



**PROTÓTIPO DE SISTEMA PARA ACESSO A BANCO
DE DADOS DISTRIBUÍDOS E HETEROGÊNEOS EM
REDES TCP/IP**

Orientando: Ademir José Fink

Orientador : Marcel Hugo



Agenda

- Introdução;
- Banco de Dados:
 - Conceito;
 - Bancos de Dados Distribuídos;
 - Arquitetura Cliente/Servidor;
- Java:
 - Características;
 - Threads;
 - Sockets TCP/IP;
- JDBC:
 - Conceito;
 - Tipos de Driver's;
- Desenvolvimento do Protótipo:
 - Ferramentas e Metodologias utilizadas;
 - Casos de Uso;
 - Diagrama de Classes
 - Diagrama de Seqüência;
 - Modelo Entidade Relacionamento;
 - Funcionamento do Protótipo;
- Conclusões;



Introdução

- É comum encontrar as informações dispersas;
- Tecnologia de diversos fornecedores;
- Aproveitamento de tecnologia existente;
- Dificuldade de acessar informações distribuídas;
- Objetivo do Trabalho:
 - Especificação e Implementação de uma camada de software intermediária;
- Transparência tanto ao tipo de SGBD quanto a sua distribuição;



Banco de Dados

- Manter informações;
- Tornar informação disponível quando solicitada;
- SGBD é o software que manipula os acessos;



Banco de Dados Distribuído

- Múltiplas Localidades interligadas por redes de comunicação;
- Transparência de Localização;
- Todos os SGBD idênticos o sistema é homogêneo, senão é heterogêneo;



Vantagens e Desvantagens de BDD

- Vantagens:
 - Tempo de resposta para acessos locais;
 - Compartilhamento;
 - Autonomia local;
- Desvantagens:
 - Heterogeneidade Global;
 - Introdução de Padrões Globais;
 - Complexidade;

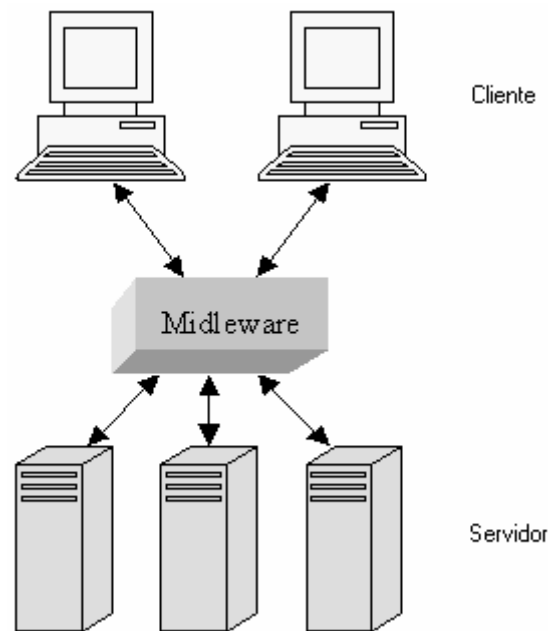


Arquitetura Cliente/Servidor

- Separa o processo em plataformas independentes podendo ou não estar na mesma máquina física;
- Permite máximo benefício no compartilhamento de recursos;
- Existe Processo Cliente e Processo Servidor que podem ser distinguidos um do outro mas podem interagir totalmente;
- Servidor pode atender vários Clientes simultaneamente e Cliente pode acessar a vários Servidores;

Sistema de Três Camadas

- Cliente conecta-se com camada de software intermediária;





Transações

- Finalidade do Sistema de Banco de Dados Distribuído é realizar Transações;
- Transação é uma unidade de trabalho;
- É a execução de uma seqüência de comandos que termina com um Commit ou Rollback;
- Mecanismo Two-Phase Commit:
 - 1 – Tenta executar o comando;
 - 2 – Se todos OK então Commit senão Rollback;



