PROJETO DE GRADUAÇÃO

APLICAÇÃO DE MODELOS DE REGRESSÃO LINEAR PARA A PREVISÃO DE RESULTADOS EM PROJETOS ÁGEIS DE DESENVOLVIMENTO DE SOFTWARE EM UMA EMPRESA DO MERCADO FINANCEIRO

Por,

Pedro Machado Gomes

Brasília, Setembro de 2024

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO

UNIVERSIDADE DE BRASÍLIA

Faculdade de Tecnologia

Departamento de Engenharia
de Produção

PROJETO DE GRADUAÇÃO

APLICAÇÃO DE MODELOS DE REGRESSÃO LINEAR PARA A PREVISÃO DE RESULTADOS EM PROJETOS ÁGEIS DE DESENVOLVIMENTO DE SOFTWARE EM UMA EMPRESA DO MERCADO FINANCEIRO

Por,

Pedro Machado Gomes

Relatório submetido como requisito parcial para obtenção do grau
de Engenheiro de Produção.

Banca Examinadora

Prof. Dr. Sanderson César Macêdo Barbalho

Brasília, Setembro de 2023

Agradecimentos

Agradeço, primeiramente, aos meus pais, Patrícia e Moisés, que são os meus maiores exemplos de vida e que sempre me apoiaram em todas as minhas decisões. Vocês são parte essencial do que eu sou e não conseguiria chegar até aqui sem vocês. Obrigado por me apoiarem e me incentivarem sempre! Meu sonho de ser engenheiro partiu de vocês, minhas maiores inspirações!

Agradeço também aos meus avós, Wander e Conceição, por serem grandes referências de vida e por sempre me acolherem e incentivarem tanto. Em memória dos meus outros avós, Margarida e Moisés, que infelizmente não verão o seu primeiro neto se formar, mas que sempre estiveram ao meu lado em todos os momentos. E ao meu primo Vítor, por ser o irmão que eu nunca tive e por estar presente em todos os momentos da minha vida. Amo vocês e obrigado por todo o apoio!

Aos meus amigos de graduação (Rafael, Gustavo, Daniela, Gabriel, Thaís, Pedro O., Pedro F., Arthur e José) pela parceria nos momentos bons e apoio nos momentos mais difíceis. Sou muito mais feliz por ter vivido tudo isso ao lado de vocês! Vocês foram essenciais para mim ao longo de todos esses anos. Ansioso por tudo que a vida ainda guarda para nós!

Ao meu orientador Professor Sanderson, por ter me acompanhado e orientado no meu último desafio da graduação. Obrigado pela paciência e disponibilidade sempre!

À Universidade de Brasília por ter sido a grande escola da minha vida. Aprendi nela muito mais do que as matérias de engenharia e me desenvolvi como cidadão durante todos esses anos. Obrigado a todos que fazem essa instituição existir!

Pedro Machado Gomes

RESUMO

Cada vez mais as empresas gastam tempo e recursos no desenvolvimento de softwares, seja para criar interfaces de atendimento ou venda para clientes, seja para apoio de seus processos internos. As empresas que desempenham essas atividades com uma maior eficiência e previsibilidade, possuem uma enorme vantagem competitiva no mercado. O presente estudo busca identificar na literatura os principais modelos e tendências na área de previsão de resultados em projetos de desenvolvimento ágil de softwares. Ademais, pretende-se explorar a relação das variáveis presentes em projetos de desenvolvimento de software de uma empresa do mercado financeiro brasileiro e propor um modelo de regressão linear capaz de prever parâmetros desses projetos. A condução do estudo revelou os principais comportamento das relações entre as variáveis presentes nos projetos da empresa e construiu um modelo estatisticamente adequado de previsão de resultados para eles.

Palavras-chave: Desenvolvimento de Software; Previsão de Resultados; Regressões Lineares; Projetos Ágeis.

ABSTRACT

Increasingly, companies are investing time and resources in software development, whether to create customer services or interfaces, or to support their internal processes. Companies that carry out these activities with greater efficiency and predictability have a significant competitive advantage in the market. This study aims to identify the main models and trends in the literature about outcome forecasting in agile software development projects. Furthermore, it seeks to explore the relationship between variables present in software development projects of a Brazilian financial market company and propose a linear regression model capable of predicting parameters of these projects. The study revealed the main behaviors of relationships between variables present in the company's projects and developed a statistically adequate model for forecasting results.

Keywords: Software Development; Outcome Forecasting; Linear Regression; Agile Projects.

Lista de Figuras

Lista de Tabelas

Sumário

1. Introdução

1.1. Justificativa

Nas últimas décadas, o mundo todo vivenciou grandes mudanças e evoluções devido ao intenso processo de globalização. Fatores como o surgimento de novas tecnologias, aumento da concorrência na escala global e a diminuição do ciclo de vida dos produtos, elevaram significativamente a complexidade da atuação de diversas empresas (SEIDELMANN, 2018); (TALIAFERRO, 2016).

Este contexto de grande complexidade imposto pelo ambiente empresarial cada vez mais competitivo, exige das empresas uma maior capacidade de planejar e gerenciar suas operações e processos (AZANHA, 2015). Essa maior exigência é aplicada a diversos processos dentro das empresas, principalmente a aqueles que necessitam de uma grande alocação de recursos, como a produção de softwares.

Segundo (SOMMERVILLE, 2011), a utilização de técnicas estruturadas de desenvolvimento de software (engenharia de software) garante a produção de sistemas de forma muito mais rápida e barata para as empresas, evitando erros nos softwares e impedindo desperdício de recursos em atividades de correção. Isso dá as empresas que se utilizam de métodos estruturados de desenvolvimento de software, uma grande vantagem frente às demais.

Além disso, o mercado brasileiro de TI somou em 2023 um total de R$ 247,4 bilhões de reais em investimentos nas áreas de software, serviços de TI e hardwares, compreendendo cerca de 4,5% do PIB nacional. Os resultados apresentados no ano passado garantiram ao país a 12ª posição no ranking de investimentos em TI no mundo (ASSOCIAÇÃO BRASILEIRA DAS EMPRESAS DE SOFTWARE, 2023). Números dessa magnitude no setor, demonstram como é importante a busca por evolução nos processos de produção de software, evitando ao máximo o desperdício de recursos e garantindo a melhor qualidade possível nos sistemas.

O contexto atual da chamada indústria 4.0, ou transformação digital, ou ainda digitalização e conectividade da economia (Corrêa et al., 2022) traz como forte tendência a perspectiva de construção de modelos preditivos que permitam estimar com maior exatidão o tempo necessário (Barbalho et al., 2022) ou o custo envolvido em projetos de novos produtos de hardware e software.

Desta forma, o presente estudo busca analisar os dados de produção de software de uma empresa do mercado financeiro brasileiro. Explorando as diversas variáveis envolvidas nos processos de desenvolvimento e propondo um modelo estatístico de previsão de resultados para os projetos. Portanto, o estudo pretende contribuir para o entendimento geral das relações entre as variáveis presentes em um projeto de desenvolvimento de sistemas, apresentando os resultados obtidos pelo modelo e sugestões de trabalhos futuros para uma melhor compreensão dos fenômenos observados.

1.2. Objetivos

1.2.1. Objetivo Geral

O estudo tem como objetivo, explorar as variáveis presentes no processo de desenvolvimento de software de uma empresa do mercado financeiro e propor um modelo estatístico para a previsão dos tempos de ciclo dos projetos/portfólios de desenvolvimento.

1.2.2. Objetivos Específicos

- Analisar a literatura existente sobre o tema e levantar os principais modelos de previsão utilizados para a produção de software;
- Aplicar métodos estatísticos para o devido uso e análise de dados;
- Identificar, observar e descrever as relações entre as principais variáveis presentes em projetos de desenvolvimento de softwares;
- Propor um modelo estatístico para previsão de resultados de projetos de software e compará-lo aos fenômenos observados na realidade.

1.3. Estrutura do Trabalho

O seguinte trabalho está organizado em cinco capítulos diferentes. O primeiro capítulo, de introdução, compreende a justificativa do estudo e seu objetivos, gerais e específicos. Já o segundo, de metodologia, apresenta a classificação acadêmica do estudo desenvolvido e os principais procedimentos metodológicos utilizados ao longo de suas diferentes etapas. No segundo capítulo são ainda expostos os procedimentos de condução do trabalho, pesquisa bibliométrica, acesso e coleta de dados, análise e manipulação dos dados e cálculo do índice de senioridade utilizado no modelo.

O terceiro capítulo, de referencial teórico, traz os principais conceitos necessários para entender e desenvolver o trabalho. Nele, são expostos os conceitos de Engenharia de Software, Processos de Desenvolvimento de Software, Aplicação de Metodologias Ágeis no Desenvolvimento de Software e Métodos de Regressão Linear e seus detalhamentos.

O quarto capítulo, de resultados, traz todo o detalhamento dos procedimentos realizados ao longo da análise dos dados e da montagem do modelo de previsão. Além disso, traz os resultados obtidos através da aplicação dos modelos criados. O último capítulo, de ponderações finais, expõe as conclusões obtidas, sugestões de trabalhos futuros complementares e as lições aprendidas ao longo da condução do estudo.

2. Metodologia

2.1. Classificação da Pesquisa

Para (GIL, 2008), pesquisa é um processo formal e sistemático de desenvolvimento do método científico. Ela tem o objetivo central de descobrir respostas para problemas gerais mediante a aplicação de procedimentos científicos.

Ainda sob essa ótica, segundo (ALMEIDA, 2014), para uma pesquisa ser considerada científica, necessariamente ela precisa ser elaborada através de métodos e procedimentos. É necessário que ela adote métodos de maneira padronizada e detalhada, de forma que qualquer outro pesquisador consiga seguir os mesmos passos, podendo replicar e aperfeiçoar a pesquisa utilizando os mesmos dados como insumo.

Segundo (SILVA; MENEZES, 2001), as pesquisas científicas podem ser classificadas de acordo com o modelo clássico, em que são categorizados os campos de natureza da pesquisa, abordagem do problema, objetivos da pesquisa e procedimentos técnicos utilizados. Cada um dos campos possuí as suas próprias subclassificações de acordo com as características da pesquisa em questão.

Sob a ótica de natureza da pesquisa, este trabalho se enquadra na **natureza aplicada** de um estudo científico. Isso se dá pois o trabalho objetiva gerar conhecimentos para a aplicação prática em problemas específicos e reais. Este tipo de natureza se baseia em verdades e interesses locais, ao contrário da natureza básica, que parte de conceitos universais (SILVA; MENEZES, 2001).

Tomando como parâmetro os conceitos apresentados por (SILVA; MENEZES, 2001), o presente estudo parte de uma **abordagem quantitativa** do problema. Abordagens quantitativas trabalham com variáveis que podem ser quantificáveis, traduzindo opiniões e informações em números para classificação e análise. Para aferir os resultados deste tipo de abordagem, utilizam-se recursos e técnicas estatísticas como porcentagem, média, moda, desvio padrão, coeficiente de correlação etc.

Em relação ao campo de objetivos de pesquisa, este estudo se enquadra na **categoria explicativa**. Estudos de categoria explicativa visam identificar fatores

que determinam ou contribuem para a ocorrência de fenômenos. Eles aprofundam o conhecimento da realidade e o porquê das coisas e acontecimentos. Normalmente assume a forma de pesquisa experimental ou pesquisa expost-facto em termos de procedimentos técnicos (SILVA; MENEZES, 2001).

Do ponto de vista de procedimentos técnicos de GIL (2002), o trabalho é classificado como uma **pesquisa "*Expost-Facto*"**. Essa classificação é dada para estudos em que o experimento é realizado depois dos acontecimentos, de maneira posterior ao recolhimento de dados do fato (SILVA; MENEZES, 2001)

2.2. Procedimento metodológico

2.2.1. Procedimento de Condução do Trabalho

De acordo com (SILVA; MENEZES, 2001), o planejamento e a execução de uma pesquisa científica fazem parte de um processo sistematizado com um total de dez etapas. As fases do processo de condução de uma pesquisa científica são sequenciais e dependentes das fases anteriores do fluxo, cada uma possuí seus respectivos outputs e objetivos.

- **Etapa 1 - Escolha do Tema:** Fase de escolha do aspecto ou área de interesse do trabalho. Estabelecendo os limites e restrições para o desenvolvimento do estudo;
- **Etapa 2 – Revisão de Literatura:** Etapa de análise dos trabalhos sobre o tema já existentes. É uma fase essencial para identificar que aspectos já foram abordados, quais lacunas existem na literatura e evita duplicidades nas pesquisas;
- **Etapa 3 – Escolha da Justificativa:** Fase de identificação das razões para a escolha do tema específico, analisando a importância da pesquisa em relação a outros temas;
- **Etapa 4 – Formulação do Problema:** Etapa de detalhamento do problema que será resolvido com o desenvolvimento da pesquisa;
- **Etapa 5 – Determinação dos Objetivos (Geral e Específicos):** Fase de síntese do que se pretende alcançar com o trabalho desenvolvido (Objetivo Geral) e o detalhamento específico deste objetivo (Objetivos Específicos);

- **Etapa 6 – Metodologia:** Etapa de definição de onde e como será realizada a pesquisa, identificando o tipo de pesquisa, a população, amostragem, instrumentos de coleta de dados e as formas de tabulação e análise;
- **Etapa 7 - Coleta de Dados:** Fase de levantamento dos dados necessários para o desenvolvimento do trabalho;
- **Etapa 8 – Tabulação e apresentação do Dados:** Etapa de "organização" dos dados para suporte das análises necessárias, usualmente realizada por meios computacionais;
- **Etapa 9 – Análise e Discussão dos Resultados:** Fase de realização da análise dos dados coletados para alcançar os objetivos definidos para o trabalho, confirmando ou rejeitando as hipóteses da pesquisa;
- **Etapa 10 – Conclusão da Análise e dos Resultados Obtidos:** Etapa de síntese dos resultados alcançados com a pesquisa. Além disso, deve-se explicitar a contribuição do trabalho para o desenvolvimento da ciência e tecnologia;

2.2.2. Procedimento da Pesquisa Bibliométrica

Segundo (GROOS; PRITCHARD, 1969), estudos bibliométricos são aqueles que contém a contagem de artigos, publicações e citações, independente da área de conhecimento, e que representam ocorrências estaticamente significativas. A bibliometria é utilizada como um método de análise quantitativa para a pesquisa científica (SU; LEE, 2010).

De acordo com (SU; LEE, 2010), os dados estatísticos obtidos através de estudos bibliométricos medem a contribuição do conhecimento científico em determinadas áreas, através das publicações feitas nelas. Deste modo, os dados obtidos podem ser utilizados para identificar tendências de pesquisa na área, da mesma forma que expõem novas oportunidades de estudo ainda não exploradas dentro dos ramos.

Desta forma, foi conduzida uma pesquisa bibliométrica para observar e analisar o estado da ciência no campo de previsão e estimação de resultados em projetos de desenvolvimento ágil de software. Seguindo o estudo de (RAO, 1968), foram utilizados dados como os autores das publicações, palavras-chave, países de publicação, citações, metodologia utilizada, entre outros, para conduzir os estudos bibliométricos deste trabalho.

A pesquisa bibliométrica conduzida neste trabalho tinha como objetivo identificar as principais tendências do campo de estudo de modelos de previsão e estimação de resultados em projetos de desenvolvimento ágil de software. Para alcançar este propósito, a análise foi realizada a partir das publicações obtidas na base de dados Scopus, devido a sua grande abrangência e diversidade de publicações.

A fim de extrair o máximo de informações possíveis da análise bibliométrica, ela foi segmentada em duas partes distintas: (1) Análise geral das publicações, por revista, autor e palavras-chave; (2) Análise específica das metodologias de previsão/estimação nas publicações classificadas como mais relevantes da base de dados. Para filtrar e obter as publicações mais relacionadas com o tema, foi desenvolvida uma chave de pesquisa com operadores booleanos para uma pesquisa avançada na base de dados escolhida. A chave utilizada foi:

(“Software Development”) **AND** (“Agile”) **AND** (“Project”) **AND** (“Predict*”) **OR** (“Estimat*”)

Inicialmente, foram obtidas 308 publicações delimitadas apenas a documentos do tipo “Artigo” e “Papers de Conferências” entre os anos de 2013 e 2023. Não foram impostas restrições relacionadas ao país de publicação, língua original de publicação e área específica de conhecimento. Todos estes artigos foram utilizados para a análise quantitativa (1) mencionada anteriormente neste tópico.

Para classificar os artigos mais relevantes da amostra obtida pela chave de pesquisa, ordenou-se as publicações de acordo com o número de citações que elas possuíam, partindo da mais citada para a menos citada. Percebeu-se então que 26 publicações acumulavam juntas 50% do número total de citações da amostra, tornando-as assim as mais relevantes para a aplicação da análise de cunho qualitativo.

Para conduzir a análise qualitativa na amostra das 26 publicações mais relevantes, foi criada uma tabela de detalhamento para orientar o estudo e armazenar as informações encontradas dentro de cada publicação. Na análise dos trabalhos classificados, o autor buscou pelas informações de: metodologia de estimação/previsão utilizada, complexidade de aplicação do modelo utilizado e setor da economia envolvido no estudo.

2.2.3. Procedimento de Acesso e Coleta de Dados

O trabalho foi realizado em uma empresa do mercado financeiro com uma grande produção de softwares e produtos digitais para amparar suas atividades de oferta de produtos e interface com clientes. Previamente a escolha, foram listadas uma série de empresas com grande volume de desenvolvimento ágil de software que poderiam fornecer as informações necessárias para o trabalho. Devido a sua grande escala de desenvolvimento para diversas aplicações e com diversas equipes, essa empresa foi escolhida para o fornecimento dos dados de insumo para a pesquisa.

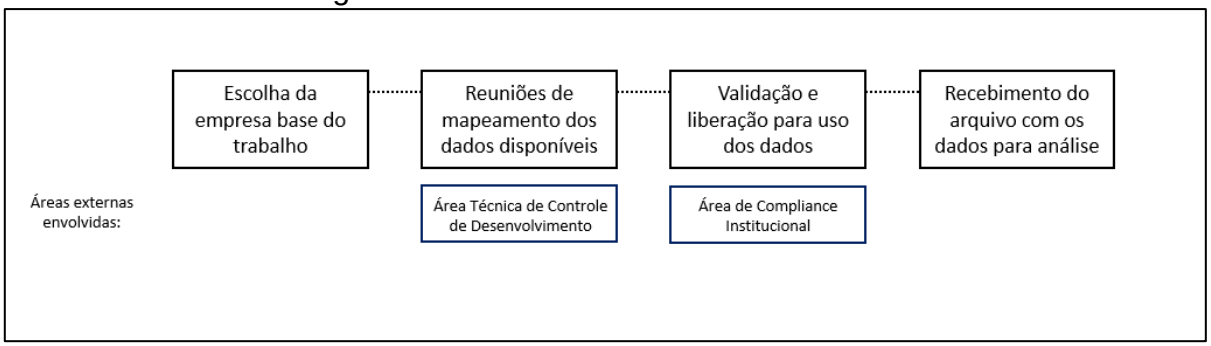
Para a coleta de dados, o autor procurou os encarregados pela área de controle de produção de software da empresa, responsável por mapear, priorizar e reportar as principais iniciativas de desenvolvimento da companhia. Foram realizadas reuniões iniciais com o objetivo de identificar quais dados eram monitorados pela área e quais poderiam ser disponibilizados para a pesquisa, mapeando junto a eles quais seriam as melhores informações para alimentar o modelo de previsão do trabalho.

Uma vez definidas as informações a serem utilizadas junto a área técnica responsável, foi iniciado um processo corporativo junto a área de *compliance* da companhia para checar a viabilidade da disponibilização das informações. Após uma análise da sensibilidade dos dados requisitados, a unidade de *compliance* aprovou a disponibilização dos dados com as seguintes exigências:

- Ausência de informações e detalhamentos aprofundados da empresa provedora dos dados. (Exemplo: Nome, Resultados Financeiros, Estrutura e Áreas Internas etc.)
- Ausência de informações sensíveis a respeito dos colaboradores envolvidos nos projetos de desenvolvimento de software. (Exemplo: Dados Pessoais, Especialidade de Desenvolvimento, entre outros)
- Permitido apenas o uso de informações de projetos finalizados a mais de 12 meses.

Desta forma, os dados foram fornecidos pela área técnica responsável em formato csv (*Comma-separated values*) sob a supervisão da área de *compliance* da companhia. De maneira geral, o processo de acesso aos dados pode ser sintetizado pelo seguinte fluxograma:

Figura 1 - Processo de Acesso aos Dados



Fonte: Autor

2.2.4. Procedimento de Tratamento e Manipulação dos Dados

Como foi informado no tópico anterior, as bases utilizadas ao longo do trabalho foram enviadas em formado CSV pela empresa e foram manipuladas dentro da ferramenta Excel, com alguns testes pontuais sendo realizados no software estatístico R. Ao todo, foram enviadas duas bases distintas com informações complementares que auxiliaram na condução do estudo:

- **Base de Dados 1:** Base com as informações dos épicos de portfólio dos três primeiros trimestres do ano de 2023. Nela se encontravam todas as informações relacionadas aos épicos (squad responsável pela execução, datas de início e fim da execução, esforço estimado pela equipe para a execução do épico etc.);
- **Base de Dados 2:** Base com as informações dos integrantes de cada squad nos três primeiros trimestres do ano de 2023. Nela se encontravam algumas informações relevantes para a análise a respeito dos desenvolvedores presentes na empresa naqueles trimestres (squad em que o desenvolvedor estava alocado, senioridade do desenvolvedor de acordo com a tabela de cargos da empresa etc.);

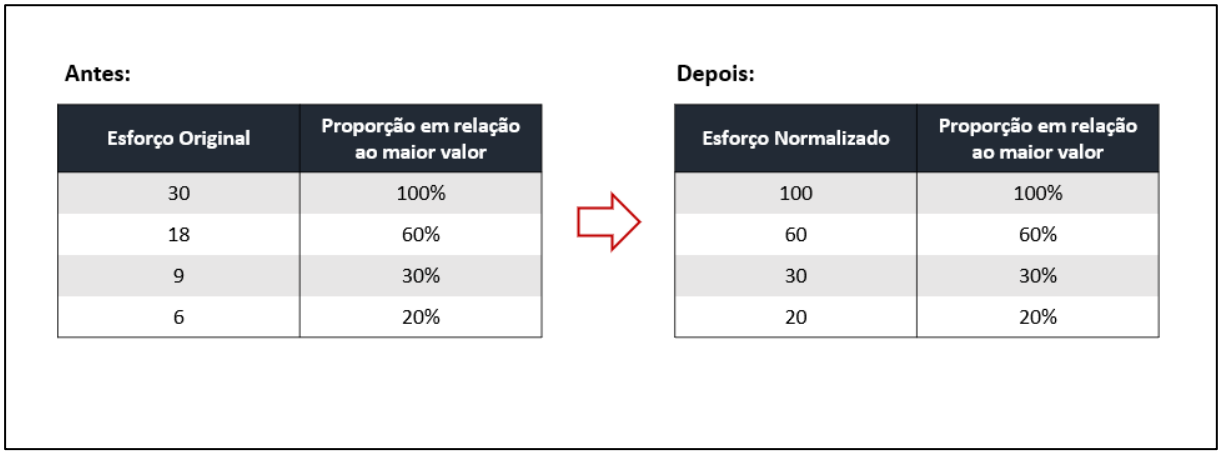
Ao todo, a base de dados 1 possuía 9196 linhas (9196 épicos diferentes) e um total de 43 colunas com diferentes informações relacionadas aos épicos. Já a base 2, possuía ao todo 4919 linhas e 10 colunas com informações distintas sobre os colaboradores.

