends_on:
      - saqr-db
    networks:
      - saqr_network

  saqr-front:
    build:
      context: ./SAQR-FRONT
      dockerfile: Dockerfile
    ports:
      - "80:3000"
    depends_on:
      - saqr-api
    networks:
      - saqr_network

networks:
  saqr_network:
    driver: bridge
```

Figura 20 – Conteúdo do docker-compose

```
1 >> FROM ruby:2.5-buster
2
3 ENV APP_DIR=/opt/app/
4
5 WORKDIR $APP_DIR
6
7 RUN apt-get update -y && \
8 apt-get install -y \
9 ubuntu-dev-tools \
10 libpq-dev libgsl0-dev \
11 postgresql-client
12
13 COPY Gemfile* $APP_DIR
14
15 RUN bundle install
16
17 COPY . $APP_DIR
18
19 # RUN bundle config --global frozen 1
20
21 CMD ["bash", "start.sh"]
```

Figura 21 – Conteúdo do dockerfile do backend

```
1 >> FROM node:10.18.0-buster
2
3 ENV APP_DIR=/opt/app/
4
5 WORKDIR $APP_DIR
6
7 COPY package* $APP_DIR
8 RUN npm install
9
10 COPY . $APP_DIR
11
12 CMD ["npm", "start"]
13
```

Figura 22 – Conteúdo do dockerfile do frontend

## 4 Estudo de Caso

Para testar a aplicação desenvolvida foram escolhidos dois conjuntos de dados. Tais são valores hipotéticos de possíveis tempos de etapas de entregas.

As amostras, chamadas de *entrega 1* e *entrega 2*, possuem características ligeiramente diferentes em especial parâmetros de algumas etapas do processo de entrega.

### 4.1 Realização de uma simulação

A utilização do sistema é relativamente simples, e para que a mesma ocorra, basta realizar os seguinte passos:

1. Acessar a página de simulação ([Figura 11](#));
2. Inserir o valores em seus respectivos campos, como descrito no subcapítulo [Algoritmo da simulação](#) e executa-la;
3. Acessar a página do gráfico e informar os parâmetros de interesse, como descrito no subcapítulo [Secção do Gráfico](#).

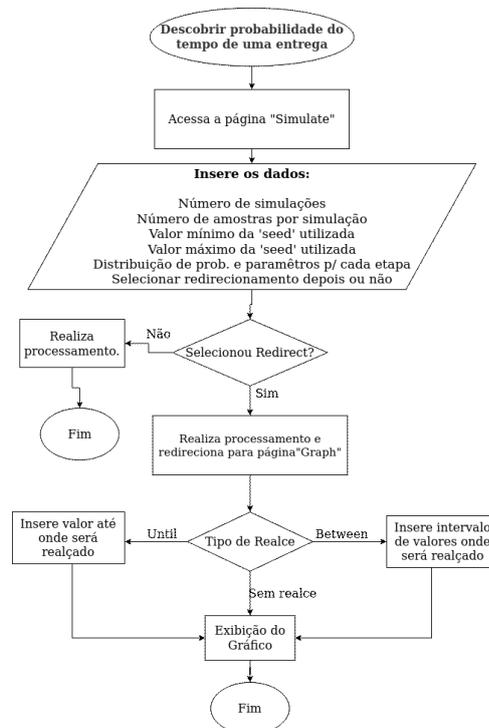


Figura 23 – Fluxograma para realização de uma simulação

## 4.2 Entrega 1

A primeira simulação de entrega foi realizada utilizando certas distribuições de probabilidade com seus respectivos parâmetros para cada uma das etapas, como pode ser observadas na [Figura 24](#).

