**CENTRO PAULA SOUZA**

**FACULDADE DE TECNOLOGIA DE MOGI MIRIM**

**Carlos Roberto de Almeida Babosa**

**Danilo Lisboa Costa da Rocha**

**MINERAÇÃO DOS DADOS DOS TRABALHOS CIENTÍFICOS DA FATEC MOGI MIRIM**

**MOGI MIRIM**

**2022**

**Carlos Roberto de Almeida Babosa**

**Danilo Lisboa Costa da Rocha**

**MINERAÇÃO DOS DADOS DOS TRABALHOS CIENTÍFICOS DA FATEC MOGI MIRIM**

Trabalho de Graduação apresentado ao Curso de Tecnologia em Análise e Desenvolvimento de Sistemas da Faculdade de Tecnologia de Mogi Mirim como pré-requisito para a obtenção do Título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

**Orientadora:** Prof. Ma. Rita de Cássia Catini

**MOGI MIRIM**

**2022**

FOLHA DE APROVAÇÃO

Será entregue pela coordenação

**LISTA DE FIGURAS OU ILUSTAÇÕES**

[**Figura 1 -** Exemplo de Figura 1 17](#_Toc82636016)

[**Figura 2 -** Exemplo de Figura 2 19](#_Toc82636017)

[**Figura 3 -** Exemplo de Figura 3 20](#_Toc82636018)

[**Figura 4** - Exemplo de Figura 3 22](#_Toc82636019)

[**Figura 5** - Exemplo de Figura 4 23](#_Toc82636020)

**LISTA DE TABELAS**

[**Tabela 1** - Tipos de XXXXXXXXX 23](#_Toc40696614)

[**Tabela 2** - Tipos de XXXXXXXXX 24](#_Toc40696615)

**LISTA DE ABREVIATURAS E SIGLAS**

CSS Cascading Style Sheets

HTML HyperText Markup Language

SQL Structured Query Language

TCC Sistema Gerenciador de Banco de Dados

TG Trabalho de Graduação

**RESUMO**

Elemento obrigatório. Destina-se a uma visão clara e rápida do conteúdo e da conclusão do trabalho, não podendo ultrapassar 500 palavras. O resumo deve conter introdução, metodologia, resultados e conclusões (o "o quê?", o "por quê?", o "como?" e o "para quê?" do trabalho monográfico). Trata-se da parte do trabalho que traz as informações essenciais acerca da monografia, sendo formatado em monobloco, pontuado, mas, sem divisão em parágrafos. Depois de digitado o texto do resumo, deixar uma linha em branco e inserir as palavras-chave (elemento obrigatório, em negrito), ou seja, palavras que identificam a essência do trabalho para indexação, por assunto, nas bibliotecas.

**Palavras-chave:** Xxx; Xxx; Xxx; Xxx; Xxx (colocar em ordem alfabética com no mínimo 3 e no máximo 5 palavras, separadas por ;)

***ABSTRACT***

*Elemento obrigatório. Destina-se a uma visão clara e rápida do conteúdo e da conclusão do trabalho, não podendo ultrapassar 500 palavras. O resumo deve conter introdução, metodologia, resultados e conclusões (o "o quê?", o "por quê?", o "como?" e o "para quê?" do trabalho monográfico). Trata-se da parte do trabalho que traz as informações essenciais acerca da monografia, sendo formatado em monobloco, pontuado, mas, sem divisão em parágrafos. Depois de digitado o texto do resumo, deixar uma linha em branco e inserir as palavras-chave (elemento obrigatório, em negrito), ou seja, palavras que identificam a essência do trabalho para indexação, por assunto, nas bibliotecas.*

***Palavras-chave:*** *Xxx; Xxx; Xxx; Xxx; Xxx (colocar em ordem alfabética com no mínimo 3 e no máximo 5 palavras, separadas por ;)*

SUMÁRIO

[1 INTRODUÇÃO 12](#_Toc119924119)

[1.1 Estrutura do Trabalho 13](#_Toc119924120)

[2 FUNDAMENTAÇÃO TEÓRICA 14](#_Toc119924121)

[2.1 Mineração de dados 14](#_Toc119924122)

[2.1.1 O que é mineração de dados 15](#_Toc119924123)

[2.1.2 Sistemas de Recomendação Educacionais 16](#_Toc119924124)

[2.1.3 Processo de Descoberta de Conhecimento em Textos 17](#_Toc119924125)

[2.1.4 Coleta 18](#_Toc119924126)

[2.1.5 Mineração 20](#_Toc119924127)

[2.1.6 Pós-processamento 21](#_Toc119924128)

[2.1.7 Método de análise dos resultados 21](#_Toc119924129)

[2.2 Sistemas online 23](#_Toc119924130)

[2.2.1 HTML 23](#_Toc119924131)

[2.2.2 CSS 24](#_Toc119924132)

[2.2.3 Frameworks 25](#_Toc119924133)

[2.3 Linguagens, banco de dados e hospedagem 27](#_Toc119924134)

[2.3.1 PhYton 28](#_Toc119924135)

[2.3.2 Javascript 30](#_Toc119924136)

[2.3.3 MySQL 31](#_Toc119924137)

[3 ESTUDO DE CASO 35](#_Toc119924138)

[3.1 Levantamento de Requisitos 35](#_Toc119924139)

[3.2 Proposta de Solução 35](#_Toc119924140)

[4 Considerações finais 36](#_Toc119924141)

[**REFERÊNCIAS BIBLIOGRÁFICAS** 37](#_Toc119924142)

# INTRODUÇÃO

De acordo com o Ministério da Educação a conclusão dos cursos de graduação no Brasil requerem a elaboração de um Trabalho de Conclusão de Curso (TCC) ou também chamados de Trabalho de Graduação (TG), no qual o aluno coloca em prática os conhecimentos adquiridos no decorrer do curso. No caso das Faculdades de Tecnologia do estado de São Paulo os trabalhos precisam ter cunho inédito ou resolverem um problema específico de um cliente real (PAZ; CAZELLA, 2018).

Um sistema de informação que contenha os trabalhos anteriores catalogados auxilia para verificar o ineditismo e dar boas ideias aos alunos sobre como abordar problemas reais. Diante disto, a catalogação automatizada dos trabalhos já realizados organiza o conhecimento produzido e dá margem para que os novos alunos tenham ideias inéditas ou indícios para resolver casos reais (MASCHIO et al., 2018).

A Faculdade de Tecnologia de Mogi Mirim Arthur Azevedo não possui um sistema que consiga catalogar e disponibilizar os principais aspectos dos TG’s elaborados por seus alunos. A consulta dos trabalhos acadêmicos feitos na instituição de ensino é de difícil acesso por não disponibilizarem uma forma de consulta digital.

A informação é o elemento central na procura do conhecimento. As tecnologias da informação se tornaram essenciais para quase todos os aspectos da vida (SIONARA; MARCACINI; REZENDE, 2021). Ao se utilizar uma ferramenta inteligente para mineração dos dados dos trabalhos científicos da Fatec Mogi Mirim, é possível realizar a consulta do tema utilizado e confrontar se já existem na base de dados cadastrada realizada com trabalhos já entregues de outros semestres. Com isso, tem-se a seguinte problemática: como a extração de dados e categorização dos trabalhos científicos da Fatec Mogi Mirim pode ser realizada por meio da mineração de dados?

O objetivo geral deste trabalho é desenvolver uma ferramenta de mineração de dados que permita a seleção e consulta futura dos dados extraídos e categorizados de trabalhos de graduação. Já seus objetivos específicos são os de a) compreender a estrutura e modelos de extração e categorização por meio da mineração de dados; b) programar um algoritmo computacional que extraia e categorize os principais dados dos artigos científicos disponibilizados pela Fatec Mogi Mirim; c) catalogar os dados extraídos em banco de dados digital; d) disponibilizar em plataforma online os dados extraídos dos trabalhos para consulta da comunidade acadêmica.

Catalogar estes dados facilitaria a pesquisa de trabalhos realizados para se obter os dados e validar se correspondem com às normas da instituição, se então conforme o padrão, se seguem a estrutura proposta pelo orientador, além de validar o seu ineditismo (PAZ; CAZELLA, 2018). A instituição de ensino, mais precisamente o orientador, precisam ter acesso a essas informações para validarem se o tema já existe em outros trabalhos já finalizados e se a instituição possui base científica de apoio para facilitar a validação do tema e realizar futuras correções nos trabalhos que estão sendo produzidos

Devido à dificuldade em se conseguir validar o tema, estrutura e itens obrigatórios de trabalhos acadêmicos, esta pesquisa se justifica pela necessidade de se ter uma ferramenta que extraia os dados mais importantes como tema, problema, objetivos e conclusão dos trabalhos.

A metodologia para o desenvolvimento será uma Revisão de Literatura, buscando nas bases de dados online das bibliotecas públicas digitais de artigos científicos publicados nos últimos 5 anos. Também, será feito um estudo de caso na Faculdade de Tecnologia de Mogi Mirim, com a análise dos TG’s que estejam em meio digital, utilizando-se as tecnologias de linguagem de programação e interpretação *phyton, javascritp, HyperText Markup Language (html) e Cascading Style Sheets (css)* e do banco de dados *Structured Query Language (SQL).*

## Estrutura do Trabalho

O presente trabalho está estruturado da seguinte forma: inicialmente, é apresentada uma breve introdução sobre a mineração de dados, seguida de uma explicação sobre a estrutura do sistema online desenvolvido para disponibilização dos resultados da mineração (GIL, 2018). Em seguida, é apresentado o algoritmo utilizado para a mineração, bem como as bibliotecas necessárias para sua execução. Por fim, são apresentados os resultados e a avaliação da qualidade dos dados retornados pela mineração.

