

# AGENTES MÓVEIS APLICADOS À INTEGRAÇÃO DE BANCO DE DADOS

Daniela Barreiro Claro  
[danclaro@inf.ufsc.br](mailto:danclaro@inf.ufsc.br)

João Bosco M. Sobral  
[bosco@inf.ufsc.br](mailto:bosco@inf.ufsc.br)

Universidade Federal de Santa Catarina  
Departamento de Informática e Estatística – INE  
Curso de Pós-Graduação em Ciência da Computação - CPGCC  
Florianópolis – Santa Catarina  
Brasil

## RESUMO

Atualmente, diversas tecnologias de armazenamento de informações estão sendo desenvolvidas e utilizadas pelas organizações. Porém, os bancos de dados existentes nestas organizações não podem ser desprezados a curto prazo e, então, surge a necessidade de integrar diversas bases de dados. Este trabalho propõe utilizar a mobilidade do código através da tecnologia dos agentes móveis, no sentido de se obter a recuperação de informação em bases de dados. A utilização de agentes móveis permite que a integração dessas bases de dados seja transparente para o usuário, visto que o agente será o responsável pela pesquisa de informações armazenadas nestas bases. Além disso, os agentes móveis também são responsáveis pela comunicação com outros agentes, delegando a responsabilidade do acesso às bases de dados a agentes estáticos específicos. Antever-se que, a utilização da mobilidade do código para a integração das bases de dados permitirá ganhos significativos para a organização em relação aos custos, performance e integridade das informações.

## ABSTRACT

Currently, many information storage technologies are being developed and used in organizations. However, the existing databases in this organizations can not be thrown away on a short term basis, thus creating a need to integrate various diverse databases using the code mobility. This paper proposes to use the code mobility between the mobile agents technology, in the sense of obtain information retrieval in databases. The use of mobile agents allows the integration of these databases be transparent for the user, since the agent will be the main one responsible for the research of the information stored in these databases. In addition, the mobile agents are also the responsible for the communication with others agents, delegating the responsibility to access the databases to certain determined static agents. It can be foreseen that the use of the mobility code for the integration of the databases will allow significant profits for the organization in terms of costs, performance and integrity of information

## **1. INTRODUÇÃO**

Atualmente, a efemeridade das mudanças no mundo tecnológico requer um tratamento especial por parte das empresas e organizações. Isso ocorre devido à contínua utilização dos sistemas antigos em conjunção com as tecnologias de última geração. Assim, surge aplicações que utilizam antigos bancos de dados juntamente com os recém lançados no mercado.

Devido a este fato, torna-se necessário para a empresa despender tempo do seu pessoal especializado para qualificá-lo, onerando as políticas de capacitação profissional da organização. Além disso, a migração dos dados dos sistemas antigos para os sistemas novos requerem uma reestruturação na modelagem, com o intuito de garantir a coesão e a veracidade dos dados. E, para finalizar, a recuperação da informação, em se tratando de bancos de dados distintos, torna-se inviável devido às diferenças estruturais e organizacionais dos dados.

Os temas Agentes e Bases de Dados, ou ainda a Integração de Bases de Dados através de Agentes Móveis está sendo foco de pesquisa em todo o mundo. O trabalho em [SAHY97] tem como propósito avaliar a utilização dos agentes na resolução ou ganhos de desempenho referentes a alguns problemas relacionados à recuperação de informação e bases de dados. Desde 1999, o problema da Integração de Bases de Dados é um tópico solicitado nos eventos ACM SIGIR e ACM SIGMIS sobre informação e gerenciamento do conhecimento. Além disso, observa-se cada vez mais o interesse nesta área como observado no trabalho em [BUSHIA99] sobre o compartilhamento de módulos de bases de dados heterogêneos usando objetos integradores. Este trabalho se restringe a utilização de bancos de dados relacionais, e não se utiliza da tecnologia de agentes em relação ao acesso ao banco de dados.

O presente trabalho está sendo desenvolvido utilizando o paradigma dos agentes móveis, e visa averiguar alguns aspectos como, por exemplo, o ganho no tempo de trabalho do pessoal especializado, a diminuição dos custos da empresa e a transparência na recuperação da informação utilizando banco de dados relacional, objeto relacional e orientado a objetos.

