

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE  
SÃO PAULO**

**MATHEUS LOPES DE OLIVEIRA**

**MODELO DE SISTEMA DE GERENCIAMENTO ESCOLAR**

**CAMPOS DO JORDÃO**

**2024**

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE  
SÃO PAULO**

**MATHEUS LOPES DE OLIVEIRA**

Este documento se trata da entrega da última etapa do relatório final da disciplina Banco de Dados 2, do curso de Tecnologia em Análise e Desenvolvimento de Sistemas, do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo - Campus Campos do Jordão (IFSP-CJO).

**PROFESSOR: Paulo Giovani de Faria  
Zeferino.**

**MODELO DE SISTEMA DE GERENCIAMENTO ESCOLAR**

**CAMPOS DO JORDÃO**

**2024**

## RESUMO

Este relatório abrange o desenvolvimento de um modelo de sistema de banco de dados destinado a instituições de ensino, com o propósito de aprimorar a eficiência operacional delas. O escopo principal deste projeto recai sobre a administração de dados referentes a alunos, docentes e disciplinas, com o intuito de aperfeiçoar os processos educacionais e administrativos. Por meio de uma metodologia minuciosa, foram estabelecidos os objetivos, limites claros de atuação e uma análise criteriosa dos desafios enfrentados neste meio. A modelagem do sistema baseou-se em pesquisas e entrevistas conduzidas junto a profissionais de diversos setores da instituição Colégio 24 de Maio, visando à coleta de dados pertinentes e o aprendizado. A expectativa é que o modelo resultante possa resultar em um sistema capaz de mitigar inconsistências, aprimorar as tomadas de decisão no âmbito administrativo e educacional, além de ampliar o acesso a informações relevantes para os gestores, docentes e os responsáveis dos alunos. Este relatório representa a fase inicial da documentação abrangente deste projeto.

**Palavras-Chave:** modelo de sistema; banco de dados; instituições de ensino; projeto.

## ABSTRACT

This report covers the development of a database system model for educational institutions, with the aim of improving their operational efficiency. The main scope of this project is the management of data relating to students, teachers and subjects, with the aim of improving educational and administrative processes. A thorough methodology was used to establish objectives, clear boundaries for action and a careful analysis of the challenges faced in this environment. The modeling of the system was based on surveys and interviews conducted with professionals from various sectors of the Colégio 24 de Maio institution, with the aim of collecting relevant data and learning. The expectation is that the resulting model will result in a system capable of mitigating inconsistencies, improving decision-making in the administrative and educational spheres, as well as increasing access to relevant information for managers, teachers and those responsible for students. This report represents the initial phase of the comprehensive documentation of this project.

**Keywords:** system model; database; educational institutions; project.

## **LISTA DE ILUSTRAÇÕES**

<b>FIGURA 1 – Modelo Conceitual</b>	<b>24</b>
<b>FIGURA 2 – Modelo Lógico</b>	<b>25</b>



## **LISTA DE QUADROS**

**QUADRO 1 –** Dicionário de Dados

18

## **LISTA DE ALGORITMOS**

<b>ALGORITMO 1 – Modelo Físico</b>	<b>37</b>
<b>ALGORITMO 2 – Consultas</b>	<b>50</b>





## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>11</b>
<b>1.1</b>	<b>Objetivos</b>	<b>11</b>
<b>1.2</b>	<b>Justificativa</b>	<b>12</b>
<b>1.3</b>	<b>Aspectos Metodológicos</b>	<b>12</b>
<b>1.4</b>	<b>Aporte Teórico</b>	<b>13</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>13</b>
<b>2.1</b>	<b>Sistemas de Banco de Dados</b>	<b>14</b>
<b>2.2</b>	<b>Engenharia de Software</b>	<b>14</b>
<b>2.3</b>	<b>Gestão Escolar</b>	<b>14</b>
<b>3</b>	<b>METODOLOGIA</b>	<b>15</b>
<b>3.1</b>	<b>Questionário</b>	<b>15</b>
<b>3.2</b>	<b>Requisitos Coletados</b>	<b>17</b>
<b>3.3</b>	<b>Regras de Negócio</b>	<b>22</b>
<b>3.4</b>	<b>Modelo Conceitual</b>	<b>23</b>
<b>3.5</b>	<b>Modelo Lógico</b>	<b>24</b>
<b>3.5.1</b>	<b>Dicionário de Dados</b>	<b>25</b>
<b>3.6</b>	<b>Modelo Físico</b>	<b>37</b>
<b>3.7</b>	<b>Consultas em SQL</b>	<b>50</b>
<b>4</b>	<b>RESULTADOS OBTIDOS</b>	<b>57</b>
<b>5</b>	<b>CONCLUSÃO</b>	<b>59</b>
	<b>REFERÊNCIAS</b>	<b>60</b>

## **1 INTRODUÇÃO**

Nos últimos anos, a educação tem sofrido uma mudança significativa devido ao avanço da tecnologia e à crescente necessidade de aprimoramento dos processos educacionais e administrativos. Diante desse contexto dinâmico, as escolas encaram desafios complexos ao buscar proporcionar uma educação eficiente e eficaz.

Este relatório aborda a criação de um modelo de sistema de banco de dados projetado exclusivamente para atender às necessidades dessas instituições. Com o objetivo de melhorar a eficiência operacional e garantir o funcionamento adequado e eficaz do ambiente escolar.

A necessidade de informatização no meio educacional é cada vez mais clara, pois a rápida e precisa disponibilidade de informações relevantes desempenha um papel crucial na tomada de decisões administrativas e educacionais. Entretanto, a situação educacional atual frequentemente precisa de sistemas integrados e eficazes para lidar de forma abrangente com a complexidade dos dados escolares, tanto presencialmente quanto remotamente.

Dentro desse cenário, a informatização não é apenas uma vantagem competitiva, mas sim uma necessidade essencial para garantir a agilidade, precisão e eficiência na gestão de dados educacionais. Este documento é o primeiro passo para lidar com essa necessidade, oferecendo uma análise detalhada do desenvolvimento de um modelo de banco de dados para melhorar as operações das instituições.

### **1.1 Objetivos**

O projeto está sendo desenvolvido com o fim de compreender e satisfazer as necessidades educacionais ao criar um modelo de banco de dados que atenda às demandas identificadas durante a análise do funcionamento das instituições educacionais, assegurando a precisão e a eficiência no gerenciamento das informações.

Além disso, é crucial o fornecimento de um modelo de sistema de banco de dados coeso e integrado que reflita as percepções obtidas sobre os desafios e

complexidades do ambiente escolar, tornando a gestão de dados eficiente e unificada.

E este relatório foi desenvolvido com o objetivo de possibilitar a transformação desse projeto em um software educacional futuramente, tornando-se uma base confiável para que haja a adaptação as necessidades dessas instituições.

## **1.2 Justificativa**

A escolha de desenvolver um sistema para a gestão integrada de farmácias é motivada pela crescente complexidade do ambiente farmacêutico, caracterizado por regulamentações rigorosas, demandas operacionais desafiadoras e a necessidade de oferecer serviços de saúde eficazes. Nesse contexto, a implementação de soluções tecnológicas torna-se essencial para superar obstáculos, aprimorar a eficiência operacional e proporcionar um atendimento ao cliente de qualidade.

Este trabalho justifica-se pela urgência de enfrentar desafios operacionais, adaptar-se às regulamentações em constante evolução e elevar os padrões de atendimento nas farmácias. O desenvolvimento do sistema proposto busca ser uma resposta pragmática e inovadora para as demandas do ambiente farmacêutico contemporâneo, almejando impactar positivamente a qualidade dos serviços prestados e contribuir para o avanço do setor.

## **1.3 Aspectos Metodológicos**

A metodologia adotada neste trabalho envolveu pesquisa bibliográfica e de campo, integrando teoria e prática para abordar de forma abrangente o desenvolvimento do modelo conceitual de banco de dados relacional utilizando a notação de Heuser, criado na ferramenta BrModelo, e a sua posterior transformação no modelo físico, com a sua criação realizada em SQL, e implementada para testes em um banco de dados Maria DB.

Além disso, como foi citado anteriormente, o relatório visa a criação posterior de um sistema para instituições educacionais que possua coerência com as regras de negócio e requisitos, coletados através das metodologias de Questionário e de

Entrevistas, realizadas com cada gestor, de cada departamento de uma instituição, e com um professor, sendo esses, funcionários da instituição de ensino Colégio 24 de Maio.

#### 1.4 Aporte Teórico

As pesquisas para este projeto foram fundamentadas em livros sobre sistemas de gerenciamento de banco de dados: **Sistemas de Bancos de Dados** (2011, ELMASRI, Ramez; NAVATHE, Shamkant B); **Banco de Dados Projeto e Implementação** (2014, NERY Rodrigues Machado, Felipe); **Sistemas de Banco de Dados: Projeto, Implementação e Gerenciamento** (2011, ROB, Peter; CORONEL, Carlos).

Artigos a respeito de gestão escolar: **Gestão escolar: o que é, estratégias e tudo sobre** (CLASSAPP); **GESTÃO ESCOLAR: ENFRENTANDO OS DESAFIOS COTIDIANOS EM ESCOLAS PÚBLICAS** (LÍLIA Imbiriba Sousa Colares, Maria; MACHADO Pacífico, Juracy; QUEIROGA Estrela, George); **GESTÃO ESCOLAR: DESAFIOS E POSSIBILIDADES** (VIEIRA de Lima, Shirley; ONETE Brandão Araújo, Maria; JOSÉ Guimarães de Carvalho, Emani; FERNANDES de Menezes, Fabíola); **Sistema escolar: descubra o que é e quais são os melhores do mercado** (VERSIANI, Rafael); **Gestão educacional: entenda que é, importância e o papel da tecnologia.** (FIA)

E em informações a respeito de projeto de software através do livro: **Engenharia de Software** (2011, SOMMERVILLE, Ian).

## 2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção serão revisados conceitos pertinentes de alguns dos livros, artigos, e textos, para o projeto, que fundamentaram o desenvolvimento deste documento.

## **2.1 Sistemas de Gerenciamento de Banco de Dados**

Segundo Rob e Coronel (2011), “O projeto de banco de dados refere-se às atividades que focam na elaboração da estrutura que será utilizada para armazenar e gerenciar dados do usuário final”, além disso, “Um banco bem projetado facilita o gerenciamento dos dados e gera informações precisas e valiosas”, e ambos são fatores que motivam o desenvolvimento deste documento.

## **2.2 Engenharia de Software**

De acordo com Sommerville (2011), “Modelagem de sistema é o processo de desenvolvimento de modelos abstratos de um sistema em que cada modelo apresenta uma visão ou perspectiva, diferente do sistema”.

O autor salienta que “Os modelos são usados durante o processo de engenharia de requisitos para ajudar a extrair os requisitos do sistema; durante o processo de projeto, são usados para descrever o sistema para os engenheiros que o implementam; e, após isso, são usados para documentar a estrutura e operação do sistema”.

## **2.3 Gestão Escolar**

Segundo o blog da Fia Business School, “A gestão educacional é fundamental para garantir a saúde financeira da instituição de ensino”. “Para tanto, é preciso estabelecer estratégias para captar e distribuir recursos financeiros de maneira correta a todos os setores da instituição”. Esses trechos destacam a importância da gestão escolar para o escopo financeiro de uma instituição, mas, além desse fator, o blog destaca que “a gestão educacional também é importante para o aspecto pedagógico da instituição”, tornando possível a melhora do ensino através de uma boa gestão.

### 3 METODOLOGIA

Conforme foi pontuado anteriormente, os dados para este projeto foram coletados através das metodologias de Questionário e de Entrevista. Realizei o questionário através da plataforma Google Forms, para os líderes dos quatro setores do Colégio 24 de Maio:

- Pedagógico: Realizei o questionário contendo as questões pedagógicas de uma instituição de ensino para a Coordenadora Pedagógica Josymeire Lopes dos Santos Santana;
- Financeiro: O questionário contendo questões sobre a parte financeira de uma instituição de ensino foi feito para o Gerente Thiago de Araújo Silva;
- Marketing: O questionário a respeito do ponto de vista de divulgação da escola, com o fim de coletar quais informações julgam como necessárias para compor os seus materiais foi feito para a Supervisora de Marketing Renata Vicente;
- Secretaria: Por fim, o questionário feito para coletar informações a respeito das matrículas foi feito para a Secretária Sueli de Almeida Santos.

#### 3.1 Questionário

As seguintes questões foram realizadas para os líderes dos departamentos:

- “Descreva como são as tarefas que realiza no seu dia a dia na escola. Se possível, com detalhes e informando as interações que realiza com os outros departamentos.”.
- “1 - No geral, quais funcionalidades julga como cruciais para um sistema escolar?”.
- “2 - No dia a dia, como iria preferir acessar o sistema? Online por meio de um site, ou por meio de um aplicativo instalado no computador?”.
- “3 - Quais informações julga como cruciais para o cadastro de uma matéria?”.
- “4 - Quais informações julga como cruciais para o cadastro de uma sala?”.
- “6 - Quais informações a senhora julga como cruciais para o cadastro de um curso?”.