Apesar das bases serem extraídas de sistemas oficiais de controle da companhia, haviam alguns tratamentos e manipulações necessárias para que as análises fossem realizadas com a melhor qualidade possível. Isso se dá, pois alguns campos advêm

do preenchimento manual de formulários por parte da equipe de desenvolvimento, o que pode acarretar inconsistências nos dados. Realizar tratamentos como a exclusão de observações com variáveis em branco (dados perdidos) evita distorções nas análises e garante resultados mais realistas (HAIR; BLACK; BABIN, 2009). As principais atividades de tratamento estão descritas abaixo:

- **Exclusão de Linhas em Branco:** Algumas linhas não possuíam preenchimento de certos campos e foram excluídas da base para que fossem usados apenas dados 100% preenchidos (Exemplo: Épicos sem data de entrega preenchida). Devido ao fato da base possuir um número considerável de observações e os dados em branco serem “ignoráveis”, a exclusão das observações com variáveis em branco não reduziu a amostra para um nível inaceitável de observações (HAIR; BLACK; BABIN, 2009);
- **Normalização de Campos Despadronizados:** Alguns campos de preenchimento manual possuíam erros de digitação ou despadronização em relação a uma mesma nomenclatura ou item. Essas discordâncias ou erros foram corrigidos para que todos os campos possuíssem itens padronizados (Exemplo: Ajuste da senioridade de “Analista Jr.” e outras variações para “Analista Júnior);
- **Normalização de Variáveis Subjetivas:** Algumas variáveis da base vinham de preenchimentos próprios das equipes e possuíam um alto grau de subjetividade, como a variável de esforço despendido para a execução de um épico. Como não havia uma escala padrão para o preenchimento deste tipo de variável, cada squad possuía sua própria noção de grandeza. Para minimizar os impactos da subjetividade e poder comparar este tipo de variável entre diferentes squads, a seguinte normalização foi feita:

Figura 2 - Esquema de Normalização da Variável de Esforço



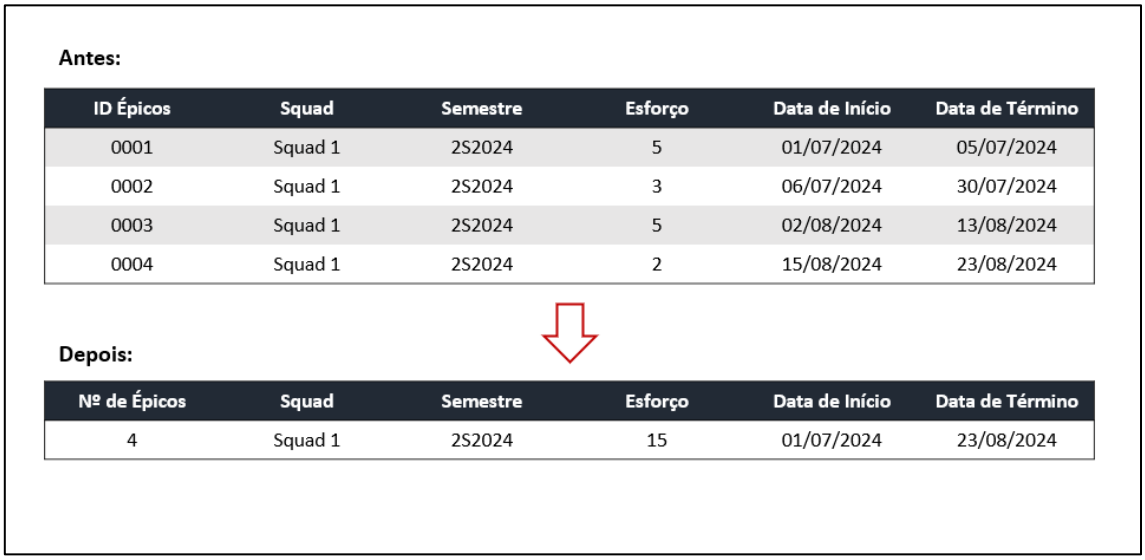
Fonte: Autor

- **Consolidação da Base 1 por Trimestres:** O trabalho objetiva analisar os resultados de projetos de software, com diferentes atividades e épicos sendo executados. Desta forma, a base de épicos de portfólio foi consolidada por trimestre para cada squad, de modo que em vez de apresentar os resultados individuais dos épicos (como mostrado na base crua), ela iria consolidar os resultados de trabalho de todos os épicos executados por uma squad ao longo de um trimestre em apenas uma linha. Isso acarretou o surgimento de novas variáveis a serem analisadas:
    - **Número de Épicos:** Esta variável foi derivada do número de linhas (épicos) que uma squad possuía em um trimestre, na base 1, e passou a ser uma nova coluna na base consolidada com a soma deste número de épicos observados;
    - **Variáveis de Data/Tempo de Execução:** Para a consolidação dos campos de data (Exemplo: data de início ou entrega de um épico) usou-se a premissa de que a data de início mais antiga e a data de entrega mais recente vistas em algum épico de um trimestre de uma squad, seriam as datas de início e fim do trabalho dessa squad no trimestre;
    - **Variáveis de Atraso:** Apesar da consolidação das datas de início e fim do trabalho das squads, as variáveis de atraso ainda foram contabilizadas de acordo com as datas individuais de cada épico. Ou seja, se em um trimestre houve dois épicos que excederam as suas respectivas datas meta de entrega, aquele “portfólio” vai apresentar

dois atrasos, mesmo que as duas datas de entrega desses épicos sejam menores ou iguais à data fim oficial do trimestre da squad;

- **Variáveis Acumulativas:** Para variáveis como esforço de execução, passíveis de acumulação, a consolidação foi feita através da soma dos esforços de cada épico presente naquele trimestre. Desta forma, quanto maior o número de épicos ou quanto maior o esforço individual de cada um deles, maior será o esforço acumulado naquele trimestre;
- **Variáveis Trimestrais:** Para variáveis como o índice de senioridade ou a porcentagem de tempo de suporte, os valores atribuídos a cada épico na base 1 já faziam referência ao resultado daquele trimestre como um todo. Isso se dá, pois essas variáveis são de características da squad, que é fixa ao longo de todo o trimestre. Para esses casos, os valores presentes na base 1 foram mantidos os mesmos para a consolidação por trimestre;

Figura 3 - Esquema de Consolidação da Base por Trimestre



Fonte: Autor

2.2.5. Procedimento de Definição do Índice de Senioridade

Segundo (BARBALHO et al., 2022), senioridade pode ser definida com o conjunto de habilidades, capacidades, conhecimentos e experiências que diferenciam profissionais altamente capacitados daqueles com capacitação média em determinada tarefa.

De acordo com (FLORES, 2012), a carência de habilidades técnicas raramente é a causa raiz principal dos resultados ruins de projetos complexos. Segundo seu estudo, a falta de habilidades de comunicação, organização do trabalho e a atenção exagerada a detalhes causada pelo desprovimento de uma visão clara e completa do projeto, são as principais causas para os resultados insatisfatórios da maioria dos projetos. Isso vai ao encontro de (DE CARVALHO; RABECHINI JUNIOR, 2015), que discorre sobre como as *softskills*, adquiridas através de experiência prática com o tema, são altamente significativas quando se está lidando com projetos complexos.

Para o desenvolvimento do trabalho, optou-se por uma definição de senioridade baseada no tempo de atuação do profissional na área e não nos títulos acadêmicos que ele possui. Com essa visão, foi construída uma escala de senioridade para cada um dos cargos possíveis para os profissionais dentro de uma equipe de desenvolvimento de software na empresa analisada.

Tabela 1 - Escala de Senioridade

Cargos	Tempo de Experiência	Índice de Senioridade
Estagiário(a)	< 1 ano	10
Assistente	1 a 2 anos	20
Analista Júnior	2 a 3 anos	30
Analista Pleno	3 a 5 anos	50
Analista Sênior	5 a 7 anos	70
Especialista 1	7 a 10 anos	85
Especialista 2	> 10 anos	100

Fonte: Autor

Desta forma, a tabela acima foi estruturada se inspirando na proposta de (DYER; GREGERSEN; CHRISTENSEN, 2011), em que há uma diferença significativa na performance de profissionais mais capacitados do que aqueles com capacitação média quando alocados em projetos inovadores e complexos. Para o trabalho, cada profissional de cada equipe recebeu o seu índice de senioridade de acordo com o cargo em que ele ocupa na empresa. Para calcular o índice geral de senioridade da equipe, é realizada uma média simples entre os indivíduos do time. Por exemplo, em uma equipe com 2 analistas juniores e 1 especialista 2, o índice geral de senioridade a ser considerado no modelo é de 53,3.

3. Referencial Teórico

3.1. Engenharia de Software

3.1.1. Surgimento da Engenharia de Software

Para compreender o conceito de Engenharia de Software em sua totalidade é necessário primeiro entender o que é um Software. Para (PRESSMAN, 2021), softwares podem ser compreendidos como elemento mais lógicos do que físicos. Consistindo em: (1) um conjunto de instruções que fornecem características, funções e desempenho desejados quando executados corretamente; (2) estruturas de dados que possibilitam aos programas a devida manipulação de informações; (3) informações descritivas da operação e uso dos programas.

Os primeiros softwares nasceram na década de 50. Até então, o esforço dos pesquisadores da área estava concentrado no estudo e desenvolvimento de hardwares. Na época, o desenvolvimento de software era feito sem nenhuma técnica de engenharia e a sua comercialização e distribuição eram bastante limitadas. Como os hardwares eram exclusivos de centros de pesquisa de ponta, os softwares eram pouco conhecidos e difundidos (BOEHM, 2006).

De acordo com (BOEHM, 2006), na metade dos anos 60 esse cenário foi modificado. Com o advento dos microprocessadores, o hardware deixou de ser o foco das atenções dos pesquisadores, que voltaram seus esforços para a produção e documentação de softwares. Isso abriu espaço para as organizações começarem a desenvolver e comercializar grandes sistemas, denominados produtos de software (PRESSMAN, 2021).

O mundo moderno não poderia existir sem a existência de softwares. Sistemas computacionais são responsáveis pelo controle dos mais diversos setores da sociedade atual. Infraestrutura e serviços nacionais, manufatura, sistema financeiro, entre outros, são todos amparados pelo funcionamento de softwares. Desta forma, a existência deles e de suas áreas paralelas, como a engenharia de software, são essenciais para o desenvolvimento das sociedades nacionais e internacionais (SOMMERVILLE, 2011).

Segundo (SOMMERVILLE, 2011), o termo “Engenharia de Software” foi proposto pela primeira vez em 1969, durante uma conferência da OTAN para a discussão sobre problemas no desenvolvimento de softwares da época. Naquele período, grandes

projetos de software atrasavam com frequência, não entregavam as funcionalidades necessárias aos clientes, custavam mais do que o esperado e não eram confiáveis.

3.1.2. Conceituação da Engenharia de Software

Neste contexto, para IEEE (2017) a engenharia de software é a aplicação de uma abordagem sistemática, disciplinada e quantificável no desenvolvimento, operação e manutenção de softwares; ou seja, é uma aplicação de engenharia nos processos do ciclo de vida de um software. Já para (SOMMERVILLE, 2011) a engenharia de software é uma disciplina no campo da engenharia, focada em apoiar o desenvolvimento profissional de softwares. Ela inclui técnicas de apoio a especificação, projeto e evolução dos programas, garantindo que eles possuam a melhor relação de custo-benefício.

Desta forma, para que os softwares cumpram devidamente as suas funções em uma realidade de constante evolução de requisitos de negócio e de plataformas computacionais, como a vivenciada hoje, é necessário que haja um esforço metodológico coordenado para garantir a construção desses programas. Neste cenário, (DAYANI, 1999) sugere justamente que o objetivo da engenharia de software é elaborar metodologias de construção baseadas nessa noção de evolução e que garantam a interação e cooperação entre sistemas novos e antigos.

Para (SOMMERVILLE, 2011), a importância da disciplina de engenharia de software reside na capacidade de gerar sistemas confiáveis de maneira rápida. Além disso, no longo prazo, é mais barato utilizar os métodos e técnicas da engenharia, uma vez que eles evitam custos posteriores mais altos de correções e alterações dos sistemas já prontos.

Segundo (PRESSMAN, 2021), a engenharia de software pode ser vista como uma tecnologia em camadas, ilustrado na Figura 4. E como qualquer outra aplicação de engenharia, é fundamentada em um compromisso organizacional com a qualidade.

Figura 4 - Esquema de Engenharia de Software em camadas

Fonte: Pressman, 2021

A base da engenharia de software é a sua camada de processos, que garante o desenvolvimento de programas de forma racional e dentro dos prazos estabelecidos. São os processos que constituem o fundamento para o controle de projetos de software e estabelecem o contexto no qual são aplicados os métodos técnicos e artefatos (PRESSMAN, 2021).

3.2. Processos de Desenvolvimento de Software

3.2.1. Conceituação de Processos de Desenvolvimento de Software

Segundo (PRESSMAN, 2021), processos de software são importantes pois propiciam estabilidade, controle e organização para uma atividade que pode se tornar muito complexa e caótica. Além disso, para ele, processos de software modernos também devem conferir agilidade ao desenvolvimento de sistemas, um aspecto imprescindível. Desta forma, devem demandar apenas atividades, controles e produtos de trabalho que realmente sejam apropriados para a equipe de projeto.

Para (SOMMERVILLE, 2011), um processo de software é uma sequência lógica de atividades relacionadas que leva à produção de um produto de software. Já segundo (PRESSMAN, 2021), um processo de desenvolvimento de software é um conjunto minimamente previsível de atividades de trabalho, ações e tarefas realizadas quando algum artefato de software será criado.

Neste contexto, processos de software são modelos complexos, e por serem processos intelectuais e criativos, dependem de pessoas para tomar decisões e fazer julgamentos. Por este motivo, não existe um processo de software ideal. Os processos estão em constante evolução para tirarem o melhor proveito das capacidades da organização, pessoas e das especificidades de cada sistema a ser desenvolvido

(SOMMERVILLE, 2011). Apesar disso, é possível fazer abstrações e criar um “modelo genérico” com representações simplificadas e atividades comuns às diferentes abordagens de desenvolvimento de software.

Segundo (PRESSMAN, 2021), a metodologia genérica para a engenharia de software estabelece cinco atividades básicas principais: Comunicação, Planejamento, Modelagem, Construção e Entrega. Além disso, as atividades de apoio, como administração de riscos, garantia da qualidade etc. são aplicadas ao longo do processo. Essas atividades básicas são aplicáveis a todos os projetos de software e podem se organizar de diferentes maneiras de acordo com a abordagem de desenvolvimento escolhida pela equipe. O nome dado a essa organização genérica é “fluxo do processo”.

- **Comunicação:** Etapa de interação com os envolvidos no projeto para o devido entendimento dos objetivos do software e para o levantamento de requisitos;
- **Planejamento:** Fase de definição do plano de projeto de software, delimitando as tarefas técnicas a serem conduzidas, os riscos envolvidos, recursos necessários, cronograma de execução e os artefatos a serem criados;
- **Modelagem:** Atividade de criação de um modelo simplificado do software para um melhor entendimento das necessidades dele e do projeto que irá construí-lo;
- **Construção:** Etapa de construção do software, de fato. Envolve a geração de código (manual ou automatizada) e a realização de testes
- **Entrega:** Entrega do software construído ao cliente, recolhendo avaliações e feedbacks do produto;

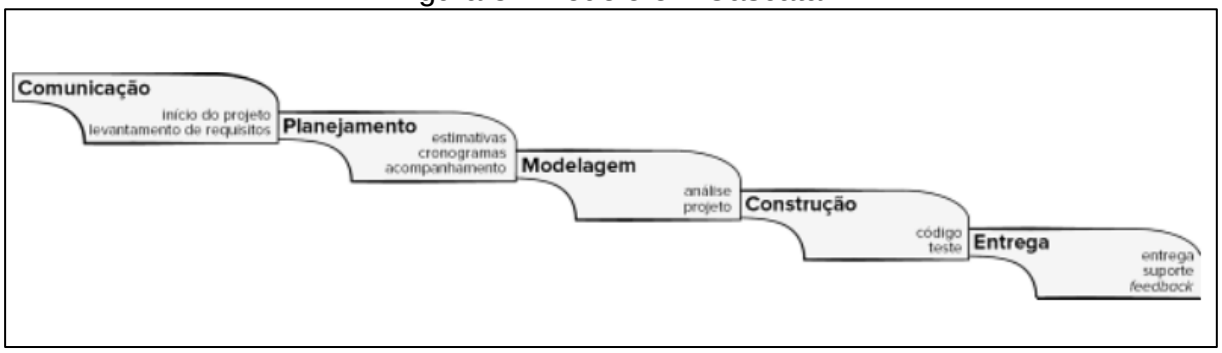
Os processos de software prescritivos são aqueles que a ordem de atividades e consistência do projeto são questões predominantes. Além disso, são chamados “prescritivos” pois prescrevem um conjunto de elementos de processos como atividades metodológicas, ações de engenharia software, tarefas etc. Segundo (PRESSMAN, 2021), os processos prescritivos podem ser organizados em três categorias distintas: Modelo Cascata, Modelo de Processo de Prototipação (Incremental) ou Modelo de Processo Evolucionário.

3.2.2. Modelo de Processo Cascata

Segundo (ROYCE, 1970), esse foi o primeiro modelo de processo de desenvolvimento de software publicado, derivando de processos gerais da engenharia de sistemas. Devido ao encadeamento e interdependência das fases deste modelo, ele recebe o nome de “modelo em cascata” (SOMMERVILLE, 2011).

O modelo de processo cascata propõe uma abordagem sequencial e sistemática para o desenvolvimento de software. Nele o projeto se inicia na fase de comunicação e avança sequencialmente até a fase de entrega, passando por todas as outras etapas do modelo genérico de organização de um processo de desenvolvimento de software (PRESSMAN, 2021). Em princípio, o resultado de cada fase do projeto é um documento final de compilação da etapa aprovado/assinado pelos responsáveis. A consolidação desse documento dá início à fase seguinte. Desta forma, no modelo cascata, uma nova fase de projeto só pode se iniciar após o término da anterior (SOMMERVILLE, 2011).

Figura 5 - Modelo em Cascata



Fonte: Pressman, 2021

Devido a sua estrutura linear, o modelo cascata é recomendado para projetos de software em que os requisitos do sistema são muito bem compreendidos e consolidados. Esse tipo de situação é mais comum em projetos de adaptações e aperfeiçoamentos bem definidos em um sistema já existente (p. ex, um projeto de adaptações em um software contábil exigida devido a mudanças de normas governamentais) (PRESSMAN, 2021).

Segundo (PRESSMAN, 2021), ao longo das últimas décadas o modelo de cascata tem sofrido duras críticas quanto a sua eficácia e aplicabilidade prática. Na prática, processos de desenvolvimento de software não são simples e lineares como o modelo propõe, as fases de um projeto acabam por se sobrepor e alimentam umas

às outras com feedbacks e novas informações não mapeadas anteriormente. Esse tipo de comunicação entre as fases e inserção de novas informações e requisitos geram retrabalho em um modelo sequencial, como o de cascata (SOMMERVILLE, 2011).

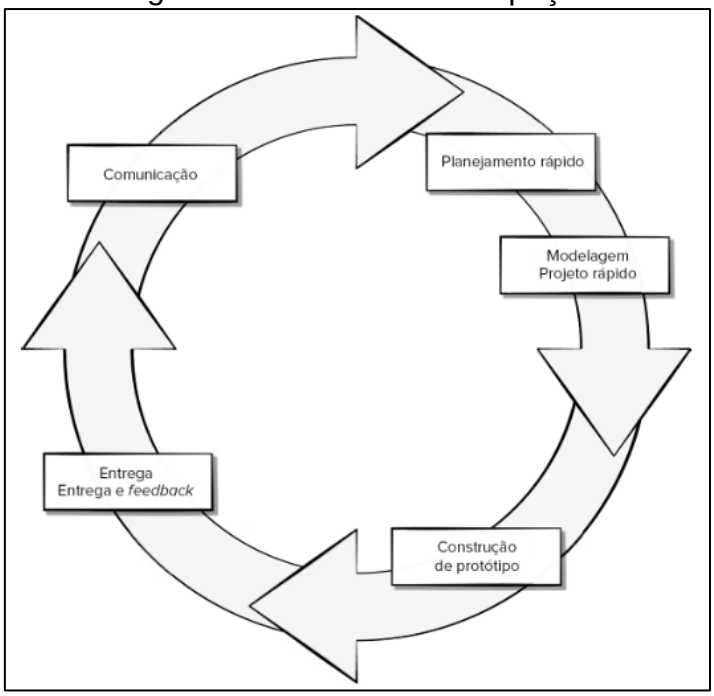
Outra grande crítica ao modelo é que por ter uma estrutura sequencial e com fases interdependentes entre si, versões operacionais do sistema só serão alcançadas ao final completo do projeto. Para muitos clientes, na realidade acelerada do mundo de hoje, isso pode ser um problema. Por fim, também devido a sua estrutura, erros graves podem não ser detectados até o final da revisão do programa operacional do projeto, causando retrabalho e mais gasto de tempo (PRESSMAN, 2021).

