

ANDERSON PAES

DESENVOLVIMENTO DE *FRAMEWORK* PARA SIMULAÇÃO DE
DISPOSITIVOS ADAPTATIVOS

Assis

2008

DESENVOLVIMENTO DE *FRAMEWORK* PARA SIMULAÇÃO DE
DISPOSITIVOS ADAPTATIVOS

ANDERSON PAES

Trabalho de Conclusão de Curso apresentado ao
Instituto Municipal de Ensino Superior de Assis,
como requisito do Curso de Graduação, analisado
pela seguinte comissão examinadora:

Orientador: Prof. Almir Rogério Camolesi

Analisador (1): _____

Analisador (2): _____

Assis

2008

ANDERSON PAES

DESENVOLVIMENTO DE *FRAMEWORK* PARA SIMULAÇÃO DE
DISPOSITIVOS ADAPTATIVOS

Trabalho de Conclusão de Curso apresentado ao
Instituto Municipal de Ensino Superior de Assis,
como requisito do Curso de Graduação, analisado
pela seguinte comissão examinadora:

Orientador: Prof. Almir Rogério Camolesi
Área de Concentração: Tecnologia Adaptativa

Assis
2008

DEDICATÓRIA

Dedico este trabalho
a Deus, que me deu forças, coragem e inspiração;
e os meus pais, que me apoiaram muito para que eu pudesse alcançar meus
objetivos e realizar os meus grandes sonhos.

AGRADECIMENTOS

Ao professor, Almir Rogério Camolesi, que apostou em mim desde começo do curso, mesmo quando às vezes eu não demonstrava tanto estímulo pelos estudos, mesmo assim ele apostou em mim, pois sabia que eu podia ser um grande profissional da computação e pela orientação e pelo constante estímulo transmitido durante o trabalho.

Aos meus amigos de classe, pela amizade, dedicação e apoio que nos momentos mais difíceis que eu passei nos estudos, pensando até em desistir, me ajudaram e me incentivaram a continuar os estudos e a todos que colaboraram diretamente ou indiretamente, na execução deste trabalho.

Aos meus familiares e amigos, pela compreensão dos momentos de minha ausência, pois esses cinco anos foram muito corridos e quase nunca pude estar junto com eles.

A minha mãe Luiza Venâncio Paes, que nesses cinco anos e toda a minha vida me amou, me acariciou e sempre se preocupou comigo, perguntando se eu estava bem nos estudos e se eu precisava de alguma coisa e se ela poderia me ajudar. Esperava-me até altas horas para abrir o portão para mim quando chegasse da faculdade.

Ao meu pai Antonio Arari Paes, que o seu sonho era de que eu estudasse, e nos momentos tão difíceis que passei sempre me apoiou e incentivou com palavras sábias que jamais vou esquecer.

À Fundação Educacional do Município de Assis (FEMA) e principalmente os professores, que me ajudaram muito a ganhar os conhecimentos profissionais e morais de um grande profissional da área de tecnologia e computação e também ao grande apoio prestado para conclusão deste trabalho.

RESUMO

Este trabalho tem por objetivo mostrar o processo de desenvolvimento da tecnologia adaptativa que é considerada uma das maiores arquiteturas nos processos de aplicação. Com esse interesse é formalizado um conjunto de classes para simular dispositivos adaptativos, ou seja, um framework para controlar o fluxo da aplicação especificando a forma de operação dos dispositivos adaptativos. Tal *framework* permitirá a um especialista utilizar uma linguagem de mais alto nível e representar o seu formalismo sem ter a necessidade de realizar codificação de programas, pois a maior necessidade da tecnologia adaptativa está na compreensão adequada do funcionalismo da mesma, então por meio de um simulador adaptativo e será realizado a execução de dispositivos adaptativos.

Palavras-chave: Tecnologia Adaptativa, dispositivos adaptativos, modelagem de sistemas, ferramentas de modelagem, programação.

ABSTRACT

This work has the objective of show the process of development of the adaptative technology that is one of the largest architectures in the application processes. With that interest a group of classes is formalized to simulate adaptatives devices, in other words, a *framework* to control the flow of the application specifying the form of operation of the adaptatives devices. Such framework will allow a specialist to use a language of higher level and to represent yours formalism without having the need to accomplish code of programs, because the largest need of the adaptative technology is in the appropriate understanding of the functionalism of the same, then through a adaptative simulator the execution of adaptative devices is accomplished.

Keywords: Adaptative technology, adaptatives devices, modeling of systems, modeling tools, programming.

LISTA DE ILUSTRAÇÕES

Figura 1 - Método para definição de dispositivos adaptativos.....	18
Figura 2 - <i>Framework</i> adaptativo na arquitetura do processo de..... aplicação adaptativa	22
Figura 3 - <i>Framework</i> e o modelo lógico no processo de..... aplicação adaptativa	23
Figura 4 - Aplicação do <i>framework</i> proposto.....	24
Figura 5 - Diagrama de seqüência para o simulador adaptativo.....	26
Figura 6 - Diagrama de execução do <i>framework</i> adaptativo.....	28
Figura 7 - Diagrama de classes adaptado para o do <i>framework</i> adaptativo.....	30
Figura 8 - Diagrama de execução da simulação adaptativa.....	33
Figura 9 – Criação dos Objetos e suas classes.....	38
Figura 10 – Criação dos pacotes do <i>Hibernate</i> e geração do banco de dados.....	39
Figura 11 – Criação do pacote DAO e as classes utilizadas para simulação.....	40
Figura 12 – O novo código em destaque de vermelho e o antigo em destaque..... de azul.	41

SUMÁRIO

1. INTRODUÇÃO	10
2. TECNOLOGIA ADAPTATIVA	12
2.1. FORMALISMO PARA DISPOSITIVOS NÃO-ADAPTATIVOS	15
2.2. FORMALISMO PARA DISPOSITIVOS ADAPTATIVOS	15
3. MODELO LÓGICO PARA DISPOSITIVOS ADAPTATIVOS	16
3.1. ESPECIFICAÇÃO DE CLASSES DA CAMADA DE DISPOSITIVO	19
3.2. ESPECIFICAÇÃO DE CLASSES DA CAMADA SUBJACENTE	20
3.3. ESPECIFICAÇÃO DE CLASSES DA CAMADA ADAPTATIVA	20
3.4. ESPECIFICAÇÃO DE CLASSES DA CAMADA DE MEMÓRIA	21
4. PROPOSTA DO AMBIENTE DE EXECUÇÃO ADAPTATIVO	21
5. <i>FRAMEWORK</i> PARA SIMULAÇÃO DE DISPOSITIVOS ADAPTATIVOS	24
6. IMPLEMENTAÇÃO DO <i>FRAMEWORK</i>	35
7. CONCLUSÃO	35
7.1. JUSTIFICATIVAS E MOTIVAÇÕES	36
7.2. OBJETIVOS	36
7.3. OBJETIVOS	37
8. REFERÊNCIAS BIBLIOGRAFICAS	37

1. INTRODUÇÃO

Segundo Camolesi (2007) uma das áreas em que a Tecnologia Adaptativa é utilizada é na modelagem de aplicações. Por exemplo, um projetista pode utilizar-se de um modelo de uma aplicação utilizando os conceitos de autômatos finitos e realizar testes com uma determinada cadeia de entrada para simular ou verificar inconsistências de sua aplicação. A mesma aplicação pode ser também modelada utilizando-se de outros formalismos, por exemplo, uma Rede de Petri Cardoso (1997). Pode-se, então, verificar que estes formalismos possuem um conjunto elementar que representam os mesmos.

Neto (2001) nos traz um estudo que demonstra uma representação geral para formalismos adaptativos dirigidos por regras. Neste estudo é definido um modelo lógico que permite representar diversos dispositivos adaptativos.

Este trabalho encontra-se organizado da seguinte forma: o capítulo 3 mostra os conceitos e realizações dos procedimentos do modelo lógico do ambiente adaptativo; no capítulo 4, é apresentado um resumo do trabalho de Camolesi, (2007). Neste trabalho foi apresentado a arquitetura de um ambiente geral e um pequeno ambiente para a modelagem de aplicações utilizando-se dos dispositivos que foram mapeados para tal estrutura.

No capítulo 5, é descrita a formalização de um *framework* para especificação da forma de operação dos dispositivos adaptativos. Esse *framework* permitirá a um especialista utilizar-se de uma linguagem de alto nível e representar o seu formalismo sem ter a necessidade de realizar codificação de programas por meio de uma linguagem de programação, forma esta, que está sendo utilizada. Como consequência, esse método ajudará a especificar melhor como são realizados os processos dos dispositivos, facilitando o aprendizado e concedendo uma forma mais fácil de compreender o dispositivo que é estudado. Desta forma, os especialistas em dispositivos adaptativos poderão dispendir um maior tempo no estudo dos problemas propostos que geralmente são difíceis de serem compreendidos.

O *framework* mostra que, por meio de um simulador, é possível demonstrar que vários dispositivos podem chegar a um resultado igual. Mesmo que suas

características sejam diferentes percebe-se que não se trata apenas de uma pequena simulação, mas sim de uma metodologia que pode resolver muitos tipos de funções e serviços.

Com isso surge uma grande motivação, pois se trata de uma tecnologia muito ampla e que há muito a ser desenvolvido. Uma tecnologia cujo o domínio está envolvendo várias teorias para fornecer os fundamentos matemáticos, ferramenta que facilitem o desenvolvimento de novas aplicações e aplicações que resolvam problemas em diversas áreas que necessitam de tecnologia, pois a adaptabilidade é a forma na qual um dispositivo pode alterar o seu próprio comportamento, de forma espontânea sem o auxílio de qualquer outro dispositivo. O desenvolvimento desse *framework* para simulação de dispositivos adaptativos tem o propósito de resolver problemas acadêmicos, mas com decorrer do tempo, também poderá ser desenvolvido para execução em serviços de empresas.

O objetivo é de se construir um *framework* para simulação de dispositivos adaptativos e com isso mostrar como ocorre a execução de uma aplicação modelada utilizando-se conceitos de adaptabilidade. Tal modelagem deverá ser realizada em uma meta ferramenta que recebe os estímulos de uma cadeia de elementos de entrada, realiza o seu processamento e determina se esta cadeia de entrada pode ser reconhecida e adaptada a um conjunto de regras e métodos que já são estabelecidos no ambiente adaptativo, com o intuito de alcançar um resultado final.

Além disso, o trabalho irá definir uma ferramenta para especificar a forma de operação de um determinado dispositivo adaptativo. Também será apresentada uma modelagem orientada a objetos que permitirá executar o simulador adaptativo com o objetivo de mostrar a sua aplicação na definição de um novo dispositivo adaptativo.

O *framework* proposto deve permitir a simulação de aplicações modeladas com base nos dispositivos adaptativos que foram mapeados para o modelo lógico proposto.

O capítulo 6 mostra as conclusões finais mostrando os trabalhos realizados, as principais contribuições e os trabalhos futuros.

2. TECNOLOGIA ADAPTATIVA

Na palavra adaptabilidade encontram-se inúmeros conceitos aplicados, mas para o contexto da Tecnologia Adaptativa refere-se ao modelo de se auto-adaptar, ou seja, um determinado dispositivo que modifique o seu próprio comportamento, de forma espontânea, sem o auxílio de qualquer outro tipo de ferramenta.

O conceito da Tecnologia adaptativa é representado por intermédio de um formalismo capaz de representar linguagens sensíveis ao contexto. Esta tecnologia está relacionado à sua gramática, que diferente das convencionais, tem a capacidade de se auto-modificar à medida que uma sentença da linguagem vai sendo derivada.

