

CURSO DE SISTEMAS DE INFORMAÇÃO – TCC ACADÊMICO		
( ) PRÉ-PROJETO	( X ) PROJETO	ANO/SEMESTRE: 2020/02

## **APLICAÇÃO PARA GERAR BANCO DE DADOS BASEADO EM INTERFACE**

Mateus Kienen

Prof.<sup>a</sup> Simone Erbs da Costa – Orientadora

### **1 INTRODUÇÃO**

Os dados possuem uma importância crescente na atual sociedade da informação e comunicação. Eles são considerados um recurso essencial que possibilitam o aumento da produtividade, eficiência e competitividade das organizações (PIRES *et al.*, 2015). Praticamente todas as atividades executadas em aplicações envolvem uma pessoa ou uma aplicação de computador que acessará um Banco de Dados (BD). Com isso, surge a necessidade da análise de requisitos para modelagem da aplicação, assim como de um Sistema de Gerenciamento do Banco de Dados (SGBD), trabalho este, que geralmente é feito por um time de analistas (ELMASRI; NAVATHE, 2011).

Em um BD relacional os dados são armazenados em uma ou mais tabelas e o conjunto de tabelas representam o modelo de dados da aplicação. Este modelo contém todas as informações necessárias para criação das tabelas no banco de dados dentro de um SGBD, mas para isso o modelo de dados deve ser previamente criado e analisado a fim de garantir sua integridade (PIRES *et al.*, 2015). Pires *et al.* (2015) relatam que é comum encontrar BDs criados por profissionais de diversas áreas e formações, por exemplo, administração e contabilidade, que não possuem devida experiência em modelagem de BD. O problema também se apresenta em empresas de pequeno porte, nas quais nem sempre há um analista de sistemas ou projetista de BD disponível para criação da modelagem. Nestes casos a tarefa poderia ser repassada a um programador que pode não possuir experiência suficiente para conclusão desta tarefa (PIRES *et al.*, 2015).

Para facilitar esta tarefa foram criados diagramas visuais. Valdameri (2007) afirma que os diagramas visuais de modelagem de BD podem ser escritos manualmente, porém foram desenvolvidas aplicações com o propósito de auxiliar nesta questão. Essas aplicações têm características em comum, como a possibilidade de criar os modelos de dados de forma gráfica, desenhar tabelas e relacionamentos e compartilhar os modelos gerados com outros projetistas, desde que estes utilizem a mesma aplicação (VALDAMERI, 2007).

Existem SGBDs que possuem funcionalidades para a criação da modelagem. Contudo, as aplicações que possuem essas funcionalidades estão geralmente associadas a um SGBD específico, impedindo a aplicação da modelagem criada a outros SGBDs. Para resolver esta questão, Aguiar (2017) coloca que é necessário recriar a modelagem ou adaptá-la a outro SGBD

em caso de necessidade de alteração. Embora algumas destas aplicações sejam mais completas, com funcionalidades mais avançadas e com a opção de exportação para diversos SGBDs, seu uso está ligado a uma licença de software, que geralmente ocorre mediante cobrança de mensalidade (AGUIAR, 2017).

Diante deste cenário, este trabalho propõe o desenvolvimento de uma aplicação que auxiliará as empresas na criação da modelagem de dados e do BD relacional, visando amparar projetistas e desenvolvedores não muito experientes neste processo. Conjectura-se que a construção desta aplicação aprimore a qualidade dos dados das empresas, apoiando-as no processo decisório.

## 1.1 OBJETIVOS

O objetivo geral do trabalho proposto é disponibilizar uma aplicação para modelagem e criação de bancos de dados relacionais baseado em Diagrama Entidade-Relacionamento (DER).

Os objetivos específicos são:

- a) disponibilizar interfaces para que o usuário possa realizar a modelagem gráfica do BD de maneira a proporcionar ao usuário uma melhor visualização, facilitando a percepção de possíveis erros cometidos na modelagem que possam acarretar futuros erros na criação de BD relacional;
- b) aplicar Design Thinking no desenvolvimento de aplicações no campo da computação, com foco na experiência do usuário;
- c) analisar e avaliar a usabilidade e a experiência de uso das interfaces desenvolvidas e de suas funcionalidades, modelando a relação dos requisitos da aplicação e das heurísticas de Nielsen, pelo Método Relationship of M3C with User Requirements and Usability and Communicability Assessment in groupware (RURUCAg).

## 2 TRABALHOS CORRELATOS

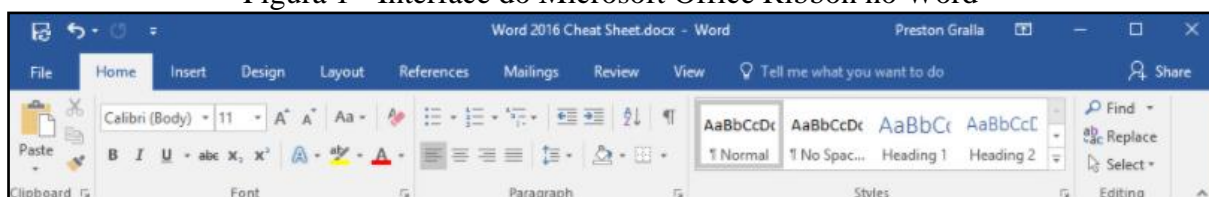
Nesta seção são descritos três trabalhos correlatos que apresentam características semelhantes ao trabalho proposto. A subseção 2.1 traz a aplicação desktop desenvolvida para criação do Diagrama Entidade-Relacionamento (DER) com foco na engenharia reversa (AGUIAR, 2017). A subseção 2.2 apresenta uma aplicação web criada para modelagem de bancos de dados relacionais com objetivo de auxiliar no aprendizado (SOUZA NETO, 2016). Por fim, a subseção 2.3 descreve uma aplicação que visa auxiliar projetistas na construção de modelo Entidade-Relacionamento (ER) focando na interface (PEQUENO *et al.*, 2020).