- “7 - Acharia válido o cadastro dos responsáveis do aluno vinculado ao cadastro do próprio aluno?”.
- “8 - Para o futuro, acha interessante um sistema que possa ser acessado também pelos pais e alunos para monitorarem o curso das atividades do aluno, horários, eventos, e entre outras coisas?”.
- “9 - Como é feito o cadastro das listas de presença das aulas?”.
- “10 - Quais documentos os responsáveis devem apresentar para o cadastro (matrícula) de um aluno?”.
- “11 - Como são organizados e registrados eventos, reuniões e atividades escolares? E como costuma registrar os cronogramas dessas atividades?”.
- “12 - Como seriam registradas as notas e o desempenho dos alunos?”.
- “13 - Como seriam registrados eventos relacionados ao aluno na escola? Por exemplo, uma advertência.”.
- “14 - O que sente falta no sistema que utiliza?”.
- “15 - Existe um sistema para controle de estoque e inventário de materiais de ensino? O que costumam estocar? E como funciona o processo de controle de estoque nas suas atividades?”.
- “16 - Como são registradas as mensalidades dos alunos? Existe um sistema de pagamento online, boletos bancários ou outra forma específica?”.
- “17 - Como são registradas as pendências quanto a materiais escolares? E as mensalidades?”.
- “18 - A escola oferece bolsas de estudo ou programas de auxílio financeiro? Se sim, quais são os critérios utilizados para que um aluno seja matriculado como bolsista?”.
- “19 - Quais são os métodos de pagamento aceitos pela escola?”.
- “20 - Como são registradas e categorizadas as despesas da escola? Existe um sistema de categorização específico?”.
- “21 - Como é elaborado o orçamento da escola? Existe um processo específico para isso?”.
- “22 - Como são registrados e gerenciados os contratos com fornecedores? Quais são os requisitos para que um fornecedor seja cadastrado?”.
- “23 - Quais são os requisitos regulatórios e legais relacionados às finanças da escola?”.



- “24 - Quem terá acesso às informações financeiras? Existem diferentes níveis de permissão?”.
- “25 - Como são gerenciados os dados dos funcionários da escola? E quais informações são necessárias para cadastrar um novo funcionário?”.
- “26 - Como são cadastradas as folhas de pagamento e benefícios dos funcionários?”.
- “27 - Como são arquivados e gerenciados os documentos administrativos, como registros acadêmicos, políticas escolares e contratos?”.
- “28 - Quem terá acesso às informações administrativas? Existem diferentes níveis de permissão?”.
- “29 - Como são desenvolvidos e atualizados os currículos escolares?”.
- “30 - Existe um sistema para criar, armazenar e compartilhar planos de aula entre os professores?”.
- “31 - Quais recursos são necessários para facilitar o processo de ensino-aprendizagem? (Livros, materiais, tecnologias, etc.)”.
- “32 - Como são planejadas e gerenciadas as atividades extracurriculares?”.
- “33 - Como são identificados e acompanhados os alunos com necessidades especiais?”.
- “33 - Como são identificados e acompanhados os alunos com necessidades especiais?”.
- “34 - Quais são as necessidades relacionadas à integração de tecnologia na sala de aula?”.
- “35 - Quanto ao cadastro de professores em específico. Como é realizado o processo, e quais são os requisitos mínimos?”.

### 3.2 Requisitos Coletados

Os requisitos funcionais para o projeto de dados são:

<b>Identificador</b>	RF001		
<b>Nome</b>	Cadastro de Alunos		
<b>Módulo</b>	Cadastro		
<b>Data de Criação</b>	31/10/2023	<b>Autor</b>	Matheus Lopes

<b>Data da última alteração</b>	31/10/2023	<b>Autor</b>	Matheus Lopes
<b>Versão</b>	1	<b>Prioridade</b>	Alta
<b>Descrição</b>	O sistema deve permitir o completo registro de dados dos alunos, abrangendo informações pessoais, como o histórico acadêmico, contatos de responsáveis legais e demais dados relevantes.		

<b>Identificador</b>	RF002		
<b>Nome</b>	Cadastro de Professores		
<b>Módulo</b>	Cadastro		
<b>Data de Criação</b>	31/10/2023	<b>Autor</b>	Matheus Lopes
<b>Data da última alteração</b>	31/10/2023	<b>Autor</b>	Matheus Lopes
<b>Versão</b>	1	<b>Prioridade</b>	Alta
<b>Descrição</b>	O sistema deve viabilizar o cadastro e atualização dos dados dos professores, incluindo detalhes pessoais, formação acadêmica, alocação de disciplinas e horários correspondentes.		

<b>Identificador</b>	RF003		
<b>Nome</b>	Cadastro de Disciplinas		
<b>Módulo</b>	Cadastro		
<b>Data de Criação</b>	31/10/2023	<b>Autor</b>	Matheus Lopes
<b>Data da última alteração</b>	31/10/2023	<b>Autor</b>	Matheus Lopes
<b>Versão</b>	1	<b>Prioridade</b>	Alta
<b>Descrição</b>	O sistema deve possibilitar o cadastro e controle das disciplinas oferecidas pela instituição, definição de horários de aulas e atribuição dos docentes responsáveis.		

<b>Identificador</b>	RF004		
<b>Nome</b>	Cadastro de Matrículas		
<b>Módulo</b>	Cadastro		
<b>Data de Criação</b>	31/10/2023	<b>Autor</b>	Matheus Lopes

<b>Data da última alteração</b>	31/10/2023	<b>Autor</b>	Matheus Lopes
<b>Versão</b>	1	<b>Prioridade</b>	Alta
<b>Descrição</b>	O sistema deve gerenciar o processo de matrícula dos alunos em disciplinas, organização de turmas e distribuição dos estudantes conforme a grade curricular.		

<b>Identificador</b>	RF005		
<b>Nome</b>	Cadastro de Notas		
<b>Módulo</b>	Cadastro		
<b>Data de Criação</b>	31/10/2023	<b>Autor</b>	Matheus Lopes
<b>Data da última alteração</b>	31/10/2023	<b>Autor</b>	Matheus Lopes
<b>Versão</b>	1	<b>Prioridade</b>	Alta
<b>Descrição</b>	O sistema deve permitir o registro e acompanhamento das notas e avaliações dos alunos em cada disciplina, facilitando a geração de relatórios de desempenho escolar.		

<b>Identificador</b>	RF006		
<b>Nome</b>	Cadastro de Frequência		
<b>Módulo</b>	Cadastro		
<b>Data de Criação</b>	31/10/2023	<b>Autor</b>	Matheus Lopes
<b>Data da última alteração</b>	31/10/2023	<b>Autor</b>	Matheus Lopes
<b>Versão</b>	1	<b>Prioridade</b>	Alta
<b>Descrição</b>	O sistema deve registrar e acompanhar a presença dos alunos em cada aula, possibilitando a manutenção do controle de frequência.		

<b>Identificador</b>	RF007		
<b>Nome</b>	Reserva de Espaços		
<b>Módulo</b>	Cadastro		
<b>Data de Criação</b>	31/10/2023	<b>Autor</b>	Matheus Lopes
<b>Data da última alteração</b>	31/10/2023	<b>Autor</b>	Matheus Lopes

<b>Versão</b>	1	<b>Prioridade</b>	Alta
<b>Descrição</b>	O sistema deve permitir a reserva e administração de espaços como salas de aula, laboratórios e espaços comuns, garantindo a disponibilidade e organização para as atividades escolares.		

<b>Identificador</b>	RF008		
<b>Nome</b>	Cadastro de Atividades Extracurriculares		
<b>Módulo</b>	Cadastro		
<b>Data de Criação</b>	31/10/2023	<b>Autor</b>	Matheus Lopes
<b>Data da última alteração</b>	31/10/2023	<b>Autor</b>	Matheus Lopes
<b>Versão</b>	1	<b>Prioridade</b>	Média
<b>Descrição</b>	O sistema deve oferecer suporte ao cadastro e gerenciamento de atividades extracurriculares, como clubes, grupos de estudo e eventos acadêmicos.		

<b>Identificador</b>	RF009		
<b>Nome</b>	Gestão de Finanças		
<b>Módulo</b>	Financeiro		
<b>Data de Criação</b>	31/10/2023	<b>Autor</b>	Matheus Lopes
<b>Data da última alteração</b>	31/10/2023	<b>Autor</b>	Matheus Lopes
<b>Versão</b>	1	<b>Prioridade</b>	Alta
<b>Descrição</b>	O sistema deve possibilitar o controle financeiro da instituição, incluindo gestão de mensalidades e pagamentos, contas à pagar e contas à receber.		

<b>Identificador</b>	RF010		
<b>Nome</b>	Cadastro de Departamentos		
<b>Módulo</b>	Cadastro		
<b>Data de Criação</b>	31/10/2023	<b>Autor</b>	Matheus Lopes
<b>Data da última alteração</b>	31/10/2023	<b>Autor</b>	Matheus Lopes
<b>Versão</b>	1	<b>Prioridade</b>	Alta
<b>Descrição</b>	O sistema deve permitir o cadastro de diferentes		

	departamentos de uma escola.
--	------------------------------

<b>Identificador</b>	RF011		
<b>Nome</b>	Cadastro de Funcionários de Outros Departamentos		
<b>Módulo</b>	Cadastro		
<b>Data de Criação</b>	31/10/2023	<b>Autor</b>	Matheus Lopes
<b>Data da última alteração</b>	31/10/2023	<b>Autor</b>	Matheus Lopes
<b>Versão</b>	1	<b>Prioridade</b>	Alta
<b>Descrição</b>	O sistema deve viabilizar o cadastro de funcionários dos departamentos criados.		

<b>Identificador</b>	RF012		
<b>Nome</b>	Cadastro de Alunos		
<b>Módulo</b>	Gestão		
<b>Data de Criação</b>	31/10/2023	<b>Autor</b>	Matheus Lopes
<b>Data da última alteração</b>	31/10/2023	<b>Autor</b>	Matheus Lopes
<b>Versão</b>	1	<b>Prioridade</b>	Alta
<b>Descrição</b>	O sistema deve permitir o cadastro de fornecedores de materiais escolares.		

<b>Identificador</b>	RF013		
<b>Nome</b>	Cadastro de Vans Escolares		
<b>Módulo</b>	Cadastros		
<b>Data de Criação</b>	31/10/2023	<b>Autor</b>	Matheus Lopes
<b>Data da última alteração</b>	31/10/2023	<b>Autor</b>	Matheus Lopes
<b>Versão</b>	1	<b>Prioridade</b>	Alta
<b>Descrição</b>	O sistema deve permitir o cadastro de vans escolares e seus motoristas.		

<b>Identificador</b>	RF014		
<b>Nome</b>	Cadastro de Responsáveis		
<b>Módulo</b>	Cadastros		
<b>Data de Criação</b>	31/10/2023	<b>Autor</b>	Matheus Lopes

<b>Data da última alteração</b>	31/10/2023	<b>Autor</b>	Matheus Lopes
<b>Versão</b>	1	<b>Prioridade</b>	Alta
<b>Descrição</b>	O sistema deve permitir o cadastro dos responsáveis de cada um dos alunos.		

<b>Identificador</b>	RF015		
<b>Nome</b>	Cadastro de Eventos		
<b>Módulo</b>	Cadastros		
<b>Data de Criação</b>	31/10/2023	<b>Autor</b>	Matheus Lopes
<b>Data da última alteração</b>	31/10/2023	<b>Autor</b>	Matheus Lopes
<b>Versão</b>	1	<b>Prioridade</b>	Alta
<b>Descrição</b>	O sistema deve permitir o cadastro de eventos realizados pela escola.		

### 3.3 Regras de Negócio

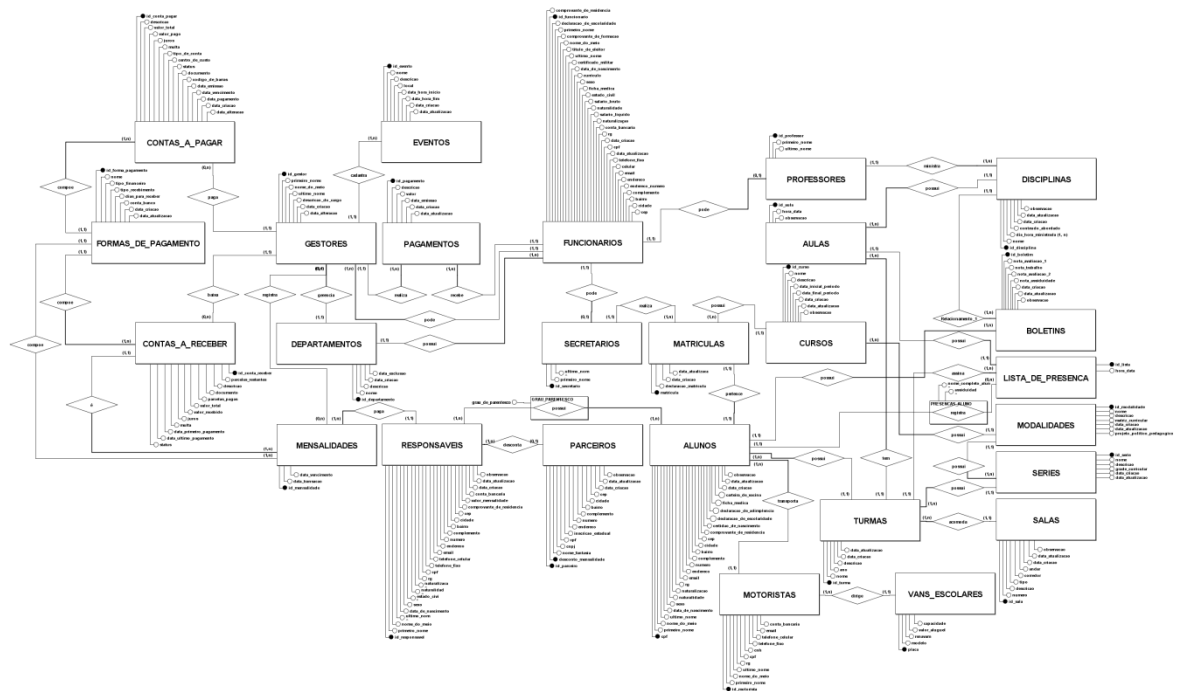
De acordo com as informações fornecidas pelos gestores da instituição de ensino Colégio 24 de Maio, as regras de negócio são as seguintes:

- A secretaria realiza o cadastro dos alunos, solicitando o seu nome completo, sexo, os seus documentos, como CPF, RG, e Certidão de Nascimento, ficha médica, comprovante de residência, seu responsável financeiro, os contatos do responsável, documentos referentes a transferência escolar, foto do aluno, e última escola;
- O gestor do departamento financeiro e administrativo realiza o controle de contas a receber e contas a pagar, como mensalidades e despesas;
- A escola possui diversas modalidades de ensino, as quais possuem as suas séries, turmas, e alunos;
- No cadastro do aluno, também é necessária a informação de qual motorista o leva ou não para a escola. Além disso, os motoristas não são funcionários contratados diretamente pela instituição;

- Para os funcionários serem cadastrados pelo RH são necessários o seu nome completo, sexo, seu estado-civil, naturalidade, comprovante de residência, contatos, declaração de escolaridade, comprovante de formação, título de eleitor, certificado militar, currículo, e a sua ficha médica;
- A nota final dos alunos é composta pela nota da primeira prova, da segunda prova, de um trabalho, e de assiduidade;
- Um responsável de aluno que realiza o cadastro em um parceiro da instituição recebe desconto nas mensalidades;
- Cada departamento possui apenas um gestor;
- Cada professor pode lecionar uma ou mais disciplinas, enquanto cada disciplina pode apenas ser lecionada por um professor;
- Cada turma possui uma única sala para ser utilizada ao longo do período letivo;
- Cada evento na instituição pode ser criado pelos gestores dos departamentos;
- Cada aula deve possuir uma lista de presença;
- A instituição aceita as formas de pagamento em boleto, Cartão de Crédito e Débito, além de aceitarem PIX somente em casos autorizados pelo responsável financeiro.
- A matriz curricular das modalidades de ensino, e a grade curricular de cada série escolar são criadas, digitalizadas e armazenadas no registro de cada uma delas.

