

ERIC KIELING e RICARDO ROSA

***Planager* - um jogo para apoio ao ensino de conceitos de gerência de projetos  
de software**

**TRABALHO DE CONCLUSÃO II**

Referente à disciplina de Trabalho  
de Conclusão II do curso de Bacharelado em  
Ciências da Computação da Pontifícia  
Universidade Católica do Rio Grande do Sul.

Orientador: MSc. Rafael Prikladnicki

Porto Alegre, novembro de 2006.

## **RESUMO**

Este trabalho busca suprir as necessidades encontradas no ensino de gerenciamento de projetos de software, principalmente para alunos de graduação. O objetivo foi criar um jogo que ensina gerência de projetos de uma forma prática e interativa, para alunos que possuem pouca experiência sobre o assunto, aprendendo os conceitos a cada etapa do jogo. Neste documento são descritas as bases teóricas sobre o assunto, a descrição das ferramentas estudadas e o projeto implementado como prova de conceito do modelo proposto.

Palavras-chave: Gerência de projetos, Ensino, Jogos.

## SUMÁRIO

<b>RESUMO</b> .....	<b>1</b>
<b>Introdução</b> .....	<b>5</b>
1.1 Motivação e Justificativa .....	6
1.2 Apresentação .....	6
<b>2. Fundamentação Teórica</b> .....	<b>8</b>
2.1 Gerenciamento de escopo do projeto .....	10
2.1.1 Estrutura analítica do projeto (EAP) .....	11
2.2 Gerenciamento de tempo do projeto.....	12
2.2.1 Método do caminho crítico.....	13
<b>3. A Proposta de Ferramenta <i>Planager</i></b> .....	<b>16</b>
3.1 Objetivo Geral .....	17
3.2 Objetivos Específicos .....	17
3.3 Escopo do trabalho .....	18
<b>4. Descrição da ferramenta <i>Planager</i></b> .....	<b>19</b>
4.1 Descrição do jogo <i>Planager</i> .....	19
4.1.1 Descrição do cenário .....	19
4.1.2 Fase escopo.....	20
4.1.3 Fase EAP .....	21
4.1.4 Fase definição de atividades .....	23
4.1.5 Fase seqüenciamento de atividades .....	24
4.1.6 Fase caminho crítico .....	26
4.1.7 Tutorial do jogo .....	27
4.2 Especificação de requisitos do jogo.....	28
4.2.1 Casos de uso e diagrama de atividades.....	28
4.2.2 Diagrama de classes conceitual .....	62
<b>5. Recursos Necessários</b> .....	<b>64</b>
5.1 Recursos Necessários .....	64
<b>6. Conclusões</b> .....	<b>65</b>
<b>7. Referências Bibliográficas</b> .....	<b>68</b>

## Lista de Figuras

Figura 1 - Sobreposição dos grupos de processos em uma fase [PMI, 2004].	9
Figura 2 – Exemplo de estrutura analítica do projeto [PMI, 2004].	11
Figura 3 – Exemplo de cálculo das datas de término mais cedo.	14
Figura 4 – Exemplo de cálculo das datas de início mais tarde.	15
Figura 5 – Escopo do trabalho de conclusão.	18
Figura 6 – Descrição do cenário	20
Figura 7 - Fase escopo	21
Figura 8 - Fase EAP	22
Figura 9 - Fase definição de atividades	23
Figura 10 - Fase seqüenciamento de atividades	25
Figura 11 - Fase caminho crítico	26
Figura 12 – Tutorial do jogo	27
Figura 13 – Casos de uso do Usuário	28
Figura 14 – Casos de uso do Administrador	32
Figura 15 – Diagrama de atividades do caso de uso <i>Criar Cenário</i>	43
Figura 16 – Diagrama de atividades do caso de uso <i>Manter fase de Escopo</i>	44
Figura 17 – Diagrama de atividades do caso de uso <i>Manter fase de EAP</i>	45
Figura 18 – Diagrama de atividades do caso de uso <i>Manter fase de definição de atividades</i>	46
Figura 19 – Diagrama de atividades do caso de uso <i>Manter fase de seqüenciamento</i>	47
Figura 20 – Diagrama de atividades do caso de uso <i>Manter fase de caminho crítico</i>	48
Figura 21 – Casos de uso do Jogo	48
Figura 22 – Casos de uso do Tutorial	57
Figura 23 – Diagrama de classes conceitual	62

## **Introdução**

Desde muito tempo as pessoas se reúnem para realizar tarefas com o objetivo de criar novos produtos. Com o passar dos anos houve a necessidade de padronizar este esforço, assim surgindo o conceito de projeto.

Um projeto é definido como um esforço temporário, conduzido por um grupo de pessoas, para criar um produto, serviço ou resultado único. Temporário significa que todos os projetos possuem início e fim definidos, o fim é atingido quando os objetivos são alcançados, caso contrário o projeto fracassou. Um projeto é caracterizado por possuir objetivo e requisitos claramente definidos (Escopo), obedecer ao orçamento aprovado (Custo) e deve ser finalizado no prazo especificado (Tempo) [SCHWALBE, 2002].

Com o desenvolvimento tecnológico os projetos começaram a ficar cada vez mais complexos, surgindo a necessidade de criar técnicas e padrões para a execução desses. Com isso vários processos de gerenciamento foram criados, por exemplo, gerenciamento de custos, criação de uma agenda de tarefas, organização das equipes, alocação de recursos, análise de riscos e muito mais. Tudo isso precisa ser observado na criação, elaboração e execução de projetos independente da área. Para que um projeto seja bem executado é imprescindível que haja ferramentas apropriadas, técnicas bem desenvolvidas, conhecimento e habilidades de quem executá-lo, para isso existe o gerenciamento de projetos.

Hoje em dia, o gerenciamento de projetos de software está muito difundido entre diversos profissionais como diretores, gerentes, clientes e outras partes interessadas. Para auxiliar estes profissionais foi criado um conjunto de conhecimentos em gerenciamento de projetos que é a soma dos conhecimentos intrínsecos à profissão de gerenciamento de projetos [PMI, 2004].

Com base neste conjunto de conhecimentos, o gerenciamento de projetos constitui-se da aplicação e integração dos seguintes grupos de processos de gerenciamento: iniciação, planejamento, execução, monitoramento e controle, e encerramento [PMI, 2004].

O gerente de projetos deve seguir estes processos ao longo do desenvolvimento. Ele é responsável pelos objetivos do projeto e que esses sejam corretamente realizados. Para isso, o gerente deve identificar as necessidades, estabelecer os objetivos de forma clara e que esses sejam alcançáveis, balancear as demandas conflitantes de qualidade, escopo, tempo e custo e

adaptar as especificações, dos planos e da abordagem para que tanto a organização como os clientes fiquem satisfeitos.

Como pode-se notar, aprender a ser um gerente de projetos não é uma tarefa fácil. Para isso existem vários tipos de treinamentos, e esses, cada dia priorizam os profissionais que já possuem um bom embasamento teórico sobre gerenciamento de projetos, assim dificultando o aprendizado daquelas pessoas que estão começando a aprender sobre gerência, como alunos e profissionais sem experiência.

### **1.1 Motivação e Justificativa**

Existe uma demanda cada vez maior pelo aprendizado de gerenciamento de projetos. Foi observado que os treinamentos disponíveis, tanto nos cursos como nas faculdades, dão ênfase a conceitos não tão básicos de gerência de projetos, visando àqueles profissionais que pretendem ser gerentes ou aperfeiçoar seus conhecimentos sobre gerência.

A abordagem didático-pedagógica deve ser diferente, pois para aqueles alunos e profissionais que estão começando a aprender os conceitos sobre gerência de projetos deve ser feita uma abordagem de uma forma mais prática e interativa, para que a falta de maturidade e a inexperiência desses não comprometa o ensino.

Para facilitar este aprendizado, este trabalho utilizou uma metodologia diferenciada dos treinamentos tradicionais. Esta metodologia constitui em um jogo que é envolvente, onde o aluno se familiariza mais facilmente com os conceitos sobre gerenciamento de projetos. Assim, diferentemente dos treinamentos tradicionais, foi desenvolvida uma ferramenta que visa a didática e não que o aluno simplesmente decore os conceitos, mas sim que ele consiga aprender de uma forma mais prática e divertida.

Na ferramenta, chamada de *Planager*, foi implementado alguns processos que o PMBOK define (ver seção 2), onde o aluno pode revisar os conceitos de gerência de projetos aprendidos em aula e ter uma visão de como é o jogo através do módulo tutorial. No módulo jogo, o aluno pode praticar seus conhecimentos de uma forma interativa. O jogador tem que passar por várias fases, sendo avaliado no final de cada uma delas.

### **1.2 Apresentação**

Este texto tem por objetivo servir como relatório final da disciplina de Trabalho de Conclusão II do curso de Bacharelado em Ciência da Computação.

Este documento está dividido nas seguintes seções:

1. Introdução: Breve explicação sobre o conteúdo a ser explorado no trabalho e detalhamento do documento;
2. Fundamentação teórica: Explicação sobre os gerenciamentos de escopo de projeto e de tempo de projeto;
3. Proposta: Apresentação do jogo construído no Trabalho de Conclusão;
4. Descrição da ferramenta *Planager*: descrição da aplicação desenvolvida ao longo do Trabalho de Conclusão;
5. Recursos necessários: recursos necessários para a implementação da ferramenta;
6. Conclusões: descrição das limitações e trabalhos futuros;
7. Referências bibliográficas.

## 2. Fundamentação Teórica

O PMBOK é um guia baseado nas boas práticas de gerenciamento de projetos e está estruturado em nove áreas de conhecimento em gerenciamento de projetos, são elas [PMI, 2004]:

1. Integração: contém os processos necessários para assegurar a coordenação entre os demais processos;
2. Escopo: contém os processos necessários para definir e controlar o que está incluído ou não no projeto;
3. Tempo: contém os processos necessários para concluir o projeto dentro do cronograma;
4. Custo: contém os processos necessários para assegurar que o projeto se encerrará dentro do orçamento;
5. Qualidade: contém os processos necessários para assegurar que o projeto satisfará as necessidades contratadas;
6. Recursos humanos: contém os processos necessários para gerenciar os recursos humanos envolvidos do projeto;
7. Comunicações: contém os processos necessários para assegurar a geração, disseminação e armazenamento das informações do projeto;
8. Risco: contém os processos necessários para identificar, analisar e responder aos riscos do projeto;
9. Aquisições: contém os processos necessários para aquisição de bens ou serviços fora da organização.

O gerenciamento de projetos é um empreendimento integrador. A integração do gerenciamento de projetos exige que cada processo do projeto esteja relacionado com outros processos. O PMBOK define 44 processos de gerenciamento de projetos, estes processos estão divididos em cinco grupos [PMI, 2004]:

1. Grupo de processos de iniciação: são utilizados para reconhecer e iniciar um novo projeto ou fase;
2. Grupo de processos de planejamento: define objetivos e planeja ações para alcançar estes objetivos;

3. Grupo de processo de execução: coordena pessoas e recursos para realizar o projeto e produzir produtos e serviços;
4. Grupo de processos de monitoramento e controle: monitora a execução do projeto para garantir que os objetivos sejam alcançados;
5. Grupo de processos de encerramento: formaliza a aceitação do projeto e o encerra de forma ordenada.

Os grupos de processos raramente são eventos únicos e distintos. Eles normalmente são compostos por atividades sobrepostas e ocorrem em níveis de intensidade variável durante cada fase do projeto (Figura 1). Geralmente os grupos de execução e planejamento possuem uma duração maior do que os outros grupos, porém isso pode variar de acordo com o projeto.