## 2.1 ARCHIDATA

Aguiar (2017) relata que o Archidata tem como principal objetivo a criação da modelagem gráfica e os scripts necessários para criação do banco de dados. Em sua construção foram utilizadas técnicas de engenharia reversa, permitindo que um novo diagrama possa ser obtido por meio de um banco de dados já existente e de uma interface disponibilizada ao usuário. Aguiar (2017) relata que os principais benefícios encontrados em sua aplicação incluem: disponibilizar a aplicação na plataforma desktop; software livre; geração do script SQL, permitindo suporte a múltiplos diagramas e notações que, embora a versão disponibilizada suporte apenas a diagramas de modelo ER, novos plug-ins baseados em outras notações podem ser desenvolvidos e disponibilizados à ferramenta; uso de engenharia reversa, na qual modelos podem ser obtidos a partir da estrutura de bancos de dados já existentes; editor de código Structured Query Language (SQL), que permite editar códigos na própria aplicação; e modelo conceitual (AGUIAR, 2017).

Aguiar (2017) utilizou o ambiente de desenvolvimento .Net Framework com a linguagem de programação C Sharp (C#). Para a construção da interface foi utilizado o *framework* Windows Presentation Foundation (WPF) que possui o conjunto de componentes Microsoft Office Ribbon, ao qual também é utilizado no Microsoft Office 2007. A Figura 1 apresenta o modelo do Microsoft Office Ribbon no Word, uma aplicação do Microsoft Office e a Figura 2 (retângulo A) exibe a barra de ferramentas disponíveis no Archidata. Por elas é possível visualizar a semelhança existente (AGUIAR, 2017).

Figura 1 - Interface do Microsoft Office Ribbon no Word

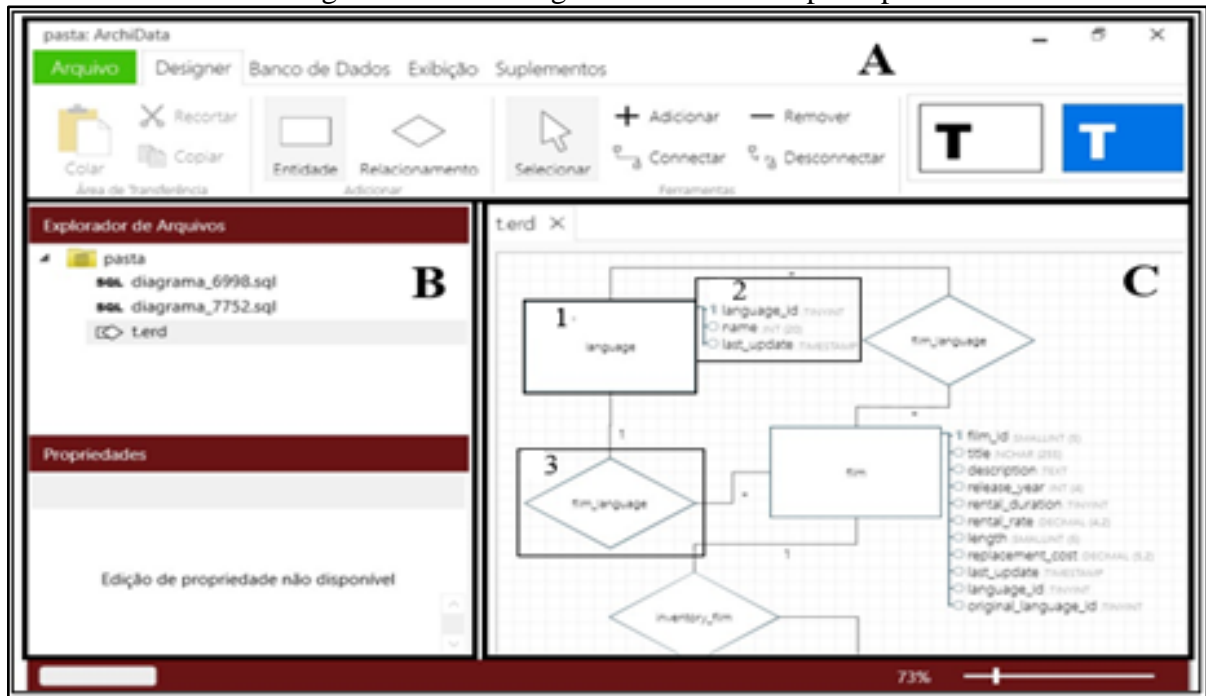


Fonte: adaptada de Microsoft (20--).

Segundo Aguiar (2017), o design do diagrama foi baseado na ferramenta BrModelo (CÂNDIDO, 2005), porém com algumas modificações para reduzir o tempo de implementação. A Figura 2 apresenta a interface principal do Archidata, que é utilizada para a manter os diagramas. No retângulo A é destacado o menu de ferramentas para criar e alterar um diagrama. O retângulo B exibe uma barra lateral que contém um navegador de arquivos e um editor de propriedades. O retângulo C apresenta a área de trabalho da aplicação com um exemplo de um diagrama criado utilizando o plug-in ER. Nele são exibidas as tabelas criadas (retângulo 1), seus atributos (retângulo 2) e a relação entre elas (retângulo 3). Por motivos didáticos, o ícone

de chave primária foi alterado para um ícone de chave, conforme apresentado no atributo `id` do retângulo 2 (AGUIAR, 2017).

Figura 2 - Interface gráfica final da tela principal



Fonte: adaptada de Aguiar (2017).

A Figura 3 apresenta um editor de propriedades disponível no Archidata, exibindo uma lista com todos os atributos (retângulo A) de determinada tabela, seus respectivos tipos de dados (retângulo B), tamanho (retângulo C), o valor padrão de determinado campo (retângulo D), identificação de chave primária (retângulo E) e demais propriedades (AGUIAR, 2017). Essa área da aplicação tem como objetivo permitir a alteração de qualquer atributo de uma tabela, visando uma melhor visualização e percepção de possíveis erros cometidos na modelagem que possam acarretar futuros erros na geração do BD (AGUIAR, 2017).