O presente trabalho está organizado em seções, sendo que a seção 2 é responsável pela estrutura do ambiente dos agentes que acessam às bases de dados, a seção 3 objetiva elucidar o ambiente de execução para a obtenção da mobilidade. As seções 4 e 5 se referem aos agentes, sendo o primeiro enfoque aos agentes móveis e o subsequente aos agentes estáticos. A seção 6 destina-se aos conceitos das bases de dados e a 7 retrata a implementação. E por fim, a seção 8 se refere as conclusões deste trabalho.

## **2. A ESTRUTURA DO AMBIENTE**

Este trabalho é fundamentado nas tecnologias dos agentes como está em [Kotay94, Chess95, Franklin96, Horizon98, Oshima98, Voyager97].

O ambiente está sendo constituído por uma rede local, estando apto para atingir os âmbitos da Internet, composta por três máquinas contendo as bases de dados, e uma máquina responsável pela inicialização do agente. Desta forma, um único cliente será o responsável pelo encaminhamento do agente móvel. O ambiente contém a seguinte estrutura.

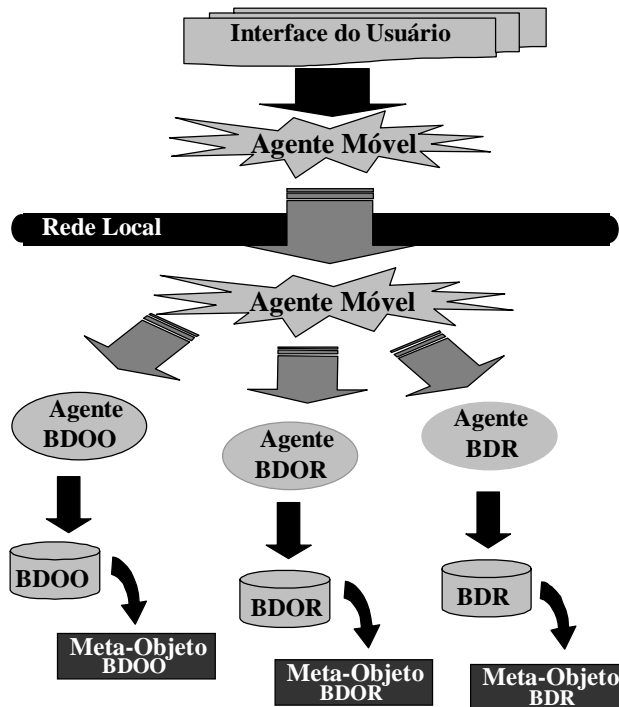


Figura 1. Estrutura do Ambiente

O usuário interage com o ambiente através de uma interface gráfica, que é responsável por disponibilizar os campos onde o usuário realizará a pesquisa desejada. Após a determinação dos campos para seleção, o agente móvel é disparado.

Uma vez inicializado, o agente móvel se desloca para uma máquina destino presente no seu itinerário, onde se comunica com o agente estático responsável pelo acesso a base de dados correspondente. Com esta estrutura, o acesso dos usuários às bases de dados é restrito, visto que as estações clientes não têm acesso direto ao banco de dados, dificultando a danificação de informações relevantes, e/ou a má manipulação das mesmas, evitando, conseqüentemente, a inconsistência do ambiente. Além disso, os agentes foram particionados em agentes móveis e agentes estáticos, para aumentar a performance do ambiente. Caso contrário, o agente móvel carregaria consigo todo o código de acesso às bases de dados singularmente, aumentando o tráfego das informações que passariam na rede, sobrecarregando o agente móvel. Assim, neste caso, o agente móvel é responsável por enviar as requisições e receber as respostas enquanto que, o estático é responsável por realizar o acesso às respectivas bases de dados.

### 3. AMBIENTE DE EXECUÇÃO

O ambiente de execução é o principal responsável pela mobilidade do agente dentro da rede. Esta migração ocorre com o intuito do agente realizar a tarefa que foi a ele designada. A primeira medida que o agente deve tomar é selecionar o *host* que provê o serviço que ele

necessita. Embora os agentes carreguem consigo seu itinerário fixo, eles, na maioria dos casos, devem decidir para onde vão. Mecanismos como pesquisas por palavras chaves, ou ainda rotas semânticas, que tentam encontrar o computador destino de um agente baseado em que tarefa ele deseja executar, podem auxiliar os agentes a encontrarem seus provedores de serviços.