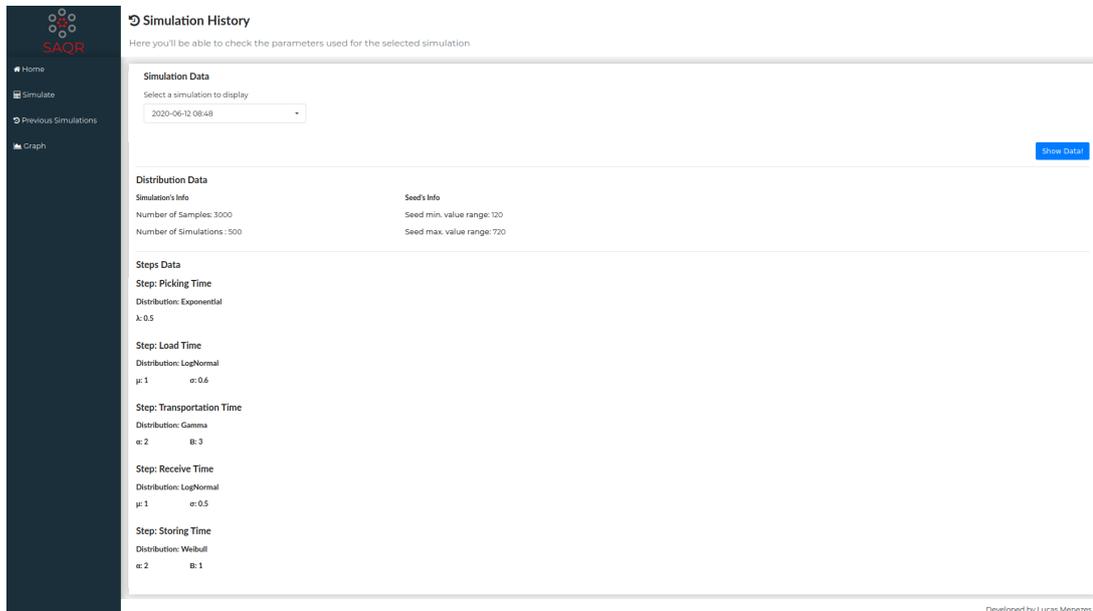


Figura 24 – Parâmetros utilizados no caso de uso da Entrega 1

O resultado da simulação pode ser verificado na página *Graph* e traz algumas informações que podem auxiliar nas tomadas de decisões, como ilustra a [Figura 25](#).

Tendo em vista o resultado da simulação realizada e os valores que a aplicação provê, é possível fazer algumas conclusões. No caso, supondo que o usuário queira saber a probabilidade do tempo da entrega demorar entre 3.2277 (Aprox. 03 horas e 13 minutos) e 3.3115 (Aprox. 03 horas e 18 minutos), é possível ver graficamente (a área verde) que a probabilidade da mesma ocorrer é de 95.43%. Além disso, existem outras informações disponíveis como tempo médio, mínimo e máximo de entrega, entre outros.

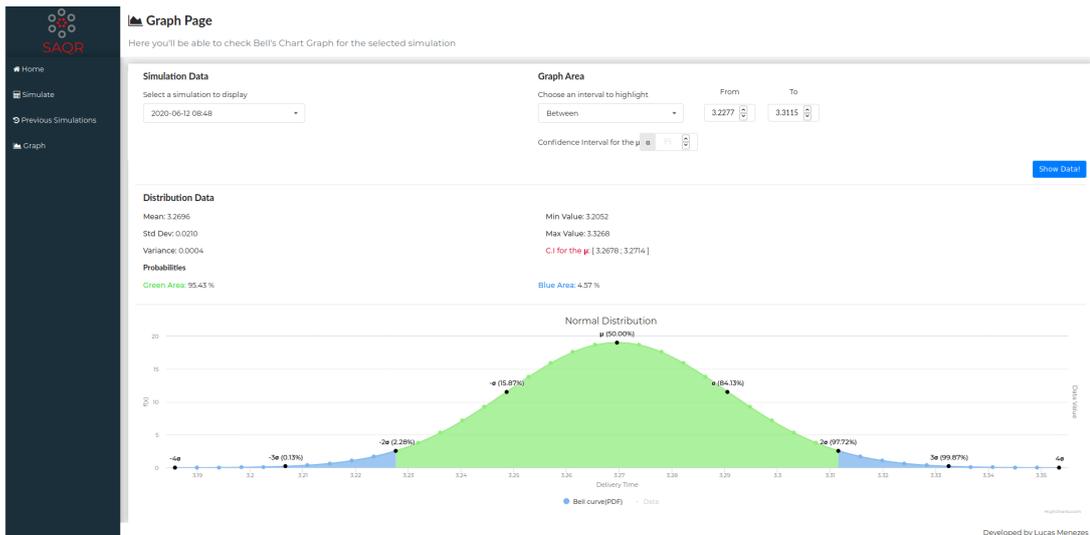


Figura 25 – Resultado gráfico da simulação do caso de uso da Entrega 1

### 4.3 Entrega 2

A segunda simulação de entrega foi realizada utilizando distribuições de probabilidade e parâmetros diferentes, emulando um outro tipo de entrega como observado na Figura 26.