Figura 3 - Editor de propriedades do plug-in

A Campo	B Tipo	C Tamanho	D Padrão	E PK	NN	AI	Inicial	Increment.
Id	INT	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
endereço	VARCHAR	140		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
numero	INT	5		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
bairro	VARCHAR	80		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
cidade	VARCHAR	120		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
cep	VARCHAR	9		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
complemento	TEXT			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
latitude	FLOAT			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
longitude	FLOAT			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>

Adicionar Atributo

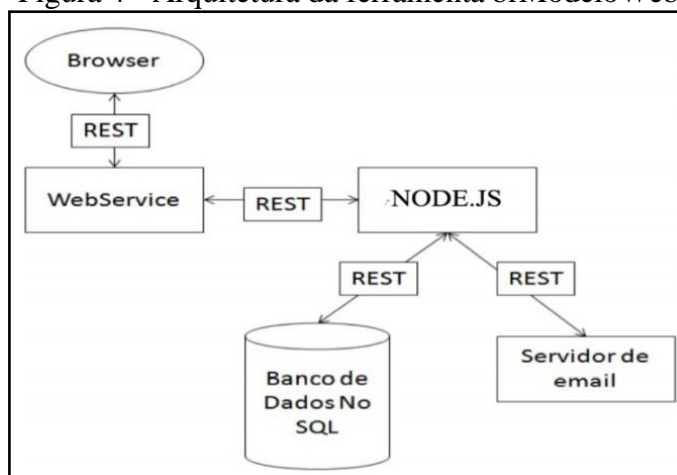
Fonte: adaptada de Aguiar (2017).

## 2.2 BRMODELOWEB

Segundo Souza Neto (2016), o brModeloWeb é uma aplicação web desenvolvida baseada no modelo ER, que pode ser acessado de qualquer dispositivo com acesso à internet e a um browser, de forma gratuita. Souza Neto (2016) teve como objetivo auxiliar na aprendizagem de modelagem de BD relacionais. O brModeloWeb possui suporte a todas as etapas de um projeto de um banco de dados (SOUZA NETO, 2016), que conforme Heuser (2009), são divididas em três etapas: modelagem conceitual, projeto lógico e projeto físico. Esta característica foi herdada da aplicação brModelo desenvolvida por Cândido (2005). A aplicação de Souza Neto (2017) também permite a geração do script SQL, porém não gera o banco de dados diretamente no SGBD, sendo necessário executar os scripts manualmente.

Souza Neto (2016) utilizou em seu desenvolvimento a arquitetura Cliente-Servidor, o padrão de projeto Model-View-Controller (MVC) e a linguagem de programação JavaScript (JS). JS foi usado no *back-end* para o processamento de dados, na interação com o banco de dados e nas páginas HyperText Markup Language (HTML), tratando o comportamento da aplicação no *front-end*. Souza Neto (2016) utilizou o *framework* AngularJS e o NodeJS, para aumentar a produtividade e organização do código no lado do cliente. Além disso, foram usados conceitos de Cascading Style Sheets (CSS) e Representational State Transfer (REST) e banco de dados MongoDB. A Figura 4 apresenta, de modo geral, a arquitetura da aplicação (SOUZA NETO, 2016).

Figura 4 - Arquitetura da ferramenta brModeloWeb



Fonte: adaptada de Souza Neto (2016).

A aplicação brModeloWeb permite ainda a criação de contas de usuário e inclusão de projetos diferentes, ou seja, o usuário poderá criar e salvar várias modelagens distintas e identificar a fase que cada projeto salvo está. É possível iniciar um projeto na fase lógica, não sendo obrigatório passar pela modelagem conceitual. Já a conversão da modelagem conceitual

para o projeto lógico é um processo feito de forma semiautomática, pois a aplicação auxilia o usuário no mapeamento, interagindo por meio de janelas que apresentam alternativas de mapeamento (SOUZA NETO, 2016).

### 2.3 CONCEPTER

O ConceptER é uma aplicação desenvolvida para criação do DER, que visa auxiliar projetistas de banco de dados e no ensino de disciplinas de bancos de dados, por meio de uma interface gráfica que possui recursos de arrastar e soltar, tornando o ambiente mais interativo e simples de ser utilizado. Após a criação dos diagramas também é possível exportar os modelos criados para arquivos de imagem, um arquivo com os scripts gerados para uso posterior, ou conectar diretamente com o SGBD escolhido para criação das estruturas, podendo ser MySQL ou SQL Server (PEQUENO *et al.*, 2020).

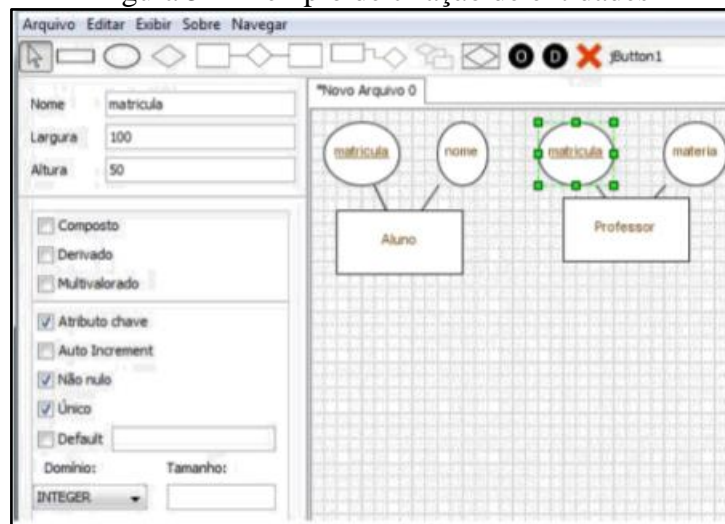
A geração dos scripts SQL é automática, na qual são utilizadas regras de conversão para manter a integridade de tabelas e relacionamentos criados. Pequeno *et al.* (2020) também destacam as seguintes características: aplicação para desktop; versão disponibilizada de forma gratuita e para uso em sala de aula no ensino da linguagem de dados SQL; geração de scripts SQL; geração automática para o SGBD MySQL ou SQL Server; e modelagem conceitual. Embora a versão apresentada ainda não possua a função para geração do modelo relacional por meio de regras de conversão do modelo ER, tal função já está prevista para ser implementada em versões futuras (PEQUENO *et al.*, 2020).