3.2.3. Modelo de Processo de Prototipação

Segundo (SOMMERVILLE, 2011), o modelo de processo de prototipação, também chamado de modelo incremental, é o modelo de desenvolvimento mais utilizado atualmente para o desenvolvimento de sistemas e aplicativos. A prototipação reflete a maneira como os humanos resolvem a maioria de seus problemas. Raramente desenvolve-se uma solução completa de um problema com antecedência. Normalmente, move-se passo a passo em direção a uma solução, fazendo pequenas alterações na medida que se conhece mais sobre o problema e sobre a própria solução desenvolvida.

O modelo de prototipação é baseado na ideia de desenvolver uma implementação prévia e rápida do sistema, recolher *feedbacks* sobre ela e continuar por meio da criação de várias versões até que o sistema final esteja pronto (SOMMERVILLE, 2011). As atividades envolvidas no projeto ocorrem de maneira cíclica e muito mais dinâmica que no modelo de processo cascata, sempre realizando ajustes para chegar em uma versão final mais adequada ao cliente. Essas implementações prévias e rápidas do sistema podem ser chamadas de protótipos.

Figura 6- Modelo de Prototipação



Fonte: Pressman, 2021

Segundo (PRESSMAN, 2021), embora a prototipação possa ser utilizada como um modelo de processo isolado (*stand-alone process*), normalmente ela é utilizada como uma técnica a ser implementada no contexto de diferentes projetos, independente do modelo de processo utilizado nele. Isso se dá pois sempre que os requisitos de um sistema estão obscuros, a prototipação auxilia os envolvidos (desenvolvedores e clientes) a compreender o que deve ser construído no projeto. Desta forma, o desenvolvimento incremental ou prototipação, é uma parte fundamental das abordagens ágeis de gerenciamento de projetos de software (SOMMERVILLE, 2011).

Para (SOMMERVILLE, 2011), o processo de prototipação tem três grandes vantagens quando comparado a outros modelos. Em primeiro lugar, o custo para acomodar mudanças de requisitos ao longo do projeto é consideravelmente menor que em modelos sequenciais. Além disso, é muito mais fácil obter feedbacks do cliente sobre o desenvolvimento já que ele vai lidar diretamente com uma versão simplificada do software que lhe será entregue. Por fim, é possível fazer entregas parciais do software com certas utilidades ao longo do projeto, sem necessitar da espera até o final do projeto para obter uma versão útil do sistema.

Apesar disso, segundo (PRESSMAN, 2021), é necessário ter atenção em alguns pontos quando se utiliza o modelo de prototipação. Muitas vezes os envolvidos

no projeto só se atentam à evolução do protótipo e perdem de vista a estrutura do programa em si, o que pode levar a defeitos na qualidade global do software e pode comprometer o processo de manutenção no longo prazo. Ademais, os engenheiros podem fazer concessões na implementação com o objetivo de fazer o protótipo entrar em operação o mais rápido possível. Essas conceções, que eram para ser ajustadas futuramente, podem se tornar parte fundamental do sistema se forem perdidas de vista em revisões posteriores.

3.2.4. Modelo de Processo Evolucionário

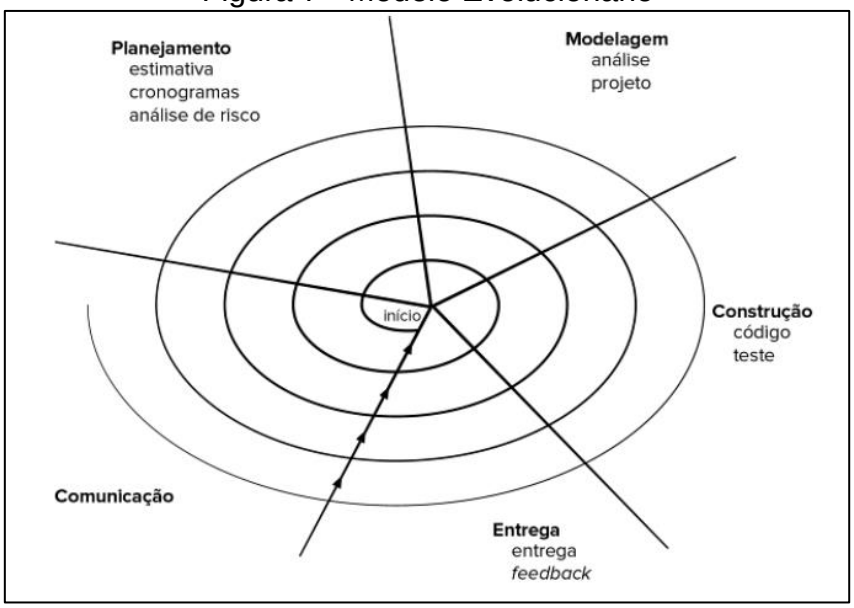
Segundo (PRESSMAN, 2021), como todos os sistemas complexos, os softwares evoluem e se modificam ao longo do tempo. Na medida que os projetos de desenvolvimento avançam, os requisitos de negócio e do produto constantemente se alteram, tornando impossível seguir um planejamento em linha reta para um produto final. Além disso, por pressões constantes de mercado, o lançamento rápido de sistemas que irão se modificar com o tempo faz necessária a existência de um modelo pensado especificamente para desenvolver produtos assim.

Pensado incialmente por (BOEHM, 1988), o modelo de processo em espiral é um modelo para softwares evolucionários que integra a natureza interativa da prototipação e os aspectos sistemáticos do modelo de cascata. Ele é desenhado para comportar o desenvolvimento rápido de versões cada vez mais completas de software que se alteram com o tempo. Segundo (PRESSMAN, 2021), com este modelo, o sistema é desenvolvido em uma série de versões evolucionárias. Nas primeiras interações a versão consiste em modelos simplificados ou protótipos. Na medida que as interações avançam, são produzidas versões cada vez mais completas do software.

De acordo com (PRESSMAN, 2021), o modelo em espiral é dividido em uma série de atividades metodológicas. Cada uma dessas atividades representa um segmento diferente do caminho em “espiral” de desenvolvimento. Quando um modelo deste é aplicado, a equipe realiza as atividades indicadas no circuito de desenvolvimento em sentido horário, começando pelo centro da espiral. Na medida que as evoluções são realizadas, os riscos de projeto são revisitados e alterados de acordo com a situação do projeto.

Desta forma, a cada evolução e passagem pelas atividades do circuito, as características do software são atualizadas e incrementadas de acordo com os feedbacks provenientes da evolução anterior. Por exemplo, a cada passagem na atividade de planejamento, os custos e cronogramas do projeto são reavaliados e atualizados. Isso torna o projeto muito mais ágil para lidar com mudanças de requisitos ao longo do desenvolvimento do sistema e tende a criar softwares mais coerentes com as necessidades dos clientes (PRESSMAN, 2021).

Figura 7 - Modelo Evolucionário



Fonte: Pressman, 2021

Diferente dos outros modelos de processo, que terminam suas atividades quando o software é entregue ao cliente, o modelo em espiral pode ser adaptado para ser aplicado ao longo de toda a vida útil do sistema. O processo de desenvolvimento em espiral é uma abordagem mais realista para o projeto de sistemas e de software em larga escala e com grande complexidade (PRESSMAN, 2021).

Por outro lado, o modelo em espiral não é uma unanimidade para todos os casos de desenvolvimento e clientes. Pode ser difícil convencer os clientes e elaborar contratos com a garantia de que o projeto que segue este modelo é controlável, uma vez que a cada revolução, os parâmetros do projeto podem mudar.

3.2.5. Modelo de Processo Unificado

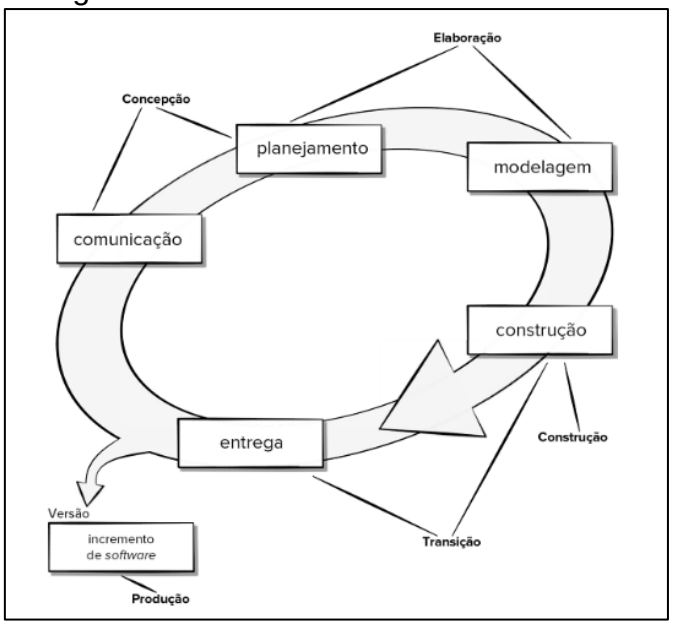
Para (JACOBSON, 1999), o modelo de processo unificado (PU) é uma tentativa de se utilizar dos melhores recursos e características dos modelos tradicionais de software, mas os adaptando para os princípios de desenvolvimento ágil de sistemas. O modelo propõe um fluxo de projeto interativo e incremental, dando a visão evolucionária ao software em construção, algo essencial para o desenvolvimento moderno de softwares complexos e com muitas mudanças de requisitos (PRESSMAN, 2021).

Ao todo, o modelo de processo unificado possuí cinco fases distintas de projeto que conversam com as atividades do modelo genérico de desenvolvimento. Na figura 6, é possível ver a relação das fases com as atividades padrão do processo genérico. Na primeira fase, a de concepção, ocorre a comunicação com o cliente e o planejamento inicial do projeto. Nela são descritos os requisitos de negócio fundamentais, são elaborados os casos de uso preliminares, são avaliados riscos e levantados o cronograma e recursos iniciais (PRESSMAN, 2021).

A fase de elaboração do modelo unificado inclui as atividades de planejamento e modelagem do processo genérico de desenvolvimento de software. Nesta fase, os casos de uso preliminares são refinados e expandidos, criando uma base de arquitetura para o novo sistema. Já a fase de construção do PU é idêntica a fase de construção do processo genérico, onde há a implementação do código fonte e de todos os recursos e funções necessárias para o incremento do software. Na medida que os componentes são implementados, realizam-se testes de unidade para cada um deles (PRESSMAN, 2021).

Por fim, a fase de transição do PU, engloba os últimos estágios da etapa de construção genérica e a entrega do software. Nela o software é entregue para testes beta dos usuários que fornecem feedbacks para a evolução do sistema. Ao final da fase de transição, o incremento realizado torna-se uma versão utilizável do sistema. A fase de produção do processo unificado coincide com a atividade de entrega do processo genérico. Nesta fase o software é entregue e monitora-se o uso contínuo dele, fornecendo suporte e avaliando possíveis defeitos e requisições de mudanças (PRESSMAN, 2021).

Figura 8 - Modelo de Processo Unificado



Fonte: Pressman, 2021

Segundo (PRESSMAN, 2021), as fases do processo unificado ocorrem concomitantemente e de forma escalonada. Devido ao fato de que os softwares possuem diversos “módulos”, é provável que ao passo que a etapa de produção esteja sendo realizada, já se tenha começado o incremento de alguma outra parte seguinte do sistema.

3.3. Metodologias Ágeis no Desenvolvimento de Software

3.3.1. Surgimento das Metodologias Ágeis

Ainda na década de 1980 e início da década de 1990, havia um cenário entre os engenheiros de software de que o melhor método para se produzir softwares de qualidade era através de um planejamento prévio cuidadoso (SOMMERVILLE, 2011). Para eles o processo de desenvolvimento deveria ser rigoroso e controlado, com a qualidade da segurança formalizada e apoiados por ferramentas CASE (*Computer-aided software engineering*). Essa percepção partia principalmente de engenheiros envolvidos em projetos de softwares grandes e duradouros, como sistemas de controle de uma aeronave moderna (SOMMERVILLE, 2011).

Este tipo de desenvolvimento pesado orientado a planos gera uma sobrecarga nas atividades de planejamento, projeto e documentação do sistema (SOMMERVILLE, 2011). Essa sobrecarga é justificada para projetos que envolvem múltiplas equipes simultâneas e sistemas tidos como críticos e de grande porte.

Apesar disso, quando essa metodologia é aplicada ao desenvolvimento de sistemas corporativos de pequeno ou médio porte, a sobrecarga é tanta que se gasta mais tempo em análises e documentações do que com o desenvolvimento do software em si, levando muitas vezes a atrasos e descasamento de requisitos (SOMMERVILLE, 2011).

Neste cenário, já se reconhecia a necessidade de métodos de desenvolvimento mais rápidos que fossem capazes de se adaptar e gerenciar melhor a mudança de requisitos ao longo do projeto (SOMMERVILLE, 2011). Ainda na década de 1980, a IBM introduziu o desenvolvimento incremental (MILLS, 1980) e ocorreu o advento das linguagens de quarta geração, que também apoiaram o desenvolvimento e entrega rápida de sistemas (MARTIN, 1982). No entanto, essa corrente de pensamento realmente ganhou força da década de 1990 com o aparecimento das abordagens ágeis de desenvolvimento, como a Metodologia de Desenvolvimento de Sistemas Dinâmicos (DSDM) (STAPLETON, 1999), Scrum (SCHWABER; BEEDLE, 2001) e Extreme Programming (BECK, 1999).

Esse tipo de desenvolvimento é ratificado em 2001, quando um grupo composto por desenvolvedores de software e consultores assinam o “Manifesto para o desenvolvimento ágil de software” (BECK, 2001). Nele, os autores defendiam a entrega de softwares operacionais acima de documentações extremamente detalhadas, interações entre indivíduos acima de processos e ferramentas, e outras convenções que deixassem o desenvolvimento de sistemas mais ágil para lidar com mudanças constantes de requisitos.

Desta forma, a engenharia de software ágil constitui uma alternativa razoável para a engenharia convencional, capaz de entregar sistemas operacionais com qualidade de maneira muito mais rápida (PRESSMAN, 2021). Isso é essencial quando se está lidando com ambientes modernos em que os requisitos e regras de negócio mudam constantemente e que a rapidez no lançamento de um sistema pode ser definidora no sucesso da empresa.

3.3.2. Conceituação das Metodologias Ágeis

Para (JACOBSON, 2002), a difusão da mudança é o principal condutor para a agilidade aplicada a engenharia de software. Os engenheiros de software deveriam ser rápidos para assimilar e englobar as mudanças de contexto dentro dos seus processos de desenvolvimento de sistemas. No entanto, a agilidade em engenharia de software não é apenas uma resposta à mudança, ela também engloba outros princípios propostos dentro da filosofia do “Manifesto para o desenvolvimento ágil de software” (PRESSMAN, 2021).

O manifesto mencionado possuí quatro princípios centrais dentro da sua filosofia de desenvolvimento de software, segundo (BECK, 2001). São eles:

- Indivíduos e interações mais do que processos e ferramentas;
- Software em funcionamento mais do que documentação abrangente;
- Colaboração com o cliente mais do que negociação de contrato;
- Respostas a mudanças mais do que seguir um plano;

Embora todos os itens mencionados sejam importantes, a agilidade em desenvolvimento de softwares valoriza mais os pontos à esquerda das sentenças (BECK, 2001).

Segundo (SOMMERVILLE, 2011), métodos ágeis baseiam-se em uma abordagem incremental para a especificação, desenvolvimento e entrega de sistemas. Eles têm como objetivo reduzir a burocracia do processo de desenvolvimento, diminuindo ao máximo as etapas de trabalho que não possuem valor agregado claro para o cliente. Portanto, a agilidade enfatiza a entrega rápida de softwares operacionais, incentivando uma comunicação mais fluída entre todas as partes envolvidas e diminuindo a relevância de artefatos intermediários como relatórios e documentação extensa (PRESSMAN, 2021).

Portanto, fica evidente que as metodologias tradicionais priorizam a geração de documentações e especificações detalhadas do projeto, ao mesmo tempo que se apoiam em uma sequência rígida de processos. Para ambientes com pouca variabilidade, metodologias assim podem funcionar muito bem. Porém, em cenários onde ocorrem mudanças constantes de requisitos do produto, as metodologias ágeis são mais recomendadas (PRIKLADNICKI; WILLI; MILANI, 2014). De maneira, geral

pode-se comparar as metodologias ágeis e tradicionais de desenvolvimento da seguinte forma (Figura 9).

Figura 9 - Quadro de Comparação de Abordagens

	TRADICIONAL	METODOLOGIAS ÁGEIS
Pressupostos fundamentais	Sistemas totalmente especificáveis, previsíveis; desenvolvidos a partir de um planejamento extensivo e meticuloso	Software adaptativo e de alta qualidade; pode ser desenvolvido por equipes pequenas utilizando os princípios da melhoria contínua do projeto e testes orientados a rápida resposta a mudanças
Controle	Orientado a processos	Orientado a pessoas
Estilo de gerenciamento	Comandar e controlar	Liderar e colaborar
Gestão do conhecimento	Explícito	Tácito
Atribuição de papéis	Individual – favorece a especialização	Times auto-organizáveis – favorece a troca de papéis
Comunicação	Formal	Informal
Ciclo do projeto	Guiado por tarefas ou atividades	Guiado por funcionalidades do produto
Modelo de desenvolvimento	Modelo de ciclo de vida (Cascata, Espiral, ou alguma variação)	Modelo iterativo e incremental de entregas
Forma/estrutura organizacional desejada	Mecânica (burocrática com muita formalização)	Orgânica (flexível e com incentivos a participação e cooperação social)

Fonte: Prikladnicki; Willi; Milani, 2014

Desta forma, as metodologias ágeis têm sido imprescindíveis no desenvolvimento de software moderno (PRIKLADNICKI; WILLI; MILANI, 2014). Elas priorizam a entrega de valor real no projeto e a melhor interação entre as partes envolvidas, em detrimento do cumprimento de prazos, custos e escopo definido inicialmente. Isso tem gerado profissionais de software mais completos, produtos de software com maior qualidade e clientes mais satisfeitos (PRIKLADNICKI; WILLI; MILANI, 2014).

3.3.3. Extreme Programming (XP)

Segundo (SOMMERVILLE, 2011), o “Extreme Programming” (XP) é talvez um dos mais conhecidos e utilizados métodos de desenvolvimento ágil de softwares. Criado por (BECK, 2004), a metodologia tinha como objetivo impulsionar e englobar práticas reconhecidamente boas na área de desenvolvimento de software, como o desenvolvimento interativo e o envolvimento do cliente.

De acordo com (PRESSMAN, 2021), a XP envolve um conjunto de normas e práticas constantes no contexto de quatro atividades metodológicas principais: Planejamento, Projeto, Codificação e Testes. Dentro de cada uma dessas atividades

existem conceitos chaves que diferenciam este método de outros frameworks ágeis de desenvolvimento existentes.

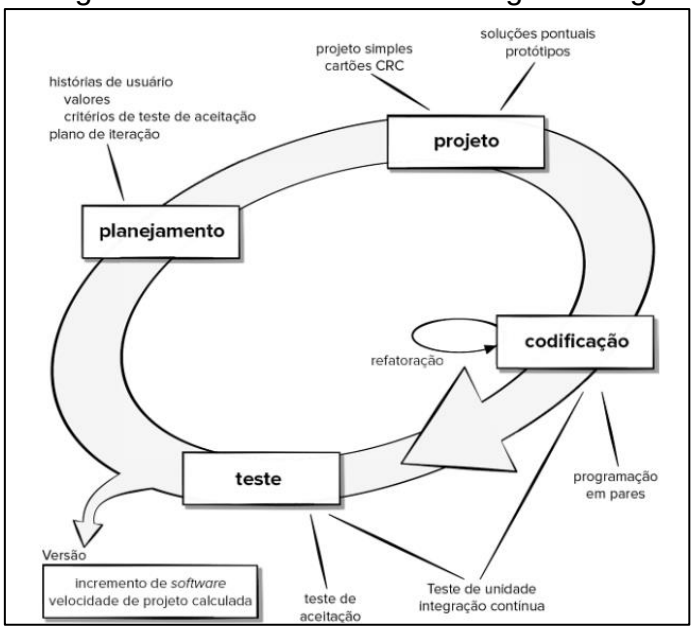
A fase de planejamento é caracterizada pelo levantamento dos resultados, características e funcionalidades esperadas pelo cliente para o software a ser construído. O nome dado a esse conjunto de requisitos de software é "*histórias de usuário*". São atribuídos pesos/notas a essas histórias tanto pelos clientes (avaliando o valor agregado do requisito) tanto pelos desenvolvedores (que avaliam o grau de esforço para a implementação do requisito). As partes interessadas então trabalham juntas para priorizar quais histórias serão contempladas em cada versão do sistema (PRESSMAN, 2021).

Segundo (PRESSMAN, 2021), a fase de projeto se caracteriza pela tentativa de simplificação ao máximo das atividades, desestimulando o projeto de funcionalidades extras não mapeadas no planejamento. A XP se utiliza de cartões CRC (classe-responsabilidade-colaborador) para identificar e organizar as classes de objetos relevantes para o incremento do software. Outro aspecto muito relevante da XP é que a fase de projeto acontece continuamente enquanto o sistema está em elaboração, na medida que os códigos são constantemente refatorados (modificados/otimizados) para a melhoria contínua do sistema.

Após o levantamento das histórias e a construção do projeto de software, os desenvolvedores elaboram testes de unidade das funcionalidades a serem construídas nos códigos. Isso é uma das mais importantes inovações da XP e garante que os códigos podem ser testados ainda quando estão sendo escritos, evitando um gasto maior de tempo (SOMMERVILLE, 2011). Além disso, uma prática essencial da XP na fase de codificação é a "*programação em pares*", em que dois desenvolvedores trabalham juntos na criação do código ao mesmo passo que revisam o trabalho desenvolvido pelo outro, garantindo soluções inteligentes e com maior qualidade (PRESSMAN, 2021).

Por fim, após criados os códigos eles deverão passar tanto pelos testes de unidade dos desenvolvedores, quanto nos testes de aceitação desenvolvidos de acordo com os requisitos das histórias feitas pelos clientes (PRESSMAN, 2021). É importante ressaltar que a automação dos testes é essencial para o desenvolvimento da fase de testes da XP, garantindo experimentações rápidas e replicáveis para os códigos.

Figura 10 - Método Extreme Programming



Fonte: Pressman, 2021

3.3.4. Gerenciamento Ágil de Projetos (Scrum)

De acordo com (PRESSMAN, 2021), Scrum é um método ágil de desenvolvimento de software muito difundido atualmente. Ele foi inicialmente concebido por Jeff Suntherland juntamente de sua equipe de desenvolvimento no início dos anos 1990. Apesar disso, (SCHWABER; BEEDLE, 2001), realizaram incrementos ao framework original, dando forma ao que é conhecido hoje como scrum e consolidando ele como um método formal de gerenciamento ágil.