A origem dessa tecnologia se deu da notação científica dos autômatos e sua formalização é designada pelos conceitos de autômatos finitos. Com as linguagens regulares toda a base é realizada por meio de estados, transições e uma cadeia de entrada, para formalizar os conceitos e fundamentos da parte de autômatos finitos. Por sua vez, esse contexto pode chegar aos autômatos de pilha estruturados que, com as linguagens livres de contexto, geram sub-máquinas que formalizam os autômatos de pilha estruturados.

Tendo assim todos os conceitos aplicados de autômatos finitos e autômatos de pilha estruturados, chega-se a conclusão que, por meio de funções adaptativas e alterações na cadeia de entrada, são formalizados o conceito dos autômatos adaptativos. A linguagem agora dependerá do contexto para que possa dirigir-se no caminho correto do autômato, tornando assim, a sua linguagem sensível ao contexto.

O domínio desta Tecnologia requer o conhecimento de três vertentes, de acordo com Neto (2001): desenvolvimento de conceitos e de fundamentos matemáticos, para a Adaptatividade; criação de Métodos, ferramentas e ambientes para o desenvolvimento de aplicações adaptativas e a aplicação da Tecnologia Adaptativa para a resolução de problemas em variadas áreas de interesse.

Um dos primeiros estudos realizados com Tecnologia Adaptativa foi demonstrado em (NETO e MAGALHÃES, 1981), no qual demonstra métodos de análise sintática e geradores de reconhecimento sintático. Esse trabalho buscou introduzir os conceitos de mecanismos adaptativos para a construção de compiladores.

Em Neto (1988), teve a incorporação de funções de transdução sintática. Com continuação do trabalho foi apresentado um aperfeiçoamento de reconhecedores sintáticos, baseando-se nos autômatos pilha estruturada. Em Neto (1988), é apresentado um transdutor adaptativo que se refere a uma classe de máquinas de estados finitos armazenado em uma pilha e com memória organizada que demonstra uma alteração dinâmica de sua configuração. De acordo com o aprendizado efetuado nas transições pelo transdutor, ele apresenta um poder maior na representação dos modelos matemáticos.

Com o trabalho de Neto (1993), foram aplicados outros trabalhos de tecnologia adaptativa em projetos de sistemas reativos, o que proporcionou o autômato e o transdutor adaptativo funcionarem como dispositivos de transdução sintática e de reconhecimento.

Em Camolesi e Neto (2004) foi formulada a Rede de Petri Adaptativa (RPA_{dp}), que, por meio de um conjunto de regras, pode modificar o seu comportamento automaticamente sem o auxílio de outra ferramenta ou processo.

Com a apresentação do AdapTree - um algoritmo de árvores de decisão em Pistori e Neto (2002), formaliza-se os conceitos da Tecnologia Adaptativa que é capaz de formalizar informações captadas na entrada e, que, por intermédio de dados fornecidos, o algoritmo do AdapTree pode reconhecer as informações mais adequadas para cada determinado tipo de situação e, desta maneira, fornecer um dos melhores conceitos de aprendizado automático através de um algoritmo adaptável.

Camolesi (2007) proporciona um gerador de ambientes (meta-ambientes) que possibilita gerar automaticamente projetos de aplicações adaptativas. Ele fundamenta-se nos conceitos da Tecnologia Adaptativa permitindo que os dispositivos possam ser redigidos por regras proporcionando um formalismo que

simula e gera automaticamente um ambiente para o desenvolvimento dos projetos em questão.

Os principais formalismos adaptativos que têm sido pesquisados no LTA incluem:

- Statecharts Adaptativos /Statecharts Adaptativos Sincronizados;
- Redes de Markov Adaptativas;
- Gramáticas Adaptativas;
- Tabelas de Decisão Adaptativas;
- Árvores de Decisão Adaptativas;
- Autômatos Finitos Adaptativos;
- Redes de Petri Adaptativas;
- Máquinas de Turing Adaptativas;
- Linguagens de Programação Adaptativas;

Os formalismos estudados resultaram em publicações, ferramentas e aplicações para a Tecnologia Adaptativa.

Por sua vasta aplicação e simplicidade os dispositivos adaptativos acoplam muitas áreas:

- Teoria: através dos modelos de computação baseado em autômatos adaptativos formalizam os conceitos adaptativos;
- Linguagens: estabelecidas para organizar as bases para um paradigma de programação aderente à tecnologia adaptativa;
- Inteligência Artificial: modelos adaptativos para representação e manipulação do conhecimento automático, ou seja, um dispositivo capaz de auto aprender através de informações pré-estipuladas e um algoritmo de tomadas de decisões.

- Tomada de decisão: representação, como árvores ou tabelas, dos mecanismos de tomada de decisão e sua manipulação automática;
- Processamento de Linguagem Natural: representação, análise, tradução automática, busca semântica etc.;
- Segurança: criptografia, compressão de dados, classificação e reconhecimento de padrões etc.;
- Jogos/Simuladores: técnicas inteligentes que podem ser acopladas a jogos e simuladores usando os formalismos adaptativos.

2.1. FORMALISMO PARA DISPOSITIVOS NÃO-ADAPTATIVOS

Um dispositivo não-adaptativo dirigido por regras pode ser qualquer máquina formal que dependa de um conjunto finito de regras para que determine cada possível configuração realizada e sua próxima configuração. Neto (2001).

O dispositivo não-adaptativo dirigido por regras é definido por $ND = (C, NR, S, c_0, A, NA)$, no qual C é o conjunto de todas possíveis configurações; NR é o conjunto de regras que definem ND ; S é o conjunto de todos possíveis eventos que compõe a cadeia de entrada ND ; c_0 é a configuração inicial; A é um conjunto de aceitação das configurações para ND ; e NA são os símbolos de saída que podem ser obtidos pelas chamadas de procedimento.

2.2. FORMALISMO PARA DISPOSITIVOS ADAPTATIVOS

Dispositivos adaptativos são formados por mecanismos adaptativos, pois em qualquer regra não-adaptativa têm a sua regra adaptativa, desde que, na execução ocorra alguma ação adaptativa. Sendo assim, é iniciado uma operação que altere o seu conjunto de regras.

Neto (2001), define o porquê de dispositivos que não são adaptativos poderem se tornar adaptativos, sendo sua forma inicial $AD = (C_0, AR_0, S, c_0, A, NA, BA, AA)$.

Um dispositivo é dito adaptativo sempre que $K \geq 0$ e o mecanismo adaptativo levar as características de ND_k até a execução de uma ação adaptativa não vazia $k+1$ e iniciar uma operação que altere o conjunto de regras. Assim então na execução de alguma ação adaptativa não-vazia ND_k evolui para ND_{k+1} e sua operação no passo $k+1$ cria um conjunto para NR_{k+1} . Depois disso, AD seguirá o comportamento de NR_{k+1} , até que uma outra ação não adaptativa não-vazia seja encontrada para iniciar outro passo. O formalismo da ação adaptativa não-vazia é $AD_{k+1} = (C_{k+1}, AR_{k+1}, S, c_{k+1}, A, NA, BA, AA)$.

3. MODELO LÓGICO PARA DISPOSITIVOS ADAPTATIVOS

Em Camolesi (2004), é apresentado um ambiente para aplicações que simule dispositivos adaptativos. Esse método permite acrescentar um mecanismo adaptativo que, por meio do ambiente, obtém ferramentas que reconhecem os dispositivos adaptativos aplicados.

O gerador de ambientes (meta-ambientes) possibilita gerar automaticamente projetos de aplicações adaptativas. Ele fundamenta-se nos conceitos da TA. Tal gerador permite que os dispositivos subjacentes dirigidos por regras possam ser definidos por um método de extensão para formalismos adaptativos e, desta forma, obter um dispositivo adaptativo e, conseqüentemente, um ambiente para o projeto de aplicações utilizando-se do dispositivo definido (Camolesi, 2007).

O ambiente obtido possui ferramentas para edição, simulação e verificação de aplicações. O ambiente deve permitir a um especialista utilizar uma linguagem de alto nível e representar as suas aplicações num certo dispositivo adaptativo. Desta forma, os especialistas em dispositivos adaptativos poderão despende um maior tempo no estudo dos problemas propostos e buscar eliminar as inconsistências da aplicação. (Camolesi 2007).

No ambiente proposto há um modelo lógico em que apresenta um método, um ambiente e um gerador para definição de dispositivos adaptativos. O mesmo permite a um especialista acrescentar mecanismos adaptativos a um dispositivo subjacente que suportem as ferramentas para a simulação de um projeto usando dispositivos adaptativos.

Sendo que para reconhecimento de dispositivos adaptativos não se utiliza apenas os conceitos do modelo lógico, mas também utiliza-se de um método constituído por três etapas: Teórico, Lógico e Físico.

A primeira Etapa consiste do modelo Teórico. Esta etapa fundamenta-se na estrutura de dispositivos adaptativos de Neto (2001). A responsabilidade nesta etapa é de modificar o comportamento de um dispositivo não-adaptativo que, por meio de

um mecanismo adaptativo, o dispositivo não-adaptativo passa a receber alterações em seu comportamento como remoções ou inserções em sua estrutura, nisto, aumentando o poder de expressão do dispositivo. Esse conceito é a principal característica da tecnologia adaptativa de aumentar o poder de representação a um custo pequeno ser danificar a estrutura do dispositivo.

O dispositivo adaptativo é mapeado para o modelo lógico no qual é representado por um modelo lógico dirigido por regras, conforme ilustra figura 1b. Esta etapa terá a responsabilidade de definir a estrutura de dados, cujo seu papel é muito importante para o conceito formal do dispositivo adaptativo, pois sua estrutura de armazenamento é a principal importância para o desenvolvimento de novas ferramentas dos dispositivos adaptativos.

Na etapa do modelo físico conforme figura 1c são produzidas as especificações para as aplicações que posteriormente serão analisadas e implementadas. Lembrando que esse processo deve ter a ajuda de um ambiente para a modelagem de aplicações com tecnologia adaptativa.

Nisto são permitidas a apresentação da especificação, a simulação do comportamento, a verificação de inconsistências e a execução da aplicação projetada. Com isso, ações adaptativas podem ser executadas e regras podem modificar e/ou remover partes da estrutura do dispositivo adaptativo.

Sendo assim, o trabalho instancia os objetos que foram definidos na etapa lógica e mostra os elementos físicos que representam os comportamentos da aplicação desejada. Nisto são geradas duas classes: uma mostra o comportamento da aplicação projetada e a outra as funções e ações adaptativas que são responsáveis pelo comportamento da aplicação em tempo de execução. Sendo que durante a simulação ações adaptativas modificaram a sua estrutura, porque regras podem ser adicionadas e/ou removidas do comportamento.

A figura 1 mostra as etapas com seus comportamentos e relacionamentos.

