



Rodoviária Itajubá

Sistema de Controle Interno de Rodoviária

DOCUMENTO FINAL DO PROJETO

Grupo: 18

Fernando José Ferreira Neto
Giovany da Silva Santos



Revisões do Documento

Revisões são melhoramentos na estrutura do documento e também no seu conteúdo. O objetivo primário desta tabela é a fácil identificação da versão do documento. Toda modificação no documento deve constar nesta tabela.

Data	Versão	Descrição	Autor
04/06/2020	1.0	Definição do objetivo e requisitos funcionais	Fernando José Giovany da Silva
11/06/2020	2.0	Revisão e correção do documento	Fernando José Giovany da Silva
23/06/2020	3.0	Revisão e correção do documento	Fernando José Giovany da Silva
07/07/2020	4.0	Documento final	Fernando José Giovany da Silva



1. Introdução

Este documento especifica os requisitos do Sistema de Controle Interno de Rodoviária, fornecendo aos desenvolvedores as informações necessárias para a execução de seu projeto e implementação, assim como para a realização dos testes e homologação.

Esta introdução fornece as informações necessárias para fazer um bom uso deste documento, explicitando seus objetivos e as convenções que foram adotadas no texto. As demais seções apresentam a especificação do Sistema de Controle Interno de Rodoviária e estão organizadas como descrito abaixo:

- **Seção 2 - Descrição geral do produto/serviço:** apresenta uma visão geral do produto/serviço, caracterizando qual é o seu escopo e descrevendo seus usuários.
- **Seção 3 - Requisitos funcionais:** lista e descreve os requisitos funcionais do produto/serviço, especificando seus objetivos, funcionalidades, atores e prioridades.
- **Seção 4 - Modelo Entidade Relacionamento:** apresenta o modelo entidade relacionamento considerando os requisitos funcionais identificados na seção 3.
- **Seção 5 – Modelo Lógico:** apresenta o mapeamento do modelo entidade relacionamento da seção 4 para tabelas do modelo relacional. O modelo lógico pode ser apresentado através de um diagrama de tabelas ou usando a notação vista em aula.
- **Seção 6 – Estratégias de desnormalização do modelo lógico:** apresenta um levantamento de quais tabelas podem ser desnormalizadas para efeito de otimizar a performance do banco
- **Seção 7 – Consultas e Visões:** apresenta uma estimativa das 10 consultas mais realizadas no banco. As consultas são apresentadas em SQL e Álgebra relacional. Além disso, deverão ser apresentadas as visões criadas no projeto (código de criação das visões).



2. Visão geral do Produto/serviço

O sistema descrito refere-se ao controle de uma rodoviária, tem como funcionalidades administrar os funcionários, ônibus, passagens, rotas, revisões dos ônibus, empresas proprietárias dos ônibus, e contratos que são feitos entre a rodoviária e as empresas. O sistema é off-line e voltado para auto administração da rodoviária.

2.1 Descrição dos usuários

Os usuários do sistema são gerente e balconista.

2.1.1 Gerente

O Gerente pode realizar todas as funções do sistema.

2.1.2 Balconista

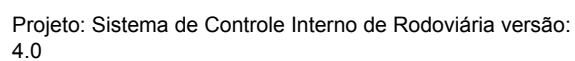
O Balconista pode realizar todas as funções do sistema, exceto a manutenção de funcionários.

3. Requisitos funcionais

Número	Descrição	Prioridade
RF1	<i>O s.d.p. manter informações sobre funcionários contendo os seguintes dados: nome, CPF (identificador), endereço (rua, número, bairro, cidade, estado), telefone(s), cargo (gerente, motorista, cobrador e balconista), data de nascimento. Um motorista faz um ou vários trajetos. Um cobrador faz um ou vários trajetos. Um balconista vende nenhuma ou várias passagens. O telefone do funcionário precisa ter como padrão tamanho igual a 9, caso não siga o padrão o telefone será deletado.</i>	(E)
RF2	<i>O s.d.p. manter informações sobre passagem contendo os seguintes dados: código (identificador) e valor. Uma passagem está associada: a somente uma rota, a um e somente um balconista.</i>	(E)
RF3	<i>O s.d.p. manter informações sobre rota considerando os seguintes dados: código (identificador), origem, destino, duração, distância, paradas, dias da semana,</i>	(E)



