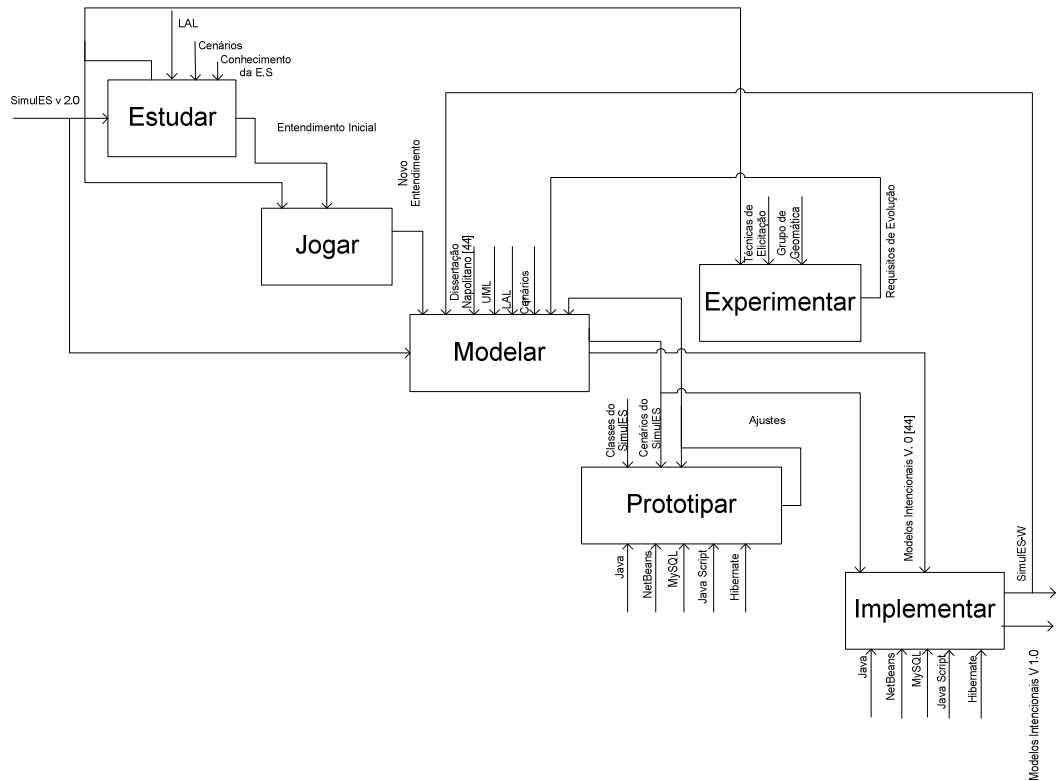


### 3 Procurando Elementos de Evolução do SimuleS

Aqui, é necessária uma explicação sobre o processo utilizado.



**Figura 10 – SADT para SimuleS-W.**

Na Figura 10 descrevemos o processo seguido para a construção da versão do SimuleS v 4.0 ou SimuleS-W apresentada neste trabalho.

Nós começamos com o estudo do SimuleS, para isso procuramos as informações disponíveis do jogo na versão 1.0 [6] e na versão 2.0 [4]. Da versão 2.0 utilizamos os léxicos e cenários e os elementos físicos para jogar e lograr um melhor entendimento do jogo. Estes mesmos elementos serviram de base para criação da primeira modelagem intencional do jogo apresentada no trabalho [44] e que também foi utilizada para a realização da atividade detalhada neste capítulo. Na Figura 10 correspondem as atividades Estudar, Jogar e Modelar.

A modelagem do trabalho [44] foi nosso primeiro insumo para começar a prototipar a SimulES-W, o primeiro protótipo criado do jogo foi apresentado na como Projeto Final de Programação, atividade prototipar da Figura 10. Continuando com a implementação do SimulES-W tivemos a oportunidade de realizar uma atividade de experimentação para identificar elementos de evolução no SimulES além de avaliar a aceitação do jogo por parte dos estudantes, isso aparece na Figura 10 como a atividade Experimentar. Finalmente foram implementadas todas as SDSitutations identificadas e modeladas apresentadas no Capítulo 6 deste trabalho e que corresponde na Figura 10 a atividade Implementar.

### **3.1. Contextualização**

A diferença entre desenvolvimento e evolução é cada vez menos relevante. Hoje, poucos sistemas são novos, o que tem maior sentido focar desenvolvimento e evolução como atividades conjuntas e contínuas. Além disso, é pertinente considerar a engenharia de software como um processo evolutivo, no qual o software muda continuamente, como resposta a mudanças no universo de informações (contexto).

A nossa perspectiva frente à evolução do SimulES é de potencializar seu uso, além de propormos modelar-lo e implementar-lo de modo que facilite futuras evoluções. Para complementar aquilo que se tinha feito até esta instancia (modelagem intencional e começo da implementação), uma experiência realizada na UERJ (Universidade do Estado do Rio de Janeiro) permitiu identificar as expectativas que poderiam ter os usuários frente ao jogo. Para isso utilizamos técnicas de elicitação de requisitos que nos ajudaram a descobrir estas expectativas além de identificar elementos de evolução para incorporar na evolução do SimulES.

Como já se mencionou, o objetivo desta atividade era de recolher informações e elementos para a versão digital do SimulES, como também, para avaliar, in situ, a participação dos jogadores. Isso será detalhado a seguir.

#### **3.1.1. O SimulES**

Antes de relatar o processo de elicitação aplicado neste trabalho para a evolução de SimulES se faz necessário contextualizar nos conceitos mais

relevantes do jogo, conceitos que foram utilizados com os estudantes na experiência realizada.