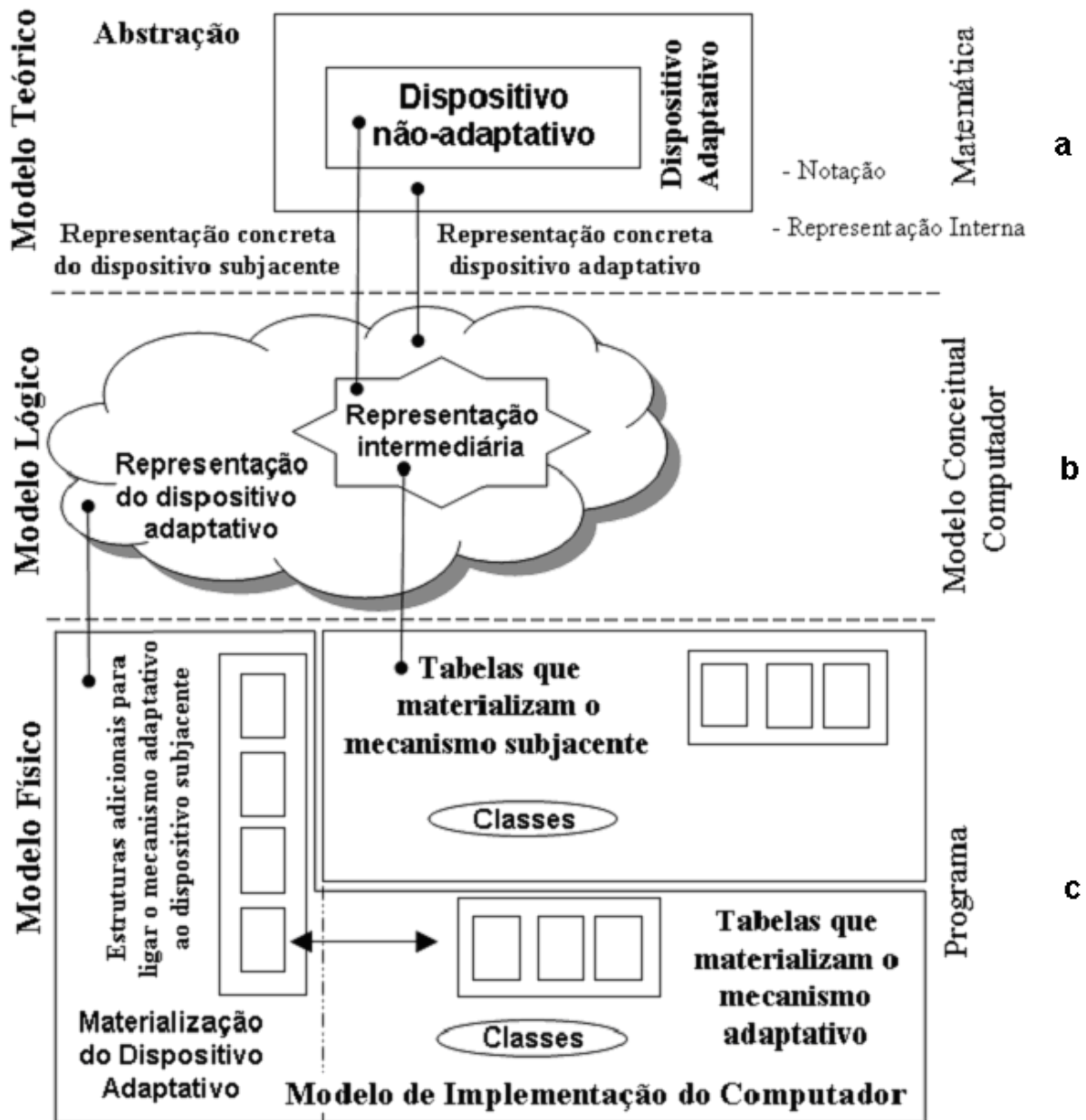


Figura 1 - Método para definição de dispositivos adaptativos (Camolesi, 2007, p. 38).

Como vimos o modelo lógico tem o papel de formalizar os dispositivos adaptativos e sua estrutura de armazenamento, pois é de principal importância para o desenvolvimento de novas ferramentas dos dispositivos adaptativos

O modelo lógico deve representar os elementos conceituais de um dispositivo adaptativo de modo que permita a representação das características dos dispositivos representados por regra.

Para facilitar e possibilitar um melhor entendimento e organização do modelo lógico, o mesmo foi feito em camadas, na qual cada camada é organizada por nível de representação de cada entidade.

Lembrando também que o modelo lógico para dispositivos adaptativos segue uma arquitetura que separa os objetos da camada adaptativa com os objetos da camada subjacente, tal estrutura é muito importante, pois as ferramentas que forem criadas no modelo lógico definido podem realizar especificações, simulações, verificações e teste de aplicações adaptativas.

Como vimos, o modelo lógico tem o papel de formalizar os dispositivos adaptativos e sua estrutura de armazenamento, pois é de principal importância para o desenvolvimento de novas ferramentas dos dispositivos adaptativos

O modelo lógico para dispositivos adaptativos ainda encontra-se separado em quatro camadas: Camada de Dispositivo, Camada Subjacente, Camada Adaptativa e Camada de Memória.

3.1. ESPECIFICAÇÃO DE CLASSES DA CAMADA DE DISPOSITIVO

A classe da camada de dispositivo é responsável pela definição de elementos básicos dirigido por regras que é estruturado por 4 classes: *dispositivo*, *Tipo de componente*, *Tipo de conexão* e *Tipo de atributo*.

A classe de *dispositivo* é onde é formalizada a descrição do dispositivo, o nome do especialista que o definiu e a data da última atualização realizada na estrutura.

Na classe *Tipo de componente* são informados os elementos físicos de um dispositivo. Tais elementos representam as características de um determinado atributo, por exemplo, estados finais e não finais de um autômato finito.

A classe *Tipo de conexão* representa os tipos de ligações existentes em um componente que faz a ligação de uma aplicação, por exemplo, transições de um autômato finito.

Já a classe de *Tipo de atributo* é responsável em armazenar as informações

referentes ao tipo de conexão e associá-las a um determinado dispositivo adaptativo para que possa ter o prosseguimento da aplicação.

3.2. ESPECIFICAÇÃO DE CLASSES DA CAMADA SUBJACENTE

A Camada Subjacente é responsável de modelar as aplicações projetadas que são organizadas estruturalmente em *Projeto*, *Componente*, *Atributo de Componente*, *Conexão* e *Atributo de Conexão*.

O *Projeto* é a classe responsável em armazenar informações do projeto.

A Classe *Componente* representa a especificação do componente que pode ser um estado, uma ação, uma iteração ou até mesmo um conector de um dispositivo adaptativo.

Já na Classe *Atributo de Componente* armazena os atributos do componente que é utilizado durante a execução manipulando os valores na fase de análise ou de tradução para representação física.

A Classe *Conexão* e *Atributo de Conexão* são responsáveis em fazer as ligações entre os componentes e os valores relacionados a uma conexão.

3.3. ESPECIFICAÇÃO DE CLASSES DA CAMADA ADAPTATIVA

A Camada Adaptativa é responsável de acoplar funções e ações adaptativas na sua especificação. Com subclasses, a Camada Adaptativa formaliza as suas propriedades que faz a modificação do comportamento da aplicação e que é acoplada a camada Adaptativa com a Camada Subjacente. Nisto são gerados contadores que durante a execução das funções adaptativas definem o novo código adicionado ao comportamento do componente de uma aplicação.

Os parâmetros armazenaram os objetos que serão os valores passados a função adaptativa no momento da execução, sendo que serão armazenados em outra classe chamada de variáveis, para posterior consulta de suas informações

3.4. ESPECIFICAÇÃO DE CLASSES DA CAMADA DE MEMÓRIA

Nesta camada são representadas as informações do ambiente adaptativo que é constituído pelas classes *cadeia de entrada*, *atributos da cadeia de entrada* e *componentes ativos*.

A classe *cadeia de entrada* é responsável em representar estímulos de entrada da aplicação. Já na classe *atributos da cadeia de entrada* são representados os atributos de cada posição da cadeia de entrada. A classe *componentes ativos* é responsável em representar os componentes que estão ativos durante a execução da aplicação, no ambiente de execução adaptativa.

4. PROPOSTA DO AMBIENTE DE EXECUÇÃO ADAPTATIVO

O trabalho constitui-se em desenvolver um *framework* que segue todos os conceitos de um modelo lógico, proposto POR Camolesi (2007), que é desenvolvido a partir de um ambiente para simulação de dispositivos adaptativos mostrando como são realizados os mais diversos casos em que um determinado dispositivo possa simular o outro, sendo que os dispositivos possuem linguagens de representação diferente, mas possuem um núcleo conceitual equivalente.

Com o *framework* adaptativo um especialista não precisará ter um grande conhecimento em uma determinada linguagem em que foi codificado o modelo lógico do ambiente, pois o ambiente agora só necessitará que o especialista tenha conhecimento sobre o que é proposto no dispositivo.

A figura 2 mostra a inclusão do *framework* no ambiente em que é processada a aplicação da arquitetura do dispositivo adaptativo sendo que a mesma auxiliará um especialista no entendimento e na compreensão dos dispositivos adaptativos que podem ser simulados no ambiente adaptativo.

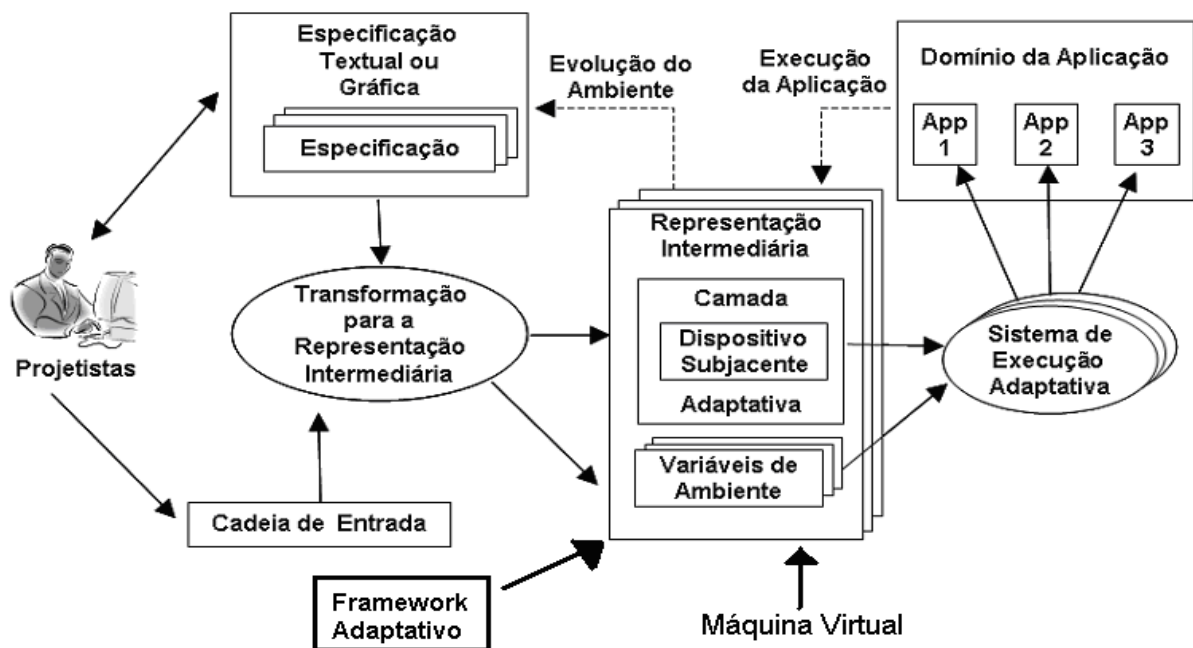


Figura 2 – *Framework* Adaptativo na arquitetura do processo adaptativo (Camolesi, 2007, p. 19).

A maior dificuldade dos especialistas em Tecnologia Adaptativa é de compreender a linguagem em que foi projetado o dispositivo adaptativo, do que a proposta dita no conteúdo. Mas com a linguagem adaptativa isso não ocorrerá, pois se o especialista já conhece o dispositivo adaptativo, a linguagem o interpretará de uma forma dinâmica tornando mais fácil ainda a compreensão e simulação da aplicação do dispositivo adaptativo.

Lembrando que a simulação ocorre de forma estruturada, automaticamente são criados objetos que instanciam e que não necessitam de recompilação para que o mesmo possa ter sua representação de modo que a simulação possa ter sua conclusão.

A figura 3 mostra a linguagem sendo aplicada juntamente com o modelo lógico dito anteriormente.

