

ANÁLISE E DESENVOLVIMENTO DE SOFTWARE

TÓPICOS INTEGRADORES II

Documentação de Projeto de Software

Versão 1.3

Autor(a): <Vinícius Marinho Pergentino de Santana>

<https://github.com/vinydev/T-picos-Integradores-II>

2020

Controle de Versão do Documento

|  |  |  |
| --- | --- | --- |
| Data | Versão | Descrição |
| 24/05/2020 | 1.3 | Projeto até as Telas da classe View |

Sumário

[Introdução ao Documento](#_heading=h.gjdgxs) **4**

[Área de negócio do Sistema](#_heading=h.30j0zll) 4

[Principais funcionalidades](#_heading=h.1fob9te) 4

[Método de trabalho](#_heading=h.3znysh7) 4

[Modelo de Dados](#_heading=h.2et92p0) **4**

[Modelo de Visão](#_heading=h.tyjcwt) 4

[Modelo Conceitual](#_heading=h.3dy6vkm) 4

[Modelo Lógico](#_heading=h.1t3h5sf) 4

[Dicionário de Dados](#_heading=h.4d34og8) 5

[Modelo Físico](#_heading=h.2s8eyo1) 5

[Deverão ser apresentado os scripts DDL do banco de dados.](#_heading=h.r3csljbljrc) 5

[Comandos Insert para alimentar cada tabela com 5 registros.](#_heading=h.17dp8vu) 5

[Os relatórios que foram solicitados e os respectivos comandos Select.](#_heading=h.kins0oxhfvyz) 5

[Análise e Design](#_heading=h.3rdcrjn) **5**

[Diagrama de Classes](#_heading=h.26in1rg) 5

[Arquitetura do Software](#_heading=h.lnxbz9) **5**

[Padrão de projeto](#_heading=h.35nkun2) 5

[Protótipo](#_heading=h.1ksv4uv) 6

[Telas e classes da camada View](#_heading=h.tyyx38t8ccqk) 6

[Classes da camada Control](#_heading=h.qmd0ffvj6asj) 6

[Classes da camada Model](#_heading=h.bowgpcfk1s0b) 6

# Introdução ao Documento

## Área de negócio do Sistema

O sistema vai auxiliar a equipe do centro de distribuição do armazém a controlar a entrada e saída de produtos e amenizar a espera caso não tenha estoque, avisando com antecedência a baixa quantidade de produto.

## Principais funcionalidades

RF01 – Antecipar aviso para reabastecimento de estoque

RF02 – Autorizar saída do produto do pedido do cliente

RF03 – Registrar a quantidade de entrada do produto

## Método de trabalho

Utilizarei o SGBD maria db, utilizarei a linguagem de programação java com a ide NetBeans, utilizarei o paradigma de orientação a objetos. Farei a interface gráfica e também a parte de dados.

# Modelo de Dados

## Modelo de Visão

O sistema fará o controle de entrada e saída dos produtos, avisará quando o produto estiver de estoque baixo, avisará quando não tiver quantidade disponível para retirada em estoque.

Quando o estoque estiver baixo ele gerará o alerta e informará o produto que está faltando junto com os dados do fornecedor para que entre em contato.

O sistema irá mapear quais produtos ficam mais tempo em estoque e quais produtos ficam menos tempo e irá mostrar decisões para reduzir o custo.

Existirão dois tipos de Cliente, o cliente pessoa física e o cliente pessoa jurídica.

No cliente pessoa física terão os seguintes campos: código, nome, telefone, endereço (cep, logradouro, numero, bairro, cidade, uf), email, rg e cpf.

No cliente pessoa jurídica terão os seguintes campos: código, nome, telefone, endereço (cep, logradouro, numero, bairro, cidade, uf), email e cnpj.

O cliente realiza vários pedidos e nesse pedido contém produtos e a quantidade do produto, data do pedido e tem também o código do pedido. Mostra se o pedido foi autorizado.

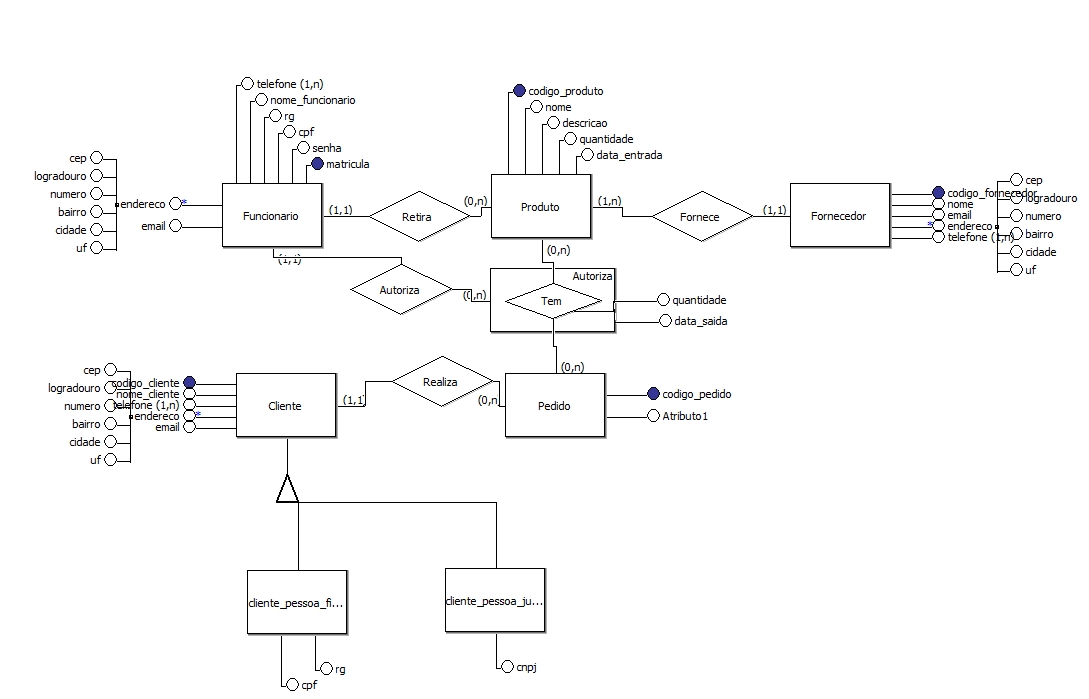
O produto será cadastrado com nome, descrição e o sistema irá controlar a quantidade do produto, entrada e saída, quem fará o controle dos produtos será o funcionário logado no sistema que será previamente cadastrado e fará login com sua matricula e uma senha.

O funcionário deverá informar no cadastro os seguintes campos: Nome, telefone, endereço (cep, logradouro, numero, bairro, cidade, uf), email, rg, cpf e será gerada uma matrícula, senha e código. A senha poderá ser alterada após o cadastro do funcionário.

Quem fornece os produtos é o fornecedor, cada produto só tem um fornecedor, mas o mesmo fornecedor pode fornecer vários produtos.

O fornecedor terá os seguintes atributos: código\_fornecedor, nome, email, endereço (cep, logradouro, numero, bairro, cidade, uf), telefone.