Java

- Ambiente de programação de propósito geral;
- Orientada a Objetos;
- Desenvolvida para ter as redes com ambiente operacional;
- Independente de Plataforma;
- Multi-Thread;
- Possui a API JDBC;



Thread

- Linha de execução que roda dentro de um processo;
- Um Processo pode suportar muitas Threads simultaneamente;
- Cada Thread é independente;



Sockets TCP/IP

- TCP/IP:

- Protocolo de comunicação de redes
- Grande portabilidade;
- Independência de hardware;
- Facilidade no desenvolvimento de aplicações;

- Socket:

- Ponto de referência para enviar e receber mensagens;
- Conexão persistente altamente confiável entre Cliente e Servidor;

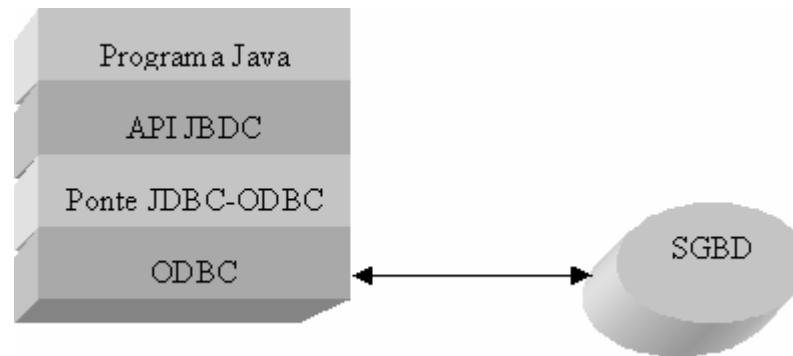


JDBC

- Interface de programação de aplicação exclusiva do Java;
- Criada para executar comandos SQL em SGBD;
- Suporte a vários SGBD;
- Necessário apenas carregar Driver específico;
- Driver é um Programa que faz a intermediação entre um Programa Java e um SGBD;

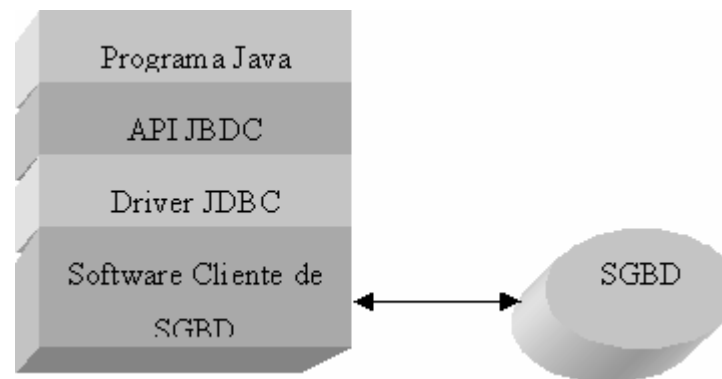
Ponte JDBC-ODBC com Driver ODBC

- Acesso ao SGBD via um *driver* ODBC convencional;