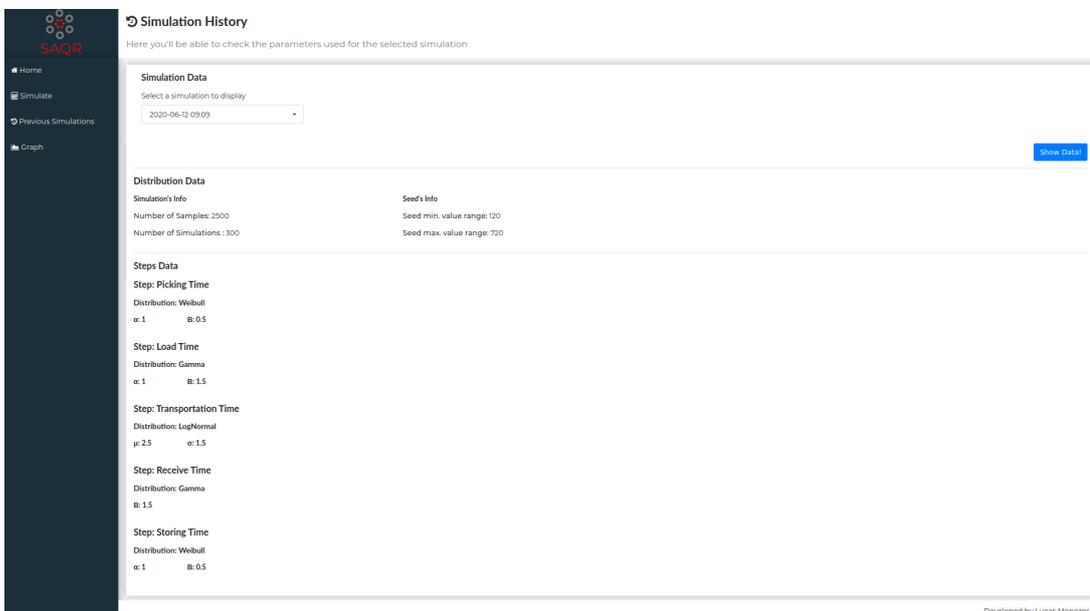


Figura 26 – Parâmetros utilizados no caso de uso da Entrega 2

O resultado da simulação da entrega 2, ilustrado a partir da Figura 27, prevê a mesma gama de informações da simulação anterior, porém com um tipo diferente de realce. No caso em questão, o usuário tinha como interesse, descobrir qual a probabilidade da entrega levar até 8.0239 (Aprox. 08 horas e 01 minuto), e como observado, a probabilidade do mesmo acontecer era de aproximadamente 50%.



Figura 27 – Resultado gráfico da simulação do caso de uso da Entrega 2

## 5 Conclusão

Durante a elaboração do presente trabalho houveram alguns desafios, como por exemplo: O estudo e a aplicação da simulação de Monte Carlo, a busca de bibliotecas de estatística em *Ruby*, e o aprendizado de *ReactJS* para a implementação de parte do projeto.

Não só, o estudo de análise quantitativa de risco, que até então era desconhecido, mas também a aplicação de tecnologias, que não haviam sido exploradas em sua totalidade durante o curso de graduação abriram as portas para uma série de conhecimentos.

O trabalho teve como foco principal uma nova ferramenta para o auxílio de tomada de decisão em relação a tempo de entrega de produtos com base em estimativas que são realizadas através de simulações.

Além do mais, uma característica relevante é que por ser um sistema de código aberto, o mesmo pode ser integrado a outras ferramentas da empresa caso necessário, podendo também ser adaptado para ser aplicado em outras áreas. Por fim, toda sua arquitetura e funcionalidades foram desenvolvidas de forma que fossem o mais intuitivo possível para o usuário.

O desenvolvimento deste trabalho foi de extrema importância para a minha compreensão da área de distribuição de probabilidade, análise de risco quantitativo, arquitetura e desenvolvimento de software.