## Modelo Conceitual



## Modelo Lógico

Tb\_Cliente (codigo\_cliente, nome\_cliente, email, cep, logradouro, numero, bairro, cidade, uf, cpf\_cnpj, rg, tipo\_cliente)

codigo\_cliente primary key

Tb\_telefone\_cliente (telefone\_cliente, codigo\_cliente)

telefone\_cliente, codigo\_cliente primary key

codigo\_cliente references tb\_cliente (codigo\_cliente)

Tb\_fornecedor (codigo\_fornecedor, nome, email, cep, logradouro, numero, bairro, cidade, uf)

codigo\_fornecedor primary key

Tb\_telefone\_fornecedor (telefone\_fornecedor, código\_fornecedor)

telefone\_fornecedor, codigo\_fornecedor primary key

codigo\_fornecedor references tb\_fornecedor(codigo\_fornecedor)

Tb\_funcionario (matricula, nome\_funcionario, email, rg, cpf, senha, cep, logradouro, numero, bairro, cidade, uf)

Matricula primary key

Tb\_telefone\_funcionario (telefone\_funcionario, matricula)

telefone\_funcionario, matricula primary key

matricula references tb\_funcionario(matricula)

Tb\_produto (codigo\_produto, nome, descrição, quantidade, data\_entrada, matricula, codigo\_fornecedor)

codigo\_produto primary key

matricula references tb\_funcionario(matricula)

codigo\_fornecedor references tb\_fornecedor(codigo\_fornecedor)

Tb\_pedido (codigo\_pedido, codigo\_cliente, data\_pedido)

codigo\_pedido primary key

codigo\_cliente references tb\_cliente(codigo\_cliente)

Tb\_pedido\_produto (codigo\_pedido, codigo\_produto, matricula, quantidade, data\_saida)

codigo\_pedido, código\_produto primary key

codigo\_pedido references tb\_pedido(codigo\_pedido)

codigo\_produto references tb\_produto(codigo\_produto)

Matricula references tb\_funcionario(matricula)

## Dicionário de Dados

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | Tabela | Objetivo |  |  | | tb\_cliente | Manter Clientes |  |  | |  |  |  |  | | Campo | Tipo | Decrição | Observações e regras | | codigo\_cliente | integer | Identificador do Cliente | not null; auto\_increment; primary key; unique | | nome\_cliente | varchar(60) | Nome do cliente | not null | | email | varchar(50) | Email do cliente | not null; unique | | cep | char(8) | Cep da residência do Cliente | not null | | logradouro | varchar(50) | Logradouro da residência do Cliente | not null | | numero | varchar(8) | Número da residência do Cliente | not null | | bairro | varchar(40) | Bairro onde o cliente mora | not null | | cidade | varchar(40) | Cidade onde o cliente mora | not null | | uf | char(2) | Sigla do estado onde o cliente reside | not null | | cpf\_cnpj | vatchar(14) | Cpf ou cnpj do cliente | not null; unique | | rg | char(7) | Rg do cliente caso seja pessoa física | unique | | tipo\_cliente | char(2) | Se o cliente é pessoa física ou pessoa jurídica | not null; 'PF' ou 'PJ' | |  |  |  |  | | Tabela | Objetivo |  |  | | tb\_telefone\_cliente | Manter o telefone do Cliente |  |  | |  |  |  |  | | Campo | Tipo | Descrição | Observações e regras | | telefone\_cliente | varchar(14) | Número do telefone do cliente | not null; primary key; unique | | codigo\_cliente | integer | Identificador do cliente | not null; primary key; foreign key(tb\_cliente) | |  |  |  |  | | Tabela | Objetivo |  |  | | tb\_fornecedor | Manter Fornecedor |  |  | |  |  |  |  | | Campo | Tipo | Descricao | Observações e regras | | codigo\_fornecedor | integer | Identificador do fornecedor | not null; auto\_increment; primary key; unique | | nome | varchar(60) | Nome do fornecedor | not null | | email | varchar(50) | Nome do fornecedor | not null; unique | | cep | char(8) | Cep do local do fornecedor | not null | | logradouro | varchar(50) | Logradouro do local do fornecedor | not null | | numero | varchar(8) | Número do local onde o fornecedor se encontra | not null | | bairro | varchar(40) | Bairro do local onde se encontra o fornecedor | not null | | cidade | varchar(40) | Cidade onde se encontra o fornecedor | not null | | uf | char(2) | Estado onde se localiza o fornecedor | not null | |  |  |  |  | | Tabela | Objetivo |  |  | | tb\_telefone\_fornecedor | Manter telefone do fornecedor |  |  | |  |  |  |  | | Campo | Tipo | Descrição | Observações e regras | | Telefone\_fornecedor | varchar(14) | Número do telefone do fornecedor | not null; primary key; unique | | codigo\_fornecedor | integer | Identificador do fornecedor | not null; primary key; foreign key(tb\_fornecedor) | |  |  |  |  | | Tabela | Objetivo |  |  | | tb\_funcionario | Manter Funcionarios |  |  | |  |  |  |  | | Campo | Tipo | Descrição | Observações e regras | | matricula | integer | Identificador do funcionário | not null; auto\_increment; primary key; unique | | nome\_funcionario | varchar(60) | Nome do funcionário | not null | | email | varchar(50) | Email do funcionário | not null; unique | | rg | char(7) | Rg do funcionário | not null; unique | | cpf | char(11) | Cpf do funcionário | not null; unique | | senha | varchar(40) | Senha do funcionário para entrar no sistema | not null | | cep | char(8) | Cep de onde mora o funcionário | not null | | logradouro | varchar(50) | Logradouro da residência do funcionário | not null | | numero | varchar(8) | Numero da residência do funcionário | not null | | bairro | varchar(40) | Bairro onde o funcionário mora | not null | | cidade | varchar(40) | Cidade onde o funcionário mora | not null | | uf | char(2) | Sigla do estado que o funcionário reside | not null | |  |  |  |  | | Tabela | Objetivo |  |  | | tb\_telefone\_funcionario | Manter telefone dos funcionarios |  |  | |  |  |  |  | | Campo | Tipo | Descrição | Observações e regras | | telefone\_funcionario | varchar(14) | Número de telefone do funcionário | not null; unique; primary key | | matricula | integer | Identificador do funcionário | not null; primary key; foreign key(tb\_funcionario) | |  |  |  |  | | Tabela | Objetivo |  |  | | tb\_produto | Manter Produtos |  |  | |  |  |  |  | | Campo | Tipo | Descrição | Observações e regras | | codigo\_produto | integer | Identificador do produto | not null; auto\_increment; primary key; unique | | nome | varchar(60) | nome do produto | not null | | descricao | varchar(300) | descrição do produto | not null | | quantidade | integer | Quantidade do produto em estoque | not null; >=0 | | data\_entrada | date | Data de chegada do produto | not null | | matricula | integer | Identificador do funcionário | not null; foreign key(tb\_funcionario) | | codigo\_fornecedor | integer | Identificador do fornecedor | not null; foreign key(tb\_fornecedor) | |  |  |  |  | | Tabela | Objetivo |  |  | | tb\_pedido | Manter pedidos |  |  | |  |  |  |  | | Campo | Tipo | Descrição | Observações e regras | | codigo\_pedido | integer | Identificador do pedido | not null; auto\_increment; primary key; unique | | codigo\_cliente | integer | Identificador do cliente | not null; foreign key(tb\_cliente) | | data\_pedido | date | Data de realização do pedido | not null | |  |  |  |  | | Tabela | Objetivo |  |  | | tb\_pedido\_produto | Manter o vínculo de produtos ao pedido |  |  | |  |  |  |  | | Campo | Tipo | Descrição | Observações e regras | | codigo\_pedido | integer | Identificador do pedido | not null; primary key; foreign key(tb\_pedido) | | codigo\_produto | integer | Identificador do produto | not null; primary key; foreign key(tb\_produto) | | matricula | integer | Identificador do funcionário | not null; foreign key(tb\_funcionario) | | quantidade | int | Quantidade de produto no pedido | not null; > 0 | | data\_saida | date | Data que o produto foi autorizado para saída | not null | |

