
Uma Abordagem de Arquitetura em Nuvem para Dados Educacionais em um Sistema Tutor Inteligente em contexto de Big Data

Regis Michel dos Santos Sousa



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Regis Michel dos Santos Sousa

**Uma Abordagem de Arquitetura em Nuvem para
Dados Educacionais em um Sistema Tutor Inteligente
em contexto de Big Data**

Tese de doutorado apresentada ao Programa de Pós-graduação da Faculdade de Engenharia Elétrica da Universidade Federal de Uberlândia como parte dos requisitos para a obtenção do título de Doutor em Ciências.

Área de concentração: Inteligência Artificial

Orientador: Luciano Vieira Lima

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema de Bibliotecas da UFU, MG, Brasil.

S725a Sousa, Régis Michel dos Santos,
2022 Uma abordagem de arquitetura em nuvem para dados educacionais
em um sistema tutor inteligente em contexto de Big Data [recurso
eletrônico] / Régis Michel dos Santos Sousa. - 2022.

Orientador: Luciano Vieira Lima.

Tese (Doutorado) - Universidade Federal de Uberlândia, Programa
de Pós-Graduação em Engenharia Elétrica.

Modo de acesso: Internet.

Disponível em: <http://doi.org/10.14393/ufu.te.2022.5039>

Inclui bibliografia.

1. Engenharia Elétrica. I. Lima, Luciano Vieira (Orient.). II.
Universidade Federal de Uberlândia. Programa de Pós-Graduação em
Engenharia Elétrica. III. Título.

CDU: 621.3

André Carlos Francisco
Bibliotecário – CRB-6/3408



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
 Coordenação do Programa de Pós-Graduação em Engenharia Elétrica
 Av. João Naves de Ávila, 2121, Bloco 3N - Bairro Santa Mônica, Uberlândia-MG, CEP 38400-902
 Telefone: (34) 3239-4707 - www.posgrad.feelt.ufu.br - copel@ufu.br



ATA DE DEFESA - PÓS-GRADUAÇÃO

Programa de Pós-Graduação em:	Engenharia Elétrica				
Defesa de:	Tese de Doutorado, 309, PPGEELT				
Data:	Vinte e quatro de novembro de dois mil e vinte e dois	Hora de início:	12:00	Hora de encerramento:	16:00
Matrícula do Discente:	11813EEL013				
Nome do Discente:	Régis Michel dos Santos Sousa				
Título do Trabalho:	Uma abordagem de arquitetura em nuvem para dados educacionais em um sistema tutor inteligente em contexto de Big Data				
Área de concentração:	Processamento da Informação				
Linha de pesquisa:	Inteligência Artificial				
Projeto de Pesquisa de vinculação:	Coordenador do projeto: Luciano Viera Lima Título do projeto: INTELIGÊNCIA ARTIFICIAL E NEUROCIÊNCIA APLICADA AO ENSINO E APRENDIZADO REMOTO OU PRESENCIAL, ASSISTIVO OU NÃO Agência financiadora: Não há. Número do processo na agência financiadora: Vigência do projeto: 2004-atual				

Reuniu-se por meio de videoconferência, a Banca Examinadora, designada pelo Colegiado do Programa de Pós-graduação em Engenharia Elétrica, assim composta: Professores Doutores: José Eduardo Ferreira Lopes - FAGEN-UFU; Sara Luzia de Melo - IC-UFF; Franciny Medeiros Barreto - FACOM-UFJ; Leiliane Pereira de Rezende - UTFPR-SH; Luciano Vieira Lima - FEELT/UFU, orientador(a) do(a) candidato(a).

Iniciando os trabalhos o(a) presidente da mesa, Dr(a). Luciano Vieira Lima, apresentou a Comissão Examinadora e o candidato(a), agradeceu a presença do público, e concedeu ao Discente a palavra para a exposição do seu trabalho. A duração da apresentação do Discente e o tempo de arguição e resposta foram conforme as normas do Programa.

A seguir o senhor(a) presidente concedeu a palavra, pela ordem sucessivamente, aos(às) examinadores(as), que passaram a arguir o(a) candidato(a). Ultimada a arguição, que se desenvolveu dentro dos termos regimentais, a Banca, em sessão secreta, atribuiu o resultado final, considerando o(a) candidato(a):

Aprovado(a).

Esta defesa faz parte dos requisitos necessários à obtenção do título de Doutor.

O competente diploma será expedido após cumprimento dos demais requisitos, conforme as normas do Programa, a legislação pertinente e a regulamentação interna da UFU.

Nada mais havendo a tratar foram encerrados os trabalhos. Foi lavrada a presente ata que após lida e achada conforme foi assinada pela Banca Examinadora.



Documento assinado eletronicamente por **Sara Luzia de Melo, Usuário Externo**, em 25/11/2022, às 12:11, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Leiliane Pereira de Rezende, Usuário Externo**, em 25/11/2022, às 12:25, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Franciny Medeiros Barreto, Usuário Externo**, em 25/11/2022, às 15:02, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **José Eduardo Ferreira Lopes, Professor(a) do Magistério Superior**, em 29/11/2022, às 09:53, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Luciano Vieira Lima, Professor(a) do Magistério Superior**, em 29/11/2022, às 16:01, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site https://www.sei.ufu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **4094114** e o código CRC **273FFE71**.

A minha mãe, Anesia do Nascimento, pelo amor incondicional, apoio e compreensão em todos os momentos da minha vida.

Agradecimentos

Agradeço...

A Deus, por ser maravilhoso e me mostrar que sempre devo buscar e persistir naquilo em que eu acredito.

A minha mãe Anesia do Nascimento, meu maior exemplo de pessoa honesta e batalhadora, que mesmo com todas as dificuldades na infância, sempre me motivou a continuar nos estudos.

A minha querida professora e madrinha Ana Mirian Rossi que desde pequenino me teve como um filho e esteve sempre presente me incentivando a aprimorar meus estudos. Foi por meio dela que percebi o quanto é nobre o papel do professor e a importância dele em contribuir para mudar a vida das pessoas.

A minha família Uberlandense, com quem partilho todos meus momentos, Fernando Lopes, Gilberto Santos, Ronaldo Luis e Sonia Miguel.

A minha família Goiana, em especial, meu amigo Thiago Perim, por quem tenho muita admiração e carinho. A Rosa Helena Perim (mais uma mãe que a vida me deu) e Paulo Roberto pelo carinho e apoio de sempre.

A minha amiga Sara, por todos os ensinamentos e parceria.

Ao meu orientador, Luciano Vieira Lima, pelos ensinamentos, paciência e dedicação durante este trabalho, e que veio a se tornar um amigo.

Ao meu amigo e professor José Eduardo, minha maior inspiração profissional. Por todos ensinamentos desde a graduação, o que foi fundamental para que eu pudesse realizar mais uma etapa na vida acadêmica.

Aos amigos do laboratório LINA e aos amigos dos laboratórios da Faculdade de Computação da Universidade Federal de Uberlândia.

A CAPES pelo apoio financeiro.

*"O que eu conheço de Deus são as pegadas de sua ausência."
(Padre Fábio de Melo)*

Resumo

O processo de ensino e aprendizagem em sistemas computacionais de ensino têm crescido de maneira acentuada nos últimos anos, e os Sistemas Tutores Inteligentes (STI) têm ganhado destaque ao propor a personalização dos ambientes ao estudante e assim aprimorar o processo de ensino e aprendizagem. Como consequência do aumento de usuários em plataformas de ensino, tem-se a geração grande volumes de dados, dos quais vêm sendo objetivos de pesquisas para propor aprimoramentos do desempenho na recuperação, armazenamento e processamento de dados em contexto Big Data. Este trabalho objetiva propor uma abordagem de arquitetura de referência em Big Data utilizando microsserviços em nuvem para o STI-LINA. Para isto, esta abordagem concentra-se em mapear os processos de ingestão, armazenamento, captura e enriquecimento de dados e definir serviços adequados para cada um deles. Como resultado deste trabalho foi possível propor arquiteturas baseadas em microsserviços utilizando serviços AWS, para todos os modelos de compõem o STI-LINA (Estudante, Domínio e Pedagógico). E, além disso, foi possível propor um novo módulo que irá disponibilizar informações analíticas para auxiliar o professor a acompanhar a trajetória do aluno no processo de ensino e aprendizagem. E, por fim, apresentou-se uma proposta de inclusão do método de acesso métrico, denominado Slim-Tree-Lina para otimizar a recuperação de objetos de aprendizagem ofertados ao estudante.

Palavras-chave: Smart Tutor Systems, Cloud Architecture, Microservices, Big Data.

Abstract

The teaching and learning process in computer education systems has grown sharply in recent years and Intelligent Tutoring Systems (STI) have gained prominence by proposing the customization of environments to the student and thus improving the teaching and learning process. As a consequence of the increase in users of teaching platforms, large volumes of data have been generated, which have been research objectives to propose performance improvement in data retrieval, storage and processing in the Big Data context. This work aims to propose a reference architecture approach in Big Data using cloud microservices for STI-LINA. For this, this approach focuses on mapping the processes of ingestion, storage, capture and enrichment of data and define adequate services for each of them. As a result of this work, it was possible to propose architectures based on microservices using AWS services, for all with models that make up the STI-LINA (Student, Domain and Pedagogical). In addition, it was possible to propose a new module/component that will provide analytical information to help the teacher follow the student's trajectory in the teaching and learning process. Finally, a proposal was made to include the metric access method, called Slim-Tree-Lina, to optimize the recovery of learning objects offered to the student.

Keywords: Intelligent Tutoring Systems. Information Retrieval, Big Data..

Lista de ilustrações

Figura 1 – Processo geral do <i>Design Research</i>	24
Figura 2 – Arquitetura Clássica de um STI.....	28
Figura 3 – Novo conhecimento implica em nova área de ignorância na estrutura dos MCE.....	30
Figura 4 – Arquitetura STI-LINA.	32
Figura 5 – Características de Big Data.....	34
Figura 6 – Ecossistema Apache Hadoop.....	37
Figura 7 – Comparativo entre as opções disponíveis.	43
Figura 8 – Escolha de Serviços Amazon Web Services (AWS) de acordo tipo de dados a serem manipulados.....	44
Figura 9 – Mapeamento das aplicações que fazem parte dos Modelo do Estudante (a) Conhecimento prévio e Conhecimento atual, (b) Perfil e de personalidade e (c) Emoções. E da Interface de Estudante responsável pelo processo de ingestão de dados neste módulo.	50
Figura 10 – Mapeamento das componentes de aplicação que fazem parte dos Módulo do Domínio (a) Conteúdo Programático, (b) Avaliação de Itens e (c) Objetos de Aprendizagem. E da Interface de Domínio que faz o processo de ingestão de dados neste módulo.	52
Figura 11 – Modelagem Entidade Relacionamento para componentes do MD e ME do STI-LINA	55
Figura 12 – Modelagem de dados para componente Emoções.	56
Figura 13 – Modelagem de dados para componente Preferências.	57
Figura 14 – Modelagem de dados para componente Preferências.	57
Figura 15 – Arquitetura com serviços AWS para suporte aos processos de ingestão de dados pela Interface do Estudante até o armazenamento dos dados no RDS.	58
Figura 16 – Processo de ingestão de dados pela Interface do professor até o armazenamento no RDS ou S3.	59

Figura 17 – Processo de proposta de Módulo Pedagógico	60
Figura 18 – O processo de atualização de dados do método Slim-Tree-Lina.....	62
Figura 19 – Visão geral a Proposta do Módulo de Analytics utilizando serviços AWS.....	64

Lista de Quadros

Quadro 1 – Áreas de aplicação do Big Data.....	35
Quadro 2 – Cenários de Uso para Tecnologias Relacionais e NoSQL.....	41
Quadro 3 – Visão geral dos componentes do Modelo do Estudante por tipo de processamento.....	51
Quadro 4 – Visão geral dos componentes do Modelo de Domínio por tipo de pro- cessamento.	53
Quadro 5 – Visão geral dos componentes do Modelo do Estudante e Modelo de Domínio por definição de tipo de dados e escolha de tecnologia para armazenamento de dados em nuvem.....	54

Lista de siglas

AVA Ambientes Virtuais de Aprendizagem

AWS Amazon Web Services

BDW *Big Data Warehousing*

DW *Data Warehousing*

EDM *Educational Data Mining*

HDFS *Hadoop Distributed File System*

LA *Learning Analytics*

MAM Método de Acesso Métrico

MCEs Teoria dos Mapas de Conhecimentos Estruturados

ME Modelo do Estudante

MD Modelo de Domínio

MP Modelo Pedagógico

NoSQL Not Only SQL

SQL *Structured Query Language*

STI-LINA Sistema Tutor Inteligente LINA

SIE Sistemas de Informações Educacionais

SQL *Structured Query Language*

TTL *Time to Live*

YARN *Yet Another Resource Negotiator*

Sumário

1	INTRODUÇÃO.....	21
1.1	Introdução	21
1.2	Objetivos e Desafios da Pesquisa	22
1.3	Hipóteses	22
1.4	Métodos.....	23
1.4.1	A Estrutura de Pensamento do <i>Design Research</i>	23
1.5	Organização da Tese.....	25
2	FUNDAMENTAÇÃO TEÓRICA.....	27
2.1	Sistema Tutores Inteligentes	27
2.1.1	Mapas de Conhecimento Estruturados.....	29
2.1.2	STI-LINA: Um Sistema Tutor Inteligente Adaptativo ao Curso e Personalizado ao Estudante.....	30
2.2	O Big Data	33
2.2.1	Áreas de aplicação do Big Data	34
2.2.2	Desafios da descoberta de conhecimentos por fases.....	35
2.2.3	Tecnologias de Big Data	36
2.2.4	Crítérios para escolha de uma arquitetura de Big Data	37
2.3	Big Data na Educação.....	39
2.3.1	Mineração de dados Educacionais (EDM).....	39
2.3.2	Big Data e Armazenamento de dados	40
2.3.3	Big Data e Definição de Tecnologias de Armazenamento em Nuvem	42
2.3.4	Arquitetura de Microserviços	46
2.4	Trabalhos Correlatos	46
3	PROPOSTA DA TESE	49
3.1	Proposta de Arquitetura para os Modelos Estudante, Domínio e Pedagógico do STI-LINA	49
3.1.1	STI-LINA – Modelo do Estudante e Interface do Estudante.....	50
3.1.2	Classificação de necessidade de processamento <i>batch</i> e <i>streaming</i> para componentes do Módulo do Estudante	51
3.1.3	Classificação de necessidade de processamento <i>batch</i> e <i>streaming</i> para componentes do Modelo de Domínio	52

3.1.4	Armazenamento de dados para Modelo do Estudante (ME) e Modelo de Domínio (MD) do STI-LINA	53
4	CONSIDERAÇÕES FINAIS	67
4.1	Contribuições.....	67
4.2	Trabalhos Futuros.....	68
4.3	Produções Bibliográficas	68
	REFERÊNCIAS.....	71

Introdução

1.1 Introdução

O processo de ensino e aprendizagem em sistemas computacionais de ensino têm crescido de maneira acentuada nos últimos anos e foi impulsionado também pela pandemia do COVID-19, no qual o ensino teve que se adaptar à modalidade não presencial (COSTA; SOUSA, 2020). Como consequência desse aumento de demanda, os sistemas educacionais passam a gerar grande volume de dados, dos quais vêm sendo objetivos de pesquisas no sentido de aprimoramento do ensino e aprendizagem nos ambientes virtuais, ao fazer uso de informações úteis para estudantes, professores e para a gestão acadêmica.

Nesse contexto, surgem desafios que consistem em determinar como os dados serão capturados, padronizados, processados, armazenados, apresentados e utilizados para benefícios dos resultados de ensino-aprendizagem (NOGUEIRA et al., 2018). Em face deste contexto, o Big Data tem como pressuposto aprimorar o desempenho à recolha, armazenamento, processamento, e análise de vastas quantidades de dados heterogêneos, gerados em velocidades cada vez maiores. Assim, surge a possibilidade de extração de conhecimentos desses dados e, com isso, a necessidade de criação de estratégias para definir como manipular grandes volumes de dados e analisar as informações úteis (WEBSTER, 2014). Um Sistema Tutor Inteligente (STI), em sua arquitetura clássica, é composto por três modelos, sendo eles, Modelo do Estudante, Modelo Pedagógico e Modelo de Domínio. Os STI incorporam técnicas de Inteligência Artificial e fornecem um rastreamento eficaz e personalizado durante o processo de ensino e aprendizagem. Além disso, os STI oferecem as seguintes vantagens: (1) Constituem uma fonte de material didático; (2) Fornecem problemas para o estudante atingir um certo nível de conhecimento; (3) Controlam o nível de dificuldade dos problemas para que os alunos enfrentem exercícios adequados às suas necessidades; (4) Ajudam o estudante a adquirir conhecimentos adicionais (RAMÍREZ-NORIEGA et al., 2019).

Nota-se que há, na literatura acadêmica, carência de especificações objetivas que visam nortear a seleção de uma arquitetura e de ferramentas para implementação de aplicações que

levam em considerações o contexto de Big Data para dados educacionais. Preencher esta lacuna é significativamente relevante, razão pela qual este trabalho propõe uma abordagem composta por modelos e métodos para conceber e implementar uma arquitetura baseada em microsserviços, a fim de satisfazer alguns requisitos, como a escalabilidade em cenários de Big Data para dados educacionais.

1.2 Objetivos e Desafios da Pesquisa

O objetivo deste trabalho é propor uma abordagem de arquitetura de referência em Big Data utilizando microsserviços em nuvem para o Sistema Tutor Inteligente, denominado STI-LINA, proposto por Melo (2021). Para isto, esta abordagem concentra-se em mapear os processos de ingestão, armazenamento, captura e enriquecimento de dados.

Sendo assim, para alcançar este objetivo, são necessários satisfazer os seguintes objetivos específicos:

1. Identificar as principais arquiteturas e tecnologias que compõem o Big Data;
2. Mapear e definir uma arquitetura em nuvem utilizando microsserviços que contemplem os processos de ingestão e armazenamento de dados para os ME e o MD do STI-LINA.
3. Definir o *schema* de dados e a tecnologia de armazenamento de dados para os ME e o MD do STI-LINA.
4. Mapear e definir uma arquitetura em nuvem utilizando microsserviços que contemplem processos de captura e enriquecimento do Modelo Pedagógico (MP) do STI-LINA.
5. Utilizar um Método de Acesso Métrico (MAM) em conjunto com módulo de recomendação de objetos de aprendizagem para otimizar a recuperação dos objetos por similaridade e, conseqüentemente, ofertar mais opções personalizadas para o aluno.
6. Desenvolver um Módulo de Analytics para que possibilite ao professor acompanhar a trajetória do aluno na plataforma de ensino, além de poder tomar decisões para melhorar o processo de ensino e aprendizagem.

1.3 Hipóteses

As hipóteses desta tese de doutorado consistem em:

- É possível definir uma arquitetura de Big Data integrada para o STI-LINA que utilize microsserviços em nuvem e também sirva de insumo para todas as tarefas que são realizadas pelas aplicações do STI-LINA, considerando os cenários *batch* e *streams*.

- É possível a utilização do Método de Acesso Métrico para otimizar o Módulo de Recomendação de Objetos de Aprendizagem Personalizado ao estudante.
- É possível integrar ao STI-LINA um módulo responsável por prover informações analíticas para a tomada de decisão pelos professores.

1.4 Métodos

O *Design Research* é um conjunto de técnicas analíticas e perspectivas para a realização de investigação em sistema de informação que resulta na criação de um artefato (SIMON, 2019). Um artefato é definido como algo físico ou ainda pode ser visualizado de forma abstrata, sendo i) **criações** (vocabulário e símbolos); ii) **modelos** (captações e representações); iii) **métodos** (algoritmos e prática) e; iv) **instanciação** (protótipo e sistemas) (HEVNER et al., 2004).

