

FUNDAÇÃO DE ENSINO "EURÍPIDES SOARES DA ROCHA"  
CENTRO UNIVERSITÁRIO "EURÍPIDES DE MARÍLIA" - UNIVEM  
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

**GUILHERME CRUZ ZACARIAS DA SILVA**

**INTERFACE TANGÍVEL SENSÍVEL AO CONTEXTO EM AMBIENTE  
COLABORATIVO**

MARÍLIA  
2011

GUILHERME CRUZ ZACARIAS DA SILVA

INTERFACE TANGÍVEL SENSÍVEL AO CONTEXTO EM AMBIENTE  
COLABORATIVO

Trabalho de Curso apresentado ao Curso de Ciência da Computação da Fundação de Ensino "Eurípides Soares da Rocha", mantenedora do Centro Universitário Eurípides de Marília - UNIVEM, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Orientador

Prof.º Ms. LEONARDO CASTRO BOTEGA

MARÍLIA

2011



CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA  
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

TRABALHO DE CONCLUSÃO DE CURSO – AVALIAÇÃO FINAL

---

Guilherme Cruz Zacarias da Silva

INTERFACE TANGÍVEL SENSÍVEL AO CONTEXTO EM AMBIENTE COLABORATIVO

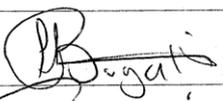
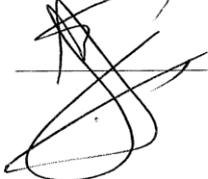
Banca examinadora da monografia apresentada ao Curso de Bacharelado em Ciência da Computação do UNIVEM/F.E.E.S.R., para obtenção do Título de Bacharel em Ciência da Computação.

Nota: 10.0 ( DEZ )

Orientador: Leonardo Castro Botega

1º. Examinador: Ildeberto de Gênova Bugatti

2º. Examinador: Elvis Fusco

  
\_\_\_\_\_  
  
\_\_\_\_\_  
  
\_\_\_\_\_

Marília, 30 de novembro de 2011.

## AGRADECIMENTOS

*Agradeço primeiramente a Deus por ter me dado um presente tão nobre que é minha vida e um dom tão bonito que é poder enxergar a beleza das pessoas que estão a minha volta.*

*Aos meus pais (Alípio e Anizia) que foram os que me impulsionaram até aqui, portanto, devo a eles um enorme agradecimento pelo amor, pelo apoio e pela paciência com este filho que tanto os ama.*

*Devo um agradecimento muito especial ao meu orientador Prof.º Leonardo Botega que me apoiou, me aconselhou, me puxou a orelha quando precisou, dedicou-se bastante ao tirar minhas dúvidas, permitiu que houvesse troca de experiência e conhecimento com outros campus, que me ajudou a atravessar as fronteiras impostas no caminho e se mostrou muito mais que um professor, saiba que para mim sempre será "O professor".*

*Aos meus amigos de graduação e professores, agradeço muito pelo apoio e pelas palavras amigas e sinceras, são de fato as pessoas mais agradáveis com quem já estive.*

*Agradeço também a minha namorada Aline pelo apoio nas horas difíceis e compreensão das dificuldades, que apesar de às vezes desistir mais rápido do que deveria, foi com você que aprendi a importância de perseverar.*

*Que Deus abençoe a todos vocês meus caros.*

CRUZ, Guilherme Z. da Silva. **Interface Tangível de Usuário Sensível ao Contexto em Ambiente Colaborativo**, 2011, 42 f. Trabalho de Curso (Bacharelado em Ciência da Computação) - Centro Universitário Eurípides de Marília, Fundação de Ensino "Eurípides Soares da Rocha", Marília, 2011.

#### RESUMO

Interfaces Tangíveis de Usuário (TUI) buscam estabelecer uma comunicação mais natural e intuitiva entre o ambiente físico e o contexto da aplicação, isso implica em uma menor interação explícita do usuário, permitindo-o desenvolver atividades com maior grau de complexidade. Nesse ambiente uma aplicação sensível ao contexto atribui autonomia ao software para reestruturar o processo por ele tratado, ou também diagnosticar ao usuário melhores práticas para a manipulação do contexto em questão. Utilizando-se dessa interface, é proposto nesse projeto o desenvolvimento de uma aplicação sensível ao contexto em busca de uma melhor integração de entidades em ambiente tangível, tendo um Quebra-cabeça como estudo de caso.

**Palavras-chave:** Sensibilidade ao Contexto; Pós-Wimp; Interface Tangível.

CRUZ, Guilherme Z. da Silva. **Interface Tangível de Usuário Sensível ao Contexto em Ambiente Colaborativo**, 2011, 42 f. Trabalho de Curso (Bacharelado em Ciência da Computação) - Centro Universitário Eurípides de Marília, Fundação de Ensino "Eurípides Soares da Rocha", Marília, 2011.

#### ABSTRACT

The Tangible User Interfaces (TUI) seek to establish a more natural and intuitive communication between the physical environment and application context, this implies a less explicit user interaction, allowing to develop activities with greater complexity. In this environment an application context-aware software to assign autonomy to restructure the process treated with it, or even diagnose the user best practices for handling context. Using this interface, it is proposed in this project to develop a context-aware application in search of a better integration of entities in such an environment tangible, having a puzzle as a case study.

**Keywords:** Context-Awareness; Pos-Winp; Tangible User Interface.

## LISTA DE ILUSTRAÇÕES

Figura 1 - Fluxo de dados através da estrutura computacional. ....	10
Figura 2 - Exemplo de uso de RV .....	12
Figura 3 - Exemplo RA utilizando um marcador. ....	13
Figura 4 - Elementos da TUI.....	13
Figura 5 - Arquitetura da reactTable. ....	14
Figura 6 - Marcadores fiduciais utilizados na reactTable.....	15
Figura 7 - Operando a reactTable. ....	16
Figura 8 - Módulos do sistema Active Cube. ....	17
Figura 9 - Funções dos módulos.....	17
Figura 10 -- Esquema de conexão do sistema Active Cube .....	18
Figura 11 - Níveis de Contexto .....	22
Figura 12 - Diagrama de atividade. ....	24
Figura 13 - Modelagem de Classes para a aplicação proprosta.....	25
Figura 14 - Exemplo de jogo Quebra-cabeça .....	26
Figura 15 - a) Exemplo de usuário soltou a peça em um espaço qualquer; b) Exemplo de sistema realizando encaixe da peça na grade.....	26
Figura 16 - Diagrama de Atividade de Mover Peça. ....	27
Figura 17 - Diagrama de Atividade de Auto-Organização.....	28
Figura 18 - Estrutura TUI disponível para o desenvolvimento do projeto visualizada em corte. ....	29
Figura 19 - CCV em Execução identificando as entradas e saídas. ....	30
Figura 20 - Fluxograma de funcionamento do MT4j .....	31
Figura 21 - Identificação do fiducial no CCV e no MT4j .....	32
Figura 22 - Grade de Peças, onde a peça número 1 está em movimento .....	33
Figura 23 - Padrão de Organização Matriz.....	34
Figura 24 - Padrão de Organização Borda.....	35
Figura 25 - Solicitação de permissão para desencaixe .....	36

## SUMÁRIO

INTRODUÇÃO.....	8
CAPÍTULO 1 -INTERFACE TANGÍVEL.....	10
1.1. Exemplos de aplicações TUI.....	14
1.1.1. reacTable .....	14
1.1.2. Active Cube .....	16
CAPÍTULO 2 - SENSIBILIDADE AO CONTEXTO.....	19
2.1. Sistemas sensíveis ao contexto.....	19
2.1.1. Exemplos de sistemas sensíveis ao contexto.....	20
2.2. Modelando o contexto .....	21
2.2.1. Contexto de baixo nível.....	22
2.2.2. Contexto de alto nível.....	23
CAPÍTULO 3 - METODOLOGIA.....	24
3.1. Levantamento de Requisitos.....	24
3.1.1. Movimentação das peças do jogo.....	25
3.1.2. Permissões de movimentação .....	27
3.1.3. Auto-Organização.....	28
3.2. Estudo de Ferramentas.....	28
3.2.1. TUI .....	28
3.2.2. Community Core Vision (CCV).....	29
3.2.3. Protocolo TUIO .....	30
3.2.4. Multitouch For Java (MT4j) .....	30
3.3. Prototipação.....	32
3.3.1. Comunicação entre aplicações.....	32
3.3.2. Peças do jogo e a grade de alinhamento .....	32
3.3.3. Auto-Organização.....	34
3.3.4. Permissão de movimentação .....	35
3.3.5. Análise dos Resultados Obtidos .....	36
CAPÍTULO 4 - CONCLUSÕES.....	38
REFERÊNCIAS .....	39

## INTRODUÇÃO

As aplicações computacionais utilizadas nas mais diversas áreas do conhecimento visam organizar virtualmente processos ou atividades do mundo físico de forma que a gerência e execução destas possam ser otimizadas através da estruturação de dados e manipulação dos mesmos. Observa-se que os grandes avanços tecnológicos realizados nos últimos anos têm permitido cada vez mais conectar o mundo físico aos sistemas computacionais de forma mais direta, onde a ação de obter dados do ambiente e suas entidades é possível por meio de uma área de pesquisa que vem crescendo chamada computação sensível ao contexto.

Muitas melhorias em aplicações comumente utilizadas são realizadas em estruturação de dados, layouts e processamento de dados, beneficiando a estrutura e desempenho, o que trata parcialmente a semântica de interação homem-máquina, uma vez que as otimizações ocorreram abordando diretamente a máquina. Observa-se que apesar da reestruturação realizada nesta situação citada, ainda é necessário tratar o outro lado desta relação, pois os obstáculos podem ainda prevalecer por parte do usuário. A observação deste tipo de dificuldade é evidenciada ao se realizar atividades com alto nível de complexidade, onde o esforço computacional é altamente dependente do usuário que o manuseia. Nesse tipo de atividade, prevalece então o esforço do usuário.

Este trabalho propõe o tratamento e otimização da interface homem-máquina através de uma interface tangível e análise de contexto. Descreve-se aqui um trabalho que consiste no levantamento e implementação de uma aplicação que utiliza os conceitos de sensibilidade ao contexto e interface tangível anteriormente citados em um escopo reduzido e bem delimitado, utilizando como estudo de caso o tradicional jogo Quebra-cabeça.

Esse jogo possui característica colaborativa, ou seja, permite a construção de um ambiente multiusuário, assemelhando-se ao desenvolvimento colaborativo de projetos.

No primeiro capítulo aborda-se o estudo de interface tangível mostrando sua relevância para a aplicação e sua dinâmica de funcionamento na interação homem-máquina.

No segundo capítulo é feito um levantamento bibliográfico de sensibilidade ao contexto onde é exibido seu conceito, sua importância e a aplicabilidade de um sistema sensível ao contexto.