Para desenvolvimento da aplicação foi utilizado o padrão de projeto MVC, a metodologia de ciclo de vida cascata, a orientação a objetos como modelo de desenvolvimento, fazendo uso da Unified Modeling Language (UML) e Java como linguagem de programação (PEQUENO *et al.*, 2020). No projeto da interface foi identificada a forma que atrativa aos estudantes e projetistas. Após a definição da interface foi feita a criação dos componentes visuais para construção do MER, envolvendo componentes para tipos de entidades, atributos e relacionamentos, que possuem variações de restrição de participação e cardinalidade. Ainda na interface foram incluídas funções utilitárias: abrir, salvar, salvar como, exportar para imagem e exportar para o SGBD MySQL ou SQL Server. Ainda é possível, converter os diagramas para scripts SQL, tanto para geração dos arquivos quanto do código para execução no SGBD (PEQUENO *et al.*, 2020).

A Figura 5 apresenta a interface do ConceptEr com duas entidades criadas: Professor e Aluno; e os atributos: matricula, nome e materia. O atributo matricula na entidade

Professor está selecionado e a partir disso é apresentado a tabela de propriedades ao lado esquerdo. Nele é possível definir atributos ao campo selecionado, podendo ser: composto, derivado, multivalorado, atributo chave, auto incrementado, não nulo, único e informar valor padrão, o tipo de dados e o seu tamanho (PEQUENO *et al.*, 2020).

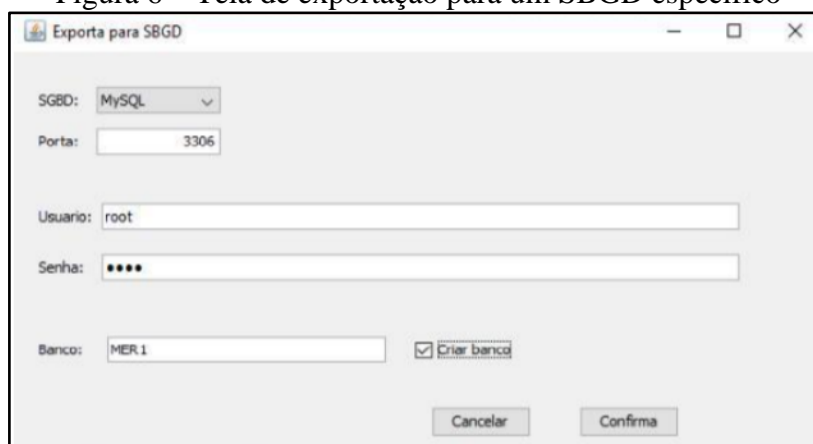
Figura 5 - Exemplo de criação de entidades



Fonte: adaptada de Pequeno *et al.* (2020).

A Figura 6 apresenta a tela de exportação para o SGBD MySQL ou para o SQL Server após o diagrama ter sido finalizado, contendo os dados necessários para liberar acesso à aplicação para executar os comandos no SGBD e criar o BD conforme os campos preenchidos nesta tela. Após a confirmação a aplicação executa os scripts SQL gerados e gera o BD.

Figura 6 – Tela de exportação para um SGBD específico



Fonte: adaptada de Pequeno *et al.* (2020).

### 3 PROPOSTA

Nesta seção serão apresentadas as justificativas para a realização do trabalho proposto (subseção 3.1), bem como serão expostos os requisitos principais (subseção 3.2), finalizando

com a metodologia e o cronograma planejado para o desenvolvimento do trabalho (subseção 3.3).

### 3.1 JUSTIFICATIVA

Nas seções 1 e 2 foram evidenciadas a relevância do tema da proposto. Neste contexto, Aguiar (2017), Souza Neto (2016) e Pequeno *et al.* (2020) identificaram a possibilidade de desenvolver soluções para modelagem e criação de bancos de dados relacionais baseado em Modelo Entidade-Relacionamento (MER). No Quadro 1 é apresentado um comparativo entre os trabalhos correlatos, de modo que as linhas representam as características e as colunas os trabalhos relacionados.

Quadro 1 - Comparativo entre os trabalhos correlatos

Correlatos Características	Archidata Aguiar (2017)	BrModeloWeb Souza Neto (2016)	Concept Pequeno <i>et al.</i> (2020)
Aplicação Web/Desktop	Desktop	Web	Desktop
Software livre	✓	✓	✓
Geração do script SQL	✓	✓	✓
Exportação para SGBD	X	X	✓
Editor de SQL	✓	X	X
Modelo conceitual	✓	✓	✓
Projeto lógico	X	✓	X
Projeto físico	X	✓	X

Fonte: elaborado pelo autor.

Conforme demonstrado no Quadro 1, percebe-se que somente Souza Neto (2016) desenvolveu sua aplicação para web, enquanto Aguiar (2017) e Pequeno *et al.* (2020) desenvolveram suas aplicações para desktop. Segundo Souza Neto (2016), a grande maioria das soluções existentes possuem a característica de serem aplicações para desktop, geralmente não comportam todas as etapas de um projeto de banco de dados e são comerciais. Portanto, há um valor a ser cobrado sobre o serviço prestado. Essa característica de software livre é encontrada tanto em Souza Neto (2016) como em Aguiar (2017) e Pequeno *et al.* (2020).

A característica de Geração do script SQL está contemplada em Aguiar (2017), Souza Neto (2016) e Pequeno *et al.* (2020). Aguiar (2017) afirma que a maioria das ferramentas de criação da modelagem estão diretamente associadas a um Sistema de SGBD específico, impedindo que o trabalho realizado seja aplicado a vários SGBDs distintos. Aguiar (2017) também relata que os sistemas que possibilitam a escolha do SGBD destino, em sua maioria estão atrelados a uma licença de software que não é gratuita. Pequeno *et al.* (2020) também se destacam por apresentarem a característica de Exportação para SGBD. Nesse sentido, Pequeno



*et al.* (2020) argumentam que as ferramentas de criação do MER são proprietárias e não permitem exportar a modelagem criada para outro SGBD.

O editor de SQL é uma característica que está presente em Aguiar (2017). Esta característica consiste em um básico editor de texto que possui algumas funções avançadas como o destaque sintático, que grifa determinadas palavras da linguagem SQL. Além disso permite ao usuário o acesso às sugestões pressionando a tecla Control (CTRL) + Espaço. A disponibilização destas funcionalidades foi possível por meio da biblioteca de código aberto AvalonEdit.