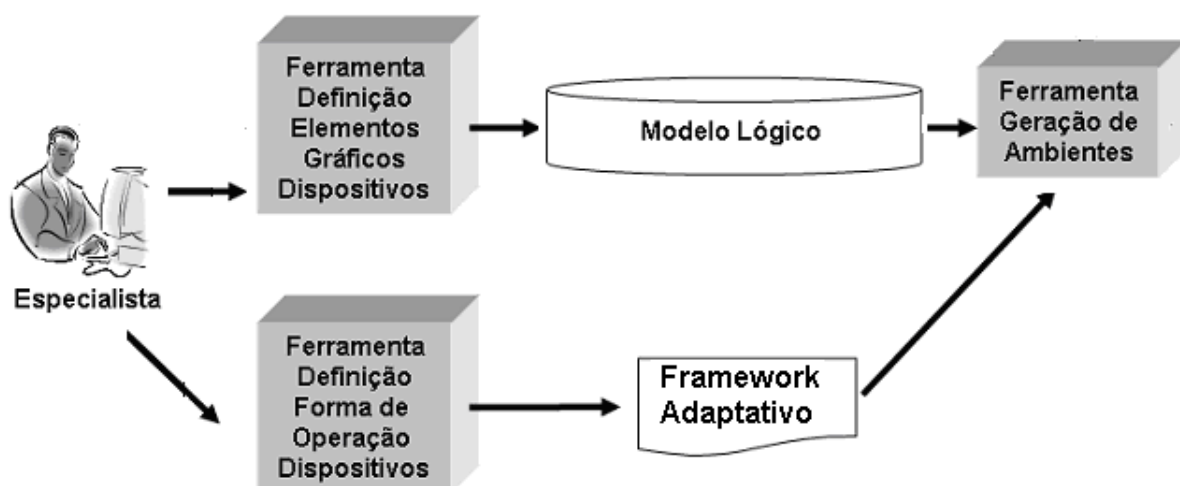


Figura 3 – Framework e o modelo lógico no processo de aplicação adaptativa.

A aplicação do *framework* adaptativo será utilizado no modelo lógico que está situado dentro do ambiente de execução de dispositivos adaptativos, que necessitará da especificação da aplicação e o conceito do modelo lógico. A aplicação do dispositivo adaptativo será simulada e mostrada para o especialista como está sendo executada sua aplicação.

A Figura 4 foi projetada a partir da arquitetura do processo da aplicação adaptativa, ilustrando de que forma o *framework* adaptativo vai agir e fazer a simulação do

dispositivo adaptativo, lembrando que toda a estrutura segue o conceito do ambiente adaptativo que foi formulado neste ambiente, sendo que o *framework* é apenas uma complementação e um aperfeiçoamento para que o ambiente possa ser mais fácil para que especialistas possam aprender e compreender melhor diversas simulações de acordo com dados já pré-estabelecidos pelo ambiente adaptativo.

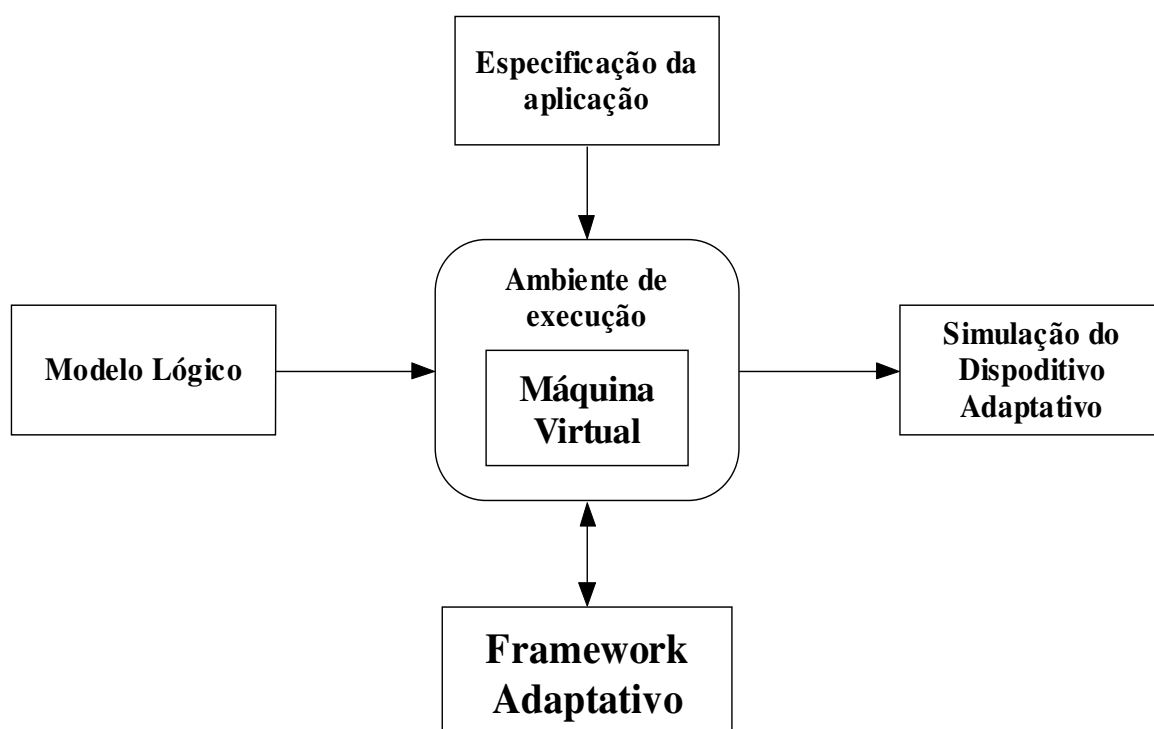


Figura 4 – Aplicação do *Framework* proposto

5. **FRAMEWORK PARA SIMULAÇÃO DE DISPOSITIVOS ADAPTATIVOS**

O trabalho começou primeiramente obtendo informações necessárias sobre as funções de cada camada em que se encontra em enlace no ambiente de execução adaptativa que são: Camada de Dispositivo, Camada de Especificação Adaptativa, Camada de Especificação Subjacente e Camada de memória.

Tendo formalizado o conceito das respectivas camadas, formalizou-se a sua estrutura, já que todo o código antes foi desenvolvido utilizando-se dos conceitos de programação procedural, e o mesmo não permite reaproveitamento de código. Então, utilizam-se todas as fontes de dados e desenvolveu-se um conjunto de classes e procedimentos de forma a seguir os conceitos de orientação ao objeto e de MVC (Model View Controller), obtendo, assim, todas as classes, métodos, controles e ligações com o banco de dados necessário para o desenvolvimento do trabalho.

O *Framework* é uma ferramenta de apoio para simular dispositivos adaptativos que são representados por Autômatos, Redes de Petri, Máquina de Turing, Pilhas e outras diversas representações em que se podem simular dispositivos adaptativos.

A representação em UML (Unified Modeling Language) é de extrema importância, pois é uma linguagem para especificação, documentação, visualização e desenvolvimentos de sistemas orientados a objetos, porque para transportar dos conceitos de programação procedural para os conceitos de Orientação ao Objeto é preciso de uma modelagem do sistema para que não fuja de nenhum enfoque do que já foi programado, pois o objetivo não é de formalizar um novo ambiente, mas sim de seguir os conceitos do Ambiente de execução adaptativa e formalizar um *framework* para auxiliar na simulação de dispositivos adaptativos.

Os simuladores gerados permitirão mostrar como ocorre a execução de uma aplicação modelada utilizando-se de certo dispositivo adaptativo.

A representação do novo conceito em orientação objeto funciona da seguinte forma.

O usuário irá acessar um método que por meio de classes terá as opções para cadastrar ações, componentes, funções e atributos para especificar as características do dispositivo adaptativo em que irá simular no ambiente de execução adaptativa.

Logo após, o simulador fará uma ligação com os objetos, ligando as especificações padrões já cadastradas por um especialista. O sistema através de métodos get's ou set's fará a conexão com o banco de dados através do objeto Controle, cuja função é de controlar o fluxo de informações entre o usuário e o banco que, automaticamente, o objeto Dao abrirá a conexão e só será fechado quando o procedimento for encerrado, porque só assim o usuário poderá cadastrar as especificações necessárias para o uso do simulador adaptativo.

Tendo todas as informações e procedimentos confirmados o objeto *Controle* terá finalizado a simulação terminando a conexão com o banco de dados por meio do objeto Dao, no qual as informações já foram inseridas e salvas ou excluídas do banco de dados.

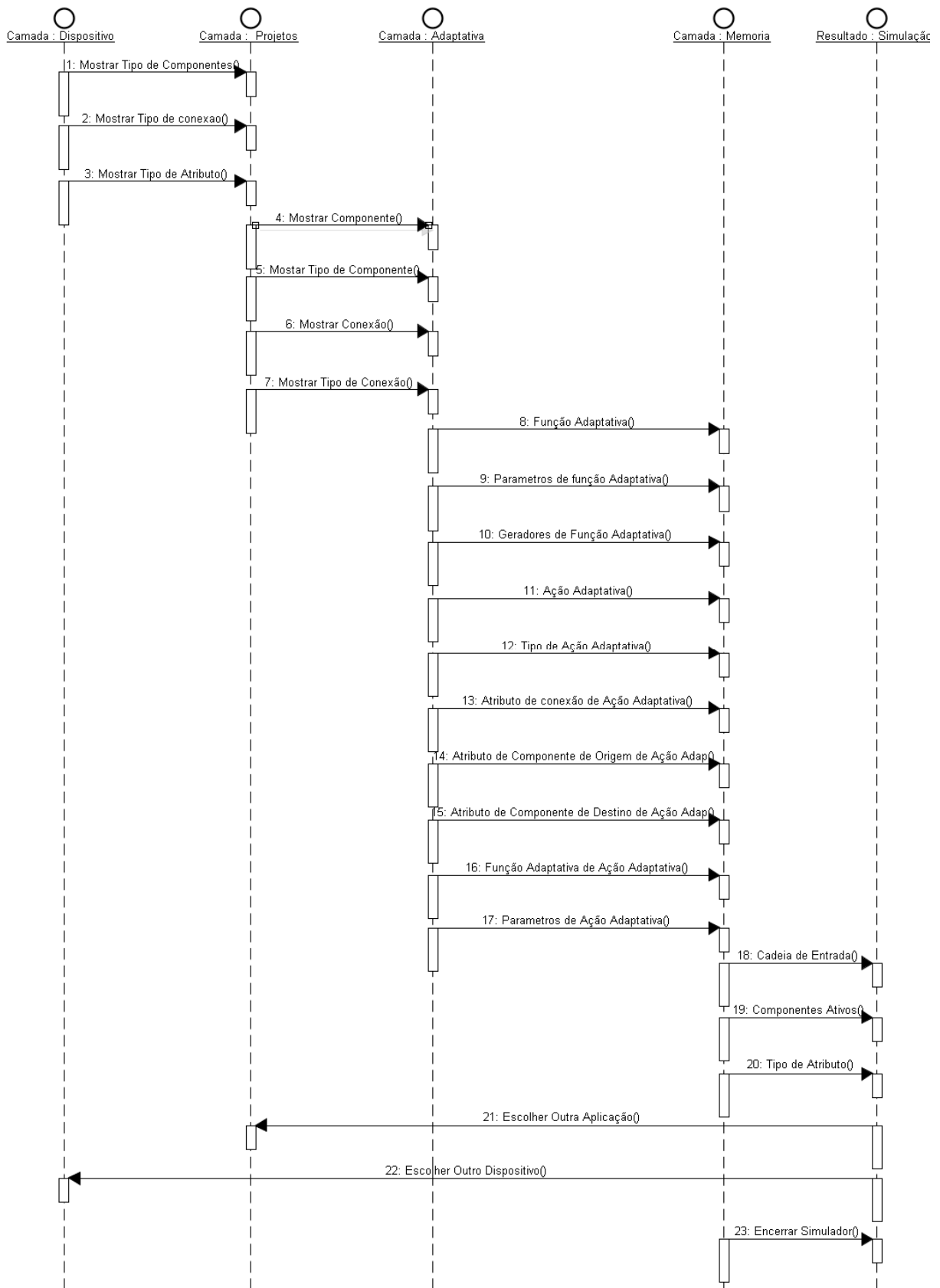


Figura 5 – Diagrama de seqüência para o simulador adaptativo.

