

UNIP – Universidade Paulista

###### Ciência da Computação (CC)

Formulário – Sistema de Estoque.

Disciplina: <Linguagem Orientada a Objeto>

Prof. Fabrício Henrique

Integrantes:

* <N313JE-4> - Gabriel Andrade Cabral
* <D660EA-4> - Eduardo Monita Dias
* <D78774-8> - Murilo Linares Mattioli

Sumário

[1. Objetivo e motivação do trabalho 3](#_Toc10308909)

[1.1 Objetivo principal 3](#_Toc10308910)

[1.2 Objetivo secundário 3](#_Toc10308911)

[2. Introdução 4](#_Toc10308912)

[3. Referencial Teórico 6](#_Toc10308913)

[4. Plano de desenvolvimento da aplicação 11](#_Toc10308914)

[5. Projeto do programa 12](#_Toc10308915)

[6. Relatório com as linhas de código 13](#_Toc10308916)

[7. Bibliografia 14](#_Toc10308917)

[8. Ficha dos participantes 15](#_Toc10308918)

# 1. Objetivo e motivação do trabalho

## 1.1 Objetivo principal

Proporcionar um Sistema de Estoque completo em para empresas de pequeno e médio porte. Com o intuito de melhoras nossas habilidades em programações Orientada a Objeto e em SGBD.

## 1.2 Objetivo secundário

Ter uma experiência de como seria o mercado a fora em relações ERPs já existentes. Sendo assim transformando a APS em si em uma experiência quase completa, pois nos dias de hoje as empresas atuantes são similares ao projeto em questão.

# 2. Introdução

A gestão empresarial dentro de uma empresa é algo que diferencia grandes negócios de pequenos negócios, mas como se destacar entre esse mercado competitivo? Desde o final da década de 50 os entendimentos sobre o controle de tecnologias e gestões corporativas começaram a acender, mas como o processo era bem caro e ainda demandava tempo no processo poucos usavam.

Durante a época de 70 com a maior disseminação computacional na história foi inventado o primeiro conceito “completo” de estoque para a época, os MRPs(Material Requirement Planning ou planejamento das requisições de materiais), eles já eram considerados um conjunto de sistemas, ou pacotes, que trocavam informações entre si onde possibilitavam o planejamento e controle financeiro da empresa.

Outro marco importante para os ERPs(Enterprise Resource Planning ou planejamento dos recurso da empresa) é o início das redes de computadores ligados a servidor, este fator escalonou milhares de vezes as funções do programa para a empresa, barateando também o seu uso empresarial, dessa maneira possibilitando que não somente negócios de grandes porte, mas os de médio poderiam adquirir esse sistema.

Após e durante a evolução tecnológica o que não faltou foi o seu crescimento, entre a década de 70 até os dias de hoje é muito difícil descrever somente com palavras a importância da tecnologia nas nossas vidas, hoje viver sem elas seria algo bem frustrante. E o mesmo vale para os ERPs, com o tempo conseguiram interligar todos os setores da empresa, que por fim acabam diminuindo as incertezas do mercado a fora. Diante deste fato, gradualmente os ERPs otimizam o trabalho de todos os funcionários e com o estudo dos seus dados a margem de erro se torna crescentemente menores, dessa maneira cada vez a certeza de uma decisão venha à tona mais facilmente para o CEO do negócio. O dono empresarial não necessita confiar tanto em seus instintos e sim em como satisfazer os números que aprecem em sua tela.

Enfim, o tema de estoque é algo que é desenvolvido antes da nossa geração sequer começar a pensar em nascer, o desenvolvimento do mesmo mostrará a importância dele e como o mercado pode evoluir a partir de mãos “milagrosas” como as nossas de programadores.

# 3. Referencial Teórico

Para ambientar como foi o desenvolvimento de um controle de estoque, e como seus dados foram armazenados, é impossível deixar de falar de Banco de Dados, tecnologia essa que revolucionou, não só a indústria, mas a vida das pessoas em um todo, porém focaremos em como essa forma de armazenar dados mudou a indústria, e o modo de como armazenamos dados, e movimentamos produtos, para isso precisamos voltar lá atrás, em uma época onde o banco de dados era uma tecnologia recente.

Antigamente, para armazenar informações era muito mais complicado e trabalhoso, era necessário papel, dentro de pastas, que ficavam dentro de arquivos, e isso ocupava muito espaço, com o avanço tecnológico isso mudou drasticamente, graças ao Banco de Dados e seus modelos.