# FUNDAMENTAÇÃO TEÓRICA

## Mineração de dados

A mineração de dados envolve a análise de grandes conjuntos de dados para “descobrir novas informações”. Este processo é conhecido como Knowledge Discovery in Databases ou KDD (AGUIAR et al., 2015). O KDD se concentra na descoberta de “novas” informações por meio da análise de grandes conjuntos de dados, com um total de ou mais pontos de dados. Este novo conhecimento pode levar a novas descobertas científicas (SILVA et al., 2016).

A mineração de dados ajuda os cientistas a investigar questões relacionadas a áreas diversas, aplicando-se, também, à esfera da educação. Também é utilizada por vendedores, bioinformáticos e cientistas de dados para encontrar informações (AGUIAR; MAGALHÃES, 2013). A mineração de dados tornou-se popular devido à sua capacidade de encontrar novas informações e respostas. Diferentes mineradores de dados a utilizam para diferentes propósitos (BORTOLOSSI, 2012).

Alguns o buscam encontrar respostas relacionadas a vendas, bioinformática ou sistematização da informação. No âmbito da educação, a mineração de dados permite a identificação de fatores que influenciam a aprendizagem, bem como a implementação dos melhores sistemas educacionais (CRUZ et al., 2018). Dentre as principais técnicas desse campo, há a “raspagem de dados”. À medida que os dados se tornam mais padronizados, as técnicas de raspagem se tornam mais comuns e fáceis de serem aplicadas (SILVA; VIERA, 2021).

Os dados podem ser facilmente copiados se forem organizados em tabelas com HTML. Extrair informações das tabelas é fácil devido ao carregamento lento das páginas dinâmicas. O aspecto mais difícil da coleta de dados é lidar com dados não padronizados (GOMES; PIMENTA; SCHNEIDER, 2019). Simular movimentos do mouse é a única maneira de proceder ao processar solicitações *REST*. Fazer isso requer o envio de filtros de pesquisa de forma assíncrona e o movimento automático do mouse para a página apropriada (SILVA et al., 2016).

O *Selenium WebDriver* é uma estrutura projetada para testes de sistema automatizados e extração de dados. Também é capaz de simular as ações de usuários reais para suporte no envio e recebimento de informações de forma assíncrona. Isso o torna a escolha ideal para realizar testes de sistema automatizados em páginas da web geradas dinamicamente, como verificação de estado, comparação de resultados e execução de fluxos de trabalho.

Ele também suporta a automação de testes em diferentes navegadores, como o *Google Chrome*, o *Internet Explorer* e o *Firefox*. Diante das contribuições da mineração de dados, neste capítulo, apresenta-se as suas principais características e técnicas específicas.

*Datamining*, ou mineração de dados, corresponde ao ato de encontrar padrões em grandes conjuntos de dados. Isso pode ser feito automaticamente ou com um processo parcialmente manual (AGUIAR et al., 2015). Os dados devem ter algum mérito para serem minerados; devem revelar algum insight que leve a uma conclusão significativa (BAKER; ISOTANI; CARVALHO, 2011). É possível prever novos dados usando padrões específicos, utilizando-se de técnicas igualmente específicas para que se obtenha um bom desempenho.

Os dados extraídos podem ser usados para criar estruturas que podem ser examinadas, testadas e reutilizadas no futuro. À medida que o volume de informação científica aumenta, surgem novas demandas e necessidades (BLUM; MERINO; MERINO, 2016). A maioria dos dados científicos é liberada a partir de arquivos digitais nos formatos PDF ou HTML. Esses dados podem ser significativamente grandes e difíceis de processar, levando a novos desafios e necessidades de soluções (CORRÊA; MARCACINI; REZENDE, 2012).

Para extrair adequadamente informações de documentos digitais, os pesquisadores dispendem uma quantidade significativa de tempo lendo seu conteúdo (FERREIRA; CORREA, 2021). Devido ao número substancial de publicações digitais e conjuntos de dados disponíveis, precisam desenvolver métodos mais rápidos e precisos para encontrar informações (GOMES; PIMENTA; SCHNEIDER, 2019). Esses métodos se tornam úteis ao permitirem a procura por informações em fontes de linguagem naturalística.

A mineração de texto envolve a pesquisa de textos para encontrar padrões específicos (GUEDES, 2018). Ademais, este campo também envolve a busca de padrões nos dados e a descoberta de novas informações. Trata-se, portando, de um processo de descoberta por informações úteis por meio da análise de dados (SANTOS, 2011). Isso pode ser feito a partir da procura por nexos de informações em fontes de dados publicadas que geralmente são ignoradas (PAZ; CAZELLA, 2018).

Dessa forma, a utilização de técnicas de mineração de dados permite a descoberta de conhecimentos inéditos, pois a mineração permite, ao pesquisador, a extração de implicações e ideias anteriormente desconhecidas.

### Sistemas de Recomendação Educacionais

Os Sistemas de Recomendação Educacional (SRE) são sistemas online que ajudam professores e alunos a encontrar recursos específicos. Eles podem ser utilizados para filtrar os recursos educacionais que pertencem a um determinado aluno, professor ou outro agente educacional (LIU et al., 2013). Em geral, esses sistemas são usados para acessar informações em um momento em que há uma ampla quantidade de dados disponível (RIGO et al., 2014). Os usuários precisam estar satisfeitos com os resultados da recomendação para que o sistema atinja seu objetivo.

As taxas de satisfação são medidas por quantas vezes as pessoas aceitam ou rejeitam as recomendações dadas pelo SRE. As métricas para medir a precisão dos dados podem ser encontradas por meio de métodos estatísticos ou por meio do uso de suporte à decisão (PAZ; CAZELLA, 2018). O método de suporte à decisão mais comum é usar o Erro Absoluto Médio, ou MAE, empregado para prever o desempenho de um SR específico (FULANTELLI et al., 2015).

Outras métricas incluem o *Root Mean Square Error*, ou RMSE, que compara as previsões estimadas com as avaliações reais, e o *Average Percent Error*, ou APE. Essas métricas são usadas lado a lado por muitos usuários do Sistema de Recomendação (SR) tradicional (COSTA; AGUIAR; MAGALHÃES, 2013). Um aspecto que diferencia o SRE de outros SR é a falta de métodos de avaliação padronizados. Também não há um conjunto definido de dados de referência ao avaliar as métricas de suporte (BAKER et al., 2011).

Outra diferença significativa entre SRE e outros SR é o número de dimensões possíveis que podem ser consideradas ao avaliar questões pedagógicas (PAIVA et al., 2013). Os usuários também utilizam as métricas Measure F, Precision e Recall como ferramentas de suporte à decisão (FRANÇA; AMARAL, 2013). Os sistemas que recomendam materiais educacionais são chamados de Sistemas Educacionais de Recomendação ou SRE.

Eles geralmente são destinados ao uso por alunos ou professores, mas também podem filtrar informações educacionais e personalizar o conteúdo para melhor atender às necessidades dos alunos (SIEMENS; LONG, 2011). Ao criar um modelo e examinar seus dados, esses sistemas são capazes, além de filtrar recursos, de fornecer sugestões, customizar conteúdo e muito mais (JOHNSON et al., 2011). Eles existem para proporcionar aos alunos uma melhor experiência de aprendizado – acredita-se que a educação é o centro do processo do sistema.

A mineração de dados educacionais é uma abordagem que usa dados coletados em ambientes educacionais para encontrar possíveis problemas, como problemas com o ensino ou o progresso acadêmico dos alunos (RIGO et al., 2014). Tem grande potencial para melhorar as recomendações pedagógicas. *Educational Analytics* (EA), por sua vez, é uma abordagem para medir e analisar dados produzidos em ambientes educacionais (PAZ; CAZELLA, 2018).

Isso dá aos educadores a capacidade de prever o futuro, bem como avaliar o progresso acadêmico e identificar possíveis problemas. *Learning Analytics* (LA), por fim, é uma técnica para analisar dados coletados em instituições de ensino, coletando, processando e interpretando-os (PAZ; CAZELLA, 2018). Isso permite que os educadores prevejam e avaliem os resultados acadêmicos futuros e identifiquem possíveis problemas (FULANTELLI et al., 2015).

Além disso, pode-se utilizar o *Learning Analytics*, ou LA, pois perite monitoramento e visualização de vários aspectos do ensino e aprendizagem. Alguns estudos usaram *Learning Analytics*, também conhecido como análise automática de dados. Essas são as principais ferramentas utilizadas para manter os Sistemas de Recomendação Educacionais assertivos.

### Processo de Descoberta de Conhecimento em Textos

A descoberta de conhecimento depende da utilização de técnicas de extração e análise para encontrar padrões nos dados. Emprega-se esta metodologia para encontrar informações ou padrões de informação em documentos textuais, bem como dados não organizados. Os dados textuais são considerados uma das fontes de conhecimento mais importantes. Os dados não estruturados requerem um processo de descoberta de conhecimento diferente dos documentos textuais (ZHAI; MASSUNG, 2016).

Os dados devem ser pré-processados, coletados, minerados e processados antes que novas descobertas possam ser feitas e validadas por outros estudos. Conjuntos de dados que são muito grandes e difíceis de serem examinados por qualquer indivíduo representam um desafio de pesquisa significativo. Quando os processos KDD são executados, requerem várias etapas. Uma dessas etapas é a Mineração de Dados, que deve ser realizada antes que qualquer algoritmo seja empregado (SILVA; PERES; BOSCARIOLI, 2017).

