

# Adaptação de transações sensíveis ao contexto com o MiD-Mobile

Daniela Flôr<sup>1,2</sup>, Maurício Ferreira<sup>2</sup>, Nilson Sant'Anna<sup>2</sup>,  
Ricardo Paulino<sup>1</sup> e Yandre Costa<sup>3</sup>

<sup>1</sup> Universidade Paranaense, Avenida Huberto Bruning – 360,  
Paranavaí, Paraná-Brasil  
danielaflor@unipar.br, ricardovasselaipaulino@gmail.com

<sup>2</sup> Instituto Nacional de Pesquisas Espaciais, Avenida Astronautas -1758,  
São José Campos, São Paulo-Brasil  
daniela.eloise.flor@gmail.com, nilson@lac.inpe.br, mauricio@ccs.inpe.br

<sup>3</sup> Universidade Estadual de Maringá, Avenida Colombo - 5790  
Maringá, Paraná-Brasil  
yandre@din.uem.br

**Resumo.** As características inerentes à computação móvel conduzem a pesquisas desafiantes, inclusive em transações de banco de dados. O dinamismo do meio sem fio, a escassez de recursos de computação dos dispositivos móveis e a singularidade dos modelos transacionais, incentivaram o MiD-Mobile. O MiD-Mobile é um middleware que isola as atividades de desenvolvimento de software do monitoramento das especificidades do ambiente distribuído móvel. Sua arquitetura agrega serviços que encapsulam funções específicas, um dos principais é o adaptador de transação ao contexto mais promissor. O serviço é baseado em um algoritmo de árvore de decisão, foi implementado em Java e faz uso de autômato finito determinístico para armazenar o conhecimento.

**Palavras-chave:** Transações móveis, Adaptação de transações, Árvore de decisão, Autômato Finito Determinístico, Propriedades ACID.

## 1 Introdução

A computação móvel é vista por [17], como um ambiente de computação distribuída adicionado de estações de base, computadores portáteis e comunicação sem fio. As estações de base gerenciam os dispositivos portáteis que estão em sua área de abrangência, denominada célula. Os computadores móveis são dispositivos portáteis que se movimentam livremente e, por intermédio das estações de base, acessam a rede fixa através da comunicação sem fio.

Tais elementos agregaram à arquitetura distribuída questões como a conectividade sem fio, mobilidade e portabilidade. A portabilidade veio com a miniaturização dos computadores, mas resultou em dispositivos com restrição de recursos. A comunicação sem fio oferece liberdade, porém com delicada qualidade de serviço. A mobilidade acrescenta novas questões como, por exemplo, modo de endereçamento.

Mesmo com as limitações, é crescente o interesse em sistemas de informação que processem transações atuantes nesse ambiente. Para [7], uma transação é um modelo de comportamento que assegura a integridade de dados compartilhados por usuários de uma aplicação. Cada transação é uma unidade atômica de execução, isolada das demais, que não viola nenhuma regra de consistência e ao final tem o seu resultado persistido, essas propriedades são conhecidas por ACID.

Devido às constantes desconexões, os pressupostos teóricos tratados em [13], que garantem o sucesso das transações nos bancos de dados distribuídos não são totalmente aplicáveis ao ambiente móvel.

Este artigo apresenta nas seções seguintes considerações sobre os modelos transacionais sugeridos para a computação móvel, os detalhes da arquitetura de um middleware que encapsula funções de avaliação do meio, adaptação e gerenciamento de transações adaptadas. Além de apresentar as implementações, conclusões e trabalhos futuros.

## 2 Modelos de Transação para Ambiente de Computação Móvel

Em [7], modelos de transação para ambiente de computação móvel são identificados como abstrações de alto nível de operações de leitura/escrita em dados compartilhados e distribuídos entre estações fixas e móveis.

Genericamente, é desejável que tais propostas sejam capazes de suportar conexões intermitentes, apoiar a mobilidade do cliente durante a transação, atualizar seu deslocamento em função do movimento, manter transações de longa duração devido à invariabilidade do meio, fornecer autonomia de trabalho em desconexões temporárias e reintegrar atualizações locais na rede fixa para manter a consistência dos dados.