Esse *framework* permite o acréscimo de um ambiente específico sem que necessite programar todos os componentes correspondentes a simulação. O programador necessita ter um bom conhecimento no *framework* e na linguagem que faz a representação física, pois assim o programador pode inserir códigos que serão necessários para funcionalidade específica com base na simulação adaptativa representadas no ambiente adaptativo.

O objetivo principal é facilitar a implementação no ambiente por meio do *framework* que, no caso, permite a um especialista instanciar objetos e representar os elementos conceituais no dispositivo adaptativo, por meio do modelo lógico.

Para a funcionalidade do simulador de aplicações adaptativas foi definido um *framework* que permite instanciar diversos dispositivos adaptativos, que são desenvolvidos simuladores de dispositivos adaptativos de forma rápida e que são possíveis de testá-los no projeto de aplicações.

O *framework* proposto segue uma arquitetura de programação que permite que a sua definição seja estruturada, sendo assim, o ambiente permite que novos módulos sejam adicionados sem que necessite recompilação e reestruturação do ambiente adaptativo, pois o mesmo já foi desenvolvido e não necessita de reformulações no código para que o dispositivo possa simular adequadamente, Camolesi (2007).

No framework foram definidos alguns requisitos para a simulação dos dispositivos adaptativos:

- Apresentar Dispositivo Disponível;
- Selecionar Dispositivo;
- Apresentar elementos conceituais do dispositivo escolhido;
- Apresentar especificações de aplicações conf. dispositivo escolhido;
- Selecionar uma especificação - Projeto;
- Apresentar especificações do projeto escolhido;

- Simular aplicação carregada;
- Encerrar Simulador;

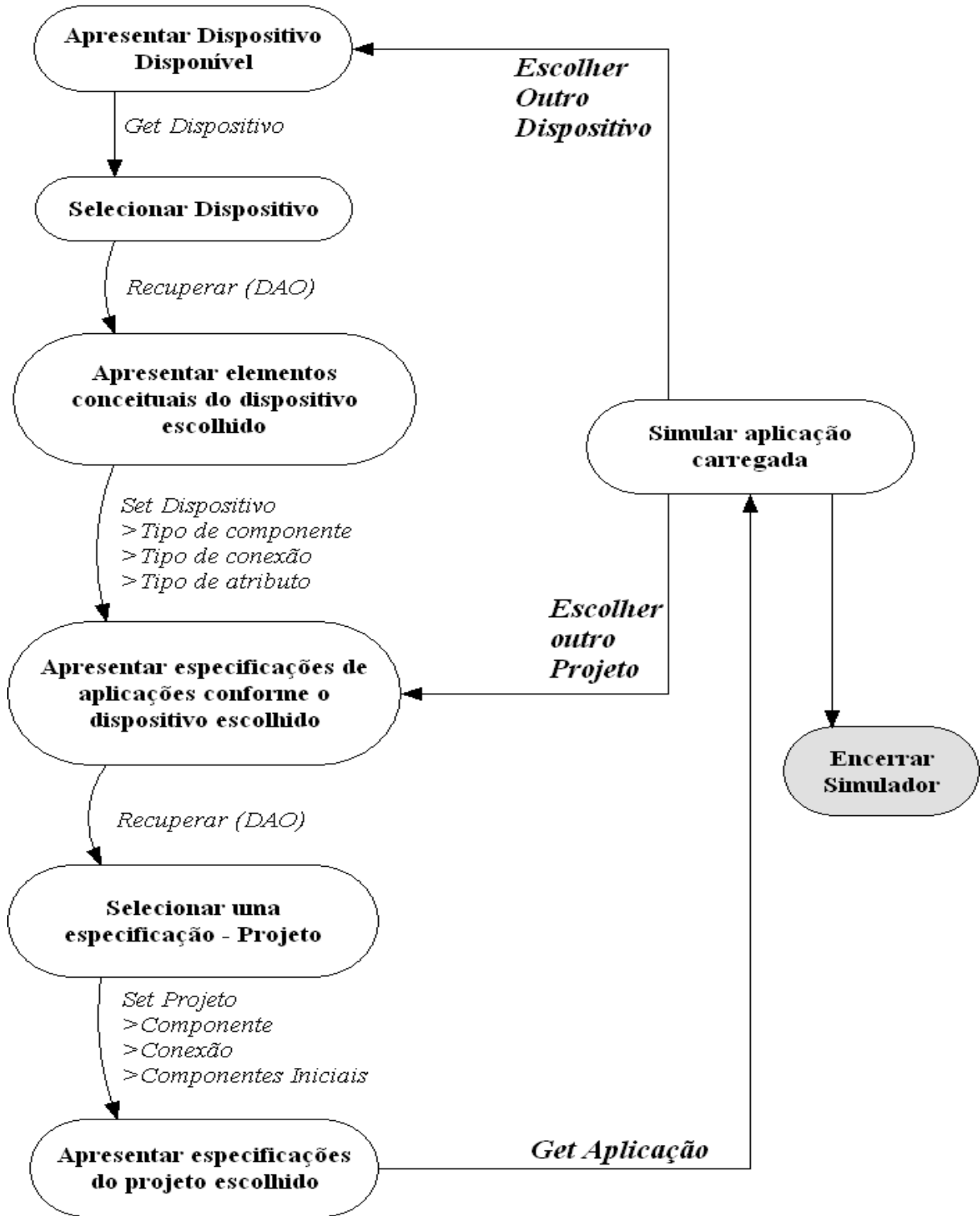


Figura 6 – Diagrama de execução do *framework* adaptativo

O *framework* composto por uma interface permitirá que o especialista visualize os elementos conceituais de um dispositivo, elementos conceituais de especificação de uma aplicação e elementos conceituais de memória.

O processo do *framework* ocorrerá da seguinte forma, ao acessar o ambiente de simulação será apresentada uma lista de dispositivos pré-cadastrados que o ambiente adaptativo reconhece e que pode ser simulado, então, o especialista terá que informar qual dispositivo que irá utilizar para simular. Depois de feita a escolha do dispositivo, será mostrada ao especialista os elementos conceituais do dispositivo requerido, disponibilizando as informações do tipo de componente, tipo de conexão e tipo de atributo.

Depois do especialista escolher o projeto que será utilizado para desenvolver a simulação do dispositivo adaptativo, será mostrado ao especialista os elementos conceituais do projeto requerido, disponibilizando as informações do componente, da conexão e dos componentes iniciais.

Isto também é demonstrado por meio de um diagrama de classe como é o funcionamento do *framework* para a simulação de dispositivos adaptativos utilizando o ambiente adaptativo. Diagrama de classe que se refere ao que ocorre na interface da aplicação.

Para ser mais compreendido, o referido diagrama de classe é dividido em seis classes representadas por camadas, que são: interface do dispositivo, interface de configuração do dispositivo, interface do projeto, interface de configuração do projeto, interface de comportamento da Camada Subjacente e interface de comportamento da Camada Adaptativa.

As classes interface do dispositivo, interface de configuração do dispositivo, referem-se interface do projeto e interface de configuração do projeto refere-se ao projeto e as classes interface de comportamento da Camada Subjacente e interface de comportamento da Camada Adaptativa correspondem às especificações da camada adaptativa.

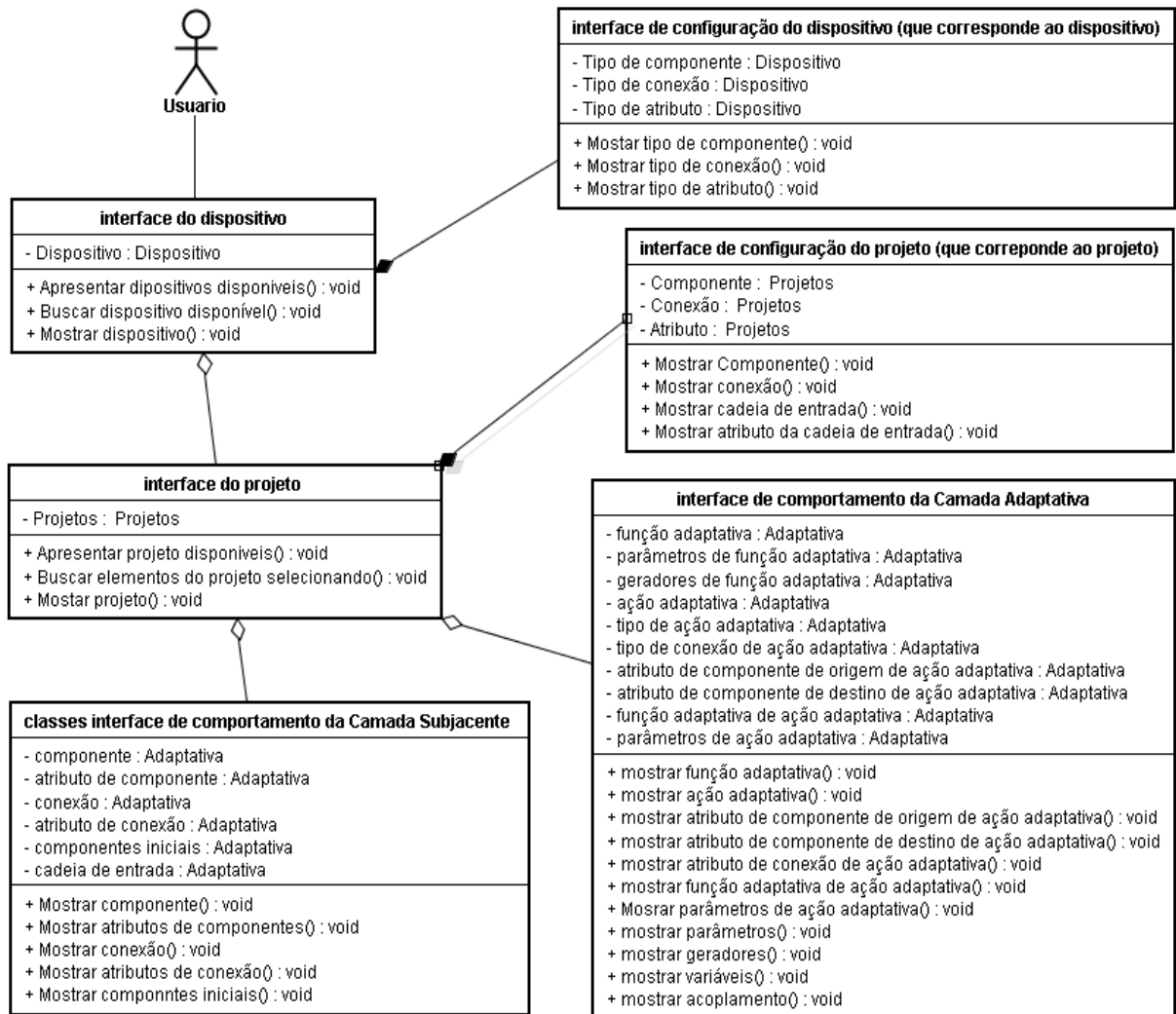


Figura 7 – Diagrama de classes adaptado para o do *framework* adaptativo.

A classe interface do dispositivo é responsável por gerenciar e representar as informações contidas no método “apresentar dispositivos disponíveis”. O método disponibiliza ao especialista uma lista de dispositivos já pré-cadastrados. Ao ser escolhido o dispositivo, ocorre um método que irá buscar no banco de dados as informações necessárias para serem visualizadas pelo especialista. Nessa mesma classe também é mostrado as informações do dispositivo selecionado.