Esse marco ocorreu no início da década de 60, onde havia dois principais modelos de banco de dados no mercado, o modelo de rede (CODASYL - Comitee for Data Systems Language) e o modelo hierárquico (IMS – Information Management System), onde os mesmos tinham seus prós e contras, porém foi somente no início dos anos 70 que começou a se popularizar o modelo relacional, utilizado até hoje, ele promovia muito mais segurança, compartilhamento, e integridade dos dados, e foi no meio dos anos 70 em que o termo “Sistema de Gerenciamento de Banco de Dados Relacional” foi definido.

A IBM (*“[...] International Business Machines (IBM) é uma empresa dos Estados Unidos voltada para a área de informática.*

*A empresa é uma das poucas na área de tecnologia da informação (TI) com uma história contínua que remonta ao século XIX. A IBM fabrica e vende hardware e software, oferece serviços de infraestrutura, serviços de hospedagem e serviços de consultoria nas áreas que vão desde computadores de grande porte até a nanotecnologia. [...]”)*  foi muito importante para os avanços tecnológicos do século XIX e XX, e como era de se esperar, teve grande importância nessa revolução de armazenagem de dados, tendo criado um projeto chamado System R, com o intuito de utilizar algumas ideias inspiradas por Edgar Frank Codd, um de seus funcionários na época e também conhecido como o Pai do modelo Relacional.

Do meio para o final dos anos 70 aparece Peter Chen, propondo o modelo Entidade-Relacionamento (ER) (“*[...]* *modelo entidade relacionamento (modelo ER) é um modelo de dados para descrever os dados ou aspectos de informação de um domínio de negócio ou seus requisitos de processo, de uma maneira abstrata que em última análise se presta a ser implementada em um banco de dados, como um banco de dados relacional. Os principais componentes dos Modelos Entidade-Relacionamento (MER) são as entidades (coisas, objetos) suas relações e armazenamento em bancos de dados. [...]*”), uma lógica para toda a construção do banco de dados relacional, facilitando a vida do projetista de banco.

No final dos anos 70, surge um dos SGBD’s mais utilizados na indústria nos dias de hoje, o Oracle (“[...] *É um SGBD (sistema gerenciador de banco de dados) que surgiu no fim dos anos 70, quando Larry Ellison vislumbrou uma oportunidade que outras companhias não haviam percebido, quando encontrou uma descrição de um protótipo funcional de um banco de dados relacional e descobriu que nenhuma empresa tinha se empenhado em comercializar essa tecnologia.* [...]”), mas só foi em 1983 que o Oracle teve sua Ascenção, derrubando seu maior competidor de mercado na época, o Igres.

(“*[...] em 1973 quando o Sistema R havia sido iniciado pela IBM, os pesquisadores liberaram muitos artigos descrevendo o sistema que construíam. Dois cientistas de Berkeley, Michael Stonebraker e Eugene Wong, se interessaram após lerem os artigos, e decidiram iniciar um projeto de banco de dados relacional para eles. [...]”.* Eles tiveram investimentos de diversas organizações, e em meados de 1970, o projeto Ingres é produzido, ele foi desenvolvido tanto por alunos como professores, tendo uma pequena comunidade de usuários na época).

E essa evolução de armazenamento de dados teve mais um estouro com a Linguagem Estruturada de Consulta – SQL (*“[...] é a linguagem de pesquisa declarativa padrão para banco de dados relacional (base de dados relacional). Muitas das características originais do SQL foram inspiradas na álgebra relacional. [...]”)* que foi um resultado do projeto System R citado anteriormente, que a partir da metade dos anos 80, se torna uma linguagem padrão no mundo inteiro deixando os modelos hierárquicos e de rede em segundo plano,

No início dos anos 90 tudo parecia muito calmo para essas inovações fervorosas que estavam vindo das décadas atrás, porém, no meio daquela década, surge algo que mudaria completamente como a tecnologia age, vem o início da internet.

Há um grande crescente das tecnologias WEB/BD, além do uso de soluções em código aberto, como o MySql e o PostgreSql por exemplo.

