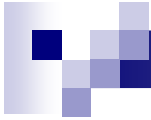


# Laboratório de ENGSOE – Estudo de Caso



# Laboratório ENGSOE

- Aula de Hoje:
  - Desenvolver um sistema UML inteiro:
    - Aplicação Banco Online.

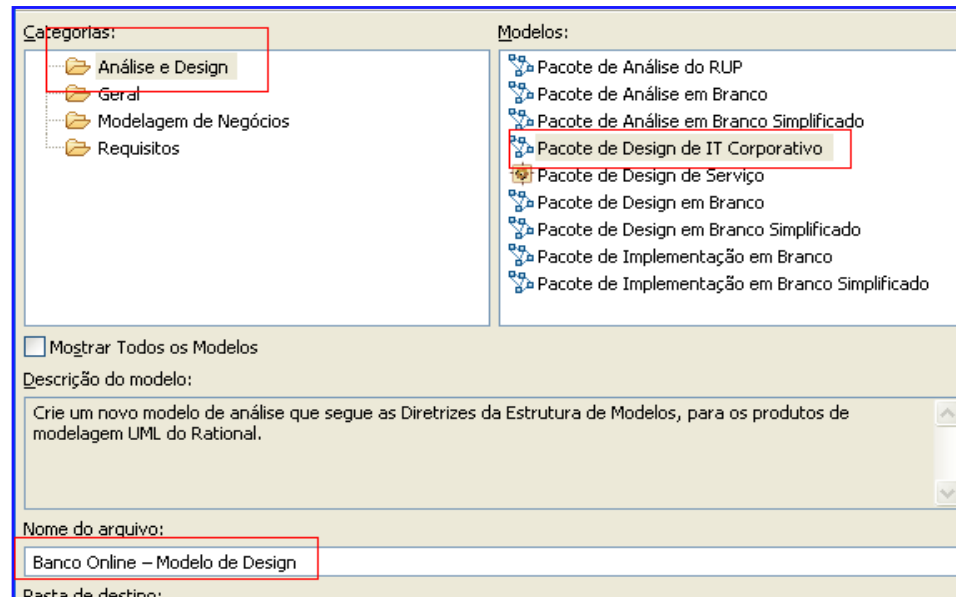
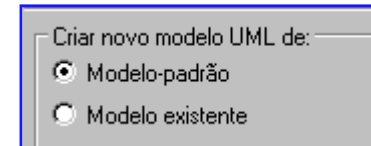


# Laboratório ENGSO

- Nosso “Estudo de Caso!”
  - RSA V7 – O que será feito para o projeto?
    - 1) Criando um Projeto UML:
      - 1) Fase de Design (Desenvolver o Projeto do Sistema):
        - 1) Preparação para o Design
        - 2) Criação do design da camada de integração.
        - 3) Aplicação de design patterns no domínio da aplicação.

# Laboratório ENGSOE

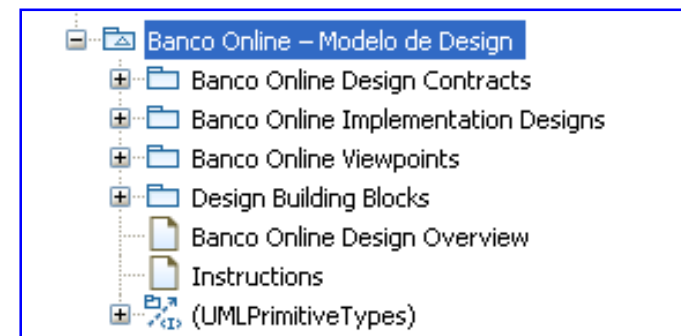
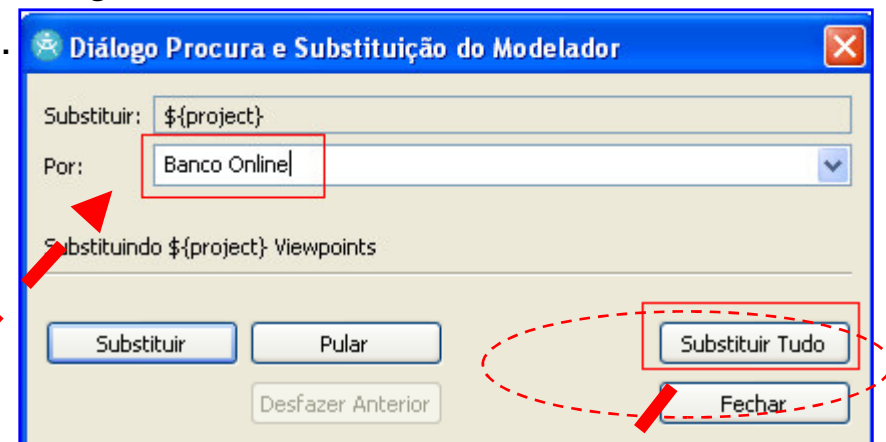
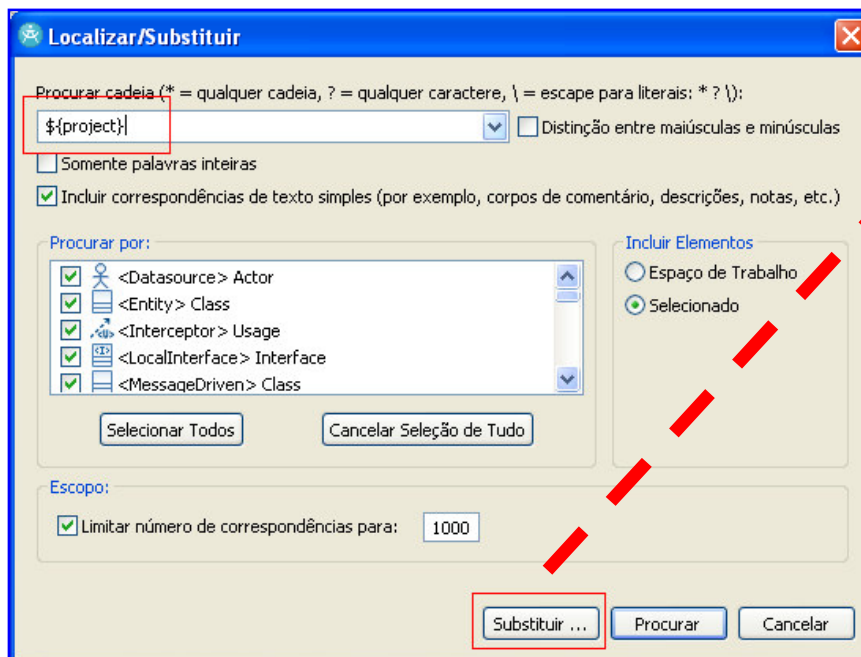
- Criando um modelo de Design “Banco Online – Modelo de Design”.
  - 1) Selecionar “ProjetoBancoOnline”, clicar com o botão direito, “Novo->Modelo UML”. Avançar.
  - 2) Escolher a opção “Modelo padrão”. Avançar.
  - 3) Escolher “Modelo do Design de IT Corporativo” e Colocar o nome “Banco Online – Modelo de Design”. Concluir.



***Não esquecer de salvar!!!***

# Laboratório ENGSO

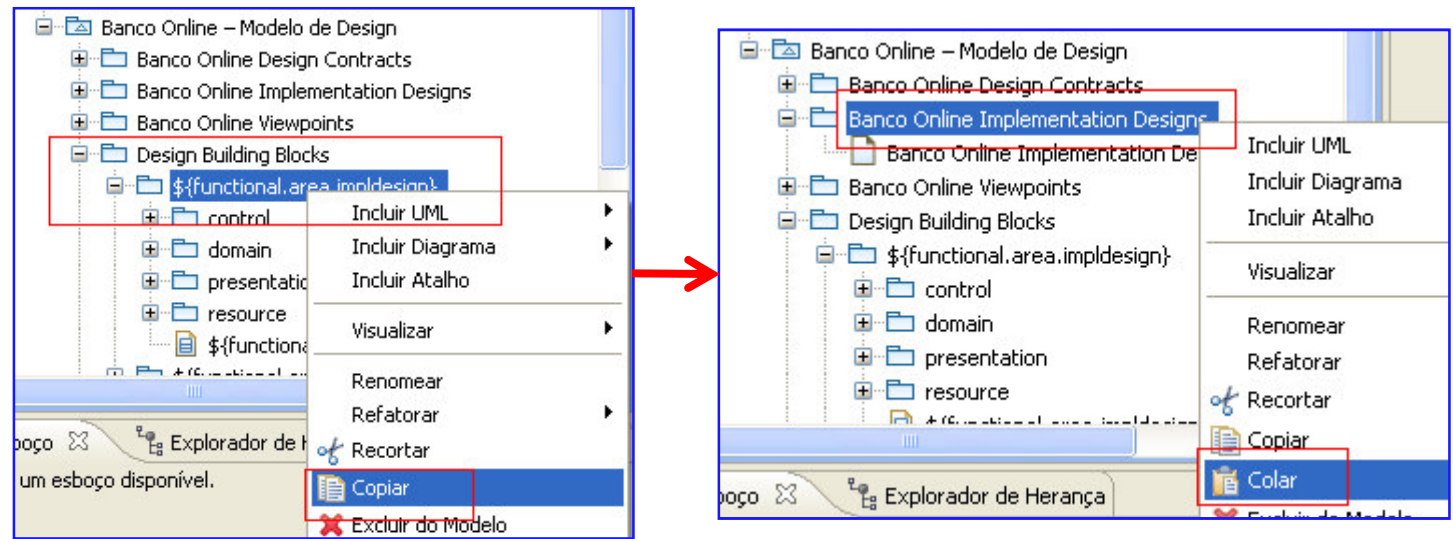
- Alterando o nome de `#{project}` para “Banco Online”.
  - Clicar em “Banco Online – Modelo de Design”.
  - No menu “editar->Localizar/Substituir”.
  - Substituir:
  - `#{project}` por “Banco Online”.



