



Ministério da
Ciência e Tecnologia



INPE-15675-TDI/1449

MODELO E ARQUITETURA PARA ADAPTAÇÃO DINÂMICA DE EXECUÇÃO DE TRANSAÇÕES EM AMBIENTES DE COMPUTAÇÃO MÓVEL

Daniela Eloise Flôr

Tese de Doutorado do Curso de Pós-Graduação em Computação Aplicada,
orientada pelos Drs. Maurício Gonçalves Vieira Ferreira, Nilson Sant' Anna e Solon
Venâncio de Carvalho, aprovada em 17 de novembro de 2008.

Registro do documento original:

<<http://urlib.net/sid.inpe.br/mtc-m18@80/2008/11.17.17.43>>

INPE
São José dos Campos
2009

PUBLICADO POR:

Instituto Nacional de Pesquisas Espaciais - INPE

Gabinete do Diretor (GB)

Serviço de Informação e Documentação (SID)

Caixa Postal 515 - CEP 12.245-970

São José dos Campos - SP - Brasil

Tel.:(012) 3945-6911/6923

Fax: (012) 3945-6919

E-mail: pubtc@sid.inpe.br

CONSELHO DE EDITORAÇÃO:

Presidente:

Dr. Gerald Jean Francis Banon - Coordenação Observação da Terra (OBT)

Membros:

Dr^a Maria do Carmo de Andrade Nono - Conselho de Pós-Graduação

Dr. Haroldo Fraga de Campos Velho - Centro de Tecnologias Especiais (CTE)

Dr^a Inez Staciarini Batista - Coordenação Ciências Espaciais e Atmosféricas (CEA)

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Dr. Ralf Gielow - Centro de Previsão de Tempo e Estudos Climáticos (CPT)

Dr. Wilson Yamaguti - Coordenação Engenharia e Tecnologia Espacial (ETE)

BIBLIOTECA DIGITAL:

Dr. Gerald Jean Francis Banon - Coordenação de Observação da Terra (OBT)

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Jefferson Andrade Ancelmo - Serviço de Informação e Documentação (SID)

Simone A. Del-Ducca Barbedo - Serviço de Informação e Documentação (SID)

REVISÃO E NORMALIZAÇÃO DOCUMENTÁRIA:

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Marilúcia Santos Melo Cid - Serviço de Informação e Documentação (SID)

Yolanda Ribeiro da Silva Souza - Serviço de Informação e Documentação (SID)

EDITORAÇÃO ELETRÔNICA:

Viveca Sant´Ana Lemos - Serviço de Informação e Documentação (SID)



Ministério da
Ciência e Tecnologia



INPE-15675-TDI/1449

MODELO E ARQUITETURA PARA ADAPTAÇÃO DINÂMICA DE EXECUÇÃO DE TRANSAÇÕES EM AMBIENTES DE COMPUTAÇÃO MÓVEL

Daniela Eloise Flôr

Tese de Doutorado do Curso de Pós-Graduação em Computação Aplicada,
orientada pelos Drs. Maurício Gonçalves Vieira Ferreira, Nilson Sant' Anna e Solon
Venâncio de Carvalho, aprovada em 17 de novembro de 2008.

Registro do documento original:

<<http://urlib.net/sid.inpe.br/mtc-m18@80/2008/11.17.17.43>>

INPE
São José dos Campos
2009

F661m Flôr, Daniela Eloise.

Modelo e arquitetura para adaptação dinâmica de execução de transações em ambientes de computação móvel / Daniela Eloise Flôr. – São José dos Campos: INPE, 2009.

177p. ; (INPE-15675-TDI/1449)

Tese (Computação Aplicada) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2008.

1. Computação móvel. 2. Sistema gerenciador de bancos de dados. 3. Transações móveis. 4. Modelo de transações. 5. Bancos de dados móveis. I. Título.

CDU 004.451.7.031.43

Copyright © 2009 do MCT/INPE. Nenhuma parte desta publicação pode ser reproduzida, armazenada em um sistema de recuperação, ou transmitida sob qualquer forma ou por qualquer meio, eletrônico, mecânico, fotográfico, microfílmico, reprográfico ou outros, sem a permissão escrita da Editora, com exceção de qualquer material fornecido especificamente no propósito de ser entrado e executado num sistema computacional, para o uso exclusivo do leitor da obra.

Copyright © 2009 by MCT/INPE. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, microfilming, recording or otherwise, without written permission from the Publisher, with the exception of any material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use of the reader of the work.

Aprovado (a) pela Banca Examinadora
em cumprimento ao requisito exigido para
obtenção do Título de Doutor(a) em
Computação Aplicada

Dr. José Demisio Simões da Silva

Presidente / INPE / SJC Campos - SP

Dr. Solon Venâncio de Carvalho

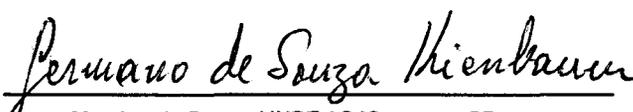
Orientador(a) / INPE / SJC Campos - SP

Dr. Mauricio Gonçalves Vieira Ferreira

Orientador(a) / INPE / SJC Campos - SP

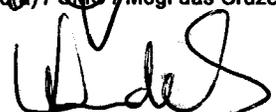
Dr. Nilson Sant'Anna

Orientador(a) / INPE / SJC Campos - SP

Dr. Germano de Souza Kienbaum

Membro da Banca / INPE / SJC Campos - SP

Dr. Henrique Jesus Quintino de Oliveira

Convidado(a) / UNIC / Mogi das Cruzes - SP

Dr. Clovis Torres Fernandes

Convidado(a) / ITA / SJC Campos - SP

Aluno (a): Daniela Eloise Flôr

São José dos Campos, 17 de dezembro de 2008

“Doutorado é assim mesmo!!!”

Maurício Gonçalves Vieira Ferreira

Às mulheres que suavizaram meus caminhos: minha mãe, irmãs e sogra!

Àquele que está dentro do meu coração e fora da minha visão: meu pai!

Aos homens que são a razão do meu empreendimento: meu marido e meu filho!

AGRADECIMENTOS

Agradecer é fazer jus aos que merecem e estes são muitos:

- a Quem abriu minha mente e meus caminhos: *Deus*, nosso Criador;
- àquele que não me deixou desistir, fazendo-me crer ser mais forte do que realmente sou: *Marcelo*, meu amor;
- àquele que veio para me incentivar: *Henrique*, meu tesouro;
- àqueles que plantaram a semente da caminhada contínua do conhecimento: *Eva e Baltazar*, meus pais;
- àqueles que se propuseram a ouvir e calar: *Luciana, Silvia, Cláudia e Priscila*, minhas irmãs;
- àqueles que compartilharam do mesmo sonho: *Heloise, Giani, Késsia, Íris, Andréia, Adriana*, minhas eternas amigas;
- àqueles que participaram ativamente desse projeto: *Yandre, Munif, Roni, Ricardo, Antônio, Danielle e Maria Olímpia*, meus parceiros;
- àquela que aumentou o número de horas dos meus dias: *Profa. Claudete*, Coordenadora do Curso de Sistemas de Informação da Unipar Campus Paranaíba;
- àqueles que me conduziram nessa jornada: *Maurício, Nilson e Solon*, meus orientadores;
- e a todos àqueles que torceram por mim!!!

RESUMO

As características inerentes ao contexto da computação móvel conduzem a pesquisas desafiantes, inclusive em transações de banco de dados. Fatores como a instabilidade do meio sem fio, a heterogeneidade de recursos de computação dos dispositivos móveis e a inflexibilidade dos modelos transacionais propostos na literatura, culminaram na motivação para esta pesquisa. Este trabalho propõe uma acomodação harmoniosa entre aplicações que processam transações e o dinamismo do ambiente de computação móvel através do MiD-Mobile. O MiD-Mobile é um *middleware* baseado em serviços que equaciona as transações com a disponibilidade de recursos computacionais, sem sacrificar as funcionalidades do software. Sua função principal é garantir a transparência de execução da transação isolando o desenvolvimento de software das conseqüências impostas pela variação de recursos computacionais e de comunicação. Outrossim, gerencia e aplica o plano de execução mais apropriado para as transações móveis adaptadas ao contexto (T_{MACs}). Uma T_{MAC} é uma transação móvel adequada ao plano de execução. A união do modelo de transações móveis adaptadas ao contexto ao MiD-Mobile viabiliza a adaptação colaborativa que provê sensibilidade ao contexto, êxito nas transações, uso consciente de recursos e diminuição do ônus do usuário. Um estudo de caso legitima as propostas ora apresentadas.

MODEL AND ARCHITECTURE TO DINAMIC ADAPTIVE OF TRANSACTION PROCESSING IN MOBILE COMPUTING ENVIRONMENT

ABSTRACT

The mobile computation context has special characteristics and these characteristics lead to challenger researches, including database transactions. Factors as the dynamism of the wireless environment, the lack of the mobile devices computations resources and the inflexibility of the proposed transactional model in literature, carry out the motivation for this research. This research presents an accordant arrangement among applications that process transactions and dynamic specific of mobile computation environment through the MiD-Mobile. The MiD-Mobile is a *middleware* software based on services that equate transactions with availability of computation resources, without impairment of the software functionalities. Its main function is to isolate the software development from the evaluation activities of computational resources and dynamic monitoring of the environment as well, besides it manages and it decides about the execution of more appropriated plan for the mobile transactions applied to context (T_{MAC}). A T_{MAC} is a transaction suitable for the mobile transactional homonym model proposed in this research. The union of the T_{MACs} model to the MiD-Mobile enables collaborative resources and decrease of the user's tax. A case study legitimates the proposals proposed in this research.

SUMÁRIO

Pág.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS E ABREVIATURAS

1	INTRODUÇÃO	29
1.1	Motivação e objetivos do trabalho de pesquisa.....	32
1.2	Metodologia de desenvolvimento e organização do trabalho de pesquisa.....	36
2	TRANSAÇÕES EM AMBIENTE DE COMPUTAÇÃO MÓVEL	39
2.1	Efeitos da mobilidade nas propriedades ACID.....	47
2.2	Modelos para transações móveis.....	48
2.2.1	Modelo IOT.....	48
2.2.2	Modelo Pro-motion.....	48
2.2.3	Modelo <i>Clustering</i>	49
2.2.4	Modelo <i>Two-tier replication</i>	50
2.2.5	Modelo <i>Reporting e Co-transations</i>	50
2.2.6	Modelo Semântico	51
2.2.7	Modelo <i>Prewrite</i>	51
2.2.8	Modelo <i>Kangaroo</i>	52
2.2.9	Modelo <i>Pre-serialization</i>	52
2.2.10	Modelo MDSTPM.....	53
2.2.11	Modelo Moflex.....	53
2.2.12	Modelo HiCoMo.....	54
2.2.13	Modelo TM.....	54

2.3	Classificação dos modelos de transações móveis.....	58
2.4	Considerações.....	61
3	ADAPTAÇÃO DE TRANSAÇÕES MÓVEIS.....	65
3.1	Comportamento adaptativo.....	65
3.2	Modelos transacionais adaptativos.....	67
3.2.1	O modelo AMT.....	67
3.2.2	Mecanismos de adaptação e a PGTA.....	70
3.3	Considerações.....	74
4	MID-MOBILE.....	77
4.1	Arquitetura do Mid-Mobile.....	78
4.1.1	Serviço de Gerenciamento de Recursos	81
4.1.2	Serviço de Gerenciamento da Comunicação.....	82
4.1.3	Serviço de Gerenciamento dos Dados	83
4.1.4	Serviço de Plano de Execução e de Aprendizagem.....	84
4.1.5	Serviço Gerenciador de Transações Móveis Adaptadas ao Contexto... ..	97
4.2	Serviço de Transmissão de Dados e Mensagens.....	98
4.3	Considerações.....	101
5.	O MODELO DE TRANSAÇÕES MÓVEIS ADAPTADAS AO	
	CONTEXTO.....	103
5.1	A estrutura de uma T_{MAC}	104
5.1.1	Plano de execução isolada de uma T_{MAC}	106
5.1.2	Plano de execução remota de uma T_{MAC}	109
5.1.3	Plano de execução compartilhada de uma T_{MAC}	111
5.1.3.1	Garantia das propriedades ACID para T_{MACs} multicomponentes... ..	111
5.2	Considerações.....	119

6 ESTUDO DE CASO	123
6.1 BancoOnline.....	123
6.2 BancoME.....	130
6.3 Considerações.....	144
7. CONCLUSÕES E TRABALHOS FUTUROS	147
7.1 Contribuições desta pesquisa.....	148
7.2 Trabalhos Futuros.....	151
REFERÊNCIAS BIBLIOGRÁFICAS	153
APÊNDICE A – TEORIA DA SERIALIZAÇÃO	159
A.1 Serialização de Conflito.....	161
A.2 Visão Serializada.....	161
APÊNDICE B – DIAGRAMAS DE SEQÜÊNCIA DAS APLICAÇÕES BANCOONLINE E BANCOME	163

LISTA DE FIGURAS

	<u>Pág.</u>
1.1 - Arquitetura genérica de um sistema de computação móvel.....	29
1.2 - Cenário para um ambiente de computação móvel.....	30
1.3 - Dispositivos portáteis que acessam sistemas de computação móvel.....	33
1.4 - Eixos usados na adaptação de transações móveis.....	34
2.1 - Ambiente com banco de dados centralizado.....	39
2.2 - Diagrama de estados de uma transação.....	39
2.3 - Ambiente com banco de dados distribuídos.....	40
2.4 - Modelo de transações distribuídas.....	41
2.5 - Diagrama de transição de estado do coordenador no protocolo 2PC.....	42
2.6 - Diagrama de transição de estado dos participantes no protocolo 2PC....	42
2.7 - Diagrama de seqüência do protocolo de consolidação em duas fases	43
2.8 - Gráfico de bloqueio 2PL	44
2.9 - Gráfico do bloqueio 2PL rigoroso.....	44
2.10 - Diagrama de seqüência do protocolo de bloqueio 2PL rigoroso.....	45
2.11 - Representação de bloqueio de réplica de dados pelo ROWA.....	45
2.12 - Organização do modelo TM.....	55
2.13 - Esquema geral da arquitetura proposta no modelo TM.....	57
3.1 - Eixos de adaptação.....	65
3.2 - Componentes de uma AMT.....	68
3.3 - Componentes de uma AMT e os descritores do ambiente.....	69
3.4 - Arquitetura Global do MTS.....	70
3.5 - Visão geral da PGTA.....	73
4.1 - Fatores consideradas para a adaptação das transações.....	73
4.2 - Arquitetura do MiD-Mobile.....	79
4.3 - Arquivo UAProf de um dispositivo móvel.....	82
4.4 - Tecnologias de rede de comunicação no protocolo CC/PP.....	83
4.5 - Método de sincronização solicitado pelo dispositivo móvel.....	83
4.6 - Método de sincronização enviado a rede fixa.....	84
4.7 - Protocolo de Consistência Global	84
4.8 - Componentes de uma Árvore de Decisão.....	85
4.9 - Modelo de árvore de decisão e de conjunto de treinamento.....	86
4.10 - Menu principal do MiD-Mobile.....	89
4.11 - Conjunto de treinamento usado na geração da árvore de decisão.....	89
4.12 - Opção de treinamento da árvore de decisão no Menu Arquivo.....	90
4.13 - Interface do Serviço de Aprendizagem.....	90
4.14 - Interface de exibição de planos generalizadas ou treinados.....	91
4.15 - Controle de limiares de confiança.....	91
4.16 - Diagrama e tabela de transições do AFD.....	93
4.17 - Tabela de transição do AFD, da árvore de decisão e do algoritmo de generalização.....	96
4.18 - Código-fonte do controle do tempo.....	97
4.19 - Código-fonte da verificação de resultados das subtmacs	98

4.20 - Conexão solicitada pelo cliente.....	98
4.21 - Conexão reconhecida pelo servidor.....	98
4.22 - Nova conexão solicitada pelo cliente.....	99
4.23 - Conexões identificadas.....	99
4.24 - Encerramento da comunicação.....	99
4.25 - Interrupção da comunicação no cliente.....	99
4.26 - Mensagens enviadas pelo STDM da rede fixa.....	100
4.27 - Mensagens não enviadas devido a desconexão.....	100
4.28 - Pacote do <i>middleware</i> disponibilizado junto com a aplicação.....	101
5.1 - Estruturas de uma T_{MAC}	104
5.2 - Estados de operações desconectadas.....	106
5.3 - Diagrama de seqüência do protocolo PCG.....	107
5.4 - Diagrama de seqüência do protocolo $2PC\Delta t$	113
5.5 - Componentes de uma T_{MAC}	115
5.6 - Protocolo de bloqueio em duas fases temporário.....	116
6.1 - Diagrama de Caso de Uso das funcionalidades do BancoOnline.....	123
6.2 - Interface principal do BancoOnline.....	124
6.3 - Interface pagamento.....	124
6.4 - Interface extrato de conta corrente.....	125
6.5 - Operações de simulação e empréstimo.....	126
6.6 - Interface de parcelamentos disponíveis para simulação de empréstimo.....	126
6.7 - Interface de avaliação de empréstimo.....	127
6.8 - Interface lançamento de empréstimo em conta corrente.....	127
6.9 - Diagrama de classes do BancoOnline.....	128
6.10 - Estrutura do BancoOnline.....	129
6.11 - Visão geral dos elementos usados na estrutura do BancoOnline.....	129
6.12 - POJO com annotation.....	130
6.13 - Interface do BancoMe no emulador WTK.....	131
6.14 - Execução das operações disponíveis do BancoMe.....	131
6.15 - Resultado da execução das operações do BancoMe.....	132
6.16 - Extrato solicitado ao BancoOnline após as operações do BancoMe...	132
6.17 - Estrutura da aplicação BancoME.....	133
6.18 - Estrutura da aplicação BancoOnline com suporte a Web Service.....	133
6.19 - Estrutura do BancoMe com suporte a Web Service.....	134
6.20 - Instância do Web Service.....	134
6.21 - ContaTO.....	134
6.22 - Classe ContaTO.....	135
6.23 - Pagamento realizado a partir do plano de execução isolada.....	136
6.25 - Seqüência de interfaces para empréstimo e sincronização com êxito.	137
6.26 - Seqüência de interfaces para empréstimo e sincronização sem êxito.	137
6.27 - Transação multicomponente para extrato de conta corrente.....	138
6.28 - Transação multicomponente de extrato com $subT_{MACs}$ de acesso.....	138
6.29 - Transação multicomponente para pagamento de contas.....	139
6.30 - Transação multicomponente de pagamento com $subT_{MACs}$ de acesso	139
6.31 - Transação multicomponente para empréstimo.....	140

6.32 - Transação multicomponente empréstimo com $subT_{MACs}$ de acesso....	140
6.33 - Transação multicomponente para simulação de empréstimo.....	141
6.34 - Transação multicomponente de simulação de empréstimo com $subT_{MAC}$ de acesso.....	141
6.35 - Log de execução operação de empréstimo no modo compartilhado...	142
6.36 - Informação do Δt para a T_{MAC}	142
6.37 - Dependência de $subT_{MACs}$	142
6.38 - Log de execução com enfoque nas dependências.....	143
6.39 - Log de execução da operação de empréstimo no modo compartilhado.....	143
6.40 - Log de execução da operação de empréstimo e configuração do Δt ..	144
B.1 - Diagrama de seqüência de extrato no plano de execução compartilhada.....	164
B.2 - Diagrama de seqüência de pagamento no plano de execução compartilhada.....	165
B.3 - Diagrama de seqüência de empréstimo no plano de execução compartilhada.....	166
B.4 - Diagrama de seqüência da transação extrato.....	168
B.5 - Diagrama de seqüência da transação simulação e empréstimo.....	169
B.6 - Diagrama de seqüência da transação pagamento.....	170
B.7 - Diagrama de seqüência de pagamento no plano de execução remota..	171
B.8 - Diagrama de seqüência de simulação e empréstimo no plano de execução remota.....	172
B.9 - Diagrama de seqüência de extrato no plano de execução remota.....	173
B.10 - Diagrama de seqüência pagamento no plano de execução isolada....	174
B.11 - Diagrama de seqüência de simulação e empréstimo no plano de execução isolada.....	175
B.12 - Diagrama de seqüência de extrato no plano de execução isolada.....	176

LISTA DE TABELAS

	<u>Pág.</u>
2.1 – Classificação dos modelos de transação de acordo com o local de processamento.....	58
6.4– Classificação dos modelos de transação de acordo com as características do ambiente de computação móvel.....	59
6.5– Classificação dos modelos de transação de acordo com as estratégias adotadas quanto às propriedades ACID.....	60
6.6– Classificação dos modelos diante da estrutura e da confiabilidade.....	61
3.1 – Dimensões do Serviço de Transação Móvel.....	69
3.2 – Níveis de Isolamento.....	71
4.1 – Aspectos avaliados durante a adaptação.....	80
6.7– Conjunto de treinamento de exemplo.....	88
5.1 – Plano de Execução de uma T_{MAC}	105
5.2 – Conflito entre operações.....	110
5.3 – Matriz de compatibilidade de bloqueios.....	111
5.4 – Estratégias para a manutenção das propriedades ACID das T_{MACs}	119
6.8– $SubT_{MACs}$ da operação de extrato.....	138
6.2 – $SubT_{MACs}$ da operação de pagamento.....	139
6.9– $SubT_{MACs}$ da operação de empréstimo.....	140
6.10– $SubT_{MACs}$ da operação de simulação de empréstimo.....	141

LISTA DE SIGLAS E ABREVIATURAS

SCM	Sistemas de Computação Móvel
ACID	Atomicidade, Consistência, Isolamento e Durabilidade
2PC	<i>Protocol Two-Phase Commit</i>
T _{MAC}	Transação Móvel Adaptada ao Contexto
SGBD	Sistema Gerenciador de Banco de Dados
SGBDD	Sistema Gerenciador de Banco de Dados Distribuído
2PL	<i>Protocol Two-Phase Locking</i>
ROWA	<i>Read Once Write All</i>
IOT	<i>Isolation-Only Transaction</i>
Pro-Motion	<i>Pro-Active Management of Mobile Transactions</i>
AAD	Agente de Acesso a Dados
MDSTPM	<i>Multidatabase Transaction Processing Manager</i>
MQF	<i>Message and Queuing Facility</i>
HiCoMo	<i>High Commit Mobile</i>
AMT	<i>Adaptable \square óbile Transactions</i>
T	Componente Transacional
CT	Transações Compensatórias
ES	Estratégias de execução
EA	Execuções alternativas
ED	Descritores de ambiente
EP	Planos de Execução
UM	Unidade Móvel
UF	Unidade Fixa
MTS	<i>\square óbile Transaction Service</i>
TM	<i>Transaction Manager</i>
TMLC	Transações Móveis Locais Compensáveis
TMG	Transações Móveis Globais
TCM	Transações Coordenadoras de Mobilidade
TCBD	Transação Coordenadora de Banco de Dados

TSMC	Transações de Suporte à Mobilidade do Cliente
TL	Transações Locais
PGTA	Plataforma de Gerenciamento de Transações Adaptáveis
ANSI	<i>American National Standards Institute</i>
SQL	<i>Structured Query Language</i>
R	Remoto
L	Local
LR	Local-Remoto
PVML	Protocolo de Validação do Modo Local
PDA	<i>Personal Digital Assistant</i>
SGT _{MAC}	Serviço Gerenciador de Transações Móveis Adaptadas ao Contexto
SPE	Serviço de Plano de Execução
SGR	Serviço de Gerenciamento de Recursos
SGR	Serviço de Gerenciamento de Comunicação
SGD	Serviço de Gerenciamento de Dados
AS	Serviço de Aprendizagem
CC/PP	<i>Composite Capability / Preference Profiles</i>
Wi-Fi	<i>Wireless Fidelity</i>
Wi-Max	<i>Worldwide Interoperability for Microwave Access</i>
SNMP	<i>Simple Network Management Protocol</i>
AD	Árvore de Decisão
AFD	Autômato Finito Determinístico
subT _{MAC}	Subtransação Móvel Adaptada ao Contexto
PCG	Protocolo de Consistência Global
DM	Dispositivo Móvel
RF	Rede Fixa
Δt	Tempo de execução global
2PC Δt	Protocolo de efetivação em duas camadas com tempo de execução global
Δst	Tempo de execução parcial

JSP	<i>Java Servlet Page</i>
JavaME	<i>Java Micro Edition</i>
JPA	<i>Java Persistence API</i>
API	<i>Application Programming Interface</i>
POJO	<i>Plain Old Java Object</i>
DAO	<i>Data Access Object</i>
TO	<i>Transfer Object</i>

1 INTRODUÇÃO

A convergência das tecnologias de comunicação sem fio com a computação portátil alavancou a Computação Móvel. Para Dunham e Helal (1995) a computação móvel é considerada uma ampliação da computação distribuída e se caracteriza por uma arquitetura composta de computadores fixos e móveis e comunicação com e sem fio, conforme a Figura 1.1

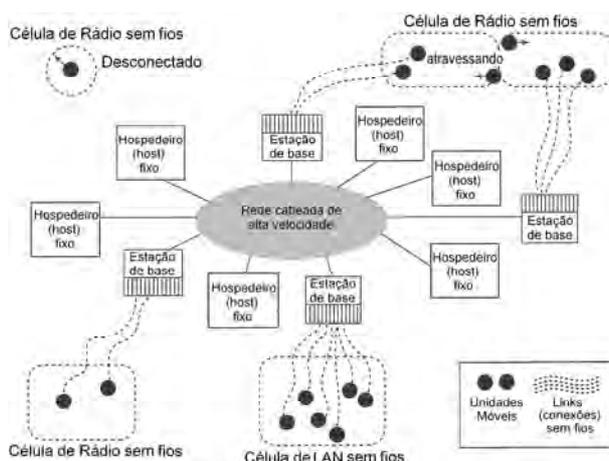


Figura 1.1 - Arquitetura genérica de um sistema de computação móvel

Fonte: Adaptado de Dunham e Helal (1995)

Os hospedeiros (*hosts*) fixos conectam-se por cabos e formam a rede fixa. As estações de base são equipamentos dotados de interface de comunicação sem fio (*wireless*). Sua principal função é o gerenciamento dos dispositivos móveis que estão em sua célula. Célula é o nome dado à área de abrangência da estação de base. Os computadores móveis são dispositivos portáteis que se movimentam livremente e, por intermédio das estações de base, acessam a rede fixa via conectividade sem fio.

Os Sistemas de Computação Móvel (SCM) tornaram-se populares ao permitirem o acesso a dados remotos, visualização de vídeos e e-mails, entre outras funções. Contudo, o acréscimo da portabilidade e da comunicação sem fio em dispositivos com capacidade de processamento e armazenamento acarretou em novos desafios.

A portabilidade veio com a miniaturização dos computadores, mas resultou em dispositivos com restrições de recursos. A comunicação sem fio apesar de oferecer liberdade, não garante a mesma qualidade de serviço que as redes cabeadas. Já a mobilidade dos usuários acrescenta novas questões como modo de endereçamento, gerência de posicionamento de usuários móveis, consultas dependentes de localização, entre outras.

Apesar dos desafios e limitações, é crescente o interesse em sistemas de computação que processem transações e atuem nesse cenário. Aplicações acessíveis a qualquer momento e de qualquer lugar, como a da abordagem explorada na Figura 1.2, podem ser beneficiadas pelos esforços empreendidos em pesquisas sobre banco de dados em ambiente de computação móvel.

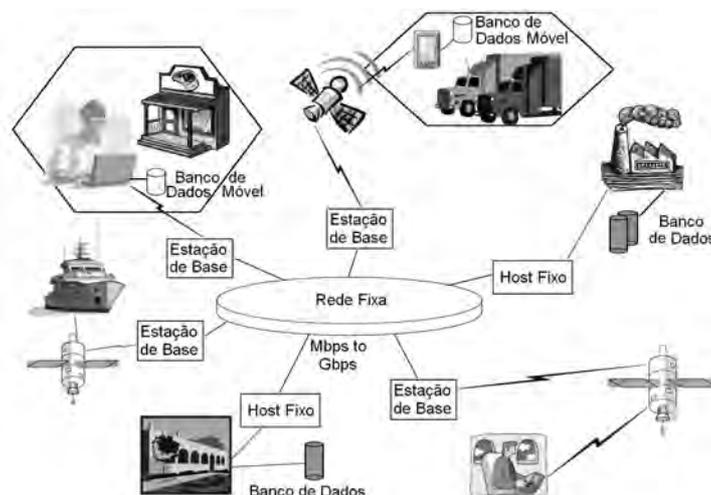


Figura 1.2 – Cenário para um ambiente de computação móvel

Fonte: Adaptado de Côrtes (2004)

A complexidade na criação de aplicações fortemente sujeitas a perda de conexão, variabilidade da largura de banda e interrupções indesejadas devido a escassez de energia, exige especial atenção com questões relativas a gerência de dados. Este trabalho aborda as características do ambiente de computação móvel que inviabilizam o uso do modelo tradicional de processamento das transações, propondo um novo modelo de processamento e uma ferramenta que auxilie na sua execução.

O conceito de transação para Melo et al. (1997), diz respeito a uma seqüência de operações realizadas sobre um conjunto de dados, delimitada por um comando de inicialização e outro de finalização. Tais operações são tratadas como unidades computacionais confiáveis relacionadas à inserção e atualização de dados.

Para assegurar que transações sejam unidades computacionais confiáveis é necessário manter a integridade e confiabilidade dos dados, para esse fim foram propostas as propriedades ACID, acrônimo de atomicidade, consistência, isolamento e durabilidade:

- a) atomicidade: uma transação deve ser uma unidade atômica de processamento sendo completamente executada ou descartada;
- b) consistência: a transação deve levar a base de dados de um estado consistente a outro;
- c) isolamento: a execução de uma transação não deve sofrer interferência de outra que ocorra concorrentemente;
- d) durabilidade: depois de verificado o sucesso na execução da transação, as mudanças operadas por ela na base de dados devem persistir.

A popularização de sistemas de computação distribuída exigiu novas formas de garantir a confiabilidade dos dados, já que o compartilhamento da computação não foi previsto inicialmente nas propriedades ACID. Alterações foram realizadas para que tais propriedades ainda fossem um mecanismo de garantia de sistemas confiáveis em computação distribuída. Mas os pressupostos teóricos que afirmam o sucesso de transações centralizadas ou distribuídas, como o protocolo de efetivação em duas fases (2PC *Two-Phase Commit*), não se ajustam às freqüentes desconexões do ambiente móvel.

No contexto da computação móvel, a coordenação da computação e a transparência de distribuição deixaram de ser as únicas preocupações dos sistemas de computação, somam-se a elas o tratamento da localização dos dados e o movimento do dispositivo portátil. À luz dessa mudança de paradigma, as particularidades no processamento das transações em ambiente de computação móvel (transações móveis) precisam ser reavaliadas.

1.1 Motivação e objetivos do trabalho de pesquisa

As transações dependem da arquitetura de distribuição dos computadores e dos dados, além do local de processamento das mesmas. Inicialmente as transações eram solicitadas em terminais e realizadas centralizadamente em *mainframes*.

A popularização de computadores pessoais a preços acessíveis permitiu alocar equipamentos para funções específicas, como os servidores de banco de dados que centralizavam o processamento. Quando os dados foram espalhados em diferentes locais, o processamento centralizado deu lugar ao processamento distribuído. O advento da mobilidade e da invisibilidade momentânea de computadores hospedeiros de dados se beneficia dos dois tipos de processamento.

O processamento autônomo acontece quando o usuário não visualiza dados localizados na rede fixa e ainda assim executa suas operações localmente. Quando a comunicação está plena, o computador móvel pode atuar como um computador fixo, colaborando em processamentos distribuídos.

Apesar da pluralidade de arquiteturas em uso, o presente trabalho baseia-se em sistemas de computação móvel, no qual ainda são bem-vindas muitas contribuições. Para delinear o contexto deste estudo, o cenário a seguir servirá para apresentar as motivações e definir os objetivos desta pesquisa.

Para vários sistemas de computação a disponibilidade de acesso e o compartilhamento dos dados estendem-se a usuários de computadores móveis, por vezes com funcionalidades diferentes do equivalente disponível para computadores fixos, mas em compensação com independência de localização, meio de comunicação, infra-estrutura física e momento.

Nestes casos o banco de dados pode estar distribuído entre as estações da rede fixa e replicado entre os dois extremos, rede com e sem fio. Na ausência da mobilidade e em períodos de forte estabilidade da comunicação sem fio, transações iniciadas por computadores fixos ou móveis não divergem. Em momentos de instabilidade ou ausência total de comunicação, a replicação dos dados permite trabalho autônomo.

Vários detalhes podem interferir nessa descrição justificando a atenção especial dada ao processamento de transação nesse cenário. Al-Bar e Wakeman (2001) detalham essas interferências sob três dimensões: dispositivos, comunicação e mobilidade.

Os dispositivos móveis possuem recursos de computação mais restritivos e dependem de fonte de energia finita. São materializados em assistentes pessoais digitais (PDA), telefones celulares, telefones inteligentes, entre outros. A heterogeneidade de modelos com perfis e funções diferentes é grande, dificultando qualquer iniciativa de padronização que uniformize o acesso às aplicações e conseqüentemente o processamento de transações. A variedade de equipamentos utilizados para acesso aos sistemas de computação móvel é apresentada na Figura 1.3.



Figura 1.3 – Dispositivos portáteis que acessam sistemas de computação móvel

A conectividade sem fio é mais frágil e vulnerável se comparada a conexões cabeadas, possuem limitada largura de banda e área de cobertura, além de maior latência. As características da comunicação sem fio exigem maior consumo de bateria dos dispositivos e a mobilidade dos mesmos aumenta as chances de desconexões, minimizando a confiabilidade das transações.

A mobilidade dos dispositivos portáteis pode deixá-los momentaneamente em áreas de sombra. Essas áreas são espaços geográficos com atenuação do sinal de comunicação que podem implicar em falta de acesso temporário aos dados disponíveis na rede fixa.

Diante do cenário apresentado, o presente trabalho propõe um modelo de transações móveis adaptáveis ao contexto (T_{MAC}) que flexibiliza a estrutura interna da transação para maximizar as chances de êxito do processamento, mantendo ao máximo as propriedades ACID. Para tanto, é necessário um mecanismo adicional, também proposto nesta tese, capaz de explorar corretamente os dados e recursos disponíveis no dispositivo móvel, além do estado da conectividade sem fio. Os elementos considerados durante a adaptação formam a Figura 1.4.

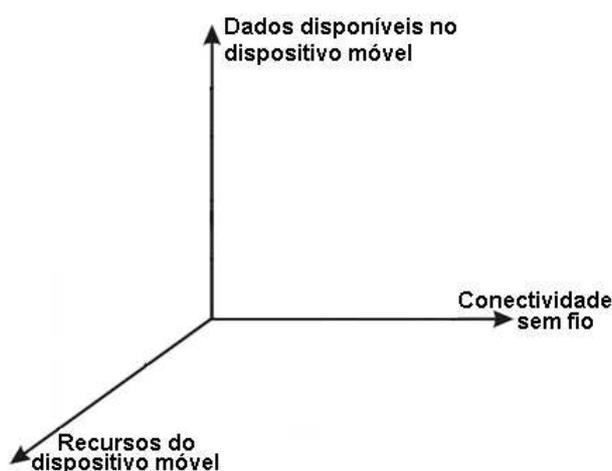


Figura 1.4 – Eixos usados na adaptação de transações móveis

A utilização dos eixos "conectividade sem fio" e "recursos do dispositivo móvel" foi explorada em trabalhos correlatos apresentados no Capítulo 2. A inserção do eixo "dados disponíveis no dispositivo móvel" é o diferencial que permite a definição da estrutura de uma T_{MAC} e do seu plano de execução.

O plano de execução de uma transação que obedece ao modelo proposto é escolhido dentre três estratégias possíveis, plano de execução isolada, remota ou compartilhada. No plano de execução isolada a transação ocorre no dispositivo móvel. A opção remota prevê o envio do processamento para a rede fixa e a execução compartilhada distribui o processamento entre os extremos.

Maximizar o sucesso do processamento das transações a despeito de todas as interferências citadas é o principal objetivo deste trabalho e para tal, estratégias foram identificadas e consistem em:

- a) formalizar um modelo transacional que permita empregar dinamicamente processamentos em locais diferentes;
- b) garantir que o modelo proposto atenda aos mecanismos que asseguram a confiabilidade e a recuperabilidade das transações;
- c) empregar os recursos de computação dos dispositivos móveis no processamento isolado ou distribuído das transações;
- d) viabilizar a sincronização de dados alterados em dispositivos móveis desconectados;
- e) prover tratamento adequado aos dados distribuídos e replicados pela rede fixa e pelos computadores móveis;

- f) adaptar as transações de acordo com a situação da rede de comunicação, da configuração do dispositivo e dos dados;
- g) apresentar a arquitetura de um *middleware* capaz de avaliar os recursos de computação e armazenamento disponíveis nos dispositivos móveis;
- h) agregar ao *middleware* a capacidade de avaliar os recursos da rede de comunicação entre computador móvel e rede fixa;
- i) fornecer ao *middleware* um instrumento de verificação dos dados disponíveis no equipamento portátil;
- j) empregar um mecanismo de aprendizagem que capacite o *middleware* a aprender com os resultados da execução das transações adaptadas;
- k) propor uma camada adicional aos sistemas gerenciadores de banco de dados populares capaz de gerenciar transações móveis adaptadas;
- l) viabilizar um canal de controle das conexões ativas para trocar dados e mensagens entre estações móveis e rede fixa.

1.2 Metodologia de desenvolvimento e organização do trabalho de pesquisa

O trabalho inicial desta pesquisa consistiu na investigação do ambiente de computação móvel, devido ao leque de questões em aberto que necessitam ser respondidas para viabilizar sistemas de computação móveis confiáveis. Nesse ínterim, o processamento de transações apresentou-se como um assunto de vital importância para o quesito confiabilidade.

Um vasto levantamento bibliográfico foi realizado para avaliar as abordagens já propostas, cuja síntese compõe o Capítulo 2. A partir da avaliação dos modelos transacionais encontrados na literatura foi possível entender os conceitos teóricos de transação, bem como as propriedades necessárias para que uma transação seja considerada uma unidade computacional confiável.

O capítulo também apresenta uma síntese dos modelos de transação para o ambiente de computação móvel. Ao final, uma seção analisa as características singulares que dificultam a aplicação de tais modelos a uma diversidade maior de domínios de negócio. O assunto da última seção transpõe o Capítulo 2 e introduz no capítulo seguinte um mecanismo que agrega às soluções apresentadas maior flexibilidade de atuação. Assim, delinea-se o emprego do conceito de adaptação às transações de banco de dados.

Do Capítulo 3 também constam os trabalhos correlatos ao tema proposto nesta pesquisa e uma breve discussão que explana as razões pelas quais esses trabalhos não encerram em si as necessidades de sistemas de computação que processam transações no cenário da computação móvel.

A junção do levantamento bibliográfico aos resultados da discussão do Capítulo 3 deu origem a abordagem apresentada nesta tese, que é descrita nos Capítulos 4 e 5. O Capítulo 4 apresenta a arquitetura de um *middleware* empregado na abordagem da adaptação colaborativa explorada neste trabalho. Sua utilização contribui efetivamente para a avaliação das características do ambiente móvel, bem como para o gerenciamento da execução das transações móveis adaptadas ao contexto, assunto do Capítulo 5.

O Capítulo 5 enfatiza o modelo de Transações Móveis Adaptadas ao Contexto, representadas pelo acrônimo T_{MAC} , que indica como uma transação pode ser elaborada para atender ao processamento isolado, remoto ou compartilhado.

Algumas seções fazem relação aos trabalhos apresentados nos Capítulos 2 e 3 para pontuar as contribuições da tese, além de tratarem objetivos secundários como a adequação das propriedades ACID ao modelo idealizado.

Para validar o modelo e a ferramenta, o Capítulo 6 apresenta um estudo de caso e as tecnologias usadas na implementação. Conclusões e trabalhos futuros constam do Capítulo 7, que é seguido das referências bibliográficas e dos apêndices.

2 TRANSAÇÕES EM AMBIENTE DE COMPUTAÇÃO MÓVEL

Para Ferreira e Finger (2000), uma transação é um modelo de comportamento que assegura a integridade dos dados de um sistema de banco de dados. Inicialmente, as transações eram enviadas a um computador, também denominado site, que centralizava o processamento e os dados. A Figura 2.1 é composta pelos sites que podem emitir as requisições. A presença do banco de dados no site 3, direcionam para ele o processamento das requisições.



