

# PROGRAMAÇÃO DE SOLUÇÕES COMPUTACIONAIS

Projeto A3

São Paulo, 2022

## Integrantes:

Gustavo Y. de Souza - RA: 125221103735.

João Lucas Pimenta Barros - RA: 12522186250.

Matheus Tonetti Ferreira - RA: 12522174089.

Rafael Vieira Craveiro - RA: 12522163896.

Vitor de Oliveira Silva - RA: 12522117004.

# Resumo

Nosso projeto foi desenvolvido usando o Netbeans 13 junto com o Banco de dados PostgreSQL. Nosso programa foi criado na intenção de ser usado em uma loja de eletrônicos, por isso conta com o cadastro de funcionários, cadastro de produtos e cadastro de novas vendas, todos integrados com o banco de dados.

# Arquivo

O projeto executável em java está localizado com o nome A3, junto com os Scripts do banco de dados.

O arquivo nomeado de SriptTabelaFuncionario é o Script de criação da tabela onde se armazena os dados do funcionário.

O arquivo nomeado de ScriptTabelaProdutos é o Script de criação da tabela onde se armazena os dados dos produtos.

O arquivo nomeado de ScriptTabelaVendas é o Script de criação da tabela onde se armazena os dados sobre as vendas.

Todas as tabelas são incrementadas automaticamente pelo projeto java, que foi separado em vários pacotes, sendo eles: Controlers, Frames, Model, dao.

Controlers: Nos controlers estão localizadas todas as classes que controlam o software, controlando tanto a inicialização quanto o controle dos funcionários, produtos e vendas.

Frames: Nos Frames estão localizadas todas as telas do software.

Model: No Model é onde ocorre a formatação de um objeto tipo funcionário, produto ou venda para o salvamento no banco de dados.

Dao: O pacote Dao armazena as classes em que fazem o controle com o banco de dados, é por lá que os objetos são salvos no banco de dados.

## Interface

Nosso projeto funciona como um esquema dominó onde por exemplo o botão de salvar funcionário quando pressionado ele executa um o processo para formar o funcionário, que ocorre no Controler dos funcionários, que depois de formado é enviado a classe Dao dos funcionários onde ele insere no banco de dados e retorna uma confirmação que o funcionário foi salvo com sucesso.

Isso ocorre para todos os processos de salvamento, já para salvar uma venda o processo é um pouco maior, pois além de salvar os dados no banco, tem que remover a quantidade de unidades selecionadas dos produtos em estoque, e além disso verificar se há unidades disponíveis para realizar está venda, e caso tenha unidades suficientes, mas acabe com o estoque realizar a venda e deletar o produto da sua tabela.

Nosso código também conta com a opção de deletar itens da tabela produtos e da tabela funcionários manualmente usando as telas de gerenciar funcionário ou produtos.

O software também é capaz de gerar relatórios com a entrada das datas pelo usuário, mostrando a quantidade de produtos vendidos e o valor total durante o período.

# Métodos por pacotes

## Pacote Controles

Classe: Executável

Método main do programa

Classe: Funcionário Controler

Método SalvarFuncionario: ele pega todos os dados que foram colocados no frame e salva no model funcionário.

Método ExcluirFuncionario: pega os dados da tela de gerenciar funcionários e usa esse dado para selecionar o funcionário e manda esse usuário salvo para o model funcionário.

Classe: LoginControler

Método autenticar: pega os dados da tela de login e os salva em um modelo de funcionário que só possui o usuário e senha.

Classe: ProdutoControler

Método SalvarProduto: pega os dados da tela de cadastrar produtos e os salva em um model de produtos.

Método excluirProdutoPorCodigo: pega o código do produto, digitado na tela de gerenciar produtos e o salva em um modelo de produto com somente o código.

Classe: VendaControler

Método salvarVenda: pega o código e a quantidade digitado na tela de cadastrar venda e os salva em um model de venda.