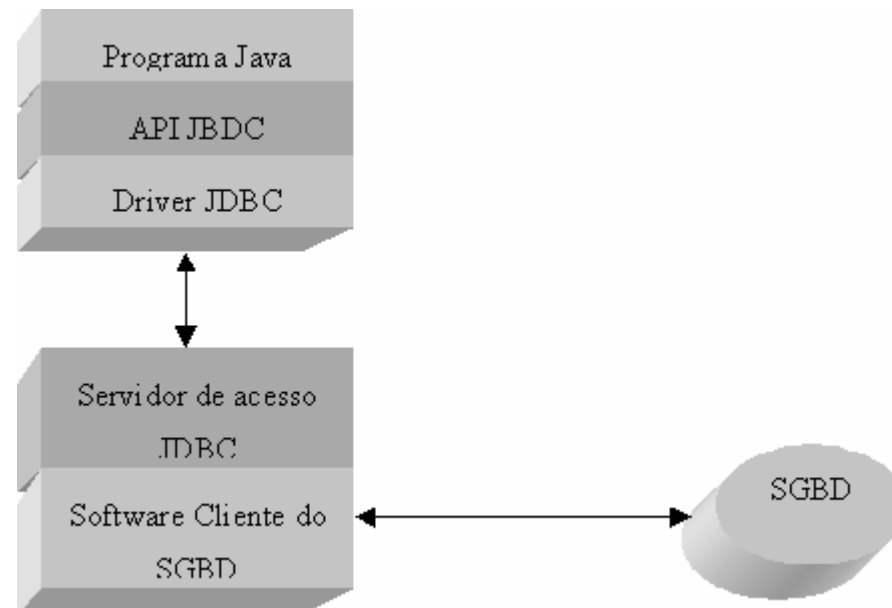
Driver com API Parcialmente Nativa

- O Driver JDBC acessa um programa cliente do SGBD



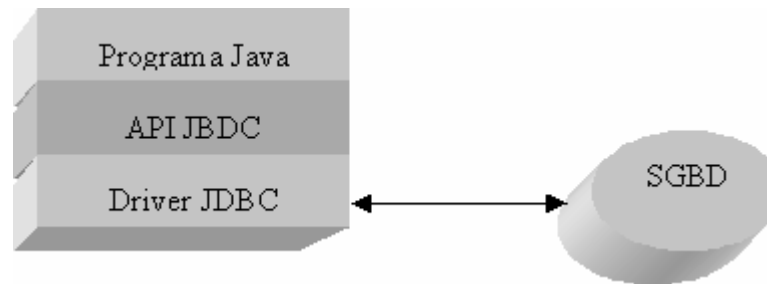
Driver Puro-Java JDBC-Rede

- O Driver acessa um servidor usando um protocolo neutro;



Driver Puro-Java com Protocolo Nativo

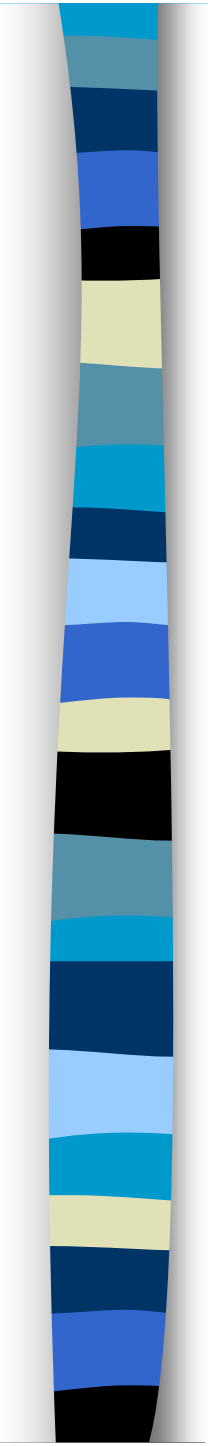
- O Driver acessa diretamente o SGBD usando o protocolo nativo do mesmo;





Ferramentas e Metodologias Utilizadas

- Ferramentas:
 - JBuilder 2.0;
 - Oracle 8i;
 - Interbase 5.5;
 - Access 7.0;
- Metodologias:
 - UML;
 - Modelo Entidade Relacionamento;

- 
- Tendo em vista que as informações estão distribuídas em diferentes servidores e diferentes SGBD precisamos de um Sistema para realizar a conexão com os SGBD;
 - Alguém precisa gerenciar a distribuição dos SGBD;