Segundo (SCHWABER, 2004), Scrum é uma metodologia ágil de gerenciamento de projetos complexos e de desenvolvimento de produtos. Ele atua de maneira incremental, interativa e objetiva, com o propósito de trazer uma nova dimensão na capacidade de resposta e adaptabilidade na gestão de processos. Para (AUDY, 2015), o Scrum deve ser aplicado no gerenciamento de projetos com equipes de pequeno porte, ágeis, multidisciplinares, auto-organizados e com foco intenso em melhoria contínua das atividades e entregas.

A abordagem Scrum (SCHWABER; BEEDLE, 2001) é um método ágil no geral, mas o seu foco principal está no gerenciamento do desenvolvimento de iniciativas interativas. Segundo (SOMMERVILLE, 2011), diferente das abordagens técnicas específicas da engenharia de software ágil, o Scrum não prescreve práticas técnicas de programação como programação em pares e testes automatizados, observadas

no framework “Extreme Programming”, por exemplo. Desta forma, ele pode ser combinado a outras abordagens mais técnicas de desenvolvimento para fornecer um framework ágil de gerenciamento de projetos de desenvolvimento de software.

Para (PRIKLADNICKI; WILLI; MILANI, 2014), o Scrum é bem aplicado a ambientes com alta variação de requisitos e grande imprevisibilidade no contexto do projeto, proporcionando uma entrega eficaz que se adapta à realidade das mudanças. Entre as boas práticas para lidar com os ambientes de mudanças constantes, está a priorização das funcionalidades de maior valor e a revisão constante da necessidade de desenvolvimento das menos prioritárias (PRIKLADNICKI; WILLI; MILANI, 2014).

No trabalho de (DE CARVALHO; MELLO, 2012), foi feito um levantamento, a partir de revisões da literatura, dos maiores benefícios na utilização do Scrum em projetos de software. Entre os nove benefícios identificados, destacam-se os três principais:

- Melhoria da comunicação e aumento da colaboração entre envolvidos;
- Melhoria da qualidade do produto produzido;
- Aumento da produtividade da equipe.

Além disso, os estudos de (RISING; JANOFF, 2000) produzidos em um ambiente de desenvolvimento de software de telecomunicações, ainda destacam melhorias atreladas ao uso do Scrum como:

- Diminuição no atraso médio dos projetos de software;
- Aumento na confiança estabelecida entre clientes e desenvolvedores ao longo das atividades;
- Melhoria no gerenciamento e compreensão das partes de produtos complexos;

Segundo (SOMMERVILLE, 2011), o Scrum possuí três fases principais. A primeira, de planejamento, em que se estabelece os objetivos gerais do projeto e arquitetura do software a ser desenvolvido. A segunda, de execução, dividida em uma série de rodadas chamadas de *sprints* (ciclos de desenvolvimento que incrementam o sistema). E por fim, a fase final de consolidação das documentações do projeto e avalição das lições aprendidas.

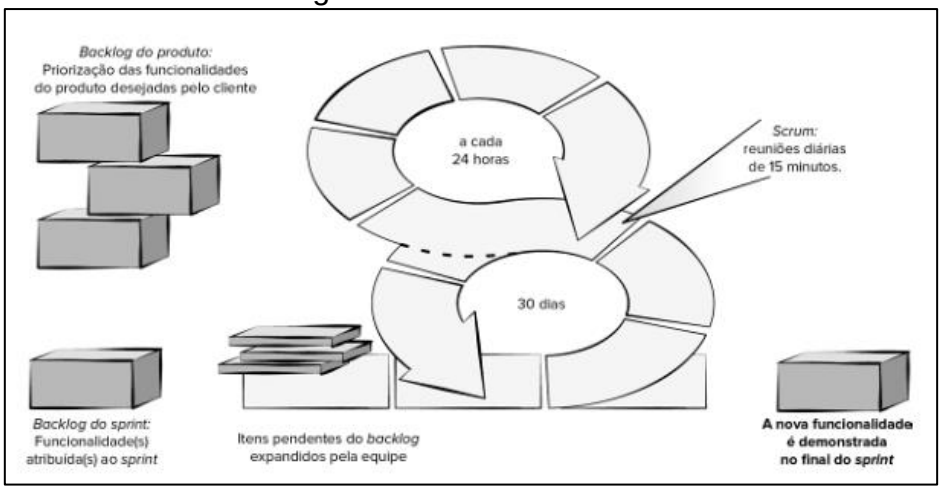
Na fase inicial, de planejamento e levantamento de arquitetura, os requisitos são apresentados no *Product Backlog*, um documento que descreve e consolida todo

o trabalho (funcionalidades, subsistemas etc.) necessário para a construção do produto. Este *Product Backlog* é totalmente dinâmico e evoluí de acordo com o desenvolvimento do projeto. Por fim, o *Product Backlog* é dividido de acordo com a priorização de execução das funcionalidades, criando os *Sprint Backlogs* (conjunto de funcionalidades ou itens a serem executados nas sprints (SCHWABER; SUTHERLAND, 2020).

De acordo com (SCHWABER; SUTHERLAND, 2020), o *Scrum Team* é o grupo/equipe responsável pela execução de um projeto através da metodologia Scrum, sendo dividido em três papéis principais. O primeiro papel, de *Product Owner*, tem a responsabilidade de gerenciar o *Product Backlog* e garantir que todos os itens necessários para a construção do produto sejam contemplados. O segundo papel é o do time de desenvolvimento, pautado nas práticas de autogerenciamento, eles têm a autonomia para decidir qual é a melhor forma de executar o desenvolvimento e incrementos necessários do produto. Por fim, o *Scrum Master* possui o papel de garantir que todos no projeto tenham o entendimento e executem da forma correta as práticas e eventos do Scrum.

Segundo (SOMMERVILLE, 2011), a característica inovadora do Scrum está no seu método de execução em *sprints*, que ao final de cada ciclo entrega uma nova funcionalidade completa ao cliente. Dentro da fase central de execução da metodologia existem os eventos do Scrum, que promovem oportunidades para a equipe realizar inspeções e adaptações no projeto. Os principais eventos do método são a *Sprint Planning* (Reunião de Planejamento da Sprint), *Daily Scrum* (Reunião Diária de Alinhamento), *Sprint Review* (Revisão da *Sprint*) e a *Sprint Retrospective* (Retrospectiva da *Sprint*) (PRIKLADNICKI; WILLI; MILANI, 2014).

Figura 11 - Método Scrum



Fonte: Pressman, 2021

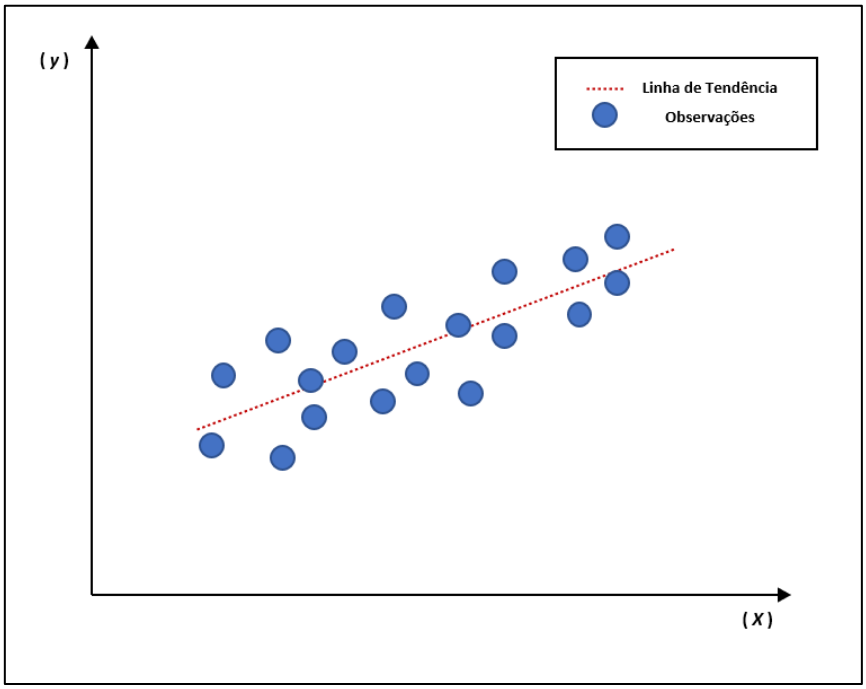
3.4. Modelos de Previsão de Resultados para Projetos de Software

3.4.1. Regressões Lineares

Muitos problemas no campo das engenharias e na ciência no geral envolvem a exploração das relações entre duas ou mais variáveis. Análises de regressão são uma das técnicas estatísticas utilizáveis para modelar e investigar essas relações entre variáveis. Além disso, podem ser utilizados para prever resultados de sistemas, bem como otimizar eles, encontrando os pontos de maximização de resultados dentro da relação entre as variáveis (MONTGOMERY; RUNGER, 2021).

Analisando a figura 12, podemos observar a relação entre duas variáveis, X e Y, em que cada ponto plotado no sistema bidimensional de coordenadas é um par de resultados (X,Y). Observando mais detalhadamente a figura, é possível constatar que nenhuma reta ou curva simples passará exatamente por todos os pontos. Apesar disso, há uma forte indicação de que os pontos estão aleatoriamente distribuídos em torno de uma linha reta.

Figura 12 - Gráfico de Dispersão das Observações



Fonte: Autor

Desta forma, é razoável considerar que a média da variável aleatória Y pode ser dada pela seguinte relação linear:

$$E(Y|x) = \beta_0 + \beta_1 x$$

Neste modelo, a intersecção (β_0) e a inclinação (β_1) da linha são chamados de coeficientes de regressão. É possível observar que a média de Y é uma função linear de X, porém o valor real observado não está posicionado perfeitamente em cima da reta. A maneira correta de se analisar essa relação em um modelo probabilístico, é considerar que o valor real de Y é dado pela função do valor médio (modelo linear abaixo) mais um termo de erro aleatório (∈) (BUSSAB; MORETTIN, 2010).

$$Y = \beta_0 + \beta_1 x + \in$$

Este modelo representado acima pode ser chamado de **modelo de regressão linear simples**, pois ele apresenta apenas uma variável independente (X). Apesar disso, existem casos em que existem duas ou mais variáveis independentes no modelo, sendo chamados assim de **modelos de regressão linear múltipla** (MONTGOMERY; RUNGER, 2021).

Muitas vezes, a relação linear observada será proveniente de um modelo teórico em que os coeficientes de relação (β_0, β_1) são conhecidos. Apesar disso, muitas vezes a identificação dessa relação linear é feita através da análise de

diagramas de dispersão, sendo necessária a estimação dos coeficientes envolvidos na relação, bem como a realização de testes para verificar a adequação do modelo. Por isso, pode-se dizer que os modelos de regressão linear são modelos empíricos (MONTGOMERY; RUNGER, 2021). Ao longo dos próximos dois subtópicos, serão abordados os processos de estimação dos coeficientes e os testes necessários para garantir que o modelo é estatisticamente válido (tanto para regressões lineares simples quanto para as múltiplas).

Vale ressaltar que muitas vezes modelos de regressão linear são aplicados de maneira errônea no estudo de fenômenos. Uma forte associação linear observada em um gráfico de dispersão nem sempre é um indicativo de relação causal entre as variáveis. Por isso é sempre necessário planejar o experimento e investigar a relação de origem das variáveis observadas (MONTGOMERY; RUNGER, 2021).

3.4.1.1. Regressões Lineares Simples

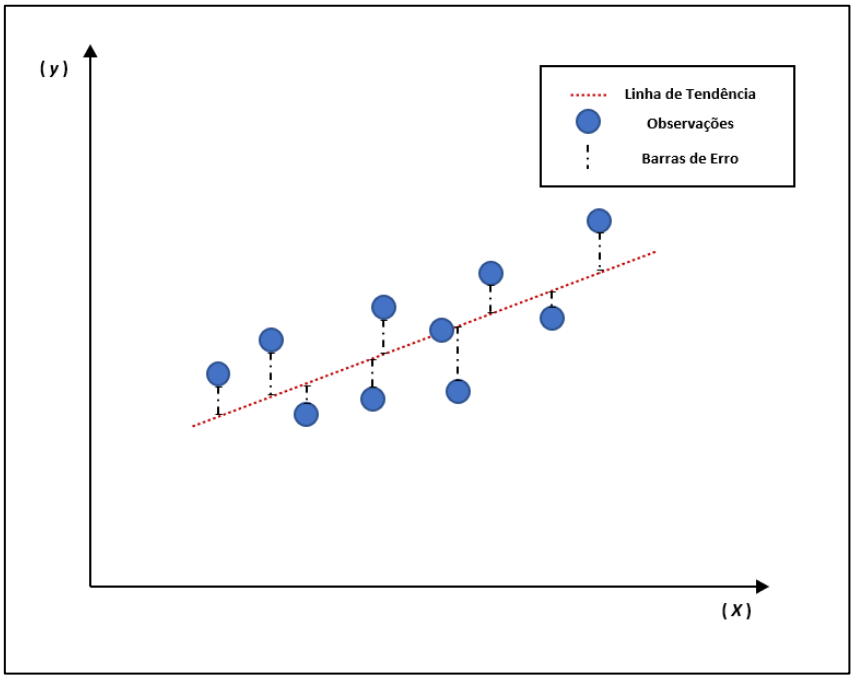
Como foi exposto no tópico anterior, modelos de regressão linear simples são aqueles que possuem apenas um preditor (X) e uma variável dependente ou de resposta (Y). Desta forma, são usados para modelar sistemas em que o comportamento de uma variável é explicado diretamente por uma única outra variável, em uma relação de 1:1.

Desta forma, relações deste tipo de modelo podem ser descritas pela equação abaixo, assim como foi exemplificado no tópico anterior. É importante ressaltar que o coeficiente de erro aleatório ($\in$), possuí média zero e uma variância σ^2. Os erros aleatórios das diferentes observações também são considerados variáveis aleatórias não correlacionadas (MONTGOMERY; RUNGER, 2021).

$$Y = \beta_0 + \beta_1 x + \in$$

Na figura 13, é possível observar um gráfico de dispersão e uma possível linha estimada de regressão. Os traços que ligam os pontos à linha de regressão são os erros aleatórios presentes no modelo através do termo ($\in$). Na teoria, o melhor modelo de regressão é aquele em que a soma dos desvios/erros verticais é a menor possível.

Figura 13 - Gráfico de dispersão das Observações com Barras de Erro



Fonte: Autor

Método dos Mínimos Quadrados

Visando criar um modelo de regressão com os menores erros possíveis, o matemático Karl Gauss (1777 – 1855) propôs um modelo de estimação dos parâmetros β_0 e β_1 de forma a minimizar a soma dos quadrados dos desvios verticais. Este método de estimação dos coeficientes de uma regressão é chamado de **método dos mínimos quadrados** (MONTGOMERY; RUNGER, 2021).

Através deste método, obtém-se as melhores estimativas teóricas possíveis para um modelo de regressão linear, dadas por $\widehat{\beta_0}$ e $\widehat{\beta_1}$ (BUSSAB; MORETTIN, 2010). As equações abaixo descrevem como esses estimadores são obtidos:

$$\widehat{\beta_0} = \bar{y} - \widehat{\beta_1}\bar{x}$$

$$\widehat{\beta_1} = \frac{\sum_{i=1}^{n} y_i x_i - \frac{(\sum_{i=1}^{n} y_i)(\sum_{i=1}^{n} x_i)}{n}}{\sum_{i=1}^{n} x_i^2 - \frac{(\sum_{i=1}^{n} x_i)^2}{n}}$$

Em que:

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Desta forma, a melhor linha estimada para uma regressão linear é dada por:

$$\hat{y} = \widehat{\beta_0} - \widehat{\beta_1} x$$

Em que a diferença entre o y observado real (y_i) e o y previsto pelo modelo ($\hat{y}_i$) é chamada de resíduo (e_i).

Estimando a Variância do Termo de Erro (σ^2)

Uma vez estimados os parâmetros β_0 e β_1, ainda há um outro parâmetro a ser definido no modelo de regressão, a variância do termo de erro aleatório ∈ (σ^2). Para calcular a variância, utiliza-se a soma do quadrado dos resíduos, dada por:

$$SQ_E = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} e_i^2$$

Através da SQ_E podemos dizer que um estimador não tendencioso para a variância do termo de erro é dado por:

$$\hat{\sigma}^2 = \frac{SQ_E}{n-2}$$

Testes de Hipótese para Regressões Lineares Simples (Testes T)

Uma vez definidos os principais parâmetros de um modelo de regressão, é importante verificar a adequação do modelo construído. Isso pode ser feito através de testes estatísticos de hipóteses em relação aos parâmetros do modelo (MONTGOMERY; RUNGER, 2021). Para testar hipóteses a respeito da inclinação e intersecção do modelo, é necessário fazer a suposição de que o componente de erro (∈) seja distribuído normalmente. Desta forma, tem-se a suposição completa de que os erros são normais e independentemente distribuídos, com média 0 e variância σ^2.

Podem ser realizados testes T para diversas hipóteses como (1) A inclinação da reta (β_1) ser uma constante, (2) Valor de intersecção da reta com o eixo Y (β_0) ou (3) Valor de inclinação da reta ser igual a 0, sendo a última uma das mais importantes (MONTGOMERY; RUNGER, 2021). Essas hipóteses se relacionam com a significância da regressão. Uma vez que não se consegue rejeitar a hipótese H_0 é equivalente a se dizer que não há relação linear entre X e Y.

Tendo como exemplo a hipótese 3 mencionada no parágrafo anterior, tem-se a seguinte estrutura para o teste T:

$$H_0 : \beta_1 = 0$$

$$H_1 : \beta_1 \neq 0$$

Partindo da definições e propriedades dos coeficientes estimados, em que ∈ é N($0, \sigma^2$), segue diretamente que as observações y_i são N($\beta_0 + \beta_1 x_i, \sigma^2$). Desta forma, $\widehat{\beta_1}$ é uma combinação linear de variáveis aleatórias normais independentes, dada por N($\beta_1, \sigma^2/S_{xx}$) (MONTGOMERY; RUNGER, 2021). Desenvolvendo as propriedades dos estimadores e sabendo que $(\mathrm{n}-2)\hat{\sigma}^2/\sigma^2$ segue uma distribuição qui-quadrada, com n-2 graus de liberdade temos que a estatística T é:

$$T_0 = \frac{\widehat{\beta_1}}{\sqrt{\hat{\sigma}^2/S_{\mathrm{xx}}}}$$

Sabendo que T_0 segue a distribuição T com n-2 graus de liberdade. A hipótese H_0 é rejeitada caso

$$|T_0| > t_{\frac{\alpha}{2},n-2}$$

Rejeitando a hipótese H_0, conclui-se que há uma relação linear entre as duas variáveis e que X é relevante para explicar os resultados de Y. Adicionalmente ao valor de T, caso o P-Valor obtido pela análise seja menor que 0,05 (quando usado um intervalo de confiança de 95%) também é um indicativo da relação linear entre as variáveis (MONTGOMERY; RUNGER, 2021).

Testes de Adequação para Regressões Lineares Simples

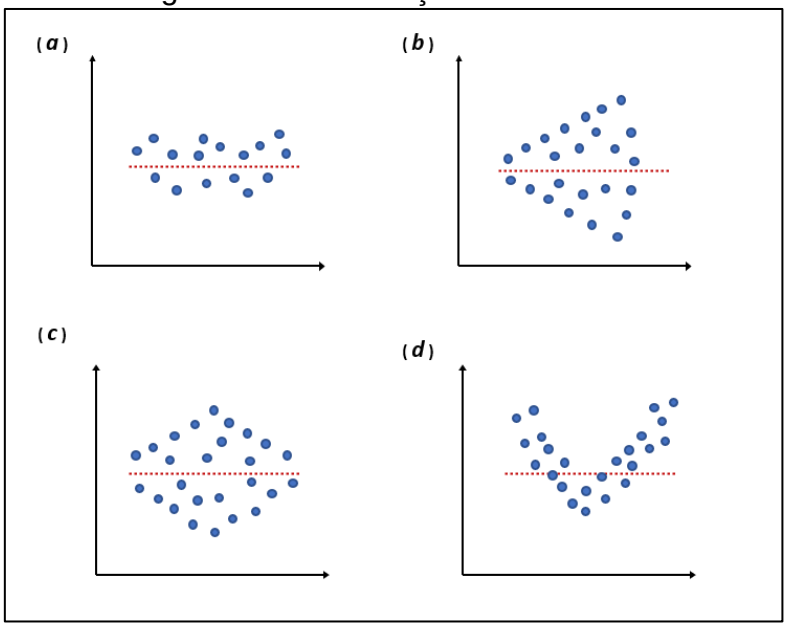
Análise Residual

Como foi observado até então, ajustar um modelo de regressão é estimar os coeficientes envolvidos nele, envolvem uma série de suposições. É necessário que os erros sejam variáveis aleatórias não correlacionadas com média zero e variância constante. Além disso, parte-se do pressuposto de que o modelo está descrevendo uma relação linear real entre as variáveis. Apesar disso, é necessário sempre realizar testes de adequação para verificar se de fato essas suposições são reais.

Um dos principais testes de adequação de um modelo de regressão é a análise residual. Este tipo de análise é aplicado para verificar se a suposição de que os erros da regressão são distribuídos de acordo com a normal e com variância constante é correta.

Um bom método para a verificação aproximada da normalidade dos resíduos é a construção de um histograma de frequência ou um gráfico de probabilidade normal dos resíduos. Caso a suposição a respeito da distribuição dos resíduos esteja correta, o histograma deve ter o comportamento aproximado da curva de sino ditada pela normal. Além disso, é bastante útil plotar os resíduos em gráficos com (1) uma sequência temporal conhecida, (2) valores de $\hat{y}_i$ e (3) variável independente X, como pode ser visto nos exemplos abaixo.

Figura 14 - Distribuição de Resíduos



Fonte: Autor

O gráfico (a) representa a situação ideal para a dispersão dos resíduos, enquanto os gráficos (b), (c) e (d) apresentam anomalias. Alterações no modelo como transformações das variáveis, para os casos (b) e (c), ou adições de termos com ordem maior que 1, para o caso (d), podem ser alternativas para a correção do modelo.