### **3.4 Modelo Conceitual**

O Modelo Conceitual do banco de dados criado neste projeto foi desenvolvido no software de modelagem BrModelo, utilizando a notação de Carlos Alberto Heuser.



**Figura 1 – Modelo Conceitual (Acervo pessoal)**

### 3.5 Modelo Lógico

O Modelo Lógico do banco de dados criado neste projeto foi desenvolvido no software de modelagem BrModelo, utilizando a notação de Carlos Alberto Heuser, utilizando a funcionalidade do software de gerar o modelo a partir do Modelo Conceitual.



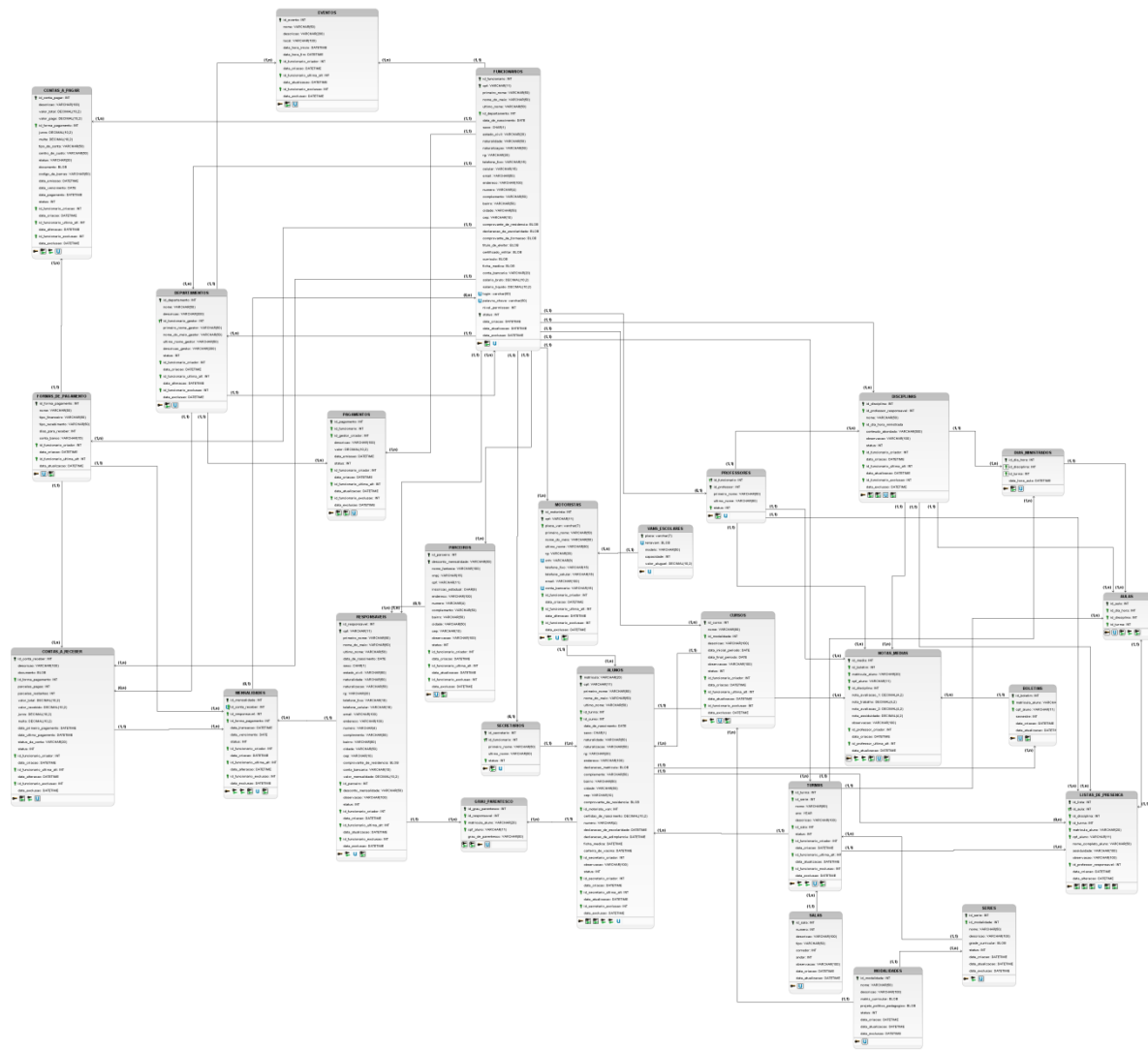


Figura 2 – Modelo Lógico (Acervo pessoal)

3.5.1 Dicionário de Dados

O quadro abaixo retrata o Dicionário de Dados das tabelas concebidas neste projeto:

Quadro 1 – Dicionário de Dados

TABELA	ATRIBUTO	CONTEÚDO	TIPOS DE DADOS	PK/FK	TABELA REFERENCIADA FK
AULAS	id_aula	Código único da aula	INT UNIQUE	PK	
	id_dia_hora	Código do dia e horário da aula	INT	PK,FK	DIAS_MINISTRADOS
	id_disciplina	Código da disciplina aplicada	INT	PK,FK	DISCIPLINAS
	id_turma	Código da turma que participou	INT	PK,FK	TURMAS

BOLETINS	id_boletim	Código único do boletim	INT	PK	
	matricula_aluno	Matrícula do Aluno	VARCHAR(20)	FK	ALUNOS
	cpf_aluno	CPF do Aluno	VARCHAR(11)	FK	ALUNOS
	semestre	Número do Semestre do Boletim	INT		
	data_criacao	Data de cadastro do Boletim	DATETIME		
	data_atualizacao	Data de alteração do Boletim	DATETIME		
DIAS_MINISTRADOS	id_dia_hora	Código único do dia ministrado	INT UNIQUE	PK	
	id_disciplina	Código da disciplina aplicada	INT	FK	DISCIPLINAS
	id_turma	Código da turma da aula	INT	FK	TURMAS
	data_hora_aula	Data e hora que será aplicada	DATETIME		
DISCIPLINAS	id_disciplina	Código único da disciplina	INT UNIQUE	PK	
	id_professor_responsavel	Código do professor responsável pela disciplina		FK	
	nome	Nome da disciplina	VARCHAR(50)		
	id_dia_hora_ministrada	Data e hora a qual será ministrada		FK	
	conteudo_abordado	Conteúdo abordado na disciplina	VARCHAR(500)		
	observacao	Observação sobre a disciplina	VARCHAR(100)		
	status	Status da disciplina. Se está ativa ou inativa.	TINYINT CHECK (status IN 0, 1)		
	id_funcionario_criador	Funcionário que cadastrou	INT	FK	FUNCIONARIOS
	data_criacao	Data do cadastro	DATETIME		
	id_funcionario_ultima_alt	Último funcionário a alterar o cadastro	INT	FK	FUNCIONARIOS
	data_atualizacao	Data da última alteração	DATETIME		
	id_funcionario_exclusao	Funcionário que excluiu o cadastro	INT	FK	FUNCIONARIOS
	data_exclusao	Data da exclusão	DATETIME		
NOTAS_MEDIAS	id_media	Código único da média	INT UNIQUE	PK	
	id_boletim	Código do boletim que a média faz parte	INT	FK	BOLETINS
	matricula_aluno	Matrícula do aluno	VARCHAR(20)	FK	ALUNOS
	cpf_aluno	CPF do aluno	VARCHAR(11)	FK	ALUNOS
	id_disciplina	Código da disciplina	INT	FK	DISCIPLINAS
	nota_avaliacao_1	Nota da primeira avaliação	DECIMAL(4,2)		
	nota_trabalho	Nota de trabalho	DECIMAL(4,2)		
	nota_avaliacao_2	Nota da segunda avaliação	DECIMAL(4,2)		
	nota_assiduidade	Nota de assiduidade	DECIMAL(4,2)		
	observacao	Observação sobre as notas	VARCHAR(100)		
	id_professor_cr	Professor que cadastrou	INT	FK	PROFESSORES

	criador				
	data_criacao	Data do cadastro	DATETIME		
	id_professor_ultima_alt	Último professor a alterar o cadastro	INT	FK	PROFESSORES
	data_atualizacao	Data da última alteração	DATETIME		
LISTAS DE PRESENÇA	id_lista	Nome da disciplina	INT UNIQUE	PK	
	id_aula	Data e hora a qual será ministrada	INT	FK	AULAS
	id_disciplina	Conteúdo abordado na disciplina	INT	FK	DISCIPLINAS
	id_turma	Observação sobre a disciplina	INT	FK	TURMAS
	matricula_aluno	Matrícula do aluno	VARCHAR(20)	FK	ALUNOS
	cpf_aluno	CPF do aluno	VARCHAR(11)	FK	ALUNOS
	nome_completo_aluno	Nome completo do aluno	VARCHAR(50)		
	assiduidade	Presença do aluno	VARCHAR(100)		
	observacao	Observação sobre o aluno na aula	VARCHAR(100)		
	id_professor_responsavel	Professor que cadastrou e alterou a lista	INT	FK	PROFESSORES
	data_criacao	Data do cadastro	DATETIME		
	data_atualizacao	Data da última alteração	DATETIME		
CURSOS	id_curso	Código único do curso	INT UNIQUE	PK	
	nome	Nome do Curso	VARCHAR(50)		
	id_modalidade	Código da Modalidade do Curso	INT	FK	MODALIDADES
	descricao	Descrição do curso	VARCHAR(100)		
	data_inicial_periodo	Data inicial do período do curso	VARCHAR(20)		
	data_final_periodo	Data final do período do curso	VARCHAR(11)		
	observacao	Observações sobre o curso	VARCHAR(50)		
	grade_curricular	Grade Curricular do curso	BLOB		
	status	Status do curso. Se está ativo ou inativo.	TINYINT CHECK (status IN 0, 1)		
	id_funcionario_criador	Funcionário que cadastrou	INT	FK	FUNCIONARIOS
	data_criacao	Data do cadastro	DATETIME		
	id_funcionario_ultima_alt	Último funcionário a alterar o cadastro	INT	FK	FUNCIONARIOS
	data_atualizacao	Data da última alteração	DATETIME		
	id_funcionario_exclusao	Funcionário que excluiu o cadastro	INT	FK	FUNCIONARIOS
	data_exclusao	Data da exclusão	DATETIME		
	id_modalidade	Código único da modalidade	INT UNIQUE	PK	
	nome	Nome da modalidade	VARCHAR(50)		
	descricao	Descrição da modalidade	VARCHAR(		

MODALIDADES	matriz_curricular	Matriz curricular da modalidade	100) BLOB		
	status	Status da modalidade. Se está ativa ou inativa.	TINYINT CHECK (status IN 0, 1)		
	observacao	Nome completo do aluno	VARCHAR(50)		
	status	Status da modalidade. Se está ativa ou inativa.	TINYINT CHECK (status IN 0, 1)		
	data_criacao	Data do cadastro	DATETIME		
	data_atualizacao	Data da última alteração	DATETIME		
	data_exclusao	Data da exclusão	DATETIME		
SERIES	id_serie	Código único da série escolar	INT UNIQUE	PK	
	id_modalidade	Código da modalidade	INT	FK	MODALIDADES
	nome	Nome da série escolar	VARCHAR(50)		
	descricao	Descrição da série escolar	VARCHAR(100)		
	grade_curricular	Grade curricular da série	BLOB		
	status	Status da série. Se está ativa ou inativa.	TINYINT CHECK (status IN 0, 1)		
	data_criacao	Data do cadastro	DATETIME		
	data_atualizacao	Data da última alteração	DATETIME		
	data_exclusao	Data da exclusão	DATETIME		
TURMAS	id_turma	Código único da disciplina	INT UNIQUE	PK	
	id_serie	Código da série que a turma se matriculou	INT	FK	SERIES
	nome	Nome da turma	VARCHAR(50)		
	ano	Ano	YEAR		
	descricao	Descrição sobre a turma	VARCHAR(100)		
	id_sala	Código da sala que a turma utiliza	INT	FK	SALAS
	status	Status da disciplina. Se está ativa ou inativa.	TINYINT CHECK (status IN 0, 1)		
	id_funcionario_criador	Funcionário que cadastrou	INT	FK	FUNCIONARIOS
	data_criacao	Data do cadastro	DATETIME		
	id_funcionario_ultima_alt	Último funcionário a alterar o cadastro	INT	FK	FUNCIONARIOS
	data_atualizacao	Data da última alteração	DATETIME		
	id_funcionario_exclusao	Funcionário que excluiu o cadastro	INT	FK	FUNCIONARIOS
	data_exclusao	Data da exclusão	DATETIME		
	id_sala	Código único da sala	INT UNIQUE	PK	
	numero	Número da sala	INT		