Diante da pluralidade de exigências impostas pelo meio, vários modelos foram propostos. Uma revisão é apresentada em [17], por razões de espaço, os modelos serão apenas agrupados em suas características determinantes. Um dos critérios para agrupar os modelos é o local de execução, as opções incluem processamento na rede fixa, no dispositivo móvel ou dividido entre ambos.

Os modelos Kangaroo em [5], Pre-serialization em [4], Moflex em [9] e MDSTPM em [21], foram propostos indicando que o processamento da transação deve ocorrer na rede fixa, após a requisição ter sido iniciada na estação móvel, sendo responsabilidade da rede fixa coordenar e executar a transação e enviar os resultados.

Uma possibilidade mais autônoma, como o tratado pelos modelos HiCoMo em [10], Pro-motion em [20], Clustering em [14], Two-tier replication em [8], IOT em [11], Prewrite em [12] e o Semantics-based em [19], permite a execução das transações no computador móvel com posterior atualização na rede fixa.

A distribuição do processamento entre a rede fixa e os dispositivos móveis é considerada pelo modelo Reporting em [1], além de também ser uma proposta adotada nos modelos Clustering e Two-tier replication.

Outra abordagem de classificação usada por [15] considera as desconexões, largura de banda, limitação no suprimento de energia, escassez de espaço em disco e mobilidade do usuário. Técnicas de caching de dados suportam desconexões, a exceção são os modelos Kangaroo em [5] e Pre-serialization em [4]. Quando a largura

de banda está alta, o modelo IOT sugere sua exploração para transferência de dados, já o Pro-motion visto em [20] e o Semantics-based tratado em [9] despacham apenas dados imprescindíveis no canal de comunicação.

As sugestões para poupar energia implicam na divisão da computação pelo modelo Reporting em [1], modo de operação desconectado pelo modelo Semantics-based e o desligamento da unidade móvel em períodos ociosos pelo modelo Kangaroo em [5].

A escassez de espaço em disco reflete na quantidade de dados enviados às caches das unidades móveis. Os modelos Semantics-based e Pro-motion transferem pequenos fragmentos, o modelo Reporting sugere a computação distribuída entre a unidade móvel e a rede fixa para diminuir os dados levados para as caches.

O apoio à mobilidade é a última consideração feita por [15] na revisão dos trabalhos. Os modelos Kangaroo e o MDSTPM dividem a transação em subtransações enviadas às estações de base por onde os usuários passam.

Outro critério de classificação pesquisado considera o tratamento das propriedades ACID pelos modelos de transações móveis [2]. Os modelos transacionais optam pela flexibilização de uma ou mais propriedades como forma de viabilizar a transação.

A flexibilização da atomicidade é conseguida com a utilização de transações aninhadas ou com a ocorrência de operações de divisão (split). Em ambos os casos o resultado é o mesmo, a divisão das transações em subtransações que são efetivadas mesmo que outras não sejam, modelos que adotam essa estratégia são Pro-motion em [20], Reporting em [1], Kangaroo em [5], Moflex em [9] e HiCoMo em [10].

As propostas para maleabilizar a consistência são bem divergentes. O Pro-motion em [20] e o Semantics-based em [19] agregam regras consistentes de acesso. Já o Moflex em [9], HiCoMo em [10] e o Two-tier replication em [8] trabalham com regras que dizem se uma transação executou corretamente ou não.

O isolamento é impulsionado pela liberação de resultados parciais nos modelos IOT, Reporting, HiCoMo, Pré-serialization, Semantics-based ou em níveis graduais de compartilhamento, como no Pro-Motion.

Devido à conexão intermitente e outras falhas de comunicação, a flexibilização da durabilidade nem sempre é trivial. Para obter sucesso o modelo HiCoMo permite que mesmo se algum resultado não se tornar persistente a transação será efetivada.

Os modelos transacionais avaliados em [17], [15] e [2] divergem em vários aspectos e acabam por enfatizar certas características em detrimento de outras. É possível perceber a inexistência de um modelo único que contemple todas as variáveis do ambiente de computação móvel. Sendo assim, é interessante que várias estratégias de atuação das transações estejam aptas a operar.

A atuação diversificada das transações é possível através da adaptação. Segundo [2], adaptar significa modelar soluções em função de certos fatores. Portanto, é necessário identificar os fatores que influenciam na execução das transações.