Figura 2.1 – Ambiente com banco de dados centralizado

Fonte: Özsu e Valduriez (2001)

Nesta arquitetura, o modelo transacional utilizado era o de transações planas, conhecido por modelo clássico. Transações com estrutura plana seguem a definição dada por Melo et al. (1997) apresentada anteriormente, tendo como característica principal a unicidade da camada de controle. Durante a execução das transações planas, diferentes estados são assumidos, a Figura 2.2, extraída de Silberschatz (2007), sintetiza esses estados.

O estado inicial da transação é o “Ativa”, se a execução atingir a última operação o estado muda para “Em efetivação parcial”, após a conclusão com sucesso o estado passar a ser “Em efetivação”.

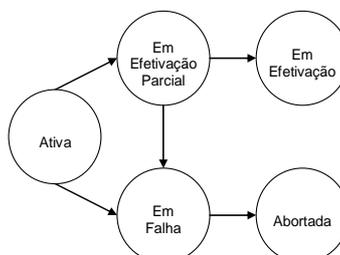


Figura 2.2 – Diagrama de estados de uma transação

Fonte: Silberschatz (2007)

Diante de anormalidades, o estado muda para “Em falha”, as ações são desfeitas e o banco de dados é restabelecido ao estado inicial e a condição muda para “Abortada”. As mudanças de estado são acompanhadas pelo Sistema Gerenciador de Banco de Dados (SGBD). Diante de resultados indesejados o componente de gerenciamento de recuperação desfaz qualquer operação. O SGBD ainda gerencia a interação entre transações concorrentes através do mecanismo de controle de concorrência, afirma Silberschatz (2007).

Avanços em redes de computadores e o refinamento dos requisitos impostos pelas aplicações exigiram das transações novas características. A mais expressiva dessas características diz respeito à capacidade de ser distribuída entre vários servidores, uma vez que espalhar os dados próximo ao seu ponto de uso mais freqüente reduz o tempo de resposta e os custos de comunicação.

A Figura 2.3 apresenta a ocorrência de vários bancos de dados distribuídos ao longo da rede, que podem inclusive permitir o armazenamento de várias réplicas dos dados em sites diferentes.



Figura 2.3 – Ambiente com banco de dados distribuídos

Fonte: Özsu e Valduriez (2001)

O modelo clássico de transações foi estendido e foram exploradas a divisão e distribuição das transações. As mudanças deram origem à segmentação da transação, explorando diferentes pontos de término para cada segmento. Neste modelo uma transação T é subdividida em várias subtransações que podem acessar diferentes sites. Cada site é responsável por uma parte da execução da transação, conforme ilustrado na Figura 2.4

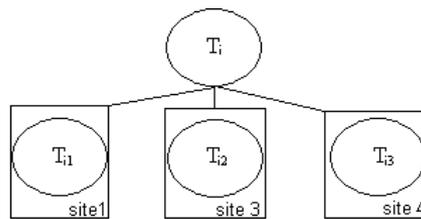


Figura 2.4 – Modelo de transações distribuídas

Diversos modelos existentes na literatura abordaram alguma forma de divisão das transações. Transações aninhadas, propostas por Moss (1985), tratam uma transação como uma árvore de subtransações e dividem-se em abertas e fechadas. As opções diferem quanto à visibilidade parcial da transação, mas nos dois casos as subtransações são executadas concorrentemente com as demais. Transações multiníveis, tratadas em Weikum e Schek (1992), são semelhantes às transações aninhadas, mas prevêm uma transação como uma árvore de mesma altura que trata operações comutativas nos nós folhas.

No modelo Sagas, proposto por Garcia-Molina e Salem (1987), uma transação é um conjunto relativamente independente de subtransações em dois níveis, associadas a transações compensatórias. Transações compensatórias desfazem os resultados obtidos por cada subtransação.

No Workflow transacional, proposto por Georgakopoulos (1995), implementam-se processos de negócio como transações ACID. A semântica é semelhante às transações aninhadas, mas impõe uma ordem parcial às subtransações que estão no primeiro nível ao invés de estruturá-las em hierarquia.

Independente do comportamento das transações e suas subtransações, a evolução do SGBD para o SGBDD (Sistema Gerenciador de Banco de Dados Distribuído), foi necessária para controlar o processamento distribuído e a réplica dos dados. Cada site passou a conter o seu próprio gerenciador de transação responsável por manter as propriedades ACID localmente, além de cooperar na execução global das transações.

Uma das táticas usadas para garantir as propriedades de atomicidade e durabilidade das transações globais é o protocolo de consolidação em duas fases (*Two-Phase Commit - 2PC*). O 2PC configura um coordenador que sincroniza a ação entre os sites que participaram do processamento, ou seja, uma transação só é consolidada caso todos os sites estejam aptos a fazê-lo.

Na Figura 2.5 o coordenador no estado “espera” aguarda as decisões dos participantes para cancelar ou efetivar a transação e manter a consistência global dos dados.

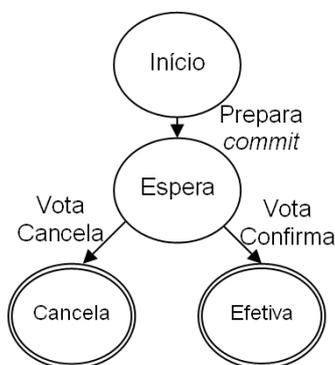


Figura 2.5 – Diagrama de transição de estado do coordenador no protocolo 2PC

Fonte: Özsu e Valduriez (2001)

Cada participante, na Figura 2.6, no estado “leitura” aguarda a decisão do coordenador para cancelar ou efetivar localmente.

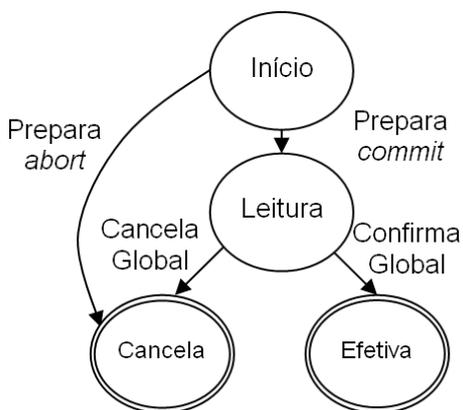


Figura 2.6 – Diagrama de transição de estado dos participantes no protocolo 2PC

Fonte: Özsu e Valduriez (2001)

Özsu e Valduriez (2001) reforçam que a solicitação de preparar do protocolo 2PC exige que os sites participantes do processamento concordem em consolidar antes dos efeitos da transação tornarem-se permanentes. A Figura 2.7 exibe as fases de preparar e agir geridas pelo coordenador da transação.

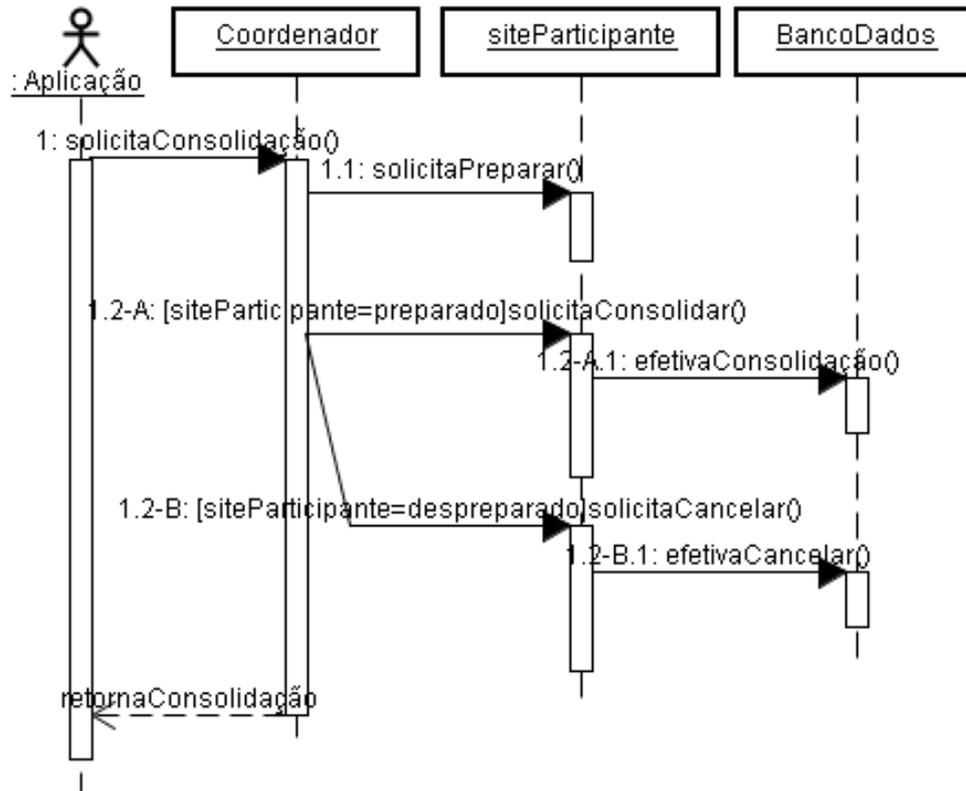


Figura 2.7 – Diagrama de seqüência do protocolo de consolidação em duas fases

Durante a execução do 2PC, a presença de falhas pode influenciar no comportamento desejado e a recuperabilidade dos dados precisa ser assegurada. Um dos sites pode não concordar com a consolidação da transação incorrendo no cancelamento de todas as suas partes.

Outra condição apresentada por Özsu e Valduriez (2001), diz respeito à falha em sites participantes do processamento ou até mesmo do canal de comunicação que os interliga, culminando na mesma decisão de cancelamento.

O isolamento e a consistência continuaram sob a atribuição do controle de concorrência, com a opção de abordagens diferentes como bloqueio ou etiqueta de tempo. Um exemplo é o algoritmo de bloqueio em duas fases (*Two-phase locking* - 2PL), retratado na Figura 2.8, que estabelece os bloqueios sobre os dados necessários à transação durante a fase de crescimento e em seguida, durante a fase de contração, libera tais bloqueios.

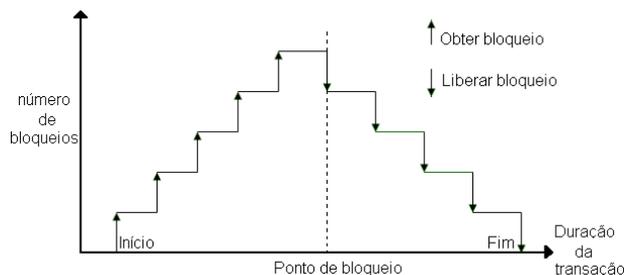


Figura 2.8 – Gráfico de bloqueio 2PL

Fonte: Özsu e Valduriez (2001)

A proposta do 2PL aumenta o grau de concorrência das transações na medida em que libera os bloqueios aos itens de dados, mas é difícil de ser concretizado, pois o gerenciador de bloqueios precisaria saber quando as transações não necessitariam mais de novos travamentos, além de conhecerem o momento certo da liberação de cada item.

Diante disto, uma versão mais viável do 2PL foi implementado, o 2PL rigoroso, visualizado na Figura 2.9. Nessa versão os bloqueios são mantidos durante o uso dos itens de dados e a liberação acontece somente ao término da transação.

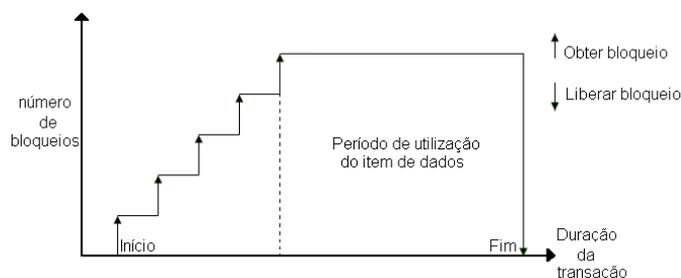


Figura 2.9 – Gráfico do bloqueio 2PL rigoroso

Fonte: Özsu e Valduriez (2001)

Nessa versão os bloqueios são liberados somente quando a transação termina. A Figura 2.10, através de um diagrama de seqüência, modela a solicitação do bloqueio dos itens X e Y na fase de crescimento e, em seguida, a liberação de ambos ao término da transação.

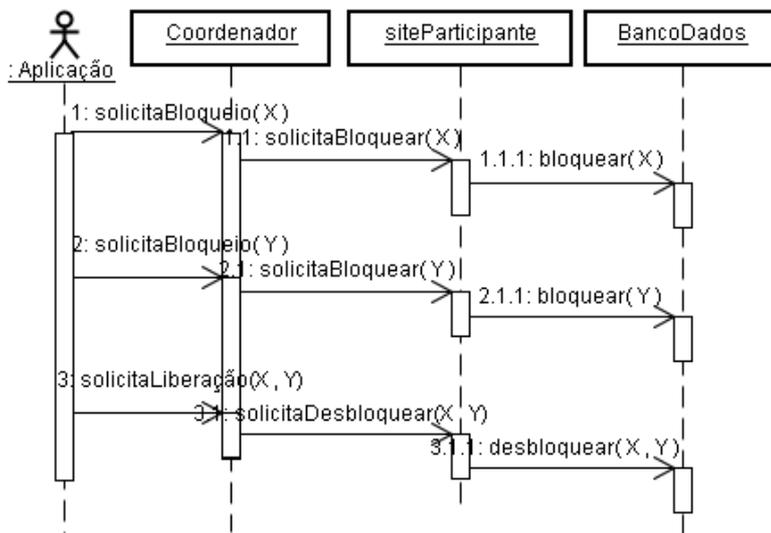


Figura 2.10 – Diagrama de seqüência do protocolo de bloqueio 2PL rigoroso

Outra questão tratada foi a replicação dos dados. O protocolo ROWA (*read-once write-all* - ler uma vez e gravar tudo), garante a liberação dos dados somente se os efeitos da transação forem propagados a todas as réplicas. Enquanto uma transação operar sobre um item de dado replicado, as demais cópias permanecem bloqueadas. Na Figura 2.11, o item de dado replicado é exemplificado por (X) e a operação de escrita de uma transação T em (X) é representada por {T,write(x)}.



Figura 2.11 – Representação de bloqueio de réplica de dados pelo ROWA

Fonte: Estendido de Özsu e Valduriez (2001)

Os protocolos e algoritmos citados não são as únicas propostas para garantir atomicidade, consistência, isolamento e durabilidade de transações distribuídas, mas representam as técnicas mais propagadas na literatura.

Uma nova evolução nas redes de comunicação, a conectividade sem fio, aliada à miniaturização de dispositivos portáteis ajudou a configurar o ambiente da computação móvel. As características intrínsecas a sistemas que operam nesse cenário, citadas na Seção 1.2, impuseram requisitos não funcionais aos sistemas para subsídio do seguinte:

- a) suporte a conexões intermitentes: a fim de permitir que o usuário mesmo em área de instabilidade da comunicação sem fio possa continuar operando;
- b) apoio a mobilidade do cliente: garantir o máximo de comunicação com a rede fixa mesmo diante de um possível movimento do usuário;
- c) atualização do deslocamento da estação móvel: viabilizar a localização da estação para recuperação de transações;
- d) autonomia em desconexões temporárias: possibilitar ao usuário o trabalho desconectado da rede fixa;
- e) reintegração de atualizações locais na rede fixa: garantir consistência global dos dados;

Em um contexto no qual a mobilidade e o trabalho isolado exercem indiscutível influência nos sistemas de computação, os efeitos impactados nas transações, nos moldes tratados anteriormente, inviabilizam sua execução como unidades computacionais confiáveis. As conseqüências que incidem em cada uma das propriedades ACID são exploradas a seguir.

2.1 Efeitos da mobilidade nas propriedades ACID

A atomicidade garante que operações parciais não se consolidem. Todas as operações são realizadas ou então descartadas. No caso da computação móvel, mesmo usando um algoritmo de consolidação distribuído como o 2PC, uma estação móvel em movimento pode perder a conexão no exato momento em que participa da sincronização de um processamento, fazendo com que todas as ações já realizadas, inclusive em outros sites, sejam desfeitas.

A capacidade de armazenamento dos dispositivos portáteis permite que todo o esquema do banco de dados ou parte dele esteja disponível para trabalho autônomo. Mas as alterações executadas isoladamente, ou seja, localmente, deixam os dados temporariamente inconsistentes se comparados aos mesmos dados na rede fixa, violando o princípio da consistência.

Os efeitos de uma transação executada em um dispositivo portátil podem ser visíveis a outras transações processadas no mesmo equipamento. O problema ocorre quando a transação inicial ainda não foi confirmada na rede fixa e o usuário continua a operar sobre os mesmos itens de dados, isto pode ocasionar cancelamento em cascata. Nessa situação há transgressão do isolamento, que trata a visibilidade das ações de uma transação sobre outra.

A durabilidade das transações que ocorrem nos dispositivos móveis está sujeita a sincronização dos dados locais com os equivalentes remotos. Caso a sincronização não seja possível devido à inconsistência dos dados, a transação mesmo consolidada acaba por ser desfeita.

Diante da pluralidade de exigências impostas aos sistemas de computação e da influência da mobilidade sobre as propriedades ACID uma reformulação do conceito de transação fez-se necessário, esse é o assunto das seções seguintes.

2.2 Modelos para transações móveis

Ao contrário do modelo clássico, não há um consenso geral sobre o conceito de transações em ambiente de computação móvel. Variações significativas são encontradas nas propostas disponíveis em literatura especializada. O comportamento dos modelos já existentes na literatura são retratados a seguir.

2.2.1 Modelo IOT

O modelo IOT (*Isolation-Only Transaction*), proposto por Lu e Satyanarayanan (1994) é uma extensão do projeto Coda. O Coda, elaborado por Satyanarayanan et al. (1990), viabilizava operações desconectadas como forma de permitir atividades continuadas diante de falhas de comunicação em ambientes distribuídos. Em uma etapa seguinte, passou a prover acesso aos dados a partir de máquinas móveis, sendo considerado um dos pioneiros nessa área.

O Coda fazia uso de replicação de servidores e de cópias de dados em cache para permitir ao usuário continuar suas operações mesmo que não conseguisse conexão com nenhum outro computador da rede, mas após a reconexão só visava a resolução de conflitos do tipo escrita/escrita. O modelo IOT passou a considerar os conflitos do tipo leitura/escrita, garantindo maior consistência aos dados manipulados pelo sistema.

2.2.2 Modelo Pro-motion

O Pro-motion (*Pro-Active Management of Mobile Transactions*), proposto por Walborn e Chrysanthis (1997,1999), é um sistema de processamento de transações que suporta o modo desconectado em um ambiente cliente/servidor móvel. Dados armazenados nas caches dos dispositivos móveis são encapsulados e chamados “compacto” (*compact*). Cada cliente móvel é dotado de um agente de compacto (*Compact Agent*) responsável por gerenciar todas as ações submetidas ao mesmo.

A elaboração dos compactos no servidor de banco de dados é função do gerente do compacto (*Compact Manager*). A estação de base hospeda o gerente de mobilidade (*Mobility Manager*), que provê a comunicação entre o agente e o gerente do compacto. As transações executadas localmente são sincronizadas durante a reconexão pelo agente na estação móvel e pelo gerente do compacto na rede fixa.

2.2.3 Modelo *Clustering*

O modelo *Clustering*, tratado por Pitoura e Bhargava (1995) e detalhado por Pitoura e Bhargava (1999), considera um sistema totalmente distribuído que aceita consistência estrita nos dados dos componentes da rede fixa e consistência relaxada nos dados disponíveis nos equipamentos móveis.

O banco de dados é dividido em grupos (*clusters*), sendo cada grupo configurado dinamicamente num dado momento em que os dispositivos estão fortemente conectados. A reconfiguração dinâmica permite que, se um dispositivo desconectar-se do grupo, um novo grupo seja formado a partir do próprio equipamento.

Os dados disponíveis na rede fixa são considerados como versão forte (*strict version*) e a versão fraca (*weak version*) são os dados armazenados nas caches dos dispositivos móveis. As transações também são divididas em fortes e fracas, sendo que as transações fortes acessam a versão forte dos dados e as transações fracas acessam a versão fraca dos dados.

Quando uma unidade móvel requisita fragmento de dados para uso em cache, nenhuma outra transação pode operar naquele fragmento. Durante a reconexão, um processo de reconciliação viabiliza a consistência global entre dados na versão forte e na versão fraca.

2.2.4 Modelo *Two-tier replication*

Gray et al. (1996) no modelo *Two-tier replication* trata transação e replicação em duas versões. Uma versão dos dados está disponível em unidades móveis ocasionalmente conectadas (cópias replicadas). A outra é uma versão mestre dos dados disponível na rede fixa. As transações se dividem em transação “tentativa” e transação “base”.

As transações base são executadas na versão mestre dos dados, enquanto as transações “tentativas” são executadas em versões locais e efetuam alterações nos dados no modo desconectado. O modelo propõe que quando a conexão for restabelecida, as transações “tentativas” serão reexecutadas como transações “base”, como forma de tornar as atualizações locais persistentes também no banco de dados da rede fixa.

2.2.5 Modelo *Reporting e Co-transactions*

Chrysanthis (1993) afirma que a estrutura atômica de uma transação é limitada para o caso móvel, sendo assim, sugere a divisão das transações em subtransações que podem operar tanto nos dispositivos móveis quanto nos fixos. A sugestão é um modelo transacional aninhado aberto que suporta transações atômicas, não-compensáveis, *reporting* e *co-transactions*.

Transações atômicas equivalem ao modelo clássico, enquanto transações não-compensáveis repassam suas operações às transações ancestrais para que estas decidam pela sua efetivação ou cancelamento.

Uma transação *reporting* pode delegar resultados parciais para outra transação e continuar em execução. *Co-transactions* também podem delegar seus resultados parciais, mas são suspensas no momento da delegação e somente continuam a operar após obterem respostas das transações delegadas.

2.2.6 Modelo Semântico

Para Walborn e Chrysanthis (1995), a unidade móvel precisa ser autônoma e continuar seu processamento no modo desconectado. Devido à limitação da capacidade de armazenamento, o modelo baseado em semântica propõe a utilização de objetos fragmentados. Cada fragmento pode ser armazenado na cache dos dispositivos móveis e pode ser manipulado de forma assíncrona.

Dois critérios são importantes no processo de fragmentação: critério de seleção e condições de consistência. O primeiro informa os dados a serem armazenados em cache, bem como o tamanho dos fragmentos. O segundo especifica restrições para preservar os dados.

Fragmentos enviados aos dispositivos móveis são considerados cópias principais exclusivas e as transações podem ser executadas sobre tais fragmentos sem preocupação com consistência global. Durante a reconexão um processo de reconciliação garante o retorno dos dados à rede fixa.

2.2.7 Modelo *Prewrite*

Neste modelo, Madria e Bhargava (2001) introduziram operações de pré-gravação (*prewrite*) e pré-confirmação (*precommit*). A execução da transação é um trabalho de cooperação entre unidade móvel, estação de base e gerenciador de dados da rede fixa.

O gerente de transações do dispositivo móvel solicita bloqueio dos dados ao gerenciador de dados da rede fixa, função intermediada pela estação de base. Uma pré-confirmação local é executada ao término da transação móvel e pré-gravações resultantes da transação móvel são enviadas para a estação de base refletindo o valor futuro dos dados. O gerente de dados efetiva as pré-gravações encaminhadas pela estação de base e confirma permanentemente a transação móvel.

2.2.8 Modelo *Kangaroo*

Esse modelo, tratado inicialmente em Dunham et al. (1997) e, em seguida, por Dunham e Kumar (1999), explora o comportamento dos dados e do movimento na estratégia de execução da transação. Quando um dispositivo móvel faz uma requisição, a estação de base gera uma transação móvel de escopo global chamada transação canguru (*Kangaroo*).

A estação de base através do Agente de Acesso a Dados (AAD) coordena a execução da transação realizada totalmente pelo sistema de banco de dados na rede fixa. Quando a unidade móvel muda de célula a coordenação da transação também muda. O modelo usa o conceito de transações aninhadas-abertas e transações divididas (*split*). Assim, quando há o movimento, a transação é subdividida em transações filhotes (*joey transaction*) e, a cada estação de base que a unidade móvel visitar, uma transação filhote será criada.

As transações filhotes são coordenadas pelo AAD da estação de base onde foram criadas e podem ser efetivadas ou canceladas independentemente das demais. Caso a transação canguru decida abortar, transações compensatórias desfazem os resultados das transações filhotes efetivadas.

2.2.9 Modelo *Pre-serialization*

O modelo *Pre-serialization*, proposto por Dirckze e Gruenwald (2000), admite um ambiente de múltiplos bancos de dados, onde cada unidade móvel executa transações compensáveis na rede fixa.

Uma transação móvel é formada por subtransações que operam sobre um dos bancos de dados do conjunto. Neste modelo, as transações possuem ordem de serialização antes da decisão de efetivação. As subtransações também são compensáveis, pois são efetivadas mesmo que a transação global ainda não tenha decidido fazer o mesmo, liberando os recursos bloqueados.

As estações de base são dotadas de coordenadores de transações globais que gerenciam a migração das unidades móveis, a desconexão e a submissão de subtransações à rede fixa. Durante a reconexão, todas as subtransações enviadas à rede fixa têm seus resultados retornados ao dispositivo que a solicitou. Para evitar recursos indefinidamente alocados, a requisição de qualquer item de dado aborta a transação órfã.

2.2.10 Modelo MDSTPM

Diante de uma arquitetura de gerenciamento de transação em múltiplos bancos de dados (MDSTPM - *Multidatabase Transaction Processing Manager*), Yeo e Zaslavsky (1994) propõem um serviço de “Mensagem e Suporte para Enfileiramento” (*Message and Queuing Facility* – MQF) que gerencia a troca de informações entre unidades móveis e o sistema de múltiplos bancos de dados.

As transações são solicitadas pelas estações móveis que podem se desconectar e trabalhar em outras tarefas. O SGBD disponível nos servidores da rede fixa irá executá-las e quando ocorrer a reconexão enviará através do MQF os resultados obtidos.

2.2.11 Modelo Moflex

Ku e Kim (2000), através do modelo Moflex, visualizam a execução de transações mesmo diante do movimento dos dispositivos móveis. A transação nessa proposta é composta por um conjunto de subtransações interdependentes. Transações moflex contêm regras para uma possível mudança de célula que permite à transação subdividir-se em subtransações residentes nas células anteriores e posteriores ao movimento.

Três camadas são identificadas nessa proposta: a unidade móvel, a estação e base e o sistema de banco de dados. Na camada da unidade móvel as transações são submetidas ao gerente de transações móveis disponíveis na

camada de estação de base. O gerente de transações móveis encaminha a transação para um coordenador global na camada do sistema de banco de dados, que executa a transação iniciada pelo equipamento portátil. Regras de junção permitem às transações Moflex validarem a execução concatenada da transação a uma execução serial equivalente.

2.2.12 Modelo HiCoMo

O HiCoMo (*High Commit Mobile*) é um modelo de transação móvel proposto por Lee e Helal (2002) para aplicações de tomada de decisão. O objetivo é permitir alterações durante as desconexões nos dados armazenados nas unidades móveis.

Dois tipos de transações são possíveis, a saber: as transações base e as transações HiCoMo. Outro item sugerido inclui a criação de tabelas base e tabelas agregadas que representam um resumo ou uma estatística do que é encontrado nos dispositivos portáteis.

As transações HiCoMo são executadas e armazenadas em tabelas agregadas durante a desconexão das estações móveis, enquanto as transações base refletem as modificações feitas pelas transações HiCoMo nas tabelas base.

2.2.13 Modelo TM

Para Côrtes (2004), uma transação móvel é uma transação distribuída, na qual alguma parte da computação é executada no equipamento móvel e a outra parte em um equipamento fixo. Assim, o modelo TM (transação móvel) opera sobre um sistema de banco de dados totalmente distribuído disponível na rede fixa e nos equipamentos móveis. A Figura 2.12 apresenta a estrutura do modelo que é subdividido em quatro níveis. Os níveis estão dispostos nas camadas de aplicação móvel (nível 1), no SGBD do equipamento móvel (nível 2), na estação de base (nível 3) e na rede fixa (nível 4).

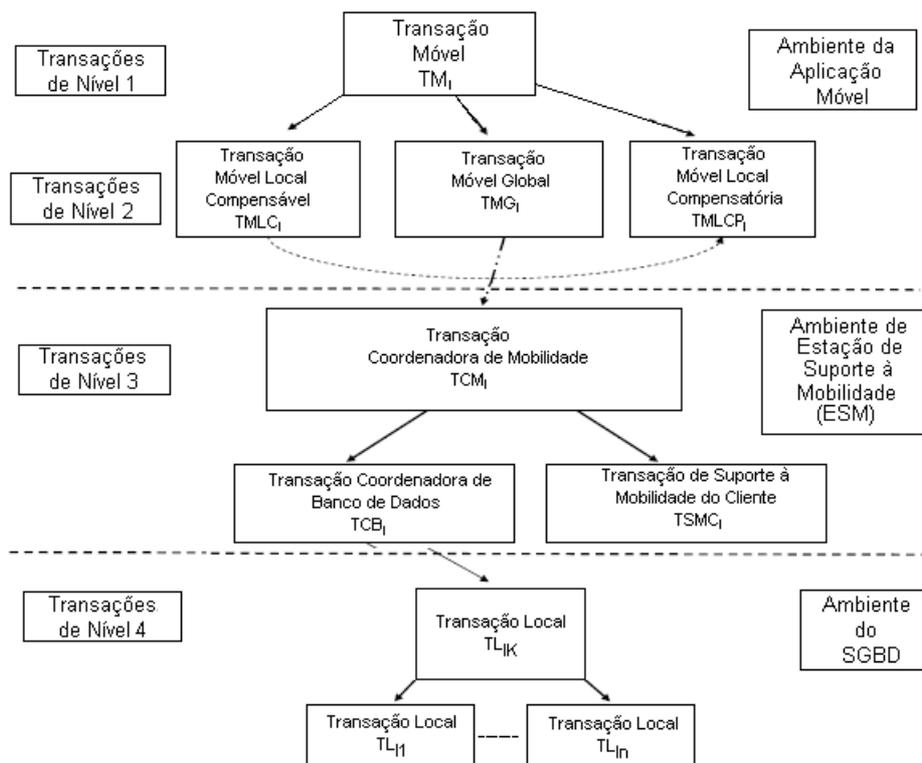


Figura 2.12 – Organização de uma TM

Fonte: Côrtes (2004)

A transação de nível 1, ou seja, a TM (Transação Móvel) faz interface com as aplicações instaladas na estação móvel. Nesse ambiente as transações móveis são geradas, iniciadas e finalizadas. Sua inicialização dispara a ocorrência de duas transações-componentes de nível 2 chamadas Transações Móveis Locais Compensáveis (TMLCs) e as Transações Móveis Globais (TMGs).

Uma TMLC ocorre no próprio equipamento móvel diante de um SGBD local. Nesse sentido, Transações Móveis Locais Compensatórias (TMLCPs) são transações que desfazem os efeitos de uma TMLC, caso esta venha a ser cancelada após sua efetivação no dispositivo móvel. A execução de uma TMLC é serial, pois suas operações são executadas uma após a outra, sem nenhum tipo de concorrência. Os dados necessários a essas transações são bloqueados e liberados após o seu término.

Transações Móveis Globais (TMGs) são enviadas à rede fixa para submissão ao SGBD. Todo o controle é realizado no dispositivo móvel, que mesmo desligado ou sem conexão mantém a espera pelos resultados oriundos das transações-componentes executadas na rede fixa.

No nível 3, as Transações Coordenadoras de Mobilidade (TCM) representam as TMGs na rede fixa. Entre suas funções estão o acompanhamento da mobilidade das estações móveis e o controle ininterrupto das transações-componentes que compõem as TMGs. Sua efetivação só ocorre após a efetivação de suas subtransações.

Uma Transação Coordenadora de Banco de Dados (TCBD) também está no nível 3 e representa as TCM e as TMGs. Sua função é coordenar a execução de uma TM no banco de dados da rede fixa. As Transações de Suporte à Mobilidade do Cliente (TSMC) estão no mesmo nível e mantêm a localização do cliente móvel no banco de dados disponível nas estações de base.

No último nível estão as Transações Locais (TL) a serem executadas e confirmadas no banco de dados da rede fixa. Neste ambiente estão presentes todas as características necessárias ao ambiente distribuído como, por exemplo, o protocolo de bloqueio em duas fases (2PL) ou qualquer outro mecanismo utilizado pelo SGBD para prover confiabilidade e recuperabilidade.

Côrtes (2004) também propôs uma arquitetura com um conjunto de funções particulares ao ambiente de computação móvel. A Figura 2.13 representa essa arquitetura enfocando três camadas. Na primeira camada estão as aplicações móveis e o banco de dados local. Na segunda estão as estações de base que formam a rede fixa e na terceira o banco de dados.

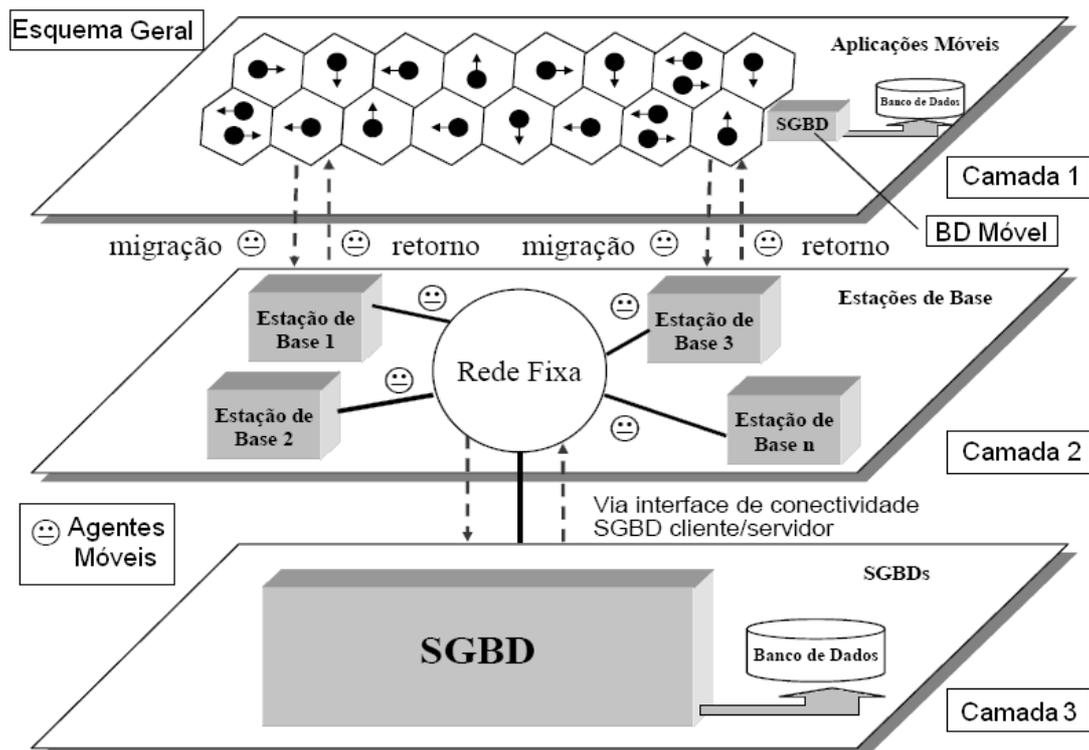


Figura 2.13 - Esquema geral da arquitetura proposta no modelo TM

Fonte: Côrtes (2004)

A camada da aplicação móvel é executada para que seja possível a criação e execução das Transações Móveis localmente, a migração das TMs para a rede fixa, o retorno dos resultados e a confirmação ou cancelamento das TMs.

A Estação de Base é responsável, principalmente, por garantir a comunicação de dispositivos móveis e rede fixa, além de viabilizar a migração das TMs, tratar os resultados do processamento das TMs, acompanhar o movimento do equipamento móvel e controlar o canal de comunicação.

Na terceira camada, onde está o sistema de banco de dados, as funções garantem a persistência e consistência dos dados armazenados, processamento das transações e manutenção da comunicação com os agentes das estações de base.

2.3 Classificação dos modelos de transações móveis

Devido à variedade de modelos e propósitos, alguns esforços para classificação foram empregados por Serrano-Alvarado et al. (2004), Côrtes (2004), Rocha e Toledo (2005), na tentativa de comparar os resultados de cada proposta.

Os autores usam critérios diferentes para agrupar os modelos, Serrano-Alvarado et al. (2004) consideram o local de execução da transação. A Tabela 2.1 apresenta uma classificação com a divisão dos modelos que executam transações na rede fixa, na estação móvel ou com processamento distribuído entre ambas.

Tabela 2.1 – Classificação dos modelos de transação de acordo com o local de processamento

Local de processamento	Modelo
Rede fixa - após a requisição ter partido da estação móvel	<i>Kangaroo</i>
	<i>Pre-serialization</i>
	Moflex
	MDSTPM
Estação móvel com reintegração dos dados na rede fixa	HiCoMo
	<i>Pro-motion</i>
	<i>Clustering</i>
	<i>Two-tier replication</i>
	IOT
	<i>Prewrite</i>
Processamento distribuído entre estação móvel e rede fixa	Semântico
	<i>Reporting</i>
	<i>Clustering</i>
	<i>Two-tier replication</i>

Rocha e Toledo (2005) investigam os modelos considerando desconexões, largura de banda, limitações de energia, escassez de espaço em disco e mobilidade. A Tabela 2.2 apresenta um agrupamento dos modelos considerando as estratégias usadas diante dos critérios de classificação citados

Tabela 2.2 - Classificação dos modelos de transação de acordo com as características do ambiente de computação móvel

Característica	Técnica	Modelo
Desconexão	Caching de dados	HiCoMo <i>Prewrite</i> Pro-motion IOT <i>Reporting</i> <i>Clustering</i> <i>Two-tier replication</i>
Largura de banda	Transferência total de dados	IOT
	Transferência parcial de dados	<i>Pro-motion</i> Semântico
Energia	Divisão da computação	<i>Reporting</i>
	Modo de operação desconectado	Semântico
	Desligamento da estação móvel em períodos ociosos	<i>Kangaroo</i>
Espaço em disco	Transferência de pequenos fragmentos	Semântico Pro-motion
	Computação distribuída	<i>Reporting</i>
Mobilidade	Subdivisão da transação	<i>Kangaroo</i> MDSTPM

Os modelos apresentados flexibilizam as propriedades ACID como forma de viabilizar as transações móveis. Côtres (2004), na Tabela 2.3, apresenta a estratégia usada pelos modelos diante de cada propriedade.

Tabela 2.3 - Classificação dos modelos de transação de acordo com as estratégias adotadas quanto às propriedades ACID

Propriedade ACID	Estratégia de flexibilização	Modelo
Atomicidade	Hierarquia de unidades atômicas	Pro-motion <i>Reporting</i> <i>Kangaroo</i> Moflex HiCoMo
Consistência	Definição de formas de acesso aos dados	Pro-motion <i>Semântico</i>
	Regras de consistência	Moflex HiCoMo <i>Two-tier replication</i>
Isolamento	Liberação de resultados parciais	IOT <i>Reporting</i> HiCoMo <i>Pre-serialization</i> Semântico
	Níveis graduais de compartilhamento	Pro-Motion
Durabilidade	Mecanismos de garantia de efetivação da transação	HiCoMo

2.4 Considerações

A análise das abordagens dos modelos de processamento de transações existentes destina-se à contextualização desta pesquisa e dirige-se ao tratamento das transações móveis, sobretudo no tocante aos requisitos que as tornam unidades confiáveis de computação. Diante disto, segue uma classificação que apresenta estrutura e mecanismos de confiabilidade.

Tabela 2.4 - Classificação dos modelos diante da estrutura e da confiabilidade

Modelo	Estrutura	Mecanismos de confiabilidade
IOT	Plana	revalidação dos dados em cache; reintegração parcial; mecanismo IOT; cache translúcido.
Pro-motion	Aninhada aberta	responsabilidade compartilhada; regras explícitas de consistência; níveis de isolamento diversificados; sincronização entre réplicas.
HiCoMo	Particionada	margem de erro; somente operações comutativas; regras de validação de execução; reexecução com avaliação de conflitos.
Semântico	Plana	cache semântico; regras explícitas de; reintegração simplificada; bloqueio estrito na rede fixa.
<i>Prewrite</i>	Aninhada	mantém versão abstrata dos dados; pretensão de operação; gerenciamento compartilhado; bloqueio dos dados.