**Não esquecer de salvar!!!**

# Laboratório ENGSO

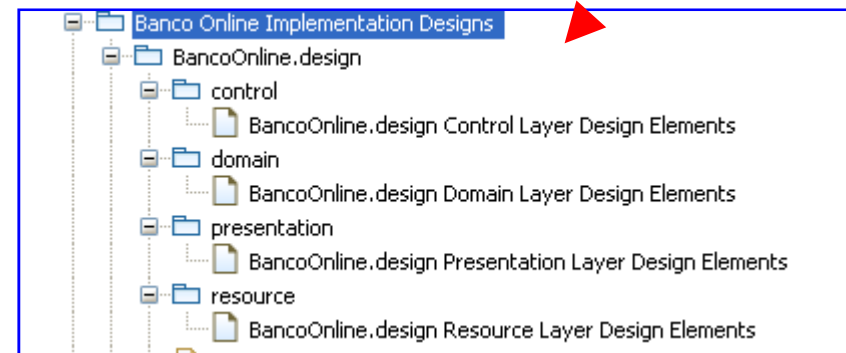
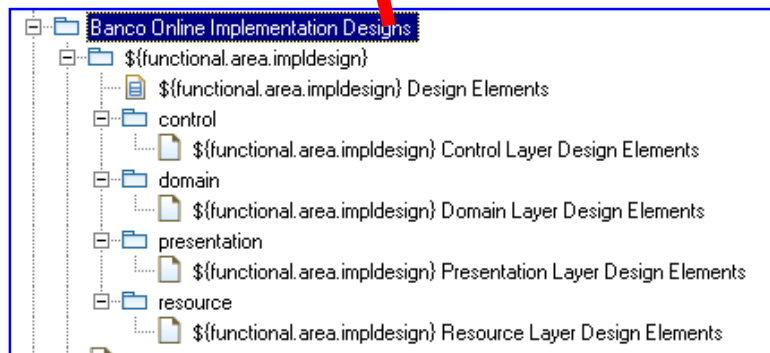
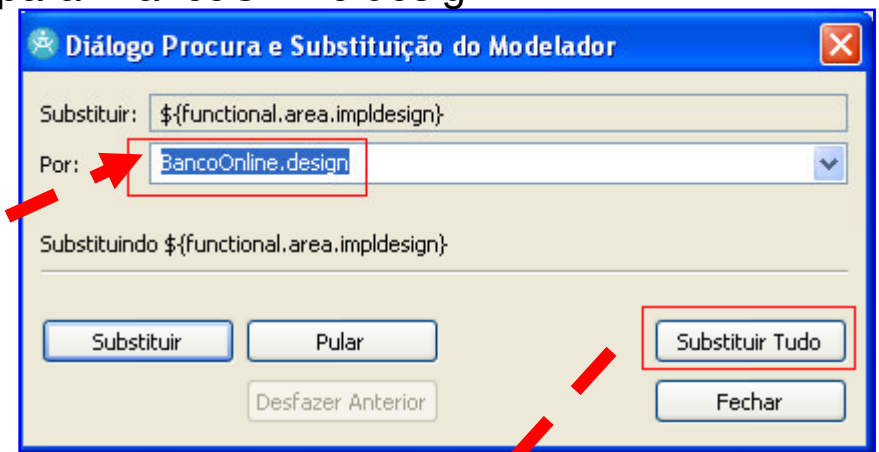
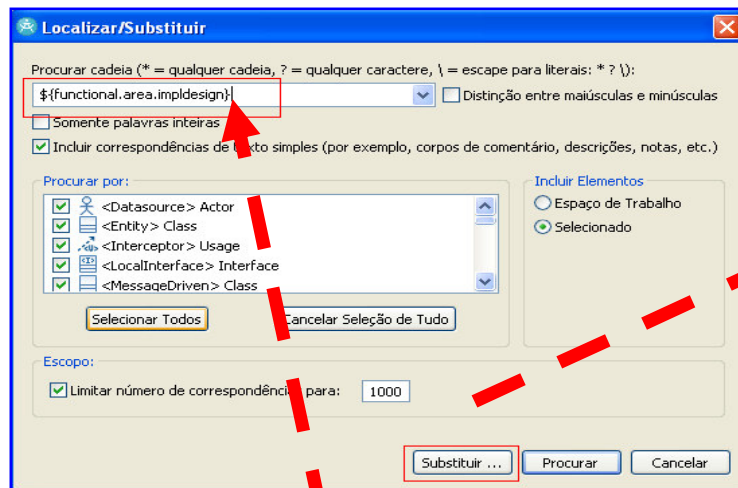
- Identificando os sub-pacotes de design de implementação e criando o diagrama de pacotes do Banco Online (Deve conter as classes de design e os diagramas de domínio de cada componente de design).
- Existem 4 subpacotes:
  - ☐ control.
  - ☐ domain.
  - ☐ presentation.
  - ☐ resource.
- Expandir “Banco Online – Modelo de Design”, “Design Building Blocks”, copiar o diretório “\${functional.area.impldesign}” e colar em “Banco Online Implementation Designs”.



***Não esquecer de salvar!!!***

# Laboratório ENGSOFT

- Alterando o nome de `#{functional.area.impldesign}` para “BancoOnline.design”.
  - Clicar em “Banco Online – Modelo de Design”.
  - No menu “editar->Localizar/Substituir”.
  - Substituir: `#{functional.area.impldesign}` para “BancoOnline.design”.

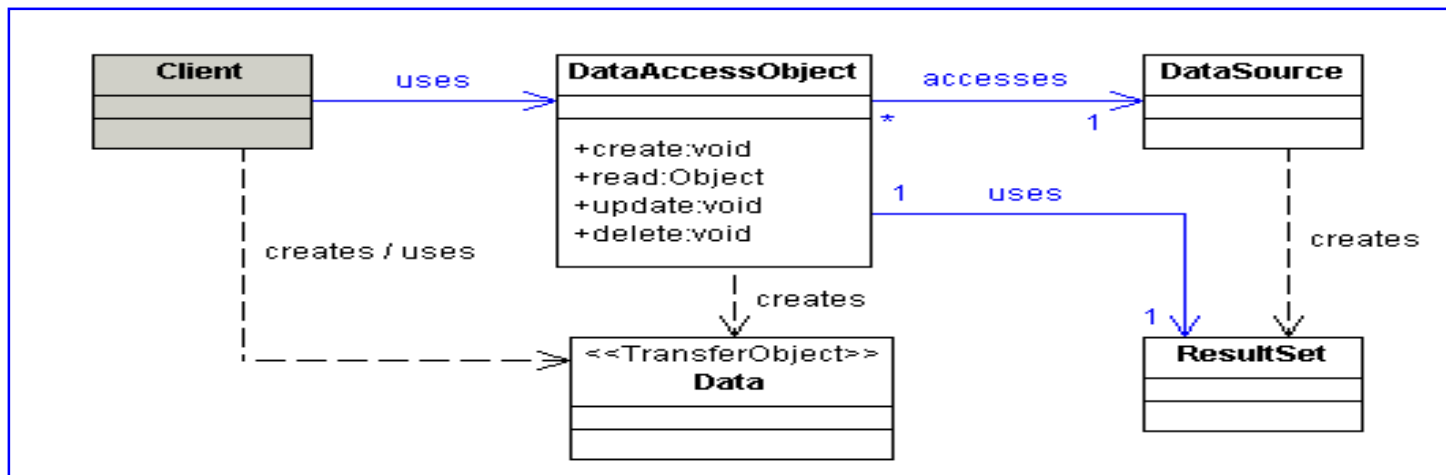


**Não esquecer de salvar!!!**

Prof. André Pereira, MSC, PMP

# Laboratório ENGSOFT

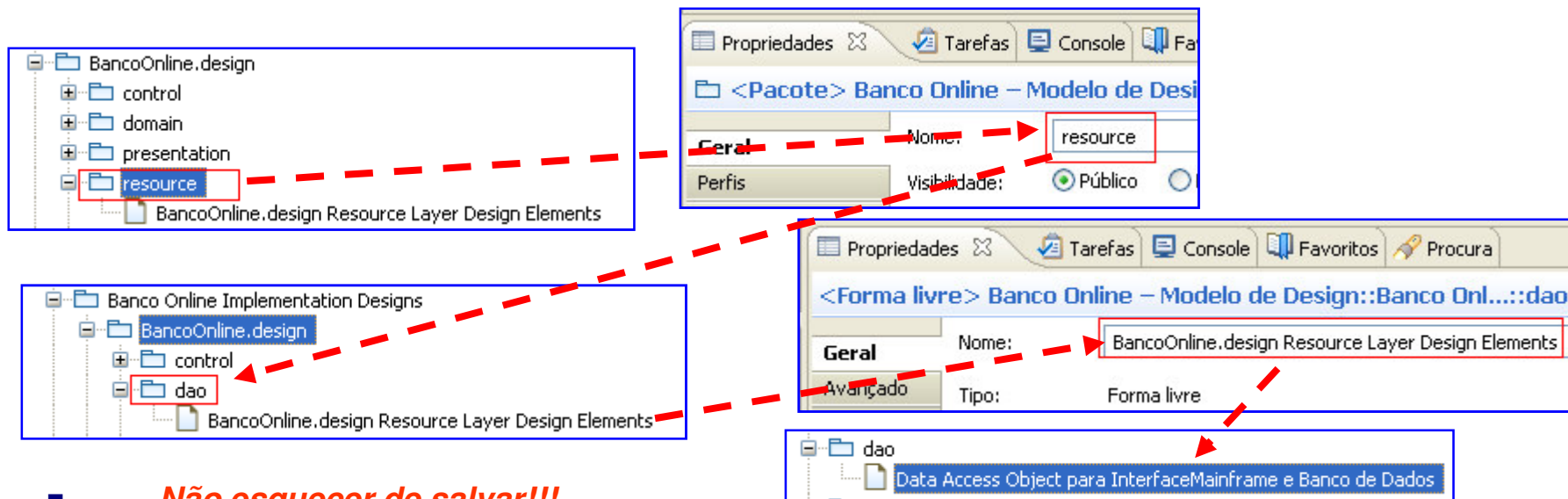
- Criando a camada do pacote “dao”.
- Essa camada deverá preparar-se para “encapsular” as classes responsáveis para acesso externo de dados, seja via interface remota, seja acesso a um banco.
- Essa é uma camada de integração que descreve como o sistema externo “mainframe” faz parte da transação descrita no “Banco Online”. Também, para o acesso aos dados locais (Em banco de dados)
- Será utilizado para isso o “design pattern” DAO:





# Laboratório ENGSOFT

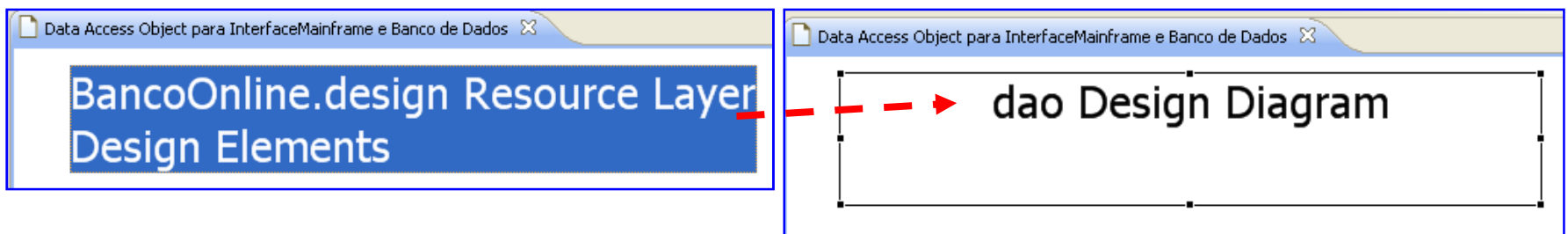
- Criando a camada “dao”.
- Utilização do Design Pattern DAO: “Data Access Object para InterfaceMainframe e Acesso a Dados”.
  - Expandir “Banco Online – Modelo de Design”, em “Banco Online Implementation Designs”, dê um duplo-clique no pacote “resource”. Em propriedades, alterar o nome “resource” para “dao”. Renomear o diagrama “BancoOnline.design Resource Layer Design Elements” para “Data Access Object para InterfaceMainframe e Banco de Dados”.



***Não esquecer de salvar!!!***

# Laboratório ENGSOFT

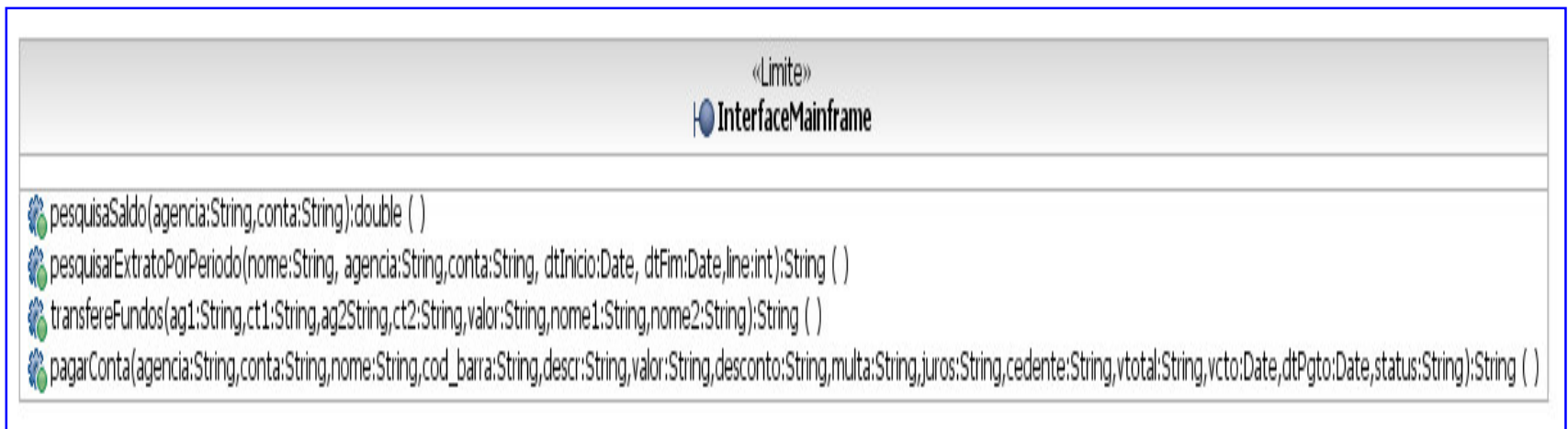
- Criando a camada “dao”.
- Utilização do Design Pattern DAO: “Data Access Object para InterfaceMainframe e Banco de Dados”.
  - Criação da classe: “BoDataAccessObject ”.
    - Abra o diagrama “Data Access Object para InterfaceMainframe e Banco de Dados”.
    - Renomear o diagrama “BancoOnline.design Resource Layer Design Elements” para “dao Design Diagram”.Dê um “F2” em cima do “label”.



