

**UNIVERSIDADE FEDERAL DE ITAJUBÁ**

**CARLOS HENRIQUE REIS**

**JEAN CARLOS DE OLIVEIRA**

**MATEUS HENRIQUE TOLEDO**

**VICTOR RODRIGUES DA SILVA**

**SISTEMA DE APOIO À GESTÃO**

**DE HOSPITAIS AMERICANOS**

**ITAJUBÁ – MG**

**2017**

**SUMÁRIO**

[1. RELATÓRIO DO BANCO DE DADOS 3](#_Toc485203089)

[1.1. Diagrama de Entidade e Relacionamento 3](#_Toc485203090)

[1.2. Diagrama Relacional 4](#_Toc485203091)

[1.2.1. Primeira Forma Normal 4](#_Toc485203092)

[1.2.2. Segunda Forma Normal 4](#_Toc485203093)

[1.2.3. Terceira Forma N 5](#_Toc485203094)

[1.3. Definição de grupos de usuários e suas permissões. 5](#_Toc485203095)

[1.4. . \*\*\*\*\*\*\* Definição de índices e suas justificativas. 5](#_Toc485203096)

[1.4.1. Estado 5](#_Toc485203097)

[1.4.2. Período 5](#_Toc485203098)

[1.4.3. Rodovia 6](#_Toc485203099)

[1.4.4. Tipo acidente 6](#_Toc485203100)

[1.4.5. Trecho 6](#_Toc485203101)

[1.4.6. Acidente 6](#_Toc485203102)

[1.5. Definição de views, triggers, procedures, funções e suas justificativas. 6](#_Toc485203103)

[2. RESULTADO DO TESTE DE PERFORMANCE 6](#_Toc485203104)

[2.1. Rodovia 6](#_Toc485203105)

[2.2. Data hora registro ocorrência 6](#_Toc485203106)

[2.3. Quantidade acidente ano 7](#_Toc485203107)

[2.3.1. Ilesos 7](#_Toc485203108)

[2.3.2. Feridos Leves 7](#_Toc485203109)

[2.3.3. Feridos Graves 7](#_Toc485203110)

[2.3.4. Mortos 7](#_Toc485203111)

[2.4. Quantidade acidentes por estado 7](#_Toc485203112)

[2.5. Tipo de acidente 7](#_Toc485203113)

[2.6. Trecho 7](#_Toc485203114)

[2.7. Município 8](#_Toc485203115)

[3. DOCUMENTO DE REQUISITOS 9](#_Toc485203116)

[4. BACKUP DO SCHEMA 10](#_Toc485203117)

[4.1. Atendimento DRG 10](#_Toc485203118)

[4.2. DRG 10](#_Toc485203119)

[4.3. Hospitais Região 10](#_Toc485203120)

[4.4. Hospital 10](#_Toc485203121)

[4.5. Agente 11](#_Toc485203122)

[4.6. Gerente 11](#_Toc485203123)

[5. COUNT DAS TABELAS 11](#_Toc485203124)

[5.1. Atendimento DRG – 1058115 11](#_Toc485203125)

[5.2. DRG – 2110 12](#_Toc485203126)

[5.3. Hospitais Região – 960527 13](#_Toc485203127)

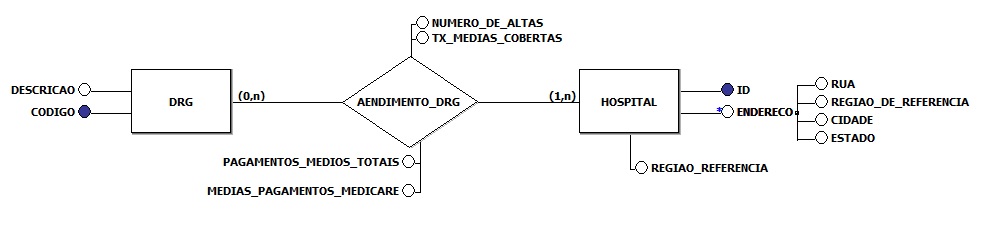
[5.4. Hospital - 193 13](#_Toc485203128)

[5.5. Agente – 17 14](#_Toc485203129)

[5.6. Gerente – 11593 14](#_Toc485203130)

# RELATÓRIO DO BANCO DE DADOS

## Diagrama de Entidade e Relacionamento

****

## Diagrama Relacional

### Primeira Forma Normal

Domínio atômico dos atributos:

* **drg** (codigo, definicao);
* **hospital** (idhospital, nomehospital);
* **endereço** (idhospital, regiao\_referencia, rua,cidade, estado);
* **atendimento\_drg (**codigo\_drg, idhospital, numero\_altas, taxas\_medias\_cobertas, pagamentos\_médios\_totais, media\_pagamentos\_medicare);

codigo\_drg referencia drg(codigo);

idhospital referencia hospital(idhospital);

**Dominância**

**DRG**: código → definição

**Hospital**: idhospital → nomehospital, região\_referência

**Endereço**: idhospital → rua,cidade,estado

**Gastos**: codigo\_drg, idhospital → numero\_altas, taxas\_medias\_cobertas, pagamentos\_médios\_totais, edia\_pagamentos\_medicare;

### Segunda Forma Normal

Todos os atributos devem depender totalmente da chave primária da tabela:

* **drg** (codigo, definição);
* **hospital** (idhospital, nomehospital);
* **hospitais\_região** (idhospital, regiao\_referencia);

idhospital referencia hospital(idhospital);

* **endereço** (idhospital, rua, cidade, estado);

idhospital referencia hospital(id);

* **atedimento\_drg** (codigo\_drg, idhospital, numero\_altas, taxas\_medias\_cobertas, pagamentos\_médios\_totais, media\_pagamentos\_medicare);

codigo\_drg referencia drg(codigo);

idhospital referencia hospital(idhospital);

### Terceira Forma N

Não podem existir dependências transitivas na tabela do tipo: a→ b e b→ c

* **drg** (codigo, definição);
* **hospital**(idhospital, nomehospital);
* **hospitais\_região**(idhospital, regiao\_referencia);

idhospital referencia hospital(idhospital);

* **endereço**(idhospital, rua,cidade,estado);

idhospital referencia hospital(id);

* **atendimento\_drg**(codigo\_DRG, idhospital, numero\_altas, taxas\_medias\_cobertas, pagamentos\_médios\_totais, media\_pagamentos\_medicare);

codigo\_DRG referencia drg(codigo);

idhospital referencia hospital(idhospital);

## Definição de grupos de usuários e suas permissões.

Para o acesso ao banco de dados, foi criado apenas um usuário chamado *‘Aplicação’*, que fica responsável por todo acesso por parte da aplicação. Seu nível de acesso é o de tabela: Inserir, alterar e remover dados de tabelas.

Para a manutenção e administração do banco de dados, o usuário ‘*root*’ é quem fica responsável, no caso, o DBA.

## . \*\*\*\*\*\*\* Definição de índices e suas justificativas.

Para fins de teste, foram implementados index para todas as tabelas do banco, com o intuito de quantificar e mensurar a otimização, a partir destes testes, foram escolhidos alguns index que demonstraram diferença satisfatória quando comparados com o SQL sem index. Segue listado abaixo todos os index criados para teste, para resultado de suas performances consultar capítulo 2 deste documento, onde cada teste será justificado, bem como serão apresentados os escolhidos para otimizar a aplicação.

### Estado

create index cidade\_uf\_municipio on cidade using btree(uf, municipio);

### Período

create index periodo\_data\_hora\_registro\_ocorrencia on periodo(data\_hora\_registro\_ocorrencia);

create index periodo\_ano on periodo(ano);

### Rodovia

create index rodovia\_br on rodovia(br);

### Tipo acidente

create index tipoacidente\_tipo\_acidente on tipoacidente(tipo\_acidente);

### Trecho

create index trecho\_trecho on trecho(trecho);

create index trecho\_id\_cidade\_fk on trecho(id\_cidade\_fk);

### Acidente

create index acidente\_id\_periodo\_fk on acidente(id\_periodo\_fk);

create index acidente\_qtd\_mortos on acidente(qtd\_mortos);

create index acidente\_qtd\_feridos\_graves on acidente(qtd\_feridos\_graves);

create index acidente\_qtd\_feridos\_leves on acidente(qtd\_feridos\_leves);

create index acidente\_qtd\_ilesos on acidente(qtd\_ilesos);

create index acidente\_qtd\_mortos on acidente(qtd\_mortos);

create index acidente\_id\_trecho\_fk on acidente(id\_trecho\_fk);

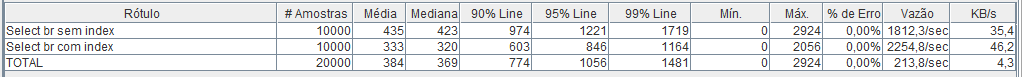
## Definição de views, triggers, procedures, funções e suas justificativas.