## Modelo Físico

### Deverão ser apresentado os scripts DDL do banco de dados.

DROP DATABASE IF EXISTS centro\_de\_distribuicao;

CREATE DATABASE centro\_de\_distribuicao;

use centro\_de\_distribuicao;

CREATE TABLE tb\_cliente (

codigo\_cliente integer not null AUTO\_INCREMENT primary key UNIQUE,

nome\_cliente varchar(60) not null,

email varchar(50) not null UNIQUE,

cep char(8) not null,

logradouro varchar(50) not null,

numero varchar(8) not null,

bairro varchar(40) not null,

cidade varchar(40) not null,

uf char(2) not null,

cpf\_cnpj varchar(14) UNIQUE,

rg char(7) UNIQUE,

tipo\_cliente char(2) not null,

CONSTRAINT CHK\_tipo\_cliente CHECK (tipo\_cliente = 'PF' or tipo\_cliente ='PJ')

);

CREATE TABLE tb\_telefone\_cliente (

telefone\_cliente varchar(14) not null unique,

codigo\_cliente integer not null,

constraint fk\_cliente\_telefone foreign key (codigo\_cliente) references tb\_cliente(codigo\_cliente) on delete cascade on update cascade,

constraint pk\_tb\_telefone\_cliente primary key (telefone\_cliente, codigo\_cliente)

);

CREATE TABLE tb\_fornecedor (

codigo\_fornecedor integer not null AUTO\_INCREMENT primary key UNIQUE,

nome varchar(60) not null,

email varchar(50) not null UNIQUE,

cep char(8) not null,

logradouro varchar(50) not null,

numero varchar(8) not null,

bairro varchar(40) not null,

cidade varchar(40) not null,

uf char(2) not null

);

CREATE TABLE tb\_telefone\_fornecedor (

telefone\_fornecedor varchar(14) not null UNIQUE,

codigo\_fornecedor integer not null,

constraint fk\_fornecedor\_telefone foreign key (codigo\_fornecedor) references tb\_fornecedor(codigo\_fornecedor) on delete cascade on update cascade,

constraint pk\_tb\_telefone\_fornecedor primary key (telefone\_fornecedor, codigo\_fornecedor)

);

CREATE TABLE tb\_funcionario (

matricula integer not null AUTO\_INCREMENT primary key UNIQUE,

nome\_funcionario varchar(60) not null,

email varchar(50) not null UNIQUE,

rg char(7) not null UNIQUE,

cpf char(11) not null UNIQUE,

senha varchar(40) not null,

cep char(8) not null,

logradouro varchar(50) not null,

numero varchar(8) not null,

bairro varchar(40) not null,

cidade varchar(40) not null,

uf char(2) not null

);

CREATE TABLE tb\_telefone\_funcionario (

telefone\_funcionario varchar(14) not null UNIQUE,

matricula integer not null,

constraint fk\_funcionario\_telefone foreign key (matricula) references tb\_funcionario(matricula) on delete cascade on update cascade,

constraint pk\_tb\_telefone\_funcionario primary key (telefone\_funcionario, matricula)

);

CREATE TABLE tb\_produto (

codigo\_produto integer not null AUTO\_INCREMENT primary key UNIQUE,

nome varchar(60) not null,

descricao varchar(300) not null,

quantidade integer not null,

data\_entrada date not null,

matricula integer not null,

codigo\_fornecedor integer not null,

constraint fk\_funcionario\_produto foreign key (matricula) references tb\_funcionario(matricula) on delete cascade on update cascade,

constraint fk\_fornecedor\_produto foreign key (codigo\_fornecedor) references tb\_fornecedor(codigo\_fornecedor) on delete cascade on update cascade,

CONSTRAINT CHK\_quantidade CHECK (quantidade >= 0)

);

CREATE TABLE tb\_pedido (

codigo\_pedido integer not null AUTO\_INCREMENT primary key UNIQUE,

codigo\_cliente integer not null,

data\_pedido date not null,

constraint fk\_cliente\_pedido foreign key (codigo\_cliente) references tb\_cliente(codigo\_cliente) on delete cascade on update cascade

);

CREATE TABLE tb\_pedido\_produto (

codigo\_pedido integer not null,

codigo\_produto integer not null,

matricula integer not null,

quantidade int not null,

data\_saida date not null,

constraint fk\_pedido\_pedido\_produto foreign key (codigo\_pedido) references tb\_pedido(codigo\_pedido) on delete cascade on update cascade,

constraint fk\_produto\_pedido\_produto foreign key (codigo\_produto) references tb\_produto(codigo\_produto) on delete cascade on update cascade,

constraint pk\_tb\_pedido\_produto primary key (codigo\_pedido, codigo\_produto),

constraint CHK\_quantidade\_item CHECK (quantidade > 0)

);

### Comandos Insert para alimentar cada tabela com 5 registros.

/\*Inserindo dados na tabela tb\_cliente\*/

INSERT INTO tb\_cliente (nome\_cliente, email, cep, logradouro, numero, bairro, cidade, uf, cpf\_cnpj, rg, tipo\_cliente) VALUES ('Isaac Rodrigues Ferreira', 'isaacrf@gmail.com', '04546001', 'RUA CASA DO ATOR', '275', 'VILA OLÍMPIA', 'SÃO PAULO', 'SP','89087643289', '3323324', 'PF');

INSERT INTO tb\_cliente (nome\_cliente, email, cep, logradouro, numero, bairro, cidade, uf, cpf\_cnpj, rg, tipo\_cliente) VALUES ('Accenture', 'accenture@gmail.com', '50030230', 'Edifício Vasco Rodrigues Rua Cais Do Apolo', '222', 'Recife Antigo', 'RECIFE', 'PE', '96534094002797', null, 'PJ');

INSERT INTO tb\_cliente (nome\_cliente, email, cep, logradouro, numero, bairro, cidade, uf, cpf\_cnpj, rg, tipo\_cliente) VALUES ('Avanade', 'avanade@gmail.com', '50030220', 'Rua, Cais do Apolo', '222', 'Recife', 'RECIFE', 'PE', '40499760001000', null, 'PJ');