### 5.1 Trabalhos Futuros

Apesar do projeto do sistema ter sido concluído, o mesmo encontra-se em um estágio inicial, o que admite a adição de várias novas funcionalidades que possam vir a ser úteis para outras diversas situações e implantações.

Concluindo que o objetivo de eventuais trabalhos futuros poderá consistir em criar novos recursos como:

- A implementação de testes de aderência como Kolmogorov-Smirnov, Chi-Quadrado e/ou Anderson-Darling para as atividades do processo de entrega a fim de identificar através de um histórico pré-existente qual distribuição e quais valores de seus parâmetros se adequam melhor.
- A implementação de *background jobs* a fim de garantir o processamento com maior número de simulações sem que haja *timeout* da aplicação.

- Implementação de aviso relacionado ao impacto do número de simulações e o resultado gerado.
- Implementação de um sistema de usuário com autenticação
- Adição de novas distribuições para a simulação de entregas
- Melhorias na acessibilidade do sistema e validação de dados

# Referências

- ANICHE, M. *Test-Driven Development*. [S.l.], 2014. Disponível em: <<http://tdd.caelum.com.br/>>. Citado na página 14.
- AXELOS. *Whats is PRINCE2?* [S.l.], 2016. Disponível em: <<https://www.axelos.com/best-practice-solutions/prince2/what-is-prince2>>. Citado na página 11.
- BARBETTA, P. A. A. C. B. M. M. R. *Estatística: para cursos de engenharia e informática*. 3. ed. [S.l.: s.n.], 2010. 153 p. Citado 2 vezes nas páginas 5 e 20.
- BARBETTA, P. A. A. C. B. M. M. R. *Estatística: para cursos de engenharia e informática*. 3. ed. [S.l.: s.n.], 2010. 155 p. Citado 2 vezes nas páginas 5 e 21.
- BARBETTA, P. A. A. C. B. M. M. R. *Estatística: para cursos de engenharia e informática*. 3. ed. [S.l.: s.n.], 2010. 186-187 p. Citado na página 19.
- BARBETTA, P. A. A. C. B. M. M. R. *Estatística: para cursos de engenharia e informática*. [S.l.: s.n.], 2010. Citado 2 vezes nas páginas 20 e 36.
- BARBETTA, P. A. A. C. B. M. M. R. *Estatística: para cursos de engenharia e informática*. 3. ed. [S.l.: s.n.], 2010. 155-156 p. Citado 2 vezes nas páginas 32 e 35.
- COOKE, T. B. R. *Probabilistic risk analysis : foundations and methods*. 1. ed. [S.l.]: Cambridge University Press, 2001. ISBN 0521773202,9780521773201. Citado na página 16.
- CORBUCCHI, M. A. H. *Test Driven Development: Teste e design no mundo real com Ruby*. 1. ed. [S.l.]: Casa Do Código, 2015. Citado 2 vezes nas páginas 9 e 24.
- D3, C. *Data-Driven Documents*. [S.l.], 2020. Disponível em: <<https://d3js.org/>>. Citado na página 37.
- DEKKING C. KRAAIKAMP, H. L. L. M. F. *A Modern Introduction to Probability and Statistics: Understanding Why and How*. [S.l.]: Springer, 2005. (Springer Texts in Statistics). Citado na página 20.
- DOCKER. *Orientation and Setup | Docker Documentation*. [S.l.], 2020. Disponível em: <<https://docs.docker.com/get-started/>>. Citado 2 vezes nas páginas 23 e 37.
- DUOMLY. *The best front-end framework to learn in 2019*. [S.l.], 2020. Disponível em: <<https://dev.to/duomly/the-best-front-end-framework-to-learn-in-2019-dn7>>. Citado na página 36.
- FACEBOOK, I. *Introdução - React*. [S.l.], 2020. Disponível em: <<https://pt-br.reactjs.org/docs/getting-started.html>>. Citado 3 vezes nas páginas 13, 23 e 36.
- FLORES-RIOS, B. L.; RODRÍGUEZ-ELIAS, O. M.; OLGUÍN-ESPINOZA, M. Comparative analysis of the iso 10006: 2003 against other standards and guidelines related to software project management. *Research in Computing Science*, v. 43, p. 55–67, 2009. Disponível em: <[http://negro.iing.mx/~/bfloros/gc/paperISO10006\\_others.pdf](http://negro.iing.mx/~/bfloros/gc/paperISO10006_others.pdf)>. Citado na página 11.

