Sistema Administrativo Antifome

1. Minimundo

O sistema a ser desenvolvido tem como objetivo:

- Dinamizar o processo de solicitação e efetivação de pedidos, assim, interligar clientes e empresas. Para isso a empresa precisa oferecer um produto relacionado a alimentação, podendo ser:
    - Lanchonetes;
    - Restaurantes;
    - Pizzarias;
    - Cafeterias;
    - Padarias;
    - Outros.
- Dar previsibilidade ao cliente, mantendo o máximo de interatividade.

Para que tudo isso ocorra é necessário manter os dados de

- Clientes;
- Empresas;
- Endereços;
- Produtos oferecidos pelas empresas;
- Transação de compra;
- Funcionários para caso haja algum problema;
- Avaliações dos cliente;
- Mensalidade de utilização do sistema por parte das empresas;
- Problemas ocorridos;
- Login de cada usuário do sistema
- Registro de cada acesso feito.

Um cliente é identificado pela seu nome, data de nascimento, data de sua entrada no sistema, cpf, idade e celular. Um cliente pode comprar vários produtos de uma empresa.

Uma empresa é mantida pelo seu cnpj, razão social, data de sua entrada no sistema e nome fantasia. Uma empresa empresa pode vender vários produtos. Cada empresa para uma mensalidade referente ao mês de uso.

Um endereço precisa ter rua, número, cidade, bairro, estado, complemento e cep. Entidades, como: empresas, clientes, funcionários possuem um endereço. E cada um desses endereços podem residir várias entidades.

O produto é inserido no sistema pela empresa que o vende. Para isso, é necessário armazenar o nome, a descrição, o valor e a categoria do produto. Uma empresa pode vender vários produtos. Os clientes podem selecionar de pelo menos um a vários produtos.

Os clientes quando efetuarem uma compra, é gerado uma data, hora, status e decidem se vão pegar o produto no local ou se disponível, será entregue pelo estabelecimento.

Um funcionário é identificado pelo seu nome, data de nascimento, cpf, idade, data de entrada no sistema, função e salário. Um funcionário pode resolver diversos problemas de várias compras.

Uma avaliação pode ser dada ou não por um cliente referente a cada compra que fizer, assim, é necessário manter um comentário e a classificação que o cliente atribuir. Uma compra só pode ser avaliada uma vez e uma compra só é avaliada por um cliente.

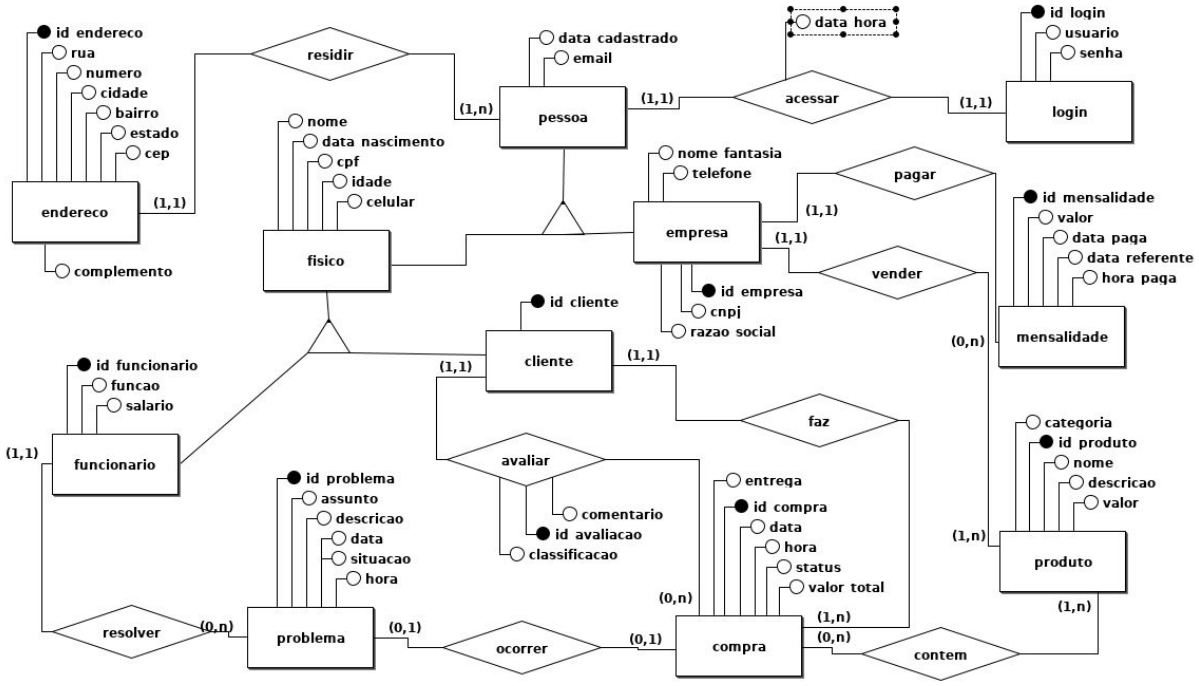
A cada mês, a empresa que utiliza o sistema paga um valor de mensalidade. É mantido junto com o valor, a data e hora que foi realizado o pagamento e o mês de referência do pagamento.