Continua

Tabela 2.4 - Conclusão

<i>Clustering</i>	Plana	controle dos níveis de inconsistência; consistência semântica; reintegração com verificação de conflito.
<i>Two-tier replication</i>	Plana	versão duplicada dos itens de dados; critérios para reexecução de transações.
<i>Reporting e Co-transations</i>	Aninhada aberta	manutenção do estado de subtransações; transações compensatórias.
<i>Kangaroo</i>	Particionada	atribuições do SGBD; gerenciamento compartilhado; cancelamento de subtransações.
<i>Pre-serialization</i>	Aninhada	atribuições do SGBD; gerenciamento compartilhado; serialização antecipada.
Moflex	Particionada	atribuições do SGBD; regras de particionamento e junção; regras de validação de execução.
MDSTPM	Plana	atribuições do SGBD; <i>message and queuing facility</i> .

Os mecanismos de confiabilidade propostos nos modelos estão relacionados ao conceito de transação móvel admitido pelos autores. Os proponentes dos modelos *Kangaroo*, *Pre-serealization*, Moflex, MDSTPM e *Reporting e Co-transations* admitem como transação móvel uma requisição que tenha partido de um dispositivo móvel com execução nos servidores da rede fixa.

O modelo *Reporting e Co-transations* também aceita que a computação possa ser realizada de forma compartilhada entre o dispositivo móvel e a rede fixa. Nessa visão, o SGBD disponível na rede fixa é a entidade responsável por manter a confiabilidade e a recuperabilidade dos dados manipulados pela transação móvel.

A execução de transações no dispositivo portátil durante desconexões é considerada pelos autores dos modelos IOT, Pro-motion, HiCoMo, Semântico, *Prewrite*, *Clustering* e *Two-tier replication*. Este cenário configura transações móveis apoiadas por reexecução das transações nos servidores ou métodos de conciliação de dados disponíveis em computadores móveis e fixos.

Os modelos TM, *Clustering* e *Two-tier replication* também aceitam o processamento duplicado, pois permitem a execução das transações na estação móvel e na rede fixa, quando há comunicação entre ambas.

Um aspecto a ser ressaltado é que independente do conceito de transação móvel ou do lugar de sua execução, os autores usam estruturas de transação divergentes para propostas equivalentes.

Outra questão perceptível é a inexistência de um modelo de transação que satisfaça todas as variáveis que têm influência em um sistema de computação móvel. Alguns modelos consideram o problema das desconexões decorrentes da mobilidade e da variabilidade da comunicação sem fio, outros supõem que a conexão sempre estará disponível, o que é difícil de assegurar.

Alguns modelos são alheios às restrições do espaço de armazenamento dos dispositivos móveis e não levam em conta o tamanho dos dados disponibilizados na cache dos mesmos. Em outros casos, a consistência dos dados é obtida à custa da indisponibilização dos mesmos.

Sistemas de computação disponíveis em dispositivos portáteis, conectados via comunicação sem fio, que exigem disponibilidade de acesso, trabalho autônomo e principalmente confiabilidade nas transações, não encontram apoio adequado nos modelos apresentados. Em vista disso, é interessante que diferentes estratégias de execução de uma transação possam ser operacionalizadas, aumentando o êxito de uma transação em tal cenário.

A execução diversificada das transações é possível através da adaptação. Adaptar é prover flexibilidade para agir diante de situações alternativas. No ambiente de computação móvel, situações alternativas configuram-se a partir de falhas de comunicação ou restrição de recursos.

Alguns trabalhos nessa linha são encontrados em Serrano-Alvarado et al. (2004) e Rocha e Toledo (2005). O próximo capítulo apresenta o estado da arte na utilização do conceito de adaptação para o processamento de transações em ambiente de computação móvel.

3 ADAPTAÇÃO DE TRANSAÇÕES MÓVEIS

Sistemas de Computação Móvel estão sujeitos a variações nas condições da comunicação sem fio, nos recursos de computação e na mobilidade dos computadores que participam da computação. Essa dinâmica sugere que características comuns à adaptação estejam disponíveis aos SCMs, capacitando-os a agir em resposta às mudanças.

3.1 Comportamento adaptativo

Comportamento adaptativo, para Satyanarayanan (1996), consiste na avaliação das condições momentâneas do ambiente e na escolha da melhor solução, dentro de um conjunto de soluções disponíveis, que atenda a aplicação dentro da configuração que se apresenta.

As estratégias de adaptação podem ocorrer sob dois extremos, conforme ilustrado na Figura 3.1. Em um deles, a adaptação é de inteira responsabilidade das aplicações. Essa abordagem, denominada *Laissez-faire*, evita a necessidade de um sistema de apoio. Por outro lado, falta uma entidade capaz de resolver demandas por recursos compartilhados. Além disso, as aplicações tornam-se mais complexas.

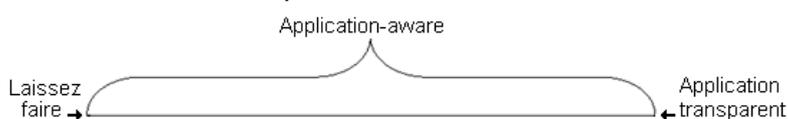


Figura 3.1 – Eixos de adaptação

Fonte: Satyanarayanan (1996)

No outro extremo está a *Application-transparent*. Neste tipo de abordagem a responsabilidade pela adaptação é toda do sistema, suavizando os encargos durante a etapa de desenvolvimento das aplicações. Mas a desvantagem é que uma adaptação padronizada pode não atender a requisitos particulares ou pode oferecer soluções que atendam a requisitos desnecessários às aplicações.

Entre os dois extremos existe outra possibilidade a *Application-aware*. Por sustentar uma parceria colaborativa entre aplicações e sistema, esta abordagem sugere que às aplicações cabe a decisão de como se adaptar e ao sistema cabe funções de controle e monitoramento dinâmico do nível dos recursos, notificação de mudanças e alocação de recursos.

Por promover a flexibilidade, a adaptação personalizada e o compartilhamento de funções, a *Application-aware* é uma estratégia atrativa. Com disponibilidade e conhecimento do nível dos recursos, as aplicações possuem maior chance de alcançarem seus propósitos, além de minimizarem possíveis interferências.

A adaptação tem sido usada em várias áreas da computação. Por exemplo, a abordagem *Application-transparent* foi aplicada ao projeto *Coda File Systems*, citado na Seção 2.1. Esse projeto, disponível em Satyanarayanan (1990), introduziu operações desconectadas e réplicas de servidores para garantir o processamento diante de falhas de partição da rede.

A estratégia *Application-aware* foi usada no projeto *Odyssey* que provê uma plataforma para o desenvolvimento de sistemas de computação móvel. O projeto oferece um mecanismo de monitoramento e notificação dos recursos, a fim de que cada aplicação reaja adequadamente às variações capturadas.

O Prayer, tratado em Bharghavan e Gupta (1997), é outro exemplo do uso da abordagem *Application-aware*. Através de uma estrutura em três camadas, o sistema promove uma adaptação colaborativa. A camada inferior monitora recursos, a intermediária os gerencia e na última estão as aplicações e suas políticas de adaptação.

Diante da inflexibilidade e incompletude dos modelos para transações móveis descritos no capítulo anterior, a busca por um formato de computação mais maleável e de maior abrangência encontrou apoio na adaptação colaborativa.

3.2 Modelos transacionais adaptativos

Aliar o conceito de adaptação às transações em ambiente de computação móvel alavancou possibilidades com maiores perspectivas de êxito, visto que a execução das mesmas é conciliada às configurações momentâneas do ambiente. As próximas subseções detalham essas possibilidades.

3.2.1 O modelo AMT

O modelo de Transações Móveis Adaptáveis (AMT – *Adaptable Mobile Transactions*) faz parte da proposta de Serrano-Alvarado (2002). Sua composição engloba várias estratégias de execução diferentes para cada componente transacional.

Um componente transacional (T) pode ser do tipo plano, distribuído ou aninhado, alocado entre unidade móvel ou rede fixa. Um coordenador em cada lugar que T ocorre garante as propriedades ACID. Transações compensatórias (CT) estão associadas a um T e desfazem os seus efeitos.

As estratégias de execução (ES - *Execution Strategy*) da transação são compostas por execuções alternativas (EA - *Execution Alternatives*). Uma execução alternativa depende da avaliação do descritor do ambiente (ED - *Environment Descriptor*) e do plano de execução (EP - *Execution Plan*).

O descritor do ambiente (ED) apresenta várias definições estáticas e previamente identificadas do ambiente e para cada definição um plano de execução (EP). Por sua vez, os planos de execução incluem a plataforma de processamento de cada componente da transação, bem como os relacionamentos de dependência entre eles. O relacionamento de dependência diz respeito à seqüência de execução de cada instrução da transação.

A definição formal de uma transação nesta proposta é $AMT = (T, CT, ES)$, onde:

- a) AMT significa transação móvel adaptável;
- b) T representa um componente transacional;
- c) CT são transações compensatórias associadas a T;
- d) ES são as estratégias de execução de uma AMT.

A Figura 3.2 ilustra a composição dos elementos de uma transação AMT. Duas execuções alternativas compõem a estratégia de execução da transação AMT. Cada uma atende a uma descrição do ambiente e propõe um plano de execução. A transação é dividida em três componentes transacionais (T1, T2 e T3) e um componente compensatório (CT1), sendo que CT1 desfaz T1.

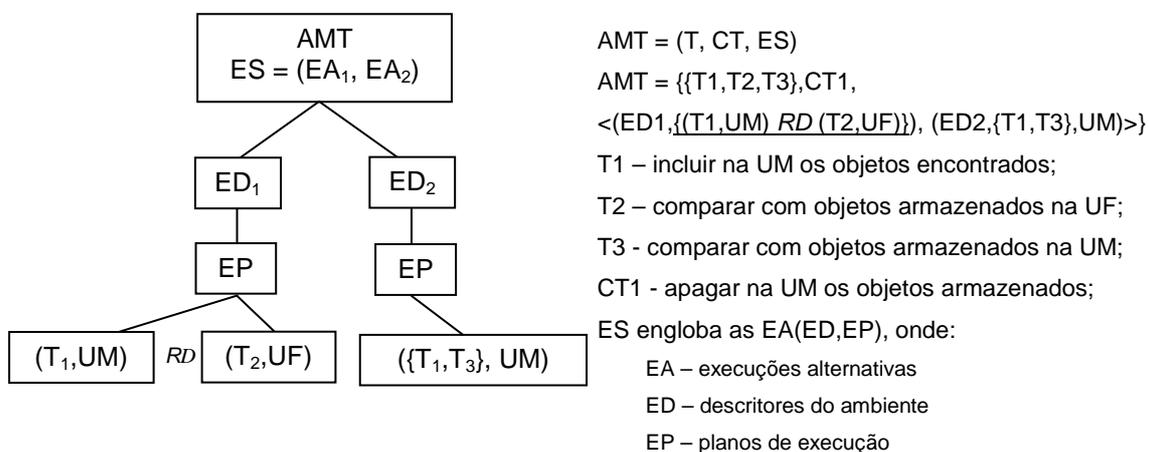


Figura 3.2 – Componentes de uma AMT

Fonte: Serrano-Alvarado (2002)

O descritor do ambiente (ED) é composto pelos conteúdos das variáveis do cenário móvel. A Tabela 3.1 apresenta todas as variáveis, seus possíveis conteúdos e as unidades de medida.

Tabela 3.1 – Dimensões do Serviço de Transação Móvel

	Dimensão	Valores	Unidade
Rede Fixa	Estado da conexão	Conectado Desconectado	
	Taxa de transmissão	Alta / Média / Baixa	Kbytes/s
	Custo de comunicação	Gratuito / Barato / Caro	Euros/t
Unidade Móvel	Disponibilidade de bateria	Total / Metade / Pouca	hh:mm:ss
	Disponibilidade de cache	Total / Metade / Pouca	Kbytes
	Disponibilidade de memória persistente	Total / Metade / Pouca	Kbytes
	Capacidade de processamento	Alta / Média / Baixa	Mhz/s
	Tempo da conexão	T	hh:mm:ss

Uma versão mais completa da Figura 3.2 é apresentada na Figura 3.3, onde pode ser visualizada a configuração do ambiente, ou seja, os elementos que formam os descritores do ambiente.

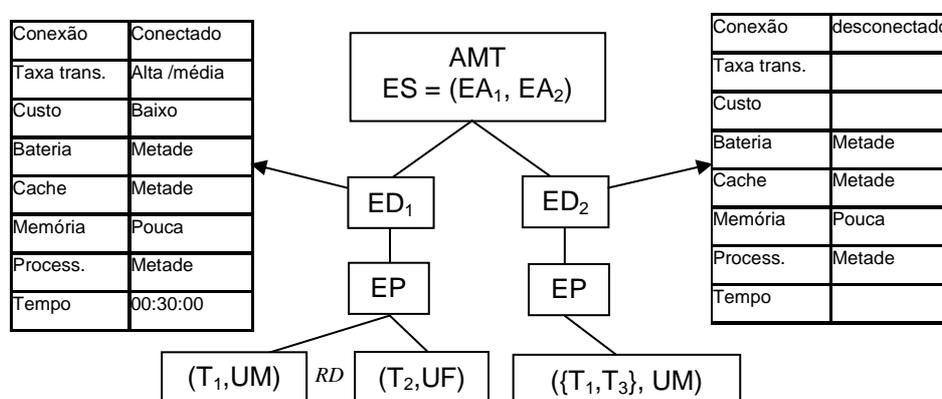


Figura 3.3 – Componentes de uma AMT e os descritores do ambiente

Fonte: Serrano-Alvarado (2002)

O relacionamento de dependência (RD) é definido nos planos de execução (EP). Na Figura 3.3 é possível verificar que o componente transacional T1 antecede o componente T2, mesmo ocorrendo em locais diferentes.

Em cada unidade onde T é executado, existe um coordenador. O coordenador no dispositivo móvel é chamado MTS Cliente (MTS - *Mobile Transaction Service*), na estação de base é denominado MTS Agente e na rede fixa MTS Servidor. A Figura 3.4 apresenta uma visão geral dos coordenadores do *Mobile Transaction Service* ou Serviço de Transação Móvel.



Figura 3.4 - Arquitetura Global do MTS

Fonte: Serrano-Alvarado (2002)

A arquitetura do MTS é composta por camadas cliente/agente/servidor, a camada cliente atua na unidade móvel, a porção agente opera na estação de base e a camada servidor atua no banco de dados da rede fixa. O serviço parte do princípio que existem funcionalidades de SGBD na rede fixa e nos dispositivos móveis.

A interface de comunicação entre o MTS e o SGBD ocorre através do gerenciador da transação (TM – *Transaction Manager*). O serviço ainda é composto por controlador de eventos, que notifica alterações no ambiente móvel e por um mecanismo de troca de mensagens

3.2.2 Mecanismos de adaptação e a PGTA

A Plataforma de Gerenciamento de Transações Adaptáveis (PGTA) foi proposta por Rocha e Toledo (2007) para apoiar os mecanismos de adaptação de transações identificados na pesquisa. Esses mecanismos incluem mudança nos níveis de isolamento, modo de operação e mobilidade.

A concorrência de transações pode resultar em efeitos indesejáveis. O isolamento propõe a ocorrência intercalada de transações simultâneas como forma de evitar tais efeitos. A PGTA admite os diferentes graus de isolamento integrados ao padrão ANSI SQL-92 como forma de viabilizar a concorrência e manter a agilidade. Os níveis de isolamento compõem a Tabela 3.2.

Tabela 3.2 - Níveis de Isolamento

Níveis	Bloqueio de Leitura	Bloqueio de Escrita
N - 0	Não há	Tranca curta
N - 1	Não há	Tranca longa
N - 2	Tranca curta	Tranca longa
N - 3	Tranca longa	Tranca longa

Os níveis mais baixos não requerem tranca de leitura, o que pode aumentar a concorrência ao item de dado acessado, obtendo melhor desempenho. Os níveis vão ficando mais estritos até atingir o nível N-3, que garante a serialização. Bloqueios de curta ou longa duração são possíveis através do algoritmo de bloqueio de dados em duas fases (2PL).

Aliar o 2PL aos níveis de isolamento foi uma das táticas encontradas por Rocha e Toledo (2007) para ser usada como mecanismo de adaptação de transações sujeitas ao dinamismo do ambiente. Outro mecanismo de adaptação usado é o modo de operação, em que três opções são consideradas: remoto (R), local-remoto (LR) e local (L). A permuta entre as alternativas ocorre em resposta às mudanças ocorridas no ambiente.

O modo R diz respeito às operações realizadas remotamente. Todas as operações da mesma sobre os itens de dados participantes são executadas nas máquinas remotas onde se encontram. Nessa situação, o algoritmo 2PL é responsável pelo controle de concorrência e obedecerá ao nível de isolamento especificado pela transação.

No modo LR, os itens de dados necessários às transações são disponibilizados na cache das máquinas onde as operações estão sendo executadas. Essas transações são chamadas de *caching*. Essa opção permite que as transações continuem a executar mesmo enquanto os dispositivos permanecem desconectados do restante da rede fixa.

O modo LR obtém acesso exclusivo aos mesmos itens de dados participantes da transação móvel disponíveis na rede fixa, garantindo que as operações realizadas localmente possam posteriormente ser efetivadas nos servidores da rede fixa. Um prazo de expiração impede travamentos indefinidos e a propagação das alterações nos itens de dados em cache mantém a correspondência entre as cópias.

Quando a transação não requer acesso exclusivo aos itens de dados remotos, encontra-se no modo L. Nessa opção, o Protocolo de Validação do Modo Local (PVML) é usado na fase de efetivação para verificar se a transação é válida ou não. Se a transação for considerada válida, ela será efetivada nos servidores da rede fixa; senão, ela será abortada.

A mobilidade do processamento é o terceiro mecanismo aplicado como forma de adaptação. Nesse caso, as transações movem-se entre unidades móveis e estações de base em busca dos recursos adequados para sua execução.

Uma transação atuando no modo de operação L ou LR pode optar pelo movimento para outra máquina diante da escassez de recursos. Diante disto, a PGTA transfere as cópias dos objetos participantes da máquina de origem para a máquina de destino juntamente com a transação. Isso permite que a transação possa continuar a sua execução na máquina de destino sobre as mesmas cópias e do mesmo ponto em que havia parado.

Uma operação no modo R também pode se mover para outra máquina. Apesar de continuar a sua execução do ponto em que parou, não precisará transferir nenhum item de dado disponibilizado em cache.

Rocha e Toledo (2007) empregam uma adaptação colaborativa entre os mecanismos apresentados e a PGTA. A plataforma é responsável por monitorar recursos de conectividade e do dispositivo, além de flexibilizar a propriedade do isolamento. Às transações cabem manter as políticas de adaptação necessárias para reagir às mudanças do ambiente.

A PGTA possui uma arquitetura distribuída e foi proposta para um cenário constituído de um conjunto de máquinas fixas interligadas que forma a rede fixa, estações de base interligadas e unidades móveis com interface de comunicação sem fio. As estações de base são intermediadoras da comunicação entre as unidades móveis e a rede fixa.

A arquitetura possui elementos nas três camadas citadas, como o Monitor de Recursos, Gerenciador de Cache, Repositório de Objetos e Gerenciador de Mobilidade. A Figura 3.5 oferece uma visão geral da Plataforma de Gerenciamento de Transações Adaptáveis.

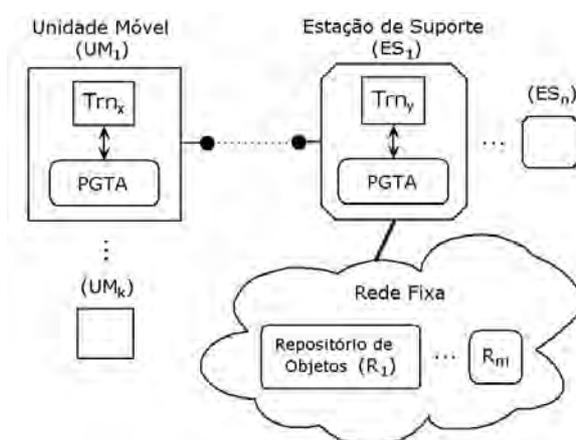


Figura 3.5 - Visão geral da PGTA

Fonte: Rocha e Toledo (2007)

O Monitor de Recursos (MR) e o Gerenciador de Cache (GC) são responsáveis respectivamente pelo acompanhamento da volatilidade de recursos como, por exemplo, energia e espaço em disco do dispositivo móvel, bem como pelos itens de dados carregados na cache dos mesmos.

A persistência dos objetos é responsabilidade do Repositório de Objetos (RO) disponível na rede fixa. O Gerenciador de Mobilidade (GM) apóia o movimento de uma transação de uma máquina para outra, atuando a partir da estação de suporte.

A PGTA aliada aos mecanismos de adaptação integra uma solução para que transações em ambiente de computação móvel sejam capazes de conhecer e adaptar-se às variações de ambientes móveis de acordo com os seus requisitos.

3.3 Considerações

Os trabalhos apresentados propuseram alguma flexibilidade no processamento de transações em ambiente de computação móvel através do emprego de mecanismos de adaptação. Uma análise apreciativa que trata as características relevantes dos trabalhos é apresentada nesta seção.

O modelo ATM e o MTS oferecem um conjunto de possibilidades de execução de uma transação. Após a avaliação e escolha da melhor alternativa, não é possível alterá-la diante de mudanças ocorridas em tempo de execução, ou seja, a definição pela estratégia de execução é estática. O trabalho apresentado por Serrano-Alvarado (2002) também considera que cópias dos dados da rede fixa estão disponíveis apenas nos dispositivos móveis, visto que não há previsão de replicação dos itens de dados também na rede fixa.

A arquitetura do MTS prevê a disponibilização de serviços na estação de base, devido à alta taxa de dispositivos que se registram nos domínios de uma estação de base. Essa decisão, contudo, pode implicar em um gargalo que atrapalhe a escalabilidade da aplicação.

O MTS pressupõe uma configuração capaz de hospedar um SGBD nos dispositivos móveis, dificultando o acesso à aplicação por uma gama de clientes com equipamentos de configuração modesta. O descritor de ambiente disponibilizado na estação móvel é composto por um conjunto de opções necessárias para cada transação poder se adaptar, o que requer um espaço de armazenamento ocasionalmente indesejável.

Na Plataforma de Gerenciamento de Transações Adaptáveis, Rocha e Toledo (2007) aplicaram a adaptação colaborativa entre as transações e a ferramenta proposta. O trabalho não define um modelo de transação e sim propõe a utilização de mecanismos que flexibilizam a consistência e o isolamento.

Essa estratégia adiciona responsabilidades às transações como, por exemplo, a requisição do monitoramento dos recursos e a definição do nível de isolamento. A execução compartilhada da transação não está presente nos modos de operação previstos, assim sua ocorrência está restrita ao dispositivo móvel ou a rede fixa.

Outra ferramenta de adaptação é a migração das transações e das cópias dos objetos em busca de recursos adequados para execução. Um *overhead* no canal de comunicação sem fio pode ser ocasionado pelo uso em demasia dessa técnica, haja vista a volatilidade dos recursos dos equipamentos móveis.

O tratamento de falhas está previsto no Repositório de Objetos disponibilizado na rede fixa. Uma abordagem específica poderia ser utilizada no dispositivo móvel para tratar falhas locais.

Apesar dos trabalhos apresentados empregarem estratégias de adaptação às transações, uma incompleteza ainda pode ser identificada nos quesitos processamento compartilhado para distribuição da carga de execução, avaliação dos dados disponíveis em cache, tratamento da desconexão e mudança dinâmica de execução, entre outros.

Através da investigação apreciativa dos trabalhos que aplicam estratégias de adaptação no processamento de transações em ambiente de computação móvel, foi possível concluir que alguns requisitos desse cenário podem ser melhor explorados ou até mesmo acrescidos. Diante disto, intensificam-se as justificativas citadas na Seção 1.1, que prevêm a proposta de um modelo transacional e de uma ferramenta capazes de agregar tais funcionalidades às aplicações de banco de dados móveis.

4 MID-MOBILE

Os trabalhos e as ponderações apresentadas no capítulo anterior favorecem o desenvolvimento de uma abordagem de adaptação baseada nas deficiências do ambiente. Segundo Coulouris et al. (2007), a adaptação só é possível a partir da modelagem de soluções computacionais em função de alguns fatores, logo, especificar tais fatores é fundamental para o emprego da solução de adaptação mais adequada.

A Figura 4.1 enfatiza dois fatores preservados de modelos anteriores e a inserção do fator "Dados disponíveis no dispositivo móvel" utilizado durante a fase de adaptação. Três pontos destacados na figura enfocam cenários tratados pela ferramenta proposta nesta pesquisa.

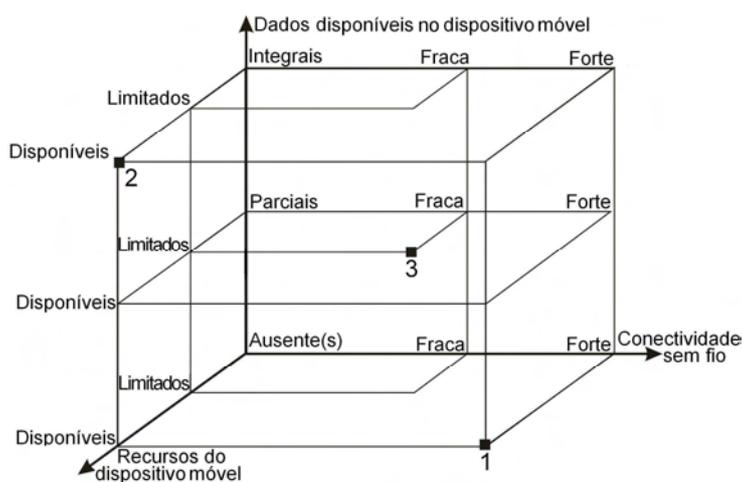


Figura 4.1 - Fatores considerados para a adaptação das transações

O ponto numerado como 1 representa uma situação em que o dispositivo móvel apesar de apto a exercer processamento e de usufruir de boa conectividade, não executa a transação por não ter dados locais, adaptando a execução para ser exercida na rede fixa. Neste caso, é necessário que a ferramenta proposta usufrua de um mecanismo que mantenha um canal de comunicação entre a aplicação disponível no computador móvel e os dados da rede fixa.

O ponto 2 exemplifica a adequação do processamento para o próprio dispositivo quando há recursos e dados disponíveis na estação móvel, mas a conectividade deixa a desejar. Disponibilizar dados em cache é a estratégia mais utilizada na ausência de conectividade, mas essa decisão agrega novas preocupações com a política de aquisição dos dados e com a manutenção da consistência. A consistência dos dados em cache interfere diretamente no processamento de transações móveis, haja visto que operar sobre dados inconsistentes pode ocasionar resultados incorretos ou até mesmo cancelamentos em cascata.

O ponto 3 ilustra a aplicação do processamento compartilhado da transação entre dispositivo móvel e rede fixa. Esse exemplo ocorre quando os dados em cache são incompletos ou desatualizados, ou quando a conectividade é insuficiente para efetuar uma transferência de dados, ou até mesmo quando os recursos estão esgotados, permitindo apenas computação limitada.

Os cenários citados apresentam a identificação dos fatores considerados durante a adaptação. É possível verificar a necessidade de um mecanismo capaz de realizar o reconhecimento do ambiente e decidir pelo plano de execução mais adequado para a transação móvel. A próxima seção formaliza a arquitetura do MiD-Mobile, *middleware* responsável pelas atividades citadas.

4.1 Arquitetura do Mid-Mobile

O MiD-Mobile é um *middleware* distribuído implementado para ambientes de computação móvel. Sua arquitetura prevê duas camadas, uma hospedada na rede fixa enquanto a outra é disponibilizada no computador móvel. A Figura 4.2 oferece uma visão geral da arquitetura do *middleware*.

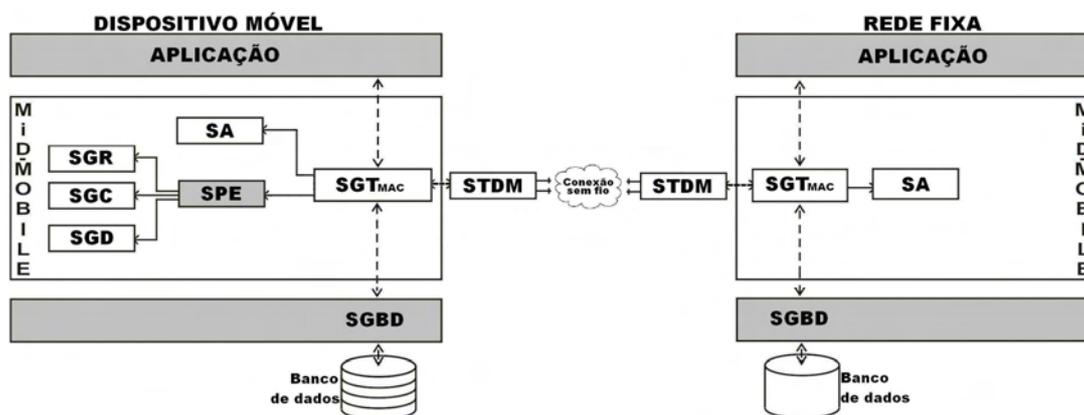


Figura 4.2 - Arquitetura do MiD-Mobile

A camada do MiD-Mobile hospedada na rede fixa é disponibilizada aos desenvolvedores de aplicação para a realização do treinamento do serviço de aprendizagem. Tal serviço é usado durante a definição do plano de execução da T_{MAC} no dispositivo móvel.

Na plataforma móvel, a atuação do MiD-Mobile é transparente e é acionada a partir da emissão das transações. Nesta camada a função principal do *middleware* é fazer o reconhecimento do ambiente, decidir e gerenciar o plano de execução adotado para a T_{MAC} solicitante.

O plano de execução prevê a estratégia mais adequada para o processamento da transação. As opções para o plano de execução são: isolada, remota ou compartilhada. No plano de execução isolada a transação ocorre no dispositivo móvel. A opção remota prevê o envio do processamento para a rede fixa. A execução compartilhada distribui o processamento entre os extremos.

O plano mais adequado é indicado a partir da caracterização do contexto móvel, realizado pelos serviços de gerenciamento de recursos, conexão e dados. A Tabela 4.1 apresenta as dimensões que influenciam na caracterização do contexto, com a discriminação detalhada de seus atributos e seus conteúdos discretizados.

Tabela 4.1 - Aspectos avaliados durante a elaboração do contexto móvel

Dimensões	Detalhes	Conteúdos
Rede	Estado da conexão	Conectado (0) Desconectado (1)
	Taxa de transferência	Suficiente (0) Intermediária (1) Insuficiente (2)
	Custo de transmissão	Gratuito (0) Tarifado (1) Indefinido (2)
Dados	Situação dos dados	Completo (0) Incompleto (1) Indefinido (2)
	Validade dos dados	Dados Válidos (0) Inválidos (1) Indefinido (2)
Dispositivo	Capacidade de Bateria	Suficiente (0) Intermediária (1) Insuficiente (2)
	Capacidade de hardware	Suficiente (0) Intermediária (1) Insuficiente (2)

Para que o plano sugerido possa oferecer chances reais de êxito, ele é proposto logo no início da transação. O Serviço Gerenciador de Transações Móveis Adaptadas ao Contexto (SGT_{MAC}) recebe a requisição e inicia o Serviço de Plano de Execução (SPE). O SPE requisita o Serviço de Gerenciamento de Recursos (SGR), o Serviço de Gerenciamento da Comunicação (SGC) e o Serviço de Gerenciamento dos Dados (SGD) para elaborar o contexto móvel.

Cada serviço atua junto a uma das dimensões apresentadas na Tabela 4.1. Os dados fornecidos são usados pelo SPE durante a definição do plano. Ao término da execução da transação no plano indicado, o SGT_{MAC} convoca o Serviço de Aprendizagem (SA) para registrar a ocorrência e os resultados obtidos. O serviço de troca de dados e mensagens (STDM) é acionado quando é necessário transmitir dados ou mensagens entre os dois extremos. As próximas seções oferecem mais detalhes sobre cada serviço do *middleware*.

4.1.1 Serviço de Gerenciamento de Recursos

O Serviço de Gerenciamento de Recursos (SGR) foi proposto devido à grande heterogeneidade de recursos de computação disponíveis nos dispositivos portáteis. A diversidade citada tem forte influência no processamento das T_{MACs} devido à divergência de capacidade de processamento, armazenamento e comunicação.

Sendo assim, a função do SGR é verificar a capacidade do dispositivo móvel agrupando as características em dois eixos principais, a capacidade de hardware e de bateria, conforme explora a dimensão "Dispositivo" da Tabela 4.1.

A avaliação da bateria interessa para checagem da autonomia que o dispositivo possui. Essa situação deve ser considerada, pois não adianta o dispositivo possuir bons recursos se a transação não tiver chance de chegar até o fim devido ao esgotamento da energia.

O segundo item diz respeito à verificação da capacidade de hardware para avaliar o potencial de processamento da transação e a disponibilidade de espaço de armazenamento para hospedagem local de dados.

As características do dispositivo podem ser adquiridas pelo CC/PP (*Composite Capability/Preference Profiles*), através da especificação UAProf da Open Mobile Alliance. O esquema é liberado pelo fabricante do equipamento e apresenta informações a respeito das características dos dispositivos.

A Figura 4.3, retirada do site da W3C, apresenta uma parte de um exemplo especificado pelo UAProf de um dispositivo. As partes ressaltadas apresentam algumas características relevantes ao SGR como as características de hardware.

```

<?xml version='1.0' ?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:prf="http://www.example.org/TR/WD-profile-vocabulary#"
xmlns:exp="http://weather.co.jp/RDF/exp-schema#">
<rdf:Description rdf:ID="MySituationProfile">
  <rdf:Description rdf:ID="MyDeviceProfile">
    <prf:component>
      <rdf:Description rdf:ID="HardwarePlatform">
        <rdf:type rdf:resource="http://www.wapforum.org/profiles/UAPROF/ccppschem-20010430#HardwarePlatform"/>
        <prf:Vendor>Ericsson</prf:Vendor>
        <prf:Model>t380</prf:Model>
        <prf:Type>PDA</prf:Type>
        <prf:ScreenSize>800x600x24</prf:ScreenSize>
        <prf:CPU>PPC</prf:CPU>
        <prf:Keyboard>No</prf:Keyboard>
        <prf:Memory>16MB</prf:Memory>
        <prf:Bluetooth>Yes</prf:Bluetooth>
        <prf:Speaker>Yes</prf:Speaker>
      </rdf:Description>
    </prf:component>
    <prf:component>
      <rdf:Description rdf:ID="SoftwarePlatform">
        <rdf:type rdf:resource="http://www.wapforum.org/profiles/UAPROF/ccppschem-20010430#SoftwarePlatform"/>
        <prf:OS>EPOC1.0</prf:OS>
        <prf:HTMLVersion>4.0</prf:HTMLVersion>
        <prf:JavaScriptVersion>4.0</prf:JavaScriptVersion>
        <prf:WAPVersion>2.0</prf:WAPVersion>
      </rdf:Description>
    </prf:component>
  </rdf:Description>

```

Figura 4.3 – Arquivo UAProf de um dispositivo móvel

Fonte: Site do W3C

4.1.2 Serviço de Gerenciamento da Comunicação

O canal de comunicação sem fio pode empregar tecnologias diferentes como Bluetooth, Wi-Fi, Wi-Max, GSM entre outras. Cada uma dessas opções possui uma taxa de transferência e a possibilidade de tarifação. O estado do canal de comunicação também exerce influência sobre a execução da transação. Um exemplo ocorre quando uma operação em andamento é cancelada devido à queda da conexão. A inconstância deve-se tanto ao movimento do usuário quanto a fragilidade da conectividade.

A indisponibilidade pode ser momentânea devido à passagem por uma área de sombra, ou até mesmo temporária devido ao distanciamento do equipamento nômade da célula de comunicação que transmite o sinal.

O SGC, a partir do SNMP (Simple Network Management Protocol - Protocolo Simples de Gerência de Rede), pode reconhecer o meio de comunicação sem fio que está sendo utilizado, avaliando o estado da conexão e a taxa de transmissão. O SGC ao identificar o tipo de conexão, deve ser informado sobre a possibilidade de cobrança pela transmissão dos dados.

A questão do custo da comunicação deve ser considerada, pois uma estratégia adequada deve minimizar os dados transmitidos pela conexão sem fio, mesmo que a conexão esteja boa.

Os tipos de tecnologias de comunicação suportadas por cada dispositivo também constam do esquema do CC/PP. A Figura 4.4 ressalta a natividade do Bluetooth e do GPRS como tecnologias de transmissão sem fio de dados.

```
- <rdf:Description rdf:ID="NetworkCharacteristics">
  <rdf:type rdf:resource="http://www.openmobilealliance.org/tech/profiles/UAPROF/ccppschem-20021212#NetworkCharacteristics"/>
  <prf:SupportedBluetoothVersion>Bluebird BTHCost2.0</prf:SupportedBluetoothVersion>
- <prf:SecuritySupport>
  - <rdf:Bag>
    <rdf:li>TLS</rdf:li>
    <rdf:li>SSL</rdf:li>
  </rdf:Bag>
  </prf:SecuritySupport>
- <prf:SupportedBearers>
  - <rdf:Bag>
    <rdf:li>CSD</rdf:li>
    <rdf:li>GPRS</rdf:li>
  </rdf:Bag>
  </prf:SupportedBearers>
</rdf:Description>
</prf:component>
```

Figura 4.4 – Tecnologias de rede de comunicação no protocolo CC/PP

Fonte: Site do W3C

4.1.3 Serviço de Gerenciamento dos Dados

Uma das formas de minimizar as conseqüências da falta de conexão é a técnica de *hoarding*. O *hoarding* diz respeito à carga antecipada dos dados, ou seja, diante de uma previsão de desconexão, o usuário pode armazenar cópias dos dados nas caches dos dispositivos móveis. A Figura 4.5 apresenta o método responsável por obter os dados do servidor.

```
public void recebeInformacoes() {
    try {
        getTbLog().insert("Obtendo dados do servidor ", getTbLog().size());
```

Figura 4.5 – Método de sincronização solicitado pelo dispositivo móvel

Fonte: Arquivo de código-fonte do BancoME

O método de sincronização, responsável por enviar dados do dispositivo móvel para a rede fixa está presente na Figura 4.6.

```
private void sincroniza() {
    new Thread() {
        public void enviaInformacoes() {
            getTbLog().insert("Enviando dados ao servidor...", getTbLog().size());
        }
    };
}
```

Figura 4.6 – Método de sincronização enviado à rede fixa

Fonte: Arquivo de código-fonte do BancoME

Tais operações são amparadas pelo Serviço Gerenciador de Dados (SGD). Outra função deste serviço é verificar a consistência global dos dados por meio do Protocolo de Consistência Global (PCG).

O PCG verifica as versões dos itens de dados disponíveis nas duas plataformas. Quando as versões divergem significa que a consistência global está em risco. A Figura 4.7 apresenta um exemplo de linha de código que valida a consistência dos dados antes de executar a transação.

```
try {
    String d = l[i].getDescricao();
    getTbLog().insert("Sincronizando: " + l[i].getDescricao() , getTbLog().size());
    if (bwss.getTaxas().getId() > bsl.getTaxas().getId()) {
        getTbLog().insert(" não realizado devido a alteração nas taxas.\n", getTbLog().size());
    } else {
```

Figura 4.7 – Protocolo Consistência Global

Fonte: Arquivo de código-fonte do BancoME

4.1.4 Serviço de Plano de Execução e de Aprendizagem

A decisão sobre o plano de execução mais adequado à T_{MAC} é vista como um problema de classificação. Classificar significa qualificar um exemplo em uma classe. No presente trabalho fez-se uso de Árvores de Decisão (AD) nessa tarefa.

De acordo com Russel e Norvig (2003), a criação de uma árvore de decisão consiste no particionamento recursivo de instâncias de treinamento, até que cada subconjunto contenha casos de uma única classe.

O nó raiz apresenta o atributo que inicia a árvore, enquanto cada ramo descendente leva a outro nó em uma camada seguinte. Cada nó contém um teste em um atributo e os ramos correspondem aos valores possíveis desses atributos. Os nós que estabelecem a classificação são chamados folhas e estão na última camada da hierarquia. Nós e ramos que compõem uma árvore de decisão são apresentados na Figura 4.8.

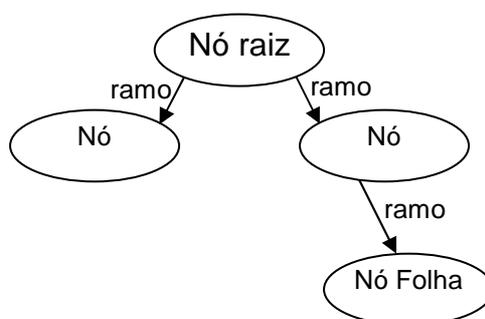


Figura 4.8 – Componentes de uma Árvore de Decisão

Cada percurso na árvore, ou seja, um caminho da raiz à folha, corresponde a uma regra de classificação. Tais regras podem ser escritas no formato SE-ENTÃO, facilitando o entendimento, principalmente em árvores grandes.