- FSF, F. S. F. I. *GSL - GNU Scientific Library*. [S.l.], 2020. Disponível em: <<https://www.gnu.org/software/gsl/>>. Citado 2 vezes nas páginas 22 e 35.
- GENTILE, S. *Project Management in the Information Technology Industry*. [S.l.], 2012. Disponível em: <[http://projectmgmt.brandeis.edu/downloads/BRU\\_MSMPP\\_WP\\_May2012\\_Project\\_Management\\_Information\\_Technology\\_Industry.pdf](http://projectmgmt.brandeis.edu/downloads/BRU_MSMPP_WP_May2012_Project_Management_Information_Technology_Industry.pdf)>. Citado na página 9.
- GENTLE, J. E. *Random Number Generation and Monte Carlo Methods*. 2nd ed. ed. [S.l.]: Springer, 2003. (Statistics and computing). ISBN 0387001786,9780387216102,9780387001784. Citado na página 18.
- GONDIM, B. E. M. de Albuquerque Filho; Bruno Rodrigo Cunha de A. C. C. F. L. M. F. F. M. *Comparação da ISO 10006 com o PMBOK: Qualidade de Software*. [S.l.], 2005. Disponível em: <[http://www.cin.ufpe.br/~ccc2/QoS/MONOGRAFIA\\_qualidade\\_ISO10006xPMBOK%5BVersaoFinal%5D.doc](http://www.cin.ufpe.br/~ccc2/QoS/MONOGRAFIA_qualidade_ISO10006xPMBOK%5BVersaoFinal%5D.doc)>. Citado na página 11.
- HIGHCHARTS. *Highcharts Javascript Charting Library*. [S.l.], 2020. Disponível em: <<https://www.highcharts.com/products/highcharts/>>. Citado na página 36.
- HIGHSMITH, J. *Agile Software Development Ecosystems*. [S.l.]: Addison-Wesley, 2002. (The Agile software development series). ISBN 0201760436,9780201760439. Citado 4 vezes nas páginas 5, 13, 23 e 24.
- IPMA. *What is IPMA?* [S.l.], 2016. Disponível em: <<http://www.ipma.world/about/answers/>>. Citado na página 11.
- ISO. *ISO 10006:2003 Quality management systems Abstract - Guidelines for quality management in projects*. [S.l.], 2016. Disponível em: <[http://www.iso.org/iso/home/store/catalogue\\_tc/catalogue\\_detail.htm?csnumber=36643](http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=36643)>. Citado na página 11.
- LANG, M. *Ruby Z-Table lookup class. Given a z-score, return approximate percentile rank*. [S.l.], 2020. Disponível em: <<https://gist.github.com/mwlang/74ef127be521f8ecc34e7201472db61c>>. Citado na página 35.
- LAW, W. D. K. A. M. *Simulation Modeling and Analysis*. 5. ed. [S.l.]: McGraw-Hill, 2014. 393-397 p. ISBN 978-0-07-340132-4. Citado na página 21.
- MACIEL, D. de M. A. *Benefícios da Gestão de Riscos em Projetos*. [S.l.], 2016. Disponível em: <[http://www.techoje.com.br/site/techoje/categoria/detalhe\\_artigo/1146](http://www.techoje.com.br/site/techoje/categoria/detalhe_artigo/1146)>. Citado na página 9.
- MASSE, M. *REST API Design Rulebook: Designing Consistent RESTful Web Service Interfaces*. O'Reilly Media, 2011. ISBN 9781449319908. Disponível em: <<https://books.google.com.br/books?id=eABpyzTcJNIC>>. Citado na página 23.
- MATIAS, R. *Análise quantitativa de risco baseada no método de monte carlo: abordagem pmbok*. v. 29, 2006. Disponível em: <[https://www.researchgate.net/publication/228773958\\_Analise\\_Quantitativa\\_de\\_Risco\\_Baseada\\_no\\_Metodo\\_de\\_Monte\\_Carlo\\_Abordagem\\_PMBOK](https://www.researchgate.net/publication/228773958_Analise_Quantitativa_de_Risco_Baseada_no_Metodo_de_Monte_Carlo_Abordagem_PMBOK)>. Citado 5 vezes nas páginas 5, 15, 17, 18 e 19.