Casos de Uso

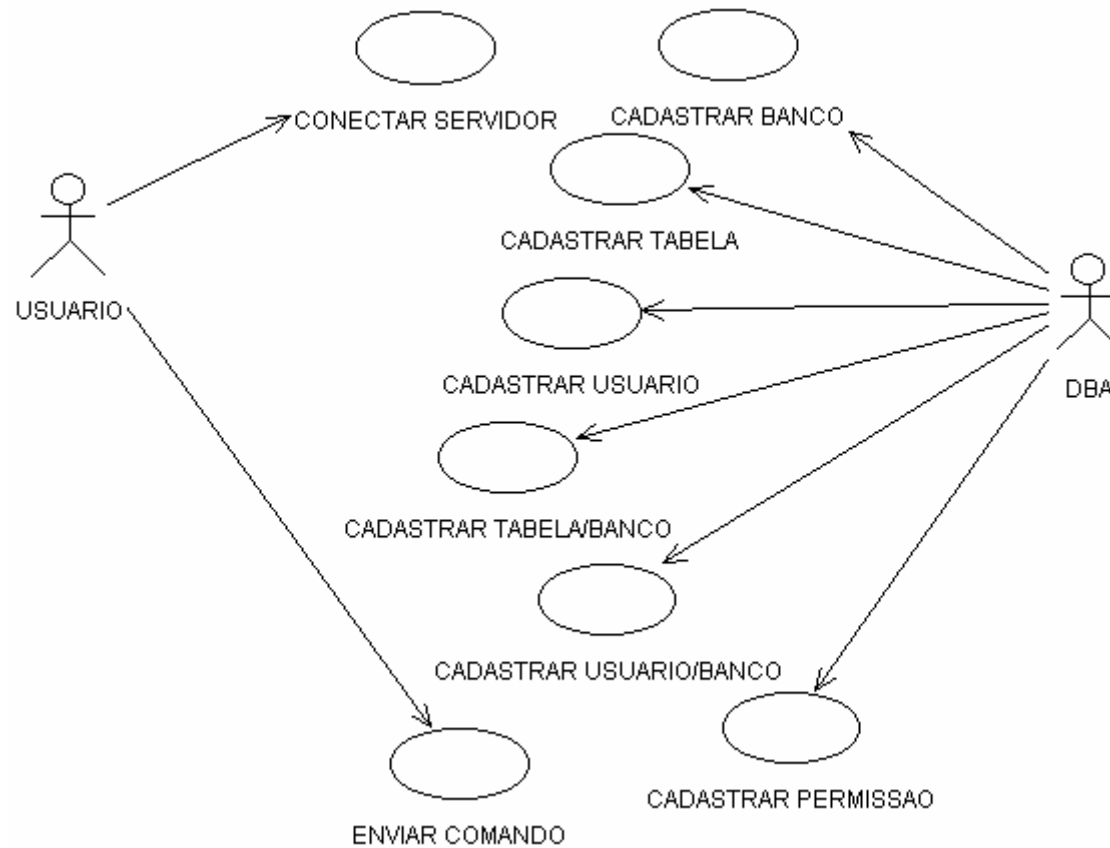
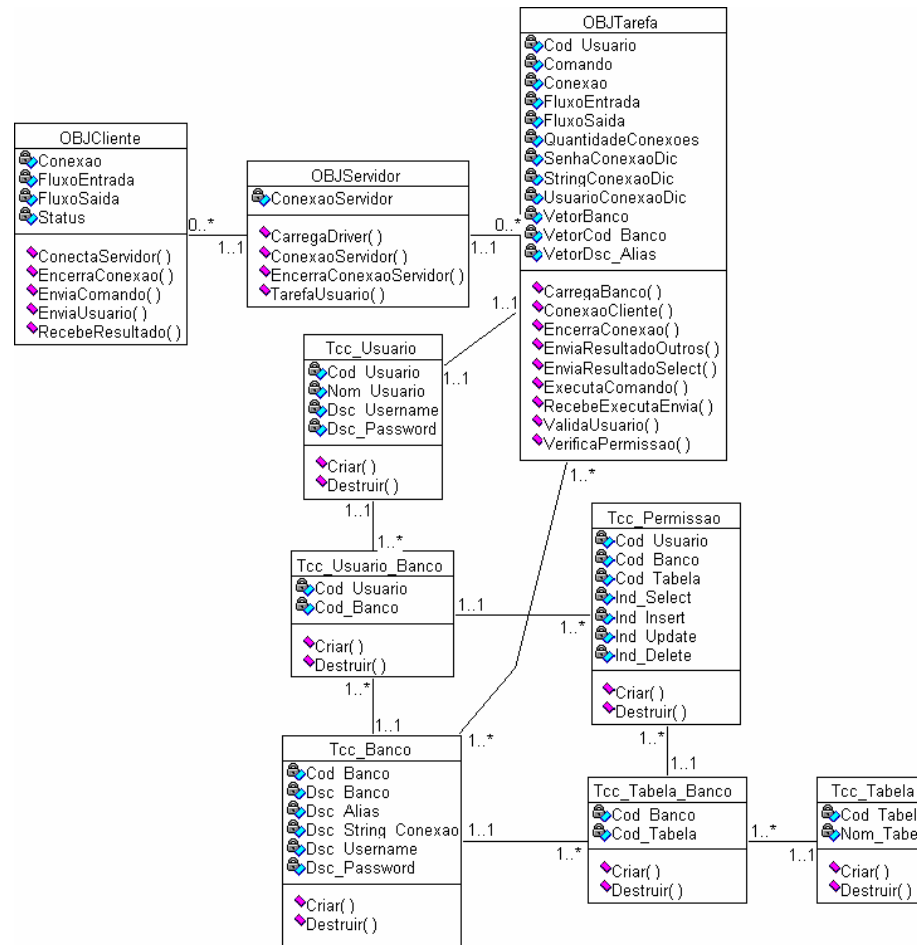