Neste estudo, a classificação diz respeito a indicação do plano de execução mais apropriado à T_{MAC} . A Figura 4.9 ilustra um percurso da árvore gerado por uma entrada no respectivo conjunto de treinamento.



Figura 4.9 - Modelo de árvore de decisão e de conjunto de treinamento

Cada linha da "Lista de Exemplos para Treinamento", interface da Figura 4.9, representa uma percurso da árvore. O nó folha exibe a classificação propriamente dita. Existem vários métodos de classificação. A escolha por árvore de decisão deve-se as seguintes características:

- instâncias representadas através de pares atributo-valor;
- representatividade discreta de valores;
- robustez diante de ruídos;
- ausência de valores de atributos;
- ocorrência de descrições disjuntas;

Cada dimensão citada na Tabela 4.1 é avaliada para proceder a estratégia de adaptação das T_{MACs} , os itens apresentados encaixam-se com as características do objeto de estudo dessa pesquisa. Os elementos identificados constituem um par de atributo-valor, como, por exemplo, estado da conexão - conectado. A discretização dos valores facilita o aprendizado quando cada atributo assume valores disjuntos (Suficiente, Insuficiente, Desconhecido), inclusive na classificação (isolado, remoto, compartilhado).

Cada percurso representa uma disjunção de combinações dos elementos da Tabela 4.1. As árvores de decisão admitem erros nas classificações dos exemplos de treinamento e nos valores dos atributos, além da ausência dos mesmos. Essa condição é valiosa quando não for possível manter a conectividade e os atributos da dimensão rede não puderem ser captados.

O algoritmo de indução utilizado no MiD-Mobile foi o ID3, apresentado em Mitchell (1997). A escolha justifica-se pela completeza no espaço de busca da melhor árvore, pelo ganho de desempenho devido à ausência de *backtracking* (reconsideração de atributos anteriores), pela robustez diante de instâncias erroneamente classificadas e até mesmo frente à ausência de valor dos atributos.

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 ou impureza) inserido em um conjunto de treinamento. Essa avaliação é chamada de entropia. Em um conjunto de treinamento S o cálculo da entropia é dado pela Equação 4.1:

$$Entropia(S) \equiv \sum_{i=1}^N p_i \log_2(p_i) \quad (4.1)$$

A entropia é máxima (1) quando as classes são equiprováveis e mínima e igual a 0 quando todos os exemplos pertencem a mesma classe. Após o cálculo da entropia de S, a fórmula do ganho de informação, descrita na Equação 4.2, é aplicada para determinar o melhor atributo a ser usado como raiz.

$$Ganho(S,A) \equiv Entropia(S) - \sum_{v \in \text{valores}(A)} \frac{|S_v|}{|S|} Entropia(S_v) \quad (4.2)$$

Em benefício da visualização, a Tabela 4.2 oferece um conjunto de treinamento (S) usado por um algoritmo ID3 no exemplo a ser explorado a seguir.

Tabela 4.2 – Conjunto de treinamento de exemplo

Instância	Estado	Taxa	Classe
1	con	sat	A
2	des	sat	A
3	des	ins	A
4	des	ins	B
5	con	ins	A
6	con	sat	B
7	des	ins	A
8	des	sat	A
9	des	ins	B
10	des	ins	B
11	des	ins	A
12	des	sat	A
13	des	ins	A
14	des	ins	B

O cálculo de entropia de S é dado por:

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

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

Com os dados disponíveis na Tabela 4.2 é possível calcular o ganho para cada atributo. Para o atributo Estado os cálculos são:

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

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

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

Para o atributo Taxa os cálculos são:

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

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

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

Comparando o ganho do atributo Estado com o ganho do atributo Taxa, percebe-se que o resultado de Estado é maior e ele será usado no início da classificação como nó raiz. A aplicação recursiva do algoritmo leva a escolha dos próximos atributos, treinando assim a árvore de decisão. A criação da árvore é responsabilidade do MiD-Mobile disponível na rede fixa. A Figura 4.10 exibe a interface do menu principal.

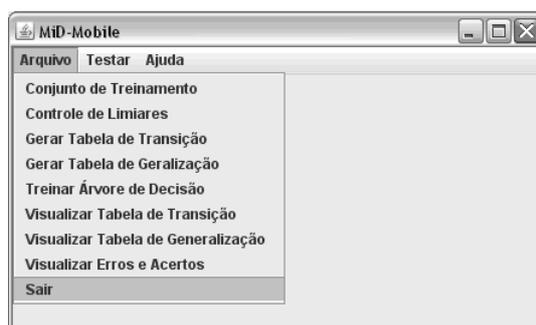


Figura 4.10 – Menu principal do MiD-Mobile

Fonte: Menu principal do MiD-Mobile

Através das opções do menu é possível gerenciar o conjunto de treinamento, os limiares de confiança, realizar os treinamentos e visualizar os elementos gerados. A Figura 4.11 retrata a interface de cadastramento do conjunto de treinamento.

Código	Estado	Taxa	Custo	Dados	Complectude	Validade	Bateria	Hardware	Plano
1	0	0	0	0	0	2	2	1	
2	0	0	0	1	1	1	2	2	2
3	1	1	1	1	1	1	1	1	4
4	1	2	2	1	1	1	1	2	3
5	0	0	0	1	1	1	1	0	1
6	1	1	1	1	1	1	1	2	4
7	0	0	0	0	0	0	1	1	1
8	0	0	0	1	0	0	2	2	1
9	1	1	1	1	0	0	2	2	4
10	1	2	2	1	1	0	1	2	3

Figura 4.11 - Conjunto de treinamento utilizado para a geração da árvore de decisão

Fonte: Menu principal do MiD-Mobile

Com o conjunto de exemplos cadastrado é possível gerar a árvore. A Figura 4.12 exibe o item do menu responsável pelo treinamento da árvore.

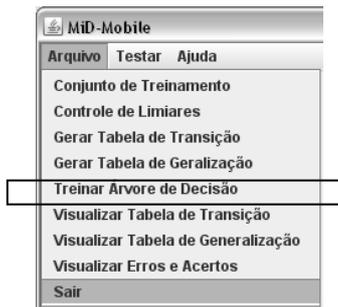


Figura 4.12 – Opção de treinamento da árvore de decisão no Menu Arquivo

Fonte: Menu principal do MiD-Mobile

Uma árvore de decisão pode ser convertida em regras de decisão do tipo SE-ENTÃO. Dessa forma, é possível manter a árvore bem ajustada, generalizando corretamente casos não treinados. Outro benefício para a utilização de regras diz respeito ao algoritmo usado no Serviço de Plano de Execução hospedado no dispositivo móvel. Uma discussão mais completa é tratada mais adiante.

O Serviço de Aprendizagem (SA) está hospedado nas duas camadas do MiD-Mobile. No ambiente estacionário o objetivo principal é avaliar a árvore diante de novas instâncias. A Figura 4.13 apresenta a interface de teste da árvore treinada, onde é possível simular e avaliar a generalização.

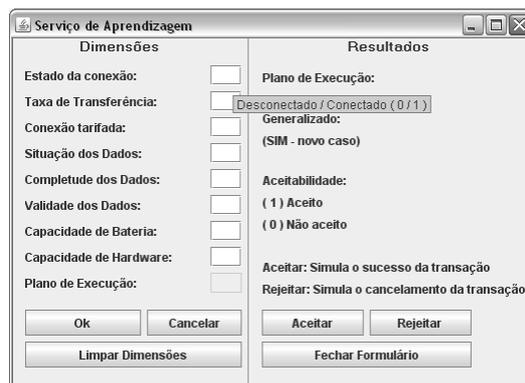


Figura 4.13 – Interface do Serviço de Aprendizagem

Fonte: Serviço de Aprendizagem do MiD-Mobile

A Figura 4.14 exibe os testes que podem resultar em planos generalizados ou não. Aceitar ou rejeitar o plano indicado exemplifica a ocorrência da consolidação (*commit*) ou do cancelamento (*rollback*) da transação.

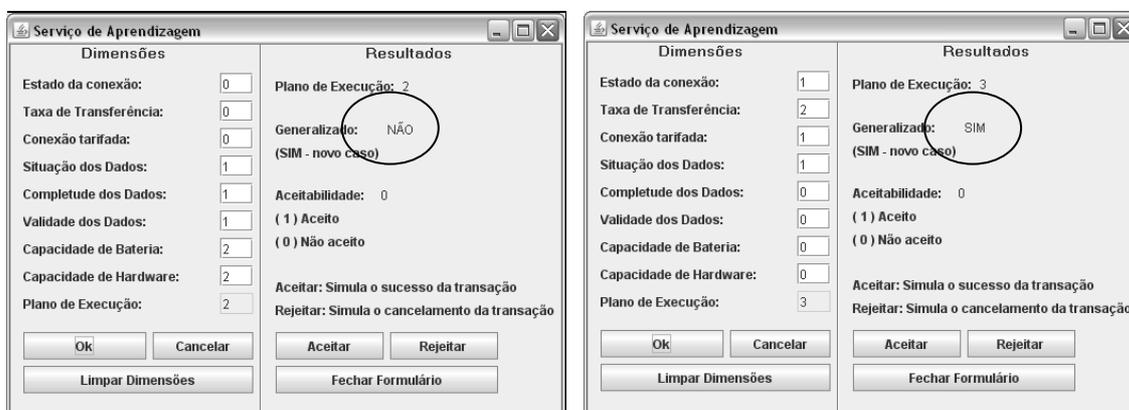


Figura 4.14 – Interface de exibição de planos generalizados ou treinados

Fonte: Serviço de Aprendizagem do MiD-Mobile

A Figura 4.15 detalha a escolha dos limiares de confiança para reter ou desprezar um caso de teste.

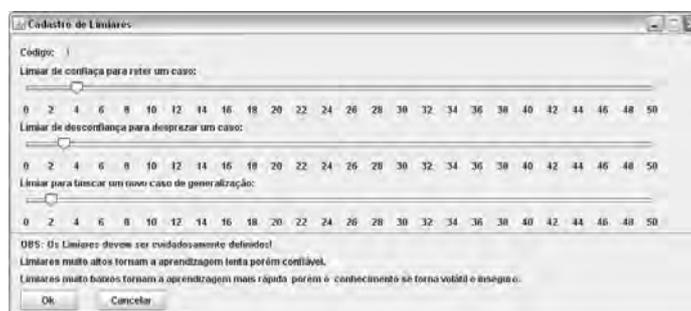


Figura 4.15 - Controle de limiares de confiança

Fonte: Cadastro de Limiares do MiD-Mobile

A generalização de novos casos oferece alternativas de execução às T_{MACs} . Por isso, o controle de limiar permite gerenciar a quantidade de vezes que os planos indicados serão aplicados (limiar de confiança para reter um caso), impedindo que um plano incorreto seja aplicado repetidas vezes (limiar de confiança para desprezar um caso). Essa política oferece ao desenvolvedor de aplicação, generalizações cada vez mais precisas e adequadas a cada caso em particular.

Uma árvore de decisão é útil se o algoritmo de generalização usado classifica corretamente exemplos não treinados. Essa é a principal função do Serviço de Plano de Execução (SPE). O SPE é um serviço mantido na camada móvel do *middleware* e, por isso, o algoritmo a ser usado na generalização deve ser leve e rápido, ao contrário do ID3 que emprega recursividade fazendo crescer progressivamente o uso da memória e do processamento.

A solução para manter a simplicidade, velocidade e economia de recursos computacionais do equipamento portátil foi inspirada em Autômatos Finitos Determinísticos (AFD). Um AFD é um modelo matemático que permite especificar algumas linguagens formais. De acordo com Menezes (1998), este modelo é capaz de descrever a classe de Linguagens Regulares, a linguagem mais restrita dentre as descritas na Hierarquia de Chomsky.

A mesma simplicidade que impõe limitações severas à descrição de linguagens formais caracteriza uma vantagem no contexto de sistemas de computação móveis. A baixa complexidade, facilidade de implementação e eficiência dos algoritmos são características bem-vindas diante da escassez de recursos computacionais dos dispositivos portáteis. O algoritmo do AFD é um reconhecedor de cadeias de símbolos baseado em um conjunto finito e não vazio de estados, podendo facilmente ser adequado para percorrer a AD.

A definição matemática do AFD é dada pela quintupla $\langle \Sigma, S, S_0, \delta, F \rangle$, onde:

- a) Σ é o alfabeto de entrada, um conjunto finito de símbolos que podem estar presentes na seqüência de entrada processada pelo AFD;
- b) S é um conjunto finito e não vazio de estados;
- c) S_0 é o estado inicial ativado no instante em que o AFD é iniciado;

- d) δ é um conjunto de funções de transição que mapeiam um par formado pelo estado ativo e o símbolo presente na seqüência de entrada. Matematicamente estas funções têm a seguinte forma: $\delta: S \times \Sigma \rightarrow S$;
- e) F é um conjunto de estados finais, que ao serem ativados ao término do processamento determinam a aceitação da cadeia de entrada.

Considere um AFD onde:

$$\Sigma = \{0, 1\}$$

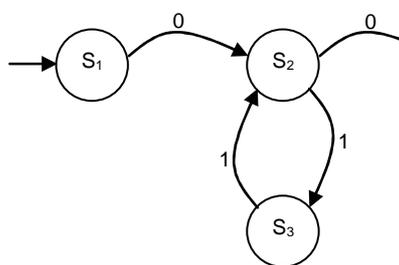
$$S = \{S_1, S_2, S_3, S_4\}$$

$$S_0 = S_1$$

$$\delta = \{(S_1, 0) \rightarrow S_2, (S_2, 0) \rightarrow S_4, (S_2, 1) \rightarrow S_3, (S_3, 1) \rightarrow S_2\}$$

$$F = \{S_4\}$$

Este AFD pode ser descrito pelo diagrama ou tabela de transições da Figura 4.16.



δ	0	1
$\rightarrow S_1$	S_2	-
S_2	S_4	S_3
S_3	-	S_2
* S_4	-	-

Figura 4.16 - Diagrama e Tabela de transições do AFD

Fonte: Menezes (1998)

No diagrama de transições, os estados são representados por círculos, devidamente rotulados com os nomes dos estados correspondentes. As funções de transição são descritas pelos arcos que aparecem no diagrama. Estes arcos têm uma correspondência com as funções de forma que ao arco está associada uma função de transição que tem como origem o estado de onde o arco parte, processa na seqüência de entrada o símbolo que rotula este arco e ativa o estado correspondente ao destino do arco.

Na Figura 4.16, o estado inicial e o estado final recebem uma marcação especial para identificá-los. S1 está caracterizado como estado inicial por uma seta e S4 está caracterizado como estado final pelo círculo duplo. Na tabela de transições, as colunas indicam os estados e as linhas o alfabeto, enquanto o cruzamento entre ambas indicam as funções de transição. O estado inicial é indicado por uma seta e o final por asterisco.

O comportamento do AFD durante o processamento de uma seqüência de entrada pode ser definido pelo algoritmo descrito a seguir:

Início

Estado Atual \leftarrow Estado Inicial;

Para I variar do símbolo inicial até o símbolo final da cadeia de entrada

Faça

Início

Se Existe δ (Estado Atual, I)

Então Estado Atual $\leftarrow \delta$ (Estado Atual, I);

Senão REJEITA;

Fim;

Se Estado Atual é estado final

Então ACEITA;

Senão REJEITA;

Fim.

Em termos de processamento, a complexidade do algoritmo do AFD é de ordem $O(n)$, onde n é o tamanho da entrada a ser processada. Em relação a memória, a implementação de um AFD apresenta gasto proporcional a $p \times q$, onde p é o tamanho de Σ e q é o tamanho de S . Este fato é bastante relevante em aplicações com sérias restrições de recursos computacionais e beneficiou a aplicação do AFD aliado às árvores de decisão no SPE.

Para a árvore de decisão induzida pelo ID3 foi gerada uma Tabela de transição a ser disponibilizada junto com a aplicação. O algoritmo do AFD percorre a Tabela de transição da árvore de decisão até verificar a existência de uma instância treinada que informe o plano mais indicado para a T_{MAC} .

Nesta tabela pode ocorrer função de transição parcial devido à ausência de função definida para um par formado por um estado de S e um símbolo de Σ , esse cenário acontece quando a instância é nova e o algoritmo tradicional do AFD trava ao encontrar em seu percurso uma função de transição que não reconhece o símbolo lido a partir do estado anterior.

Duas operações complementares foram usadas nessa situação. Uma adaptação no algoritmo do AFD e a criação de uma tabela de transição gerada pelo algoritmo de generalização. O algoritmo foi implementado para percorrer a tabela de transição da árvore até o subconjunto conhecido, em seguida passa para a tabela de generalização e quantifica a incidência de cada plano nos subconjuntos descentes.

Assim, a tabela de generalização difere da tabela de transição da AD por armazenar quantas vezes um determinado nó esteve presente na classificação de um plano e não o próximo atributo a ser avaliado. Percorrer a tabela de transição e de generalização através de um algoritmo de mudança de estado, conforme razões previamente apresentadas, é mais simples e econômico em termos de memória e processamento do que o algoritmo de indução da AD.

Conforme explorado na Figura 4.16, uma tabela de transição pode ser usada para representar um AFD. Dessa forma, essa característica foi estendida às árvores de decisão e ao algoritmo de generalização. A Figura 4.17 apresenta a equivalência dos elementos.

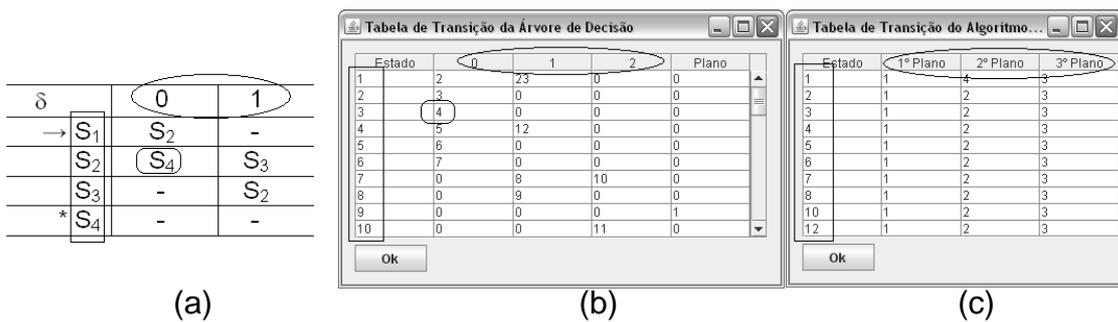


Figura 4.17 - Tabela de transição do AFD (a), da árvore de decisão (b) e do algoritmo de generalização (c)

Fonte: Adaptado de Menezes (1998) e MiD-Mobile

Na tabela de transição da árvore de decisão ilustrada na Figura 4.17 (b), os nós da árvore equivalem aos estados do AFD da Figura 4.17 (a) e são descritos na coluna “Estado”. O mesmo ocorre com a tabela de transição do algoritmo de generalização da Figura 4.17 (c). Os ramos 0, 1 e 2 substituíram os símbolos do alfabeto na Figura 4.17 (b) e os planos de execução (1, 2 e 3) na Figura 4.17 (c). O acréscimo ficou por conta da coluna plano na tabela de transição da árvore que só é preenchida com uma classificação. O cruzamento entre os nós e os ramos direciona o próximo nó a ser testado, Figura 4.17 (b), bem como o número de vezes que aquele nó contribuiu para o plano de execução indicado, Figura 4.17 (c).

O Serviço de Aprendizagem no dispositivo móvel não possui interface de treinamento. Sua função compreende o armazenamento das ocorrências do SPE diante das T_{MACs} . Os dados armazenados englobam as configurações do ambiente, o plano indicado pelo SPE e o resultado da aplicação do plano. As instâncias armazenadas nos dispositivos móveis são reenviadas ao Serviço de Aprendizagem da rede fixa para que uma nova árvore seja gerada.

Em cada domínio de aplicação o desenvolvedor deve decidir o momento mais adequado para o envio das instâncias de testes. O passo seguinte é o encaminhamento da árvore retreinada ao dispositivo móvel.

4.1.5 Serviço Gerenciador de Transações Móveis Adaptadas ao Contexto

O Serviço Gerenciador de Transações Móveis Adaptadas ao Contexto (SGT_{MAC}) é o elemento responsável por coordenar as ações que garantem a execução da transação móvel. Dentre suas funções destacam-se as seguintes:

- a) acionamento do SPE em busca do plano de execução para a T_{MAC};
- b) controle do tempo das subT_{MAC}s;
- c) cancelamento de subtransações ativas;
- d) garantia da execução da transação conforme o modelo indicado;
- e) acionamento do SA do dispositivo móvel.

Quando necessário, as funções são exercidas pelas camadas do SGT_{MAC} da estação móvel e da rede fixa. A Figura 4.18 especifica o método de inicialização do controle do tempo para uma possível interrupção de subT_{MAC}s.

```
public static void executaCompartilhado(final long timeOut, final SubTMAC subTMAC[]) {
    long ti = System.currentTimeMillis();

    final int n = subTMAC.length;
    Thread mataPorTimeOut = new Thread() {

        public void run() {
            try {
                sleep(timeOut);
                for (int i = 0; i < n; i++) {
                    if (subTMAC[i].isAlive()) {
                        subTMAC[i].interrupt();
                    }
                }
            } catch (InterruptedException ex) {
                System.out.println("Timeout Cancelado");
            }
        }
    };
    mataPorTimeOut.start();
}
```

Figura 4.18 – Código-fonte do controle do tempo

Fonte: Código-fonte do BancoOnline

Enquanto o tempo de execução estiver dentro do prazo, as subtransações permanecem ativas, caso contrário são interrompidas e a T_{MAC} é cancelada. Dois trechos de código explorados na Figura 4.19 apresentam o sucesso de todas as $subT_{MACs}$ ou o cancelamento geral.

```

boolean sucesso = true;
for (int i = 0; i < n; i++) {
    sucesso = sucesso && subTMAC[i].isSucesso();
}
if (sucesso) {
    System.out.println("Commit");
} else {
    for (int i=0;i<n;i++){
        System.out.println ("subTMAC id:"+subTMAC[i].getId()+" Sucesso:"+subTMAC[i].isSucesso());
    }
    System.out.println("RollBack");
}

```

Figura 4.19 – Código-fonte da verificação de resultado das $subT_{MACs}$

Fonte: Código-fonte do BancoME

4.2 Serviço de Transmissão de Dados e Mensagens

A comunicação entre o SGT_{MAC} disponível em cada plataforma é provida pelo serviço de troca de dados e mensagens (STDM), inclusive durante desconexões. Uma interface, para fins didáticos, foi gerada para exemplificar as ações do serviço. A Figura 4.20 apresenta a comunicação iniciada por um cliente (dispositivo móvel) e a Figura 4.21 a identificação pelo servidor.



Figura 4.20– Conexão solicitada pelo cliente

Fonte: STDM da camada móvel do MiD-Mobile

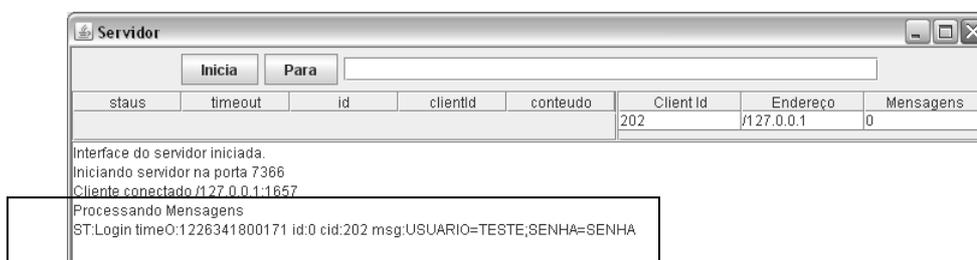


Figura 4.21 – Reconhecimento da Conexão

Fonte: STDM da camada fixa do MiD-Mobile

As Figuras 4.22 e 4.23 mostram uma nova conexão partindo de outro cliente móvel sendo identificada pelo STDM da rede fixa.

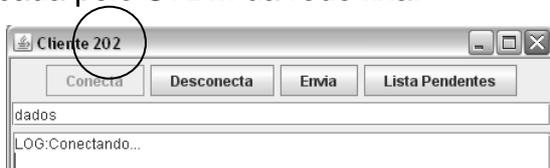


Figura 4.22 – Nova conexão solicitada pelo cliente

Fonte: STDM da camada móvel do MiD-Mobile

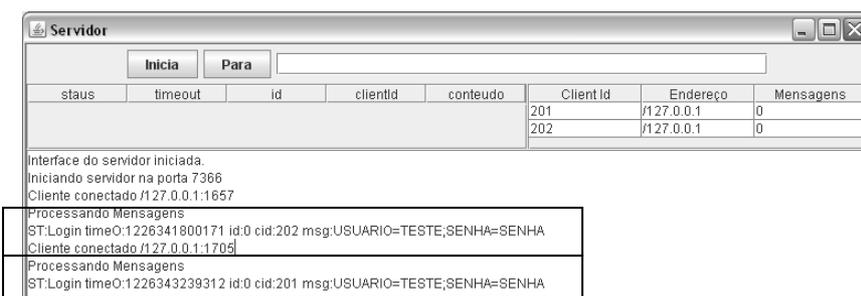


Figura 4.23 – Conexões identificadas

Fonte: STDM da camada fixa do MiD-Mobile

A queda de qualquer um dos elementos ocasiona a falha na comunicação. As Figuras 4.24 e 4.25 representam respectivamente a interrupção da conexão no STDM da rede fixa e do dispositivo móvel.

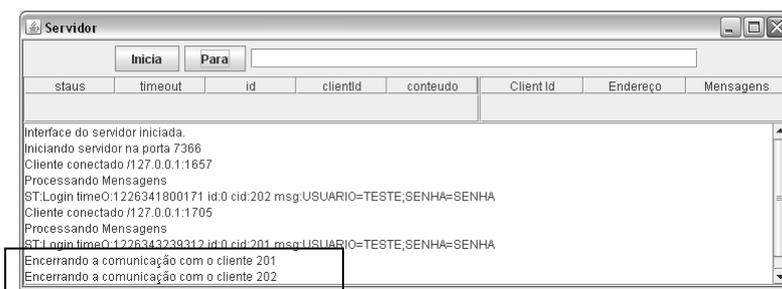


Figura 4.24 – Encerramento da comunicação

Fonte: STDM da camada fixa do MiD-Mobile

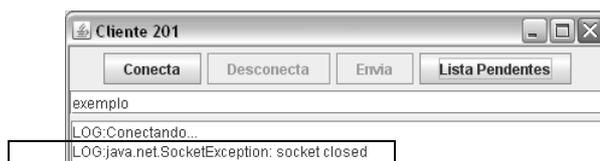


Figura 4.25 – Interrupção da comunicação no cliente

Fonte: STDM da camada móvel do MiD-Mobile

Quando a comunicação é iniciada pelo STDM da rede fixa, os clientes móveis recebem as mensagens, conforme ilustrado na Figura 4.26. A linha destacada na interface da direita representa a mensagem enviada aos clientes móveis. Do lado esquerdo, o conteúdo dentro do retângulo representa o recebimento da mesma mensagem pelos clientes móveis.

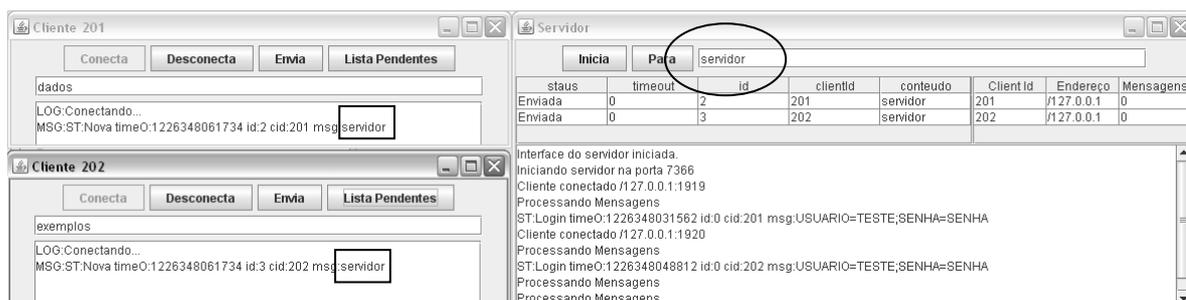


Figura 4.26 – Mensagens enviadas pelo STDM da rede fixa

Fonte: STDM da camada fixa e móvel do MiD-Mobile

Diante de uma desconexão a mensagem permanece em espera para o STDM realizar o envio no momento adequado, conforme ilustrado pela Figura 4.27. A interface da direita mostra o envio de uma mensagem. Do lado esquerdo estão os clientes móveis, um deles está desconectado e não recebe a mensagem, enquanto o outro não encontra problemas no recebimento. A seta, na interface da direita, indica que há uma "nova" mensagem a ser encaminhada para um determinado cliente.

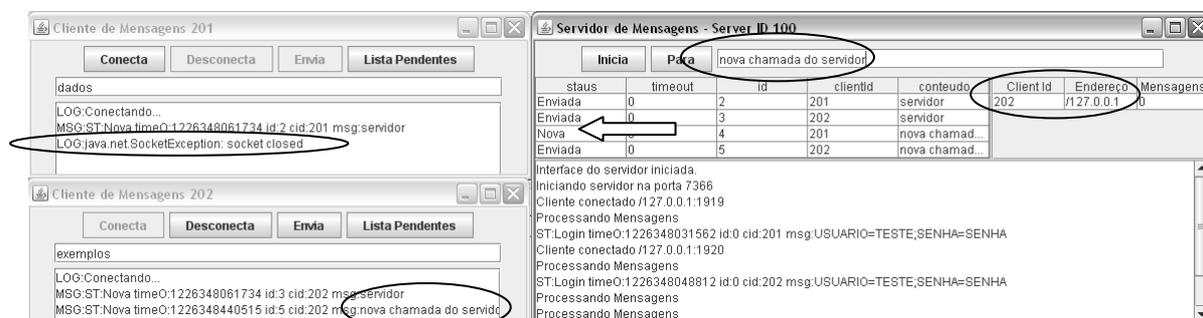


Figura 4.27 – Mensagens não enviadas devido a desconexão

Fonte: STDM da camada fixa e móvel do MiD-Mobile

4.3 Considerações

O MiD-Mobile é um *middleware* disponibilizado junto com a aplicação, conforme a visualização da estrutura de pacotes da aplicação BancoME na Figura 4.28. Seu objetivo é maximizar as chances de sucesso de execução de transações iniciadas a partir de clientes móveis em sistemas de computação móvel.

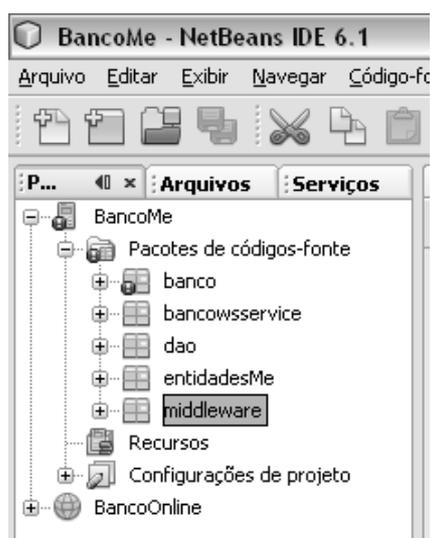


Figura 4.28 – Pacote do *middleware* disponibilizado junto com a aplicação

Dois fatores influenciam negativamente a execução das transações em tal ambiente: a instabilidade do canal de comunicação sem fio e a heterogeneidade dos dispositivos de acesso.

As conseqüências desses fatores são os freqüentes cancelamentos das operações iniciadas pelos usuários. As causas da instabilidade foram identificadas e tratadas com o intuito de minimizar seus efeitos.

A solução adotada para minimizar as interferências resultantes do contexto está baseada na identificação e seleção da estratégia de processamento de maior êxito. Por vezes, essa seleção requer uma adaptação na estrutura da transação, assunto abordado no próximo capítulo.

Os resultados obtidos a partir das decisões tomadas realimentam o processo decisório oferecendo novas ocorrências a serem consideradas. Dessa forma, o MiD-Mobile implementa um mecanismo de aprendizagem capaz de refinar suas escolhas diante de novos casos.

Outro mecanismo usado para contornar as interferências e ainda assim manter as transações como unidades computacionais confiáveis é o controle do tempo. Esta ferramenta auxilia na manutenção das propriedades ACID mesmo frente às mudanças na estrutura das T_{MACS} .

As modificações na organização interna da transação devem ser gerenciadas, essa é outra função designada ao MiD-Mobile. Trabalhar em ambientes distintos exige maior controle de processamento, consolidação ou cancelamento das transações. O próximo capítulo apresenta em detalhes as estratégias adotadas para cada estrutura aplicada pelo modelo de transações móveis adaptadas ao contexto.

5 O MODELO DE TRANSAÇÕES MÓVEIS ADAPTADAS AO CONTEXTO

Os modelos transacionais encontrados na literatura não atendem completamente aos anseios desta pesquisa, na medida em que fazem concessões à violação das propriedades ACID ou não oferecem mecanismos de adaptação condizentes com o ambiente da computação móvel. Com o intuito de oferecer maior flexibilidade com o mínimo de transgressão às propriedades ACID, o presente trabalho propõe um modelo de transações móveis adaptáveis ao contexto.

O modelo de transações móveis adaptadas ao contexto agrega características peculiares aos modelos clássico, aninhado e multinível por meio da alteração de sua organização interna. Dentre os trabalhos pesquisados, nenhum prevê a mudança de estrutura da transação como mecanismo de adaptabilidade.

O modelo proposto é caracterizado pelo maior tempo de execução comparado ao modelo tradicional, já que no contexto da computação móvel, a conexão intermitente ou a insuficiência da largura de banda podem fazer uma transação de sistemas de computação móvel ter maior duração que o usual. Uma transação que perdura mais que o comum traz conseqüências às aplicações e às outras transações já que tende a tomar conta de recursos por mais tempo.

O modelo de transações aninhadas abertas prevê a liberação de itens de dados e resultados parciais mesmo antes do final da transação. Essas características beneficiam a propriedade do isolamento, aumentando a concorrência e a disponibilidade de acesso.

O modelo clássico parte de uma estrutura plana sem flexibilização, porém tal rigidez favorece a consistência dos dados disponíveis no dispositivo móvel e na rede fixa por meio da reexecução das transações ou da sincronização entre as réplicas dos dados.

O paralelismo intra-transação explorado no modelo de transação de múltiplos níveis agiliza a execução da transação nos casos em que os dados estão disponíveis em plataformas diferentes, ou seja, na rede fixa e no dispositivo móvel. A próxima seção apresenta detalhes da estrutura de uma T_{MAC} de acordo com o plano de execução indicado pelo SPE do MiD-Mobile.

5.1 A estrutura de uma T_{MAC}

A estrutura de uma T_{MAC} depende do seu plano de execução. Cada plano pode optar por uma dentre as três estratégias possíveis: execução isolada, remota ou compartilhada. No plano de execução isolada, a transação ocorre no dispositivo móvel. A opção remota prevê o envio do processamento para a rede fixa. A execução compartilhada distribui o processamento entre os extremos.

Uma T_{MAC} no plano de execução isolada ou no plano de execução remota possui estrutura plana. Para a opção compartilhada, a estrutura da transação passa a ser multicomponente. A Figura 5.1 está dividida em duas partes para apresentar uma T_{MAC} com estrutura plana (a) e a mesma T_{MAC} com estrutura multicomponentizada (b).

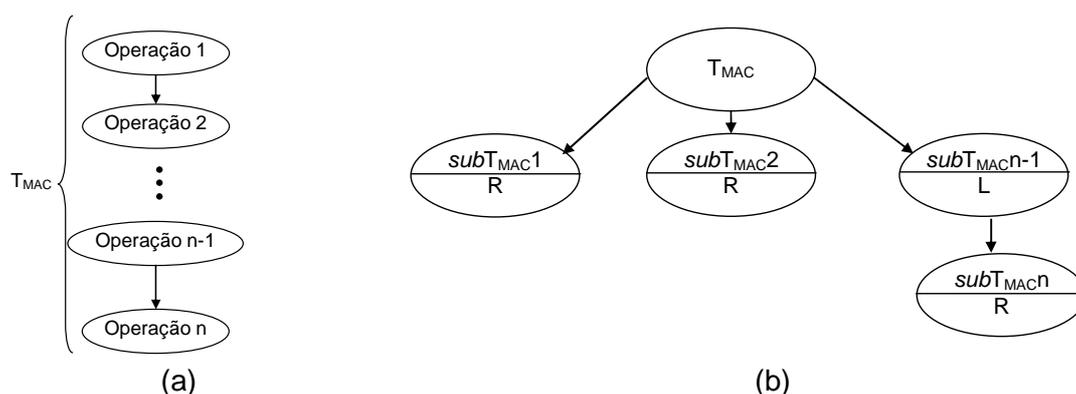


Figura 5.1 – Estruturas de uma T_{MAC}

Transações multicomponentes possuem componentes de computação e de acesso chamados de subtransações ou $subT_{MACs}$. Os componentes são concorrentes e possuem relação de dependência de execução entre eles.

Dos modelos abordados nos Capítulos 2 e 3, somente o AMT prevê processamento compartilhado a partir do modelo aninhado aberto. Transações aninhadas abertas permitem que subtransações tenham autonomia unilateral, exigindo estratégias da transação-raiz para tratar a queda da conexão sem fio.

O argumento utilizado para a proposta de transações aninhadas aplicadas ao ambiente móvel considera a transação passível de execução em horas ou dias. Nessa pesquisa, a subjetividade do tempo submete-se à semântica da aplicação e à arquitetura ou tecnologia da conectividade sem fio. O entendimento da falha de comunicação concebe desconexões voluntárias ou involuntárias. O acréscimo de elementos de suporte à mobilidade e à conexão permite o tratamento adequado das transações e suas possíveis conseqüências à integridade e confiabilidade dos dados.

Quando uma T_{MAC} é disparada, o MiD-Mobile é chamado para decidir qual o plano de execução mais adequado à configuração do momento. A Tabela 5.1 apresenta as três opções de plano de execução para uma transação.

Tabela 5.1 – Planos de Execução de uma T_{MAC}

Identificação da T_{MAC}	Plano de Execução	Plataforma de Execução	Estrutura da T_{MAC}	Componentes
$T_s(T_{MAC})$	ISOLADA	Dispositivo Móvel (Local)	Plana	<i>não se aplica</i>
$T_s(T_{MAC})$	REMOTA	Rede Fixa (Remoto)	Plana	<i>não se aplica</i>
$T_{sT_{MAC}}$	COMPARTILHADA	Dispositivo Móvel Rede Fixa (Local / Remoto)	Multicomponente	$subT_{MAC1}$ (Local) $subT_{MAC2}$ (Remoto)

As próximas seções apresentam detalhes de cada plano de execução e as mudanças estruturais que podem ser necessárias.

5.1.1 Plano de execução isolada de uma T_{MAC}

Quando o MiD-Mobile indica o plano de execução isolada, a transação é executada como no modelo clássico e a subdivisão não se aplica. A T_{MAC} plana é vista atomicamente e as propriedades ACID precisam ser garantidas. O processamento ocorre no mesmo local em que a transação foi disparada, ou seja, no próprio equipamento móvel.

Restrições de recursos podem inviabilizar a hospedagem de um banco de dados e o seu SGBD no dispositivo portátil, fazendo das ferramentas de persistência um mecanismo mais leve e, portanto, mais adequado para ser utilizado nesse contexto.

Essa permuta só é possível se o mecanismo de persistência oferecer suporte à unicidade de execução, isolamento de transações concorrentes, manutenção da consistência e durabilidade dos resultados. Em qualquer plano de execução, se as ferramentas usadas não oferecerem o apoio adequado, é necessário buscar outros meios para fazê-lo.

O plano de execução isolada viabiliza o trabalho autônomo através de operações desconectadas, já que falhas de comunicação são comuns diante da mobilidade. Durante a carga antecipada dos dados (*hoarding*) em períodos de estabilidade da conexão, réplicas de dados são disponibilizadas no equipamento móvel. Em seguida, a reintegração das alterações garante a consistência global. A Figura 5.2 modela os estados citados.