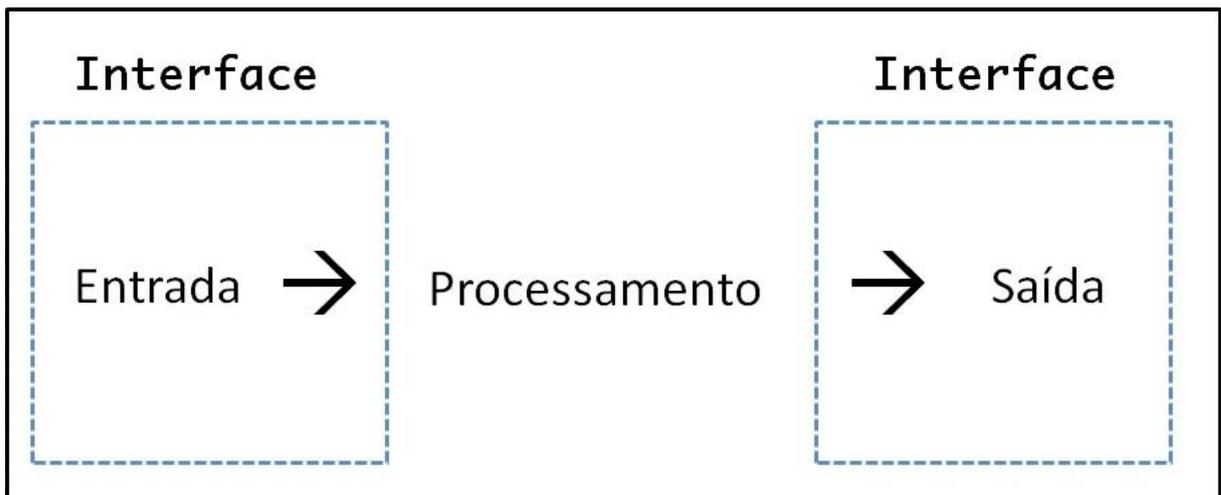
No terceiro capítulo aborda-se a metodologia de desenvolvimento do trabalho, que vai da regulação do hardware, passando pelo levantamento de requisitos e chegando à utilização de frameworks para a construção da aplicação final.

No quarto e último capítulo são apresentados as conclusões deste trabalho e possíveis futuros trabalhos para continuação ou evolução deste estudo.

## CAPÍTULO 1 -INTERFACE TANGÍVEL

Uma vez presente a necessidade de interagir com um sistema computacional torna-se primeiramente necessário traduzir a linguagem natural humana em computacional. A interface na área da computação é o elemento responsável por realizar essa tradução. É um intermediador na relação homem-máquina que ao exercer sua função torna a visão do usuário mais simples e mais clara em relação aos processos executados, de forma que este se preocupe menos com a linguagem computacional e mais como problema a ser tratado.

Independente da forma de atuação da interface, ela consiste na entrada e saída de dados. É recebido o estímulo externo do usuário na interface, gerando os dados de entrada para o sistema, que por sua vez analisa as informações recebidas e realiza o processamento necessário, alimentando quando necessário a interface de saída que interpreta os dados e apresenta ao usuário. O fluxo básico desta operação pode ser observado na Figura 1.



**Figura 1 - Fluxo de dados através da estrutura computacional. Fonte própria.**

As primeiras interfaces computacionais criadas nas décadas de 40 e 50 tinham como as entradas do sistema chaves eletro-mecânica e a saída de dados uma sequência de lâmpadas respondendo em linguagem bem próxima à linguagem de máquina. Alguns anos mais tarde, nas décadas de 70 e 80 foi expandido a utilização da interface baseada no paradigma de linha de comando, utilizando-se de teclados como dispositivos de entrada e monitores como saída, ou seja, os comandos desejados pelo usuário são digitados no teclado e submetidos a processamento e retornada a resposta em monitor, temos como exemplo os terminais Linux ou Unix. Esse tipo de interface exige que o usuário possua certo grau de conhecimento da linguagem da interface que não é totalmente natural humana. Independente desse requisito,

foi uma grande evolução na interface computacional que permitiu um significativo crescimento e popularização dos microcomputadores (PC - Computador pessoal).

No final da década de 70, através de pesquisas desenvolvidas principalmente nos laboratórios *Stanford Research Institute (SRI)* e *Massachusetts Institute of Technology (MIT)*, concebeu-se um novo paradigma de interface chamado WIMP formado basicamente por representações gráficas como janelas, ícones, menu e ponteiro (cursor). Atualmente essa interface é largamente aplicada nos sistemas operacionais mais utilizados e é conhecida como GUI (*Graphic User Interface*).

Já no início da década de 80 a empresa Xerox (XEROX, 2011) uniu o chamado paradigma WIMP ao dispositivo hardware mouse criando o conceito de "*Desktop*", onde os ícones simbolizavam arquivos e aplicações do sistema bem como o gerenciamento das janelas. Essa evolução contribuiu bastante para a evolução da interface computacional e conseqüentemente para o gerenciamento de atividades do usuário por permitir uma experiência mais simplificada e agradável.

Observando as interfaces atuais em relação às suas precursoras, destaca-se o aumento de elementos facilitadores na interface, visando a otimização da interação do usuário com o ambiente virtual. Uma vez que a interface é caracterizada pela reunião dos aspectos computacionais que afetam o uso do sistema (SMITH,1982), ela é quem diretamente se responsabiliza por utilizar os recursos disponíveis e construir relação homem-máquina através da organização dos dados afim de facilitar essa interação do usuário com os elementos contextuais da aplicação.

O conceito de interface exprimido neste trabalho, refere-se à Interface Homem Computador (IHC) que trata a relação entre o usuário e a aplicação, mas é importante deixar claro que há também o conceito de interface aplicado a duas aplicações distintas que necessitam interagir, precisando assim compartilhar a mesma linguagem através de uma interface tradutora. Esse tipo de interface não é o foco do estudo em questão, mas também demonstra o principal papel das interfaces na tradução das linguagens.

O estudo da IHC foi amplamente realizado a partir da década de 90, procurando explorar funções multimídia como o som e a imagem, ambas como entrada e saída de dados: comandos de voz (SHNEIDERMAN, 1998) e reconhecimento de padrões como gestos ou toque (BOUCHET e NIGAY, 2004).

Observa-se que IHC geralmente abordam dois níveis de interpretação: o nível físico (hardware) e o lógico (software). Ambos evoluíram significativamente enquanto interface, cada um da sua forma específica. Um exemplo interessante de interface é o mouse, onde em

seu nível físico consiste no objeto com seus componentes eletroeletrônicos que é movimentado em uma superfície qualquer e em nível lógico consiste na interpretação e atualização de suas coordenadas no sistema que são indicadas pela sua movimentação física capturada pelo hardware.

A partir da evolução constante da interface observa-se uma aproximação maior entre o usuário e a aplicação, dando espaço à Realidade Virtual (RV). Trata-se de um conceito em interfaces avançadas, que consiste na capacidade de aproximar o usuário da interface através de interações de resposta em tempo real feitas a partir de sensores no ambiente do usuário, permitindo-o ser inserido de uma forma artificial, mas próxima da real no ambiente computacional (KIRNER *et al*, 1995). Exemplifica-se a RV na Figura 2.



**Figura 2 - Exemplo de uso de RV (VRCIM, 2010)**

Em contrapartida à RV existe a Realidade Aumentada (RA) que tem caráter inverso, baseia-se em projetar elementos virtuais em elementos reais através de aparelhos tecnológicos, ou seja, trazer elementos presentes no sistema computacional sobrepostos ao objetos físicos como demonstrado na Figura 3 através da utilização de um marcador.

Interfaces Tangíveis (*Tangible User Interface*, TUI) permitem a otimização da interação homem-máquina através da utilização de diversos elementos no contexto de interação com o usuário, reduzindo as limitações de interface. A TUI consiste no uso de objetos diversos do ambiente físico, já assimilados de certa forma ao sistema, para que o

usuário manipule estes objetos como elementos de entrada, gerando assim as informações digitais (ULLMER e ISHII, 1997).



Figura 3 - Exemplo RA utilizando um marcador (KIRNER, 2007).

Atravessando as limitações da GUI, a TUI dispõe de uma forma diferenciada de manipular os elementos contextuais do sistema. Essa interface define os objetos reais como entrada para o sistema, que uma vez atrelados aos respectivos objetos virtuais permite que o usuário modifique seu contexto através desta ligação (FITZMAURICE *et al*, 1995). Esse tipo de interação caracteriza a manipulação da informação digital e ainda busca a sincronização harmoniosa entre os artefatos físicos com os virtuais da interface. Esse nível de abstração da interface permite ao usuário uma maior capacidade de imersão e interação, estabelecendo uma comunicação e relacionamento mais natural, uma experiência mais agradável e também simplificada. A interface ganha elementos como pode-se ver na Figura 4.

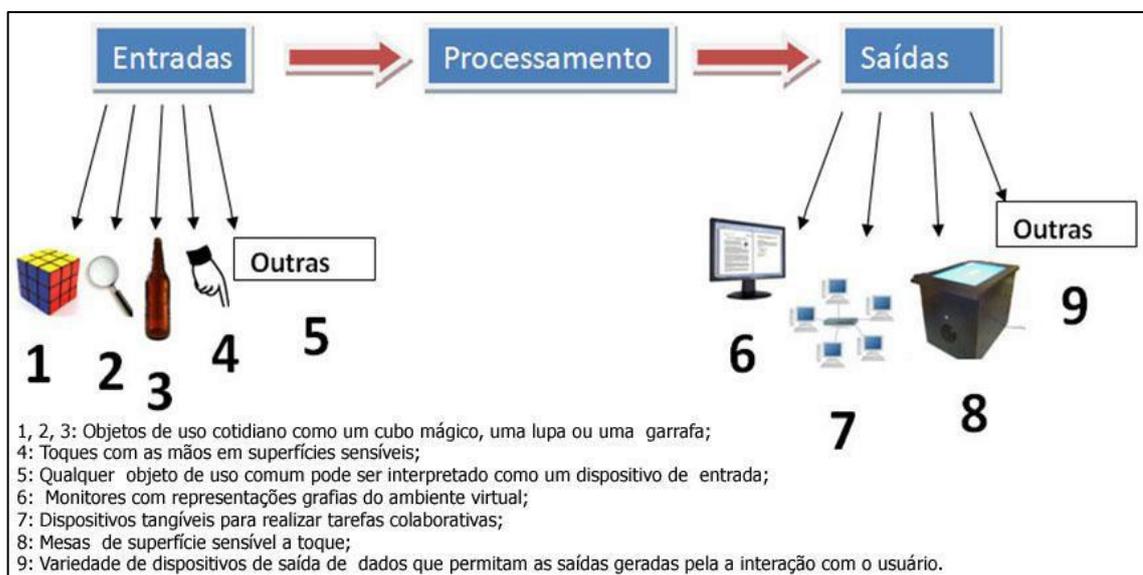


Figura 4 - Elementos da TUI (RADICCHI, 2010)

Os elementos de contexto presentes na TUI e suas interações são manipuláveis e visíveis de uma melhor forma pelo fato de que o sistema ao aplicar este tipo de interface passa a dispor de uma maior "superfície" de interação. Isso acontece porque o meio de contato do usuário com o ambiente não é limitado pelo uso de dispositivos padrões de interação como o teclado e o mouse somente, o que permite uma maior quantidade de interferências no sistema.

## 1.1. Exemplos de aplicações TUI

Dois exemplos de aplicações TUI são trazidos para uma melhor contextualização deste tipo de interface, sendo eles a *reactTable* e o *Active Cube*.

### 1.1.1. *reactTable*

Esta IHC permite que seus usuários construam músicas eletrônicas de uma forma mais simples e intuitivas do que da forma convencional utilizando a *pick up* (mesa de mixagem) e programas de edição. A arquitetura (Figura 5) consiste em uma mesa tangível redonda de superfície translúcida e no seu interior há uma câmera responsável por observar toda a movimentação e interação que ocorrem na superfície da mesa e um projetor para a saída de dados.