### 3 Adaptação Sensível ao Contexto

Aplicações baseadas em transações que operam em ambiente de computação móvel devem coexistir com a instabilidade do meio e a despeito dela concluírem suas ações com êxito. Identificar as ameaças à execução das transações ajuda a evitar insucessos.

Os dispositivos portáteis possuem recursos limitados, requerendo uma avaliação no tamanho dos dados armazenados localmente, na capacidade computacional dos serviços disponibilizados e em modos de operação diferenciados visando a economia de energia.

Modos mais flexíveis de operação também atendem a variabilidade da comunicação sem fio. Quando a rede estiver descongestionada é o período ideal para uma estação móvel estar conectada à rede fixa, transferindo dados. Uma conexão parcial pode ser a melhor estratégia quando a largura de banda for suficiente para receber dados.

Quando o canal de comunicação estiver indisponível é possível trabalhar desconectado. O modo vigília é indicado quando a estação móvel não tem ou está no limite dos seus recursos, assim as requisições podem ser enviadas à rede fixa enquanto o dispositivo economiza recursos e aguarda os resultados.

Os dispositivos móveis são personificados em assistentes pessoais digitais (PDA), telefones celulares, smart phones, notebooks, entre outros. É certo que avanços tecnológicos vão popularizar equipamentos com funções e capacidades diferentes. A heterogeneidade de dispositivos e recursos é uma constante. Isso também se aplica às diferentes formas de comunicação sem fio, por exemplo, bluetooth, Wi-Fi e GPRS.

Outro fator é o enlace de rede sem fio que alterna a intensidade do sinal e ocasiona cobertura insuficiente, impedindo o cliente de usar recursos disponíveis na rede fixa.

Após a verificação das possíveis interferências a um sistema de computação móvel se faz oportuna a proposta de um middleware, que encapsule funções de monitoramento e adequação ao meio. Estratégias de adaptação são exploradas em diferentes áreas da Ciência da Computação. Para adaptar transações [18] propôs um Serviço de Transações Móveis (STM) que trata algumas dimensões importantes a respeito do dispositivo móvel, mas exige uma configuração razoável.

A adaptação colaborativa entre as transações e a Plataforma de Gerenciamento de Transações Adaptáveis (PGTA) é uma proposta de [16]. A plataforma é responsável por monitorar recursos e às transações cabe reagir às mudanças do ambiente.

A adaptação aumenta as chances das transações serem concluídas com êxito. Nesse sentido, o MiD-Mobile, durante a adaptação, avalia a capacidade de armazenamento de dados do dispositivo. Esse acréscimo, em relação aos trabalhos pesquisados, permite a isenção de uma configuração mínima dos dispositivos que acessam a aplicação, atendendo a heterogeneidade de opções disponíveis no mercado.

Outro quesito inovador é o controle no modo de operação que viabiliza a economia de recursos, além do tratamento estendido de todas as propriedades ACID para dar suporte às opções de adaptação da transação. Os trabalhos correlatos analisados são de teor acadêmico, o que permitiu apenas equiparações conceituais, dificultando comparações sobre os resultados obtidos.

O MiD-Mobile idealizado nessa pesquisa visa atender diferentes domínios de aplicação através da harmonização entre a necessidade e a disponibilidade de recursos computacionais, sem sacrificar as funcionalidades de cada software. A proposta é adaptar as transações considerando a conectividade sem fio, os dados e os recursos do dispositivo móvel, a próxima seção trás o detalhamento dessas transações.

### 3.1 Transações móveis adaptadas ao contexto

O MiD-Mobile ao adaptar uma transação define-a como transação móvel adaptada ao contexto ( $t_{mac}$ ), as características que impactam no seu molde constam da tabela 1.

**Tabela 1.** Tabela de recursos dinâmicos

Dimensões	Itens considerados
Canal de comunicação	Estado da conexão
	Taxa de transferência
	Intensidade do sinal
Dados	Situação dos dados
	Completeness dos dados
	Validade dos dados
Dispositivo	Capacidade de Bateria
	Capacidade de hardware

O plano de execução da  $t_{mac}$  ( $PET_{mac}$ ), exposto na tabela 2, inclui o local de execução, o modo de operação e o tratamento das propriedades ACID.