E não podemos esquecer do banco de dado orientado a objeto, onde no meio dos anos 90 o Java estava em alta, e a programação orientada a objeto também teve um relacionamento importante no mundo da tecnologia, já que naquela época, a Sun (proprietária do Java) estava guiando o rumo da internet naquele momento, era impossível essas duas tecnologias não se relacionarem, foi ai que surgiu o banco de dados orientado a objetos (“[...] *Um banco de dados orientado a objetos é um banco de dados em que cada informação é armazenada na forma de objetos, ou seja, utiliza a estrutura de dados denominada orientação a objetos, a qual permeia as linguagens mais modernas. Começou a ser comercialmente viável em 1980. O gerenciador do banco de dados para um orientado a objeto é referenciado por vários como ODBMS ou OODBMS. [...]”.)*

É obvio que nesse projeto que estamos desenvolvendo, utilizamos outras tecnologias, como a programação orientada a objeto em que utilizamos Java (“*Java é uma linguagem de programação orientada a objetos desenvolvida na década de 90 por uma equipe de programadores chefiada por James Gosling, na empresa Sun Microsystems. Em 2008 o Java foi adquirido pela empresa Oracle Corporation. Diferente das linguagens de programação modernas, que são compiladas para código nativo, a linguagem Java é compilada para um bytecode que é interpretado por uma máquina virtual (Java Virtual Machine, mais conhecida pela sua abreviação JVM). [...]”*) onde a mesma foi um grande marco no meio dos anos 90, e até hoje é muito importante, porém acreditamos que a presença do banco de dados e como ele age em todo esse método de armazenamento, controle, e movimento de estoque (“[...] controle de estoque *é uma área muito importante de uma empresa, grande ou pequena, pois é através dele que ela será capaz de prever o quanto que será necessário comprar no próximo pedido ao fornecedor, além de fornecer informações úteis sobre as vendas, já que muitas vezes os relatórios do setor de vendas não são muito claros e não condizem com a realidade, afinal, o setor de vendas quer comissões. O principal objetivo do controle de estoque é otimizar o investimento em estoques, aumentando o uso eficiente dos meios internos de uma empresa, e minimizar as necessidades de capital investido em estoque. [...]”*), se tornou mais presente em todo o desenvolvimento do trabalho.

# 4. Plano de desenvolvimento da aplicação

A aplicação se consiste em dois tópicos principais Cadastro – Este tópico será responsável por inserir novos dados ao banco, tendo validações necessárias para caso os dados não sejam validos, dentro desta parte da aplicação, haverá telas especificas para cada tipo de cadastro, como por exemplo:

Cadastro Item: Aqui será cadastrado novos itens com os seguintes campos:

* + - Código de barras
    - Nome
    - Descrição
    - Tipo
    - Valor unitário.

Cadastro Usuário: Aqui serão cadastrados os usuários que irão acessar a aplicação:

* + - Nome
    - Senha
    - Tipo
    - Usuário.

Cadastro Localização: Aqui será informado a localização dos itens e o responsável pelo setor com os campos:

* + - Código localização
    - Nome
    - Setor
    - CPF

Cadastro Pedido: Aqui será informado os dados que referenciam o pedido do item, pode ser de venda ou de compra dependendo do usuário:

* + - Nota Fiscal
    - Item
    - Usuário
    - Quantidade
    - Valor Total

Cadastro Estoque: Aqui será informado os dados que referenciam o a tabela estoque:

* + - Código do item
    - Localização
    - Quantidade

Cadastro Funcionário: Nesta tela, será incluído os dados do Funcionário, e seu devido cargo e localização no estoque, sendo necessário preencher estes campos:

* + - CPF
    - Localização
    - Nome
    - Cargo

Cadastro Solicitação: Nesta tela será onde os pedidos serão cadastrados e referenciados aos funcionários responsáveis:

* + - Código de solicitação
    - Pedido
    - Funcionário
    - Item Estoque

Há cadastros que são inteiramente independentes como é o exemplo do: Cadastro Item, Cadastro Localização e o Cadastro Usuário, eles não dependem de outros dados para serem criados.

Porém alguns cadastros são dependentes, por exemplo:

Cadastro Pedido, que depende de um usuário e um item para ser cadastrado

Cadastro Estoque: Depende de um cadastro de item, de um cadastro de localização, e uma solicitação cadastrada.

Cadastro Funcionário: Que dependerá de uma localização para ser cadastrada.

Todas as telas de cadastro terão o campo referenciado por uma label ao seu lado esquerdo, e um botão de salvar, os cadastros dependentes serão validados antes de salvar.