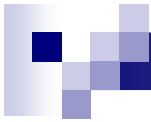
- ***Não esquecer de salvar!!!***

# Laboratório ENGSO

- Criando a camada “dao”.
- Utilização do Design Pattern DAO: “Data Access Object para InterfaceMainframe e Banco de Dados”.
  - Todas os diagramas da análise, utilizam-se da classe “InterfaceMainframe” com diversas operações”. Essa classe de análise, será transformada em uma classe cliente da “api” do mainframe. Vamos assumir que no mainframe exista uma Webservice bem definido.



- ***Não esquecer de salvar!!!***

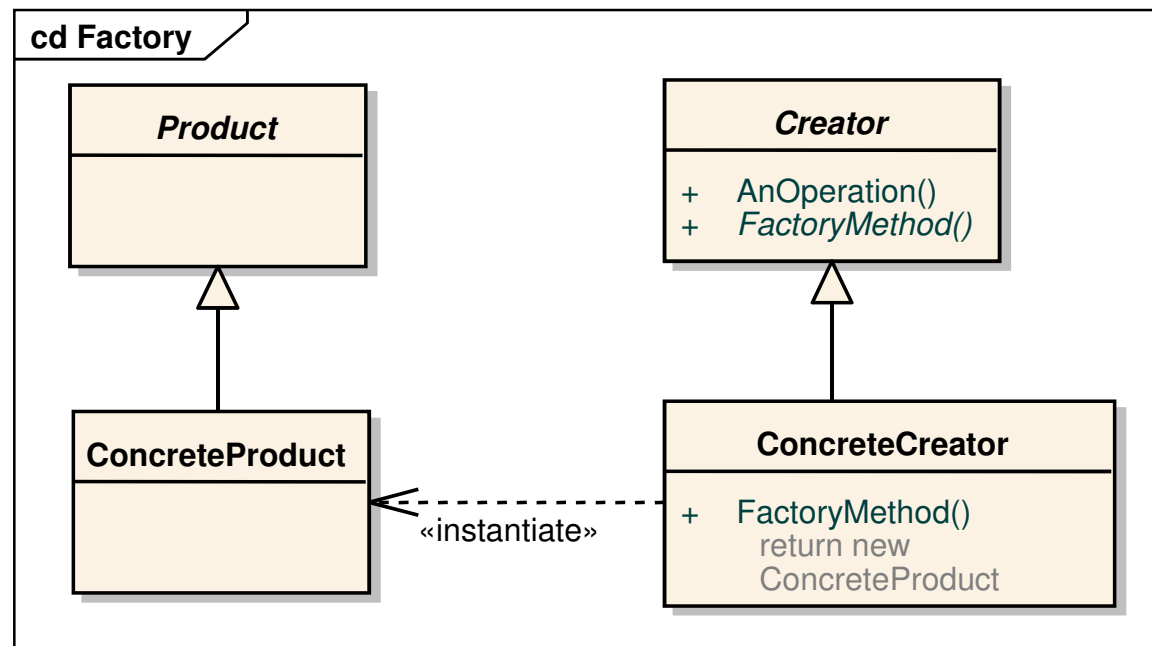


# Laboratório ENGSO

- Criando a camada “dao”.
- Utilização do Design Pattern DAO: “Data Access Object para InterfaceMainframe e Banco de Dados”.
  - Vamos fazer várias coisas:
    - Vamos implementar uma factory (Fabrica de DAOs)
    - Criar a classe de design “InterfaceMainframe” (<- um cliente web service)
    - Criar a classe DAO de acesso a um banco de dados.

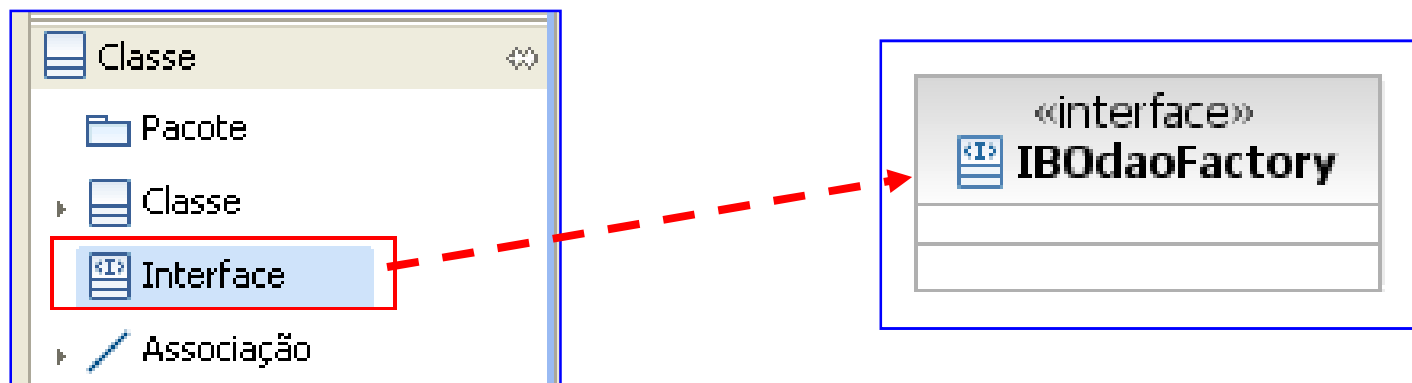
# Laboratório ENGSOE

- Criando a camada “dao”.
- Utilização do Design Pattern DAO: “Data Access Object para InterfaceMainframe e Banco de Dados”.
  - Vamos implementar uma factory (Fabrica de DAOs)



# Laboratório ENGSOB

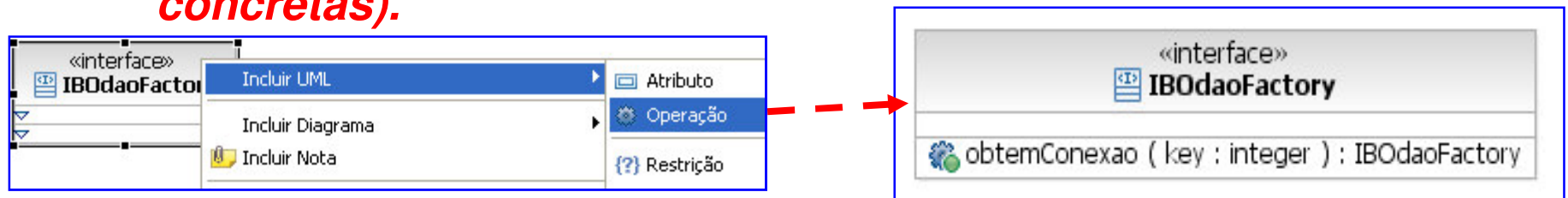
- Criando a camada “dao”.
- Utilização do Design Pattern DAO: “Data Access Object para InterfaceMainframe e Banco de Dados”.
  - Vamos implementar uma factory (Fabrica de DAOs)
  - Mantenha aberto o diagrama “Data Access Object para InterfaceMainframe e Banco de Dados”.
  - Clique na paleta “Interface” e arraste para o diagrama. De o nome de “IBOdaoFactory”



■ ***Não esquecer de salvar!!!***

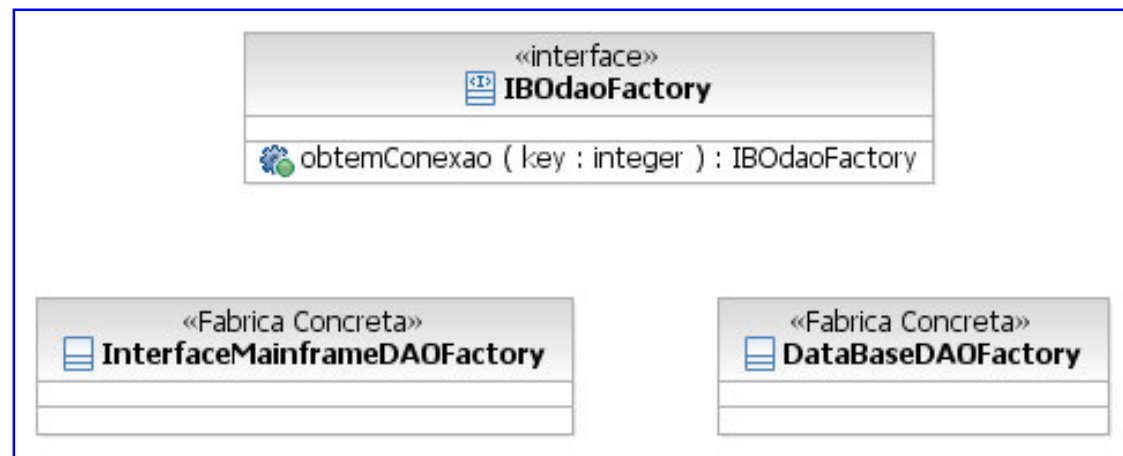
# Laboratório ENGSOFT

- Criando a camada “dao”.
- Utilização do Design Pattern DAO: “Data Access Object para InterfaceMainframe e Banco de Dados”.
- Clique em “IBOdaoFactory” com o botão direito e escolha “Incluir UML->Operação”. Coloque o nome da operação: “obtemConexao(key: int):IBOdaoFactory”.
- ***Obs: essa operação permitirá que qualquer classe obtenha um objeto do tipo “IBOdaoFactory” que poderá ser “InterfaceMainframeDAO” ou “DataBaseDAO” (que são classes concretas).***



# Laboratório ENGSOE

- Criando a camada “dao”.
- Utilização do Design Pattern DAO: “Data Access Object para InterfaceMainframe e Banco de Dados”.
- Clique na Paleta e escolha classe e arraste-a até o diagrama. Renomei-a para “InterfaceMainframeDAOFactory”.
- Repita para a classe “DataBaseDAOFactory”.
- Coloque os esteriótipos para ambas de “Fabrica Concreta”.

