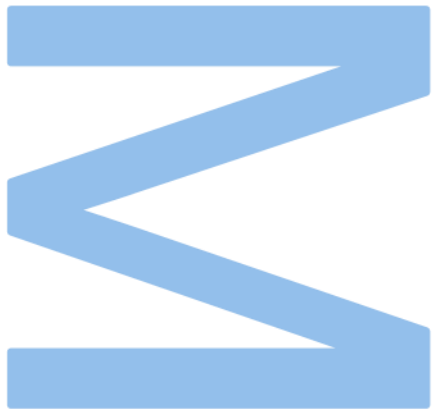
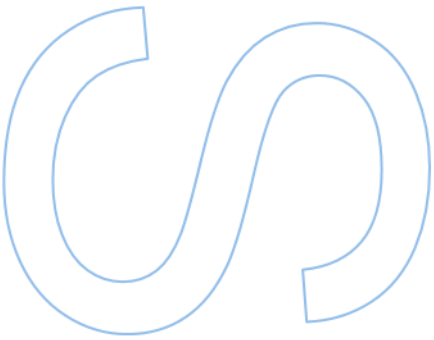
Incident Ticket Database Root Cause Analysis for Top Recurrent Issues





Vasco Fernando Coelho Soares
Mestrado em Engenharia e Sistemas Informáticos
Departamento de Ciências e Computadores
2023

Orientador
João Manuel Portela da Gama, Professor Catedrático, FEP

Coorientador
Aleksandra Rolof, Analista de Sistemas, Infineon


infineon


U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

Agradecimentos

Gostaria de expressar a minha sincera gratidão a todos aqueles que desempenharam um papel significativo na realização deste trabalho.

Em primeiro lugar, à minha família, em especial à minha mãe, pai e irmã, pelo apoio constante ao longo desta jornada. O vosso incentivo inabalável e crença em mim nos momentos mais desafiadores foram fundamentais.

À Inês Fonseca e ao Emerson Ribeiro da Infineon, agradeço profundamente por tornarem este projeto possível. A vossa colaboração e envolvimento foram essenciais para o sucesso desta tese.

À equipa composta por Emerson, Marlon e Anabela da Infineon, agradeço pelo valioso apoio e orientação ao longo de todo o processo. A vossa dedicação e partilha de conhecimento foram fundamentais.

À Aleksandra Rolof, minha mentora, expresso a minha gratidão por guiar-me com sabedoria e paciência durante esta jornada de investigação. Os vossos conselhos e insights foram inestimáveis.

Ao meu orientador de tese, João Gama, quero agradecer pela orientação experiente, disponibilidade constante e motivação.O profissionalismo e conhecimento foram cruciais para o desenvolvimento deste trabalho.

A todas as outras pessoas que, de alguma forma, contribuíram para este projeto, o meu sincero agradecimento. A todos, o meu profundo reconhecimento.

Este trabalho não teria sido possível sem o apoio e contribuições de cada um de vocês.

Muito obrigado.

Resumo

O presente estudo tem como objetivo investigar os desenvolvimentos baseados em *SAP ABAP* utilizadas pela *Infineon Technologies AG,* uma empresa alemã de semicondutores fundada em 1999. Com mais de 50.000 empregados e um volume de negócios elevado, a *Infineon Technologies AG* necessita de uma plataforma eficiente para gerenciar os seus processos internos. Para tal, a empresa usa o *SAP* e desenvolvimentos baseados em *SAP ABAP*, que são realizados pela sua equipa interna de *IT* e executados diariamente por seus funcionários.

Quando existe uma necessidade de mudança ou correção num desenvolvimento, é criado um Ticket na plataforma *Remedy.* Em seguida, é transferido para a equipe de IT na plataforma *Octane,* onde é adicionada informação adicional, como prioridade ou gravidade, para que a equipe de desenvolvimento possa resolver o problema. Durante a fase de desenvolvimento, a plataforma SAP Conigma é utilizada para armazenar os componentes modificados ou criados no Ticket em questão.

No entanto, existe uma questão relevante quanto à gestão dos Tickets no *Remedy* e *Octane* por uma equipe externa à empresa. Com isso, a equipe de IT não tem uma compreensão clara sobre quantas vezes um desenvolvimento foi alterado, se por erro ou por novas necessidades, e assim, não pode tomar uma decisão informada sobre a necessidade de refazer um desenvolvimento.

Este estudo pretende fornecer uma análise detalhada das ferramentas utilizadas pela *Infineon Technologies AG*, bem como investigar as implicações desses processos na gestão eficiente dos desenvolvimentos baseados em *SAP ABAP.* Com a obtenção destas informações, a equipe de IT terá a oportunidade de tomar decisões informadas sobre quais módulos e desenvolvimentos são mais críticos e precisam de ser refeitos, minimizando assim o número de tickets criados por erros no futuro.

Palavras-chave: SAP ABAP, *Dashboard*, Incidentes, Melhoria Contínua, Gestão de Desenvolvimentos, Análise de Dados, Previsão, Eficiência, Processos Internos, Sap Ui5, Fiori, Python, Oracle Db, Base De Dados, Remedy, Octane, Conigma, Visual Studio Code, Chart, Dashboard.

Abstract

The present study aims to investigate developments based on SAP ABAP used by Infineon Technologies AG, a German semiconductor company founded in 1999. With over 50,000 employees and a high turnover, Infineon Technologies AG needs an efficient platform to manage your internal processes. To this end, the company uses SAP and developments based on SAP ABAP, carried out by its internal IT team and executed daily by its employees.

When there is a need for change or correction in a development, a Ticket is created on the Remedy platform. It is then transferred to the IT team on the Octane platform, where additional information, such as priority or severity, is added so that the development team can resolve the issue. During the development phase, the SAP Conigma platform stores the components modified or created in the Ticket.
However, there is a relevant issue regarding the management of Tickets in Remedy and Octane by a team external to the company. As a result, the IT team needs a clearer understanding of how often a development has been changed, whether due to error or new needs. Therefore, it cannot make an informed decision about the need to redo a development.

This study aims to provide a detailed analysis of the tools used by Infineon Technologies AG and investigate the implications of these processes for the efficient management of SAP ABAP-based developments. By obtaining this information, the IT team will have the opportunity to make informed decisions about which modules and developments are most critical and need to be redone, thus minimizing the number of tickets created due to errors in the future.

Keywords: Sap, SAP ABAP, Dashboard, Incidentes, Melhoria Contínua, Gestão de Desenvolvimentos, Análise de Dados, Previsão, Eficiência, Processos Internos, Sap Ui5, Fiori, Python, Oracle Db, Base De Dados, Remedy, Octane, Conigma, Visual Studio Code, Chart, Dashboard.

Índice

Lista de Tabelas

Lista de Figuras

Lista de Abreviaturas

UP	**UNIVERSIDADE DO PORTO**
MSE	**MEAN SQUARED ERROR**
MAE	**MEAN ABSOLUTE ERROR**
NLP OU PLN	**NATURAL LANGUAGE PROCESSING**
SVM	**SUPPORT VECTOR MACHINES**
SAP	**SYSTEMS, APPLICATIONS, AND PRODUCTS**
ABAP	**ADVANCED BUSINESS APPLICATION PROGRAMMING**
IT	**INFORMATION TECHNOLOGY**
ML	**MACHINE LEARNING**
KPI	**KEY PERFORMANCE INDICATOR**
BI	**BUSINESS INTELLIGENCE**
CRM	**CUSTOMER RELATIONSHIP MANAGEMENT**
ERP	**ENTERPRISE RESOURCE PLANNING**
SQL	**STRUCTURED QUERY LANGUAGE**
API	**APPLICATION PROGRAMMING INTERFACE**
UI	**USER INTERFACE**
UX	**USER EXPERIENCE**
CSV	**COMMA-SEPARATED VALUES**
GUI	**GRAPHICAL USER INTERFACE**
HTML	**HYPERTEXT MARKUP LANGUAGE**
CSS	**CASCADING STYLE SHEETS**
JVM	**JAVA VIRTUAL MACHINE**
OS	**OPERATING SYSTEM**
VM	**VIRTUAL MACHINE**

1. Introdução

1.1 Contextualização e Motivação

A gestão eficiente dos processos internos de uma empresa é um pilar fundamental para garantir o seu sucesso e competitividade no mercado global. A *Infineon Technologies AG*, uma empresa alemã líder na indústria de semicondutores desde a sua fundação em 1999, possui uma força de trabalho de mais de 50.000 colaboradores e opera com um volume significativo de negócios. Para atender às complexas demandas do mercado, a *Infineon* utiliza extensivamente a plataforma SAP e desenvolvimentos baseados em SAP ABAP, que são realizados internamente pela equipa de TI da empresa e são executados diariamente pelos seus funcionários.

Contudo, uma questão crucial emerge no âmbito da gestão dos incidentes e da análise das causas-raízes, conhecida como "Root Cause Analysis". Estes incidentes são tratados por uma equipa externa à empresa, que é responsável pela administração das ferramentas Remedy e Octane. Esta separação introduz um desafio significativo, pois a equipa interna de TI não possui uma visão clara do número de vezes que um desenvolvimento baseado em SAP ABAP foi alterado, seja devido a erros ou novas necessidades. Essa falta de transparência compromete a capacidade da equipa interna de TI de tomar decisões fundamentadas sobre a necessidade de reajustar ou refazer um determinado desenvolvimento. Esta lacuna pode, por sua vez, resultar num aumento no número de incidentes devido a erros no futuro, prejudicando a eficiência e a eficácia dos processos internos da empresa.

Neste contexto, a presente tese assume a missão de investigar as ferramentas empregues pela *Infineon Technologies AG* para gerir os seus desenvolvimentos baseados em SAP ABAP. Ademais, almeja-se analisar as implicações destes processos no que concerne à gestão eficiente dos desenvolvimentos e explorar como a equipa de TI pode tomar decisões informadas quanto a módulos e desenvolvimentos cruciais que necessitam de reavaliação. Ao obter estas informações, a equipa de TI estará capacitada a otimizar a gestão dos processos internos da empresa, reduzindo o número de incidentes decorrentes de erros no futuro e reforçando a eficiência e eficácia dos processos internos da *Infineon Technologies AG*.

Durante a escrita da presente tese não será possível mostrar ou indicar alguns dados devido à imposição da empresa em questão, de forma a manter a proteção de dados internos.

1.2 Incident Tickets e Root Cause Analysis

No contexto da gestão de desenvolvimentos baseados em SAP ABAP, a Infineon Technologies AG confronta-se com a gestão de "*incident tickets*", que representam os problemas e erros identificados no funcionamento dos programas. Esses *incident tickets* são submetidos por utilizadores e constituem os desafios a serem superados pela equipa de desenvolvimento. A ferramenta *Remedy* é utilizada como plataforma central para o registo e acompanhamento destes incidentes. Porém, devido à operação desta ferramenta por uma equipa externa, a equipa interna de TI nem sempre possui uma visão completa da frequência e natureza das alterações efetuadas nos desenvolvimentos.

Para enfrentar este desafio, a prática da "*Root Cause Analysis*" é essencial. Esta abordagem envolve a identificação e tratamento da causa-raiz de um incidente em vez de tratar apenas os seus sintomas. O processo de "*Root Cause Analysis*" visa determinar por que ocorreu um incidente e quais os fatores que contribuíram para a sua manifestação. Isto possibilita a eliminação da origem do problema, resultando em soluções duradouras e evitando reincidências.

Um aspeto crítico da gestão eficiente dos desenvolvimentos é a identificação dos incidentes recorrentes de maior impacto. Através da análise dos "*top recurrent incidents*", a equipa de TI pode priorizar a alocação de recursos para solucionar os problemas mais frequentes e prejudiciais. A identificação destes incidentes recorrentes permite uma abordagem proativa, em que a equipa de desenvolvimento se concentra em corrigir as causas-raízes desses problemas, evitando assim um ciclo contínuo de correções de sintomas.

1.3 Desenvolvimento do *Dashboard*

O propósito desta tese é a conceção e desenvolvimento de um dashboard final, destinado a facultar à equipa de TI da *Infineon Technologies AG* uma abordagem eficaz

para avaliar a necessidade de modificar ou reavaliar programas baseados em SAP ABAP. Especificamente, o objetivo central deste *dashboard* consiste em disponibilizar duas abordagens distintas para calcular a importância de um programa: uma baseada nos custos associados em termos de dias de trabalho e outra com base na criticidade dos problemas detetados.

Para além disso, o *dashboard* final deve incorporar uma funcionalidade de previsão futura, permitindo à empresa antecipar se a importância de um programa está a evoluir positiva ou negativamente ao longo do tempo. Esta capacidade proporcionará à equipa de TI os conhecimentos necessários para tomar decisões ponderadas quanto à pertinência de reavaliar ou reformular um programa em determinados intervalos temporais.

Um outro objetivo relevante é a inclusão de gráficos detalhados no *dashboard* final que ofereçam uma visão granular sobre as áreas problemáticas num programa específico. Por exemplo, o *dashboard* poderá elucidar que a maioria dos problemas se encontra concentrada numa classe particular, facultando assim à equipa de desenvolvimento um foco direcionado para melhorar o desempenho dessa área específica.

1.4 Contribuições Esperadas e Benefícios

O resultado deste trabalho proporcionará à *Infineon Technologies AG* uma ferramenta poderosa para melhorar a eficiência da gestão dos seus desenvolvimentos SAP ABAP. O *dashboard* final permitirá à equipa de TI tomar decisões embasadas e estratégicas sobre a reavaliação de programas, otimizando assim os processos internos da empresa e evitando problemas recorrentes. A abordagem analítica adotada neste projeto contribuirá para uma cultura de melhoria contínua, na qual os desenvolvimentos são refinados e aprimorados de acordo com métricas sólidas e informação concreta, resultando em operações mais eficazes e na maximização dos resultados.

1.5 Organização da Tese

Capítulo 2. Organização dos Processos e Integração das Ferramentas

No segundo capítulo, exploraremos a organização dos processos internos da Infineon Technologies AG e a integração das principais ferramentas utilizadas. Abordaremos em detalhes como as ferramentas SAP, Remedy, ALM Octane, Oracle DataBase, Tableau e Red Hat são utilizadas na empresa. Além disso, discutiremos o fluxo de trabalho e a interação entre essas ferramentas, bem como os benefícios decorrentes dessa organização e integração.

Capítulo 3. Conceitos Fundamentais

O terceiro capítulo introduzirá os conceitos fundamentais necessários para compreender o nosso trabalho. Exploraremos o papel da aprendizagem de máquina *(Machine Learning)* neste projeto, assim como as avaliações de modelos, aprendizagem supervisionada e processamento de linguagem natural (NLP).

Capítulo 4. Revisão Bibliográfica

No quarto capítulo, realizaremos uma revisão bibliográfica detalhada. Abordaremos temas relacionados à gestão de incidentes em desenvolvimentos de software, análise de tendências e padrões de incidentes, visualização de dados e gráficos de heatmap, previsão de incidentes em desenvolvimentos de software e processamento de linguagem natural (NLP). Essa revisão proporcionará uma base sólida para o desenvolvimento do nosso projeto.

Capítulo 5. Desenvolvimento e Implementação

O quinto e último capítulo é dedicado ao desenvolvimento e implementação do projeto. Iniciaremos com a obtenção e tratamento de dados, seguido pela aplicação de técnicas de previsão. Analisaremos os resultados obtidos e investigaremos o heatmap gerado. Por fim, descreveremos a criação do dashboard final no Tableau, destacando suas funcionalidades e aplicações.

2. Organização dos Processos e Integração das Ferramentas

Neste capítulo, vamos explorar a organização dos processos, a integração das diferentes ferramentas utilizadas na gestão e resolução de problemas relacionados a programas SAP ABAP para alem de uma introdução a cada uma das ferramentas.

O campo da tecnologia empresarial está em constante evolução, e o sucesso de uma organização pode estar diretamente relacionado a sua capacidade de aproveitar essas mudanças e implementar soluções tecnológicas apropriadas. Nesse contexto, a seleção e implementação de softwares de gerenciamento de operações é uma questão crítica para as empresas. Existem várias soluções de software disponíveis no mercado, cada uma com seus próprios recursos e capacidades únicas.
Neste capítulo, o foco será na explicação de várias ferramentas de software utilizadas no mercado atual e na empresa em questão. As ferramentas que serão discutidas incluem o *SAP, o Remedy, o Octane, o SAP UI5, o Fiori e o Oracle DB.* Cada ferramenta será analisada em termos de sua funcionalidade e propósito, fornecendo uma compreensão abrangente de suas capacidades e limitações. Essa análise permitirá ao leitor obter uma compreensão abrangente dessas ferramentas, incluindo as várias indústrias em que são utilizadas e os benefícios que proporcionam. Ao final deste capítulo, o leitor terá uma compreensão abrangente das ferramentas de software discutidas e será capaz de avaliar sua adequação para diferentes necessidades empresariais.

2.1 Ferramentas com os dados a serem tratados

2.1.1 SAP

SAP (Sistemas, Aplicações e Produtos) é uma empresa alemã de software multinacional que fornece software de negócios para gerenciar operações de negócios e relações com clientes. É uma das maiores empresas de software de negócios do mundo e é considerada líder em software de planeamento de recursos empresariais (ERP).
A *SAP* foi fundada em 1972 em Walldorf, Alemanha e desde então cresceu para ter operações em mais de 180 países. Seus produtos de software são utilizados por mais

de 400.000 clientes, incluindo pequenas e médias empresas, bem como grandes corporações multinacionais [1].