Na classe interface de dispositivo de configuração são disponibilizados os seguintes métodos: mostrar tipo de componente, mostrar tipo de conexão e mostrar tipo de atributo, representando os elementos conceituais de um dispositivo. Ao ser chamado, esses métodos recebem o nome do campo como parâmetros e que faz uma busca no banco de dados e apresenta as informações ao especialista.

Depois que as informações do dispositivo forem apresentadas, será apresentada as informações do projeto.

Na classe interface do projeto é apresentando os projetos que foram estipulados para serem selecionados pelo especialista. Quando for selecionado um dos projetos apresentados, um método “buscar elementos” irá buscar as descrições do projeto, com o nome do projetista, a data de criação etc. Nesta classe também irá ser definida a cadeia de entrada através de um método “mostrar cadeia de entrada” que irá mostrar os atributos da cadeia de entrada para serem visualizados pelo especialista.

Após ter sido mostrada as especificações do projeto, o próximo passo será disponibilizar a especificação de comportamento, que é representada pela camada subjacente e a camada adaptativa. Para as especificações de comportamento foram criadas as classes de interface de comportamento da Camada Subjacente e interface de comportamento da Camada Adaptativa. A classe interface de comportamento da Camada Subjacente é representada pelos métodos e seus respectivos atributos: “mostrar componente”, “mostrar atributo de componente”, “mostrar conexão”, “mostrar atributo de conexão” e “mostrar componente inicial”.

O método mostrar componente irá receber dois parâmetros para informar o nome do campo e o código do projeto, nisto são mostrados os seus respectivos atributos que servem para filtrar os atributos de um respectivo componente, que são representados por um método que é “mostrar atributo de componente”.

Para que seja possível ver a especificação de uma aplicação devem ser apresentadas também as suas conexões. As conexões representam as relações existentes ente os componentes, sendo, então, utilizado o método “mostrar conexões”. Caso a conexão possua um atributo, o mesmo é buscado por meio do nome do campo que armazena o código do projeto e valor do código do projeto (método mostrar atributo de conexão).

A classe interface de comportamento da Camada Subjacente também é mostrada o método componente inicial, que apresenta as informações dos componentes iniciais ativos para que seja utilizado quando for iniciada a simulação.

Sabendo que nem todas as aplicações precisam ser adaptativas, pois se, por exemplo, um especialista quiser representar um autômato simples, ele pode aplicar por meio da camada subjacente. Mas caso seja realizado uma aplicação adaptativa, foi estruturada uma classe interface de comportamento da classe adaptativa, que é composto pelos seguintes métodos: mostrar função adaptativa, mostrar ação adaptativa, mostrar atributo de componente de origem, mostrar atributo de componente de destino, mostrar atributo de conexão de ação adaptativa, mostrar função adaptativa de ação adaptativa, mostrar parâmetros, mostrar geradores, mostrar variáveis e mostrar acoplamento.

Para filtrar e apresentar ao especialista as funções adaptativas que fazem parte da especificação de um projeto foi criado o método mostrar função adaptativa, o qual possui parâmetros que permitem mostrar todas as funções adaptativas disponíveis dentro de um conjunto de ações adaptativas, representadas pelo método mostrar ações adaptativas.

Para cada ação adaptativa deve ser apresentado o seu respectivo atributo de origem, de destino e de suas conexões, através dos métodos, mostrar atributo de componente de origem, mostrar atributo de componente de destino, mostrar atributo de conexão de ação adaptativa.

As ações adaptativas podem descrever novas funções adaptativas para serem adicionadas e, para isso, devem ser visualizadas, portanto, foram definidos os métodos “mostrar função adaptativa de ação adaptativa” e “mostrar parâmetros” que são responsáveis em mostrar as funcionalidades das ações adaptativas.

Além de ações adaptativas, as ações adaptativas possuem elementos relacionados que é representado pelos métodos mostrar parâmetros, mostrar geradores e mostrar variáveis. Métodos que permitem a visualização dos elementos de um projeto ou função adaptativa. O método “mostrar acoplamento” corresponde à associação de uma função adaptativa a um componente ou conexão, classe interface de comportamento da classe adaptativa segue o conceito conforme Camolesi (2007).

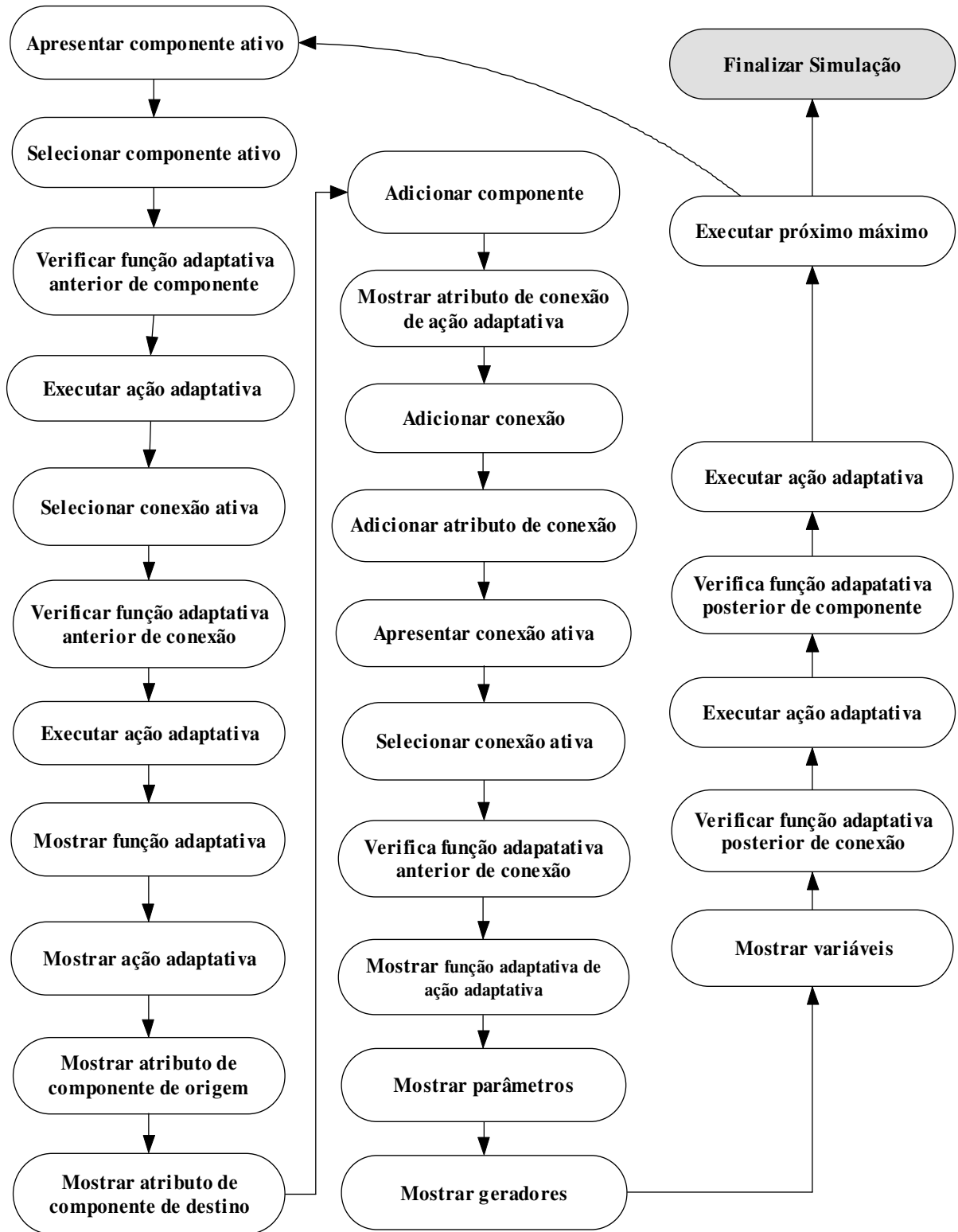


Figura 8 – Diagrama de execução da simulação adaptativa.

Camolesi (2007), propõe que inicialmente o especialista deve verificar se o componente está ativo, portanto, será utilizado o “método apresentar componente

ativo”, após ter verificado se o componente está ativado, o especialista pode escolher qual componente deseja utilizar através do método selecionar componente ativo.

Logo depois de selecionado o componente ativo, é verificado se existe uma função adaptativa. Se o componente ativo que foi selecionado tiver uma função adaptativa, será executado o método “executar função adaptativa”, que é o responsável por executar as ações adaptativas da simulação e, na seqüência, são retornadas as conexões que estão ativas por meio do método “apresentar conexão ativa”, da mesma forma do componente. Se houver mais de uma conexão ativa, o especialista tem a opção de selecionar a conexão que irá realizar a simulação, através do método selecionar conexão ativa. Na seqüência será realizada a verificação da existência da função adaptativa anterior para a conexão, por meio de um método chamado verificar função adaptativa anterior de conexão. Quando for executado esse procedimento, o mesmo retornará ao processo de simulação.

Depois de selecionado o componente e sua respectiva conexão o método mostrar função adaptativa retornará as informações das funções adaptativas, sendo que para cada função adaptativa existe uma ação adaptativa, que no qual é representado pelo método “mostrar ação adaptativa”.

O próximo procedimento será mostrar as informações dos atributos que correspondem ao componente, sendo que ao componente é designado uma origem e um destino, representado pelos métodos: “mostrar atributo de componente de origem” e “mostrar atributo de componente de destino”.

Se o processo precisar acrescentar ou modificar algum componente neste momento, um método “adicionar componente” fará a inserção de um componente novo na simulação do dispositivo adaptativo.

Ainda na execução “ação adaptativa”, da mesma forma que foi mostrado os atributos de componente, um método mostrará as informações dos atributos de uma conexão que contenha uma ação adaptativa, o qual se utiliza do método “mostrar componente de conexão de ação adaptativa”.

Se uma conexão precisar acrescentar ou modificar uma conexão, um método

“adicionar conexão” fará a inserção do novo procedimento da conexão e também do atributo de conexão, por meio do método “adicionar atributo de conexão”.

Para verificar a função adaptativa que havia anteriormente à nova conexão, utiliza-se o método “verificar função adaptativa anterior de conexão”.

Na execução da ação adaptativa se o dispositivo tiver uma ação adaptativa, o método “mostrar função adaptativa de ação adaptativa” irá informar quais são as ações adaptativas do determinado dispositivo representado pelo ambiente de simulação adaptativa.

Além das ações adaptativas, as funções adaptativas possuem seus elementos relacionados com parâmetros que são representados pelo método “mostrar parâmetros”, os geradores que são apresentados pelo método “mostrar geradores” e variáveis que são denominadas pelo método “mostrar variáveis”. Elementos estes que permitem a um especialista visualizar todos os elementos relacionados a um projeto a uma respectiva função adaptativa.

Para finalizar a execução da ação adaptativa, verifica-se que se há uma função adaptativa posterior de conexão e, na seqüência, verifica-se também a função adaptativa posterior de componente. Se a simulação tiver outros procedimentos a serem executadas, é executado o próximo passo repetindo novamente todo o procedimento, mas se não houver, é encerrado a simulação.

6. IMPLEMENTAÇÃO DO *FRAMEWORK*

Para a implementação do *framework* foram utilizados o NetBeans 6.1¹, o banco de dados PostgreSQL 8.2² e a estrutura de mapeamento objeto-relacional Hibernate para desenvolvido do trabalho.

A figura 9, mostra como foram desenvolvido a estrutura principal do trabalho.