O texto apresentado nessa seção é fortemente baseado na monografia documentada em [4] que corresponde com a versão 2.0 do SimulES. E nas informações sobre o entendimento do Jogo de aqueles que de alguma ou outra forma tem interagido com ele.

Na Figura 11 é ilustrado o processo que é seguido durante uma partida, segundo o entendimento do jogo na versão 2.0 do SimulES. Na *jogada de inicio* é escolhido o projeto que deverá ser tratado durante todo o jogo, aqui também é escolhido o primeiro engenheiro de software para cada um dos jogadores. Na *jogada de ações* deve se tratar os artefatos dispostos no tabuleiro individual (*aqui é possível construir artefato, inspecionar artefato, corrigir artefato ou integrar artefatos em modulo*). Na *jogada de conceitos e tratamento de problemas* serão usadas aquelas cartas relacionadas com conceitos e problemas típicos da engenharia de software, além disso, os jogadores serão atacados e atacam seus adversários e por ultimo temos *submeter produto*, o primeiro jogador que chegue nesta instância e que satisfaça o critério de qualidade do software e que não tenha penalidade alguma a ser resolvida, ganhará o jogo.

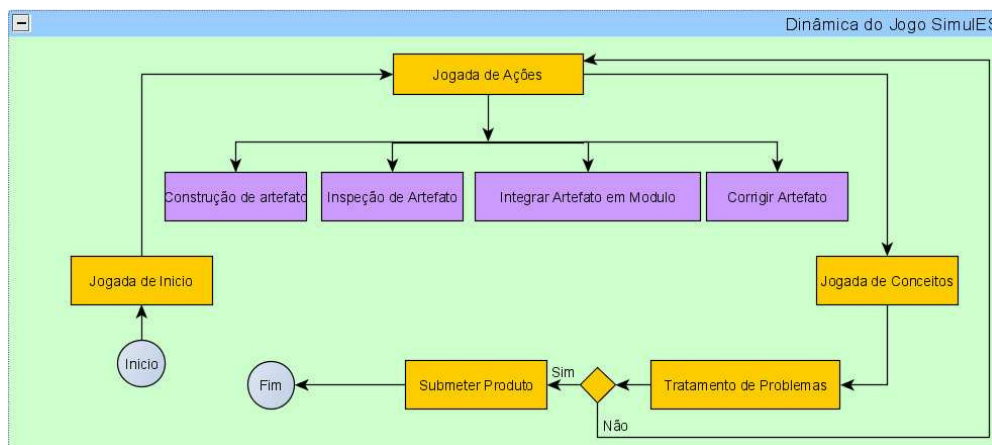


Figura 11 – Dinâmica do jogo SimulES v 2.0.

### 3.1.1.1. Recursos do Jogo

- **Cartões do projeto:**

São recursos dispostos no tabuleiro principal que apresentam as principais informações do projeto a ser desenvolvido pelos jogadores ao longo do jogo. Um exemplo de cartão de projeto é apresentado na Figura 12.

A seguir é feita uma breve descrição das informações contidas neste recurso:

**Descrição:** texto em linguagem natural que descreve as principais características do projeto. Esta descrição torna o jogo mais realista e ajuda o jogador a compreender os principais requisitos do projeto a ser construído. Como informação complementar temos abaixo da descrição as referências para artigos e livros relacionadas ao projeto ou seus principais conceitos.

**Complexidade:** indica quantos *pontos de tempo* um engenheiro de software precisa gastar para construir um **bom** artefato, ou seja, nos informa qual é o valor da *carta branca*. Artefatos de menor qualidade (carta cinza) custam metade deste valor. A complexidade do projeto varia entre dois e quatro.

**Tamanho:** indica quantos módulos devem ser construídos e integrados para que o produto software seja completado. O número máximo de módulos que um projeto pode ter é seis. O cartão indica também a forma que os módulos devem ser construídos. Por exemplo, o projeto “Expert Committee”, apresentado na Figura 5, é composto de cinco módulos, sendo o primeiro formado de duas cartas de requisitos (RQ), uma de desenho (DS) e uma de código (CD).

**Qualidade:** representa o quão livre de defeitos (bugs) deve estar o produto final. O valor varia entre um e cinco. Essa informação indica o número mínimo de módulos sem defeitos necessários para submeter o produto e vencer o jogo. A qualidade do sistema é verificada na fase de submissão do produto, após a integração de todos os módulos especificados no cartão de projeto.

**Orçamento:** determina a quantidade de dinheiro disponível para gastar com o projeto e representa também uma restrição para contratação de engenheiros de software e para o uso de cartas de conceitos. Durante qualquer momento do jogo os valores dos salários de todos os engenheiros somados aos custos das cartas de conceito (caso estas tenham valor) não devem ultrapassar o orçamento do projeto.

|   |       |  |
|---|-------|--|
| Projeto   |       | PR 1   |
| Expert Committee  |       |  |
| Expert Committee é um sistema multi-agente aberto para suporte ao gerenciamento de submissões e revisões de artigos submetidos a uma conferência ou workshop. O sistema oferece suporte a diferentes atividades, tais como, envio de trabalhos, atribuição de um artigo a um revisor, seleção de revisores, notificação da aceitação e recusa de artigos. |       |  |
| [Garcia et al, 2004]  |       |  |
| Complexidade  | 4     | Módulos<br>1 2 RQ + 1 DS + 1 CD<br>2 1 RQ + 1 RT + 1 CD<br>3 1 DS + 1 RT + 1 CD<br>4 2 DS + 1 RT + 1 CD<br>5 1 DS + 2 AJ + 1 CD<br>6 |
| Tamanho   | 5     |  |
| Qualidade   | 2     |  |
| Orçamento   | 220 K |  |

Figura 12 – Exemplo de cartão de projeto Simules v 2.0 [4].