A característica de modelo conceitual está presente em Aguiar (2017), Souza Neto (2016) e Pequeno *et al.* (2020), enquanto as características de projeto lógico e físico estão presentes somente em Souza Neto (2016). O modelo conceitual, que deve ser a primeira etapa de um projeto de banco de dados e tem objetivo de elencar formalmente os requisitos de informação de um banco de dados. O projeto lógico, que é a segunda etapa, define as estruturas de dados construídas na primeira etapa, porém no nível do SGBD escolhido. Na terceira etapa, o projeto físico, são incluídos detalhes que procuram otimizar o desempenho do sistema como um todo (HEUSER, 2009).

A aplicação proposta neste trabalho se destaca ainda por permitir que alunos e projetistas de bancos de dados criem a modelagem de um banco de dados e consigam exportar para o SGBD desejado, bem como propõe a geração e edição dos scripts SQL. A proposta também visa o desenvolvimento em código aberto (open source), permitindo que novos desenvolvedores contribuam com o projeto e que empresas poupem custos de licenças para ferramentas de modelagem.

Com base nestas características, bem como foram apresentadas no Quadro 1, nota-se que o trabalho proposto se torna relevante ao buscar disponibilizar uma aplicação desktop na qual seja possível criar todas as fases do projeto de banco de dados e exportá-lo para os SGBDs Oracle, MySQL, Microsoft SQL Server, SQLite e PostgreSQL, sendo um diferencial tecnológico. Além disso, esta proposta visa permitir que o usuário consiga criar e trabalhar com vários modelos, oferecendo maior flexibilidade.

A proposta trará ainda a contribuição acadêmica do uso de Design Thinking durante a construção de aplicações na área da computação. O trabalho proposto também contribuirá ao apresentar o Método RURUCAg, que poderá ser utilizado em outros trabalhos para modelar a relação entre requisitos de aplicação e práticas no design de interface como as heurísticas de Nielsen, avaliando a usabilidade e a experiência de usuários em sistemas na área da

computação. Além disso, traz a contribuição acadêmica ao trazer a fundamentação necessária para que se possa aplicar as avaliações com os usuários finais, sendo eles os especialistas da aplicação. Como contribuição social pode-se destacar o desenvolvimento de uma aplicação para instituições de ensino e empresas de maneira gratuita e de código aberto.

### 3.2 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

Nesta subseção serão abordados os principais Requisitos Funcionais (RF), assim como os principais Requisitos Não Funcionais (RNF), conforme Quadro 2.

Quadro 2 – Principais Requisitos Funcionais e Não Funcionais

A aplicação deverá:	Tipo
permitir ao usuário manter uma modelagem de banco de dados (Create, Read, Update and Delete – CRUD)	RF
permitir ao usuário gerar o script SQL da modelagem criada	RF
permitir ao usuário editar o SQL gerado	RF
permitir ao usuário exportar a modelagem para um SGBD escolhido	RF
permitir ao usuário criar o banco de dados no SGBD escolhido	RNF
utilizar a biblioteca AvalonEdit para edição do SQL	RNF
utilizar o Ambiente de Desenvolvimento Integrado (Integrated Development Environment - IDE) Visual Studio	RNF
ser disponibilizada na plataforma desktop	RNF
ser construída utilizando <i>framework</i> Windows Presentation Foundation (WPF)	RNF
ser desenvolvida na linguagem de programação C Sharp (C#)	RNF
utilizar arquivos de configuração YAML Ain't Markup Language (YAML) para construção do BD	RNF
utilizar o Método RURUCAg para modelar os requisitos da aplicação com as heurísticas de Nielsen	RNF
utilizar o Método RURUCAg para avaliar a usabilidade e a experiência de uso da aplicação	RNF

Fonte: elaborado pelo autor.

### 3.3 METODOLOGIA

O trabalho proposto será desenvolvido diante dos seguintes instrumentos metodológicos e respeitará as etapas que serão realizadas nos períodos relacionados no Quadro 3:

- aprofundamento bibliográfico: realizar uma revisão mais aprofundada da literatura sobre os assuntos citados na revisão bibliográfica e trabalhos correlatos;
- inspiração: identificar os problemas dos usuários por meio de entrevistas com o público-alvo da solução e de pesquisas exploratórias, usando técnicas de Design Thinking;
- ideação: transformar as informações e problemas identificados na etapa anterior em ideias, usando técnicas de Design Thinking;
- levantamento dos requisitos: reavaliar os requisitos funcionais e não funcionais já definidos e, se necessário, especificar outros a partir do aprofundamento bibliográfico e da etapa ideação;
- especificação e análise: formalizar as funcionalidades da aplicação proposta por

- meio da Unified Modeling Language (UML) para criação de diagramas afim de definir casos de uso, classes, tabelas de banco etc., por meio da ferramenta Draw.io;
- f) implementação da aplicação: implementação da aplicação proposta utilizando a linguagem C Sharp (C#) e a Integrated Development environment (IDE) Visual Studio;
- g) verificação e validação: testar a aplicação para garantir que todas as funcionalidades citadas estejam em pleno funcionamento. Validar as funcionalidades bem como a usabilidade da aplicação proposta pelo Método RURUCAg.

Quadro 3 – Cronograma

Etapas	Quinzenas		2021									
			Fev.		Mar.		Abr.		Maio		Jun.	
			1	2	1	2	1	2	1	2	1	2
Aprofundamento bibliográfico												
Inspiração												
Ideação												
Levantamento dos requisitos												
Especificação e análise												
Implementação da aplicação												
Verificação e validação												

Fonte: elaborado pelo autor.

## 4 REVISÃO BIBLIOGRÁFICA

Nesta seção são apresentados os conceitos e fundamentos mais importantes para a pesquisa em questão, estando organizado da seguinte forma: a subseção **Erro! Fonte de referência não encontrada.** apresenta o tema de modelagem de banco de dados; a subseção **Erro! Fonte de referência não encontrada.** aborda o conceito de Design Thinking; e a subseção **Erro! Fonte de referência não encontrada.** aborda o Método RURUCAg.