Com relação aos dois últimos, destacamos que:

1. Método é um conjunto de passos, ou algoritmo, ou guia para desempenhar uma tarefa específica. Esses passos são baseados em um conjunto de construtos e em um modelo do espaço de soluções;
2. Instanciação é a realização dos artefatos em seu ambiente. Após a operacionalização dos construtos, modelos e métodos, busca-se demonstrar a possibilidade e a efetividade dos modelos e métodos em um ambiente real.

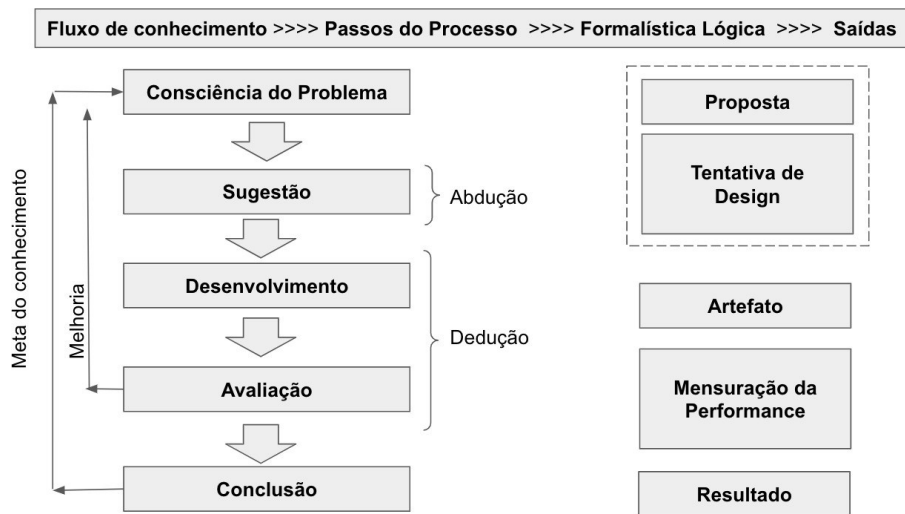
A utilização de conhecimentos para a concepção e criação de artefatos úteis é o que define o conceito de *Design Research*, o qual deve utilizar vários métodos rigorosos para analisar por que razão ou não um artefato é especialmente eficaz. Deve, também, contribuir para a análise e para a construção de artefatos, a fim de compreender e explicar o comportamento e os aspectos de um determinado sistema (MARCH; SMITH, 1995).

1.4.1 A Estrutura de Pensamento do *Design Research*

O processo, proposto por Takeda, Verkamp e Yoshikawa (1990), permite a construção de artefatos, ou seja, modelos e métodos de instanciação, como ilustrado na Figura 1.

O processo do *Design Research* se inicia quando o pesquisador propõe-se a solucionar um problema. Para isso, o mesmo necessita entender a natureza do problema, o contexto, as potencialidades e suas limitações, o que ajuda a compreender melhor o ambiente dentro do qual o problema se insere.

Essa fase é denominada **consciência do problema**, e é nela que começa a construção formal ou informal que permite o processo de investigação. A análise de um conjunto de

Figura 1 – Processo geral do *Design Research*

Fonte: Takeda et al. (1990) e adaptado por Vaishnavi e Kuecheler (2005).

conceitos, teorias e relações verificadas experimentalmente é útil para explicar processos e resultados organizacionais. Logo, em seguida, tem-se a fase de **sugestão** que é um passo essencialmente criativo. Nesta fase é que os diferentes itens investigados alcançam os diferentes desenhos dos artefatos a serem propostos. Nessa fase também são realizadas proposições imperativas, fundamentais nos estados da arte da teoria, para produzir projetos ou reinventar projetos existentes.

A construção efetiva de um artefato pelo investigador é considerada como a fase de **desenvolvimento**, no qual ele irá construir um ou mais artefatos com o objetivo de equacionar o problema. O investigador poderá compor um entrelaçamento de técnicas, o que também pode variar de acordo com as características do objeto que está sendo arquitetado. Uma vez construído o artefato, é dada início à fase de **avaliação**.

Essa fase consiste em avaliar o artefato que foi desenvolvido. Este deve ser analisado e testado de acordo com as condições estabelecidas para validação. As descobertas da avaliação provavelmente servirão de complementação do artefato, e será sugerido um processo de **melhoria**. Transposta a fase de avaliação, o artefato é direcionado para a **conclusão**. Nessa etapa, por meio da análise e interpretação dos resultados, o artefato é consolidado.

Após a apresentação do método de pesquisa, o sentido da consolidação do objetivo proposto para este estudo, faz-se necessário explicar o desenvolvimento do trabalho.

Com relação à **consciência do problema**, esta foi evidenciada por meio de pesquisas na literatura, e pela vivência do autor no mercado de trabalho, o que contribuiu amplamente para o desenvolver esta pesquisa. Notou-se que, à medida que novas tecnologias e ferramentas vão surgindo no mercado, há necessidade inerente de avaliar e reavaliar quais melhores tecnologias disponíveis, ou conjunto destas, para solucionar determinados problemas, com a premissa de sempre garantir melhor aproveitamento dos recursos. Nesse contexto, foi

identificada a necessidade de construção de um modelo de referência para guiar o processo de escolha de tecnologia para cada aplicação que compõem os componentes do STI-Lina.

Para isso, foi necessária a investigação na literatura para identificar alguns critérios gerais, sendo eles:

1. Compreensão dos Modelos que compõem STI-LINA;
2. Identificação dos Modelos responsáveis por captar e armazenamento de dados;
3. Identificação dos Modelos responsáveis recuperar e processar dados;
4. Compreensão dos Fundamentos de Big Data e critérios de escolha de arquitetura com foco em microserviços;
5. Ferramentas de Armazenamento em Big Data levando em consideração os tipo de dados;
6. Abordagem em cenários *bacth* e (*streaming*) para captura e processamento dados;

Sendo assim, o artefato é criado incluindo os modelos e métodos de Big Data como abordagem. Estes devem cobrir as características de BD identificadas na literatura para prover as funcionalidades adequadas. As entradas de dados utilizados são técnicas e tecnologias desenvolvidas para BD, como as melhores práticas presentes na literatura.

1.5 Organização da Tese

Este documento de qualificação está estruturado em quatro capítulos conforme dispostos a seguir:

- **Capítulo 1:** No Capítulo 1 foi apresentado a introdução do trabalho, os objetivos e desafios de pesquisa. Além disso, são apresentadas as hipóteses formuladas e, por fim, temos um seção destinada a apresentar os métodos utilizados na pesquisa.
- **Capítulo 2:** Apresenta, inicialmente, conceitos teóricos necessários para compreensão de Sistema Tutores Inteligente (STI) e entendimento do STI-LINA, que é o foco desta pesquisa. Também, são apresentados conceitos fundamentais sobre o termo Big Data e suas aplicações. Nessa mesma seção também são apresentados os conceitos relacionados a serviços de armazenamento de dados em nuvem, e sobre microserviços. Por fim, são apresentas os trabalhos correlatos a essa proposta de tese.
- **Capítulo 3:** Apresenta a abordagem proposta nessa tese, que consiste em uma proposta de arquitetura baseada em microserviços em nuvem focada a atender requisitos de Big Data para o STI-LINA.

- **Capítulo 4:** São explanadas as considerações finais deste trabalho por meio das contribuições, trabalhos futuros, e as contribuições bibliográficas.

Fundamentação Teórica

Este capítulo introduz os conceitos teóricos utilizados no desenvolvimento desta pesquisa.

A seção 2.1, aborda conceitos sobre Sistemas Tutores Inteligentes e apresenta o STI-LINA. Na sequência, na seção 2.2 são apresentados conceitos relacionados a Big Data buscando elucidar as tecnologias que o compõem. A seção 2.3 é dedicada ao big data na educação, e, por fim, na seção 2.4 são apresentados os trabalhos correlatos que norteiam esta pesquisa.

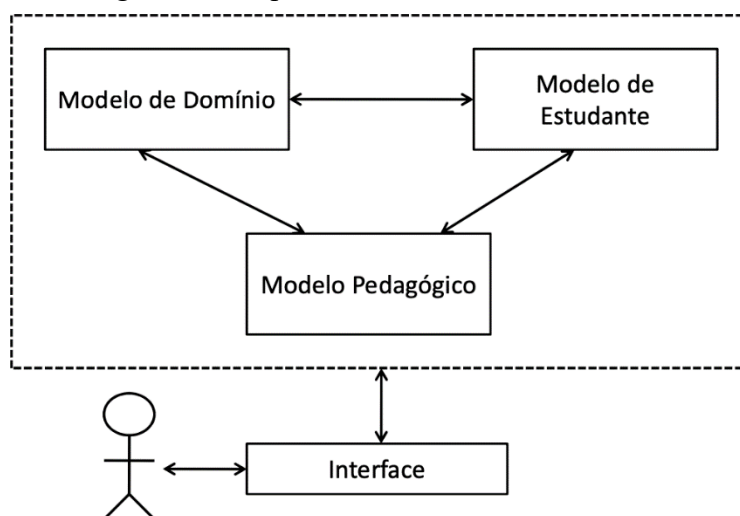
2.1 Sistema Tutores Inteligentes

Sistemas Tutores Inteligentes (STI) são aplicações educacionais que utilizam técnicas de Inteligência Artificial para fornecer instruções personalizadas ao estudante, assim proporcionando uma solução eficiente aos problemas de ensino e aprendizagem. A arquitetura clássica de um STI é composta por três modelos principais, sendo eles: **Modelo do Estudante** que apresenta as características individuais do aprendiz, o **Módulo Pedagógico**, que consiste em determinar o como ensinar, e **Modelo de Domínio**, que determina o que ensinar. Além destes modelos, também possui o ambiente de interação usuário/sistema, denominado Módulo Interface, responsável pela interação do sistema com o estudante. A arquitetura da Figura 2 ilustra a arquitetura clássica do STI.

O principal objetivo da modelagem do STI consiste em adaptar o conteúdo de aprendizagem de acordo com as preferências dos estudantes, estilos de aprendizagem, nível de conhecimento, dentre outras características especificadas no Modelo do Estudante (BRUSILOVSKY; MILLAN, 2007).

Segundo Karampiperis e Santos (2005), e Venugopalan, Srinath e Rodrigues (2016), alguns ambientes computacionais de aprendizagem possuem problemas relacionados com inconsistência nos conjuntos de regras do modelo de adaptação, resultando por vezes, em falhas conceituais na sequência dos objetos de aprendizagem oferecidos ao estudante. Sendo assim, os STI necessitam incluir métodos com a finalidade de efetuar a determinação e o gerenciamento dos conteúdos educacionais de forma eficaz para evitar caminhos únicos de aprendizagem (NAFEA; SIEWE; HE, 2018).

Figura 2 – Arquitetura Clássica de um STI.



Fonte: (CHRYSAFIADI; VIRVOU, 2015)

Para isso, recomenda-se apresentar uma sequência de materiais de aprendizagem, em ordem apropriada, correspondente às necessidades e ao conhecimento do estudante. O Sequenciamento Curricular Adaptativo (SCA) possibilita indicar um caminho de aprendizagem personalizado ao estudante. Entretanto, ao se utilizar o SCA, apresentam-se alguns problemas, dentre os quais destacam-se: ajustar o aprendizado conforme o conhecimento prévio do estudante, como também compreender uma sequência de conteúdo, por meio dos conceitos definidos, conforme o perfil e nível de habilidade de cada estudante (DWIVEDI et al., 2018, AQUINO et al., 2020).

Além da adaptação dos recursos educacionais, a construção de um ambiente propício à aprendizagem necessita integrar à arquitetura do STI informações referentes aos aspectos emocionais, motivacionais e pedagógicos (RODRIGUES; CARVALHO, 2005). Isto se deve ao fato de que as emoções interferem diretamente no processo de ensino e aprendizagem. Estudantes que se encontram em uma emoção prejudicial ao aprendizado, não conseguem assimilar, da melhor maneira possível, o conteúdo que lhes é proposto. No entanto, para Chabot et al (2005), estudantes que se encontram em uma emoção favorável ao aprendizado têm um maior interesse no conteúdo de aprendizagem, como também ocorre um aumento na memorização dos itens compreendidos.

A personalização de ambientes computacionais de aprendizagem está entre as áreas de pesquisa mais importantes na educação baseada em computador. Todavia, estes ambientes devem rastrear, com precisão, as atividades dos estudantes, monitorar suas características e elaborar intervenções e/ou estímulos conforme estratégias pedagógicas eficientes. Desse modo, além adaptação do material de aprendizagem, julga-se necessário fornecer instruções adaptadas ao estudante, considerando suas características individuais, respon-

sáveis por otimizar o ganho de aprendizagem (WOOLF, 2010; NARCISS et al., 2014).

2.1.1 Mapas de Conhecimento Estruturados

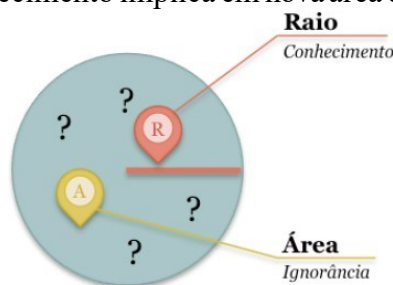
A Teoria das Categorias (EILENBERG; MACLANE, 1945) possibilita levantar questões sobre epistemologia e ontologias matemáticas. Sendo assim, é uma generalização da teoria dos conjuntos. A metodologia dos Mapas de Conhecimentos Estruturados (MCE) baseia-se matematicamente na Teoria das Categorias e Funtores. Assim, a estrutura dos MCE, conforme demonstrada por Filho (2013), é composta pelas seguintes definições:

- ❑ **Categorias:** indivíduos envolvidos nos processos (professor, estudante);
- ❑ **Objetos:** conceitos e definições existentes no domínio que deseja ensinar;
- ❑ **Morfismos:** métodos, ações, aplicações de como manipular os conceitos e definições que estão envolvidos no processo de transmissão do conhecimento em questão;
- ❑ **Funtores:** efetiva a adaptação dos objetos e morfismos preexistentes na categoria estudante para adquirir ou transformar um novo conhecimento.

A metodologia MCE propõe efetuar um mapeamento da categoria do professor na categoria do estudante, através do Funtor, com a finalidade de construção efetiva do conhecimento durante o processo de ensino e aprendizagem. Além disso, no processo de transmissão do conhecimento são preservadas as estruturas da categoria estudante. Nesta abordagem, os conceitos são palavras que não necessitam de ações para serem assimiladas. Os conhecimentos são os morfismos que são transmitidos como ações. Para implementar um Funtor entre duas categorias, deve-se cumprir a exigência mínima de que as duas categorias possuam objetos e morfismos equivalentes (FILHO, 2013).

O objetivo do Funtor é garantir a transmissão do conhecimento entre as categorias (professor e estudante). Para isto, o ideal é que o conhecimento inicial seja o mais básico possível. A transferência do conhecimento será realizada quando o estudante possuir uma subcategoria com objetos e morfismos mínimos, estabelecidos pelo professor, e uma base sólida para que o Funtor se estabeleça entre o estudante e o professor e, consequentemente, promova a efetivação do aprendizado (FERREIRA et al., 2014). Além disso, os MCE possuem uma ferramenta de avaliação, denominada Tabela de Avaliação Qualitativa (TAQ), responsável pela identificação dos pré-requisitos de objetos (conceitos e definições) e morfismos (conhecimentos) que possam existir na categoria estudante. Durante a transmissão do conhecimento pode haver dúvidas em relação aos conceitos e conhecimentos que são abordados, esta etapa é denominada área da ignorância, como ilustrada na Figura 3, onde o raio representa o conhecimento e a área representa as dúvidas que este raio desperta no estudante.

Figura 3 – Novo conhecimento implica em nova área de ignorância na estrutura dos MCE.



Fonte: Adaptado de Ferreira et al. (2014)

Na Figura 3 é possível observar que o raio de conhecimento determina o conteúdo a ser ensinado ao estudante. Este conteúdo exige conceitos que precisam ser adquiridos antes de aumentar o raio de conhecimento e avançar o conteúdo. As dúvidas do estudante são consideradas como pontos de ignorância que representa a ausência de determinado conhecimento ou conceito. Assim, faz-se necessário um levantamento sobre o nível de informação existente na categoria receptora do conteúdo (FERREIRA et al., 2014). Basicamente, na estrutura dos MCE, para cada novo conhecimento (estabelecido pelo Funtor) define-se os conceitos e ferramentas do aprendiz (objetos), bem como o domínio do uso de tais ferramentas básicas (morfismos). Se, na categoria do estudante, faltar um objeto (desconhecimento de conceitos) ou um morfismo (saber manipular os objetos), a aplicação do Funtor (transmissão do novo conhecimento) não terá sucesso (FILHO, 2013). Dessa maneira, ao utilizar os MCE, o estudante nunca avança no aprendizado de um novo conhecimento sem que antes não tenha aprendido o conhecimento precedente. A passagem do conhecimento, segundo Costa, N.S. et al. (2016), só será efetivada quando o estudante tiver, em uma subcategoria de si mesmo, os mesmos objetos e morfismos existentes na categoria professor, relacionada apenas ao que se deseja ensinar no momento. Satisfazendo tal exigência, um Funtor pode ser estabelecido entre as categorias, promovendo o mapeamento correto dos conhecimentos entre professor e estudante e, em seguida, a transferência de um novo conhecimento.

2.1.2 STI-LINA: Um Sistema Tutor Inteligente Adaptativo ao Curso e Personalizado ao Estudante

Esta seção irá apresentar o Sistema Tutor Inteligente, denominado STI-Lina, proposto por Melo (2021), para a recomendação dos recursos educacionais com base nas necessidades individuais de aprendizagem de cada estudante sobre determinado conteúdo. Além da personalização dos recursos de aprendizagem, o STI-Lina considera que as emoções interferem diretamente no processo cognitivo, de modo que um estudante que encontra-se em uma emoção prejudicial ao aprendizado não consegue assimilar adequadamente o

conteúdo proposto.

Sendo assim, para uma aprendizagem efetiva, não basta apenas recomendar o material de aprendizagem adequado, julga-se necessário observar as emoções do estudante durante o aprendizado e, posteriormente, fornecer instruções adaptadas ao seu perfil. Isto posto, nesta abordagem inclui-se o perfil afetivo do estudante e, por meio dessas informações, a definição das estratégias pedagógicas que possibilite a regulação das emoções negativas ao aprendizado. Além disso, técnicas de memorização do conteúdo também são consideradas. Comumente, conforme exposto por Raabe (2005), a modelagem de sistemas adaptativos para educação não considera a participação do professor, porém, ao inserir o professor como agente ativo é possível acrescentar abordagens pedagógicas personalizadas.

Com a finalidade de definir as regras do modelo de adaptação que aborda o problema de sequenciamento dos recursos educacionais, o STI-Lina propõe a integração do módulo/interface do professor em sua arquitetura.

Amparado pela metodologia dos Mapas de Conhecimentos Estruturados (MCE), descrita na Seção 2.1.1, o professor será responsável por verificar os itens da matriz curricular e determinar os conceitos (definições) e os conhecimentos (ações sobre o conceito) relacionados à compreensão de cada item. Este processo é definido como ramificação do ementa. Tendo em vista as teorias de Miller (1956) e Cowan (2001), para não comprometer o raciocínio e aprendizagem, esta ramificação do item não poderá ultrapassar o total de cinco informações, em sua complexidade.