Além da síntese de escolha da rota do agente, o ambiente de execução é responsável pela garantia da mobilidade do agente na rede. O processo de mobilidade envolve dois passos, o primeiro é mandar o agente da origem para o destino e o outro é receber o agente no destino. Para trafegar o agente da máquina origem para o *host* destino é necessário suspender a execução do agente, no sentido de interromper a ação que o mesmo esteja desempenhando, codificar o agente para que seja transmitido e por fim liberar os recursos que o agente estava utilizando na origem, tornando estes disponíveis para outros usuários, agentes ou ambientes. Quando o mesmo chega no destino é necessário que, em primeiro lugar, o agente seja decodificado, verificando logo em seguida se o recurso que o agente necessita está disponível naquela máquina. Em caso positivo, o agente é excluído efetivamente do seu ambiente de execução original. Este processo transitório ocorre porque, se por algum motivo, o agente não consiga atingir o *host* desejado, ou sofra alguma modificação externa, ele retornará ao seu ambiente original, recompondo-se. Neste sentido, aspectos de segurança são envolvidos com o cuidado de garantir a integridade tanto do ambiente quanto do agente [CHESS 95].

Para transmissão dos agentes, protocolos como TCP/IP, HTTP, X.25 e SMTP podem ser utilizados como mecanismos de transporte dos agentes entre *hosts*, sendo os mesmos definidos pelo ambiente de execução em questão.

Os ambientes de execução dos agentes móveis devem prover mecanismos de controle dos recursos disponíveis. Este tipo de controle permite uma utilização eficiente dos recursos, e evitam que sejam manipulados inadequadamente pelos agentes. Os agentes podem conter informações relativas às necessidades dos seus recursos e a capacidade de utilização dos mesmos. Quando o agente é despachado para outro ambiente, os recursos por ele outrora utilizados, são liberados.

Em relação ao ambiente que está sendo desenvolvido, foi necessário executar um destes ambientes de execução em cada máquina presente no itinerário do agente móvel. Assim, um servidor referente a este ambiente deve estar inicializado na estação onde o agente for criado, além dos respectivos servidores em cada máquina que contiver os bancos de dados. Neste caso específico, o agente será direcionado de acordo com o endereço IP da máquina em questão e com a porta correspondente, visto que em uma única máquina podem ter dois servidores aguardando conexões em portas distintas.

#### **4. AGENTE MÓVEL**

A utilização de um agente com características de mobilidade permite uma melhor utilização do tempo disponível do profissional responsável por realizar a pesquisa, visto que o mesmo poderá disparar o agente e logo em seguida desconectar-se da rede para realizar outras tarefas, obtendo os resultados em outro momento que a conexão for restabelecida. Além disso, a utilização de um agente móvel fornece transparência na recuperação das informações, visto

que o mesmo se comunica com outros agentes das diversas bases de dados interoperando as informações independente do banco em questão.

Uma vez instanciado local, ou remotamente, e realizada suas tarefas na máquina específica, o agente móvel prepara-se para o início do seu itinerário desempenhando suas funções. Com este intuito, o agente móvel carrega consigo dois objetos. Um responsável pelos requisitos necessários à execução da pesquisa, e o outro responsável por armazenar os resultados da mesma.

O objeto que contém os requisitos é fornecido pelo usuário, após a criação do agente, e tem como conteúdo os dados dos campos presentes na interface gráfica, pelo qual o cliente deseja que a pesquisa seja realizada.

Com a posse deste objeto, o agente invoca outro método que é responsável pela sua movimentação para a próxima estação onde o mesmo busca os demais resultados. Após a comunicação com o agente estático, que tem como objetivo passar o objeto contendo as requisições obtidas no cliente, este agente do banco de dados fornece como retorno um novo objeto responsável pelo armazenamento do resultado da pesquisa em questão.

Obtendo estes resultados, o agente móvel invoca o método responsável pela sua movimentação e passa a realizar sua tarefa na próxima máquina designada, onde ocorre o mesmo processo anteriormente explicitado.