• **Tabuleiro Individual**

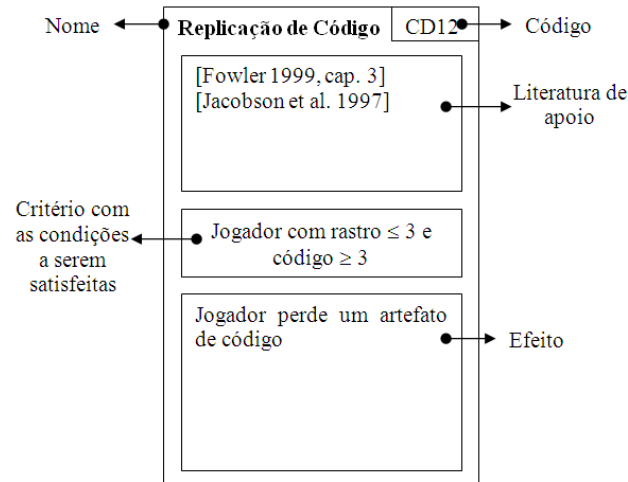
Este recurso é uma área ou folha de papel na qual cada jogador coloca seus engenheiros de software em colunas e os artefatos que vai construindo em linhas, ela deve ficar em frente ao jogador a que pertence e suas informações são públicas para os demais jogadores. Os artefatos podem ser dos seguintes tipos: requisito, desenho, código, rastro e ajuda. A quantidade e tipos de artefatos a construir dependeram do projeto escolhido no início do jogo. A Figura 13 apresenta o tabuleiro do jogo Simules, na ilustração os dois engenheiros de software contratados já construíram vários artefatos. Como pode ser observado, as cartas brancas e cinzas são colocadas nas células do tabuleiro, abaixo do engenheiro que as construíram e nas linhas referentes aos seus tipos.

|            | Engenheiro ES1<br>Janaina                       | Engenheiro ES2<br>Carlos                        | Engenheiros de Software |              |              |
|------------|---|---|-------------------------|--------------|--------------|
|            | Salário: 40 K<br>Habilidade: 1<br>Maturidade: 4 | Salário: 70 K<br>Habilidade: 5<br>Maturidade: 1 | Engenheiro 3            | Engenheiro 4 | Engenheiro 5 |
| Requisitos | [Carta cinza]                                   | [Carta cinza]                                   |                         |              |              |
| Desenhos   | [Carta cinza]                                   | [Carta cinza]                                   |                         |              |              |
| Códigos    | [Carta cinza]                                   | [Carta cinza]                                   |                         |              |              |
| Rastros    | [Carta cinza]                                   | [Carta cinza]                                   |                         |              |              |
| Ajudas     | [Carta cinza]                                   | [Carta cinza]                                   |                         |              |              |

Figura 13 – Tabuleiro individual Simules v 2.0 [4].

- **Cartas de Problema e Conceito**

**Cartas de Problema:** descrevem problemas clássicos de Engenharia de Software resultantes de falhas no Processo de Desenvolvimento de Software. Essas cartas são utilizadas para que os jogadores criem obstáculos ao progresso dos jogos de outros jogadores. Além das informações de nome e um código, as cartas de problema possuem outros atributos como (i) restrições ou condições a serem satisfeitas por um jogador para que seu adversário lhe jogue o problema; (ii) opcionalmente podem ter referências relacionadas; e (iii) efeito ou repercussão que a carta vai dar ao jogo quando for utilizada. Um exemplo desse tipo de carta é apresentado na Figura 14.



**Figura 14 – Exemplo de uma carta problema do jogo SimulES v 2.0 [4].**

**Cartas de Conceito:** descrevem boas práticas de Engenharia de Software e podem ser utilizadas pelo jogador para neutralizar uma carta problema se as condições de ambas cartas estão relacionadas. Os principais atributos das cartas de conceito são: (i) uma literatura de apoio ou referência; (ii) efeito: minimiza ou bloqueia as consequências das cartas de problema ou mesmo melhorar a características do próprio jogo; e (iii) custo (nem sempre presente na carta) que incorre em gastos após o conceito ser aplicado. Um exemplo desse tipo de carta é apresentado na Figura 15.

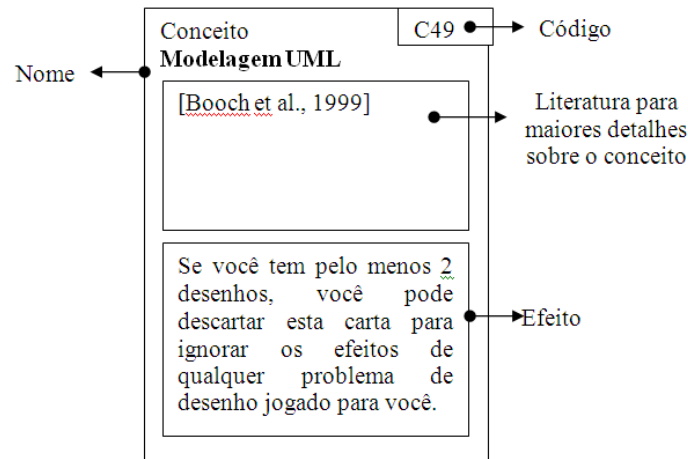


Figura 15 – Exemplo de carta conceito do jogo Simules v 2.0 [4].

**Cartas de Engenheiro de Software:** Com elas o jogador pode executar as atividades referentes ao tabuleiro individual. Os engenheiros constroem, inspecionam e corrigem artefatos que são necessários para o produto software ser completado. Estas cartas apresentam os seguintes atributos: (i) nome; (ii) breve descrição das características do engenheiro; (iii) salário a ser pago ao engenheiro. É necessário verificar que a soma dos salários dos engenheiros de software colocados no tabuleiro individual deve respeitar o limite de orçamento do projeto; (iv) habilidade ou “*pontos de tempo*” que um engenheiro possui a cada rodada para desempenhar ações no jogo, esta deve respeitar a complexidade do projeto; e (v) maturidade que reflete a tendência do engenheiro de software em trabalhar bem em equipe. A habilidade e a maturidade são medidas em uma escala de um a cinco. Figura 16 nos ilustra duas cartas desse tipo.