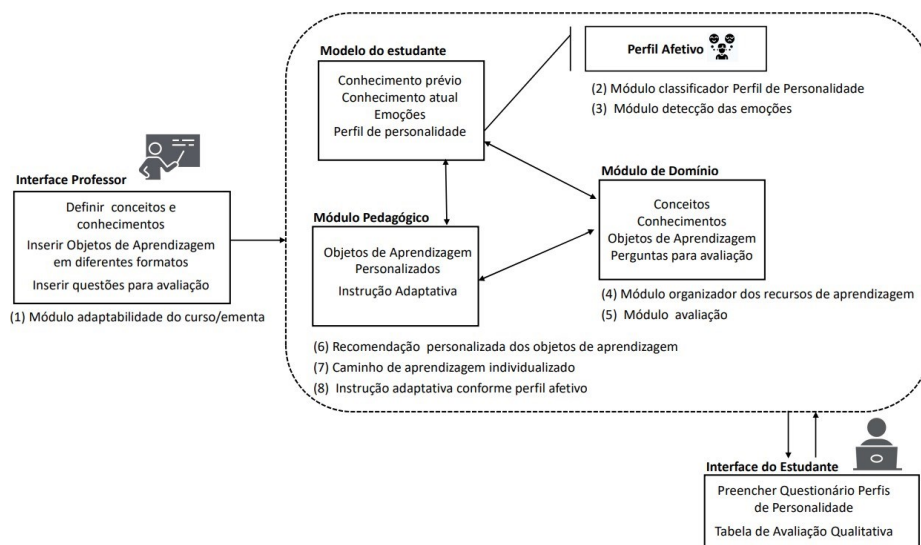
Na interface do estudante aqui proposta, seguindo os MCE e a sequência curricular preestabelecida na interface do professor, efetua-se o mapeamento das dúvidas de pré-requisitos de conceitos e pré-requisitos de conhecimentos. Este mapeamento é realizado pela Tabela de Avaliação Qualitativa (TAQ). Por meio da TAQ, inicializa-se o primeiro conhecimento (conteúdo do curso a ser ministrado) a partir da ignorância zero (ausência de dúvidas), identificada na categoria do estudante, aplicando o primeiro Funtor para mapear o primeiro conhecimento que será acrescentado à categoria estudante. Dessa forma, o STI manterá um equilíbrio estável avançando para um novo conhecimento, subsequente ao conhecimento anterior, apenas após eliminar novamente as ignorâncias (dúvidas) do estudante.

Assim sendo, o estudante receberá exclusivamente os recursos educacionais, em diferentes formatos, responsáveis por cobrir as dúvidas existentes, garantindo também que o estudante não consuma tempo desnecessário na procura de materiais, bem como na assimilação desnecessária de conteúdos naquele momento do curso. De modo geral, existem três principais Modelos, sendo eles:

- **Modelo do Estudante:** constituindo por informações relevantes do estudante que contribuem para um modelo de aprendizagem eficiente. É responsável por armazenar informações sobre o conhecimento prévio, conhecimento atual, o perfil de personalidade do estudante, preferências do estudante e estado emocional do estudante durante todo o processo cognitivo (emoções básicas).

- ❑ **Modelo de Domínio:** a organização e a estrutura são garantidas por meio da teoria dos MCE na qual serão armazenadas as relações de conceitos e conhecimentos para cada item, o qual é necessário determinar uma sequência adequada de conteúdos que certifiquem o sucesso no aprendizado. Todas essas informações são agrupadas no modelo do domínio, e contém todo o conhecimento do que deve ser ensinado e avaliado (MELO, 2021).
- ❑ **Modelo de Pedagógico:** é responsável por executar instruções com base nas informações consumidas dos Modelo do Estudante e no Módulo de Domínio. Nesta abordagem, as estratégias pedagógicas são adaptadas a cada perfil de personalidade considerando seus estados afetivos (emoções e personalidade), como também seu conhecimento sobre cada conteúdo. Caso a emoção do estudante esteja em um estado prejudicial ao seu aprendizado, o modelo pedagógico executará instruções adaptadas aos seus traços de personalidade, responsáveis por manter e/ou aproximar o estudante da emoção que estimule suas habilidades de aprendizagem. Além disso, o modelo pedagógico será responsável também pela personalização dos objetos a serem aprendidos, e pelo momento adequado da recomendação do conteúdo de aprendizagem, de acordo com as necessidades de aprendizado do estudante.

Figura 4 – Arquitetura STI-LINA.



Fonte: MELO, 2021.

A Figura 4, apresenta uma visão geral do Sistema Tutor Inteligente LINA (STI-LINA), com os seguintes módulos: (1) Interface do Professor; (2) Interface do Estudante; (3) Módulo gerenciador e organizador dos objetos de aprendizagem e conteúdos; (4) Módulo organizador dos itens de avaliação; (5) Módulo Classificador das Emoções (6) Módulo

classificador dos Perfis de Personalidade (7) Módulo recomendação personalizada de objetos de aprendizagem (8) Módulo adaptabilidade das instruções e estratégias pedagógicas para seu funcionamento efetivo.

Nota-se que a interface do professor, interface do estudante e, por fim, o perfil afetivo são responsáveis por prover a ingestão de dados no STI. Enquanto que, em 6, 7 e 8 são camadas de ações que o sistema irá executar, ou seja, elas utilizam as informações contidas Módulo de Domínio e Módulo de Estudante que serão processadas e irão direcionar um processo dentro do sistema.

2.2 O Big Data

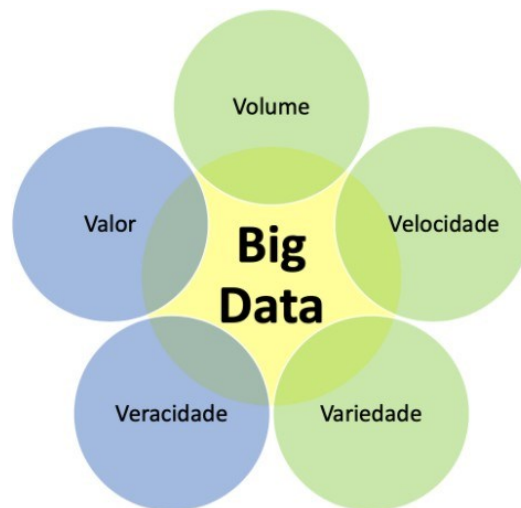
O Big data não é uma única tecnologia, mas uma combinação de tecnologias novas e antigas que possibilitam às organizações viabilizar ideias de aplicações. O termo refere-se à capacidade de administrar uma volumetria enorme de dados de diferentes formatos, na velocidade certa e dentro do prazo certo para permitir análise e reações em tempo real (HURWITZ et al., 2016). De acordo com Fan e Bifet (2013), as principais características podem ser resumidas em:

- ❑ **Volume:** representa uma grande quantidade de dados a ser coletada e analisada.
- ❑ **Variedade:** refere-se ao uso de diferentes estruturas de dados como dados estruturados, não estruturados, e semiestruturados.
- ❑ **Velocidade:** representa a velocidade com que os dados são processados.
- ❑ **Veracidade:** consiste na classificação em relação às fontes de dados de acordo com diferentes qualidades, como: cobertura, precisão e atualidades de dados fornecidos.
- ❑ **Valor:** Representa às influências que a informação terá no processo de tomada de decisão.

Ao se definir um significado para o termo Big Data, os autores Hurwitz et al (2016), e Fan e Bifet (2013), enfatizam a variedade dos tipos de dados, sendo estes classificados em: estruturados, semiestruturados e não estruturados, conforme as características apresentadas a seguir:

- ❑ **Dados Estruturados:** São dados que têm estrutura rígida previamente delineada com formato e comprimento definido. Como por exemplo, os dados advindos de *Enterprise Resource Planning* (ERPs), programas de gestão, históricos das últimas compras realizadas por um cliente, transações realizadas com cartões de crédito, entre outros.

Figura 5 – Características de Big Data.



Fonte: Adaptado de Hurwitz et al (2016).

- **Dados semiestruturados:** São dados que apresentam um alto grau de heterogeneidade e, dessa forma, pode-se concluir que estes não possuem uma estrutura fixa. Alguns exemplos desses dados são *Extensible Markup Language* (XML), conteúdo de e-mails, bem como outros elementos.
- **Dados Não Estruturados:** Esses dados não possuem necessariamente um formato ou sequência, não seguem regras e não são previsíveis. Estes dados também não são dados relacionados, mas sim diversificados. Alguns exemplos deste tipo de dados são: textos, vídeos, imagens, entre outros.

Conhecer a estrutura pela qual os dados se concretizam é importante no momento da coleta e armazenamento destes, para que uma busca posterior possa ser realizada de forma efetiva.

2.2.1 Áreas de aplicação do Big Data

O Big data está presente em uma série de áreas de negócios, entre elas na área da saúde em que se utiliza para a personalização de medicamentos, através de técnicas para extrair informações sobre grandes volumetrias de dados. Destaca-se também seu uso na área do Meio Ambiente, especificamente na captura de dados por meio de sensores. No setor público, vem sendo utilizado para criar algoritmos para evitar fraudes e erros tributários, entre outros. No quadro 1 são apresentados mais áreas de negócio e seus respectivos exemplos do uso de Big Data.

Quadro 1 – Áreas de aplicação do Big Data.

Área de Negócio	Exemplo de Aplicação
Saúde	Utilizado na personalização de medicamentos, entender as causas das doenças, usando técnicas para extrair valor de grandes quantidades de dados de histórico médico, medicamentos e fabricação de medicamentos. Faz uso de dados não estruturados, como imagens médicas.
Ambiente	Utilizado para encontrar correlação entre os valores medidos e as implicações para o meio ambiente, por meio da coleta de dados de vários sensores (por exemplo, qualidade do ar e da água, metrologia e emissões de gás).
Setor público	Utilizar Big Data para prevenir fraudes e erros tributários; Em ações segmentando a população; Criar transparência por meio da disponibilidade de dados.
Comércio	Previsão de eventos e segmentação de clientes, Criando produtos personalizados; Marketing baseado em localização, análise de sentimento e venda cruzada; Otimização de logística, entre outros.
Manufatura	Previsão de demanda para planejamento de abastecimento; Uso de sensores em produtos manufaturados para oferecer manutenção proativa.
Vida e ciência	Análise das variações genéticas utilizando uma grande quantidade de dados.

Fonte: Adaptado de: Costa (2019).

O Big Data traz vantagens competitivas para as organizações, conforme exemplificado anteriormente no Quadro 1, no entanto, existem características particulares das quais o definem. Sendo assim, na próxima seção, serão abordados alguns desafios dos quais são enfrentados quando se almeja a implementação de projetos de Big Data.

2.2.2 Desafios da descoberta de conhecimentos por fases

De acordo com Chen e Zhang (2014) podemos verificar alguns desafios trazidos pela uso do Big Data com relação a:

- ❑ **Captura e armazenamento de dados:** a capacidade de armazenar dados não cresce na mesma velocidade que a produção de dados exige, desse modo, ocorre muitas das vezes o descarte de muitos dados que custarão caro para serem gerados e obtidos (CHEN; ZHANG, 2014).
- ❑ **Transmissão dos dados:** a popularização do armazenamento e do processamento distribuído em nuvem esbarra na largura limitada dos *links* de transmissão de dados, ao mesmo tempo em que o armazenamento de dados levanta questões relacionadas à segurança dos dados e à integração (CHEN; ZHANG, 2014).
- ❑ **Curadoria de dados:** manter os dados implica armazenamento e recuperação, garantia da qualidade, reuso e preservação durante determinado tempo. Os modelos de bancos de dados relacionais lidam com dados estruturados armazenados num esquema e recuperados por um modelo relacional. Os dados devem ser pré-processados, transformados e catalogados para serem armazenados no modelo relacional. As abordagens de *Data Warehouses* e *Data Marts* são as maneiras com as quais os bancos de dados relacionais lidam com grandes volumes de dados, não comparáveis

aos volumes de dados dos problemas Big Data. Novas arquiteturas de bancos de dados são requeridas para prover curadoria dos dados com desempenho adequado (CHEN; ZHANG, 2014).

- **Análise de dados:** O grande volume de dados dos problemas Big Data indica que o maior desafio nesse campo é a escalabilidade. Esforços têm sido feitos no desenvolvimento de novos algoritmos de análise de grandes volumes de dados. O aumento do número de núcleos dos processadores viabilizou o processamento paralelo, e a computação paralela em nuvem surgiu como proposta viável para a demanda por processamento em tempo real. Esta mesma arquitetura distribuída da nuvem levanta questões de segurança, privacidade, direitos autorais e adequação a legislações locais (CHEN; ZHANG, 2014).
- **Visualização de dados:** O volume e a alta dimensionalidade dos dados em problemas Big Data requerem ferramentas de visualização com desempenho adequado, que sejam escaláveis e que tenham um bom tempo de resposta. A presença de incertezas nos dados requer modelos de análise e de visualização adequados (CHEN; ZHANG, 2014).

Com base na percepção dos desafios que foram apresentados, é necessário o uso de tecnologias adequadas para resolver tais problemas. Nesse sentido, a próxima seção irá apresentar um panorama de tecnologias utilizando o ecossistema apache Hadoop.

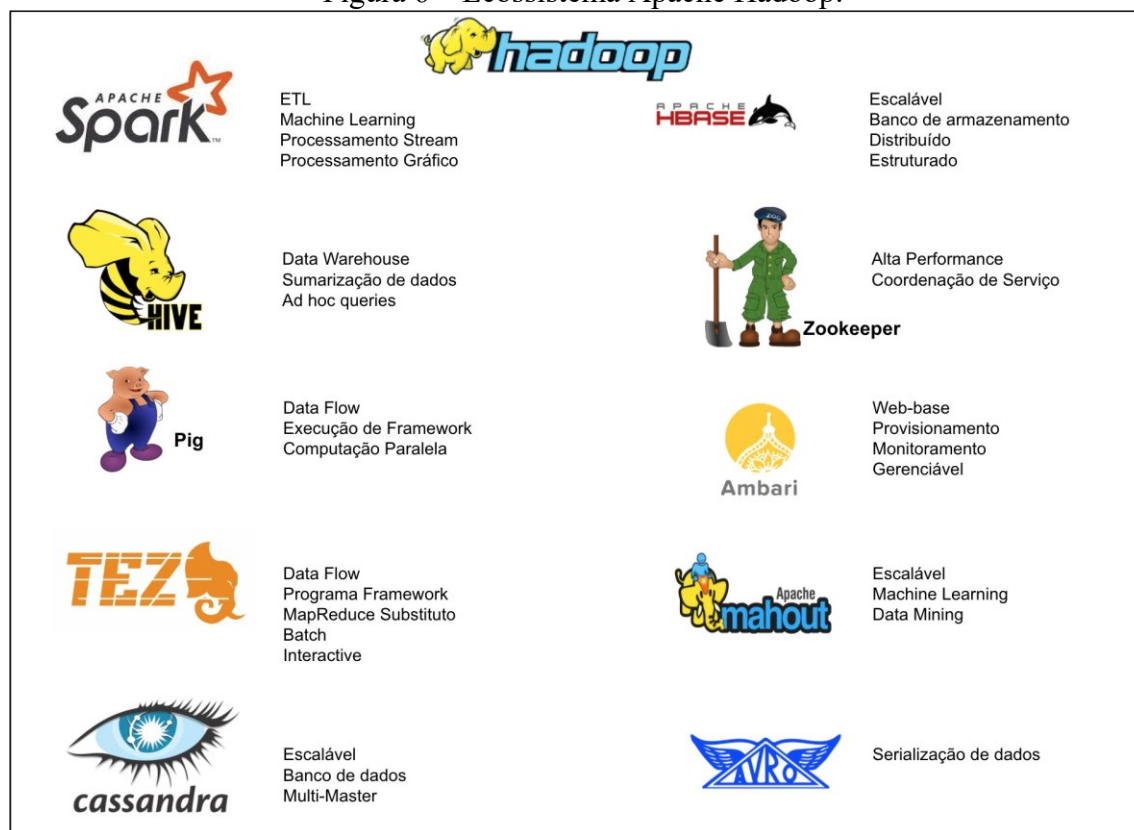
2.2.3 Tecnologias de Big Data

O Hadoop é um sistema de arquivos distribuído para armazenar e processar grande volume de dados por meio de MapReduce, sendo ele um projeto da Apache de código aberto (GHEMAWAT; GOBIOFF; LEUNG, 2003). Ele é composto por dois componentes principais: o *Hadoop Distributed File System* (HDFS) e uma estrutura de processamento distribuído chamada Hadoop MapReduce.

O Hadoop possui a capacidade para armazenar e processar grandes quantidades de dados, por meio da distribuição do armazenamento e processamento em um cluster escalonável, de vários nós, construídos por meio de hardware comum. No HDFS, os arquivos são divididos em blocos distribuídos e replicados entre nós do cluster. Ele garante preencher certos requisitos, como a tolerância a falhas e disponibilidade. O Hadoop MapReduce é um modelo de programação e um mecanismo de execução para processamento de grandes conjuntos de dados armazenados no HDFS, com base no método dividir e conquistar, dividindo um problema complexo em muitos problemas mais simples e, em seguida, combinando cada solução mais simples em um solução geral para o problema principal.

Em relação ao HDFS, existem dois tipos de nós no cluster: um *NameNode*, que é responsável por armazenar metadados sobre blocos e nós; e um *DataNode*, que armazena

Figura 6 – Ecossistema Apache Hadoop.



Fonte: Adaptado de Costa e Santos (2017).

blocos de dados. Em relação ao Hadoop MapReduce, também existem dois tipos de nós, a saber, um *JobTracker* que agenda trabalhos e distribui tarefas entre escravos chamados *TaskTrackers* (BENGFORT; KIM, 2019).

Com o passar do tempo, o Hadoop apresentou uma evolução considerável, incluindo a transição do MapReduce para o *Yet Another Resource Negotiator* (YARN). O YARN revê os componentes *JobTracker* e *TaskTracker* e os substitui pelo *ResourceManager*, um *NodeManager* e um *Application Master*, com a finalidade de resolver alguns problemas em Hadoop MapReduce, como escalabilidade em grandes clusters ou suporte para programação alternativa (GOLDMAN et al., 2012). Além disso, o Hadoop tem vários projetos relacionados, como a Figura 6 demonstra, também destacando seus principais recursos.

2.2.4 Critérios para escolha de uma arquitetura de Big Data

Para projetos relacionados ao Big Data, a escolha pela arquitetura é essencial. Para isso, Singh e Reddy (2015) elenca alguns questionamentos importantes para esta finalidade. Sendo estes:

1. Qual é a velocidade na qual os resultados devem ser obtidos?

2. Qual a volumetria dos dados que serão processados?

3. Qual a quantidade de iterações terá o modelo?

Outro fator importante que deve ser considerado é em relação aos requisitos de sistema e de plataformas, sendo estes:

4. Haverá necessidade de aumento da capacidade de processamento?

5. A taxa de transferência de dados é crítica para o sistema?

No trabalho de Singh e Reddy (2015), discute-se também o escalonamento de tecnologias atualmente utilizadas para o processamento Big Data, classificando-as em tecnologias de escalabilidade horizontal e tecnologias de escalonamento vertical. O escalonamento define o quanto um sistema consegue adaptar-se ao crescimento da demanda de processamento, que é implementado horizontalmente, pela distribuição de carga de processamento entre máquina de baixo custo agrupadas, ou verticalmente, pelo aumento da capacidade de processamento em um único servidor, seja pelo acréscimo de processadores, de memória, ou por *hardware* mais rápido.

Algumas tecnologias e sistemas são apresentadas pelos referidos autores e avaliadas de acordo com fatores, tais como: a volumetria dos dados, otimização de velocidade ou de carga de trabalho (taxa de transferência) e treinamento e aplicação de um modelo.