Cada resultado obtido através da comunicação com os agentes estáticos aos acessos às bases de dados relacional, objeto-relacional e orientadas a objetos serão adicionados ao objeto no agente móvel responsável pelo armazenamento dos resultados. Assim, no final do itinerário, o agente móvel terá todos os dados resultantes da pesquisa requisitada, se deslocando para a máquina de origem.

Chegando na máquina de origem, este é responsável por enviar as informações obtidas ao cliente, completando assim sua meta de execução.

## **5. AGENTES ESTÁTICOS**

Os agentes caracterizados como estáticos, são em número de três, de acordo com a natureza dos bancos de dados considerados neste trabalho: BDR, BDOR, e BDOO. Para cada banco de dados específico é instanciado um novo agente estático, que contém as funções específicas para o acesso aos dados, de acordo com o banco de dados do ambiente acessado.

No momento de instanciação do agente estático, fato este que ocorre quando o agente móvel chega na máquina destino, ele recebe o objeto contendo as requisições da pesquisa do usuário. De posse das requisições, os agentes estáticos realizam uma consulta à base de dados utilizando para isso as funções referentes ao banco em questão. Como estas funções são divergentes, torna-se necessário a implementação de um agente estático para cada tipo de banco de dados.

Outra característica importante dos agentes estáticos é a propensão que o mesmo tem em trabalhar com os gerenciadores de drivers e conexões diretas ao banco. Sendo esta tarefa tão

singular para cada banco e, ao mesmo tempo, não correspondendo às funções principais dos agentes estáticos, observa-se uma necessidade de criação de um meta-objeto. Este meta-objeto, situado em um meta-nível desta aplicação, é responsável pela definição do driver correspondente ao banco, utilização e realização da conexão através do mesmo. Assim, como também os drivers são específicos para cada banco de dados, por exemplo o driver utilizado pelo Oracle, é diferente do driver utilizado pela Microsoft, que por sua vez é diferente do driver utilizado pela ObjectStore, efetiva-se a criação de meta-objetos para cada agente estático em questão.

## **6. BASES DE DADOS**

Desde o surgimento dos primeiros sistemas de computação, a necessidade de armazenar e recuperar dados de forma confiável e segura vem se apresentando como um dos principais desafios.

A princípio o auge do armazenamento de dados ocorreu com o surgimento dos bancos de dados relacionais (BDR) [Date91, Navathe94, Melo98]. Este fez emergir a abstração dos dados e o desenvolvimento de aplicações não direcionadas a apontadores ou registros, mas a um conjunto de dados.

Aproximadamente uma década posterior, surgiram os bancos de dados orientados a objetos, devido a complexidade das aplicações da época que estavam tornando os bancos relacionais inadequados ao armazenamento. Os BDOO (Banco de Dados Orientado a Objeto) adotaram muitos dos conceitos implementados nas linguagens de programação com este mesmo paradigma. A principal diferença entre os mesmos é que, no banco de dados, os objetos têm características persistentes, enquanto que nas linguagens, os mesmos são transientes. Objetos persistentes trazem como característica básica existirem permanentemente em uma base de dados, para que possam ser acessados por outros programas e aplicações.

O conceito fundamental de um banco de dados orientado a objetos é o próprio objeto. Um objeto é uma representação abstrata de uma entidade do mundo real.

Apesar das vantagens citadas em relação aos BDOO, estes representam uma mudança muito brusca referente a tecnologia atualmente adotada pelos banco de dados relacionais (BDR). Adicionando-se ao fato de que são incompatíveis com as tecnologias anteriores (BDR), os mesmos ainda requerem novos conjuntos de habilidades por parte dos profissionais da área, além de necessitarem de treinamento especializado.

Diante deste contexto surgem os Banco de Dados Objeto-Relacional (BDOR), que permitem uma integração das atuais tecnologias com os conceitos de orientação a objetos. Neste sentido, os BDOR representam uma progressão natural baseada nos benefícios da combinação entre uma base de dados relacional com uma base de dados orientada a objetos [Fortier99].

### **6.1. Manipulação dos Dados**

A manipulação dos dados nestas diversas bases de dados ocorre de maneira distinta. Em se tratando de um BDR, o acesso a base de dados efetiva-se através da linguagem SQL-92 que

permite a inserção, exclusão e consulta do dados através de estruturas denominadas tabelas ou relações.