Outra etapa normalmente realizada em processos KDD é o pré-processamento. Após a mineração de dados, o processamento adicional é necessário para avaliar adequadamente o conhecimento descoberto. Isso ocorre porque a mineração de dados nem sempre precisa de pós-processamento. No entanto, isso não significa que o pós-processamento seja desnecessário. De fato, muitas vezes é necessário garantir que o conhecimento descoberto tenha os resultados desejados (SILVA; VIERA, 2021).

Para alcançar resultados, um processo de KDD precisa de um elemento interativo que permita a progressão do processo geral. É por isso que os termos inicial e iterativo descrevem a análise e interpretação dos fatos realizada pelo homem. Ao repetir os processos de KDD de forma iterativa, os pesquisadores podem chegar a uma solução com refinamentos sucessivos. A dificuldade em realizar operações de KDD fica evidente em uma grande e diversificada quantidade de dados, bem como a necessidade de tratar os resultados em diferentes formatos (ZHAI; MASSUNG, 2016).

Alguns exemplos de fatores operacionais são a manipulação de grandes conjuntos de dados e o tratamento de resultados em diferentes formatos. Existem muitas ferramentas que simplificam as operações de KDD, fornecendo ambientes integrados que minimizam as dificuldades impostas por esses fatores.

### Coleta

Antes que os dados possam ser analisados, eles precisam ser preparados. Isso envolve a remoção de qualquer informação estranha dos dados brutos, incluindo erros ortográficos e aspectos morfológicos incorretos. Além disso, os dados devem ser normalizados e analisados quanto a diferenças semânticas. Uma vez concluído esse processo, as próximas etapas envolvem a aplicação de algoritmos específicos para coletar dados (KRASSMANN et al., 2017).

A fase final do exame envolve a interpretação dos resultados. Os algoritmos de mineração de dados dependem de dados preparados, e é por isso que esta etapa vem antes deles. A seleção de dados é uma das principais etapas do pré-processamento. Outras funções incluem binarização e fragmentação de dados para processamento. Essa função do processo KDD visa determinar quais dados devem ser considerados ao coletar informações para uma determinada análise KDD (Parreño et al., 2016).

Após esta primeira etapa, os dados são limpos para remover imprecisões, impurezas e quaisquer outras alterações feitas nos mesmos. Esta segunda fase também inclui qualquer tratamento de dados necessário para garantir a qualidade, exaustividade e veracidade da informação representada. As inconsistências do banco de dados devem ser corrigidas antes da extração dos modelos de conhecimento para o processo KDD (SCHIESSL; BRASCHER, 2011).

Isso evita que informações imprecisas sejam incluídas nos algoritmos de mineração de dados finais, que são usados para coletar modelos de conhecimento (PAZ; CAZELLA, 2018). A codificação de dados representa a terceira etapa. Ela deve acontecer antes que qualquer algoritmo seja executado nos dados. Os dados podem ser transformados em diferentes intervalos ou categorias usando codificação categórica (SILVA; VIERA, 2021).

Alternativamente, a codificação numérica é usada para representar os valores de atributos categóricos. A codificação é então aprimorada com uma quarta etapa, que geralmente é omitida da pesquisa acadêmica (SILVA; VIERA, 2021). O enriquecimento de dados é o processo de adicionar novas informações aos registros existentes (KRASSMANN et al., 2017). É a última etapa do pré-processamento dos dados.

Ele aumenta a densidade de informações, tornando mais fácil para um algoritmo de descoberta de conhecimento encontrar padrões. Isso pode ser feito por meio de pesquisas que combinam com os dados, consultas a bancos de dados externos e outros métodos.

### Mineração

Ao optar pela abordagem de mineração de dados, as equipes de KDD precisam considerar o problema que desejam resolver. Essa decisão geralmente depende do tipo de tarefa KDD que eles planejam executar (AGUIAR et al., 2015). Técnicas comuns de mineração de dados são algoritmos genéticos, modelos estatísticos e probabilísticos e redes neurais. A primeira etapa é a realização de associações a partir dos dados relacionados e envolve a pesquisa em entradas de banco de dados comuns (AGUIAR; MAGALHÃES, 2013).

O próximo passo é classificação dos dados. É uma função que converte dados registrados em valores reais. A terceira tarefa, regressão, envolve apenas o cálculo de propriedades numéricas (BLUM; MERINO; MERINO, 2016). Esta terceira tarefa de recuperação de dados requer encontrar uma função que mapeie os registros do banco de dados em seus equivalentes do mundo real (BAKER; ISOTANI; CARVALHO, 2011). Esta tarefa é semelhante à de classificação, pois trata apenas de atributos numéricos.

As redes neurais fornecem ferramentas para executar tarefas de regressão com eficácia. Também realizam agrupamento, que é o processo de separar os bancos de dados em grupos que possuem propriedades semelhantes, mantendo esses aspectos exclusivas entre cada grupo (BORTOLOSSI, 2012). Ao executar uma tarefa de agrupamento, algoritmos automáticos como *KMeans, K-Modes, K-Prototypes e Kohonen* podem ser usados para classificar os dados (CORRÊA; MARCACINI; REZENDE, 2012).

Eles dão forma à etapa da clusterização. Esses algoritmos devem classificar os dados entre clusters enquanto maximizam as semelhanças intracluster (CRUZ et al., 2018). O objetivo desta tarefa é facilitar o processo de classificação, eliminando a necessidade de rótulos pré-determinados. A sumarização é a etapa seguinte (FERREIRA; CORREA, 2021). Requer a identificação e transmissão de tendências comuns entre conjuntos de dados.

Muitos algoritmos KDD se utilizam de algoritmos genéticos e lógica indutiva para lidar com essa tarefa comum. Por fim, na etapa da detecção de desvios, os outliers devem ser identificados; são registros de banco de dados que não seguem os padrões estabelecidos (GUEDES, 2018). Isso ocorre porque outliers desafiam as normas estabelecidas dos dados que estão sendo analisados. As estatísticas fornecem informações que ajudam a realizar essa tarefa (GOMES; PIMENTA; SCHNEIDER, 2019).

Na última etapa é feita a descoberta de sequências. É uma fase semelhante à classificatória, uma vez que também realiza associações. O objetivo é o de detectar itens frequentes em uma transação ao longo de um período específico. Essas são as principais tarefas da mineração de dados indicadas para que se obtenha dados seguros e assertivos.

### Pós-processamento

Executadas todas as etapas apresentadas no tópico 4.1.5, o profissional poderá partir para a fase do pós-processamento dos dados. Esta última fase engloba os mecanismos voltados ao tratamento de todo o conhecimento obtido com a aplicação das demais técnicas de mineração de dados. O principal objetivo é o de facilitar o processo de interpretação e avaliação da qualidade dos dados, ou seja, busca-se atestar a qualidade do conhecimento descoberto (PAZ; CAZELLA, 2018).

Dentre as principais atividades a serem realizadas nesta fase final, menciona-se a elaboração e organização da análise. Pode-se utilizar gráficos, diagramas ou relatórios com recursos textuais e visuais para apresentar os resultados de maneira simplificada. Intenta-se que esses dados possam ser úteis e eficazes e, para isto, devem ser representados de maneira textual e visual para que se tome ciência das etapas perseguidas que atestam a devida qualidade (SILVA; VIERA, 2021).

### Método de análise dos resultados

Antes de iniciar um processo de análise de texto, os analistas de dados primeiro coletam textos e documentos. Em seguida, é feita a “limpeza da linguagem”, no qual os elementos desnecessários são removidos para revelar a estrutura subjacente da linguagem (YANG; MANOHARAN; BARBER, 2014). Os métodos de pré-processamento são utilizados antes das técnicas de mineração. A dobragem de maiúsculas e minúsculas é uma técnica usada para combinar *strings* que envolvem a conversão de todas as palavras em maiúsculas ou minúsculas (ORENGO; HUYCK, 2001).

Variações de ênfase devem ser consideradas ao analisar as palavras. Preparações adicionais são necessárias porque algumas palavras podem ser acentuadas enquanto outras não (ZHAI; MASSUNG, 2016). Suplementos são necessários porque os acentos podem ser removidos para dar conta dos casos em que algumas formas da palavra são acentuadas enquanto outras não. É comum usar a primeira etapa de criação de um índice em qualquer tipo de dados de texto, transformando caracteres em palavras ou tokens (WEISS et al., 2010).

Além disso, recursos como a remoção de *stopwords* geralmente são implementados nesta etapa. Baeza-Yates e Ribeiro-Neto (2013) apontam que os termos que aparecem com frequência representam 80% dos documentos. Essas palavras não ajudam na recuperação. Em vez disso, reduz o termo à sua raiz para obter resultados mais eficazes – isso é chamado de lematização. Palavras citadas em documentos são analisadas durante a lematização. Esse processo remove os sufixos das palavras para revelar suas formas de raiz. Um método comum usado durante este processo é o *Bag of Words*, ou BoW.

Meena e Lawrance (2019) observam que o saco de palavras constrói um vocabulário a partir do documento fornecido. Um modelo de ponderação usado em vários estudos acadêmicos é o TF-IDF, que leva dois cálculos para criar — o IDF (Frequência de Documento Inverso) e TF (Frequência de Termo). Ambos medem a frequência com que vários termos aparecem em um documento. Este modelo foi descoberto em vários estudos graças aos seus resultados positivos de extração de recursos. Nesta fase inicial do processamento, os n-gramas são usados para extrair dois ou três termos próximos com uma frequência específica.

