

UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação
Bacharelado em Ciências de Computação

Projeto de Curso
SCC0540 - Bases de Dados
Professor Moacir Ponti Júnior

PARTE 1

**Modelagem, Especificação do Problema
e Esquema Conceitual**

São Carlos, 14 de setembro de 2011

Projeto Desenvolvido pelo Aluno:

Ubiratan Soares (5634292) - ubiratan.f.soares@gmail.com

1. Objetivos do Projeto

Este projeto tem por objetivo a construção de uma aplicação completa de Banco de Dados, envolvendo desde a compreensão do problema a ser resolvido até o desenvolvimento da interface do sistema para os usuários finais.

A intenção inicial é desenvolver a aplicação em uma arquitetura cliente-servidor em três camadas, atentando em especial para o ciclo de vida que norteia o desenvolvimento de um Sistema de Banco de Dados, em conjunto com o desenvolvimento dos respectivos softwares de aplicação [1][4].



Figura 1.1 - Ciclo de Vida em Desenvolvimento de Sistemas a partir da especificação dos dados

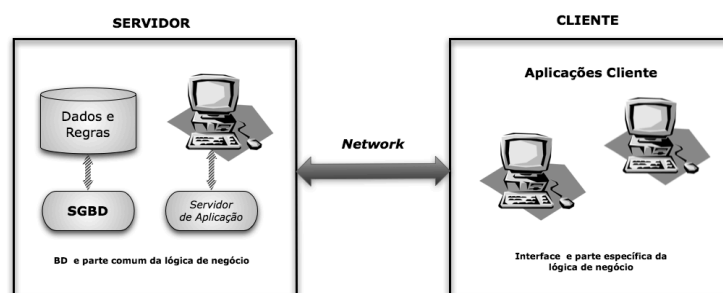


Figura 1.2 - Arquitetura Cliente/Servidor em Três Camadas

A forma de implementação das aplicações clientes - incluindo todos os detalhes técnicos referentes a plataformas de operação, aplicação SGBD, protocolos de comunicação utilizados, linguagens de programação e outros - será definida em um próximo momento do projeto.

2. Definição de Escopo e Descrição das Entidades e Relacionamentos

Conforme proposto na especificação do projeto, esse trabalho consiste em desenvolver uma aplicação para armazenar e recuperar dados em um contexto de venda de passagens, com a natureza final da aplicação a ser escolhida pelo aluno.

Assim, foi definido como o contexto da aplicação um sistema de venda de passagens, para uma empresa de transportes rodoviários. Com base na descrição dos pré-requisitos mínimos que o sistema deve suportar e na natureza desse sistema, são definidas a seguir as entidades e relacionamentos que descrevem o comportamento dos dados e indicam desde já alguns aspectos da lógica de negócio.

Também é preciso destacar que o modelo semântico empregado na confecção do esquema conceitual é o Modelo Entidade Relacionamento Estendido [3][4]. Em adição ao modelo proposto por Chen [1], o modelo adotado prevê maior arcabouço de ferramentas de modelagem, capazes de angariar mais semântica em relação aos dados sendo modelados. O diagrama desenvolvido referente ao esquema segue no anexo I.

Descrição das Entidades Fortes

- **MOTORISTA** : modela um motorista apto a conduzir um dos veículos da empresa de transportes. É identificado unicamente por um número de identificação (**id**), e possui como atributos iniciais nome completo (**nome**), data de nascimentos (**nascimento**) e data de admissão (**admissão**).
- **ÔNIBUS** : modela um dos veículos da empresa. Possui como chave-primária um número de identificação (**id**), e possui como atributos a marca do fabricante do veículo (**marca**), o modelo do veículo (**modelo**) e o ano em que foi fabricado (**ano**).
- **CIDADE** : modela uma cidade atendida pela empresa. Possui um número de identificação (**id**) como chave-primária, além do nome do município (**nome**).
- **PASSEIRO** : modela o cliente da empresa. É identificado pelo documento de CPF (**CPF**), e tem como atributos adicionais nome completo (**nome**) e o documento de identidade (**RG**).
- **LINHA** : modela uma das linhas de ônibus da companhia e traz as informações associadas sobre itinerários e trajetos. Possui como chave uma atributo numérico (**id**), além de um nome de referência (**nome**).
- **VIAGEM** : modela uma viagem executada pela empresa de ônibus. É indenticada por um atributo numérico (**id**), além de possuir os atributos data de viagem (**data**) e horário de partida da cidade de origem (**horário**). Esse modelo permite, por exemplo, que dois ônibus possam ser alocados em uma mesma linha, executando uma viagem partindo do mesmo dia e horário.