Coeficiente de Determinação (R^2)

Um outro método para se verificar a adequação de um modelo é o cálculo do coeficiente de determinação ou também chamado de R^2. O coeficiente de determinação é o quadrado do coeficiente de correlação entre duas variáveis aleatoriamente distribuídas de um modelo (MONTGOMERY; RUNGER, 2021). O coeficiente de determinação é tido como a quantidade da variabilidade da variável resposta que é explicada/considerada no modelo de regressão. Desta forma, valores altos de R^2 são um indicativo de um modelo "bem" ajustado e que explica a variável resposta de maneira "completa".

O coeficiente de determinação pode ser obtido através da seguinte fórmula:

$$R^2 = 1 - \frac{SQ_E}{SQ_T}$$

Em que que SQ_T é igual a:

$$SQ_T = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

Apesar do R^2 dar um bom indicativo da adequação do modelo e da sua capacidade de explicar a variável resposta, é necessário ter alguns cuidados na sua interpretação. A adição de novas variáveis ou termos ao modelo podem aumentar o valor do coeficiente sem necessariamente aumentar a qualidade da regressão. É necessário avaliar o modelo como um todo e verificar se os resultados estão coerentes com o observado na realidade (MONTGOMERY; RUNGER, 2021).

3.4.1.2. Regressões Lineares Múltiplas

Diversas aplicações de regressões lineares envolvem sistemas com duas ou mais variáveis preditoras. Modelos de regressão com essa característica, são chamados de modelos de regressão múltipla (MONTGOMERY; RUNGER, 2021). De maneira geral, os sistemas de regressão múltipla podem ser descritos pela seguinte

estrutura, onde a variável dependente y, pode estar relacionada a K variáveis independentes:

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k + \in$$

Neste modelo, os parâmetros β_j, j = 0, 1, 2, ... k, são tidos como os coeficientes da regressão, assim como no modelo de regressão linear simples. Por definição, o parâmetro β_i representa a variação esperada na variável resposta y por unidade de variação em x_i, quando todas as outras variáveis regressoras são mantidas constantes. Além disso, por se tratarem de múltiplas variáveis, o modelo passa a descrever um hiperplano em um espaço com k dimensões.

Modelos deste tipo são usados de forma frequente como funções de aproximação. Ou seja, a relação funcional real que envolve as variáveis y e x_1, x_2, ... x_k é desconhecida. Porém, é possível identificar faixas das variáveis independentes em que o modelo de regressão fornece uma aproximação adequada para o comportamento observado, sendo assim utilizado para descrever essas relações.

Além disso, modelos com efeitos de interação entre as variáveis independentes também podem ser ajudados para um modelo de regressão múltipla. Por exemplo, observe o modelo XX.

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2 + \in$$

É possível definir que $x_3 = x_1 x_2$ e que $\beta_3 = \beta_{12}$ e então se obtém o seguinte modelo linear:

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \in$$

Este exemplo é interessante para demonstrar que mesmo como modelo linear, a forma da superfície gerada por ele em um espaço de três dimensões é não-linear. De maneira geral, pode-se dizer que qualquer modelo de regressão que seja linear nos parâmetros (β_k) é um modelo de regressão linear, independente da forma gerada por ele.

Método dos Mínimos Quadrados

Assim como no método de regressão linear, o método dos mínimos quadrados desenvolvido por Gauss também pode ser aplicado em regressões múltiplas para estimar os coeficientes de regressão do modelo. Tomando como exemplo o seguinte modelo, tem-se:

$$y_1 = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik} + \in_i$$

Também demonstrado por:

$$y_1 = \beta_0 + \sum_{j=1}^{k} \beta_j x_{ij} + \in_i$$

Desta forma, obtém-se a função dos mínimos quadrados para modelos de regressão múltipla, dada por:

$$L = \sum_{i=1}^{n} \in_i^2 = \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{k} \beta_j x_{ij})^2$$

Desenvolvendo os termos dessa função, chega-se as equações normais de mínimos quadrados, demonstradas abaixo. Vale ressaltar que vai haver uma para cada coeficiente da regressão. A solução para o sistema de equações normais é o conjunto de coeficientes mais adequados para a regressão. Por se tratar de um sistema de equações lineares, ele pode ser resolvido através de qualquer método adequado para tal situação (MONTGOMERY; RUNGER, 2021).

$$n\hat{\beta}_0 + \hat{\beta}_1 \sum_{i=1}^{n} x_{i1} + \hat{\beta}_2 \sum_{i=1}^{n} x_{i2} + \ldots + \hat{\beta}_k \sum_{i=1}^{n} x_{ik} = \sum_{i=1}^{n} y_i$$

$$\hat{\beta}_0 \sum_{i=1}^{n} x_{i1} + \hat{\beta}_1 \sum_{i=1}^{n} x_{i1}^2 + \hat{\beta}_2 \sum_{i=1}^{n} x_{i1}x_{i2} + \cdots + \hat{\beta}_k \sum_{i=1}^{n} x_{i1}x_{ik} = \sum_{i=1}^{n} x_{i1}y_i$$

... ...

$$\hat{\beta}_0 \sum_{i=1}^{n} x_{ik} + \hat{\beta}_1 \sum_{i=1}^{n} x_{ik}x_{i1} + \hat{\beta}_2 \sum_{i=1}^{n} x_{ik}x_{i2} + \cdots + \hat{\beta}_k \sum_{i=1}^{n} x_{ik}^2 = \sum_{i=1}^{n} x_{ik}y_i$$

Estimando a Variância do Termo de Erro (σ^2)

Assim como na regressão linear simples, para se obter todos os parâmetros do modelo construído, é necessário estimar a variância do termo de erro ∈ aleatório da regressão (σ^2). Vale relembrar que na regressão linear simples, o estimador da variância foi obtido através da divisão da soma dos quadrados dos resíduos por n-2, já que haviam dois parâmetros no modelo. Logo, para uma regressão linear múltipla com *p* parâmetros, um estimador não tendencioso da variância do termo de erro é dado por:

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^{n} e_i^2}{n-p} = \frac{SQ_E}{n-p}$$

Como se observa na regressão linear simples, o numerador da equação é chamado de soma dos quadrados dos resíduos ou erro. Já o denominador *n – p* é o grau de liberdade do erro ou do resíduo.

Testes de Significância para Regressões Lineares Múltiplas (Teste F)

Tal como na regressão linear simples, testes de hipótese relativos aos parâmetros do modelo são importantes para a verificar a adequação da regressão. Assim como na regressão simples, os testes de hipóteses exigem que o termo de erro ($\in_i$) seja normalmente distribuído, com média 0 e variância σ^2.

Um dos testes mais relevantes para aferir a adequação do modelo é o de significância da regressão. Ele é usado para determinar se há uma relação linear entre a variável resposta y e o conjunto de regressores x_1, x_2,... x_k. Para isso, as hipóteses do teste são:

$$H_0 : \beta_1 = \beta_2 = \cdots = \beta_{\mathrm{k}} = 0$$

$$H_1 : \beta_{\mathrm{j}} \neq 0$$

A rejeição de H_0 exige apenas que ao menos uma das variáveis preditoras contribua significativamente para o modelo (MONTGOMERY; RUNGER, 2021). Desta forma, desenvolvendo a relação abaixo de acordo com as propriedades dos coeficientes da regressão

$$SQ_T = SQ_R + SQ_E$$

Obtém-se a fórmula da estatística de teste

$$F_0 = \frac{SQ_R/k}{SQ_E/(n-p)} = \frac{MQ_R}{MQ_E}$$

A condição para a rejeição da hipótese H_0 é que o valor da estatística obtido através da fórmula deve ser maior que $F_{\alpha,p,n-p-1}$. O procedimento de análise da variância geralmente é resumido através da tabela ANOVA (MONTGOMERY; RUNGER, 2021).

Além do teste F, existem outras opções para se avaliar a significância de uma regressão linear, como o teste T de Student. Neste caso, o teste T é específico para a avaliação da significância dos coeficientes individuais de uma regressão, enquanto o teste F pode ser usado para avaliação global (múltiplas variáveis) de um modelo. Por fim, o teste T permite testar hipóteses alternativas unilaterais, sendo mais flexível que o teste F que é restrito a alternativas bilaterais (MONTGOMERY; RUNGER, 2021).

Coeficiente de Determinação (R^2) e R^2 Ajustado

Da mesma forma que na regressão linear, também é possível utilizar o coeficiente de determinação múltipla (R^2) como uma estatística global para avaliar o ajuste do modelo. Como foi dito no tópico de regressões lineares, o R^2 avalia, de maneira geral, a porcentagem de explicação da variável resposta através do modelo construído. Desta forma, R^2 é dado por:

$$R^2 = \frac{SQ_R}{SQ_T} = 1 - \frac{SQ_E}{SQ_T}$$

Como foi mencionado no tópico de regressões lineares, o uso sem consciência do coeficiente de determinação traz alguns problemas. Visto que o coeficiente aumenta de acordo com a inserção de novas variáveis ao modelo (sem necessariamente melhorar o modelo em si), sistemas de regressão múltipla tendem a ser mais impactados por esse efeito uma vez que possuem várias variáveis. Pode ser difícil julgar se o aumento do coeficiente é explicado por uma melhora na qualidade do modelo ou se é devido ao efeito natural de aumento por inserção de novas variáveis (MONTGOMERY; RUNGER, 2021).

Desta forma, é preferível o uso do R^2 ajustado, demonstrado pela fórmula abaixo. Uma vez que o numerador é composto pela média quadrática do erro e o

denominador é uma constante, o R^2 ajustado só irá aumentar em caso de redução do numerador. Isto é, caso a nova variável reduza a média quadrática do erro, em outras palavras, tornando o modelo mais preciso (MONTGOMERY; RUNGER, 2021).

$$R^2_{ajustado} = 1 - \frac{SQ_E/(n-p)}{SQ_T/(n-1)}$$

Análise de Resíduos

Assim como na regressão simples, a análise dos resíduos desenvolve um importante papel na verificação da adequação do modelo. Essa análise pode ser feita da mesma forma para as regressões simples, plotando os resíduos contra (1) uma sequência temporal conhecida, (2) valores de $\hat{y}_i$ e (3) variável independente X. O comportamento desses gráficos pode trazer conclusões sobre possíveis ajustes necessários no modelo (MONTGOMERY; RUNGER, 2021).

Além da análise usual dos resíduos, é possível avaliá-los através da padronização dos resíduos. Essa padronização escalona os resíduos de modo a aproximar os desvios padrão deles para 1. Desta forma, resíduos grandes serão mais nítidos a partir de uma inspeção visual e poderão indicar possíveis outliers (MONTGOMERY; RUNGER, 2021). A fórmula de padronização dos resíduos é descrita abaixo:

$$d_1 = \frac{e_i}{\sqrt{MQ_E}} = \frac{e_i}{\sqrt{\hat{\sigma}^2}}$$

3.5. Análise Bibliométrica

3.5.1. Resultados da Análise Bibliométrica

3.5.1.1. Análise Quantitativa

A análise quantitativa deste trabalho utilizou todos os 308 artigos recolhidos através da busca bibliométrica realizada na base Scopus. Ela tem como objetivo identificar macrotendências e características da amostra de publicações obtida. Essa análise foi dividida em quatro campos principais para uma melhor segmentação e

visualização dos resultados, sendo eles: (1) Resultados Gerais (compreendendo ano e país de publicação); (2) Resultados de Editoras; (3) Resultados de Autores e (4) Resultados de Palavras-Chave.

3.5.1.2. Resultados Gerais

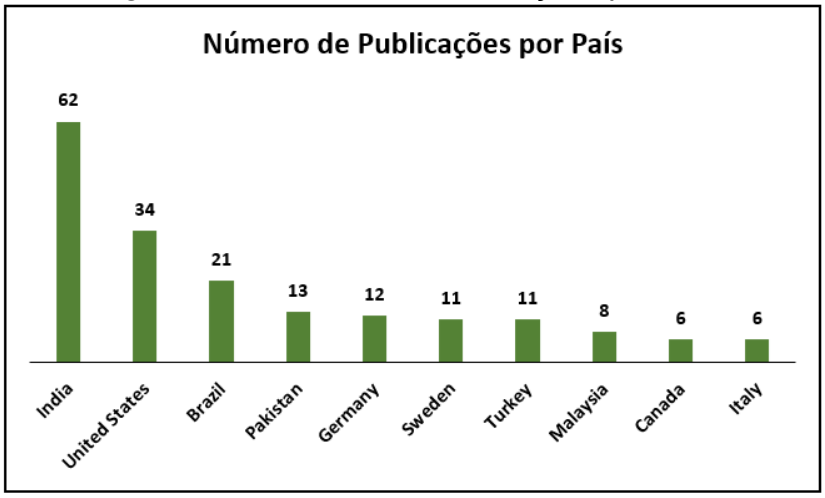
A relevância do tema pesquisado foi analisada através da contagem de publicações ao longo dos anos, Figura 15, e em termos da distribuição dessas publicações nos seus países de origem, Figura 16. Através dessas análises é possível identificar tendências da evolução dos estudos no tema ao longo do tempo, bem como criar hipóteses sobre quais países tem se dedicado mais a este campo de conhecimento.

Figura 15 - Número de Publicações por Ano

Fonte: Autor

A Figura 15 mostra a distribuição das publicações da amostra ao longo dos anos recortados para o estudo bibliométrico (2013 a 2023). Através de sua análise é possível identificar uma tendência de crescimento no número de publicações ao longo dos anos. A grande queda no ano de 2020 está possivelmente ligada à pandemia mundial de Covid-19. Além disso, é provável que a baixa no ano de 2023 se deva ao fato da amostra ter sido retirada ao longo do ano, com quase três meses antes de seu fim. É possível que a tendência de aumento no número de publicações se deva a um maior interesse dos pesquisadores em entender e propor novas e melhores soluções para a estimação de resultados em projetos de software ágeis.

Figura 16 - Número de Publicações por País



Fonte: Autor

A Figura 16 mostra a distribuição das publicações nos dez países mais relevantes da amostra, através da contabilização do país das instituições científicas por trás de cada trabalho. Observa-se que os dez países com mais publicações, são responsáveis por quase 60% do volume total de trabalhos presente na amostra de estudo, somando 184 publicações. Neste contexto, é possível observar uma grande dominância de instituições indianas em termos de volume de publicações, seguidas por instituições americanas e brasileiras, respectivamente. Além disso, outros quarenta e quatro países foram identificados na amostra, responsáveis por cerca de 40% do volume de publicações observado. Vale ressaltar a posição de destaque do Brasil dentro da amostra avaliada, ocupando o terceiro lugar do ranking de países com mais publicações, somando um total de 21 trabalhos.

3.5.1.3. Resultados de Editoras

A fim de analisar o panorama de publicações sobre o tema estudado, realizou-se um aprofundamento à respeito das instituições editoras dos trabalhos. Na tabela abaixo, é possível ver a distribuição de trabalhos entre as dez instituições de edição mais relevantes na amostra.

Tabela 2 - Editoras mais relevantes

Publicadores	Número de Artigos	Porcentagem	Número de Citações
Institute of Electrical and Electronics Engineers Inc.	67	22%	540
Association for Computing Machinery	20	6%	186
Springer Verlag	17	6%	106
Springer Science and Business Media Deutschland GmbH	17	6%	31
IEEE Computer Society	15	5%	249
Elsevier B.V.	15	5%	208
Springer	10	3%	40
Knowledge Systems Institute Graduate School	10	3%	26
Elsevier Inc.	7	2%	333
CEUR-WS	6	2%	16

Fonte: Autor

É possível observar que as dez editoras mais relevantes somam juntas 184 publicações e 1735 citações em seus diferentes trabalhos, acumulando cerca de 60% do número de publicações da amostra. Além disso, foram identificadas outras 74 instituições de edição, que somam juntos 124 trabalhos. Assim como na análise de publicações por país, mostrada no tópico anterior, é possível identificar uma grande concentração no volume de trabalhos em alguns grupos editoriais. Apesar de possuírem diferentes entidades editoras individuais, observa-se uma grande concentração dos grupos IEEE (Institute of Electrical and Electronics Engineers), Springer e Elsevier, sendo estas, respectivamente, as editoras mais relevantes dentro da amostra.

3.5.1.4. Resultados de Autores

Também foi feita uma análise para identificar os autores mais relevantes dentro do tema estudado. Para isso, foram calculadas as médias de citação anual de todos os autores, apenas os que possuíam média global maior que a média geral da amostra (1,68 citações por ano por autor) foram considerados. A tabela abixo mostra os dez autores que atendem ao requisito de média de citações por ano com maior número de publicações na base. Juntos, eles possuem uma média de 5,12 citações por ano, bem maior do que a média global da amostra.

Tabela 3 - Autores mais relevantes

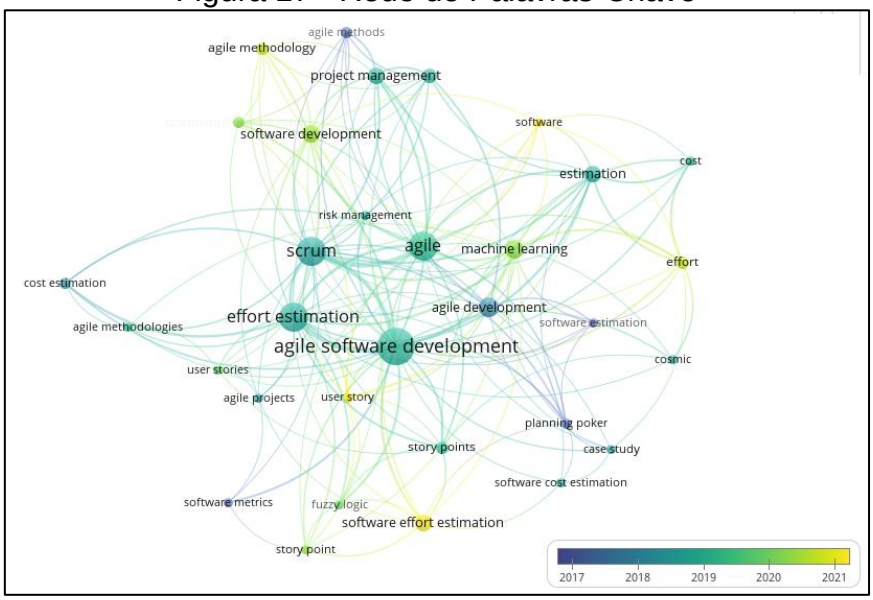
Autores	Número de Artigos
Demirors O.	8
Perkusich M.	8
Chauhan N.	6
Verma S.	6
Hacaloglu T.	6
Misra S.	6
Mendes E.	5
Mashkoor A.	5
Moussa R.	5
Celar S.	4

Fonte: Autor

3.5.1.5. Resultados de Palavras Chaves

Por fim, foram analisadas as palavras chaves das publicações para a identificação dos temas mais relevantes dentro desta área de conhecimento. Através do estudo dos termos chaves, é possível identificar as tendências de pesquisa na área e traçar panoramas para o futuro do tema. Para uma análise mais rica dos termos-chave, foram excluídas da tabela de palavras-chave as *Strings* de busca utilizadas na pesquisa bibliométrica. Desta forma, observa-se apenas as principais palavras-chave relacionadas com os termos de busca. Na Figura 17 é possível observar a rede de palavras gerada pelos termos chave da amostra analisada.

Figura 17 - Rede de Palavras-Chave



Fonte: Autor

É possível observar na Figura 17 que a maioria dos termos chave possuem uma coloração referente à média do recorte de tempo da análise, o que pode indicar que estes termos são usados de forma perene nos estudos deste campo. Apesar disso, os termos em amarelo vivo, como “Software Effort Estimation” e “User Story” começaram a ser mais usados nos anos finais do recorte, o que pode indicar uma tendência futura de estudos nestas áreas. Abaixo, é possível observar os dez termos chaves mais relevantes entre os trabalhos analisados.

Tabela 4 - Palavras-Chave mais relevantes

Palavras Chaves	Contagem
Scrum	47
Effort Estimation	46
Machine learning	22
Project management	17
Estimation	17
Software effort estimation	16
Software Engineering	11
Effort	10
Story points	8
Cost estimation	8

Fonte: Autor

É possível observar que dentro da amostra analisada, o termo chave mais relevante é diretamente ligado ao campo de metodologias ágeis. Isso possivelmente se dá devido à grande presença dessas metodologias no cenário mundial de desenvolvimento de software (PRESSMAN, 2021). Além disso, devido à natureza de mudanças constantes de requisitos e regras de negócio em que são aplicadas essas metodologias, o estudo de previsão de resultados de projetos ágeis pode despertar muito interesse dos pesquisadores ao mesmo tempo que representam uma enorme oportunidade de redução de custos para o mercado.

3.5.1.6. Análise Qualitativa

A fim de se aprofundar mais no campo de conhecimento pesquisado foi conduzida uma análise qualitativa entre os artigos mais relevantes dentro da base de publicações. A análise tinha como objetivo levantar informações mais específicas sobre as publicações e o tema de previsão de resultados em projetos de software

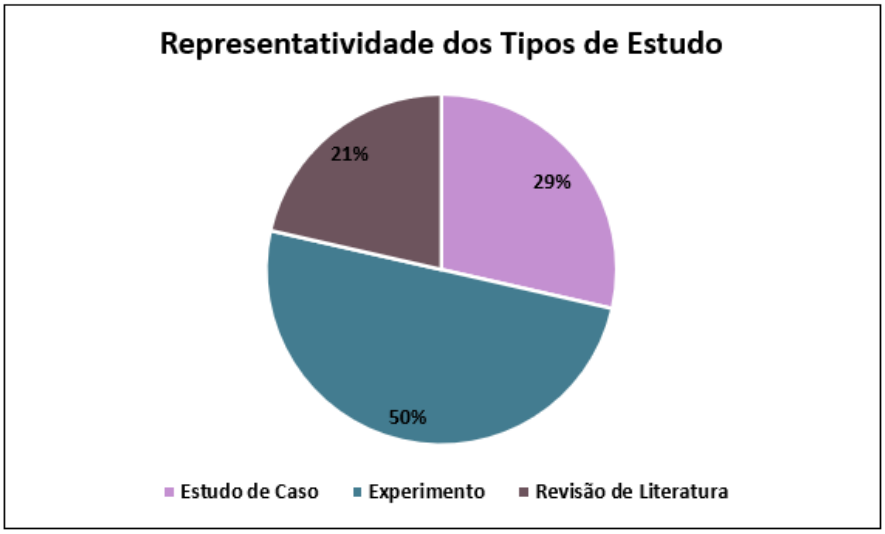
ágeis, como: modelos de previsão utilizados, setor da economia em que o estudo foi realizado, complexidade de aplicação do modelo entre outros.

