

**Instituto Federal de Educação da
Paraíba - Campus Campina Grande**
Engenharia de Computação

David Medeiros Santos

Projeto de Banco de Dados:
Desenvolvimento de um Sistema para
Aplicativos de Transporte

Campina Grande
07/03/2025

SUMÁRIO

1 INTRODUÇÃO	2
2 MODELO CONCEITUAL	4
3 MODELO LÓGICO	5
3.1 MODELO RELACIONAL	5
3.2 DICIONÁRIO DE DADOS	6
4 PROJETO FÍSICO	7
4.1 SCRIPTS DE ESTRUTURAÇÃO	9
4.2 POVOAMENTO COM DADOS INICIAIS	9
4.3 SCRIPTS DE ATUALIZAÇÕES	10
4.4 SCRIPTS DE CONSULTAS	10
4.5 SIMULANDO CORRIDA	10
4 CONSIDERAÇÕES FINAIS	11

1 INTRODUÇÃO

O mini-mundo escolhido para este projeto é um sistema de gerenciamento de corridas de motoristas e passageiros de um aplicativo de transporte. Para isso, o sistema visa organizar, gerenciar e monitorar a interação entre motoristas, passageiros, corridas, veículos, localizações e avaliações, promovendo uma gestão eficiente das informações e facilitando a comunicação entre as partes envolvidas.

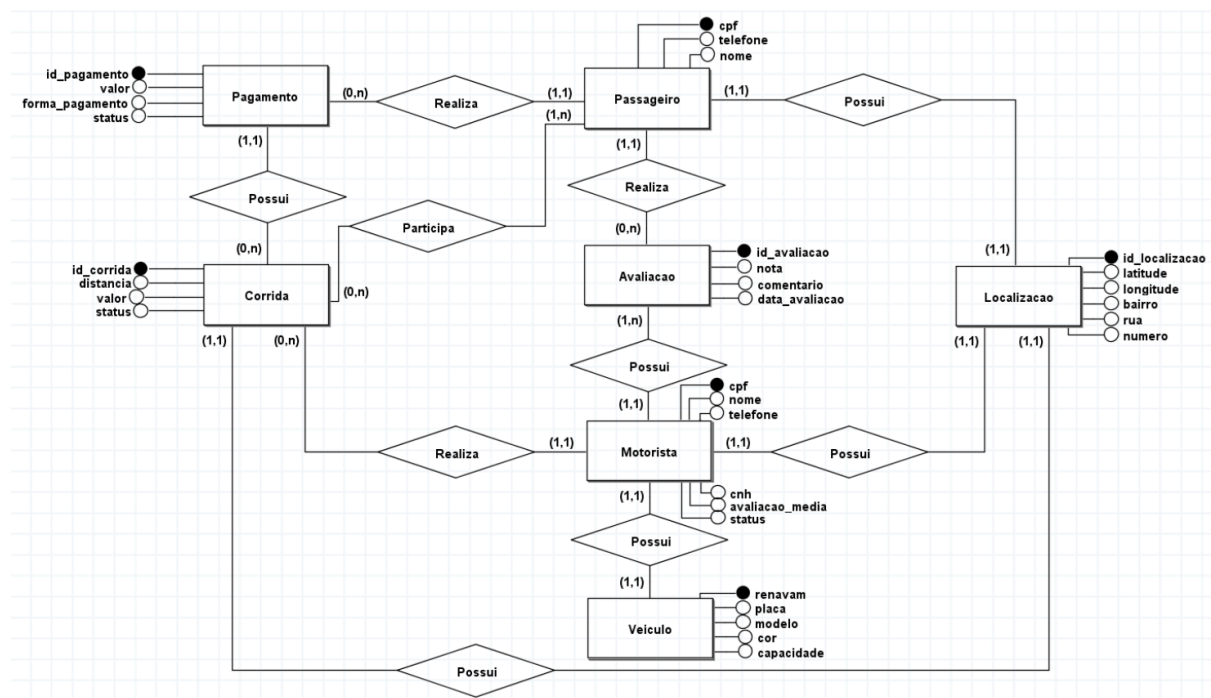
Motoristas possuem CPF como chave primária, avaliação entre 0 e 5 e telefone, podendo ter um veículo vinculado, que contém placa, modelo e é identificado pelo Renavam. Passageiros possuem CPF, telefone e e-mail. As corridas registram motorista, passageiro, status e valor, refletindo com precisão as informações de cada viagem. Avaliações, que devem conter notas entre 1 e 5, data, comentário, são exclusivas para passageiros. O banco de dados calcula automaticamente a média de avaliação de cada motorista com base nas avaliações recebidas. As localizações dos motoristas, passageiros e destinos são armazenadas em coordenadas de latitude e longitude, além de informações como bairro, rua e número. O pagamento da corrida, com valor, método de pagamento e status, é liberado ao final da viagem, e a corrida é então finalizada.