	<i>horário de partida, horário de chegada. Uma rota está associada: a uma ou várias passagens, a um ou vários trajetos.</i>	
<i>RF4</i>	<i>O s.d.p. manter informações sobre o trajeto contendo os seguintes dados: código do trajeto (identificador), data e consumo. Um trajeto só existe se estiver associado a uma rota. Um trajeto está associado: a uma e somente uma rota, a um e somente um ônibus, a um e somente um motorista e a um e somente um cobrador. Além disso, um motorista e um cobrador não podem estar alocados para trajetos diferentes na mesma data.</i>	<i>(E)</i>
<i>RF5</i>	<i>O s.d.p. manter informações sobre ônibus considerando as seguintes informações: código (identificador), modelo, ano, quilometragem, tanque, consumo de combustível, data da próxima revisão, lugares. Um ônibus está associado: a um ou vários trajetos, a nenhuma ou várias vistorias e a uma empresa ou mais empresas.</i>	<i>(E)</i>
<i>RF6</i>	<i>O s.d.p. manter informações sobre vistoria do ônibus considerando as seguintes informações: código da vistoria (identificador), quilometragem, data da revisão, estado. Uma vistoria está associada a um e somente um ônibus.</i>	<i>(E)</i>
<i>RF7</i>	<i>O s.d.p. manter informações sobre empresa considerando os seguintes dados: nome e CNPJ (identificador). Uma empresa está associada a um ou vários ônibus.</i>	<i>(E)</i>
<i>RF8</i>	<i>O s.d.p. manter informações sobre o contrato que será gerado entre a empresa que possui o ônibus e a rodoviária. O contrato contém as seguintes informações: código do contrato (identificador), vigência, valor. Quando um ou mais ônibus estão associados a uma ou mais empresas pode ser gerado um contrato. Um contrato está associado a uma ou mais relações entre ônibus e empresa.</i>	<i>(E)</i>





5. Modelo Lógico (Modelo Relacional)

rota(codigo, origem*, destino*, duracao*, distancia, paradas, dias_semana*, horario_partida*, horario_chegada*)

passagem(codigo, cod_rota*, valor*, id_funcionario*)

cod_rota referencia rota(codigo)

id_funcionario referencia a funcionario(CPF)

funcionário(CPF, nome*, rua*, numero*, bairro*, cidade*, estado*, cargo*, data_nascimento*)

chave candidata: nome

telefone(id_funcionario, telefone)

id_funcionario referencia o funcionario(CPF)

trajeto(cod_trajeto, cod_rota, cod_motorista*, cod_cobrador*, data*, consumo, cod_onibus*)

cod_rota referencia rota(codigo)

cod_motorista referencia funcionario(CPF)

cod_cobrador referencia funcionario(CPF)

cod_onibus referencia onibus(codigo)

chave candidata: data

onibus(codigo, modelo*, ano*, quilometragem*, tanque*, consumo_combustivel, data_prox_revisao*, lugares*)

vistoria(cod_vistoria, cod_onibus*, quilometragem*, data_revisao*, estado*)

cod_onibus referencia onibus(codigo)

empresa(CNPJ, nome*)

chave candidata: nome

contrato(codigo, vigência*, valor*)



possui(CNPJ, cod_onibus, cod_contrato*)

CNPJ referencia empresa(CNPJ)

cod_onibus referencia onibus(codigo)

cod_contrato referencia contrato(codigo)

6. Projeto de desnormalização

rota(codigo, origem*, destino*, duracao*, distancia*, paradas*, dias_semana*,
horario_partida*, horario_chegada*)

~~passagem(codigo, cod_rota, valor*, id_funcionario*)~~

~~cod_rota referencia rota(codigo)~~

~~id_funcionario referencia a funcionario(CPF)~~

~~funcionario(CPF, nome*, rua*, numero*, bairro*, cidade*, estado*, cargo*,
data_nascimento*)~~

~~chave candidata: nome~~

funcionario(CPF, id_passagem, nome*, rua*, numero*, bairro*, cidade*, estado*, cargo*,
data_nascimento*, cod_rota, valor)

cod_rota referencia rota(codigo)

Está tabela não está na 2FN pois contém dependências funcionais parciais em

CPF, id_passagem -> nome,

CPF, id_passagem -> rua,

CPF, id_passagem -> numero,

CPF, id_passagem -> bairro,

CPF, id_passagem -> cidade,

CPF, id_passagem -> estado,

CPF, id_passagem -> cargo,

CPF, id_passagem -> data_nascimento.

CPF, id_passagem->cod_rota

CPF, id_passagem-> valor.

**Ou seja, nome, rua, numero, bairro, cidade, estado, cargo, data_nascimento dependem
somente de CPF e cod_rota e valor dependem somente de id_passagem**

telefone(id_funcionario, telefone)

id_funcionario referencia o funcionario(CPF)



trajeto(cod_trajeto, cod_rota, cod_motorista*, cod_cobrador*, data*, consumo, cod_onibus*)

cod_rota referencia rota(codigo)

cod_motorista referencia funcionario(CPF)

cod_cobrador referencia funcionario(CPF)

cod_onibus referencia onibus(codigo)

chave candidata: data

onibus(codigo, modelo*, ano*, quilometragem*, tanque*, consumo_combustivel, data_prox_revisao*, lugares*)

vistoria(cod_vistoria, cod_onibus*, quilometragem*, data_revisao*, estado*)

cod_onibus referencia onibus(codigo)

empresa(CNPJ, nome*)

chave candidata: nome

contrato(codigo, vigência*, valor*)

possui(CNPJ, cod_onibus, cod_contrato*, vigência*, valor*)

CNPJ referencia empresa(CNPJ)

cod_onibus referencia onibus(codigo)