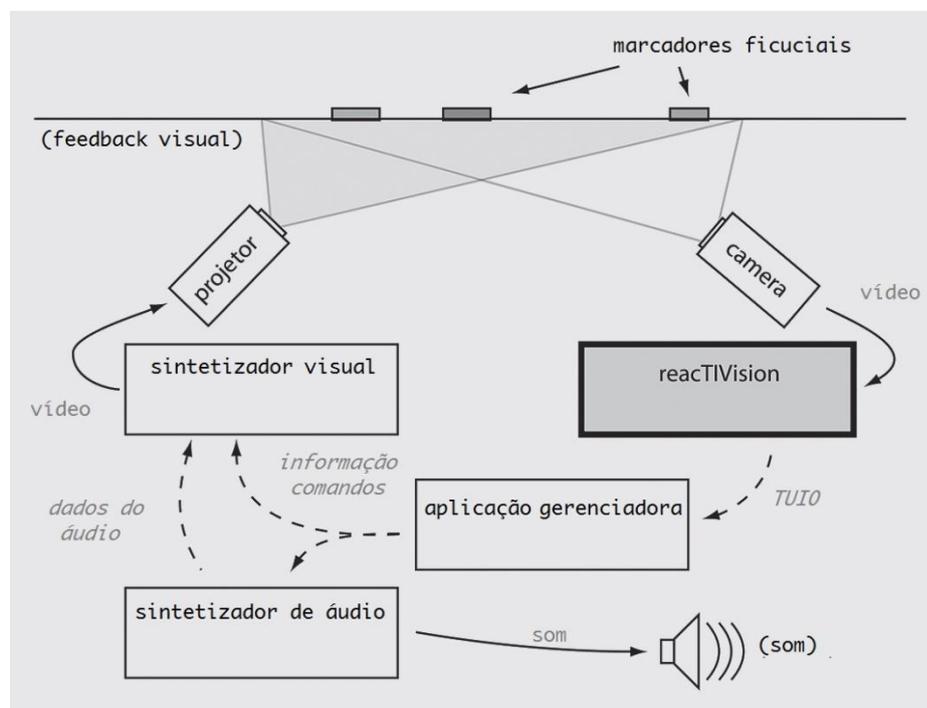
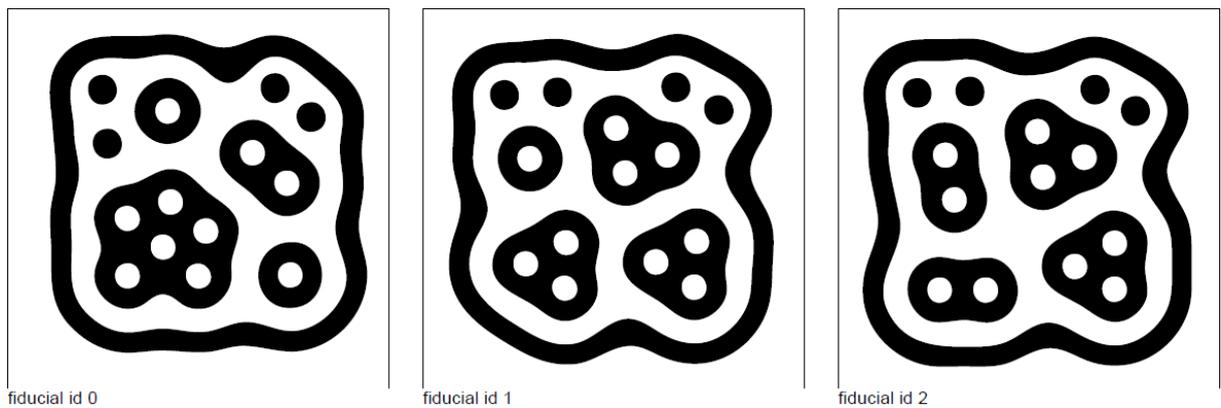


Figura 5 - Arquitetura da *reactTable* (GEIGER *et al*, 2010).

Os objetos de interação com a mesa são marcadores fiduciais (Figura 6), que consistem em símbolos para formarem um padrão de linguagem, onde cada um deles possui

um código que representa uma característica específica, podendo ser um som ou uma distorção sonora. Ao entrarem em contato com a superfície esses marcadores são capturados pela câmera do interior da mesa, que comunica-se com a aplicação de visão computacional, a *reactIVision*, responsável por interpretar os dados capturados e identificar os marcadores em sua posição, momento e código. Após o reconhecimento, esses dados do marcador são transmitidos pelo barramento de rede utilizando o protocolo de comunicação TUIO, protocolo este, que é abordado no item 3.2.3 deste trabalho. A aplicação gerenciadora captura esses dados, interpreta e chama a ação pertinente de acordo com os dados do marcador.



**Figura 6 - Marcadores fiduciais utilizados na reactTable (NUIGROUP, 2011).**

A posição do marcador, o seu ângulo de rotação sobre a superfície em seu eixo Y e a sua distância em relação ao centro são fatores importantes para determinar se há combinação de sons e efeitos entre os marcadores, o volume do som a ser reproduzido e a velocidade de sua execução, permitindo uma grande quantidade de possibilidades ao usuário que passa a executar esses comandos de forma intuitiva.

A partir da decisão tomada da aplicação gerenciadora baseada nos dados capturados e interpretados, é fornecida a saída do sistema para o usuário. A estrutura da *reactTable* conta com um sintetizador de áudio, que é o alvo da aplicação gerenciadora, ou seja, recebe os comandos e executa as reproduções sonoras. Entre estas reproduções estão aplicação de filtros sonoros, harmonizadores, quantizadores de passo, fusores e sincronizadores. Este resultado é transmitido para alto-falantes do sistema, entretanto não é o único dispositivo que responde ao usuário, há também um sistema de projeção na superfície translúcida da mesa. Isso é possível porque paralelamente ao sintetizador de áudio, há também um sintetizador visual, ou também chamado de motor gráfico, que também recebe os comandos da aplicação gerenciadora somado aos dados fornecidos pelo sintetizador de áudio e tem como função renderizar os objetos virtuais baseado nas posições dos objetos reais e também as suas possíveis conexões

que simbolizam a combinação dos elementos. Essa renderização é transmitida ao projetor no interior da mesa que gera a imagem sobreposta aos elementos posicionados na mesa. O resultado dessa iteração visual pode ser visto na Figura 7.



**Figura 7 - Operando a reactTable (GEIGER *et al*, 2010).**

### **1.1.2. Active Cube**

Esta interface se baseia em montagem construtiva, ou seja, através da interligação física dos módulos disponíveis na interface. Este tipo de IHC tem como foco principal o espaço físico dos objetos e o relacionamento entre os elementos que o compõe, garantindo ao usuário uma grande quantidade de opções para compor seu resultado pretendido.

O *Active Cube* é composto por cubos físicos (Figura 8) também chamados de módulos que podem ser de diversos tipos, cada qual com sua funcionalidade. Estes cubos possuem suas próprias entradas de dados e saídas, entre elas podemos destacar as funções de entrada como sensor tátil, sensor ultra-sônico de presença, sensor sonoro e giroscópio. E as funções de saída são alto-falantes, LED's (*Light Emissor Diode*) emissores de luz (semelhante a um semáforo de trânsito urbano), vibrador, motor rotor, ventilador e matriz de LED (Figura 9).

Nesse sistema é utilizado um computador para o reconhecimento da estrutura composta por cubos que é montada no ambiente físico e sincronizada em tempo real através de uma rede previamente estabelecida. Nota-se que há dois tipos de comunicação existentes

nesse ambiente: a comunicação entre os cubos e a comunicação realizada entre o grupo de blocos e o computador.

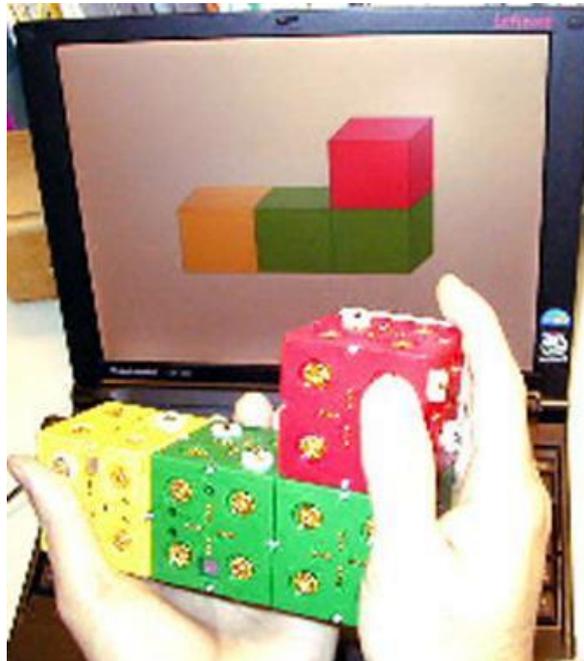


Figura 8 - Módulos do sistema *Active Cube* (KITAMURA *et al*, 2001).



Figura 9 - Funções dos módulos (WATANABE *et al*, 2004).

A comunicação entre os módulos é possível através de uma rede que é formada entre eles ao conectá-los e que é gerenciada por cada um destes módulos que possuem um processador. Cada um desses cubos possui um código único, identificando cada peça de forma singular, o que permite a gerência dos elementos na rede. Cada face do cubo possui um sensor

infra-vermelho responsável por reconhecer outros cubos que realizem uma conexão ou uma desconexão.

A comunicação com o computador é realizado através de protocolo de rede intermediado por um roteador que é conectado diretamente ao cubo intermediador chamado de cubo base. O roteador troca dados de rede com este módulo e também energiza-o, assim, quando um novo cubo é anexado ao cubo base, a energia recebida no primeiro módulo é compartilhada com os demais (Figura 10).

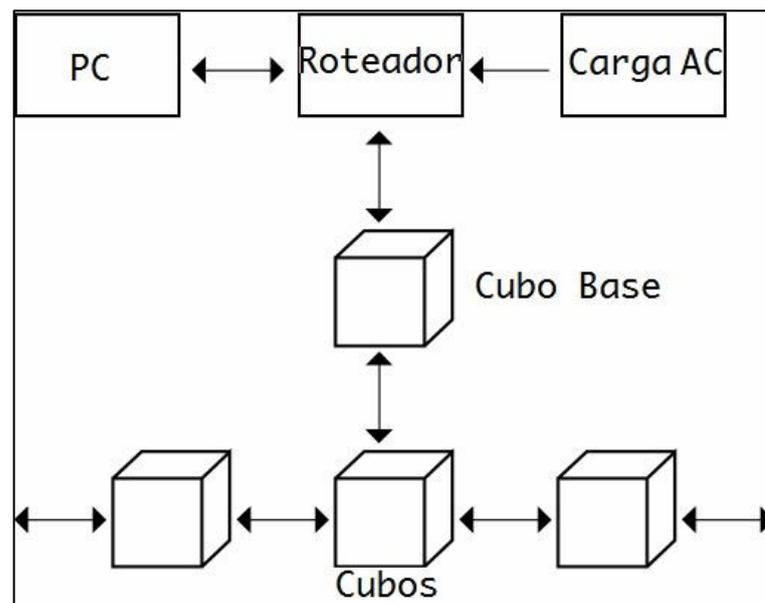


Figura 10 -- Esquema de conexão do sistema Active Cube (WATANABE *et al*, 2004).