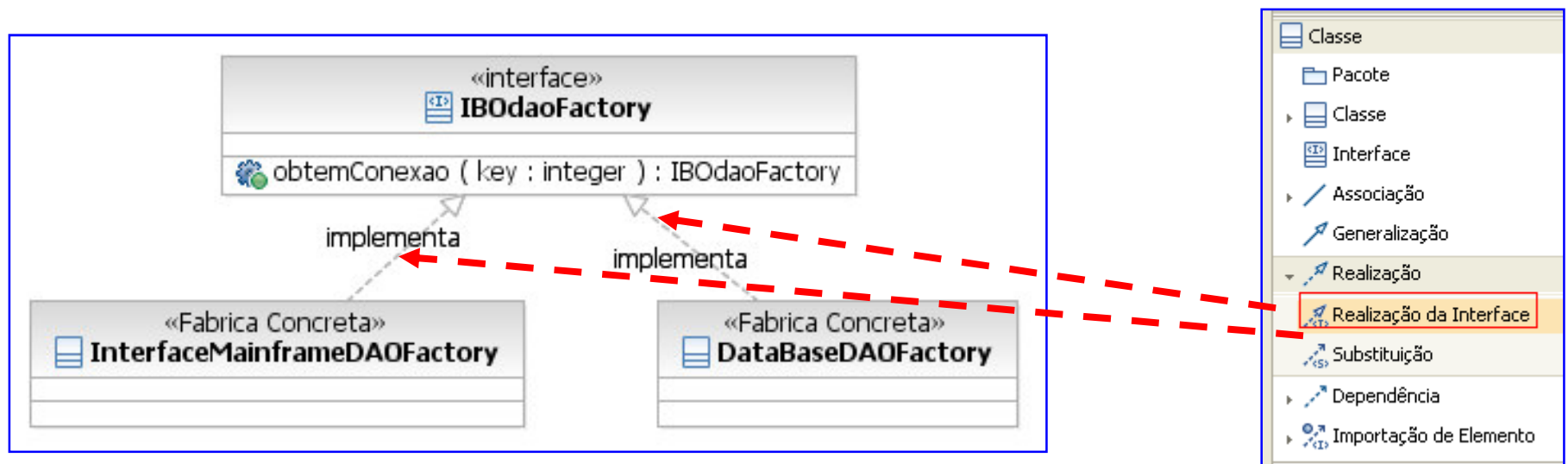


***Não esquecer de salvar!!!***



# Laboratório ENGSO

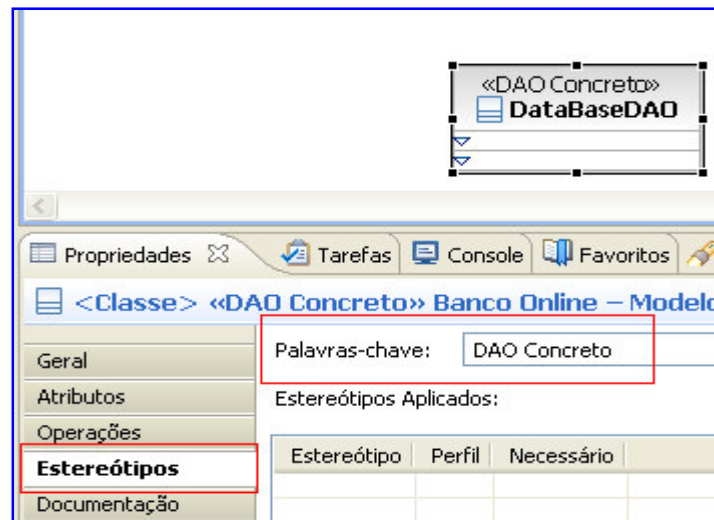
- Criando a camada “dao”.
- Utilização do Design Pattern DAO: “Data Access Object para InterfaceMainframe e Banco de Dados”.
- Clique na paleta, escolha “Realização da Interface” e clique nas classe concretas criadas e ligue cada uma com a interface. Coloque o label “Implementa”.



***Não esquecer de salvar!!!***

# Laboratório ENGSOE

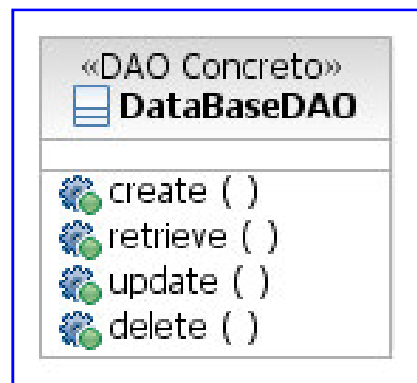
- Criando a camada “dao”.
- Utilização do Design Pattern DAO: “Data Access Object para InterfaceMainframe e Banco de Dados”.
- Vamos criar a classe de acesso a banco e fazer um CRUD (Create, Retrieve, Update e Delete).
- Clique na paleta “Classe” e arraste para o diagrama. De o nome de “DataBaseDAO”. Clique em “Esteriótipos” e coloque em palavra-chave: “DAO Concreto”.



■ **Não esquecer de salvar!!!**

# Laboratório ENGSO

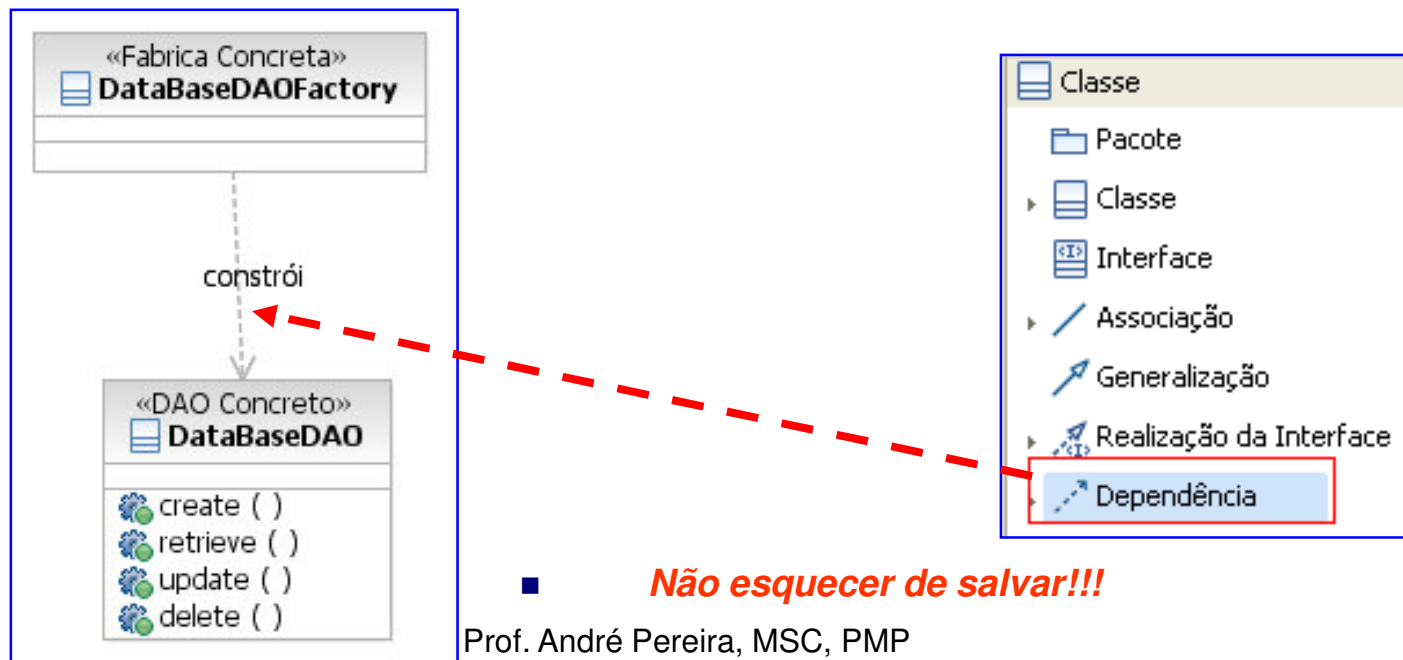
- Criando a camada “dao”.
- Utilização do Design Pattern DAO: “Data Access Object para InterfaceMainframe e Banco de Dados”.
- Clique na classe “DataBaseDAO” e insira 4 operações: “create, retrieve, update e delete”.



- ***Não esquecer de salvar!!!***

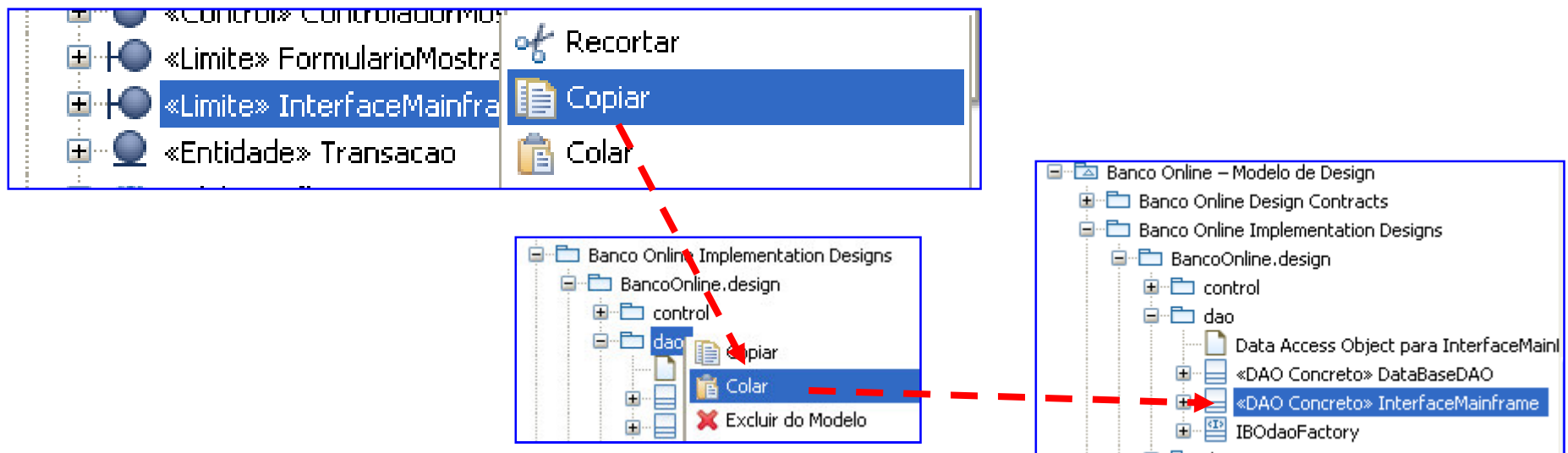
# Laboratório ENGSOB

- Criando a camada “dao”.
- Utilização do Design Pattern DAO: “Data Access Object para InterfaceMainframe e Banco de Dados”.
- Clique na paleta, escolha “dependência” e clique na classe concreta criada “DataBaseDAOFactory” e ligue em “DataBaseDAO”. Coloque o label “constroi”.



# Laboratório ENGSO

- Criando a camada “dao”.
- Utilização do Design Pattern DAO: “Data Access Object para InterfaceMainframe e Banco de Dados”.
- Copie de “Banco Online – Modelo de Análise” a classe “InterfaceMainframe” e cole dentro do pacote “dao”.