Para mapear as publicações mais relevantes, utilizou-se a contagem de citações de cada uma delas. Observou-se que 50% do total de citações de toda a base analisada estava concentrado em apenas 26 publicações, tornando-as as mais relevantes para a análise qualitativa. Analisando-as mais detalhadamente, apenas 14 dessas estavam relacionadas especificamente ao tema de previsão de resultados de projetos, as demais discorriam sobre assuntos como metodologias de priorização de atividades, gestão de conhecimento entre outros. Desta forma, o autor se aprofundou apenas nas 14 publicações com tema relacionado ao trabalho para a 1ª etapa do projeto de graduação. Para a 2ª etapa do projeto, as publicações foram analisadas em sua totalidade para identificação de mercados com maior maturidade em desenvolvimento de software, bem como para observar metodologias paralelas envolvidas no desenvolvimento de softwares.

Analisando o detalhamento dos 14 trabalhos escolhidos, é possível observar que 86% deles foram realizados em empresas de desenvolvimento de software de diversos tamanhos. O setor industrial foi abordado por 7% dos estudos e setores diversos também acumularam 7% na amostra da análise. Por se tratar de trabalhos que exigem uma grande quantidade de dados sobre os projetos de desenvolvimento de software, faz sentido que o setor mais buscado pelos pesquisadores seja o mercado raiz de desenvolvimento desses sistemas.

Quando se analisa o tipo de estudo que foi realizado em cada uma dessas publicações, é possível observar uma predominância de experimentos, com cerca de 50% dos trabalhos se encaixando neste grupo. Neste tipo de trabalho analisado, os autores usualmente se utilizam de dados reais a respeito de projetos de desenvolvimento de software ágeis e aplicam modelos estatísticos de predição para a comparação de resultado com a realidade. O segundo modelo de trabalho mais comum na amostra são os “Estudos de Caso”, com 29% de representatividade, seguidos pela categoria de “Revisão de Literatura” com 21%. A distribuição das categorias pode ser observada na Figura 18.

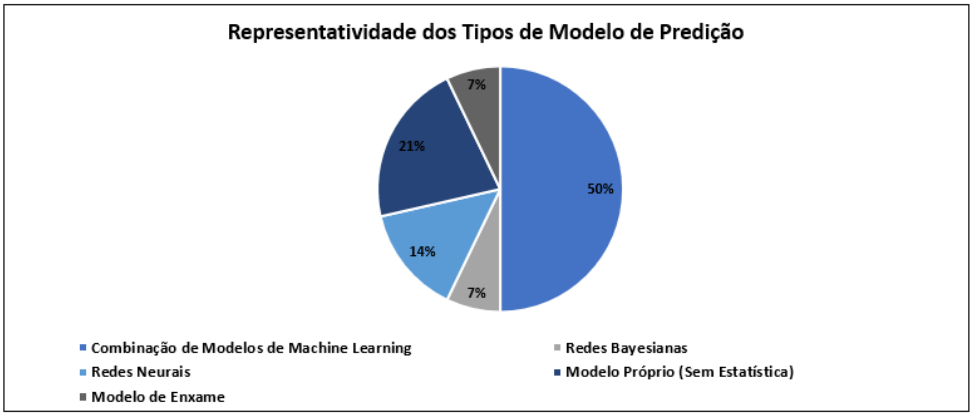
Figura 18 - Representatividade dos Tipos de Estudo



Fonte: Autor

Também foi feita uma análise dos tipos de modelo preditivo utilizados nos estudos da amostra. Foi observada uma grande predominância de estudos que utilizaram mais de um modelo de predição com machine learning em suas análises. Na maioria das vezes os autores testavam vários modelos e comparavam os seus resultados em diferentes cenários, a fim de encontrar a metodologia que melhor descrevia o projeto em questão. Entre os modelos usados nesses estudos estão alguns como: “Redes Neurais”, “Árvore de Decisão”, “Floresta Aleatória” e “Redes Bayesianas”. Em segundo lugar, com 21% representatividade, estão os estudos que desenvolveram o próprio modelo de estimação sem embasamento estatístico. Muitas vezes estão ligados a implementação de novos processos ou checklists e envolvem operações matemáticas simplificadas como o estabelecimento de médias históricas e aplicação delas para o futuro.

Figura 19 - Representatividade dos Modelos de Predição



Fonte: Autor

4. Resultados

Este capítulo tem como objetivo apresentar o desenvolvimento da pesquisa de fato, expondo os principais passos seguidos na aplicação do método e os resultados obtidos através dele. No primeiro subtópico, foi realizada uma breve contextualização da empresa que forneceu os dados para o estudo, bem como o motivo pelo qual ela foi escolhida.

Em seguida, foi apresentado o processo completo de exploração e preparação dos dados para a construção do modelo de regressão. Nele são exploradas as metodologias de limpeza e normalização dos dados necessárias para a aplicação de uma regressão linear.

No terceiro subtópico é exposta a aplicação de fato dos modelos de regressão nos dados já preparados. Detalhando os passos seguidos para a construção e verificação do modelo e expondo os parâmetros e resultados obtidos.

4.1. Contextualização da Empresa

O estudo foi conduzido em uma corretora de valores brasileira de grande porte. A instituição financeira possuí mais de 20 anos de existência e conta com mais de 6 mil colaboradores espalhados pelo Brasil, atendendo mais de 4 milhões de clientes. Por se tratar de uma corretora de valores, seus principais serviços estão ligados a intermediação e oferta de produtos de investimentos de diversos tipos e emissores.

Seguindo as tendências de inovação no mercado financeiro, grande parte da oferta de produtos e relacionamento com os clientes é feita através de uma plataforma digital e um aplicativo mobile para celulares. Desta forma, para dar sustentação aos serviços e produtos oferecidos, é necessário um alto volume de iniciativas constantes de desenvolvimento e manutenção de softwares. Toda essa infraestrutura de desenvolvimento exige um grande número de colaboradores dedicados às disciplinas de tecnologia, bem como altos investimentos na área.

Buscando rapidez e flexibilidade nos processos de desenvolvimento de software, a empresa gerencia suas iniciativas através de metodologias ágeis como o *Scrum* e a *Extreme Programming.* Desta forma, contam com squads independentes, livres para priorizar suas iniciativas de acordo com as necessidades da unidade de negócio em que estão alocadas. Além disso, os ciclos de trabalho, ou seja, os

períodos de reorganização macro dos backlogs e dos membros da squad, acontecem de maneira trimestral. Por fim, entre as principais práticas das metodologias ágeis utilizadas pela empresa, podem se destacar a programação em pares, modularização do projeto em sprints e criação de backlogs de trabalho.

Devido ao alto volume de iniciativas de desenvolvimento de software, a empresa se mostrou viável entre as demais consideradas para a aplicação do estudo. Além disso, metodologias ágeis trazem consigo uma maior dificuldade de previsão de resultados nos projetos que as utilizam. Desta forma, a criação de modelos que consigam prever comportamentos e resultados de projetos ágeis de software, trazem um grande benefício para a indústria de desenvolvimento como um todo. Visto que a empresa unia um alto volume de projetos, bem como a aplicação de metodologias ágeis, ela foi escolhida para o estudo.

4.2. Processo de Exploração e Preparação dos Dados

Neste subtópico serão abordados todos os passos seguidos para analisar e preparar os dados antes da construção do modelo de regressão. Etapas de análise prévia são importantes para identificar possíveis falhas ou comportamentos específicos da base, que podem ter um impacto significativo nos resultados do modelo.

Desta forma, na primeira etapa do subtópico serão detalhadas as diferentes bases e variáveis utilizadas no processo. Já na segunda parte, serão expostas as premissas e restrições adotadas inicialmente, bem como a abordagem para lidar com dados faltantes/”vazios”. Além disso, também serão detalhadas as etapas de verificação da distribuição e normalidade dos dados, assim como a de identificação e retirada de outliers. Por fim, são demonstrados a metodologia e os cálculos para as transformações e criação de novas variáveis utilizadas no modelo.

4.2.1. Caracterização dos Dados

Conforme mencionado no tópico 2.2.3 a respeito do procedimento de acesso aos dados, foram utilizadas duas bases distintas, com informações complementares, que foram fundamentais para a condução do estudo:

- **Base de Dados 1**: Continha informações sobre os épicos de portfólio dos três primeiros trimestres de 2023. Esta base incluía todos os detalhes relacionados

aos épicos, como o squad responsável pela execução, as datas de início e término, o esforço estimado pela equipe, entre outros.

- **Base de Dados 2**: Incluía informações sobre os membros de cada squad nos três primeiros trimestres de 2023. Nesta base, estavam detalhados dados como a squad de alocação e as senioridades dos membros, conforme a tabela de cargos da empresa.

A primeira base de dados continha 9196 linhas (correspondendo a 9196 épicos diferentes) e 43 colunas com diversas informações sobre os épicos. Já a segunda base tinha um total de 4919 linhas e 10 colunas, contendo informações variadas sobre os colaboradores.

Além disso, buscando explorar a relação entre as variáveis, foram criadas variáveis secundárias através da interação entre as variáveis originais. O objetivo dessas novas variáveis era fornecer uma gama maior de possibilidades a serem exploradas nas etapas de montagem do modelo de regressão. Através de variáveis assim, é possível a observação da relação de uma variável resposta (explicada) com a interação de duas ou mais variáveis explicativas, de maneira simultânea.

Abaixo, estão detalhadas as principais variáveis presentes nas duas bases, bem como as variáveis secundárias criadas a partir da interação de outras. Vale ressaltar que várias das variáveis fornecidas nas bases originais não possuíam uma aplicação prática para a construção dos modelos imaginados, portanto não são apresentadas neste trabalho.

Tabela 5 - Principais Variáveis Independentes Envolvidas

Variáveis	Abreviação Variável	Base de Origem	Grupo	Categoria Regressão	Descrição
Número de Épicos	*NE*	Base 1	Original	Explicativa	Contagem do número de épicos
Esforço Original	*EO*	Base 1	Original	Explicativa	Variável de esforço preenchida pelas squads
Esforço Normalizado	*EN*	-	Calculada	Explicativa	Variável de esforço normalizada (Tópico 2.2.4)
Índice de Senioridade	*IS*	-	Calculada	Explicativa	Índice da squad calculado a partir da senioridade dos membros (Tópico 2.2.5)
Porcentagem de Suporte	*%S*	Base 1	Original	Explicativa	Porcentagem do tempo das squads gasto com atividades de suporte (manutenção de sistema, correção de bugs etc.)
C1	*C1*	-	Calculada	Explicativa	Variável de interação entre *NE* e *IS*
C2	*C2*	-	Calculada	Explicativa	Variável de interação entre *EO* e *IS*
C3	*C3*	-	Calculada	Explicativa	Variável de interação entre *EN* e *IS*
C4	*C4*	-	Calculada	Explicativa	Variável de interação entre *NE, IS* e *%S*
C5	*C5*	-	Calculada	Explicativa	Variável de interação entre *EN, IS* e *%S*
C6	*C6*	-	Calculada	Explicativa	Variável de interação entre *EN* e *%S*

Fonte: Autor

Tabela 6 - Principais Variáveis Dependentes Envolvidas

Variáveis	Abreviação Variável	Base de Origem	Grupo	Categoria Regressão	Descrição
Tempo de Execução	*TE*	Base 1	Original	Explicada	Tempo gasto (em dias) para a execução de todos os épicos de um trimestre
Número de Atrasos da Meta	*NA*	Base 1	Original	Explicada	Número de épicos que foram entregues após a data limite de meta

Fonte: Autor

Abaixo podem ser observadas as fórmulas de cálculo utilizadas para a obtenção das variáveis de interação:

Tabela 7 - Fórmulas de Cálculo das Variáveis Calculadas

Variáveis	Fórmula de Cálculo
C1	$C_1 = NE \div IS$
C2	$C_2 = EO \div IS$
C3	$C_3 = EN \div IS$
C4	$C_4 = (NE \div IS) \times (1 + \%S)$
C5	$C_5 = (EN \div IS) \times (1 + \%S)$
C6	$C_6 = EN \times (1 + \%S)$

Fonte: Autor

4.2.2. Definição de Premissas para o Uso dos Dados

Com o objetivo de melhorar os resultados obtidos pelo modelo a ser construído, foram definidas algumas premissas iniciais para o tratamento e utilização dos dados. Além disso, por se tratar dos dados de uma empresa de grande porte com processos de compliance bem estruturados, a própria instituição fez exigências sobre o uso dos dados.

Os processos e premissas detalhadas abaixo foram adotadas antes de qualquer análise dos dados. Por se tratar de restrições e princípios bem definidos na fase de idealização do estudo, os dados excluídos pelas premissas adotadas não foram considerados nem para a fase inicial de visualização da distribuição e normalidade dos dados.

Desta forma, as premissas adotadas para a delimitação dos dados a serem utilizados, foram:

- **Recorte temporal:** Partindo da exigência feita pela própria empresa, não poderiam ser utilizados dados referentes a períodos do ano de 2024 ou do último trimestre de 2023. Além disso, na virada do ano de 2022 para 2023, foi realizada uma grande mudança no sistema de controle que gera os dados utilizados no estudo, acarretando uma alteração significativa na estrutura dos dados. Desta forma, visando respeitar as demandas da

empresa fornecedora dos dados e a utilização apenas de dados confiáveis, a análise foi feita apenas com os dados do período de **01/01/2023 a 30/09/2023**;

- **Dados em Branco:** Assim como em qualquer base de dados gerada por sistemas de input manual, alguns campos das bases estavam em branco, ou seja, com ausência de dados. Uma vez que o volume de dados era grande e o número de campos em branco não era tão representativo (HAIR; BLACK; BABIN, 2009), foi escolhida a exclusão completa das linhas com dados faltantes;
- **Dados Conflitantes:** Buscando o uso apenas de dados consistentes, linhas com informações conflitantes ou incoerentes foram excluídas da análise para garantir a assertividade do modelo (Exemplo: Datas de entrega anteriores às datas de início dos trabalhos);
- **Escolha das Squads:** Devido a inconsistências na base de informações dos colaboradores, algumas linhas não possuíam a informações de senioridade preenchida. Desta forma, foram escolhidas apenas as squads que possuíam 100% dos seus membros com a senioridade definida.

Desta forma, após as definições de utilização dos dados e a consolidação da base de épicos em trimestres e squads, o número de linhas da nova base passou a ser 184 e o de colunas 22. A partir desta nova base que foram realizados os estudos de distribuição dos dados e identificação de outliers.

4.2.3. Distribuição e Normalidade dos Dados

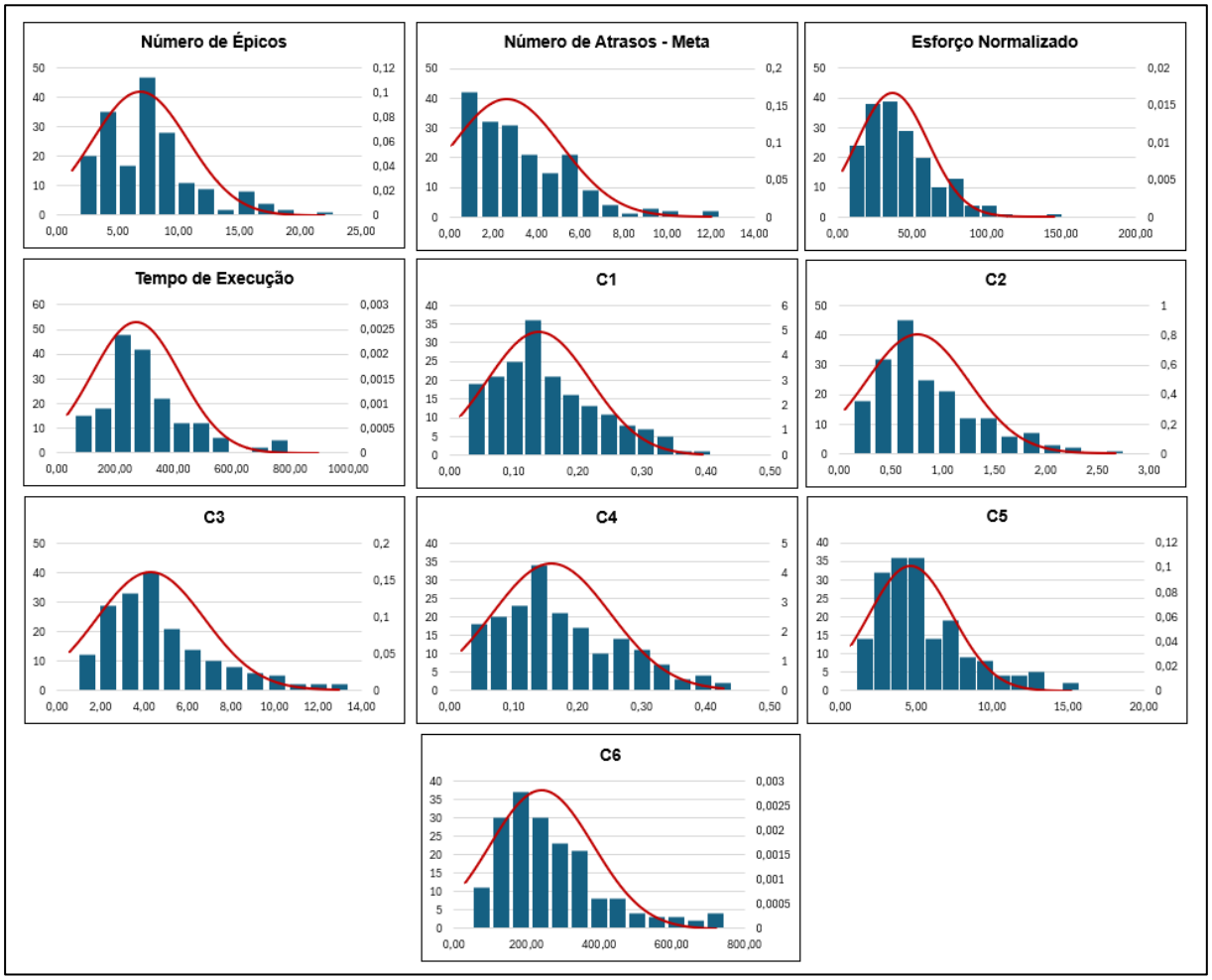
Como ponto inicial de uma análise de dados, é interessante avaliar o comportamento geral das variáveis envolvidas e da relação entre elas. Isso dá ao avaliador insumos importantes sobre qual é o melhor modelo estatístico a se utilizar na análise, bem como se há ou não algumas inconsistências nos dados (HAIR; BLACK; BABIN, 2009).

Em primeiro lugar, foi feita uma avaliação da distribuição das principais variáveis da base, candidatas a serem utilizadas para a construção o modelo. Essa avaliação foi feita com perfil univariado (cada variável sendo analisada de maneira individual). Segundo (HAIR; BLACK; BABIN, 2009), a montagem de histogramas é um bom método para avaliações visuais da distribuição de variáveis.

Os histogramas foram montados definindo o número de classes através do valor da raiz quadrada do número de observações totais presente na base. Com isso, foram definidos 13 intervalos numéricos onde as observações foram distribuídas de acordo com os seus respectivos valores. Desta forma, as barras mais altas, representam os intervalos numéricos em que mais observações se encaixavam.

O objetivo principal desta análise era verificar a possível normalidade dos dados. Isso é usualmente representado através de um histograma com formato de curva de “sino”, em que as barras centrais são maiores e vão diminuindo na medida que se distanciam da média ou do centro da distribuição, assim como a curva normal. Os histogramas das principais variáveis da base podem ser observados abaixo:

Figura 20 - Histogramas das Principais Variáveis



Fonte: Autor

Pode-se observar que os histogramas das variáveis de Esforço Normalizado, Número de Épicos, Tempo de Execução, C1, C2, C3, C4, C5 e C6 possuem um

comportamento que remete com o de uma curva normal (também plotada no histograma) com um deslocamento considerável para a direita e diferentes perfis de curtoses. Além disso, é possível verificar que a variável de Número de Atrasos da Meta possuí uma estrutura bem diferente da observada na curva normal, apresentando um comportamento decrescente de frequência na medida que os valores dos intervalos crescem.

Apesar de bons indicativos visuais, os histogramas não são testes estatísticos definitivos. Ou seja, não é possível atestar a normalidade ou não dos dados apenas através da análise deles (HAIR; BLACK; BABIN, 2009). Para complementar a verificação da distribuição dos dados, foi realizado o teste estatístico de Anderson-Darling, capaz de atestar a normalidade de dados com amostras médias ou grandes de forma robusta. Devido a maior facilidade de se realizar testes estatísticos em softwares próprios, as testagens foram feitas no software R Studio.

Tabela 8 - Resultados do Teste de Normalidade (Anderson-Darling)

Variáveis	**P-Valor**
Número de Épicos	5,83E-05
Esforço Normalizado	9,18E-09
Tempo de Execução	9,71E-08
Número de Atrasos da Meta	1,10E-11
C1	5,42E-06
C2	1,86E-08
C3	2,93E-11
C4	4,05E-03
C5	8,08E-09
C6	6,63E-10

Fonte: Autor

Observa-se claramente que nenhuma das variáveis obteve um Valor-P maior do que o nível de significância adotado de 0,05. Isso indica que não existem evidências suficientes para concluir que as variáveis seguem uma distribuição normal. Resultados deste tipo, podem indicar influências de assimetria nos dados, impacto de outliers da amostra entre outros fatores. Buscando a obtenção de variáveis normalmente distribuídas para a aplicação das regressões propostas e normalização dos resíduos delas, foram realizadas transformações de variáveis e retirada de outliers (HAIR; BLACK; BABIN, 2009).

4.2.4. Retirada de Outliers Estatísticos

Analisando os histogramas apresentados e alguns gráficos de dispersão auxiliares, observou-se possíveis outliers estatísticos relevantes. Esses candidatos a outliers foram identificados nas variáveis de Tempo de Execução, Número de Épicos e Esforço Normalizado. A partir desta verificação inicial, foi conduzido um processo de retirada de outliers através do método interquartil (IQR).