Alguns estudos também creditam aos métodos *n-gram* benefícios como agilidade no processamento de dados, independentemente do idioma e arquivos com erros linguísticos e ruído (COSTANTINO et al., 2017). Para a apresentação dos resultados, a nuvem de palavras é uma forma simples de visualizar conceitos. O profissional que analisa os dados consegue entender conceitos mais importantes em um nível superior. Quanto mais termos significativos em um texto, maior se torna a nuvem de palavras ao exibir os resultados (KUANG; BRANTINGHAM; BERTOZZI, 2017).

Na próxima etapa, o agrupamento de texto é usado para agregar conjuntos de dados. Algoritmos de aprendizado de máquina não supervisionados permitem o agrupamento de documentos de texto em subgrupos coesos sem serem guiados por um ser humano. Esses algoritmos visam criar clusters diferentes e internamente consistentes que não estão vinculados uns aos outros (AL-SAIF; AL-DOSSARI, 2018). Essas técnicas permitem, àqueles que necessitam de dados, a chegada até as fontes relevantes de maneira mais rápida.

## Sistemas online

Há diferentes tipos de sistemas online que permitem o emprego de técnicas de mineração de dados, como o HTML, CSS e *frameworks*. Eles otimizam a realização de tarefas que envolvem diferentes etapas relacionadas à pesquisa científica que envolvem o manuseio de uma ampla gama de dados. Cada um desses sistemas tem características próximas a serem explicitadas ao longo desta seção.

### HTML

O HTML é uma linguagem de marcação utilizada na elaboração de páginas para a *World Wide Web* (SILVA, 2007). Com a linguagem, os documentos podem conter texto, gráficos, links, formulários e outros conteúdos multimídia, além de possuírem uma estrutura definida (SILVA, 2008). A história do HTML começa em 1980, quando o físico britânico Tim Berners-Lee, então um pesquisador do CERN, criou a linguagem de marcação para hipertexto com a intenção de facilitar a troca de informações entre as pessoas que participavam de seus experimentos (FLATSCHART, 2011). Em 1991, a linguagem foi publicada pelo CERN, tornando-se pública e, consequentemente, adotada por outras instituições.

A partir daí, o HTML sofreu diversas evoluções, com o surgimento das versões 1.0, 2.0 e 3.2, até chegar à versão 4.0, que foi lançada em 1997. A partir daí, o HTML foi padronizado pelo W3C, que é uma organização que padroniza as linguagens de marcação utilizadas na internet, ganhando novas *features* e sendo padronizado pelo W3C (SILVEIRA; ALMEIDA, 2014). Nos dias de hoje, a linguagem é considerada a base para a construção de qualquer página para a internet. Atualmente, a versão mais utilizada do HTML é a 5.0, lançada em 2014. A linguagem de marcação HTML é utilizada para criar páginas na internet (FREEMAN, 2014).

As páginas são criadas com o uso de *tags*, que são elementos que delimitam o conteúdo das páginas para definir a estrutura de uma página, seus títulos, parágrafos, listas, links, imagens, tabelas, formulários, entre outros elementos (SILVA, 2008). Além disso, a linguagem também é responsável por estabelecer a semântica do conteúdo, ou seja, o seu significado (SILVA, 2007). Com o HTML é possível criar páginas para a web de forma simples e intuitiva. A linguagem é bastante flexível e pode ser utilizada para criar páginas com conteúdo diversos, desde um simples texto até complexos sistemas interativos (MILETTO; BERTAGNOLLI, 2014).

Nos dias de hoje, o HTML ainda é a linguagem de marcação mais utilizada para a criação de páginas para a web. Porém, com o surgimento de novas tecnologias, como o CSS e o *Javascript*, a sua utilização foi se limitando ao estabelecimento da estrutura e semântica do conteúdo.

### CSS

O CSS (*Cascading Style Sheets*, em português Folha de Estilo em Cascata) foi criado em 1994 pelo Hakon Wium Lie, um engenheiro da empresa norueguesa de *software* Telenor (SILVA et al., 2021). Lie tinha como objetivo facilitar a troca de estilos nas páginas da web, uma vez que, até então, os estilos eram aplicados diretamente no código HTML (OLIVEIRA JUNIOR; RAABE; ZEFERINO, 2014). Foi criado com o objetivo de separar o conteúdo da formatação de um documento, o que tornava mais fácil a manutenção e atualização do código (RIBEIRO, 2018). Anteriormente, toda formatação era definida no HTML, o que tornava o código muito confuso e difícil de gerenciar.

Com o CSS, os desenvolvedores podiam definir a formatação em um arquivo separado, e aplicar essa formatação a todos os elementos HTML de um site. Sua utilização se torna importante porque é responsável pelo design e estilo de um site (BIÇER; DIRI, 2016). Sem o CSS, um site seria apenas um documento HTML estruturado. O CSS é usado para controlar a formatação e o *layout* de um documento HTML (MISHRA; GUPTA, 2018). Ele pode ser usado para mudar a cor do texto, a fonte, o espaçamento entre linhas, a margem, a largura e a altura das colunas, e muito mais (SILVA et al., 2021). Desde a sua criação, o CSS vem sofrendo constantes evoluções, tanto no que diz respeito à sua sintaxe quanto às suas funcionalidades.

Com o passar dos anos, o CSS foi ganhando novos recursos, o que o tornou cada vez mais poderoso e versátil. A primeira versão do CSS foi criada em 1996, e desde então já foram lançadas mais seis versões: CSS2 (1998), CSS2.1 (1999), CSS3 (2002), CSS4 (2011), CSS5 (2012) e CSS6 (2015) (GANAPATHY et al., 2021). A cada nova versão, o CSS ganha novos recursos e funcionalidades, o que o torna cada vez mais poderoso e versátil (MISHRA; GUPTA, 2018). Atualmente, o CSS é um dos principais pilares da web moderna, sendo utilizado por milhões de sites e aplicações web ao redor do mundo (BIÇER; DIRI, 2016). Com o CSS, é possível criar interfaces ricas e dinâmicas, além de aplicações web complexas.

É uma linguagem de estilo usada para definir a apresentação de documentos HTML e XML. Ele pode ser aplicado a um único elemento ou a um grupo de elementos (RIBEIRO, 2018). O CSS é um recurso muito útil para web designers e desenvolvedores, pois permite que eles criem páginas mais atraentes e interativas. A linguagem CSS é cada vez mais utilizada na web (OLIVEIRA JUNIOR; RAABE; ZEFERINO, 2014). Seu objetivo é separar o conteúdo da forma, permitindo que os desenvolvedores criem páginas mais dinâmicas e atraentes. Além disso, o CSS é muito útil para ajustar a formatação de um documento para diferentes dispositivos, como computadores, *tablets* e telefones celulares sendo possível criar páginas *web* mais bonitas e funcionais (BIÇER; DIRI, 2016).

A linguagem CSS está se tornando cada vez mais popular entre os desenvolvedores e *web designers*.

### Frameworks

Desenvolver um aplicativo do zero pode ser uma tarefa desafiadora. É por isso que os desenvolvedores costumam usar frameworks (MENDES; SILVA; MACIEL, 2021). Um *framework* é um conjunto de bibliotecas de código-fonte que fornecem funcionalidades pré-codificadas e podem ser usadas como base para o desenvolvimento de um aplicativo (FUJISAWA; MACIEL, 2018). Ele fornece um padrão para o código, permitindo que o programador se concentre no código da aplicação em si, em vez de ter que reimplementar as funcionalidades básicas de um programa (STORCK et al., 2017). Usar um framework pode economizar muito tempo e esforço, pois muitas das tarefas de desenvolvimento já foram realizadas.

Isso significa que os desenvolvedores podem se concentrar em criar a lógica do aplicativo e adicionar suas próprias funcionalidades personalizadas (SILVA; CARVALHO; MACIEL, 2021). Os *frameworks* modernos oferecem uma variedade de vantagens, como um ambiente de desenvolvimento unificado, um conjunto de ferramentas integradas e uma base de código comprovada (COSTA et al., 2013). Além disso, muitos *frameworks* são mantidos por grandes comunidades de desenvolvedores, o que significa que eles recebem constantes atualizações e melhorias (RIGO et al., 2014). Existem frameworks para a maioria das linguagens de programação, incluindo Java, .*NET, PHP e Ruby*.

Alguns frameworks são mais populares do que outros, mas todos eles têm seus prós e contras (FUJISAWA; MACIEL, 2018). É importante escolher o framework certo para o seu projeto, pois isso pode facilitar ou dificultar o desenvolvimento (MENDES; SILVA; MACIEL, 2021). Um *framework* é uma estrutura de software que fornece um conjunto de funcionalidades comuns que podem ser utilizadas como base para o desenvolvimento de um determinado tipo de aplicação e que hoje existem diversos tipos de *frameworks*, como por exemplo:

* *Frameworks* para desenvolvimento web: são frameworks voltados para o desenvolvimento de aplicações web, como o *Ruby on Rails, ASP.NET MVC e Django*;
* *Frameworks* para desenvolvimento mobile: são *frameworks* voltados para o desenvolvimento de aplicações *mobile*, como o *Ionic* e o *React Native*;
* *Frameworks* para desenvolvimento de games: são frameworks voltados para o desenvolvimento de games, como o *Unity* e o *Unreal Engine*;
* *Frameworks* para desenvolvimento de aplicações empresariais: são *frameworks* voltados para o desenvolvimento de aplicações empresariais, como o SAP Fiori e o *Oracle Application Express*.