- NAZÁRIO, P. *A importância de sistemas de informação para a competitividade logística*. [S.l.], 1999. v. 5. Disponível em: <<http://www.tecspace.com.br/paginas/aula/faccamp/TI/Texto04.pdf>>. Citado na página 16.

PICININ JOÃO LUIZ KOVALESKI, L. S. C. T. *Gerenciamento de risco e rastreadores apoiando no processo de gestão da logística: Um estudo de caso*. [S.l.], 2016. Disponível em: <[http://pg.utfpr.edu.br/dirppg/ppgep/ebook/2009/CONGRESSOS/Nacionais/2009%20-%20SIMPEP/XVI\\_SIMPEP\\_Art\\_9\\_a.pdf](http://pg.utfpr.edu.br/dirppg/ppgep/ebook/2009/CONGRESSOS/Nacionais/2009%20-%20SIMPEP/XVI_SIMPEP_Art_9_a.pdf)>. Citado na página 17.

PMI. *Guide to the project management body of knowledge*. 3. ed. [S.l.]: USA: Project Management Institute, 2004. Citado na página 15.

PMI. Gerenciamento dos riscos do projeto. In: \_\_\_\_\_. *Guia PMBOK®: Um guia do conhecimento em gerenciamento de projetos*. 5. ed. [S.l.: s.n.], 2013. cap. 11, p. 309–354. Citado na página 12.

PMI. *Guia PMBOK®: Um guia do conhecimento em gerenciamento de projetos*. 5. ed. [S.l.]: EUA: Project Management Institute, 2013. Citado 3 vezes nas páginas 10, 11 e 13.

PMI. *PMBOK Guide & Standards*. [S.l.], 2016. Disponível em: <<http://www.pmi.org/PMBOK-Guide-and-Standards.aspx>>. Citado na página 11.

POSTGRESQL. *PostgreSQL: About*. [S.l.], 2020. Disponível em: <<https://www.postgresql.org/about/>>. Citado 2 vezes nas páginas 14 e 23.

RAILS, C. *Ruby on Rails*. [S.l.], 2020. Disponível em: <<http://rubyonrails.org/>>. Citado 3 vezes nas páginas 13, 14 e 23.

REDUX, C. *Getting Started with Redux*. [S.l.], 2020. Disponível em: <<https://redux.js.org/introduction/getting-started>>. Citado na página 37.

RETAMAL, A. M. *Fluxo do FDD*. [S.l.], 2016. Disponível em: <<http://heptagon.com.br/fdd>>. Citado 2 vezes nas páginas 5 e 24.

RUBY, C. *Ruby : A Programmer's best friend*. [S.l.], 2016. Language creator: Yukihiro “Matz” Matsumoto. Disponível em: <<https://www.ruby-lang.org/>>. Citado na página 14.

SILVA, D. P. M. A. *Introdução A Logística*. 9. ed. [S.l.], 2017. Disponível em: <[http://portal.unisepe.com.br/unifa/wp-content/uploads/sites/10001/2018/06/036\\_logistica.pdf](http://portal.unisepe.com.br/unifa/wp-content/uploads/sites/10001/2018/06/036_logistica.pdf)>. Citado na página 9.

TANAKA, M. *GSL interface for Ruby with Numo::NArray*. [S.l.], 2020. Disponível em: <<https://github.com/ruby-numo/numo-gsl>>. Citado na página 35.

TEAM, R. D. *RSpec*. [S.l.], 2016. Disponível em: <<http://rspec.info/>>. Citado na página 14.

VOSE, D. *Risk Analysis - A Quantitative Guide*. 3. ed. [S.l.]: Wiley, 2008. ISBN 0470512849,9780470512845. Citado na página 15.

WIDEMAN, R. M. Comparing prince2® with pmbok®. *AEW Services, Vancouver*, 2002. Disponível em: <[http://www.logro.sk/na\\_stiahnutie/project\\_management\\_-\\_comparing\\_prince2\\_with\\_pmbok.pdf](http://www.logro.sk/na_stiahnutie/project_management_-_comparing_prince2_with_pmbok.pdf)>. Citado na página 11.