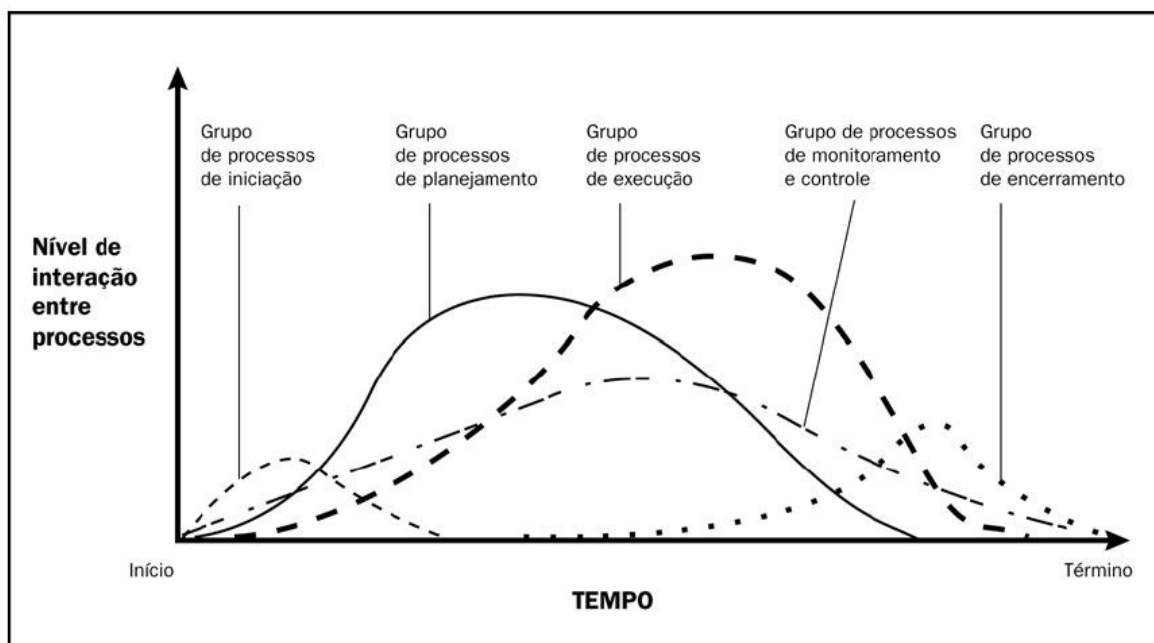


Figura 1 - Sobreposição dos grupos de processos em uma fase [PMI, 2004].

Este trabalho de conclusão utilizou a intersecção entre o grupo de processos de planejamento e as duas áreas de conhecimento de gerenciamento de projetos, a de escopo e a de tempo, para escolher os cinco processos que foram implementados (ver seção 3), onde cada processo representa uma fase do jogo. Estas áreas e seus processos, dentre eles os que foram selecionados, serão descritos nas duas próximas seções.

## 2.1 Gerenciamento de escopo do projeto

O gerenciamento do escopo do projeto inclui os processos necessários para garantir que o projeto possua todo o trabalho necessário, e somente ele, para terminar o projeto com sucesso [PMI, 2004]. Um dos objetivos dos processos de gerenciamento do escopo é que a equipe do projeto e os *stakeholders* possuam o mesmo entendimento de quais produtos serão gerados como resultado do projeto e quais processos serão utilizados na produção deles.

O PMBOK define cinco processos utilizados no gerenciamento do escopo do projeto [PMI, 2004]:

1. Planejamento do escopo: este processo é utilizado para determinar como a equipe irá definir o escopo do projeto, criar a estrutura analítica do projeto (EAP), verificar o escopo e controlar as mudanças de escopo.
2. Definição do escopo: neste processo o escopo do projeto será detalhado de forma que os *stakeholders* entrem em acordo com todo o trabalho que deverá ser realizado para alcançar os objetivos do projeto e quais são estes objetivos.
3. Criar a EAP: este processo é utilizado para criar a estrutura analítica do projeto. A EAP é decomposição hierárquica do trabalho a ser realizado no projeto. Ela deve conter todo o trabalho necessário para realizar o projeto. Alguns especialistas acreditam que o trabalho não deve ser realizado se ele não está incluído na EAP [SCHWALBE, 2002].
4. Verificação do escopo: neste processo ocorre a aceitação formal das entregas pelas partes interessadas do projeto. O processo de verificação do escopo documenta quais entregas foram aceitas e quais não foram aceitas juntamente com as justificativas para a não aceitação.
5. Controle do escopo: este processo é utilizado para controlar os fatores que geram mudanças no escopo do projeto e o impacto destas mudanças. O processo de controle do escopo define procedimentos para analisar as solicitações de mudanças e é responsável pela atualização da declaração do escopo e EAP.

Cada projeto exige um balanceamento cuidadoso de ferramentas, fontes de dados, metodologias, processos e procedimentos, para garantir que o esforço gasto nas atividades de determinação do escopo esteja de acordo com o tamanho, complexidade e importância do projeto [SCHWALBE, 2002].

### 2.1.1 Estrutura analítica do projeto (EAP)

A criação da EAP serve como base para a maior parte do planejamento de projeto. O importante é que a estrutura da EAP deverá conter todo e somente o trabalho necessário para o alcance dos objetivos do projeto, o trabalho que não está representado nesta estrutura estará fora do escopo do projeto.

É importante também utilizar a EAP para dividir o trabalho do projeto em partes menores e mais facilmente gerenciáveis, em que cada nível descendente da EAP representa uma definição cada vez mais detalhada do trabalho [PMI, 2004].

A quebra do trabalho em elementos menores que podem ser mais bem visualizados e entendidos facilita os processos de planejamento, execução, monitoramento e controle.

Um exemplo de EAP pode ser visualizado a seguir (Figura 2):

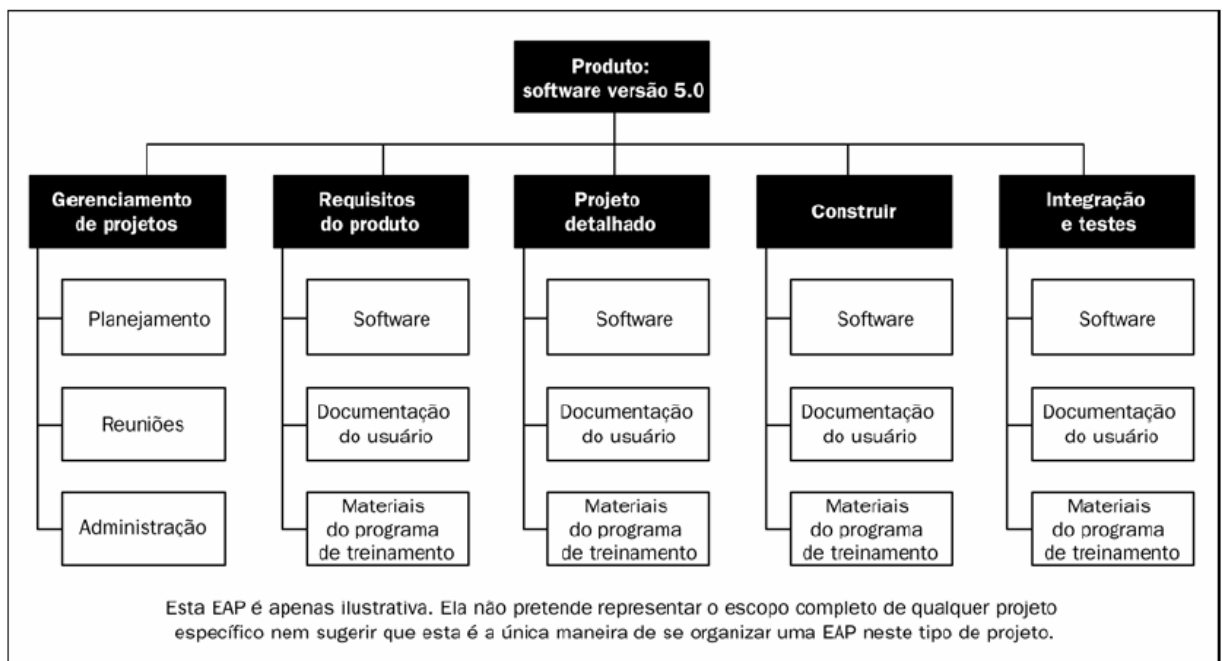


Figura 2 – Exemplo de estrutura analítica do projeto [PMI, 2004].

Vantagens que esta estrutura proporciona:

1. Proporciona uma visão geral de todo o escopo do projeto;
2. Fornece uma base para a maior parte do planejamento do projeto;
3. Facilita a definição clara de papéis e atribuição de responsabilidades;
4. Facilita a comunicação entre a equipe do projeto e os todos os envolvidos;
5. Define uma linha base para medir e controlar o desempenho do projeto;
6. Pode ser reutilizada em outros projetos.

A EAP deve ser decomposta até um nível que permita a efetiva gerência do projeto, sendo as folhas da árvore consideradas como os pacotes de trabalho.

Cada pacote de trabalho deve possuir:

1. Um código de identificação único;
2. Uma entrega específica e verificável;
3. Um único responsável pela sua entrega;
4. Uma equipe alocada;
5. Um custo definido.

## **2.2 Gerenciamento de tempo do projeto**

O gerenciamento de tempo do projeto inclui os processos necessários para concluir o projeto no prazo. Muitos gerentes afirmam que concluir o projeto dentro do cronograma é um dos maiores obstáculos da gerência de projeto. Segundo o CHAOS Report [STANDISH GROUP, 1995], um estudo utilizando como base 8380 projetos de TI, mostrou que apenas 16,2% dos projetos foram entregues respeitando os prazos e os custos e com todas as funcionalidades especificadas e que em média eles foram entregues utilizando 222% do tempo estipulado.

Os processos envolvidos na gerência do tempo do projeto são [PMI, 2004]:

1. Definição da atividade: este processo consiste na identificação das entregas (pacotes de trabalho da EAP) e na sua divisão em componentes menores e mais gerenciáveis (atividades). A principal saída deste processo é a lista de atividades. Ela deverá conter todo o trabalho necessário para a realização do projeto e deve possuir um nível de detalhamento suficiente para que a equipe do projeto possa entender o trabalho que deverá ser realizado.
2. Seqüenciamento de atividades: este processo utiliza a lista de atividades que foi gerada pelo processo de definição de atividades juntamente com algumas premissas e restrições para determinar as dependências entre as atividades. Existem três tipos de dependências: dependências obrigatórias (inerentes à natureza do trabalho), dependências arbitradas (utilizar para especificar uma seqüência preferida) e dependências externas (relacionamento com atividades que não são do projeto).
3. Estimativa de recursos da atividade: este processo consiste na identificação de quais recursos e em que quantidade estes recursos serão utilizados durante cada atividade. Recursos podem ser pessoas, equipamentos ou outros tipos de materiais. Após a

estimativa de recursos nas atividades é possível agregar os recursos para determinar a estimativa de recursos para os pacotes de trabalho.

4. Estimativa de duração da atividade: este processo utiliza o escopo da atividade, tipos de recursos necessários, estimativas das quantidades de recursos e a disponibilidade dos recursos para determinar a duração de cada atividade. Como as durações das atividades são geralmente difíceis de estimar, utilizam-se opiniões de especialistas nas áreas das atividades e analogias com atividades semelhantes.
5. Desenvolvimento do cronograma: este processo utiliza todos os resultados dos outros processos de gerenciamento de tempo para determinar a data de início e término de cada atividade do projeto. Diversas técnicas são utilizadas para auxiliar no desenvolvimento do cronograma, as principais são: gráfico de Gantt, análise de PERT e caminho crítico.
6. Controle do cronograma: este processo é utilizado para monitorar o andamento do cronograma durante o projeto. O controle do cronograma deve medir e avaliar o desempenho dos prazos e definir os procedimentos para efetuar mudanças no cronograma.

### **2.2.1 Método do caminho crítico**

O método do caminho crítico é uma técnica utilizada para calcular as datas teóricas de início e término de todas as atividades do projeto com o objetivo de descobrir quais são as folgas entre as atividades (a folga de uma atividade é a diferença entre a data que ela deve ser iniciada e a data que ela pode ser iniciada).