A carteira de produtos da *SAP* inclui uma ampla gama de soluções de software para várias funções de negócios, incluindo finanças, recursos humanos, gestão de cadeia de suprimentos e gestão da relação com o cliente. O sistema ERP da empresa é o produto principal da SAP, que fornece às empresas uma solução abrangente para gerenciar suas operações, incluindo finanças, recursos humanos e cadeia de suprimentos.

Um dos principais benefícios do software da *SAP* é sua capacidade de integração com outros sistemas e fontes de dados, permitindo que as empresas otimizem seus processos e melhorem a eficiência. Por exemplo, o software de finanças da SAP pode ser integrado com o sistema de contabilidade existente da empresa, permitindo relatórios e análises em tempo real de dados financeiros.

Outro benefício do software da *SAP* é sua escalabilidade. Seus produtos podem ser usados por empresas de todos os tamanhos, desde pequenas empresas até grandes corporações multinacionais. Conforme uma empresa cresce e se expande, seu software da *SAP* pode ser configurado e personalizado para atender às suas necessidades em constante mudança.

A *SAP* também é conhecida por seu compromisso com a inovação e seu investimento em pesquisa e desenvolvimento. A empresa está constantemente atualizando seu software para acompanhar as mudanças nas necessidades das empresas e no cenário tecnológico. Por exemplo, a *SAP* agora oferece soluções baseadas na nuvem, permitindo que os clientes acessem o seu software de qualquer lugar, a qualquer momento [1].

Em conclusão, a *SAP* é líder em software de negócios e fornece uma solução abrangente para gerir operações de negócios. Seus produtos são utilizados por empresas de todos os tamanhos, são escaláveis e estão constantemente evoluindo para atender às mudanças nas necessidades das empresas. Seja uma pequena empresa ou uma grande corporação, o software da *SAP* pode ajudar a otimizar os processos, melhorar a eficiência e obter uma vantagem competitiva.

Figura 1 - Exemplo das áreas de uso do SAP

2.1.2 Remedy

Remedy é uma poderosa ferramenta para gerir Tickets que é utilizada por organizações de todos os tamanhos para otimizar suas operações de serviço e suporte. Desenvolvido pela *BMC Software*, o *Remedy* é uma solução abrangente de gestão de serviços de TI (ITSM) que permite às organizações gerir incidentes, problemas, mudanças e solicitações de maneira eficiente e efetiva [2].

Uma das principais características do *Remedy* é sua flexibilidade. Ele pode ser personalizado para atender às necessidades únicas de cada organização, tornando-o ideal para empresas de todos os tamanhos e setores. Seja uma pequena empresa com poucos funcionários ou uma grande corporação com milhares de funcionários, o *Remedy* pode ser configurado de acordo com as necessidades específicas.

Outro benefício importante do *Remedy* é sua capacidade de automatizar e otimizar os processos de serviço e suporte. Com o *Remedy*, os tickets podem ser encaminhados automaticamente para a pessoa certa com base na prioridade, habilidade ou outros critérios. Isso ajuda a garantir que os tickets sejam tratados de maneira rápida e eficiente, o que leva a tempos de resolução mais rápidos e a uma maior satisfação do cliente [2].

O *Remedy* também fornece às organizações um repositório centralizado para todos os seus dados de serviço e suporte. Esses dados podem então ser usados para gerar relatórios e realizar análises, o que pode fornecer valiosas insights sobre as operações de serviço e suporte. Por exemplo, as organizações podem usar os dados do *Remedy*

para identificar tendências e padrões nas submissões de tickets, o que pode ajudá-las a priorizar seus esforços e melhorar seus processos de serviço e suporte.

Um dos benefícios mais importantes do *Remedy* é sua integração com outros sistemas. O *Remedy* pode ser integrado a uma ampla gama de sistemas, incluindo sistemas de gestão do relacionamento com o cliente (CRM), sistemas de planejamento de recursos empresariais (ERP) e outras ferramentas ITSM. Essa integração permite que as organizações usem o *Remedy* em conjunto com seus outros sistemas, o que ajuda a otimizar as operações e a melhorar a eficiência [3].

Além disso, o *Remedy* também é conhecido por sua segurança e confiabilidade. A BMC Software está comprometida em garantir que o *Remedy* seja seguro e confiável, e a empresa fornece atualizações contínuas e suporte para garantir que a ferramenta permaneça atualizada e eficaz. Com o *Remedy*, as organizações podem ter a certeza de que suas operações de serviço e suporte estão em boas mãos [4].

Em conclusão, o *Remedy* é uma ferramenta poderosa para gestão de tickets que oferece às organizações a flexibilidade, automação e insights que eles precisam para otimizar suas operações de serviço e suporte. Seja uma pequena empresa ou uma grande corporação, o *Remedy* é uma solução eficaz que pode ajudar a melhorar seus processos de serviço e suporte e fornecer aos seus clientes o suporte de alta qualidade.

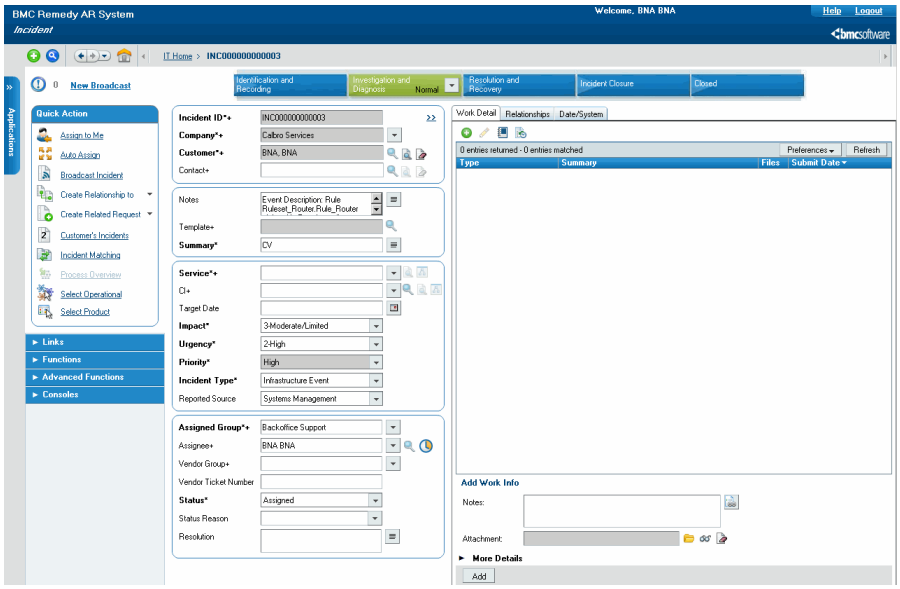


Figura 2 - Imagem ilustrativa da tool Remedy

2.1.3 ALM Octane

O *ALM Octane* é uma ferramenta de software projetada para gestão do ciclo de vida de aplicações no campo do desenvolvimento de software. É um produto da *Micro Focus* e é projetado para ajudar as equipas a gerir e acompanhar seus projetos de desenvolvimento de software do início ao fim.

O *ALM Octane* fornece uma variedade de recursos para apoiar o processo de desenvolvimento de software, incluindo gestão de requisitos, gestão de testes e rastreamento de defeitos. Ele também inclui suporte para metodologias agile, com recursos como gestão de *backlog* e planeamento de *sprint* [5].

Um dos principais benefícios do *ALM Octane* é a sua capacidade de integração com outras ferramentas de software comumente usadas no processo de desenvolvimento de software, como *JIRA* e *Jenkins*. Isso permite que as equipas movam dados entre as ferramentas e garantam que todas as partes interessadas tenham acesso às informações mais atualizadas [5].

Em termos de gestão de projetos, *o ALM Octane* fornece visibilidade em tempo real sobre o estado e o progresso do projeto, facilitando a identificação de possíveis problemas e riscos. Ele também inclui recursos de colaboração, como a capacidade de compartilhar painéis e relatórios, que podem ajudar as equipas a trabalhar de forma mais eficaz.

Em geral, o *ALM Octane* pode ser uma ferramenta valiosa para equipas de desenvolvimento de software que desejam simplificar seus processos de *ALM* e melhorar a colaboração e a comunicação ao longo do ciclo de vida do desenvolvimento de software.

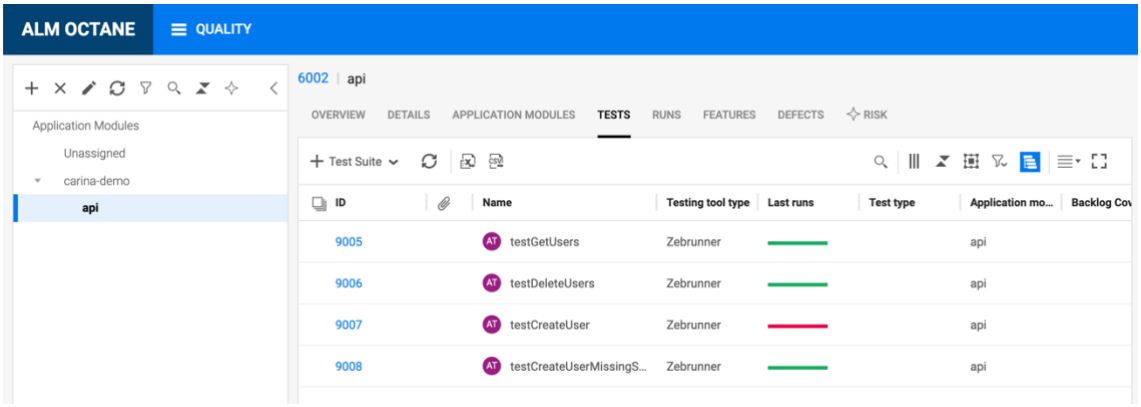


Figura 3 - Imagem ilustrativa da tool Octane

2.2 Ferramenta de Suporte para guardar e tratar dados

2.2.1 Oracle DataBase

O *Oracle Database*, frequentemente referido simplesmente como Oracle, é um sistema de gestão de base de dados relacionais (RDBMS) desenvolvido pela *Oracle Corporation.* É um dos sistemas de gestão de base de dados mais amplamente utilizados no mundo e é conhecido pela sua escalabilidade, confiabilidade e segurança.

O *Oracle Database* é projetado para lidar com grandes volumes de dados e transações, tornando-o adequado para uso em aplicações de nível empresarial. Ele suporta uma ampla gama de tipos de dados, incluindo dados de texto, numéricos e de data/hora, e fornece uma variedade de recursos de indexação e otimização de consulta para garantir que os dados possam ser recuperados de forma rápida e eficiente [6].

Além da sua funcionalidade de base de dados central, a *Oracle* fornece uma gama de ferramentas e recursos adicionais para ajudar os desenvolvedores e administradores a gerir e otimizar as suas bases de dados. Estes incluem ferramentas para backup e recuperação de bases de dados, monitorização e ajuste de desempenho e gestão de segurança.

Um dos principais benefícios do *Oracle Database* é a sua capacidade de escalar para atender às necessidades de aplicações grandes e complexas. Ele suporta *clustering* e

replicação multi-nó, o que permite às organizações distribuir dados em vários servidores para garantir alta disponibilidade e recuperação de desastres [6].

Em geral, o *Oracle Database* é um sistema robusto e confiável de gestão de base de dados que é adequado para uso em aplicações de nível empresarial. O seu conjunto de recursos avançados e escalabilidade fazem dele uma escolha popular para organizações que precisam de gerir grandes volumes de dados e transações.

2.3 Ferramentas para correr scripts e dar display dos resultados

2.3.1 Tableau

O *Tableau* é uma ferramenta de visualização de dados e inteligência empresarial que permite aos utilizadores criar dashboards, relatórios e gráficos interativos e visualmente apelativos. Foi originalmente desenvolvido pela *Tableau Software,* que foi adquirida pela *Salesforce* em 2019 [7].

Uma das principais vantagens do *Tableau* é a sua facilidade de uso. O *Tableau* suporta uma ampla gama de fontes de dados, incluindo folhas de cálculo, bases de dados e plataformas baseadas em nuvem, como *Amazon Web Services e Google Cloud.*

Outra vantagem do *Tableau* é a sua capacidade de lidar com grandes volumes de dados. Inclui funcionalidades como a mescla de dados e agregação de dados, que permitem aos utilizadores combinar dados de várias fontes e analisá-los de várias maneiras. O *Tableau* também fornece uma variedade de opções de visualização, incluindo gráficos de barras, gráficos de linhas, gráficos de dispersão e mapas.

As funcionalidades interativas do *Tableau* são outra vantagem chave. Os utilizadores podem interagir com as visualizações de dados ao passar o cursor sobre os pontos de dados, filtrar os dados por vários critérios e aprofundar a informação em detalhes mais específicos. Essas funcionalidades permitem aos utilizadores explorar os dados de uma maneira que não é possível com gráficos e relatórios estáticos [7].

O *Tableau* também fornece uma gama de funcionalidades adicionais para utilizadores avançados, incluindo análise estatística, modelagem de dados e previsão. Também

permite aos utilizadores partilhar e colaborar em visualizações de dados através da sua plataforma online, o *Tableau Public*.

Em geral, o *Tableau* é uma ferramenta poderosa de visualização de dados e inteligência empresarial que pode ajudar as organizações a tomar melhores decisões, transformando dados complexos em visualizações compreensíveis. A sua ampla gama de funcionalidades e flexibilidade tornam-no uma escolha popular para empresas e organizações de todos os tamanhos.

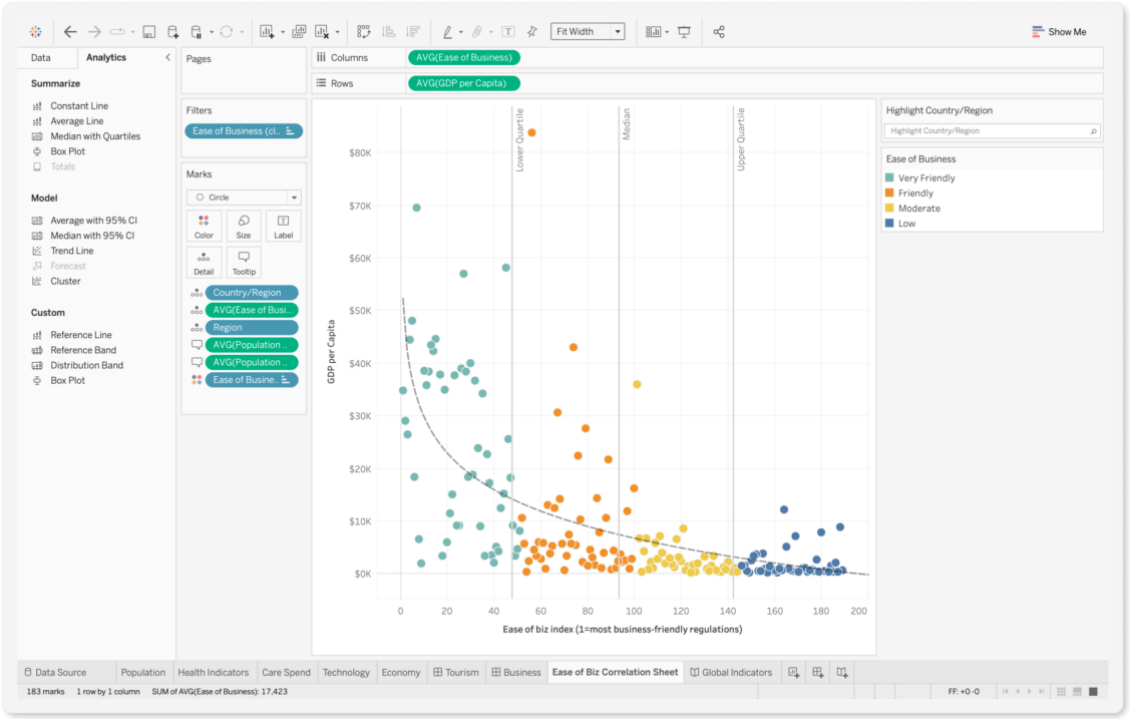


Figura 4 - Imagem ilustrativa da tool Tableau

2.3.2 Red Hat

A *Red Hat* é uma empresa líder em soluções de software empresarial de código aberto. O seu sistema operativo, o *Red Hat Enterprise Linux,* é um dos mais populares no mercado de servidores corporativos e é utilizado por empresas em todo o mundo.

Além disso, a *Red Hat* oferece uma plataforma chamada *OpenShift*, que é uma plataforma de containers baseada em *Kubernetes* para desenvolvimento, implantação e gestão de aplicativos. Essa plataforma permite que os desenvolvedores criem e implantem aplicativos em containers de forma rápida e eficiente.

Uma das funcionalidades do *OpenShift* é permitir a execução de *notebooks Jupyter* de *Python* em seus containers. Isso é feito através da criação de um projeto Jupyter no *OpenShift*, que permite que os *notebooks Python* sejam criados e executados dentro de um ambiente seguro e controlado pelo administrador do sistema.