### 4.1 MODELAGEM DE BANCOS DE DADOS

A modelagem de um banco de dados é uma descrição dos tipos de informações que estão armazenadas em um banco de dados, que pode ser representada em modelos textuais ou modelos gráficos (HEUSER, 2009). Além disso, a modelagem de dados visa criar uma abstração da realidade com o objetivo de garantir que todos os objetos de dados relacionados a um determinado contexto estejam corretamente representados dentro do banco de dados. Uma modelagem de dados mal feita, pode comprometer o sucesso de um projeto, tendo como consequência diversos problemas, como por exemplo o baixo desempenho (LEHMKUHL; EGER, 2013).

Siebra (2010) descreve que o modelo de dados como um conjunto de conceitos que podem ser usados para estruturar uma base de dados. Machado (2020) afirma que a modelagem de dados busca informações que representam um determinado contexto criando o modelo lógico de dados. Portanto, o modelo de dados é o conjunto de conceitos que podem ser utilizados para descrever as estruturas lógicas e físicas de um banco de dados (MACHADO, 2020). Siebra (2010) salienta que é importante efetuar a modelagem de dados com objetivo de conhecer melhor as informações dos usuários e como elas se relacionam para representar da melhor forma o ambiente observado e por consequência, bancos de dados mais corretos e eficientes.

Para Siebra (2010), um modelo de dados é a forma abstrata como aspectos do mundo real se relacionam, para que possam ser representados no mundo computacional. Machado (2020) relata ainda que uma das principais características da modelagem de BD é que ela fornece vários níveis de abstração dos dados, o que omite alguns detalhes sobre o armazenamento dos dados para o usuário final. Lehmkuhl e Eger (2013) relatam que esta representação dos dados pode estar submetida a diferentes níveis de abstração. Tais níveis podem ser classificados como: modelo conceitual, modelo lógico e modelo físico (LEHMKUHL; EGER, 2013).

## 4.2 DESIGN THINKING

O design tem como objetivo melhorar os aspectos funcionais, ergonômicos e visuais de um produto para atender às necessidades do consumidor, melhorando o conforto, a segurança e a satisfação dos usuários (SANTOS; CLAUDINO, 2019). Neste sentido, está o Design Thinking (DT), que segundo Paula (2015), é um processo capaz de identificar problemas e conceber soluções inovadoras para clientes e usuários, ou seja, as pessoas envolvidas no processo de desenvolvimento de uma aplicação. Luz (2018) afirma que o DT é um processo que visa compreender e assimilar um problema para então buscar soluções e ideias, composto por três frases (BROWN, 2010): inspiração, ideação e implementação.

Para Simon (1969), o DT é notado como uma metodologia de inovação desenvolvido no Instituto da Universidade de Stanford, no vale do Silício na Califórnia. Ele é uma forma de pensar, abordar, e encarar problemas, com foco na empatia colaboração e experimentação (SIMON, 1969). Na percepção de Paiva e Marques (2014), o DT é um processo que trabalha por meio do pensamento abduutivo, procurando questionar e compreender os fatos. Por esta razão o DT se destaca por primeiramente buscar compreender e assimilar o problema, para,

então, elaborar ideias e soluções que possam melhorar a vida das pessoas (PAIVA; MARQUES, 2014).

Para Brown (2010), a missão do DT é compreender observações e traduzi-las em ideias, e estas em produtos ou serviços para melhorar a vida das pessoas. Pode-se dizer, que o DT trata de uma forma abstrata do modelo utilizado para consolidação de ideias, seus conceitos podem ser utilizados por qualquer pessoa interessada, aplicando-os em uma gama de cenários de negócios (BROWN, 2010). Martins *et al.* (2016) relatam que o DT é uma abordagem que permite o desenvolvimento de uma solução centralizada no usuário final, empregando métodos do design para atender às necessidades destes.

#### 4.3 MÉTODO RELATIONSHIP OF M3C WITH USER REQUIREMENTS AND USABILITY AND COMMUNICABILITY ASSESSMENT TO GROUPWARE (RURUCAG)

O método RURUCAg foi criado para avaliar a User eXperience (UX), usabilidade e comunicabilidade das interfaces desenvolvidas bem com as suas funcionalidades. Segundo Costa (2018), o método está fundamentado em quatro métodos consolidados, sendo eles: as heurísticas de Nielsen, as expressões de comunicabilidade, pelo Modelo 3C de Colaboração e pelo Experience Sampling Method (ESM). Esses métodos são utilizados em conjunto para validar e avaliar as interfaces desenvolvidas e as suas funcionalidades, tanto para plataformas web quanto móvel.

Costa (2018) coloca que a avaliação baseada em heurística é requerida em situações que se torna necessário identificar incidentes relacionados com a usabilidade, baseando-se nas melhores práticas definidas por um conjunto de heurísticas. Para Silva (2017), o conjunto de heurísticas desenvolvido por Nielsen e Molich (1990) é o mais utilizado entre os especialistas de usabilidade, apesar de não ser inteiramente suficiente em contextos específicos. O Quadro 4 traz o conjunto de Heurísticas de Nielsen que são utilizadas na avaliação.

Quadro 4 - Heurísticas de Nielsen

Heurística		Explicação
1	Visibilidade do estado do sistema	A aplicação deve sempre manter os usuários informados sobre o que está acontecendo, por meio de um feedback apropriado dentro de um tempo razoável.
2	Concordância entre o sistema e o mundo real	A aplicação deve utilizar a linguagem do usuário, com palavras, frases e conceitos familiares ao usuário. Seguir as convenções do mundo real e fazer a informação aparecer na ordem natural e lógica.
3	Controle e liberdade ao usuário	A aplicação deve dar apoio a ações como desfazer e refazer, funções que permitam ao usuário utilizar “saídas de emergência” em caso de escolhas de funções erradas ou para sair de um estado não esperado.
4	Consistência e padrões	Devem ser seguidas convenções da plataforma de desenvolvimento e padrões de interface normalmente aceitos. Usuários não devem ter que adivinhar se palavras, situações ou ações diferentes significam a mesma coisa.