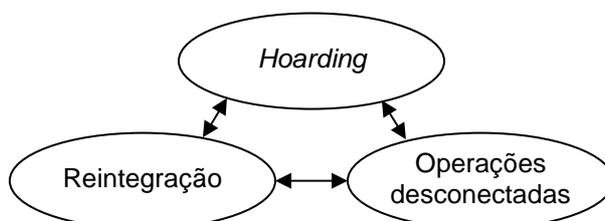


Figura 5.2 – Estados de operações desconectadas

Fonte: Pitoura e Chrysanthis (2000)

A consistência global é obtida pela sincronização dos dados alterados localmente com os mesmos dados disponíveis na rede fixa. Essa estratégia é acompanhada pelo MiD-Mobile disponível no cenário estacionário, através do Protocolo de Consistência Global (PCG).

No momento da execução do protocolo, o *timestamp* enviado inicialmente ao dispositivo móvel junto com os dados é comparado com o *timestamp* dos dados equivalentes na rede fixa. O resultado da comparação define as operações posteriores. A Figura 5.3, através de um diagrama de seqüência, exhibe a interação entre o *middleware* instalado no dispositivo móvel (DM) e na rede fixa (RF) durante o processo de validação do *timestamp* e sincronização dos dados.

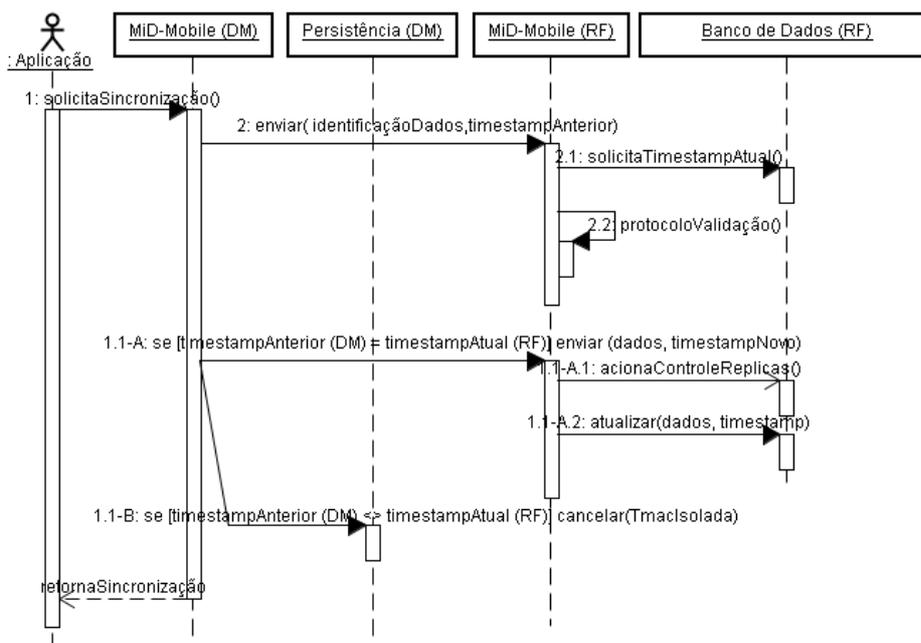


Figura 5.3 – Diagrama de seqüência do PCG

Se o resultado da comparação permitir, a sincronização terá início e os dados dos servidores da rede fixa passarão a receber os dados e o novo *timestamp* gerado no dispositivo móvel. Caso a impossibilidade de sincronismo dos dados ocasione o cancelamento da T_{MAC} , estratégias específicas são acionadas para minimizar os efeitos propagados.

Nesse último caso, um resultado indesejado impediu o funcionamento ideal da transação, impondo falhas na durabilidade da T_{MAC} e das transações posteriores.

Outra falha comum em ambientes distribuídos móveis que atinge a durabilidade acontece quando o estoque de bateria das estações nômades chega ao fim. Diante desse cenário os processamentos em execução ou armazenados em memória principal serão perdidos.

As falhas citadas impõem técnicas de recuperação por parte do SGBD ou do mecanismo de persistência, pelo controle de *log* das operações já consolidadas, através de algoritmos de refazimento (REDO), ou pela ocorrência do desfazimento (UNDO) das transações.

A replicação dos dados provê recuperabilidade no caso de falhas de mídia. Por restrição de recursos, uma vez que normalmente o dispositivo móvel não mantém cópias duplicadas dos itens de dados em cache, esse tipo de falha exige mecanismos de recuperação mais específicos. Novas cópias podem ser requisitadas quando o equipamento estiver operacional novamente.

A consistência global dos dados exige que no momento da sincronização o mecanismo de controle de réplicas seja acionado, o que permite igualdade de valor às cópias. Um exemplo teórico desse mecanismo, citado no Capítulo 2, foi o ROWA (*read-once write-all* - ler uma vez e gravar tudo).

Existem outros algoritmos que permitem replicação somente aos sites alcançados e outros que mantêm a atualização inicialmente sobre uma cópia mestre e depois fazem a propagação das alterações às demais réplicas.

5.1.2 Plano de execução remota de uma T_{MAC}

Quando o plano de execução indicado for o remoto, a T_{MAC} é vista atômicamente como no plano anterior, mas a transação é encaminhada para a rede fixa onde é executada. Nesse cenário, um computador fixo será o representante da T_{MAC} no ambiente estacionário. Sua execução remota a condiciona às estratégias de gerenciamento empregadas nos servidores da rede fixa que abrigam um banco de dados distribuído.

Transações em ambientes distribuídos estão sujeitas às técnicas citadas no Capítulo 2 para tratar a atomicidade, consistência, isolamento e durabilidade. O efeito do tudo ou nada é garantido pelo algoritmo de consolidação em duas fases (2PC), aliado ao mecanismo que mantém a consistência mútua entre as réplicas dos itens de dados.

Outra característica empregada nos servidores da rede fixa que atinge as T_{MACs} remotas é a presença da concorrência entre transações. A execução em série das transações submetidas ao SGBDD ocasiona lentidão no processamento. Assim, escalas de execução não seriais entremeiam o acesso aos itens de dados, aumentando o desempenho. O isolamento nesse tipo de execução é de vital importância para a garantia da consistência dos dados.

Nem todas as escalas de execução não seriais produzem resultados corretos. Assim, escalonamentos serializáveis são escalas de execução intercalada das operações de transações concorrentes equivalentes ao resultado da execução serial das mesmas. No Apêndice A constam exemplos de escalas inválidas.

A equivalência de escalonamentos serializáveis e planos seriais garantem a ordem das operações aplicadas aos itens de dados. Elmasri e Navathe (2005) citam duas técnicas de avaliação da equivalência, a baseada em conflito e a baseada em visão. O Apêndice A apresenta detalhes dessas técnicas.

O mecanismo de controle de concorrência é responsável pela garantia da serialização das escalas geradas por transações concorrentes. Os esquemas mais explorados são os de bloqueios dos dados, ordenação por etiquetas de tempo (*timestamp*) e validação, conforme Özsu e Valduriez (2001).

Em escalas de execução, operações que partem de transações diferentes envolvendo a escrita em um mesmo item de dados, podem conflitar. A Tabela 5.2 mostra quando as operações de escrita e leitura em um sistema de banco de dados podem entrar em conflito.

Tabela 5.2 – Conflito entre operações

	Leitura	Escrita
Leitura	não conflitam	conflitam
Escrita	conflitam	conflitam

Fonte: Elmasri e Navathe (2005)

Algumas anomalias são decorrentes de operações conflitantes, como a leitura suja, não repetível e perda de atualização.

Qualquer esquema utilizado para assegurar o isolamento precisa tratar as anomalias apresentadas. A técnica mais utilizada nos sistemas de banco de dados comerciais é a do bloqueio. Nesta pesquisa é utilizada uma técnica híbrida que envolve bloqueio e controle de tempo.

Bloqueios se dividem em compartilhado ou exclusivo. O tipo compartilhado libera a leitura do dado, enquanto o exclusivo permite leitura e escrita. A transação só pode realizar suas operações mediante a concessão do bloqueio. A incompatibilidade entre modos de bloqueios correspondem aos conflitos entre operações das mesmas transações. A Tabela 5.3 apresenta a relação de compatibilidade entre os bloqueios.

Tabela 5.3 – Matriz de compatibilidade de bloqueios

	Compartilhado	Exclusivo
Compartilhado	Verdadeiro	Falso
Exclusivo	Falso	Falso

Fonte: Elmasri e Navathe (2005)

Dentre os algoritmos de bloqueio existentes um dos que mais se destaca é o 2PL (*two-phase locking*) rigoroso, citado no Capítulo 2. No 2PL nenhuma transação solicita novos bloqueios após liberar qualquer um dos bloqueios já adquiridos. Um escalonamento gerado por um algoritmo de controle de concorrência que obedeça ao 2PL é dito serializável.

A ocorrência da T_{MAC} no plano de execução remota é amparada por todos os elementos citados nesta subseção, mas a falta de comunicação com o equipamento portátil, quer seja por falha na conectividade ou omissão do dispositivo móvel, impede que os resultados da T_{MAC} sejam encaminhados, apesar de não ocasionar o seu cancelamento. Quando do restabelecimento do contato, a informação final será direcionada.

5.1.3 Plano de execução compartilhada de uma T_{MAC}

Diante do plano de execução compartilhada, a estrutura multicomponente é aplicada e cada $subT_{MAC}$ será requisitada em uma plataforma de execução específica, informada previamente durante o desenvolvimento. Nesse contexto, as técnicas que garantem a integridade e a confiabilidade dos dados diferem do modelo tradicional. Elementos adicionais foram agregados às propriedades ACID para assegurar às T_{MACs} a denominação de unidades de computação confiáveis.

5.1.3.1 Garantia das propriedades ACID para as T_{MACs} multicomponentes

Durante a ocorrência de uma T_{MAC} multicomponente, uma falha no canal de comunicação pode gerar impasses, inconsistências e perda de processamento. Um elemento importante, o tempo de execução global (Δt), foi adicionado à transação para promover tratamento adequado diante de ocorrências como essas.

A composição do tempo de execução prevê também um retardo (*delay*) no início da transação. Cada domínio de negócio possui um Δt adequado para suas T_{MACs} , por isso esse elemento é configurável.

A espera indefinida por recursos ou respostas diminui a concorrência e desfavorece a unicidade de execução. O princípio da atomicidade multicomponente é garantido pelo acréscimo do tempo de execução (Δt) ao protocolo de consolidação em duas fases, o 2PC, dando origem ao $2PC\Delta t$.

Quando o MiD-Mobile, responsável pelo monitoramento do tempo, intercepta uma T_{MAC} multicomponente, ele inicializa o serviço responsável por essa função nas duas plataformas. O *timestamp* e o Δt da T_{MAC} são usados pelo *middleware* para evitar diferenças entre os inícios dos monitoramentos.

Ao término da transação ou do tempo de execução, as camadas distribuídas do MiD-Mobile verificam a situação parcial das $subT_{MACs}$. Se os sites participantes concordarem e a conexão permitir, o *middleware* da rede fixa recebe os resultados das subtransações executadas no dispositivo móvel e decide pela consolidação ou pelo cancelamento de toda a transação. O diagrama de seqüência da Figura 5.4 exhibe as atividades do $2PC\Delta t$.

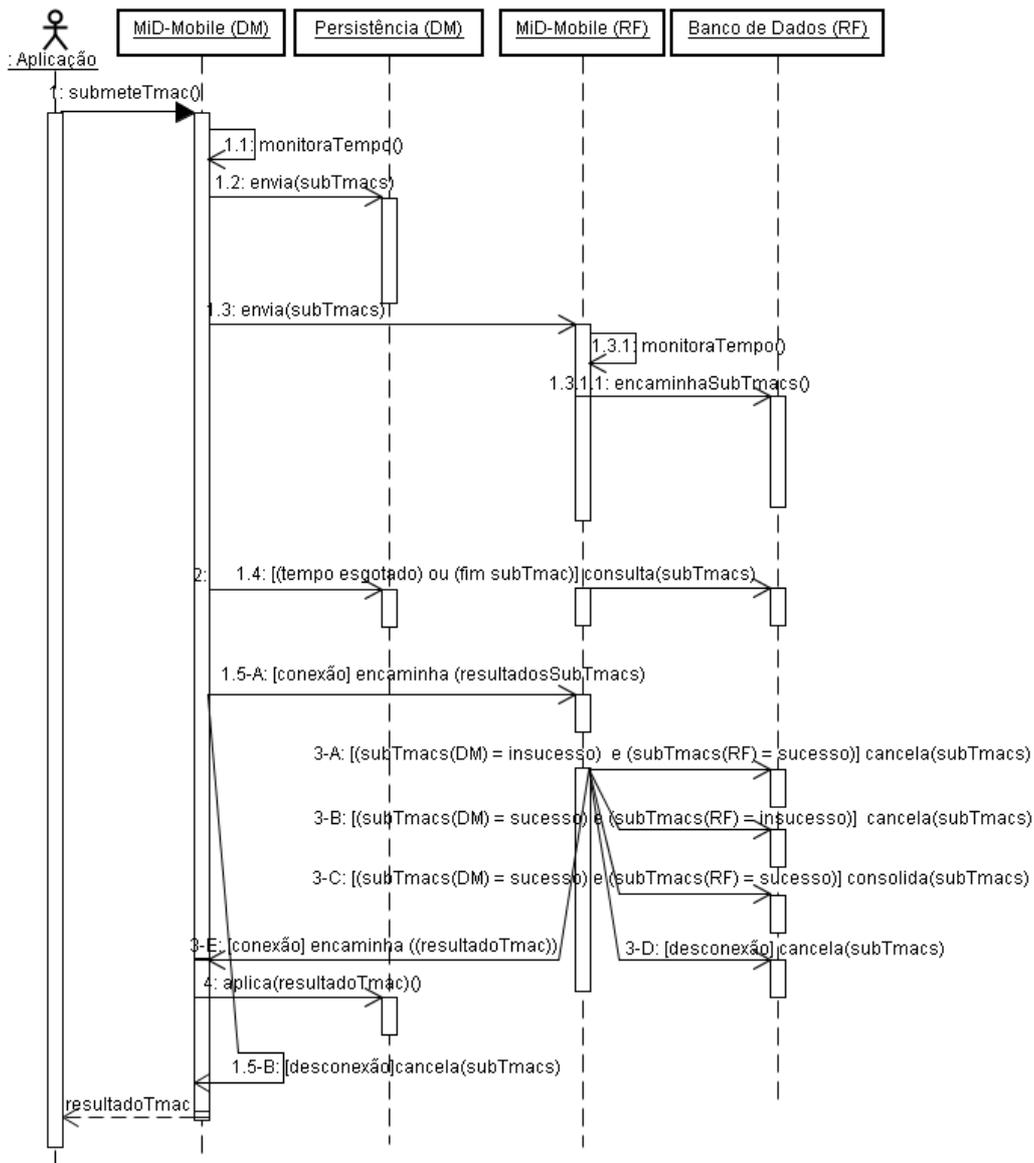


Figura 5.4 – Diagrama de seqüência do protocolo 2PC Δt

A principal diferença do protocolo 2PC, em relação ao 2PC Δt diz respeito à duplicidade de opção para a inicialização do algoritmo, ou seja, no modo tradicional com o fim da T_{MAC} ou diante do término do prazo de execução (Δt). No 2PC Δt , o site coordenador é o MiD-Mobile residente na rede fixa. O recebimento dos resultados das $subT_{MACs}$ do dispositivo móvel é o sinal de "preparar" para o início da consolidação.

Diante do resultado encaminhado é que a decisão de cancelar ou consolidar é determinada e a mesma ação é aplicada aos dois extremos. As condições para o sucesso da T_{MAC} multicomponente são os resultados parciais positivos de todas as $subT_{MACs}$, respeitado o tempo de execução (Δt). A falha pode ocorrer devido ao esgotamento do Δt , ao insucesso de qualquer $subT_{MAC}$ ou a queda da conexão.

A indisponibilidade da conexão pode coincidir com o final da T_{MAC} e a inicialização do $2PC\Delta t$, por isso o canal de comunicação é monitorado no restante do tempo e as $subT_{MACs}$ são mantidas para o caso da conexão ser restabelecida. A falha também pode acontecer em outras duas ocasiões, quando os resultados das $subT_{MACs}$ do dispositivo móvel são encaminhados para a rede fixa, ou após a decisão de consolidar ou abortar do MiD-Mobile.

O cancelamento de subtransações baseado na monitorização do tempo é providencial na ocorrência de *deadlocks*, na presença de subtransações órfãs e diante de falha de site dos dispositivos móveis ou dos servidores da rede fixa.

A execução distribuída dificulta a manutenção do ponto em que a T_{MAC} precisaria ser reiniciada, além do mais, mesmo que a recuperação da falha no site ocorra dentro do período de processamento da T_{MAC} , o tempo acabaria sendo reiniciado em uma das plataformas, permitindo indesejáveis discordâncias entre os prazos.

Nesse sentido, a utilização de apenas um ponto de término garante a unicidade da transação. Se o modelo da T_{MAC} no plano compartilhado fosse aninhado, subtransações já confirmadas poderiam não sofrer com a falha, mas posteriormente seriam desfeitas. E a execução de outras transações sobre o mesmo item de dados também teriam que ser canceladas. Isso tudo não ocorre com a utilização da estrutura multicomponentizada.

O isolamento no caso do plano de execução compartilhada deve ser garantido nos dois ambientes. Aplicações monousuárias sem suporte à execução concorrente e que aplicam processamento serial dispensam esse controle. Do contrário, a aplicação de mecanismos de controle de concorrência assegura a consistência dos dados manipulados por várias transações.

As $subT_{MACs}$ remotas concorrem pelos recursos junto com as transações oriundas da própria rede fixa, como também com subtransações de outras T_{MACs} , sendo submetidas a escalas de execução que permitam paralelismo. No dispositivo móvel a concorrência também depende de fatores como as características do sistema operacional nativo e a natureza da aplicação.

A Figura 5.5 (a) mostra componentes de processamento (sombreados) e de acesso se entrelaçando para compor uma T_{MAC} multicomponente. Para o escalonador apenas as subtransações de acesso interessam, assim os componentes avaliados são os apresentados na Figura 5.5 (b).

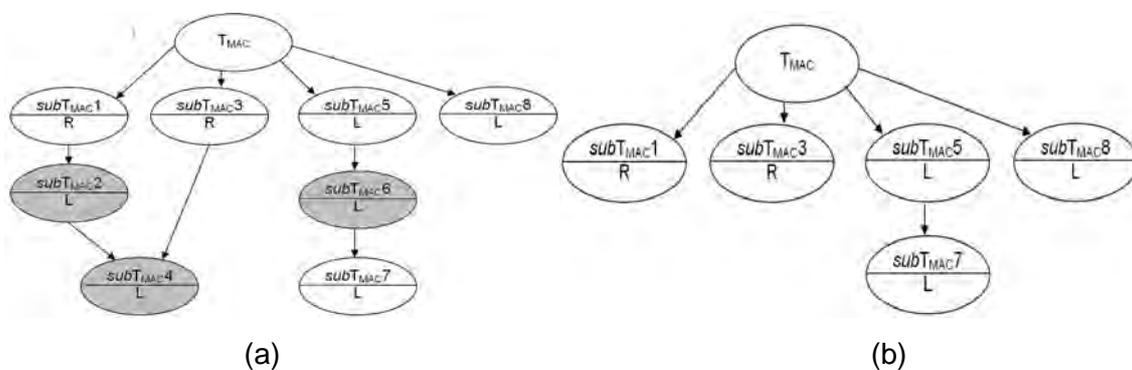


Figura 5.5 – Componentes de uma T_{MAC}

Transações multicomponentes são seriais quando cada componente é executado na ordem em que aparece em sua estrutura. A serialização de um escalonamento de T_{MACs} multicomponentes é testada pela equivalência com a execução serial a cada camada. Na Figura 5.5 (b), requer-se que as $subT_{MACs}$ 1, 3, 5 e 8 sejam seriáveis, pois estão no mesmo nível.

A serialização das $subT_{MACs}$ esbarra no conflito de operação entre as mesmas. A noção de conflito entre operações é ligeiramente diferente do tradicional. Duas subtransações conflitam se operam no mesmo site, sobre o mesmo item de dado, sendo uma delas operação de escrita, além de não serem ancestrais uma da outra.

No exemplo da Figura 5.5 (b), a $subT_{MAC}$ 1 poderia conflitar com a $subT_{MAC}$ 3, caso operasse sobre o mesmo item e uma delas fosse de escrita, já que uma não é ancestral da outra e executarão o processamento no mesmo site (R). No caso das $subT_{MACs}$ 5 e 7 não há conflito, visto que existe uma relação de dependência entre as mesmas, uma vez que qualquer que seja o tipo de bloqueio adquirido pela $subT_{MAC}$ 5, a outra subtransação vai herdar o bloqueio seja através de promoção ou de rebaixamento.

O método usado para alcançar serialização de transações multicomponentes é o controle de concorrência por protocolo de bloqueio bifásico rigoroso aliado ao Δt , denominado nesse trabalho de $2PL\Delta t$. A Figura 5.6 ilustra o efeito da adaptação.

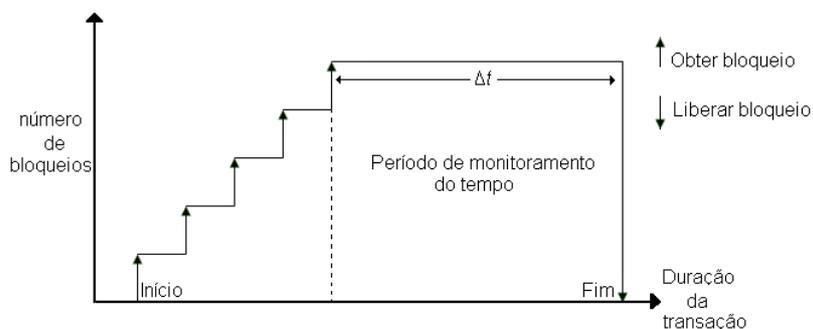


Figura 5.6 – Protocolo de bloqueio em duas fases temporário

Essa técnica inicialmente tratada em Mohan et al. (1986), permite que o bloqueio permaneça em duas situações, até que o tempo predeterminado se esgote ou ao término da transação. As seguintes regras são impostas para verificar a corretude dos resultados produzidos pelas transações multicomponentes:

- a) uma subtransação de T_{MAC} só opera após obter o bloqueio dos itens de dados;
- b) bloqueios internos em conflito são arbitrados pelo tempo particular de execução (Δst);
- c) os bloqueios mantidos por Δst são liberados para outras subtransações da mesma T_{MAC} em decorrência da dependência de execução e de bloqueio entre as mesmas. Dessa forma a serialização interna é mantida;
- d) diante de bloqueios externos em conflito, uma $subT_{MAC}$ concorre com outras transações ou $subT_{MACs}$ pela obtenção do bloqueio, mantendo a serialização externa;
- e) a ocorrência de bloqueios conflitantes, tanto interna ou externamente, ocasionam a espera da $subT_{MAC}$ pela liberação do bloqueio. O *delay* inserido no Δt representa a espera pela liberação dos recursos;
- f) T_{MAC} ativas são canceladas ao fim do tempo de execução designado (Δt);
- g) os bloqueios são liberados em casos de término, falha ou expiração do tempo da T_{MAC} ;
- h) uma T_{MAC} só se confirma pela execução do $2PC\Delta t$.

A regra (a) garante que as subtransações da T_{MAC} só ocorrerão a partir do bloqueio dos itens de dados, impedindo a violação da consistência. Itens já bloqueados podem causar o retardo no início das subtransações e por isso, um *delay* (atraso) é previsto no Δt .

Os bloqueios são divididos em internos e externos. Bloqueios internos ocorrem pelas solicitações das $subT_{MACs}$, enquanto os externos se concretizam diante das solicitações de transações diferentes.

Conflitos em bloqueios internos, conforme regra (b), são controlados pelo Δt que é obtido pelo resultado da divisão do tempo total da T_{MAC} (Δt) pela somatória da quantidade de todas as $subT_{MACs}$ que operam sobre o mesmo item de dados no mesmo site. A aplicação do Δt garante a serialização interna, conforme regra (c), já que os bloqueios são oferecidos para outra $subT_{MAC}$ da transação móvel (dependência de execução e de bloqueio).

As condições de Bernstein ditam as regras de dependência entre as $subT_{MACs}$ que implicam na manutenção do bloqueio durante o momento de execução e transição entre subtransações dependentes.

Bloqueios externos, de acordo com a regra (d), são passíveis de disputa por um item de dados com qualquer transação ou subtransação de outra T_{MAC} vigente, o que garante a serialização entre as transações da rede fixa. A ocorrência de bloqueios conflitantes ocasiona a espera pela liberação do recurso solicitado, conforme regra (e).

Caso a espera exceda o *delay* previsto no Δt , o sucesso da T_{MAC} pode ficar comprometido. Dentro da validade total da transação, ainda é permitido que as $subT_{MACs}$ sejam processadas. Caso ao término do Δt ainda se mantenham ativas, elas serão canceladas de acordo com a regra (f). Sobre a liberação dos bloqueios, a regra (g) esclarece que este pode ocorrer ao término (consolidação ou cancelamento) da T_{MAC} multicomponente, na ocorrência de falhas ou, devido ao término do tempo que controla os bloqueios. A última opção é imprescindível para evitar que subtransações órfãs mantenham bloqueios indefinidamente. A regra (h) garante que a atualização permanente dos dados só aconteça após a execução do $2PC\Delta t$.

5.2 Considerações

O modelo de transações móveis adaptadas ao contexto sugere a mudança na estrutura da transação de acordo com o plano de execução que oferece maior chance de êxito à T_{MAC} . Assim, as duas opções possíveis, plana e multicomponente, foram avaliadas quanto à corretitude dos resultados.

Independente da organização interna, as propriedades ACID foram garantidas, ou seja, a T_{MAC} é totalmente executada ou nada é confirmado e todas as transações perduram mesmo diante de falhas. As operações invocadas mantêm a consistência do banco de dados e os estados intermediários entre transações concorrentes não afetam outras transações.

Para tanto, alguns mecanismos e algoritmos foram adotados e acrescentados (elementos sombreados). A Tabela 5.4 associa todas as estratégias empregadas com a manutenção das propriedades ACID.

Tabela 5.4 – Estratégias para a manutenção das propriedades ACID das T_{MACs}

Propriedades	T_{MAC} plana		T_{MAC} multicomponente
	Plano isolado	Plano remoto	
Atomicidade	Um ponto de término	Um ponto de término 2PC	Um ponto de término $2PC\Delta t (2PC + \Delta t)$ <i>timestamp</i>
Consistência	Cancelamento Controle de réplicas PCG Sincronização <i>Timestamp</i>	Cancelamento Controle de réplicas Escalonamentos serializáveis	Cancelamento Controle de réplicas PCG Sincronização <i>Timestamp</i>
Isolamento	Depende do Sistema Operacional e da natureza da aplicação	2PL	$2PL\Delta t (2PL + \Delta t)$ Δst Plataformas de execução diferentes
Durabilidade	<i>Log (REDO)</i> Replicação / <i>Hoarding</i>	Consolidação Replicação <i>Transmissão</i>	Consolidação Replicação

O plano de execução remota exige apenas a transmissão entre as plataformas de processamento e é ideal para usuários com dispositivos de recursos modestos, que enviam requisições e recebem resultados. O plano de execução isolada é interessante para perfis de usuários que perdem conexão freqüentemente, mas que necessitam operar sobre os dados mesmo diante da desconexão.

O plano de execução compartilhada da T_{MAC} é o que exigiu o maior número de estratégias diferenciadas, sendo que duas características importantes serviram de base para tal idealização. A primeira foi o compartilhamento da computação proposta por Serrano-Alvarado (2004). A segunda foi a utilização do tempo como forma de indisponibilização temporária dos dados, tratada em Rocha e Toledo (2007).

Apesar de basear-se nestas referências, dois aspectos destes trabalhos diferenciam-se da presente pesquisa: a forma de tratamento das propriedades ACID e a organização interna no modelo AMT, tratado em Serrano-Alvarado (2004); e a quantidade de dados transmitidos pelo canal de comunicação sem fio previsto no modo de operação LR, visto em Rocha e Toledo (2007).

A estrutura da transação no modelo AMT é aninhada aberta e relaxa a atomicidade e o isolamento, admitindo vários pontos de término com liberação de resultados parciais. Devido à forte influência da desconexão, a consistência global também foi maleabilizada, já que os mecanismos de recuperação têm mais trabalho para manter a consistência diante das falhas.

As propriedades ACID no modelo proposto, em qualquer plano de execução, são sempre asseguradas no seu formato mais estrito. A unicidade de ponto de término garante a atomicidade da T_{MAC} mesmo na estrutura multicomponente. A desconexão influencia na consistência dos dados, por isso o protocolo de validação é acionado para permitir apenas operações sobre dados

consistentes. O isolamento foi garantido pelas escalas serializáveis e a durabilidade é assistida pela replicação e técnicas de recuperação.

O modo de operação LR, apresentado por Rocha e Toledo (2007), leva dados indisponibilizados na rede fixa até a cache do equipamento portátil para o processamento, retornando em seguida. Essa estratégia é válida quando o canal de comunicação está bom e a taxa de transferência está alta. Qualquer instabilidade no tráfego de dados pode ocasionar eventual corrompimento de pacotes e excesso de tempo na transmissão.

Aumentar o *throughput* num cenário delicado é uma decisão que compromete a confiabilidade dos dados transmitidos ou acarreta retransmissão, ocasionando lentidão. O *delay* (atraso) no tempo de transmissão pode incidir sobre o prazo em que a transação mantém os dados indisponibilizados. Isto ocorre porque o término da transmissão pode coincidir com o fim do bloqueio dos dados e, enquanto a transação no modo LR opera localmente, os dados podem ser alterados na rede fixa, ocasionando inconsistência.

No plano de execução compartilhada, o canal de comunicação é usado para troca de mensagens, o que admite uma rede mais instável. A transmissão dos dados ocorre apenas uma vez no sentido dispositivo móvel → rede fixa, e compreende somente os dados que foram utilizados no processamento local e não os usados pela transação inteira. O plano se mostra adequado a cenários de escassez de recursos, indisponibilidade de dados e baixas taxas de transmissão.

O paralelismo interno e a distribuição da computação entre os extremos pode proporcionar a diminuição da sobrecarga nos servidores fixos, minimização da transmissão de dados, agilização das transações de maior duração. A principal contribuição do modelo de transações móveis adaptadas ao contexto é a possibilidade de maximização das chances de êxito da transação pela

aplicação dinâmica do plano de execução mais adequado às configurações do momento. O próximo capítulo apresenta um estudo de caso que esclarece a interação entre os sistemas de computação móveis, as T_{MACs} e o MiD-Mobile.

6 ESTUDO DE CASO

Todos os esforços destinados a pesquisas em computação móvel como infraestrutura, formas de interação, ferramentas, entre outras questões, são justificados pela variedade de modelos de negócio que podem usufruir de tais avanços. O foco dessa pesquisa diz respeito aos sistemas de computação móvel que acessam dados compartilhados, nesse contexto encontram-se aplicações como, por exemplo, agendamento, força de vendas, serviços, movimentações bancárias entre outros.

Para validar o modelo e o *middleware* proposto, foi desenvolvido um sistema para execução de operações financeiras denominado BancoOnline acessível também via computador portátil, por meio da aplicação BancoME. A aplicação foi desenvolvida em Java e usou várias técnicas e tecnologias como JSP, Servlets, JME, Web Services, JPA, Hibernate, Annotations e Generics. A Figura 6.1 apresenta o diagrama de casos de uso das funcionalidades previstas na aplicação.

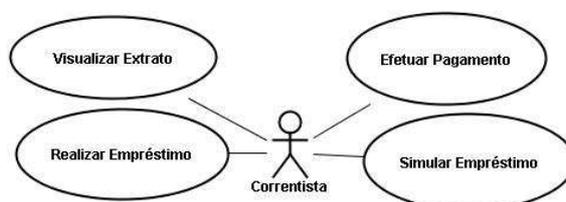


Figura 6.1 – Diagrama de Casos de Uso das funcionalidades do BancoOnline e BancoME

As próximas seções apresentam detalhes das aplicações desenvolvidas para o estudo de caso.

6.1 BancoOnline

O BancoOnline disponibiliza as funcionalidades previstas na Figura 6.1. As Figuras 6.2, 6.3, 6.4, 6.5, 6.6, 6.7 e 6.8 apresentam as interfaces dessas funcionalidades acessíveis a partir da rede fixa.



Figura 6.2 – Interface principal do BancoOnline

As setas que constam da Figura 6.2 indicam as operações possíveis. Cada operação será tratada pelo modelo de transações móveis adaptadas ao contexto em todos os planos de execução. A primeira transação diz respeito ao pagamento de contas, por razões de simplificação apenas descrição e valor são solicitados aos usuários. A Figura 6.3 apresenta a execução da operação.



Figura 6.3 – Interface pagamento

O exemplo mais simples é a consulta ao extrato da conta corrente. A transação que apresenta o extrato, busca pelos últimos lançamentos e o saldo atual, a Figura 6.4 ilustra a ocorrência da mesma.



Extrato

Conta AGENCIA 044 CONTA 12345-6

Limite R\$ 1.000,00

Disponível R\$ 2.526,09

Lançamentos Efetivados

19/10/2008	Saldo Anterior	R\$ 200,00
20/10/2008	Deposito	R\$ 160,00
23/10/2008	Saque	-R\$ 10,00
26/10/2008	Deposito	R\$ 220,00
29/10/2008	Deposito	R\$ 50,00
01/11/2008	Deposito	R\$ 280,00
03/11/2008	Emprestimo	R\$ 1.000,00
03/11/2008	Prestacao Emprestimo 1 de 6	-R\$ 173,91
03/11/2008	Pagamento	-R\$ 100,00
03/11/2008	Pagamento Água	-R\$ 100,00
	Saldo Atual	R\$ 1.526,09

Lançamentos Programados e não efetivados

03/01/2009	Prestacao Emprestimo 2 de 6	-R\$ 173,91	Efetivar
03/02/2009	Prestacao Emprestimo 3 de 6	-R\$ 173,91	Efetivar
03/03/2009	Prestacao Emprestimo 4 de 6	-R\$ 173,91	Efetivar
03/04/2009	Prestacao Emprestimo 5 de 6	-R\$ 173,91	Efetivar
03/05/2009	Prestacao Emprestimo 6 de 6	-R\$ 173,91	Efetivar

Figura 6.4 – Interface extrato de conta corrente

O exemplo seguinte trata duas situações, a simulação de um empréstimo e a efetivação do mesmo. Na Figura 6.5, as informações circuladas apresentam o valor a ser emprestado e a quantidade de parcelas. Os dados são usados para o cálculo do valor das parcelas.

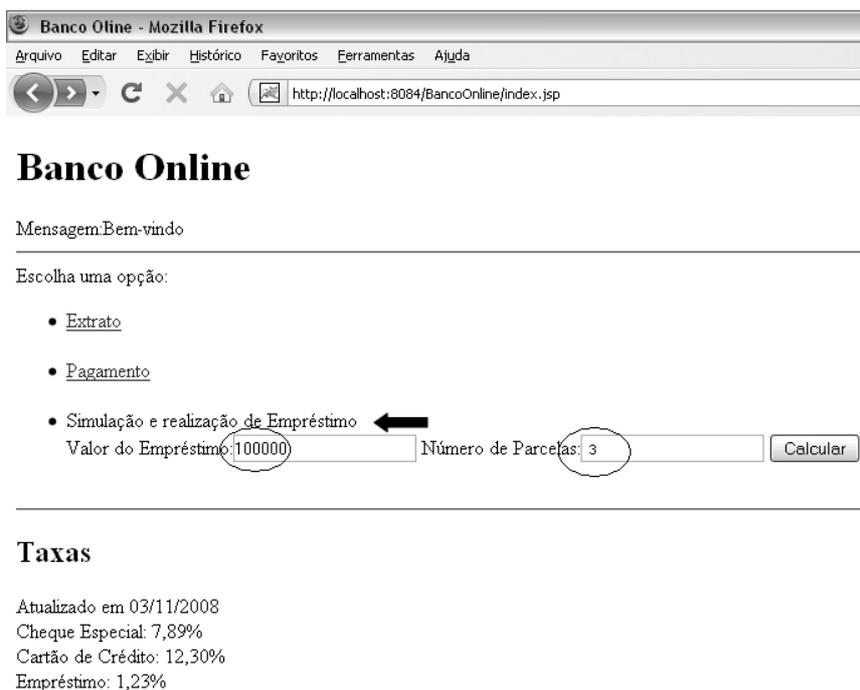


Figura 6.5 – Operações de simulação e empréstimo

A Figura 6.6 apresenta os valores das parcelas e o total final do empréstimo considerando os juros incididos. Além de apresentar o número de parcelas solicitadas o BancoOnline oferece outras opções ao correntista.

Simulação para R\$ 100.000,00
Empréstimo: 1,23%

Parcelas	Valor	Total	Operação
3	R\$ 34.156,67	R\$ 102.470,02	Realizar
6	R\$ 17.391,48	R\$ 104.348,85	Realizar
12	R\$ 9.014,51	R\$ 108.174,12	Realizar
18	R\$ 6.227,19	R\$ 112.089,41	Realizar
24	R\$ 4.837,27	R\$ 116.094,48	Realizar
30	R\$ 4.006,30	R\$ 120.188,98	Realizar
36	R\$ 3.454,79	R\$ 124.372,49	Realizar

[Voltar](#)

Figura 6.6 – Interface de parcelamentos disponíveis para efetivação de empréstimo

A Figura 6.7 apresenta a verificação do valor das parcelas comparado ao valor da prestação que o cliente pode assumir. Nessa situação, a mensagem emitida diz que o valor da parcela supera o limite, impedindo a realização do empréstimo.

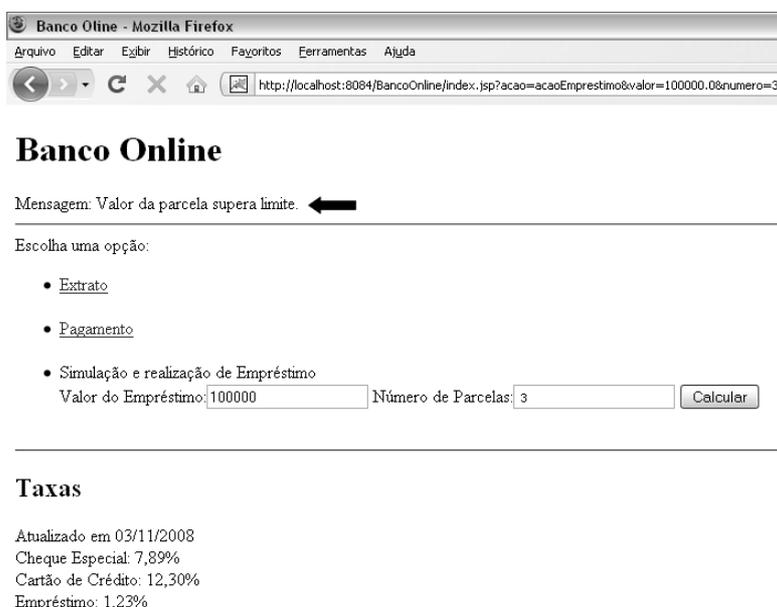


Figura 6.7 – Interface de avaliação de empréstimo

A Figura 6.8 ilustra a situação em que o empréstimo foi aprovado e efetivado. As linhas circuladas apresentam o lançamento do valor emprestado e a efetivação da primeira parcela, através da operação de extrato.

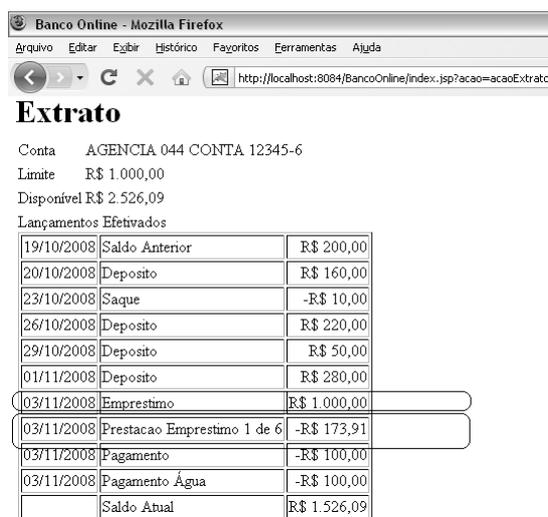


Figura 6.8 – Interface lançamento de empréstimo em conta corrente

O diagrama de classe da aplicação BancoOnline é constituído de três classes apresentadas na Figura 6.9. A classe "Conta" possui detalhes da conta, o atributo saldo é amplamente utilizado em todas as transações que o exemplo disponibiliza. A conta "Lançamento" armazena todos os lançamentos efetivados a partir das transações como, por exemplo, o empréstimo realizado na Figura 6.8.