Consulta – Este tópico tem como objetivo consultar dados já cadastrados, elimina-los ou alterá-los, com validações com relação as possíveis alterações efetuadas, segue em seguida quais serão essas telas:

Consulta Item: Aqui será consultado quais itens já foram cadastrados, e seus devidos campos preenchidos anteriormente:

* + - Código de barras
    - Nome
    - Descrição
    - Tipo
    - Valor unitário

Consulta Usuário: Aqui será efetuada a consulta dos usuários já cadastrados com os seguintes campos:

* + - Nome
    - Senha
    - Tipo
    - Usuário

Consulta Localização: Aqui será feita a consulta da localização do estoque, onde o funcionário se encontra, e onde o item pode estar armazenado:

* + - Código localização
    - Nome
    - Setor
    - CPF

Consulta Estoque: Onde será possível visualizar o item e sua quantidade, além de mostrar onde ele está localizado:

* + - Código de Barra
    - Localização
    - Quantidade

Consulta Pedido: Nesta tela, onde vamos poder verificar o tipo de pedido, e sua situação atual:

* + - Nota Fiscal
    - Item
    - Usuário
    - Quantidade
    - Valor Total
    - Situação

Consulta Funcionário: Onde será possível verificar dados como CPF, Cargo, e data de Admissão:

* + - CPF
    - Localização
    - Nome
    - Cargo
    - Situação
    - Data Admissão

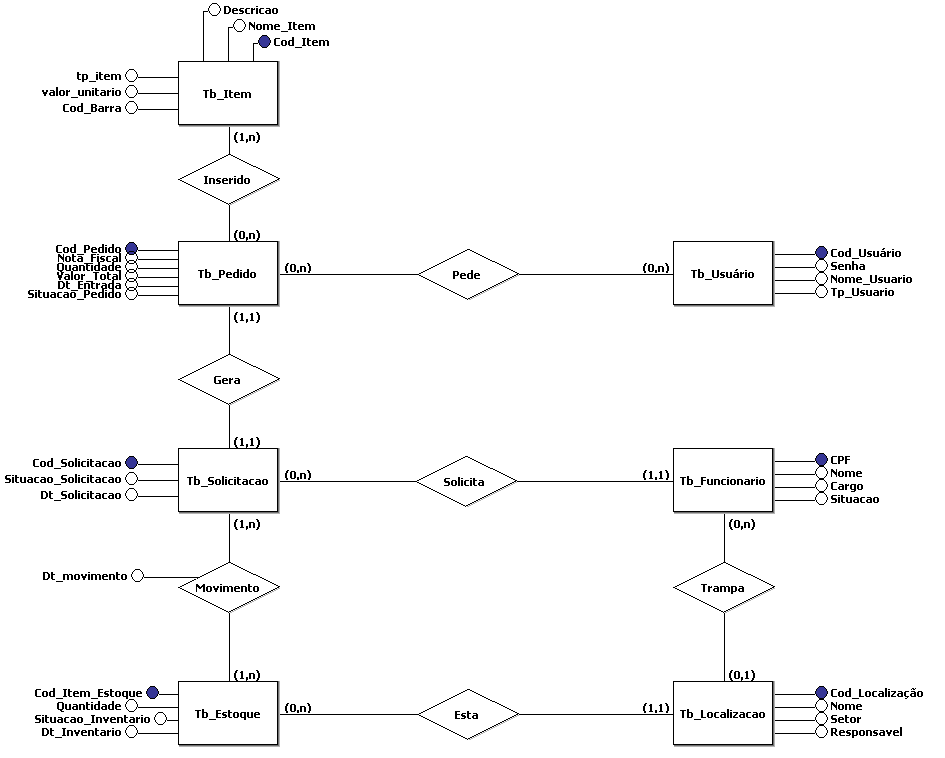
Consulta Solicitação: Onde será possível verificar a atual situação do pedido, e mais dados, como o código do solicitante, entre outras coisas:

* + - Código de Solicitação
    - Pedido
    - Nome Solicitante
    - Item Estoque
    - Situação
    - Data Solicitação