A estrutura do sistema foi projetada para garantir a eficiência e a integridade dos dados. Para isso, a vinculação de motoristas a veículos, o rastreamento por coordenadas geográficas e o cálculo automático das avaliações asseguram dados precisos e operabilidade otimizada. Além disso, a gestão do pagamento e status das corridas ao final de cada viagem reforça a segurança e controle. Desta forma, a estrutura foi planejada para oferecer eficiência e agilidade em processos reais, buscando refletir as necessidades e dinâmicas do mundo prático.

2 MODELO CONCEITUAL

Para iniciar o projeto, foi criado o projeto conceitual por meio do modelo entidade e relacionamento (MER), que visa solidificar a concepção do sistema de gerenciamento de corridas. O MER organiza as principais entidades, como motorista, passageiro, veículo, corrida, avaliação, pagamento e localização, juntamente com seus atributos e relacionamentos. Essa estrutura reflete a operação real do sistema de transporte, proporcionando uma base sólida para a implementação do banco de dados, garantindo consistência nos dados e facilitando o desenvolvimento inicial.

Imagem 1 - Modelo entidade e relacionamento (MER).



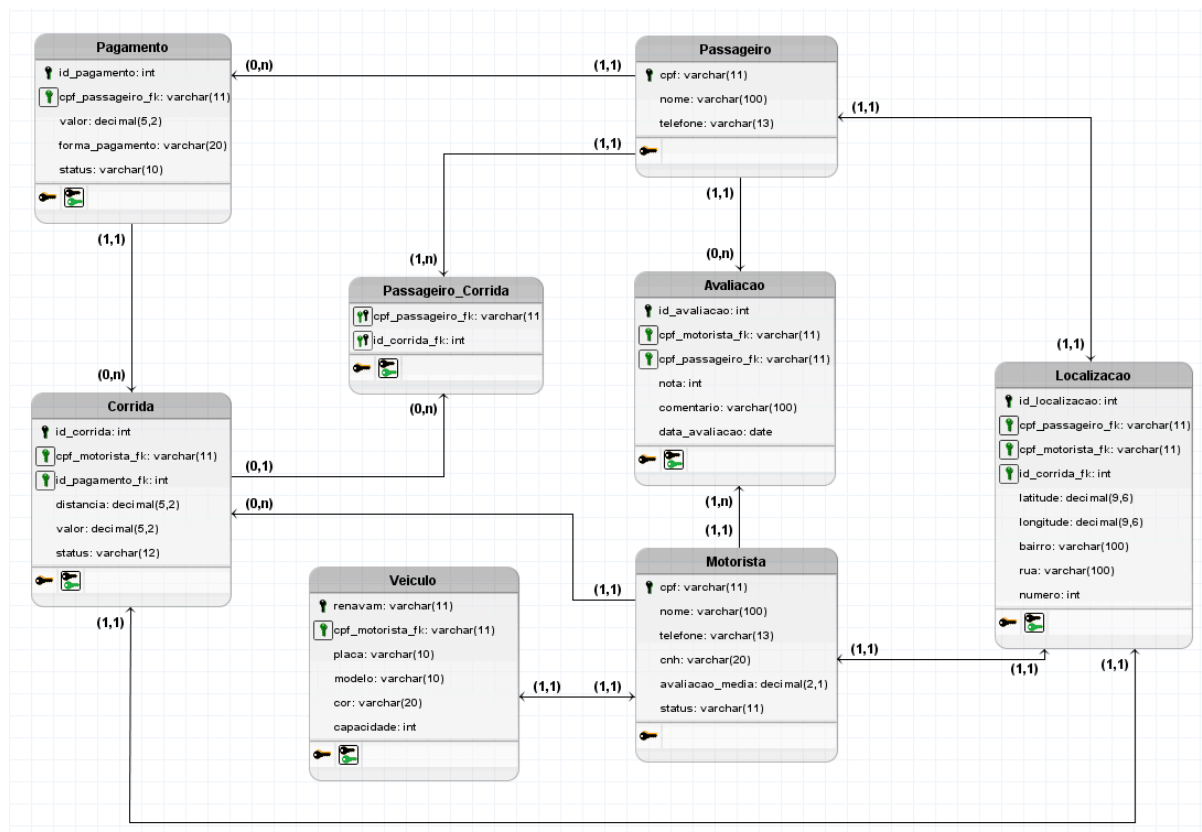
Fonte: Fotografia do autor - Captura de tela (07/03/2025).