O sistema *Active Cube* resume-se em capturar os dados do ambiente por meio dos cubos e suas conexões, reproduzir o resultado obtido e atuar na aplicação executada, isso garante ao usuário uma nova ferramenta de interação com o sistema.

## CAPÍTULO 2 - SENSIBILIDADE AO CONTEXTO

Em complemento à TUI observa-se um outro fator de dinamização do sistema: o contexto em que ele está inserido. Há grande discussão sobre a exata e precisa definição de contexto, pois ele é variável a cada ambiente e ou situação. De uma forma bem abrangente, contexto é identificado como um conjunto de condições e influências relevantes que tornam uma situação única e compreensível, referindo-se a uma pessoa, grupo de pessoas, um objeto no espaço, ou outra entidade genérica (BRÉZILLON, 1997).

No âmbito computacional, aceita-se o contexto como qualquer informação caracterizadora de uma entidade, sendo esta última uma pessoa, um objeto ou um lugar considerado relevante para a interação entre o ambiente computacional e o usuário. As informações das entidades de forma generalizada tratam a localização, identidade e estados de cada uma delas (DEY e ABOWND, 2001).

A interpretação do conceito de contexto pode ser mais naturalmente compreendida em atividades cotidianas, como por exemplo a conversa entre duas pessoas, partindo do princípio que estas observam o contexto em que a comunicação ocorre. Para fazer essa análise elas observam os elementos contextuais (CE), e nesse caso esses elementos estão relacionados com a riqueza da linguagem que utilizam (vocabulários, gírias ou formalismo), o entendimento de ambos de como o mundo funciona e a interpretação das mensagens implícitas na entonação. Todos esses elementos circundam o contexto que é gerado nas mentes dos indivíduos em questão. Embora esse seja um elemento presente no cotidiano no relacionamento humano, a aplicação do contexto continua sendo pouco explorado pelos sistemas computacionais (VIEIRA *et al*, 2009) e também mostra-se um nicho de pesquisa para que seja aplicado em otimização de sistemas diversos.

### 2.1. Sistemas sensíveis ao contexto

As áreas de Inteligência Artificial (IA) e Computação Ubíqua foram as que utilizaram o termo Computação Ciente de Contexto (*Context-Aware Computing*) pela primeira vez para classificar um sistema que era capaz de observar o ambiente computacional e atuar em função das mudanças deste ambiente, ou seja, desse contexto (SCHILIT *et al*, 1994). A IA utilizou este conceito devido ao fato de que um contexto não é estático e suas regras são bem inconstantes, portanto é necessária a aplicação dos conhecimentos desta área para o reconhecimento das mudanças de padrão e perspectivas do sistema. A Computação

Ubíqua por sua vez se apresenta como o estudo e aplicação de informações e recursos computacionais em qualquer momento, em qualquer lugar e em qualquer meio (WEISER, 1991). Esta área de atuação garante a si mesma uma grande proximidade com a análise de contexto, uma vez que sua base consiste na adaptação da tecnologia para o envolvimento do ambiente, empregando então a principal característica da Sensibilidade ao Contexto.

### **2.1.1. Exemplos de sistemas sensíveis ao contexto**

Alguns exemplos de aplicações que utilizam a sensibilidade ao contexto para promover um resultado mais específico aos seus usuários em seus diversos domínios.

#### **2.1.1.1. *Dynamic Tour Guide***

Em países que tem o turismo como principal gerador de riquezas em seu país há uma necessidade de tratar o turista sob uma perspectiva atenciosa, onde permitir que ele possua boas experiências naquele país é o objetivo para que possa retornar outras vezes. No âmbito de orientar o turista, criou-se uma aplicação que tem função de guia turístico para os usuários: *Dynamic Tour Guide* (DTG). Este guia é executado em dispositivos móveis e auxilia na criação de rotas personalizadas ao usuário-turista considerando diferentes preferências de usuário, informações ambientais e necessidade de transporte (KRAMER *et al*, 2005).

A indicação de roteiros e pontos turísticos é realizada através da análise do contexto onde o usuário está inserido, levando em consideração alguns dados relevantes que compõe o contexto da aplicação: intervalo de tempo disponível para o passeio, a localização atual do turista (utilizando-se como referência os locais próximos ao mesmo), o horário atual (determina a estação do ano ou eventos turísticos decorrentes daquele período), interesses pessoais do usuário (afinidade com temas ou hábitos) e a velocidade de locomoção do usuário (determinando a área de cobertura dos eventos). Além de traçar as rotas, esta aplicação também fornece dados sobre os pontos turísticos.

#### **2.1.1.2. *GMail***

A análise do contexto pode ser utilizada em hipermídia de forma adaptativa afim de promover personalização de seu conteúdo em função do usuário. No caso do *GMail*, que é um sistema de gerenciamento de emails da *Google*, a análise realizada sobre o usuário implica em uma personalização na exibição de propagandas. A aplicação em questão utiliza-se de uma ferramenta de análise de texto que realiza uma varredura na mensagem que o usuário está

visualizando em sua tela. Nessa leitura são identificadas palavra chaves que são as responsáveis por selecionar a publicidade mais adequada o conteúdo da mensagem (GMAIL, 2011).

O tratamento do contexto neste domínio permite ao sistema garantir uma maior proximidade entre o conteúdo da mensagem e o conteúdo da publicidade apresentada no gerenciador de email o que garante um efeito de sincronização.

## 2.2. Modelando o contexto

Não há um número definido da quantidade e tipos de elementos que circundam o contexto, porém é possível realçar e utilizar aqueles caracterizados como prioridade em um sistema computacional.

A unidade mais significativa para a análise do contexto é o anteriormente citado CE que consiste em um dado, informação ou conhecimento que permite caracterizar uma entidade em um domínio (VIEIRA *et al*, 2009). Sendo assim, o CE é caracterizado com grande relevância porque o contexto formado a partir de uma interação entre o agente e a aplicação é montado a partir do conjunto de CE's que apóiam esta tarefa.

Os dados expressos pelos CE's são conhecidos e possíveis de serem determinados em tempo de projeto, são dados seguros e que podem ser previamente determinados, em contrapartida ao contexto que é dinâmico, dependente do tipo de aplicação executada, pois é criado em tempo de execução, ou seja, quando uma interação ocorre. Nesse ponto da construção e execução da aplicação é possível enxergar a fusão da análise do contexto em IA com a análise do ambiente em Computação Ubíqua.

Para que seja possível realizar a correta extração das informações que possam melhor representar o contexto, um conceito coreano (DEY, 2001) adota os chamados 5W's para representam 5 dados considerados indispensáveis para a correta caracterização e definição dos elementos do contexto. Os 5W's são: Quem (*Who*), Onde (*Where*), Quando (*When*), O que (*What*) e Por que (*Why*).

Esses dados permitem uma coleção de dados satisfatória para a correta interpretação do contexto, uma vez que são utilizados para extrair informações básicas dos elementos do contexto. É possível ainda realizar uma abordagem hierárquica em torno desse conceito, que divide os 5W's em dois principais grupos: o contexto de alto e baixo nível. A divisão pode ser observada na Figura 11.

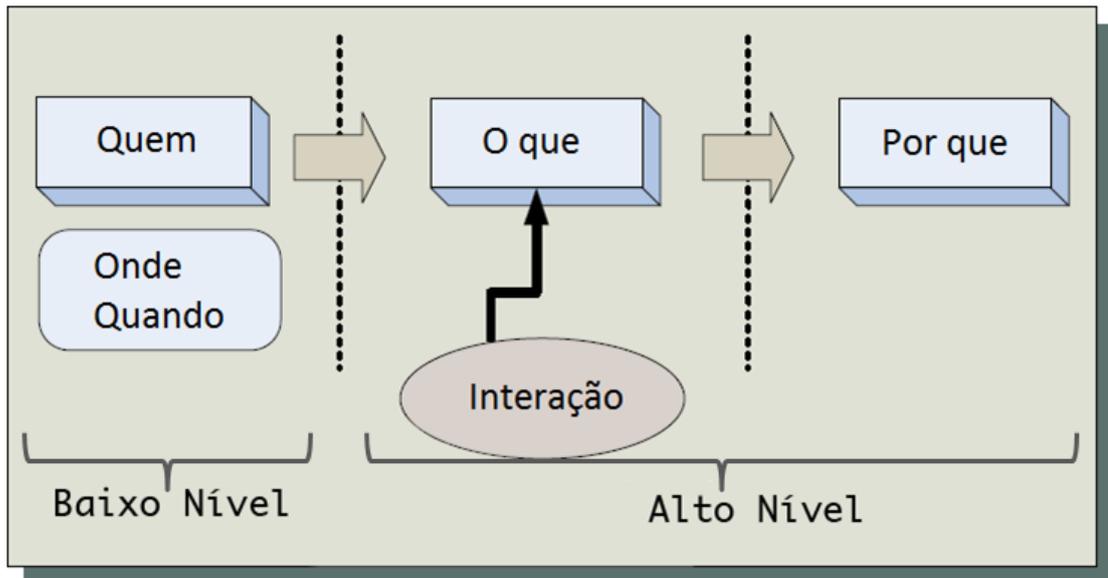


Figura 11 - Níveis de Contexto. Fonte: Traduzido de "Context-Awareness for Ubiquitous Computing System" (KANG *et al*, 2005).

O contexto de baixo nível são os dados adquiridos de forma explícita e determinam "Quem", "Onde" e "Quando". Essas três informações são definidas como elementos de contexto por representarem explicitamente seus objetivos e seguem um padrão ao descrever seus dados. Em contrapartida a esse nível, o contexto de alto nível é uma composição das informações do baixo nível somada às interações e interferências realizadas no elemento.

O contexto de alto nível ainda pode ser dividido em duas categorias: "O que" e "Por que". "O que" descreve as interações (alterações) realizadas e é definido como relação das entidades entre si através das interações. O "Por que", por sua vez, corresponde ao contexto implícito da interação baseando-se em eventos ocorridos naquele determinado período, e descreve a intenção da interação atual, caracterizando-se como contexto intencional.

### 2.2.1. Contexto de baixo nível

O contexto de baixo nível, como descrito anteriormente, é composto por 3 (três) dos 5W's, sendo então o responsável por representar dados explícitos dos elementos.

Para obter-se o dado "Onde" de uma entidade é preciso informar o sistema sobre a localização de um elemento. A localização é uma referência atribuída a partir de um ponto, logo, a sua determinação é feita desde que haja uma entidade de referência, ou seja, esta informação só pode ser determinada a partir do momento em que se assume a existência de algumas identidades no contexto ou referência em um ambiente estático.

Em contrapartida com o "Onde", o "Quando" é obtido de uma forma simplificada, pois é o momento ou período em que aquela informação é capturada e não depende do comportamento de uma entidade ou local, baseia-se apenas no tempo cronológico. Esta informação combinada com o "Onde" pode fornecer informações importantes ou relevantes ao sistema, como a possibilidade de calcular a estação do ano à qual o contexto pertence e até efetuar consultas de previsão meteorológica.

Adota-se o "Quem" como entidade ativa que pode se comportar de forma independente, ou seja, uma entidade que altera, que não é estática. Por exemplo, uma impressora pode ser considerada como "Quem", mas o papel não, porque o papel não pode agir por si só: ele é estático. Esta determinação das entidades ativas é útil para focar-se nos elementos que podem ser analisados como relevantes na análise do contexto, enquanto as entidades passivas são consideradas como os objetos que são afetados pelas entidades ativas ("Quem") (KANG *et al*, 2005).