Um problema pode acontecer com um cliente ao efetuar uma compra. A cada registro de um problema é mantido o assunto, descrição, data e hora da abertura do problema e a atual situação.

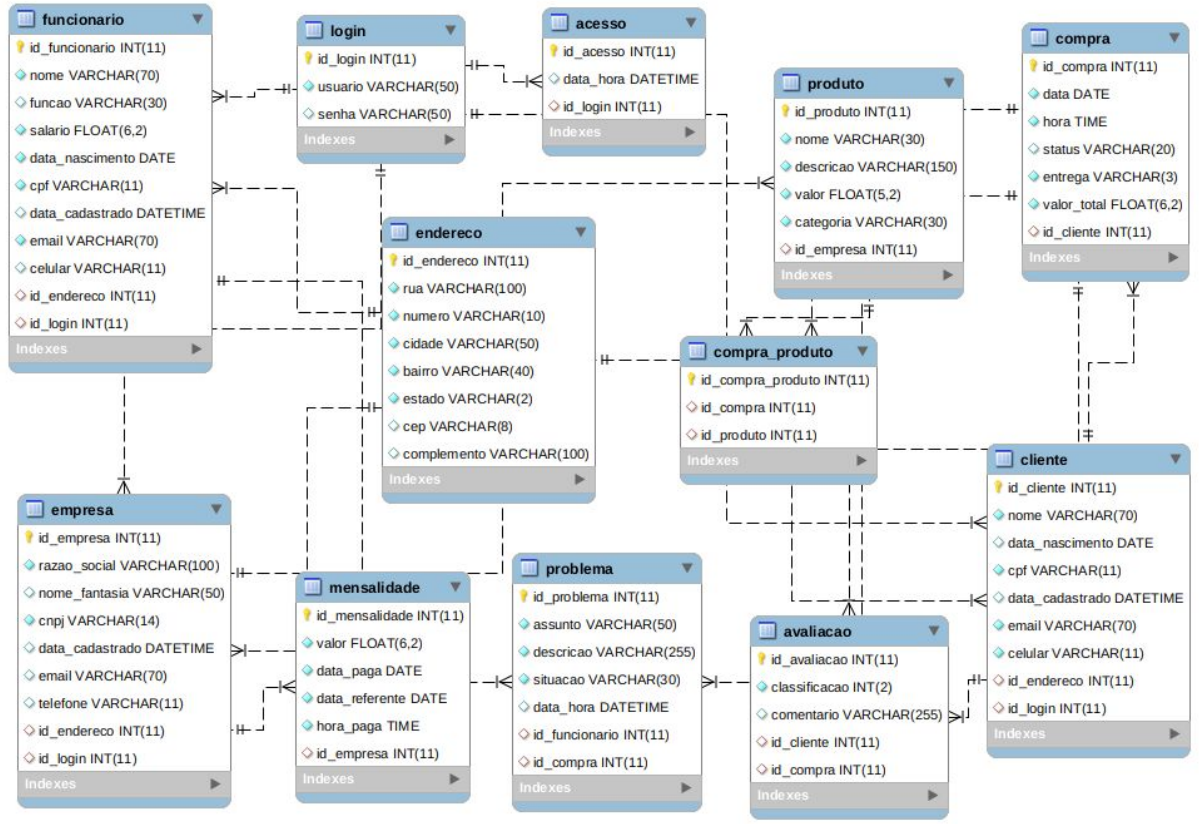
O login é mantido pelo usuário e senha. Um cliente, funcionário e um empresa possuem apenas um login cada um para acessar o sistema.