INSERT INTO tb\_cliente (nome\_cliente, email, cep, logradouro, numero, bairro, cidade, uf, cpf\_cnpj, rg, tipo\_cliente) VALUES ('Pietro Martins da Silva Gomes', 'pietromsg@gmail.com', '04576020', 'RUA GEORGE HOM', '230', 'BROOKLIN NOVO', 'SÃO PAULO', 'SP', '45398763521', '3231098', 'PF');

INSERT INTO tb\_cliente (nome\_cliente, email, cep, logradouro, numero, bairro, cidade, uf, cpf\_cnpj, rg, tipo\_cliente) VALUES ('Helena Kettelin Dias Porto De Oliveira','helenakdpo@gmail.com', '13420640', 'AVENIDA ANTONIA PAZZINATO STURION', '1221', 'JD PETROPOLIS- PIRACICABA', 'SÃO PAULO', 'SP', '78954389787', '0929332', 'PF');

/\*Inserindo dados na tabela tb\_telefone\_cliente\*/

INSERT INTO tb\_telefone\_cliente (telefone\_cliente, codigo\_cliente) VALUES ('+5511987965467',1);

INSERT INTO tb\_telefone\_cliente (telefone\_cliente, codigo\_cliente) VALUES ('8121236733',2);

INSERT INTO tb\_telefone\_cliente (telefone\_cliente, codigo\_cliente) VALUES ('8121222234',3);

INSERT INTO tb\_telefone\_cliente (telefone\_cliente, codigo\_cliente) VALUES ('+5511998765843',4);

INSERT INTO tb\_telefone\_cliente (telefone\_cliente, codigo\_cliente) VALUES ('+5511998765678',5);

/\*Inserindo dados na tabela tb\_fornecedor\*/

INSERT INTO tb\_fornecedor (nome, email, cep, logradouro, numero, bairro, cidade, uf) VALUES ('Celite', 'celite@gmail.com', '50950000', 'Rua João Lopes', '10', 'Curado', 'Recife', 'PE');

INSERT INTO tb\_fornecedor (nome, email, cep, logradouro, numero, bairro, cidade, uf) VALUES ('Eternit','eternit@gmail.com', '53635745', 'R. Alameda dos Cravos', '20', 'Cruz de Rebouças', 'Igarassu', 'PE');

INSERT INTO tb\_fornecedor (nome, email, cep, logradouro, numero, bairro, cidade, uf) VALUES ('Deca', 'deca@gmail.com', '51180020', 'Rua Madrid', '190', 'Imbiribeira', 'Recife', 'PE');

INSERT INTO tb\_fornecedor (nome, email, cep, logradouro, numero, bairro, cidade, uf) VALUES ('Pial', 'pial@gmail.com', '54340320', 'Rodovia Antiga', '2141', 'Prazeres', 'Jaboatão dos Guararapes', 'PE');

INSERT INTO tb\_fornecedor (nome, email, cep, logradouro, numero, bairro, cidade, uf) VALUES ('Botafogo', 'botafogo@gmail.com', '22290040', 'Av. Venceslau Brás', '72', 'Botafogo', 'Rio de Janeiro', 'RJ');

/\*Inserindo dados na tabela tb\_telefone\_fornecedor\*/

INSERT INTO tb\_telefone\_fornecedor (telefone\_fornecedor, codigo\_fornecedor) VALUES ('8134568790', 1);

INSERT INTO tb\_telefone\_fornecedor (telefone\_fornecedor, codigo\_fornecedor) VALUES ('8132256778', 2);

INSERT INTO tb\_telefone\_fornecedor (telefone\_fornecedor, codigo\_fornecedor) VALUES ('8121223285', 3);

INSERT INTO tb\_telefone\_fornecedor (telefone\_fornecedor, codigo\_fornecedor) VALUES ('8132222732', 4);

INSERT INTO tb\_telefone\_fornecedor (telefone\_fornecedor, codigo\_fornecedor) VALUES ('2122348531', 5);

/\*Inserindo dados na tabela tb\_funcionario\*/

INSERT INTO tb\_funcionario (nome\_funcionario, email, rg, cpf, senha, cep, logradouro, numero, bairro, cidade, uf) VALUES ('Marianna Martins da Silva Gomes', 'mariannamsg@gmail.com', '8796443', '87694488876', 'mari123', '04546001' , 'RUA CASA DO ATOR', '294', 'VILA OLÍMPIA', 'SÃO PAULO', 'SP');

INSERT INTO tb\_funcionario (nome\_funcionario, email, rg, cpf, senha, cep, logradouro, numero, bairro, cidade, uf) VALUES ('Adam Vitor Dias Porto De Oliveira', 'adamvdpo@gmail.com', '1234879', '01293883392', 'adam123', '01333010', 'RUA CINCINATO BRAGA', '306', 'VILA MARIANA', 'SÃO PAULO', 'SP');

INSERT INTO tb\_funcionario (nome\_funcionario, email, rg, cpf, senha, cep, logradouro, numero, bairro, cidade, uf) VALUES ('Melissa da Fonseca Diniz', 'melissafd@gmail.com', '9328732', '98724787324', 'melissa123','04532082' , 'RUA JESUINO ARRUDA', '676', 'ITAIM BIBI ', 'SÃO PAULO', 'SP');

INSERT INTO tb\_funcionario (nome\_funcionario, email, rg, cpf, senha, cep, logradouro, numero, bairro, cidade, uf) VALUES ('Calebe Mendes de Oliveira', 'calebemo@gmail.com', '2138943', '23239108213', 'calebe123', '04544000', 'RUA DR ALCEU DE CAMPOS RODRIGUES', '213', 'ITAIM BIBI', 'SÃO PAULO', 'SP');

INSERT INTO tb\_funcionario (nome\_funcionario, email, rg, cpf, senha, cep, logradouro, numero, bairro, cidade, uf) VALUES ('Isabelly Victoria de Carvalho Moreira', 'isabellyvcm@gmail.com', '9833242', '21332098213', 'isabelly123','04571090' , 'RUA SANSÃO ALVES DOS SANTOS', '373', 'BROOKLIN', 'SÃO PAULO', 'SP');

/\*Inserindo dados na tabela tb\_telefone\_funcionario\*/

INSERT INTO tb\_telefone\_funcionario (telefone\_funcionario, matricula) VALUES ('+5511998343434' , 1);

INSERT INTO tb\_telefone\_funcionario (telefone\_funcionario, matricula) VALUES ('+5511983247923' , 2);

INSERT INTO tb\_telefone\_funcionario (telefone\_funcionario, matricula) VALUES ('+5511949823749' , 3);

INSERT INTO tb\_telefone\_funcionario (telefone\_funcionario, matricula) VALUES ('+5511934892723' , 4);

INSERT INTO tb\_telefone\_funcionario (telefone\_funcionario, matricula) VALUES ('+5511998743298' , 5);

/\*Inserindo dados na tabela tb\_produto\*/

INSERT INTO tb\_produto (nome, descricao, quantidade, data\_entrada, matricula, codigo\_fornecedor) VALUES ('Vaso Sanitário Celite', 'Vaso Sanitário com Caixa Acoplada 3/6L Saída Vertical Eco Plus Branco Celite',10,'2020-03-26', 2, 1);