SALAS	descricao	Descrição da sala	VARCHAR(100)		
	tipo	Tipo da sala	VARCHAR(50)		
	corredor	Corredor onde está localizada	INT		
	andar	Andar onde está localizada	INT		
	observacao	Observação sobre a sala	VARCHAR(100)		
	data_criacao	Data do cadastro	DATETIME		
	data_atualizacao	Data da última alteração	DATETIME		
VANS_ESCOLARES	placa	Placa do veículo	INT UNIQUE	PK	
	renavam	RENAVAM do veículo	INT UNIQUE		
	modelo	Modelo do Veículo	VARCHAR(50)		
	capacidade	Capacidade do veículo	INT		
	valor_aluguel	Valor do aluguel da van	DECIMAL(10,2)		
MOTORISTAS	id_motorista	Código único do motorista	INT UNIQUE	PK	
	cpf	CPF do motorista	VARCHAR(11) UNIQUE	PK	
	placa_van	Placa da van que dirige	VARCHAR(7)	FK	VANS_ESCOLARES
	primeiro_nome	Primeiro nome do motorista	VARCHAR(50)		
	nome_do_meio	Nome do meio do motorista	VARCHAR(50)		
	ultimo_nome	Último nome do motorista	VARCHAR(50)		
	rg	RG do motorista	VARCHAR(20)		
	cnh	CNH do motorista	VARCHAR(9) UNIQUE		
	telefone_fixo	Telefone fixo do motorista	VARCHAR(15)		
	telefone_celular	Celular do motorista	VARCHAR(15)		
	email	Email de contato do motorista	VARCHAR(100)		
	conta_bancaria	Agência e Conta Bancária do Motorista	VARCHAR(15) UNIQUE		
	id_funcionario_criador	Funcionário que cadastrou	INT	FK	FUNCIONARIOS
	data_criacao	Data do cadastro	DATETIME		
	id_funcionario_ultima_alt	Último funcionário a alterar o cadastro	INT	FK	FUNCIONARIOS
	data_atualizacao	Data da última alteração	DATETIME		
	id_funcionario_exclusao	Funcionário que excluiu o cadastro	INT	FK	FUNCIONARIOS
	data_exclusao	Data da exclusão	DATETIME		
GRAU_PARENTESCO	id_grau_parentesco	Código único do grau de parentesco	INT UNIQUE	PK	
	id_responsavel	Código do responsável pelo aluno	INT	FK	RESPONSAVEIS
	matricula_aluno	Matrícula do aluno	VARCHAR(20)	FK	ALUNOS
	cpf_aluno	CPF do aluno	VARCHAR(11)	FK	ALUNOS

			11)		
	grau_de_paren tesco	Descrição do grau de parentesco	VARCHAR( 50)		
ALUNOS	matricula	Código único do aluno	VARCHAR( 20) UNIQUE	PK	
	cpf	CPF do aluno	VARCHAR( 11) UNIQUE		
	primeiro_nome	Primeiro nome do aluno	VARCHAR( 50)		
	nome_do_meio	Nome do meio do aluno	VARCHAR( 50)		
	ultimo_nome	Último nome do aluno	VARCHAR( 50)		
	id_turma	Código da turma do aluno	INT	FK	TURMAS
	id_curso	Código do curso da matrícula	INT	FK	CURSOS
	data_de_nasci mento	Data de nascimento do aluno	DATE		
	sexo	Sexo do aluno	CHAR(1)		
	naturalidade	Naturalidade do aluno	VARCHAR( 50)		
	naturalizacao	Naturalização do aluno	VARCHAR( 50)		
	rg	RG do aluno	VARCHAR( 20)		
	endereco	Endereço	VARCHAR( 100)		
	numero	Número	VARCHAR( 4)		
	complemento	Complemento	VARCHAR( 50)		
	bairro	Bairro	VARCHAR( 50)		
	cidade	Cidade	VARCHAR( 50)		
	cep	CEP	VARCHAR( 10)		
	comprovante_d e_residencia	Documento que comprove a residência	BLOB		
	declaracao_de _matricula	Declaração de matrícula	BLOB		
	id_motorista_v an	Motorista que leva o aluno de van	INT	FK	MOTORISTAS
	certidao_de_na scimento	Certidão de Nascimento do aluno	BLOB		
	declaracao_de _escolaridade	Declaração de escolaridade	BLOB		
	declaracao_de _adimplencia	Declaração de adimplência	BLOB		
	ficha medica	Ficha médica	BLOB		
	carteira_de_va cina	Carteira de vacina	BLOB		
	observacao	Observação sobre o aluno	VARCHAR( 100)		
	status	Status do aluno, se está ativo ou inativo	TINYINT CHECK (status IN 0, 1)		
	id_secretario_c riador	Secretário que realizou o cadastro	INT	FK	SECRETARIOS
	data_criacao	Data do cadastro	DATETIME		
	id_secretario_u	Secretário que realizou a última	INT	FK	SECRETARIOS

	ltima_alt	alteração no cadastro			
	data_atualizacao	Data da atualização do cadastro	DATETIME		
	id_secretario_exclusao	Secretário que realizou a exclusão do aluno	INT	FK	SECRETARIOS
	data_exclusao	Data de exclusão do cadastro	DATETIME		
PROFESSORES	id_funcionario	Código único do funcionário	INT UNIQUE	PK,FK	FUNCIONARIOS
	id_professor	Código do professor	INT UNIQUE	PK	
	primeiro_nome	Primeiro nome do professor	VARCHAR(50)		
	ultimo_nome	Último nome do professor	VARCHAR(50)		
	status	Status do professor, se está ativo ou inativo	TINYINT CHECK (status IN 0, 1)	FK	FUNCIONARIOS
SECRETARIOS	id_funcionario	Código único do funcionário	INT UNIQUE	PK,FK	FUNCIONARIOS
	id_secretario	Código do secretário	INT UNIQUE	PK	
	primeiro_nome	Primeiro nome do secretário	VARCHAR(50)		
	ultimo_nome	Último nome do secretário	VARCHAR(50)		
	status	Status do secretário, se está ativo ou inativo	TINYINT CHECK (status IN 0, 1)	FK	FUNCIONARIOS
PARCEIROS	id_parceiro	Código único do parceiro	VARCHAR(20) UNIQUE	PK	
	desconto_mensalidade	Desconto fornecido através da parceria	DECIMAL(4,2)	PK	
	nome_fantasia	Nome Fantasia da empresa	VARCHAR(50)		
	cnpj	CNPJ do parceiro	VARCHAR(15)		
	cpf	CPF do parceiro	VARCHAR(11) UNIQUE		
	inscricao_estadual	Inscrição Estadual da empresa	VARCHAR(9)		
	endereco	Endereço	VARCHAR(100)		
	numero	Número	VARCHAR(4)		
	complemento	Complemento	VARCHAR(50)		
	bairro	Bairro	VARCHAR(50)		
	cidade	Cidade	VARCHAR(50)		
	cep	CEP	VARCHAR(10)		
	observacao	Observação sobre a parceria	VARCHAR(100)		
	status	Status do parceiro, se está ativo ou inativo	TINYINT CHECK (status IN 0, 1)		
	id_funcionario_criador	Funcionário que realizou o cadastro	INT	FK	FUNCIONARIOS
	data_criacao	Data do cadastro	DATETIME		
	id_funcionario_ultima_alt	Funcionário que realizou a última alteração no cadastro	INT	FK	FUNCIONARIOS
	data_atualizacao	Data da atualização do cadastro	DATETIME		
	id_funcionario_exclusao	Funcionário que realizou a exclusão do aluno	INT	FK	FUNCIONARIOS
	data_exclusao	Data de exclusão do cadastro	DATETIME		

RESPONSÁVEIS	id_responsavel	Código único do responsável	VARCHAR(20) UNIQUE	PK	
	cpf	CPF do responsável	VARCHAR(11) UNIQUE	PK	
	primeiro_nome	Primeiro nome do responsável	VARCHAR(50)		
	nome_do_meio	Nome do meio do responsável	VARCHAR(50)		
	ultimo_nome	Último nome do responsável	VARCHAR(50)		
	data_de_nascimento	Data de nascimento do responsável	DATE		
	sexo	Sexo do responsável	CHAR(1)		
	naturalidade	Naturalidade do responsável	VARCHAR(50)		
	naturalizacao	Naturalização do responsável	VARCHAR(50)		
	rg	RG do responsável	VARCHAR(20)		
	endereco	Endereço	VARCHAR(100)		
	numero	Número	VARCHAR(4)		
	complemento	Complemento	VARCHAR(50)		
	bairro	Bairro	VARCHAR(50)		
	cidade	Cidade	VARCHAR(50)		
	cep	CEP	VARCHAR(10)		
	comprovante_de_residencia	Documento que comprove a residência	BLOB		
	conta_bancaria	Agência e Conta Bancária do responsável	VARCHAR(15)		
	valor_mensalidade	Valor Bruto da mensalidade paga	DECIMAL(4,2)		
	id_parceiro	Código do parceiro caso tenha chegado a escola através de indicação	INT	FK	PARCEIROS
	desconto_na_mensalidade	Desconto obtido através de parceiros	DECIMAL(4,2)	FK	PARCEIROS
	status	Status do responsável, se está ativo ou inativo	TINYINT CHECK (status IN 0, 1)		
	id_funcionario_criador	Funcionário que realizou o cadastro	INT	FK	FUNCIONARIOS
	data_criacao	Data do cadastro	DATETIME		
	id_funcionario_ultima_atualizacao	Funcionário que realizou a última alteração no cadastro	INT	FK	FUNCIONARIOS
	data_atualizacao	Data da atualização do cadastro	DATETIME		
	id_funcionario_exclusao	Funcionário que realizou a exclusão do aluno	INT	FK	FUNCIONARIOS
	data_exclusao	Data de exclusão do cadastro	DATETIME		
	id_mensalidade	Código Único da mensalidade	INT UNIQUE	PK	
	id_conta_a_receber	Código da Conta a Receber	INT	FK	CONTAS_A_RECEBER
	id_responsavel	Código do Responsável	INT	FK	RESPONSÁVEIS
	id_forma_de_pagamento	Forma de pagamento utilizada	INT	FK	FORMAS_DE_PAGAMENTO



<b>MENSALIDADES</b>	agamento				MENTO
	data_transacao	Data de pagamento da transação	DATETIME		
	data_de_vencimento	Data de vencimento da mensalidade	DATE		
	status	Status da mensalidade, se está ativa ou inativa	TINYINT CHECK (status IN 0, 1)		
	id_funcionario_criador	Funcionário que cadastrou	INT	FK	FUNCIONARIOS
	data_criacao	Data do cadastro	DATETIME		
	id_funcionario_ultima_alt	Último funcionário a alterar o cadastro	INT	FK	FUNCIONARIOS
	data_atualizacao	Data da última alteração	DATETIME		
	id_funcionario_exclusao	Funcionário que excluiu o cadastro	INT	FK	FUNCIONARIOS
	data_exclusao	Data da exclusão	DATETIME		
<b>FORMAS_DE_PAGAMENTO</b>	id_forma_de_pagamento	Código único da forma de pagamento	INT UNIQUE	PK	
	nome	Nome da forma de pagamento	VARCHAR(50)		
	tipo_financeiro	Tipo financeiro da forma de pagamento	VARCHAR(50)		
	tipo_recebimento	Tipo de recebimento da forma de pagamento	VARCHAR(50)		
	dias_para_receber	Dias para receber o pagamento da adquirente	INT		
	conta_banco	Conta recebedora	VARCHAR(15)		
	id_funcionario_criador	Funcionário que cadastrou	INT	FK	FUNCIONARIOS
	data_criacao	Data do cadastro	DATETIME		
	id_funcionario_ultima_alt	Último funcionário a alterar o cadastro	INT	FK	FUNCIONARIOS
	data_atualizacao	Data da última alteração	DATETIME		
<b>CONTAS_A_RECEBER</b>	id_conta_a_receber	Código único da conta	INT UNIQUE	PK	
	descricao	Descrição da conta	VARCHAR(100)		
	documento	Documento da conta	BLOB		
	id_forma_de_pagamento	Forma de pagamento utilizada	INT	FK	FORMAS_DE_PAGAMENTO
	parcelas_pagas	Parcelas pagas pelo cliente	INT		
	parcelas_restantes	Parcelas restantes para o cliente pagar	INT		
	valor_total	Valor total da conta	DECIMAL(10,2)		
	valor_recebido	Valor já recebido do cliente	DECIMAL(10,2)		
	juros	Juros em R\$	DECIMAL(10,2)		
	multa	Multa em R\$	DECIMAL(10,2)		
	data_primeiro_pagamento	Data do primeiro pagamento feito pelo cliente	DATETIME		
	data_ultimo_pagamento	Data do último pagamento feito pelo cliente	DATETIME		
	status_da_conta	Descrição sobre o status atual da conta	VARCHAR(20)		