O caminho crítico é a seqüência de atividades que não possuem folga, ou seja, é o caminho com maior duração. Este caminho determina o menor tempo possível para concluir o projeto. As atividades que compõe o caminho crítico são chamadas de atividades críticas, pois se qualquer uma destas atividades for atrasada, o projeto inteiro será atrasado.

A importância do caminho crítico se dá pelo fato que só é possível diminuir a duração do projeto modificando as atividades críticas. Se o gerente do projeto aumentar os recursos utilizados em atividades não críticas, ele não estará modificando a data de término do projeto.

É necessário atualizar o caminho crítico conforme o andamento do projeto, pois atividades fora do caminho crítico podem ter sido atrasadas (gerando um caminho mais longo), ou

atividades do caminho crítico podem ter sido realizadas em menos tempo gerando folga em algumas atividades que eram atividades críticas.

Este método utiliza o resultado dos processos de definição de atividades, sequenciamento de atividades e estimativa de duração das atividades e não leva em consideração a estimativa de recursos necessários para cada atividade e por isso pode produzir um cronograma preliminar de início mais cedo e um cronograma preliminar de início mais tarde que podem exigir mais recursos durante determinados períodos de tempo do que os disponíveis.

Para determinar o caminho crítico é necessário descobrir as folgas de todas as atividades e para isso é necessário calcular a data de início mais cedo (IMC), data de início mais tarde (IMT), data de término mais cedo (TMC) e data de término mais tarde (TMT) de todas as atividades.

Dado que a data de início mais cedo da primeira atividade sempre é zero e data de término mais tarde da última atividade sempre é igual à data de término mais cedo da mesma, pode-se calcular as datas de término mais cedo (TMC) através da seguinte fórmula:  $TMC = IMC + \text{duração da atividade}$  (Figura 3).

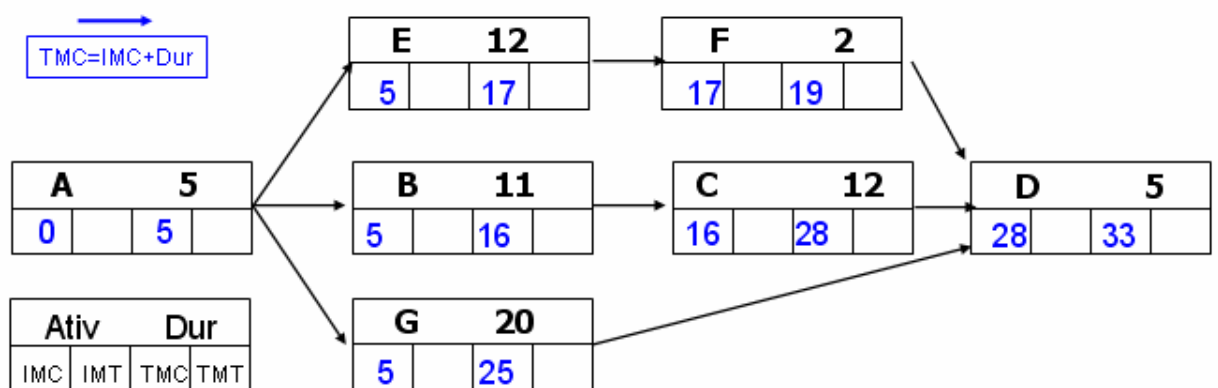


Figura 3 – Exemplo de cálculo das datas de término mais cedo.

Após calcular as datas de término mais cedo é possível calcular as datas de início mais tarde (IMT) utilizando a seguinte fórmula:  $IMT = TMT - \text{duração da atividade}$  (Figura 4).

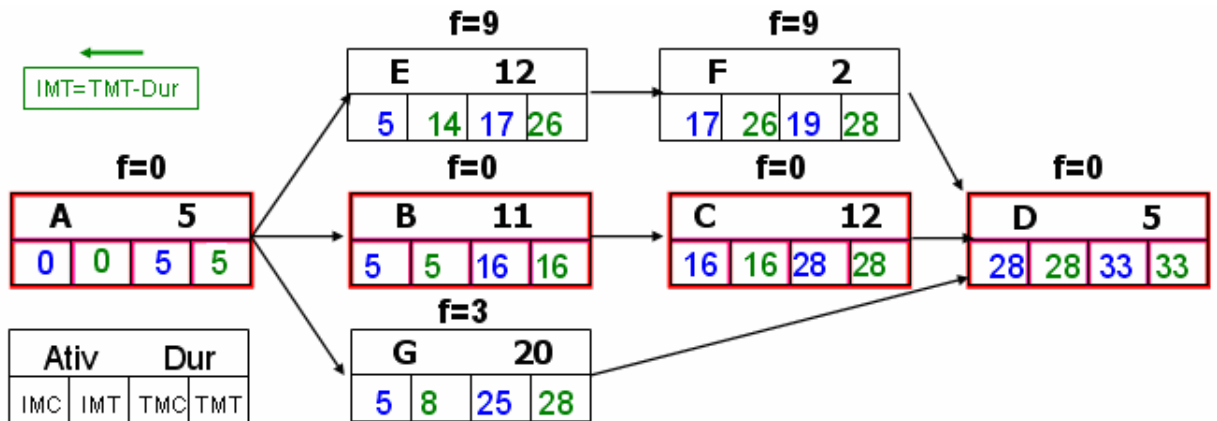


Figura 4 – Exemplo de cálculo das datas de início mais tarde.

O próximo passo para encontrar o caminho crítico é o cálculo das folgas. Para encontrar a folga individual de cada atividade pode-se subtrair a data de término mais tarde (TMT) com a data de término mais cedo (TMC). No exemplo da Figura 4 o caminho crítico é composto pelas atividades A, B, C, D, pois elas possuem folga zero enquanto as atividades G, E, F possuem folga três, nove e nove respectivamente.

### 3. A Proposta de Ferramenta *Planager*

Este trabalho desenvolveu um jogo (*Planager*) para apoio ao ensino de conceitos de gerência de projetos de software. Este jogo busca suprir as necessidades encontradas no ensino de gerenciamento de projetos de software, principalmente para alunos de graduação.

Muitas vezes os alunos de graduação não possuem um grau de maturidade e nem experiência suficiente para entender alguns conceitos apresentados na literatura. A grande maioria dos livros e softwares sobre gerenciamento de projetos possui como objetivo a formação de gerentes de projeto ou a certificação na área de gerência de projetos (PMP).

O jogo proposto neste trabalho de conclusão tem como objetivo ajudar o ensino da gerência de projetos de uma maneira mais didática do que as encontradas na literatura e nos cursos existentes, focando principalmente os alunos de graduação, mas possibilitando o uso por qualquer interessado iniciante na área.

Jogos são utilizados para ajudar no ensino de diversas áreas do conhecimento e muitas vezes despertam maior interesse e motivação por parte do aluno [GRAMIGMA, 1994]. Segundo Betz [BETZ, 1995], existe uma grande associação entre jogos e aprendizado. Os jogos de computador permitem a visualização e experimentação de conceitos abstratos e possuem ambientes que despertam a criatividade dos jogadores. No jogo *Planager* tentamos trabalhar com os conceitos de gerenciamento de projetos, tais como EAP de forma visual e intuitiva. O jogador pode, por exemplo, criar diagramas de rede e associar atividades a pacotes de trabalho utilizando *drag and drop* de objetos gráficos.

Neal [NEAL, 1990] afirma que competição e definição de objetivos são componentes motivadores em jogos de computador. O jogo *Planager* possui um sistema de pontuação e um ranking que serão utilizados para motivar o jogador a aprender.

O jogo proposto possui algumas semelhanças com os jogos de empresa utilizados na área de administração. Jogos de empresa são simulações de um ambiente empresarial, onde os participantes atuam como executivos de uma empresa avaliando e analisando cenários hipotéticos e suas possíveis conseqüências [GRAMIGMA, 1994].

O jogo proposto não possui como objetivo simular todos os processos utilizados na gerência de projetos, pois ele irá focar apenas o grupo de processos de planejamento. Ele irá auxiliar a

execução dos processos e a criação dos documentos para o planejamento de projetos hipotéticos, mas não irá simular a execução dos mesmos.

Devido à limitação de tempo, foi escolhido o grupo de processos de planejamento e foram escolhidas duas entre as nove áreas de conhecimento em gerenciamento de projetos definidas pelo PMBOK para serem desenvolvidas no jogo. As duas áreas escolhidas são: gerenciamento do escopo e gerenciamento do tempo, pois elas possuem processos, como por exemplo, os processos de criação da EAP e cronograma que servem de base para uma grande quantidade de processos e que poderão ser utilizados em futuros módulos do jogo. Estas duas áreas foram desenvolvidas como prova de conceito e o software foi desenvolvido modularmente para facilitar a criação de novas fases que ajudem a praticar os outros processos de outras áreas de conhecimento do gerenciamento de projetos.

Pela intersecção do grupo e das duas áreas escolhidos foram selecionados cinco processos que formam as fases do jogo. Na área de gerenciamento de escopo foram escolhidos os processos: definição do escopo e criação da EAP, já na área de gerenciamento de tempo foram escolhidos os processos: definição da atividade, seqüenciamento de atividades e desenvolvimento do cronograma. No capítulo 4 será explicado o que foi implementado de cada processo.

O jogo *Planager* foi desenvolvido utilizando a linguagem de programação Java. Sendo assim, uma ferramenta flexível e portátil, podendo ser utilizada pelas diversas plataformas que suportam a JVM (*Java Virtual Machine*). O jogo não foi desenvolvido utilizando-se a arquitetura cliente/servidor, pois optamos por manter a aplicação independente de recursos tais como a Internet. Esta abordagem possibilita que o jogo seja utilizado em diversos locais e que os alunos possam levar cópias dele para casa para aperfeiçoar os seus conhecimentos.

### **3.1 Objetivo Geral**

O objetivo geral foi criar um jogo que ensina gerência de projetos de uma forma prática e interativa, para alunos que possuem pouca experiência sobre o assunto, aprendendo os conceitos a cada etapa do jogo.

### **3.2 Objetivos Específicos**

Para alcançar o objetivo geral foram identificados os seguintes objetivos específicos:

1. Estudo sobre gerenciamento de tempo de projeto;
2. Estudo sobre gerenciamento de escopo de projeto;

3. Modelagem do sistema;
4. Estudo sobre a utilização de jogos na área de Informática na Educação.

### 3.3 Escopo do trabalho

O escopo do trabalho de conclusão foi definido através de uma estrutura analítica do projeto (Figura 5). Esta EAP possui três itens principais, são eles:

- Gerência de projeto: todas as atividades referentes ao gerenciamento do trabalho de conclusão;
- TC1: possui as atividades referentes às pesquisas sobre interface jogos e gerência de projetos, assim como a modelagem inicial do jogo e o volume final de TC1;
- TC2: possui as atividades referentes ao desenvolvimento, teste e documentação do jogo, assim como o volume final e a apresentação para a banca examinadora.