Método preencherTabelaVenda: pega as datas iniciais e finais digitadas na tela de gerar relatórios e os salva em um model de relatório.

## Pacote Frames

Classe TelaCadastroFuncionario: tela do programa onde tem a entrada dos dados do funcionário.

Classe TelaCadastroProduto: tela do programa onde tem a entrada dos dados do produto.

Classe TelaCadastroVendas: tela do programa onde tem a entrada dos dados da venda.

Classe TelaGerenciarFuncionarios: tela do programa onde tem a entrada do usuário do funcionário que se pretende excluir.

Classe TelaGerenciarProdutos: tela do programa onde tem a entrada do código do produto que se pretende excluir.

Classe TelaLogin: tela do programa onde tem a entrada do usuário, senha e função do funcionário para fazer o login no programa.

Classe TelaPrincipal: tela do programa feita exclusivamente para os Administradores onde é aberta todas as outras telas dentro do seu JPanel, exceto a tela de login.

Classe TelaPrincipal\_1: tela do programa feita exclusivamente para atendentes, para que possam apenas cadastrar vendas.

Classe TelaRelatorio: tela do programa onde tem a entrada da data inicial e data final para poder gerar o relatório, que é apresentado na tabela da própria classe.

## Pacote Model

Classe Funcionario: onde cria as variáveis dos funcionários, são elas: nome, e-mail, data de nascimento, CPF, função, telefone, usuário e senha. Nessa classe é também onde se cria os construtores de funcionários para poder fazer o login, salvar no banco de dados e excluir funcionários.

Classe Produto: onde cria as variáveis dos produtos, são elas: nome, fabricante, quantidade, código e valor. Nessa classe é também onde se cria os construtores de produtos para salvar no banco de dados, para poder calcular o valor final de uma compra pelo valor do produto comprado, excluir produtos no banco de dados.

Classe Relatorio: onde cria as variáveis do relatório, são elas: data Inicial e data final. Nessa classe é também onde se cria os construtores do relatório para poder selecionar os dados das vendas no banco.

Classe Venda: onde cria as variáveis das vendas, são elas: código, quantidade e valor. Nessa classe é também onde se cria os construtores das vendas para salvar no banco de dados e para poder gerar o relatório.



## Pacote Dao

### Classe Conexao

Método getConnection: onde faz a conexão inicial com o banco de dados.

### Classe FuncionarioDao

Método insertFuncionario: método usado para inserir o funcionário criado pelo FuncionarioControler no banco de dados.

Método existePorUsuarioESenha: método usado para selecionar os dados do login que foi selecionado no LoginControler, e verificar se eles existem no banco de dados devolvendo assim sua função para poder fazer o login na função correta.

Método removerFuncionario: método usado para deletar o funcionário criado pelo FuncionarioControler do banco de dados.

### Classe ProdutoDao

Método insertProduto: método usado para inserir o produto criado pelo ProdutoControler no banco de dados.

Método removerProdutoPorCodigo: método usado para deletar o produto criado pelo ProdutoControler do banco de dados.

### Classe VendaDao

Método insertVenda: método usado para inserir a venda criada pelo VendaControler no banco de dados, preenchendo automaticamente o valor da venda feito pela quantidade do produto vendido pelo seu preço selecionado pelo código do produto. Além disso, esse método também reduz a quantidade de produtos em estoque pela subtração

das unidades vendidas das unidades em estoque. Também verifica se tem unidades disponíveis para realizar essa venda e se tiver, mas acabar as unidades, ele deleta o produto no banco de dados.

Método `preencherTabela`: seleciona no banco de dados todas as vendas em um período selecionado na `VendaController` e salva os dados em uma lista.

Método `preencherTabelaVenda`: utiliza os valores da lista feita em `preencherTabela` para popular a tabela com as informações das vendas, além de somar todos os valores das vendas e somar também as unidades vendidas para assim poder gerar um relatório que mostra o código do produto vendido, quantidade vendida, valor da compra, unidades totais vendidas e a renda total no período.