Para executar *notebooks Python no OpenShift*, é necessário primeiro criar um projeto *Jupyter no OpenShift*. Em seguida, é possível carregar os *notebooks Python* para o projeto *Jupyter* e executá-los usando o *kernel Python*. Os *notebooks* podem ser executados interactivamente e as saídas são exibidas diretamente no *notebook*.

Além disso, o *OpenShift* também suporta o uso de bibliotecas e módulos *Python* externos. Isso significa que os *notebooks Python* podem usar qualquer biblioteca ou módulo que esteja disponível no ambiente *OpenShift* [8].

Em resumo, a *Red Hat* é uma plataforma de software empresarial de código aberto que permite a execução de *notebooks Jupyter de Python* em seus containers por meio da criação de um projeto *Jupyter* no *OpenShift*. Essa funcionalidade permite que os desenvolvedores criem e executem *notebooks Python* de forma rápida e eficiente num ambiente seguro e controlado.

2.4 Fluxo de Trabalho e Interação das Ferramentas

O fluxo de trabalho começa quando um utilizador solicita o desenvolvimento de um programa SAP ABAP. A equipa de desenvolvimento recebe a solicitação e inicia o processo de criação e codificação do programa. Após o desenvolvimento, o programa é submetido a testes rigorosos para garantir sua funcionalidade e ausência de erros. Caso não haja problemas, o programa é enviado ao utilizador para implantação.

No entanto, em alguns casos, podem ocorrer problemas ou erros no programa após a implantação. Quando isso acontece, os utilizadores abrem um ticket na plataforma Remedy, que serve como uma ferramenta de gestão de incidentes. A equipa de suporte recebe o ticket e cria uma entrada correspondente no Octane, uma plataforma de gestão de testes e desenvolvimento. Essa entrada no Octane está associada ao ticket do Remedy e ao ID de transporte, que será utilizado para vincular o programa no SAP.

2.5 Utilização em Conjunto das Ferramentas

A utilização em conjunto das ferramentas Remedy e Octane é fundamental para o processo de resolução de problemas. O Remedy atua como um ponto central para os utilizadores reportarem incidentes, enquanto o Octane é utilizado para monitorar o desenvolvimento e os testes relacionados aos tickets do Remedy.

Quando um ticket é associado a um ID de transporte no Octane, isso permite que os desenvolvedores associem o problema específico ao programa SAP correspondente. Um desenvolvedor pega o ticket e utiliza a informação contida no Octane para compreender o problema e refazer parte do código para corrigi-lo. Além disso, o desenvolvedor também tem acesso à informação adicional sobre o programa, como histórico de alterações e informações de testes.

2.6 Benefícios da Organização e Integração

A organização dos processos e a integração das ferramentas Remedy, Octane e análise de dados constituem uma abordagem abrangente para a gestão de problemas e melhoria contínua dos programas SAP ABAP. Essa abordagem permite uma comunicação mais eficiente entre utilizadores, equipa de suporte e desenvolvedores, resultando em resoluções mais rápidas e eficazes de problemas. Além disso, a análise de dados e a previsão contribuem para a identificação proativa de questões e a tomada de decisões informadas, promovendo a qualidade e a eficiência do desenvolvimento de software.

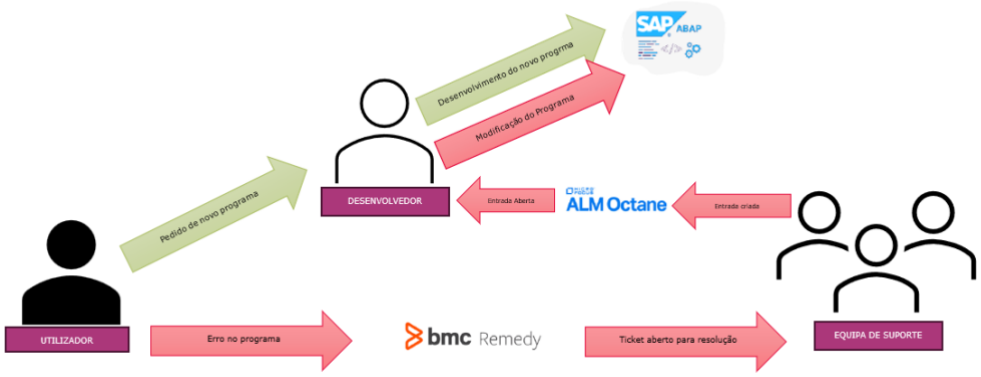


Figura 5 - Passo a Passo desde o pedido do programa a resolução de um possível erro

3. Conceitos Fundamentais

3.1 Introdução à *Machine Learning* e sua utilidade neste caso

Machine Learning é uma área da Inteligência Artificial que se concentra em desenvolver algoritmos capazes de aprender a partir de dados. Esses algoritmos são capazes de identificar padrões, fazer previsões e tomar decisões, sem serem explicitamente programados para isso. Eles são treinados com dados de entrada e de saída, e a partir desse treino, são capazes de generalizar para novos dados [9].

No contexto da gestão eficiente de desenvolvimentos baseados em *SAP ABAP*, a utilização de técnicas de *Machine Learning* pode ser extremamente útil. Com a grande quantidade de dados gerados diariamente pela *Infineon Technologies AG*, é difícil para a equipe de IT analisar e compreender todas as informações de maneira eficiente. A utilização de modelos de *Machine Learning* pode permitir a identificação automática de padrões e a criação de previsões precisas sobre o comportamento futuro dos desenvolvimentos.

Por exemplo, a utilização de algoritmos de *Machine Learning* pode ajudar a equipa de IT a identificar os desenvolvimentos que mais precisam de atenção e a prever a probabilidade de futuras mudanças serem necessárias. Isso permitirá que a equipa tome decisões informadas sobre quando é necessário reavaliar ou reescrever um programa, economizando tempo e recursos.

Além disso, a utilização de técnicas de *Machine Learning* pode permitir a criação de modelos de previsão mais precisos para calcular a importância dos programas. Esses modelos podem ser treinados com dados históricos de mudanças e problemas encontrados, permitindo a criação de previsões mais precisas e confiáveis [9].

Portanto, a utilização de técnicas de *Machine Learning* pode ser uma ferramenta poderosa para a *Infineon Technologies AG* na gestão eficiente de seus desenvolvimentos baseados em *SAP ABAP*. Ela pode ajudar a equipa de IT a tomar decisões informadas e a priorizar seus recursos, economizando tempo e dinheiro, além de melhorar o desempenho geral do sistema.

3.2 Avaliação de modelos

Após a construção de modelos de *Machine Learning,* é necessário avaliá-los para determinar a sua eficácia na solução do problema em questão. A avaliação de modelos é uma parte fundamental do processo de desenvolvimento de soluções baseadas em *Machine Learning*, pois permite determinar se o modelo é capaz de fazer previsões precisas e confiáveis [10].

Existem várias métricas utilizadas na avaliação de modelos, cada uma delas com um propósito específico. Algumas das métricas mais comuns para problemas de classificação incluem:

- *Accuracy*: É a proporção de previsões corretas em relação ao total de previsões realizadas. É uma métrica amplamente utilizada, mas não é apropriada para todos os tipos de problemas, especialmente quando as classes não estão equilibradas.

- Precisão: É a proporção de verdadeiros positivos em relação ao total de previsões positivas. É uma métrica útil para problemas em que é importante minimizar os falsos positivos.

- *Recall*: É a proporção de verdadeiros positivos em relação ao total de exemplos positivos. É uma métrica útil para problemas em que é importante minimizar os falsos negativos.

- *F1 Score*: É uma média harmônica entre precisão e *recall*. É uma métrica útil para problemas em que é importante equilibrar a precisão e o *recall*.

- *AUC-ROC:* É a área sob a curva *ROC*, que é uma representação gráfica do desempenho do modelo em relação a diferentes limiares de probabilidade. É uma métrica útil para problemas em que é importante avaliar a capacidade do modelo de classificar exemplos positivos e negativos [10].

Já para problemas de regressão, algumas das métricas mais comuns incluem:

- *Mean Squared Error (MSE)*: É uma métrica utilizada para medir a média do quadrado dos erros entre os valores previstos e os valores reais. É uma medida muito utilizada em problemas de regressão, onde o objetivo é prever um valor contínuo. Quanto menor o valor do MSE, melhor é o desempenho do modelo [11].

- *Mean Absolute Error (MAE):* É uma métrica que mede a média do valor absoluto dos erros entre os valores previstos e os valores reais. Também é muito utilizada em problemas de regressão e indica a magnitude média dos erros de previsão. Quanto menor o valor do MAE, melhor é o desempenho do modelo [11].

Tanto as métricas de classificação quanto as métricas de regressão são importantes na avaliação dos modelos de *Machine Learning*, pois fornecem informações sobre a precisão das previsões e ajudam a identificar quais modelos são mais adequados para o problema em questão [11].

Na avaliação de modelos de *Machine Learning*, é importante ter em mente que a escolha do modelo e dos Hiper parâmetros pode afetar significativamente o desempenho do modelo. É necessário realizar experimentos com diferentes modelos e Hiper parâmetros para determinar a melhor configuração para o problema em questão.

Por fim, é importante ressaltar que a avaliação de modelos é um processo contínuo. À medida que mais dados se tornam disponíveis ou à medida que o problema muda, pode ser necessário atualizar o modelo ou reavaliar.

3.3 Aprendizagem Supervisionada

A Aprendizagem Supervisionada é uma das principais abordagens de *Machine Learning,* que permite que um modelo seja treinado para fazer previsões com base em dados rotulados. Nessa abordagem, o modelo é alimentado com dados de entrada e suas respetivas saídas esperadas, com o objetivo de aprender a relação entre os dados de entrada e saída e, assim, ser capaz de fazer previsões precisas em novos dados. A Aprendizagem Supervisionada é amplamente utilizada em diversas áreas, como finanças, medicina, marketing e muitas outras, e tem sido responsável por grandes avanços em muitos campos. Neste tipo de abordagem, é importante selecionar as variáveis mais relevantes para treinar o modelo e utilizar técnicas de avaliação para escolher o melhor algoritmo e ajustar os Hiper parâmetros. Com a Aprendizagem Supervisionada, é possível resolver uma ampla variedade de problemas, desde classificação até regressão [12].

3.3.1 Regressão Linear

Regressão Linear é um método de Aprendizagem Supervisionada utilizado para modelar a relação entre uma variável dependente contínua e uma ou mais variáveis independentes. É uma técnica amplamente utilizada em *Machine Learning* e análise de dados para prever valores numéricos [13].

A ideia por trás da Regressão Linear é encontrar uma linha reta que melhor se ajuste aos dados, minimizando a distância entre os valores reais e os valores previstos pela linha. Essa linha é representada pela equação $y = mx + b$ onde y é a variável dependente, x é a variável independente, m é o coeficiente angular e b é o intercepto.

O objetivo da Regressão Linear é encontrar os valores de m e b que minimizem a soma dos quadrados dos erros (SSE), que é a diferença entre os valores reais e os valores previstos pela linha. Essa técnica é conhecida como Regressão Linear de Mínimos Quadrados [14].

Existem vários tipos de Regressão Linear, incluindo Regressão Linear Simples e Regressão Linear Múltipla. Na Regressão Linear Simples, há apenas uma variável independente, enquanto na Regressão Linear Múltipla há duas ou mais variáveis independentes.

A Regressão Linear é uma técnica útil para previsão de valores numéricos em uma ampla variedade de problemas, desde previsão de preços de imóveis até previsão de demanda de produtos. No entanto, é importante lembrar que a Regressão Linear tem suas limitações e pode não ser adequada para todos os problemas. É necessário avaliar cuidadosamente os dados e as hipóteses antes de aplicar a técnica de Regressão Linear [14].

3.3.2 Regressão Logistica

Regressão Logística é um algoritmo de aprendizagem supervisionada utilizado em problemas de classificação binária, ou seja, em que o objetivo é prever uma classe dentre duas possíveis opções. Ela é uma extensão da Regressão Linear e baseia-se num modelo probabilístico que estima a probabilidade de um exemplo pertencer a uma determinada classe [9].

O modelo de Regressão Logística usa uma função sigmoide para converter a saída do modelo, que pode variar de menos infinito a mais infinito, em uma probabilidade que varia entre 0 e 1. Essa função sigmoide é definida como:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Para treinar o modelo, é necessário ajustar os parâmetros de forma que a função de perda, que mede a diferença entre as probabilidades estimadas e as classes reais dos exemplos, seja minimizada. A função de perda mais comumente usada na Regressão Logística é a função de entropia cruzada, que é definida como:

$$L(y, \hat{y}) = -[y \log \hat{y} + (1 - y) \log (1 - \hat{y})]$$

A ideia é penalizar o modelo caso ele atribua uma baixa probabilidade para a classe correta.

Uma vez que o modelo é treinado, ele pode ser usado para fazer previsões para novos exemplos, que serão classificados de acordo com a probabilidade estimada pelo modelo. É comum definir um limiar de probabilidade (geralmente 0.5) para decidir qual classe atribuir ao exemplo [9].

A Regressão Logística é um algoritmo simples e eficiente para problemas de classificação binária, e é muito utilizado em áreas como detecção de spam, análise de sentimentos e diagnóstico médico. No entanto, ela pode não ser adequada para problemas com mais de duas classes ou em que as classes não são linearmente separáveis. Nesses casos, é necessário utilizar outras técnicas de aprendizagem de máquina.

3.3.3 Árvores de Decisão

As árvores de decisão são um algoritmo de aprendizagem supervisionada usado para resolver problemas de classificação e regressão. O algoritmo divide o conjunto de dados em subconjuntos menores e mais simples, até que o subconjunto final contenha exemplos homogêneos em relação à variável de destino. Essa estratégia de divisão permite que o modelo tome decisões com base num conjunto de regras que podem ser facilmente interpretadas por seres humanos [15].

Uma árvore de decisão é composta por um conjunto de nós interconectados por meio de arestas. Cada nó na árvore representa uma decisão, enquanto as arestas conectam as decisões às possíveis ações ou consequências. O nó inicial da árvore é chamado de

nó raiz, enquanto os nós que não têm filhos são chamados de folhas. Cada nó interno da árvore é associado a uma variável de entrada, e as arestas que saem dele representam as possíveis saídas para essa variável. O processo de construção da árvore continua até que não seja possível fazer mais divisões ou até que um critério pré-definido de parada seja atendido [15].

Para construir uma árvore de decisão, é necessário escolher uma métrica de impureza para avaliar a qualidade da divisão em cada nó. As métricas mais comuns são a entropia e o índice Gini. A entropia mede a incerteza em relação à distribuição das classes na amostra, enquanto o índice Gini mede a probabilidade de classificação incorreta de uma observação aleatória da mesma classe.

Uma das vantagens das árvores de decisão é a sua capacidade de lidar com dados em falta e *outliers*, pois eles são tratados como uma classe separada. Além disso, a interoperabilidade das árvores de decisão as torna úteis em aplicações em que é importante entender como o modelo chegou a uma determinada decisão [9].

No entanto, as árvores de decisão podem ser sensíveis a mudanças nos dados e propensas ao *overfitting*, o que pode levar a uma baixa capacidade de generalização. Para superar esses problemas, é possível utilizar técnicas de poda, que consistem em remover ramos desnecessários da árvore, ou *ensemble methods*, que combinam várias árvores de decisão para obter um modelo mais robusto e geralmente mais preciso.

3.3.4 K-Nearest Neighbours (k-NN)

K-Nearest Neighbors (KNN) é um algoritmo de Machine Learning utilizado tanto para problemas de classificação quanto de regressão. Ele é um dos algoritmos mais simples e fáceis de entender [15].

O algoritmo KNN funciona da seguinte maneira: dado um conjunto de dados de treino, ele calcula a distância entre os pontos desse conjunto e um novo ponto de entrada (que precisa ser classificado ou estimado). Em seguida, ele encontra os K pontos mais próximos do novo ponto e usa as suas classes ou valores de saída para determinar a classe ou valor de saída do novo ponto.

A escolha do valor de K é um hiper parâmetro do modelo e pode afetar significativamente o desempenho do algoritmo. Valores pequenos de K podem tornar o modelo mais sensível a ruídos nos dados, enquanto valores grandes de K podem reduzir a capacidade do modelo de capturar a estrutura dos dados.

O algoritmo KNN pode ser aplicado em problemas de classificação e regressão. No caso da classificação, a saída do modelo será a classe mais comum entre os K vizinhos mais próximos do novo ponto. Já no caso da regressão, a saída do modelo será a média dos valores de saída dos K vizinhos mais próximos [13].

O KNN pode ser uma boa opção para problemas com conjuntos de dados pequenos ou com estruturas complexas e não lineares, uma vez que o algoritmo é capaz de se adaptar a essas estruturas sem a necessidade de especificar uma função de decisão paramétrica. No entanto, ele pode ter um desempenho ruim em conjuntos de dados grandes ou em problemas com muitas dimensões, uma vez que o cálculo da distância entre pontos pode se tornar computacionalmente caro. Além disso, a escolha do valor de K pode ser difícil e pode afetar significativamente o desempenho do modelo.

3.3.5 Maquinas de Vetores de Suporte (SVM)

As Máquinas de Vetores de Suporte, ou SVM (do inglês, *Support Vector Machines*), são um algoritmo de aprendizagem supervisionada que pode ser usado tanto para tarefas de classificação quanto de regressão.