O método IQR consiste na definição estatística de um intervalo de "normalidade" dos dados. Observações fora deste intervalo são consideradas outliers e devem ser removidas da base para evitar distorções na análise. O intervalo de normalidade é dado pela fórmula abaixo, em que IQR é a diferença entre o valor limite do 1º quartil (Q1) e o 3º quartil (Q3) e constante *k* recebe o valor de 1,5 (MAHAJAN; KUMAR; PANT; TIWARI, 2020).

$$Q_1 - k(IQR) < x_i < Q_3 + k(IQR)$$

Aplicando essa fórmula às variáveis analisadas, obtém-se os seguintes intervalos para os dados:

Tabela 9 - Limites do IQR

Variáveis	IQR	L. Superior	L. Inferior
Tempo de Execução	150,8	498,1	45,8
Número de Atrasos da Meta	3,0	7,1	-1,9
Número de Épicos	5,0	14,3	-0,7

Fonte: Autor

Desta forma, todos as observações das variáveis que foram identificadas fora dos limites delimitados pelo método, foram excluídas da análise. Ao todo, foram excluídas 27 observações (~14% da base de análise). Com isso, restaram 157 linhas na base para a construção das análises.

4.2.5. Transformações de Variáveis

Após a retirada de outliers, o próximo passo seguido para a obtenção das variáveis normalmente distribuídas foi a realização de transformações das variáveis já existentes. Segundo, (HAIR; BLACK; BABIN, 2009) a transformação de variáveis

fornece (1) um meio para corrigir violações das suposições inerentes às técnicas estatísticas ou (2) melhorar a correlação entre variáveis. As transformações foram realizadas através de interações de tentativa e erro em que eram coletadas as melhorias de um método de transformação para o outro (HAIR; BLACK; BABIN, 2009).

Tomando o estudo de (HAIR; BLACK; BABIN, 2009) como referência, para casos em que há uma assimetria ou deslocamento na distribuição das variáveis, as transformações mais usuais para a obtenção de normalidade são as de raiz quadrada e logarítmica. Desta forma, essas transformações foram realizadas em todas as variáveis principais da base e foi repetido o teste de Anderson-Darling para a verificação das melhorias nos resultados de normalidade. Vale ressaltar que essa segunda interação do teste de Anderson-Darling foi feita com a base de dados já sem outliers.

Desta forma, seguem abaixo as fórmulas das principais transformações feitas. Também foram testadas transformações como o inverso das variáveis, aplicações exponenciais e outras. Apesar disso, elas não obtiveram resultados significativos como as observadas abaixo:

- **Transformação Logarítmica:**

$$X_{Logaritmo} = \log X$$

- **Transformação de Raiz Quadrada:**

$$X_{Raiz} = \sqrt{X}$$

Através das transformações aplicadas, foram criadas variáveis derivadas das variáveis originais da base. Essa maior gama de variáveis é interessante no processo de montagem de modelos de regressão em que há a necessidade de interações de tentativa e erro na busca pelo melhor modelo possível. Um número maior de variáveis, significa uma quantidade maior de opções para se montar um bom modelo. As novas variáveis podem ser observadas na tabela abaixo, bem como as suas variáveis de origem e as transformações utilizadas em suas criações.

Tabela 10 - Novas Variáveis Criadas

Novas Variáveis	Variável Originadora	Transformação Aplicada
Épicos (Log)	Número de Épicos	Logarítmica
Épicos (Raiz)	Número de Épicos	Raiz
Esforço Normalizado (Log)	Esforço Normalizado	Logarítmica
Esforço Normalizado (Raiz)	Esforço Normalizado	Raiz
C1 (Log)	C1	Logarítmica
C1 (Raiz)	C1	Raiz
C2 (Log)	C2	Logarítmica
C2 (Raiz)	C2	Raiz
C3 (Log)	C3	Logarítmica
C3 (Raiz)	C3	Raiz
C4 (Log)	C4	Logarítmica
C4 (Raiz)	C4	Raiz
C5 (Log)	C5	Logarítmica
C5 (Raiz)	C5	Raiz
C6 (Log)	C6	Logarítmica
C6 (Raiz)	C6	Raiz
T de Execução (Log)	Tempo de Execução	Logarítmica
T de Execução (Raiz)	Tempo de Execução	Raiz
N de Atrasos Meta (Raiz)	Número de Atrasos da Meta	Raiz

Fonte: Autor

Desta forma, após a retirada de outliers e transformações das variáveis, foi realizado um segundo teste de Anderson – Darling para aferir as mudanças na distribuição das variáveis. Os resultados foram coletados e podem ser observados na tabela abaixo:

Tabela 11 - Novos Resultados de Normalidade (Anderson-Darling)

Variáveis	P-Valor	Flag - Normalidade
Número de Épicos	0,00	-
Épicos (Log)	0,00	-
Épicos (Raiz)	0,00	-
Esforço Normalizado	0,00	-
Esforço Normalizado (Log)	0,03	-
Esforço Normalizado (Raiz)	**0,34**	**Normal**
C1	0,03	-
C1 (Log)	0,00	-
C1 (Raiz)	**0,48**	**Normal**
C2	0,00	-
C2 (Log)	0,00	-
C2 (Raiz)	0,02	-
C3	0,00	-
C3 (Log)	**0,08**	**Normal**
C3 (Raiz)	**0,23**	**Normal**
C4	0,02	-
C4 (Log)	0,00	-
C4 (Raiz)	**0,41**	**Normal**
C5	0,00	-
C5 (Log)	**0,31**	**Normal**
C5 (Raiz)	**0,21**	**Normal**
C6	0,01	-
C6 (Log)	**0,25**	**Normal**
C6 (Raiz)	**0,52**	**Normal**
Tempo de Execução	0,04	-
T de Execução (Log)	0,00	-
T de Execução (Raiz)	**0,34**	**Normal**
Número de Atrasos da Meta	0,00	-
N de Atrasos Meta (Raiz)	0,00	-

Fonte: Autor

Observa-se que a transformação de Raiz Quadrada foi capaz de normalizar grande parte das variáveis analisadas (p-valor >0,05). Além disso, o método por Logaritmo obteve resultados positivos para as variáveis C3, C5 e C6 (variáveis de interação que envolviam a variável original de Esforço Normalizado). Todas as novas variáveis geradas, normalmente distribuídas ou não, foram testadas ao longo dos passos seguintes no processo de aplicação da regressão em si.

4.3. Aplicação dos Métodos de Regressão Linear

4.3.1. Cálculo das Correlações

Anteriormente a construção dos modelos de regressão de fato, foi feito um estudo a respeito da correlação entre as variáveis presentes na base. Esse estudo visava entender melhor a força das relações lineares entre as variáveis para elencar as combinações candidatas a um modelo de regressão linear com qualidade. Seguindo esse método, parte-se do pressuposto que variáveis com alta correlação possivelmente podem "se explicar" melhor em modelos de regressão linear.

Apesar disso, vale ressaltar que a correlação não é um indicativo absoluto sobre a qualidade do modelo de regressão gerado pelas variáveis. Fatores como a ausência de causalidade entre as variáveis podem gerar modelos de regressão não adequados, apesar da presença de valores altos de correlação entre elas (MONTGOMERY; RUNGER, 2021). Desta forma, as correlações foram utilizadas apenas para eleger uma gama de variáveis candidatas a modelos de regressão adequados. As regressões por sua vez, foram avaliadas individualmente através de outros fatores que não a correlação, como o R^2 e os resíduos gerados.

Para estruturar o estudo de correlações, foram montadas duas matrizes, uma para as variáveis normalmente distribuídas (usando a correlação de Pearson) e outra para as que não seguem uma distribuição normal (usando a correlação de Spearman). As variáveis de entrada (variáveis explicativas) foram posicionadas nas colunas da matriz, enquanto as variáveis de saída (variáveis explicadas) compuseram as linhas dela. Desta forma, foi possível analisar as correlações entre todas as variáveis da base de maneira unificada. As matrizes completas podem ser analisadas no Anexo 1 deste trabalho.

Analisando as matrizes de maneira geral, é possível observar que as variáveis normalmente distribuídas obtiveram valores de correlação entre si inferiores às não normais, variando entre 0,09 e 0,19. Valores de correlação dessa magnitude são tidos como baixos (DANTAS, 1998). Já as variáveis não normais, obtiveram resultados melhores, apresentando índices de correlação de até 0,56, tido como um valor de correlação moderada. Abaixo, é possível analisar as cinco maiores correlações das duas matrizes construídas.

Tabela 12 - Cinco Melhores Correlações das Variáveis Normais

Variável Explicativa	Variável Explicada	Correlação (Pearson)
Esforço Normalizado (Raiz)	T de Execução (Raiz)	0,185
C1 (Raiz)	T de Execução (Raiz)	0,142
C3 (Log)	T de Execução (Raiz)	0,141
C6 (Log)	T de Execução (Raiz)	0,180
C6 (Raiz)	T de Execução (Raiz)	0,163

Fonte: Autor

Tabela 13 - Cinco Melhores Correlações das Variáveis Não Normais

Variável Explicativa	Variável Explicada	Correlação (Spearman)
Número de Épicos	Número de Atrasos da Meta	0,557
Esforço Normalizado	Número de Atrasos da Meta	0,446
C1	Número de Atrasos da Meta	0,519
C4	Número de Atrasos da Meta	0,521
C6	Número de Atrasos da Meta	0,445

Fonte: Autor

De maneira geral, correlações moderadas positivas indicam que na medida que a variável explicativa cresce, a variável explicada tende a crescer também em alguma medida (MONTGOMERY; RUNGER, 2021). Apesar disso, resultados assim tem de ser interpretados com cautela pois podem não indicar uma relação causal entre as variáveis, necessitando da avaliação dos parâmetros da regressão em si para que se conclua algo concreto sobre a relação das variáveis e do modelo de regressão gerado por elas.

4.3.2. Aplicação de Regressões Lineares Simples

Após a análise das correlações exposta no tópico anterior, foram feitas um total de 14 regressões lineares simples com diferentes variáveis. Os critérios de priorização para a realização dessas regressões foram (1) Valor de correlação encontrado entre as variáveis e (2) Relação de causalidade teórica entre as variáveis (critério qualitativo). Desta forma, as diferentes regressões foram comparadas entre si através de critérios distintos para a obtenção do modelo mais adequado. A tabela de resumo das regressões realizadas e de seus parâmetros de comparação pode ser observada na Figura 21.

Figura 21 - Resumo Regressões Lineares Simples

Cód	Variável Entrada	Variável Saída	B0	B1	Correlação	R^2	R^2 Ajustado	Erro Padrão	P-Valor	Média Resíduos	Anderson Darling (Resíduos)
Reg1.9	Número de Épicos	Número de Atrasos da Meta	-0,11	0,40	0,56	0,33	0,33	1,59	0,00	0,00	0,02
Reg1.10	Épicos (Raiz)	Número de Atrasos da Meta	-1,90	1,78	0,56	0,32	0,31	1,61	0,00	0,00	0,01
Reg1.13	Número de Épicos	N de Atrasos da Meta (Raiz)	0,29	0,17	0,56	0,31	0,30	0,69	0,00	0,00	0,06
Reg1.3	C1	Número de Atrasos da Meta	0,20	17,03	0,52	0,30	0,29	1,64	0,00	0,00	0,01
Reg1.2	C4	Número de Atrasos da Meta	0,22	14,84	0,52	0,30	0,29	1,64	0,00	0,00	0,00
Reg1.4	C1 (Raiz)	Número de Atrasos da Meta	-1,41	10,94	0,52	0,28	0,28	1,65	0,00	0,00	0,01
Reg1.1	C4 (Raiz)	Número de Atrasos da Meta	-1,36	10,13	0,52	0,28	0,28	1,65	0,00	0,00	0,01
Reg1.11	Épicos Log	Número de Atrasos da Meta	-0,48	1,69	0,56	0,28	0,27	1,66	0,00	0,00	0,00
Reg1.14	C4	N de Atrasos da Meta (Raiz)	0,42	6,06	0,52	0,27	0,27	0,71	0,00	0,00	0,00
Reg1.12	C1	N de Atrasos da Meta (Raiz)	0,42	6,93	0,52	0,27	0,26	0,71	0,00	0,00	0,00
Reg1.8	C2 (Raiz)	T de Execução (Raiz)	12,15	3,68	0,28	0,09	0,08	3,25	0,00	0,00	0,82
Reg1.6	C2 (Raiz)	T de Execução (Log)	4,97	0,51	0,28	0,08	0,07	0,00	0,00	0,00	0,00
Reg1.7	C2 (Log)	T de Execução (Raiz)	15,78	1,20	0,28	0,07	0,07	3,27	0,00	0,00	0,51
Reg1.5	C2 (Log)	T de Execução (Log)	5,47	0,17	0,28	0,07	0,06	0,47	0,00	0,00	0,00

Fonte: Autor

Observando a tabela acima, é possível constatar que a variável explicada de Número de Atrasos da Meta, e suas transformações, foi a variável que obteve os resultados mais expressivos em relação ao parâmetro R^2. As dez regressões com os melhores R^2, ou seja, as que em teoria conseguem explicar a maior parte da variabilidade da variável resposta, advém de modelos que envolvem o número de atrasos da meta. Além disso, observa-se que as variáveis explicativas atreladas de alguma forma ao número de épicos, foram as que compuseram as regressões com maior R^2, o que indica uma boa relação linear aparente entre elas e o número da atrasos da meta.

Apesar disso, quando se olha para os parâmetros atrelados aos resíduos das regressões, os resultados obtidos não foram tão positivos assim. Ainda que as médias dos resíduos estejam todas próximas a zero, como é requisitado pela literatura do tema, os testes de normalidade da distribuição dos resíduos foram negativos para a maior parte das regressões testadas. Apenas três modelos obtiveram um P-Valor acima de 0,05 para a distribuição de seus resíduos, sendo eles o (1) Reg1.13, (2) Reg1.8 e (3) Reg1.7. Desta forma, respeitando a teoria acerca da adequação de modelos de regressões lineares, apenas estes três modelos mencionados poderiam ser classificados como estatisticamente apropriados. Devido ao seu valor maior de R^2, o modelo Reg1.13 é tido como o mais adequado entre os analisados. Abaixo, estão detalhados mais parâmetros acerca deste modelo.

Figura 22 - Detalhamento dos Parâmetros (Modelo Reg1.13)

Estatística de regressão	
R múltiplo	0,56
R-Quadrado	0,31
R-quadrado ajustado	0,30
Erro padrão	0,69
Observações	152

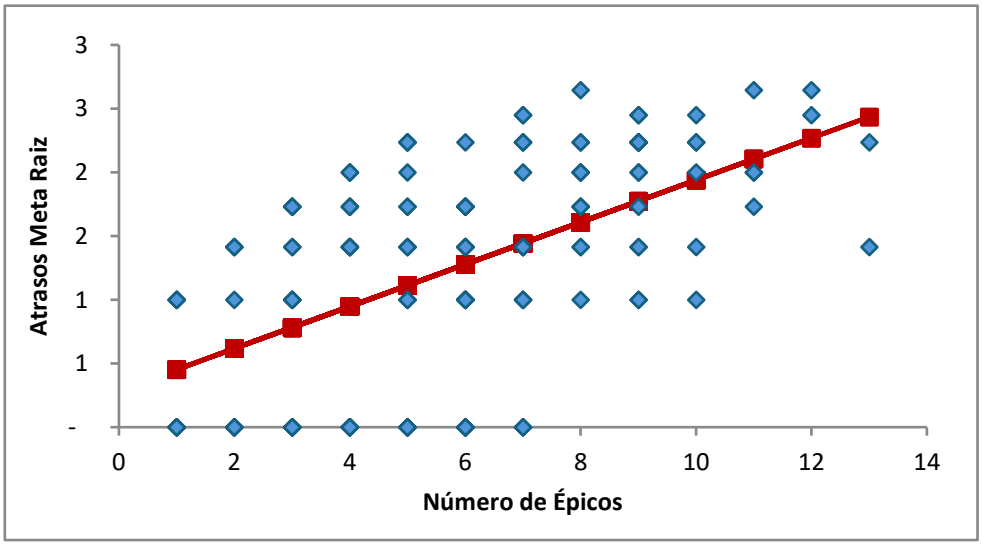
ANOVA	Grau de Liberade	SQ	MQ	F	P-Valor
Regressão	1	31,74	31,74	65,89	0,00
Resíduo	150	72,25	0,48		
Total	151	103,99			

	Coeficientes	Erro padrão	Stat t	P-Valor
Interseção	0,29	0,13	2,18	0,03
Número de Épicos	0,17	0,02	8,12	0,00

Fonte: Autor

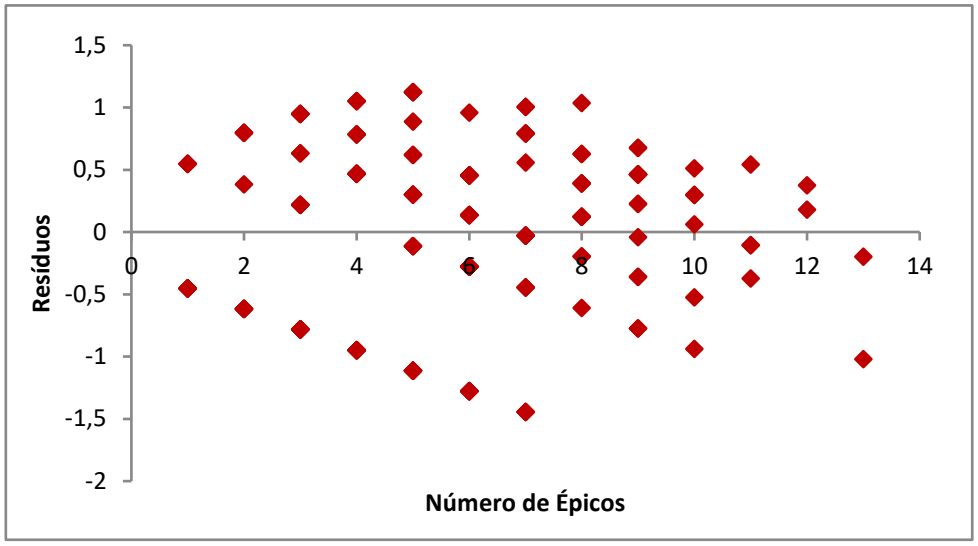
Analisando as saídas advindas do modelo Reg1.13, observamos um R^2 e um R^2 ajustado com valores de 0,31 e 0,30, respectivamente. Isto indicia que o modelo é capaz de explicar cerca de 30% da variabilidade observada na variável resposta. Além disso, o teste ANOVA foi bem-sucedido ao constatar um p-valor inferior a 0,05 pelo teste F, o que rejeita a hipótese H_0 em que o coeficiente angular da reta do modelo é igual a 0 e comprova a significância do modelo.

Figura 23 - Gráfico de Plotagem da Linha de Ajuste (Modelo Reg1.13)



Fonte: Autor

Figura 24 - Gráfico de Dispersão dos Resíduos (Modelo Reg1.13)



Fonte: Autor

Apesar do gráfico de dispersão dos resíduos apresentar um comportamento visual diferente do que é tido como uma dispersão aleatória, o teste de Anderson Darling foi aplicado aos resíduos e constatou uma distribuição normal através de p-valor maior que 0,05. Por fim, o modelo Reg1.13 pode ser aplicado através da seguinte fórmula:

$$\widehat{NA} = 0{,}29 + 0{,}17 * NE$$

Constata-se que apesar do modelo Reg1.13 atender a todos os requisitos de adequação de um modelo de regressão linear, a sua aplicação em cenários reais ainda não é aconselhada devido ao seu baixo R^2. Por só conseguir explicar 30% da variabilidade da variável resposta, a sua aplicação pode levar a erros de previsão e embasar de maneira rasa o processo de tomada de decisão na empresa.

4.3.3. Aplicação de Regressões Lineares Múltiplas

Seguindo o mesmo processo observado no tópico anterior de regressões lineares simples, foram construídos 14 modelos de regressão linear múltipla com as variáveis disponíveis. A priorização dos modelos construídos também seguiu os critérios de maior correlação e causalidade teórica entre as variáveis. Os resultados obtidos por cada modelo estão presentes na Figura 25.

Figura 25 - Resumo das Regressões Lineares Múltiplas

Cód	Variável Entrada I	Variável Entrada II	Variável Saída	B0	B1	B2	Correlação	R^2	R^2 Ajustado	Erro Padrão	P-Valor	Média Resíduos	Anderson Darling (Resíduos)
Reg2.5	C6 (Raiz)	Número de Épicos	Número de Atrasos da Meta	-1,20	0,11	0,33	0,61	0,37	0,36	1,56	0,00	0,00	0,02
Reg2.6	C6	Número de Épicos	Número de Atrasos da Meta	-0,46	0,00	0,33	0,60	0,37	0,36	1,56	0,00	0,00	0,01
Reg2.2	C5 (Raiz)	Número de Épicos	Número de Atrasos da Meta	-1,09	0,68	0,34	0,60	0,36	0,35	1,56	0,00	0,00	0,01
Reg2.1	C5	Número de Épicos	Número de Atrasos da Meta	-0,41	0,15	0,34	0,60	0,36	0,35	1,56	0,00	0,00	0,01
Reg2.7	C5	Épicos (Raiz)	Número de Atrasos da Meta	-1,88	0,16	1,47	0,59	0,34	0,33	1,58	0,00	0,00	0,00
Reg2.8	C5 (Raiz)	Épicos (Raiz)	Número de Atrasos da Meta	-2,54	0,69	1,45	0,59	0,34	0,33	1,58	0,00	0,00	0,00
Reg2.11	C6 (Raiz)	Número de Épicos	N de Atrasos da Meta (Raiz)	-0,21	0,05	0,13	0,59	0,34	0,33	0,68	0,00	0,00	0,00
Reg2.12	C6	Número de Épicos	N de Atrasos da Meta (Raiz)	0,14	0,00	0,13	0,58	0,34	0,33	0,68	0,00	0,00	0,00
Reg2.13	C6 (Raiz)	Épicos (Raiz)	N de Atrasos da Meta (Raiz)	-0,80	0,05	0,58	0,58	0,33	0,32	0,68	0,00	0,00	0,00
Reg2.14	C5 (Raiz)	Épicos (Raiz)	N de Atrasos da Meta (Raiz)	-0,75	0,31	0,59	0,58	0,33	0,32	0,68	0,00	0,00	0,00
Reg2.9	C5	Épicos (Log)	Número de Atrasos da Meta	-0,71	0,18	1,34	0,56	0,31	0,30	1,62	0,00	0,00	0,00
Reg2.10	C5 (Raiz)	Épicos (Log)	Número de Atrasos da Meta	-1,46	0,78	1,32	0,56	0,31	0,30	1,62	0,00	0,00	0,00
Reg2.3	C2	Número de Épicos	T de Execução (Raiz)	12,84	1,82	0,16	0,32	0,10	0,09	3,23	0,00	0,00	0,89
Reg2.4	C2 (Raiz)	Número de Épicos	T de Execução (Raiz)	11,81	3,00	0,15	0,31	0,10	0,09	3,24	0,00	0,00	0,85

Fonte: Autor

Assim como foi observado nas regressões simples, a variável de número de atrasos em relação a meta foi a que obteve os maiores valores de R^2 em seus modelos, estando presente em 12 dos 14 modelos testados. Em relação às outras variáveis de entrada, excluindo o número de épicos, constata-se uma predominância das variáveis C5 e C6 (e suas transformações) nos modelos com alto valor de R^2. Ambas as variáveis possuem interações que envolvem a variável original de esforço normalizado, o que pode explicar os valores altos de R^2, já que foi observada uma correlação moderada entre os números de atraso da meta e o esforço normalizado.