O Laravel é o *framework* PHP mais popular do mundo, com mais de 40 mil estrelas no *GitHub* e mais de 1,3 milhão de sites no ar. Sua comunidade é gigante, ativa e amigável, composta por profissionais de todos os níveis (COSTA et al., 2013). Isso torna muito fácil aprender e usar o Laravel. O Laravel é um *framework* de código aberto PHP, criado por Taylor Otwell e *maintainers* voluntários (RIGO et al., 2014). É um framework com foco na simplicidade, elegância e eficiência. Sua filosofia é baseada em 4 pilares: satisfação do desenvolvedor, flexibilidade, testabilidade e sustentabilidade.

## Linguagens, banco de dados e hospedagem

Os sistemas de gerenciamento de banco de dados – ou DBMS – são um componente obrigatório de qualquer sistema de banco de dados. Eles exigem que os usuários criem e monitorem os seus bancos de dados. Isso é feito por meio de um conjunto de programas chamados de Sistemas de Banco de Dados. Assim, permite-se que os usuários executem todas as tarefas envolvidas na interação com um sistema online que armazena dados (SILVA et al., 2019).

Um sistema de banco de dados corresponde a um programa de software que permite que as pessoas pesquisem e atualizem dados ou informações internas. Os bancos de dados também estão conectados a *hardware*, software e usuários. O DBMS é um sistema computadorizado de uso geral que protege, mantém e repara bancos de dados. Também é usado para compartilhar bancos de dados entre diferentes aplicativos e usuários (AGUIAR et al., 2015).

Este sistema ajuda a proteger os dados contra falhas ou avarias no *hardware* ou *software*, acesso não autorizado ou malicioso e segurança contra avarias no *software* ou *hardware*. Os dados são armazenados em um banco de dados para criar informações. Assim, esses sistemas online são importantes para as empresas e para o mundo, inclusive para o sistema educacional. Eles são úteis porque contêm dados que não podem ser alterados ou destruídos (SILVA; VIEIRA, 2021).

Todas as empresas se utilizam de bancos de dados para rastrear dados e informações, e algumas até os usam para armazenar informações sobre funcionários. Para entender melhor a indústria de Tecnologia da Informação e Comunicação (TIC), é importante entender como os dados e as tecnologias são usados em seus processos. Isso pode ser feito por meio do uso de boas práticas de um banco de dados, bem como por meio da utilização de linguagens de programação que automatizam os processos que envolvem a coleta e gerenciamento de dados (SILVA, 2016).

Muitas empresas aproveitam as redes de comunicação para organizar seus dados e disponibilizá-los aos usuários. Além disso, muitas organizações, incluindo as educacionais, trabalham em pequenas empresas ou mesmo em parques industriais. Esses processos de transformação usam máquinas e dados para criar um terceiro produto. Para utilizar a informação de forma eficaz, é necessário compreender a sua função (GOMES; PIMENTA; SCHNEIDER, 2019).

Armazenar, organizar e acessar informações de forma eficaz pode aumentar sua utilidade. Todavia, compreender as principais linguagens da programação é essencial, pois elas tornam essas operações que envolvem dados mais efetivas e seguras para todos.

### Phyton

No início dos anos 90, Guido Van Rossum criou o *Python* com a intenção de melhorar a eficiência da decodificação e diminuir o tempo necessário para desenvolver projetos. Nascida da necessidade, a criação de Van Rossum serviu a um único propósito. Uma necessidade o inspirou a desenvolver uma linguagem de programação descomplicada que pudesse ser facilmente compreendida por todos. Como pesquisador de uma instituição holandesa, o pesquisador observou muitas necessidades como essa durante o desenvolvimento de seu projeto (GUEDES, 2018).

Nos últimos anos, vários profissionais começaram a usar o *Python* para melhorar seu trabalho. Este crescimento pode ser creditado a vários motivos, como sua facilidade de uso e ampla gama de aplicações. Algumas dessas razões são a crescente demanda por seu uso e a diminuição da popularidade de outras línguas. *Python* é uma linguagem de programação sofisticada que visa facilitar compreensão e a implementação. É uma linguagem de programação dinâmica, interativa, modular, orientada a objetos e multiplataforma (BORGES, 2014).

Isso o torna uma escolha atraente para profissionais da indústria de tecnologia — como engenheiros, cientistas de dados e acadêmicos — que não são programadores. *Python* tem muitas bibliotecas nativas e de terceiros. Isso o torna ideal para uma ampla gama de projetos de desenvolvimento web e aplicativos de *software* em outros setores, como IA, *machine learning* e análise de dados. Além disso, é muito popular entre as pessoas que procuram usar bibliotecas *Python* para *software* em outros setores, como educação e finanças (RAMALHO, 2015).

Muitos computadores agora vêm com bibliotecas integradas que facilitam a criação de scripts de automação na linguagem *Python*. Esses scripts podem ser empregados para realizar tarefas comuns de forma automática, que é um dos principais objetivos perseguidos pelo profissional de tecnologia. Em relação às aplicações do *Python*, uma de suas principais tarefas é a elaboração de aplicativos da web. Possui uma ampla variedade de estruturas — incluindo *FastAPI*, *Flask* e *Django* — que podem ser usadas para diversos fins (MUELLER, 2016).

Alguns usos populares do *Python* incluem a venda de produtos e o gerenciamento de empresas, bem como a criação de sites para ações de *marketing*. Os desenvolvedores de software TDD usam *Python* para criar projetos mais seguros e confiáveis. Isso porque o *Python* vem com um grande volume de frameworks de teste, além de módulos especiais para este objetivo. Essas estruturas ajudam os desenvolvedores a criar testes eficientes e fáceis de implementar. *Python* é uma ótima linguagem voltada à análise e interpretação de volumes expressivos de dados (GUEDES, 2018).

Chamado de *Big Data*, esse campo é um dos usos mais populares da linguagem de programação. Por ser fácil de aprender e usar por profissionais de diferentes áreas, o *Python* é fortemente preferido em relação a outras linguagens de programação. Além disso, os recursos de criação de *scripts* da linguagem de programação a tornam ainda mais desejada. As muitas bibliotecas gratuitas desenvolvidas em colaboração com especialistas de diferentes áreas fazem do *Python* uma escolha ideal para esse propósito (BORGES, 2014).

O *Python* provou ser uma ferramenta eficaz para lidar com grandes conjuntos de dados. É utilizado por muitos profissionais para analisar, processar e exibir suas informações. Além disso, um grande número de bibliotecas está disponível em *Python* que fornecem funções para qualquer finalidade. Isso porque o *Python* ganhou muita popularidade entre os profissionais de tecnologia e entre os cientistas de dados. Existem muitas razões pelas quais as pessoas escolhem aprender *Python*. Uma delas é que a linguagem tem uma curva de aprendizado baixa (BARBOSA et al., 2017).

Alguns acreditam que isso torna os programas *Python* fáceis de entender por novos programadores. Além disso, muitos argumentam que a grande comunidade *Python* do Brasil acredita que compartilhar informações e soluções com profissionais da área ajuda a reduzir as chances de encontrar um problema de programação persistente. O *Python* foi criado para ser uma linguagem simples, fácil de usar e versátil. Sua sintaxe é fácil de entender e é portátil, então pode ser empregada em vários sistemas operacionais sem instalar um interpretador (GUEDES, 2018).

Também é compatível com muitas outras plataformas graças às suas propriedades extensíveis. Existem cerca de 125.000 bibliotecas úteis em *Python* disponíveis e acessíveis. Existem muitas possibilidades ao criar novos aplicativos em razão da ampla coleção de estruturas e bibliotecas disponíveis para a linguagem de programação *Python*. Esta linguagem de programação tem vários caminhos de desenvolvimento abertos. Muitos trabalhos estão disponíveis para profissionais de *Python* graças à versatilidade estrutural da linguagem (RAMALHO, 2015).

Isso porque eles podem criar diferentes sistemas usando *Django*, um dos frameworks mais populares usados para criar aplicações web. Isso permite que eles escolham entre diferentes trabalhos com uma sintaxe simples e uma curva de aprendizado fácil. Por tais motivos, esta linguagem tem sido útil ao campo da mineração de dados.

### Javascript

Conforme Cruz et al (2018), embora computadores e dispositivos móveis sempre tenham sido usados nos tempos modernos, a taxa em que são empregados está aumentando rapidamente. Isto é especialmente verdadeiro nos campos comercial, agrícola, aeronáutico e outros. Além dos computadores comuns, *smartphones*, *laptops* e *smartvs* também podem ser programados usando *Javascript*. Essa linguagem de programação teve um grande aumento em popularidade graças à revolução da indústria 4.0.

Por causa disso, como aponta Bortolossi (2012), cada vez mais comum que os sites usem *Javascript* em vez da programação HTML tradicional. O Javascript está sendo usado por muitos desenvolvedores diferentes - de startups a agências governamentais - na criação de software de desktop, aplicativos móveis, aplicativos da Web e muito mais. O *Javascript* foi criado com a intenção de programar e agilizar páginas da web. Antes do *Javascript*, as páginas da web eram estáticas e continham apenas informações que não podiam ser alteradas.

Como resultado disso, de acordo com Bera, Mine e Lopes (2015), o *Javascript* tornou-se uma linguagem de programação que ajudou a simplificar os sites. Algumas páginas podem ser chamadas de documentos - elas são apenas informativas e não contêm nenhuma informação dinâmica. Essas páginas da Web se assemelham a aplicativos em vez de documentos. Podem carregar novas informações conforme necessário, operar off-line e armazenar dados localmente para que possam ser restaurados quando você revisitar a página.