Com o intuito de estender o padrão SQL-92 (SQL-2) foi desenvolvida a linguagem SQL-3 [Fortier99]. Isso significa que os bancos de dados e as aplicações que utilizam o padrão SQL2 vão continuar funcionando, sem nenhuma alteração, com o modelo SQL-3. A extensão ocorre no momento da inclusão dos tipos abstratos de dados que permitem a manipulação dos objetos. Esta linguagem é utilizada no acesso aos dados de bases objeto-relacional.

O armazenamento dos dados, assim como a sua manipulação em um BDOO ocorre através da persistência dos objetos, onde se permite armazenar o estado dos objetos em um ambiente não volátil.

Devido ao crescimento do paradigma orientado a objeto e novas formas de acesso às bases de dados, tanto o objeto-relacional, quanto o orientado a objeto, é necessário permitir a integração destas bases com a atual tecnologia relacional.

Os bancos de dados estão sendo projetados contendo estruturas semelhantes, diferindo apenas no tocante a sua forma de organização. Em relação ao relacional, estão sendo analisadas e estruturadas tabelas, contendo as chaves primárias e suas chaves estrangeiras quando houver necessidade, além dos campos de código, que são responsáveis pela identificação das tuplas (registros). Em relação ao banco de dados objeto-relacional, sua forma de armazenamento é semelhante aos bancos de dados relacionais, diferindo na forma de acesso aos dados, através do SQL-3. Já em relação aos bancos de dados orientados a objetos, estão sendo definidas as classes e objetos de acordo com as necessidades preestabelecidas.

## **7. IMPLEMENTAÇÃO**

A implementação tem como objetivo principal exemplificar as características da mobilidade do código, juntamente com a integração das bases de dados, elucidando as etapas essenciais para a conclusão do desenvolvimento.

### **7.1. A Plataforma**

A plataforma é responsável por disparar o ambiente de execução, permitindo assim a mobilidade do agente. Após a avaliação de diversas plataformas, dentre elas os Aglets, Grasshopper, Concordia e Voyager, este último foi o escolhido devido principalmente a disponibilidade dos recursos, visto que outros não disponibilizavam o *freeware* completo, inibindo assim o desenvolvedor ao acesso a todos os módulos. Outra característica importante em relação ao Voyager, é que o mesmo é totalmente desenvolvido em Java, com isso, a utilização deste ambiente com os agentes que podem transitar pelo mundo, torna-se evidente, devido à independência de plataforma inerente à linguagem.

### **7.2. Configuração das máquinas**

Uma vez escolhido o ambiente de execução do agente, segue-se para a configuração da máquina onde o mesmo possa ser executado. A princípio é necessário que se tenha instalado o JDK (Java Developers Kit) na versão superior ao 1.1. Além disso, é necessário que o Voyager esteja instalado em todas as máquinas que o agente irá percorrer, visto que o mesmo será o responsável por dar início ao ambiente de execução.

Para inicializar o ambiente de execução, basta dentro da tela do DOS executar o seguinte comando:

***voyager 8000***

Sendo 8000 o número da porta de comunicação dos agentes. Após realizada esta tarefa, o ambiente de execução estará em funcionamento, assim como mostrado na figura 2.



```
MS-DOS java
Auto
Microsoft(R) Windows 95
(C)Copyright Microsoft Corp 1981-1996.
C:\WINDOWS>voyager 8000
voyager core technology 2.0, copyright objectspace
```

*Figura 2. Ambiente de execução do Voyager*

### **7.3. O Usuário**

O usuário deste ambiente será o responsável por disparar o agente móvel para realizar a pesquisa de acordo com as suas necessidades. Para tal, foi desenvolvida uma interface gráfica, mostrada na figura 3, onde o mesmo seleciona a informação que deseja pesquisar, disparando logo em seguida o agente para o seu itinerário.





Figura 3. Interface gráfica para realizar a pesquisa como Agente.

No exemplo elucidado acima, pode-se observar que o agente móvel está sendo inicializado e tem como requisito buscar todos os usuários que tenham como profissão a fisioterapia.

Internamente, para o agente móvel ser criado em uma máquina remota ou em uma máquina local, é necessário que antes seja definida a sua interface de comunicação, ou seja, os métodos que podem ser executados por objetos remotos.