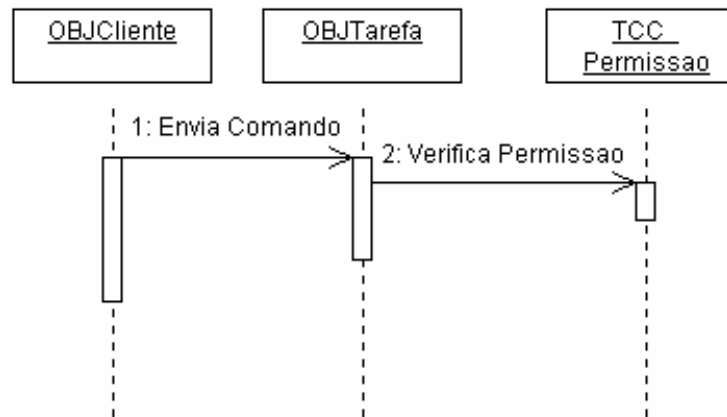
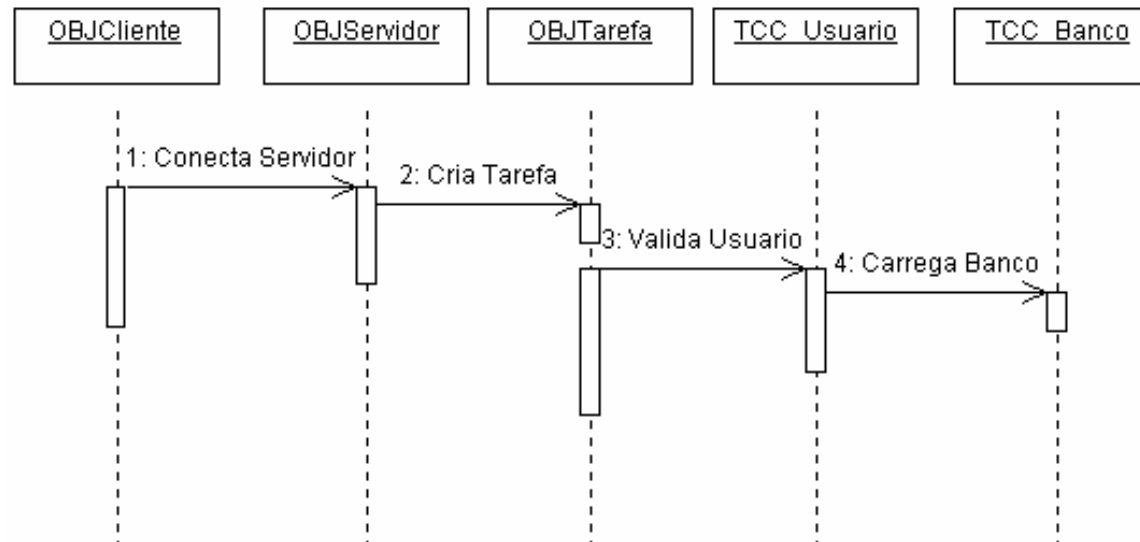