1. **Volumetria dos dados:** considerando como um dos mais importantes fatores. Se o dado couber na memória do sistema, então poderá ser processado por uma única máquina. Plataformas como GPU ou CPUs de vários núcleos podem ser utilizados para o aumento da velocidade neste caso. Caso a quantidade de dados não caiba na memória do sistema, então, serão necessários grupos de máquinas, e soluções como Apache Hadoop ou Apache Spark (SINGH; REDDY, 2015).
2. **Otimização de velocidade ou carga de trabalho:** a velocidade é entendida como a capacidade de processar dados em tempo real e carga de trabalho é entendida como a quantidade de dados que o sistema é capaz de manipular e processar simultaneamente. Se houver grandes quantidades de dados sem restrições de tempo de processamento, podem ser aplicadas soluções de escalonamento horizontal, mas se a necessidade for de processamento rápido, são aplicadas soluções de processamento em tempo real por meio de escalonamento vertical. (SINGH; REDDY, 2015).
3. **Treinamento e aplicação de um modelo:** um modelo de análise de dados é tipicamente treinado *off-line*, e este processo toma um tempo considerável. A quantidade de dados necessária para o treinamento é grande, mas não há restrições de tempo latentes quando comparamos com o processamento *on-line*. O modelo treinado é

aplicado em tempo real em Big Data, e os resultados devem ser obtidos rapidamente. Estas diferenças entre treinamento e aplicação de um modelo sugerem a utilização de plataformas diferentes: a plataforma de treinamento pode ser escalada horizontalmente, e a plataforma de aplicação, por sua vez, escalonada verticalmente (SINGH; REDDY, 2015).

2.3 Big Data na Educação

O Big Data no contexto educacional pode ser utilizado para a criação de plataformas educacionais nas quais personalize as experiências dos agentes que interagem neste sistema, sendo eles: alunos, professores ou tutores. Essa personalização pode levar em consideração as necessidades individuais desses atores no sistema. Uma das áreas de conhecimento que se dedica à análise dos dados produzidos nestas interações é Mineração de dados Educacionais, conhecida como *Educational Data Mining* (EDM).

2.3.1 Mineração de dados Educacionais (EDM)

O conceito de EDM consiste em compreender o comportamento dos estudantes bem como o ambiente na qual a aprendizagem ocorre, com a finalidade de melhorar a aprendizagem. O foco pode ser no aprimoramento da qualidade do ensino, ou ainda otimizando a absorção do conhecimento pelo aluno (ROMERO; VENTURA; GARCÍA, 2008). Há três fases do EDM, a saber: pré-processamento, mineração e pós-processamento.

Segundo Romero, Ventura e Garcia (2008), as características do EDM incluem a busca por melhorar o processo de aprendizagem ao realizar o processamento de dados educacionais, o que possibilita conhecer melhor os fenômenos da aprendizagem, nos quais estes podem conter semânticas intrínsecas, relacionamentos diversos, bem como hierarquia em múltiplos níveis. Segundo o mesmo autor, na literatura, o EDM tem sido aplicada para diversas tarefas no contexto educacional tais como: 1) Análise e Visualização de dados; 2) Informações para apoiar educadores; 3) Recomendação para estudantes; 4) Predição de desempenho de estudantes; 5) Modelagem de estudantes; 6) Detecção de comportamentos indesejáveis; 7) Agrupamento de estudantes; 8) Análise de redes de sociais; 9) Desenvolvimento de mapas conceituais; 10) Construção de conteúdo programático e 11) Planejamento e agendamento de curso (ROMERO; VENTURA; GARCÍA, 2008).

As organizações podem ser orientadas por dados, e podem explorar o potencial de inúmeras fontes, sem lidar com técnicas e tecnologias de Big Data, como apresentado anteriormente. No entanto, considerando o uso de várias aplicações sobre diferentes formatos de dados, e a crescente volumetria destes no presente contexto educacional, exige-se a criação de uma mudança de paradigma na forma com a qual coletamos, armazenamos, recuperamos, processamos e analisamos dados que vai ao encontro de conceitos e características de BigData.

Nesse sentido, pode-se elencar alguns dos benefícios do uso de Big data em contexto educacional. Morabito (2015) elenca esforços em Big Data com a finalidade de prover análises de dados na perspectiva do aluno. Sendo elas:

1. **A descoberta e avanços na ciência da aprendizagem:** um exemplo é a utilização para o rastreamento de padrões de aprendizagem durante o processo de ensino e aprendizagem. O que vai ao encontro dos estudos de Romero, Ventua e Garcia (2008) e também na proposta de modelagem do estudante, realizada por Melo (2021) por meio da implementação do STI-LINA.
2. **A avaliação mais imparcial do aluno:** nesse caso a ideia é considerar as dimensões sociais, cognitivas, emocionais e meta cognitivas (MORABITO, 2015). Também propõem considerar os níveis de grupo e de indivíduos. Além disso, também considera fatores para recomendação de material pedagógico, sendo algo também proposto por Melo (2021) que, além dessa premissa, estende o conceito ao considerar, na recomendação, a aplicação baseada no raio de conhecimento do qual o aluno encontra-se no momento da avaliação.
3. **O acompanhamento das atividades em tempo real:** dar melhor suporte ao aluno, uma vez de posse do desempenho do mesmo durante a realização de atividades avaliativas (MORABITO, 2015). O sistema deve traçar um plano de estudos adaptado ou personalizado (produção de cursos mais personalizados), de acordo com as necessidades do aluno. Algo que também foi proposto por Melo (2021) e realizado considerando os Teoria dos Mapas de Conhecimentos Estruturados (MCEs).

2.3.2 Big Data e Armazenamento de dados

Ao abordar o conceitos de Big Data é de extrema importância compreender os mecanismos de banco de dados utilizados para manter os dados íntegros e de fácil recuperação. Nesse sentido, os mecanismos precisam ser rápidos, escaláveis e sólidos. Em ambiente de Big Data, embora seja possível utilizar apenas um sistema de banco de dados (SGDB) para todas as implementações, esta não é uma prática popularmente adotada, devido ao baixo desempenho, à escalabilidade, e até mesmo ao alto custo de manutenção. (HURWITZ et al.,2016).

No mercado de dados há diferentes sistemas disponíveis, e deve-se, primeiramente, compreender as características de aplicação a serem implementadas, além de entender com que tipo de dados deve ser persistido, como o mesmo pode ser acessado e atualizado, e também como utilizar tais dados para tomada de decisão.

Assim, à medida que se concebe a arquitetura para suportar uma aplicação, depara-se com a necessidade de escolher entre tecnologias de bancos de dados relacionais ou não relacionais. O Quadro 2 propõe uma breve explicação sobre as diferenças entre elas, e apresenta alguns cenários de uso que podem guiar a escolha sobre o tipo de tecnologia.

Quadro 2 – Cenários de Uso para Tecnologias Relacionais e NoSQL.

Casos de Uso	Relacional	NoSQL
Consultas e Relatório Complexos	x	
Tabela com esquema fixo	x	
Repositório de documento		x
Integridade Referencial	x	
Milhares de transações por segundo	x	x
Consistência, disponibilidade e desempenho	x	x
Replicação de dados assíncrona	x	x
Estruturas definidas	x	
Armazenamento de logs e estado da aplicação		x

Fonte: Elaborado pelo autor.

O modelo relacional é composto basicamente por tabela e colunas e os relacionamentos entre entidades. Já o modelo Not Only SQL (NoSQL) é um modelo mais simples, que não possui tantas entidades, e é voltado para o alto desempenho, não adotando SQL como linguagem padrão, apesar de que alguns bancos de dados, como o Cassandra, utilizar a linguagem *CQL*, semelhante, ao SQL, embora não possua todas as funcionalidades deste. O foco dessa tecnologia é a escalabilidade e alta performance. O Quadro 2 apresenta alguns casos de uso que direcionam para a escolha mais assertiva da tecnologia.

Com base no exposto é possível concluir que, ao construir um repositório de documentos ou armazenamento de logs de aplicação, necessariamente as tecnologias NoSQL têm melhor desempenho na recuperação, na facilidade de manutenção e implementação de modelo e, posto isso, é sempre a melhor escolha entre soluções de armazenamento para posterior recuperação de dados. Quando se trabalha com consulta e relatórios complexos, tabelas com esquema fixo, integridade referencial e estruturas definidas, as tecnologias relacionais são sempre consideradas a melhor escolha. Já quando temos consistência, disponibilidade e desempenho, milhares de transações por segundo e replicação de dados assíncrona, ambas as tecnologias, NoSQL e Relacionais, tendem a atender bem a estes quesitos. Nesse caso, é necessário entender melhor o contexto de aplicação e suas particularidades, para uma escolha mais assertiva. Sendo assim, faz-se necessário traçar um direcionamento para melhor escolha entre tecnologias relacional e não relacional, baseado nas características presentes nas aplicações, como carga de trabalho, *schema* de dados, performance e escalabilidade. Estas características serão elucidadas de maneira mais aprofundada a seguir.

- ❑ **Carga de trabalho:** está relacionada com a quantidade de transações que serão realizadas na aplicação. Em aplicações que trabalham com um alto volume de transações por segundo, tanto em ambiente transacional (OLTP) quanto analítico (OLAP), a característica mais marcante desses modelos é a consistência e integridade dos dados, nesse caso, o uso de tecnologias relacionais são mais adequadas. Já para aplicações de baixa latência na qual

cria-se uma tabela com base em uma consulta almejada, consulta planejada e não *adhoc*, o ideal é o uso de banco NoSQL.

- **Schema:** O tipo relacional é coposto por linhas e colunas, e pelo relacionamento entre entidades. Já o NoSQL, modelo mais aberto e flexível, pode ser do tipo chave e valor, documento ou grafo, o que o torna diferente do tipo relacional. Importante notar também que as propriedades ACID¹ são aderentes ao modelo relacional, enquanto que, no NoSQL, é parcialmente aderente. A consistência eventual é o que permite uma alta disponibilidade, já que, neste caso, tem-se as réplicas, e não necessariamente as transações precisam esperar para que todas as réplicas estejam atualizadas. Uma vez gravada a primeira transação, assincronamente esta irá refletir na atualização das demais réplicas. Sendo assim, a consistência é preferivelmente eventual. Em tecnologias como o DynamoDB é possível atribuir uma consistência forte, mas pode-se perder em disponibilidade.
- **Performance:** para o modelo relacional, quando se propõe melhorar a performance, - uma vez que o acesso aos dados é feito diretamente em disco -, utiliza-se, como estratégia, a criação de índices e de particionamentos de tabelas. Já no NoSQL, desde que o modelo tenha sido bem construído (escolha de chaves bem definidas) o ganho de performance é obtido com o aumento do *cluster*.
- **Escalabilidade:** no modelo relacional, quando há necessidade de escalar, é alocado mais memória e CPU e, em alguns casos, cria-se réplicas de leitura. Já no NoSQL, basta adicionar mais nós.
- **API:** no modelo relacional as consultas são compatíveis com *Structured Query Language* (SQL), e no NoSQL, pensamos em objeto e chave de partições de acesso.

Com base nos que foi exposto acima, a Figura 7 ilustra um resumo sobre o comparativo entre as opções disponíveis.

Esses cenários de uso são guias que auxiliam na tomada de decisão de qual banco utilizar, baseados nas peculiaridades de aplicação. Porém, é importante salientar que nenhum caso de uso substitui a avaliação da arquitetura e a engenharia do projeto investigado.

2.3.3 Big Data e Definição de Tecnologias de Armazenamento em Nuvem

O aumento do volume e da complexidade dos dados que são gerados atualmente pressupõe que seja feita a escolha assertiva das melhores tecnologias para armazenamento de dados, com a finalidade de auxiliar o contexto analítico e a evolução de produtos e serviços,

¹ ACID: Atomicidade, Consistência, Isolamento e Durabilidade

Figura 7 – Comparativo entre as opções disponíveis.

Diferenças	Relacionais	NoSQL
Cargas de trabalho ideais	Aplicativos transacionais e fortemente consistentes de processamento de transações online (OLTP) ou processamento analítico online (OLAP)	Aplicativos de baixa latência. Os bancos de dados NoSQL são projetados para consulta de dados semiestruturados.
Schema	Tabelas compostas por linhas e colunas, esquemas, relações entre tabelas e outros elementos de banco de dados.	Variedade de modelo de dados, como chave-valor, documento e grafos.
*ACID	Aderente	Parcialmente Aderente
Performance	A performance normalmente depende do subsistema do disco. A otimização de consultas, índices e estrutura de tabela é necessária para alcançar máxima performance	A performance geralmente é uma função do tamanho do cluster do hardware subjacente, da latência de rede e do aplicativo que faz a chamada.
Escala	Escala verticalmente o tamanho ao aumentar os recursos de computação do hardware, ou escalam horizontalmente o tamanho ao adicionar réplicas para cargas de trabalho somente leitura.	Escalabilidade horizontalmente o tamanho usando arquitetura distribuída para aumentar a taxa de transferência que fornece performance consistente em escala quase limitada
API	Consultas compatíveis com uma SQL	Baseadas em objeto - chave de partição

Fonte: Adaptado de Hurwitz et al. (2016).

bem como contribuir para melhorar a integração entre os grandes volumes de dados. Assim sendo, nota-se que novas tecnologias vêm surgindo para suprir as necessidades, sendo elas cada vez mais efetivas e seguras. Nesse contexto, encontram-se alternativas tanto gratuitas quanto pagas, cada uma com suas particularidades, vantagens e limitações. Nesta seção, iremos explorar as tecnologias disponíveis na plataforma Cloud. Atualmente, temos vários serviços disponíveis no mercado, sendo ofertados por empresas como AWS, Azure, GCP e Cloudera. Para esse trabalho optou-se pela escolha de serviços ofertados pela AWS, em ambiente Cloud Pública e Cloud Privada. Inicialmente, para fazer a definição de uma tecnologia de banco de dados para cada aplicação, foi necessário conhecer qual o tipo de dado do qual cada uma destas aplicações farão uso. Partimos primeiramente dos tipos de dados: se ele é estruturado, semi-estruturado ou não estruturado, como já detalhamos na seção 2.2.

Como apresentado na Figura 8, para os dados estruturados, temos opções de bancos de dados relacionais, que tem como características a integridade referencial, que é o ponto forte desse tipo de tecnologia. Nesse caso, algo que deve-se ter em mente para construir um projeto é: (1) há necessidade de integridade referencial? (2) há necessidade de garantir a consistência dos dados? Esse tipo de banco também possui características como o suporte para o particionamento de tabelas, a criação de réplicas de leitura, entre outras opções para propósito geral de armazenamento de dados e informações. Dentro das tecnologias em Cloud Pública tem-se os serviços do RDS e do Aurora.

- O RDS é um serviço web que facilita a configuração, operação e dimensionamento de um banco de dados relacional na nuvem AWS. Ele fornece capacidade econômica e redimensionável para um banco de dados relacional padrão do setor (AMAZON, 2022i). O RDS possui suporte aos SGBDs MySQL, Postgres, SQL Server e Oracle, apesar de que as escolhas costumam ser voltadas mais para o MySQL e Postgres por motivos

Figura 8 – Escolha de Serviços AWS de acordo tipo de dados a serem manipulados.

Tipo de dados	Caso de uso	Características Gerais	Ferramentas / Serviços
			Cloud Pública (AWS)
Estruturado	Transacional / ACID (OLTP)	* Suporte limitado a particionamento horizontal (sharding) * Integridade Referencial * Banco de dados de propósito geral	RDS ou Aurora
	Análítico (OLAP)	* Ferramentas de Relatórios	Redshift
Semi-Estruturado	Chave-Valor / Documento / Hierárquico / Schemaless	* Schemaless, apenas PrimaryKey obrigatória ** * Padrão de acesso conhecido * Suporte a particionamento (sharding) * Time to Live (expiração de registros) * Suporte a Eventos de Mudanças-Streams * Consistência de Leitura Flexível	DynamoDB
	Cache, Chave-Valor	* Time to Live (expiração de registros / chaves)	ElasticCache
	2D Key-Value / Wide Column	* Padrão de acesso conhecido * Suporte a particionamento (sharding) * Modelo otimizado para consulta * Time to live (expiração de registros) * Consistência de Leitura Flexível	KyeSpaces
	Relacionamento entre Entidades / Grafos	* Ligações em Rede * Detecção de Fraude / Rede Social / Recomendações * Suporte a Eventos de Mudanças - Streams	Netune **
	Full Text Search	* Permite expandir de capacidade de pesquisa de aplicações	ElasticSearch
Não Estruturado	Arquivos Multimídia	* Possibilidade de complementar informações de registros Large Objects	S3

Fonte: Adaptado de Amazon (2022n).

relacionados aos custos menores de aquisição dessas tecnologias (SONI; SONI, 2021).

- Aurora é um banco de dados relacional compatível com MySQL e PostgreSQL, criado para a nuvem, e que combina a performance e a disponibilidade de banco de dados empresariais tradicionais com a simplicidade e a economia de bancos de dados de código aberto. O Aurora é utilizado quando o projeto necessita de uma capacidade maior de processamento. Esse está disponível nas versões do PostgreSQL e do MySQL. Já na Cloud Privada tem-se a opções do SQL Server, Oracle e MySQL. (AMAZON, 2022c).

Para o banco relacional que têm integridade, mas as buscas nos dados são mais voltadas a relatórios e consultas mais complexas (*ad hoc*) há outras opções como Redshift (AMA- ZON, 2022j) e Teradata (TERADATA, 2022) disponíveis em Cloud Pública e Privada, respectivamente.

Para dados que são semiestruturados, temos um leque com mais opções. Nesse caso, há necessidade de maior detalhamento sobre o tipo de dado e sua usabilidade para a aplicação. Caso esses dados sejam Chave-Valor, Documento, Hierárquico ou *Schemaless*, uma opção seria a escolha por uma tecnologia de banco como o DynamoDB ² utilizado quando há um padrão de acesso conhecido e queries bem definidas. Nesse tipo de tecnologia, há peculiaridades, como o particionamento horizontal, que irá possibilitar o crescimento em nós e em partições, para melhorar a performance das consultas sendo possível crescer sem preciar

² DynamoDB é um banco de dados de chave-valor e documentos que oferecem desempenho em milise- gundos de um dígito em qualquer escala. É um banco de dados totalmente gerenciado, multi-regional, e durável com segurança, para aplicativos em escala de Internet (AMAZON, 2022e).

³ Redis é um banco de chave-valor de código aberto construído para funcionar em memória. Utilizado usado como banco de dados, cache, agente de mensagens e fila. (AMAZON, 2022h).

alterar a estrutura de dados, além de contar com uma funcionalidade de tempo, denominada *Time to Live* (TTL) para expiração dos registros de forma nativa (AMAZON, 2022e). Além disso, há o suporte a mudanças *streams*, na qual consegue constatar qual era a imagem do dado antes e depois de um processo de atualização. Além disso, a consistência neste caso é flexível, o que significa que não necessariamente terá aquele dado idêntico naquele milissegundo em todas as réplicas. Nesse cenário, é possível ainda atribuir mais consistência, porém faz com que haja a perda na disponibilidade de dados, necessitando esperar mais tempo para que esses dados estejam iguais em todas as réplicas antes de devolver para a aplicação requisitante (AMAZON, 2022e). Vale notar que, nesse tipo de tecnologia, não há a integridade referencial, a qual está presente no cenário relacional. No trabalho de Kalid et al. (2017), são apresentados detalhes e a comparação do DynamoDB com outros bancos, como o Cassandra.

Para dados do tipo chave e valor em memória, ou seja, cache, há o *Redis*³ como opção de tecnologias, como *ElastiCache*, na Cloud Pública e na Cloud Privada. No trabalho de Pelle et al. (2019) são apresentados estudos com bases nestes bancos e suas performance em ambiente Cloud.

Para a manipulação de dados do tipo *Key-Value / Wide Column* é possível ter um modelo de dados no qual tem-se uma tabela dentro de outra tabela, ou seja, uma coluna pode ser uma referência para uma tabela composta de várias informações. Este tipo de tecnologia é útil quando há um modelo voltado para consultas planejadas. É possível ter um modelo otimizado para consulta, similar ao existente no DynamoDB, com características específicas, como o crescimento em número de nós para ganho de performance, TTL e a consistência flexível. Como opções de tecnologias há, na Cloud Pública, o Keyspaces, e na Cloud Privada, o Cassandra. Porém, para os dados do tipo semiestruturados, que são voltados para pesquisa, ou seja, para aumentar a capacidade de pesquisa de aplicações, há como opção o ElasticSearch (AMAZON, 2022d).