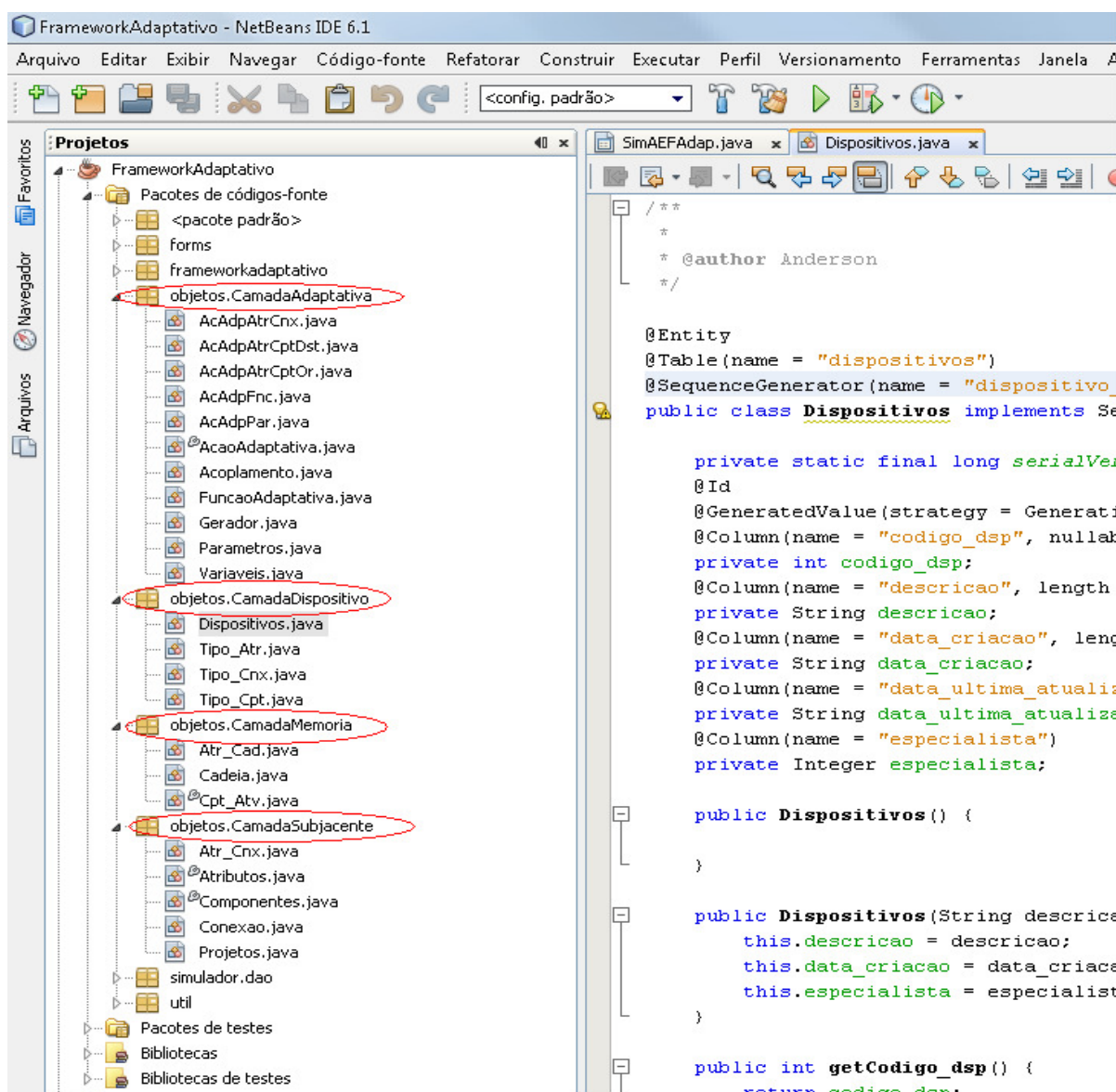


Figura 9 – Criação dos Objetos e suas classes

¹ <http://www.netbeans.org/>

² <http://www.postgresql.org.br/>

O primeiro passo foi desenvolver os objetos e que são separados por camadas, chamando-se de objetos.CamadaAdaptativa, objetos.CamadaDispositivo, objetos.CamadaMemoria e objetos.CamadaSubjacente, sendo que cada objeto possui suas respectivas classes contendo as informações de chave primária e chaves estrangeiras e os métodos get's e set's.

Também foram implementados dois pacotes chamados Pacotes de códigos-fonte e útil que servem para fazerem a conexão da estrutura do Hibernate com o banco de dados Postgress e a criação das tabelas e seus respectivos campos, tornando assim um trabalho reutilizável podendo ser acrescentado outras classes e métodos em trabalhos futuros.

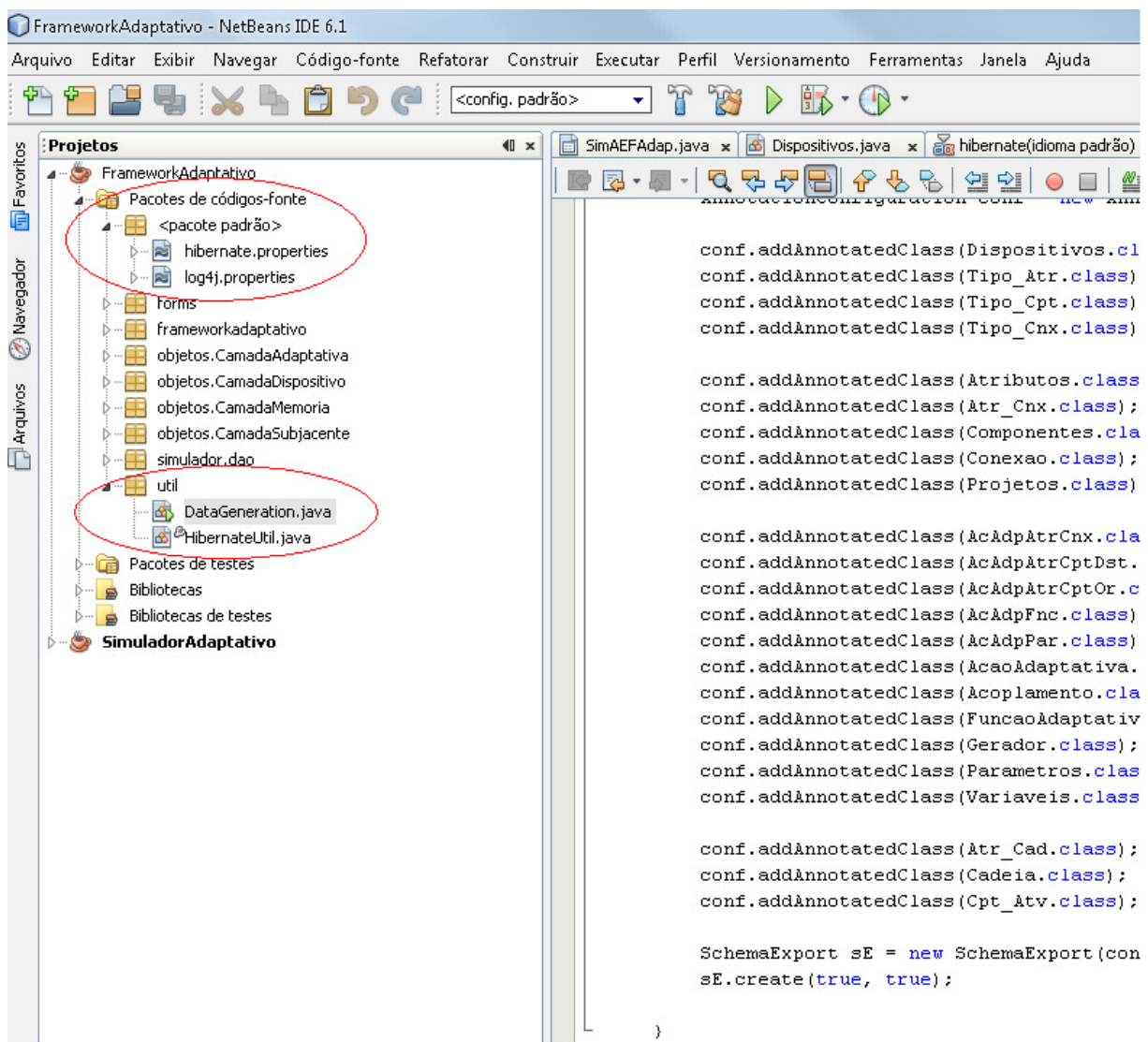


Figura 10 – Criação dos pacotes do *Hibernate* e geração do banco de dados

Para controlar o acesso do simulador com o banco de dados foram implementados os métodos DAO proporcionando um método mais seguro e prático para realização de inserções, alterações e exclusões no banco de dados.