INSERT INTO tb\_produto (nome, descricao, quantidade, data\_entrada, matricula, codigo\_fornecedor) VALUES ('Telha de Fibrocimento', 'Telha de Fibrocimento Vogatex 4mm 50x244cm',10,'2020-03-26', 2, 2);

INSERT INTO tb\_produto (nome, descricao, quantidade, data\_entrada, matricula, codigo\_fornecedor) VALUES ('Vaso Sanitário Deca', 'Vaso Sanitário Convencional Saída Vertical Monte Carlo Gelo Deca',10,'2020-03-26', 2, 3);

INSERT INTO tb\_produto (nome, descricao, quantidade, data\_entrada, matricula, codigo\_fornecedor) VALUES ('Interruptor Pial', 'Conjunto de Interruptor Simples 10A Branco Pial Plus Pial Legrand',10,'2020-03-26', 2, 4);

INSERT INTO tb\_produto (nome, descricao, quantidade, data\_entrada, matricula, codigo\_fornecedor) VALUES ('Escada Botafogo', 'Banco Escada em Alumínio 3 Degraus Prata',10,'2020-03-26', 2, 5);

/\*Inserindo dados na tabela tb\_pedido\*/

INSERT INTO tb\_pedido (codigo\_cliente, data\_pedido) VALUES (2, '2020-01-20');

INSERT INTO tb\_pedido (codigo\_cliente, data\_pedido) VALUES (2, '2020-01-19');

INSERT INTO tb\_pedido (codigo\_cliente, data\_pedido) VALUES (1, '2020-03-27');

INSERT INTO tb\_pedido (codigo\_cliente, data\_pedido) VALUES (3, '2020-04-06');

INSERT INTO tb\_pedido (codigo\_cliente, data\_pedido) VALUES (5, '2020-04-16');

/\*Inserindo dados na tabela tb\_pedido\_produto\*/

INSERT INTO tb\_pedido\_produto (codigo\_pedido, codigo\_produto, matricula, quantidade, data\_saida) VALUES (1, 1, 4, 1, '2020-03-31');

INSERT INTO tb\_pedido\_produto (codigo\_pedido, codigo\_produto, matricula, quantidade, data\_saida) VALUES (2, 3, 3, 1, '2020-05-07');

INSERT INTO tb\_pedido\_produto (codigo\_pedido, codigo\_produto, matricula, quantidade, data\_saida) VALUES (3, 4, 2, 5, '2020-03-27');

INSERT INTO tb\_pedido\_produto (codigo\_pedido, codigo\_produto, matricula, quantidade, data\_saida) VALUES (4, 5, 5, 1, '2020-04-06');

INSERT INTO tb\_pedido\_produto (codigo\_pedido, codigo\_produto, matricula, quantidade, data\_saida) VALUES (5, 5, 5, 1, '2020-04-16');

### Os relatórios que foram solicitados e os respectivos comandos Select.

/\* 1 - Listar do produto: nome, nome do fornecedor e quantidade em estoque. \*/

select pro.nome as nome\_produto, pro.quantidade as quantidade\_estoque, forn.nome as nome\_fornecedor from tb\_produto pro, tb\_fornecedor forn WHERE forn.codigo\_fornecedor = pro.codigo\_fornecedor;

/\* 2 - Informar a quantidade de pedidos realizados por nome do cliente no mês de janeiro de 2020 \*/

select cli.nome\_cliente, count(ped.data\_pedido) as quantidade\_pedido from tb\_cliente cli, tb\_pedido ped where cli.codigo\_cliente = ped.codigo\_cliente and ped.data\_pedido BETWEEN '2020-01-01' and '2020-01-31' GROUP by cli.nome\_cliente;

/\* 3 - Informar qua quantidade de autorizações geradas por funcionário. \*/

select fun.matricula as matricula\_funcionario, fun.nome\_funcionario, count(pep.data\_saida) as quantidade\_autorizacoes from tb\_funcionario fun left join tb\_pedido\_produto pep on fun.matricula = pep.matricula group by fun.matricula;

/\* 4 - Listar os funcionários: nome, email, matricula, cpf e rg, ordenando pelo nome. \*/

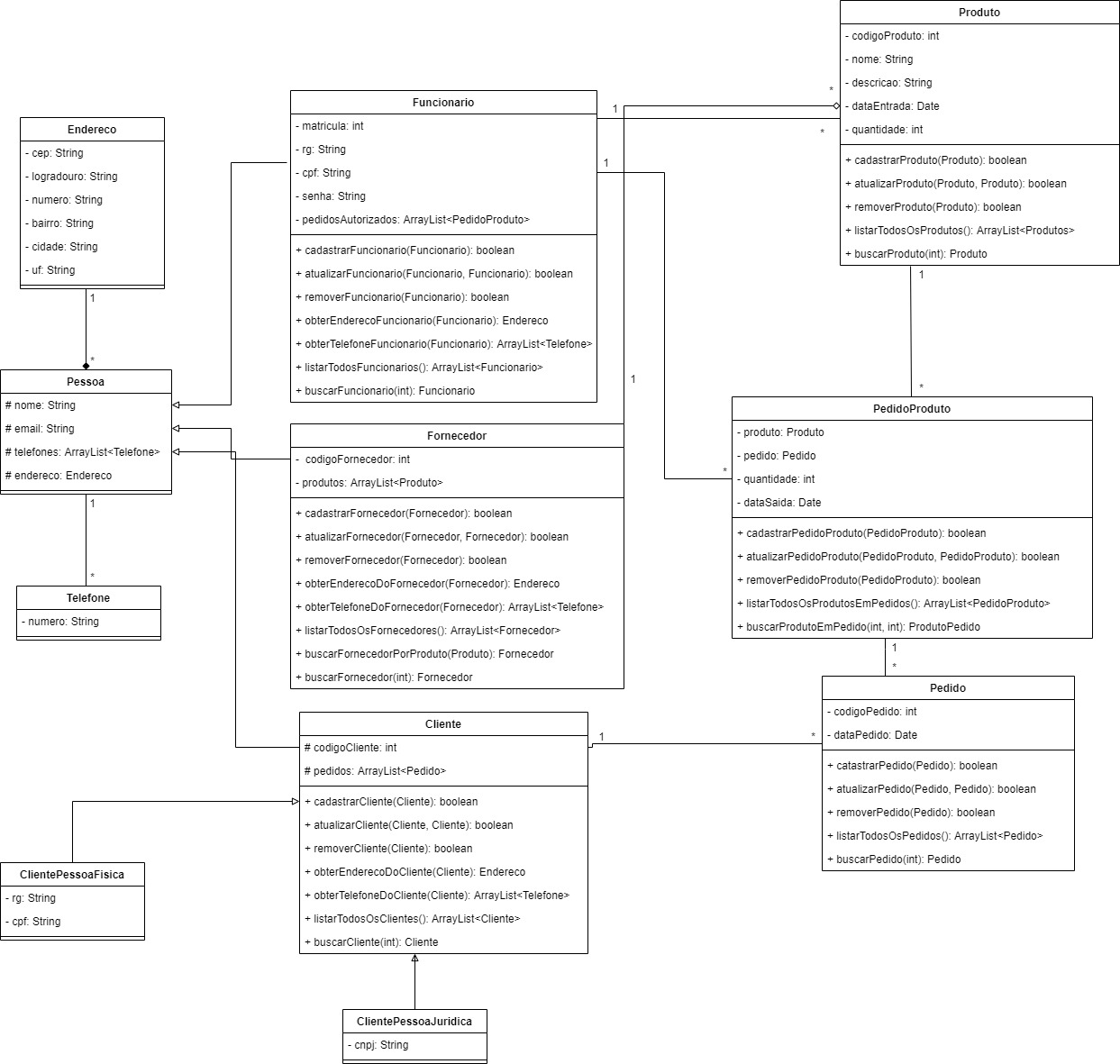
select nome\_funcionario, matricula, cpf, rg, email from tb\_funcionario order by nome\_funcionario;