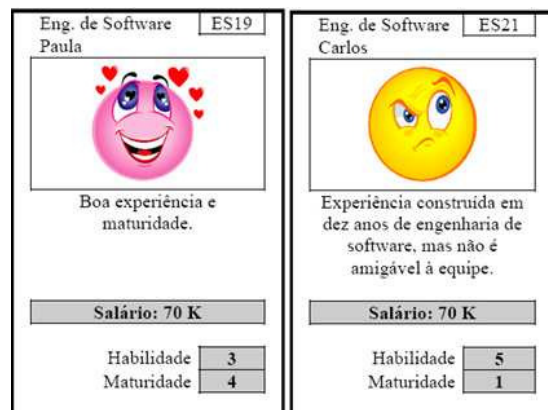


Figura 16 – Exemplo de cartas de engenheiros de software Simules v 2.0 [4].

**Cartas de Artefato:** simbolizam os produtos construídos pelos engenheiros de software que podem ou não conter defeitos (erros ou *bugs*). Além disso, as cartas de artefato podem ser de duas cores, branca ou cinza, de acordo com a sua qualidade. Cartas brancas custam “*pontos de tempo*” segundo a complexidade do projeto (Figura 12) para serem compradas e as cartas cinza custam a metade, as cartas de cor branca contêm defeitos na proporção de 5 cartas para 1 defeito; enquanto nas cartas de cor cinza esta proporção é de 3 para 2. Ou seja, sendo mais custoso construir com carta branca pode trazer maior benefício na hora de fazer inspeção nos artefatos, já que sua proporção de erro é menor.

Para vencer o jogo, é preciso completar os módulos do projeto definidos no cartão de projeto selecionado no início do jogo. Após serem construídos, os artefatos devem ser integrados e depois submetidos. A Figura 17 é um exemplo desse tipo de carta (artefato).

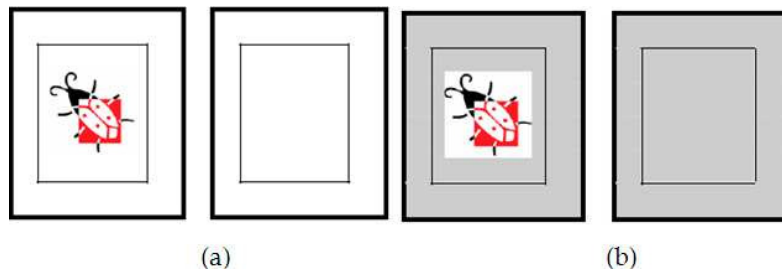


Figura 17 – Artefatos (a) brancos com ou sem erro e (b) cinzas com ou sem erro SimulES v 2.0 [4].

- **Tabuleiro principal**

Este recurso é a área ou folha de papel colocado no centro da mesa do Jogo, em ele estão arranjados os recursos descritos acima, cartões de projeto, cartas brancas e cinzas, cartas conceito e cartas problemas, assim como também os engenheiros de software. O projeto escolhido deve ficar desvirado neste lugar para que todos os jogadores possam vê-lo. Os lançamentos do dado também são efetuados nesta área, como é mostrado na Figura 18.