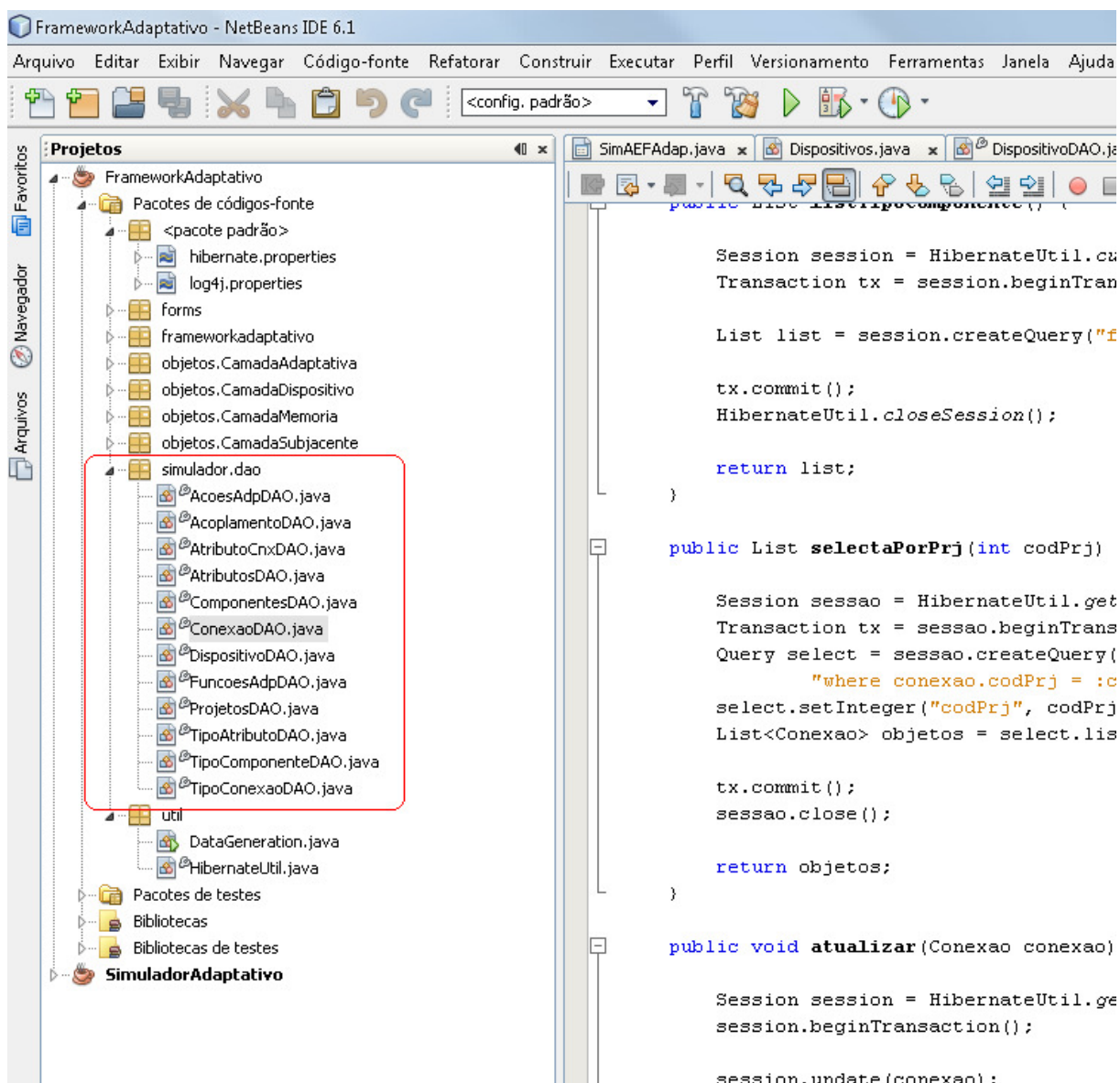


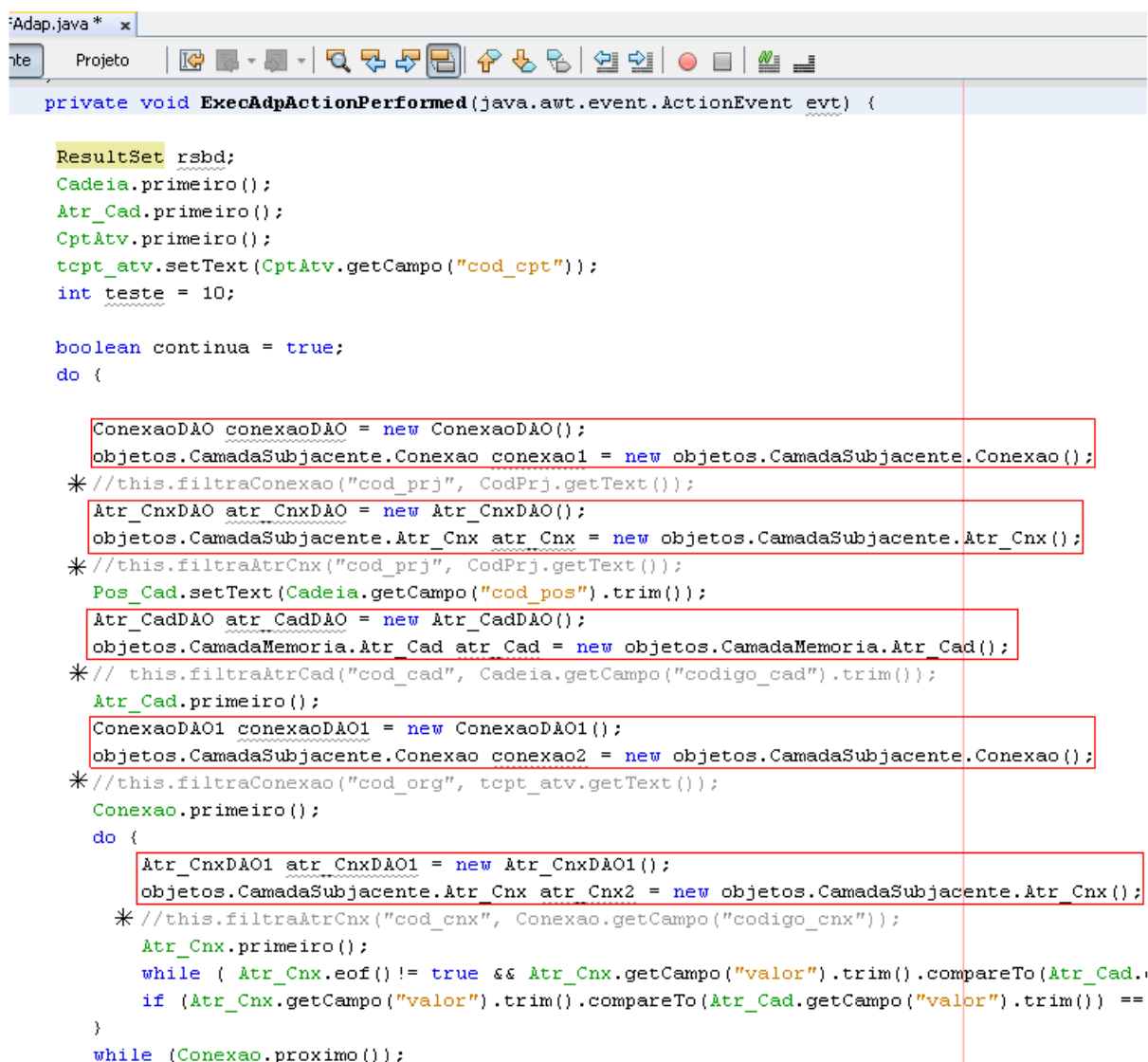
Figura 11 – Criação do pacote DAO e as classes utilizadas para simulação.

Na figura 12 mostra uma parte da implementação de como o trabalho do *framework* proporcionará um código reutilizável e mais enxuto fazendo com que assim não precise de muitas funções para o controle do simulador com o banco de dados.

Os códigos que estão em destaque em um quadrado representam a implementação do novo código desenvolvido para o simulador adaptativo e os códigos que estão

comentados e com um asterisco, referem-se ao código que era utilizado. Esse código utilizava-se de funções e não seguia nenhuma estrutura e sua programação é totalmente de forma procedural.

Com o *framework* os trabalhos futuros poderão ser bem compreendidos, pois foram feitos vários estudos de diagramas e a implementação do *framework* que contém toda a base de objetos, classes e métodos para a simulação de dispositivos adaptativos.



```
private void ExecAdpActionPerformed(java.awt.event.ActionEvent evt) {  
  
    ResultSet rsbd;  
    Cadeia.primeiro();  
    Atr_Cad.primeiro();  
    CptAtv.primeiro();  
    tcpt_atv.setText(CptAtv.getCampo("cod_cpt"));  
    int teste = 10;  
  
    boolean continua = true;  
    do {  
  
        ConexaoDAO conexaoDAO = new ConexaoDAO();  
        objetos.CamadaSubjacente.Conexao conexao1 = new objetos.CamadaSubjacente.Conexao();  
        *//this.filtraConexao("cod_prj", CodPrj.getText());  
        Atr_CnxDAO atr_CnxDAO = new Atr_CnxDAO();  
        objetos.CamadaSubjacente.Atr_Cnx atr_Cnx = new objetos.CamadaSubjacente.Atr_Cnx();  
        *//this.filtraAtrCnx("cod_prj", CodPrj.getText());  
        Pos_Cad.setText(Cadeia.getCampo("cod_pos").trim());  
        Atr_CadDAO atr_CadDAO = new Atr_CadDAO();  
        objetos.CamadaMemoria.Atr_Cad atr_Cad = new objetos.CamadaMemoria.Atr_Cad();  
        *// this.filtraAtrCad("cod_cad", Cadeia.getCampo("codigo_cad").trim());  
        Atr_Cad.primeiro();  
        ConexaoDAO1 conexaoDAO1 = new ConexaoDAO1();  
        objetos.CamadaSubjacente.Conexao conexao2 = new objetos.CamadaSubjacente.Conexao();  
        *//this.filtraConexao("cod_org", tcpt_atv.getText());  
        Conexao.primeiro();  
        do {  
            Atr_CnxDAO1 atr_CnxDAO1 = new Atr_CnxDAO1();  
            objetos.CamadaSubjacente.Atr_Cnx atr_Cnx2 = new objetos.CamadaSubjacente.Atr_Cnx();  
            *//this.filtraAtrCnx("cod_cnx", Conexao.getCampo("codigo_cnx"));  
            Atr_Cnx.primeiro();  
            while ( Atr_Cnx.eof() != true && Atr_Cnx.getCampo("valor").trim().compareTo(Atr_Cad.  
                if (Atr_Cnx.getCampo("valor").trim().compareTo(Atr_Cad.getCampo("valor").trim()) ==  
            )  
        }  
        while (Conexao.proximo());  
    }  
}
```

Figura 12 – O novo código em destaque em um quadrado e o antigo que está comentado com um asterisco na frente.

7. CONCLUSÃO

Este trabalho buscou contribuir no aperfeiçoamento do ambiente para simulação de dispositivos adaptativos, para que o mesmo siga um método único no qual seu código seja reaproveitado e que a área teórica e técnica sejam mais especificadas para o especialista em tecnologia adaptativa. Este capítulo descreve as conclusões do trabalho realizado, suas principais contribuições e sugestões para trabalhos futuros.

7.1. TRABALHO REALIZADO

Inicialmente foram desenvolvidos estudos sobre a tecnologia adaptativa, levantando os conceitos do funcionalismo matemático e teórico. Com base nos estudos realizados no ambiente de simulação de dispositivos adaptativos, realizou-se um estudo sobre o modelo lógico usado para representar os elementos conceituais dos dispositivos subjacente e/ou adaptativos.

Com relação ao *framework*, foi desenvolvida uma análise para representar os conceitos por intermédio de diagramas de classe, diagramas de seqüência e diagramas de execução.

Com o objetivo de auxiliar o aprendizado de dispositivos adaptativos o *framework* foi desenvolvido para auxiliar o especialista no desenvolvimento de simulações para dispositivos adaptativos.

7.2. PRINCIPAIS CONTRIBUIÇÕES

Resumidamente, as principais contribuições alcançadas foram a formalização de classes para o desenvolvimento do *framework* para simulação de dispositivos adaptativos e, como conseqüência, essas classes formam métodos que ajudam a especificar melhor a execução dos dispositivos representados no modelo lógico. No *framework* são descritos os passos das funções e ações adaptativas. Tais definições

facilitam o desenvolvimento de ferramentas de simulação para os dispositivos que são estendidos para as funcionalidades de tecnologia adaptativa e mapeados para o modelo lógico.

Com o auxílio dos simuladores obtidos, os especialistas em dispositivos adaptativos poderão simular os seus dispositivos sem depender de um grande tempo no desenvolvimento de simuladores e, com isso, obter um maior tempo para o estudo dos problemas propostos, que geralmente são difíceis de ser compreendidos.

7.3. TRABALHOS FUTUROS

Os trabalhos desenvolvidos serviram para o desenvolvimento da tecnologia adaptativa e também para material didático para futuros estudos. É possível sugerir alguns tópicos para serem realizados em relação ao trabalho e a tecnologia adaptativa, como:

- O término desse *framework* para concluir toda parte de simulação de dispositivos adaptativos;
- Construção do ambiente completo para auxiliar os especialistas na definição dos dispositivos adaptativos;
- O desenvolvimento de simulações para comprovar a eficácia do procedimento do *framework*.

8. REFERÊNCIAS BIBLIOGRÁFICAS

Camolesi, A. R.; Neto, J.J. - Representação Intermediária para Dispositivos Adaptativos Dirigidos por Regras. 3rd International Information and Telecommunication Technologies Symposium, UFSCar, São Carlos, Brasil, 2004.

Camolesi, A. R. - Proposta de um gerador de ambientes para a modelagem de aplicações usando tecnologia adaptativa – Escola Politécnica (EP) – USP Universidade de São Paulo, 2007.

Neto, J. J. e Magalhães, M.E.S. - Um Gerador Automático de reconhecedores sintáticos para o SPD. VIII SEMISH - Seminário de Software e Hardware, pp. 213-228, Florianópolis, 1981.

Neto, J. J. - Uma Solução Adaptativa para Reconhecedores Sintáticos. Anais EPUSP - Engenharia de Eletricidade - série B, vol. 1, pp. 645-657, São Paulo, 1988.

Neto, J. J. - Contribuições à metodologia de construção de compiladores. Tese de Livre Docência, USP, São Paulo, 1993.

Cardoso, J.; Vallete, R. - Redes de Petri. - Editora UFSC, Florianópolis, 1997.

Pistori, H.; Neto, J.J. - AdapTree - Proposta de um Algoritmo para Indução de Árvores de Decisão Baseado em Técnicas Adaptativas - Anais Conferência Latino Americana de informática - CLEI 2002, Montevideo, Uruguai, Novembro, 2002.

Pistori, H - Tecnologia adaptativa em engenharia de computação: Estado da Arte e Aplicações – Escola Politécnica (EP) – USP Universidade de São Paulo, 2003.