Cada acesso de um cliente, funcionário e um empresa no sistema tem que ser gravado a hora e data.

2. Modelo Conceitual



3. Modelo Lógico



4. DDL

```
create database if not exists antifome;
use antifome;

CREATE TABLE endereco (
	id_endereco int not null auto_increment PRIMARY KEY,
	rua varchar(100) not null,
	numero varchar(10) not null,
	cidade varchar(50) not null,
	bairro varchar(40) not null,
	estado varchar(2) not null,
	cep varchar(8),
	complemento varchar(100)
);

CREATE TABLE login (
	id_login int not null auto_increment PRIMARY KEY,
	usuario varchar(50) not null,
	senha varchar(50),
	index dx_usuario(usuario)
);

CREATE TABLE empresa (
	id_empresa int not null auto_increment PRIMARY KEY,
	razao_social varchar(100) not null,
	nome_fantasia varchar(50),
	cnpj varchar(14) not null,
	data_cadastrado datetime default current_timestamp,
	email varchar(70),
	telefone varchar(11),
	id_endereco int,
	id_login int,
	index dx_cnpj(cnpj),
	FOREIGN KEY (id_endereco) REFERENCES endereco(id_endereco) on
delete cascade on update cascade,
	FOREIGN KEY (id_login) REFERENCES login(id_login) on delete cascade
on update cascade
);

CREATE TABLE funcionario (
	id_funcionario int not null auto_increment PRIMARY KEY,
	nome varchar(70) not null,
```

```
	funcao varchar(30),
	salario float(6,2) not null,
	data_nascimento date not null,
	cpf varchar(11) not null,
	data_cadastrado datetime default current_timestamp,
	email varchar(70) not null,
	celular varchar(11),
	id_endereco int,
	id_login int,
	index dx_cpf(cpf),
	FOREIGN KEY (id_endereco)REFERENCES endereco (id_endereco) on delete cascade on update cascade,
	FOREIGN KEY (id_login)REFERENCES login (id_login) on delete cascade on update cascade
);

CREATE TABLE cliente (
	id_cliente int not null auto_increment PRIMARY KEY,
	nome varchar(70) not null,
	data_nascimento date,
	cpf varchar(11) not null,
	data_cadastrado datetime default current_timestamp,
	email varchar(70) not null,
	celular varchar(11) not null,
	id_endereco int,
	id_login int,
	index dx_cpf(cpf),
	FOREIGN KEY (id_endereco) REFERENCES endereco (id_endereco) on delete cascade on update cascade,
	FOREIGN KEY (id_login) REFERENCES login (id_login) on delete cascade on update cascade
);

CREATE TABLE produto (
	id_produto int not null auto_increment PRIMARY KEY,
	nome varchar(30) not null,
	descricao varchar(150) not null,
	valor float(5,2) not null,
	categoria varchar(30) not null,
	id_empresa int,
	index dx_nome(nome),
```

```
	FOREIGN KEY (id_empresa)REFERENCES empresa (id_empresa) on delete cascade on update cascade
);

CREATE TABLE mensalidade (
	id_mensalidade int not null auto_increment PRIMARY KEY,
	valor float(6,2) not null,
	data_paga date not null,
	data_referente date not null,
	hora_paga time not null,
	id_empresa int,
	index dx_data_paga(data_paga),
	FOREIGN KEY (id_empresa)REFERENCES empresa (id_empresa) on delete set null on update cascade
);

CREATE TABLE compra (
	id_compra int not null auto_increment PRIMARY KEY,
	data date not null,
	hora time not null,
	status varchar(20),
	entrega varchar(3) not null,
	valor_total float(6,2) not null,
	id_cliente int,
	FOREIGN KEY (id_cliente)REFERENCES cliente (id_cliente) on delete set null on update cascade
);

create table compra_produto(
   id_compra_produto int not null auto_increment primary key,
   id_compra int,
   id_produto int,
   FOREIGN KEY (id_compra) REFERENCES compra(id_compra) on delete cascade on update cascade,
   FOREIGN KEY (id_produto) REFERENCES produto(id_produto) on delete cascade on update cascade
);

CREATE TABLE problema (
	id_problema int not null auto_increment PRIMARY KEY,
	assunto varchar(50) not null,
	descricao varchar(255) not null,
```

```
	situacao varchar(30) not null,
	data_hora datetime default current_timestamp,
	id_funcionario int,
	id_compra int,
	index dx_data_hora(data_hora),
	FOREIGN KEY (id_funcionario) REFERENCES funcionario(id_funcionario) on
delete set null on update cascade,
	FOREIGN KEY (id_compra) REFERENCES compra(id_compra) on delete
cascade on update cascade
);

CREATE TABLE acesso (
	id_acesso int not null auto_increment PRIMARY KEY,
	data_hora datetime default current_timestamp,
	id_login int,
	index dx_data(data_hora),
	FOREIGN KEY (id_login) REFERENCES login (id_login) on delete set null on
update cascade
);

CREATE TABLE avaliacao (
	id_avaliacao int not null auto_increment PRIMARY KEY,
	classificacao int(2) not null,
	comentario varchar(255),
	id_cliente int,
	id_compra int,
	FOREIGN KEY (id_cliente) REFERENCES cliente (id_cliente) on delete
cascade on update cascade,
	FOREIGN KEY (id_compra) REFERENCES compra (id_compra) on delete
cascade on update cascade
);
```