Conforme Rauschmayer (2014), as informações apresentadas nessas páginas são estáticas; alterado apenas por *Javascript*. A utilidade do *Javascript* tanto para o cliente quanto para o servidor de aplicativos da web é evidente. Ao criar páginas da Web dinâmicas por meio de *Javascript* no lado do cliente, uma página da Web se torna muito mais do que apenas um documento estático. Esta ferramenta também contribui com o gerenciamento de banco de dados. O *RethinkDB* utiliza o *Javascript* para manipular funções em tempo real.

Mardan (2015) reitera que se trata de um banco de dados *JSON* de software livre que pode ser usado para armazenar dados em tempo real; foi criado a partir de expressões *Javascript*. Os bancos de dados *MongoDB*, por sua vez, armazenam dados no formato *BSON*, que é uma versão binária do *JSON*. Eles também utilizam comandos *Javascript* para operar o terminal de comando do banco de dados. Todos esses bancos de dados utilizam-se de linguagens de programação baseadas na mesma sintaxe; alguns deles até contêm scripts que executam funções dentro deles.

Flanagan (2011) pontua que os benefícios em usar um aplicativo baseado em *Javascript* — desempenho e escalabilidade — são alcançados levando em consideração os recursos e as solicitações feitas por um número maior de usuários. *Javascript* é uma linguagem de programação orientada a eventos que oferece algumas vantagens significativas sobre outras linguagens de programação. Uma dessas vantagens é a capacidade do *Javascript* de ser orientado a eventos, o que significa que suas operações de programação são acionadas somente quando necessário.

Outros benefícios do uso de *Javascript* para gestão de dados incluem desempenho, facilidade de dimensionamento e capacidade de programar orientado a eventos. Essas vantagens tornam o *Javascript* uma opção positiva para desenvolver aplicativos em qualquer plataforma.

### MySQL

Segundo Alves (2018), os sistemas utilizados para o gerenciamento de grandes bibliotecas virtuais, conhecidas como banco de dados, são utilizados para manter as informações organizadas e acessíveis. Alguns exemplos desses sistemas são *MySQL, MongoDB* e outros. Os gerenciadores de banco de dados ajudam os usuários a encontrar informações rapidamente e garantir sua confiabilidade. Silva (2001) afirma que essas bibliotecas virtuais correspondem essencialmente a sistemas de armazenamento de informações baseados em computador.

Conforme Milani (2016), esses sistemas permitem aos usuários organizar as informações e as tornem acessíveis quando necessário, de forma rápida e confiável. Com volumes crescentes de dados na sociedade moderna, essa tendência crescente continua todos os dias. Dessa forma, o *MySQL* foi desenvolvido para atuar enquanto sistema que agrega e monitora dados em qualquer circunstância. É bastante conhecido por ser utilizado por grandes empresas, como NASA, Bradesco, HP e SONY. Na verdade, sua popularidade no mercado cresceu a ponto de ultrapassar os sistemas anteriores.

Ele superou outros SGBDs em termos de sucesso de uso no mercado. Nesse sentido, afirma-se que o *MySQL* é um programa de banco de dados gratuito e de código aberto capaz de desenvolvimento contínuo. É compatível com muitos sistemas operacionais, como *Windows*, *Linux, Unix, FreeBSD* e *Mac OS X Server*. Segundo Silva (2001), o MySQL é distribuído sob a licença GNU/GLP, ou *General Public License*. Silberschatz, Sundarshan e Korth (2016) afirmam que alguns sistemas de banco de dados funcionam apenas com uma versão específica.

Essas afirmações permitem compreender que os bancos de dados relacionais possuem tabelas únicas com nomes. Cada nome pertence a uma tabela específica. Silberschatz et al (2002) definem um esquema de relacionamento como uma lista de domínios e atributos associados. Uma linguagem de programação de uma variável é representada pelo conceito de um esquema de relação, enquanto uma linguagem de programação de duas variáveis é representada pelo conceito de uma relação. Richard Stallman iniciou o projeto GNU em 1989.

Conforme Bento Júnior (2021), definiu-se uma licença para esse projeto como “*software* livre”, também conhecido como software de domínio público. Diversos projetos de *software* com código aberto se utilizam de licenças GLP; elas são os únicos tipos de licenças de software que podem ser utilizadas com suas ferramentas e recursos de maneira gratuita. Existem quatro licenças públicas básicas: *Creative* *Commons*, GPL, GLP e *Mozilla Public License*. Há alguns princípios básicos a serem observados.

Souza, Oliveira e Taquaritinga (2019), quanto à utilização, pontuam que o *MySQL* permite que o usuário possa utilizar o software para qualquer tipo de finalidade. Quanto à distribuição, visa que o modus operandi, isto é, o funcionamento do programa, possa ser uma atividade para todos, de modo que este deve ser adaptado para cada necessidade, a depender do propósito que está atendendo e, para isso, o acesso ao código fonte é essencial. Outra característica é a didática. O sistema permite a redistribuição de cópias para que possa ajudar diversos tipos de interessados.

Por fim, no eixo da colaboração, permite que o código-fonte seja alterado para que o programa seja continuamente aperfeiçoado. Esse aperfeiçoamento deve ser acessível para que todos se beneficiem e possam contribuir. Dessa forma, o GLP permite que diversos programas possam ser compartilhados e reaproveitados, sem deixar de resguardar os direitos do autor. Além disso, o *MySQL* tem cinco características essenciais. Silva (2001) afirma que a linguagem de banco de dados SQL é atualmente o principal método para lidar com bancos de dados relacionais.

É uma ferramenta fácil de usar, devido à sua natureza mundial padronizada. Atualmente, é considerada como a linguagem de programação mais universal, pois é reconhecida pelo American National Standards Institute (ANSI) e pela *International* *Standards Organization* (ISO). De acordo com Silva (2001), praticamente não resta nenhuma linguagem de programação que não suporte SQL. Diferentes sistemas operacionais, fornecedores e até idiomas suportam diferentes SQL. Sua velocidade de acesso aos dados se deve ao seu desenvolvimento com tabelas ISAM, que também utilizam caches nas consultas.

Alves (2018) observa que migrar de uma linguagem para outra é mais fácil quando se conhece a SQL. Isso porque não é uma linguagem de programação, mas sim de manipulação de banco de dados. O gerenciamento de caches em um sistema que armazena dados é dispensável; o método de armazenamento mais rápido e caro é a memória cache. É pequeno, mas caro de usar, pois usa o hardware do sistema do computador. Quanto à memória cache em aplicações web, é gerenciada pelo sistema.

Bento Junior (2021) afirma que os engenheiros de estrutura de dados consideram as implicações do armazenamento em cache ao projetar algoritmos e bancos de dados de processamento de dados. Também se utilizam da indexação, algoritmos de busca e outros recursos ao criar suas estruturas de banco de dados. Existem muitos tipos diferentes de bancos de dados que podem ser empregados nos diversos sistemas de computador. Memória flash, armazenamento em fita óptica, armazenamento em disco magnético e cache são algumas das mídias de armazenamento que podem ser adquiridas.

De acordo com Milani (2016), sistemas diferentes usam linguagens e compiladores diferentes para facilitar a portabilidade do *MySQL* em várias plataformas e sistemas. Assim, pode-se adquirir mídia de armazenamento de dados para cada classificação usando estas três medidas: custo por unidade de dados, velocidade de acesso aos dados e confiabilidade. Embora suas tabelas possam armazenar um grande volume de dados, o limite está vinculado ao tamanho máximo de arquivo suportado pelo software em uso. Os módulos de interface do MySQL também incluem linguagens de programação como *Ruby, PHP, ASH* e *Python*.

Souza, Oliveira e Taquaritinga (2019) reiteram que existem também módulos de interface Java atualmente disponíveis no mercado. As tabelas do tipo InooDB podem armazenar dados em vários arquivos e armazenar sub-TBs de dados. O MySQL suporta até 61 milhões de tabelas de junções e, de acordo com o banco de dados, é um dos sistemas online mais rápidos em termos de velocidade de execução. Além disso, as tabelas InooDB podem armazenar até TBs de dados. O *MySQL* é um banco de dados capaz de realizar bilhões de consultas por dia.

Além disso, o pode processar milhões de transações por minuto. Uma das características mais importantes de um sistema que visa gerenciar dados é a sua segurança; O *MySQL* apresenta vários tipos de tabelas que são exclusivas ao software. Isso permite um DBMS seguro e consistente com integridade referencial. O servidor *MySQL* possui um banco de dados chamado *MySQL*, bem como uma tabela contendo informações sobre todos os usuários que se conectaram ao sistema.

O banco de dados é acionado por aquele que monitora o servidor para fins de manutenção de registros e a tabela é usada para gerenciar os privilégios do usuário. Silva (2001) observa que o sistema de gerenciamento de banco de dados do MySQL é confiável quando se trata de segurança. Ele se utiliza de recursos de segurança que oferecem vários níveis de acesso. O comando "conceder" dá aos usuários acesso às funções específicas, enquanto "revogar" elimina as permissões anteriores. Isso concede privilégios totais a qualquer usuário, tabela ou coluna.

Existe outro comando que pode ser usado para limitar o acesso a uma única pessoa, tabela ou coluna. Seu nome é “conceder todos os privilégios. Essas são algumas características que tornam esse sistema vantajoso.