Está tabela não está na 3FN porque tem uma dependência funcional transitiva de vigência e valor em relação a CNPJ e cod_onibus, pois (CNPJ , cod_onibus) -> cod_contrato; cod_contrato-> vigência e cod_contrato-> valor



7. Consultas

7.1 Junção externa

Consulta o nome, cpf e código da passagem dos funcionários que vendem passagem, incluindo os que não vendem passagem

SQL

```
select f.nome, f.cpf, pa.codigo from funcionario f left outer join passagem pa on f.cpf = passagem.id_funcionario
```

Álgebra Relacional

$$\pi_{\text{nome}, \text{cpf}, \text{codigo}}(\sigma_{\text{funcionario.cpf}=\text{passagem.id_funcionario}}(\text{funcionario} \bowtie \text{passagem}))$$

7.2 Seleção com like

Consulta o nome e o cpf de todos os funcionários que possuem (“ovan”) no nome.

SQL

```
select nome, cpf from funcionario where nome like '%ovan%'
```

Álgebra Relacional

$$\pi_{\text{nome}, \text{cpf}}(\sigma_{\text{nome} = \%ovan\%}(\text{funcionario}))$$

7.3 Função de agregação

Consulta a quantidade de funcionários que moram em Minas Gerais (“mg”)

SQL

```
select count(*) from funcionario where estado = 'mg'
```

Álgebra Relacional

$$T \leftarrow \sigma_{\text{estado}=\text{"mg"}}(\text{funcionario})$$
$$Gcount(id_funcionario)(T)$$

7.4 Função de agregação com having

Consulta os dias da semana e a soma das distâncias apresentando somente os dias da semana que possui a soma das distâncias maiores que 1000



SQL

```
select dias_semana, sum(distancia) from rota group by (dias_semana) having
sum(distancia)>1000
```

Álgebra Relacional

$$T \leftarrow \text{dias_semana} \text{Gsum}(\text{distancia}) \text{ as dist (rota)}$$

$$S \leftarrow \sigma_{\text{dist}>1000}(T)$$

$$R \leftarrow \pi_{\text{dias_semana, sum}}(S)$$

7.5 Junção interna com mais de duas tabelas

Consulta os telefones de todos distintos dos funcionários que vendem passagem.

SQL

```
select distinct telefone from telefone tel inner join funcionario f on f.cpf =
tel.id_funcionario inner join passagem pa on pa.id_funcionario = f.cpf
```

Álgebra Relacional

$$T \leftarrow \text{passagem} \times (\text{funcionario} \times \text{telefone})$$

$$R \leftarrow \pi_{\text{distinct}(\text{telefone})} (\sigma_{\text{telefone.id_funcionario} = \text{funcionario.cpf} \wedge \text{funcionario.cpf} = \text{passagem.id_funcionario}}(T))$$

7.6 Operador de conjunto

Consulta o cpf de todos os funcionários que são motoristas e que estão alocados em algum trajeto.

SQL

```
(select cpf from funcionario) intersect (select cod_motorista from trajeto)
```

Álgebra Relacional

$$\pi_{\text{cpf}}(\text{funcionario}) \cap \pi_{\text{cod_motorista}}(\text{trajeto})$$

7.7 Junção Natural

Consulta o código e modelo dos ônibus que realizam o trajeto ('00002').

SQL

```
select codigo, modelo from onibus o natural join trajeto tra where tra.cod_trajeto =
'00002' and tra.cod_onibus=o.codigo
```

Álgebra Relacional

$$\pi_{\text{codigo, modelo}}(\sigma_{\text{trajeto.cod_trajeto}='00002'} \wedge \sigma_{\text{onibus.codigo} = \text{trajeto.cod_onibus}}(\text{onibus} \bowtie \text{trajeto}))$$



7.8 Consulta aninhada ‘=’

Consulta o nome e a cidade dos funcionários que moram na rua (“palmeiras”).

SQL

```
select nome, cidade from funcionario where rua = (select rua from funcionario where  
rua = ‘palmeiras’ )
```

Álgebra Relacional

$\pi_{\text{nome, cidade}} (\sigma_{\text{rua}=\text{“palmeiras”}}(\text{funcionario}))$

7.9 Consulta aninhada com in ou not in

Consulta a cidade dos funcionários que moram em cidades não são iguais a origem de alguma rota.

SQL

```
select cidade from funcionario where cidade not in (select origem from rota)
```

Álgebra Relacional

$\pi_{\text{cidade}} (\text{funcionario}) - \pi_{\text{origem}} (\text{rota})$

7.10 Divisão

Consulta o nome dos funcionários que vendem todas as passagens.

SQL

```
SELECT nome FROM funcionario WHERE NOT EXISTS ((SELECT  
id_funcionario FROM passagem)  
except (SELECT id_funcionario FROM passagem WHERE  
passagem.id_funcionario = funcionario.cpf))
```

Álgebra Relacional

$R \leftarrow \pi_{\text{nome, cpf}} (\text{funcionario})$

$S \leftarrow \pi_{\text{id_funcionario}} (\text{passagem})$

$R \div S$