Para o desenvolvimento desta aplicação não foram utilizados views, triggers, procedures ou funções, todas as permissões de consultas foram tratadas a nível de aplicação.

# RESULTADO DO TESTE DE PERFORMANCE

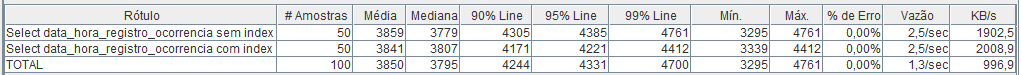
## Rodovia

Apesar de apresentar melhoria significativa, este índice não foi implementado por não haver nenhum relatório que utiliza tal consulta, desta forma, não se espera que seja uma consulta crítica para o banco, assim foi admitido que manter este índice apenas iria sobrecarregar o banco sem apresentar melhora que justificasse tal sobrecarga, index descartado.



## Data hora registro ocorrência

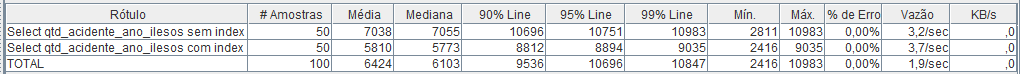
Não apresentou melhora significativa com a criação do índice, index descartado.



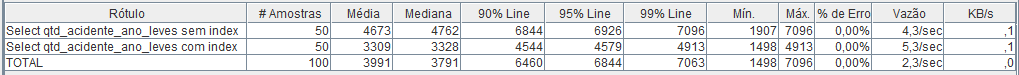
## Quantidade acidente ano

Apresentaram melhora significativa, e como as consultas são utilizadas no [RF12] subentendeu-se que se fazia necessário, para melhor performance da aplicação manter tais índices, index implementado.

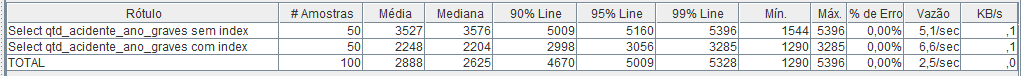
### Ilesos

****

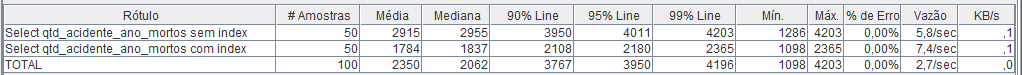
### **Feridos Leves**



### Feridos Graves

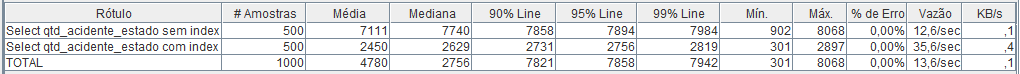
****

### Mortos



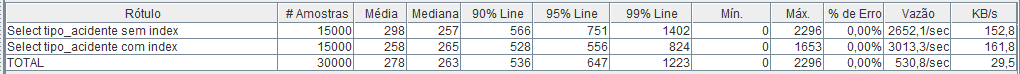
## Quantidade acidentes por estado

Apresentou melhora significativa, e como a consulta é utilizada no [RF13] subentendeu-se que se fazia necessário, para melhor performance da aplicação manter tal índice, index implementado.



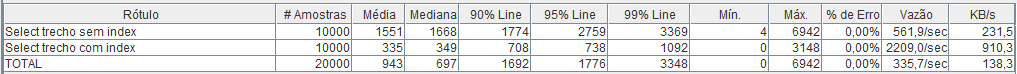
## Tipo de acidente

Não apresentou melhora significativa com a criação do índice, index descartado.



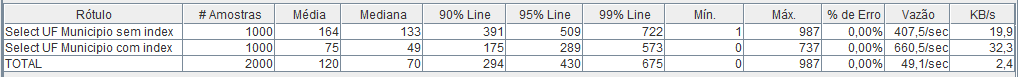
## Trecho

Apresentou melhora significativa, apesar de não ser implementada por nenhum relatório, subentende-se que consultas por trecho, bem como por município serão recorrentes no sistema, assim melhorando a performance da aplicação, índex implementado.



## Município

Apresentou melhora significativa, apesar de não ser implementada por nenhum relatório, subentende-se que consultas por município, bem como por trecho serão recorrentes no sistema, assim melhorando a performance da aplicação, índex implementado.