### **2.2.2. Contexto de alto nível**

O contexto de alto nível possui um grau de complexidade superior ao contexto de baixo nível. A análise dos dados é realizada de uma forma diferente, pois seus dados podem ser ambíguos ou imprecisos. Apesar de diferente, é baseado e construído sobre o contexto de baixo nível, e expressa 2 (dois) dos 5W's.

Neste nível de contexto "O que" representa as interações entre as entidades. Este elemento do contexto necessita de cálculos computacionais para ser determinado, ou seja, não é um dado explícito, sendo o resultado da observação do contexto de baixo nível somado ao processamento computacional em função das regras da aplicação, adequando então as alterações realizadas no contexto à aplicação.

O "Por que" do contexto de alto nível é a informação que representa a intenção da interferência realizada pela entidade na aplicação. Este dado é obtido a partir da análise realizada na obtenção do "O que" em paralelo ao objetivo da aplicação. Isso exige do interpretador do contexto que possua o conhecimento dos objetivos do sistema em questão, porém estes objetivos podem nem sempre serem previstos, sendo assim, em alguns casos utiliza-se IA para análise do sistema para a formalização de objetivos. A intenção da entidade que realiza a interferência pode não ser explícita, ou seja, subjetiva e em alguns casos ambígua.

## CAPÍTULO 3 - METODOLOGIA

Uma aplicação que realiza a análise de contexto em altos e baixos níveis exige um processo de concepção diferenciado do padrão, portanto o presente trabalho tem como objetivo o levantamento e implementação de uma aplicação sensível ao contexto em interface tangível e de caráter colaborativo. É utilizado um jogo de Quebra-cabeça como estudo de caso. Esta aplicação foi escolhida pela sua lógica de interação que se assemelha ao desenvolvimento de projetos e trabalhos cooperativos.

O processo de concepção deste estudo pode ser dividido em 4 (quatro) principais etapas (Figura 12) que abordam (i) levantamento de requisitos e estruturação do projeto de software em questão, (ii) escolha e estudo das ferramentas para desenvolvimento da aplicação de interface tangível, (iii) prototipação da aplicação e (iv) validação e avaliação do resultado obtido.



Figura 12 - Diagrama de atividade. Fonte própria.

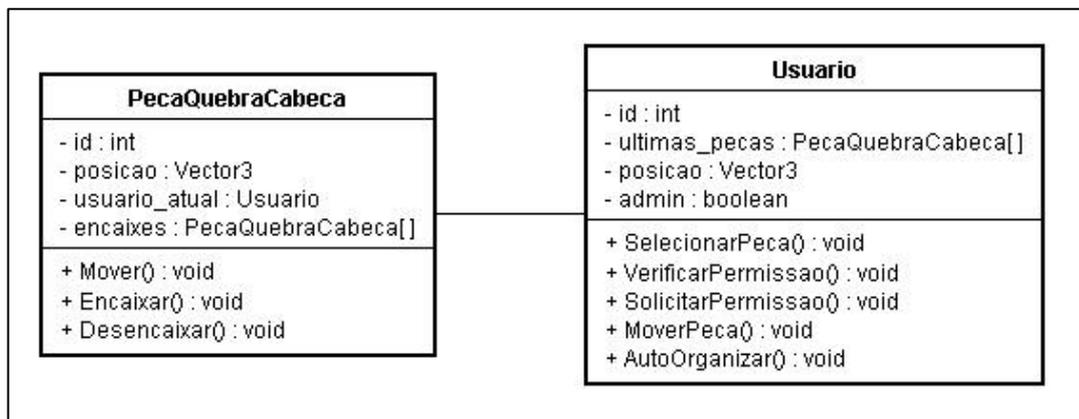
### 3.1. Levantamento de Requisitos

Tratando-se de uma aplicação, um estudo de caso, inicia-se pelo seu planejamento ou levantamento de requisitos para obter o resultado desejado. Essa etapa do processo de desenvolvimento exige a organização dos elementos funcionais do sistema, formalizando assim as estruturas e regras do projeto. Esse planejamento deve prever situações em que o sistema analisa o contexto em que está imerso, o momento em que deve atuar e qual a forma mais coerente de realizá-lo. No seu âmbito geral, esta etapa gera documentos como diagramas de atividades e diagramas de classe, sendo a concepção dessa documentação extensível à

etapa de implantação da aplicação, onde espera-se especificar descrições sobre as classes e estruturas afim de facilitar manutenções, modificações e potenciais expansões.

A partir dos requisitos explicados a seguir, obteve-se a modelagem das classes representada na Figura 13, onde cada usuário e peça do jogo possui um código de identificação próprio, uma posição cartesiana no plano e atributos específicos para cada classe com seus respectivos métodos.

Com base no levantamento de requisitos, nesta etapa do projeto pode-se realizar um balanço geral sobre o resultado obtido. Este conjunto de documentos servem de base para a construção e aplicação das regras do protótipo. Todos os objetivos traçados no documento base são abordados em aplicação.



**Figura 13 - Modelagem de Classes para a aplicação proposta. Fonte própria.**

O desenvolvimento de uma aplicação sensível ao contexto exige um levantamento do estudo de caso diferenciado, pois é preciso prever as situações onde o sistema deve se adaptar ao meio em que está inserido, assim como a aplicação ao usuário, torna-se então necessário a rígida especificação, documentação e levantamento de regras do sistema. Este tipo de atividade num ambiente ideal abrange todas as possibilidades, mas nos requisitos restringiu-se apenas à hierarquização e à individualização dos usuários que são o foco do estudo proposto por serem considerados relevantes para a aplicação em questão.

### **3.1.1. Movimentação das peças do jogo**

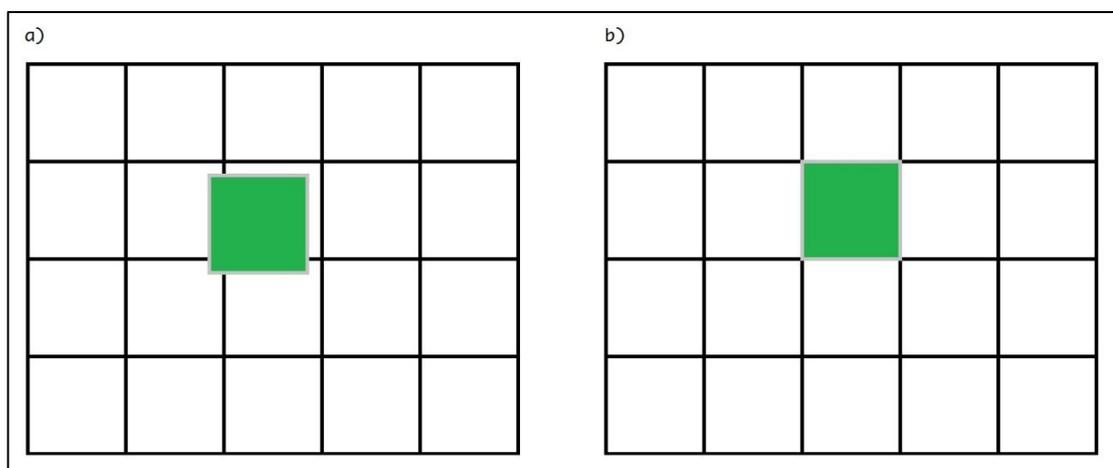
O jogo de Quebra-cabeça (Figura 14) em si não possui um número determinado de peças desde que seja possível construir uma imagem ao reunir todas elas. Nesta aplicação são utilizadas 25 peças formando uma matriz 5x5.



**Figura 14 - Exemplo de jogo Quebra-cabeça. Fonte própria.**

As peças do jogo de Quebra-cabeça possuem encaixe único, ou seja, só são corretamente encaixadas se forem anexadas às peças previamente determinadas. Em nível de aplicação cada peça possuirá um código identificador único que servirá de referência para verificar quais delas podem realizar encaixes.

Quanto à movimentação das peças no plano, pode ser realizada de forma livre, bastando selecionar a peça e "arrastá-la" à sua posição de destino desejada, porém ao terminar a movimentação ela deve se encaixar em uma grade predeterminada (Figura 15), evitando que a peça não realize os encaixes corretamente.



**Figura 15 - a) Exemplo de usuário soltou a peça em um espaço qualquer; b) Exemplo de sistema realizando encaixe da peça na grade. Fonte própria.**

### 3.1.2. Permissões de movimentação

É desejado que haja uma hierarquia entre os usuários da aplicação afim de organizar o ambiente colaborativo. A divisão dos usuários consiste em 2 (dois) grupos: os usuários administradores e os usuários comuns.

Foi estabelecido que ao selecionar uma peça o sistema verifica se esta já não está encaixada a uma outra. Caso a peça esteja livre não haverá impedimentos, porém se a peça já estiver anexada a outra, o usuário simples irá solicitar permissão do administrador, porém se a interação for do administrador não haverá impedimento.

Também foi determinado que ao posicionar a peça que se está manipulando próxima a uma outra peça que permita o seu encaixe não há necessidade de controle de hierarquia, pois identifica-se que há uma colaboração na construção do objetivo.

Neste ponto é importante observar que o usuário movimenta sua peça livremente pelo plano, alterando apenas o contexto de baixo nível. Porém, quando ele a "solta", a análise gira em torno do contexto de alto nível, tentando identificar "O que" o usuário está querendo fazer (encaixar, desencaixar ou mover na grade) e também o "Por que" (o usuário está se aproximando do objetivo final, ou está se distanciando).

Baseado nas regras de negócio estabelecidas acima, foi construído um diagrama de atividade para uma melhor documentação e ilustração (Figura 16).

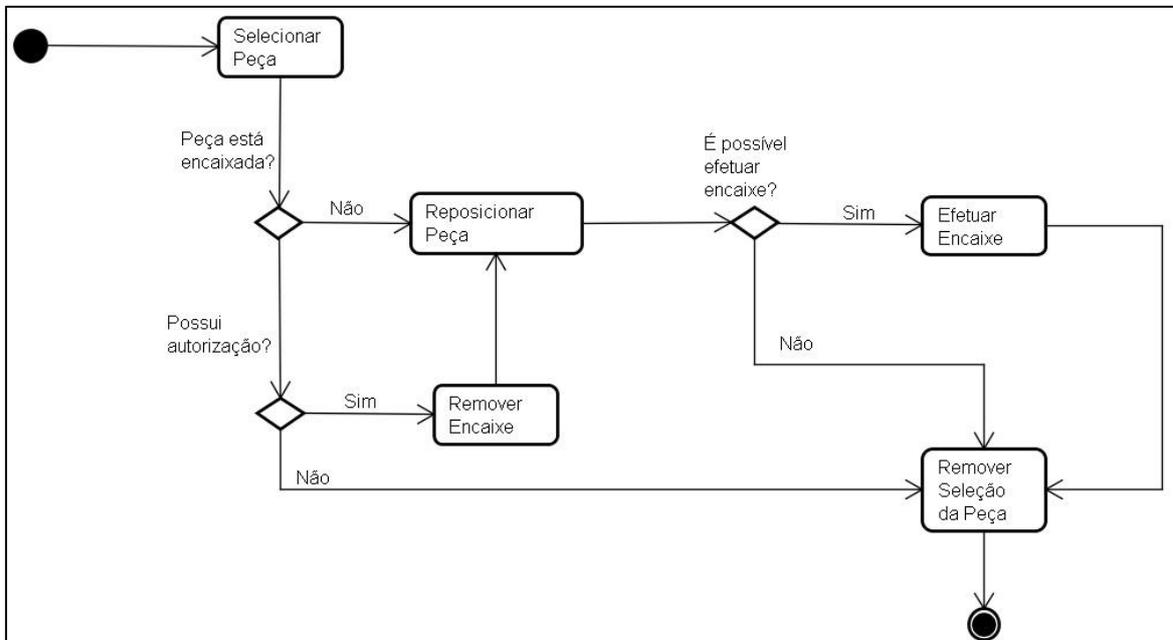


Figura 16 - Diagrama de Atividade de Mover Peça. Fonte própria.