A figura a seguir representa a estrutura analítica do projeto, considerando o desenvolvimento das disciplinas de Trabalho de Conclusão I e Trabalho de Conclusão II:

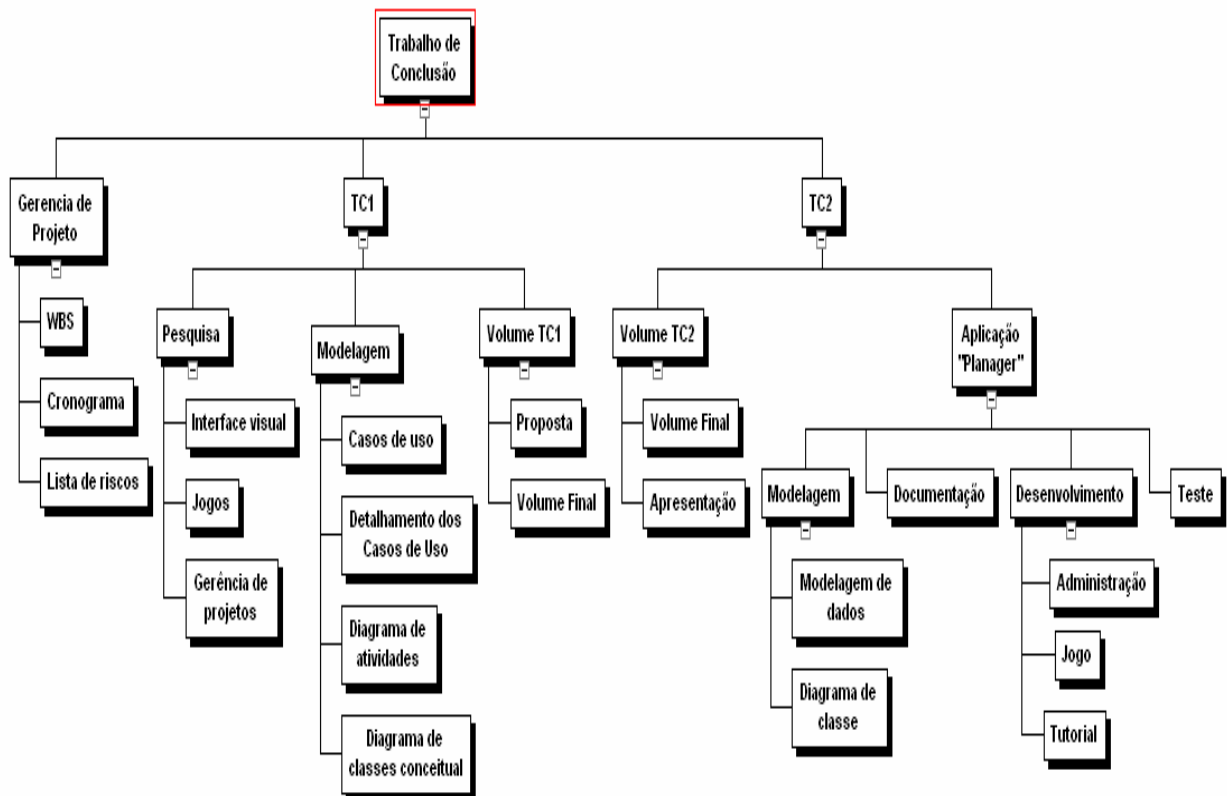


Figura 5 – Escopo do trabalho de conclusão.

#### **4. Descrição da ferramenta *Planager***

A ferramenta possui uma grande quantidade de funcionalidades e cada uma com suas peculiaridades. A seguir será feita uma explicação de cada uma delas, dando uma visão coesa sobre a ferramenta e como foi desenvolvida.

##### **4.1 Descrição do jogo *Planager***

O jogo *Planager* possui dois tipos de usuários, o administrador e o jogador. O jogador pode utilizar o modulo tutorial para aprender sobre gerência de projetos e pode jogar em diversos cenários cadastrados na base de dados do jogo. Somente o administrador é capaz de adicionar, modificar e remover cenários da base de dados. Cenários são representações de projetos e são compostos por uma descrição e por cinco fases: fase escopo, fase EAP, fase definição de atividades, fase seqüenciamento de atividades e fase caminho crítico.

Esta seqüência de fases é geralmente a mesma seqüência que os gerentes de projetos utilizam para planejar o escopo e o tempo de um projeto. O gerente de projetos utiliza as informações geradas pelos primeiros processos como entrada para vários outros processos (por exemplo, a definição de atividades gera informações que são utilizadas no seqüenciamento de atividades).

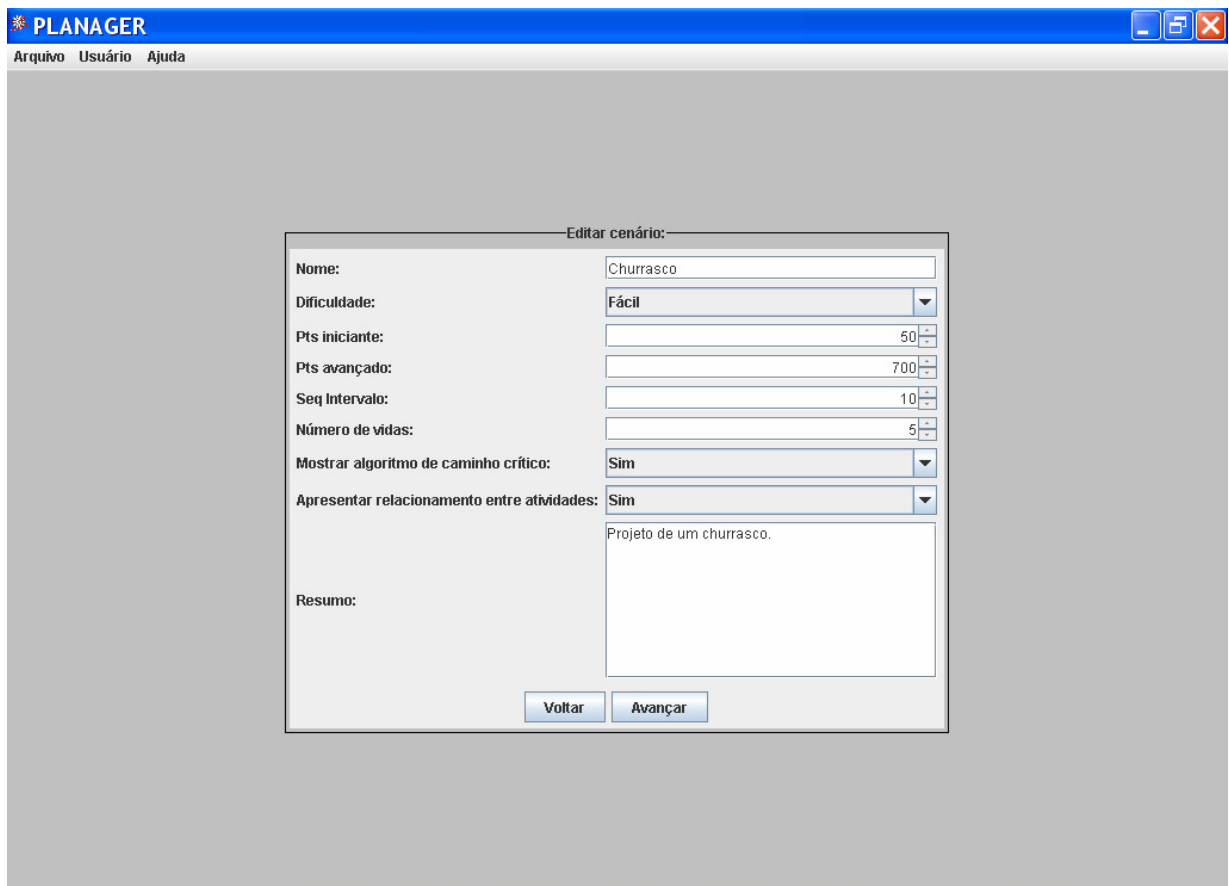
O jogador, assim como o gerente de projetos, deverá utilizar as informações contidas nas fases anteriores para conseguir resolver corretamente as próximas fases do jogo. Em futuras versões do jogo poderão ser acrescentadas fases que utilizarão informações geradas pelas fases atuais (por exemplo, pode-se incluir a fase de criação do cronograma).

O usuário deve estar logado para utilizar qualquer funcionalidade do jogo. Ele é capaz de modificar o seu cadastro básico que contém seu nome, senha, tipo e nível de experiência. Ele também é capaz de consultar o ranking que contém as melhores pontuações obtidas pelos jogadores em cada cenário.

##### **4.1.1 Descrição do cenário**

Na descrição do cenário o jogador tem o nome do cenário, a dificuldade, a pontuação mínima para passar de fase e um resumo. O jogador ao ler a descrição pode desistir de jogar o cenário e escolher outro ou iniciar o jogo.

O administrador do jogo além de cadastrar as informações que o jogador visualiza também cadastra informações auxiliares ao cenário, como os pontos mínimos para o jogador iniciante e para o avançado passarem de fase, o intervalo que o caminho crítico utiliza para sortear as durações, o número inicial de vidas que o usuário ganha quando começa a jogar o cenário, a condição se o cenário mostra o algoritmo do caminho crítico ao jogador e a condição se o cenário mostra o relacionamento entre as atividades. A Figura 6 mostra a tela de descrição, pela visão do administrador, do projeto “Churrasco”.



The screenshot shows the PLANAGER application window with a menu bar (Arquivo, Usuário, Ajuda) and standard window controls. A dialog box titled "Editar cenário:" is open, displaying the following fields:

Nome:	Churrasco
Dificuldade:	Fácil
Pts iniciante:	50
Pts avançado:	700
Seq Intervalo:	10
Número de vidas:	5
Mostrar algoritmo de caminho crítico:	Sim
Apresentar relacionamento entre atividades:	Sim
Resumo:	Projeto de um churrasco.

At the bottom of the dialog box are two buttons: "Voltar" and "Avançar".

Figura 6 – Descrição do cenário

#### 4.1.2 Fase escopo

A fase de escopo possui a descrição detalhada do escopo do projeto e nela o jogador deverá responder questões de múltipla escolha. As questões de múltipla escolha podem ser utilizadas para verificar se o jogador entendeu corretamente a descrição do projeto.

O administrador do jogo também utiliza esta fase para cadastrar questões que não estão relacionadas diretamente com a descrição do escopo do projeto. Estas perguntas podem ser teóricas, como por exemplo, questões sobre como calcular o caminho crítico ou como criar uma estrutura analítica de projeto.

A pontuação desta fase é calculada utilizando a quantidade de acertos e a quantidade total de questões. A Figura 7 mostra um jogador respondendo perguntas sobre o escopo do projeto “Churrasco”.

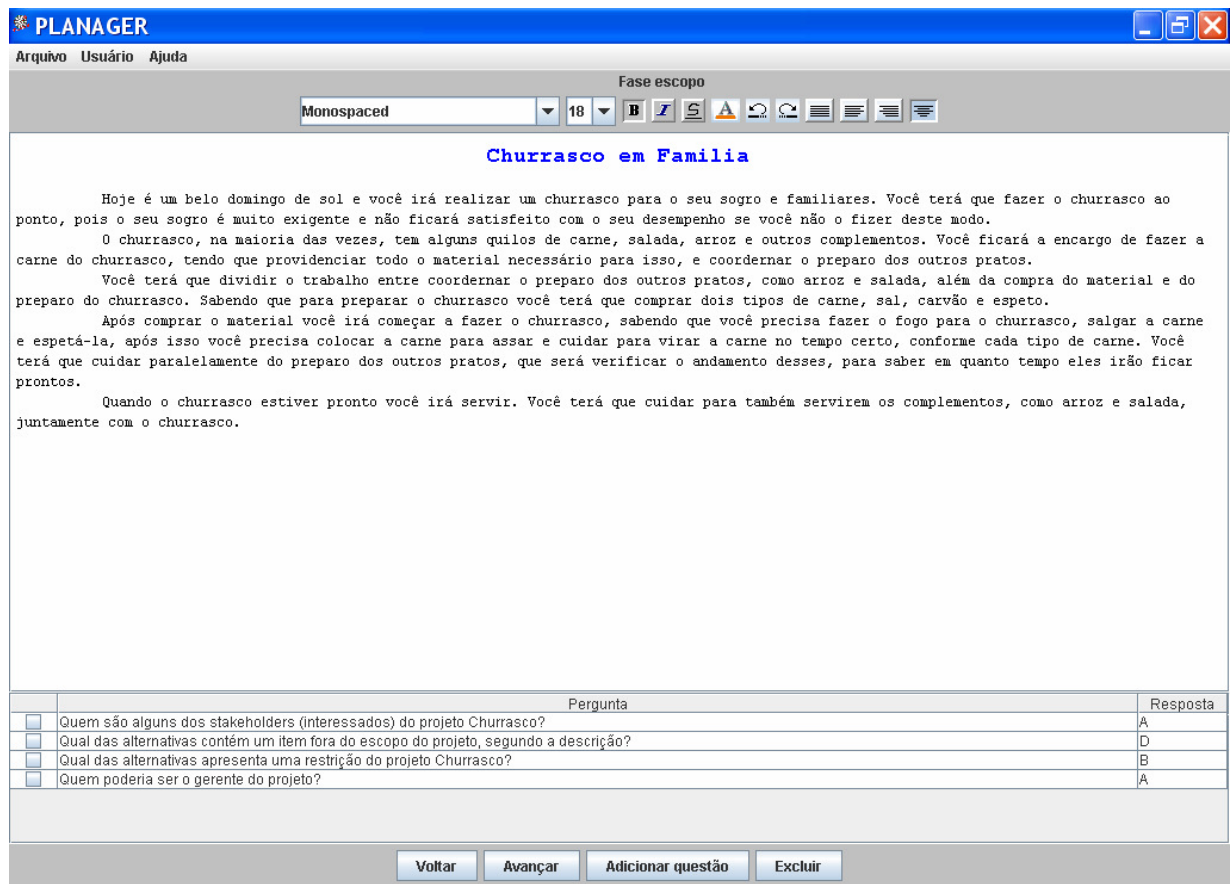


Figura 7 - Fase escopo

#### 4.1.3 Fase EAP

Nesta fase o jogador exercita o conceito de estrutura analítica do projeto (EAP) de forma prática. Ele recebe vários rótulos (alguns que fazem parte da EAP e outros que não fazem parte) e deve associá-los aos nodos da EAP. O administrador do jogo também utiliza esta fase para