O objetivo do SVM é encontrar um hiperplano que possa separar os exemplos em duas ou mais classes. Em problemas de classificação binária, esse hiperplano é uma linha que divide o espaço em dois lados, cada um correspondendo a uma das classes. Em problemas de classificação multiclasse, o hiperplano pode ser uma superfície ou uma hipersuperfície que separa as classes [14].

O SVM é capaz de lidar com dados que não são linearmente separáveis por meio do uso de uma técnica chamada *kernel*. O *kernel* transforma os dados de entrada num espaço de dimensão superior, onde pode ser mais fácil encontrar um hiperplano que separe as classes. Alguns exemplos de *kernels* comuns são o *kernel* linear, o *kernel* polinomial e o kernel gaussiano (também conhecido como RBF).

Além de separar as classes, o SVM também busca maximizar a margem entre o hiperplano e os exemplos mais próximos de cada classe, chamados de vetores de suporte. A margem é a distância entre o hiperplano e os vetores de suporte. Essa abordagem é conhecida como margem máxima e é uma forma de reduzir a generalização do modelo e evitar *overfitting* [15].

Uma das principais vantagens do SVM é que ele é capaz de lidar com conjuntos de dados de alta dimensionalidade e complexos. Além disso, o SVM é um algoritmo eficiente e rápido em muitos casos. No entanto, o SVM pode ser sensível à escolha do *kernel* e dos hiperparâmetros, o que pode afetar significativamente o desempenho do modelo [15].

Em resumo, as Máquinas de Vetores de Suporte são um algoritmo de aprendizagem supervisionada poderoso que pode ser usado para classificação e regressão em conjuntos de dados complexos e de alta dimensão. Eles são particularmente úteis quando a separação linear das classes não é possível e quando é importante maximizar a margem entre as classes.

3.3.6 Redes Neuronais Artificiais (ANNs)

Redes Neuronais Artificiais (ANNs) são um tipo de modelo de Machine Learning que baseia-se no funcionamento dos neurônios do cérebro humano para realizar tarefas de classificação ou regressão. As ANNs são compostas por camadas de neurônios artificiais que processam a informação e geram uma saída. Essas camadas podem ser divididas em três tipos: camadas de entrada, camadas ocultas e camadas de saída.

A camada de entrada é responsável por receber os dados de entrada e enviar para a camada oculta, que processa os dados e envia para a camada de saída, que retorna a resposta final do modelo. Cada neurônio na camada oculta é conectado a todos os neurônios da camada anterior e posterior, permitindo que as informações sejam transmitidas e processadas em todas as camadas da rede [9].

Os pesos e os vieses são ajustados durante o treino da rede, que consiste em alimentar a rede com exemplos de dados rotulados e ajustar os pesos e os vieses para minimizar o erro entre a saída esperada e a saída produzida pela rede. Isso é feito usando um algoritmo de otimização, como o gradiente descendente [15].

As ANNs têm sido usadas com sucesso em muitas áreas, incluindo reconhecimento de fala, visão computacional, processamento de linguagem natural e previsão financeira. No entanto, o treino de uma rede neural pode ser computacionalmente intensivo e requer grandes conjuntos de dados rotulados para obter bons resultados.

Uma das principais vantagens das ANNs é a capacidade de lidar com dados não lineares e realizar tarefas complexas de classificação e regressão. Além disso, as ANNs são capazes de aprender padrões e relações entre os dados, o que pode ser útil em tarefas de previsão [15].

No entanto, as ANNs podem ser propensas a *overfitting*, ou seja, ajustar os dados de treino com muita precisão, mas falhar em generalizar para novos dados. Além disso, as ANNs podem ser difíceis de interpretar, pois as conexões entre os neurônios são complexas e não intuitivas.

3.3.7 Random Forest Regressor

Random Forest Regressor é um algoritmo de aprendizagem de máquina utilizado para problemas de regressão. É uma técnica baseada em árvores de decisão que usa um conjunto de árvores de decisão aleatórias para gerar previsões precisas.

A *Random Forest Regressor* é construída a partir de um conjunto de árvores de decisão, em que cada árvore é construída a partir de um subconjunto aleatório dos dados de treino e de um subconjunto aleatório das variáveis de entrada. Durante o processo de treino, cada árvore é treinada com um subconjunto diferente dos dados e variáveis, garantindo que as árvores sejam independentes e diversificadas.

Ao fazer previsões, a *Random Forest Regressor* faz a média das previsões de todas as árvores individuais, resultando em uma previsão mais precisa e robusta do que qualquer árvore individual [14].

A *Random Forest Regressor* tem várias vantagens em relação a outros modelos de regressão. Ele é capaz de lidar com dados em falta, valores atípicos e variáveis categóricas sem a necessidade de pré-processamento adicional. Além disso, ele é

menos propenso a *overfitting* do que outros modelos de regressão, tornando-se uma escolha popular para problemas de regressão em geral.

Em resumo, a *Random Forest Regressor* é uma técnica poderosa de aprendizagem de máquina que pode ser usada para prever valores numéricos com precisão. É um algoritmo versátil que pode lidar com uma ampla variedade de tipos de dados e é especialmente útil quando a precisão é uma prioridade e o *overfitting* é uma preocupação.

3.3.8 Hurdle model HuberRegressor

O *Hurdle Model Huber Regressor* é uma técnica de aprendizagem de máquina utilizada para realizar análises de regressão com dados que apresentam excesso de zeros ou heterocedasticidade. É uma combinação de duas técnicas: o modelo *Hurdle* e o modelo *Huber* [14].

O modelo *Hurdle* é um modelo estatístico que trata da presença de excesso de zeros nos dados. Ele é utilizado quando os dados apresentam uma alta frequência de zeros e, por isso, não se ajustam bem a uma distribuição normal. O modelo *Hurdle* consiste em duas partes: a primeira parte modela a probabilidade de um dado observado ser zero ou não-zero, enquanto a segunda parte modela a distribuição da variável de interesse condicionalmente aos valores não nulos.

Já o modelo *Huber* é uma técnica de regressão robusta que lida com dados que apresentam heterocedasticidade, ou seja, quando a variância dos erros de previsão varia ao longo dos valores da variável independente. Ele é capaz de lidar com dados que possuem *outliers*, que podem influenciar negativamente a estimativa dos parâmetros de um modelo de regressão [15].

O *Hurdle Model Huber Regressor* combina as duas técnicas para criar um modelo que lida com ambos os problemas: excesso de zeros e heterocedasticidade. Ele é capaz de modelar a probabilidade de um dado observado ser zero ou não-zero, e depois modelar a distribuição da variável de interesse condicionalmente aos valores não nulos, enquanto lida com a heterocedasticidade dos dados.

O modelo é especialmente útil para dados econômicos e financeiros, onde é comum encontrar excesso de zeros e heterocedasticidade. Ele também pode ser aplicado em outras áreas, como em análises de dados ambientais e de saúde [14].

O *Hurdle Model Huber Regressor* é implementado utilizando técnicas de aprendizagem de máquina, como a regressão linear e a regressão logística. Ele é uma opção interessante para análises de dados que apresentam problemas de excesso de zeros e heterocedasticidade, e pode ser uma alternativa eficaz aos modelos tradicionais de regressão linear.

3.3.9 SARIMA

O modelo SARIMA (*Seasonal AutoRegressive Integrated Moving Average*) é uma extensão do modelo ARIMA que inclui uma componente de sazonalidade para modelar séries temporais com padrões sazonais. Ele é capaz de capturar tanto a tendência quanto a sazonalidade da série, tornando-o um modelo muito poderoso para previsão de séries temporais [16].

O modelo SARIMA consiste em quatro componentes principais: a componente autorregressiva (AR), a componente integrada (I), a componente de média móvel (MA) e a componente sazonal (S). A ordem de cada componente é definida pelos parâmetros p, d, q e P, D, Q, respetivamente.

A componente AR modela a relação entre uma observação e um número fixo de observações passadas, conhecido como a ordem AR (p). A componente MA modela a relação entre a observação e um erro de previsão passado, conhecido como a ordem MA (q). A componente I é responsável por transformar a série temporal em uma série estacionária, subtraindo a série da sua média ou diferenciando a série. A ordem I (d) é o número de diferenciações necessárias para tornar a série estacionária [16].

A componente sazonal é adicionada para capturar os padrões sazonais na série. Ela modela a relação entre uma observação e as observações no mesmo período em temporadas anteriores. A ordem sazonal AR (P) e MA (Q) referem-se à ordem das componentes autorregressiva e de média móvel sazonal, respetivamente. A ordem de diferenciação sazonal (D) é o número de diferenciações necessárias para tornar a série estacionária na dimensão sazonal.

A partir dessas componentes, o modelo SARIMA é capaz de modelar a série temporal e fazer previsões. Os parâmetros do modelo (p, d, q, P, D, Q) são determinados por meio de técnicas de otimização, como o método de máxima verossimilhança [16].

Em resumo, o modelo SARIMA é uma extensão do modelo ARIMA que inclui uma componente sazonal para modelar séries temporais com padrões sazonais. Ele é capaz de capturar tanto a tendência quanto a sazonalidade da série e seus parâmetros são determinados por meio de técnicas de otimização.

3.4 Processamento de Linguagem Natural

O Processamento de Linguagem Natural (PLN ou NLP) é um campo da inteligência artificial que se concentra na interação entre computadores e a linguagem humana. Ele visa capacitar os sistemas computacionais a entender, interpretar e gerar textos de maneira semelhante aos seres humanos. O PLN desempenha um papel fundamental em diversas aplicações, como *chatbots*, sistemas de tradução automática, análise de sentimentos, resumos automáticos, entre outros [15].

3.4.1 Princípios do Processamento de Linguagem Natural

O PLN envolve várias etapas e técnicas para processar e compreender a linguagem humana. Essas etapas incluem:

3.4.1.1 Pré-processamento de texto

Antes de iniciar a análise, os textos são submetidos a etapas de pré-processamento, como tokenização (divisão do texto em palavras ou unidades menores), remoção de *stopwords* (palavras comuns que não contribuem significativamente para o sentido), *stemming* (redução de palavras à sua forma raiz) e lematização (redução de palavras à sua forma base) [15].

3.4.1.2 Análise sintática e semântica

A análise sintática envolve a compreensão da estrutura gramatical do texto, identificando as relações entre palavras e as funções sintáticas de cada termo. Já a

análise semântica busca extrair o significado do texto, identificando entidades, relacionamentos e inferências.

3.4.1.3 Extração de recursos

Nessa etapa, são extraídos recursos relevantes dos textos para alimentar modelos de Machine Learning. Esses recursos podem incluir características como frequência de palavras, presença de termos específicos, estruturas gramaticais e características semânticas.

3.4.1.4 Modelagem e treino

Nesta etapa, são aplicados modelos de Machine Learning, como redes neurais, árvores de decisão, SVM (*Support Vector Machines)* ou modelos de linguagem, para treinar os sistemas a realizar tarefas específicas, como classificação de textos, sumarização automática ou tradução.

3.4.2 Aplicações do Processamento de Linguagem Natural

O PLN tem diversas aplicações em diferentes setores. Algumas delas incluem:

3.4.2.1 Chatbots e assistentes virtuais

Os *chatbots* e assistentes virtuais são sistemas que utilizam PLN para entender e responder a perguntas ou solicitações de usuários de forma natural e coerente.

3.4.2.2 Análise de sentimentos

A análise de sentimentos é uma aplicação que utiliza técnicas de PLN para identificar e classificar emoções expressas em textos, como comentários de redes sociais, avaliações de produtos, entre outros [15] .

3.4.2.3 Tradução automática

A tradução automática utiliza técnicas de PLN, como modelos de linguagem e algoritmos de Machine Learning, para traduzir automaticamente textos de um idioma para outro, facilitando a comunicação global [9]

3.4.2.4 Sumarização automática

A sumarização automática é a tarefa de extrair as informações mais relevantes de um texto longo e gerar um resumo conciso. O PLN é utilizado para identificar as informações-chave e gerar sumários que capturam os aspetos essenciais do texto original.

3.4.2.5 Análise de tópicos

A análise de tópicos envolve a identificação e o agrupamento de documentos com base nos temas discutidos neles. O PLN é utilizado para extrair os principais tópicos de um conjunto de documentos, permitindo uma compreensão abrangente de grandes volumes de texto [16].

3.4.2.6 Desafios e tendências no Processamento de Linguagem Natural

Embora o Processamento de Linguagem Natural tenha alcançado avanços significativos, ainda há desafios a serem superados. Alguns desses desafios incluem [17]:

3.4.2.7 Ambiguidade linguística

A linguagem humana é rica em ambiguidades, como palavras com múltiplos significados e construções sintáticas complexas. Lidar com a ambiguidade é um desafio para os sistemas de PLN, pois requer uma compreensão mais profunda do contexto.

3.4.2.8 Variações linguísticas

Os idiomas podem variar significativamente entre diferentes regiões e culturas, incluindo diferenças de vocabulário, gramática e expressões idiomáticas. Adaptar os sistemas de PLN para essas variações é um desafio importante.

3.4.2.9 Dados não estruturados

A maioria dos dados de texto disponíveis para análise é não estruturada, o que significa que não segue um formato padronizado. Isso dificulta a extração de informações relevantes e a criação de modelos eficazes [17].

3.4.2.10 Escassez de dados

O treino de modelos de PLN requer grandes volumes de dados anotados, ou seja, dados com rótulos que indicam a classificação ou a estrutura correta. No entanto, a anotação manual de dados é uma tarefa trabalhosa e cara, limitando a disponibilidade de conjuntos de dados anotados [17].

Em termos de tendências futuras, algumas áreas de pesquisa promissoras incluem:

3.4.2.10.1 Machine Learning

O uso de técnicas de deep learning, como redes neurais profundas, tem impulsionado avanços significativos no PLN. Esses modelos são capazes de aprender representações de texto mais complexas e obter melhores resultados em várias tarefas [16] .

3.4.2.10.2 Integração com conhecimento externo

A incorporação de conhecimento externo, como bases de conhecimento estruturadas ou ontologias, pode melhorar a compreensão e a geração de texto. Essa integração permite que os sistemas de PLN acedam informações adicionais e enriqueçam seu desempenho.

3.4.2.10.3 Abordagens multilíngues

A capacidade de lidar com múltiplos idiomas de forma eficiente é uma área de pesquisa em expansão. O desenvolvimento de modelos e técnicas que possam ser aplicados a diferentes idiomas facilitará a expansão e a adoção global de soluções de PLN [9] [17].

4. Revisão Bibliográfica

4.1 Introdução

A presente revisão bibliográfica abrange uma ampla gama de estudos e pesquisas relacionados ao projeto de tese proposto, que visa criar um gráfico de *heatmap* para analisar áreas da empresa com um alto número de incidentes em desenvolvimentos de software. A revisão aborda os seguintes tópicos: gerenciamento de incidentes, análise de tendências, visualização de dados, previsão de incidentes e processamento de linguagem natural.

4.2 Gestão de incidentes em desenvolvimentos de software

A gestão de incidentes é uma área crucial para garantir a qualidade e a estabilidade dos sistemas de software. Diversos estudos têm se dedicado a abordar esse tema, propondo diferentes metodologias e técnicas. Por exemplo, Smith et al. (2018) discutem a importância da análise de incidentes como uma prática essencial para melhorar a qualidade do software [17].

4.3 Análise de tendências e padrões de incidentes

A análise de tendências e padrões de incidentes é fundamental para identificar áreas problemáticas e tomar medidas corretivas adequadas. Estudos como Jones et al. (2016) utilizaram técnicas de mineração de dados para identificar padrões de incidentes em projetos de software [18].

4.4 Visualização de dados e gráficos de heatmap

A visualização de dados desempenha um papel crucial na compreensão e interpretação de grandes conjuntos de dados. Gráficos de *heatmap* são uma forma eficaz de representar dados multidimensionais. Por exemplo, Silva et al. (2017) propuseram o uso de *heatmaps* para visualizar padrões de incidentes em sistemas de software [19].

4.5 Previsão de incidentes em desenvolvimentos de software

A previsão de incidentes é uma área de pesquisa importante, pois permite antecipar e mitigar problemas futuros. Modelos de previsão, como ARIMA (*Autoregressive Integrated Moving Average*) e SARIMA (*Seasonal ARIMA*), têm sido amplamente utilizados nesse contexto. Por exemplo, Wang et al. (2019) aplicaram o modelo SARIMA para prever incidentes em projetos de desenvolvimento de software [17].

4.6 Processamento de Linguagem Natural (PLN)

O processamento de linguagem natural é uma área que oferece ferramentas e técnicas para extrair informações úteis de dados textuais. No contexto de incidentes em desenvolvimentos de software, o PLN pode ser aplicado para analisar descrições de incidentes e extrair insights relevantes. Por exemplo, Chen et al. (2018) utilizaram técnicas de PLN para analisar relatórios de incidentes e identificar padrões e tendências [17] [19].

4.7 Conclusão

A revisão bibliográfica apresentada oferece uma visão geral abrangente dos principais estudos e pesquisas relacionados ao projeto de tese proposto. Os estudos revisados abordam questões-chave, como gestão de incidentes, análise de tendências, visualização de dados, previsão de incidentes e processamento de linguagem natural. Essa revisão fornece uma base sólida para o desenvolvimento da pesquisa, abrindo caminho para a aplicação de abordagens e modelos que foram explorados nos estudos revisados.