- **Não esquecer de salvar!!!**

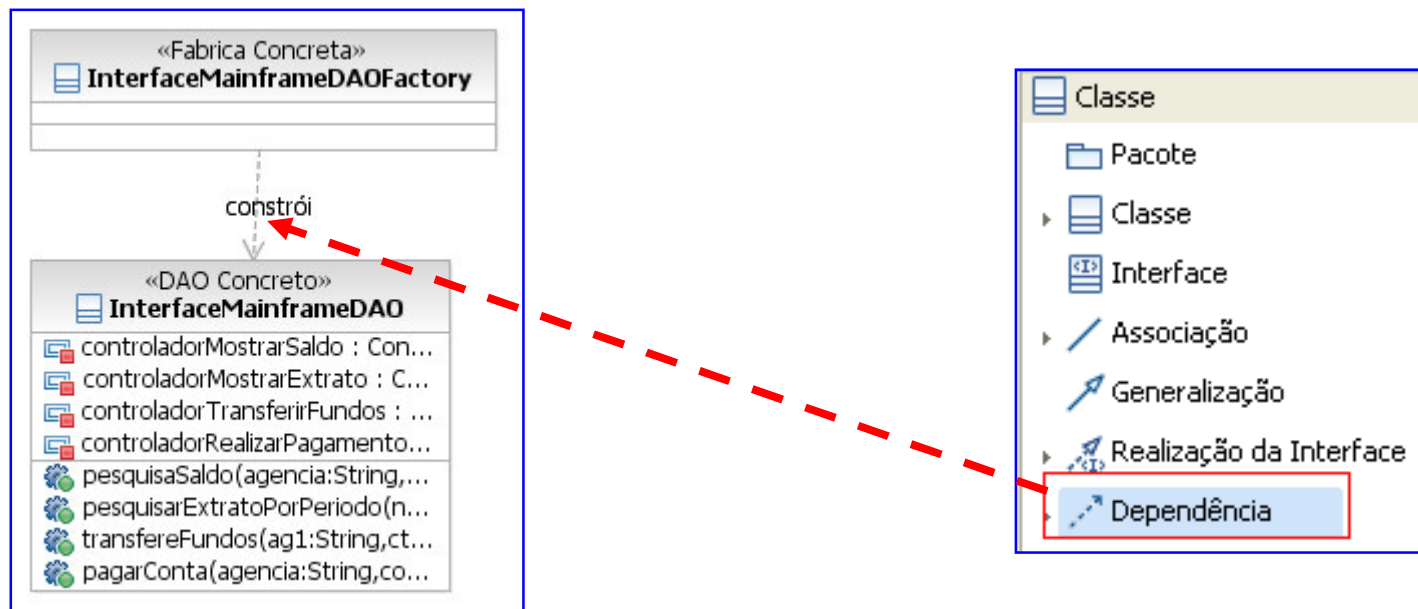
# Laboratório ENGSO

- Criando a camada “dao”.
- Utilização do Design Pattern DAO: “Data Access Object para InterfaceMainframe e Banco de Dados”.
- Arraste a classe “InterfaceMainframe” para dentro do diagrama. Mude o nome para “InterfaceMainframeDAO” e coloque o esteriótipo “DAO Concreto”.



# Laboratório ENGSOFT

- Criando a camada “dao”.
- Utilização do Design Pattern DAO: “Data Access Object para InterfaceMainframe e Banco de Dados”.
- Clique na paleta, escolha “dependência” e clique na classe concreta criada “InterfaceMainframeDAOFactory” e ligue em “InterfaceMainframeDAO”. Coloque o label “constroi”.

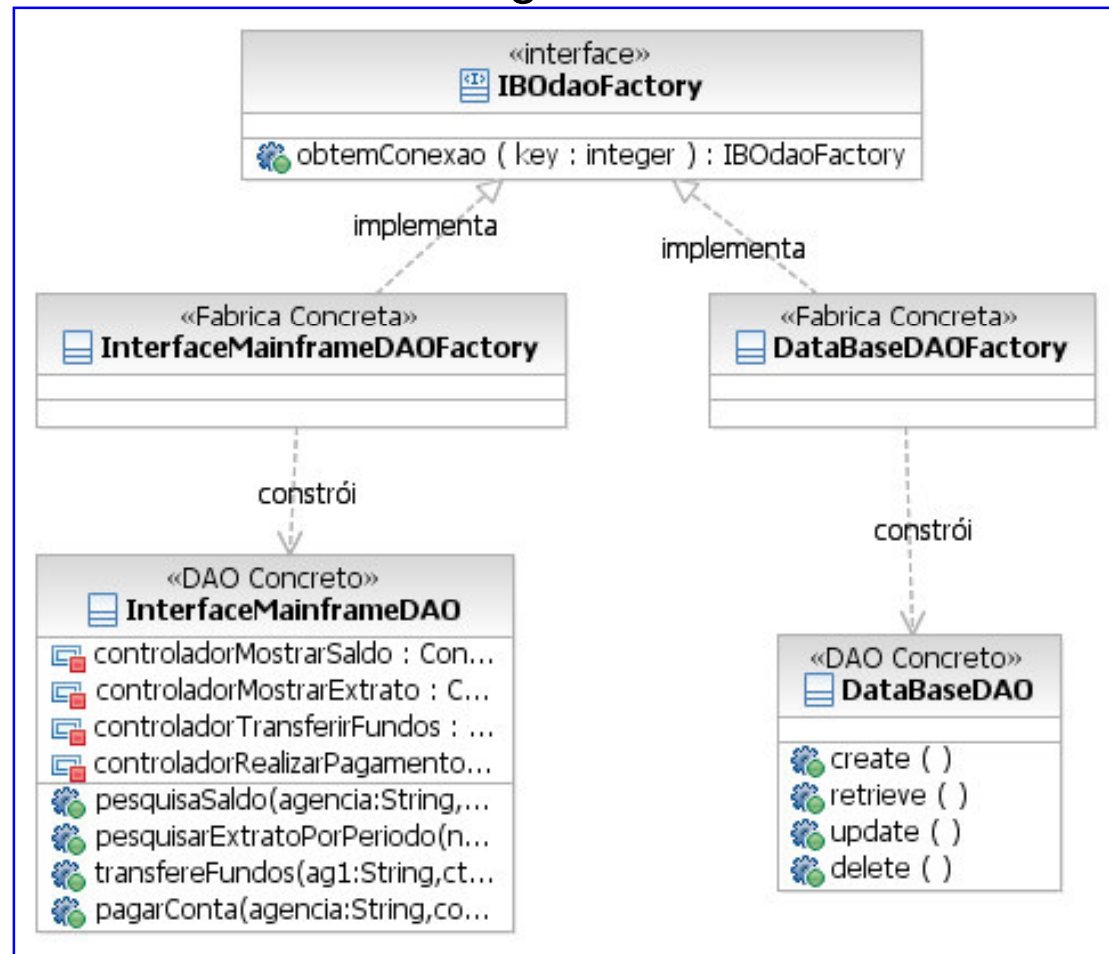


**Não esquecer de salvar!!!**

Prof. André Pereira, MSC, PMP

# Laboratório ENGSO

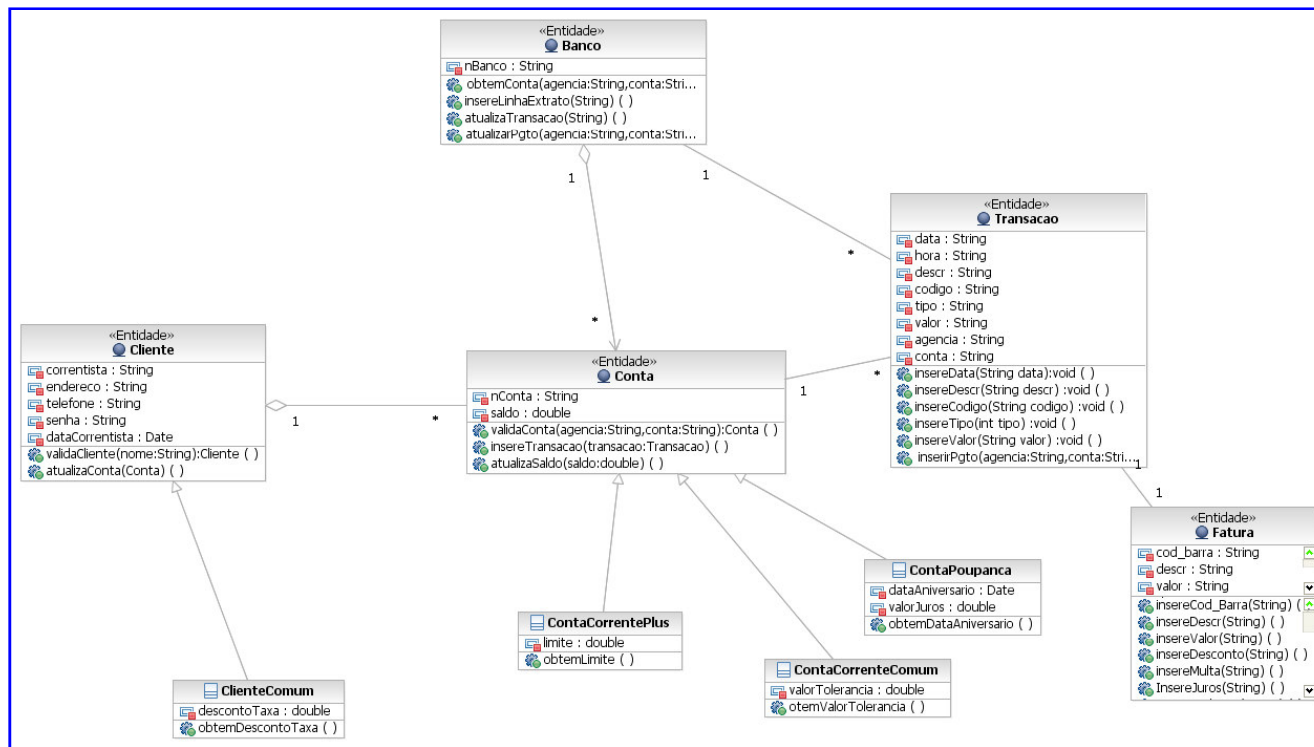
- Criando a camada “dao”. Diagrama final:





# Laboratório ENGSO

- Criando a camada de Negócio.
  - Basicamente, tem-se as seguintes classes de análise que correspondem à camada de negócio. Nesse diagrama corresponde ao diagrama de domínio mais completo que engloba todas as operações criadas.

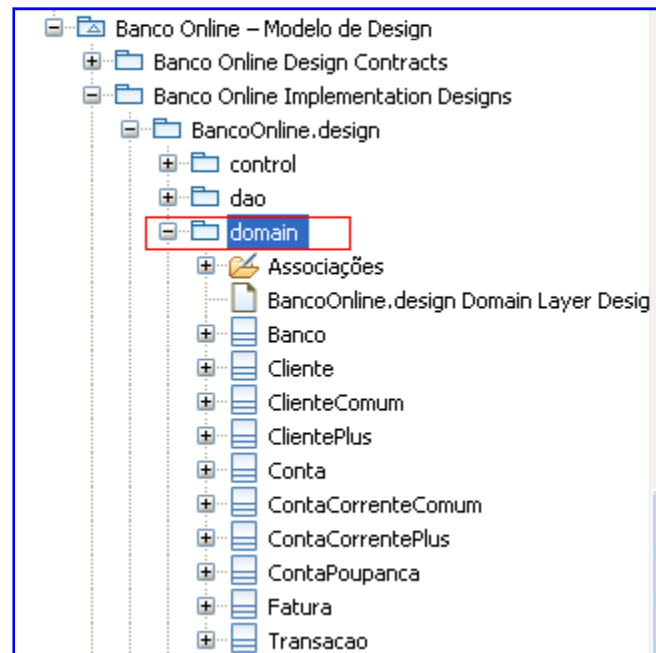


■ **Não esquecer de salvar!!!**

Prof. André Pereira, MSC, PMP

# Laboratório ENGSOFT

- Criando a camada de Negócio.
  - As classes de negócio podem ser modificadas para melhorar a manutenção e flexibilidade do sistema. Aplicaremos mais alguns Design Patterns.
  - Primeiramente, copiar todas as classes de domínio que foram criadas durante a análise para a pasta “domain” conforme figura abaixo.