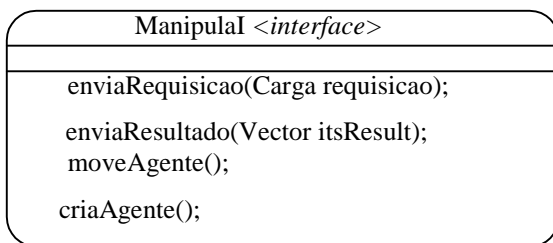


Figura 4. Interface ManipulaI

O parâmetro passado através do método *enviaRequisição*, é um objeto da classe *Carga*, mostrada na figura 5, que contém as informações fornecidas pelo usuário através da interface gráfica.

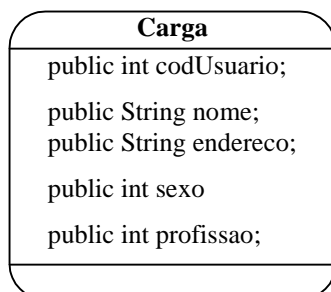


Figura 5. Classe Carga

#### 7.4. O Agente Móvel e o Agente Estático

A classe *AgenteMovel* é o efetivo agente que trafega através da rede em busca da informação que o usuário requisitou. Para isso, a mesma herda as características de uma classe do Voyager denominada *com.objectspace.voyager.agent.Agent*, fornecendo assim funções inerentes a um agente, como sua mobilidade.

Além disso, esta classe *AgenteMovel*, implementa duas outras interfaces, a *ManipulaI*, previamente caracterizada, e a *Serializable*, que fornece os subsídios necessários para o transporte dos agentes entre máquinas na rede.

Para que haja a mobilidade do agente entre máquinas, é necessário que explicitamente o desenvolvedor do agente relacione o seu itinerário, descrevendo a sua trajetória de pesquisa. Para tal, faz-se necessário a utilização da linha abaixo mencionada:

```
Agent.of(this).moveTo("//iris:8000","criaAgente");
```

Neste caso, há uma ordem para o agente se mover para a máquina denominada *iris*, se comunicar com o ambiente de execução através da porta 8000, e depois que o mesmo for criado nesta máquina destino, ele terá como tarefa executar o método “criaAgente”. Este método será o responsável por criar o *AgenteBDR*, o agente estático na máquina destino, que realizará a conexão com a base de dados, assim como a pesquisa, armazenando os valores de retorno.

O retorno gerado pelo *AgenteBDR*, denominação concedida por se tratar do agente estático que realiza a pesquisa na base de dados relacional, é armazenado em um objeto denominado *Resultado*, instanciado da classe *Carga*. O objeto *Resultado*, por sua vez, está organizado em um vetor, e é passado para o *AgenteMovel* como sendo a carga de retorno do mesmo.

Além disso, o *AgenteBDR* é o responsável pela efetiva conexão ao banco de dados, utilizando *JDBC*[Reese97]. Para tal, é necessário a criação de três classes inerentes ao mesmo: *Connection*, *Statement* e *ResultSet*. Através destas classes ocorre a conexão do driver específico para o *JDBC* com a fonte de dados definida através do *ODBC* (*Open Database Conectivity*), que permite o acesso a base de dados relacional.

Após a passagem do *AgenteMovel* pelo banco de dados relacional, o mesmo irá se encaminhar para as demais estações, estando esta parte em desenvolvimento.

## 8. CONCLUSÃO

A integração destas bases de dados permite uma redução nos custos de uma organização, onde os sistemas legados e as novas tecnologias de armazenamento de dados poderão se inter-relacionar, fornecendo aos usuários finais informações mais consistentes e providas de uma gama maior de base de dados, como resultado do processo de integração.

Além de trazer estes benefícios, a utilização de Agentes Móveis deve permitir uma maior consistência nos dados, visto que os agentes se deslocam para as máquinas e executam suas

tarefas localmente. Assim, os dados não ficam trafegando pela rede, mas ficam armazenados no compartimento dos agentes móveis, fornecendo mais um nível de segurança.

Assim, a elaboração deste trabalho visa minimizar os custos da empresa, garantindo a performance, integridade e transparência na obtenção dos dados.