5. Desenvolvimento e Implementação

5.1 Acesso a Dados

No contexto deste projeto, o acesso aos dados da empresa apresentou desafios significativos, uma vez que os dados estavam divididos em vários programas e bases de dados sem uma conexão direta entre eles. Nesta subsecção , descreveremos em detalhes os procedimentos adotados para obter acesso aos dados necessários nas diferentes plataformas utilizadas, como *Remedy, Octane e SAP*, destacando as implementações relevantes.

5.1.1 Acesso aos dados do Remedy

Os dados dos tickets na plataforma *Remedy* estavam armazenados num servidor externo. No entanto, ao realizar a extração dos dados, percebeu-se que havia uma quantidade significativa de informações não relevantes referentes a outras funcionalidades do *Remedy*. Para lidar com essa questão, foi necessário aplicar uma filtragem nos dados extraídos, selecionando apenas as informações pertinentes para o âmbito do projeto. Essa filtragem permitiu reduzir o volume de dados e focar nas áreas específicas da empresa em que os desenvolvimentos apresentavam incidentes, de acordo com os critérios estabelecidos. Para obter acesso a esses dados, foi necessário copiá-los para um servidor Oracle que foi criado para este projeto. Esse processo envolveu a configuração de um servidor Oracle e a criação de uma conexão segura com o servidor externo do *Remedy*. Por meio dessa conexão, os dados dos tickets foram transferidos e armazenados no servidor Oracle, garantindo assim a disponibilidade dos dados em tempo real. Além disso, uma conexão foi estabelecida para atualizar periodicamente os dados no servidor Oracle, mantendo-os atualizados.

5.1.2 Acesso aos dados do Octane

Da mesma forma que no caso do *Remedy*, os dados de resoluções de tickets na plataforma *Octane* estavam armazenados num servidor externo. Durante o processo de extração desses dados, também foi identificado um grande volume de informações não relevantes relacionadas a outras funcionalidades do *Octane*. Para contornar esse problema, foi necessário realizar uma filtragem dos dados, excluindo as informações

desnecessárias e mantendo apenas aquelas relacionadas aos desenvolvimentos e incidentes em questão. Essa filtragem contribuiu para a criação de um conjunto de dados mais focado e relevante para a análise e visualização no *heatmap* proposto. Para aceder esses dados, foi realizada uma cópia dos dados do *Octane* para o servidor Oracle criado anteriormente. Essa abordagem permitiu a centralização dos dados do *Octane* no mesmo ambiente do *Remedy*, facilitando a integração e análise dos dados num único local. Assim como no caso do *Remedy*, uma conexão foi estabelecida para garantir a atualização regular dos dados no servidor Oracle.

É importante ressaltar que a filtragem dos dados nos sistemas *Remedy* e *Octane* foi realizada com base em critérios definidos previamente, a fim de selecionar as informações mais pertinentes para a análise e previsão de incidentes nos desenvolvimentos da empresa. Essa abordagem permitiu concentrar os esforços nos aspetos específicos que impactam a qualidade e a estabilidade dos programas desenvolvidos, proporcionando uma visão mais precisa das áreas da empresa que requerem maior atenção e melhorias.

5.1.3 Acesso aos dados do SAP

Os dados dos programas da empresa estão armazenados na plataforma SAP. Para obter acesso a esses dados, foi necessário desenvolver um programa em SAP ABAP, uma linguagem específica para a plataforma SAP. Esse programa permitiu extrair os dados necessários para análise, incluindo informações sobre os programas alterados e as classes associadas a eles. Esses dados foram acedidos por meio de tabelas internas do SAP, que registavam as alterações e forneciam informações relevantes para a análise. Para garantir a segurança e a consistência dos dados, a conexão com o SAP foi estabelecida diretamente no *notebook Python* utilizado na implementação, evitando assim a exposição desnecessária dos dados fora da plataforma *SAP*.

5.1.4 Criação do servidor Oracle

Para a centralização dos dados provenientes das plataformas *Remedy* e *Octane*, foi necessário criar um servidor Oracle dedicado. Esse processo envolveu a instalação e configuração do banco de dados Oracle num ambiente adequado, garantindo a disponibilidade e o desempenho necessários para a manipulação dos dados. Além

disso, foram realizadas configurações de segurança e permissões adequadas para garantir a proteção dos dados e o acesso restrito somente aos usuários autorizados.

5.2 Análise e Tratamento dos Dados

5.2.1 Dados Recebidos e Irrelevantes

Durante o desenvolvimento do projeto, foram recebidos dados de diferentes fontes, como a plataforma *Remedy* e o *Octane*, que continham informações relacionadas aos desenvolvimentos e incidentes da empresa. No entanto, foi identificado que muitas das colunas presentes nos conjuntos de dados não eram relevantes para o âmbito do projeto, sendo necessário realizar uma análise e filtragem dessas informações.

Ao comparar os dados provenientes do *Remedy* e do *Octane*, verificou-se que algumas colunas representavam a mesma informação, mas em alguns casos, algumas estavam preenchidas e outras não. Para ter um conjunto de dados final mais completo, foi necessário realizar um processo de correspondência (match) entre os dois conjuntos de dados, garantindo que as informações correspondentes fossem combinadas em uma única coluna. Isso permitiu obter um conjunto de dados consolidado e mais abrangente para a análise.

5.2.2 Preenchimento de Colunas com NLP

Durante a análise dos dados, foi constatado que certas colunas, como "*Functional Area*", continham informações preenchidas de forma livre nas duas fontes de dados (*Remedy* e *Octane*). Isso levava a várias formas de representar a mesma área funcional ou, em alguns casos, o campo era deixado em branco. Para lidar com essa variabilidade, utilizou-se o processamento de linguagem natural (NLP) para preencher essas colunas.

Por meio da análise do texto livre, foi possível identificar referências a áreas funcionais e preencher a coluna "*Functional Area*" com base no conteúdo do texto. Esse processo envolveu a utilização de técnicas de NLP, como identificação de palavras-chave e análise de sentimentos, para determinar a área funcional mais adequada para cada registo [17].

5.2.3 Prioridade com Base em Análise de Sentimento

Outro exemplo de tratamento dos dados foi a coluna "Prioridade" no conjunto de dados. Em alguns casos, essa coluna não estava preenchida. Para resolver essa questão, utilizou-se a abordagem de análise de sentimentos com base no conteúdo do texto. Por meio de um dicionário de palavras-chave associadas a diferentes níveis de prioridade, foi possível atribuir valores de 0 a 4 à coluna "Prioridade", refletindo a prioridade atribuída pelo utilizador.

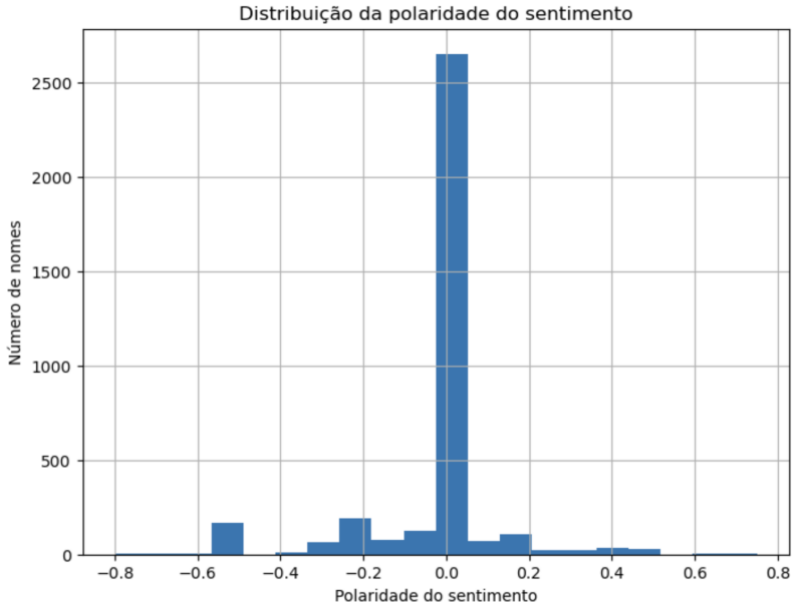


Figura 6 - Distribuição da polaridade do sentimento

5.2.4 Tratamento de Dados Repetidos

No conjunto de dados, foi observado que um único ticket criado no *Remedy* poderia ter vários incidentes associados no *Octane*. Para lidar com essa situação, os incidentes associados foram divididos em linhas separadas por meio da utilização de uma expressão regular (*regex*). Como não era possível determinar previamente o número de incidentes associados em cada linha, a expressão regular foi projetada para lidar com diferentes cenários [18].

No entanto, durante a análise dos dados, verificou-se a presença de números de incidentes inválidos, uma vez que esse campo era preenchido manualmente. Para garantir a integridade dos dados, utilizou-se uma função para validar se o conteúdo do

campo seguia o padrão "INC" seguido de 12 dígitos. Dessa forma, apenas os incidentes com números válidos foram considerados para a análise.

Adicionalmente, identificou-se que os incidentes repetidos não eram relevantes para o objetivo do projeto, pois apenas indicavam que mais de uma pessoa havia criado um ticket para o mesmo problema. Portanto, realizou-se uma limpeza no *dataframe*, mantendo apenas os números de incidentes únicos para evitar duplicidades e garantir a representatividade adequada dos dados.

5.2.5 Exclusão de Colunas Irrelevantes

Com o objetivo de manter apenas as informações relevantes para o projeto, algumas colunas foram excluídas do conjunto de dados resultante da junção do *Remedy* com o *Octane*. Essas colunas, como "*BPLM*", "*Incident No*" (que continha a lista inicial de incidentes associados ao ticket) e "*Defects*", não contribuíam diretamente para a análise e visualização dos desenvolvimentos com incidentes.

5.2.6 Tratamento dos Dados da Coluna "Conigma Transport Number"

Para estabelecer a ligação com a tabela do SAP, era necessário lidar com os dados da coluna *"Conigma Transport Number".* Inicialmente, as entradas que continham essa coluna com valores nulos foram removidas, uma vez que não eram úteis para o projeto. Da mesma forma que nos incidentes, foi aplicada a mesma abordagem de divisão em várias linhas para lidar com os casos em que um ticket possuía mais de um *"Conigma Transport Number"* associado.

A fim de garantir que apenas informações relevantes para o projeto fossem consideradas, foi utilizada uma coluna denominada "*New Feature or CR".* Essa coluna permitiu filtrar os dados que não estavam relacionados a bugs ou incidentes, mas sim a novas implementações ou alterações do sistema. Por meio da aplicação de uma expressão regular, foi possível remover os registos que continham informações não pertinentes ao âmbito do projeto.

5.2.7 Criação de um Novo DataFrame

Foi então criado um *DataFrame* que contém as seguintes colunas*: Incident Number, Go Live Date, Functional Area,* Prioridade para o Utilizador e *Effort* em Dias de trabalho do ticket. Essas informações são relevantes para a análise posterior dos incidentes.

Além disso, foi realizado um processo de concatenação dos *Conigma Transport Numbers* em uma única coluna. Para garantir a integridade dos dados, foram aplicadas expressões regulares para filtrar apenas os valores que seguiam o padrão de um *Transport Number* válido. Dessa forma, foram removidos quaisquer dados inconsistentes ou inválidos presentes nessa coluna.

5.2.8 Análise por Área Funcional

Com base nesse novo *DataFrame (df_inc),* foi criado outro *DataFrame* chamado *df_plot* para analisar os incidentes por área funcional. Esse DataFrame foi gerado utilizando o método *group by* da coluna *Functional Area*, seguido de uma contagem dos Incidentes dessa área. Os dados foram ordenados por *Functional Area* e, em seguida, foi configurado um gráfico de barras para visualizar os resultados. No gráfico, o eixo x representa as áreas funcionais e o eixo y mostra o número de tickets associados a cada área.

5.2.9 Filtro por Áreas de Desenvolvimento

Durante essa etapa do projeto, foi identificado que apenas os tickets criados nas áreas de DIO (Desenvolvimento de Infraestrutura e Operações) e DPO (Desenvolvimento de Produtos e Operações) eram relevantes para a análise, uma vez que essas áreas são responsáveis pelo desenvolvimento das aplicações e não estão relacionadas a testes ou ambientes de produção. Para filtrar os dados e obter apenas as informações dessas áreas, foi aplicada uma expressão regular como filtro nos dados do *DataFrame*.

5.2.10 Criação de Métricas

Com o objetivo de obter diferentes formas de análise dos dados, foram criadas métricas para atribuir relevância a cada incidente. Essas métricas são diferentes das que estão

presentes nos *DataFrames* até o momento e são calculadas e armazenadas em novas colunas [17].

A primeira métrica criada foi a *Priority* (Prioridade), que indica a prioridade atribuída pelo utilizador. Para tornar a métrica numérica, os valores originais, como *Low, Medium, High* e *Emergency*, foram convertidos para valores de 1 a 4.

A segunda métrica é o *Effort* (Esforço), que representa os dias de trabalho gastos para resolver o incidente. Nessa métrica, os dias de trabalho são agrupados em diferentes faixas. Se o esforço for de 1 dia, recebe o peso 1. Se estiver entre 2 e 4 dias, recebe o peso 2. Se estiver entre 4 e 7 dias, recebe o peso 3. E se for superior a 7 dias, recebe o peso 4. Essa atribuição de pesos leva em consideração valores contratuais definidos pela empresa *Infineon*, que remunera os desenvolvedores com base no tempo gasto para solucionar cada incidente.

A terceira e última métrica é a Geral, que atribui um peso de 0.7 a essa métrica ao *Effort* calculado anteriormente e um peso de 0.3 à prioridade também calculada anteriormente. Essa métrica visa considerar a relevância de cada uma das outras duas métricas para a empresa.

Essas métricas fornecem uma base para a análise posterior dos dados e auxiliam na identificação e priorização dos incidentes de acordo com diferentes critérios de relevância.

```
d_priority = {"Low": 1, "Medium": 2, "High": 3 , "Emergency": 4}
df_inc['Priority'] = df_inc['Priority'].replace(d_priority)
d_effort = {1:1, range(2,4): 2, range(4,7): 3, range(7,157): 4}
df_inc["Effort in MD"] = df_inc["Effort in MD"].replace(d_effort)
df_inc["Effort in MD"] = df_inc["Effort in MD"].astype(int)

df_inc['Weight_geral'] = 0.7 * df_inc["Effort in MD"] + 0.3 * df_inc['Priority']
df_inc['Weight_Effort'] = df_inc["Effort in MD"]
df_inc['Weight_Priority'] = df_inc['Priority']
```

Figura 7 - Criação das métricas

5.2.11 Conversão das Datas para um Formato Uniforme

Outra etapa importante do tratamento de dados envolveu a conversão das datas presentes no *DataFrame* para um formato uniforme. As datas podem estar em diferentes

formatos e representações, o que pode dificultar a análise correta dos dados. Portanto, foi necessário padronizar todas as datas num único formato reconhecido.

Por meio de técnicas de manipulação de *strings* e expressões regulares, foi possível extrair as informações relevantes de cada data e convertê-las para o formato desejado. Dessa forma, todas as datas no *DataFrame* foram normalizadas e tornaram-se comparáveis e consistentes para análises posteriores.

5.2.12 Criação de DataFrames por Métrica

Visando facilitar o trabalho de previsão futura, foram criados três *DataFrames* adicionais, um para cada métrica estabelecida anteriormente. Esses *DataFrames* foram agrupados por área funcional, mês, classe e método, e também calcularam a média da métrica em questão para cada combinação.

Essa abordagem permitiu ter uma visão mais granular dos dados, analisando as métricas em diferentes contextos e permitindo a identificação de padrões específicos em cada categoria. Esses *DataFrames* auxiliarão nas análises de previsão e no entendimento das relações entre as métricas e os fatores que as influenciam em cada contexto específico.

Essas etapas adicionais de tratamento de dados garantiram a consistência e a qualidade das informações, preparando o conjunto de dados para análises mais aprofundadas e fornecendo uma base sólida para a identificação de tendências e a realização de previsões futuras.

5.2.13 Criação de um Histograma dos Sistemas com Mais Tickets

Com base no DataFrame df_inc, foi necessário realizar um tratamento adicional dos dados para criar um histograma que permitisse visualizar os sistemas com maior número de tickets. Essa análise é importante para identificar quais sistemas apresentam um maior volume de incidentes e podem requerer maior atenção.

Para criar o histograma, foi necessário agrupar os dados por sistema e contar o número de tickets associados a cada sistema. Em seguida, os dados foram organizados em

ordem decrescente de acordo com o número de tickets. Com essa estrutura de dados preparada, foi possível gerar um histograma que apresentava os sistemas no eixo x e a contagem de tickets no eixo y.