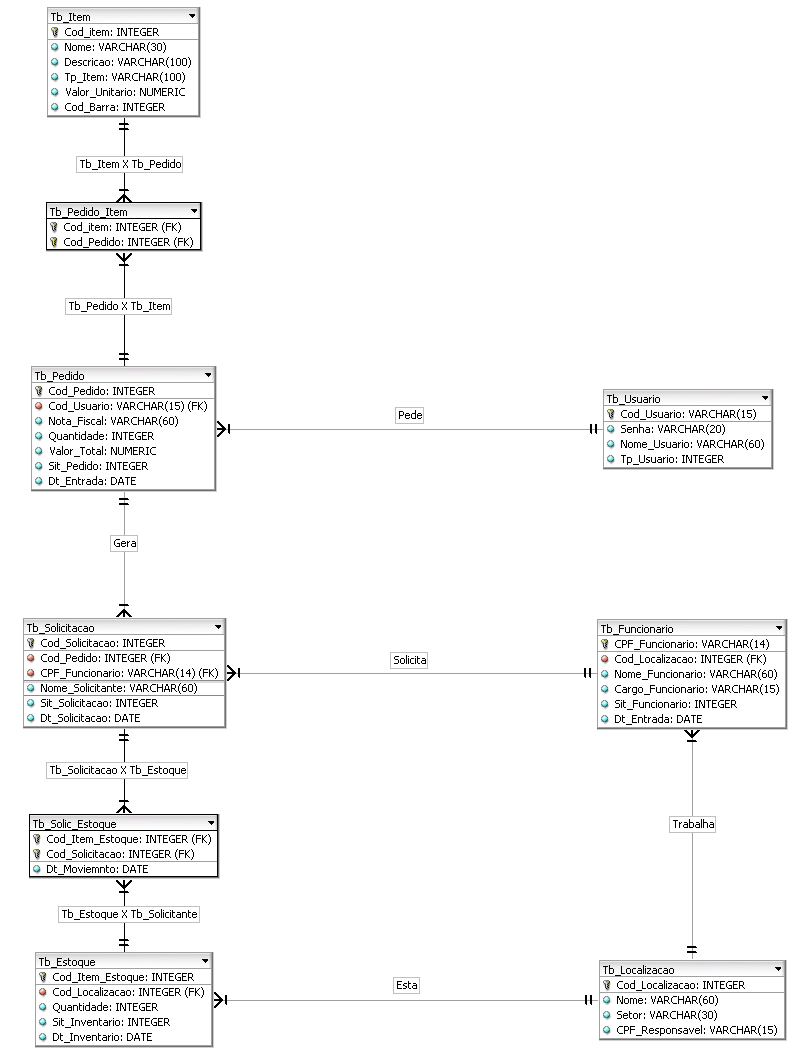
Assim como as telas de cadastro, as telas de consulta terão campos com a label referenciando os campos no seu lado esquerdo, e dois botões, um para fazer alterações que também serão validadas, e exclusões de determinados dados.

Todas as telas de consultas são dependentes.

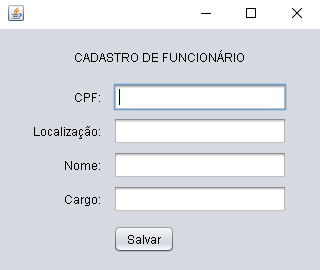
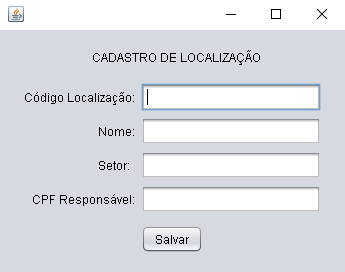
# 5. Projeto do programa

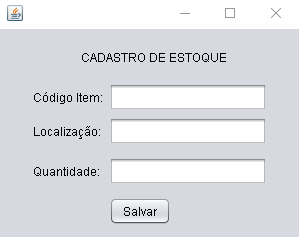
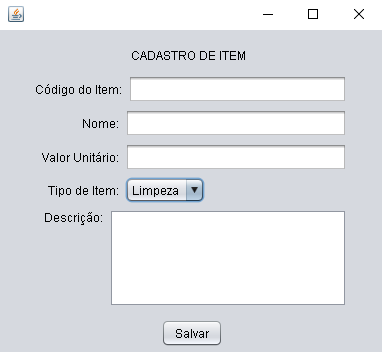
**Modelo Conceitual**

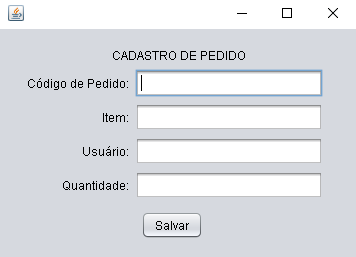
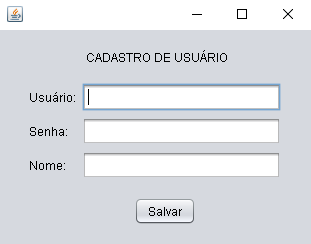
**Modelo Lógico**