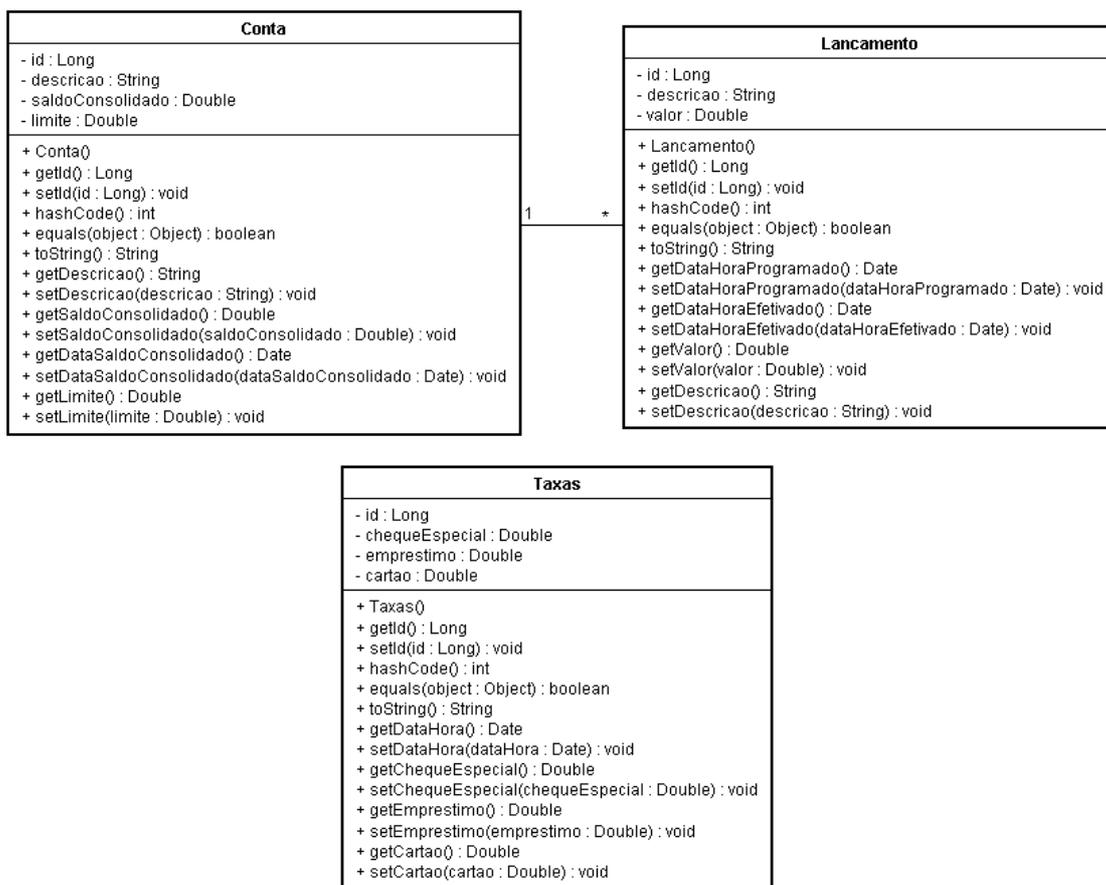


Figura 6.9 – Diagrama de classes do BancoOnline

A classe "Taxas" apresenta os percentuais de juros usados para a execução da transação de simulação de empréstimo e o próprio empréstimo. A implementação do BancoOnline contempla o pacote da visão (páginas JSP), controle, entidades, negócios e persistência. A Figura 6.10 oferece uma visão geral do empacotamento da aplicação a partir da janela projetos do NetBeans.



Figura 6.10 – Estrutura do BancoOnline

A Figura 6.11 exibe os elementos que constituem a aplicação e a interação necessária para a execução das transações a partir do encapsulamento ilustrado na Figura 6.10.

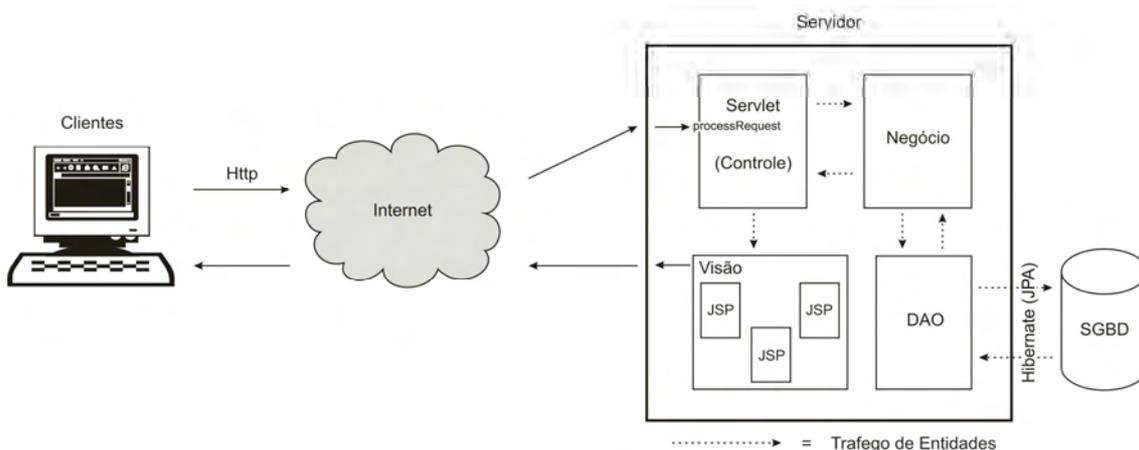


Figura 6.11 – Visão geral dos elementos usados na estrutura do BancoOnline

Fonte: Site da Sun Microsystems

As páginas Web, desenvolvidas em JSP, compõem a camada da Visão e é o mecanismo de interação entre o correntista e a aplicação. O pacote de controle possui um servlet chamado ControlaPrincipal que é responsável por acionar as ações solicitadas pelo correntista. O pacote entidade possui os POJOS (*Plain Old Java Object* – classes Java simples) que permeiam os demais pacotes e são representados na Figura 6.11 pelas linhas tracejadas.

O pacote de Negócio implementa as interfaces e os métodos necessários para executar as operações. O último encapsulamento engloba as classes DAO, DAOException e EntityManagerFactory. A classe DAO utiliza o conceito de Genéricos (Generics) para passar entidades como parâmetros para a tarefa de persistência. O EntityManagerFactory executa tal tarefa a partir de uma única instanciação capaz de persistir todos os objetos repassados pelo DAO.

A utilização da Java Persistência API (JPA) mapeada pelo Hibernate (*framework* de mapeamento objeto relacional) se beneficia da estrutura do tipo POJO. As anotações (*annotation*) como o "@Entity" da Figura 6.12 são necessárias para oferecer metadados a respeito de entidades, relacionamentos, tipos, entre outros.

```
@Entity
public class Conta implements Serializable {
    private static final long serialVersionUID = 1L;
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;

    private String descricao;

    private Double saldoConsolidado;
```

Figura 6.12 – POJO com *annotation*

6.2 BancoME

Essa aplicação, feita em JavaME, é usada para acessar via dispositivo móvel o BancoOnline disponível no servidor da rede fixa. Os testes foram realizados com o Wireless Toolkit (WTK), ferramenta usada para executar a aplicação a partir de um emulador, disponibilizada gratuitamente no site da Sun Microsystems. Quando necessário o BancoME acessa as funcionalidades do BancoOnline via Web Services (serviços web), por meio da classe BancoCompartilhado disponível em BancoME. A Figura 6.13 apresenta o menu de opções de BancoME.



Figura 6.13 – Interface do BancoMe no emulador WTK

Figura 6.14 exibe as interfaces de execução de cada item disponível.

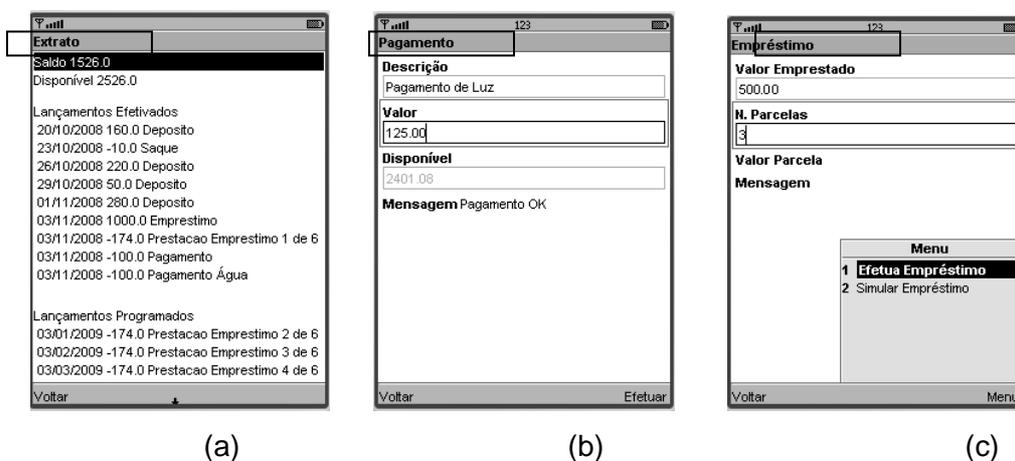


Figura 6.14 – Execução das operações disponíveis do BancoMe

A Figura 6.14 (a) explora a execução do extrato, a Figura 6.14 (b) focaliza o pagamento e a Figura 6.14 (c) o menu de opções para uma transação de simulação/empréstimo. As figuras 6.15 (a) e 6.15 (b) exibem respectivamente a simulação do empréstimo e a efetivação do mesmo. Um novo extrato foi solicitado para ressaltar a execução do pagamento e do empréstimo, interface da Figura 6.15 (c).

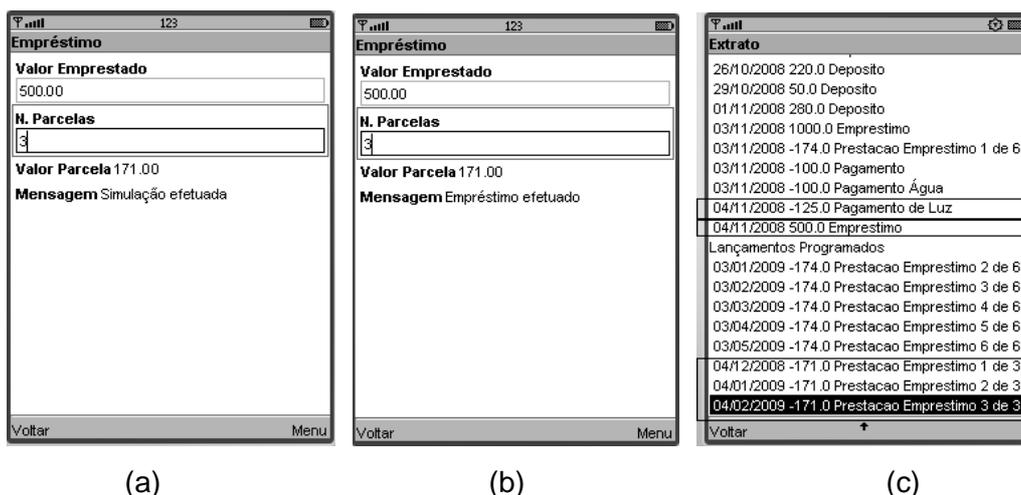


Figura 6.15 – Resultado da execução das operações do BancoMe

Outro extrato solicitado diretamente no servidor do BancoOnline, Figura 6.16, reforça que as operações efetuadas a partir do dispositivo móvel estão sendo realizadas na rede fixa.

Banco Online - Mozilla Firefox			
Extrato			
Conta AGENCIA 044 CONTA 12345-6			
Limite R\$ 1.000,00			
Disponível R\$ 2.901,09			
Lançamentos Efetuados			
19/10/2008	Saldo Anterior	R\$ 200,00	
20/10/2008	Deposito	R\$ 160,00	
23/10/2008	Saque	-R\$ 10,00	
26/10/2008	Deposito	R\$ 220,00	
29/10/2008	Deposito	R\$ 50,00	
01/11/2008	Deposito	R\$ 280,00	
03/11/2008	Emprestimo	R\$ 1.000,00	
03/11/2008	Prestacao Empréstimo 1 de 6	-R\$ 173,91	
03/11/2008	Pagamento	-R\$ 100,00	
03/11/2008	Pagamento Água	-R\$ 100,00	
04/11/2008	Pagamento de Luz	-R\$ 125,00	
04/11/2008	Emprestimo	R\$ 500,00	
	Saldo Atual	R\$ 1.901,09	
Lançamentos Programados e não efetuados			
03/01/2009	Prestacao Empréstimo 2 de 6	-R\$ 173,91	Efetivar
03/02/2009	Prestacao Empréstimo 3 de 6	-R\$ 173,91	Efetivar
03/03/2009	Prestacao Empréstimo 4 de 6	-R\$ 173,91	Efetivar
03/04/2009	Prestacao Empréstimo 5 de 6	-R\$ 173,91	Efetivar
03/05/2009	Prestacao Empréstimo 6 de 6	-R\$ 173,91	Efetivar
04/12/2008	Prestacao Empréstimo 1 de 3	-R\$ 170,78	Efetivar
04/01/2009	Prestacao Empréstimo 2 de 3	-R\$ 170,78	Efetivar
04/02/2009	Prestacao Empréstimo 3 de 3	-R\$ 170,78	Efetivar

Figura 6.16 – Extrato solicitado ao BancoOnline após as operações do BancoMe

A organização interna do BancoMe segue a estrutura apresentada na Figura 6.17.



Figura 6.17 – Estrutura da aplicação BancoME

A estrutura foi desenvolvida para executar transações a partir do envio da mesma para o BancoOnline ou para processamento no próprio dispositivo, essa opção será tratada posteriormente. Algumas alterações foram inseridas no BancoOnline para permitir a requisição das operações do BancoMe a partir de serviços web. A nova estrutura é representada na Figura 6.18.

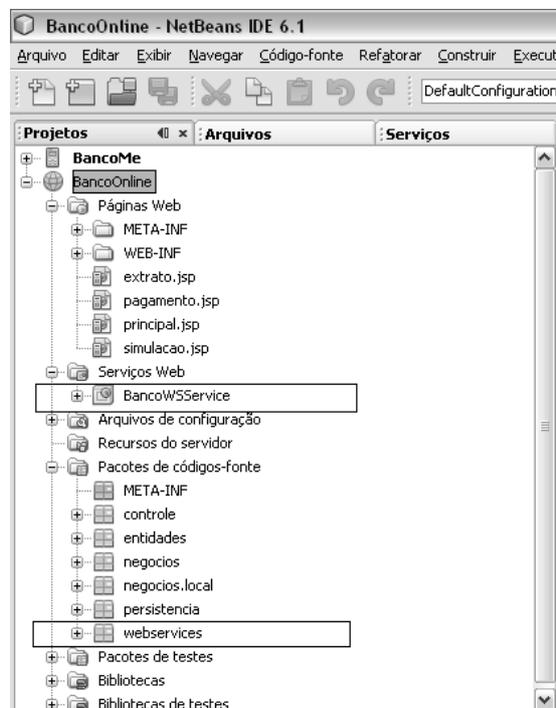


Figura 6.18 – Estrutura da aplicação BancoOnline com apoio a serviços web

Um pacote importante está selecionado na Figura 6.18 o BancoWSService. Esse pacote oferece os mecanismos necessários para a solicitação remota de operações, a Figura 6.19 ressalta o BancoWSService_Stub.java. Essa classe representa os serviços disponíveis no BancoWSService da Figura 6.18.



Figura 6.19 – Estrutura do BancoMe com apoio a serviços web

Todas as chamadas serão realizadas a partir do instanciamento do *stub* do serviço web, método disponível na Figura 6.20.

```

public BancoMeRemoto() {
    bws = new BancoWSService_Stub();
    txDao = TaxasDao.getInstance();
    lnDao = LancamentoDao.getInstance();
    ctDao = ContaDao.getInstance();
}

```

Figura 6.20 – Instância do Serviço Web

Quando necessário, os serviços web trafegam objetos através de TOs (*transfer object*). A Figura 6.21 mostra o retorno dos dados da conta a partir de ContaTO depois da chamada do método getConta().

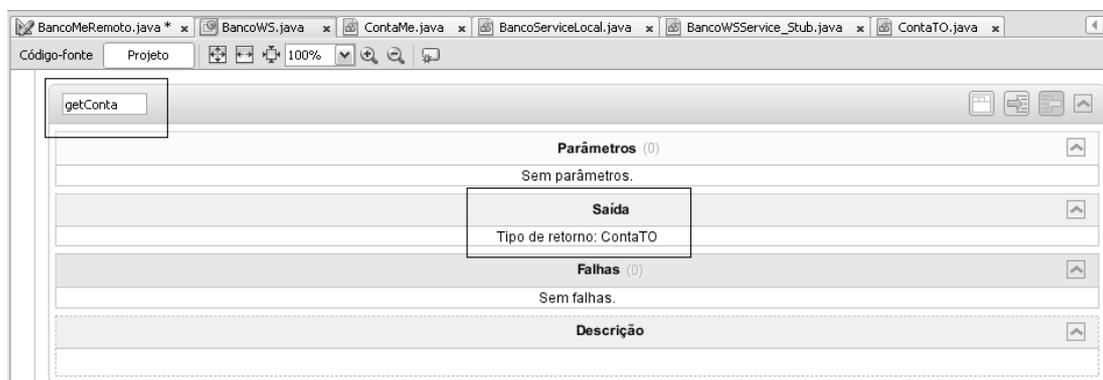


Figura 6.21 – ContaTO

A Figura 6.22 exibe o método `getConta` que pertence a `BancoWSService` (dentro de `BancoOnline`), além de `ContaTO` que faz parte do pacote de serviços web. O código a direita mostra a estrutura de um objeto de transferência.

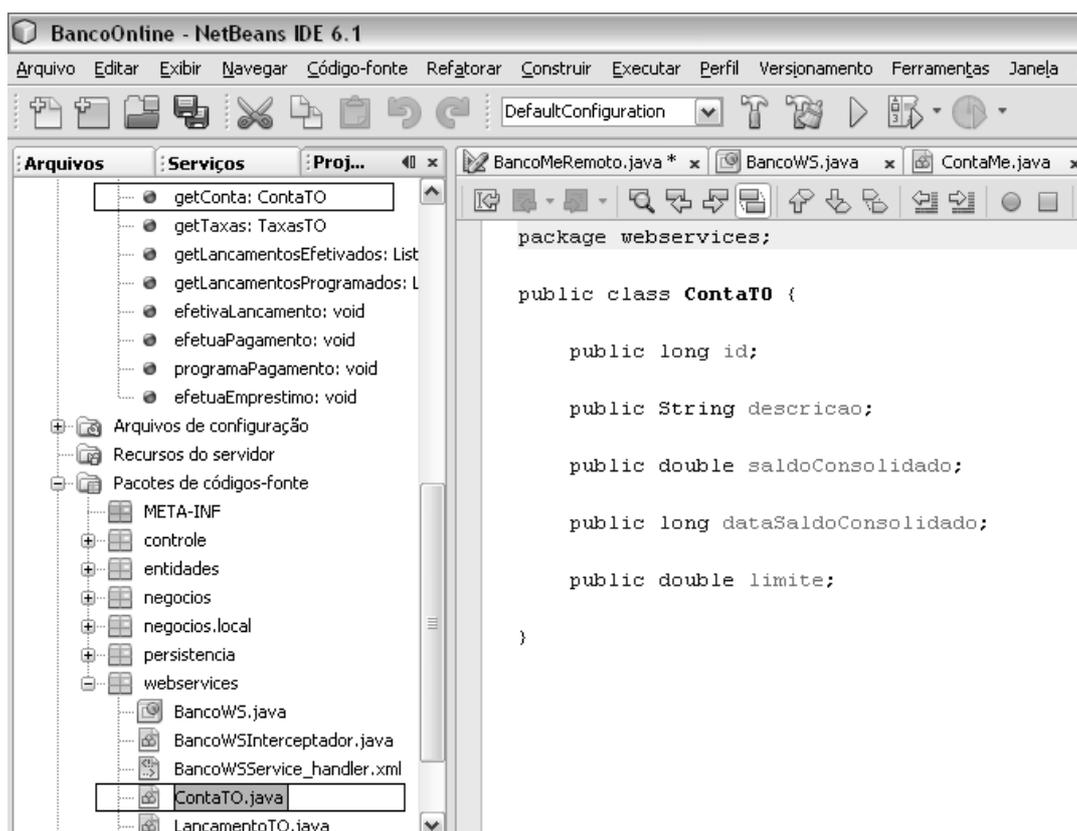


Figura 6.22 – Classe `ContaTO`

Outra opção de processamento de uma T_{MAC} é o plano isolado. Nesse caso, dados em cache são necessários. É fundamental uma política adequada para validação/invalidação desses dados para evitar transmissões desnecessárias, Berkenbrock e Dantas (2005) apresentam um estudo sobre o assunto. No `BancoOnline`, uma estratégia empregada para minimizar a transmissão durante o *hoarding* consistiu na persistência de dados resultantes de transações processadas no plano remoto. O Apêndice B, por meio de diagramas de seqüência, apresenta detalhes da ordem temporal de chamada dos métodos para todos os planos de execução possíveis.

A mudança do plano remoto para o isolado não impõe reflexos na forma como o usuário usa a aplicação, pois as interfaces apresentadas anteriormente continuam as mesmas. A diferença ocorre internamente, confira a Figura 6.23.

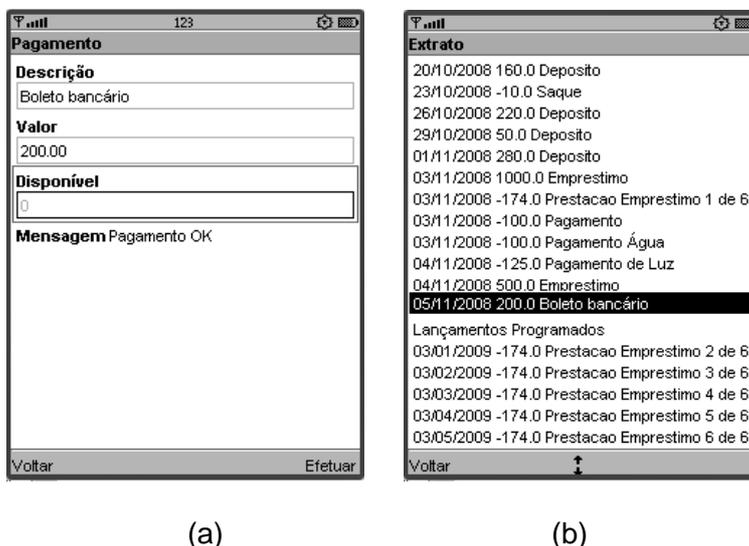


Figura 6.23 – Pagamento realizado a partir do plano de execução isolada

A Figura 6.23 (a) apresenta um pagamento a partir do BancoMe. Apesar de o lançamento constar do extrato seguinte, Figura 6.23 (b), devido à desconexão, o mesmo não aparece no extrato solicitado a partir do BancoOnline no servidor, disponível na Figura 6.24.

Banco Online - Mozilla Firefox

Arquivo Editar Exibir Histórico Favoritos Ferramentas Ajuda

Extrato

Conta AGENCIA 044 CONTA 12345-6

Limite R\$ 1.000,00

Disponível R\$ 2.901,09

Lançamentos Efetivados

19/10/2008	Saldo Anterior	R\$ 200,00
20/10/2008	Deposito	R\$ 160,00
23/10/2008	Saque	-R\$ 10,00
26/10/2008	Deposito	R\$ 220,00
29/10/2008	Deposito	R\$ 50,00
01/11/2008	Deposito	R\$ 280,00
03/11/2008	Emprestimo	R\$ 1.000,00
03/11/2008	Prestacao Emprestimo 1 de 6	-R\$ 173,91
03/11/2008	Pagamento	-R\$ 100,00
03/11/2008	Pagamento Água	-R\$ 100,00
04/11/2008	Pagamento de Luz	-R\$ 125,00
04/11/2008	Emprestimo	R\$ 500,00
	Saldo Anual	R\$ 1.901,09

Figura 6.24 – Extrato disponível através de BancoOnline

Durante a reconexão as tarefas realizadas no modo isolado serão submetidas à avaliação durante a sincronização. A Figura 6.25 mostra uma seqüência na qual uma operação de empréstimo é realizada, Figura 6.25 (a), em seguida os lançamentos são conferidos no extrato, Figura 6.25 (b). A primeira tentativa de sincronização informa que não há comunicação e por isso, o dispositivo opera em modo isolado, Figura 6.25 (c). Após a reconexão e durante a sincronização o empréstimo é confirmado, Figura 6.25 (d).

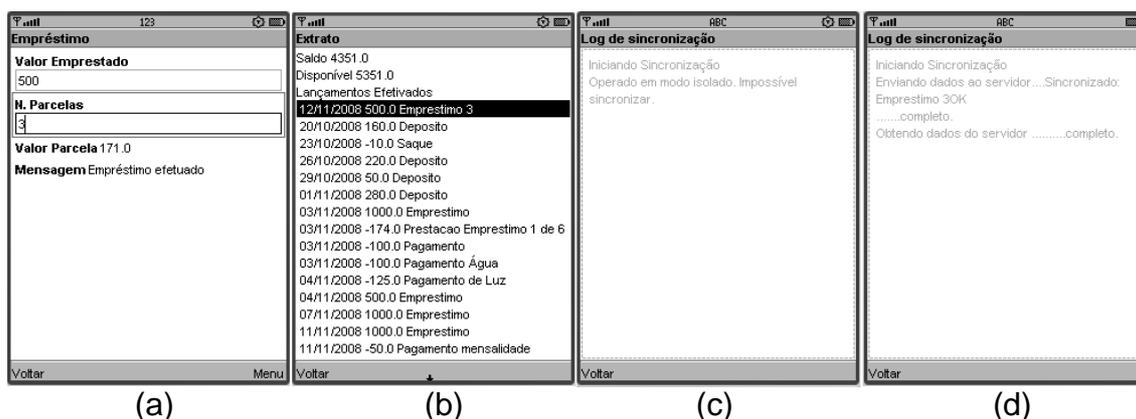


Figura 6.25 – Seqüência de interfaces para empréstimo e sincronização com êxito

A Figura 6.26 mostra outro cenário, na tentativa de sincronização, Figura 6.26 (c), após a execução de um empréstimo no plano de execução isolada, Figura 6.26 (a) e Figura 6.26 (b), uma mensagem informa que as taxas praticadas pelo banco foram alteradas, Figura 6.26 (c) e o empréstimo não poderá ser efetivado, Figura 6.26 (d).

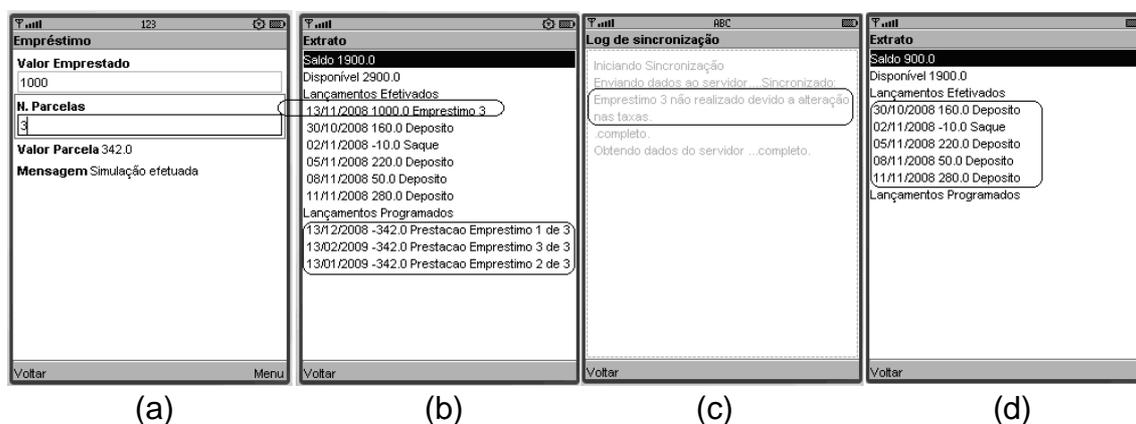


Figura 6.26 – Seqüência de interfaces para empréstimo e sincronização sem êxito

O plano de execução compartilhada impõe a estrutura multicomponente à T_{MAC} . Os exemplos tratados a seguir constituem as mesmas operações executadas pelos planos anteriores. Para cada transação uma Tabela e um grafo acíclico dirigido oferecem detalhes sobre as plataformas de execução e as dependências entre as subtransações. A Figura 6.27 apresenta um modelo da T_{MAC} multicomponente para a execução do extrato da conta corrente.

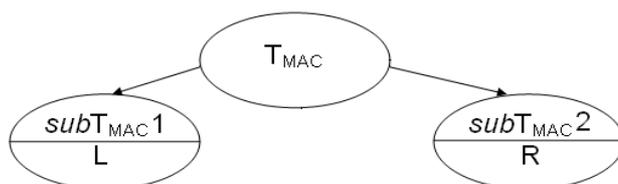


Figura 6.27 – Transação multicomponente para extrato de conta corrente

A Tabela 6.1 exibe as operações que constam de cada $subT_{MAC}$ do modelo da T_{MAC} multicomponente da Figura 6.27.

Tabela 6.1 – $SubT_{MACs}$ da operação de extrato

Nºda $subT_{MAC}$	Operação	Dependência	Plataforma
1	ler (lançamentos)	0	L
2	ler (saldo)	0	R

As $subT_{MACs}$ que interessam ao escalonador do banco de dados são as de acesso, assim a Figura 6.28 apresenta a estrutura multicomponente da transação de extrato somente com subtransações de leitura e escrita.

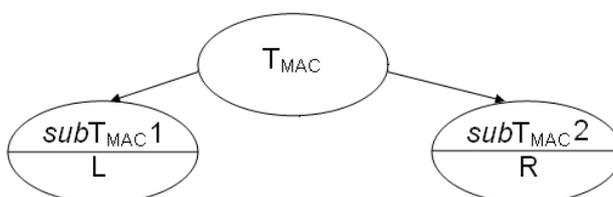


Figura 6.28 – Transação multicomponente de extrato com $subT_{MACs}$ de acesso

Das operações tratadas no estudo de caso, somente extrato não possui $subT_{MACs}$ de computação, devido à isso as Figuras 6.27 e 6.28 são iguais.

A Figura 6.29 apresenta um modelo da T_{MAC} multicomponente para a execução de pagamento.

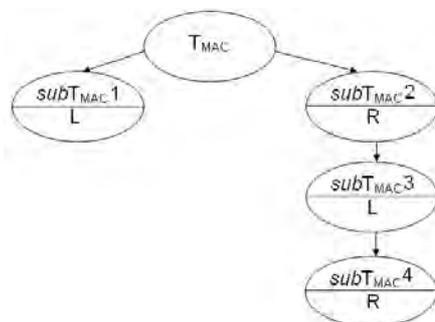


Figura 6.29 – Transação multicomponente para pagamento de contas

A Tabela 6.2 exibe as operações que constam de cada $subT_{MAC}$ do modelo da T_{MAC} multicomponente da Figura 6.29.

Tabela 6.2 – $subT_{MAC}$ s da operação de pagamento

Nº da $subT_{MAC}$	Operação	Dependência	Plataforma
1	escreve (lançamento)	0	L
2	ler (saldo)	0	R
3	se (lançamento < saldo) saldo := saldo – lançamento	2	L
4	escreve (saldo)	3	R

A Figura 6.30 apresenta a estrutura multicomponente da transação de pagamento somente com subtransações de acesso. A subtransação retirada foi a 3 pois executa computação.

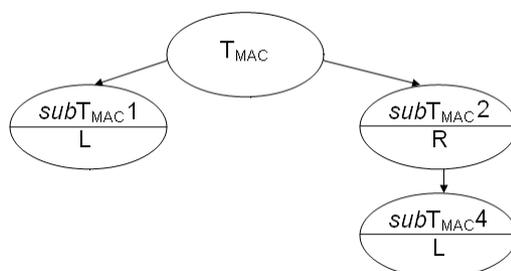


Figura 6.30 – Transação multicomponente de pagamento com $subT_{MAC}$ s de acesso

A Figura 6.31 apresenta um modelo da T_{MAC} multicomponente para a execução do empréstimo.

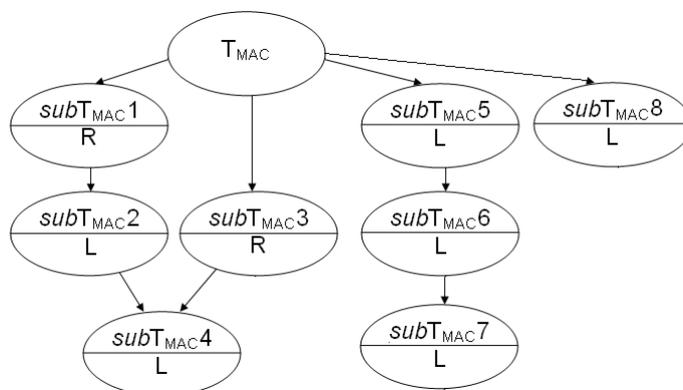


Figura 6.31 – Transação multicomponente para empréstimo

A Tabela 6.3 exibe as operações que constam de cada $subT_{MAC}$ do modelo da T_{MAC} multicomponente da Figura 6.31.

Tabela 6.3 – $SubT_{MACs}$ da operação de empréstimo

Nº da $subT_{MAC}$	Operação	Dependência	Plataforma
1	ler (taxa)	0	R
2	calcula (parcela)	1	L
3	ler (limite)	0	R
4	parcela > limite	2,3	L
5	ler (saldo)	0	L
6	saldo := saldo + valor	5	L
7	escreve (saldo)	6	L
8	escreve (lançamento)	0	L

Da Figura 6.31 foram retiradas as subtransações 2, 4 e 6, dando origem a Figura 6.32.

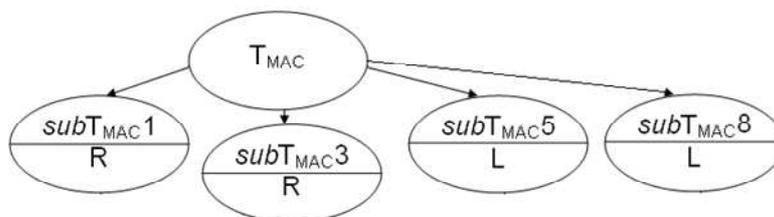


Figura 6.32 – Transação multicomponente de empréstimo com $subT_{MACs}$ de acesso

A Figura 6.33 apresenta um modelo da T_{MAC} multicomponente para a execução da simulação de empréstimo.

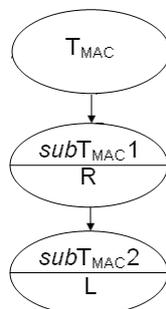


Figura 6.33 – Transação multicomponente para simulação de empréstimo

A Tabela 6.4 exibe as operações que constam de cada $subT_{MAC}$ do modelo da T_{MAC} multicomponente da Figura 6.33.

Tabela 6.4 – $SubT_{MACs}$ da operação de simulação de empréstimo

Nº da $subT_{MAC}$	Operação	Dependência	Plataforma
1	ler (taxa)	0	R
2	calcula (parcela)	1	L

A Figura 6.34 apresenta a estrutura multicomponente da transação de simulação de empréstimo somente com subtransações de acesso. A subtransação 2 foi retirada, pois realiza somente computação.

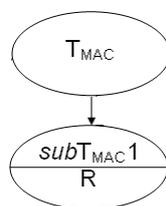


Figura 6.34 – Transação multicomponente de simulação de empréstimo com $subT_{MACs}$ de acesso

A criação das $subT_{MACs}$ e as dependências são implementadas durante a etapa de desenvolvimento. Com o intuito de explorar a visibilidade da ocorrência do plano de execução compartilhada da operação de empréstimo, o *log* de execução foi capturado e é apresentado na Figura 6.35.