5. DML

5.1 Insert

```
use antifome;
/* Endereço */
insert into endereco(rua,numero,cidade,bairro,estado,cep)
values ('Avenida nao te interessa', '500', 'Campos dos Goytacazes', 'Parque Aurora',
'RJ', '28083200');
insert into endereco(rua,numero,cidade,bairro,estado,cep)
```

```
values ('Rua Barão da Lagoa Dourada', '340', 'Campos dos Goytacazes', 'Parque
Leopoldina', 'RJ', '28083700');
insert into endereco(rua,numero,cidade,bairro,estado,cep)
values ('Rua Liceu', '340', 'Campos dos Goytacazes', 'Parque Presidente', 'RJ',
'28083800');
insert into endereco(rua,numero,cidade,bairro,estado,cep)
values ('Rua Caramba', '340', 'Campos dos Goytacazes', 'Parque Porta', 'RJ',
'28083110');
insert into endereco(rua,numero,cidade,bairro,estado,cep)
values ('Rua Sala', '340', 'Campos dos Goytacazes', 'Parque Armario', 'RJ',
'28083200');

/* Login */
insert into login(usuario,senha)
values ('robertinho21','senha');
insert into login(usuario,senha)
values ('carolina','senha');
insert into login(usuario,senha)
values ('roteador','senha');
insert into login(usuario,senha)
values ('eu','senha');
insert into login(usuario,senha)
values ('marcao','senha');
insert into login(usuario,senha)
values ('funcionario1','senha');
insert into login(usuario,senha)
values ('funcionario2','senha');
insert into login(usuario,senha)
values ('empresa1','senha');
insert into login(usuario,senha)
values ('empresa2','senha');

/* Cliente */
insert into cliente(nome,data_nascimento,cpf,email,celular,id_endereco,id_login)
values ('Roberto Almeida', '1998-01-25', '15721558798',
'robertinhodavila@gmail.com', '22997852505', 1,1);
insert into cliente(nome,data_nascimento,cpf,email,celular,id_endereco,id_login)
values ('Maria Doida', '2000-01-25', '25636525633', 'mariadatia@gmail.com',
'22955455415', 4,2);
insert into cliente(nome,data_nascimento,cpf,email,celular,id_endereco,id_login)
values ('Roberta', '2001-01-25', '12515889322', 'robertapop@gmail.com',
'22955455415', 2,3);
```

```
insert into cliente(nome,data_nascimento,cpf,email,celular,id_endereco,id_login)
values ('Carla', '1996-01-25', '20354892015', 'carla@gmail.com', '22955455415',
3,4);
insert into cliente(nome,data_nascimento,cpf,email,celular,id_endereco,id_login)
values ('Lucas', '2002-01-25', '02158763152', 'luquinha@gmail.com', '22955455415',
1,5);

/* Funcionario */

insert into
funcionario(nome,funcao,salario,data_nascimento,cpf,email,celular,id_endereco,id_l
ogin)
values ('Carlos', 'Atendente', 1300, '1972-10-18', '15725152154', 'carlos@gmail.com',
'22987725422', 4,6);
insert into
funcionario(nome,funcao,salario,data_nascimento,cpf,email,celular,id_endereco,id_l
ogin)
values ('Mara', 'Atendente', 3500.5, '1972-10-18', '15725155554', 'mara@gmail.com',
'22987725422', 5,7);

/* Empresa */
insert into
empresa(razao_social,nome_fantasia,cnpj,email,telefone,id_endereco,id_login)
values ('Doces', 'Adoçando a Vida', '15333372515215', 'adocando@gmail.com',
'2227314547
', 3,8);
insert into
empresa(razao_social,nome_fantasia,cnpj,email,telefone,id_endereco,id_login)
values ('Salgados para todos', 'Salgado do Povo', '25469805423152',
'salgadando@gmail.com', '2227282116
', 5,9);

/* Compra */
insert into compra(data,hora,status,entrega,valor_total,id_cliente)
values ('2019-6-15', '21:21:01', 'finalizada','nao', 56.2,1);
insert into compra(data,hora,status,entrega,valor_total,id_cliente)
values ('2019-6-01', '10:54:01', 'andamento','nao', 76.8,2);


/* Produto */
insert into produto(nome,descricao,valor,categoria,id_empresa)
values ('Chocolate', 'Barra de Chocolate 500g', 8.5, 'Doce', 1);
```

```
insert into produto(nome,descricao,valor,categoria,id_empresa)
values ('Caramelo', 'Bisnaga de Caramelo 50g', 1.5, 'Doce', 1);
insert into produto(nome,descricao,valor,categoria,id_empresa)
values ('Enroladinho', 'Presunto e Queijo', 3.5, 'Salgado', 2);
insert into produto(nome,descricao,valor,categoria,id_empresa)
values ('Enroladinho', 'Frengo', 3.5, 'Salgado', 2);

/* Mapeando compra e produtos */
insert into compra_produto(id_compra,id_produto)
   values (1,1);
insert into compra_produto(id_compra,id_produto)
   values (1,2);
insert into compra_produto(id_compra,id_produto)
   values (2,3);
insert into compra_produto(id_compra,id_produto)
   values (2,4);

/* Mensalidade */
insert into mensalidade(valor,data_paga,data_referente,hora_paga,id_empresa)
   values (50.5, '2019-7-2', '2019-6-15', '21:21:01',1);
insert into mensalidade(valor,data_paga,data_referente,hora_paga,id_empresa)
   values (50.5, '2019-7-2', '2019-6-15', '21:21:01',2);


/* Problema */
insert into problema(assunto, descricao, situacao, id_funcionario, id_compra)
   values ('Compra Invalida', 'Minha compra foi nagada e invalida', 'Resolvido',1, 1);
insert into problema(assunto, descricao, situacao, id_funcionario, id_compra)
   values ('Compra Invalida', 'Minha compra foi nagada e invalida', 'Não esolvido',2,
2);

/* Avaliacao */
insert into avaliacao (classificacao, comentario, id_cliente, id_compra)
   values(5, 'Chegou Rápido, adorei',  1,1);
insert into avaliacao (classificacao, comentario, id_cliente, id_compra)
   values(1, 'Demorou muito, Odiei',  2,2);

/* Acesso */
insert into acesso (id_login)
   values(1);
insert into acesso (id_login)
   values(2);
```