Para armazenamento de arquivos grandes e do tipo multimídias (texto, imagens, vídeos) o ideal é o uso do S3. Este é um serviço que disponibiliza uma interface *web* para guardar e recuperar qualquer quantidade de dados de dados, em qualquer lugar. É do tipo *Object Storage*, no qual o objeto armazenado pode ter de 0 a 5TB. Os arquivos são armazenados em *buckets*, o que é basicamente representado por pasta de partição na AWS.(AMAZON, 2021).

A Figura 8, apresenta um quadro que sumariza os conceitos apresentados sobre tipos de dados e características de gerias de aplicação para guiar na escolha de um serviços de nuvem mais adequando no que refere-se a armazenamento de dados, uma vez que saber quais dados estão armazenados, e onde estão, são fundamentais para a implementação de projetos de Big Data.

2.3.4 Arquitetura de Microserviços

A maioria dos sistemas são frequentemente implementados sob um modelo de arquitetura monolítica devido a sua popularidade no mercado (JÚNIOR, 2017). Nesta abordagem, tanto o banco de dados quanto o servidor web e o servidor de aplicativos são instalados em um mesmo servidor, já em outras abordagens, o banco de dados está uma instância e o servidor web (aplicação) em outra, desencadeando desafios no que refere-se à disponibilidade e escalabilidade. Desse modo, uma maneira de resolver esses problemas é o dimensionamento vertical, ou seja, aumentar a capacidade dos recursos (memória, disco e CPU) na mesma instância, o que pode acarretar em altos custos.

Em uma arquitetura de microserviços, é o conjunto de aplicações menores e independentes que, juntas, formam uma aplicação completa, ideal para aplicações que necessitam de um alto escalonamento, e facilidade na integração dos dados (JÚNIOR, 2017). A separação permite que cada componente da aplicação seja executado em próprio processo e até mesmo em computadores diferentes. Caso determinado serviço venha a falhar, apenas aquele serviço em específico é afetado (GOS; ZABIEROWSKI, 2020). Assim, observa-se que cada serviço tem sua independência para escolher sua própria tecnologia. Além disso, o desacoplamento em diferentes serviços permite que a cada um destes sejam atribuídos os recursos que realmente necessitam.(NEWMAN, 2021).

2.4 Trabalhos Correlatos

Esta seção descreve os principais trabalhos encontrados na literatura e que serviram de base para o levantamento das hipóteses de pesquisa propostas neste trabalho. Além disso, também foram selecionados trabalhos que utilizam conceitos de Big Data para a implementação de aplicações para trabalhar com grandes volumetrias de dados.

No trabalho de Jahn (2017) é apresentada uma proposta de arquitetura genérica para processamento de dados em larga escala, de forma paralela, utilizando o ecossistema Hadoop Apache. O trabalho apresenta ainda aspectos fundamentais sobre uso de dados em disco e de memória, e como deve ser utilizado para melhor performance de processamento. Além disso, o trabalho contribui com uma visão detalhada do funcionamento dos bancos de dados relacionais e NoSQL, sendo este último explorado de forma detalhada, no qual o autor apresenta soluções de mercado para diferentes aplicações. Vale ressaltar que não é apresentanda uma comparação com outras tecnologias disponíveis no mercado, demonstrando pontos que levam à escolha de determinadas tecnologias em detrimento de outras. Além disso, o trabalho também não discute aspectos essenciais relacionados à escalabilidade horizontal e vertical, que podem ser decisivos para a escolha da arquitetura a ser implementada.

Pääkkönen (2016) explora possíveis arquiteturas de Big Data como referência para facilitar a implementação de projetos e a seleção de tecnologias ou soluções comerciais para a construção de sistemas robustos em termos de volumetrias de dados.

No trabalho, é realizado um estudo de caso que relaciona conceitos, como a forma de captura e o processamento de dados, que são realizados por aplicações de *streaming* de vídeos. Como exemplo de aplicação, é apresentado um estudo de caso realizado com dados do Netflix.

Com o objetivo de diferenciar as principais camadas na arquitetura em relação ao modelo de banco de dados tradicional, o trabalho de Krikun (2017) elucida as necessidades de pesquisas no que refere-se à organização dos dados, tomada de decisão, ferramentas específicas de domínio e ferramentas de plataforma. Neste estudo, o foco foi apontar para que as próximas gerações abordem tais temas e permitir que os usuários possam extrair o máximo de benefícios das grandes quantidades de dados disponíveis. O trabalho apresentou ferramentas e possibilidades para utilização, mas não trouxe uma solução clara para aplicação do Big Data em algumas áreas específicas, tampouco sobre dados educacionais.

No trabalho de Nogueira et al. (2018) é apresentado o uso de tecnologias, tais como Ambientes Virtuais de Aprendizagem (AVA) e Sistemas de Informações Educacionais (SIE) no ensino superior, e são abordados como o uso destas culminou para geração em massa de dados sobre diferentes formatos, sendo eles categorizados como estruturados e não estruturados. A o trabalho propõe apresentar uma arquitetura de Big Data, no contexto de tecnologia de armazenamento de dados, para servir como insumo para a utilização de técnicas de *Learning Analytics* (LA). A ideia do trabalho é a utilização de dados para gerar informações úteis para estudantes, professores e gestão acadêmica. Os autores ainda destacam o desafio nesta abordagem que consiste em determinar como os dados serão capturados, padronizados, processados, armazenados e apresentados para os usuários. Porém, o foco central é dado à questão da variabilidade e quantidade de dados que podem ser gerados, embora não se elucide, necessariamente, como deveria ser empregada uma infraestrutura robusta para suportar os processos de LA em ensino e aprendizagem no contexto de Big Data.

Jahn (2017) objetiva apresentar uma arquitetura de referência para tratamento de dados não estruturados, inerentes à realidade dos Institutos Federais de educação, para que possam prover análise sobre dados oriundos de redes sociais. Inicialmente, a revisão bibliográfica expõe os conceitos, linguagens e ferramentas das principais tecnologias a respeito dos conceitos que envolvem as tecnologias NoSQL. Apresenta também alguns produtos como Hadoop, Hive, HBase e outros, e denotam a grande quantidade de soluções NoSQL disponíveis no mercado. Apresenta também um levantamento de dados institucionais e mostra que o tratamento de dados não estruturados ainda é considerado um ineditismo para estas instituições que não faziam gestão destes dados. No entanto, o trabalho não apresenta soluções sobre o ponto de vista da escalabilidade horizontal e vertical, trazendo um questionamento sobre como proceder com a escolha de arquiteturas, conceitos como arquiteturas *On Premises*⁴ e em *Cloud*⁵ não são considerados. O principal foco foi nas questões relacionadas aos dados estruturados e não estruturados unicamente. Mas é necessário considerar que, ao se referir ao conceito de Big Data, vários aspectos devem ser considerados na definição de uma arquitetura, tais como cenários em *batch* ou *streaming*, nos quais impactam em vários outros aspectos como apontados em 2.2.4.

⁴ *On Premises*: Refere-se as aplicações onde própria organização tem a responsabilidade de processar suas aplicações de hardware e software. (PÉREZ et al.,2019).

No trabalho de Costa (2019) são apresentados os conceitos de *Data Warehousing* (DW) utilizando nas organizações para a recuperação de informações gerais, e propõe a evolução dessa tecnologia, de acordo com o que é aplicado ao conceito de *Big Data Warehousing* (BDW), destacando as características de sistemas em tempo real, escaláveis, de elevado desempenho, com armazenamento flexível, e baseados em *commodity* hardware, sendo capazes de ultrapassar as limitações dos DWs tradicionais, de forma a assegurar uma variedade de tarefas complexas de Big Data Analytics. Como resultado, o trabalho contribui para uma abordagem que pode ser aplicada e generalizada para suportar vários contextos de aplicação, disponibilizando modelos de dados adequados e flexíveis que conseguem reduzir o tempo de implementação entre a coleta de dados e a análise de dados. Além disso, essa abordagem apresenta a comparação entre tempos de execução de *queries*, e um uso de recursos mais eficiente do que uma abordagem dimensional tradicional, que se mostrou mais rápida, logo, a concepção e implementação de BDW, seguindo uma abordagem sistêmica, foi uma contribuição para a comunidade técnico-científica relacionada com este tópico.

⁵ *Cloud*: Refere-se ao fato do servidor e todos os seus recursos são fornecidos remotamente. Assim os dados são hospedados em um Data Center externo. Ao contratar o serviço é pago pelo consumo de recursos, como disco e memória, mas não precisa realizar um alto investimento inicial.(TAURION, 2009).

Proposta da Tese

Este capítulo apresenta a proposta de arquitetura baseada em microsserviços no que tange a camadas de ingestão, armazenamento e enriquecimento de dados para o STI-LINA, utilizando serviços AWS em contexto de aplicações que lidam com o Big Data. Assim sendo, o foco será nos modelos Estudante, Modelo Domínio e Modelo Pedagógico. Além disso, essa proposta acrescenta um novo módulo, denominado Módulo de Analytics. Este último foi incluído com a finalidade de o professor poder acompanhar o desempenho dos alunos nas atividades propostas. Por fim, é proposto também um novo algoritmo baseado em Método de Acesso Métrico (MAM), denominado SLIM-Tree-LINA, para otimizar o desempenho na recuperação de objetos de aprendizagem no momento de recomendação de material pedagógico personalizado ao estudante.

3.1 Proposta de Arquitetura para os Modelos Estudante, Domínio e Pedagógico do STI-LINA

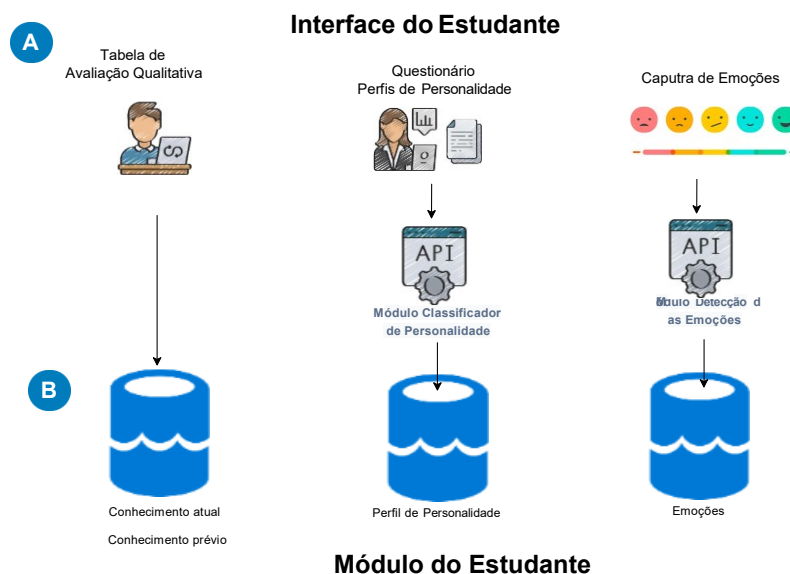
O trabalho de Melo (2021) propõe a modelagem de um Sistema Tutor Inteligente, denominado STI-LINA. O foco do trabalho foi apresentar os componentes/módulos que deveriam estar presentes no sistema. Porém, não foi apresentado como seria implementada a arquitetura deste sistema e quais as tecnologias necessárias para viabilizá-lo, no que tange à captura, armazenamento e processamento de dados. Nota-se que definir a arquitetura de uma aplicação não é uma tarefa simples, e existem vários padrões que podem ser seguidos, sendo difícil escolher o que melhor se enquadre no propósito de cada aplicação. Neste contexto, o trabalho proposto é uma continuação do trabalho de Melo (2021), porém com o foco em apresentar um modelo de arquitetura baseada em microsserviços viabilizando o dimensionamento horizontal para satisfazer certos requisitos, como a escalabilidade em cenários de Big Data.

Esta abordagem está amparada nas teorias apresentadas sobre as características de Big Data e sua aplicação no contexto de dados educacionais, na modelagem do STI-LINA e nas áreas de Mineração de Dados Educacionais (EDM). Como objeto de estudo as ar-

quitetas propostas a seguir serão utilizadas para promover a implementação do Modelo do Estudante, Modelo de Domínio e Modelo Pedagógico, e também um novo módulo, denominado de Módulo de Analytics. Assim sendo, leva-se em consideração as necessidades específicas no que refere-se à captura (ingestão de dados), modelagem e armazenamento de dados nos Modelos Estudante e Domínio, conforme arquitetura apresentada em (MELO et al., 2021). Além disso, considera-se o consumo e disponibilização dos dados para aplicações que fazem o processamento/enriquecimento de dados, sendo estes utilizados como insumo para fomentar a tomada de ação no Modelo Pedagógico, e também prover informações importantes de acompanhamento no Módulo de Analytics.

3.1.1 STI-LINA – Modelo do Estudante e Interface do Estudante

Figura 9 – Mapeamento das aplicações que fazem parte dos Modelo do Estudante (a) Conhecimento prévio e Conhecimento atual, (b) Perfil e de personalidade e (c) Emoções. E da Interface de Estudante responsável pelo processo de ingestão de dados neste módulo.



Fonte: Adaptado de Melo et al. (2021)

Na Figura 9, em B é apresentado como é constituído o ME (Modelo Estudante), que consiste em uma camada relacionada ao armazenamento de dados. Esta camada é responsável por armazenar informações sobre o conhecimento prévio e o conhecimento atual, que são provenientes da ação de preenchimento da Tabela de Avaliação Qualitativa (TAQ), como demonstrado na Figura 9-A. Além disso, temos a ingestão relacionada ao perfil de personalidade do estudante, no qual o mesmo deverá responder a um questionários composto por 16 questões, o qual será consumido pelo Módulo Classificador de Personalidade, que consiste em um algoritmo que irá classificar a personalidade do estudante. Maiores detalhes sobre este algoritmo podem ser encontrados no trabalho de Melo (2021). Após

esse processo, deve-se armazenar essa informação em uma estrutura adequada. Por fim, temos o estado emocional do estudante durante todo o processo cognitivo (emoções básicas), que é capturado por meio da câmera ligada, e enviado para o Módulo Detecção de emoções, que consiste em um algoritmo preparado para classificar as emoções do estudante. Após esse processo, os dados serão armazenados para consumo posterior em uma estrutura eficiente que permita a recuperação de forma rápida e em tempo de execução. Diante dessa apresentação do módulo estudante, surgem algumas questões que devem ser abordadas para cada um dos componentes (a), (b) e (c), tais como: Como é o processo de ingestão desses dados utilizando o processamento *batch* ou *streaming* de dados? Quais tecnologias adequadas para armazenamento destes dados? Para responder tais questões foi necessário compreender os conceitos e definir o requisito para cada componente no que tange às duas indagações acima.

3.1.2 Classificação de necessidade de processamento *batch* e *streaming* para componentes do Módulo do Estudante

Ao serem analisados os dados em movimento, ou seja, gerados continuamente, estamos lidando com cenários de dados em *streaming*. A característica principal é a chegada contínua de dados ‘pequenos’ na ordem de *kilobytes*. Um *streaming* é um conjunto de eventos, e o evento é tudo o que acontece a todo momento. Já no cenário **batch** temos uma transmissão maior de dados que foram agrupados em um intervalo de tempo específico.

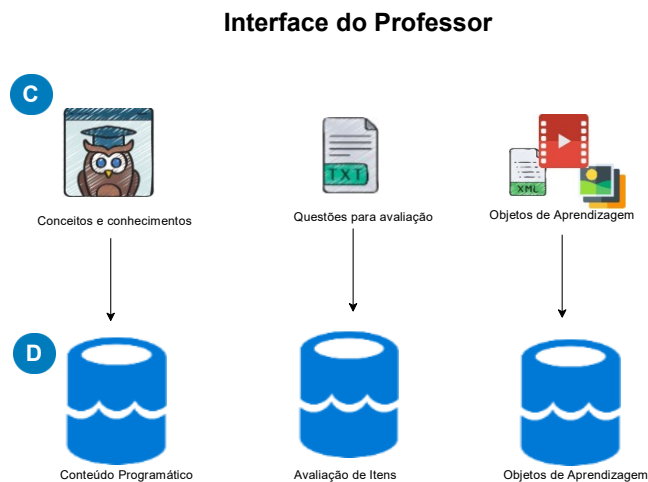
Quadro 3 – Visão geral dos componentes do Modelo do Estudante por tipo de processamento.

Componentes	Processamento
Preferências do Estudantes	<i>Batch</i>
Conhecimento atual	<i>Batch</i>
Conhecimento Prévio	<i>Batch</i>
Personalidade	<i>Batch</i>
Emoções	<i>Streaming</i>

Fonte: Elaborado pelo autor.

No Quadro 3 podemos notar que o componente Personalidade é processado em cenários *batch*, isso porque, o aluno, ao interagir com a plataforma e responder ao questionário de personalidade, só realizará uma única vez. Após o preenchimento desse questionário, o mesmo é enviado para o Módulo Classificador de Personalidade (mais detalhes podem ser vistos na Figura 9). Os componentes de preferência do estudante, conhecimento atual e conhecimento prévio, apesar de estarem em constante atualização, não são aplicações que lidam com dados em cenário *streaming*, isso porque a interação não ocorre em fluxo constante, e sim em momentos pontuais da interação do aluno com a plataforma. Nesse sentido, todos também são considerados como processamento *batch*. Já as emoções são capturadas a todo momento que o aluno está interagindo com a Plataforma, a captura é realizada com câmera ativa, e suas expressões faciais são capturadas e enviadas como um evento para o Módulo Detecção das Emoções.

Figura 10 – Mapeamento das componentes de aplicação que fazem parte dos Módulo do Domínio (a) Conteúdo Programático, (b) Avaliação de Itens e (c) Objetos de Aprendizagem. E da Interface de Domínio que faz o processo de ingestão de dados neste módulo.



Fonte: Elaborado pelo Autor.

Na Figura 10, em C é apresentada a camada de ingestão de dados que consiste na Interface do Professor. Nessa camada, o professor deverá interagir com o sistema para realizar as ingestões de dados. É possível a inserção de conceitos e conhecimento, questões referentes à avaliação do conteúdo, e os objetos de aprendizagem disponibilizados ao aluno.

3.1.3 Classificação de necessidade de processamento *batch* e *streaming* para componentes do Modelo de Domínio

Mediante essa apresentação do Modelo do estudante, surgem algumas questões que devem ser abordadas para definição de tecnologias para cada um dos componentes (a), (b) e (c), tais como: Como é o processo de ingestão desses dados utilizando o processamento *batch* ou *streaming* de dados? Para responder a esta questão foi necessário compreender os conceitos e definir os requisitos para cada componente no que tange às duas indagações acima.

No Quadro 4 podemos notar que, para todos os componentes deste módulo, teremos processamento de dados considerando o cenário *batch*. Isso porque todos os conteúdos de ingestão são inseridos periodicamente pelo professor ao interagir com o sistema. As seções 3.1.2 e 3.1.3, tem como finalidade apresentar a proposta para analisar os componentes dos Modelo do Estudante e de Domínio na ótica de cenários de processamento de dados em requisitos relacionados a *streaming* e *batch*, sendo necessário levar em consideração as características individuais de cada componente. O foco consiste em entender o comportamento dos dados e a forma de ingestão dos mesmos.

Quadro 4 – Visão geral dos componentes do Modelo de Domínio por tipo de processamento.

Aplicação	Processamento
Conteúdo Programático	<i>Batch</i>
Avaliação de Itens	<i>Batch</i>
Ramificação dos pré requisitos	<i>Batch</i>

Fonte: Elaborado pelo autor.