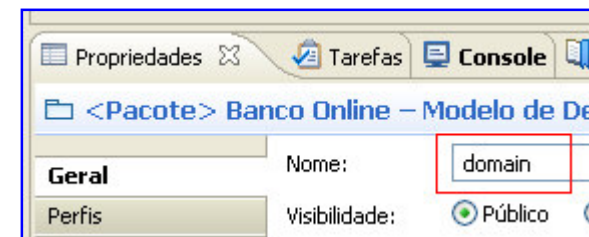
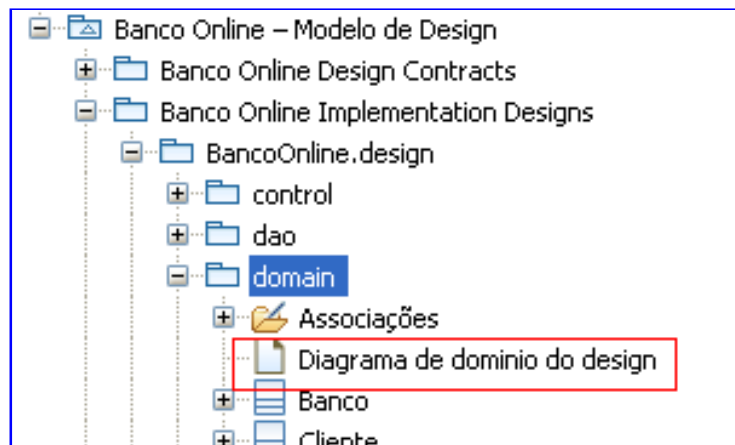


***Não esquecer de salvar!!!***

Prof. André Pereira, MSC, PMP

# Laboratório ENGSOFT

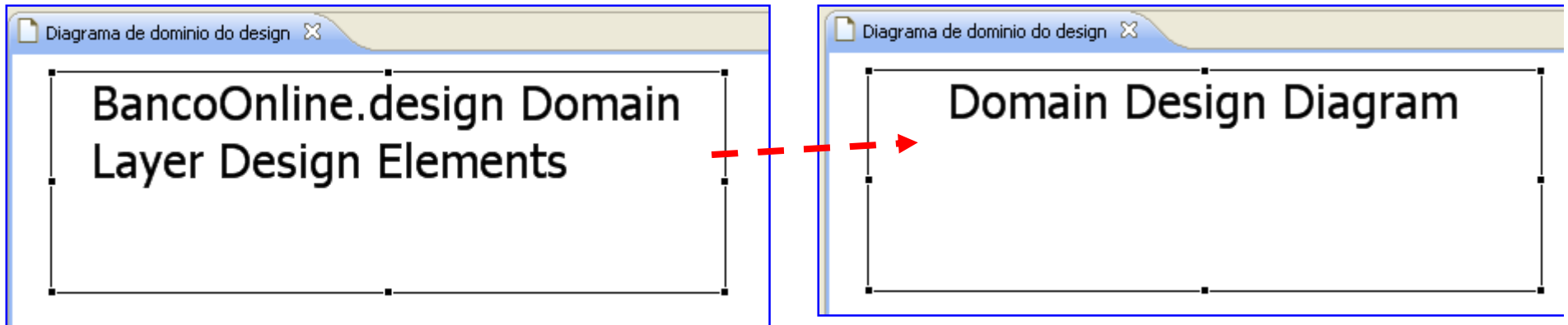
- Criando a camada de Negócio.
- Utilização do pacote “domain”.
  - Expandir “Banco Online – Modelo de Design”, em “Banco Online Implementation Designs”, renomear o diagrama “BancoOnline.design Resource Layer Design Elements” para “Diagrama de dominio do design”.



- ***Não esquecer de salvar!!!***

# Laboratório ENGSOE

- Criando a camada de Negócio.
- Aplicação de alguns patterns para arrumar a camada de negócio.
  - Abra o diagrama “Diagrama de dominio do design”.
  - Renomear o diagrama “BancoOnline.design Domain Layer Design Elements” para “Domain Design Diagram”.Dê um “F2” em cima do “label”.

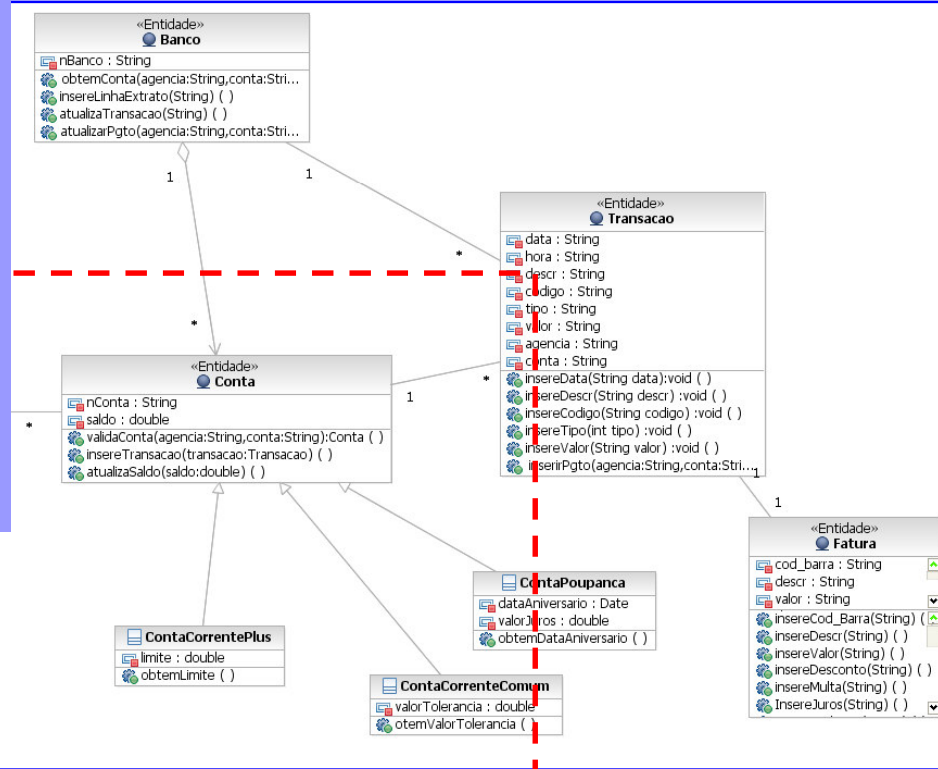


- ***Não esquecer de salvar!!!***

# Laboratório ENGSOFT

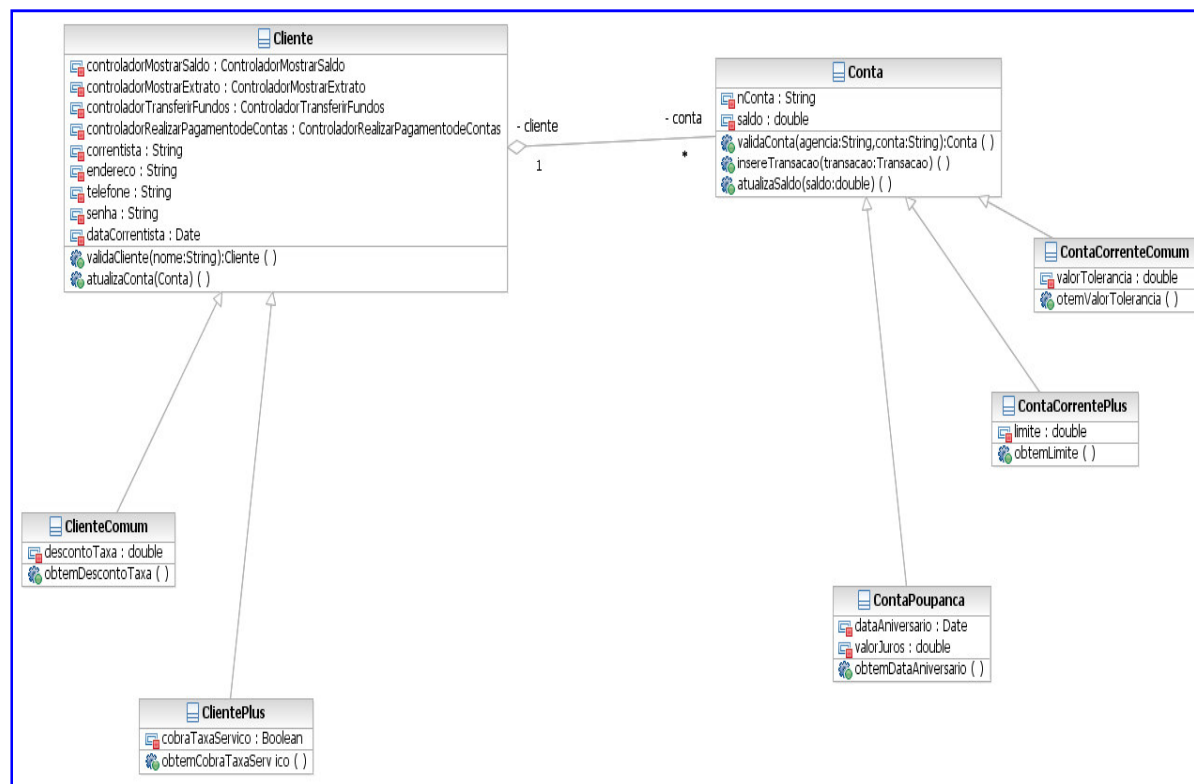
- Criando a camada de Negócio.
  - Analisemos a figura abaixo:

O relacionamento entre Cliente e Conta pode ser modificado para a aplicação do pattern Strategy (Para evitar incluir “ifs” desnecessários no código). Incluiremos também uma Fabrica para obter esses objetos!



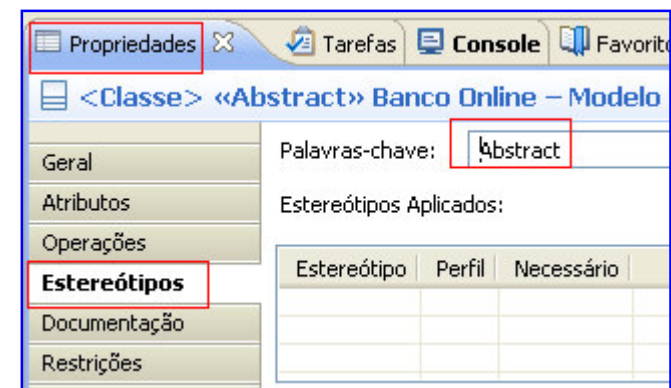
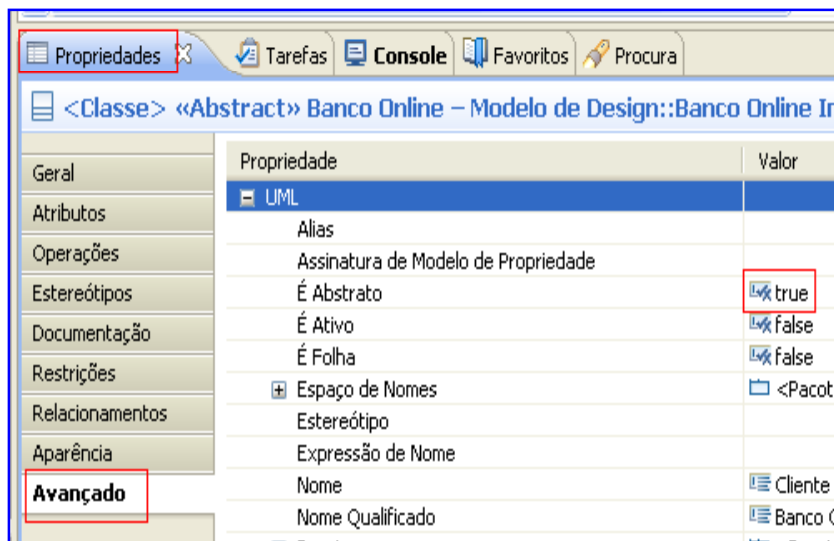
# Laboratório ENGSO

- Criando a camada de Negócio.
  - Inserir no diagrama as seguintes classes:



# Laboratório ENGSO

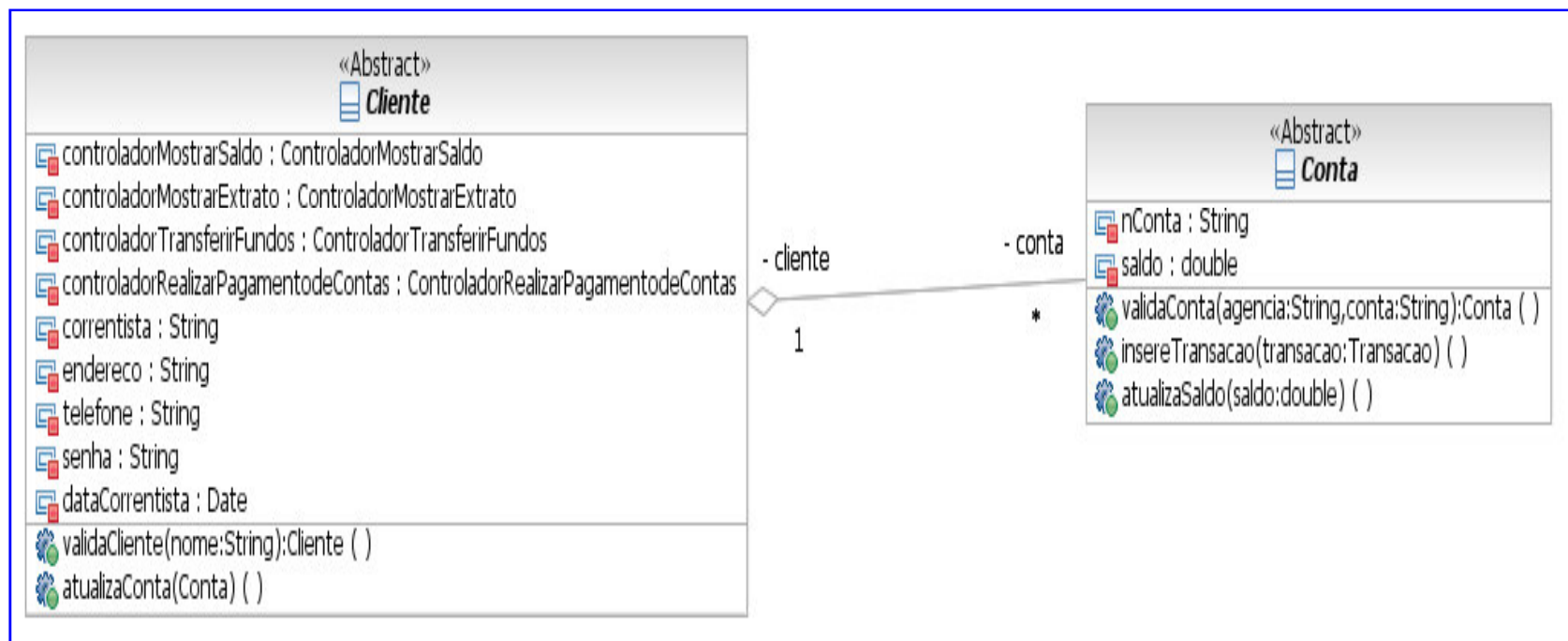
- Criando a camada de Negócio.
  - Modifique a classe Cliente e Conta para serem abstratas alterando em “Propriedades” as abas “Avançado” e “Esteriótipos” conforme abaixo:



- **Não esquecer de salvar!!!**

# Laboratório ENGSOB

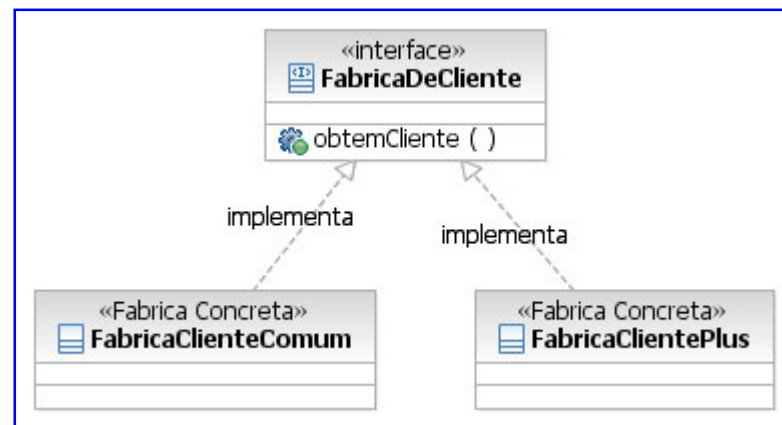
- Criando a camada de Negócio.
  - As classes ficarão assim:





# Laboratório ENGSOE

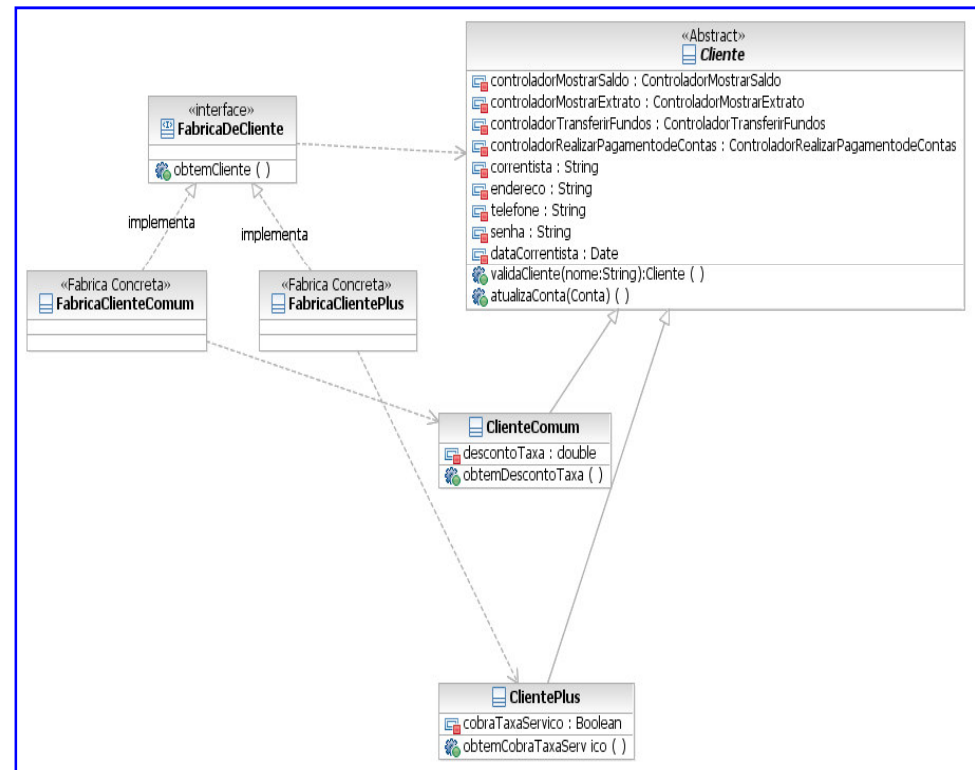
- Criando a camada de Negócio.
  - Iremos construir a Fabrica de clientes.
  - Inclua uma interface, coloque o nome de “FabricaDeCliente” e crie a operação “obtemCliente():Cliente”
  - Crie uma classe, coloque o nome de “FabricaClienteComum”. Coloque o esteriótipo de “Fabrica Concreta”.
  - Crie outra classe, coloque o nome de “FabricaClientePlus”. Coloque o esteriótipo de “Fabrica Concreta”.
  - Faça um relacionamento de “realização de interface” entre as mesmas e coloque o label “implementa”.



■ **Não esquecer de salvar!!!**

# Laboratório ENGSOFT

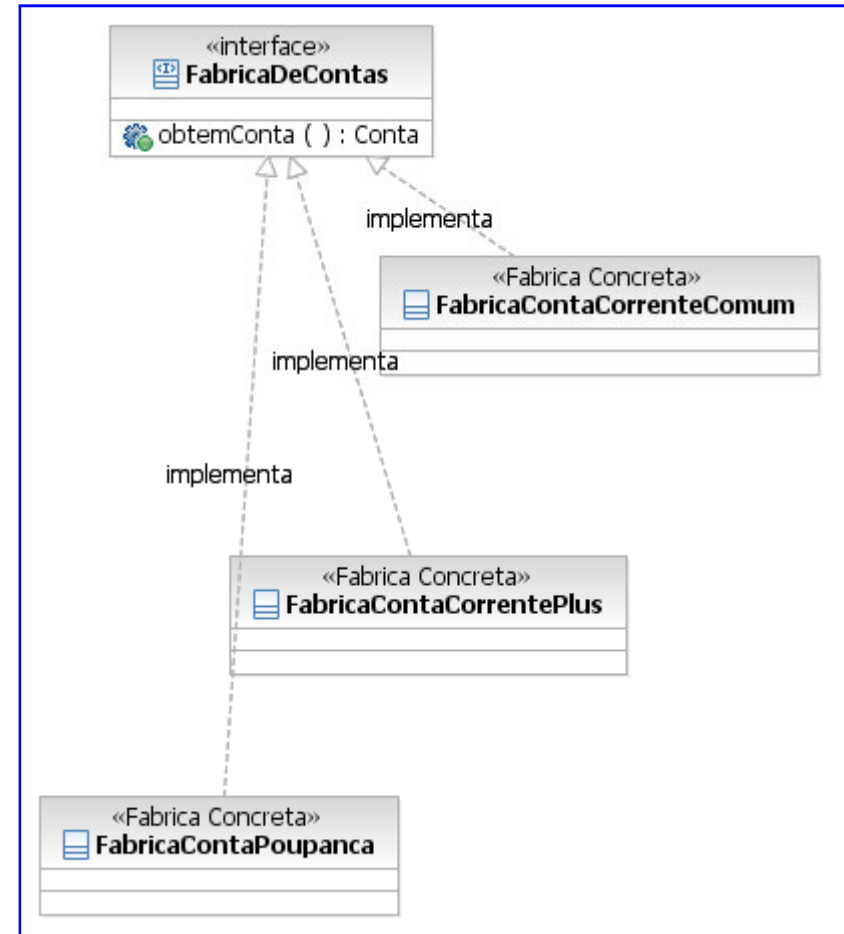
- Criando a camada de Negócio.
  - Iremos construir a Fabrica de clientes.
  - Clique na paleta o relacionamento de dependência e crie as seguintes dependências:
    - Entre “FabricaDeCliente” e “Cliente”
    - Entre “FabricaClienteComum” e “ClienteComum”
    - Entre “FabricaClientePlus” e “ClientePlus”