De forma parecida com o que foi observado nas regressões lineares, os resultados de normalidade dos resíduos foram negativos para a maioria dos modelos construídos. Apenas os modelos Reg2.3 e Reg2.4, que possuem como variável explicada o tempo de execução dos épicos no trimestre, obtiveram um p-valor maior que 0,05. Apesar disso, os R^2 ajustados de ambos são bem inferiores aos demais observados na tabela, o que demonstra uma baixa capacidade do modelo de explicar a variabilidade da variável resposta. O segundo modelo que obteve o p-valor do teste de Anderson-Darling mais próximo ao nível de significância de 0,05 foi o Reg2.5. O detalhamento deste modelo pode ser observado abaixo.

Figura 26 - Detalhamento dos Parâmetros (Modelo Reg2.5)

Estatística de regressão	
R múltiplo	0,61
R-Quadrado	0,37
R-quadrado ajustado	0,36
Erro padrão	1,56
Observações	152

ANOVA	Grau de Liberdade	SQ	MQ	F	P-Valor
Regressão	2	208,88	104,44	43,09	0,00
Resíduo	149	361,11	2,42		
Total	151	569,99			

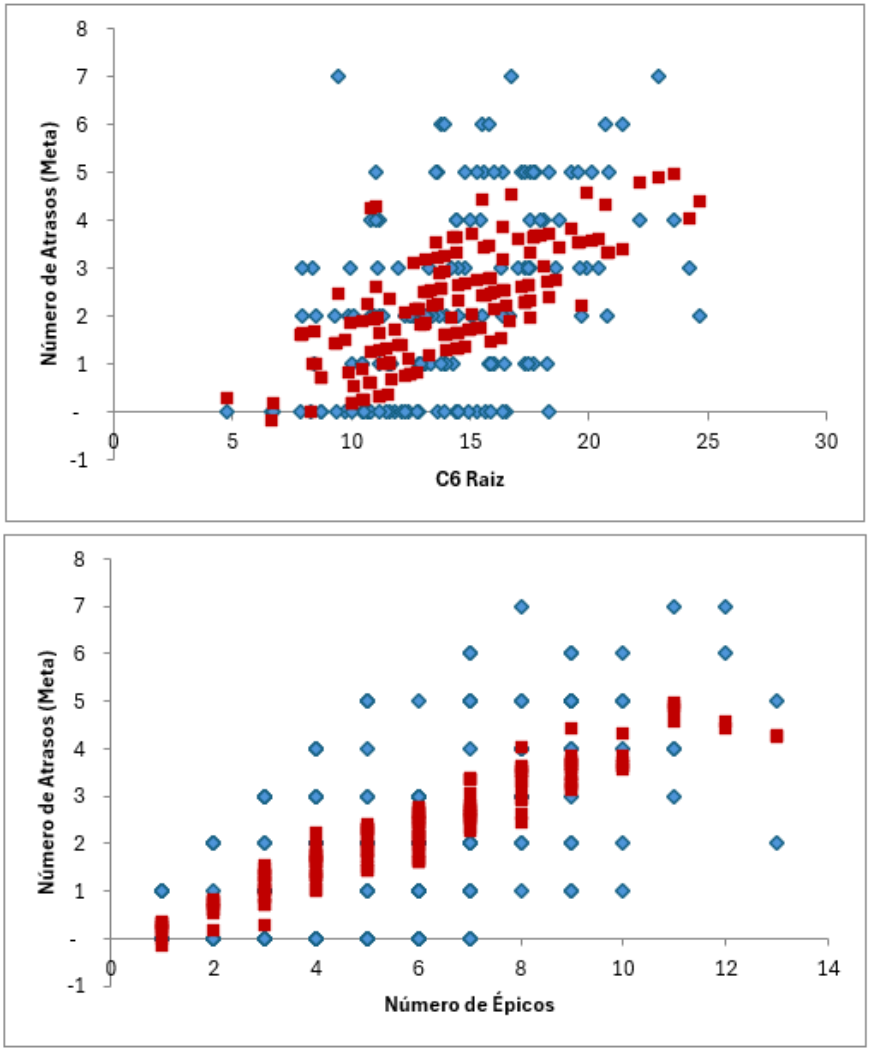
	Coeficientes	Erro padrão	Stat t	P-Valor
Interseção	-1,20	0,49	-2,44	0,02
C6 Raiz	0,11	0,04	2,75	0,01
Número de Épicos	0,33	0,05	6,29	0,00

Fonte: Autor

Através da análise dos parâmetros obtidos, é possível observar que o modelo obteve um R^2 ajustado de 0,36, o que demonstra uma capacidade de explicar 36% da variabilidade do número de atrasos da meta. Isso mostra um aumento da capacidade de explicação se comparada com o modelo linear simples analisado anteriormente, o que demonstra que a inserção de uma nova variável de explicação contribuiu para uma “melhora” no modelo. Além disso, observa-se que a regressão Reg2.5 obteve um P-Valor menor que 0,05 no teste F da ANOVA, atestando a sua significância.

Por fim, quando analisamos a tabela de parâmetros individuais das variáveis envolvidas, observa-se que as duas variáveis explicativas obtiveram um p-valor menor que 0,05. Isso demonstra que ambas são significativas para a explicação do número de atrasos em relação a meta e que são adequadas para estarem no modelo.

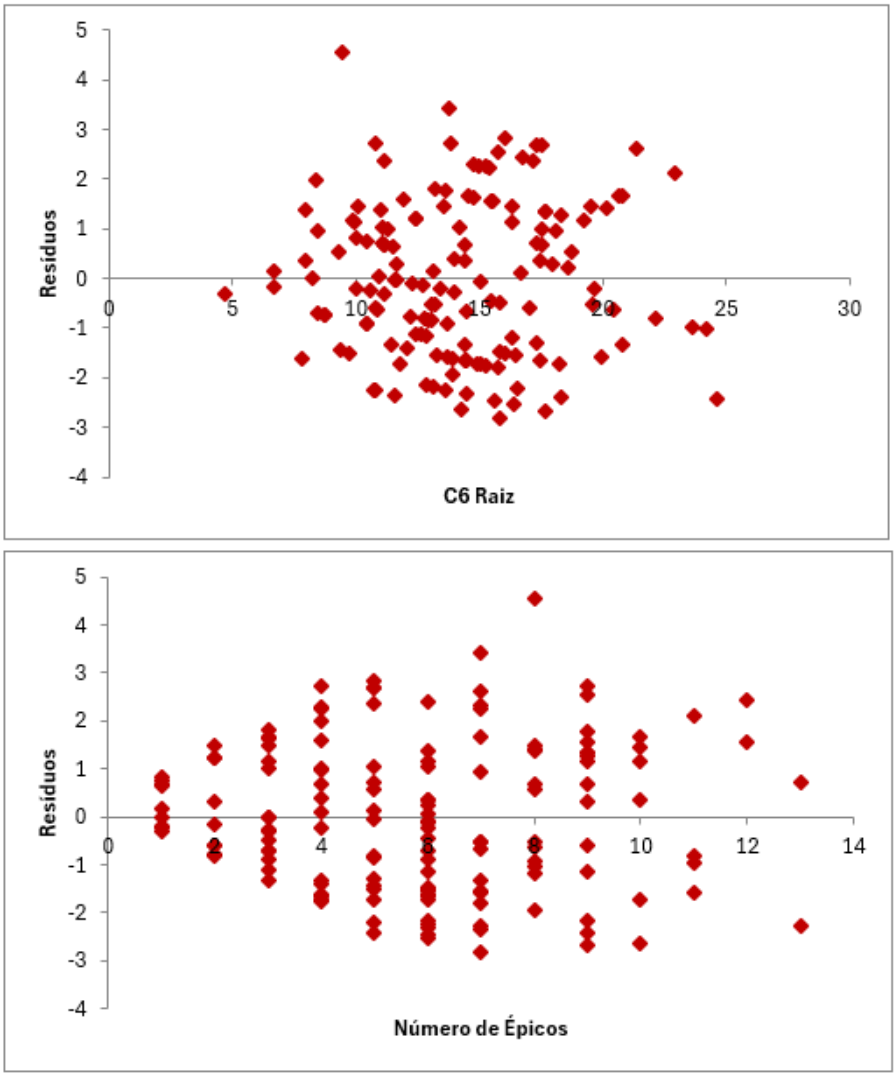
Figura 27- Gráficos de Plotagem do Modelo de Ajuste (Modelo Reg2.5)



Fonte: Autor

Por se tratar de uma regressão múltipla com duas variáveis explicativas, acima estão os dois gráficos da linha de ajuste do modelo em relação a cada uma das variáveis explicativas, o de cima para a variável C6 raiz e o de baixo para o Número de Épicos. Os resíduos em relação a cada uma das variáveis podem ser vistos nos gráficos abaixo.

Figura 28 - Gráfico de Dispersão dos Resíduos (Modelo Reg2.5)



Fonte: Autor

Observa-se que os resíduos em relação à variável C6 Raiz (normalmente distribuída) seguem uma dispersão aparentemente normal e aleatória. Já a variável de número de épicos apresenta um comportamento anômalo assim como os vistos no tópico 3.4.1.1, um indicativo de não normalidade dos resíduos. Esse comportamento não normal dos resíduos também é evidenciado pelo teste de Anderson-Darling, que forneceu valor de 0,02, inconclusivo para atestar a normalidade da distribuição. Por fim, o modelo Reg2.5 é obtido pela seguinte expressão:

$$\widehat{NA} = -1{,}2 + 0{,}11 \times C_{6\,raiz} + 0{,}33 \times NE$$

Constata-se que apesar de um valor maior do R^2 ajustado, o modelo não pode ser tido como adequado devido ao comportamento não normal de seus resíduos. Sendo assim, não é recomendado o seu uso em situações reais na empresa, visto que por não ser estatisticamente relevante, o modelo pode induzir previsões erradas sobre o comportamento do número de atrasos em relação a meta.

4.4. Conclusões dos Resultados das Regressões Lineares

Através dos estudos desenvolvidos no último tópico do trabalho, foi possível entender de forma mais clara a relação entre as variáveis presentes na base. Com o estudo de correlação, observou-se que a variável explicada de Número de Atrasos da Meta foi a que obteve os maiores índices de correlação linear com as demais variáveis da base. Esse fenômeno se intensifica quando a analisamos frente as variáveis de número de épicos e suas derivadas (C1, C4 e suas respectivas transformações), tornando-as boas candidatas para a elaboração de modelos de regressão linear.

Além disso, quando se olha para correlações das regressões lineares múltiplas, é possível identificar uma influência positiva das variáveis de esforço normalizado, como a C5 e C6, na explicação da variável de Número de Atrasos da Meta. Isso também demonstra a possibilidade da criação de modelos de regressão linear se utilizando deste tipo de variável.

Por outro lado, os resultados dos modelos construídos não alcançaram um nível de explicação da variável resposta satisfatório para uma aplicação prática na realidade. O modelo de regressão linear múltipla, não obteve sucesso nos testes de normalidade de distribuição dos resíduos, o que o torna não adequado estatisticamente. O modelo de regressão linear simples, por sua vez, cumpriu com todos os requisitos de adequação de um modelo de regressão, apresentando normalidade em seus resíduos. Desta forma, ele poderia ser aplicado em situações reais para orientar a tomada de decisão na empresa originadora dos dados, apesar do seu baixo índice de explicação da variável resposta.

Desta forma, conclui-se que a variabilidade do Número de Atrasos da Meta pode ser explicada em parte pela variável de Número de Épicos e suas derivadas em modelos de regressão linear. Acredita-se também que as variáveis de esforço normalizado (C5 e C6) exercem uma influência positiva na explicação desta variável resposta e são boas candidatas para a construção de modelos lineares. Para que

esses modelos sejam mais precisos e adequados estatisticamente do que os construídos nesse trabalho, recomenda-se mais testes de transformação das variáveis para a observação do comportamento dos resíduos, além de testes com mais variáveis envolvidas no modelo.

Por fim, é válida também a exploração e teste de modelos não lineares entre as variáveis da base. Como foi observado, grande parte das variáveis não apresenta uma distribuição normal, bem como não apresentam índices fortes de correlação linear entre si. Estes fatos as tornam boas candidatas para a aplicação de modelos não lineares de previsão.

5. Considerações Finais

A elaboração deste trabalho buscou explorar e compreender a relação entre as diferentes variáveis envolvidas nos projetos de desenvolvimento de software de uma empresa do mercado financeiro brasileiro. Além disso, foram construídos modelos de regressão linear com o objetivo de prever certos resultados desses projetos através do comportamento de variáveis explicativas chave.

A primeira etapa do trabalho consistiu no embasamento teórico dos principais temas relacionados ao desenvolvimento do estudo. Desta forma, foram levantados junto a literatura, os conceitos e características da engenharia de software, as principais metodologias de desenvolvimento de software, as metodologias ágeis no gerenciamento de projetos de software e as características de modelos de regressão linear de dados. Além disso, também foi desenvolvida uma pesquisa bibliométrica que levantou resultados quantitativos e qualitativos a respeito das tendências na literatura, a respeito do tema de previsão de resultados em projetos de desenvolvimento de software. Desta forma, obteve-se o devido embasamento e compreensão dos temas relacionados para se desenvolver a pesquisa da melhor forma possível.

A segunda etapa do estudo consistiu no processo completo de construção dos modelos de regressão linear para prever os resultados nos projetos de desenvolvimento de software. Desta forma, como todo trabalho realizado com dados, foi realizada uma etapa inicial de exploração de preparação dos dados para a aplicação dos modelos. A primeira fase consistiu na definição de premissas de uso dos dados, delimitando os recortes temporais e as exigências básicas para a construção dos modelos desejados. Isso incluiu a exclusão de observações com inconsistências, normalização dos de campos com preenchimentos divergentes, entre outras adequações de bases.

Em seguida, foram verificadas as distribuições dos dados analisados, a fim de entender como se comportavam as variáveis presentes na base. Foram constatadas diversas variáveis que não seguiam uma distribuição normal, bem como a presença de diversos outliers estatísticos que estavam impactando negativamente os testes de normalidade realizados. Para a identificação e retirada dos outliers presentes na base, foi utilizado o método IQR de identificação de observações “desviantes”. Também com o objetivo de buscar distribuições normais nas variáveis, foram realizadas transformações como a aplicação de logaritmo ou raiz quadrada nas variáveis já

existentes. Essas medidas adotadas foram bem-sucedidas e novas variáveis com distribuição normal foram obtidas.

Após a normalização das variáveis, foi conduzido um estudo de correlações lineares entre elas para eleger as principais combinações candidatas a bons modelos de regressão. Este estudo observou que a variável de Número de Atrasos da Meta se destacou como a variável explicada com os maiores níveis de correlação linear com a demais presentes na base. Isso a tornou a variável explicada mais promissora para a construção dos modelos. De maneira, geral observou-se um comportamento de correlação tido como moderado para grande parte das variáveis, variando entre 0,28 e 0,56.

Na última etapa, priorizadas pelos índices de correlação linear do estudo e pela causalidade teórica entre as variáveis, foram realizadas 28 regressões lineares com diferentes variáveis. Metade delas foram regressões lineares simples e a outra metade regressões múltiplas, envolvendo duas variáveis explicativas. As regressões foram analisadas através de diferentes parâmetros e comparadas entre si para a obtenção do modelo mais adequado possível. Desta forma, uma das regressões lineares simples, obteve um coeficiente de determinação 0,30 e obteve sucesso em todos os testes de adequação realizados, podendo ser considerado um modelo “válido” para a aplicação prática. Apesar disso, os níveis observados nos coeficientes de determinação ainda são tidos como baixos para a exigência de aplicações reais na empresa. Sendo assim, não se recomenda o uso dos modelos construídos no estudo para a previsão de resultados em situações reais.

Buscando a melhoria na qualidade de previsão de resultados em projetos de desenvolvimento de software, propõe-se para trabalhos futuros:

- Realização de mais transformações diferentes nas variáveis da base para obtenção de resíduos normalmente distribuídos (exponenciais, inversões, combinações etc.);
- Adição e teste de mais variáveis nos modelos de regressão múltipla;
- Aplicação de modelos estatísticos não lineares entre as variáveis da base para a previsão de resultados;
- Aplicação de outros modelos lineares para a previsão de resultados através de múltiplas variáveis.

6. Referências

ALMEIDA, M. **Elaboração de Projeto, TCC, Dissertação e Tese: Uma Abordagem Simples, Prática e Objetiva**. 2. ed. São Paulo: Editora Atlas S.A, 2014. v. 1

ASSOCIAÇÃO BRASILEIRA DAS EMPRESAS DE SOFTWARE. **Mercado Brasileiro de Software: Panorama e Tendências 2023**. [s.l: s.n.].

AUDY, J. **Scrum 360: um guia completo e prático de agilidade**. 1. ed. [s.l.] Casa do Código, 2015. v. 1

BARBALHO, S. et al. Exploring the relation among product complexity, team seniority and project performance as a path for planning new product development projects: a predictive model applying the system dynamics theory. **IEEE Transactions on Engineering Management**, v. 69, n. 5, p. 1823–1836, out. 2022.

BECK, K. Embracing change with extreme programming. **Computer**, v. 32, n. 10, p. 70–77, out. 1999.

BECK, K. **Manifesto for Agile software Development**.

BECK, KENT. **Extreme programming explained : embrace change**. [s.l.] Pearson Education, 2004. v. 2

BOEHM, B. A view of 20th and 21st century software engineering. **Proceedings - International Conference on Software Engineering**, v. 2006, p. 12–29, 2006.

BOEHM, B. W. A Spiral Model of Software Development and Enhancement. **Computer**, v. 21, n. 5, p. 61–72, 1988.

BUSSAB, W.; MORETTIN, P. **Estatística Básica**. [s.l: s.n.].

DE CARVALHO, B. V.; MELLO, C. H. P. Aplicação do método ágil scrum no desenvolvimento de produtos de software em uma pequena empresa de base tecnológica. **Gestão & Produção**, v. 19, n. 3, p. 557–573, 2012.

DE CARVALHO, M. M.; RABECHINI JUNIOR, R. **Impact of risk management on project performance: The importance of soft skills**. **International Journal of Production Research**Taylor and Francis Ltd., , 17 jan. 2015.

DYER, J.; GREGERSEN, H.; CHRISTENSEN, C. **The Innovator's DNA: Mastering the Five Skills of Disruptive Innovators**. [s.l.] Havard Business Review Press, 2011.

FLORES, J. Senior Project Design Success and Quality: A Systems Engineering Approach. **Procedia: Computer Science**, v. 8, p. 452–460, 2012.

GIL, A. **Como elaborar projetos de pesquisa**. 4. ed. São Paulo: Editora Atlas S.A, 2002. v. 1

GIL, A. **Métodos e Técnicas de Pesquisa Social**. 6. ed. São Paulo: Editora Atlas S.A, 2008. v. 1

GROOS, O.; PRITCHARD, A. DOCUMENTATION NOTES. **Journal of Documentation**, v. 25, n. 4, p. 344–349, 1969.

HAIR, J.; BLACK, W.; BABIN, B. **Análise Multivariada de Dados**. 6. ed. Porto Alegre: Bookman, 2009.

IEEE. **International Standart: Systems and software engineering-Vocabulary**. [s.l: s.n.]. Disponível em: <www.iso.orgwww.ieee.org>.

JACOBSON, I. **THE UNIFIED SOFTWARE DEVELOPMENT PROCESS**. Boston: Addison-Wesley Longman Publishing Co., Inc., 1999.

JACOBSON, I. A Resounding 'Yes' to Agile Processes – But Also More. **Cutter IT Journal**, v. 15, n. 4, p. 2–24, 2002.

MARTIN, J. Application development without programmers. p. 350, 1982.

MILLS, H. D. The management of software engineering, Part I: Principles of software engineering. **IBM Systems Journal**, v. 9, n. 4, p. 414–420, 5 abr. 1980.

MONTGOMERY, D.; RUNGER, G. **Estatística Aplicada e Probabilidade Para Engenheiros**. 7. ed. [s.l.] LTC, 2021. v. 1

PRESSMAN, R. **Engenharia de Software**. [s.l: s.n.].

PRIKLADNICKI, R.; WILLI, R.; MILANI, F. **Métodos Ágeis Para Desenvolvimento de Software**. [s.l.] Bookman, 2014.

RISING, L.; JANOFF, N. S. Scrum software development process for small teams. **IEEE Software**, v. 17, n. 4, p. 26–32, jul. 2000.

ROYCE, W. W. **MANAGING THE DEVELOPMENT OF LARGE SOFTWARE SYSTEMS**. [s.l: s.n.].

SCHWABER, K. **Agile Project Management with Scrum**. 1. ed. Seattle: Microsoft Press, 2004. v. 1

SCHWABER, K.; BEEDLE, M. **Agile Software Development with Scrum**. [s.l.] Prentice Hall PTR, 2001.

SCHWABER, K.; SUTHERLAND, J. **Guia do Scrum: Regras definitivas do jogo**.

SEIDELMANN, J. **Digital Transformation/Industrie 4.0**.

SILVA, E.; MENEZES, E. **Metodologia da Pesquisa e Elaboração de Dissertação**. 3. ed. Florianópolis: UFSC, 2001. v. 1

SOMMERVILLE, IAN. **Engenharia de Software**. [s.l.] Pearson Prentice Hall, 2011.

STAPLETON, J. DSDM: Dynamic Systems Development Method. **Proceedings of the Conference on Technology of Object-Oriented Languages and Systems, TOOLS**, p. 406, 1999.

SU, H. N.; LEE, P. C. Mapping knowledge structure by keyword co-occurrence: A first look at journal papers in Technology Foresight. **Scientometrics**, v. 85, n. 1, p. 65–79, 2010.

TALIAFERRO, A. Industry 4.0 and Distribution Centers: Transforming distribution operations through innovation. **Deloitte University Press**, p. 16–16, 2016.