3 MODELO LÓGICO

3.1 MODELO RELACIONAL

Após finalizar o modelo conceitual, é essencial reduzir o nível de abstração para o modelo relacional (MR), a fim de detalhar como as entidades serão representadas nas tabelas e como se relacionarão entre si. Nesse estágio, é necessário mapear cada entidade para uma tabela, definindo as chaves primárias e estrangeiras, e estabelecendo suas restrições de integridade. Criado através do MER, o modelo relacional traduz a estrutura conceitual em um formato pronto para ser implementado em um banco de dados, de forma que a comunicação entre as tabelas e suas operações sejam sempre consistentes e eficazes. Desta maneira, a partir desse modelo, é possível estabelecer um relacionamento claro entre as entidades, refletindo as interações do sistema de forma concreta e otimizada.

Imagem 2 - Modelo relacional (MR).



Fonte: Fotografia do autor - Captura de tela (07/03/2025).

3.2 DICIONÁRIO DE DADOS

Após a finalização do MR, com o intuito de visualizar melhor os dados, seus tipos e suas restrições, foi criado um dicionário de dados. Esse dicionário é essencial para o gerenciamento de informações dentro de um banco de dados, fornecendo detalhes sobre as tabelas, campos e tipos de dados que compõem a estrutura. Com isso, é utilizado para documentar as características de cada elemento do banco de dados, promovendo uma melhor organização e compreensão do sistema. A seguir, apresento o dicionário de dados com base no projeto de gerenciamento de corridas:

PASSAGEIRO

Atributo	Tipo	Nul o	Descrição	Domínio	Pri	Est	Cand
cpf	texto(11)	N	cpf do passageiro		X		
nome	texto(100)	N	nome do passageiro				
telefone	texto(13)		telefone do passageiro				

MOTORISTA

Atributo	Tipo	Nulo	Descrição	Domínio	Pri	Est	Cand
cpf	texto(11)	N	cpf do motorista		X		
nome	texto(100)	N	nome do motorista				
telefone	texto(13)		telefone do motorista				
cnh	texto(20)	N	cnh do motorista				
avaliacao_media	dec(2,1)		avaliação média do app				
status	texto(11)	N	status de atividade				