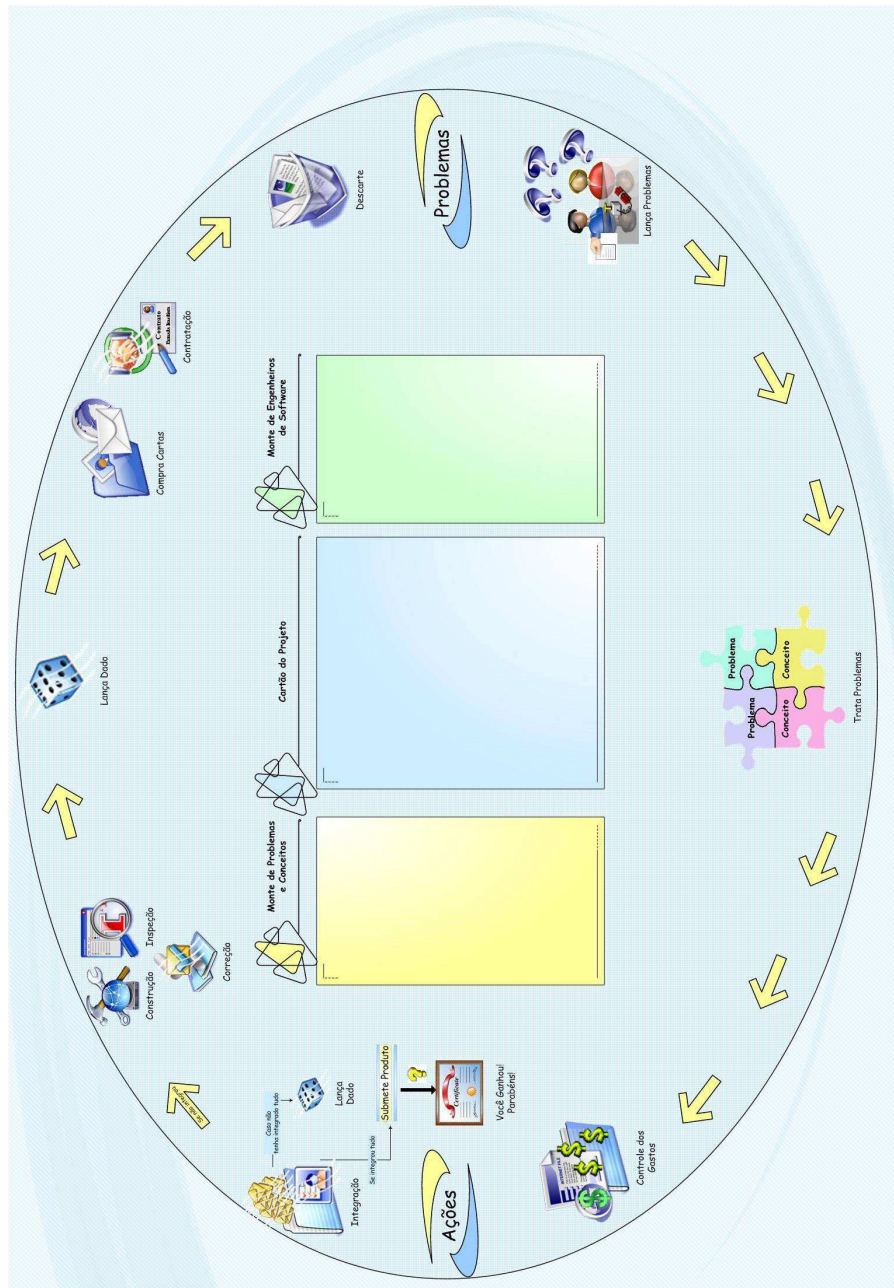


Figura 18 – Tabuleiro principal de SimulES v 2.0 [4].

### 3.1.1.2. Papéis dos Jogadores

Jogador é um participante do jogo que tem por objetivo vencer o jogo. Em uma partida ele pode desempenhar dois papéis:

**Jogador da vez:** participante ativo de todos os cenários do jogo, quando através de seus engenheiros de software ele constrói, corrige e inspeciona artefatos. Ou quando integra artefato ou submete produto.

**Adversário:** oponente do jogador da vez, principalmente, na rodada de conceitos, no tratamento de problemas e na submissão de produtos. Este jogador recebe cartas de problemas do jogador da vez.

### **3.1.2. Técnicas Utilizadas para Identificar Elementos de Evolução para o SimulES**

A engenharia de requisitos é um processo sistemático de desenvolvimento de requisitos. Esta atividade é feita através de um processo iterativo e cooperativo de entendimento do problema, documentação das observações resultantes em vários formatos de representação e análise da precisão do conhecimento obtido [40]. As técnicas de elicitação escolhidas para a evolução do SimulES proposta neste trabalho foram observação e questionário.

#### **3.1.2.1. Observação**

É uma técnica amplamente utilizada porque permite que o engenheiro de requisitos esteja em uma posição passiva no contexto observado [41]. Esta técnica é considerada importante, pois permite que o engenheiro de requisitos tenha um contato suficientemente estreito com o fenômeno sob pesquisa. Assim, esta técnica nos ajudou a fazer anotações da atividade e sobre as interações dos jogadores como também o vocabulário utilizado.

#### **3.1.2.2. Questionário**

Permite ter uma idéia clara de certos aspectos do sistema ou contexto, abrange um maior numero pessoas, além de permitir análise estatística [41]. Os questionários são utilizados para avaliar os sistemas de podendo usar grande número de respondentes, como é o caso de questionários para avaliação de sítios Web [9], onde uma série de perguntas pode ponderar conceitos de transparência que contribuem com a qualidade do software. Os questionários são um instrumento útil naqueles casos onde usar reuniões ou entrevistas para fazer o levantamento de requisitos é difícil [42], este foi o nosso caso uma vez que só tínhamos um espaço de tempo fixo com todos os envolvidos no experimento que efetuamos.

### 3.1.3. Usando as Técnicas de Elicitação para a Evolução do Simules

Foi considerado adequado fazer um laboratório didático-pedagógico para os estudantes, e principalmente aqueles que estão iniciando seus estudos na área da computação.

Foi escolhido um grupo de 8 estudantes do mestrado em Geomática<sup>2</sup> da UERJ (Universidade do Estado do Rio de Janeiro) que conta com estudantes formados em áreas como a estatística, cartografia, biologia, matemática além dos alunos das ciências da computação. A idéia principal deste experimento foi o de fortalecer o ensino de engenharia de software em os estudantes através do jogo Simules além de identificar os pontos fortes e fracos do jogo.

Foram distribuídos em formato impresso os cenários e um diagrama de atividades para contextualizar aos participantes. Os cenários [18] são descrições de situações em linguagem natural semi-estruturado, que nos permitem compreender, padronizar, analisar, rastrear relacionamento de um sistema, como também nos permite descrever as interações entre seus componentes. Nós consideramos o cenário como uma notação adequada como material de apoio no entendimento do jogo.

Na Figura 19 apresentamos um dos cenários que foram utilizados na atividade feita com estes estudantes. Estes cenários pertencem à versão 2.0 de Simules [4].

|                  |   |
|------------------|---|
| <b>Título</b>    | Início de jogo  |
| <b>Objetivo</b>  | Descrever os preparativos para início do jogo.  |
| <b>Contexto</b>  | Tabuleiro de cada jogador colocado na mesa de jogo.   |
|                  | Cartas embaralhadas sobre a mesa.   |
| <b>Atores</b>    | Cada jogador tem uma carta de engenheiro de software.   |
| <b>Atores</b>    | jogador, jogador da vez.  |
| <b>Recursos</b>  | dado, cartas, cartão do projeto, tabuleiro.   |
| <b>Episódios</b> | Jogador joga dado. Restrição: Jogador que tirar o maior número no dado, inicia o jogo e é o jogador da vez.   |
|                  | Jogador da vez escolhe aleatoriamente do monte de cartões de projeto. Restrição: Outros jogadores têm que concordar com o cartão escolhido pelo jogador da vez. |
|                  | Cada jogador compra uma carta de engenheiro de software e coloca-a no tabuleiro. Restrição: Sentido de jogo é horário.  |

Figura 19 – Exemplo de cenário Simules v 2.0 [4].