/\* 5 - Infomar a quantidade de produtos por fornecedor. \*/

SELECT forn.codigo\_fornecedor, forn.nome as nome\_fornecedor, sum(pro.quantidade)\*count(pro.quantidade) as quantidade\_de\_produto from tb\_fornecedor forn left join tb\_produto pro on pro.codigo\_fornecedor = forn.codigo\_fornecedor GROUP BY forn.codigo\_fornecedor;

# Análise e Design

## Diagrama de Classes



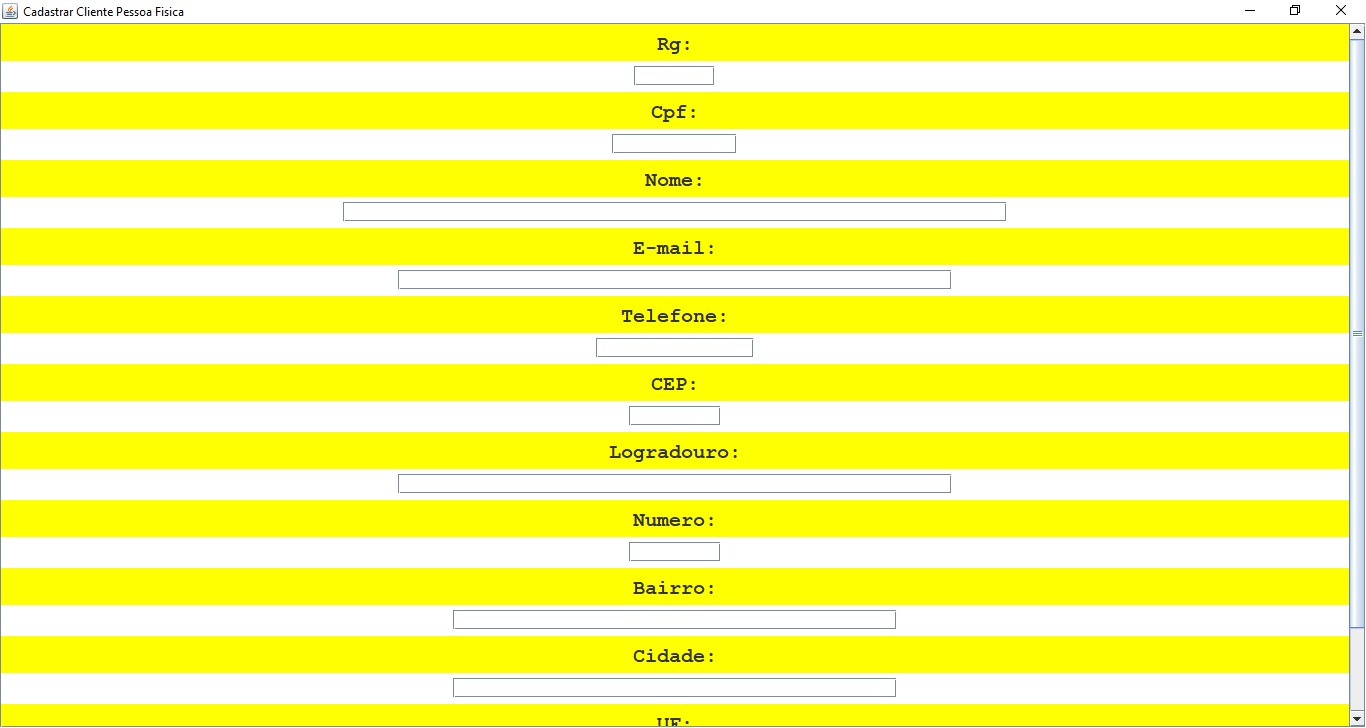
# Arquitetura do Software

## Padrão de projeto

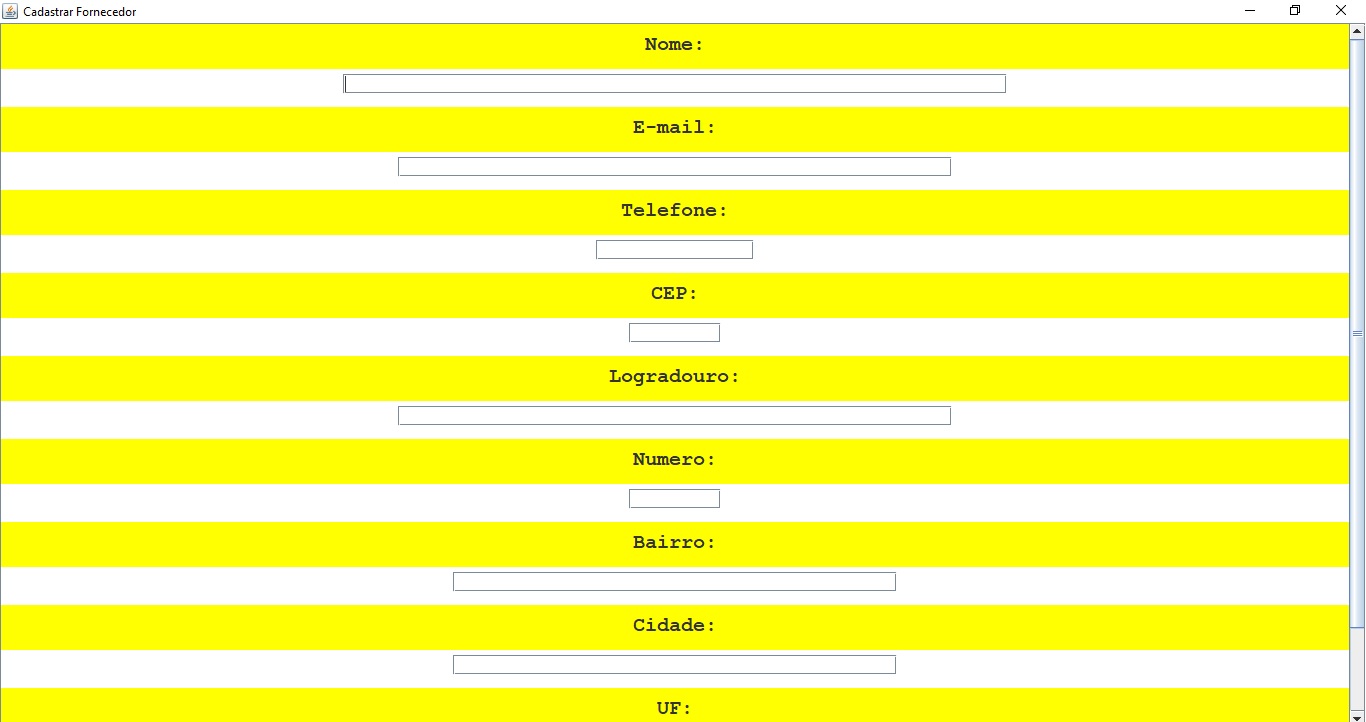
O padrão de projeto utilizado será o MVC cada classe acima terá o model que será feito com os atributos, métodos construtores, gets e sets. Cada classe terá uma tela de cadastro e também terá uma tela com os seguintes relatórios: 1 - Listar do produto: nome, nome do fornecedor e quantidade em estoque; 2 - Informar a quantidade de pedidos realizados por nome do cliente no mês de janeiro de 2020; 3 - Informar quantidade de autorizações geradas por funcionário; 4 - Listar dos funcionários: nome, email, matricula, cpf e rg, ordenando pelo nome; 5 - Infomar a quantidade de produtos por fornecedor. E no controller será por onde passarão os parâmetros recebidos pela view para fazer a requisição utilizando a model e fazendo a requisição no banco via classes DAO.