	status	Status da conta, se está ativa ou inativa	TINYINT CHECK (status IN (0, 1))		
	id_funcionario_criador	Funcionário que cadastrou	INT	FK	FUNCIONARIOS
	data_criacao	Data do cadastro	DATETIME		
	id_funcionario_ultima_alt	Último funcionário a alterar o cadastro	INT	FK	FUNCIONARIOS
	data_atualizacao	Data da última alteração	DATETIME		
CONTAS_A_PAGAR	id_conta_a_pagar	Código único da conta	INT UNIQUE	PK	
	descricao	Descrição da conta	VARCHAR(100)		
	documento	Documento da conta	BLOB		
	id_forma_de_pagamento	Forma de pagamento utilizada	INT	FK	FORMAS_DE_PAGAMENTO
	parcelas_pagas	Parcelas pagas pelo cliente	INT		
	parcelas_restantes	Parcelas restantes para o cliente pagar	INT		
	valor_total	Valor total da conta	DECIMAL(10,2)		
	valor_pago	Valor já pago	DECIMAL(10,2)		
	juros	Juros em R\$	DECIMAL(10,2)		
	multa	Multa em R\$	DECIMAL(10,2)		
	tipo_de_conta	Data do primeiro pagamento feito pelo cliente	DATETIME		
	centro_de_custo	Data do último pagamento feito pelo cliente	DATETIME		
	status_da_conta	Descrição sobre o status atual da conta	VARCHAR(20)		
	codigo_de_barras	Código de Barras da conta	VARCHAR(50)		
	data_emissao	Data de emissão da conta	DATETIME		
	data_vencimento	Data de vencimento da conta	DATE		
	data_pagamento	Data de pagamento da conta	DATETIME		
	status	Status da conta, se está ativa ou inativa	TINYINT CHECK (status IN (0, 1))		
	id_funcionario_criador	Funcionário que cadastrou	INT	FK	FUNCIONARIOS
	data_criacao	Data do cadastro	DATETIME		
	id_funcionario_ultima_alt	Último funcionário a alterar o cadastro	INT	FK	FUNCIONARIOS
	data_atualizacao	Data da última alteração	DATETIME		
	id_pagamento	Código único do pagamento	INT UNIQUE	PK	
	id_funcionario	Código único do funcionário recebedor	INT	FK	FUNCIONARIOS
	id_gestor_criador	Código do gestor que cadastrou o pagamento	VARCHAR(100)	FK	DEPARTAMENTOS
	valor	Valor já pago	DECIMAL(10,2)		
	data_emissao	Data de emissão da conta	DATETIME		

PAGAMENTOS	status	Status do pagamento, se está ativa ou inativa	TINYINT CHECK (status IN (0, 1))		
	id_funcionario_criador	Funcionário que cadastrou	INT	FK	FUNCIONARIOS
	data_criacao	Data do cadastro	DATETIME		
	id_funcionario_ultima_alt	Último funcionário a alterar o cadastro	INT	FK	FUNCIONARIOS
	data_atualizacao	Data da última alteração	DATETIME		
EVENTOS	id_evento	Código Único do evento	INT UNIQUE	PK	
	nome	Nome do evento	VARCHAR(50)		
	descricao	Descrição sobre o evento	VARCHAR(200)		
	local	Local onde será realizado	VARCHAR(100)		
	data_hora_inicio	Data e hora de início	DATETIME		
	data_hora_fim	Data e hora do fim	DATETIME		
	status	Status do evento, se está ativo ou inativo	TINYINT CHECK (status IN 0, 1)		
	id_funcionario_criador	Funcionário que cadastrou	INT	FK	FUNCIONARIOS
	data_criacao	Data do cadastro	DATETIME		
	id_funcionario_ultima_alt	Último funcionário a alterar o cadastro	INT	FK	FUNCIONARIOS
	data_atualizacao	Data da última alteração	DATETIME		
	id_funcionario_exclusao	Funcionário que excluiu o cadastro	INT	FK	FUNCIONARIOS
	data_exclusao	Data da exclusão	DATETIME		
DEPARTAMENTOS	id_departamento	Código único do departamento	INT UNIQUE	PK	
	nome	Nome do departamento	VARCHAR(50)		
	descricao	Descrição sobre as funções do departamento	VARCHAR(200)		
	id_funcionario_gestor	Código do funcionário que é gestor do departamento	INT	FK	FUNCIONARIOS
	primeiro_nome_gestor	Primeiro nome do gestor	VARCHAR(50)		
	nome_do_meio_gestor	Nome do meio do gestor	VARCHAR(50)		
	ultimo_nome_gestor	Último nome do gestor	VARCHAR(50)		
	descricao_gestor	Descrição sobre o gestor	VARCHAR(200)		
	status	Status do departamento, se está ativo ou inativo	TINYINT CHECK (status IN 0, 1)		
	id_funcionario_criador	Funcionário que cadastrou	INT	FK	FUNCIONARIOS
	data_criacao	Data do cadastro	DATETIME		
	id_funcionario_ultima_alt	Último funcionário a alterar o cadastro	INT	FK	FUNCIONARIOS
	data_atualizacao	Data da última alteração	DATETIME		

FUNCIONARIOS	id_funcionario_exclusao	Funcionário que excluiu o cadastro	INT	FK	FUNCIONARIOS
	data_exclusao	Data da exclusão	DATETIME		
	id_funcionario	Código único do funcionário	VARCHAR(20) UNIQUE	PK	
	cpf	CPF do aluno	VARCHAR(11) UNIQUE	PK	
	primeiro_nome	Primeiro nome do aluno	VARCHAR(50)		
	nome_do_meio	Nome do meio do aluno	VARCHAR(50)		
	ultimo_nome	Último nome do aluno	VARCHAR(50)		
	id_departament o	Código do curso da matrícula	INT	FK	DEPARTAMENTOS
	data_de_nasci mento	Data de nascimento do funcionário	DATE		
	sexo	Sexo do funcionário	CHAR(1)		
	naturalidade	Naturalidade do funcionário	VARCHAR(50)		
	naturalizacao	Naturalização do funcionário	VARCHAR(50)		
	rg	RG do funcionário	VARCHAR(20)		
	telefone_fixo	Telefone Fixo do funcionário	VARCHAR(15)		
	celular	Telefone Celular do funcionário	VARCHAR(15)		
	email	Email de contato do funcionário	VARCHAR(50)		
	endereco	Endereço	VARCHAR(100)		
	numero	Número	VARCHAR(4)		
	complemento	Complemento	VARCHAR(50)		
	bairro	Bairro	VARCHAR(50)		
	cidade	Cidade	VARCHAR(50)		
	cep	CEP	VARCHAR(10)		
	comprovante_d e_residencia	Comprovante de Residência	BLOB		
	declaracao_de _escolaridade	Declaração de Escolaridade do funcionário	BLOB		
	comprovante_d e_formacao	Comprovante de formação do funcionário	BLOB		
	titulo_de_eleitor	Título de Eleitor do funcionário	BLOB		
	certificado_milit ar	Certificado Militar do funcionário	BLOB		
	curriculo	Curriculo do funcionário	BLOB		
	ficha_medica	Ficha médica do funcionário	BLOB		
	salario_bruto	Salário bruto do funcionário	DECIMAL(10,2)		
	salario_liquido	Salário líquido do funcionário	DECIMAL(10,2)		
	login	Login do sistema	VARCHAR(50)		
	palavra_chave	Senha do login	VARCHAR(50)		

	nivel_de_permissao	Nível de permissão do funcionário no sistema	INT		
	status	Status do funcionário, se está ativo ou inativo	TINYINT CHECK (status IN 0, 1)	PK	
	data_criacao	Data do cadastro	DATETIME		
	data_atualizacao	Data da atualização do cadastro	DATETIME		
	data_exclusao	Data de exclusão do cadastro	DATETIME		

### 3.6 Modelo Físico

O Modelo Físico do banco de dados criado neste projeto foi desenvolvido no software de modelagem BrModelo, utilizando a notação de Carlos Alberto Heuser, utilizando a funcionalidade do software de gerar o modelo conceitual, lógico, e logo após gerar o modelo físico. Foram necessários alguns ajustes para adequar o código para o correto. A seguir está o modelo físico do bando de dados de Gestão Escolar:

#### Quadro 2 – Modelo Físico

```

/* Projeto de Banco de Dados 2 */
/* Mod_Lógico_Definitivo */
/* Matheus Lopes de Oliveira - C33023494 */

CREATE DATABASE escola_v1;

USE DATABASE escola_v1;

CREATE TABLE FUNCIONARIOS (
    id_funcionario INT,
    cpf VARCHAR(11),
    primeiro_nome VARCHAR(50),
    nome_do_meio VARCHAR(50),
    ultimo_nome VARCHAR(50),
    id_departamento INT,
    data_de_nascimento DATE,
    sexo CHAR(1),
    estado_civil VARCHAR(20),
    naturalidade VARCHAR(50),
    naturalizacao VARCHAR(50),
    rg VARCHAR(20),
    telefone_fixo VARCHAR(15),
    celular VARCHAR(15),
    email VARCHAR(50),

```

```

    endereco VARCHAR(100),
    numero VARCHAR(4),
    complemento VARCHAR(50),
    bairro VARCHAR(50),
    cidade VARCHAR(50),
    cep VARCHAR(10),
    comprovante_de_residencia BLOB,
    declaracao_de_escolaridade BLOB,
    comprovante_de_formacao BLOB,
    titulo_de_eleitor BLOB,
    certificado_militar BLOB,
    curriculo BLOB,
    ficha_medica BLOB,
    conta_bancaria VARCHAR(20),
    salario_bruto DECIMAL(10,2),
    salario_liquido DECIMAL(10,2),
    login varchar(50),
    palavra_chave varchar(50),
    nivel_permissao INT,
    status TINYINT CHECK (status IN (0, 1)),
    data_criacao DATETIME,
    data_atualizacao DATETIME,
    data_exclusao DATETIME,
    PRIMARY KEY (id_funcionario, status, cpf),
    UNIQUE (id_funcionario, login, palavra_chave, cpf),
    FOREIGN KEY (id_departamento)
    REFERENCES DEPARTAMENTOS (id_departamento)
    ON DELETE RESTRICT
);

CREATE TABLE DEPARTAMENTOS (
    id_departamento INT UNIQUE,
    nome VARCHAR(50),
    descricao VARCHAR(200),
    id_funcionario_gestor INT,
    primeiro_nome_gestor VARCHAR(50),
    nome_do_meio_gestor VARCHAR(50),
    ultimo_nome_gestor VARCHAR(50),
    descricao_gestor VARCHAR(200),
    status TINYINT CHECK (status IN (0, 1)),
    id_funcionario_criador INT,
    data_criacao DATETIME,
    id_funcionario_ultima_alt INT,
    data_alteracao DATETIME,
    id_funcionario_exclusao INT,
    data_exclusao DATETIME,
    PRIMARY KEY (id_departamento, id_funcionario_gestor)
    FOREIGN KEY (id_funcionario_gestor, id_funcionario_exclusao,
id_funcionario_ultima_alt, id_funcionario_criador)

```

```

    REFERENCES FUNCIONARIOS (id_funcionario, id_funcionario, id_funcionario,
id_funcionario)
);

CREATE TABLE EVENTOS (
    id_evento INT PRIMARY KEY UNIQUE,
    nome VARCHAR(50),
    descricao VARCHAR(200),
    local VARCHAR(100),
    data_hora_inicio DATETIME,
    data_hora_fim DATETIME,
    id_funcionario_criador INT,
    data_criacao DATETIME,
    id_funcionario_ultima_alt INT,
    data_atualizacao DATETIME,
    id_funcionario_exclusao INT,
    data_exclusao DATETIME,
    FOREIGN KEY (id_funcionario_criador, id_funcionario_ultima_alt,
id_funcionario_exclusao)
    REFERENCES FUNCIONARIOS (id_funcionario, id_funcionario, id_funcionario)
    ON DELETE RESTRICT
);

CREATE TABLE PAGAMENTOS (
    id_pagamento INT PRIMARY KEY UNIQUE,
    id_funcionario INT,
    id_gestor_criador INT,
    descricao VARCHAR(100),
    valor DECIMAL(10,2),
    data_emissao DATETIME,
    status TINYINT CHECK (status IN (0, 1)),
    id_funcionario_criador INT,
    data_criacao DATETIME,
    id_funcionario_ultima_alt INT,
    data_atualizacao DATETIME,
    id_funcionario_exclusao INT,
    data_exclusao DATETIME,
    FOREIGN KEY (id_funcionario, id_funcionario_criador,
id_funcionario_exclusao, id_funcionario_ultima_alt)
    REFERENCES FUNCIONARIOS (id_funcionario, id_funcionario, id_funcionario,
id_funcionario)
    ON DELETE RESTRICT,
    FOREIGN KEY (id_funcionario, id_funcionario_criador,
id_funcionario_exclusao, id_funcionario_ultima_alt)
    REFERENCES FUNCIONARIOS (id_funcionario, id_funcionario, id_funcionario,
id_funcionario)
    ON DELETE RESTRICT,
    FOREIGN KEY (id_gestor_criador)
    REFERENCES DEPARTAMENTOS (id_funcionario_gestor)