Entidades Fracas

- **POLTRONA** : modela uma das poltronas de cada ônibus. Possui como chave parcial um número de identificação (**número**). Será utilizada para manter o controle de poltronas por trecho da viagem, mas pode ser dependente da existência de um ônibus e não poder ser identificada unicamente, é modelada como uma entidade fraca.

Relacionamentos

- **CONTÉM** : relaciona as entidades ÔNIBUS e POLTRONA. Por ser uma entidade fraca, POLTRONA tem participação total nesse relacionamento, que tem cardinalidade **1:N**, indicando que um ônibus é formado por várias poltronas.

- **DIRIGE** : relaciona as entidades **motorista**, **ônibus** e **viagem**. Essa decisão de modelagem foi adotada considerando que, para uma mesma viagem pode haver troca de motoristas. Contudo, o modelo não admite que mais do que um veículo de ônibus participando de uma mesma viagem, de maneira que a cardinalidade desse relacionamento é **1:N:M**.
- **LUGAR** : relaciona a entidade forte **passageiro**, a entidade fraca **poltrona** e a agregação **trecho_viagem**, a ser descrita adiante. Traz a semântica de que um passageiro tem um lugar na viagem, ocupando uma poltrona durante um trecho do percurso, ou até durante o percurso inteiro do itinerário. Como cada poltrona só é ocupada por um passageiro, a cardinalidade aqui fica **1:N:M**.
- **CONECTA** : auto-relacionamento para a entidade cidade. Indica que duas cidades estão conectadas, trazendo os papéis **é_destino** e **é_origem**, com cardinalidade **N:M**.
- **FORMA** : relaciona a agregação **trecho_viagem** com a entidade **linha**. Traz a semântica de que uma linha atende várias cidades através de trechos de viagem distintos. A cardinalidade aqui é **N:M**.
- **INSTANCIA** : relaciona as entidades **linha** e **viagem**. Traz a semântica de que uma viagem é a instância de uma linha existente, de forma que a participação da entidade viagem no relacionamento é total. A cardinalidade aqui é **1:N**.
- **REGISTRA** : relaciona a entidade **viagem** com a agregação **passagem**, a ser descrita adiante. Traz a semântica de que uma viagem existe para o usuário através de uma passagem. Tem a função de indicar o histórico de passagens adquiridas por um passageiro. A cardinalidade aqui é **1:N**.

Agregações

- **REGISTRO_VIAGEM** : essa agregação se dá sobre o relacionamento **dirige**, no sentido de que é preciso manter um registro sobre quais motoristas atuam em cada viagem. Assim, essa agregação é determinada unicamente por uma chave composta por cinco atributos : a data em que a viagem é realizada (**data**), o horário em que o motorista parte com o ônibus (**partida**), o horário em que o motorista deixa o ônibus - para outro motorista ou no fim da viagem - (**chegada**), a identificação do motorista (**motorista**) e a identificação do veículo (**ônibus**). Os atributos data, motorista e ônibus derivam de outras entidades do modelo.
- **TRECHO_VIAGEM** : essa agregação se dá sobre o auto-relacionamento **conecta**, e indica que duas cidades estão conectadas por um ou mais trajetos distintos, que devem ser identificados dentro do itinerário da linha. Dessa forma, a viagem é a instância de uma linha, que por sua vez é composta por diversos trechos, cada qual com uma **origem** e **destino**, de forma que trechos adjacentes estão unidos. Esse controle deverá ser realizado por software.