VEÍCULO

Atributo	Tipo	Nulo	Descrição	Domínio	Pri	Est	Cand
renavam	texto(11)	N	renavam do veículo		X		
placa	texto(10)	N	placa do veículo				
modelo	texto(10)	N	modelo do veículo				
cor	texto(20)		cor do veículo				
capacidade	inteiro	N	capacidade de passageiros				
cpf_motorista	texto(11)	N	cpf do motorista			X	

AValiação

Atributo	Tipo	Nulo	Descrição	Domínio	Pri	Est	Cand
id_avaliacao	inteiro	N	identificador da avaliação		X		
nota	inteiro	N	nota da corrida				
comentario	texto(100)		comentário sobre a corrida				
data_avaliacao	data	N	data da avaliação				
cpf_motorista	texto(11)	N	cpf do motorista			X	
cpf_passageiro	texto(11)	N	cpf do passageiro			X	

LOCALIZAÇÃO

Atributo	Tipo	Nulo	Descrição	Domínio	Pri	Est	Cand
id_localizacao	inteiro	N	identificador da localização		X		
latitude	dec(9,6)	N	latitude da coordenada				
longitude	dec(9,6)	N	longitude da coordenada				
bairro	texto(100)		bairro do usuário/destino				
rua	texto(100)		rua do usuário/destino				
numero	inteiro		número do usuário/destino				
cpf_motorista	texto(11)		cpf do motorista			X	
cpf_passageiro	texto(11)		cpf do passageiro			X	
id_corrida	inteiro		identificador da corrida			X	

PAGAMENTO

Atributo	Tipo	Nulo	Descrição	Domínio	Pri	Est	Cand
id_pagamento	inteiro	N	identificador do pagamento		X		
valor	dec(5,2)		valor da corrida				
forma_pagamento	texto(20)		método de pagamento				
status	texto(10)	N	status do pagamento				
cpf_passageiro	texto(11)		cpf do passageiro			X	

CORRIDA

Atributo	Tipo	Nulo	Descrição	Domínio	Pri	Est	Cand
id_corrida	inteiro	N	identificador da corrida		X		
distancia	dec(5,2)		distância da corrida				
valor	dec(5,2)		valor da corrida				
status	texto(12)	N	status da corrida				
cpf_motorista	texto(11)		cpf do motorista			X	
id_pagamento	inteiro		identificador do pagamento			X	

PASSAGEIRO_CORRIDA

Atributo	Tipo	Nulo	Descrição	Domínio	Pri	Est	Cand
cpf_passageiro	varchar(11)	N	cpf do passageiro		X	X	
id_corrida	inteiro	N	identificador da corrida		X	X	

4 MODELO FÍSICO

O modelo físico é a etapa final do desenvolvimento do banco de dados, onde as estruturas definidas no modelo lógico são implementadas em um SGBD. Nessa fase, tabelas, restrições e relacionamentos são convertidos em elementos concretos do sistema, garantindo sua funcionalidade e eficiência. Desta forma, com o intuito de implementar os scripts foram utilizadas as ferramentas MySQL, para a criação e gerenciamento do banco de dados, e VSCode, como ambiente de desenvolvimento para organizar e executar as queries.

Além disso, para aderir boas práticas ao desenvolvimento, as queries foram organizadas em arquivos separados conforme sua finalidade, facilitando a manutenção e reutilização do código. Para isso, foram criados quatro arquivos distintos, cada um com uma função específica: um para a criação da estrutura do banco de dados, outro para o povoamento com dados iniciais, um terceiro para operações de atualização, mais um para consultas e, por fim, um arquivo dedicado à simulação real de uma corrida. Desta maneira, utilizando a modularização das queries a estruturação do projeto fica mais clara e eficiente.