cadastrar a EAP e os nodos complementares, que servem para dificultar a fase. A Figura 8 mostra um jogador associando os rótulos a EAP do cenário “Churrasco” nos seus respectivos nodos.

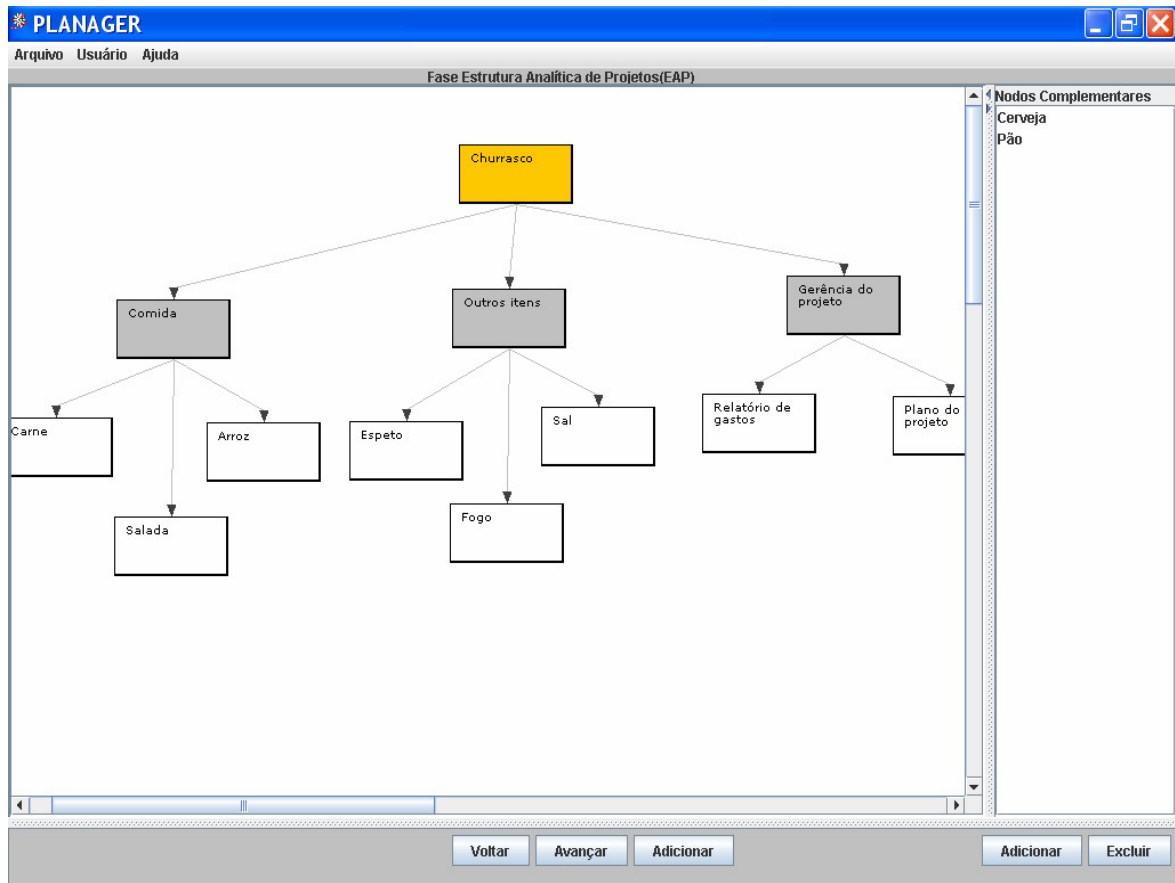


Figura 8 - Fase EAP

A fase possui como entrada a declaração do escopo do projeto, que foi definida na fase escopo. As outras entradas, como ativos de processos organizacionais, não foram incluídos, pois envolveria processos mais complexos, como a análise de uma organização, o que não se enquadra no objetivo do jogo.

Foi utilizado o método de decomposição da EAP, sendo prático para o jogador aprender a dividir o escopo em uma estrutura organizada. A decomposição é a subdivisão das entregas do projeto em componentes menores e mais facilmente gerenciáveis, até que o trabalho e as entregas estejam definidos até o nível de pacote de trabalho [PMI, 2004], o que serve como entrada para a fase definição de atividades.

Para facilitar a avaliação e o aprendizado o jogo disponibiliza a EAP já montada na tela sem os rótulos (eles estão em uma lista). O jogador tem que montar a EAP a partir dos rótulos fornecidos, sabendo que tem rótulos falsos para confundir.

Nesta fase o jogador ganha pontos se ele associar um rótulo ao seu lugar correto na EAP e se suas ligações também estiverem corretas. Caso o rótulo esteja no lugar errado ele não ganha pontos e se algumas das ligações estiverem erradas há uma proporção de quantos pontos ele ganhará.

#### 4.1.4 Fase definição de atividades

Nesta fase o jogador recebe uma lista contendo diversas atividades (algumas que fazem parte do escopo do projeto e outras que não fazem parte) e ele deve associar estas atividades aos pacotes de trabalho da EAP. O administrador do jogo também utiliza esta fase para associar as atividades aos pacotes de trabalho da EAP. A Figura 9 mostra um jogador associando as atividades do cenário “Churrasco” nos seus respectivos pacotes de trabalho.

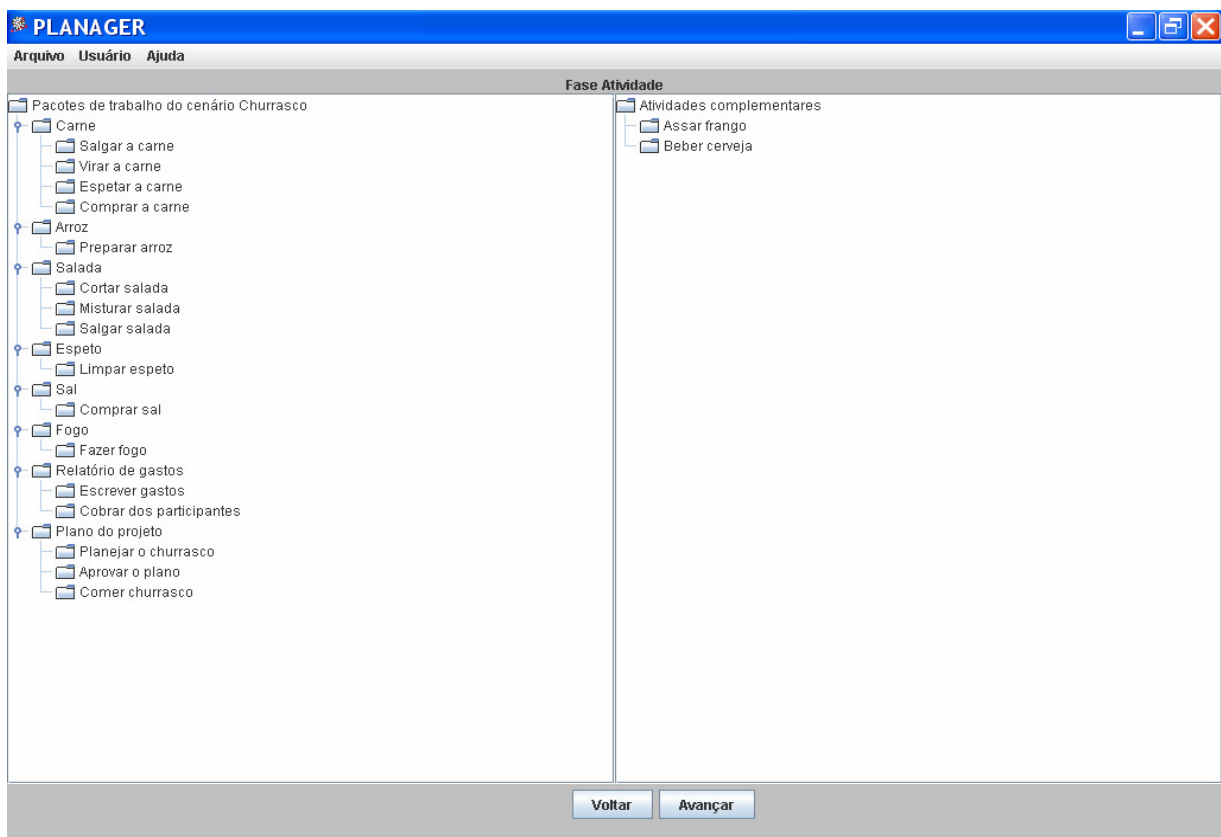


Figura 9 - Fase definição de atividades

O objetivo desta fase é ensinar o jogador a decompor os pacotes de trabalho da EAP em componentes menores, chamados de atividades. Em um projeto real, estas atividades seriam utilizadas para definir os custos e o cronograma do projeto. No jogo *Planager* elas serão utilizadas como entrada para o processo de cálculo do caminho crítico do projeto.

O jogo não possui uma fase para a estimativa da duração das atividades, pois este é um processo que normalmente é realizado por especialistas e é baseado no histórico dos projetos anteriores. Como o jogador não possui conhecimento nas áreas de todos os cenários ele não será capaz de realizar a estimativa de duração e por esse motivo a duração é escolhida aleatoriamente na fase de cálculo do caminho crítico. Não foi criada uma fase para o processo de estimativa de recursos de atividades pelo mesmo motivo.

Para ser possível avaliar este conceito tão subjetivo o jogo só permite ao jogador identificar se as atividades fazem parte do escopo do projeto e a qual pacote de trabalho elas estão vinculadas. Se ele fosse livre para criar qualquer atividade seria impossível verificar se elas fazem ou não parte do escopo.

Nesta fase o jogador perde pontos se ele associar uma atividade que não esta no escopo. O jogador ganha pontos se a atividade for associada ao pacote de trabalho correto e se a atividade pertencer ao escopo, mas caso não for associada ao pacote correto o jogador irá ganhar um quarto da pontuação normal.

#### **4.1.5 Fase seqüenciamento de atividades**

Nesta fase o jogador deve criar relações de dependência entre as atividades da fase anterior. A fase possui como entradas a declaração do escopo do projeto (fase escopo) e lista de atividades (fase definição de atividades). As outras entradas, como atributos da atividade, não foram incluídas, pois não se enquadram no objetivo do jogo. O administrador do jogo também utiliza esta fase para criar o seqüenciamento das atividades cadastradas. A Figura 10 mostra um jogador associando os rótulos ao diagrama de seqüenciamento, do cenário “Churrasco”, nos seus respectivos nodos.