## 9. TRABALHOS FUTUROS

Em uma extensão deste trabalho, os aspectos específicos de segurança dos agentes móveis podem beneficiar este ambiente. A questão da segurança está sendo estudada em [Jucá00]. Além disso, questões relativas às transações necessárias no processo do agente estático também podem ser observadas.

## 10. REFERÊNCIAS

- [Bushia99]Bushia, G.; Ferreira, J.E. “Compartilhamento de Módulos de Bases de Dados Heterogêneas através de Objetos Integradores” In proceedings of the 1999 Simpósio Brasileiro de Engenharia de Software (Florianópolis,SC, Oct.13-15) Brazil, 129-145, 1999.
- [Chess95]Chess, D.M.; Harrison, C. G.;Lebine, D. 1995.“Itininerant Agents for Mobile.” Computing IBM Reserach Report RC 20010, Research Division, IBM, 1995.
- [Date91]Date, C.J. Introdução a Sistemas de Banco de Dados, Campus, Rio de Janeiro, Brasil, 1991.
- [Fortier99]Fortier, P.J. SQL-3 Implementing the Object-Relational Database, McGraw-Hill, 1999.
- [Franklin96]Franklin, S.Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents\_Institute for Intelligent Systems, University of Memphis, 1996.
- [Horizon98]Horizon Systems Laboratory “Mobile agent computing: A white paper”. Technical report, Mitsubishi Electric ITA,January.(<http://www.meitca.com/HSL/Projects/Concordia>), 1998.
- [Jucá00]Jucá, K.R.L., "Aspectos de Segurança em Sistemas de Agentes Móveis", Trabalho Individual, CPGCC,UFSC, 2000.
- [Kotay94]Kotay, K; Kotz, D. Transportable Agents.\_Department of Computer Science, Darmouth, 1994.
- [Melo98]Melo, R.N.; Silva, S.D.; Tanaka, A.K. Banco de Dados em Aplicações Cliente-Servidor, Infobook S.A, 1998.

- [Navathe94]Navathe, S.B.;Elmasri, R. Fundamentals of Database Systems, Addison-Wesley, Second Edition, 1994.
- [Oshima98]Oshima, M.; Lange, D. B. Programing and Deploying Java Mobile Agents with Aglets, Addison-Wesley Publishing Company, 1998.
- [Reese97]Reese, G. Database Programming with JDBC and Java, O'Reilly.
- [Soler97]Soler, O.O.;Prieto, F.M.;Gutiérrez, A.B.A. "A Reflective Persistence Middleware over na Object Oriented Database Engine." In Proceedings of the 1999 Simpósio Brasileiro de Banco de Dados (Florianópolis, SC,Oct11-13) Brasil, 137-151, 1997.
- [Sahy97]Sahuguet, Arnaud; "About Agents and Databases", CIS-650, <http://www.x.com.br>
- [Voyager97]Voyager Core Technology Group. "Agent-enhanced distributed computing for java". Technical report, - ObjectSpace, Inc.(Mar) <http://www.objectspace.com/developers/voyager/white/index.html>, 1997.

## 11.BIOGRAFIA

**Daniela Barreiro Claro** é mestranda de Ciência da Computação na UFSC - Universidade Federal de Santa Catarina, na área de Engenharia de Sistemas Distribuídos. Ela é bacharel em Ciência da Computação pela UNIFACS - Universidade Salvador. Atualmente, ela trabalha com planejamento e desenvolvimento de sistemas computacionais em uma empresa privada além de participar do programa de Mestrado Inter-Institucional do CPGCC/UFSC. Suas áreas de interesse são Computação Distribuída com Agentes Móveis e Banco de Dados e suas aplicações.

**João Bosco Mangueira Sobral** é professor na Universidade Federal de Santa Catarina, na área de Engenharia de Sistemas Distribuídos. Ele é doutor pela Universidade Federal do Rio de Janeiro(COPPE) em 1996. Ele é mestre em Engenharia de Sistemas e Computação pela UFRJ/COPPE desde 1977, e é graduado em Matemática pelo Instituto de Matemática da UFRJ (1973). Atualmente, ele trabalha no Departamento de Informática e Estatística e do Curso de Graduação da UFSC. Suas áreas de interesse são Computação Distribuída, Mobilidade em Computação e Gerenciamento de Redes de Computadores e de Telecomunicações.