5	Prevenção de erros	A aplicação deve evitar a ocorrência de erros na sua utilização. Melhor do que apresentar boas mensagens de erro, é ter um projeto cuidadoso que evite a ocorrência de um problema.
6	Reconhecer ao invés de lembrar	Tornar objetos, ações e opções visíveis, para que o usuário não tenha que se lembrar de informações de uma parte do diálogo para outra. Instruções para uso da aplicação devem estar visíveis, ou facilmente recuperáveis, quando necessário.
7	Flexibilidade e eficiência de uso	Aceleradores (abreviações, teclas de função) podem tornar mais rápida a interação com o usuário. Permitir aos usuários customizar ações frequentes.
8	Projeto minimalista e estético	Diálogos não devem conter informação irrelevante ou raramente necessária. Todas as unidades extras de informações em um diálogo competem com aquelas que são realmente relevantes, e diminuem sua visibilidade relativa.
9	Reconhecimento, diagnóstico e recuperação de erros	Mensagens de erro devem ser expressas em linguagem simples (sem códigos), indicando precisamente o problema, e sugerindo construtivamente uma solução.
10	Ajuda e documentação	As informações de ajuda e documentação devem ser fáceis de procurar, com foco na tarefa do usuário, listando passos concretos que devem ser seguidos e não serem grandes demais.

Fonte: adaptado de Costa (2018) elaborado de Nielsen (2002).

Cabe destacar ainda, que os usuários especialistas da aplicação são aqueles que possuem expertise do que será avaliado (COSTA, 2018). Outra questão diz respeito a se fazer uso de *emoticons* junto com texto e utilizar a escala likert para compor as respostas das perguntas realizadas, trazendo opções pares, para que o participante se posicione. A Figura 4 traz os emoticons utilizados pelo método em conjunto com a escala likert. Ressalta-se que o método foi aceito sob o protocolo de número 87266318.6.-0000.0118 pelo Comitê de Ética em Pesquisa, sendo composto de três partes: termos de compromissos para realizar a avaliação, roteiro no uso da aplicação e o questionário da avaliação (COSTA, 2018).

Figura 4 – *Emoticons* com a escala Likert



Fonte: Costa (2018).

Neste sentido, Seibt (2020) utilizou o método RURUCAg para avaliar a usabilidade, a experiência do usuário e a comunicabilidade das interfaces desenvolvidas e de suas funcionalidades. Seibt (2020) relata que após a aplicação da avaliação, obteve feedbacks importantes para identificar pontos de melhorias e extensões (SEIBT, 2020). Além de Seibt (2020), o método também foi utilizado e seu uso foi significativo para Leitão (2018), Wruck (2019), Todesco (2020) entre outros.

## REREFERÊNCIAS

- AGUIAR, Ítalo Almeida. **ARCHIDATA**: ferramenta para modelagem de banco de dados com suporte a extensões e múltiplos SGBDs. 2017. 55 f. Tese (Doutorado) - Curso de Sistemas de Informação, Departamento de Computação e Sistemas, Universidade Federal de Ouro Preto, João Monlevade, 2017.
- BROWN, Tim. **Design thinking**: uma metodologia poderosa para decretar o fim das velhas ideias. Rio de Janeiro: Elsevier, 2010.
- CÂNDIDO, Carlos Henrique. **Aprendizagem em banco de dados**: implementação de ferramenta de modelagem E.R. 2005. Monografia (Especialização em Banco de Dados) - Pós-Graduação em Banco de Dados, Universidade Federal de Santa Catarina - UFSC, Universidade de Várzea Grande - UNIVAG, Várzea Grande-MT, 2005.
- COSTA, Simone Erbs da. **iLibras como Facilitador na Comunicação efetiva do Surdo**: Uso de Tecnologia Assistiva e Colaborativa Móvel. 2018. 263 f. Dissertação (Mestrado em Computação Aplicada) - Programa de Pós-Graduação em Computação Aplicada, Universidade do Estado de Santa Catarina, Joinville, 2018.
- ELMASRI, R.; NAVATHE, S. B. **Sistemas de banco de dados**. 6ª. ed. [S.l.]: Pearson Addison Wesley, 2011.
- HEUSER, Carlos Alberto. **Projeto de banco de dados**. 6. ed. Porto Alegre: Bookman, 2009. 282 p.
- LEITÃO, Geovani Finoti. **Análise de dados para usuários não técnicos utilizando o método SSBI**. 2018. 92 f. TCC (Graduação) - Curso de Ciência da Computação, Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau, 2018.
- LEHMKUHL, Décio; EGER, Djayson Roberto. **Princípios de Banco de Dados**. Indaial: Uniasselvi, 2013. 199 p.
- LUZ, Felipe Guedes da. **Empreendedorismo e design thinking**: um estudo sobre a articulação da etapa inicial dos processos de design thinking com as etapas iniciais de novos empreendimentos. 2018. 125 f. Dissertação (Mestrado) - Curso de Programa de Pós Graduação em Design da Universidade do Vale do Rio dos Sinos, Universidade do Vale do Rio dos Sinos, Porto Alegre, 2018.
- MACHADO, Felipe Nery Rodrigues. **Banco de Dados Projeto e Implementação**. 4. ed. São Paulo: Érica, 2020. 376 p.
- MARTINS, Amilton Rodrigo de Quadros; SIGNORI, Gláuber Guilherme; CAPELLARI, Márcia Rodrigues dos Santos; SOTILLE, Suellen Spinello; KALIL, Fahad. Uso de Design Thinking como Experiência de Prototipação de Ideias no Ensino Superior. **Future Journal**. São Paulo, p. 208-224. abr. 2016.
- NIELSEN, J.; MOLICH, R. Heuristic evaluation of user interfaces. In: **Proceedings of ACM CHI'90 Conference**, Seattle, WA, p. 249-256, 1990.
- PAIVA, M.; MARQUES, E. Design Thinking como ferramenta para implantação de rotinas informatizadas no serviço público. In: SIMPÓSIO DE ADMINISTRAÇÃO DA PRODUÇÃO, LOGÍSTICA E OPERAÇÕES INTERNACIONAIS, Fundação Getúlio Vargas, Rio de Janeiro, 2014. **Anais...** São Paulo: SIMPOI, 2014.

PAULA, Danielly Ferreira Oliveira de. **Model for the Innovation Teaching (MoIT)**: um modelo baseado em Design Thinking, Lean Startup e Ágil para estudantes de graduação em computação. 2015. 84 f. Dissertação (Mestrado em Ciência da Computação) - Programa de Pós-graduação em Ciência da Computação, Universidade Federal de Pernambuco, Recife, 2015.