```

```

    ON DELETE RESTRICT
);

CREATE TABLE CONTAS_A_PAGAR (
    id_conta_pagar INT PRIMARY KEY UNIQUE,
    descricao VARCHAR(100),
    valor_total DECIMAL(10,2),
    valor_pago DECIMAL(10,2),
    id_forma_pagamento INT,
    juros DECIMAL(10,2),
    multa DECIMAL(10,2),
    tipo_de_conta VARCHAR(50),
    centro_de_custo VARCHAR(50),
    status_da_conta VARCHAR(20),
    documento BLOB,
    codigo_de_barras VARCHAR(50),
    data_emissao DATETIME,
    data_vencimento DATE,
    data_pagamento DATETIME,
    status TINYINT CHECK (status IN (0, 1)),
    id_funcionario_criacao INT,
    data_criacao DATETIME,
    id_funcionario_ultima_alt INT,
    data_alteracao DATETIME,
    id_funcionario_exclusao INT,
    data_exclusao DATETIME,
    FOREIGN KEY (id_funcionario_exclusao, id_funcionario_ultima_alt,
id_funcionario_criacao)
REFERENCES FUNCIONARIOS (id_funcionario, id_funcionario, id_funcionario)
ON DELETE CASCADE,
    FOREIGN KEY (id_forma_pagamento)
REFERENCES FORMAS_DE_PAGAMENTO (id_forma_pagamento)
ON DELETE RESTRICT,
);

CREATE TABLE CONTAS_A_RECEBER (
    id_conta_receber INT PRIMARY KEY UNIQUE,
    descricao VARCHAR(100),
    documento BLOB,
    id_forma_pagamento INT,
    parcelas_pagas INT,
    parcelas_restantes INT,
    valor_total DECIMAL(10,2),
    valor_recebido DECIMAL(10,2),
    juros DECIMAL(10,2),
    multa DECIMAL(10,2),
    data_primeiro_pagamento DATETIME,
    data_ultimo_pagamento DATETIME,

```



```

        status_da_conta VARCHAR(20),
        status TINYINT CHECK (status IN (0, 1)),
        id_funcionario_criador INT,
        data_criacao DATETIME,
        id_funcionario_ultima_alt INT,
        data_alteracao DATETIME,
        id_funcionario_exclusao INT,
        data_exclusao DATETIME,
        FOREIGN KEY (id_funcionario_criador, id_funcionario_ultima_alt,
id_funcionario_exclusao)
        REFERENCES FUNCIONARIOS (id_funcionario, id_funcionario, id_funcionario)
        ON DELETE CASCADE,
        FOREIGN KEY (id_forma_pagamento)
        REFERENCES FORMAS_DE_PAGAMENTO (id_forma_pagamento)
        ON DELETE RESTRICT
    );

CREATE TABLE FORMAS_DE_PAGAMENTO (
    id_forma_pagamento INT PRIMARY KEY UNIQUE,
    nome VARCHAR(50),
    tipo_financeiro VARCHAR(50),
    tipo_recebimento VARCHAR(50),
    dias_para_receber INT,
    conta_banco VARCHAR(15),
    id_funcionario_criador INT,
    data_criacao DATETIME,
    id_funcionario_ultima_alt INT,
    data_atualizacao DATETIME,
    FOREIGN KEY (id_funcionario_ultima_alt, id_funcionario_criador)
    REFERENCES FUNCIONARIOS (id_funcionario, id_funcionario)
);

CREATE TABLE MENSALIDADES (
    id_mensalidade INT PRIMARY KEY,
    id_conta_receber INT,
    id_responsavel INT,
    id_forma_pagamento INT,
    data_transacao DATETIME,
    data_vencimento DATE,
    status TINYINT CHECK (status IN (0, 1)),
    id_funcionario_criador INT,
    data_criacao DATETIME,
    id_funcionario_ultima_alt INT,
    data_alteracao DATETIME,
    id_funcionario_exclusao INT,
    data_exclusao DATETIME,
    UNIQUE (id_mensalidade, id_conta_receber),
    FOREIGN KEY (id_forma_pagamento)
    REFERENCES FORMAS_DE_PAGAMENTO (id_forma_pagamento)
);

```

```

ON DELETE RESTRICT,
FOREIGN KEY (id_conta_receber)
REFERENCES CONTAS_A_RECEBER (id_conta_receber)
ON DELETE RESTRICT,
FOREIGN KEY (id_funcionario_exclusao, id_funcionario_ultima_alt,
id_funcionario_criador)
REFERENCES FUNCIONARIOS (id_funcionario, id_funcionario, id_funcionario)
ON DELETE RESTRICT,
FOREIGN KEY (id_responsavel)
REFERENCES RESPONSAVEIS (id_responsavel)
);

CREATE TABLE RESPONSAVEIS (
    id_responsavel INT,
    cpf VARCHAR(11),
    primeiro_nome VARCHAR(50),
    nome_do_meio VARCHAR(50),
    ultimo_nome VARCHAR(50),
    data_de_nascimento DATE,
    sexo CHAR(1),
    estado_civil VARCHAR(50),
    naturalidade VARCHAR(50),
    naturalizacao VARCHAR(50),
    rg VARCHAR(20),
    telefone_fixo VARCHAR(15),
    telefone_celular VARCHAR(15),
    email VARCHAR(100),
    endereco VARCHAR(100),
    numero VARCHAR(4),
    complemento VARCHAR(50),
    bairro VARCHAR(50),
    cidade VARCHAR(50),
    cep VARCHAR(10),
    comprovante_de_residencia BLOB,
    conta_bancaria VARCHAR(15),
    valor_mensalidade DECIMAL(10,2),
    id_parceiro INT,
    desconto_mensalidade VARCHAR(50),
    observacao VARCHAR(100),
    status TINYINT CHECK (status IN (0, 1)),
    id_funcionario_criador INT,
    data_criacao DATETIME,
    id_funcionario_ultima_alt INT,
    data_atualizacao DATETIME,
    id_funcionario_exclusao INT,
    data_exclusao DATETIME,
    PRIMARY KEY (id_responsavel, cpf),
    UNIQUE (id_responsavel, cpf),
    FOREIGN KEY (id_parceiro, desconto_mensalidade)

```

```

REFERENCES PARCEIROS (id_parceiro, desconto_mensalidade)
ON DELETE CASCADE,
FOREIGN KEY (id_funcionario_criador, id_funcionario_ultima_alt,
id_funcionario_exclusao)
REFERENCES FUNCIONARIOS (id_funcionario, id_funcionario, id_funcionario)
);

CREATE TABLE PARCEIROS (
    id_parceiro INT UNIQUE,
    desconto_mensalidade VARCHAR(50),
    nome_fantasia VARCHAR(100),
    cnpj VARCHAR(15),
    cpf VARCHAR(11),
    inscricao_estadual CHAR(9),
    endereco VARCHAR(100),
    numero VARCHAR(4),
    complemento VARCHAR(50),
    bairro VARCHAR(50),
    cidade VARCHAR(50),
    cep VARCHAR(10),
    observacao VARCHAR(100),
    status TINYINT CHECK (status IN (0, 1)),
    id_funcionario_criador INT,
    data_criacao DATETIME,
    id_funcionario_ultima_alt INT,
    data_atualizacao DATETIME,
    id_funcionario_exclusao INT,
    data_exclusao DATETIME,
    PRIMARY KEY (id_parceiro, desconto_mensalidade),
    FOREIGN KEY (id_funcionario_criador, id_funcionario_ultima_alt,
id_funcionario_exclusao)
REFERENCES FUNCIONARIOS (id_funcionario, id_funcionario, id_funcionario)
);

CREATE TABLE SECRETARIOS (
    id_secretario INT,
    id_funcionario INT,
    primeiro_nome VARCHAR(50),
    ultimo_nome VARCHAR(50),
    status TINYINT CHECK (status IN (0, 1)),
    PRIMARY KEY (id_secretario, id_funcionario),
    UNIQUE (id_secretario, id_funcionario),
    FOREIGN KEY (id_funcionario, status)
REFERENCES FUNCIONARIOS (id_funcionario, status)
ON DELETE CASCADE
);

CREATE TABLE ALUNOS (
    matricula VARCHAR(20),

```

```

    cpf VARCHAR(11),
    primeiro_nome VARCHAR(50),
    nome_do_meio VARCHAR(50),
    ultimo_nome VARCHAR(50),
    id_turma INT,
    id_curso INT,
    data_de_nascimento DATE,
    sexo CHAR(1),
    naturalidade VARCHAR(50),
    naturalizacao VARCHAR(50),
    rg VARCHAR(20),
    endereco VARCHAR(100),
    numero VARCHAR(4),
    complemento VARCHAR(50),
    bairro VARCHAR(50),
    cidade VARCHAR(50),
    cep VARCHAR(10),
    comprovante_de_residencia BLOB,
    declaracao_matricula BLOB,
    id_motorista_van INT,
    certidao_de_nascimento DECIMAL(10,2),
    declaracao_de_escolaridade DATETIME,
    declaracao_de_adimplencia DATETIME,
    ficha_medica DATETIME,
    carteira_de_vacina DATETIME,
    observacao VARCHAR(100),
    status TINYINT CHECK (status IN (0, 1)),
    id_secretario_criador INT,
    data_criacao DATETIME,
    id_secretario_ultima_alt INT,
    data_atualizacao DATETIME,
    id_secretario_exclusao INT,
    data_exclusao DATETIME,
    PRIMARY KEY (matricula, cpf),
    UNIQUE (matricula, cpf),
    FOREIGN KEY (id_secretario_criador, id_secretario_criador,
id_secretario_exclusao, id_secretario_ultima_alt)
    REFERENCES SECRETARIOS (id_secretario, id_secretario, id_secretario,
id_secretario)
    ON DELETE RESTRICT,
    FOREIGN KEY (id_motorista_van)
    REFERENCES MOTORISTAS (id_motorista)
    ON DELETE RESTRICT,
    FOREIGN KEY (id_turma)
    REFERENCES TURMAS (id_turma)
    ON DELETE RESTRICT,
    FOREIGN KEY (id_curso)
    REFERENCES CURSOS (id_curso)
    ON DELETE RESTRICT

```

```

);

CREATE TABLE GRAU_PARENTESCO (
    id_grau_parentesco INT PRIMARY KEY UNIQUE,
    id_responsavel INT,
    matricula_aluno VARCHAR(20),
    cpf_aluno VARCHAR(11),
    grau_de_parentesco VARCHAR(50),
    FOREIGN KEY (id_responsavel)
    REFERENCES RESPONSAVEIS (id_responsavel),
    FOREIGN KEY (matricula_aluno, cpf_aluno)
    REFERENCES ALUNOS (matricula, cpf)
);

CREATE TABLE MOTORISTAS (
    id_motorista INT,
    cpf VARCHAR(11),
    placa_van varchar(7),
    primeiro_nome VARCHAR(50),
    nome_do_meio VARCHAR(50),
    ultimo_nome VARCHAR(50),
    rg VARCHAR(20),
    cnh VARCHAR(9),
    telefone_fixo VARCHAR(15),
    telefone_celular VARCHAR(15),
    email VARCHAR(100),
    conta_bancaria VARCHAR(15),
    id_funcionario_criador INT,
    data_criacao DATETIME,
    id_funcionario_ultima_alt INT,
    data_alteracao DATETIME,
    id_funcionario_exclusao INT,
    data_exclusao DATETIME,
    PRIMARY KEY (id_motorista, cpf),
    UNIQUE (id_motorista, cpf, cnh, conta_bancaria),
    FOREIGN KEY (placa_van)
    REFERENCES VANS_ESCOLARES (placa)
    ON DELETE RESTRICT,
    FOREIGN KEY (id_funcionario_criador, id_funcionario_ultima_alt,
id_funcionario_exclusao)
    REFERENCES FUNCIONARIOS (id_funcionario, id_funcionario, id_funcionario)
);

CREATE TABLE VANS_ESCOLARES (
    placa varchar(7) PRIMARY KEY,
    renavam BLOB,
    modelo VARCHAR(50),
    capacidade INT,
    valor_aluguel DECIMAL(10,2),

```

```

        UNIQUE (renavam, placa)
    );

CREATE TABLE SALAS (
    id_sala INT PRIMARY KEY UNIQUE,
    numero INT,
    descricao VARCHAR(100),
    tipo VARCHAR(50),
    corredor INT,
    andar INT,
    observacao VARCHAR(100),
    data_criacao DATETIME,
    data_atualizacao DATETIME
);

CREATE TABLE TURMAS (
    id_turma INT PRIMARY KEY UNIQUE,
    id_serie INT,
    nome VARCHAR(50),
    ano YEAR,
    descricao VARCHAR(100),
    id_sala INT,
    status TINYINT CHECK (status IN (0, 1)),
    id_funcionario_criador INT,
    data_criacao DATETIME,
    id_funcionario_ultima_alt INT,
    data_atualizacao DATETIME,
    id_funcionario_exclusao INT,
    data_exclusao DATETIME,
    FOREIGN KEY (id_sala)
    REFERENCES SALAS (id_sala)
    ON DELETE RESTRICT,
    FOREIGN KEY (id_serie)
    REFERENCES SERIES (id_serie)
    ON DELETE RESTRICT,
    FOREIGN KEY (id_funcionario_criador, id_funcionario_ultima_alt,
id_funcionario_exclusao)
    REFERENCES FUNCIONARIOS (id_funcionario, id_funcionario, id_funcionario)
);

CREATE TABLE SERIES (
    id_serie INT PRIMARY KEY UNIQUE,
    id_modalidade INT,
    nome VARCHAR(50),
    descricao VARCHAR(100),
    grade_curricular BLOB,
    status TINYINT CHECK (status IN (0, 1)),
    data_criacao DATETIME,
    data_atualizacao DATETIME,

```

```

        data_exclusao DATETIME,
        FOREIGN KEY (id_modalidade)
        REFERENCES MODALIDADES (id_modalidade)
        ON DELETE RESTRICT
    );

CREATE TABLE MODALIDADES (
    id_modalidade INT PRIMARY KEY UNIQUE,
    nome VARCHAR(50),
    descricao VARCHAR(100),
    matriz_curricular BLOB,
    projeto_politico_pedagogico BLOB,
    status TINYINT CHECK (status IN (0, 1)),
    data_criacao DATETIME,
    data_atualizacao DATETIME,
    data_exclusao DATETIME
);

CREATE TABLE CURSOS (
    id_curso INT PRIMARY KEY UNIQUE,
    nome VARCHAR(50),
    id_modalidade INT,
    descricao VARCHAR(100),
    data_inicial_periodo DATE,
    data_final_periodo DATE,
    observacao VARCHAR(100),
    status TINYINT CHECK (status IN (0, 1)),
    id_funcionario_criador INT,
    data_criacao DATETIME,
    id_funcionario_ultima_alt INT,
    data_atualizacao DATETIME,
    id_funcionario_exclusao INT,
    data_exclusao DATETIME,
    FOREIGN KEY (id_modalidade)
    REFERENCES MODALIDADES (id_modalidade)
    ON DELETE RESTRICT,
    FOREIGN KEY (id_funcionario_criador, id_funcionario_ultima_alt,
id_funcionario_exclusao)
    REFERENCES FUNCIONARIOS (id_funcionario, id_funcionario, id_funcionario)
);

CREATE TABLE LISTAS_DE_PRESENCA (
    id_lista INT,
    id_aula INT,
    id_disciplina INT,
    id_turma INT,
    matricula_aluno VARCHAR(20),
    cpf_aluno VARCHAR(11),
    nome_completo_aluno VARCHAR(50),