<sup>2</sup> <http://www.geomatica.eng.uerj.br/>

Ao distribuir todos os recursos e dar explicações básicas estávamos prontos para jogar. Os alunos foram guiados através das diversas fases do processo de desenvolvimento de software para a criação de artefatos de requisitos, desenho, código, rastro e ajuda. Durante a construção dos diferentes artefatos os jogadores foram alvo de cartas de problemas jogadas por os outros jogadores. Igualmente, o jogador podia utilizar este mecanismo dentro do jogo para enfraquecer o jogo dos outros. Com estas ações foram testados conhecimentos relacionados à engenharia de software dos jogadores. Eles também foram aprendendo como usar as cartas conceito que podiam neutralizar certas cartas problemas. Estas cartas são uma compilação de problemas típicos de engenharia de software já testados em outras aulas onde o jogo havia sido jogado.

Depois da experiência do jogo, um questionário foi aplicado. Isso permitiu junto com as informações coletadas através da observação, identificar elementos de evolução do jogo SimulES.

**Tabela 3 – Questionário SimulES aplicado a estudantes da UERJ.**

| Questionario SimulES (Simulador de Engenharia de Software)                                    |  |
|---|--|
| 1. Formação Acadêmica :   | <input type="radio"/> Graduação em :<br><input type="radio"/> Pos-Graduação em:<br><input type="radio"/> Outros  |
| 2. Participa em áreas relacionadas com Engenharia de Software ou desenvolvimento de Software? | <input type="radio"/> Sim<br><input type="radio"/> Não<br><input type="radio"/> ouais  |
| 3. O jogo SimulES cumpriu com as expectativas?  | <input type="radio"/> Sim<br><input type="radio"/> Não<br><input type="radio"/> ouais  |
| 4. O jogo SimulES aportó algum conhecimento novo?   | <input type="radio"/> Sim<br><input type="radio"/> Não<br><input type="radio"/> ouais  |
| 5. Pontos fortes do Jogo  |  |
| 6. Pontos fracos do Jogo  |  |
| 7. Sugestões de melhora   |  |
| 8. Qualificação do Jogo (de 1 até 5)  | <input type="radio"/> 5 Ótimo<br><input type="radio"/> 4 Bom<br><input type="radio"/> 3 Médio<br><input type="radio"/> 1 Ruim<br><input type="radio"/> 0 Sem opinião |

A Tabela 2 mostra o questionário e seu conteúdo. No total, foram 8 questões, 3 do tipo abertas e 5 fechadas. As questões abertas permitiram que os

alunos expressarem suas opiniões livremente e as fechadas permitiram um tratamento estatístico.

O objetivo específico de cada uma das perguntas foi: (i) na primeira questão queríamos identificar qual era a formação acadêmica dos jogadores, e assim conhecer quão familiarizados eles estavam com conceitos de engenharia de software e processos. (ii) a segunda questão nos permitiu saber, independente da formação os jogadores, quanto envolvidos estavam ou participavam em atividades relacionadas com a área. (iii) com a terceira questão pretendíamos determinar a opinião dos estudantes com respeito a atividades práticas e iterativas, e se estas podiam captar o interesse entre os jogadores. (iv) com a quarta questão a idéia era identificar se uma abordagem mais prática e interativa realmente podia apoiar o ensino tradicional. (v) na quinta questão, pretendíamos saber os pontos fortes do jogo como um método de ensino. (vi) na sexta questão, pretendíamos identificar problemas identificados pelos jogadores. (vii) na sétima questão queríamos conhecer, sob a perspectiva dos alunos, quais eram os elementos que poderiam ser melhorados dentro do jogo. (viii) e na oitava e última questão desejávamos avaliar o nível de aceitação do jogo.

### **3.1.3.1. Análise das Perguntas Fechadas**

Observe-se a seguinte análise na Figura 20: (i) na pergunta um (1) pode-se observar que a grande maioria dos alunos não tinha formação em informática, por serem estudantes de Geomática, aplicam conceitos de computação desde a perspectiva de processamento digital de imagens, banco de dados, registros de técnicas, sistemas SIG, cartografia digital e sistemas de posicionamento. Em muitos casos, eles não estão familiarizados com os conceitos de engenharia de software. (ii) do resultado da pergunta dois (2), pudemos determinar que a participação dos alunos em áreas relacionadas à informática é o suficientemente significativa (75%), onde, se sugere que a informação fornecida através da atividade foi de utilidade dos alunos, pois a grande maioria dos estudantes atua na área de informática como usuário, embora não conheçam os processos de desenvolvimento de software (iii) com a pergunta três identificamos que o jogo cumpria com as expectativas dos alunos. Um resultado motivador para continuar com o projeto SimulES. (iv) a pergunta quatro indicou que a maioria dos

estudantes (75%) considerou o jogo como um mecanismo gerador de conhecimento novo em tópicos de engenharia de software. (v) na avaliação final do jogo SimulES os alunos acharam que era bom, sendo que metade deu o grau máximo, ou seja, consideraram o jogo ótimo.