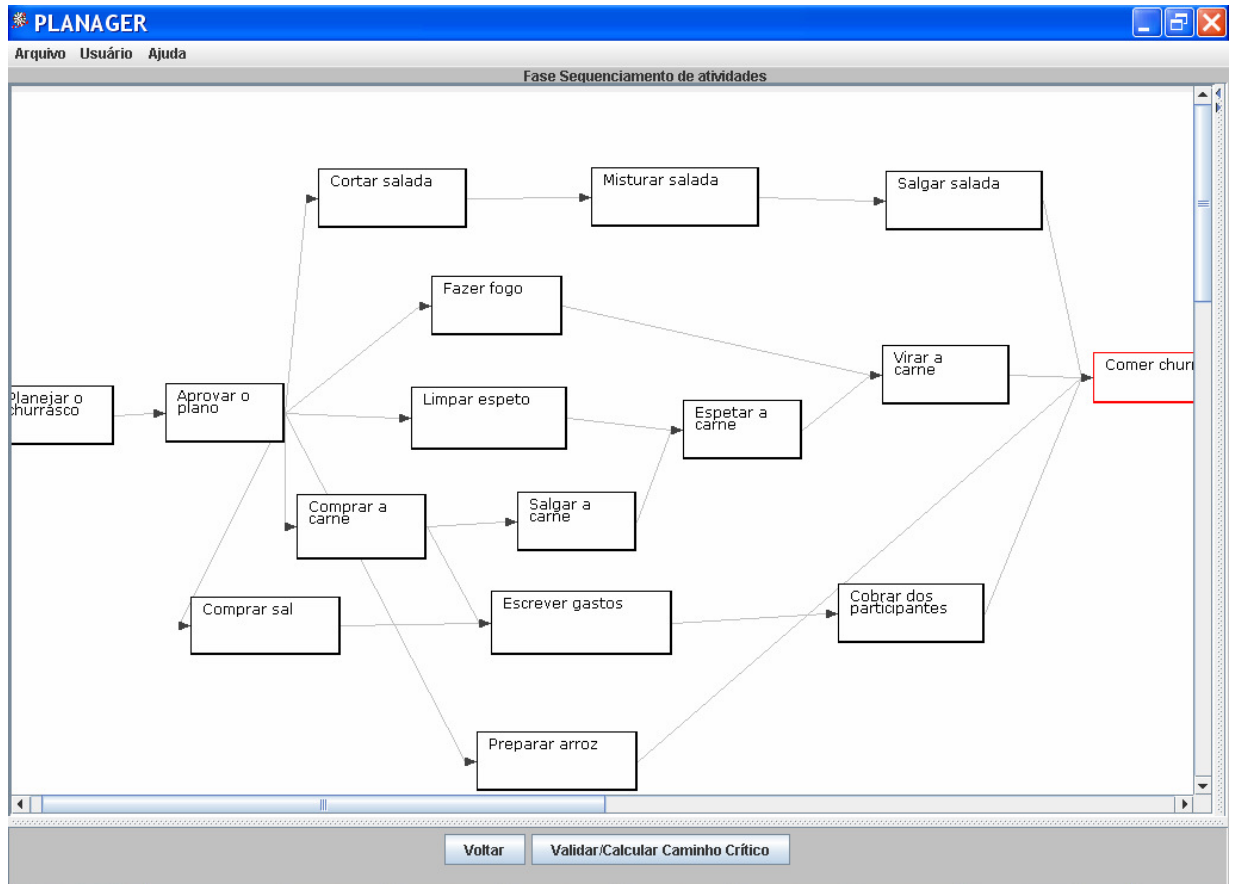


Figura 10 - Fase seqüenciamento de atividades

Foi utilizado o método do diagrama de precedência (MDP), pois é um método fácil de ser implementado e prático para o jogador aprender a seqüenciar as atividades do projeto. O MDP é um método de construção de um diagrama de rede do cronograma do projeto que usa caixas ou retângulos, chamados de nós, para representar atividades e os conecta por setas que mostram as dependências [PMI, 2004].

A aplicação do método foi restringida para facilitar a implementação e avaliação da fase. Foi usado somente o término para início como a relação de precedência, pois é a relação mais utilizada [PMI, 2004]. Outra restrição foi que o diagrama só poderá ter um nodo inicial e um nodo final, para simplificar a implementação no escopo do Trabalho de Conclusão. A partir deste cenário o jogador deve montar o diagrama com dos rótulos fornecidos em uma lista e dependendo do nível do jogador tem que ligar os nodos entre si.

Nesta fase o jogador ganha pontos se ele associar um rótulo ao seu lugar correto no diagrama de seqüenciamento e suas ligações também estiver corretas. Quando o rótulo está no lugar errado ele não ganha pontos. Se algumas das ligações estiverem erradas há uma proporção de quantos pontos ele ganhará e se ele colocar mais ligações que o nodo deveria ter ele perder pontos por cada ligação a mais.

#### 4.1.6 Fase caminho crítico

Nesta fase o jogador utiliza o diagrama gerado na fase de seqüenciamento para calcular as folgas de todas as atividades e com isso descobrir o caminho crítico do projeto. O jogador recebe a estrutura do diagrama do caminho crítico pronta na tela e precisa calcular para cada nodo do diagrama a data de inicio mais cedo (IMC), data de inicio mais tarde (IMT), data de término mais cedo (TMC), data de término mais tarde (TMT) e a folga. O administrador do jogo já recebe esta fase pronta, pois o sistema realiza todo o cálculo a partir do seqüenciamento sorteando durações para cada atividade. A Figura 11 mostra um jogador preenchendo os nodos do caminho crítico do projeto “Churrasco”.

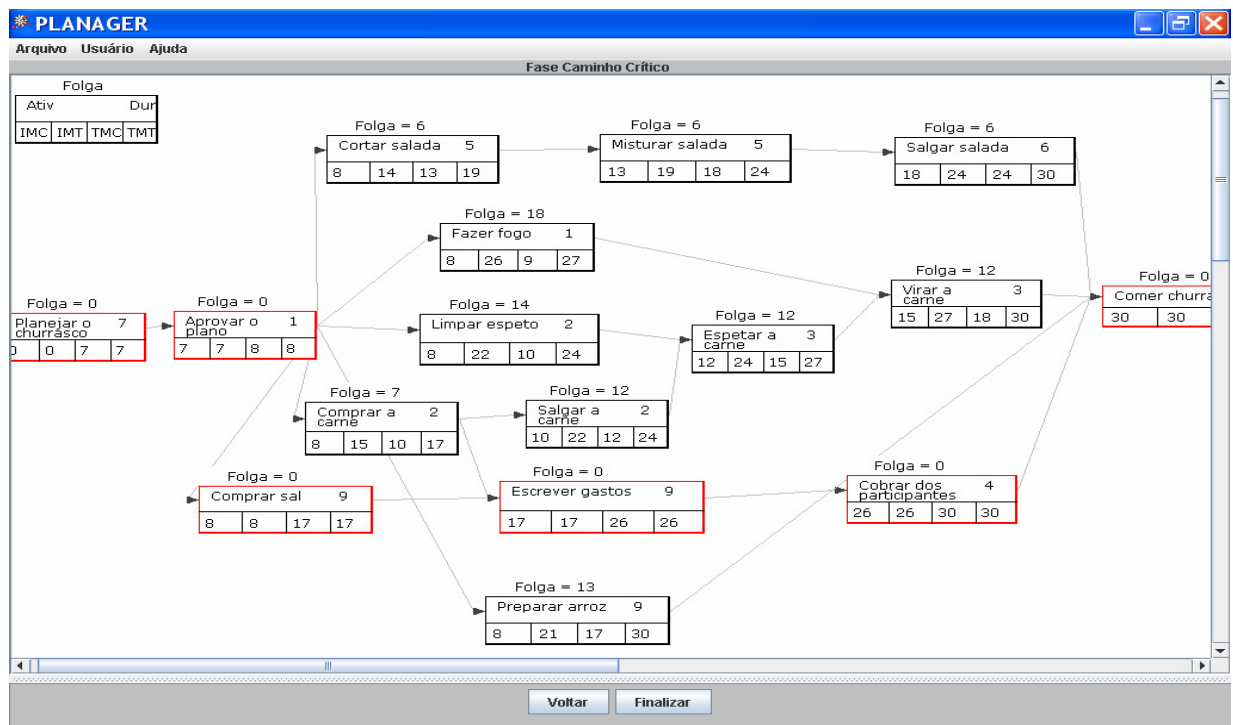


Figura 11 - Fase caminho crítico

Foi restringido o processo de cálculo do cronograma do projeto para apenas realizar o cálculo do caminho crítico. Considerou que pelo pouco tempo para a implementação, o caminho crítico daria uma boa noção para o jogador de quais atividades ele tem que cuidar para que o projeto não atrase, além de ser facilmente implementado por possuir regras bem definidas.

Nesta fase o jogador ganha pontos se ele calcular o IMC, o IMT, o TMC, o TMT e a folga corretamente. Quando o jogador errar um dos cálculos ele ganhar pontos proporcionalmente somente aos acertos.

#### 4.1.7 Tutorial do jogo

Um tutorial foi desenvolvido para mostrar aos jogadores um exemplo de como jogar um cenário completo. O tutorial foi criado na forma de um vídeo em *Flash* que mostra a execução do cenário “Churrasco” por um jogador. Este vídeo mostra quais são os elementos da interface que o jogador deve utilizar para completar todas as cinco fases do jogo. O tutorial também tem como objetivo relembrar alguns conceitos de gerência de projetos, tais como EAP e caminho crítico.

O cenário “Churrasco” foi escolhido para ser o cenário exemplo do tutorial porque ele possui uma baixa dificuldade e ele não exige conhecimento técnico de nenhuma área (engenharia, computação, etc.). O tutorial mostra para o jogador como que ele poderá obter vidas extras e mostra algumas dicas que podem ser utilizadas durante o jogo. A Figura 12 mostra o tutorial explicando para o jogador alguns dos elementos gráficos que ele pode utilizar durante a fase de criação da EAP.

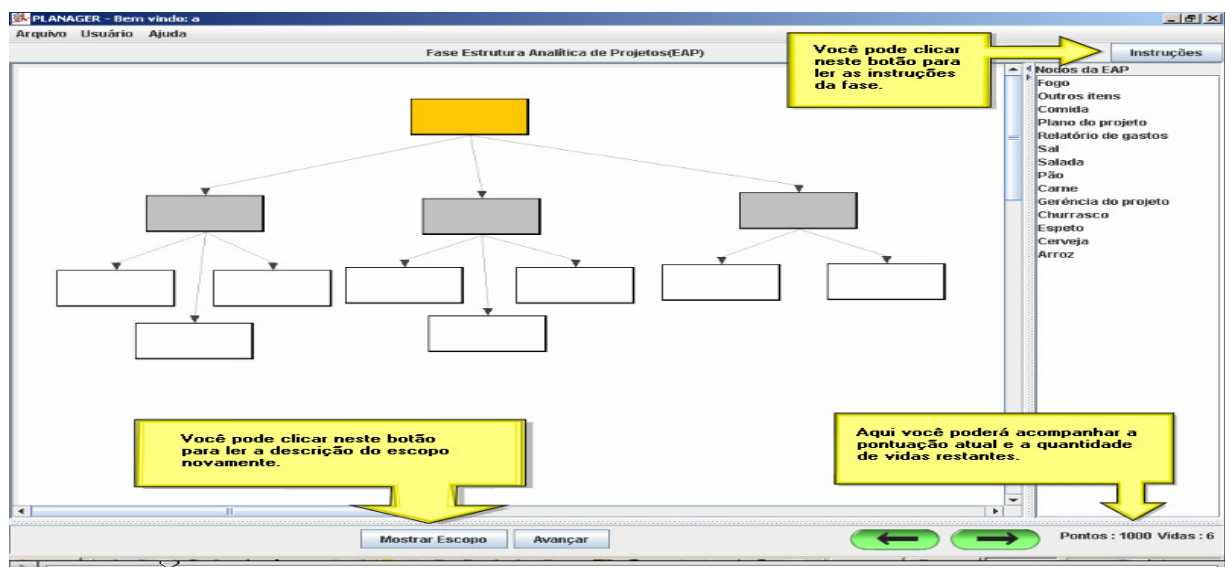


Figura 12 – Tutorial do jogo

## 4.2 Especificação de requisitos do jogo

Serão descritos os casos de uso do sistema com suas descrições e diagrama de atividades correspondentes. Também será apresentado o diagrama de classes conceitual do sistema, com sua descrição.