**Tabela 2.** Tabela de dimensões adaptadas

Dimensões do PE	Conteúdos
Plataforma de Execução	Local
	Remoto
	Distribuído
Modo de Operação	Totalmente conectado
	Parcialmente conectado
	Desconectado
	Soneca
Propriedades ACID	Atomicidade tradicional ou estendida
	Consistência tradicional ou estendida
	Isolamento tradicional ou estendido
	Durabilidade tradicional ou estendida

A dimensão plataforma de execução define se uma  $t_{mac}$  pode atuar localmente (no dispositivo móvel), se pode ser processada remotamente (na rede fixa) ou se será particionada entre as opções anteriores. As propriedades ACID também são definidas no  $PET_{mac}$ . No plano remoto não são necessárias mudanças, na opção local a preocupação é com a consistência. A divisão das transações requer atomicidade local e global e reavaliação do isolamento.

Transações móveis operando localmente permitem que o dispositivo trabalhe parcialmente conectado ou desconectado. Transações móveis executadas remotamente admitem todos os modos de operação. Já as transações móveis distribuídas toleram o modo parcialmente conectado ou em vigília.

O MiD-Mobile é composto por módulos que provêm todo o suporte necessário para avaliar as dimensões do meio, executar as  $t_{mac}$ s, além de aplicar algoritmos e protocolos específicos para cada estratégia de adaptação. Segue seu detalhamento.

### 4 Arquitetura do MiD-Mobile

O MiD-Mobile é composto por serviços que interagem permitindo à aplicação atingir resultados desejados. A figura 1 oferece uma visão geral da sua arquitetura.

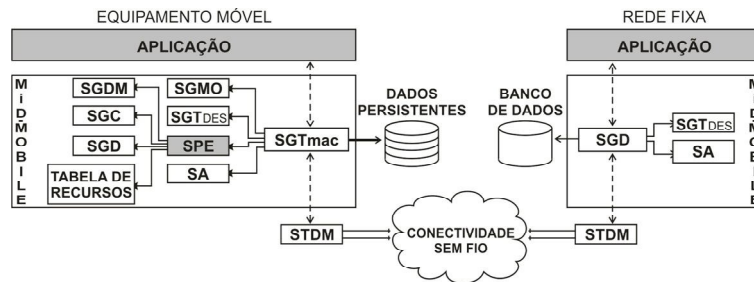


Fig. 1. Arquitetura do MiD-Mobile

A transação ao ser iniciada é enviada ao middleware, o SGT<sub>mac</sub> (serviço de gerenciamento de transações móveis adaptadas ao contexto) recebe e aciona o SPE (serviço de plano de execução), que indicará o plano de execução mais favorável. O SPE emite esse parecer após receber a colaboração de outros três serviços: o SGDM (serviço de gerenciamento do dispositivo móvel), o SGC (serviço de gerenciamento da comunicação) e o SGD (serviço de gerenciamento dos dados).

O SPE é um mecanismo de classificação, implementado em Java através do ID3, algoritmo que elabora árvores de decisão (AD). Classificação consiste na inferência de um modelo de agrupamento em classes a partir de uma amostra. Dentre os métodos de classificação existentes, a escolha por AD deve-se às características do problema, como atributos representados por valores discretos, robustez ao verificar ruídos ou a ausência de valores das instâncias, entre outros.

Uma árvore de decisão particiona recursivamente um conjunto de treinamento, como visto na figura 2, até que cada subconjunto obtido contenha casos de uma única classe, no caso do MiD-Mobile, um único plano de execução.

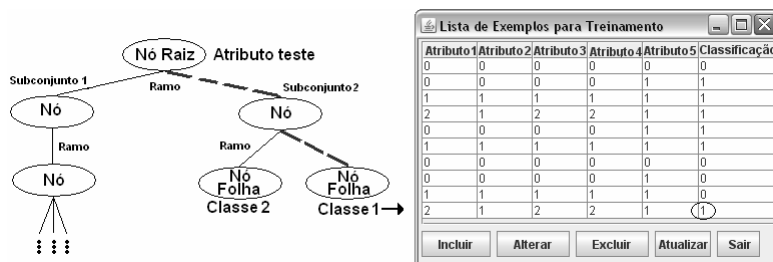


Fig. 2. Árvore criada pelo conjunto de treinamento

A responsabilidade pela indução da árvore é do SA (serviço de aprendizagem) da rede fixa. A escolha pelo algoritmo indutor, ID3, deve-se a sua completude no espaço de busca das árvores e a ausência de backtracking, ou seja, reconsideração da escolha dos atributos, o que garante maior desempenho.