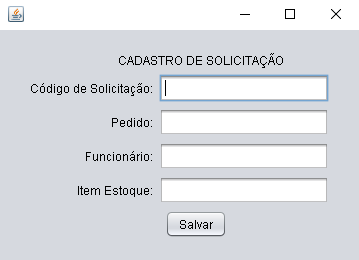
****

# 6. Relatório com as linhas de código

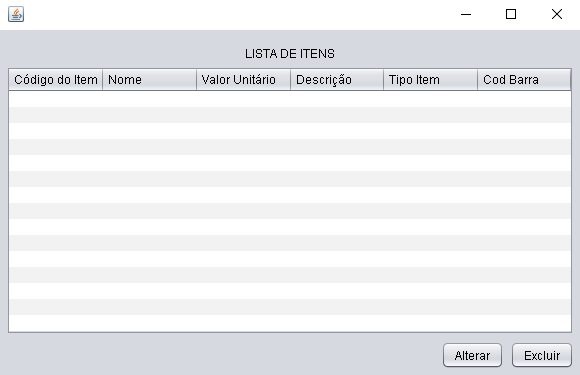
**Telas de Cadastrado**

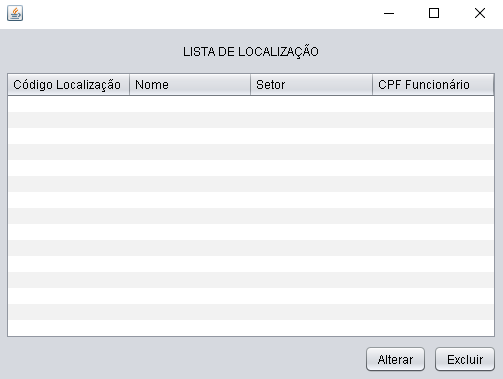
****

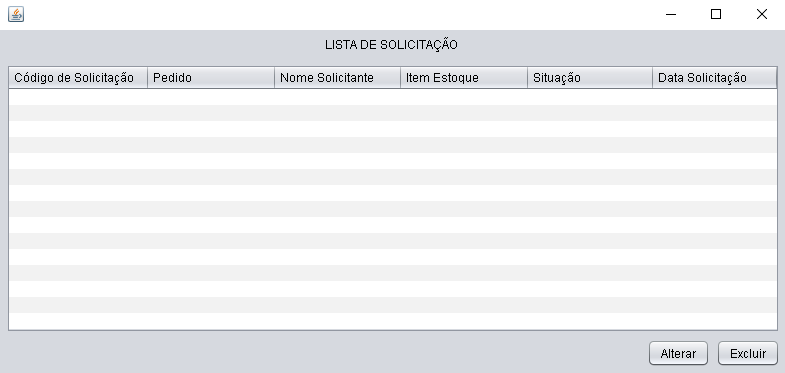


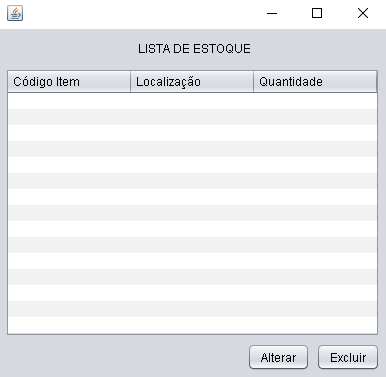


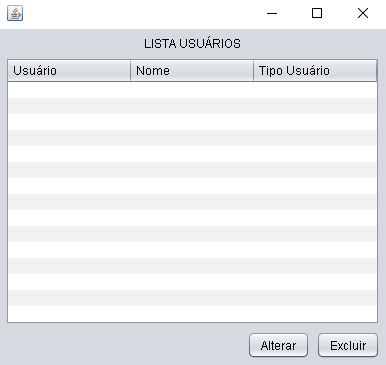
**Telas de Consulta**

****

****

****

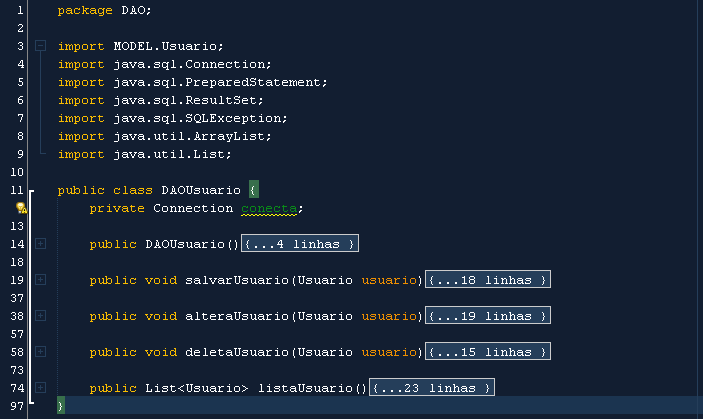
****

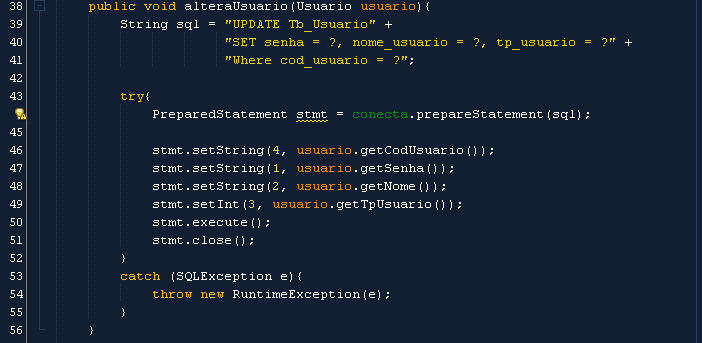
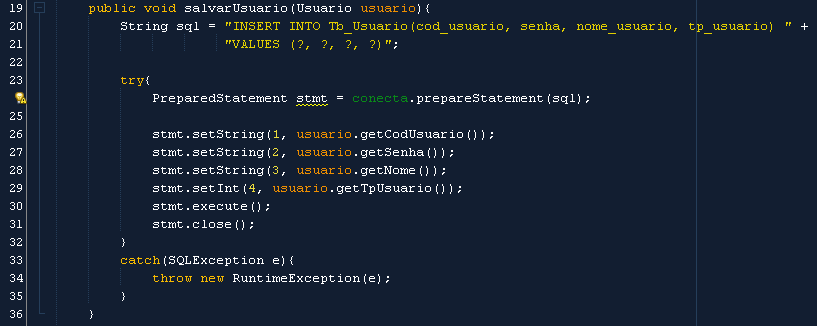
****

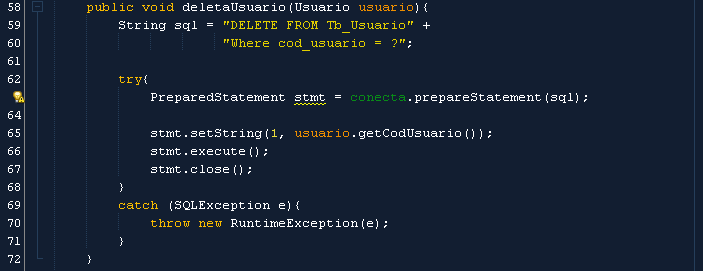
**Código**

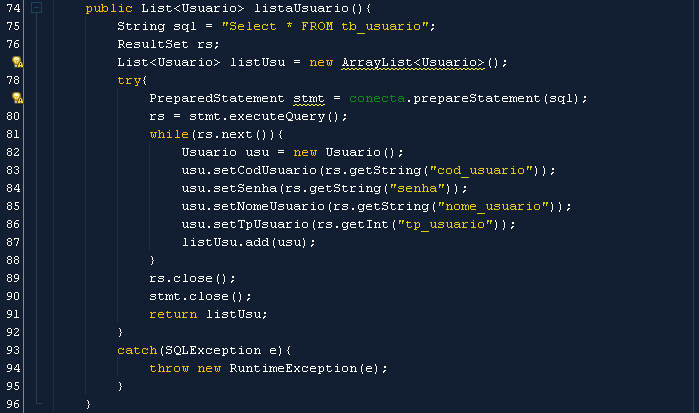
Os mesmos códigos foram usados nas outras telas, para emxemplificar segue abaixo:











# 7. Bibliografia

<https://teamsoftware.wordpress.com/2011/05/27/erp-definicao-historico-e-principais-objetivos/>

https://pt.wikipedia.org/wiki/Modelo\_entidade\_relacionamento

https://www.savepoint.blog.br/2017/10/24/um-pouco-sobre-a-historia-dos-bancos-de-dados-parte-i/

http://www.savepoint.blog.br/2017/10/24/um-pouco-sobre-a-historia-dos-bancos-de-dados-parte-ii/

https://dicasdeprogramacao.com.br/a-historia-dos-bancos-de-dados/

https://pt.wikipedia.org/wiki/SQL

https://pt.wikipedia.org/wiki/IBM

https://pt.wikipedia.org/wiki/Oracle\_(banco\_de\_dados)

https://pt.wikipedia.org/wiki/Ingres\_(banco\_de\_dados)