### 4.2.1 Casos de uso e diagrama de atividades

O diagrama a seguir (Figura 13), representa os casos de uso relacionados ao atores *Administrador* e *Jogador*.

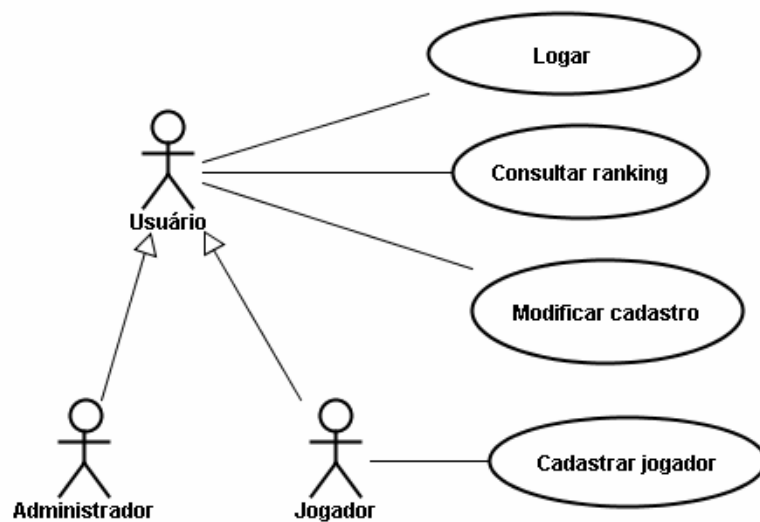


Figura 13 – Casos de uso do Usuário

Foram descritos os casos de uso *Logar*, *Consultar Ranking*, *Modificar Cadastro* e *Cadastrar Jogador*.

#### Caso de Uso 01 – Logar

ID do Caso de Uso:	01
Nome do Caso de Uso:	Logar
Criado por:	Eric Kieling
Data de Criação:	12/06/2006
Última Atualização Realizada por:	Ricardo Rosa
Data da Última Atualização:	13/11/2006
Atores:	Jogador ou Administrador
Propósito:	O usuário deve ser capaz de se logar no sistema.
Pré-condições:	

Pós-condições:	1. Usuário logado no sistema.
----------------	-------------------------------

<b>Fluxo Básico de Eventos</b>	
<b>Ações do Ator:</b>	<b>Ações do Sistema:</b>
	1. O sistema apresenta a tela de login contendo o campo “Usuário” e o campo “Senha”.
2. O usuário informa o seu nome e a sua senha.	
	2. O sistema valida se o nome e senha do usuário estão corretos. Caso as informações estejam inconsistentes ir para <b>Subseção 2.1</b> .
	3. O sistema efetua o logon do usuário.
<b>Subseção 2.1</b>	
<b>Ações do Ator:</b>	<b>Ações do Sistema:</b>
	2.1.1 O Sistema informa que o nome ou a senha são inválidos.
2.1.2 O usuário corrige as informações.	

### **Caso de Uso 02 – Consultar ranking**

ID do Caso de Uso:	02
Nome do Caso de Uso:	Consultar ranking
Criado por:	Eric Kieling
Data de Criação:	14/06/2006
Última Atualização Realizada por:	Eric Kieling
Data da Última Atualização:	13/11/2006
Atores:	Jogador ou Administrador
Propósito:	O usuário deve ser capaz de consultar o ranking que contém a pontuação dos melhores jogadores em um determinado cenário.
Pré-condições:	1. Usuário logado no sistema
Pós-condições:	1. O ranking é apresentado para o usuário.

<b>Fluxo Básico de Eventos</b>	
<b>Ações do Ator:</b>	<b>Ações do Sistema:</b>
	1. O sistema apresenta uma lista contendo todos os cenários cadastrados no sistema.
2. O usuário seleciona o cenário que ele deseja consultar o ranking.	
	2. Se existem pontuações associadas a este cenário, o sistema informa os 20 melhores jogadores e para cada jogador as suas três melhores pontuações. Senão, o sistema informa para o usuário que não existe ranking para este cenário.

### **Caso de Uso 03 – Modificar cadastro**

ID do Caso de Uso:	03
Nome do Caso de Uso:	Modificar cadastro
Criado por:	Eric Kieling
Data de Criação:	12/06/2006
Última Atualização Realizada por:	Ricardo Rosa
Data da Última Atualização:	13/11/2006
Atores:	Administrador e Jogador
Propósito:	O usuário deve ser capaz de modificar seus dados cadastrais (senha e nível).
Pré-condições:	1. Usuário deve estar logado no sistema.
Pós-condições:	1. Alterações cadastradas.

<b>Fluxo Básico de Eventos</b>	
<b>Ações do Ator:</b>	<b>Ações do Sistema:</b>
	1. O sistema apresenta os dados cadastrais do usuário (nome e nível).
2. O usuário pode alterar a sua senha e seu nível de experiência. Para alterar a senha o usuário irá informar a senha atual e duas vezes a nova senha.	

	2. O sistema valida se ele escolheu um nível de experiência e se as duas senhas informadas são iguais. Caso as informações estejam inconsistentes ir para <b>Subseção 2.1</b> .
	3. O sistema modifica os dados cadastrais do usuário.
<b>Subseção 2.1</b>	
<b>Ações do Ator:</b>	<b>Ações do Sistema:</b>
	2.1.1 O sistema informa que as senhas estão inconsistentes.
2.1.2 O usuário corrige as informações.	

#### **Caso de Uso 04 – Cadastrar Jogador**

ID do Caso de Uso:	04
Nome do Caso de Uso:	Cadastrar Jogador
Criado por:	Eric Kieling
Data de Criação:	12/06/2006
Última Atualização Realizada por:	Ricardo Rosa
Data da Última Atualização:	13/11/2006
Atores:	Jogador
Propósito:	O jogador deve informar seu nome, senha e o seu nível de experiência (iniciante ou avançado), para se cadastrar no sistema.
Pré-condições:	
Pós-condições:	1. Jogador cadastrado no sistema.

#### **Fluxo Básico de Eventos**

<b>Ações do Ator (Jogador):</b>	<b>Ações do Sistema:</b>
1. O jogador informa o seu nome, senha (duas vezes) e o seu nível de experiência (iniciante ou avançado).	

	2. O sistema valida se o nome do usuário é único e se as duas senhas informadas são iguais. Caso as informações estejam inconsistentes ir para <b>Subseção 2.1</b> .
	3. O sistema cadastra os dados do usuário.
Subseção 2.1	
Ações do Ator (Jogador):	Ações do Sistema:
	2.1.1 O sistema informa quais informações estão inválidas (nome repetido ou senhas diferentes).
2.1.2 O jogador corrige as informações.	

O diagrama a seguir (Figura 14), representa os casos de uso relacionados ao ator *Administrador*.

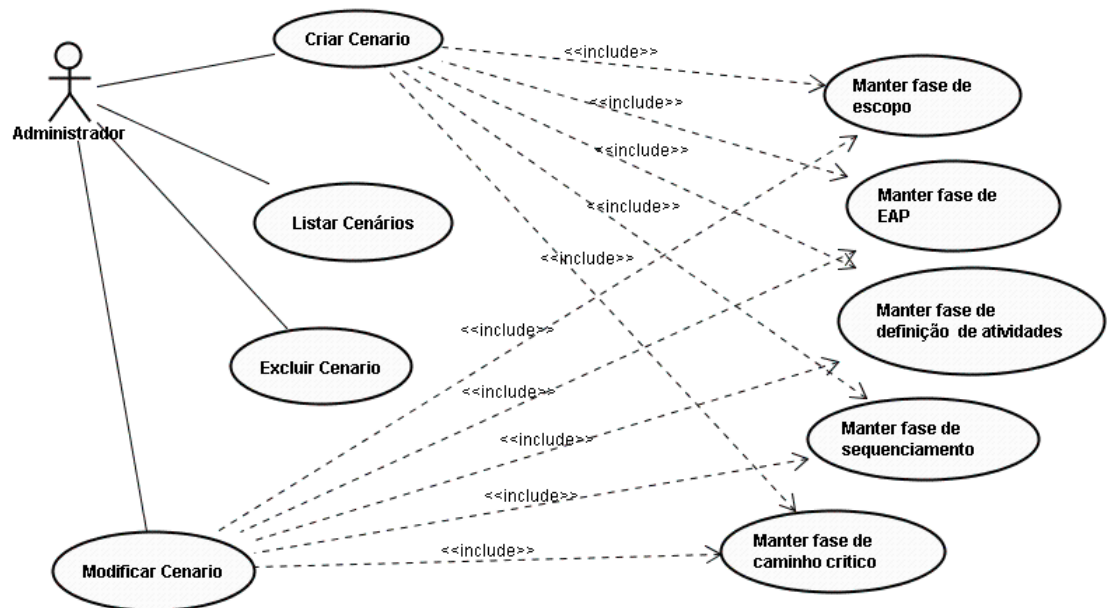


Figura 14 – Casos de uso do Administrador

Foram descritos os casos de uso *Listar Cenários*, *Excluir Cenário*, *Criar Cenário*, *Manter fase de escopo*, *Manter fase de EAP*, *Manter fase de definição de atividades*, *Manter fase de sequenciamento*, *Manter fase de caminho crítico* e *Modificar Cenário*.

**Caso de Uso 05 – Listar Cenários**

ID do Caso de Uso:	05
Nome do Caso de Uso:	Listar Cenários
Criado por:	Eric Kieling e Ricardo Rosa
Data de Criação:	12/06/2006
Última Atualização Realizada por:	Eric Kieling
Data da Última Atualização:	13/11/2006
Atores:	Administrador
Propósito:	O administrador do jogo deve ser capaz de listar os cenários cadastrados e utilizar esta lista para modificá-los e excluí-los.
Pré-condições:	1. Administrador logado.
Pós-condições:	1. A lista dos cenários é apresentada para o administrador.

**Fluxo Básico de Eventos**

<b>Ações do Ator (Administrador):</b>	<b>Ações do Sistema:</b>
1. O administrador seleciona a opção para listar os cenários do jogo.	
	2. O sistema apresenta a lista contendo o nome, o resumo e a dificuldade de todos os cenários cadastrados no jogo.
3. A partir da lista de cenários o administrador pode selecionar cenários para exclusão ou alteração (ver caso de uso <i>Excluir cenário</i> e caso de uso <i>Modificar cenário</i> ).	

**Caso de Uso 06 -- Excluir Cenário**

ID do Caso de Uso:	06
Nome do Caso de Uso:	Excluir Cenário
Criado por:	Eric Kieling e Ricardo Rosa
Data de Criação:	12/06/2006
Última Atualização Realizada por:	Eric Kieling
Data da Última Atualização:	13/11/2006
Atores:	Administrador

Propósito:	O Administrador do jogo deve ser capaz de excluir qualquer cenário.
Pré-condições:	1. Administrador logado.
Pós-condições:	1. O(s) cenário(s) é (são) excluído(s) do sistema. 2. O ranking referente ao(s) cenário(s) também é excluído.