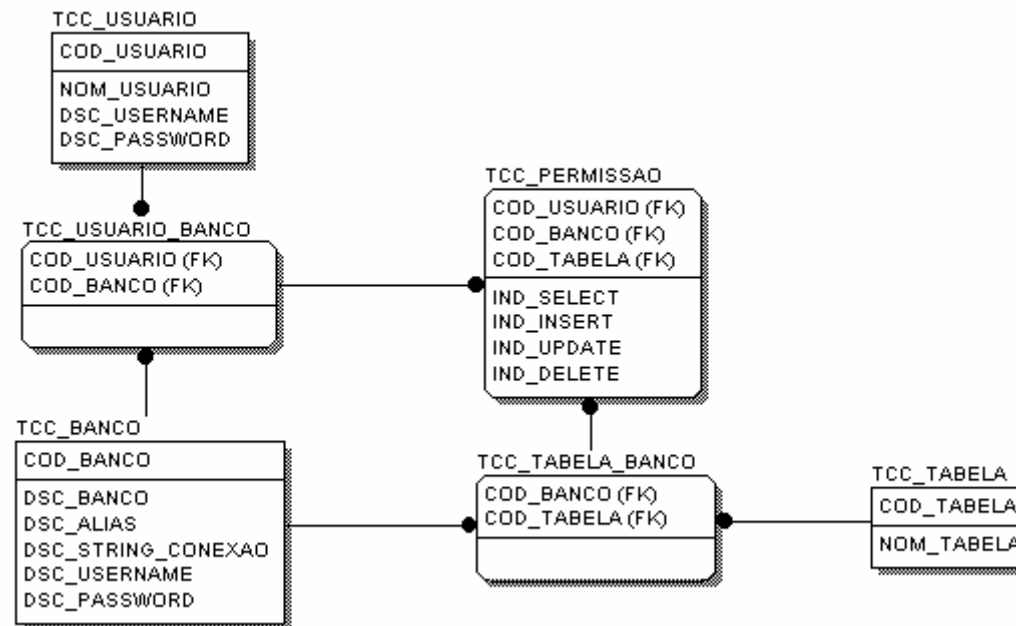
Diagrama de Classes



Diagramas de Seqüência



Modelo Entidade Relacionamento





Funcionamento do Protótipo

- Três Classes:
 - OBJCliente;
 - OBJServidor;
 - OBJTarefa;



OBJCliente

- Responsável pela conexão com o OBJServidor;
- Interface com o usuário;
- Envia Comandos;
- Exibe Resultados;



