



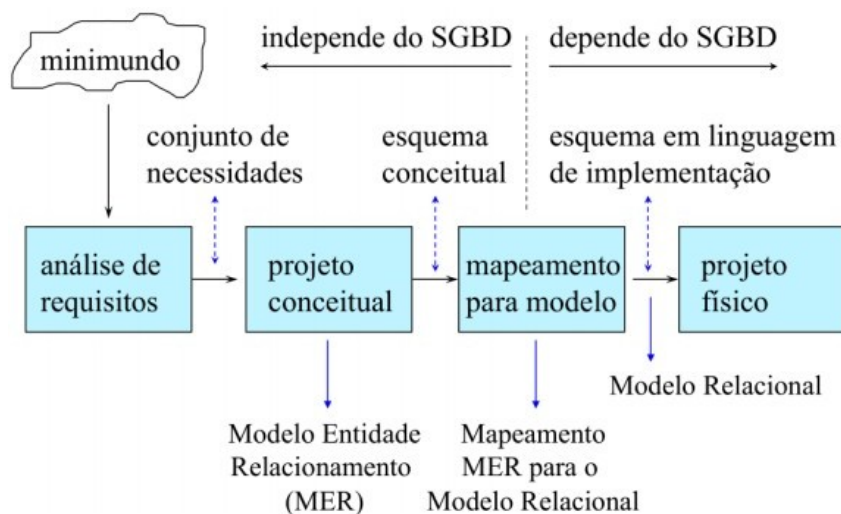
## TRABALHO FINAL DE DISCIPLINA

### Avisos Importantes:

- Trabalho vale 50% da Nota Final!
- Trabalho a ser realizado em grupo!
- Não compartilhem suas respostas. Haverá verificação de cópia entre os trabalhos (inclusive implementação). Vocês podem ser os maiores prejudicados em casos de plágio!
- Não deixem de enviar o relatório por e-mail no prazo especificado em aula!

### Especificação do Trabalho

Implementar um projeto de banco de dados relacional desde a sua modelagem lógica até a sua efetiva implementação em um SGBD relacional, passando, necessariamente, pelas etapas de (i) mapeamento entre os Modelos Entidade-Relacionamento/Relacional e (ii) normalização (1FN, 2FN e 3FN). O relatório entregue deve conter todos os passos da modelagem de acordo com o algoritmo de projeto de banco de dados discutido em aula e resumido na figura abaixo.



A implementação das tabelas deve seguir a especificação padrão da linguagem SQL 2003 em diante. Recomenda-se o uso do SGBD PostgreSQL. Cada tabela gerada deve ser povoada com pelo menos 20 tuplas que podem ser geradas a partir de dados fictícios. Os *scripts* de criação e povoamento das tabelas devem ser entregues junto com os relatórios. Os comandos SQL para responder às consultas especificadas ao final do trabalho devem ser entregues como um *script* separado e também devem ser incluídas no relatório. Os *scripts* entregues serão utilizados na avaliação do trabalho.

### **Requisitos do banco de dados**

Deseja-se projetar a base de dados de um sistema de transporte de passageiros que reservam suas viagens por celular e são atendidos por motoristas autônomos que efetivamente realizam o traslado (corrida) em seus veículos particulares. Esse sistema tem por objetivo atender corridas longas (intermunicipais).

Construa o modelo ER referente à base de dados. A base de dados não deve conter redundância de dados. O modelo ER deve ser representado com a notação vista nas aulas. **Não é permitido o uso de chaves artificiais no modelo ER.** Mapeie o modelo ER para o modelo relacional. Normalize o modelo relacional mapeado. **É permitido o uso de chaves artificiais no modelo relacional**, desde que sejam apresentadas justificativas adequadas.

A base de dados deve incluir motoristas, identificados por seus CPFs, RG, endereço, data de nascimento, número da CNH e conta corrente para pagamento, bem como passageiros, identificados por seus CPFs, RG, endereço, data de nascimento e número do cartão de crédito. Além deles, deve-se registrar os atendentes, que são identificados por seus CPFs, RG, endereço, data de nascimento e formação escolar. Motoristas e atendentes podem usar os serviços de transporte como passageiros. Um atendente pode ser um motorista nas horas vagas.

Um ou mais veículos são associados a um motorista, sendo que cada veículo deve ser identificado pelo seu número de RENAVAM, data de compra, marca, modelo, ano, preço e característica, que deve ser um valor entre {econômico, SUV, luxo, rural}.

Uma seguradora é identificada por um CNPJ, nome, telefone e endereço postal. Uma seguradora pode ser responsável por um ou mais seguros de veículos, que são identificados por número e valor da apólice. O sistema não armazena o histórico de apólices dos veículos; logo, no máximo, apenas um único número de apólice por veículo aparece no sistema por vez.

Uma corrida é registrada como o serviço de um motorista para um ou vários passageiros, sendo identificada por um número sequencial de corrida (definido por passageiro), endereço de destino, data de início, data de fim, hora de início, hora de fim, valor total da corrida e preço da corrida

por passageiro. Cada corrida tem ao menos um passageiro que a solicita (passageiro principal) e que, portanto, determina qual será o número sequencial da corrida.

Os demais passageiros, quando existirem, são considerados passageiros "caronistas". O preço total da corrida é fixo, mas o valor por passageiro é arbitrário. Cada passageiro, principal ou caronista, pode pagar uma quantia até que o valor total da corrida esteja completo. Todos os passageiros saem da mesma origem e vão para o mesmo destino. Um mesmo passageiro pode pedir várias corridas, desde que em horas e dias diferentes e que não esteja atualmente realizando uma corrida, em qualquer condição.

Um atendente é vinculado ao suporte de cada corrida, sendo que os passageiros podem enviar questionamentos, identificados por um número sequencial por corrida e texto da pergunta, ao atendente durante a corrida. O atendente deve responder com um texto da resposta. O sistema armazena o histórico de corridas e, conseqüentemente, dos atendimentos realizados.

### **Consultas a serem respondidas**

1. Dado uma seguradora, identificar quais são os veículos com seguros sob sua responsabilidade e listar as vigências das apólices.
2. Dado um atendente, descubra se ele já usou o serviço de corridas e quando.
3. Listar quais foram os passageiros que nunca fizeram uma corrida.
4. Dado um passageiro, mostrar a quantidade de vezes onde ele foi o passageiro principal ou o caronista.
5. Dada uma corrida, listar todas as perguntas e respostas feitas ao atendente responsável.
6. Listar todos os passageiros e o valor percentual pago por eles em corridas com mais de um passageiro.
7. Listar todos os passageiros e mostrar a quantidade de vezes onde ele foi o passageiro principal ou o passageiro caronista.
8. Duas perguntas que o grupo julgar relevante para os requisitos.

**Pontos extras NÃO são obrigatórios. Somente serão considerados os pontos extras APÓS o grupo responder os oito itens obrigatórios!**

9. **[Ponto extra 01 - Grau de dificuldade: Baixo]**. Repita as perguntas 6 e 7 com os resultados por passageiros e por ano durante a última década (2020-2025).
10. **[Ponto extra 02 - Grau de dificuldade: Sofisticado]**. Dado o modelo relacional final e a implementação do modelo no Postgres, o grupo vislumbra alguma possibilidade de consultar os dados com **consultas diretamente em português e sem usar SQL explicitamente ou via ORM?** Como seria isso? Ilustre com uma prova de conceito.