### 3.1.3. Auto-Organização

Ao iniciar a aplicação todas as peças estarão sobrepostas no mesmo ponto na grade o que obriga o usuário a primeiramente organizar as peças antes de começar a montar o Quebra-cabeça. Quando algum dos usuários movimentar quatro peças, deve-se realizar uma análise do padrão de organização dessas peças para que o sistema possa concluir essa organização (Figura 17). A análise do contexto de alto nível nesse ponto da aplicação se faz fortemente presente, pois há a tentativa da interpretação do "Por que" e "O que" o usuário está fazendo.

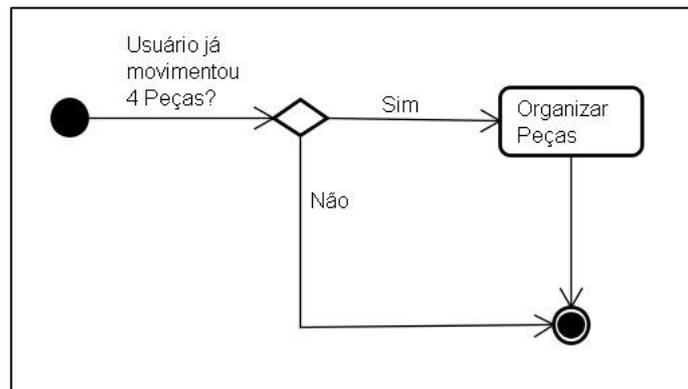


Figura 17 - Diagrama de Atividade de Auto-Organização. Fonte própria.

## 3.2. Estudo de Ferramentas

Esta etapa do projeto consiste no estudo das ferramentas que são utilizadas para realizar o desenvolvimento da aplicação, sendo elas a TUI, a Community Core Vision (CCV), TUIO e o Multitouch for Java (MT4j). Os requisitos previamente estabelecidos são essenciais para objetivar o uso das ferramentas em questão.

### 3.2.1. TUI

A aplicação proposta neste projeto é executada em TUI, portanto trata-se de um ponto relevante para a pesquisa. A estrutura disponível para o desenvolvimento deste projeto foi previamente concebida pela instituição de pesquisa. A interface é exibida na Figura 18 que é mantida (executada) por um *desktop* conectado ao projetor e à câmera de maneira bem semelhante à *reacTable* anteriormente descrita no capítulo 1.

A superfície da TUI é translúcida composta por uma placa de acrílico coberta com plástico semi-transparente. Esta característica permite que imagens sejam projetadas na superfície assim como a captura de imagens de objetos em contato com a mesma.

A câmera localizada no fundo da estrutura é responsável por observar as movimentações (interações) do usuário em contato com a superfície e transmiti-las para o

*desktop* que irá processar os dados e fornecer uma resposta para o usuário através do projetor. Em função da altura da estrutura, é utilizado um espelho para refletir a imagem de saída provida pelo projetor permitindo uma maior distância entre o projetor e a superfície, o que garante uma maior área de projeção.

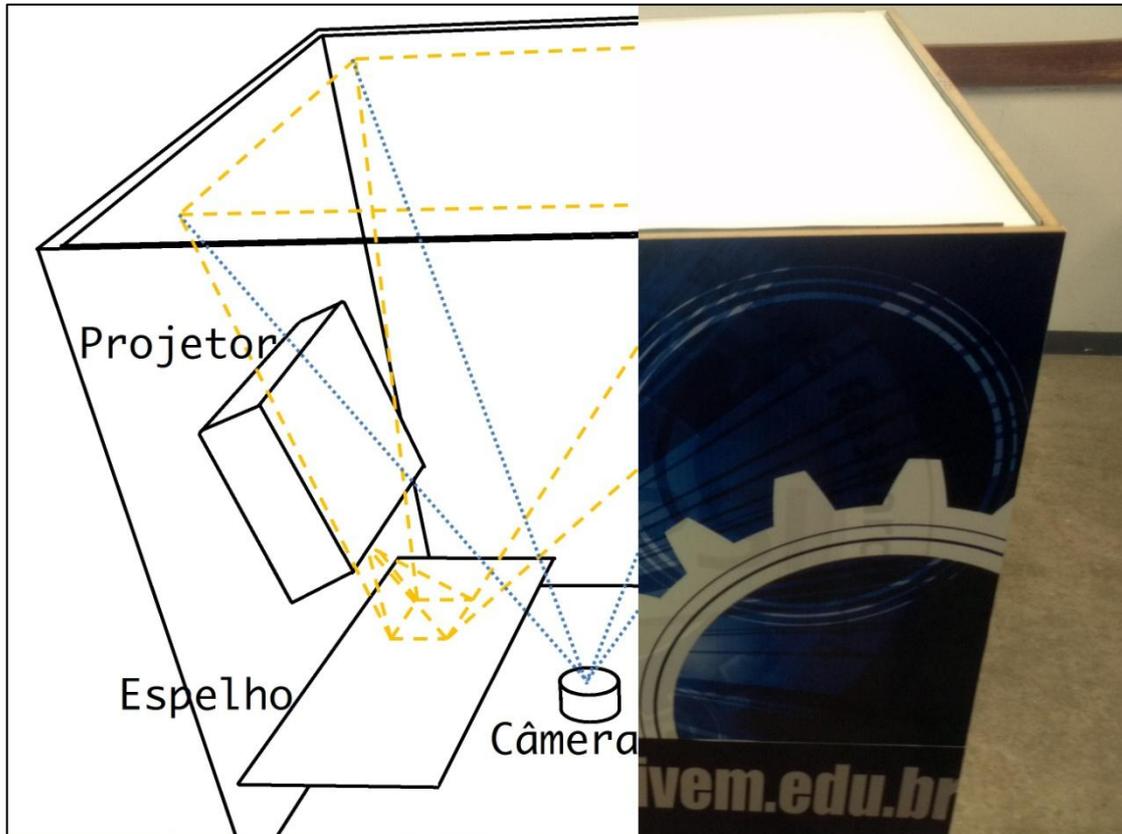


Figura 18 - Estrutura TUI disponível para o desenvolvimento do projeto visualizada em corte. Fonte própria.

### 3.2.2. Community Core Vision (CCV)

O CCV (Figura 19) é uma aplicação de visão computacional *open-source* (código aberto) multi-plataforma (disponível para vários sistemas operacionais) que caracteriza-se em receber uma entrada *stream* de vídeo (através de uma câmera), realizar o rastreamento da imagem e responder com dados do toque (detecção, atualização da posição e término) ou outros eventos (fiduciais da *reactTable* e posições).

Esta ferramenta é utilizada na construção de aplicações multi-toque pois suporta este tipo de detecção. Ela pode inclusive usar-se da GPU para efetuar a varredura, o que agrega mais poder de processamento.

A escolha desta ferramenta foi realizada pela sua interface simples, grande possibilidade de configuração e tratamento das imagens das câmeras e também pela sua característica multi-plataforma.

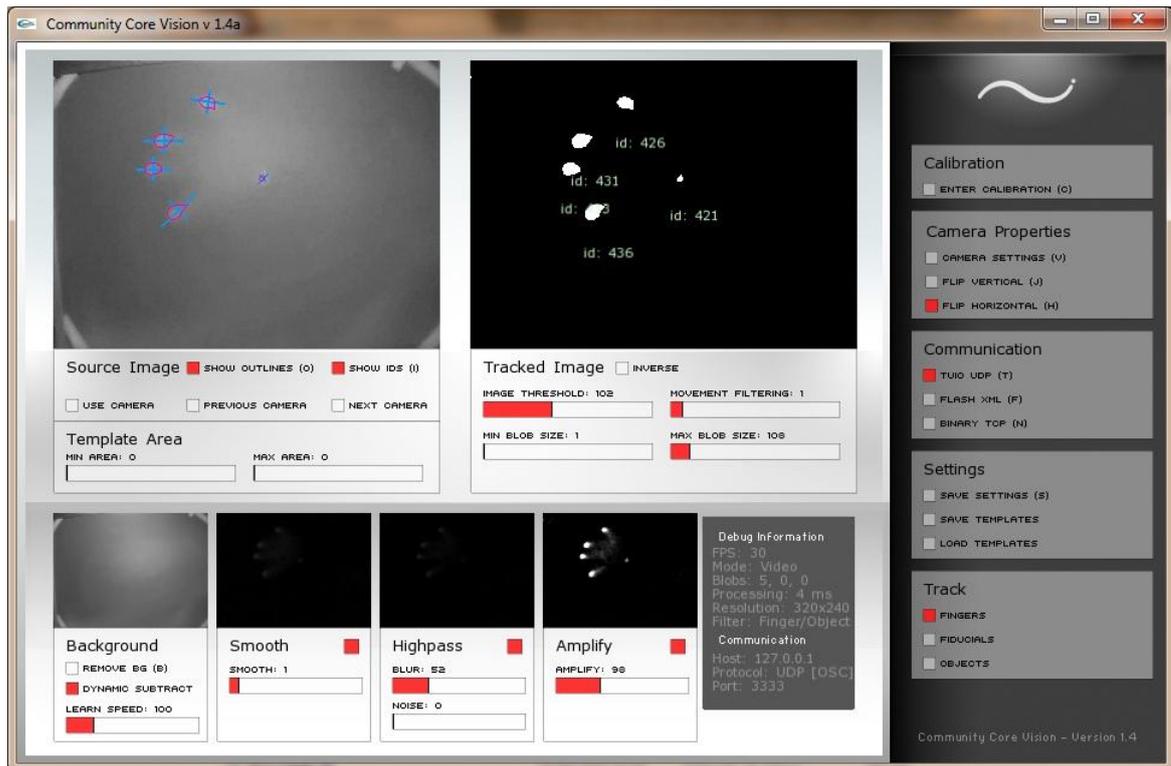


Figura 19 - CCV em Execução identificando as entradas e saídas. Fonte própria.