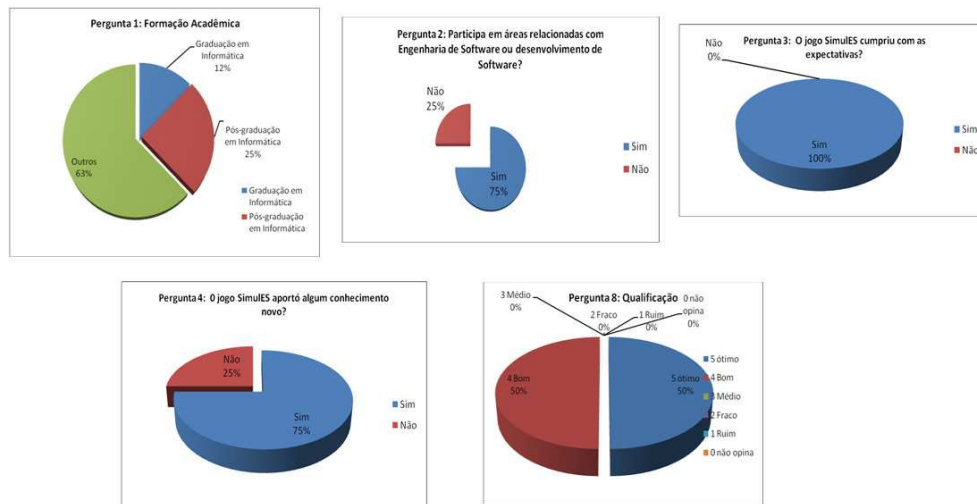


Figura 20 – Representação gráfica dos resultados das questões fechadas.

### 3.1.3.2. Análise das Perguntas Abertas

Os pontos positivos relacionados com as perguntas abertas estiveram entre as informações mais destacadas fornecidas pelos alunos, assim: o jogo promove a capacidade de desenvolver artefatos, atendendo sempre critérios de requisitos, projeto, acompanhamento e testes, ele também incentiva a interação entre os jogadores e trabalho em equipe e gera uma concorrência saudável.

Como pontos fracos os alunos relataram que o jogo foi confuso no início, só após a segunda rodada eles conseguiram melhorar o entendimento, além disso, relataram que a organização da atividade precisava de um planejamento melhor com respeito à distribuição previa da informação do jogo, e que uma explicação mais detalhada no início do jogo deve estar presente.

Como sugestões de melhora os alunos expressaram que o jogo deveria ter menos variáveis e regras mais claras.

### 3.1.3.3. Utilizando a Técnica de Observação

**Interação dos estudantes com o jogo:** os estudantes estiveram atentos às instruções iniciais do jogo, a informação impressa foi distribuída contendo instruções e detalhes do jogo, mas não foi suficientemente atraente para os alunos. Neste aspecto podemos tirar como lição aprendida que o material impresso deve conter informações concisas, de fácil leitura e atraente para captar a atenção dos estudantes, além disso, o pessoal envolvido na atividade deve ser informado previamente.

**Os termos contidos no jogo:** os alunos foram guiados durante atividade toda, o entendimento geral do jogo veio após da realização da primeira rodada.

- Na execução da *jogada de conceitos e problemas* os alunos sentiam se inseguros para realizar suas jogadas principalmente aqueles que tinham pouco conhecimento sobre os conceitos de engenharia de software, ou seja, as informações contidas nas cartas conceito e problema em alguns casos não foram entendidas pelos participantes ou não sabiam como podiam aplicar tanto conceitos como problemas. Nós identificamos que o papel de instrutor é sempre importante para orientar os estudantes principalmente nesta atividade, além disso, é importante destacar que o jogo por si só não permitirá o desenvolvimento e a aprendizagem. O instrutor deve se encarregar de reforçar os conceitos e informações a que o jogo está se referindo, se a ação de jogar desenvolve a compreensão, o instrutor servirá de guia durante todo o processo de aprendizado.

- Identificou-se que *cartas problemas e cartas conceito* podem ser classificadas de acordo com o tipo de aluno que estarão envolvidos na atividade, ou seja, as cartas podem ser tipificadas para reforçar conceitos de testes, desenvolvimento de software, engenharia de software e assim por diante, Dependendo das necessidades do grupo onde a atividade será aplicada.

- Uma análise do conteúdo das *cartas problema e conceito* deve ser realizada para determinar se eles são o suficientemente claros, compreensíveis e legíveis para os usuários que interagem com elas (*cartas problema e conceito*), pois notamos que alguns não conseguiam entender os conceitos. Existem mecanismos que permitem avaliar a transparência da informação [43] e qualidade no conteúdo pode ser avaliados em atributos como: simplicidade, clareza, uniformidade, intuitividade, corretude entre outros.

*SimulES como um ambiente de fácil entendimento:* Basicamente nós identificamos duas questões que foram foco de difícil compreensão para os participantes:

i) A jogada de conceitos consiste em duas atividades principais. Na primeira temos envolvido o tabuleiro individual e os artefatos. Lá pode se construir artefato, inspecionar artefato, corrigir e / ou integrar artefatos em um módulo. Na segunda atividade se faz o lançamento do dado para a compra de cartas de engenheiros de software e cartas de conceitos e problemas. Tudo isso dentro da mesma jogada. As duas atividades que não têm uma aparente relação geraram confusão entre os jogadores, portanto deve ser analisada uma maior desagregação dos cenários, para aqueles casos em que as atividades não estão relacionadas umas com as outras.

ii) A complexidade do projeto está diretamente relacionada à habilidade do engenheiro de software para a construção de artefatos dentro do jogo. Por exemplo: se a complexidade de um projeto é de 2, isso indica que os custos de cada carta branca é de dois (2) “pontos de tempo” da mesma maneira, cada carta cinza custará 1 ponto. Portanto, se o engenheiro de software tem habilidade de quatro (4) só pode obter por jogada duas (2) cartas brancas ou quatro (4) cartas cinza, de modo que cada vez que o jogador faz este movimento deve ter em sua mente este cálculo, coisa que não foi assimilado prontamente no jogo.