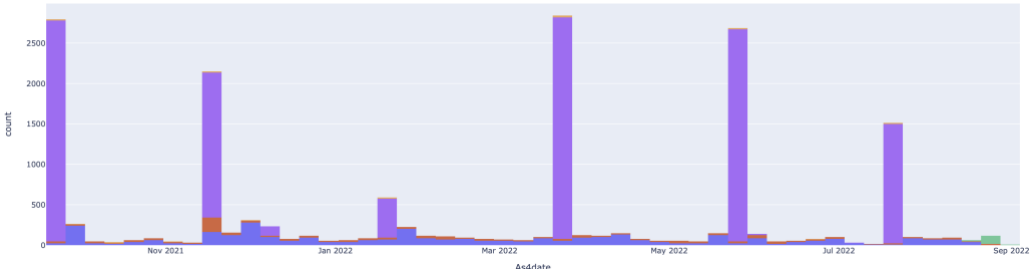


Figura 8 - Histograma com a distribuição dos tickets por sistema

5.2.14 Áreas Funcionais numa Word Cloud

Além das etapas anteriores, uma outra análise realizada foi a criação de uma *Word Cloud* para visualizar as áreas funcionais com o maior número de tickets. *A Word Cloud* é uma representação visual das palavras mais frequentes num texto ou conjunto de dados, onde o tamanho das palavras é proporcional à sua frequência.

Para criar a *Word Cloud*, foram utilizados os dados do *DataFrame df_inc* e a coluna referente às áreas funcionais dos tickets. Primeiramente, foram agrupados os tickets por área funcional e contabilizado o número de ocorrências de cada área. Em seguida, essas informações foram utilizadas para gerar a *Word Cloud.*

A *Word Cloud* resultante apresentou as áreas funcionais como palavras, sendo que o tamanho de cada palavra era determinado pelo número de tickets associados àquela área. Assim, as áreas com maior número de tickets ficaram visualmente mais destacadas na nuvem de palavras.

Essa análise por meio da *Word Cloud* permitiu uma rápida visualização das áreas funcionais que possuíam um maior volume de tickets, fornecendo uma visão geral das prioridades em relação ao suporte e manutenção das aplicações. Essa informação pode ser útil para direcionar recursos e esforços de desenvolvimento, focando nas áreas que apresentam maior demanda e impacto nos processos de negócio da empresa. [20]

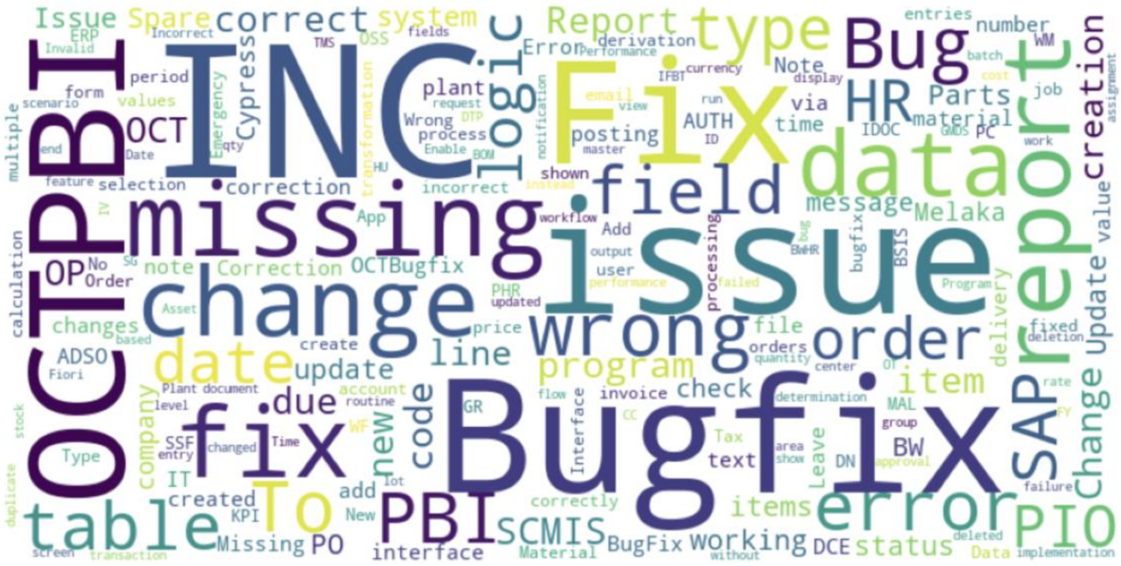


Figura 9 - Word Cloud com as palavras mais usadas nos tickets

5.2.15 Areas mais problemáticas num gráfico de barras

Estes gráficos tinham como objetivo visualizar a distribuição das métricas por área funcional, proporcionando uma compreensão mais detalhada do desempenho de cada área em relação aos incidentes.

Para cada métrica *(Priority, Effort e Geral),* foi criado um gráfico de barras separado. O eixo x do gráfico representava as áreas funcionais, enquanto o eixo y representava o valor da métrica. Cada barra no gráfico correspondia a uma área funcional e seu comprimento era proporcional ao valor da métrica calculada para aquela área.

Ao analisar esses gráficos de barras, era possível identificar as áreas funcionais com maior prioridade, esforço ou relevância geral, conforme definido pelas métricas. Essas informações permitiam uma comparação direta entre as áreas e auxiliavam na identificação de padrões, tendências ou áreas que requeriam maior atenção e ação por parte da equipe responsável.

A utilização dos gráficos de barras para visualização das métricas fornecia uma representação clara e concisa do desempenho das áreas funcionais, facilitando a análise e tomada de decisões baseadas nos resultados obtidos. Essa visualização contribuía para uma compreensão mais aprofundada do impacto dos incidentes em

cada área e auxiliava na alocação adequada de recursos e esforços para melhorias contínuas.

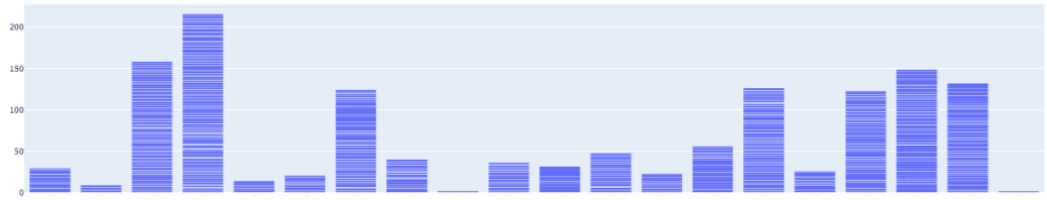


Figura 10 - PlotBar que indica as areas com mais problemas

5.3 Previsões

5.3.1 Random Forest Regressor

O primeiro algoritmo testado foi *Random Forest Regressor* escolheu-se devido à sua capacidade de lidar com conjuntos de dados complexos e não lineares, além de ser adequado para problemas de regressão, como é o caso das métricas de importância, esforço e prioridade.

Para avaliar o desempenho do modelo, foram utilizadas métricas como o *Mean Square Error (MSE) e o Mean Absolute Error (MAE).* O MSE mede a média dos erros ao quadrado, fornecendo uma medida do quão próximo os valores previstos estão dos valores reais. O MAE, por sua vez, calcula a média dos valores absolutos dos erros, fornecendo uma medida da magnitude dos erros cometidos pelo modelo [21].

Valores aceitáveis para essas métricas variam de acordo com o contexto do problema e as expectativas. Em geral, quanto menores os valores do MSE e do MAE, melhor é o desempenho do modelo. Por exemplo, um MSE próximo de zero indica uma boa capacidade de previsão, pois os valores previstos estão muito próximos dos valores reais. O mesmo se aplica ao MAE, onde um valor próximo de zero indica menor magnitude de erros cometidos [20].

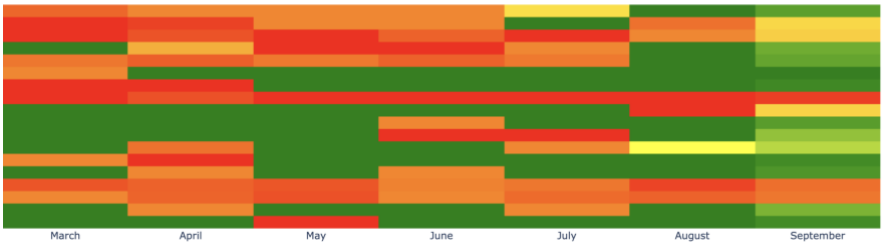


Figura 11 - HeatMap gerado usando random forest regressor (métrica Prioridade)

Metricas	Valor do Erro
MSE	1.376
MAE	0.966

Tabela 1 - Valores de Erro para random forest regressor usando a métrica de prioridade

No entanto, no caso em questão, os valores obtidos de MSE e MAE não foram considerados aceitáveis, o que indica que o modelo inicial não estava a produzir resultados satisfatórios. Para melhorar o desempenho, foi adotada a estratégia de otimização dos Hiper parâmetros do modelo por meio do método *Random Search*. Esse método busca de forma aleatória entre diferentes combinações de Hiper parâmetros, visando encontrar a configuração que resulte num melhor desempenho. Mas os valores de MAE e MSE, não sofreram uma alteração relevante.

Antes de aplicar os dados ao modelo, foi necessário converter a coluna da Área Funcional utilizando o método *LabelEncoder*. Isso deve-se ao fato de que o algoritmo de regressão *Random Forest* requer que todas as variáveis sejam numéricas. Ao realizar a codificação da coluna da Área Funcional, os nomes das áreas foram substituídos por valores numéricos correspondentes, permitindo que o *dataframe* fosse utilizado no modelo.

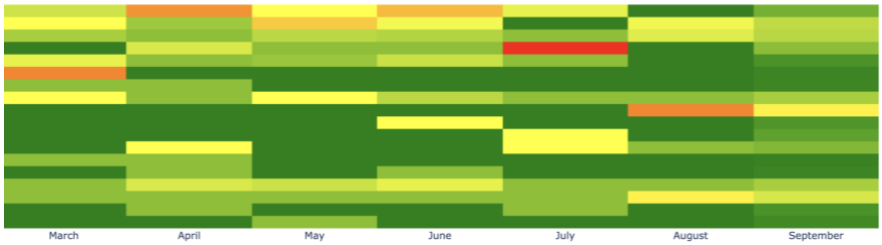


Figura 12 - HeatMap gerado usando random forest regressor (métrica Effort)

Metricas	Valor do Erro
MSE	1.173
MAE	0.834

Tabela 2 - Valores de Erro para random forest regressor usando a métrica de effort

Para calcular o valor do próximo mês, foi utilizado uma função do *Python* para obter a data atual do computador e, em seguida, extrair apenas o mês. Em seguida, foi adicionado 1 ao valor do mês para obter o valor numérico do próximo mês. Para o processamento, os meses no *dataframe* foram convertidos em valores numéricos utilizando a biblioteca "*calendar*" do *Python*. No final da previsão, os valores foram novamente convertidos para seus equivalentes por extenso.

O modelo foi treinado com os dados disponíveis e, em seguida, fez-se a previsão adicionando os valores previstos à coluna de *Importance/Effort/Priority.* Para calcular os dois meses seguintes, os valores previstos foram armazenados no *dataframe* e o processo de treino e previsão foi repetido utilizando um conjunto de dados que incluía os valores previamente previstos.

Ao final de cada previsão, foi criada uma tabela pivot e, em seguida, um *heatmap*. Essa visualização permitia uma compreensão mais intuitiva e visual dos valores previstos, facilitando a identificação de padrões, tendências ou áreas de maior importância, esforço ou prioridade. O *heatmap* proporcionava uma representação gráfica dos valores previstos, onde as cores indicavam a magnitude ou intensidade dos valores, auxiliando na interpretação dos resultados.

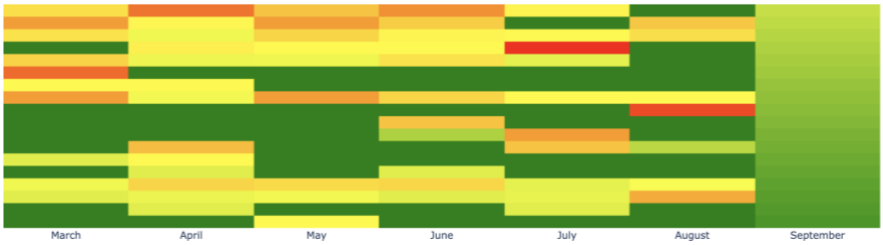


Figura 13 - HeatMap gerado usando random forest regressor (métrica importância)

Metricas	Valor do Erro
MSE	0.834
MAE	1.021

Tabela 3 - Valores de erro para random forest regressor usando a métrica de importância

5.3.2 Redes Neuronais

Além do *Transformed Target Regressor*, também foi utilizado o *MLPRegressor*, que é um algoritmo de Machine Learning baseado em redes neurais do tipo *Perceptron* de Múltiplas Camadas (*Multilayer Perceptron).* Essa escolha baseia-se na capacidade das redes neurais de aprender relacionamentos complexos entre as variáveis de entrada e a variável de saída [21].

No *MLPRegressor,* foram definidas duas camadas ocultas com tamanhos de 100 e 50 neurônios, respetivamente, assim como no modelo anterior. O número máximo de iterações foi definido em 1000, o que representa o número máximo de vezes que o modelo será ajustado aos dados durante o treino. A estratégia utilizada para a transformação da variável de destino foi novamente a média.

A coluna de Área Funcional foi codificada usando o *LabelEncoder*, conforme mencionado anteriormente, para permitir o processamento dos dados pela rede neural. A estratégia de cálculo dos meses e a abordagem de treino também foram mantidas consistentes com os algoritmos anteriores.

O modelo *MLPRegressor* foi treinado dividindo os dados num conjunto de treino, correspondendo a 80% dos dados, e um conjunto de teste, correspondendo aos 20% restantes. Essa divisão permite avaliar a capacidade do modelo de generalizar e fazer previsões em dados não vistos durante o treino.

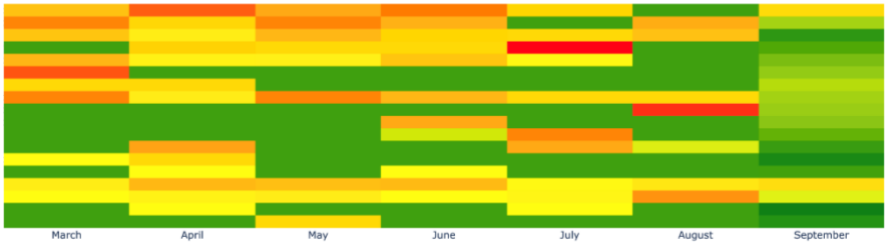


Figura 14 - HeatMap gerado usando Redes Neuronais usando a métrica importância

Após o treino, foi gerado um *heatmap* para visualizar os resultados da previsão. Essa visualização gráfica ajuda a comparar os valores previstos com os valores reais e a identificar possíveis padrões ou discrepâncias nas previsões geradas pela rede neural.

Por fim, o processo de avaliação e comparação dos modelos de redes neurais foi concluído com a geração dos *heatmaps*, que fornecem uma representação visual dos resultados obtidos, facilitando a interpretação e a análise dos valores previstos para as métricas de importância, esforço e prioridade.

Metricas	Valor do Erro
MSE	1.021
MAE	0.810

Tabela 4 - Valores de erro para redes neuronais usando a métrica de importância

5.3.3 Regressão Linear

A Regressão Linear é um modelo estatístico clássico que se adapta bem a casos em que existe uma relação linear entre as variáveis de entrada e a variável de saída. Nesse caso, busca-se encontrar uma relação linear que represente melhor os dados e possa ser utilizada para fazer previsões [17].

No contexto do projeto em questão, a Regressão Linear pode ser aplicada para prever as métricas de importância, esforço e prioridade com base nas variáveis de entrada disponíveis. Por exemplo, é possível explorar se existe uma relação linear entre a área funcional de um ticket e sua importância, ou se o esforço necessário para resolver um incidente está relacionado de forma linear com a prioridade atribuída pelo usuário.

Na implementação da Regressão Linear, a coluna de Área Funcional foi novamente codificada utilizando o *LabelEncoder*, e a estratégia de cálculo dos meses foi mantida consistente com os modelos anteriores.

O conjunto de dados foi dividido num conjunto de treino (80%) e um conjunto de teste (20%) para avaliação do desempenho do modelo. Foram utilizadas estratégias de Hiper parâmetros para ajustar os parâmetros do modelo e tentar obter os melhores resultados possíveis.

No entanto, é importante mencionar que a Regressão Linear pode não ser a escolha mais adequada para este projeto, considerando a natureza complexa e não linear dos dados. As métricas de importância, esforço e prioridade podem depender de uma combinação de fatores não lineares, como a interação entre diferentes variáveis ou a presença de padrões não lineares nos dados. Portanto, é esperado que os resultados obtidos pela Regressão Linear sejam menos precisos e tenham um desempenho inferior em comparação com outros modelos mais flexíveis, como as Redes Neurais ou *Random Forest.*

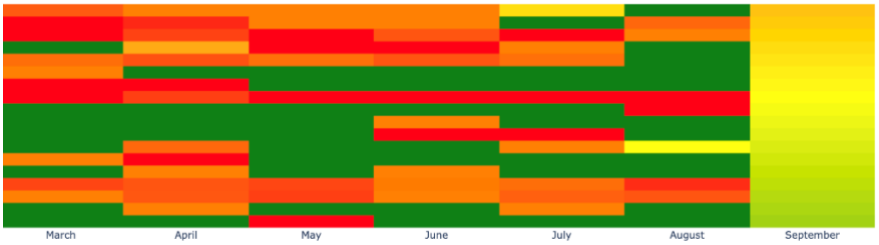


Figura 15 - HeatMap usando Regressão Linear (métrica importância)

Apesar dos resultados insatisfatórios da Regressão Linear, ainda assim foram criadas uma tabela pivot e um *heatmap* para visualizar os valores previstos e possibilitar uma análise comparativa entre as métricas. Essas visualizações podem fornecer insights adicionais sobre as relações lineares presentes nos dados e ajudar a identificar possíveis padrões ou tendências.