O algoritmo começa definindo o melhor atributo raiz para iniciar a classificação. Essa definição aplica um cálculo estatístico chamado ganho de informação, usado para avaliar quão bem um atributo classifica os exemplos de treinamento.

Antes do cálculo do ganho de informação verifica-se o ruído (classificação incorreta) inserido em um conjunto de treinamento. Essa avaliação é chamada de entropia. Em um conjunto de treinamento  $S$  o cálculo da entropia é dado por:

$$Entropia(S) = \sum_{i=1}^n - p_i \log_2(p_i) \quad (1)$$

Por razões de espaço, é apresentado um caso de cálculo de entropia para uma classificação binária. O exemplo contém 14 ocorrências no conjunto de treinamento, destes 9 são classificados como A e 5 como B, a entropia de  $S$  é dada por:

$$Entropia([9_A, 5_B]) = -(9/14) \log_2(9/14) + -(5/14) \log_2(5/14)$$

$$Entropia([9_A, 5_B]) = 0,940$$

Após o cálculo da entropia de  $S$ , a fórmula do ganho de informação é aplicada para determinar o melhor atributo a ser usado como raiz.

$$Ganho(S, A) = Entropia(S) - \sum_{v \in \text{valores}(A)} \frac{|S_v|}{|S|} Entropia(S_v) \quad (2)$$

Para exemplificar dois atributos serão considerados: estado e taxa. O atributo estado possui os conteúdos “conectado” e “desconectado”. A taxa pode ser “satisfatória” ou “insatisfatória”. A ocorrência do conteúdo “conectado” para o atributo estado é de 3 vezes, para “desconectado” 11. Para taxa “satisfatória” ocorre 5 vezes e “insatisfatória” 9. Com esses dados é possível calcular o ganho para cada atributo. Para estado os cálculos são:

$$Ganho(S, estado) = 0,940 - (|S_{conectado}|/14) * (Entropia(S_{conectado})) - (|S_{desconectado}|/14) * (Entropia(S_{desconectado}))$$

$$Ganho(S, estado): 0,940 - 0,476 - 0,274$$

$$Ganho(S, estado): 0,19$$

Para o atributo taxa os cálculos são:

$$Ganho(S, taxa) = 0,940 - (|S_{satisfatória}|/14) * (Entropia(S_{satisfatória})) - (|S_{insatisfatória}|/14) * (Entropia(S_{insatisfatória}))$$

$$Ganho(S, taxa): 0,940 - 0,530 - 0,442$$

$$Ganho(S, taxa): 0,032$$

Comparando o ganho do atributo estado com o ganho do atributo taxa, percebe-se que o resultado de estado é maior, ou seja, ele será usado no início da classificação como nó raiz. A aplicação recursiva do algoritmo leva a escolha dos próximos atributos, criando assim a árvore de decisão.

Para armazenar a árvore gerada, o MiD-Mobile emprega autômato finito determinístico (AFD). À luz da teoria dos grafos, autômato finito determinístico assemelha-se à árvore de decisão, por serem comparados a grafos orientados.

Assim os recursos do AFD foram adequados à árvore gerada, o que foi providencial visto que um autômato é um modelo computacional simples e de implementação algorítmica leve, mesmo perfil exigido para do middleware.

Um AFD é uma quintupla  $(Q, \Sigma, \delta, q_0, F)$ , onde  $Q$  é um conjunto finito de estados internos,  $\Sigma$  é o alfabeto,  $\delta$  é uma função de transição,  $q_0$  é um estado inicial e  $F$  é um conjunto de estados finais. Um AFD trata uma string de entrada dividida em quadros, cada quadro possui um símbolo do alfabeto ( $\Sigma$ ). Uma função de transição ( $\delta$ ) leva o foco do estado inicial para o próximo estado, considerando o símbolo informado, até que toda a string seja percorrida e o estado final identificado ( $f' \in F$ ).