# ESTUDO DE CASO

O estudo de caso se refere ao desenvolvimento de um sistema de mineração de dados para a extração e catalogar dos trabalhos científicos produzidos pela Faculdade de Tecnologia de Mogi Mirim A ferramenta desenvolvida para a mineração dos dados dos trabalhos científicos será programada em linguagem Python, utilizando-se as bibliotecas *Beautiful Sou*p*,* Pandas e *Scikit-Learn*.

A solução proposta para o estudo de caso é o desenvolvimento de uma ferramenta de mineração de dados que permita a extração e categorização dos principais dados dos trabalhos científicos disponibilizados pela Fatec Mogi Mirim. Esta ferramenta será capaz de extrair os dados e categorizá-los em banco de dados, para posteriormente serem disponibilizados em uma plataforma online para consulta da comunidade acadêmica.

## Levantamento de Requisitos

Para o desenvolvimento do sistema de mineração dos dados dos trabalhos científicos da Fatec Mogi Mirim, foi necessário o levantamento de requisitos específicos para o desenvolvimento da ferramenta. Alguns dos principais requisitos foram:

a) A utilização de linguagens de programação como Python, Javascript, HTML e CSS;

b) A utilização de um banco de dados MySQL;

c) A análise das estruturas dos trabalhos científicos;

d) A criação de um algoritmo para a extração e categorização dos dados;

e) A criação de um sistema de busca inteligente.

Além destes requisitos formais, também foi considerado o nível de complexidade dos trabalhos científicos, pois estes podem conter diferentes estruturas e conteúdo, o que torna a mineração dos dados mais complexa.

## Proposta de Solução

Para atender os requisitos levantados, será desenvolvida uma ferramenta de mineração de dados que permita a seleção e consulta futura dos dados extraídos e categorizados de trabalhos de graduação. O sistema será desenvolvido utilizando-se as linguagens de programação Python, Javascript, HTML e CSS, além do banco de dados MySQL.

O algoritmo de mineração desenvolvido será baseado no modelo de mineração de dados basicamente dividido em três etapas: Coleta, Mineração e Pós-Processamento. O processo de coleta dos dados será realizado por meio do acesso aos trabalhos disponibilizados pela Fatec Mogi Mirim e armazenamento dos dados em um banco de dados.

Já o processo de mineração dos dados será realizado por meio da utilização de técnicas de mineração de texto para extração dos dados relevantes. Por fim, o processo de pós-processamento consistirá na análise dos dados extraídos e na criação de um sistema de busca inteligente.

A escolha pelas tecnologias citadas foi feita com base nos requisitos levantados, bem como pela facilidade de utilização e pela sua versatilidade. Isto, porque a linguagem Python possui um alto nível de abstração e facilidade de programação, o que torna o processo de mineração dos dados mais ágil. Já o Javascript permite a criação de sistemas de busca inteligentes, o que auxilia na recuperação de informações. Por fim, a utilização do banco de dados MySQL foi escolhida pela sua flexibilidade e boa performance para este fim de sistemas online.

# CONSIDERAÇÕES FINAIS

**REFERÊNCIAS BIBLIOGRÁFICAS**

AGUIAR, J. J. B. et al. Avaliação de Sistemas de Recomendação Educacionais no Brasil: uma revisão sistemática da literatura. *In*: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 26., 2015, Maceió, AL. **Anais** [...]. Maceió, AL: UFAL, 2015.

AGUIAR, J.; MAGALHÃES, J. Sistemas de Recomendação de Recursos Educacionais: conceitos, técnicas e aplicações. *In*: CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 2., 2013, Campinas. **Anais** [...]. Campinas, SP: UNICAMP, 2013.

AL-SAIF, H.; AL-DOSSARI, H. Detecting and classifying crimes from arabic twitter posts using text mining techniques. **International Journal of Advanced Computer Science and Applications**, v. 9, n. 10, 2018.

ALVES, W. P. **Construindo uma aplicação web completa com PHP e MySQL**. Santa Terezinha, SP: Novatec Editora, 2018.

BAEZA-YATES, R.; RIBEIRO-NETO, B. **Recuperação de Informação**: conceitos e tecnologia das máquinas de busca. 2ª ed. Traduzido por Leandro Krug Wives e Viviane Pereira Moreira. Porto Alegre: Bookman, 2013.

BAKER, R. et al. Mineração de dados educacionais: Oportunidades para o brasil. **Revista Brasileira de Informática na Educação**, v. 19, n. 2, p. 3-13, 2011.

BAKER, R.; ISOTANI, S.; CARVALHO, A. Mineraçao de dados educacionais: oportunidades para o brasil. **Revista Brasileira de Informática na Educação**, v. 19, n. 02, p. 3-13, 2011.

BARBOSA, J. et al. Introdução ao processamento de linguagem natural usando python. **III Escola Regional de Informática do Piauí**, v. 1, p. 336-360, 2017.

BENTO, E. J. **Desenvolvimento web com PHP e MySQL**. São Paulo: Casa do Código, 2021.

BERA, M. H. G; MINE, A. F; LOPES, L. F. B. **MEAN Stack**: desenvolvendo aplicações web utilizando tecnologias baseadas em Javascript. Maringá: Faculdade Cidade Verde, 2015.

BIÇER, M. S.; DIRI, B. Defect prediction for cascading style sheets. **Applied Soft Computing**, v. 49, p. 1078-1084, 2016.

BLUM, A.; MERINO, E. A. D.; MERINO, G. S. A. D. Método visual para revisão sistemática em Design com base em conceitos da Mineração de Dados. **DAPesquisa**, v. 11, n. 16, p. 124-139, 2016.

BORGES, L. E. **Python para desenvolvedores**: aborda Python 3.3. Santa Terezinha, SP: Novatec Editora, 2014.

BORTOLOSSI, H. J. Criando conteúdos educacionais digitais interativos em matemática e estatística com o uso integrado de tecnologias: GeoGebra, JavaView, HTML, CSS, MathML e Javascript. **Revista do Instituto GeoGebra Internacional de São Paulo**, v. 1, n. 1, p. 28-36, 2012.

CORRÊA, G. N.; MARCACINI, R. M.; REZENDE, S. O. **Uso da mineração de textos na análise exploratória de artigos científicos. Relatório Técnico Nº 30**. São Carlos: ICMC, 2012.

COSTA, E. et al. Mineração de dados educacionais: conceitos, técnicas, ferramentas e aplicações. **Jornada de Atualização em Informática na Educação**, v. 1, n. 1, p. 1-29, 2013.

COSTA, E.; AGUIAR, J.; MAGALHÃES, J. Sistemas de Recomendação de Recursos Educacionais: conceitos, técnicas e aplicações. *In*: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO E JORNADA DE ATUALIZAÇÃO EM INFORMÁTICA NA EDUCAÇÃO (JAIE), 24., 2013, Campinas. **Anais** [...]. Campinas: CBIE, 2013.

COSTANTINO, G. et al. Privacy-preserving text mining as a service. *In*: IEEE Symposium on Computers and Communications (ISCC), 1., 2017, Heraklion, Greece. **Anais** [...] Piscataway, Nova Jersey: IEEE, 2017. p. 890-897.

CRUZ, V. da. S. et al. A linguagem Javascript como alternativa para o desenvolvimento de aplicações multiplataforma. **Revista Interface Tecnológica**, v. 15, n. 2, p. 39-49, 2018.

FERREIRA, M. H. W.; CORREA, R. F. Mineração de textos científicos: análise de artigos de periódicos científicos brasileiros da área de Ciência da Informação. **Em Questão**, v. 27, n. 1, p. 237-262, 2021.

FLANAGAN, D. **Javascript**: the definitive guide. 6ª ed. California: O’Reilly Media, 2011.

FLATSCHART, F. **HTML 5 – Embarque Imediato**. Rio de Janeiro: Brasport, 2011.

FRANÇA, R. S.; AMARAL, H. J. C. Mineração de dados na identificação de grupos de estudantes com dificuldades de aprendizagem no ensino de programação. **RENOTE**, v. 11, n. 1, 2013.

FREEMAN, E. **Use a Cabeça! Programação em HTML**. Rio de Janeiro: Alta Books Editora, 2014.

FUJISAWA, I. Y.; MACIEL, A. M. A. Desenvolvimento de um framework integrador de mineraçao de dados educacionais. **Revista de Engenharia e Pesquisa Aplicada**, v. 3, n. 3, 2018.

FULANTELLI, G. et al. A framework to support educational decision making in mobile learning. **Computers in Human Behavior**, v. 47, p. 50-59, 2015.

GANAPATHY, A. et al. HTML Content and Cascading Tree Sheets: Overview of Improving Web Content Visualization. **Turkish Online Journal of Qualitative Inquiry**, v. 12, n. 3, p. 2428-2438, 2021.

GIL, A. C. **Como elaborar projetos de pesquisa**. 7ª ed. São Paulo: Atlas, 2018.

GOMES, J. C.; PIMENTA, R. M.; SCHNEIDER, M. Mineração de dados na pesquisa em ciência da informação: desafios e oportunidades. *In*: ENANCIB, 20., 2019, Florianópolis. **Anais** [...]. Florianópolis, SC: UFSC, 2019.

GUEDES, D. B. **Linguagem de programação Python e arduino como ferramenta para motivar estudantes iniciantes em programação**. 2018. 80f. Trabalho de Conclusão de Curso (Tecnólogo em Tecnologia em Sistemas de Telecomunicações) - Universidade Tecnológica Federal do Paraná, Curitiba, PE, 2018.

IMRAN, H. et al. A framework to provide personalization in learning management systems through a recommender system approach. **Springer**, p. 271-280, 2014.