Metricas	Valor do Erro
MSE	1.878
MAE	1.113

Tabela 5 - Valores de erro para regressão linear usando a métrica de importância

5.3.4 Huber Regressor

O *Huber Regressor* é um modelo de regressão robusto que combina características da regressão linear e da regressão por mínimos quadrados. Ele é adequado para casos em que os dados podem conter *outliers* ou valores discrepantes que podem afetar negativamente o desempenho de modelos de regressão tradicionais. O *Huber Regressor* é menos sensível a *outliers*, pois utiliza uma função de perda que penaliza menos os erros menores e mais os erros maiores [20].

No contexto do projeto, onde estamos lidar com dados de métricas e buscamos prever valores de importância, esforço e prioridade, pode haver casos em que os dados contenham *outliers* ou valores discrepantes. Por exemplo, pode haver incidentes com uma importância extremamente alta ou uma prioridade muito baixa que não sigam a distribuição geral dos dados. O *Huber Regressor* pode ajudar a lidar com esses casos, fornecendo previsões mais robustas e menos influenciadas por esses *outliers*.

Para aplicar o *Huber Regressor,* a coluna de Área Funcional, que estava em formato de texto por extenso, foi novamente codificada utilizando o *LabelEncoder* para representação numérica, permitindo que o modelo trabalhe com esses dados. A estratégia de cálculo dos meses foi mantida consistente com os modelos anteriores.

Foram realizados ajustes do modelo com o objetivo de encontrar os melhores Hiper parâmetros para *o Huber Regressor*. O modelo foi ajustado por Área Funcional e Mês, juntamente com a métrica escolhida (importância, esforço ou prioridade). Os resultados obtidos foram considerados aceitáveis, uma vez que o MSE e o MAE se aproximaram de zero, indicando que as previsões estavam próximas dos valores reais [17].

Além disso, também foi utilizado o *Transformed Target Regressor*, que é um *wrapper do Scikit-learn* que permite aplicar transformações aos alvos (variáveis dependentes) antes de treinar o modelo de regressão. Essa abordagem é útil quando os valores dos alvos possuem uma distribuição não normal ou quando existem *outliers*. No caso específico do *Huber Regressor,* a estratégia utilizada foi a média, que busca minimizar a função de perda de forma geral.

Os modelos foram treinados com um conjunto de teste de tamanho de 20% e as previsões foram realizadas. Em seguida, foram criadas pivot *tables* e *heatmaps* para visualizar os resultados e permitir uma análise comparativa entre as métricas previstas.

No caso do *MLRegressor*, foi utilizado um modelo de regressão baseado em redes neurais, com uma camada oculta de tamanho 100,50 e um número máximo de iterações de 1000. A estratégia de média foi escolhida para a mesma finalidade de buscar uma boa estimativa do alvo.

Todas as estratégias e ajustes mencionados foram realizados com o objetivo de obter previsões mais precisas e confiáveis para as métricas de importância, esforço e prioridade com base nas informações disponíveis no conjunto de dados.

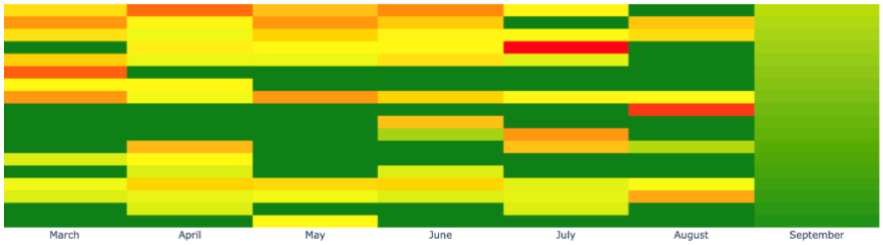


Figura 16 - HeatMap usando Huber Regressor (métrica importância)

Metricas	Valor do Erro
MSE	1.431
MAE	1.096

Tabela 6 - Valores de erro para huber regressor usando a métrica de importância

5.3.5 SARIMA

O modelo SARIMA *(Seasonal Autoregressive Integrated Moving Average*) é uma extensão do modelo ARIMA que leva em consideração a sazonalidade dos dados. Ele é adequado para análise e previsão de séries temporais, onde a existência de padrões sazonais é observada, como é o caso dos dados do projeto [3] [21].

No contexto do projeto, onde estamos lidar com dados que podem exibir padrões sazonais, como a importância, o esforço e a prioridade dos tickets ao longo do tempo, o modelo SARIMA se adapta bem, pois considera a influência de componentes sazonais, além de tendências e variações aleatórias.

Assim como nos modelos anteriores, a coluna de Área Funcional foi codificada utilizando o *LabelEncoder* para representação numérica, permitindo que o modelo trabalhe com esses dados. A estratégia de cálculo dos meses foi mantida consistente com os modelos anteriores.

O SARIMA foi ajustado por Área Funcional e Mês, juntamente com a métrica escolhida (importância, esforço ou prioridade). Através do ajuste do modelo SARIMA, foi possível obter previsões para o próximo mês, bem como para os dois meses subsequentes, utilizando as previsões anteriormente calculadas como entrada para os modelos subsequentes.

Os resultados obtidos com o modelo SARIMA foram considerados aceitáveis, uma vez que o MSE e o MAE estavam próximos de zero, indicando que as previsões estavam alinhadas com os valores reais. A aplicação do SARIMA permitiu capturar os padrões sazonais dos dados e fornecer previsões mais precisas e confiáveis para as métricas em questão.

Após as previsões, foram novamente criadas pivot *tables* e *heatmaps* para visualizar os resultados e permitir uma análise comparativa entre as métricas previstas, agora considerando a capacidade de prever os valores para os próximos meses.

A transformação dos dados em uma série temporal é essencial para a aplicação do modelo SARIMA. Ao considerar a natureza sequencial dos dados e os componentes sazonais, o modelo SARIMA é capaz de capturar os padrões temporais e fornecer previsões mais precisas. A previsão para múltiplos meses à frente foi realizada utilizando as previsões anteriores como entrada, o que permite uma projeção mais precisa dos valores das métricas no futuro.

*considerar como mês atual o mês de Agosto

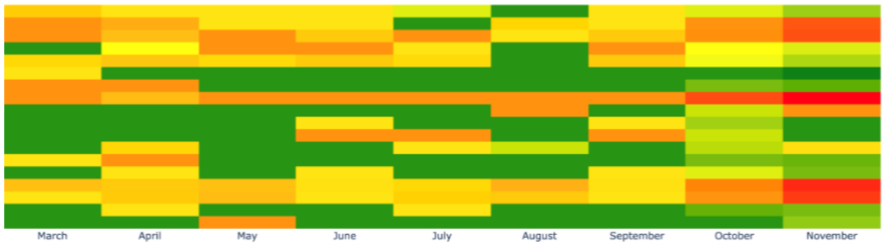


Figura 17 - Modelo Sarima, previsão dos próximos 3 meses

Métricas	Valor do Erro
MSE	0.231
MAE	0.456

Tabela 7 - lores de erro para Sarima usando a métrica de importância para o próximo mês

5.4 Analise dos Resultados

A fim de comparar os resultados dos diferentes modelos de previsão, foi elaborada uma tabela que apresenta os valores de MSE e MAE obtidos por cada modelo. Essa tabela permite uma visualização direta das métricas de desempenho e facilita a comparação entre os modelos testados.

Aqui está a tabela de comparação dos modelos:

	Random Forest Regressor	Reudes Neuronais	Regressão Linear	Huber Regressor	Sarima
MSE	1.34	0.98	1.88	1.43	0.23
MAE	1.02	0.81	1.11	1.10	0.46

Tabela 8 - Comparação dos valores de MSE e MAE nos diferentes modelos usados. (previsão de um mês)

Observando os valores de MSE e MAE, fica evidente que o modelo SARIMA obteve os resultados mais baixos, indicando uma melhor precisão e ajuste aos dados. Portanto, considerando a tabela de comparação e os critérios de avaliação, o modelo SARIMA foi

escolhido como o mais adequado para a previsão das métricas de importância, esforço e prioridade.

Após realizar a avaliação dos diferentes modelos de previsão, verificou-se que o modelo SARIMA obteve os valores de MSE (Erro Quadrático Médio) e MAE (Erro Médio Absoluto) mais baixos em comparação aos outros modelos testados.

O valor de MSE igual a 0.231450356778825 indica que a média dos erros quadrados entre os valores previstos e os valores reais é baixa, o que indica uma boa precisão nas previsões. Já o valor de MAE igual a 0.4565142183694756 mostra que a média dos erros absolutos entre os valores previstos e os valores reais também é baixa, indicando que o modelo é capaz de fazer previsões próximas aos valores reais.

Com base nesses resultados, o modelo SARIMA foi escolhido como o modelo final a ser implementado no *dashboard*. Ele será responsável por realizar as previsões das métricas de importância, esforço e prioridade para o próximo mês, assim como para os dois meses subsequentes, utilizando as previsões anteriores como entrada.

Ao implementar esse modelo no *dashboard* final, podemos ter acesso às previsões atualizadas das métricas, permitindo uma melhor tomada de decisão e planejamento nas atividades relacionadas ao projeto em questão.

5.5 Analise do HeatMap

Com base nas necessidades de identificação dos problemas no código, foram desenvolvidas análises adicionais para auxiliar na localização específica do problema. Essas análises envolvem a criação de *heatmaps*, gráficos de barras e análise de métodos dentro das classes. Vamos detalhar cada uma dessas etapas:

5.5.1 Heatmap de Classes Afetadas por Tickets

Foi criado um *heatmap* que indica o número de vezes que uma classe foi afetada devido a um ticket. Para isso, utilizou-se o campo *"Transport Number"* presente nos tickets, que foi comparado com as informações do SAP para extrair o nome do programa relacionado. O *heatmap* foi gerado agrupando as classes por área funcional, mostrando visualmente quais classes estão a enfrentar mais problemas ao longo do tempo. Essa informação permite aos desenvolvedores identificar as áreas que estão em vermelho ou que se tornarão vermelhas com base no *heatmap* do modelo SARIMA. A partir dessa identificação, é possível investigar e solucionar os problemas nessas classes.

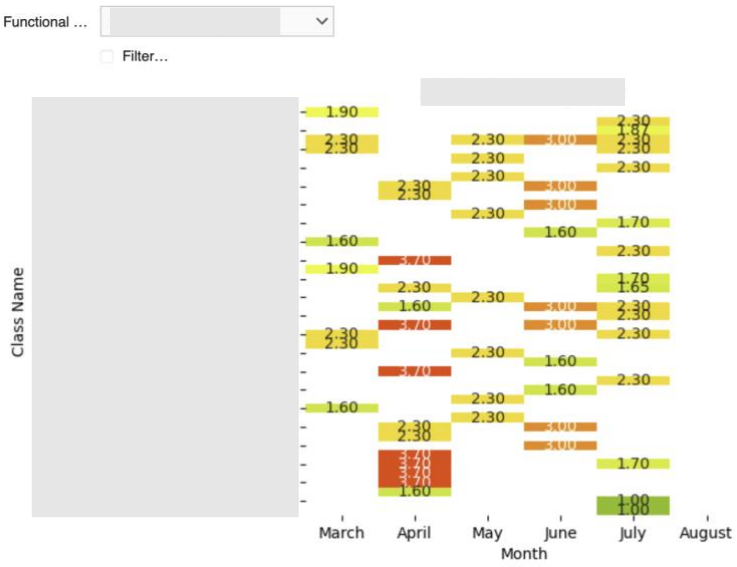


Figura 18 - Heatmap de Classes Afetadas por Tickets

5.5.2 Gráfico de Barras - Média de Mudanças nas Classes devido a Tickets

Foi criado um gráfico de barras que apresenta a média de mudanças em cada classe devido a tickets nos últimos 5 meses. Esse gráfico auxilia na avaliação do *heatmap* anterior, fornecendo uma visão mais detalhada das classes que estão a enfrentar mais alterações devido a tickets. É possível filtrar os programas do tipo "Z", que são programas desenvolvidos internamente na empresa, para focar na análise desses programas.

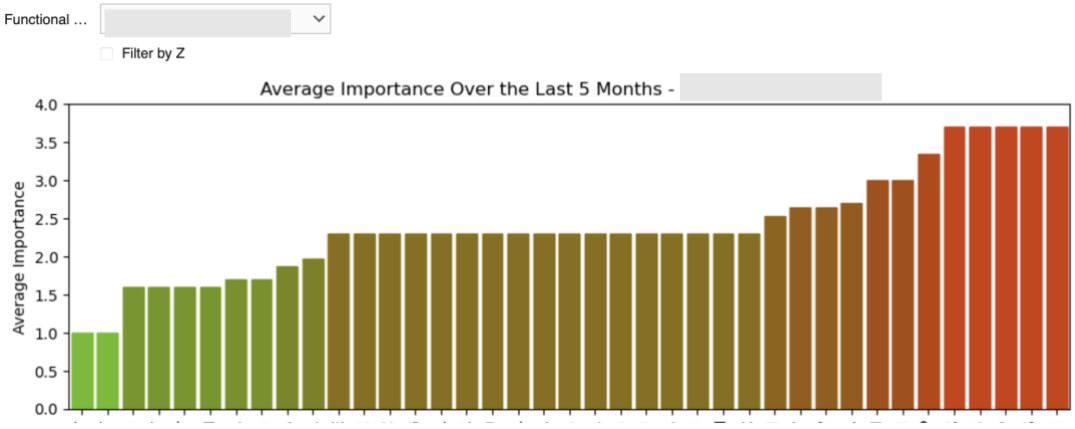


Figura 19 - Gráfico de Barras - Média de Mudanças nas Classes devido a Tickets

5.5.3 Heatmap de Métodos Afetados por Tickets

Por fim, quando os desenvolvedores já têm informações sobre as classes que estão enfrentando problemas, eles podem analisar mais detalhadamente os métodos dentro dessas classes. Foi criado um *heatmap* que indica quais métodos estão sendo mais impactados. Essa análise permite uma identificação mais precisa das partes do código que estão a causar problemas e podem exigir intervenção.

Essas análises visuais permitem que os desenvolvedores localizem rapidamente as áreas problemáticas no código, desde as classes até os métodos específicos, auxiliando no processo de investigação e resolução de problemas.

5.6 Criação do Dashboard no Tableau

Nesta etapa do projeto, o foco foi a criação de um *dashboard* no *Tableau*, que permitiria uma visualização mais interativa e intuitiva dos resultados obtidos. O processo envolveu a criação de uma base de dados virtual no *Tableau*, por meio de conexões com a base de dados original. Os dados foram tratados de maneira semelhante ao que foi feito no arquivo *Python*, garantindo consistência e coerência nos resultados.

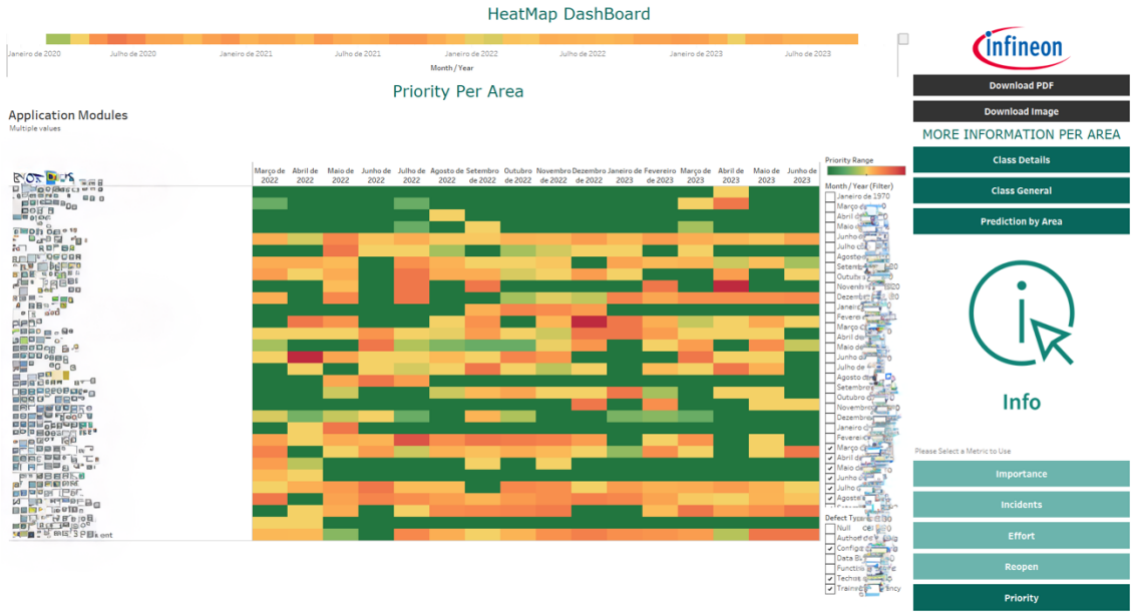


Figura 20 - Dashboard Principal

5.6.1 Visualização dos Gráficos

No *dashboard* do *Tableau*, foram incorporados diversos gráficos que foram gerados durante a análise dos dados. Um dos gráficos adicionados foi a média de cada uma das métricas *(Importance, Effort e Priority)* para cada área, considerando os últimos 6 meses. Essa visualização proporcionou uma visão geral do desempenho dessas métricas ao longo do tempo.