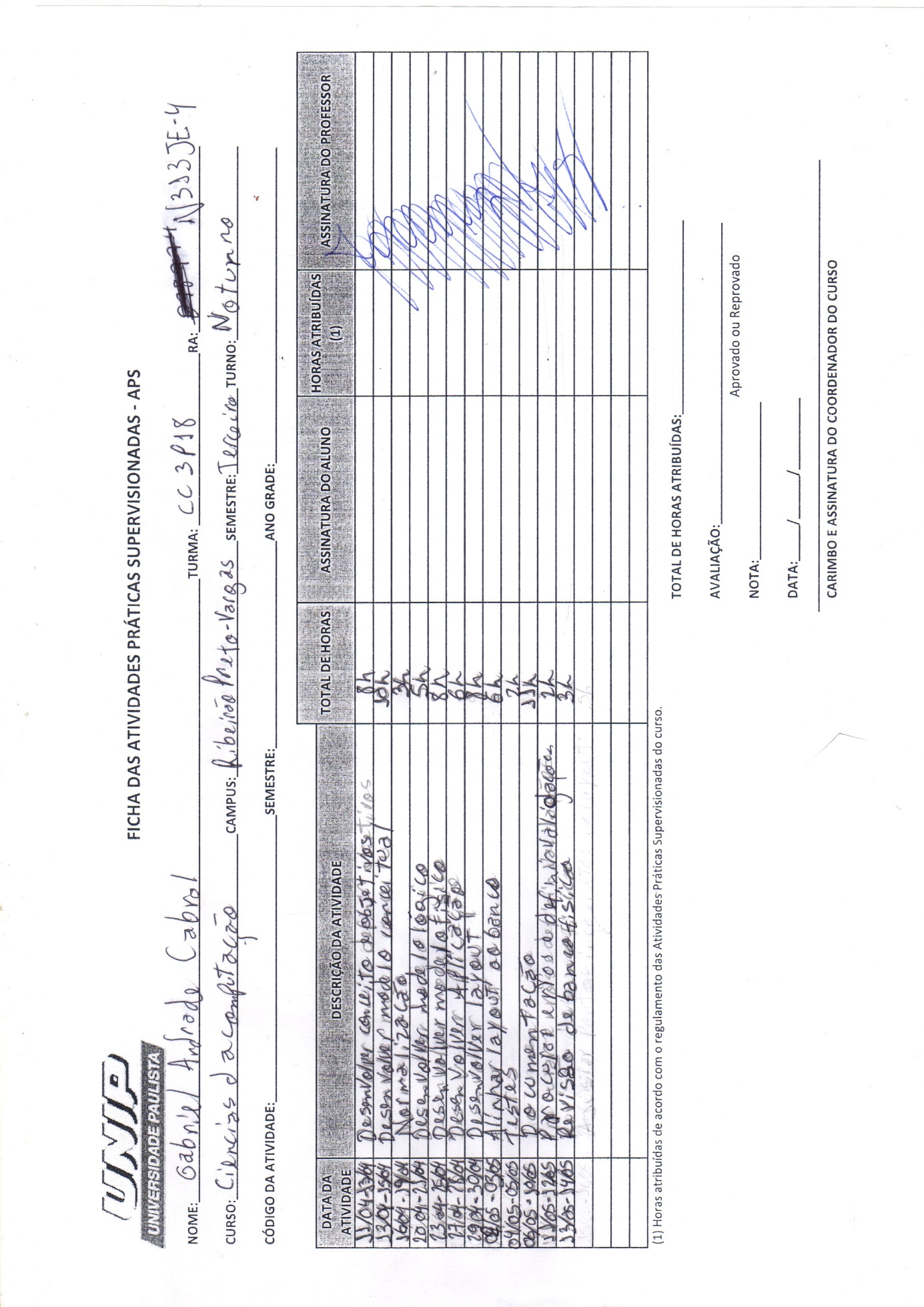
https://pt.wikipedia.org/wiki/Banco\_de\_dados\_orientado\_a\_objetos

https://www.devmedia.com.br/a-historia-dos-banco-de-dados/1678

Slides feitos pelo Professor Geraldo Henrique Neto que leciona a disciplina de Banco de Dados na instituição Unip Ribeirão Preto

https://pt.wikipedia.org/wiki/Java\_(linguagem\_de\_programação)

# 8. Ficha dos participantes



# 