### 3.2.3. Protocolo TUIO

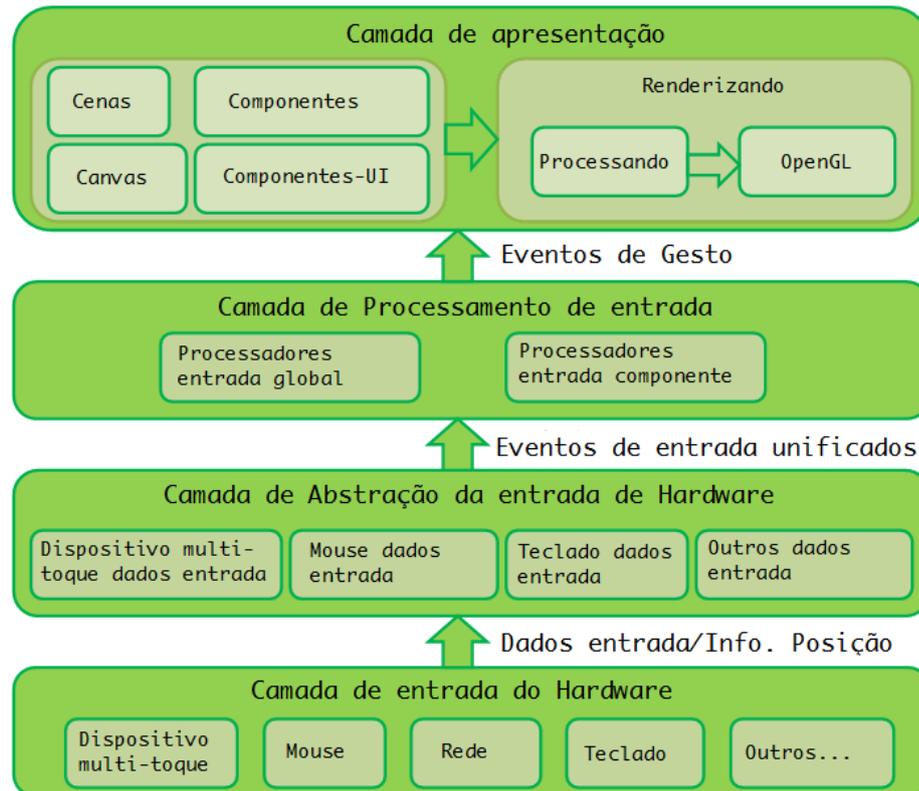
O protocolo TUIO é largamente utilizado no desenvolvimento de aplicações tangíveis. Ele é o responsável pela comunicação entre a aplicação de visão computacional com a aplicação que utiliza entrada de dados tangíveis.

A saída de dados do CVV nativamente pode ser gerada no protocolo TUIO, pois ele permite a transmissão e representação de um dado de descrição abstrata identificada na superfície tangível incluindo o toque e o estado de objetos tangíveis (marcadores fiduciais por exemplo).

Existe um número crescente de aplicações de visão computacional e aplicações cliente que adotam o protocolo TUIO como padrão, tal fato ocorreu através da utilização desse protocolo na *reactTable* que acabou popularizando esse formato *open-source*.

### 3.2.4. Multitouch For Java (MT4j)

O MT4j é um framework escrito em Java criado para agilizar o desenvolvimento de aplicações com visual de grande opções de elementos de interface e orienta-se por suportar diferentes tipos de dispositivos de entrada com um foco especial ao suporte multi-toque. A Figura 20 ilustra o fluxo do framework em questão.



**Figura 20 - Fluxograma de funcionamento do MT4j (MT4J, 2011).**

O MT4j divide a sua estrutura em 4 principais camadas:

- Camada de entrada do Hardware: responsável por capturar os dados providos dos diferentes dispositivos de entrada (Mouse, rede, teclado, dispositivo multi-toque e outros) e repassar à sua camada superior;
- Camada de Abstração da entrada de Hardware: responsável por unificar os dados, convertendo as entradas de hardware em um evento de entrada unificado;
- Camada de Processamento de entrada: dividida em duas partes principais: reconhecimento das entradas em sua totalidade como potenciais interferências e em seguida tipos de interferência como as multi-toque ou toque único (rotação, translação e dimensionamento por exemplo);
- Camada de apresentação: responsável por gerar os componentes que serão exibidos e em seguida renderizá-los utilizando OpenGL.

O OpenGL e o MT4j são de código-aberto, permitindo um ambiente de desenvolvimento colaborativo, que é apoiado pelas suas comunidades, são flexíveis por permitirem modificações em camadas inferiores das aplicações e não impactam no custo do projeto.

### 3.3. Prototipação

A partir do levantamento de requisitos e do estudo das ferramentas a serem utilizadas nesta aplicação, fundem-se os conhecimentos para a prototipação, ou seja, a implementação da aplicação propriamente dita.

#### 3.3.1. Comunicação entre aplicações

O primeiro passo tomado na construção da aplicação é a configuração e teste da comunicação entre a aplicação de visão computacional (CCV) utilizando o protocolo TUIO para se comunicar com o framework de desenvolvimento (MT4j). Na Figura 21 é demonstrada a identificação do marcador fiducial de código 0 (zero) no CCV e a sua projeção sendo realizada na aplicação exemplo do MT4j.

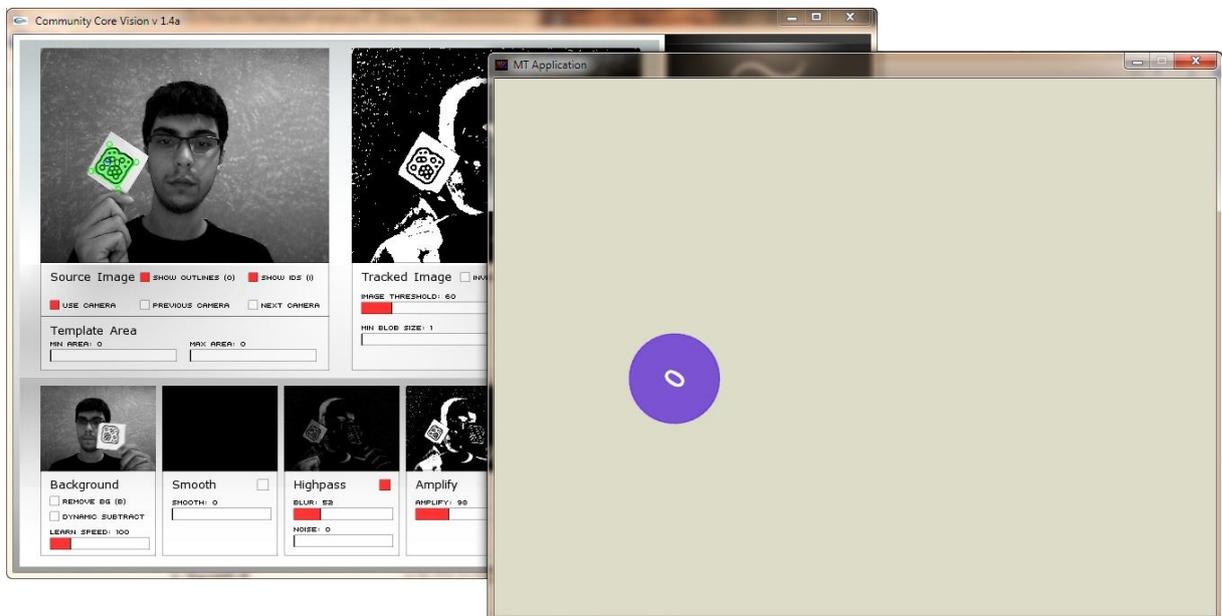


Figura 21 - Identificação do fiducial no CCV e no MT4j. Fonte própria

Neste ponto do desenvolvimento foi possível detectar um conflito na aplicação de visão computacional quando se opta por reconhecer marcadores fiduciais e toques simultaneamente. Essa tentativa resultou no reconhecimento da imagem do marcador como toque. Dessa forma optou-se apenas por um dos tipos de reconhecimento, o feito por toque.

#### 3.3.2. Peças do jogo e a grade de alinhamento

Ao utilizar o MT4j nota-se que existem exemplos junto ao seu pacote que abordam as inúmeras ferramentas do software. Há aplicações básicas com a inserção de palavras no ambiente, criação de polígonos, reconhecimento de gestos e manipulação de polígonos assim

como aplicações avançadas projetando um objeto tridimensional modelado, simulação de características físicas tais como colisão e gravidade, visualização de partículas, comportamento de fluídos, entre outros. Nesse mesmo conjunto de exemplos há uma aplicação que simula um Quebra-cabeça, porém não possui nenhum mecanismo de identificação de usuário e também não realiza análise de contexto. Em um primeiro momento foi optado por alterar esta aplicação, uma vez que o jogo de Quebra-cabeça poderia ser reaproveitado, porém seus elementos apresentaram muitos conflitos ao serem desacoplados, não permitindo um bom uso dos recursos ali dispostos. Esse processo de reaproveitamento foi descontinuado e a partir de uma nova aplicação foi construído o Quebra-cabeça e sua grade de alinhamento.

Buscando uma maior facilidade para o desenvolvimento foi feita uma alteração na visualização das peças em relação ao jogo original que consiste em não utilizar imagens como conteúdo das peças, mas sim números que na sua sequência crescente completam o objetivo do jogo. Essa alteração facilita a visualização das peças para efeito de teste.

O usuário pode movimentar ("arrastar") as peças livremente, porém com a implementação da grade, ao "soltá-la" a peça se encaixa automaticamente na grade. A grade foi implementada através do ponto central da peça (Figura 22).



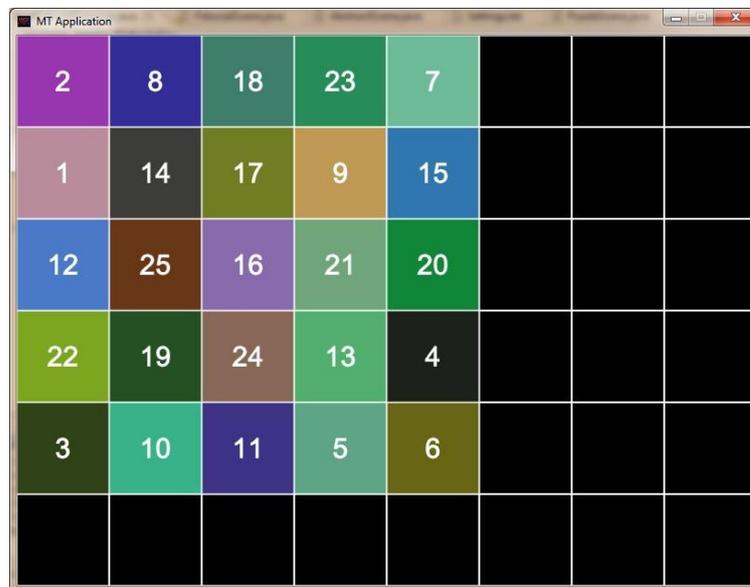
**Figura 22 - Grade de Peças, onde a peça número 1 está em movimento. Fonte própria.**

### 3.3.3. Auto-Organização

Ao iniciar a aplicação todas as peças encontram-se todas na mesma posição central, sobrepostas umas às outras, sendo assim exige-se do usuário a organização delas na grade. Porém com a funcionalidade de auto-organização, é necessário que o usuário apenas defina um padrão, cabendo ao sistema prosseguir com a organização.

O desenvolvimento restringiu-se a dois padrões por conta do nível de complexidade desta tarefa que pode resultar em erro por interpretação ambígua. Define-se então os padrões em matriz e borda.

O padrão matriz consiste na organização da peça em formato de matriz, ou seja, de forma aleatória as peças são colocadas em linhas e colunas, formando um quadrado, conforme Figura 23.



2	8	18	23	7		
1	14	17	9	15		
12	25	16	21	20		
22	19	24	13	4		
3	10	11	5	6		

Figura 23 - Padrão de Organização Matriz. Fonte própria.

O padrão borda consiste na organização das peças nas bordas da grade da aplicação, deixando o espaço central sem nenhuma peça (Figura 24).

Para obter este resultado foi utilizado um identificador de padrões bastante utilizado na área de IA, trata-se de um neurônio artificial denominado *Perceptron* que se baseia em separar linearmente vetores de entrada em classes de padrão [RUSSEL e NORVIG, 1995]. Isso implica na conversão da matriz (grade) em um vetor para a análise do padrão, resultando na identificação do mesmo.