Outro gráfico relevante foi o gráfico de barras que mostrou os programas/classes mais frequentemente alterados nos últimos meses, independentemente da área funcional. Essa informação auxiliou na identificação das partes do código que passaram por mais mudanças e podem requerer atenção especial.

Figura 21 - DashBoard com previsão por área funcional

5.6.2 Adição de Novas Métricas e Atualização da Métrica de Importância

Para aprimorar a análise no *dashboard*, foram adicionadas novas métricas, como o número de incidentes e o número de vezes que um ticket foi reaberto. Essas métricas forneceram insights adicionais sobre a ocorrência de problemas e permitiram um melhor entendimento do contexto. Com a inclusão dessas novas métricas, a métrica de importância foi atualizada para considerar os diferentes elementos, utilizando uma fórmula ponderada:

$$0.50 * \text{Priority} + 0.30 * \text{Effort} + 0.20 * \text{Reopens}$$

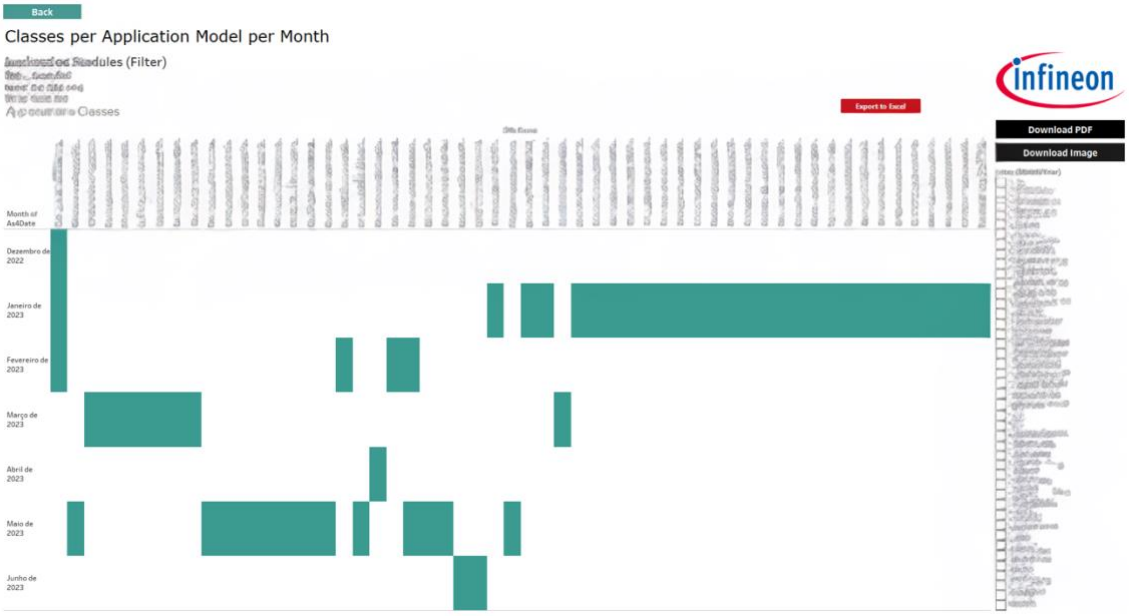


Figura 22 - Dashboard que indica quando ouve alteraçoes em cada programa num espaço temporal

5.6.3 Inclusão de Filtros no Dashboard

No intuito de tornar o *dashboard* mais versátil e adaptável às diferentes equipas que o utilizariam, foram adicionados filtros. Esses filtros permitiram a seleção de informações específicas e a personalização dos dados exibidos no *dashboard*. Entre os filtros adicionados estão as Áreas Funcionais e o Tipo de Ticket (*Defect Type).* Além desses, foi incluído o filtro de Mês/Ano, que possibilitou a alteração dos dados utilizados para a previsão em cada área funcional, assim como a previsão geral, e a modificação dos dados exibidos em cada um dos gráficos do *dashboard*.

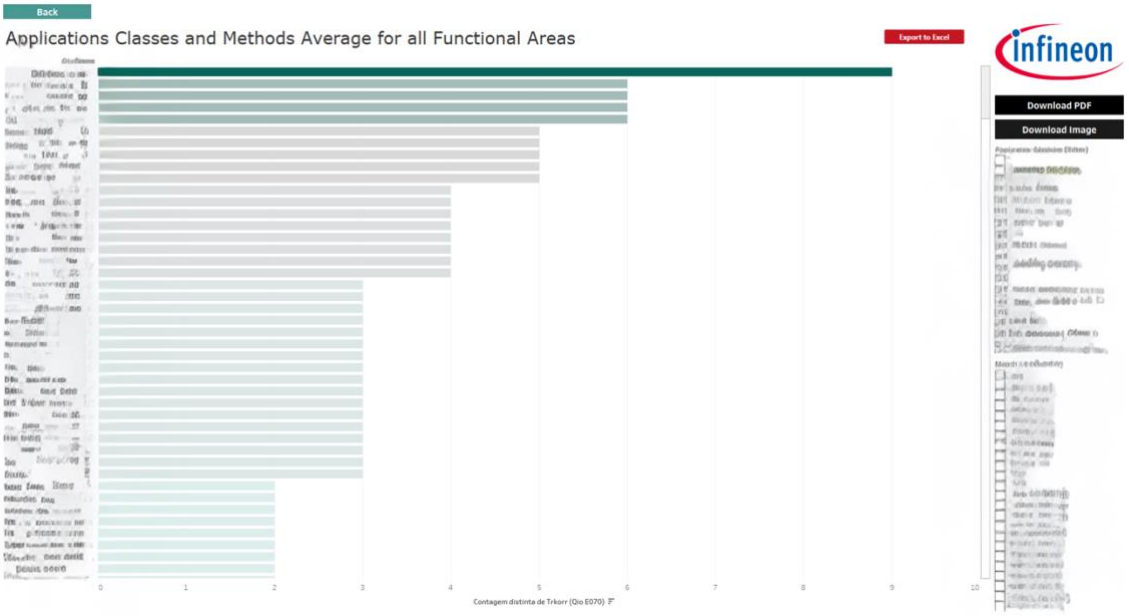


Figura 23 - Dashboard com os programas com mais alterações nos últimos meses

5.6.4 Hospedagem

Os *dashboards* foram hospedados no servidor da empresa designado para essa finalidade. Para viabilizar a limpeza e a previsão dos dados diretamente no *Tableau*, a extensão *Python* foi adicionada. Essa extensão permitiu a execução de scripts Python na ferramenta *Tableau*, possibilitando a integração dos resultados obtidos anteriormente com as funcionalidades do *dashboard*.

5.6.5 Uso da Ferramenta Tableau

É relevante mencionar que durante o processo, houve a necessidade de aprendizagem e familiarização com a ferramenta *Tableau*. Dessa forma, foi possível explorar todas as funcionalidades disponíveis, aprimorar as habilidades na criação de *dashboards* e obter uma compreensão mais aprofundada do seu potencial para análise de dados. Essa capacidade de adaptação e aquisição de novas habilidades demonstram a sua dedicação em enfrentar os desafios do projeto de forma eficiente.

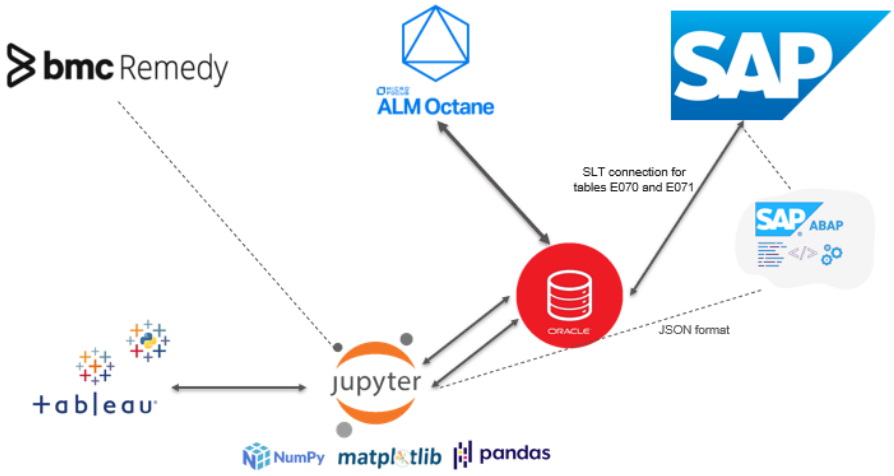


Figura 24 - Esquema da organização final

6. Trabalho Futuro

Neste capítulo, discutiremos os próximos passos após a conclusão deste projeto, que incluem a implementação do dashboard desenvolvido e o plano para o seu uso contínuo na Infineon Technologies AG. Abordaremos também a importância da colaboração com os programadores e a criação de uma equipa de suporte para garantir que o dashboard atenda às necessidades em constante evolução da empresa.

6.1 Implementação do Dashboard em Produção

Uma etapa fundamental após o desenvolvimento do dashboard é a sua implementação em ambiente de produção. Isso envolve a transição do sistema do ambiente de desenvolvimento ou teste para o ambiente operacional da empresa. Dado que o dashboard foi projetado para ser executado na plataforma Tableau em nuvem, essa transição deve ser facilitada, pois a infraestrutura necessária já está em vigor.

6.2 Uso pelos Programadores e Recolha de Feedback

O dashboard será disponibilizado aos programadores da Infineon Technologies AG, que serão os principais utilizadores finais. Eles serão incentivados a começar a utilizar o dashboard no seu trabalho diário de gestão de desenvolvimentos SAP ABAP. A recolha de feedback dos programadores é uma parte crítica deste processo, uma vez que suas sugestões e observações serão valiosas para a melhoria contínua do dashboard.

6.3 Revisão e Ajustes

Com base no feedback dos programadores, serão realizadas revisões e ajustes no dashboard, conforme necessário. Isso pode incluir otimizações de desempenho, melhorias na interface do utilizador, adição de funcionalidades adicionais ou aprimoramentos na precisão das previsões. A colaboração próxima com os programadores será essencial para garantir que o dashboard atenda às suas expectativas e necessidades.

6.4 Definição de uma Equipa de Suporte

Para garantir o bom funcionamento contínuo do dashboard e fornecer assistência aos utilizadores, será estabelecida uma equipa de suporte dedicada. Esta equipa será composta por especialistas que possuam um profundo conhecimento do dashboard e das ferramentas subjacentes. Além disso, eles serão responsáveis por manter a documentação atualizada e fornecer assistência técnica quando necessário.

6.5 Documentação e Treino

Será desenvolvida documentação detalhada sobre o uso do dashboard, incluindo guias de utilizador e manuais de referência. Além disso, serão oferecidos programas de formação aos programadores e outros utilizadores-chave para garantir que possam aproveitar ao máximo as capacidades do dashboard.

6.6 Gestão de Acessos e Segurança

A gestão adequada dos acessos ao dashboard é fundamental para proteger os dados e garantir que apenas utilizadores autorizados tenham acesso às informações sensíveis. Será implementado um sistema de gestão de acessos e segurança para controlar quem pode ver e interagir com o dashboard.

6.7 Monitorização e Melhoria Contínua

Após a implementação em produção, o dashboard será continuamente monitorizado para garantir o seu desempenho e utilidade. Serão definidos indicadores-chave de desempenho para avaliar a eficácia do dashboard na gestão de desenvolvimentos SAP ABAP. Com base nas métricas recolhidas, serão realizadas melhorias contínuas para garantir que o dashboard continue a atender às necessidades da *Infineon Technologies AG*.

6.8 Conclusão

O futuro do dashboard desenvolvido neste projeto envolve uma abordagem colaborativa e orientada para a melhoria contínua. Com a participação ativa dos programadores e uma equipa de suporte dedicada, a Infineon Technologies AG estará bem posicionada para otimizar a gestão dos seus desenvolvimentos baseados em SAP ABAP e tomar

decisões informadas com base em dados sólidos e análises precisas. A implementação em ambiente de nuvem facilitará todo o processo, tornando-o ágil e escalável para atender às necessidades em constante evolução da empresa.

7. Conclusões e Considerações Finais

Este trabalho teve como objetivo principal analisar e desenvolver um *dashboard* para avaliar a importância de programas baseados em SAP ABAP *na Infineon Technologies AG*. O processo de investigação e desenvolvimento levou em consideração as ferramentas e processos existentes na empresa, com foco na gestão dos incidentes, análise das causas fundamentais e identificação dos problemas recorrentes mais impactantes.

Ao longo deste estudo, foi possível observar a complexidade inerente à gestão dos desenvolvimentos SAP ABAP, com a interação entre as diferentes ferramentas, processos e equipas desempenhando um papel crítico no resultado final. As análises detalhadas das métricas de importância, esforço e prioridade permitiram uma compreensão mais clara dos fatores que influenciam a necessidade de reavaliação ou reformulação de um programa.

A criação e implementação do *dashboard* final proporcionou uma plataforma interativa e visualmente informativa para a equipa de TI. Os modelos de previsão incorporados no *dashboard* oferecem uma maneira eficaz de calcular a importância dos programas, permitindo avaliações objetivas e baseadas em dados. A inclusão de gráficos detalhados destacando áreas problemáticas específicas, bem como as tendências temporais, proporciona uma visão abrangente dos desenvolvimentos, permitindo tomadas de decisão mais assertivas.

Ao explorar a relação entre as ferramentas utilizadas, os processos internos e as métricas analisadas, tornou-se evidente que a combinação de abordagens analíticas e tecnológicas pode resultar em melhorias significativas na gestão de desenvolvimentos SAP ABAP. A identificação precoce de problemas recorrentes e a capacidade de avaliar a importância dos programas de forma objetiva contribuem para a eficiência operacional e a excelência dos resultados finais.

Este trabalho também enfatiza a importância da aprendizagem e adaptação ao longo do processo. A necessidade de aprender a usar novas ferramentas e tecnologias, como a plataforma *Tableau*, para desenvolver o *dashboard* final, demonstrou a importância de

permanecer flexível e disposto a adquirir novas habilidades para atingir os objetivos propostos.

Em suma, este estudo oferece um olhar aprofundado sobre a gestão de desenvolvimentos SAP ABAP na *Infineon Technologies AG*, evidenciando a relevância de uma abordagem analítica, integrada e baseada em dados para otimizar os processos internos. O dashboard desenvolvido e as análises realizadas têm o potencial de impactar positivamente a eficiência, a qualidade e a tomada de decisões na empresa, contribuindo para um ambiente de trabalho mais eficaz e uma entrega de resultados excecionais.

Referências Bibliográficas

[1] R. Zaidi, The Ultimate SAP(R) User Guide, eCruiting Alternatives, Incorporated, 2015.

[2] BMC, "BMC Helix ITSM is the next generation of remedy," BMC, [Online]. Available: https://www.bmc.com/it-solutions/remedy-itsm.html. [Acedido em 04 2023].

[3] BMC, Remedy IT Service Management Suite 9.1, BMC Software , 2010.

[4] A. Gramm, IT Service Management with ITIL and Remedy, Packt Publishing, 2011.

[5] D. G. W. R. B. v. V. Rik Marselis, Quality for DevOps Teams, Sogetibooks, 2020.

[6] K. B. Hansen, Practical Oracle SQL: Mastering the Full Power of Oracle Database, Apress, 2020.

[7] U. Priyadarshiny, "What is Tableau?," Edureka, 2017. [Online]. Available: https://medium.com/edureka/what-is-tableau-1d9f4c641601. [Acedido em 04 2023].

[8] RED HAT, "RED HAT BLOG," RED HAT BLOG, 02 2023. [Online]. Available: https://www.redhat.com/en/blog.

[9] S. Brown, "Machine learning, explained," MIT, 04 2021. [Online]. Available: https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained. [Acedido em 05 2023].

[10] S. Liu, Visual Informatics, science direct, 2017.

[11] Y. Dodge, The Concise Encyclopedia of Statistics, Springer-Verlag, 2008.

[12] I. G. e. Y. Bengio, Deep Learning, MIT, 2015.

[13] A. Burkov, The Hundred-Page Machine Learning Book, Andriy Burkov, 2019.

[14] O. Theobald, Machine Learning For Absolute Beginners, Oliver Theobald, 2018.

[15] S. G. Andreas Müller, Introduction to Machine Learning with Python, OREILLY, 2016.

[16] D. N. S. N. D. S. F. S. Dr Zul Ariffin Maarof, Implementation of Genetic Algorithm in Model Identification, Paperback , 2019.

[17] G. Eichhorn, "Predict IT Support Tickets with Machine Learning and NLP," *Towards Data Science,* 2020.

[18] H. B. Feras Al-hawari, "A Machine Learning Based Help Desk System for IT Service Management," *Journal of King Saud University - Computer and Information Sciences,* 2019.

[19] G. K. Gábor Békés, Data Analysis for Business, Economics, and Policy, Cambridge University Press, 2021.

[20] H. B. Feras Al-Hawari, "A machine learning based help desk system for IT service management," *Journal of King Saud University - Computer and Information Sciences,* 2021.

[21] C. D. W. Simon Fuchs, "Improving Support Ticket Systems Using Machine Learning:," *HICSS,* 2022.