- **Não esquecer de salvar!!!**

# Laboratório ENGSOFT

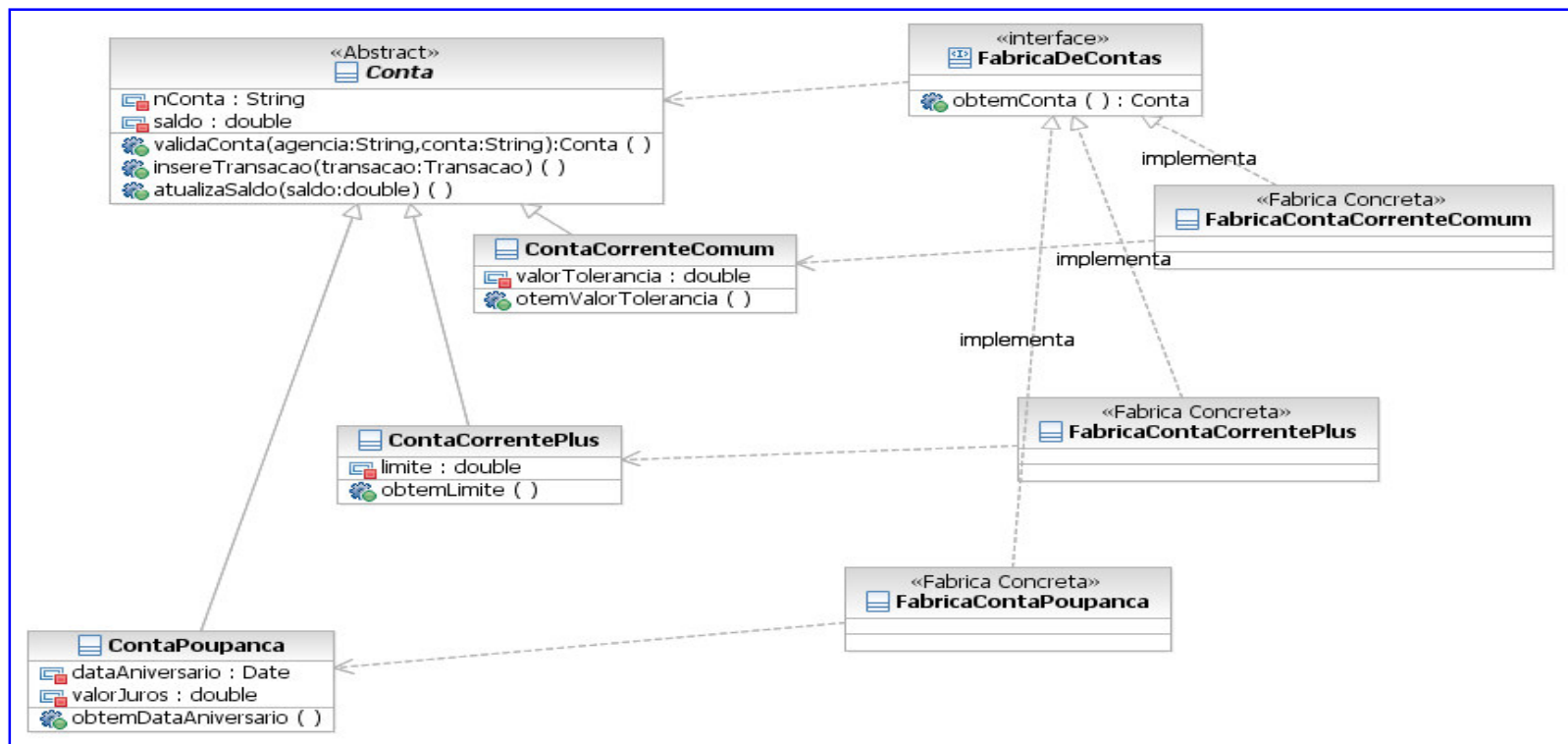
- Criando a camada de Negócio.
  - Iremos construir a Fabrica de Contas.
  - Inclua uma interface, coloque o nome de “FabricaDeContas” e crie a operação “obtemConta():Conta”
  - Crie uma classe, coloque o nome de “FabricaContaCorrenteComum”. Coloque o esteriótipo de “Fabrica Concreta”.
  - Crie outra classe, coloque o nome de “FabricaContaCorrenteComum”. Coloque o esteriótipo de “Fabrica Concreta”.
  - Crie outra classe, coloque o nome de “FabricaContaPoupanca”. Coloque o esteriótipo de “Fabrica Concreta”.
  - Faça um relacionamento de “realização de interface” entre as mesmas e coloque o label “implementa”.



- **Não esquecer de salvar!!!**

# Laboratório ENGSO

- Criando a camada de Negócio.
  - Iremos construir a Fabrica de clientes.
  - Clique na paleta o relacionamento de dependência e crie as seguintes dependências:
    - Entre “FabricaDeContas” e “Conta”
    - Entre “FabricaDeContaCorrenteComum” e “ContaCorrenteComum”
    - Entre “FabricaDeContaCorrentePlus” e “ContaCorrentePlus”
    - Entre “FabricaDeContaPoupanca” e “ContaPoupanca”

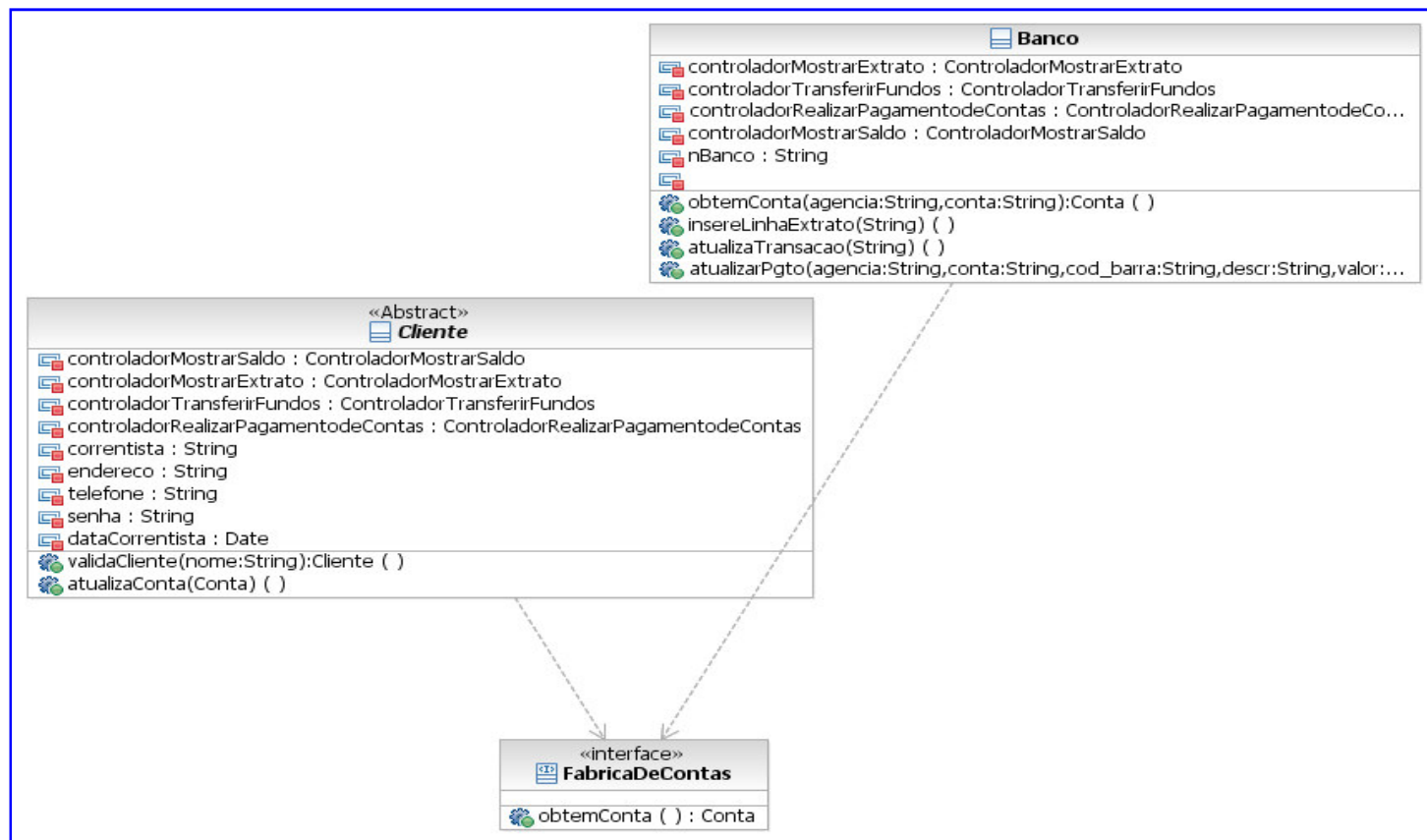


**Não esquecer de salvar!!!**

Prof. André Pereira, MSC, PMP

# Laboratório ENGSO

- Criando a camada de Negócio.
  - Completando o diagrama.
  - Inserir a classe “Banco”.
  - Criar dependências entre “Cliente->FabricaDeContas” e “Banco->FabricaDeContas”.

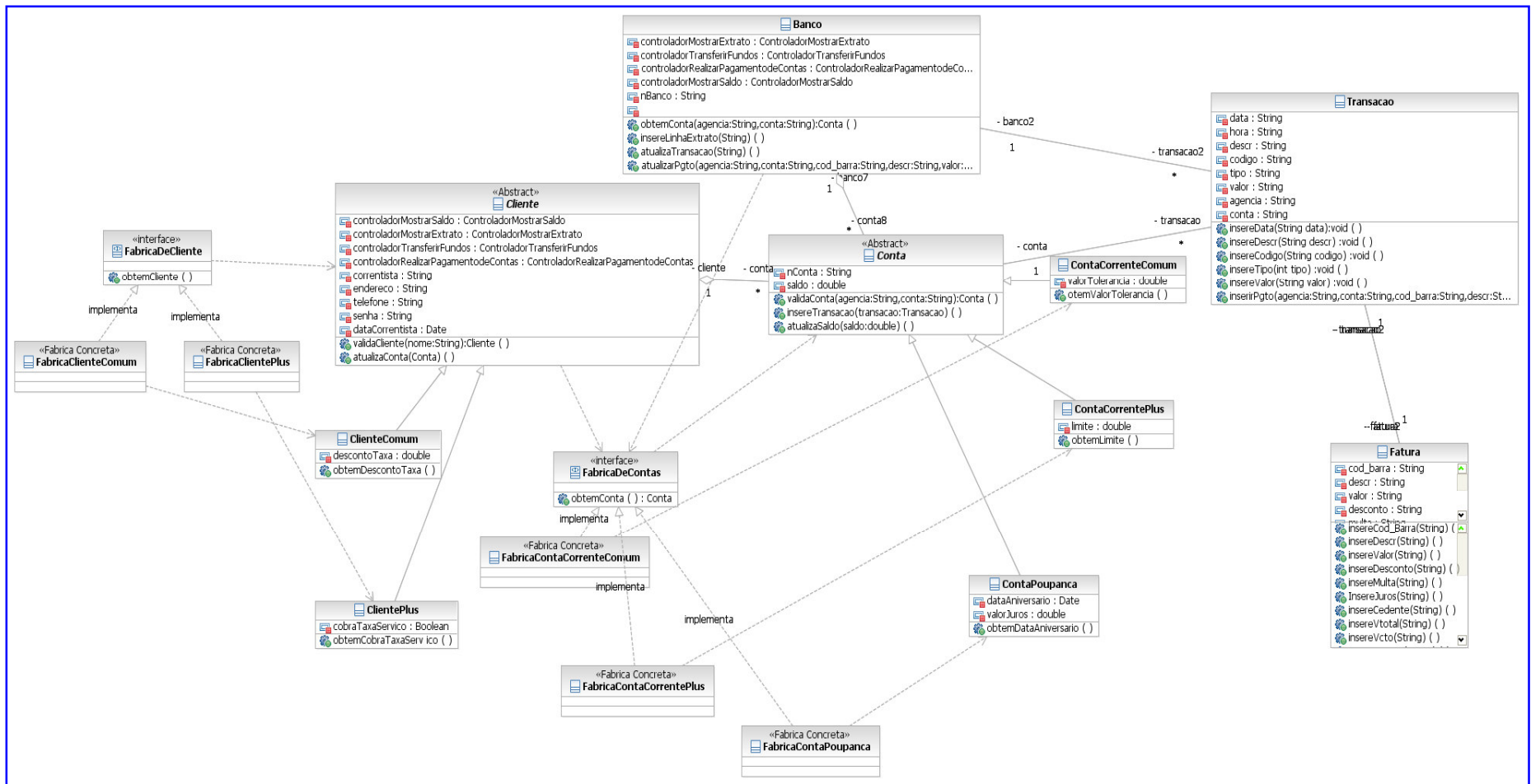


**Não esquecer de salvar!!!**

# Laboratório ENGSO

■ Criando a camada de Negócio.

- Completando o diagrama.
- Inserir as classes faltantes (Transacao e Fatura).



■ **Não esquecer de salvar!!!**

Prof. André Pereira, MSC, PMP



# Laboratório ENGSO

- O que você aprendeu até aqui?
- Qual é a diferença entre modelagem de classes de análise e modelagem de classes de design?
- Percebeu o beneficio de design patterns aplicados ao dominio?
- Próximos passos:
  - Aplicação de design patterns na lógica de aplicação: MVC
  - Desenho do diagrama de componentes
  - Desenho do diagrama de deploy/físico.