Importante ressaltar que, na definição do tipo de tecnologia de armazenamento, na definição do schema para armazenamento dos dados, e também nas propostas de arquiteturas baseadas em microsserviços, foi extremamente necessário seguir uma metodologia. Para isso, foi fundamental compreender bem os conceitos delineados na seção 1.4. Assim, todo o processo foi guiado considerando o *Design Research*. Inicialmente, definiu-se os problemas de pesquisa, que consistem em desenvolver propostas de implementação para os ME, MD e MP, baseadas em tecnologias AWS, para viabilizar os módulos e os componentes do STI-LINA em contexto de Big Data. Delimitou-se o contexto, as potencialidades e as limitações, com base no referencial teórico. Logo, iniciou-se a construção formal que nos permitiu o processo de investigação das várias possibilidades de combinação de serviços AWS. Esse passo foi importante para a construção efetiva dos artefatos que, nesse caso, são propostas acerca de qual tecnologia de armazenamento utilizar para cada componente, além da formulação da arquitetura que permite viabilizar os processos de ingestão, armazenamento, captura e processamento dos dados em contexto de Big Data.

Uma vez que os SGDBs foram escolhidos e as arquiteturas foram desenvolvidas, essas definições passaram por avaliações e testes dos quais simulou-se em uma conta AWS gratuita a orquestração dos serviços propostos. Nesse processo, foi necessário remodelar mais de uma vez a arquitetura proposta até, por fim, estar aderir às necessidades de cada componente. Nas próximas seções serão apresentados os resultados alcançados após transpor todas as etapas da metodologia.

A próxima seção será destinada à apresentação da camada referente ao armazenamento de dados.

3.1.4 Armazenamento de dados para ME e MD do STI-LINA

Esta seção tem como fundamento explorar os aspectos fundamentais para definição de tipos de tecnologia apropriados para o armazenamento dos dados de cada um dos componentes que compõem o MD e ME. Neste contexto, surge a necessidade de compreender qual(is) tecnologias adequadas para armazenamento destes dados em nuvem e apresentar uma proposta de modelagem física para estes dados. Para nortear essa definição foi necessário seguir alguns passos, tais como:

1. Definir o tipo de dados (estruturado, semi ou não estruturado);
2. Compreender fundamentos de tecnologias relacionais ou não relacionais;
3. Escolha do serviço de armazenamento;
4. Elaborar o *Schema* de dados adequado com a escolha do serviço;

Quadro 5 – Visão geral dos componentes do Modelo do Estudante e Modelo de Domínio por definição de tipo de dados e escolha de tecnologia para armazenamento de dados em nuvem.

Modelos	Componentes	Tipo de dados	Tecnologia	SGBD
Módulo do Estudante	Preferências do Estudante	Não estruturado	Não Relacional	DynamoDB
	Conhecimento atual	Estruturado	Relacional	RDS
	Conhecimento Prévio	Estruturado	Relacional	RDS
	Personalidade	Estruturado	Relacional	RDS
	Emoções	Não Estruturado	Não Relacional	DynamoDB
Módulo de Domínio	Conteúdo Programático	Estruturado	Relacional	RDS
	Avaliação de Itens	Estruturado	Relacional	RDS
	Ramificação Pre-requisitos	Estruturado	Relacional	RDS

Fonte: Elaborado pelo autor.

Inicialmente, para fazer a definição de uma tecnologia de banco de dados para cada componente, foi necessário conhecer qual o tipo de dado do qual cada uma destas aplicações farão uso. Partimos dos tipos de dados, sendo eles estruturado, semi-estruturado ou não estruturado, como já detalhamos na seção 2.2. No Quadro 5 é possível notar que no ME temos dois componentes que têm por definição características de tipo de dados não estruturados que são as preferências do estudante e suas emoções. Os demais dados que serão inseridos no sistema serão classificados como estruturados, que seguem um padrão fixo. Ao fazer a escolha de tecnologias de banco de dados, sempre surge a dúvida sobre qual(ais) tecnologias relacionais ou não relacionais são mais adequadas às aplicações. Com base no que foi exposto na fundamentação teórica desta tese em 3.1.3 e 3.1.4, esta serviu de base para nortear as escolhas feitas para cada componente. Como também apresentado no Quadro 5, nesse caso, somente os componentes do ME emoções e preferências utilizaram tecnologias não relacionais para armazenamento dos dados, os demais devem seguir o modelo relacional.

A Figura 11 apresenta uma proposta de modelagem entidade relacionamento para os componentes do ME e MD, os respectivos componentes que serão armazenados no RDS.

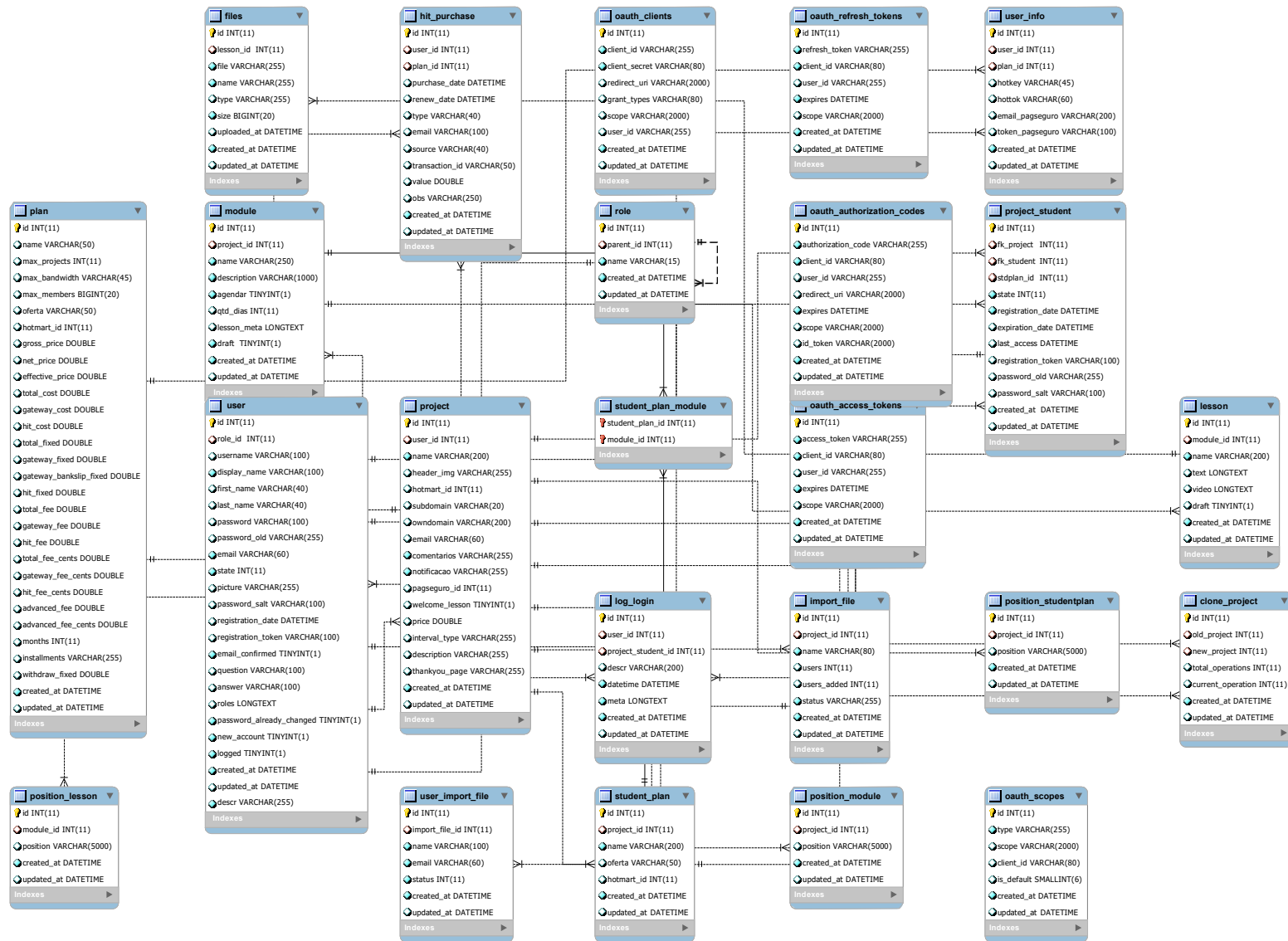


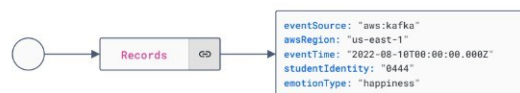
Figura 11 – Modelagem Entidade Relacionamento para componentes do MD e ME do STI-LINA

Fonte: Elaborado pelo Autor.

A Figura 11 exemplifica a modelagem proposta para o STI-LINA para o Módulo Estudante, especificamente para armazenamento dos componentes Conhecimento atual, Conhecimento Prévio e Personalidade. Já para o Módulo de domínio, a modelagem contempla os componentes Conteúdo Programático, Avaliação de Itens e Ramificação dos pré requisitos. A seguir serão apresentadas as modelagem do Modelo de Estudante, que são aplicações que serão armazenadas em banco de dados NoSQL ou não relacional, nesse caso, o serviço de nuvem escolhido foi do DynamoDB.

Figura 12 – Modelagem de dados para componente Emoções.

```
{
  "Records":
  {
    "eventSource": "aws:kafka",
    "awsRegion": "us-east-1",
    "eventTime": "2022-08-10T00:00:00.000Z",
    "studentIdentity": "0444",
    "emotionType": "happiness"
  }
}
```

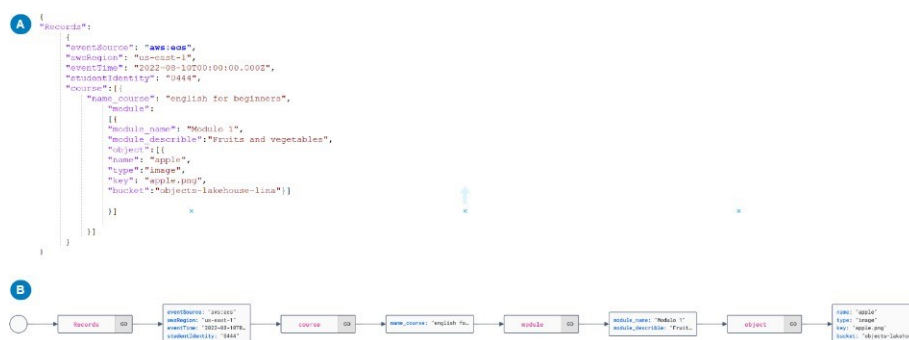


Fonte: Elaborado pelo Autor.

A Figura 12 apresenta o *schema* proposto para o armazenamento de dados referentes às emoções após terem sido classificadas pelo Módulo Classificador de Emoções. Em A, temos o schema do Json que será armazenado, e em B, o mesmo schema, embora de forma mais visual. Todos os dados coletados deverão ser armazenados no DynamoDB por período específico. Nessa proposta, a ideia é utilizar o serviços de TTL que irá garantir que os dados passados não ocupem espaço no banco de dados por um longo período de tempo. Para o processo de recomendação de objetos, será a atual classificação da emoção que servirá como input para o algoritmo, ou seja, o estado emocional no momento de interação com a plataforma. Considerando que as informações armazenadas não serão utilizadas em momento posterior, o ideal é manter configurado um tempo específico para a expiração destes dados.

A Figura 13 apresenta a proposta de armazenamento de dados do componente preferências no banco Dynamo. Em A, podemos ver a estrutura em formato json que será armazenada no banco, e, em B, temos a mesma estrutura, de forma visual, o que ajuda compreender melhor como está estruturada a camada de dados do Json. Diferentemente da modelagem para as emoções, aqui não é necessário usar um serviço de TTL para o dados, visto que é necessário sempre manter informações referente às preferências do usuário, pois são informações de insumo constante para o algoritmo de recomendação.

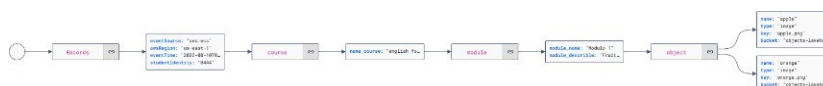
Figura 13 – Modelagem de dados para componente Preferências.



Fonte: Elaborado pelo Autor.

Nesse caso, a cada nova preferência de um objeto de aprendizagem, são adicionadas novas informações, desde que estejam no mesmo nível (curso/disciplina/módulo), como apresentado na Figura 14.

Figura 14 – Modelagem de dados para componente Preferências.



Fonte: Elaborado pelo Autor.

Esta seção tratou das ingestões e armazenamento de dados nos Modelos Estudante e Domínio, no qual considerou-se a necessidade de processamento por vezes em *batch*, por vezes em *streaming*, além de considerar os tipos de dados provenientes de cada componente, bem como foi possível definir tecnologias de armazenamento adequadas para tal finalidade e, por fim, foi proposto o *schema* de dados de acordo com o serviço utilizado.

3.1.4.1 STI-LINA: Proposta de Ingestão - Interface Estudante

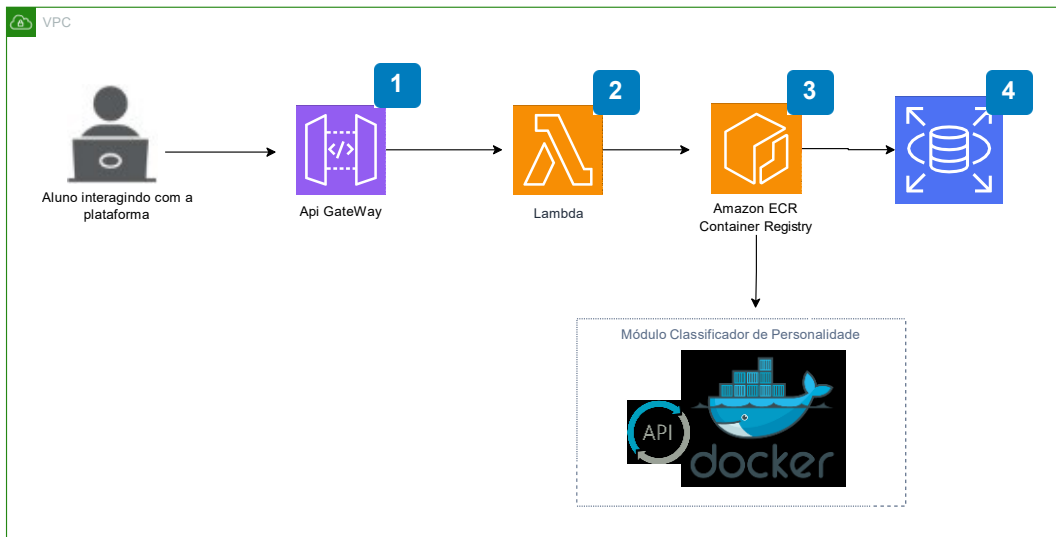
A Figura 15 apresenta uma proposta de arquitetura baseada em serviços AWS para suportar o processo de ingestão de dados provenientes da interface do Estudante. Nessa interface, temos dois tipos ações, no que refere-se à ingestão de dados, conforme especificado no trabalho de Melo (2021), a saber:

- (a) Preenchimento do questionário perfis de personalidade,
- (b) Preenchimento da Tabela de Avaliação Qualitativa.

Figura 15 – Arquitetura com serviços AWS para suporte aos processos de ingestão de dados pela Interface do Estudante até o armazenamento dos dados no RDS.

STI-LINA - Proposta de Ingestão - Interface Estudante

Proposta de Ingestão para a Interface de Estudante utilizando tecnologias AWS



Fonte: Elaborado pelo Autor.

Conforme visto na Figura 15, no passo 1, é utilizado o serviço API Gateway que recebe uma requisição do usuário no formato json. Essa requisição é enviada para uma Lambda¹ (passo 2), na forma de evento que irá iniciar um serviço de container (passo 3); após o processamento que será feito no container, será enviado uma requisição para o RDS. Nesse RDS, a estrutura utilizada para nossos testes, foi o banco de dados MariaDB, entretanto, poderia ter sido utilizado qualquer um dos bancos tratados anteriormente, como: MySQL ou PostgreSQL. Além disso, vale ressaltar que a escolha do SGBD foi baseada nos estudos realizados na seção 3.1.4.

3.1.4.2 STI-LINA: Proposta de Ingestão - Interface Professor

A Figura 16 apresenta uma proposta de arquitetura baseada em serviços AWS para suportar o processo de ingestão de dados provenientes da interface do professor. Nessa interface, temos três tipos ações no que refere-se a ingestão de dados, conforme especificado no trabalho de Melo (2021), a saber:

- (a) Ingestão de conceitos e conhecimentos,
- (b) Ingestão de questões para avaliação,

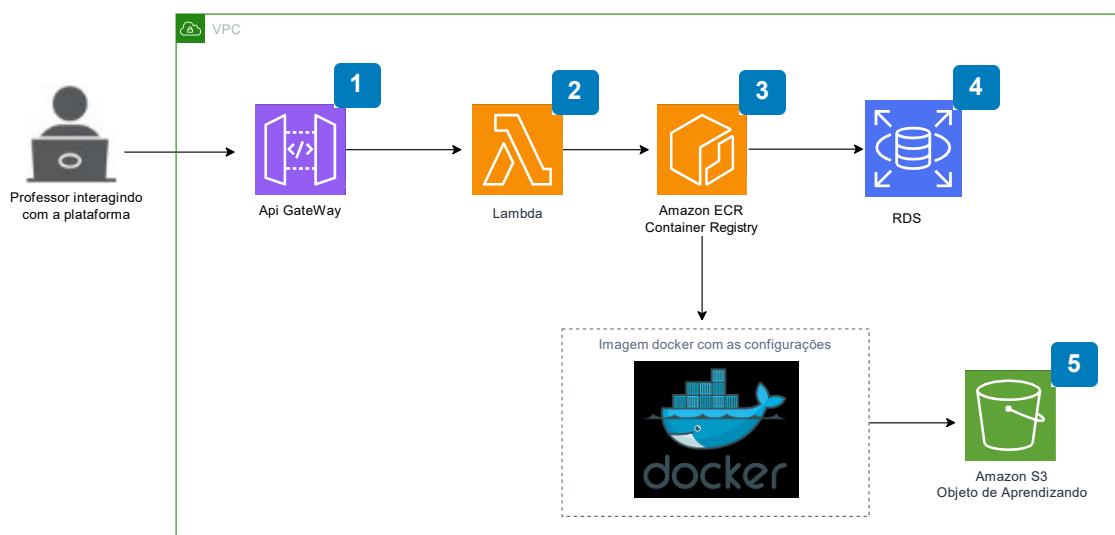
¹ O AWS Lambda consiste em um serviço de computação que não utiliza servidor, sendo que é orientado a eventos o que permite a execução de código para praticamente qualquer tipo de aplicação ou serviço de backend sem provisionar. (AMAZON, 2021)

- (c) Ingestão de Objetos de Aprendizagem em diferentes formatos.

STI-LINA - Proposta de Ingestão - Interface Professor

Figura 16 – Processo de ingestão de dados pela Interface do professor até o armazenamento no RDS ou S3.

Proposta de Ingestão para a Interface de Professor utilizando tecnologias AWS



Fonte: Elaborado pelo Autor.

Ao interagir com o sistema, o professor poderá realizar quaisquer ações (a), (b) ou (c). Assim, como apresentado na Figura 16, no passo 1, o serviço API GateWay recebe uma requisição em formato json a qual é direcionada para uma lambda, que faz a captura da requisição por meio de evento. De acordo com o parâmetro de ação advindo do evento, inicia-se o serviço ECR ². No passo 3, ao iniciar o ECR, realiza-se o processamento de dados que irá manipular o tipo e ação do usuário, e realizar o que foi proposto. Em todas as ações (a), (b) e (c) são gravadas informações nas tabelas do RDS. Além disso, se o tipo de ação for o (c) o arquivo físico é armazenado no bucket do serviço S3. A escolha para camada de armazenamento se dá devido a diversos fatores, tais como: custo extremamente baixo, suporte a múltiplos formatos de arquivos, e estar altamente disponível.