Deste item surgiram duas importantes contribuições a nosso trabalho: na primeira identificamos que a modelagem intencional pode ser pertinente quando queremos modelar interação entre os jogadores, ou seja, partir do modelo situacional com o qual contamos [4] para uma modelagem intencional que permita representar elementos de interação. Segundo [7] modelagem intencional é geralmente usada para modelar contextos organizacionais baseados nas relações entre os atores. Os atores na modelagem intencional são participantes ativos deste contexto organizacional, **eles são entidades que estão obrigadas a ter e alcançar objetivos através do exercício das suas competências e conhecimentos**. Estas metas podem ter dependências intencionais entre si, que ocorrem quando existe uma relação entre os atores, assim nós acreditamos que é pertinente para o contexto que pretende ser representado. Modelos intencionais preliminares já foram desenvolvidos após esta atividade [44].



Outra importante contribuição que surgiu a partir desta experiência foi a possibilidade de identificar situações e controles próprios do jogo suscetíveis de sistematização. Assim delegando estas atividades ao sistema o estudante ou jogador não teriam que lidar com elas e poderia concentrar-se nas atividades do jogo que são relacionadas com os conceitos de engenharia de software. Portanto, delegar ao sistema as tarefas de validação e controle poderia potencializar a aprendizagem em engenharia de software dos estudantes durante o jogo.

#### 3.1.3.4. Considerações Finais da Atividade

Nosso experimento demonstrou que técnicas de elicitación são importantes não apenas para começar um novo software. Elas também são úteis sempre que há necessidade de captar novos conhecimentos e no nosso caso, esse conhecimento foi importante para a melhora e a evolução do SimulES.

A elicitación de requisitos é a primeira atividade que deve ser considerada ao abordar o desenvolvimento de software. Apesar de ser a primeira atividade não significa que isto aconteça uma vez, ele deve ser um processo iterativo e presente em todas as etapas do software, por isso acreditamos que em todas as etapas da evolução de SimulES se faz pertinente novos levantamentos para melhorar e aperfeiçoar o produto final.

Para elicitar e modelar jogos é necessário levar em consideração a interação entre atores e os elementos de seu ambiente. Modelar jogos usando uma abordagem intencional é inovador, isso foi identificado com a literatura revisada, pois nenhum dos jogos aqui descritos utiliza este tipo de abordagem.

Lembremos que o foco desta abordagem se baseia na análise das dependências estratégicas entre os atores de um sistema. Portanto, seria então possível representar as responsabilidades dos atores dentro do jogo.

Do mesmo modo, o jogo possui as seguintes características que pretendemos sejam representadas: (i) **regras** já que o jogo tem organização e coordenação que o inserem num quadro de natureza lógica, (ii) **o jogo envolve operações entre pessoas** e contato social, o tratamento de problemas e a utilização de conceitos que o jogo oferece dão ao participante oportunidade de empregar procedimentos cooperativos para alcançar o objetivo que é ganhar o jogo. (iii) **o jogo assume um significado funcional** onde a realidade é

incorporada e transformada no ambiente onde este é inserido. (iv) o jogo tem natureza estrutural além das intenções e objetivos por trás dos jogadores. (v) os jogadores possuem estratégias que são incorporadas ao jogo (vi) jogadores possuem motivações, intenções e razões para executar as atividades. A utilizarmos a modelagem intencional para evidenciar estes elementos que acreditamos de vital importância, estaremos mais próximos de uma boa modelagem, importante para o entendimento do contexto e futuras evoluções.

Se a tendência é que os jogos sejam considerados como um ambiente de aprendizagem em determinadas áreas do conhecimento (exemplo, engenharia de software) onde esta tendência crescente é abastecida por recursos tecnológicos cada vez mais sofisticados, queremos responder a esta tendência visando melhoras no ambiente do jogo SimulES. Para isso esperamos evoluí-lo com base nos modelos intencionais gerados, além de explorar mecanismos que tornem mais próximas modelagem e implementação.

### **3.1.3.5. Evolução do SimulES a Software**

Na revisão da literatura identificamos que a modelagem de jogos educacionais não considera a interação entre jogadores. Além disso, identificamos que para o desenvolvimento de jogos educacionais na engenharia de software não é prioritário usar modelos, em vários casos modelos foram omitidos ou tratados superficialmente.

Outro motivo para o uso da modelagem intencional nesta evolução do SimulES é a utilização destes modelos como insumo para a criação do protótipo como uma forma de passar de forma mais transparente de modelos à implementação. Essa modelagem evoluirá os léxicos e cenários previamente definidos em [44]

Na criação do protótipo de SimulES-W estamos incorporando elementos de transparência de software aqueles que permitiram descrição do que o software está fazendo, levando em consideração o nível de compreensão do usuário.

Dentro desta evolução do SimulES exploramos aspectos pedagógicos como incorporação de hipertexto como imagens e animações, interação entre nodos, reprodução de situações reais em cenários por médio de interfaces, internet como um mecanismo de comunicação, software em rede que permite trabalho

colaborativo e *e-learning*. Acreditamos que o uso das tecnologias e a rede podem melhorar as estratégias de aprendizado.

Dentro dos aspectos de melhora reportados pelos usuários temos a complexidade de regras e manejo de variáveis dentro do jogo. Como resposta a estas observações feitas achamos que mecanismos de controle dentro do jogo auxiliariam para que certas regras sejam validadas pelo sistema.

A versão do SimulES-W apresentada neste trabalho é projetada baseada em modelos, métodos, técnicas e padrões de arquitetura e projeto para dar suporte ao desenvolvimento de software com qualidade, com ênfase em sua natureza **evolutiva** [46]. Além disso, projetamos o SimulES-W para que atenda diferentes tipos de usuários e para isso as cartas conceito e problema podem ser customizadas como será apresentado no Capítulo 6.