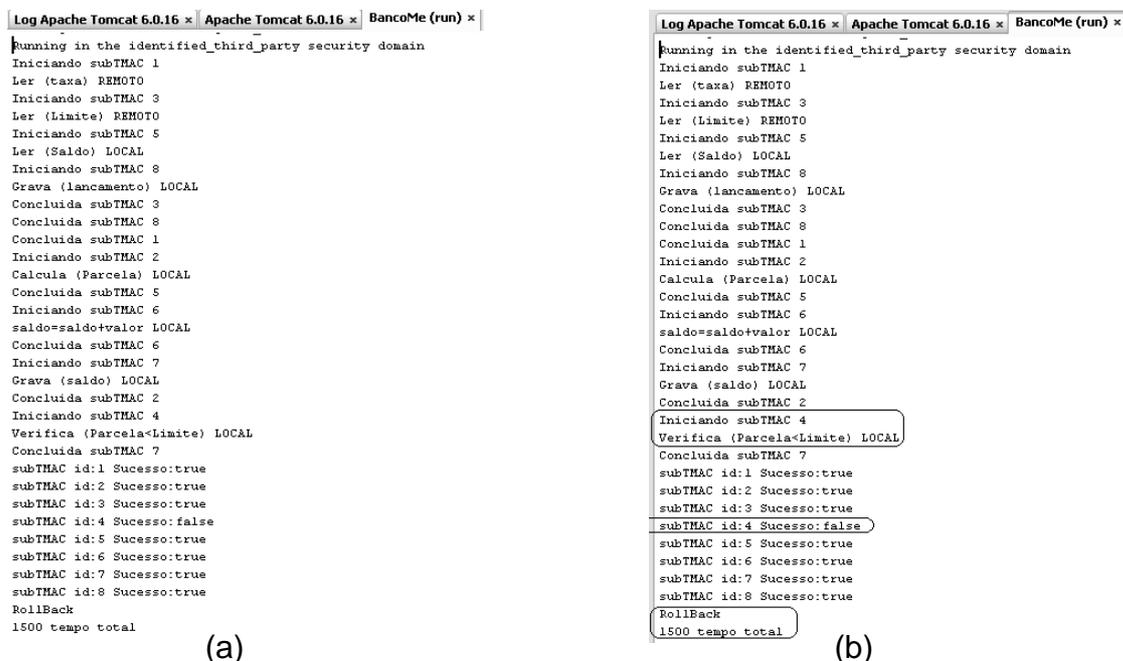


Figura 6.35 – Log de execução da operação de empréstimo no modo compartilhado

A parte (a) da Figura 6.35 exibe o log de execução, enquanto a parte (b) explora a ocorrência de falha na $subT_{MAC}$ 4, ocasionando o cancelamento (*rollback*) de toda a transação. O Δt informado para o processamento da transação foi de 1500 milissegundos. A Figura 6.36 mostra essa informação dentro do código-fonte.

```
MidMobile.executaCompartilhado(1500, sts);
```

Figura 6.36 – Informação do Δt para a T_{MAC}

Outra informação importante diz respeito às dependências entre as transações. A Figura 6.37 apresenta como essa característica foi tratada.

```

subtmac2.setDp(new SubTMAC[] {subtmac1});
subtmac4.setDp(new SubTMAC[] {subtmac2, subtmac3});
subtmac6.setDp(new SubTMAC[] {subtmac5});
subtmac7.setDp(new SubTMAC[] {subtmac6});

```

Figura 6.37 – Dependência de $subT_{MACs}$

O log de execução confirma o processamento de acordo com a dependência declarada. A Figura 6.38 (a) mostra o início das $subT_{MAC}$ s de primeiro nível, a Figura 6.38 (b) enfoca a execução da $subT_{MAC}$ 2 logo após a conclusão da 1.

```

Log Apache Tomcat 6.0.16 x Apache Tomcat 6.0.16 x BancoMe (run) x
Running in the identified_third_party security domain
Iniciando subTMAC 1
Ler (taxa) REMOTO
Iniciando subTMAC 3
Ler (Limite) REMOTO
Iniciando subTMAC 5
Ler (Saldo) LOCAL
Iniciando subTMAC 8
Grava (lancamento) LOCAL
Concluida subTMAC 3
Concluida subTMAC 8
Concluida subTMAC 1
Iniciando subTMAC 2
Calcula (Parcela) LOCAL
Concluida subTMAC 5
Iniciando subTMAC 6
saldo=saldo+valor LOCAL
Concluida subTMAC 6
Iniciando subTMAC 7
Grava (saldo) LOCAL
Concluida subTMAC 2
Iniciando subTMAC 4
Verifica (Parcela<Limite) LOCAL
Concluida subTMAC 7
subTMAC id:1 Sucesso:true
subTMAC id:2 Sucesso:true
subTMAC id:3 Sucesso:true
subTMAC id:4 Sucesso:false
subTMAC id:5 Sucesso:true
subTMAC id:6 Sucesso:true
subTMAC id:7 Sucesso:true
subTMAC id:8 Sucesso:true
RollBack
1500 tempo total

```

```

Log Apache Tomcat 6.0.16 x Apache Tomcat 6.0.16 x BancoMe (run) x
Running in the identified_third_party security domain
Iniciando subTMAC 1
Ler (taxa) REMOTO
Iniciando subTMAC 3
Ler (Limite) REMOTO
Iniciando subTMAC 5
Ler (Saldo) LOCAL
Iniciando subTMAC 8
Grava (lancamento) LOCAL
Concluida subTMAC 3
Concluida subTMAC 8
Concluida subTMAC 1
Iniciando subTMAC 2
Calcula (Parcela) LOCAL
Concluida subTMAC 5
Iniciando subTMAC 6
saldo=saldo+valor LOCAL
Concluida subTMAC 6
Iniciando subTMAC 7
Grava (saldo) LOCAL
Concluida subTMAC 2
Iniciando subTMAC 4
Verifica (Parcela<Limite) LOCAL
Concluida subTMAC 7
subTMAC id:1 Sucesso:true
subTMAC id:2 Sucesso:true
subTMAC id:3 Sucesso:true
subTMAC id:4 Sucesso:false
subTMAC id:5 Sucesso:true
subTMAC id:6 Sucesso:true
subTMAC id:7 Sucesso:true
subTMAC id:8 Sucesso:true
RollBack
1500 tempo total

```

(a) (b)
 Figura 6.38 – Log de execução com enfoque nas dependências

A Figura 6.39 (a) mostra a conclusão da $subT_{MAC}$ 2 e 3 e o início da 4. A Figura 6.39 (b) exhibe o início da $subT_{MAC}$ 7 após o início e conclusão da 5 e 6.

```

Log Apache Tomcat 6.0.16 x Apache Tomcat 6.0.16 x BancoMe (run) x
Running in the identified_third_party security domain
Iniciando subTMAC 1
Ler (taxa) REMOTO
Iniciando subTMAC 3
Ler (Limite) REMOTO
Iniciando subTMAC 5
Ler (Saldo) LOCAL
Iniciando subTMAC 8
Grava (lancamento) LOCAL
Concluida subTMAC 3
Concluida subTMAC 8
Concluida subTMAC 1
Iniciando subTMAC 2
Calcula (Parcela) LOCAL
Concluida subTMAC 5
Iniciando subTMAC 6
saldo=saldo+valor LOCAL
Concluida subTMAC 6
Iniciando subTMAC 7
Grava (saldo) LOCAL
Concluida subTMAC 2
Iniciando subTMAC 4
Verifica (Parcela<Limite) LOCAL
Concluida subTMAC 7
subTMAC id:1 Sucesso:true
subTMAC id:2 Sucesso:true
subTMAC id:3 Sucesso:true
subTMAC id:4 Sucesso:false
subTMAC id:5 Sucesso:true
subTMAC id:6 Sucesso:true
subTMAC id:7 Sucesso:true
subTMAC id:8 Sucesso:true
RollBack
1500 tempo total

```

```

Log Apache Tomcat 6.0.16 x Apache Tomcat 6.0.16 x BancoMe (run) x
Running in the identified_third_party security domain
Iniciando subTMAC 1
Ler (taxa) REMOTO
Iniciando subTMAC 3
Ler (Limite) REMOTO
Iniciando subTMAC 5
Ler (Saldo) LOCAL
Iniciando subTMAC 8
Grava (lancamento) LOCAL
Concluida subTMAC 3
Concluida subTMAC 8
Concluida subTMAC 1
Iniciando subTMAC 2
Calcula (Parcela) LOCAL
Concluida subTMAC 5
Iniciando subTMAC 6
saldo=saldo+valor LOCAL
Concluida subTMAC 6
Iniciando subTMAC 7
Grava (saldo) LOCAL
Concluida subTMAC 2
Iniciando subTMAC 4
Verifica (Parcela<Limite) LOCAL
Concluida subTMAC 7
subTMAC id:1 Sucesso:true
subTMAC id:2 Sucesso:true
subTMAC id:3 Sucesso:true
subTMAC id:4 Sucesso:false
subTMAC id:5 Sucesso:true
subTMAC id:6 Sucesso:true
subTMAC id:7 Sucesso:true
subTMAC id:8 Sucesso:true
RollBack
1500 tempo total

```

(a) (b)
 Figura 6.39 – Log de execução da operação de empréstimo no modo compartilhado

A execução satisfatória (*commit*) da T_{MAC} é conseguida com o sucesso de todas as $subT_{MACs}$, respeitado o tempo informado para cada transação. A Figura 6.40 (a) e (b) mostram respectivamente o log de execução com sucesso de uma T_{MAC} de empréstimo e o tempo (Δt) designado para a transação.

The image shows a screenshot of a Java application window with three tabs: 'Apache Tomcat 6.0.16', 'BancoOnline (run)', and 'BancoMe (run)'. The main content area displays a log of subtransactions (subTHAC) and their completion status. At the bottom, a box highlights the commit details, including the total time (1922) and a timeout (Cancelado). To the right, a code snippet shows a call to `MidMobile.executaCompartilhado(2500, sts);` with a box around the number 2500.

```

Apache Tomcat 6.0.16 x BancoOnline (run) x BancoMe (run) x
Iniciando subTHAC 1
Ler (taxa) REMOTO
Iniciando subTHAC 3
Ler (Limite) REMOTO
Iniciando subTHAC 5
Ler (Saldo) LOCAL
Iniciando subTHAC 8
Grava (lancamento) LOCAL
Concluida subTHAC 3
Concluida subTHAC 8
Concluida subTHAC 1
Iniciando subTHAC 2
Calcula (Parcela) LOCAL
Concluida subTHAC 5
Iniciando subTHAC 6
saldo=saldo+valor LOCAL
Concluida subTHAC 2
Iniciando subTHAC 4
Verifica (Parcela<Limite) LOCAL
Concluida subTHAC 6
Iniciando subTHAC 7
Grava (saldo) LOCAL
Concluida subTHAC 4
Concluida subTHAC 7
Commit
1922 tempo total
Timeout Cancelado
MidMobile.executaCompartilhado(2500, sts);
  
```

Figura 6.40 – Log de execução da operação de empréstimo e configuração do Δt

Cada operação deve ser avaliada cuidadosamente para que o tempo de execução seja satisfatório diante da tecnologia de comunicação utilizada. Cada subtransação deve ser alocada à plataforma de processamento que contenha os dados necessários ou os mais recentes.

6.3 Considerações

Este capítulo apresentou um protótipo que serviu para validação do MiD-Mobile, apresentado no Capítulo 4 e do modelo de transações móveis adaptadas ao contexto exposto no Capítulo 5. O *middleware* é a ferramenta responsável por gerenciar a execução do modelo de transações proposto neste trabalho. Sua existência subtrai funções de decisão da aplicação, fazendo-a mais simples, além de fazer as chamadas dos serviços e o tratamento das dependências entre as $subT_{MACs}$.

O desenvolvimento do MiD-Mobile foi providencial no gerenciamento da mudança da estrutura da transação, mecanismo utilizado para prover a adaptação colaborativa. A colaboração de um reconhecedor de características do ambiente e da adequação da estrutura da transação diante do ambiente identificado maximiza as chances de sucesso de transações iniciadas em dispositivos portáteis, principalmente quando submetidas à instabilidade do canal de comunicação sem fio.

As propriedades ACID, mesmo sendo consideradas restritivas, ainda são amplamente empregadas como medida de confiabilidade das unidades computacionais. Diante disto, o estudo de caso apresentou a ocorrência de situações nas quais as propriedades poderiam ser violadas e exibiu as medidas tomadas em cada caso para evitar tais ocorrências, abrangendo qualquer plano de execução indicado à transação.

A junção do modelo das T_{MACs} ao MiD-Mobile, demonstrou a viabilidade da proposta deste trabalho, na medida em que permitiu a harmonização das transações com a disponibilidade de recursos computacionais, sem sacrificar as funcionalidades do software. A seguir são analisadas as conclusões e principais contribuições desta pesquisa, bem como são feitas sugestões para trabalhos futuros.

7. CONCLUSÕES E TRABALHOS FUTUROS

Vários esforços nas áreas de hardware e software para a computação móvel vêm sendo feitos para viabilizar o desenvolvimento de sistemas de computação móvel. O principal problema enfrentado nesta área diz respeito às características restritivas do ambiente, constituído pelos canais de comunicação sem fio e equipamentos móveis de capacidade limitada, que limitam o processamento das transações e acarretam em cancelamento.

Diante disto, esta pesquisa despendeu esforços direcionados aos desafios implícitos às transações iniciadas a partir de equipamentos móveis. Com vistas a esse objetivo, uma ampla revisão bibliográfica permitiu avaliar o estágio atual de estudos nessa área. Este trabalho visa dar uma contribuição ao tema, adotando um modelo baseado em transações móveis adaptadas ao contexto e demonstrando sua operacionalização, através de um gerenciador de transações, denominado MiD-Mobile.

Os trabalhos formalizados inicialmente dizem respeito a modelos transacionais que oferecem solução de processamento com características únicas. A singularidade de propósito identificada nos modelos comparada à pluralidade de interferências que afetam os sistemas de computação móvel requereu a aplicação de abordagens mais dinâmicas como a adaptação, a fim de reutilizar o software em contextos múltiplos.

Uma visão mais crítica possibilitou o aproveitamento da adaptação, aliando-a ao conceito de aprendizagem, estratégia proposta nesta pesquisa. A subseção seguinte elenca as contribuições dessa junção ao processamento de transações em ambiente de computação móvel, que foram apresentadas em Flôr et al. (2005a), (2005b), (2006), (2007), (2008a), (2008b), (2008c), (2008d) e (2008e) e Paulino (2006a) e (2006b).

7.1 Contribuições da pesquisa

Permitir a execução de requisitos funcionais em sistemas de computação móvel significa submeter transações às variações de recursos do dispositivo portátil e da conectividade sem fio. Um cenário sujeito às desconexões ou a falhas de site compromete as funções disponibilizadas aos usuários. Dessa forma, um instrumento de decisão dinâmica foi desenvolvido para adaptar a transação ao processamento isolado, remoto ou compartilhado.

O plano de processamento isolado prevê uma situação de falta de comunicação com a rede fixa e completa hospedagem dos dados usados em tal execução. A abordagem remota é indicada para dispositivos de recursos modestos, que usufruem de algum tipo de comunicação sem fio, configuração ideal para emitir requisições, mas não para processá-las. A execução compartilhada é ideal diante da instabilidade da rede de comunicação ou devido à parcialidade dos dados usados nas transações.

Assim, a adaptação das transações é exercida em sua organização interna, com base na situação da rede de comunicação, nas configurações do equipamento móvel e nos dados disponíveis. Esse tipo de adaptação possui a colaboração da aplicação e do sistema.

A contribuição da aplicação para o provimento da adequação conduz à mudança da estrutura da transação, sendo possível o modelo plano ou multicomponente. A primeira opção é a mais tradicionalmente conhecida, enquanto a segunda compõe as iniciativas desta pesquisa e diz respeito à divisão da transação em componentes interdependentes executados em plataformas diferentes.

A contribuição do sistema foi implementada em uma camada alocada entre a camada da aplicação e a camada dos sistemas gerenciadores de banco de dados, denominada MiD-Mobile. Os serviços internos são responsáveis pelo reconhecimento e monitorização do ambiente. Outro elemento identificado pelo *middleware* engloba a avaliação dos recursos da rede de comunicação sem fio.

A comunicação direta entre as camadas do *middleware* é outro empreendimento deste estudo. O Serviço Gerenciador de Dados e Mensagens objetiva a troca de dados e mensagens entre estações móveis e rede fixa, evitando serviços adicionais às estações de base. A mobilidade intensa dos dispositivos móveis ocasiona um grande fluxo de registros. Agregar mais responsabilidade a essa camada pode contribuir para o prejuízo dos serviços prestados ou representar um gargalo no processamento.

Todas as requisições entre as camadas do MiD-Mobile são tratadas, mesmo se o dispositivo não estiver conectado. O serviço implementado oferece uma ferramenta de armazenamento que assegura que, quando o dispositivo móvel se reconectar, receberá tudo o que foi enviado a ele.

Independente da estrutura da transação, as propriedades ACID permanecem em uso como requisitos que garantem a integridade e a confiabilidade dos sistemas de computação. No plano de execução remota e isolada a organização interna da transação é plana. Apesar disso, transgressões diferentes podem atingir as características ACID.

No plano de execução isolada, a consistência pode ser questionada frente a réplicas desatualizadas. Nesse sentido, foi proposto um Protocolo de Consistência Global que certifica a possibilidade de sincronização entre dados alterados em dispositivos móveis desconectados.

A sincronização empregada pelo MiD-Mobile verifica a versão dos dados mantidos em cache para evitar transmissão desnecessária no canal de comunicação sem fio, principalmente diante de tarifação.

No plano de execução compartilhada várias adequações foram necessárias, além do Protocolo de Consolidação Global. Algoritmos conhecidos como o 2PC e o 2PL foram adaptados para trabalharem em conjunto com um controlador de tempo de execução (Δt). Os algoritmos implementados foram denominados $2PC\Delta t$ e $2PL\Delta t$ e evitam impasses e mantém a unicidade de operação.

O tempo de execução global (Δt) foi tratado de forma global e parcial para atender as $subT_{MACs}$. Para isso, foi definido o tempo de execução parcial denominado Δst . O Δt assegurou o tempo máximo para a transação e o Δst fez o mesmo para cada subtransação de uma T_{MAC} adequada ao plano compartilhado.

Outra contribuição do MiD-Mobile diz respeito à mudança de local de execução, permitindo que o processamento seja executado no lugar onde os dados estão. Essa estratégia viabiliza a diminuição dos dados trafegados além de não sobrecarregar os servidores da rede fixa com tantas requisições.

O mecanismo de aprendizagem agregou ao *middleware* a capacidade de refinar suas decisões, aumentando as chances de sucesso da T_{MAC} com o devido plano de execução. O retreinamento a partir de todas as experiências dos dispositivos portáteis propiciará ao algoritmo da árvore de decisão generalizar corretamente um número cada vez maior de casos de testes.

O MiD-Mobile foi proposto no formato de serviços acessíveis por interfaces padronizadas. A abordagem facilita a chamada dos métodos e viabiliza a utilização por outros sistemas de computação móvel. Muitos outros cenários

desafiantes se configuram a partir da utilização do *middleware*, principalmente os que adicionam inteligência aos sistemas de computação móvel.

7.2 Trabalhos Futuros

O MiD-Mobile encerra em si essas funções, mas não esgota a pluralidade de temas desafiantes como, por exemplo, escalabilidade do sistema de computação móvel. O *middleware* objetiva a popularização do acesso à aplicação por dispositivos móveis com modesta configuração, aumentando assim as chamadas à rede fixa. Em pesquisas futuras, faz-se necessário verificar o comportamento das aplicações e como garantir que a disponibilidade de serviço não seja prejudicada.

Outra consideração é sobre a desejável onipresença da computação móvel. Quanto mais o MiD-Mobile dispensar a colaboração do usuário, menor será o seu ônus com a aplicação, e mais a evolução dessa tendência coincidirá com a idéia de interação natural e imperceptível do usuário com o ambiente. Para alcançar esse estado, a sensibilidade ao contexto é a estratégia que merece atenção. É importante que as aplicações saibam agir dependendo do contexto e da semântica da aplicação.

Para validar a arquitetura completamente é necessária a implementação e integração de todos os serviços propostos. Dessa forma, será possível avaliar resultados reais da utilização do *middleware*.

Outra tarefa diz respeito a uma avaliação mais profunda sobre a discretização das dimensões do contexto móvel, bem como alguns aspectos na aprendizagem, como a manipulação de atributos com diferentes custos (devido a pluralidade de formas de comunicação sem fio) e o tratamento de valores desconhecidos (para considerar dados com completeza e validade indefinidos). Tais trabalhos podem refinar a atuação da ferramenta.

REFERÊNCIAS BIBLIOGRÁFICAS

AL-BAR, A.; WAKEMAN, I. A survey of adaptive applications in mobile computing. In: INTERNATIONAL CONFERENCE ON DISTRIBUTED COMPUTING SYSTEMS WORKSHOPS, n. 21, 2001, Phoenix. **Proceedings ...**, Phoenix: IEEE, 2001. p. 246.

BERKENBROCK, C. D. M.; DANTAS, M. A. R. Investigação e Implementação de Estratégias de Coerência de Cache em um Ambiente de Computação Móvel sem Fio. In: SEMINÁRIO INTEGRADO DE SOFTWARE E HARDWARE, n. 32, 2005, São Leopoldo. **Anais ...**, São Leopoldo: SBC, 2005. p.1601-1615.

BHARGHAVAN, V. GUPTA, V. A Framework for Application Adaptation in Mobile Computing Environments. In: INTERNATIONAL COMPUTER SOFTWARE AND APPLICATIONS CONFERENCE, n. 21, 1997, Washigton. **Proceedings ...**, Washigton: IEEE, 1997. p. 573-579.

CHRYSANTHIS, P. K. Transaction Processing in Mobile Computing Environment. In: WORKSHOP ON ADVANCES IN PARALLEL AND

DISTRIBUTED SYSTEMS, n.1, 1993, Princeton. **Proceedings** Princeton: IEEE, 1993. p.77-82.

CÔRTEZ, S. C. **Um modelo de transações para integração de SGBD a um ambiente de computação móvel**. 215 f. Tese (Doutorado) – PUC, Rio de Janeiro, 2004. Disponível em: http://www2.dbd.puc-rio.br/pergamum/tesesabertas/9924918_04_pretextual.pdf. Acesso em: 16 fev. 2007.

COULOURIS, G.; DOLLIMORE, J.; KINDBERG, T. **Sistemas distribuídos: conceitos e projeto**. 4. ed. Porto Alegre: Bookman, 2007, 784 p.

DIRCKZE, R. A.; GRUENWALD, L. A Pre-Serialization Transaction Management Technique for Mobile Multidatabases. **Mobile Networks and Applications**, v. 5, n. 4, p. 311-321, 2000.

DUNHAM, M. H.; HELAL, A.; BALAKRISHNAN, S. A Mobile Transaction Model that Captures both the Data and Movement Behavior. **Mobile Networks and Applications**, v.2, n. 2, p. 149-162, 1997.

DUNHAM, M.H.; KUMAR, V. Impact of mobility on transaction management. In: INTERNATIONAL WORKSHOP ON DATA ENGINEERING FOR WIRELESS AND MOBILE ACCESS, n. 1, 1999, Seattle. **Proceedings** ... New York: ACM, 1999. p. 14-21.

DUNHAM, M. H.; HELAL, A.; BALAKRISHNAN, S. Mobile Computing and Database: Anything new?. In: WIDOM, J.; DUNHAM, M. H.; HELAL, A. **ACM SIGMOD RECORD**. ACM, v. 24 , n. 4, p.5-9, 1995.

ELMASRI, R.; NAVATHE, S. **Sistemas de banco de dados: fundamentos e aplicações**. 4. ed. São Paulo: Addison Wesley, 2005, 724 p.

FERREIRA, J. E.; FINGER, M. **Controle de concorrência e distribuição de dados: a teoria clássica, suas limitações e extensões modernas**. São Paulo: IME-USP, 2000, 179 p.

FLÔR, D. E.; FERREIRA, M. G. V.; SANT'ANNA, N. Arquitetura Adaptável para execução de transações móveis. In: INTERNATIONAL INFORMATION AND TELECOMMUNICATION TECHNOLOGIES SYMPOSIUM, n. 4, 2005, Florianópolis. **Proceedings** ..., Florianópolis: UFSC, 2005.

FLÔR, D. E.; FERREIRA, M. G. V.; SANT'ANNA, N. Arquitetura Adaptável para execução de transações móveis. In: WORKSHOP DOS CURSOS DE COMPUTAÇÃO APLICADA DO INPE, n. 5, 2005, São José dos Campos. **Livro de resumos do V Workshop dos cursos de computação aplicada do INPE**, 2005. Disponível em:

<http://hermes2.dpi.inpe.br:1905/col/dpi.inpe.br/hermes2@1905/2005/10.04.00.28/doc/Arquitetura%20Adaptavel%20para%20Execucao%20de%20Transacoes%20Moveis%20-%20Daniela%20Flor.pdf>. Acesso em: 16 fev. 2006.

FLÔR, D. E.; FERREIRA, M. G. V.; SANT'ANNA, N. Transactions adaptation for mobile computation environment. In: WORKSHOP DOS CURSOS DE COMPUTAÇÃO APLICADA DO INPE, n. 6, 2006, São José dos Campos. **Livro de resumos do VI Workshop dos cursos de computação aplicada do INPE**, 2006. Disponível em:

http://hermes2.dpi.inpe.br:1905/col/dpi.inpe.br/hermes2@1905/2006/10.20.22.50/doc/danielaflor_resumoworcap2006.pdf. Acesso em: 16 fev. 2007.

FLÔR, D. E.; COSTA, Y. M. G.; PAULINO, R. V. Aplicação de Técnica de Inteligência Artificial na tomada de decisão em ambiente de computação móvel. **Diálogos e Saberes**, v. 2, n.2, p. 81-90, 2007.

FLÔR, D. E.; FERREIRA, M. G. V.; SANT'ANNA, N.; COSTA, Y. M. G.; PAULINO, R. V.; GEBARA JUNIOR, M. Gerenciamento de transações móveis adaptadas ao contexto através do MID-MOBILE. In: VIII WORKSHOP DOS CURSOS DE COMPUTAÇÃO APLICADA DO INPE, n. 8, 2008, São José dos Campos. **Livro de resumos do VIII Workshop dos cursos de computação aplicada do INPE**, 2008. Disponível em:

<http://plutao.dpi.inpe.br/rep/dpi.inpe.br/plutao@80/2008/12.18.16.22.04/Cover?mirror=dpi.inpe.br/plutao@80/2008/08.19.15.01.21&metadataarepository=dpi.inpe.br/plutao@80/2008/12.18.16.22.05>.

FLÔR, D. E.; FERREIRA, M. G. V.; SANT'ANNA, N.; COSTA, Y. M. G.; PAULINO, R. V. MiD-Mobile: Middleware distribuído para adaptação e gerenciamento de transações em ambiente de computação móvel. In: SEMINÁRIO INTEGRADO DE SOFTWARE E HARDWARE, n. 28, 2008, Belém. **Anais ...**, Belém: SBC, 2008. p. 136-150.

FLÔR, D. E.; FERREIRA, M. G. V.; SANT'ANNA, N.; COSTA, Y. M. G.; PAULINO, R. V. Adaptação de transações sensíveis ao contexto com o MiD-Mobile. In: CONFERENCIA LATINOAMERICANA DE INFORMÁTICA, n. 34, 2008, Santa Fe. **Anais ...** Santa Fé: SADIO, 2008. p. 35-50.

FLÔR, D. E.; FERREIRA, M. G. V.; SANT'ANNA, N.; COSTA, Y. M. G.; PAULINO, R. V.; GEBARA JUNIOR, M. Adaptação de transações para sistemas de computação móvel. Aceito, **Revista Tecnológica**, nov., 2008.

FLÔR, D. E.; FERREIRA, M. G. V.; SANT'ANNA, N.; COSTA, Y. M. G.; PAULINO, R. V.; GEBARA JUNIOR, M. Gerência de transações adaptadas ao contexto de sistemas de computação móvel com o MiD-Mobile. In: INTERNATIONAL INFORMATION AND TELECOMMUNICATION TECHNOLOGIES SYMPOSIUM, n. 7, 2008, Foz do Iguaçu. **Proceedings ...**, Florianópolis: UFSC, 2008. p. 139-149.

GARCIA-MOLINA, H.; SALEM, K. Sagas. **ACM SIGMOD RECORD**, ACM, v. 16, n. 3, p. 249-259, 1987.

GEORGAKOPOULOS, D. Transactional Workflow Management in Distributed Object Computing Environments. In: INTERNATIONAL CONFERENCE ON DATA ENGINEERING, n.10, 1994, Washigton. **Proceedings...**, Washigton: IEEE, 1994. p. 540.

GRAY, J.; HELLAND, P.; O'NEIL, P.; SHASHA, D. The dangers of replication and a solution. In: INTERNATIONAL CONFERENCE ON MANAGEMENT DATA, 1996, Montreal. **Proceedings...**, Montreal: ACM, 1996. p. 173-182.

KU, K.; KIM, Y. Moflex Transaction Model for Mobile Heterogeneous Multidatabase Systems. In: WORKSHOP ON RESEARCH ISSUES IN DATA ENGINEERING, n. 10, 2000, San Diego. **Proceedings...**, Washigton: IEEE, 2000. p. 39.

LEE, M.; HELAL, S. HiCoMo: High Commit Mobile Transactions. **Distributed and Parallel Databases**, v. 11, n. 1, p. 73-92, 2002.

LU, Q.; SATYANARANYAN, M. Isolation-Only Transactions for Mobile Computing. **Operating Systems Review**, v. 28, n. 2. p. 81-87, 1994.

MADRIA, S. K.; BHARGAVA, B. A Transaction Model for Improving Data Availability in Mobile Computing. **Distributed and Parallel Databases**, v.10, n 2, p. 127-160. 2001.

MELO, R.; SILVA, S.; TANAKA, A. **Banco de dados em aplicação cliente-servidor**. Rio de Janeiro: São Paulo: Infobook, 1997.

MENEZES, P. B. **Linguagens formais e autômatos**. 2. ed. Rio Grande do Sul: Sagra Luzzatto, 1998.

MITCHELL, T. M. **Machine learning**. Washigton: McGraw-Hill, 1997.

MOHAN, C.; LINDSAY, B.; OBERMARCK, R. Transaction Management in the R* Distributed Database Management System. **Transactions on Database Systems**, v. 11, n. 4, p. 378-396, 1986.

MOSS, J. E. B. **Nested Transactions: An approach to reliable computing**. 348 f. Tese (Doutorado), MIT, Massachusetts, 1981. Disponível em: <http://www.lcs.mit.edu/publications/pubs/pdf/MIT-LCS-TR-260.pdf>. Acesso em: 10, jan. 2008.

ÖZSU, M. T.; VALDURIEZ, P. **Princípios de sistemas de bancos de dados distribuídos**. Rio de Janeiro: Campus, 2001, 711 p.

PAULINO, R.; FLÔR, D. E.; COSTA, Y. M. G. Aplicação de técnica de Inteligência artificial na tomada de decisão em ambiente de transação móvel. In: ENCONTRO ANUAL DE INICIAÇÃO CIENTÍFICA E ENCONTRO DE PESQUISA DA UEPG, n. 15, 2006, Ponta Grossa. **Anais ...**, Ponta Grossa: UEPG, 2006. p.13.

PAULINO, R.; FLÔR, D. E.; COSTA, Y. M. G. Inteligência Artificial Aplicada à Transação Móvel. In: ENCONTRO DE INICIAÇÃO CIENTÍFICA E FÓRUM DE PESQUISA, n. 5, 2006, Umuarama. **Anais ...**, Umuarama: Unipar, 2006. p. 78.

PITOURA, E.; BHARGAVA, B. Maintaining Consistency of Data in Mobile Distributed Environments. In: INTERNATIONAL CONFERENCE ON DISTRIBUTED COMPUTING SYSTEMS, n. 15, 1995, Vancouver. **Proceedings...**, Washigton: IEEE, 1995. p. 404.

PITOURA, E.; BHARGAVA, B. Data Consistency in Intermittently Connected Distributed Systems. **Transactions on Knowledge and Data Engineering**, v. 11, n. 6, p. 896–915, 1999.

ROCHA, T.; TOLEDO, M. B. F. **Estudo de Modelos de Transação para o ambiente de Computação Móvel**. Campinas: Instituto de Computação, Unicamp, 2005. p. 32.

ROCHA, T.; TOLEDO, M.B.F. Mecanismos de Adaptação para Transações em Ambientes de Computação Móvel. **Latin America Transactions**, vol.5, no.8, 2007.

RUSSEL, S.; NORVIG, P. **Inteligência artificial**. São Paulo: Campus, 2003.

SATYANARAYANAN, M.; KLISTER, J.; KUMAR, P.; OKASAKI, E.; SIEGEL, H.; STEERE, C. Coda: A highly available file system for distributed workstation environment. **Transactions on Computers**. v. 39, n. 4, 1990. p. 447-459

SATYANARAYANAN, M. Mobile information Access. **Personal Communications**, v. 3, n.1, 1996. p. 1-8.

SERRANO-ALVARADO, P. Defining an adaptable mobile transaction service. In: WORKSHOP ON DATA MANAGEMENT AND MULTIMEDIA ENGINEERING, 2002, Prague. London: Springer-Verlag, 2002. p. 616-626.

SERRANO-ALVARADO, P.; RONCANCIO, C.; ADIBA, M. A Survey of Mobile Transactions. **DISTRIBUTED AND PARALLEL DATABASES**, v. 16, n.2, 2004. p. 193-230.

SERRANO-ALVARADO, P. **Transactions Adaptables pour les Environments Mobiles**. 322 f. Tese (Doutorado), Université Joseph Fourier, 2004. Disponível em <http://www-lsr.imag.fr/Les.Personnes/Claudia.Roncancio/biblio.html>. Acesso em: 14. nov. 2006.

SILBERSCHATZ, A.; KORTH, H. F.; SURDASHAN, S. **Sistemas de banco de dados**. São Paulo: Makron Books, 2007. 778 p.

W3C. UAPROF. Disponível em <http://www.w3.org>. Acesso em: 12 set. 2008.

WALBORN, G. D.; CHRYSANTHIS, P. K. Supporting Semantics-Based Transaction Processing in Mobile Database Applications. In: SYMPOSIUM ON RELIABLE DISTRIBUTED SYSTEMS, n. 14, Bad Neuenahr, 1995. **Proceedings ...**, Washington: IEEE, 1995. p. 31.

WALBORN, G. D.; CHRYSANTHIS, P. K. PRO-MOTION: Management of Mobile Transactions. In: SYMPOSIUM ON APPLIED COMPUTING, San Jose, 1997. **Proceedings ...**, New York: ACM, 1997. p. 101-108.

WALBORN, G. D.; CHRYSANTHIS, P. K. Transaction Processing in PRO-MOTION. SYMPOSIUM ON APPLIED COMPUTING, San Antonio, 1999. **Proceedings ...**, New York: ACM, 1999. p.389-398.

WEIKUM, G.; SCHEK, H. Concepts and Applications of Multilevel Transactions and Open Nested Transactions. In: ELMAGARMID, A. **Database Transaction Models for Advanced Applications**. Morgan Kaufmann, 1992. cap. 13.

Disponível em

<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.17.7962>. Acesso em: 19 out. 2008.

SUN MICROSYSTEMS. MVC2. Disponível em

<http://java.sun.com/products/jfc/tsc/articles>. Acesso em: 17 set. 2008.

YEO, L. H.; ZASLAVSKY, A. Submission of Transactions from Mobile Workstations in a Cooperative Multidatabase Processing Environment. In: INTERNACIONAL CONFERENCE ON DISTRIBUTED COMPUTING SYSTEMS, n. 14, Poznan, 1994. **Proceedings ...**, ..., Washington: IEEE, 1994. p. 372-379

A APÊNDICE A – SERIALIZAÇÃO

Por razões de desempenho, os sistemas de banco de dados permitem a execução concorrente de várias transações. A execução de transações concorrentes não pode interferir na consistência dos dados. Dessa forma, são necessárias técnicas que impeçam tal interferência. A serialização propõe o escalonamento das operações de transações concorrentes em execução, como mecanismo de garantia da correitude dos resultados obtidos.

Mas nem todas as escalas de execução concorrente produzem o mesmo resultado da execução serial. Com o intuito de exemplificar tal afirmação, Silberschatz (2007) utiliza duas transações, T1 e T2. A Figura A.1 apresenta as operações das transações executadas serialmente.

T1	T2
read(A)	
A := A - 50	

write(A) read(B) B := B + 50 write(B)	read(A) temp := A * 0,1 A := A - temp write(A) read(B) B := B + temp write(B)
--	---

Figura A.1 – Escala de execução serial na qual T1 é seguida por T2

Sejam R\$ 1.000,00 e R\$ 2.000,00 reais os valores das contas A e B, ao término da aplicação da escala de execução serial, os resultados passam a ser respectivamente R\$ 855,00 e R\$ 2.145,00.

Caso T1 e T2 sejam executadas concorrentemente, algumas escalas são possíveis. A Figura A.2, retirada de Silberschatz (2007), apresenta uma escala de execução que resulta nos mesmos valores da escala serial da Figura A.1.

T1	T2
read(A) A := A - 50 write(A)	read(A) temp := A * 0,1 A := A - temp write(A)
read(B) B := B + 50 write(B)	read(B) B := B + temp write(B)

Figura A.2 – Escala de execução concorrente equivalente à escala seqüencial

Silberschatz (2007) também apresenta um exemplo de escala de execução não equivalente exibido na Figura A.3.

T1	T2

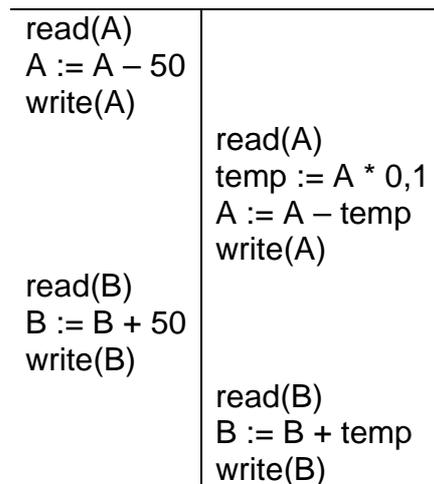


Figura A.3 – Escala de execução concorrente não equivalente à escala seqüencial

Neste caso, os resultados finais da escala de execução da Figura A.3 passam a ser R\$ 950,00 e R\$ 2.100,00. Os resultados da escala da Figura A.2 são os mesmos obtidos pela escala serial, ou seja, a serialização da escala da Figura A.2 é aceitável e a da Figura A.3 não. As técnicas mais divulgadas para avaliar as escalas de execução possíveis são as de serialização de conflito e visão serializada.

A.1 Serialização de Conflito

Uma escala com serialização de conflito permite a troca de operações conflitantes até alcançar uma escala de execução seqüencial. A equivalência de visão considera que uma escala com visão serializável tem visão equivalente a um plano serial. Uma escala visão serializável observa que em ambas as escalas cada leitura de uma transação enxerga o resultado de uma mesma escrita e por último escreve sobre o mesmo item de dados.

A.2 Visão Serializável

A visão serializável é menos restritiva que o conflito serializável. Um escalonamento serializável em conflito também é serializável em visão, porém a recíproca nem sempre é verdadeira. Avaliar a serialização das escalas em um ambiente que recebe transações ininterruptamente é inviável, por isso

outras técnicas de controle de concorrência são mais comumente utilizadas.

B APÊNDICE B – DIAGRAMAS DE SEQÜÊNCIA DE BANCOONLINE E BANCOME

As transações disponíveis em BancoOnline e BancoME foram modeladas por diagramas de seqüência. O intuito é facilitar a visualização dos métodos que compõem as aplicações e as chamadas aos serviços do MiD-Mobile. A Figura B.1 apresenta o diagrama de seqüência do pagamento de contas, também é possível verificar a realização dos lançamentos originados, desde que o correntista tenha o valor informado em saldo.

O diagrama da figura B.2 mostra as operações de simulação e empréstimo. A primeira ação do correntista é fazer a simulação, caso os valores sejam atrativos, o empréstimo pode ser efetivado.

Todas as operações executadas no BancoOnline e no BancoME podem ser visualizadas a partir da operação de extrato. O diagrama da Figura B.3 apresenta a interação entre os métodos que informam o extrato.