5.2. Case When

- Aumentar o salário dos funcionários que ganham abaixo de mil e quinhentos reais em 10%.
- Aumentar o salário dos funcionários que ganham a partir de mil e quinhentos reais em 20%.

```
update funcionario set salario =
   case when salario < 1500
        then salario + salario * 0.1
        else salario * 1.1
   end;
```

5.3. Relatórios

Os relatórios foram escolhidos considerando sua necessidade e agregação para o negócio.

1. Busca o nome dos funcionários que não resolveram nenhum problema:
    - select nome from funcionario f left join problema p on f.id_funcionario = p.id_funcionario where f.id_funcionario not in (select id_funcionario from problema);

2. Lista dos clientes que mais compraram, mostrando apenas nomes e a quantidade de vezes que compraram:
    - select c.nome, count(*) quantidade from cliente c join compra cp on c.id_cliente = cp.id_cliente group by c.id_cliente order by quantidade desc;

3. A soma do valor de todas as compras realizadas no mês de junho:
    - select sum(valor_total) valor_todas_as_compras from compra where data between '2019-06-01' and '2019-06-30';

4. Lista com nome dos clientes e a respectiva quantidade de produtos que já comprou em ordem decrescente:
    - select cli.nome, count(*) produtos from compra c join compra_produto cp on c.id_compra = cp.id_compra join produto p on cp.id_produto = p.id_produto join cliente cli on cli.id_cliente = c.id_cliente group by cli.id_cliente order by produtos desc;