PEQUENO, Pedro Vinícius de Oliveira et al. ConceptER: uma ferramenta para criação e manutenção do modelo entidade-relacionamento e geração automática de instruções sql para banco de dados. **Brazilian Journal Of Development**. Curitiba, p. 49345-49354. jul. 2020.

PIRES, Carlos Eduardo Santos et al. Um Jogo Didático para Detecção de Problemas de Qualidade de Dados em Bancos de Dados Relacionais. **Revista Brasileira de Informática na Educação**, Porto Alegre, v. 23, n. 3, p. 99-112, mar. 2015.

SANTOS, Magno Brazil dos; CLAUDINO, Saulo Vinicius. **Design**: sua importância em todos os aspectos. 2019. 10 f. Curso de Sistemas de Informação, Fundação Educacional São José Faculdade de Santos Dumont, Santos Dumont, 2019.

SEIBT, Gustavo Merini. **Assinômetro**: sistema web para controle de gastos com assinatura. 2020. 109 f. TCC (Graduação) - Curso de Sistemas de Informação, Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau, 2020.

SIEBRA, Sandra de Albuquerque. **Banco de Dados**. 2. ed. Recife: Universidade Federal Rural de Pernambuco, 2010. 78 p.

SILVA, Christofer Ramos da. **Desenvolvimento de um Conjunto de Heurísticas para Avaliação da Usabilidade em Sistemas do Domínio Específico de M-Learning**. 2017. 148 f. Dissertação (Mestrado) - Curso de Programa de Pós-Graduação em Design, Universidade do Estado de Santa Catarina - Udesc, Florianópolis, 2017.

SIMON, H. A. **The Sciences of the Artificial**. Cambridge, MA. 1969.

SOUZA NETO, Milton Bittencourt de. **BrModeloWeb**: ferramenta web para ensino e modelagem de banco de dados. 2016. 65 f. TCC (Graduação) - Curso de Curso de Graduação em Sistemas de Informação, Departamento de Informática e Estatística, Universidade Federal de Santa Catarina, Florianópolis, 2016.

TODESCO, Gabriel Jensen. **Team Stats**: sistema para gestão de equipes que utilizam práticas ágeis em seu processo de desenvolvimento. 2020. 109 f. TCC (Graduação) - Curso de Sistemas de Informação, Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau, 2020.

VALDAMERI, A. R. ; BACHMANN, J. WebModeler: uma Ferramenta CASE para Modelagem de Banco de Dados Relacional na Web. In: XVI Seminário de Computação, 2007, Blumenau. **Anais [...]**. Blumenau: Nova Letra, 2007. v. 16.

WRUCK, André Zimmermann. **Cachaça Digital**: sistema para controle de vendas, produção e envelhecimento de cachaça. 2019. 99 f. TCC (Graduação) - Curso de Sistemas de Informação, Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau, 2019.



**ASSINATURAS**

(Atenção: todas as folhas devem estar rubricadas)

Assinatura do(a) Aluno(a): \_\_\_\_\_

Assinatura do(a) Orientador(a): \_\_\_\_\_

Assinatura do(a) Coorientador(a) (se houver): \_\_\_\_\_

Observações do orientador em relação a itens não atendidos do pré-projeto (se houver):
--

## FORMULÁRIO DE AVALIAÇÃO – PROFESSOR AVALIADOR

Acadêmico(a): Mateus Kienen

Avaliador(a): Alexander Roberto Valdameri

ASPECTOS AVALIADOS <sup>1</sup>		atende	atende parcialmente	não atende
ASPECTOS TÉCNICOS	1. INTRODUÇÃO O tema de pesquisa está devidamente contextualizado/delimitado?	X		
	O problema está claramente formulado?	X		
	1. OBJETIVOS O objetivo principal está claramente definido e é passível de ser alcançado?	X		
	Os objetivos específicos são coerentes com o objetivo principal?	X		
	2. TRABALHOS CORRELATOS São apresentados trabalhos correlatos, bem como descritas as principais funcionalidades e os pontos fortes e fracos?	X		
	3. JUSTIFICATIVA Foi apresentado e discutido um quadro relacionando os trabalhos correlatos e suas principais funcionalidades com a proposta apresentada?	X		
	São apresentados argumentos científicos, técnicos ou metodológicos que justificam a proposta?	X		
	São apresentadas as contribuições teóricas, práticas ou sociais que justificam a proposta?	X		
	4. REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO Os requisitos funcionais e não funcionais foram claramente descritos?	X		
	5. METODOLOGIA Foram relacionadas todas as etapas necessárias para o desenvolvimento do TCC?	X		
	Os métodos, recursos e o cronograma estão devidamente apresentados e são compatíveis com a metodologia proposta?	X		
	6. REVISÃO BIBLIOGRÁFICA (atenção para a diferença de conteúdo entre projeto e pré-projeto) Os assuntos apresentados são suficientes e têm relação com o tema do TCC?	X		
	As referências contemplam adequadamente os assuntos abordados (são indicadas obras atualizadas e as mais importantes da área)?	X		
ASPECTOS METODOLÓGICOS	7. LINGUAGEM USADA (redação) O texto completo é coerente e redigido corretamente em língua portuguesa, usando linguagem formal/científica?	X		
	A exposição do assunto é ordenada (as ideias estão bem encadeadas e a linguagem utilizada é clara)?	X		

### PARECER – PROFESSOR AVALIADOR: (PREENCHER APENAS NO PROJETO)

O projeto de TCC ser deverá ser revisado, isto é, necessita de complementação, se:

- qualquer um dos itens tiver resposta NÃO ATENDE;
- pelo menos **5 (cinco)** tiverem resposta ATENDE PARCIALMENTE.

**PARECER:** ( X ) APROVADO ( ) REPROVADO

Assinatura: \_\_\_\_\_ Data: \_\_\_\_\_

<sup>1</sup> Quando o avaliador marcar algum item como atende parcialmente ou não atende, deve obrigatoriamente indicar os motivos no texto, para que o aluno saiba o porquê da avaliação.