# DOCUMENTO DE REQUISITOS

**Sistema de Apoio à Gestão de**

**Hospitais Americanos**

**Cliente: Departamento de Saúde e Serviços Humanos**

**dos Estados Unidos da América (****HHS)**

SAGHA – Sistema de Apoio à Gestão de Hospitais Americanos  
DOCUMENTO DE REQUISITOS

Versão 1.0

**Responsáveis pela Proposta:**

***Carlos Henrique Reis – 30415***

**Graduando em Ciências da Computação**

**carlos\_henreis@unifei.edu.br**

***Jean Carlos de Oliveira – 35138***

**Graduando em Sistemas de Informação**

**jeancarlosdeoliveira@outlook.com**

***Mateus Henrique Toledo – 34849***

**Graduando em Sistemas de Informação**

**mateushtoledo@gmail.com**

***Victor Rodrigues da Silva –******31054***

**Graduando em Ciências da Computação**

**vitor1908@gmail.com**

# BACKUP DO SCHEMA

## Atendimento DRG

CREATE TABLE ‘atendimento\_drg’ (

‘codigodrg’ smallint(6) NOT NULL,

‘idhospital’ int(11) NOT NULL,

‘numeroaltas’ smallint(6) NOT NULL,

‘taxasmediascobertas’ float NOT NULL,

‘pagamentosmediostotais’ float NOT NULL,

‘mediapagamentosmedicare’ float NOT NULL,

PRIMARY KEY (‘idhospital’,’codigodrg’),

KEY ‘codigodrg’ (‘codigodrg’),

CONSTRAINT ‘atendimento\_drg\_ibfk\_1’ FOREIGN KEY (‘idhospital’)

REFERENCES ‘hospital’ (‘id’) ON UPDATE CASCADE,

CONSTRAINT ‘atendimento\_drg\_ibfk\_2’ FOREIGN KEY (‘codigodrg’)

REFERENCES ‘drg’ (‘codigo’) ON UPDATE CASCADE

)

## DRG

CREATE TABLE ‘drg’ (

‘codigo’ smallint(6) NOT NULL,

‘definicao’ varchar(128) NOT NULL,

PRIMARY KEY (‘codigo’)

)

## Hospitais Região

CREATE TABLE ‘hospitais\_regiao’ (

‘idhospital’ int(11) NOT NULL,

‘regiao\_referencia’ varchar(64) NOT NULL,

PRIMARY KEY (‘idhospital’,’regiao\_referencia’),

CONSTRAINT ‘hospitais\_regiao\_ibfk\_1’ FOREIGN KEY (‘idhospital’)

REFERENCES ‘hospital’ (‘id’) ON DELETE CASCADE

)

## Hospital

CREATE TABLE ‘hospital’ (

‘id’ int(11) NOT NULL,

‘nome’ varchar(64) NOT NULL,

‘rua’ varchar(52) NOT NULL,

‘cidade’ varchar(64) NOT NULL,

‘estado’ char(2) NOT NULL,

PRIMARY KEY (‘id’)

)

## Agente

CREATE TABLE agente(

‘nome’ varchar(64),

‘cpf’ char(11),

‘senha’ char(8),

PRIMARY KEY (‘cpf’)

)

## Gerente

CREATE TABLE gerente(

‘nome’ varchar(64),

‘cpf’ char(11),

‘senha’ char(8),

‘idhospital’ int,

PRIMARY KEY (‘cpf’),

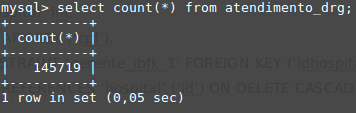
CONSTRAINT ‘gerente\_ibfk\_1’ FOREIGN KEY (‘idhospital’)

REFERENCES ‘hospital’ (‘id’) ON DELETE CASCADE

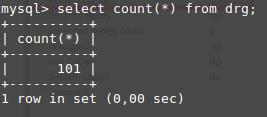
)

# COUNT DAS TABELAS

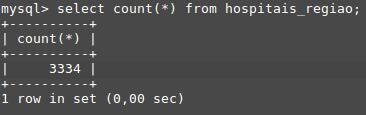
## Atendimento DRG – 145719

****

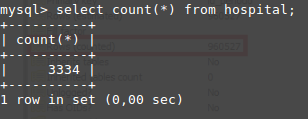
## DRG – 101

****

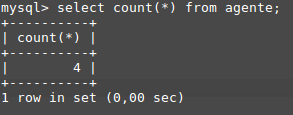
## Hospitais Região – 3334

****

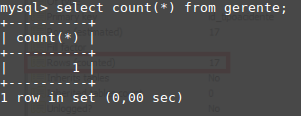
## Hospital - 3334

****

## Agente – 17



## Gerente – 1

****