OBJServidor

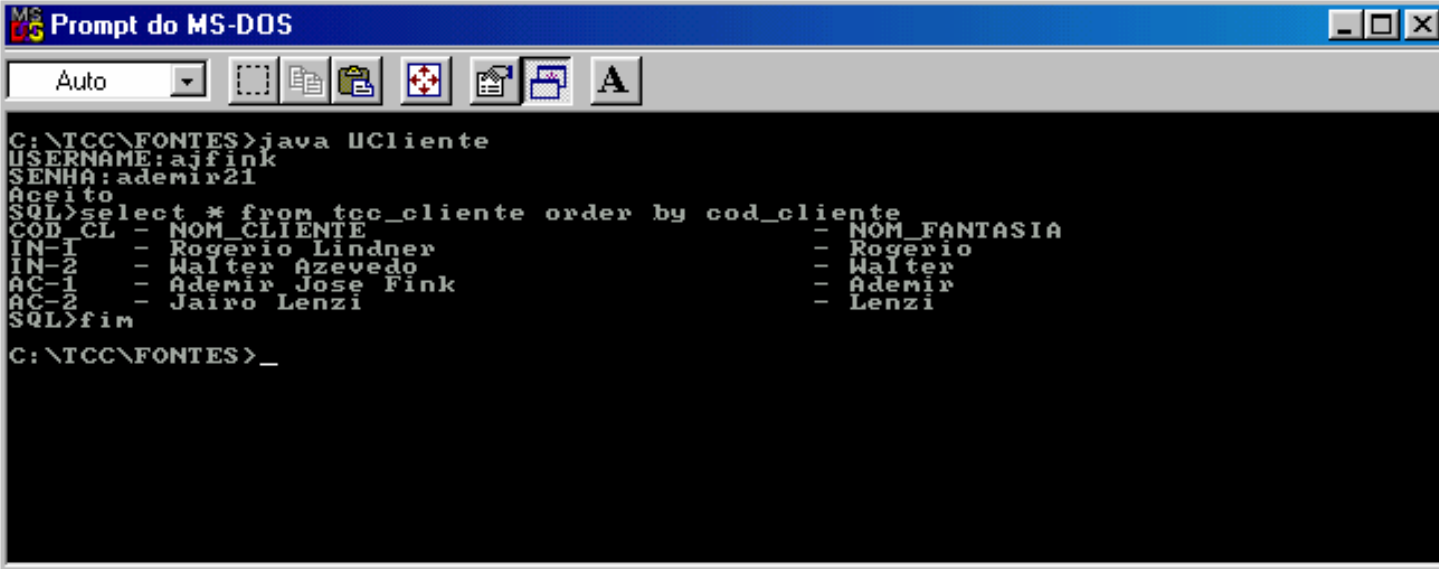
- Monitora a Porta de comunicação aguardando conexões do OBJCliente;
- Para cada conexão é instanciada um novo OBJTarefa que passa a manter e gerenciar a conexão;
- Não possui Interface;



OBJTarefa

- Recebe a conexão com o OBJCliente;
- Valida o Usuário;
- Estabelece a conexão com os SGBD que o Usuário esta cadastrado;
- Recebe comandos do Usuário;
- Valida o comando do Usuário;
- Verifica as permissões do Usuário em relação ao comando enviado;
- Executa o comando;
- Faz o controle da transação (two-phase commit);
- Envia o resultado para o OBJCliente;

Interface do OBJCliente



```
MS-DOS Prompt do MS-DOS
Auto
C:\TCC\FONTES>java UCliente
USERNAME:ajfink
SENHA:ademir21
Aceito
SQL>select * from tcc_cliente order by cod_cliente
COD_CL - NOM_CLIENTE - NOM_FANTASIA
IN-1 - Rogerio Lindner - Rogerio
IN-2 - Walter Azevedo - Walter
AC-1 - Ademir Jose Fink - Ademir
AC-2 - Jairo Lenzi - Lenzi
SQL>fim
C:\TCC\FONTES>_
```



Conclusões

- Objetivos foram alcançados;
- Manutenção do Dicionário de Distribuição a qualquer momento;
- Facilidades propiciadas pelo Java e pela API JDBC;
- Popularidade do Protocolo TCP/IP;