3.1.4.3 STI-LINA: Arquitetura proposta Modelo Pedagógico

A Figura 17 apresenta a proposta para o Módulo Pedagógico, no qual inclui as características, tais como recomendação de objetos de aprendizagem baseada na personalidade (Perfil de Personalidade), Caminho de aprendizagem individualizado (Nível de conhecimento), e Instrução adaptativa, conforme perfil afetivo (Emoções), as quais serão entradas para o algoritmo, que recomendará os objetos de aprendizagem, levando em consideração essas três características. Os módulo-componentes que integram esse modelo são:

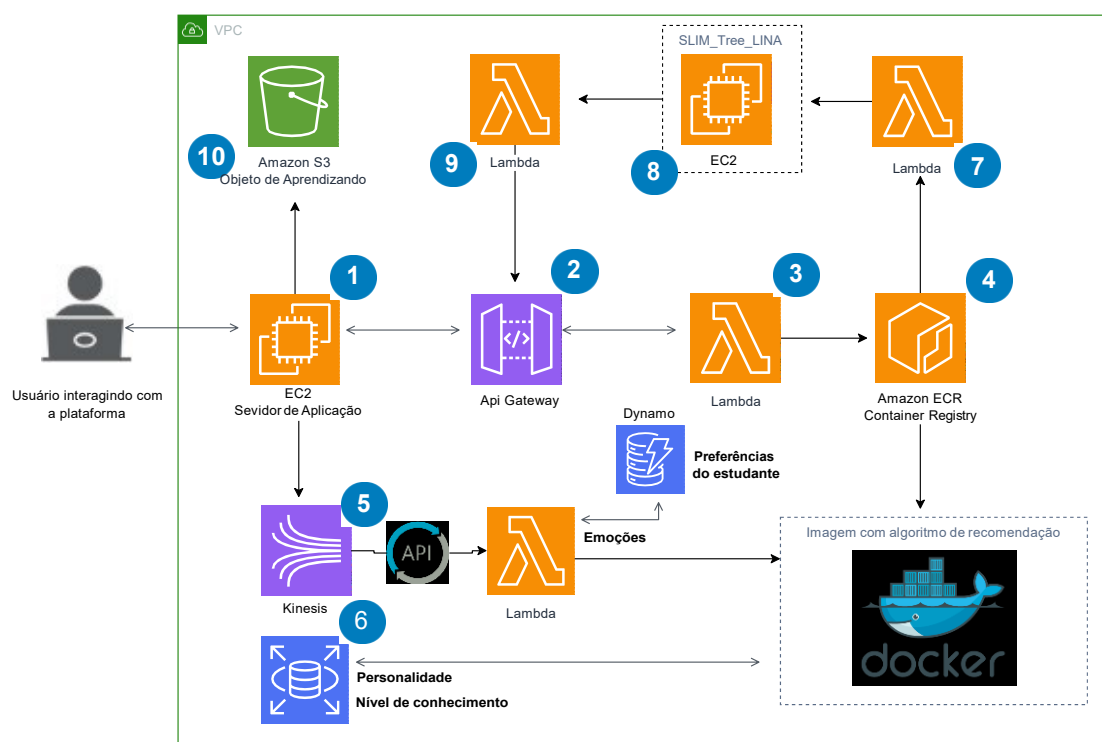
² ECR: O Amazon ECR consiste em um registro de contêiner gerenciado que oferece hospedagem de alta performance para implantar imagens e artefatos de aplicações de forma confiável.(AMAZON, 2022g)

- (a) Recomendação personalizada dos objetos de aprendizagem,
- (b) Caminho de aprendizagem individualizado,
- (c) Instrução adaptativa conforme perfil afetivo.

STI-LINA - Arquitetura proposta para Módulo Pedagógico

Figura 17 – Processo de proposta de Módulo Pedagógico.

Proposta para Módulo Pedagógico para Recomendação Personalizada de Objetos de Aprendizagem utilizando tecnologias AWS



Fonte: Elaborado pelo Autor.

Conforme observado na Figura 17, no passo 1, é apresentada a interação do aluno com o sistema que está hospedado em uma EC2, denominado servidor da aplicação. À medida que as interações são realizadas no processo de aprendizagem, ocorrem ações que disparam eventos para os sistemas, os quais requisitarão serviços personalizados para a recomendação de objetos de aprendizagem ao aluno. Já no passo 2, é apresentado o serviço API Gateway ³ que recebe uma requisição em formato *json*, e é direcionada para uma Lambda (passo 3), a qual realiza a captura da requisição por meio de evento.

³ Amazon API Gateway consiste em um serviço gerenciado que permite a “porta de entrada” para aplicativos acessarem dados, lógica de negócios ou funcionalidades de serviços de back-end.(AMAZON, 2022a).

De acordo com o parâmetro de ação advindo do evento, inicia-se o serviço ECR. No passo 4, ao iniciá-lo, é realizado o processamento de dados que irá manipular o tipo e ação solicitada. Nesse momento, é realizado o processamento do algoritmo de recomendação que receberá como ingestão de dados: (a) a recomendação de objetos aprendizagem baseada na personalidade (Perfil de Personalidade), e (b) o Caminho de aprendizagem individualizado (Nível de conhecimento). Esses dados são recuperados das tabelas do RDS, como apresentado no passo 6. Além disso, o algoritmo também recebe dados referentes a (c) Instrução adaptativa conforme perfil afetivo (Emoções), no qual são capturados no exato momento em que o usuário está interagindo com a plataforma, o que é demonstrado no passo 6. De posse desses dados, o algoritmo recomendará um objeto de aprendizagem que será ofertado para o aluno. Após esse passo, a lambda recebe a referência do objeto a ser recuperado (passo 3). Esse objeto será ofertado para o aluno no momento da interação com o sistema.

Na modelagem do STI-Lina define-se que, caso o aluno não consiga assimilar o conhecimento com o objeto ofertado, é preciso ofertar outro objeto, necessitando realizar novamente a busca por objeto similar, até que o conteúdo seja assimilado. Para aperfeiçoar esse processo, no passo 8 há um serviço EC2 executando o método de acesso métrico, denominado SLIM-TREE-LINA (ver em 3.1.4.4), que recebe uma requisição da lambda e devolve um array com os n objetos de aprendizagem mais similares àqueles relacionados como parâmetro de input. Assim sendo, o aluno não consegue assimilar o conhecimento com o objeto ofertado, logo, não são necessárias novas buscas, uma vez que, além do objeto inicial ofertado, há cerca de n outros objetos que poderão servir para o aluno. O uso desse algoritmo traz vantagem reduzindo o custo com relação ao processamento do algoritmo de recomendação, e ajuda a melhorar o tempo de buscas dos objetos, trazendo mais fluidez durante o processo de ensino. Importante notar que a quantidade de objetos pode ser parametrizada de acordo com a necessidade. No passo 9 há uma lambda que recebe esse array e recupera o endereço desses objetos no Bucket e deverá enviar para o API Gateway (passo 2), o qual devolve para o servidor de aplicação (passo 1), que terá a referência desse arquivo no Bucket (passo 10), possibilitando ser apresentado na tela para o usuário que está interagindo com a plataforma.

3.1.4.4 Método SLIM-TREE-LINA

A Indexação e recuperação por similaridade são operações fundamentais para aplicações que lidam com dados complexos. Considerando os dados atualmente produzidos no STI-LINA, no qual o professor faz a inserção de objetos de aprendizagem para um determinado conteúdo, esses dados podem ser do tipo multimídia (imagens, áudios e vídeos). Para esses dados é empregada a noção de similaridade, quando o objetivo é a recuperação, sendo útil recuperar os documentos similares (ou menos dissimilares) aos de referência.

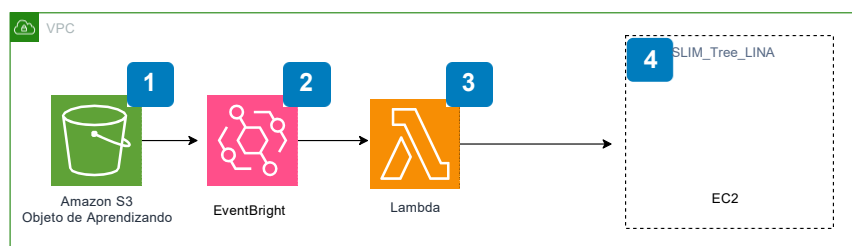
As estruturas que permitem indexar e recuperar rapidamente um determinado objeto são conhecidas como Métodos de Acesso Métricos (MAMs). Neste contexto, propõe-se a implementação do método de acesso métrico denominado SLIM-TREE-LINA, descrito no trabalho de Razente, Barioni e Sousa (2021), em uma instância EC2⁴ para auxiliar no momento da Recomendação personalizada dos objetos de aprendizagem ao estudante. Importante notar que esse método foi escolhido em detrimento a tantos outros, na literatura, pois é o único até o momento que está preparado para trabalhar tanto em cenário *batch*, quanto em cenário *streaming*, e demonstra aderência à proposta dessa tese. Esse método SLIM-TREE-LINA receberá, como entrada, um objeto de aprendizagem, o qual sairá no Módulo Recomendação Personalizado. O retorno do método será uma lista de índices contendo n objetos mais similares àquele adotado como referência. Assim, após a recuperação, esses objetos serão apresentados na tela principal como sugestões para o aluno, diferentemente da proposta anterior prevista no trabalho de Melo (2021), que devolvia apenas um objeto de preferência para o estudante. Desse modo, garante-se que, além de recomendar um conteúdo personalizado ao estudante, teremos outros n conteúdos similares a este, que estarão disponíveis ao estudante. Pode-se ressaltar que, nessa abordagem, reduz-se o número de requisições que poderiam ser feitas para o sistema para recuperar novos objetos, com isso tem-se um menor custo e um sistema que responde melhor aos usuários.

A seguir na Figura 18 é apresentada a proposta de como serão feitas as atualizações referentes aos índices do método Slim-Tree-Lina. Importante notar que, à medida que novas inserções de objetos de aprendizagem são feitas, é necessário fazer a atualização no método, para que a performance no momento da busca seja eficiente.

SLIM_Tree_LINA - Proposta de atualização

Figura 18 – O processo de atualização de dados do método Slim-Tree-Lina

Proposta de atualização para os método de acesso métrico Slim_Tree_LINA com tecnologias AWS



Fonte: Elaborado pelo Autor.

Como observado na figura 18, no passo 1, temos o repositório de objetos de aprendizado e, a partir do recebimento de um novo objeto, este será triggado via Eventbridge (passo 2), e envia

uma requisição para

⁴ Amazon EC2 é uma infraestrutura de computação alta performance. No qual pode ser instalado diferentes sistemas operacionais.(AMAZON,2022f).

a lambda que irá se comunicar com o EC2 para a atualização do índice do método Slim-Tree-LINA. Esse é um processo que ocorre constantemente e, por isso, a necessidade de um serviço de EC2 sendo executado constantemente por 24 horas. Os demais serviços Eventbridge e Lambda são serviços *serverless*.

3.1.4.5 STI-LINA: Proposta Módulo Analytics

Tradicionalmente, a arquitetura do STI-LINA não inclui o um Módulo de Analytics no qual é possível ao professor poder acompanhar o desempenho do aluno durante o caminho da aprendizagem. Desse modo, propõe-se a inclusão desse módulo com a finalidade de acompanhamento pelo professor e controle das atividades do estudante. Para isso, é necessário a definição de alguns aspectos tais como: armazenamento e o processo de tratamento dos dados analíticos. Para a camada de armazenamento é importante lembrar que esta é responsável por fornecer componentes duráveis, escalonáveis, seguros e econômicos para armazenar grandes quantidades de dados. Ela suporta o armazenamento de dados estruturados, não estruturados e conjuntos de dados de uma variedade de estruturas e formatos. Para essa arquitetura proposta de solução, será utilizado o S3 como camada de armazenamento, em função de diversos fatores, tais como: custo extremamente baixo, suporte a múltiplos formatos, e estar altamente disponível. Inicialmente, é necessário criar um fluxo de trabalho para preparação dos dados, essas tarefas contemplam a limpeza e preparação dos dados, e implica em particionar, indexar, catalogar e transformar os dados, especialmente em um formato colunar para a otimização do desempenho no momento de recuperação dos mesmos. Na figura 19, é apresentado como está orquestrado os serviços AWS proposto para servirem ao Módulo de Analytics.

No passo 1, é apresentado o RDS, no qual há o sistema legado da aplicação, que armazena os dados referentes ao MD e ME, como visto nas seções anteriores. No passo 2, temos o serviços da AWS Glue, que faz a leitura das tabelas, e grava esse dados no passo 3, bucket stage. Esse processo de leitura dos dados ocorre por meio de um agendamento, que pode ser configurado diretamente no Job. Em relação ao agendamento, o ideal é que ocorra entre 23 horas e 5 horas da manhã, pois a taxa de requisições ao banco é menor nesse período, uma vez que há menor utilização do sistema. Esse job pode conter *n* tabelas nesse processo, e deve gravar esse dados no bucket stage, como definido no passo 3. Os dados devem ser gravados no formato parquet⁵, isso porque irá auxiliar a otimizar o desempenho de determinadas consultas de leitura.

Como a chegada desses arquivos no bucket stage, a cada novo arquivo será disparado um evento via Amazon Eventbridge⁶ (passo 4), o qual também dispara a chama de um lambda que irá receber o arquivo e mover para o Bucket Raw, como demonstrado na Figura

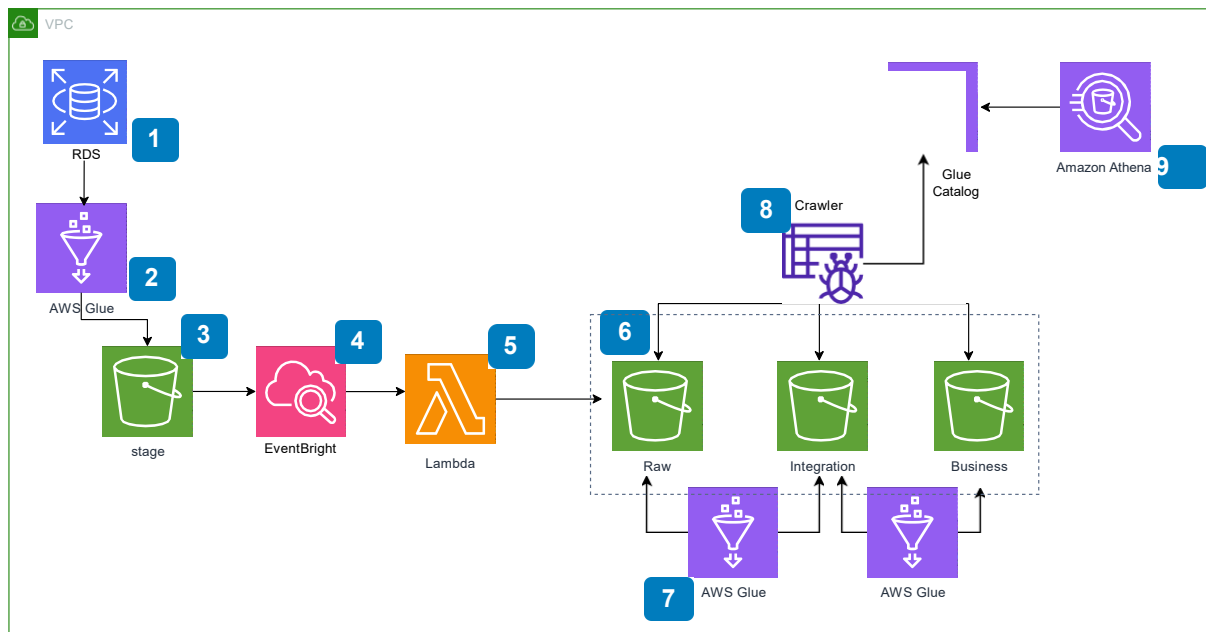
⁵ Parquet é um formato de código aberto que foi projetado para um formato de armazenamento colunar que é utilizado para implementações de arquitetura de Big Data.(OLIVEIRA et al.,2021)

⁶ O Amazon EventBridge é um barramento de eventos com tecnologia sem servidor que permite receber, filtrar, transformar, rotear e entregar eventos.(AMAZON,2022m)

STI-LINA - Proposta de Módulo de Analytics

Figura 19 – Visão geral a Proposta do Módulo de Analytics utilizando serviços AWS.

STI-LINA proposta de Módulo de Analytics utilizando tecnologias AWS



Fonte: Elaborado pelo Autor.

19. Os passos feitos até esse momento são considerados como processo de ‘pouso’ das bases que serão utilizadas. Importante notar que os dados são carregados sempre D-1 (carregamento *batch*), isso infere que haverá sempre informações históricas com um dia de defasagem.

Em 6, apresentam-se os três buckets principais para armazenamento de dados, sendo eles:

- ❑ **Raw:** Armazena os dados em formato originais sem modificações,
- ❑ **Integration:** Armazena dados em primeira fase de tratamento,
- ❑ **Business:** Armazena dados já tratados e refinados em sua forma final para consumo.

No passo 7 é realizado o tratamento de dados e, para isso, é proposta a utilização da ferramenta de ETL Glue. Por meio dela serão realizadas todas as tratativas, incluindo agregações, exclusões e junção desses dados. A proposta é que, para cada tabela que ‘pouse’ no bucket Raw, seja realizado um job que leia esses dados e faça transformações simples, como cast em tipo de dados, renomeação de campos, e filtro para campos que devem ir para a próxima camada de armazenamento.

Após realizadas as transformações, os dados devem ser gravados na camada de Integration. Essa camada receberá a leitura de dados por outros processos do AWS Glue, que será responsável por realizar o processo de enriquecimento dos dados, no qual, tem-se

agregações de dados, junções, e também são aplicadas regras de transformações nesses dados, para que sejam gravados na camada Business. Nessa camada, os dados já encontraram-se no formato correto, limpos e agregados. As informações serão disponibilizadas no serviço do Amazon Athena⁷, para poder explorar esses dados com consultas SQL e VIEWS, possibilitando o trabalho de analistas de negócio e cientistas/analistas de dados que desejam extrair *insights*. Nessa camada, o negócio poderá utilizar qualquer ferramenta de visualização para consumir os dados. Um exemplo é o uso do Power BI como ferramenta voltada para a visualização e entendimento dos *insights* e informações gerenciais para o usuário de negócio.

Importante notar que, na etapa 8, há o serviço de Crawler, que é responsável por mapear os dados que estão no bucket, e gravar um metadado no catálogo de dados, para que seja possível a visualização dos mesmos pela ferramenta do Athena – ferramenta esta que funciona como uma abstração de uma interface de banco de dados, a qual lê os dados em formato colunar, e possibilita utilizar da sintaxe SQL para consulta-los. A integração dos dados e o catálogo de dados se dará por pipelines com workflows específicos do AWS Glue⁸, os quais serão aplicadas as regras de negócio unificadas para cada job. Os jobs serão disparados por *triggers*, com periodicidade programada, levando em consideração a prioridade e a dependências dos dados.

⁷ O Amazon Athena é um serviço de consultas interativas que facilita a análise de dados no Amazon S3 usando SQL padrão e não possui servidor.(AMAZON,2022b).

⁸ O AWS Glue é um serviço de integração de dados com tecnologia sem servidor e é utilizado para ingestão, preparação e movimentação de integração de dados de variadas fontes.(AMAZON,2022k).

Considerações Finais

A proposta desta teste configurou-se como uma proposta de arquitetura para implementação do Sistema Tutor Inteligente STI-LINA, baseado em microserviços com proposta de atender os requisitos de Big Data. O foco desta proposta foi nos processos que tangem às camadas de ingestão, armazenamento, captura e enriquecimento de dados, que compõem os Modelos proposto no trabalho de Melo (2021). Assim, este estudo foi direcionado e desenvolvido com o intuito de validar a hipótese formulada na Seção 1.3 e, desse modo, foi definido o objetivo geral, que consiste em propor uma abordagem de arquitetura de referência em Big Data ao se utilizar de microserviços em nuvem para viabilizar a implementação do STI-LINA.