```

```

    assiduidade VARCHAR(100),
    observacao VARCHAR(100),
    id_professor_responsavel INT,
    data_criacao DATETIME,
    data_alteracao DATETIME,
    PRIMARY KEY (id_aula, id_lista),
    UNIQUE (id_lista, id_aula),
    FOREIGN KEY (id_turma)
    REFERENCES TURMAS (id_turma),
    FOREIGN KEY (id_disciplina)
    REFERENCES DISCIPLINAS (id_disciplina),
    FOREIGN KEY (matricula_aluno, cpf_aluno)
    REFERENCES ALUNOS (matricula, cpf),
    FOREIGN KEY (id_aula)
    REFERENCES AULAS (id_aula),
    FOREIGN KEY (id_professor_responsavel)
    REFERENCES PROFESSORES (id_professor)
);

CREATE TABLE NOTAS_MEDIAS (
    id_media INT PRIMARY KEY UNIQUE,
    id_boletim INT,
    matricula_aluno VARCHAR(20),
    cpf_aluno VARCHAR(11),
    id_disciplina INT,
    nota_avaliacao_1 DECIMAL(4,2),
    nota_trabalho DECIMAL(4,2),
    nota_avaliacao_2 DECIMAL(4,2),
    nota_assiduidade DECIMAL(4,2),
    observacao VARCHAR(100),
    id_professor_criador INT,
    data_criacao DATETIME,
    id_professor_ultima_alt INT,
    data_atualizacao DATETIME,
    FOREIGN KEY (matricula_aluno, cpf_aluno)
    REFERENCES ALUNOS (matricula, cpf)
    ON DELETE RESTRICT,
    FOREIGN KEY (id_disciplina)
    REFERENCES DISCIPLINAS (id_disciplina)
    ON DELETE RESTRICT,
    FOREIGN KEY (id_boletim)
    REFERENCES BOLETINS (id_boletim),
    FOREIGN KEY (id_professor_ultima_alt, id_professor_criador)
    REFERENCES PROFESSORES (id_professor, id_professor)
);

CREATE TABLE PROFESSORES (
    id_funcionario INT,
    id_professor INT,

```



```

    primeiro_nome VARCHAR(50),
    ultimo_nome VARCHAR(50),
    status TINYINT CHECK (status IN (0, 1)),
    PRIMARY KEY (id_professor, id_funcionario),
    UNIQUE (id_funcionario, id_professor),
    FOREIGN KEY (id_funcionario, status)
    REFERENCES FUNCIONARIOS (id_funcionario, status)
    ON DELETE CASCADE
);

CREATE TABLE DISCIPLINAS (
    id_disciplina INT PRIMARY KEY UNIQUE,
    id_professor_responsavel INT,
    nome VARCHAR(50),
    id_dia_hora_ministrada INT,
    conteudo_abordado VARCHAR(500),
    observacao VARCHAR(100),
    status TINYINT CHECK (status IN (0, 1)),
    id_funcionario_criador INT,
    data_criacao DATETIME,
    id_funcionario_ultima_alt INT,
    data_atualizacao DATETIME,
    id_funcionario_exclusao INT,
    data_exclusao DATETIME,
    FOREIGN KEY (id_dia_hora_ministrada)
    REFERENCES DIAS_MINISTRADOS (id_dia_hora)
    ON DELETE NO ACTION,
    FOREIGN KEY (id_professor_responsavel)
    REFERENCES PROFESSORES (id_professor)
    ON DELETE RESTRICT,
    FOREIGN KEY (id_funcionario_criador, id_funcionario_ultima_alt,
id_funcionario_exclusao)
    REFERENCES FUNCIONARIOS (id_funcionario, id_funcionario, id_funcionario)
);

CREATE TABLE DIAS_MINISTRADOS (
    id_dia_hora INT NOT NULL PRIMARY KEY UNIQUE,
    id_disciplina INT,
    id_turma INT,
    data_hora_aula DATETIME,
    FOREIGN KEY (id_disciplina)
    REFERENCES DISCIPLINAS (id_disciplina),
    FOREIGN KEY (id_turma)
    REFERENCES TURMAS (id_turma)
);

CREATE TABLE BOLETINS (
    id_boletim INT PRIMARY KEY UNIQUE,
    matricula_aluno VARCHAR(20),

```

```

    cpf_aluno VARCHAR(11),
    semestre INT,
    data_criacao DATETIME,
    data_atualizacao DATETIME,
    FOREIGN KEY (matricula_aluno, cpf_aluno)
    REFERENCES ALUNOS (matricula, cpf)
);

CREATE TABLE AULAS (
    id_aula INT PRIMARY KEY UNIQUE,
    id_dia_hora INT,
    id_disciplina INT,
    id_turma INT,
    FOREIGN KEY (matricula_aluno, cpf_aluno)
    REFERENCES ALUNOS (matricula, cpf),
    FOREIGN KEY (id_dia_hora)
    REFERENCES DIAS_MINISTRADOS (id_dia_hora),
    FOREIGN KEY (id_disciplina)
    REFERENCES DISCIPLINAS (id_disciplina),
    FOREIGN KEY (id_turma)
    REFERENCES TURMAS (id_turma)
);

```

**Algoritmo 1 – Modelo Físico**

### 3.7 CONSULTAS EM SQL

Para realizar as consultas de exemplo, foi necessária a inserção de diversos dados aleatórios no banco de dados criado a partir do código. Para a inserção dos dados, utilizei a ferramenta de inteligência artificial Gemini, da Google. Além disso, para a realização das seguintes consultas, utilizei um banco de dados Maria DB, por meio do software HeidiSQL.

1) Listar todas as disciplinas ministradas em uma determinada turma:

```

SELECT DISTINCT D.nome
FROM DISCIPLINAS D
INNER JOIN DIAS_MINISTRADOS DM ON D.id_disciplina = DM.id_disciplina
INNER JOIN TURMAS T ON DM.id_turma = T.id_turma
WHERE T.id_turma = 1;

```

2) Listar todos os alunos matriculados em uma disciplina específica:

```

SELECT *

```

```

FROM ALUNOS A
INNER JOIN BOLETINS B ON A.matricula = B.matricula_aluno AND A.cpf =
B.cpf_aluno
INNER JOIN NOTAS_MEDIAS NM ON B.id_boletim = NM.id_boletim
INNER JOIN DISCIPLINAS D ON NM.id_disciplina = D.id_disciplina
WHERE D.nome = 'Matemática';

```

3) Listar as notas dos alunos em uma disciplina específica:

```

SELECT A.primeiro_nome, A.ultimo_nome, NM.nota_avaliacao_1, NM.nota_trabalho,
NM.nota_avaliacao_2, NM.nota_assiduidade
FROM ALUNOS A
INNER JOIN BOLETINS B ON A.matricula = B.matricula_aluno AND A.cpf =
B.cpf_aluno
INNER JOIN NOTAS_MEDIAS NM ON B.id_boletim = NM.id_boletim
INNER JOIN DISCIPLINAS D ON NM.id_disciplina = D.id_disciplina
WHERE D.nome = 'Ciências Naturais';

```

4) Listar os alunos que estão com notas abaixo da média em uma disciplina específica:

```

SELECT A.primeiro_nome, A.ultimo_nome
FROM ALUNOS A
INNER JOIN BOLETINS B ON A.matricula = B.matricula_aluno AND A.cpf =
B.cpf_aluno
INNER JOIN NOTAS_MEDIAS NM ON B.id_boletim = NM.id_boletim
INNER JOIN DISCIPLINAS D ON NM.id_disciplina = D.id_disciplina
WHERE D.nome = 'Geografia'
AND (NM.nota_avaliacao_1 + NM.nota_trabalho + NM.nota_avaliacao_2 +
NM.nota_assiduidade) / 4 < <nota_media_esperada>;

```

5) Listar todas as notas dos alunos em todas as disciplinas:

```

SELECT A.primeiro_nome, A.ultimo_nome, D.nome AS disciplina,
NM.nota_avaliacao_1, NM.nota_trabalho, NM.nota_avaliacao_2,
NM.nota_assiduidade
FROM ALUNOS A
INNER JOIN BOLETINS B ON A.matricula = B.matricula_aluno AND A.cpf =
B.cpf_aluno
INNER JOIN NOTAS_MEDIAS NM ON B.id_boletim = NM.id_boletim
INNER JOIN DISCIPLINAS D ON NM.id_disciplina = D.id_disciplina;

```

6) Listar todas as disciplinas ministradas por um determinado professor:

```

SELECT D.nome AS disciplina, P.primeiro_nome, P.ultimo_nome

```

```
FROM DISCIPLINAS D
INNER JOIN PROFESSORES P ON D.id_professor_responsavel = P.id_professor;
```

7) Listar todos os alunos que faltaram em uma determinada aula:

```
SELECT A.primeiro_nome, A.ultimo_nome
FROM ALUNOS A
INNER JOIN LISTAS_DE_PRESENCA LP ON A.matricula = LP.matricula_aluno AND A.cpf = LP.cpf_aluno
INNER JOIN AULAS AU ON LP.id_aula = AU.id_aula
WHERE AU.id_aula = <id_da_aula> AND LP.assiduidade = 'Falta';
```

8) Listar todos os alunos que obtiveram nota máxima em uma disciplina específica:

```
SELECT A.primeiro_nome, A.ultimo_nome
FROM ALUNOS A
INNER JOIN BOLETINS B ON A.matricula = B.matricula_aluno AND A.cpf = B.cpf_aluno
INNER JOIN NOTAS_MEDIAS NM ON B.id_boletim = NM.id_boletim
INNER JOIN DISCIPLINAS D ON NM.id_disciplina = D.id_disciplina
WHERE D.nome = 'Educação Física'
AND (NM.nota_avaliacao_1 + NM.nota_trabalho + NM.nota_avaliacao_2 + NM.nota_assiduidade) / 4 = 10;
```

9) Listar todas as turmas e seus respectivos números de alunos matriculados:

```
SELECT T.nome, COUNT(A.matricula) AS numero_alunos
FROM TURMAS T
LEFT JOIN ALUNOS A ON T.id_turma = A.id_turma
GROUP BY T.nome;
```

10) Alunos com Melhor e Pior Desempenho em Cada Disciplina:

```
SELECT D.nome AS disciplina,
       CONCAT((SELECT A.primeiro_nome FROM ALUNOS A WHERE A.matricula = B.matricula_aluno AND A.cpf = B.cpf_aluno), ' ', (SELECT A.ultimo_nome FROM ALUNOS A WHERE A.matricula = B.matricula_aluno AND A.cpf = B.cpf_aluno)) AS melhor_aluno,
       MAX((NM.nota_avaliacao_1 + NM.nota_trabalho + NM.nota_avaliacao_2 + NM.nota_assiduidade) / 4) AS melhor_desempenho,
       CONCAT((SELECT A.primeiro_nome FROM ALUNOS A WHERE A.matricula = B.matricula_aluno AND A.cpf = B.cpf_aluno), ' ', (SELECT A.ultimo_nome FROM ALUNOS A WHERE A.matricula = B.matricula_aluno AND A.cpf = B.cpf_aluno)) AS pior_aluno,
```

```

    MIN((NM.nota_avaliacao_1 + NM.nota_trabalho + NM.nota_avaliacao_2 +
    NM.nota_assiduidade) / 4) AS pior_desempenho
FROM BOLETINS B
INNER JOIN NOTAS_MEDIAS NM ON B.id_boletim = NM.id_boletim
INNER JOIN DISCIPLINAS D ON NM.id_disciplina = D.id_disciplina
GROUP BY D.nome;

```

11) Percentual de Alunos por Gênero:

```

SELECT A.sexo, COUNT(*) AS quantidade_alunos,
       (COUNT(*) / (SELECT COUNT(*) FROM ALUNOS)) * 100 AS percentual
FROM ALUNOS A
GROUP BY A.sexo;

```

12) Média de Idade dos Alunos por Turma:

```

SELECT T.nome AS turma,
       AVG(TIMESTAMPDIFF(YEAR, A.data_de_nascimento, CURDATE())) AS media_idade
FROM ALUNOS A
INNER JOIN TURMAS T ON A.id_turma = T.id_turma
GROUP BY T.nome;

```

13) Quantidade de Alunos por Responsável:

```

SELECT R.primeiro_nome, R.ultimo_nome, COUNT(*) AS quantidade_alunos
FROM ALUNOS A
INNER JOIN RESPONSAVEIS R ON A.id_responsavel = R.id_responsavel
GROUP BY R.primeiro_nome, R.ultimo_nome;