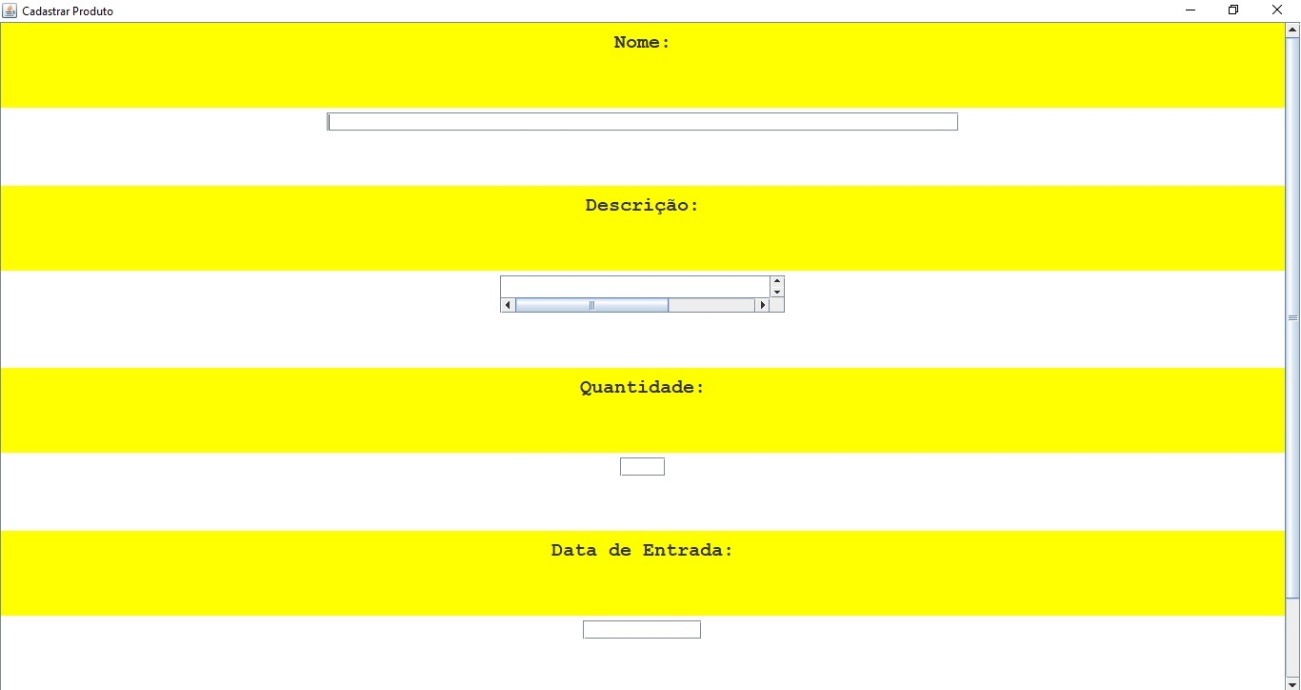
## Protótipo

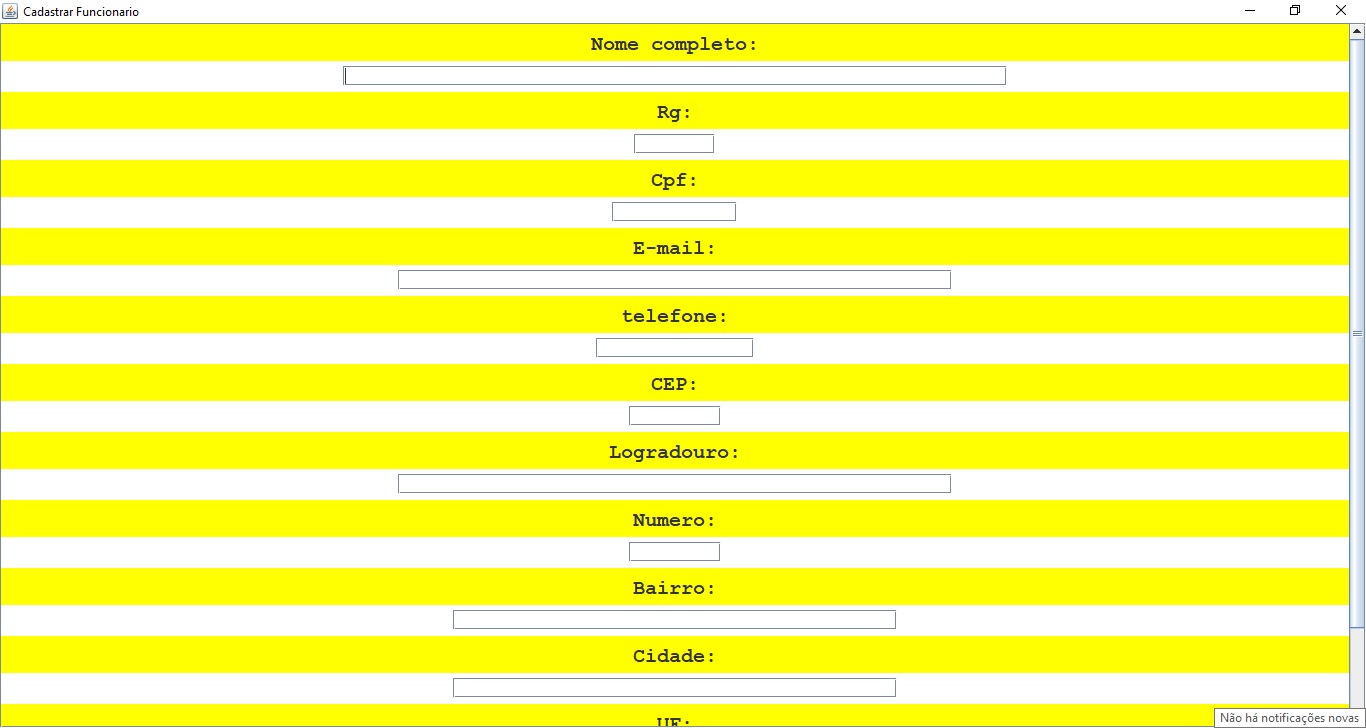
### Telas e classes da camada View

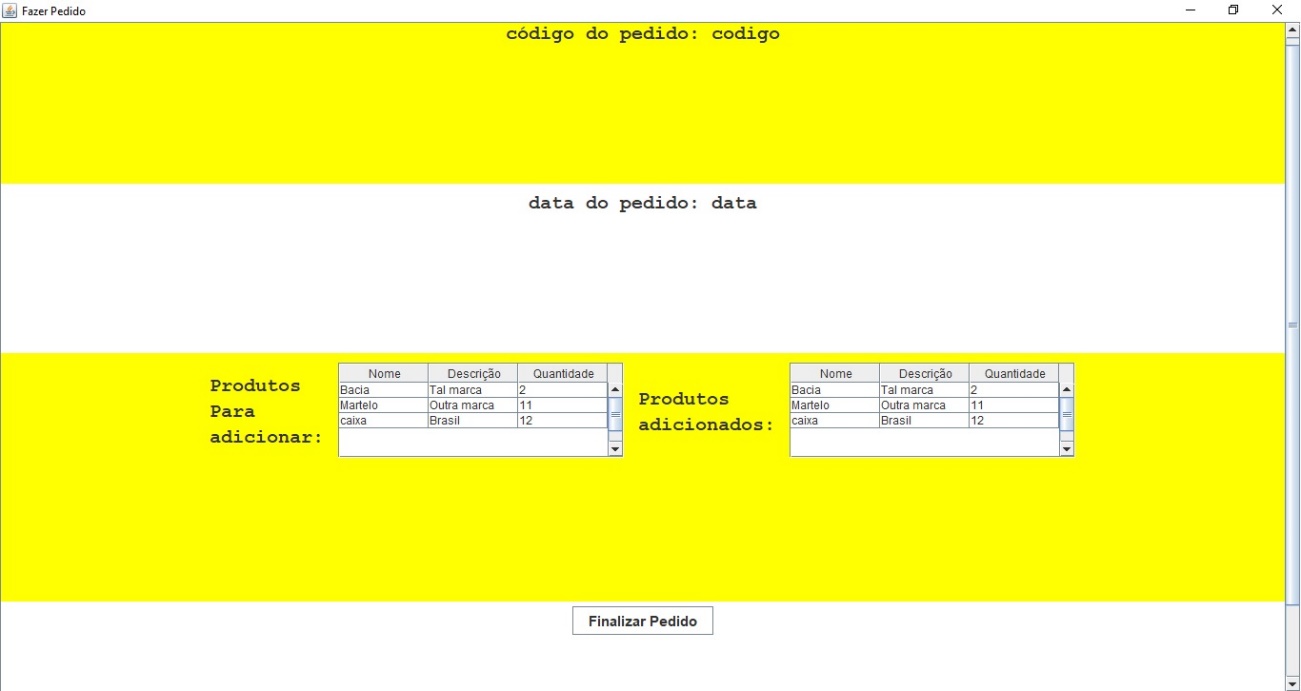
Tela de Cadastro de Cliente sendo Pessoa Física: RF01 – Cadastrar Cliente Pessoa Física

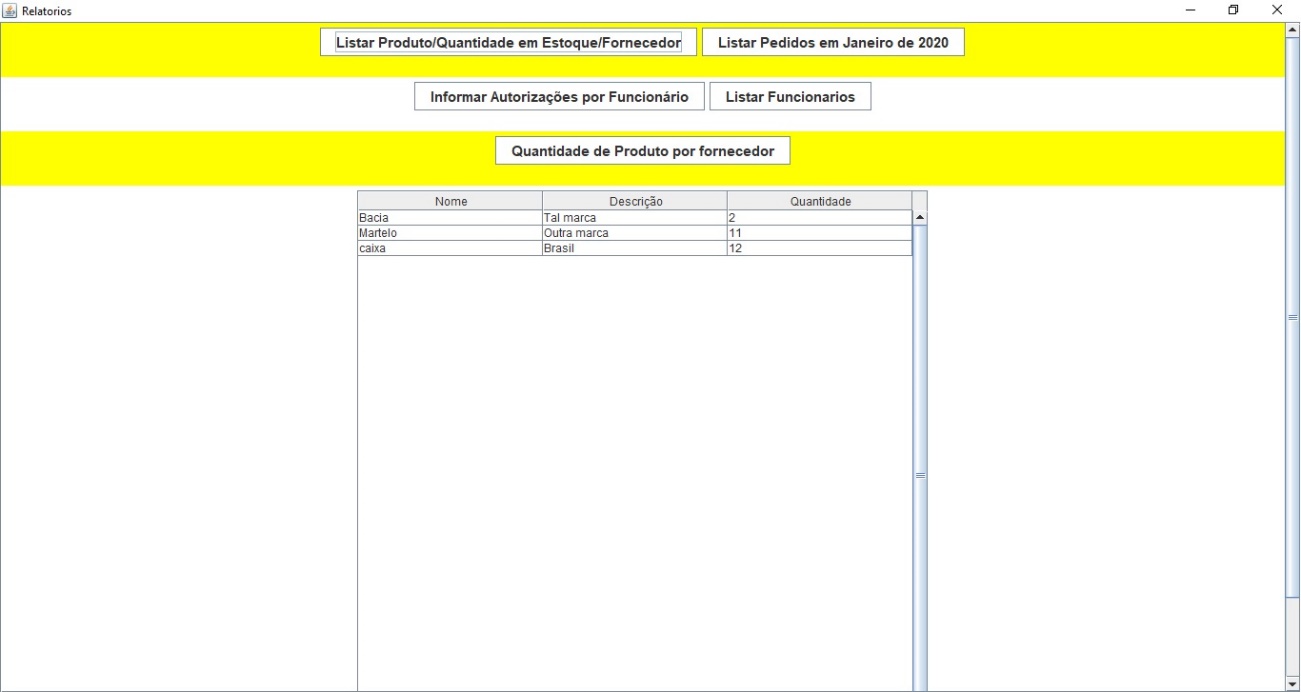
Tela de Cadastro de Cliente sendo Pessoa Jurídica: RF02 – Cadastrar Cliente Pessoa Jurídica

Tela para cadastrar Fornecedor: RF03 – Cadastrar Fornecedor

Tela para cadastrar produto: RF04 – Cadastrar Produto

Tela para cadastrar Funcionário: RF05 – Cadastrar Funcionário

Tela Para Fazer Pedido: RF06 – Realizar Pedido

Tela com alguns Relatórios: RF07 - Listar do produto: nome, nome do fornecedor e quantidade em estoque; RF08 - Informar a quantidade de pedidos realizados por nome do cliente no mês de janeiro de 2020; RF09 - Informar quantidade de autorizações geradas por funcionário; RF10 - Listar dos funcionários: nome, email, matricula, cpf e rg, ordenando pelo nome; RF11 - Infomar a quantidade de produtos por fornecedor.

### Classes da camada Control

Apresentar a descrição do código da classe.

### Classes da camada Model

Apresentar a descrição do código da classe.