5. A soma do valor de todas as mensalidades pagas no ano de 2019:

- select sum(valor) receita_2019 from mensalidade where data_paga like '2019%';

6. DCL

No controle de acesso foi pensado em 4 usuários:

- 1 acesso total:
  create user 'gerencia'@'localhost' identified by 'gerencia';
  grant all on *.* to 'gerencia'@'localhost';

- 1 acesso de visualização (select):
  create user 'olhada'@'localhost' identified by 'olhada';
  grant select on antifome.* to 'olhada'@'localhost';

- 1 algumas tabelas. (Apenas Cliente, Empresa e Funcionário):
  create user pessoas@'localhost' identified by pessoas;
  grant all on antifome.empresa to pessoas@'localhost';
  grant all on antifome.cliente to pessoas@'localhost';
  grant all on antifome.funcionario to pessoas@'localhost';

- 1 algumas colunas de uma tabela:
  create user vercliente@'localhost' identified by vercliente;
  grant select(nome), select(data_nascimento) on antifome.cliente to vercliente@'localhost';

7. Backup

Foi escolhido o sistema operacional linux para hospedar o servidor de banco de dados, o script contém informações importantes e instruções. O backup vai ser feito diariamente, antes de ser feito o backup será feito uma verificação para ver se não há tabelas corrompidas e uma otimização para manter o máximo de dinâmica do banco de dados MySQL.

```
#!/bin/sh
# bkp_antifome.sh

# permissao para execucao: chmod +x bkp_antifome.sh
# no crontab: */1 * * * * /bin/sh /home/usuario/bkp_antifome.sh

DATAHORA=`/bin/date +%d-%m-%Y-%H-%M`
NOME="/home/usuario/bkp_antifome-$DATAHORA.sql"
ANALYZE="/home/usuario/check-analyze"
OTIMIZATION="/home/usuario/check-otimization"

# variaveis do MySQL
# HOST="localhost"
USER="gerencia"
```

```
PASSWORD="gerencia"
DATABASE="antifome"

mysqldump -u $USER -p$PASSWORD $DATABASE > $NOME
mysqlcheck -a $DATABASE -u $USER -p$PASSWORD > $ANALYZE
mysqlcheck -o $DATABASE -u $USER -p$PASSWORD > $OTIMIZATION
```

8. View

Lista contendo os nomes dos clientes e a quantidade total de produtos que já adquiriram em todas as compras.

```
create view v_cliente_produto as
   select cli.nome, count(*) produtos
   from compra c
   join compra_produto cp on c.id_compra = cp.id_compra
   join produto p on cp.id_produto = p.id_produto
   join cliente cli on cli.id_cliente = c.id_cliente
   group by cli.id_cliente
   order by produtos desc;
```

9. Programação

9.1 Função

Retornar a quantidade de empresas cadastradas em um determinado mês (Intervalo restrito em 1 ano).

```
create function empresas_cadastradas(mes varchar(2)) returns integer
begin
        declare quantidade integer;
        declare ano varchar(4);

        if(mes > date_format(now(),'%m')) then
         set ano = date_format(date_sub(now(), interval 1 year),'%Y');
   else
         set ano = date_format(now(),'%Y');
   end if;

        select count(*) into quantidade from empresa where data_cadastrado like
(concat(ano,'-',mes,'%'));

        return quantidade;
end;
```

9.2 Procedimento

Quando a aplicação é fechada, o status de todos as compras tem que ser definido para finalizado e mostrar na tela quantas compras ficaram com status diferente de finalizado.

```
create procedure finalizar_todos()
begin
        declare tamanho integer;
        declare i integer;
        declare quantidade integer;
        declare statu varchar(20);
   set quantidade = 0;
        set i = 0;

        select max(id_compra) into tamanho from compra;
        while i <= tamanho do
         select status into statu from compra where id_compra = i;
         if(statu <> 'Finalizada' ) then
                 set quantidade = quantidade + 1;
        end if;
        set i = i + 1;
        end while;
        select concat('A quantidade de compras diferente de Finalizada: ',
quantidade);
   update compra set status = 'Finalizada';
end;
```

9.3 Trigger

Toda vez que o funcionario resolve um problema é adicionado 20 reais ao seu salario.

```
create trigger adiciona_funcionario after insert
on problema
for each row
begin
   update funcionario set salario = salario + 20 where id_funcionario =
new.id_funcionario;
end
```

10. Outros

Para mais informações consulte os arquivos em anexo.