```

14) Média de Notas dos Alunos por Idade:

```

SELECT CASE
    WHEN TIMESTAMPDIFF(YEAR, A.data_de_nascimento, CURDATE()) BETWEEN 5 AND 10
    THEN '5-10'
    WHEN TIMESTAMPDIFF(YEAR, A.data_de_nascimento, CURDATE()) BETWEEN 11 AND
    15 THEN '11-15'
    WHEN TIMESTAMPDIFF(YEAR, A.data_de_nascimento, CURDATE()) BETWEEN 16 AND
    20 THEN '16-20'
    ELSE 'Acima de 20'
END AS faixa_etaria,
    AVG((NM.nota_avaliacao_1 + NM.nota_trabalho + NM.nota_avaliacao_2 +
    NM.nota_assiduidade) / 4) AS media_notas
FROM ALUNOS A
INNER JOIN BOLETINS B ON A.matricula = B.matricula_aluno AND A.cpf =
    B.cpf_aluno

```

```
INNER JOIN NOTAS_MEDIAS NM ON B.id_boletim = NM.id_boletim
GROUP BY faixa_etaria;
```

15) Quantidade de Disciplinas por Curso:

```
SELECT C.nome AS curso, COUNT(DISTINCT D.id_disciplina) AS num_disciplinas
FROM CURSOS C
INNER JOIN DISCIPLINAS D ON C.id_curso = D.id_curso
GROUP BY C.nome;
```

16) Alunos com Maior Número de Faltas em Cada Disciplina:

```
SELECT D.nome AS disciplina, CONCAT(A.primeiro_nome, ' ', A.ultimo_nome) AS
aluno,
    COUNT(*) AS faltas
FROM ALUNOS A
INNER JOIN BOLETINS B ON A.matricula = B.matricula_aluno AND A.cpf =
B.cpf_aluno
INNER JOIN DISCIPLINAS D ON B.id_disciplina = D.id_disciplina
INNER JOIN LISTAS_DE_PRESENCA LP ON B.id_turma = LP.id_turma AND A.matricula =
LP.matricula_aluno AND A.cpf = LP.cpf_aluno
WHERE LP.assiduidade = 'Falta'
GROUP BY D.nome, aluno
ORDER BY faltas DESC;
```

17) Alunos com Menor Média de Notas por Turma:

```
SELECT T.nome AS turma, CONCAT(A.primeiro_nome, ' ', A.ultimo_nome) AS aluno,
    AVG((NM.nota_avaliacao_1 + NM.nota_trabalho + NM.nota_avaliacao_2 +
NM.nota_assiduidade) / 4) AS media_notas
FROM ALUNOS A
INNER JOIN TURMAS T ON A.id_turma = T.id_turma
INNER JOIN BOLETINS B ON A.matricula = B.matricula_aluno AND A.cpf =
B.cpf_aluno
INNER JOIN NOTAS_MEDIAS NM ON B.id_boletim = NM.id_boletim
GROUP BY T.nome, aluno
ORDER BY media_notas ASC;
```

18) Desempenho dos Alunos em Cada Disciplina por Ano Letivo:

```
SELECT YEAR(B.data_inicio) AS ano_letivo, D.nome AS disciplina,
    CONCAT(A.primeiro_nome, ' ', A.ultimo_nome) AS aluno,
    (NM.nota_avaliacao_1 + NM.nota_trabalho + NM.nota_avaliacao_2 +
NM.nota_assiduidade) / 4 AS media_notas
FROM ALUNOS A
```

```
INNER JOIN BOLETINS B ON A.matricula = B.matricula_aluno AND A.cpf =
B.cpf_aluno
INNER JOIN NOTAS_MEDIAS NM ON B.id_boletim = NM.id_boletim
INNER JOIN DISCIPLINAS D ON NM.id_disciplina = D.id_disciplina;
```

19) Alunos que Obtiveram Aprovação Automática em Todas as Disciplinas:

```
SELECT CONCAT(A.primeiro_nome, ' ', A.ultimo_nome) AS aluno
FROM ALUNOS A
INNER JOIN BOLETINS B ON A.matricula = B.matricula_aluno AND A.cpf =
B.cpf_aluno
INNER JOIN NOTAS_MEDIAS NM ON B.id_boletim = NM.id_boletim
GROUP BY aluno
HAVING MIN((NM.nota_avalicao_1 + NM.nota_trabalho + NM.nota_avalicao_2 +
NM.nota_assiduidade) / 4) >= 7.0;
```

20) Quantidade de Alunos por Turma e Sexo:

```
SELECT T.nome AS turma, A.sexo, COUNT(*) AS quantidade_alunos
FROM ALUNOS A
INNER JOIN TURMAS T ON A.id_turma = T.id_turma
GROUP BY T.nome, A.sexo;
```

21) Contagem de Funcionários por Departamento:

```
SELECT D.nome AS departamento, COUNT(*) AS total_funcionarios
FROM FUNCIONARIOS F
INNER JOIN DEPARTAMENTOS D ON F.id_departamento = D.id_departamento
GROUP BY D.nome;
```

22) Média Salarial por Estado Civil dos Funcionários:

```
SELECT estado_civil, AVG(salario_bruto) AS media_salarial
FROM FUNCIONARIOS
GROUP BY estado_civil;
```

23) Número de Funcionários por Sexo:

```
SELECT sexo, COUNT(*) AS total_funcionarios
FROM FUNCIONARIOS
GROUP BY sexo;
```

24) Total de Funcionários por Ano de Nascimento:

```
SELECT YEAR(data_de_nascimento) AS ano_nascimento, COUNT(*) AS
total_funcionarios
FROM FUNCIONARIOS
GROUP BY ano_nascimento;
```

25) Departamento com Maior Número de Funcionários:

```
SELECT D.nome AS departamento, COUNT(*) AS total_funcionarios
FROM FUNCIONARIOS F
INNER JOIN DEPARTAMENTOS D ON F.id_departamento = D.id_departamento
GROUP BY D.nome
ORDER BY total_funcionarios DESC
LIMIT 1;
```

26) Média Salarial por Tipo de Departamento:

```
SELECT D.descricao AS tipo_departamento, AVG(F.salario_bruto) AS
media_salarial
FROM FUNCIONARIOS F
INNER JOIN DEPARTAMENTOS D ON F.id_departamento = D.id_departamento
GROUP BY D.descricao;
```

27) Funcionários por Faixa Etária:

```
SELECT CASE
    WHEN TIMESTAMPDIFF(YEAR, data_de_nascimento, CURDATE()) BETWEEN 20 AND 30
    THEN '20-30'
    WHEN TIMESTAMPDIFF(YEAR, data_de_nascimento, CURDATE()) BETWEEN 31 AND 40
    THEN '31-40'
    WHEN TIMESTAMPDIFF(YEAR, data_de_nascimento, CURDATE()) BETWEEN 41 AND 50
    THEN '41-50'
    ELSE 'Acima de 50'
END AS faixa_etaria,
COUNT(*) AS total_funcionarios
FROM FUNCIONARIOS
GROUP BY faixa_etaria;
```

28) Funcionários com Salário Bruto Acima da Média:

```
SELECT *
FROM FUNCIONARIOS
WHERE salario_bruto > (SELECT AVG(salario_bruto) FROM FUNCIONARIOS);
```

29) Departamento com Maior Média Salarial:



```
SELECT D.nome AS departamento, AVG(F.salario_bruto) AS media_salarial
FROM FUNCIONARIOS F
INNER JOIN DEPARTAMENTOS D ON F.id_departamento = D.id_departamento
GROUP BY D.nome
ORDER BY media_salarial DESC
LIMIT 1;
```

30) Funcionários que São Gestores de Departamento:

```
SELECT F.primeiro_nome, F.ultimo_nome, D.nome AS departamento
FROM FUNCIONARIOS F
INNER JOIN DEPARTAMENTOS D ON F.id_funcionario = D.id_funcionario_gestor;
```

#### Algoritmo 2 – Consultas

## 4 RESULTADOS OBTIDOS

O desenvolvimento do modelo de banco de dados proporcionou uma oportunidade significativa de aprendizado em diversos aspectos relacionados à modelagem de dados e sistemas de informação. Primeiramente, a análise detalhada dos requisitos do sistema foi fundamental para o entendimento completo das entidades envolvidas, seus atributos e os relacionamentos entre elas. Esse processo destacou a importância da fase de levantamento de requisitos no desenvolvimento de sistemas de informação robustos e eficientes.

Ademais, a aplicação de conceitos de modelagem de dados, como normalização, permitiu evitar a redundância de dados e garantir a integridade e consistência dos dados armazenados. A compreensão e aplicação correta das restrições de integridade referencial foram cruciais para manter a consistência dos dados ao longo do tempo, evitando inconsistências e anomalias.

Por fim, a modelagem dos relacionamentos entre as entidades exigiu uma análise cuidadosa das associações entre elas, utilizando chaves primárias e estrangeiras para representar de forma precisa a estrutura do domínio do problema. Esse processo ressaltou a importância da representação clara e concisa dos relacionamentos para facilitar a compreensão e manutenção do modelo de banco de dados.

O diagrama de banco de dados foi desenvolvido com foco na concisão,

clareza e eficiência. Isso foi alcançado por meio da identificação cuidadosa das entidades principais e seus atributos essenciais, evitando a inclusão de informações redundantes ou irrelevantes que pudessem complicar o modelo.

Além disso, a aplicação de técnicas de normalização permitiu minimizar a redundância de dados, garantindo a consistência e integridade do banco de dados. A definição clara dos relacionamentos entre as entidades, por meio de chaves estrangeiras, contribuiu para uma representação precisa da estrutura do sistema e facilitou a compreensão do modelo.

Embora o diagrama de banco de dados atenda aos requisitos atuais do sistema, algumas melhorias podem ser consideradas para aprimorar ainda mais o modelo e prepará-lo para implementações futuras:

- **Otimização de Desempenho:** Avaliar e otimizar o desempenho do banco de dados, especialmente para consultas frequentes e operações de escrita, pode ser uma área de melhoria importante.
- **Segurança de Dados:** Reforçar a segurança dos dados por meio de medidas de autenticação, autorização e criptografia é crucial para proteger informações sensíveis contra acessos não autorizados.
- **Escalabilidade:** Projetar o banco de dados visando à escalabilidade futura é essencial para lidar com um aumento no volume de dados e na carga de trabalho do sistema.
- **Backup e Recuperação:** Desenvolver e implementar estratégias abrangentes de backup e recuperação é fundamental para garantir a disponibilidade e integridade dos dados em caso de falhas.
- **Documentação Adequada:** Manter uma documentação detalhada do modelo de banco de dados é importante para facilitar a compreensão e manutenção do sistema ao longo do tempo.

Em conclusão, considerar essas melhorias pode contribuir significativamente para o aprimoramento do modelo de banco de dados, tornando-o mais robusto, seguro e escalável para suportar as necessidades do sistema no futuro.

## 5 CONCLUSÃO

O processo de elaboração do diagrama de entidade-relacionamento, e a criação de um modelo físico de banco de dados concebido a partir das informações compiladas junto aos gestores do Colégio 24 de Maio mediante questionários e entrevistas, representa um marco significativo rumo à modernização e otimização dos processos internos desta instituição educacional.

Este modelo oferece uma representação visual concisa e abrangente das interconexões entre os dados e as entidades da escola, estabelecendo uma base sólida para a implementação de um programa de gestão automatizado.

Sua utilidade se destaca na transição das atividades manuais, como a gestão financeira realizada por meio de planilhas, para um sistema integrado e altamente eficiente. Porém, ainda há pontos a serem explorados nele, para que se torne o mais completo possível de acordo com o regimento de cada um dos setores da instituição.

Ao implementar este modelo em um programa operacional, a instituição poderá agilizar suas operações cotidianas, reduzindo equívocos e aumentando a eficiência na gestão administrativa. Ademais, este avanço tecnológico permitirá que os gestores e colaboradores concentrem seus esforços em atividades mais estratégicas, fomentando o aprimoramento contínuo do Colégio 24 de Maio.

Essa abordagem não apenas simplificará tarefas operacionais, mas também criará um ambiente propício para o aprimoramento educacional dos alunos, ao possibilitar que a equipe dedique mais tempo e recursos para a melhoria da qualidade do ensino proporcionado.

## REFERÊNCIAS

CLASSAPP. **Gestão escolar: o que é, estratégias e tudo sobre.** Disponível em:<<https://www.classapp.com.br/artigos/gestao-escolar>>. Acesso em: 29 out 2023.

COLARES, Maria Lília Imbiriba Sousa; PACÍFICO, Juracy Machado; ESTRELA, George Queiroga. **GESTÃO ESCOLAR: ENFRENTANDO OS DESAFIOS COTIDIANOS EM ESCOLAS PÚBLICAS.** Disponível em:<[http://portal.mec.gov.br/index.php?option=com\\_docman&view=download&alias=2170-livro-unir-2009&Itemid=30192](http://portal.mec.gov.br/index.php?option=com_docman&view=download&alias=2170-livro-unir-2009&Itemid=30192)>. Acesso em: 29 out 2023.

ELMASRI, Ramez; NAVATHE, Shamkant B. **Sistemas de Bancos de Dados.** nº ed: 6. São Paulo : Pearson Education, 2011.

LIMA, Shirley Vieira de; ARAÚJO, Maria Onete Brandão; CARVALHO, Emani José Guimarães de; MENEZES, Fabíola Fernandes de. **GESTÃO ESCOLAR: DESAFIOS E POSSIBILIDADES.** nº ed. 1. São Paulo : Editora Científica, 2021.

NERY Rodrigues Machado, Felipe. **Banco de Dados Projeto e Implementação.** nº ed. 3. São Paulo : Érica, 2014.

ROB, Peter; CORONEL, Carlos. **Sistemas de Banco de Dados: Projeto, Implementação e Gerenciamento.** nº ed: 8. São Paulo : Cengage Learning, 2011.

SOMMERVILLE, Ian. **Engenharia de Software.** nº ed: 9. São Paulo : Pearson Education, 2011.

VERSIANI, Rafael. **Sistema escolar: descubra o que é e quais são os melhores do mercado.** Disponível em:<<https://enotas.com.br/blog/sistema-escolar/>>. Acesso em: 28 out 2023.