Considerando o conteúdo dos atributos da árvore de decisão como um elemento dentro da string de um AFD, é possível visualizar um grafo da função de transição como um subconjunto de uma árvore de decisão, figura 3.



Fig. 3. Grafo de uma função de transição com string de entrada de um subconjunto da AD

Para serem inseridos na string, os valores obtidos pelo SGC, SGDM e SGD, representados na tabela 1, foram discretizados e representados por 0, 1 e 2. A tabela de transição de um AFD foi usada para armazenar o treinamento e a generalização da AD, figura 4.

Considere  $M = \langle K, \Sigma, \delta, i, F \rangle$ , onde  $K = \{q_0, q_1, q_2, q_3\}$ ,  $\Sigma = \{a, b\}$ ,  $i = q_0$ ,  $F = \{q_3\}$  e  $\delta: \{q_0, q_1, q_2, q_3\} \times \{a, b\} \rightarrow \{q_0, q_1, q_2, q_3\}$  é dada pela seguinte tabela de transição

	a	b
q0	q1	q2
q1	q0	q3
q2	q3	q0
q3	q2	q1

Fig. 4. Tabela de transição e de generalização gerada pelo AFD

Na tabela de transição estão todos os estados possíveis (coluna estado) gerados pela árvore resultante do conjunto de treinamento. Os símbolos do alfabeto 0, 1 e 2 possuem colunas próprias e após a aplicação do algoritmo guardam o próximo estado. A coluna plano é preenchida quando encontrar um estado que represente um nó folha.

Classificar novos casos é uma forma de verificar a capacidade de generalização da árvore, a figura 5 apresenta um ambiente de simulação onde os planos de execução obtidos a partir de novos conjuntos de dados podem ser aceitos ou rejeitados.

Fig. 5. Simulação da generalização de novos casos

Simular o aceite do plano de execução sugerido pelo SPE representa o término satisfatório da transação (commit) e rejeitar significa o seu insucesso (rollback).

O algoritmo pode se perder ao percorrer a tabela de transição e encontrar um estado que não reconhece o símbolo lido. Nesse caso o algoritmo do AFD foi adaptado para percorrer a árvore até o subconjunto conhecido, em seguida avaliar a incidência dos planos em todas as possíveis subárvores que constam da tabela de generalização, figura 4. A tabela de generalização tem funcionamento semelhante à tabela de transição, só que armazena a quantidade de vezes que aquele estado leva à aplicação dos planos de execução.

Ao quantificar o plano de execução mais comum às árvores subjacentes, o SPE faz sua indicação, o middleware permite a configuração de um limiar de tentativas, toda vez que a execução de um plano for rejeitada, um erro é computado e ao atingir o limiar especificado a mudança de plano é sugerida.

Para aumentar a capacidade de generalização da árvore, o SA do dispositivo móvel armazena os erros e acertos junto com a string de entrada de todas as  $t_{macs}$  enviadas ao middleware. Posteriormente o SA envia esses dados, através do STDm, à rede fixa para um retreinamento da árvore de decisão.

Após o novo treinamento, as informações são despachadas para a estação móvel aumentando as chances de sucesso das  $T_{macs}$  sucessoras. No dispositivo móvel, o MiD-Mobile emprega o algoritmo e a tabela de generalização e transição na composição do SPE. Já para o SA consta a tabela de erros e acertos. A implementação do simulador foi feita em Java Desktop.

## 5 Conclusões e Trabalhos Futuros

A parceria das técnicas utilizadas confere ao MiD-Mobile a leveza e agilidade necessárias à uma aplicação para dispositivos móveis com restrições significativas de recursos. A arquitetura baseada em serviços permite a adaptação, em tempo de execução, das transações ao plano de execução de maior chance de sucesso, além de poder agregar serviços futuros a partir de alterações controladas.

Entre as questões em aberto que constituem trabalhos futuros, está uma avaliação do ambiente que adota o MiD-Mobile comparado a outro que não o aplica, com o intuito de quantificar as transações concluídas com sucesso. Uma avaliação mais profunda sobre a discretização de valores da tabela de recursos faz-se necessária, bem como em alguns aspectos na aprendizagem, como a manipulação de atributos com diferentes custos (devido a pluralidade da conectividade sem fio) e o tratamento de valores desconhecidos (para considerar dados com completude e validade indefinidos).