Além disso, para satisfazer ao objetivo geral, delimitaram-se os objetivos específicos explicitados na Seção 1.2, nos quais foram todos alcançados no decorrer do trabalho. Mesmo que tenham sido alcançados todos os objetivos do estudo, ainda é possível citar algumas limitações no processo de desenvolvimento da pesquisa. Dentre elas, destacam-se as limitações com relação às propostas de arquiteturas de microserviços, às quais foram realizadas levando em consideração apenas os serviços AWS, e não terem sido feitos comparativos com outras plataformas de serviços. Além disso, podemos notar que o trabalho não chegou a explorar explicitamente os custos com relação à implementação das soluções propostas. As limitações apresentadas anteriormente, juntamente com outras ideias para melhorar os estudos que permearam essa pesquisa, são apresentadas como sugestões para trabalhos futuros na Seção 4.2.

4.1 Contribuições

Esta pesquisa contribuiu para ampliar o conhecimento dos estudos sobre Sistema Tutores Inteligentes, e como implementar arquiteturas robustas baseadas em microserviços para viabilizar requisitos de escalabilidade em cenários de Big Data. Além disso, propõe um direcionamento sobre melhores tipo de serviços para armazenamento de dados em nuvem, de acordo com as características de cada aplicação.

Especificamente, as principais contribuições deste trabalho podem ser resumidas em:

- Proposta de arquitetura integrada baseada em microsserviços para os Módulos do STI-LINA.
- Uma proposta de arquitetura baseada em microsserviços para o Módulo de Estudante e Módulo de Domínio com foco na ingestão e armazenamento de dados.
- Uma proposta de arquitetura para o Módulo de Pedagógico para a recuperação e processamento de dados.
- Definição do tipo de tecnologia de banco de dados para o armazenamento de dados dos componente dos Modelos de Estudante e Domínio. Além, disso propõe o *schema* de dados para cada um destes módulos.
- Utilização do Método de Acesso Métrico (MAM), denominado SLIM-Tree-LINA, proposto por Razente, Barion e Sousa (2021) para auxiliar na recuperação de objetos de aprendizagem baseado em consultas por similaridade.
- Inclusão do Módulo *Analytics* para o STI-LINA com a finalidade de auxiliar o professor no acompanhamento e controle das atividades dos alunos.

4.2 Trabalhos Futuros

Para dar continuidade a este trabalho, seguindo os objetivos apresentados no Capítulo 1, as seguintes atividades precisam ser realizadas:

1. Comparativos de serviços em nuvem entre outras plataformas populares de mercado como Azure, GCP e Cloudera.
2. Desenvolvimento de um Módulo de Analytics para contemplar a atualização em tempo real.
3. Implementação do Módulo de Recomendação Personalizada de Objetos de Aprendizagem integrado com o método SLIM-TREE-LINA.
4. Explorar os custos com relação às propostas de arquitetura.

4.3 Produções Bibliográficas

As produções bibliográficas resultantes deste trabalho de pesquisa até o momento são apresentadas a seguir:

1. Artigo intitulado **"Proposta de um sistema tutor inteligente considerando as características afetivas e o conhecimento do estudante para a recomendação de objetos de aprendizagem"**, publicado como capítulo de livro na Editora Atena, 2020. Refere-se aos métodos que amparam a arquitetura proposta. A partir deste trabalho são apresentados os módulos de estudante e módulo de domínio, que serão utilizados nos testes para definir a arquitetura proposta para o STI-Lina em formato de Lakehouse. Sendo assim, esta tese é uma continuidade da pesquisa anterior proposta pelo grupo de pesquisa ao qual participa o autor.
2. Artigo intitulado **"A comprehensive analysis of delayed insertions in metric access methods"** publicado na revista *Information Systems*, 2021. - Quális A1. Refere-se aos métodos que amparam a arquitetura de Big Data na recuperação de informação em cenário *batch*. Por meio deste trabalho, foi possível propor um novo método de acesso métrico otimizado para recuperação de objetos de aprendizagem por similaridade, do qual se mostrou aderente no contexto de dados educacionais sobre os quais esta tese aborda.

Parceiras em Periódicos

1. Periódico aceito para publicação **"Arquitetura de um sistema tutor inteligente para recomendação personalizada de objetos de aprendizagem considerando a teoria dos mapas de conhecimento estruturados"**, publicado na revista *Research, Society and Development*, 2021. - Quális A3.

Referências

AMAZON, I. **Amazon S3 + Amazon CloudFront: A Match Made in the Cloud**. 2021. Disponível em: <https://aws.amazon.com/dms/>. Acesso em: 17 jul. 2022.

AMAZON, I. **Amazon API Gateway**. 2022. Disponível em: <https://aws.amazon.com/pt/api-gateway/>. Acesso em: 17 jul. 2022.

AMAZON, I. **Amazon Athena**. 2022. Disponível em: <https://aws.amazon.com/pt/athena/?whats-new-cards.sort-by=item.additionalFields.postDateTime&whats-new-cards.sort-order=desc>. Acesso em: 14 fev. 2022.

AMAZON, I. **Amazon Aurora**. 2022. Disponível em: <https://aws.amazon.com/pt/rds/aurora/>. Acesso em: 14 fev. 2022.

AMAZON, I. **Amazon com Elasticsearch**. 2022. Disponível em: <https://aws.amazon.com/pt/opensearch-service/the-elk-stack/what-is-elasticsearch/>. Acesso em: 14 fev. 2022.

AMAZON, I. **Amazon DynamoDB**. 2022. Disponível em: <https://aws.amazon.com/dynamodb/?p=ft&c=db&z=3>. Acesso em: 14 fev. 2022.

AMAZON, I. **Amazon EC2**. 2022. Disponível em: <https://aws.amazon.com/pt/ec2/>. Acesso em: 14 fev. 2022.

AMAZON, I. **Amazon Elastic Container Registry (Amazon ECR)**. 2022. Disponível em: https://aws.amazon.com/pt/ecr/?nc1=h_ls. Acesso em: 14 fev. 2022.

AMAZON, I. **Amazon MemoryDB for Redis**. 2022. Disponível em: <https://aws.amazon.com/pt/redis/>. Acesso em: 17 fev. 2022.

AMAZON, I. **Amazon RDS**. 2022. Disponível em: <https://aws.amazon.com/rds/?p=ft&c=db&z=3>. Acesso em: 14 fev. 2022.

AMAZON, I. **Amazon Redshift**. 2022. Disponível em: <https://aws.amazon.com/redshift/?p=ft&c=db&z=3>. Acesso em: 14 fev. 2022.

AMAZON, I. **AWS Glue**. 2022. Disponível em: <https://aws.amazon.com/pt/glue/>. Acesso em: 14 fev. 2022.

AMAZON, I. **AWS Lambda**. 2022. Disponível em: <https://aws.amazon.com/pt/lambda/>. Acesso em: 14 fev. 2022.

AMAZON, I. **AWS Lambda**. 2022. Disponível em: <https://aws.amazon.com/pt/eventbridge/>. Acesso em: 17 fev. 2022.

- AMAZON, I. **Build Modern Applications with Free Databases on AWS**. 2022. Disponível em: <https://aws.amazon.com/free/database/?trk=eca03f9c-ce0f-4704-b08e-e6fe66f1f54d&sc_channel=ps&s_kwcid=AL!4422!3!548640877337!e!!g!!aws%20redis&ef_id=CjwKCAiAvK2bBhB8EiwAZUbP1AWT7JV2zxLhXdbDnKtrKnCw7xkAqmFn5WQG3kaq5ZKl0Lnu9-R BwE:G:s&s_kwcid=AL!4422!3!548640877337!e!!g!!aws%20redis>.
- BENGFORT, B.; KIM, J. **Analítica de dados com Hadoop: Uma introdução para cientistas de dados**. [S.l.]: Novatec Editora, 2019.
- BRUSILOVSKY, P.; MILLAN, E. User models for adaptive hypermedia and adaptive educational systems. **The adaptive Web**, Springer-Verlag Berlin Heidelberg, 2007.
- CHABOT, D. et al. **Pedagogia Emocional: sentir para aprender**. [S.l.]: Sá Editora, 2005.
- CHEN, C. P.; ZHANG, C.-Y. Data-intensive applications, challenges, techniques and technologies: A survey on big data. **Information sciences**, Elsevier, v. 275, p. 314–347, 2014.
- COSTA, C.; SANTOS, M. Y. The data scientist profile and its representativeness in the european e-competence framework and the skills framework for the information age. **International Journal of Information Management**, Elsevier, v. 37, n. 6, p. 726–734, 2017. <<https://doi.org/10.1016/j.ins.2014.01.015>>.
- COSTA, C. F. M. d. S. **Advancing the design and implementation of big data warehousing systems**. 2019. <<https://doi.org/10.1016/j.ijinfomgt.2017.07.010>>.
- COSTA, M. R. M.; SOUSA, J. C. Educação a Distância e Universidade Aberta do Brasil: reflexões e possibilidades para o futuro pós-pandemia. **Revista Thema**, [S. l.], v. 18, n. ESPECIAL, p. 124–135, 2020. DOI: 10.15536/thema.V18.Especial.2020.124-135.1832. Disponível em: <https://periodicos.ifsul.edu.br/index.php/thema/article/view/1832>. Acesso em: 18 dez. 2022. <<https://doi.org/10.15536/thema.V18.Especial.2020.124-135.1832>>.
- COSTA, N. S. et al. Teaching and learning with structured knowledge maps: An analysis about the new approach. **Creative Education**, Scientific Research Publishing, v. 7, n. 15, p. 2212, 2016. <<https://doi.org/10.4236/ce.2016.715218>>.
- COWAN, N. The magical number 4 in short-term memory: A reconsideration of mental storage capacity. **Behavioral and brain sciences**, Cambridge University Press, v. 24, n. 1, p. 87–114, 2001. <<https://doi.org/10.1017/S0140525X01003922>>.
- EILENBERG, S.; MACLANE, S. General theory of natural equivalences. **Transactions of the American Mathematical Society**, American Mathematical Society, v. 58, n. 2, p. 231–294, 1945. <<https://doi.org/10.2307/1990284>>.
- FAN, W.; BIFET, A. Mining big data: current status, and forecast to the future. **ACM SIGKDD explorations newsletter**, ACM New York, NY, USA, v. 14, n. 2, p. 1–5, 2013. <<https://doi.org/10.1145/2481244.2481246>>.
- FERREIRA, D. C. M. et al. Mapas de conhecimento estruturado: proposta de uma nova abordagem metodológica de ensino e aprendizagem. **Educere et Educare**, v. 9, 2014.

FILHO, R. B. **Uma abordagem para ensino baseada na teoria da aprendizagem significativa utilizando a teoria das categorias**. 110f. Tese (Doutorado) — Universidade Federal de Uberlândia - Departamento de Engenharia Elétrica, 2013.

GHEMAWAT, S.; GOBIOFF, H.; LEUNG, S.-T. The google file system. In: **Proceedings of the nineteenth ACM symposium on Operating systems principles**. [S.l.: s.n.], 2003. p. 29–43.<<https://doi.org/10.1145/1165389.945450>>.

GOLDMAN, A. et al. Apache hadoop: conceitos teóricos e práticos, evolução e novas possibilidades. **XXXI Jornadas de atualizações em informática**, p. 88–136, 2012.

GOS, K.; ZABIEROWSKI, W. The comparison of microservice and monolithic architecture. In: IEEE. **2020 IEEE XVth International Conference on the Perspective Technologies and Methods in MEMS Design (MEMSTECH)**. [S.l.], 2020. p. 150–153.<<https://doi.org/10.1109/MEMSTECH49584.2020.9109514>>.

HEVNER, A. R. et al. Design science in information systems research. **MIS quarterly**, JSTOR, p. 75–105, 2004.<<https://doi.org/10.2307/25148625>>.

HURWITZ, J. et al. **Big Data para leigos**. [S.l.]: Alta Books Editora, 2016.

JAHN, G. F. **Uma proposta de arquitetura para tratamento de dados não estruturados no âmbito dos institutos federais de educação**. 131f. Dissertação (Mestrado em Ciências da Computação) — Universidade Federal de Pernambuco - UFPE, 2017.

JÚNIOR, O. A. **Arquitetura de micro serviços: uma comparação com sistemas monolíticos**. 13f, Trabalho de Conclusão de Curso (Sistemas de Informação). Universidade Federal da Paraíba - UFPB, 2017.

KALID, S. et al. Big-data nosql databases: A comparison and analysis of “big-table”, “dynamodb”, and “cassandra”. In: IEEE. **2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)**. [S.l.], 2017. p. 89–93.<<https://doi.org/10.1109/ICBDA.2017.8078782>>.

KARAMPIPERIS, P.; SAMPSON, D. Adaptive learning resources sequencing in educational hypermedia systems. **Journal of Educational Technology & Society**, JSTOR, v. 8, n. 4, p. 128–147, 2005.

KRIKUN, I. Applying learning analytics methods to enhance learning quality and effectiveness in virtual learning environments. In: IEEE. **2017 5th IEEE Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE)**. [S.l.], 2017. p. 1–6.<<https://doi.org/10.1109/AIEEE.2017.8270532>>.

MARCH, S. T.; SMITH, G. F. Design and natural science research on information technology. **Decision support systems**, Elsevier, v. 15, n. 4, p. 251–266, 1995.<[https://doi.org/10.1016/0167-9236\(94\)00041-2](https://doi.org/10.1016/0167-9236(94)00041-2)>.

MELO, S. L. de. **Arquitetura de um sistema tutor inteligente para recomendação personalizada de objetos de aprendizagem considerando os estados afetivos e o conhecimento do estudante**. 2021. 119 f. Tese (Doutorado em Engenharia Elétrica) - Universidade Federal de Uberlândia, Uberlândia, 2021.<<https://doi.org/10.14393/ufu.te.2021.559>>.

MELO, Sara Luzia de; DOS SANTOS SOUSA, Régis Michel; LIMA, Luciano Vieira. **Arquitetura de um sistema tutor inteligente para recomendação personalizada de objetos de aprendizagem considerando a teoria dos mapas de conhecimentos estruturados**. Research, Society and Development, v. 10, n. 16, p. e518101623831-e518101623831. 2021.<<https://doi.org/10.33448/rsd-v10i16.23831>>.

- MILLER, G. A. The magical number seven, plus or minus two: Some limits on our capacity for processing information. **Psychological review**, American Psychological Association, v. 63, n. 2, p. 81, 1956.< <https://doi.org/10.1037/h0043158>>.
- MORABITO, V. Big data and education: Massive digital education systems. In: **Big Data and Analytics**. [S.l.]: Springer, 2015. p. 47–64.< https://doi.org/10.1007/978-3-319-10665-6_3>.
- NAFEA, S.; SIEWE, F.; HE, Y. Ulearn: Personalized course learning objects based on hybrid recommendation approach. **International Journal of Information and Education Technology**, v. 8, n. 12, p. 842–847, 2018.< <https://doi.org/10.18178/ijiet.2018.8.12.1151>>.
- NARCISS, S. et al. Exploring feedback and student characteristics relevant for personalizing feedback strategies. **Computers & Education**, Elsevier, v. 71, p. 56–76, 2014.< <https://doi.org/10.1016/j.compedu.2013.09.011>>.
- NEWMAN, S. **Building microservices**. [S.l.]: "O'Reilly Media, Inc.", 2021.
- NOGUEIRA, S. P. et al. Big data com learning analytics para apoiar o planejamento pedagógico acadêmico. In: SBC. **Anais do VII Workshop de Desafios da Computacao aplicada a Educacao**. [S.l.], 2018.< <https://doi.org/10.5753/desafie.2018.3101>>.
- OLIVEIRA, B. F. P. de et al. Análise da influência da modelagem e formato de dados no desempenho de data warehouse baseado em hadoop-hive. In: SBC. **Anais do XXXVI Simpósio Brasileiro de Bancos de Dados**. [S.l.], 2021. p. 271–276.< <https://doi.org/10.5753/sbbd.2021.17884>>.
- PÄÄKKÖNEN, P. Feasibility analysis of asterixdb and spark streaming with cassandra for stream-based processing. **Journal of Big Data**, Springer, v. 3, n. 1, p. 1–25, 2016.< <https://doi.org/10.1186/s40537-016-0041-8>>.
- PELLE, I. et al. Towards latency sensitive cloud native applications: A performance study on aws. In: IEEE. **2019 IEEE 12th International Conference on Cloud Computing (CLOUD)**. [S.l.], 2019. p. 272–280.< <https://doi.org/10.1109/CLOUD.2019.00054>>.
- PÉREZ, A. et al. On-premises serverless computing for event-driven data processing applications. In: IEEE. **2019 IEEE 12th International Conference on Cloud Computing (CLOUD)**. [S.l.], 2019. p. 414–421.< <https://doi.org/10.1109/CLOUD.2019.00073>>.
- RAABE, A. L. A. **Uma proposta de arquitetura de sistema tutor inteligente baseada na teoria das experiências de aprendizagem mediadas**. 155f, Tese (Doutorado em Informática na Educação) — Universidade Federal do Rio Grande do Sul, 2005.< <https://doi.org/10.22456/1982-1654.9718>>.
- RAMÍREZ-NORIEGA, A. et al. Towards the automatic construction of an intelligent tutoring system: Domain module. In: SPRINGER. **World Conference on Information Systems and Technologies**. [S.l.], p. 293–302, 2019.< https://doi.org/10.1007/978-3-030-16181-1_28>.
- RAZENTE, H.; BARIONI, M. C. N.; SOUSA, R. M. S. A comprehensive analysis of delayed insertions in metric access methods. **Information Systems**, Elsevier, v. 95, p. 101492, 2021.< <https://doi.org/10.1016/j.is.2020.101492>>.
- RODRIGUES, L. M. L.; CARVALHO, M. Sti-i: Sistemas tutoriais inteligentes que integram cognição, emoção e motivação. **Revista Brasileira de Informática na Educação**, v. 13, n. 1, p. 20–34, 2005.

- ROMERO, C.; VENTURA, S.; GARCÍA, E. Data mining in course management systems: Moodle case study and tutorial. **Computers & Education**, Elsevier, v. 51, n. 1, p. 368–384, 2008.< <https://doi.org/10.1016/j.compedu.2007.05.016>>.
- SIMON, H. A. **The Sciences of the Artificial, reissue of the third edition with a new introduction by John Laird**. [S.l.]: MIT press, 2019.
- SINGH, D.; REDDY, C. K. A survey on platforms for big data analytics. **Journal of big data**, Springer, v. 2, n. 1, p. 1–20, 2015.<<https://doi.org/10.1186/s40537-014-0008-6>>.
- SONI, R. K.; SONI, N. Deploy mysql as a database in aws with rds. In:_. **Spring Boot with React and AWS: Learn to Deploy a Full Stack Spring Boot React Application to AWS**. Berkeley, CA: Apress, 2021. p. 77–102. ISBN 978-1-4842-7392-0. Disponível em: https://doi.org/10.1007/978-1-4842-7392-0_3. Acesso em: 15 jun. 2022. <https://doi.org/10.1007/978-1-4842-7392-0_3>.
- TAKEDA, H.; VEERKAMP, P.; YOSHIKAWA, H. Modeling design process. **AI magazine**, v. 11, n. 4, p. 37–37, 1990.
- TAURION, C. **Cloud computing-computação em nuvem**. [S.l.]: Brasport, 2009.
- TERADATA, I. **Teradata**. 2022. Disponível em:<<https://www.teradata.com/c/HomeV1>>.
- VENUGOPALAN, S.; SRINATH, M.; RODRIGUES, P. Recommender system for e-learning through content and profile based approach. In: **Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies**. [S.l.: s.n.], 2016. p. 1–5.
- WOOLF, B. **Building intelligent interactive tutors: Student-centered strategies for revolutionizing e-learning**. [S.l.]: Morgan Kaufmann, 2010.