4.1 SCRIPTS DE ESTRUTURAÇÃO

Para dar início ao sistema, foi produzido um arquivo com scripts de estruturação chamado *Criar_Estrutura.sql* responsáveis por criar a base do banco de dados, definindo suas tabelas, atributos e restrições essenciais para garantir a integridade dos dados. Nele, foram estabelecidas as chaves primárias e estrangeiras, que garantem a relação entre as tabelas, assim como as restrições de seus campos, como *NOT NULL* e *CHECK*, que asseguram a integridade e a consistência dos dados.

4.2 POVOAMENTO COM DADOS INICIAIS

Após a criação da estrutura do banco de dados, o próximo passo é o povoamento das tabelas com dados iniciais. Esse processo consiste em inserir registros fictícios nas tabelas, permitindo que o sistema seja testado de forma prática e para isso foi utilizado um arquivo chamado *Inserir_Dados.sql*. Após inseridos, esses dados são essenciais para validar a integridade das relações entre as tabelas e garantir o funcionamento adequado das operações no banco de dados.

4.3 SCRIPTS DE ATUALIZAÇÕES

Para modificar dados no banco de dados foi necessário utilizar um arquivo chamado *Atualizar_Dados.sql* que contém instruções como *UPDATE* e *DELETE* para atualizar e deletar registros. Estes scripts são essenciais para manter a integridade e a atualização das informações no banco, garantindo que o sistema reflita as mudanças realizadas pelas interações dos usuários.

4.4 SCRIPTS DE CONSULTAS

Elaborados com o objetivo de simular demandas reais do sistema, os scripts de consultas criados no arquivo *Consultar_Dados.sql* utilizam conceitos de agregação, agrupamento e junção de tabelas. Ademais, foram criadas consultas que refletissem cenários cotidianos, como a busca por corridas realizadas, a análise das avaliações de motoristas ou o histórico de pagamentos. A utilização de funções de agregação, como *COUNT*, *SUM* e *AVG*, e a aplicação de *GROUP BY* garantem a extração de dados consolidados, enquanto os *JOINS* asseguram que as informações de diferentes tabelas sejam corretamente combinadas e apresentadas de maneira eficiente.

4.5 SIMULANDO CORRIDA

Neste tópico, é realizada a simulação da manipulação dos dados com base em uma corrida completa, desde o cadastro inicial do motorista até a avaliação final de sua corrida. O processo começa com o cadastro do motorista e seu veículo, seguido pela ativação de seu status para "ativo" e a atualização de sua localização. Em seguida, um usuário aciona a corrida, informando sua localização e destino. O motorista, ao receber a solicitação, aceita a corrida e segue para o destino. Após a conclusão da corrida, o pagamento é liberado e realizado, e a corrida é marcada como finalizada. Por fim, o passageiro realiza a avaliação da corrida, o que resulta na atualização dos dados do motorista com base na nota recebida, concluindo todo o ciclo de uma corrida. Esse processo simula de forma realista a sequência de operações e interações entre os dados no sistema, desde o início até o término de uma corrida.

4 CONSIDERAÇÕES FINAIS

Por fim, este projeto foi essencial para aplicar na prática os conceitos adquiridos durante as aulas da disciplina de Banco de Dados do curso de Engenharia de Computação. Ao desenvolver cada etapa, desde a modelagem conceitual até a implementação do banco de dados físico, foi possível consolidar o entendimento sobre as melhores práticas no design e na gestão de dados. Além disso, obtive um aprendizado valioso em banco de dados, aprimorando minhas competências em modelagem, otimização de consultas, definição de restrições e implementação de soluções eficientes para armazenamento de dados.

Ademais, vale lembrar que durante o desenvolvimento do projeto, todas as alterações realizadas foram constantemente atualizadas e armazenadas no repositório do GitHub, garantindo o versionamento adequado e a rastreabilidade das modificações. Isso não só facilitou o progresso, mas também proporcionou uma documentação organizada e acessível. O repositório pode ser acessado através do seguinte link: https://github.com/David-Mdrs/Aplicativo_Corridas_MySQL.