## Referências

- [1] Chrysanthis, P. K.: Transaction Processing in Mobile Computing Environment. In IEEE Workshop on APADS, Princeton, USA, October (1993).
- [2] Côrtes, S. C.: Um modelo de transações para integração de SGBD a um ambiente de computação móvel. Departamento de Informática, PUC - Rio de Janeiro, (2004).

- [3] Couloris, G., Dollimore, J., Kindberg, T.: *Sistemas distribuídos: conceitos e projetos*. Bookman, Porto Alegre (2007).
- [4] Dirckze, R. e Gruenwald, L.: A pre-serialization transaction management technique for mobile multidatabases. *MONET*, vol. 5, no. 4, (2000).
- [5] Dunham, M. H., Helal, A. e Balakrishnan, S.: A mobile transaction model that captures both the data and movement behavior. *ACM/Baltzer Journal on Special Topics in Mobile Networks and Applications*, vol.2, (1997).
- [6] Dunham, M., Helal, A.: *Mobile Computing and Databases: Anything New?*. *ACM SIGMOD Record*, vol 24, Issue 4. (1995).
- [7] Ferreira, J. E., Finger, M.: *Controle de concorrência e distribuição de dados: a teoria clássica, suas limitações e extensões modernas*. IME-USP, São Paulo (2000).
- [8] Gray, J., Helland, P., O'neil, P. e Shasha, D.: The dangers of replication and a solution. In *ACOMA SIGMOD Conference Montreal, Canada: June*, (1996).
- [9] Ku, K., Kim, Y.: Moflex transaction model for mobile heterogeneous multidatabase systems". In *IEEE Workshop on Research Issues in Data Engineering*, San Diego, USA, Feb, (2000).
- [10] Lee, M., e Helal, S.: HiCoMo High Commit Mobile Transactions. *Kluwer Academic Publishers Distributed and Parallel Databases (DAPD)*, vol . 11, no. 3, (2002).
- [11] Lu, Q. e Satyanarayanan, M.: Isolation-only transactions for mobile computing. In *ACM Operating Systems Review*, vol. 28, n. 3, (1994).
- [12] Madria, S. K. e Bhargava, B.: "A transaction model for improving data availability in mobile computing". *Kluwer Academic Publishers Distributed and Parallel Databases (DAPD)*, vol . 10, no. 2, (2001).
- [13] Özsü, M.T., Valduriez, P.: *Princípios de sistemas de bancos de dados distribuídos*. Campus, Rio de Janeiro (2001).
- [14] Pitoura, E. e Bhargava, B.: Maintaining Consistency of Data in Mobile Computing Environments. In *proceedings of 15th IEEE International Conference on Distributed Computing Systems*, June, (1995).
- [15] Rocha, T., Toledo, M.: *Estudo de Modelos de Transação para o ambiente de Computação Móvel*. Instituto de Computação, Unicamp, Relatório Técnico IC-05-25, (2005).
- [16] Rocha, T., Toledo, M.: Mecanismos de Adaptação para Transações em Ambientes de Computação Móvel. In *IEEE Latin America Transactions*, vol. 5, n.8. (2007).
- [17] Serrano-Alvarado, P., Roncancio, C., Adiba, M.: *A Survey of Mobile Transactions*. *Kluwer Academic Publishers Distributed and Parallel Databases* (2004).
- [18] Serrano-Alvarado, P.: *Transactions Adaptables pour les Environments Mobiles*. Université Joseph Fourier (2004).
- [19] Walborn, G. e Chrysanthis, P.: Suporting Semantics-Based Transaction Processing in Mobile Database Applications. Dans *Symposium on Reliable Distributed Systems (SRDS)*, Bad Neuenahr, Germany, September, (1995).
- [20] Walborn, G. e Chrysanthis, P.: Transaction Processing in PRO-MOTION. *Proc. Of the Symposium on Applied Computing*. *ACM SAC V. 11*, p.389-398, (1998).
- [21] Yeo, L. e Zaslavsky, H.: "A Submission of Transactions from Mobile Workstations in a Cooperative Multidatabase Processing Environment". *Proceedings of the 14th Internacional Conference on Distributed Computing Systems*, (1994).