Além disso, o passageiro requer um lugar em um trecho da viagem, ou seja, entre uma origem e um destino qualquer dentro do itinerário da linha. Essa agregação possui como chave um atributos **origem** e **destino** indicado o percurso indicado, além de atributos para o comprimento em quilômetros do trecho (**comprimento**) e as estradas que compõem o mesmo (**estradas**). Esse último deve ser um atributo multi-valorado.

- **PASSAGEM** : essa agregação se dá sobre o relacionamento **lugar**, e indica que a passagem identifica o lugar do passageiro em um trecho da viagem. Possui seis atributos, sendo a chave primária um atributo numérico (**id**). Os demais atributos que indicam propriedades da passagem e são comumente encontrados nas passagens de ônibus do mundo real : **origem**, **destino**, **data** da viagem, **horário** de partida e **poltrona**.

3. Principais Funcionalidades a Serem Implementadas

O sistema proposto admite em princípio dois tipos de usuário, com poderes e motivações distintos. É de responsabilidade do sistema garantir que qualquer operação realizada por um usuário não leve os dados a um estado de inconsistência, prevendo e gerenciando todas as operações e zelando de maneira que as restrições de integridade dos dados sejam sempre respeitadas.

Assim, as funcionalidades para cada tipo de usuário são descritas a seguir.

3.1. Administrador do sistema

Será o responsável por gerenciar o *backend* do sistema aqui proposto. As principais funcionalidades desse usuário são :

- Cadastrar, alterar e remover motoristas do sistema;
- Cadastrar, alterar e remover veículos de ônibus no sistema;
- Cadastrar, alterar e remover cidades no sistema;
- Cadastrar, alterar e remover linhas de ônibus do sistema;
- Criar, alterar e remover trechos de viagem (origem e destino) no sistema;
- Criar, alterar e remover registros de viagem no sistema;
- Criar, alterar e remover viagens no sistema;
- Consultar todos os passageiros presentes em uma viagem
- Consultar quando um passageiro específico viajou pela última vez

3.2. Cliente do sistema

Será o usuário no *frontend*, representa o passageiro e deseja adquirir e gerenciar suas passagens.

Um detalhe importante a ser tratado aqui está relacionado ao caráter prático da compra de passagens. Por motivos de escopo desse projeto, o sistema não será integrado nenhum sistema externo de pagamentos, de maneira que a validação de uma compra se dá de forma simulada, sempre supondo que a transação de pagamento ocorreu com sucesso. Após esse procedimento, o cliente pode imprimir um *voucher* da passagem, que hipoteticamente seria utilizado para embarque.

Assim, as principais funcionalidades desse usuário são :

- Criar, alterar e remover uma conta no sistema;
- Consultar as cidades de origem e destino para viagens oferecidas pela empresa;
- Consultar disponibilidade de lugares em uma viagem;
- Comprar passagens, reservando lugares no ônibus entre origem e destino de interesse;
- Imprimir um *voucher* referente à compra de passagens;
- Cancelar uma passagem com uma antecedência de tempo pré-definida;
- Consultar o histórico de passagens adquiridas, utilizadas, em aberto e canceladas
- Remover do histórico passagens já utilizadas
- Restituir passagens canceladas, através da aquisição de uma nova passagem de mesmas características (origem e destino)

4. Considerações Finais

O modelo conceitual desenvolvido reflete a proposta do projeto, dadas as considerações semânticas aqui apresentadas e alguns aspectos já visualizáveis referentes à implementação.

O maior desafio do projeto reside na complexidade do mesmo. Tanto o *frontend* quanto o *backend* nas aplicações clientes são extensos, com diversas funcionalidades apoiadas em sobre diversas tabelas, o que aumenta a complexidade de tempo no momento da implementação.

5. Referências

1. **DeSOUZA, E. P. M.**; Notas de Aulas - *“Introdução aos Sistemas BD”, “MER Parte 1”, “MER Parte 2”, “Agregação”, “Especialização”* referentes à disciplina de Banco de Dados, ICMC, USP, São Carlos, 2009.
2. **ELMASRI, R. ; NAVATHE, S. B.** - *“Sistemas de Banco de Dados”*. Addison Wesley, 4a Edição.
3. **SILBERSCHATZ, A. et al.** - *“Sistemas de Banco de Dados”*. McGraw-Hill, 3a Edição.