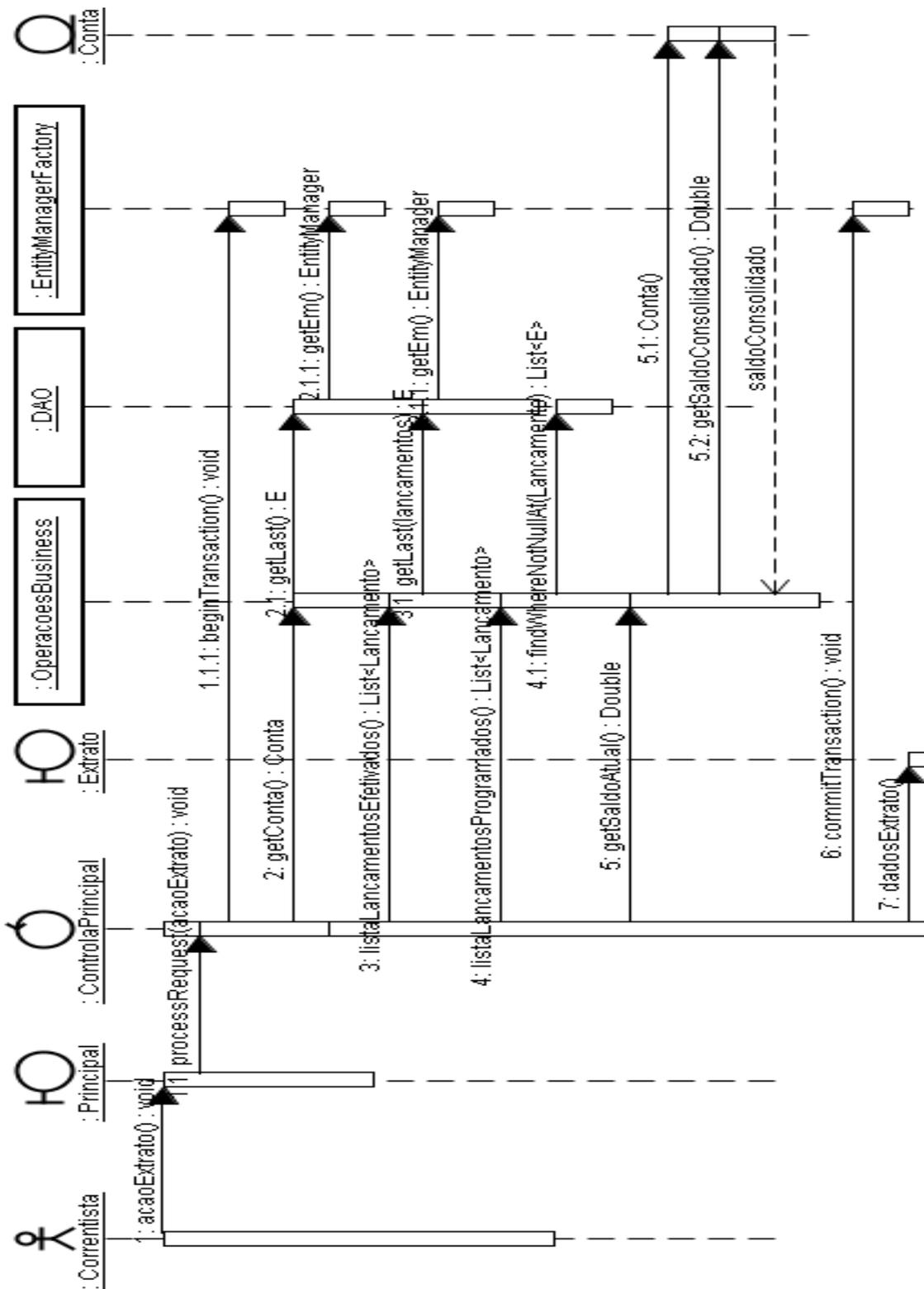


Figura B.1 – Diagrama de seqüência da operação de extrato

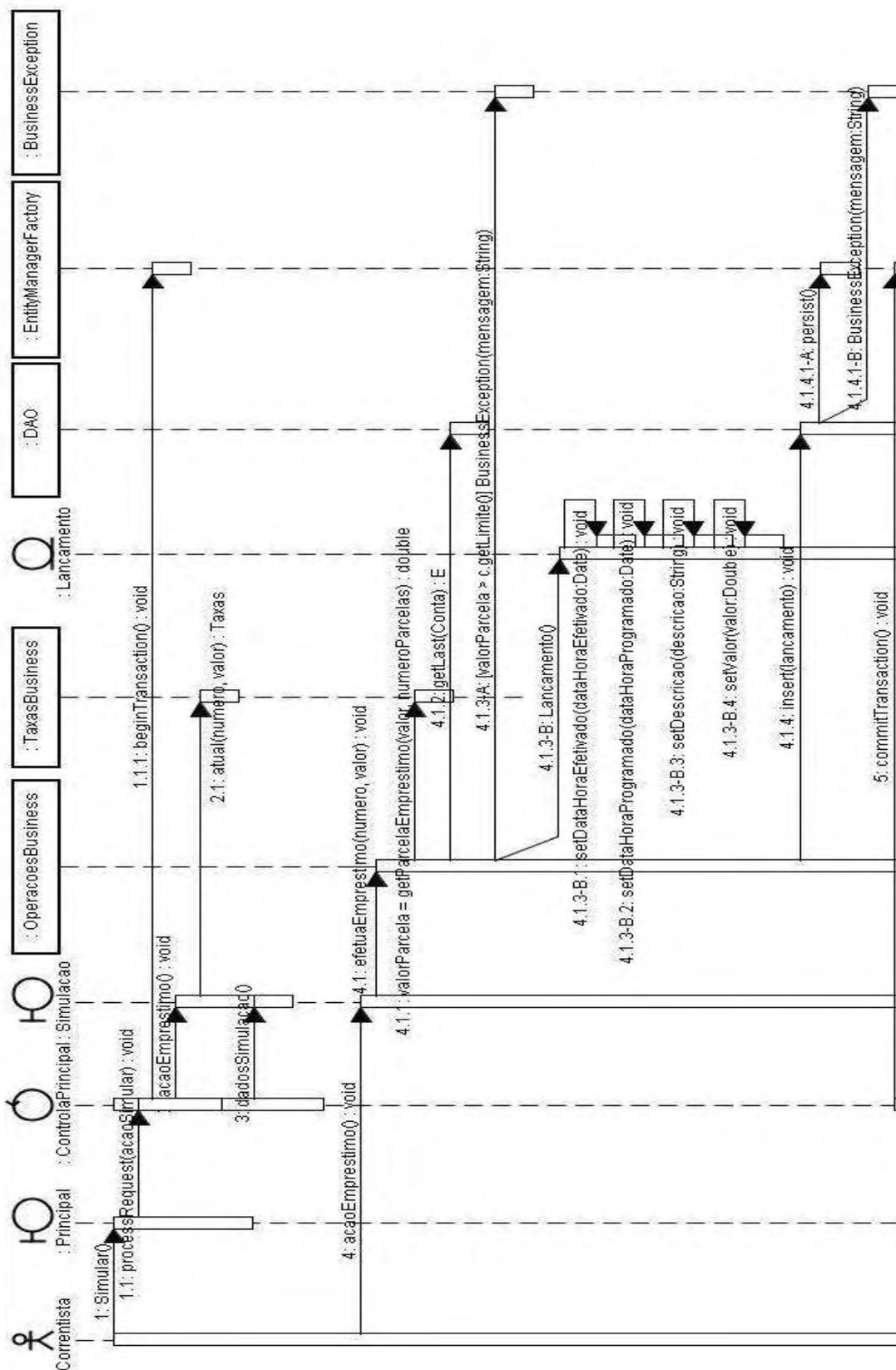


Figura B.2 – Diagrama de seqüência da operação de simulação e empréstimo

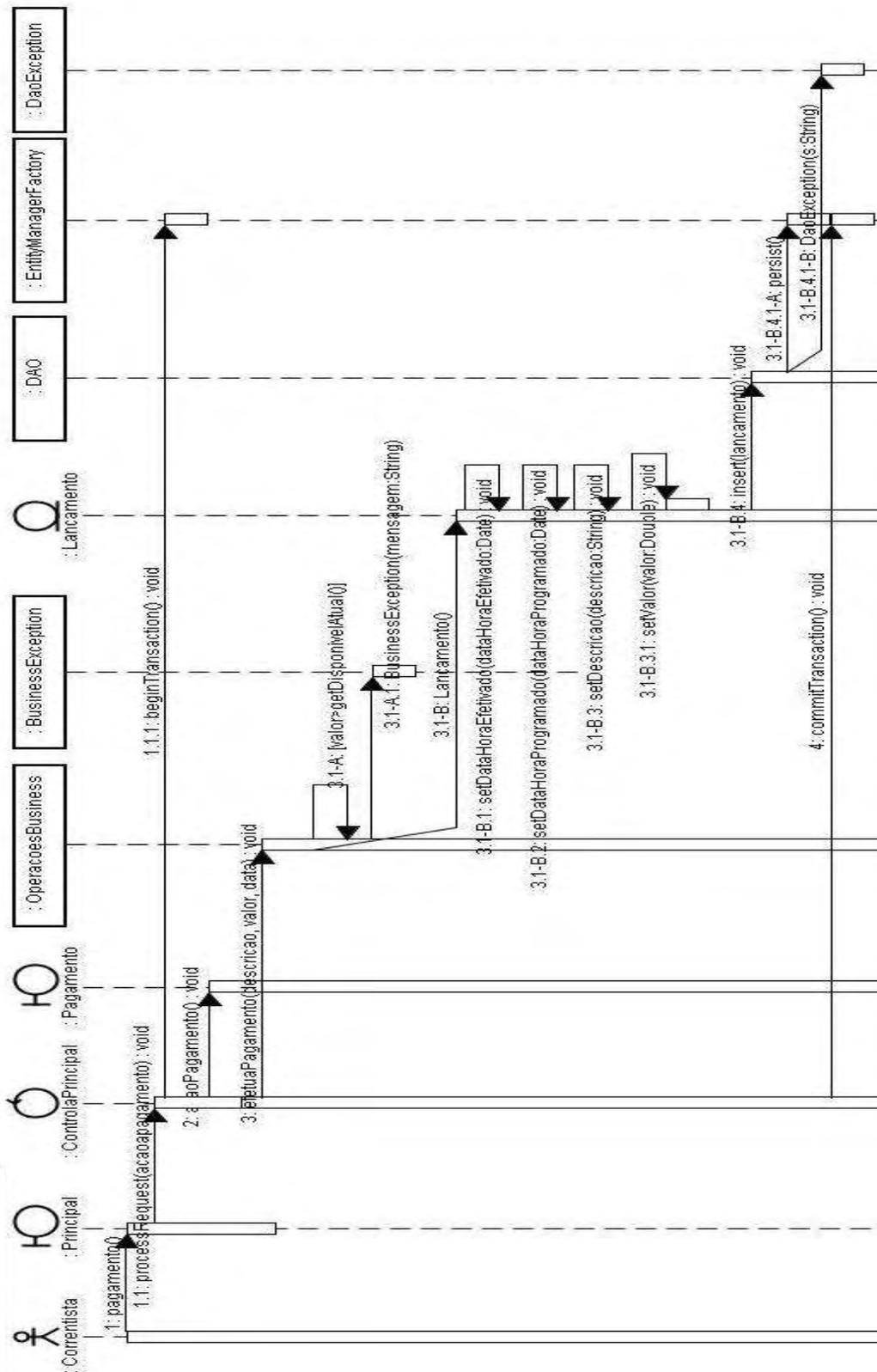


Figura B.3 – Diagrama de seqüência da operação de pagamento

Diagramas de seqüência também foram empregados para viabilizar o acompanhamento dos possíveis planos de execução das operações de BancoOnline e BancoME.

No plano de execução remota, as classes pertencentes ao dispositivo móvel estão marcadas com o estereótipo <<movel>>, as disponíveis na rede fixa são identificadas por <<fixo>>. A Figura B.4 apresenta o exemplo da transação de pagamento, a Figura B.5 trata a simulação e a efetivação do empréstimo. O último diagrama é o do extrato da Figura B.6.

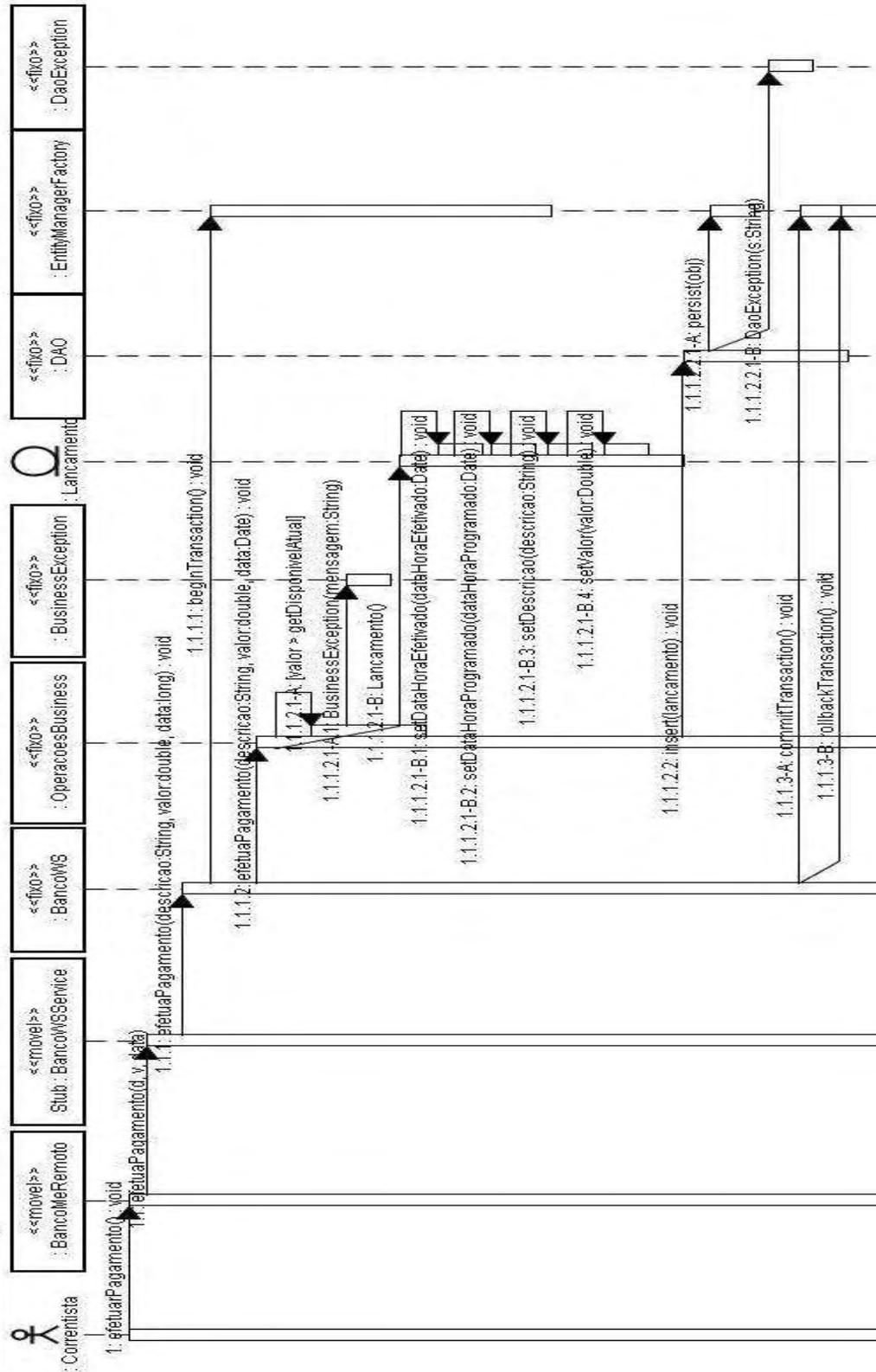


Figura B.4 – Diagrama de seqüência de pagamento no plano de execução remota

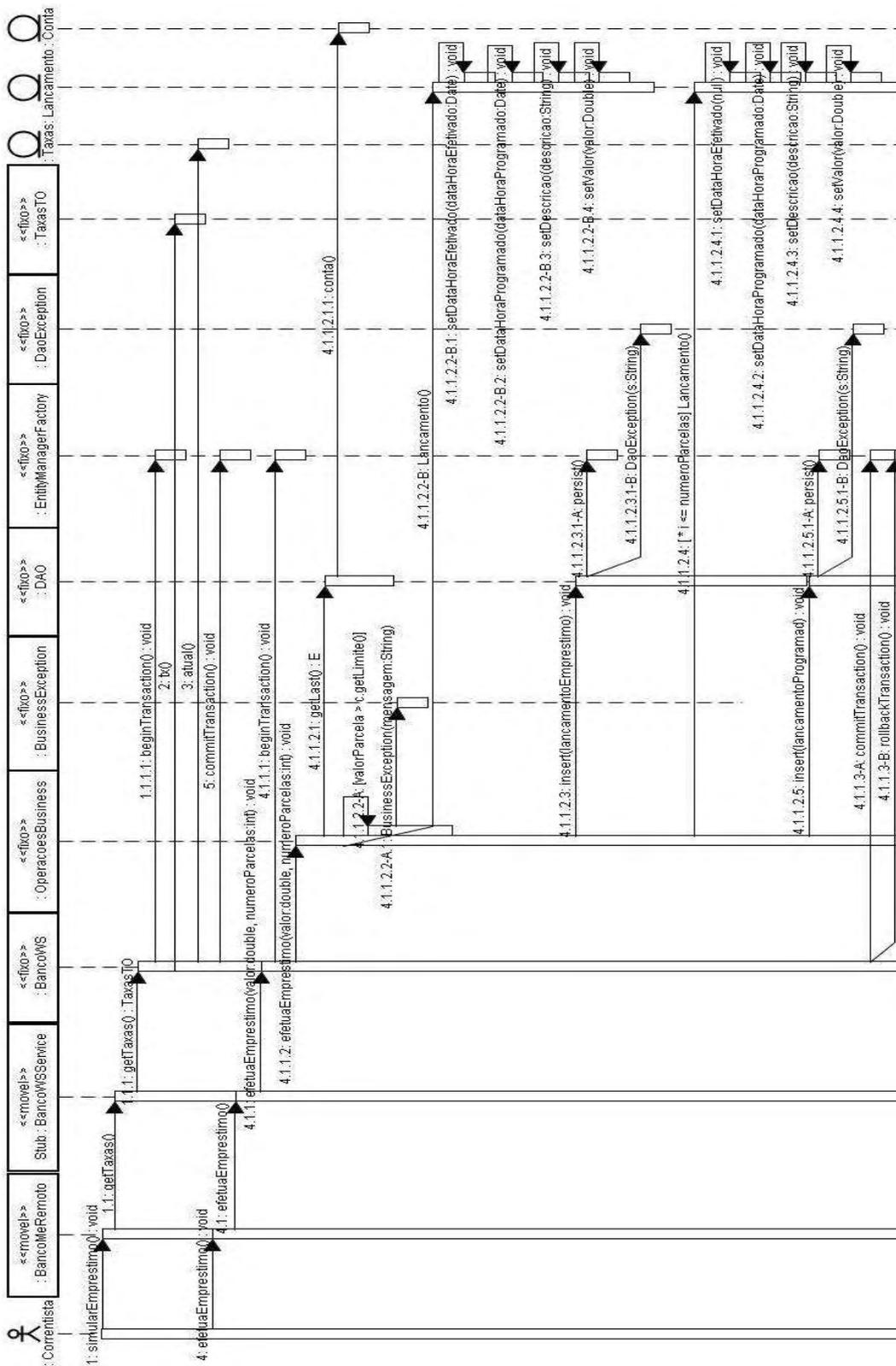


Figura B.5 – Diagrama de seqüência de simulação e empréstimo no plano de execução remota

O diagrama de seqüência da Figura B.7 apresenta a execução da operação de extrato, explorando somente classes disponíveis no dispositivo móvel. A Figura B.8 trata a operação de extrato e a Figura B.9 a execução da simulação e da efetivação do empréstimo.

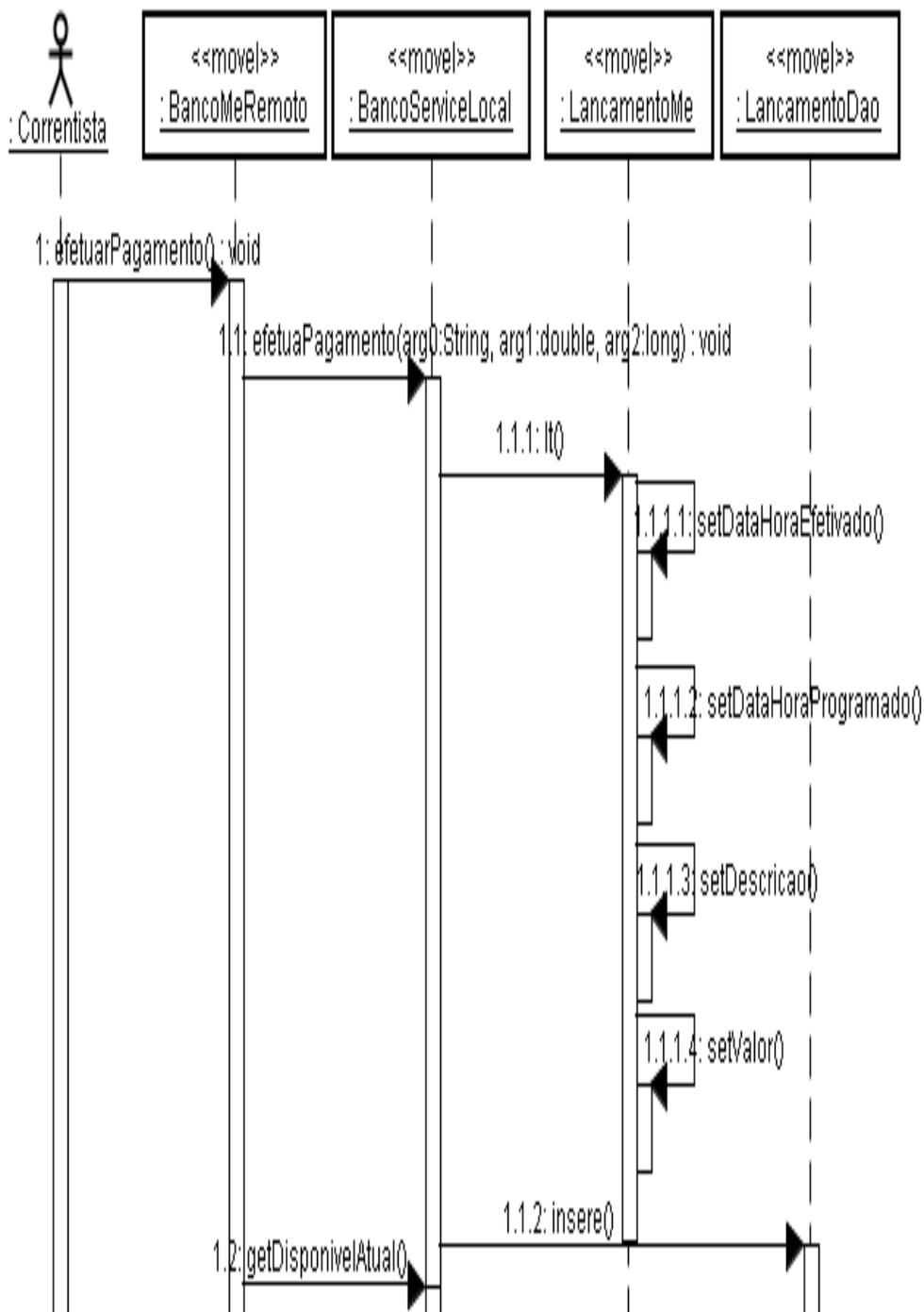


Figura B.7 – Diagrama de seqüência de pagamento no plano de execução isolada

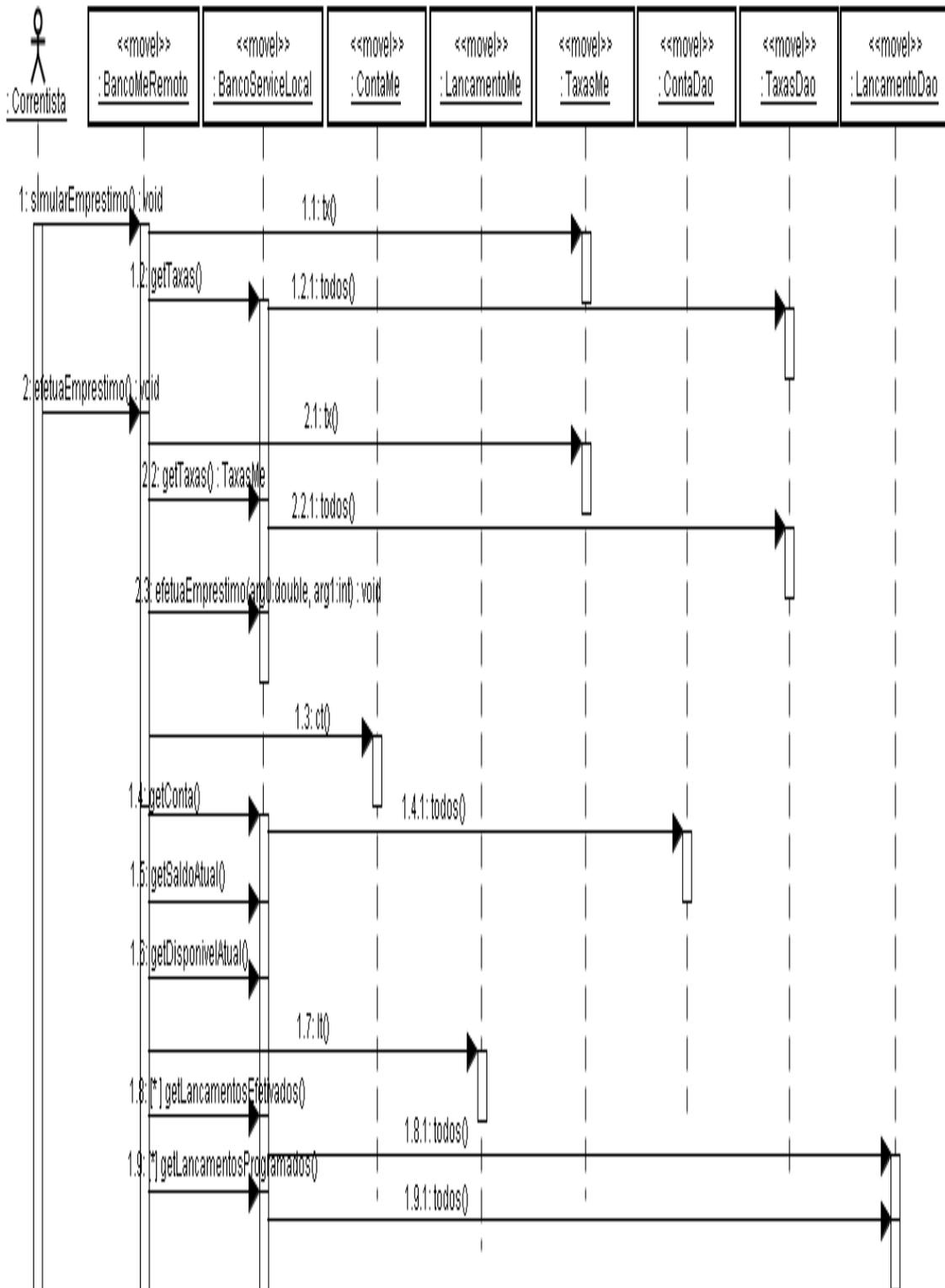


Figura B.8 – Diagrama de seqüência de simulação e empréstimo no plano de execução isolada

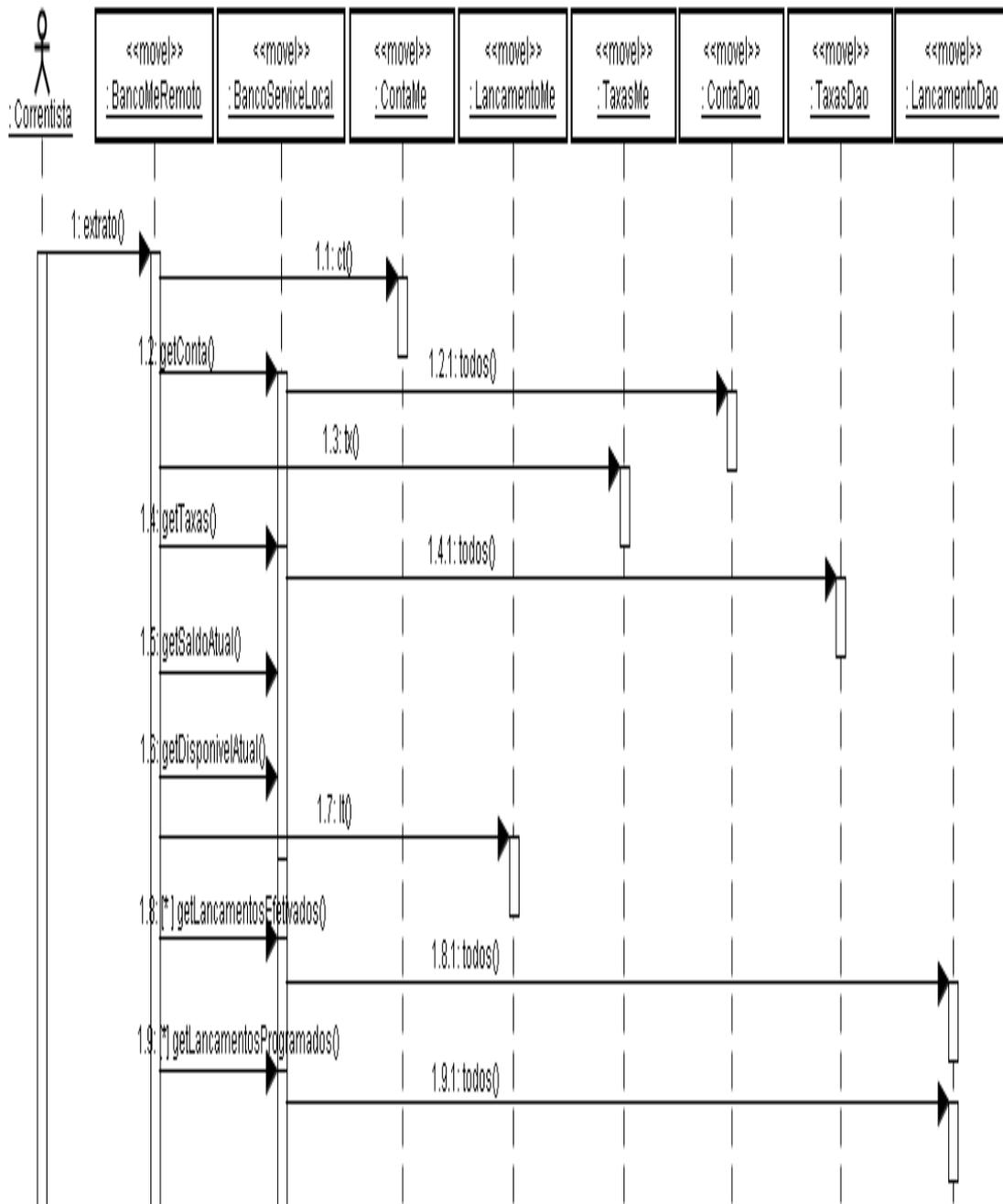


Figura B.9 – Diagrama de seqüência de extrato no plano de execução isolada

Os diagramas de seqüência das Figuras B.10, B.11 e B.12 apresentam respectivamente a execução compartilhada das operações de extrato, pagamento, simulação e empréstimo. A classe BancoCompartilhado gerencia as chamadas à BancoMe ou à BancoOnline.

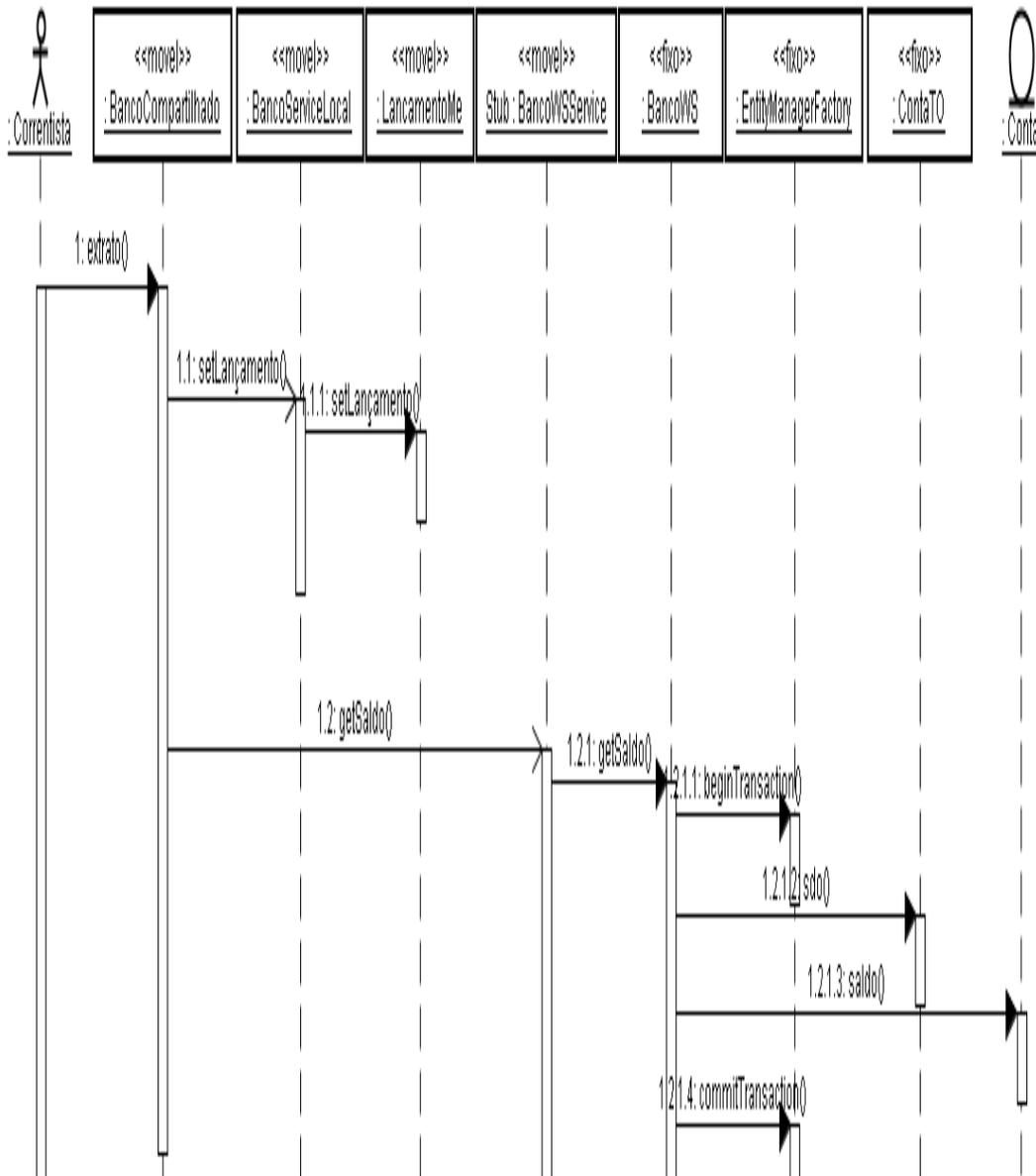


Figura B.10 – Diagrama de seqüência de extrato no plano de execução compartilhada

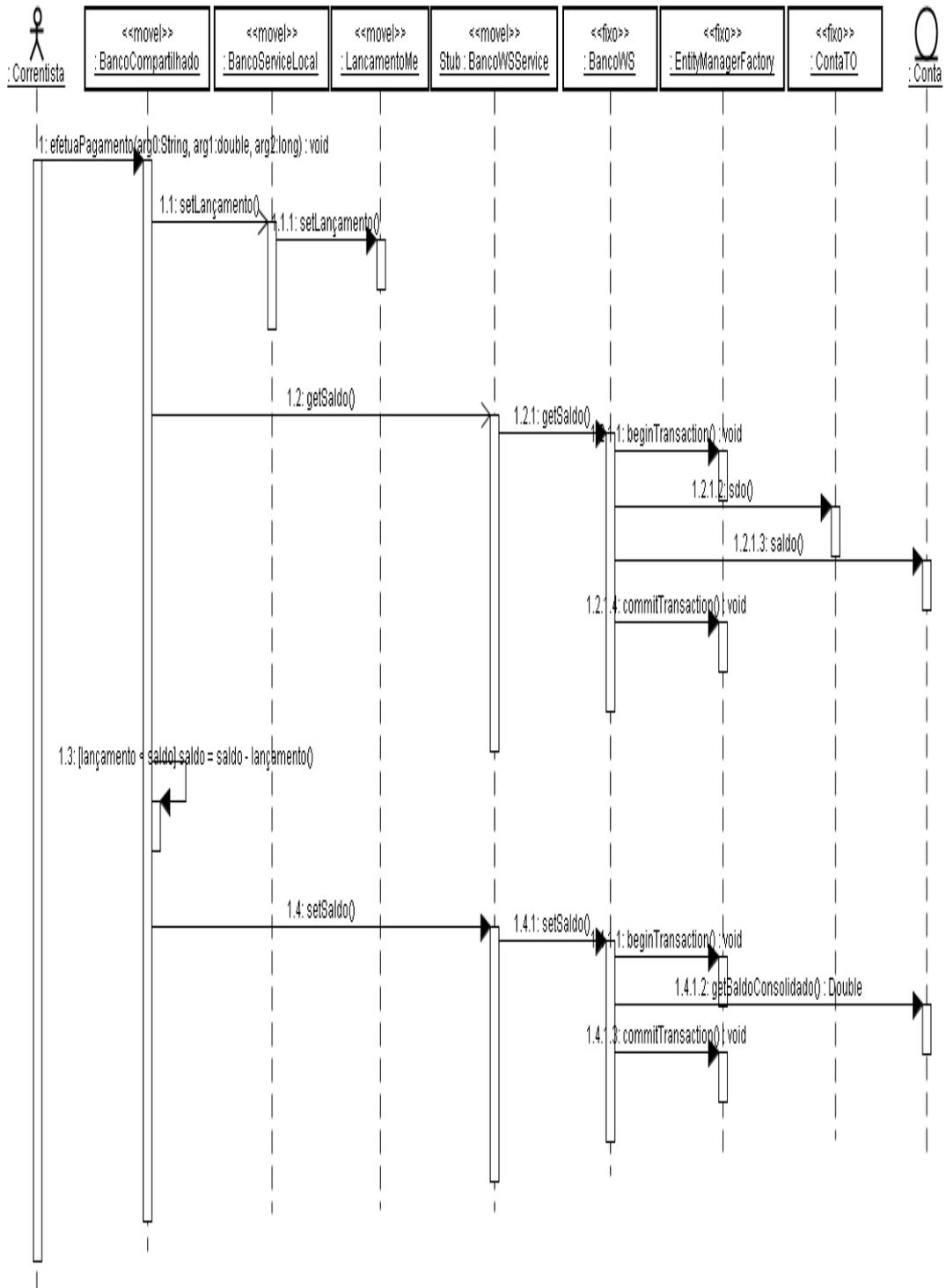


Figura B.11 – Diagrama de seqüência de pagamento no plano de execução compartilhada

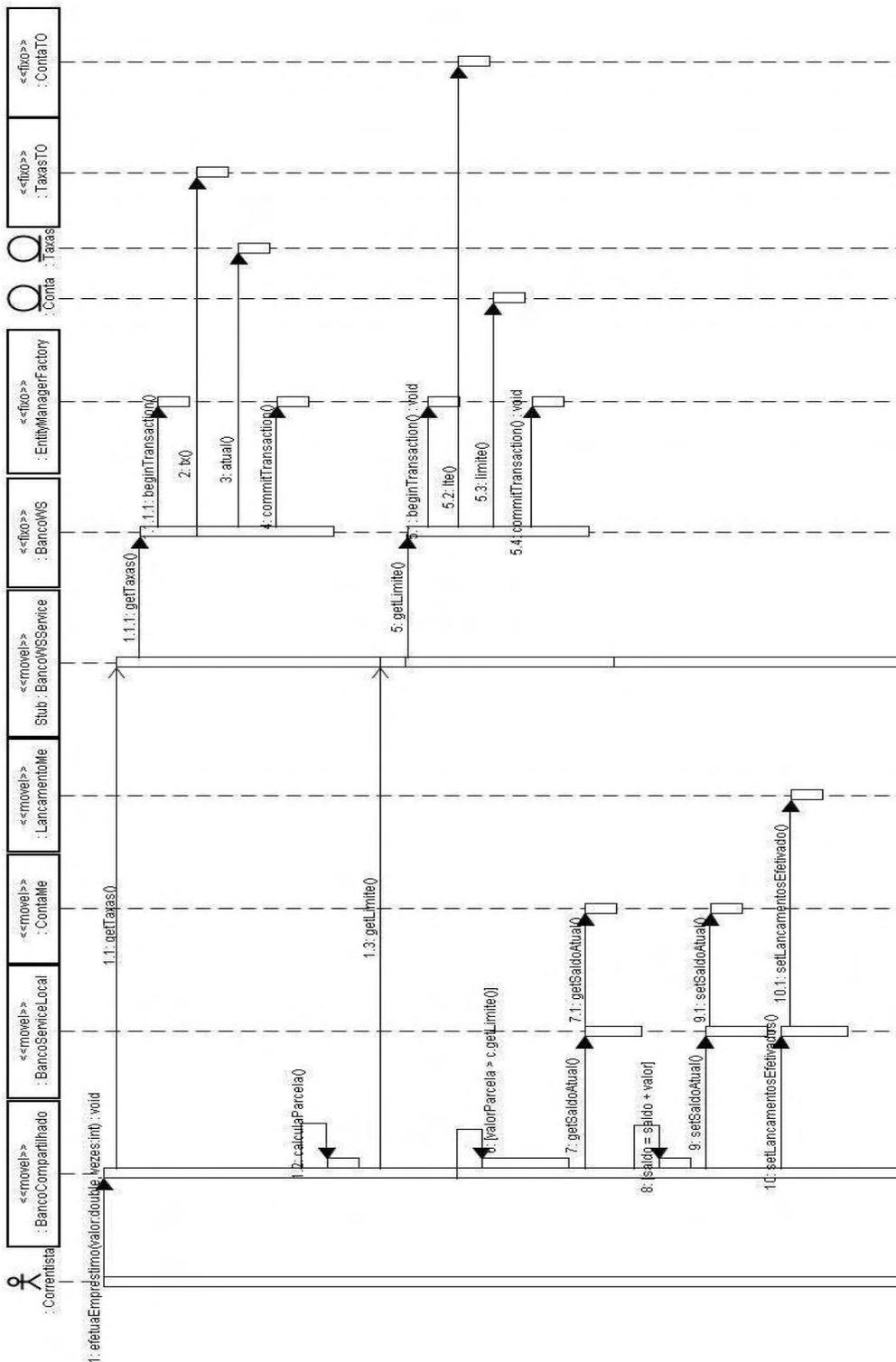


Figura B.12 – Diagrama de seqüência de simulação e empréstimo no plano de execução compartilhada

PUBLICAÇÕES TÉCNICO-CIENTÍFICAS EDITADAS PELO INPE

Teses e Dissertações (TDI)

Teses e Dissertações apresentadas nos Cursos de Pós-Graduação do INPE.

Manuais Técnicos (MAN)

São publicações de caráter técnico que incluem normas, procedimentos, instruções e orientações.

Notas Técnico-Científicas (NTC)

Incluem resultados preliminares de pesquisa, descrição de equipamentos, descrição e ou documentação de programa de computador, descrição de sistemas e experimentos, apresentação de testes, dados, atlas, e documentação de projetos de engenharia.

Relatórios de Pesquisa (RPQ)

Reportam resultados ou progressos de pesquisas tanto de natureza técnica quanto científica, cujo nível seja compatível com o de uma publicação em periódico nacional ou internacional.

Propostas e Relatórios de Projetos (PRP)

São propostas de projetos técnico-científicos e relatórios de acompanhamento de projetos, atividades e convênios.

Publicações Didáticas (PUD)

Incluem apostilas, notas de aula e manuais didáticos.

Publicações Seriadas

São os seriados técnico-científicos: boletins, periódicos, anuários e anais de eventos (simpósios e congressos). Constam destas publicações o Internacional Standard Serial Number (ISSN), que é um código único e definitivo para identificação de títulos de seriados.

Programas de Computador (PDC)

São a seqüência de instruções ou códigos, expressos em uma linguagem de programação compilada ou interpretada, a ser executada por um computador para alcançar um determinado objetivo. São aceitos tanto programas fonte quanto executáveis.

Pré-publicações (PRE)

Todos os artigos publicados em periódicos, anais e como capítulos de livros.