JOHNSON, L. R. et al. **Learning Analytics. The 2011 Horizon Report**. Austin, Texas: The New Media Consortium, 2011.

KRASSMANN, A. L. et al. Analyzing trends in academic papers about ubiquitous virtual worlds in education using text mining. **International Journal for Innovation Education and Research**, v. 5, n. 4, p. 167-180, 2017.

KUANG, D. et al.; BRANTINGHAM, P. J.; BERTOZZI, A L. Crime topic modeling. **Crime Science**, v. 6, n. 1, p. 1-20, 2017.

LIU, C‐C. et al. The effect of recommendation systems on I nternet‐based learning for different learners: A data mining analysis. **British Journal of Educational Technology**, v. 44, n. 5, p. 758-773, 2013.

MARDAN, A. **Full Stack Javascript**. 2ª ed. New York: Apress Media, 2015.

MASCHIO, P. et al. Um panorama acerca da mineração de dados educacionais no Brasil. In: **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)**. 2018.

MEENA, K.; LAWRANCE, R. An automatic text document classification using modified weight and semantic method. **International Journal of Innovative Technology and Exploring Engineering**, v. 8, n. 12, p. 2608-2622, 2019.

MENDES, V. M. P.; SILVA, R. G.; MACIEL, A. M. A. Usability Analysis of an Educational Data Mining Framework. **Revista de Engenharia e Pesquisa Aplicada**, v. 6, n. 3, p. 31-38, 2021.

MILANI, A. **Construindo Aplicações Web com PHP e MySQL**. 2ª ed. Santa Terezinha, SP: Novatec Editora, 2016.

MILETTO, E. M.; BERTAGNOLLI, S. de. C. **Desenvolvimento de Software II**: Introdução ao Desenvolvimento Web com HTML, CSS, Javascript e PHP-Eixo: Informação e Comunicação-Série Tekne. Porto Alegre: Bookman Editora, 2014.

MISHRA, A.; GUPTA, B. B. Intelligent phishing detection system using similarity matching algorithms. **International Journal of Information and Communication Technology**, v. 12, n. 1-2, p. 51-73, 2018.

MUELLER, J. P. **Começando a programar em Python para leigos**. Rio de Janeiro: Alta Books Editora, 2016.

OLIVEIRA JUNIOR, N. P.; RAABE, A.; ZEFERINO, C. A. Implementando suporte a novas linguagens de programação e outros idiomas no ambiente de desenvolvimento integrado bipide. **International Journal of Computer Architecture Education**, v. 3, p. 5-8, 2014.

ORENGO, V. M., HUYCK, C. A stemming algorithm for the portuguese language. *In*: Symposium on String Processing and Information Retrieval, 8., 2001, Laguna de San Rafael, Chile. **Anais** [...]. Piscataway, Nova Jersey: IEEE, 2001.

PAIVA, R. et al, A. Uma ferramenta para recomendação pedagógica baseada em mineração de dados educacionais. *In*: CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO - CBIE, 2., 2013, Campinas. **Anais** [...] Campinas: UNICAMP, 2013.

PARREÑO, J. et al. The use of gamification in education: a bibliometric and text mining analysis. **Journal of Computer Assisted Learning**, v. 32, n. 6, p. 663-676, 2016.

PAZ, F. J.; CAZELLA, S. C. Aplicando Mineração de Textos na análise de artigos científicos sobre Sistemas de Recomendação com Mineração de Dados Educacionais e​​ Learning Analytics. **RENOTE**, v. 16, n. 1, p. 1-11, 2018.

RAMALHO, L. **Python Fluente**: Programação clara, concisa e eficaz. Santa Terezinha, SP: Novatec Editora, 2015.

RAUSCHMAYER, A. **Speaking Javascript**: an in-depth guide for programmers. 1ª ed. Califórnia: O’Reilly Media, 2014.

RIBEIRO, M. T. Utilização do Gtmetrix em Websites: Estudo de Viabilidade para a Fábrica de Tecnologias Turing. **Revista Ada Lovelace**, v. 2, p. 72-77, 2018.

RIGO, S. J. et al. Aplicações de Mineração de Dados Educacionais e Learning Analytics com foco na evasão escolar: oportunidades e desafios. **Revista Brasileira de Informática na Educação**, v. 22, n. 1, p. 132-146, 2014.

SANTOS, L. A. N. dos. **Contribuição da mineração de dados e da otimização heurística para a interpretação dos dados da produção científica brasileira**. 2011. 114f. Dissertação (Mestrado em Ciência da Informação) - Universidade Federal da Bahia, Salvador, BA, 2011.

SCHIESSl, M.; BRASCHER, M. Descoberta de conhecimento em texto aplicada a um sistema de atendimento ao consumidor. **RICI: Revista Ibero-americana de Ciência da Informação**, v. 4, n. 2, p. 94-110, 2011.

SIEMENS, G.; LONG, P. Penetrating the fog: Analytics in learning and education. **EDUCAUSE Review**, v. 46, n. 5, p. 30, 2011.

SILBERSCHATZ, A. et al. **Fundamentos de bases de datos**. Ciudad de México, México: McGraw-Hill, 2002.

SILBERSCHATZ, A.; SUNDARSHAN, S.; KORTH, H. F. **Sistema de banco de dados**. São Paulo: Elsevier Brasil, 2016.

SILVA, A. T. I. da. et al. A importância do banco de dados para o controle em uma indústria. **Tekhne e Logos**, v. 10, n. 2, p. 76-89, 2019.

SILVA, G. L. B.; CARVALHO, J. A. de.; MACIEL, A. M. A. Desenvolvimento de um Learning Analytics Dashboard a partir de Modelos de Mineração de Dados Educacionais. **Revista de Engenharia e Pesquisa Aplicada**, v. 6, n. 3, p. 59-69, 2021.

SILVA, L. C. **Banco de Dados para Web**. 1ª ed. São Paulo: Érica, 2001.

SILVA, L.; PERES, S.; BOSCARIOLI, C. **Introdução a Mineração de Dados com aplicações em R**. Rio de Janeiro: Elsevier, 2017.

SILVA, M. L. da. et al. Um estudo exploratorio sobre dados abertos em Universidades. In: JORNADA DE INFORMÁTICA DO MARANHÃO, 6., 2016, São Luís. Anais [...]. São Luís, MA: UFMA, 2016.

SILVA, M. P. da.; VIERA, A. F. G. Descoberta de conhecimento com uso de técnicas de mineração de textos aplicadas em documentos textuais da investigação policial brasileira. **Investigación Bibliotecológica**, v. 35, n. 88, p. 161-183, 2021.

SILVA, M. S. **Construindo sites com CSS e (X) HTML**: sites controlados por folhas de estilo em cascata. Santa Terezinha, SP: Novatec Editora, 2007.

SILVA, M. S. **Criando sites com HTML**: sites de alta qualidade com HTML e CSS. Santa Terezinha, SP: Novatec Editora, 2008.

SILVA, P. H. B. et al. Desenvolvimento de software destinado a gestão de estoque em uma unidade escolar: integração dos saberes adquiridos na graduação. **Prospectus**, v. 3, n. 2, p. 124-146, 2021.

SILVA, R. M. da. **Modelo de mineração de dados em bases de dados acadêmicas**. 2016. 111f. dissertação (Mestrado em Engenharia Elétrica) - Universidade de Brasília, Brasília, DF, 2016.

SILVEIRA, P.; ALMEIDA, A. **Lógica de programação**: crie seus primeiros programas usando Javascript e HTML. São Paulo: Casa do Código, 2014.

SINOARA, Roberta Akemi; MARCACINI, Ricardo Marcondes; REZENDE, Solange Oliveira. Mineraçao de Textos e Semântica: desafios, abordagens e aplicaçoes. **Revista de Sistemas de Informaçao da FSMA**, v. 27, n. ja-ju 2021, p. 41-53, 2021.

STORCK, C. R. et al. Proposta de um Framework Baseado em Mineração de Dados para Redes 5G. **Revista Eletrônica de Sistemas de Informação**, v. 16, n. 2, p. 1-16, 2017.

WEISS, S. M. et al. **Text Mining**: Predictive Methods for Analyzing Unstructured Information. New York: Springer, 2010.

YANG, Y.; MANOHARAN, M.; BARBER, K. S. Modelling and Analysis of Identity Threat Behaviors through Text Mining of Identity Theft Stories”. *In*: IEEE Joint Intelligence and Security Informatics Conference, 2014, The Hague, Netherlands. **Anais** [...]. Piscataway, Nova Jersey: IEEE, 2014. p. 50-63.

ZHAI, C.; MASSUNG, S. **Text Data Management and analysis**: a practical introduction to Information Retrieval and Text Mining. USA: Association for Computing Machinery and Morgan & Claypool, 2016.

Lista de autores e datas:

Neto (2013)

Lawrance (2019)

al (2018)

Bortolossi (2012)

Lopes (2015)

Rauschmayer (2014)

Mardan (2015)

Flanagan (2011)

Alves (2018)

Silva (2001)

Milani (2016)

Korth (2016)

al (2002)

Júnior (2021)

Taquaritinga (2019)

Junior (2021)

Lista de autores e datas:

Lista de autores e datas:

Lista de autores e datas:

Neto (2013)

Lawrance (2019)

al (2018)

Bortolossi (2012)

Lopes (2015)

Rauschmayer (2014)

Mardan (2015)

Flanagan (2011)

Alves (2018)

Silva (2001)

Milani (2016)

Korth (2016)

al (2002)

Júnior (2021)

Taquaritinga (2019)

Junior (2021)