Observa-se que a grade possui mais posições para peças do que peças do jogo, mas essa diferença é desejada para que possa se desenvolver essa funcionalidade de auto-

organização, caso contrário não seria possível, pois não haveriam espaços livres disponíveis para o posicionamento das peças.

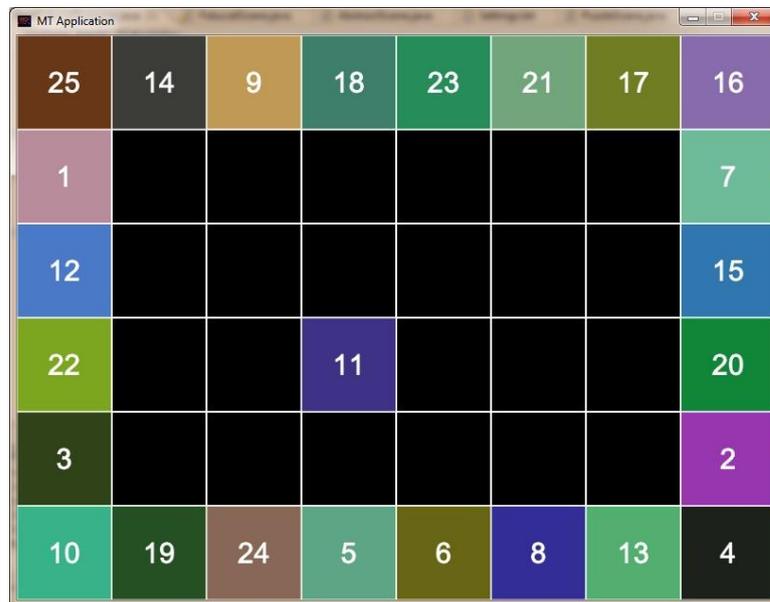


Figura 24 - Padrão de Organização Borda. Fonte própria.

### 3.3.4. Permissão de movimentação

O objetivo do jogo é solucionar o problema, ou seja, encaixar todas as peças às suas respectivas posições, porém no caso do usuário solicitar um desencaixe de uma peça que já estava posicionada corretamente deve-se obter a aprovação do administrador para que a operação seja efetuada.

A identificação do usuário não é realizada por meio do simples toque na mesa (interface), pois essa interferência pode ser similar ou igual entre dois usuários distintos, uma vez que a aplicação de visão computacional consegue apenas reconhecer o toque e não a impressão digital do usuário. Dessa forma, essa identificação pode ser realizada através do uso de marcadores fiduciais onde cada usuário possui um. Os fiduciais são os aplicados na reactTable, onde cada um deles possui seu código único, permitindo a identificação de até 256 usuários distintos.

A funcionalidade de permissão de movimentação pode ser visualizada na Figura 25 onde é solicitado o desencaixe da peça "1" com a peça "2". A confirmação do botão "Sim" só pode ser acionada pelo administrador, podendo então qualquer usuário usar a opção não para fechar a janela.



Figura 25 - Solicitação de permissão para desencaixe. Fonte própria.

### 3.3.5. Análise dos Resultados Obtidos

A aplicação construída atende aos requisitos levantados, cumprindo assim seu papel de estudo de caso.

Ela permite o auto-encaixe das peças em grade, não as deixando soltas pela aplicação. Também é possível auto-organizar as peças a partir de um padrão de organização que deve ser indicado pelo usuário. É possível também a hierarquização de seus usuários, garantindo que o projeto seja administrado de uma forma que a construção seja livre e a desconstrução (desencaixe das peças) seja restrito à vontade do administrador.

A hierarquização de usuários e as permissões de manipulação foram pontos tratados com êxito, onde o resultado esperado foi o alcançado. A auto-organização das peças, por sua vez, ficou restrita a duas opções apenas, quando na proposta original era desejado que a aplicação reconhecesse o padrão e pudesse reproduzir a sequência, mas essa restrição à essas opções disponibilizadas ainda permite ao usuário ter essa funcionalidade de auto-organizar.

Ao utilizar o MT4J, foi encontrada uma aplicação que já implementa o jogo Quebra-Cabeça. Foram alterados alguns parâmetros como a quantidade de peças e configurações de ajuste de resolução espacial. Porém a manipulação das peças do Quebra-cabeça não foi alcançada, o que gerou a construção de um novo modelo de jogo.

Foi construído o protótipo em linguagem orientada a objeto para a estruturação dos algoritmos de análise de contexto, assim como a manipulação das entidades. Nesse caminho, o MT4j permitiu a identificação e distinção dos fiduciais, tratando assim o contexto de baixo nível. A construção de classes unida a essa funcionalidade do framework, permitiu que os elementos do contexto fossem manipulados de forma mais simples do que da forma convencional (teclado e mouse).

Com base nessa satisfação dos requisitos levantados utilizando os conceitos de sensibilidade ao contexto e interface tangível propostos é possível formular conclusões sobre o trabalho aqui apresentado.

## CAPÍTULO 4 - CONCLUSÕES

O presente trabalho propôs o levantamento e a implementação de uma aplicação sensível ao contexto em interface tangível utilizando como estudo de caso o tradicional jogo Quebra-cabeça.

Verificou-se que é possível implementar um sistema sensível ao contexto multiusuário colaborativo que permite a hierarquização entre os membros da equipe em dois grupos, bem como a atuação em soluções para os usuários por meio de análise de comportamento dos indivíduos utilizando este estudo de caso em interface tangível.

Disposto desse resultado, observa-se que a análise do contexto de baixo nível foi um fator de baixo risco no projeto, pois através da atribuição do marcador fiducial a um determinado usuário solucionou-se com naturalidade uma parte do problema proposto. Entretanto, a análise do contexto de alto nível foi realizado com um nível de complexidade maior, pois precisou ser realizada de três formas diferentes: através da solicitação da intervenção do usuário administrador (desencaixe), através do uso de IA (auto-organizar) e regras de ambiente (grade).

Considerando a interface tangível utilizada, alterou-se alguns comportamentos básicos da aplicação desenvolvida. Foi o ocorrido na identificação de usuários, pois a interação padrão do usuário com a interface é feita através do toque na superfície desta, mas para que a sensibilidade ao contexto consiga identificar os diferentes usuários que estão interagindo ("Quem") é essencial o uso de marcadores fiduciais, pois o simples toque do usuário e seus gestos não permitem a identificação dos diferentes indivíduos presentes.

Este projeto pode se expandir através da utilização de uma camada de persistência para o armazenamento de informações do usuário, permitindo que a análise do contexto seja realizada não só pelos dados atuais da aplicação, mas também pela análise do histórico dos usuários e do ambiente da aplicação. Ainda há como explorar em um trabalho futuro o tratamento de conflitos e exceções que surgem na utilização de interfaces tangíveis.

Conclui-se que um sistema sensível ao contexto tem a capacidade de analisar vários ambientes para atuar sobre o sistema desejado, mas ao tratar interface tangível, observa-se a necessidade de inserir a análise do contexto da aplicação para sua interface, pois os padrões de comunicação com o usuário são diferentes, o que implica em alterações nos padrões de estruturação e desenvolvimento.

## REFERÊNCIAS

DEY, A.K. "Providing Architectural Support for Building Context-Aware Applications," Georgia Institute of Technology, Estados Unidos da América, 2001.

BOUCHET, J.; NIGAY, L. ICARE: a component-based approach for the design and development of multimodal interfaces. Proceedings of CHI'04, 2004, pp. 1325-1328.

BRÉZILLON, P. "Context in Artificial Intelligence: IA Survey of the Literature", Paris, França, v. 18, p. 321-340, 1997.

DEY, A. K. ; ABOWND, G. D. "A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications", v. 16, n.2-4, p. 97-166, 2001.

FITZMAURICE, G. W. ; ISHII, H., BUXTON, W. A. "Bricks: Laying the Foundations for Graspable User Interfaces", Conference on Human Factors in Computing Systems, Nova Iorque: ACM Press, 1995.

GEIGER, G. ; ALBER, N. ; JORDÀ, S. ; ALONSO, M. "The Reactable: A Collaborative Musical Instrument for Playing and Understanding Music" pp.36-43, 2010.

GMAIL. "Anúncios no Gmail e os seus dados pessoais" Disponível em: <<https://mail.google.com/support/bin/answer.py?hl=pt&answer=6603&topic=1668949>> Acesso em 11/2011.

ISHII, H. ; ULLMER, B. "Tangible bits: towards seamless interfaces between people, bits and atoms". Nova Iorque, Estados Unidos da América, p. 234-241, 1997.

KANG, D. ; AHN, S. ; KOA, H. ; CHOB, W ; PARKC, Y. "Context-Awareness for Ubiquitous Computing System", Coreia do Sul, 2005.

KIRNER C. ; SISCOOTTO, R. A. "Realidade Virtual e Aumentada: Conceitos, Projeto e Aplicações". Livro do Pré-Simpósio IX Symposium on Virtual and Augmented Reality Petrópolis –RJ, 28 de Maio de 2007.

KIRNER,C.; et. al. "Virtual Environments for Shared Interactive Visualization". Workshop of the german-brazilian cooperative program in informatics. Berlin –Alemanha.1995.

KITAMURA, Y. ; et. al. "Real-time 3D interaction with ActiveCube.Ext. Abstracts" CHI, ACM Press, 355-356, 2001.

KRAMER, R. ; MODSCHING, M. ; SCHULZE, J. ; HAGEN, K. "Context-Aware Adaptation in a Mobile Tour Guide", In: Proc. of the 5th International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT'05), LNCS 3554, pp. 210-224, Paris, França, 2005.

MT4J. Disponível em: <<http://www.mt4j.org/mediawiki/index.php/Architecture>> Acesso em 11/2011.

NUIGROUP. Disponível em: <<http://ccv.nuigroup.com/>> Acesso em: 11/2011.

RADICCHI, Adriel O. "Desenvolvimento de um Protótipo de Hardware de Interface Tangível", Marília/SP, 2010.

RUSSEL, Stuart ; NORVIG, Peter. "Artificial Intelligence - A Modern Approach". Prentice-Hall, Nova Jersey, 1995.

SCHILIT, B., ADAMS, N., WANT, R. "Context-Aware Computing Applications", Santa Cruz, Estados Unidos da América, p. 90-99, 1994.

SHNEIDERMAN , B. "Designing the User Interface: Strategies for Effective Human-Computer Interaction" (3rd Edition ed.). Reading, MA: Addison-Wesley, 1998.

SMITH, S. L. "User-system interface". Human Factors Society Bulletin, 1982.

VIEIRA, V. ; TEDESCO, P. ; SALGADO, A. C. "A Process for the Design of Context-Sensitive Systems", Santiago, Chile, p. 143-148, 2009.

VRCIM, Virtual Reality and Computer Integrated Manufacturing Laboratory, Washington

State University, Disponível em: <<http://vrcim.wsu.edu/pages/gallery/>>. Acesso em: 05/2011.

WATANABE R. ; ITOH, Y. ; ASAI, M. ; KITAMURA, Y. ; KISHINO, F. ; KIKUCHI, H.  
"The Soul of ActiveCube: Implementing a Flexible, Multimodal, Three-Dimensional Spatial  
Tangible Interface", Singapura, 2004.

WEISER, M. "The computer for the 21st century", Scientific American, v. 265, n. 3, p. 66-75,  
1991.

XEROX. Disponível em: <<http://www.xerox.com/about-xerox/enus.html>> Acesso em:  
11/2011.