Fluxo Básico de Eventos	
Ações do Ator (Administrador):	Ações do Sistema:
1. A partir da lista de cenários o administrador pode marcar um ou mais cenários para exclusão e selecionar a opção <i>Excluir</i> .	
	2. O sistema verifica se o administrador selecionou pelo menos um cenário e solicita a confirmação do administrador.
3. O administrador confirma a exclusão.	
	4. O sistema remove este(s) cenário(s) do sistema e também remove todas as informações do ranking referentes a este(s) cenário(s).

#### Caso de Uso 07 – Criar Cenário

ID do Caso de Uso:	07
Nome do Caso de Uso:	Criar Cenário
Criado por:	Eric Kieling e Ricardo Rosa
Data de Criação:	16/05/2006
Última Atualização Realizada por:	Ricardo Rosa
Data da Última Atualização:	13/11/2006
Atores:	Administrador
Propósito:	O administrador do jogo deve ser capaz de criar cenários para o jogo. Cada cenário poderá ser considerado fácil ou difícil. Cada cenário possui cinco fases (escopo, EAP, atividades, sequenciamento e caminho crítico).
Pré-condições:	1. Administrador logado.
Pós-condições:	1. Cenário cadastrado no jogo.

#### Fluxo Básico de Eventos

Ações do Ator (Administrador):	Ações do Sistema:
<p>1. O administrador informa o nome do cenário, a dificuldade, a quantidade de pontos necessários para o jogador iniciante e o jogador avançado passar de fase e a quantidade de vidas do jogador. Ele também informa se o jogador poderá consultar o algoritmo de caminho crítico e se o jogador deverá criar as ligações na fase de seqüenciamento. O administrador deve informar o número máximo de segundos que será utilizado para sortear as durações das atividades.</p>	
	<p>2. O sistema verifica se a pontuação necessária para os jogadores passarem de fase não é maior do que a pontuação total da fase.  O sistema valida se os pontos que o jogador avançado deve atingir para passar de fase são maiores que os pontos que o jogador iniciante deve atingir e os pontos desse são maiores que zero.  O sistema valida se a quantidade inicial de vidas é maior que zero.  O sistema valida se o número de segundos é um número entre um e trinta.  Caso as informações estejam inconsistentes ir para <b>Subseção 2.1</b>.</p>
	3. Ir para <i>include</i> “Manter fase escopo”.
	4. Ir para <i>include</i> “Manter fase EAP”.
	5. Ir para <i>include</i> “Manter fase definição de atividades”.
	6. Ir para <i>include</i> “Manter fase sequenciamento”.
	7. Ir para <i>include</i> “Manter fase caminho crítico”.
<b>Subseção 2.1</b>	
Ações do Ator:	Ações do Sistema:
	2.1.1 O sistema informa quais informações estão inválidas.
2.1.2 O administrador corrige as informações.	

**Caso de Uso 08 – Manter fase de escopo**

ID do Caso de Uso:	08
Nome do Caso de Uso:	Manter fase de escopo
Criado por:	Eric Kieling e Ricardo Rosa
Data de Criação:	16/05/2006
Última Atualização Realizada por:	Eric Kieling
Data da Última Atualização:	13/11/2006
Atores:	Administrador
Propósito:	O administrador do jogo deve ser capaz de criar ou modificar um texto descrevendo o escopo do cenário. Ele pode criar ou remover perguntas relacionadas a este texto e definir alternativas para cada pergunta.
Pré-condições:	1. Passar pelo caso de uso “Criar cenário”.
Pós-condições:	1. Fase de escopo cadastrada.

<b>Fluxo Básico de Eventos</b>	
<b>Ações do Ator (Administrador):</b>	<b>Ações do Sistema:</b>
1. O administrador descreve textualmente o escopo do cenário.	
2. O administrador adiciona e/ou remove perguntas, para cada uma define quatro alternativas e indica qual é a correta.	
	3. O sistema valida se a descrição textual possui o mínimo de linhas exigido. Caso as informações estejam inconsistentes ir para <b>Subseção 3.1</b> .
	4. O sistema valida se foram cadastradas no mínimo quatro perguntas e se para cada pergunta foi indicado qual alternativa é a correta. Caso as informações estejam inconsistentes ir para <b>Subseção 3.1</b> .
	5. O sistema armazena as informações.
<b>Subseção 3.1</b>	
<b>Ações do Ator:</b>	<b>Ações do Sistema:</b>
	3.1.1 O sistema informa quais informações estão inválidas.

3.1.2 O administrador corrige as informações.	
---	--

### Caso de Uso 09 – Manter fase de EAP

ID do Caso de Uso:	09
Nome do Caso de Uso:	Manter fase de EAP
Criado por:	Eric Kieling e Ricardo Rosa
Data de Criação:	20/05/2006
Última Atualização Realizada por:	Eric Kieling
Data da Última Atualização:	13/11/2006
Atores:	Administrador
Propósito:	O administrador do jogo pode criar ou modificar uma EAP de forma visual (esta EAP será utilizada como gabarito durante o jogo). O administrador também pode cadastrar ou remover itens que não fazem parte da EAP (estes itens serão utilizados como opção para o jogador quando ele jogar esta fase).
Pré-condições:	1. Passar pelo caso de uso “Manter fase de escopo”
Pós-condições:	1. Fase EAP cadastrada.

### Fluxo Básico de Eventos

Ações do Ator (Administrador):	Ações do Sistema:
1- O administrador adiciona e/ou remove nodos da EAP (mínimo dois nodos) inserindo primeiramente o nodo raiz e depois criando os nodos filhos.	
2- O administrador informa para cada nodo da EAP um rótulo.	
3- O administrador pode criar ou remover uma lista de rótulos que não fazem parte da EAP, mas que servirão como opções, juntamente com os rótulos corretos, para o jogador criar a EAP.	
	4- O sistema valida se a EAP possui no mínimo dois nodos, se todos os nodos possuem rótulos e se todos os nodos possuem ligações. Caso as informações estejam inconsistentes ir para <b>Subseção 4.1</b> .
	5- O sistema armazena as informações.

<b>Subseção 4.1</b>	
<b>Ações do Ator:</b>	<b>Ações do Sistema:</b>
	4.1.1 O sistema informa que a EAP possui menos itens que o mínimo aceitável ou que existem nodos sem rótulos ou nodos não conectados.
4.1.2 O administrador corrige a EAP.	

#### **Caso de Uso 10 – Manter fase de definição de atividades**

ID do Caso de Uso:	10
Nome do Caso de Uso:	Manter fase de definição de atividades
Criado por:	Eric Kieling e Ricardo Rosa
Data de Criação:	20/05/2006
Última Atualização Realizada por:	Ricardo Rosa
Data da Última Atualização:	13/11/2006
Atores:	Administrador
Propósito:	O administrador pode criar ou modificar, para cada pacote de trabalho da EAP, uma lista de atividades. Ele pode criar ou modificar uma lista de atividades que não estão relacionadas à EAP, para dificultar a escolha do jogador.
Pré-condições:	1. Passar pelo caso de uso “Manter fase EAP”
Pós-condições:	1. Fase de definição de atividades cadastrada.

<b>Fluxo Básico de Eventos</b>	
<b>Ações do Ator (Administrador):</b>	<b>Ações do Sistema:</b>
	1 - O sistema irá apresentar todos os pacotes de trabalho da EAP criada na fase anterior.
2 - O administrador adiciona e/ou remove atividades para cada pacote de trabalho da EAP.	
3 - O administrador pode criar ou remover uma lista de atividades que não fazem parte dos pacotes de trabalho da EAP, mas que servirão como opções, juntamente com as atividades dos pacotes, para dificultar a escolha das atividades pelo jogador.	

	4 - O sistema valida se existe pelo menos quatro atividades vinculadas à EAP. Caso as informações estejam inconsistentes vá para <b>Subseção 4.1</b> .
	5- O sistema armazena os dados.
<b>Subseção 4.1</b>	
<b>Ações do Ator:</b>	<b>Ações do Sistema:</b>
	4.1.1 O sistema informa que no mínimo quatro atividades devem ser cadastradas.
4.1.2 O administrador corrige as informações.	

#### **Caso de Uso 11 – Manter fase de sequeciamento**

ID do Caso de Uso:	11
Nome do Caso de Uso:	Manter fase de sequeciamento
Criado por:	Eric Kieling e Ricardo Rosa
Data de Criação:	22/05/2006
Última Atualização Realizada por:	Ricardo Rosa
Data da Última Atualização:	13/11/2006
Atores:	Administrador
Propósito:	O administrador do jogo recebe a lista de atividades, que foi definida na fase anterior. Ele deve ligar as atividades conforme suas dependências.
Pré-condições:	1. Passar pelo caso de uso “Manter fase definição de atividades”
Pós-condições:	1. Fase sequeciamento cadastrada.

<b>Fluxo Básico de Eventos</b>	
<b>Ações do Ator (Administrador):</b>	<b>Ações do Sistema:</b>
	1 – O sistema apresenta as atividades que foram cadastradas na fase anterior.
2 - O administrador utiliza as atividades cadastradas na fase anterior para criar o diagrama de rede.	
3 - O administrador liga cada atividade conforme suas dependências, sendo que só pode associar uma atividade à outra que esteja em um nível maior.	

4 – O administrador informa qual é o nodo final do diagrama.	
	5 - O sistema valida se no mínimo duas atividades foram utilizadas no diagrama e se uma foi marcada como nodo final. Caso as informações estejam inconsistentes ir para <b>Subseção 5.1</b> .
	6 - O sistema valida se cada atividade possui no mínimo uma seta entrando e uma seta saindo. Exceção para o <i>start</i> (só sai seta) e <i>finish</i> (só entra seta). Caso as informações estejam inconsistentes ir para <b>Subseção 5.1</b> .
	7- O sistema armazena as informações.
<b>Subseção 5.1</b>	
<b>Ações do Ator:</b>	<b>Ações do Sistema:</b>
	5.1.1 O sistema informa quais informações estão inválidas.
5.1.2 O administrador corrige as informações.	

#### **Caso de Uso 12 – Manter fase de caminho crítico**

ID do Caso de Uso:	12
Nome do Caso de Uso:	Manter fase caminho crítico
Criado por:	Eric Kieling e Ricardo Rosa
Data de Criação:	22/05/2006
Última Atualização Realizada por:	Ricardo Rosa
Data da Última Atualização:	13/11/2006
Atores:	Administrador
Propósito:	Atualizar o diagrama de sequenciamento de atividades com durações aleatórias e com os valores de IMC, IMT, TMC e TMT.
Pré-condições:	1. Passar pelo caso de uso “Manter fase sequenciamento”
Pós-condições:	1. Fase caminho crítico cadastrada.

#### **Fluxo Básico de Eventos**

<b>Ações do Ator (Administrador):</b>	<b>Ações do Sistema:</b>
---------------------------------------	--------------------------

1. O administrador clica no botão “Validar / Calcular caminho crítico”.	
	2. O sistema sorteia durações para todas as atividades (utilizando as configurações do cenário) e calcula o IMC, IMT, TMC e TMT para cada atividade.
2. O administrador verifica se os dados foram gerados corretamente e clica no botão “Salvar”.	
	3. O sistema armazena as informações do cenário.

### Caso de Uso 13 – Modificar Cenário

ID do Caso de Uso:	13
Nome do Caso de Uso:	Modificar Cenário
Criado por:	Eric Kieling
Data de Criação:	15/06/2006
Última Atualização Realizada por:	Ricardo Rosa
Data da Última Atualização:	20/06/2006
Atores:	Administrador
Propósito:	O Administrador do jogo deve ser capaz de modificar qualquer informação de um cenário desde que ele mantenha o cenário em um estado consistente.
Pré-condições:	1. Cenário cadastrado. 2. Administrador logado.
Pós-condições:	1. Informações do cenário alteradas.

### Fluxo Básico de Eventos

Ações do Ator (Administrador):	Ações do Sistema:
	1. O sistema apresenta as informações gerais do cenário (nome, dificuldade, quantidade inicial de vidas e a quantidade de pontos necessários para o jogador iniciante e o jogador avançado passar de fase, número máximo de segundos para a duração das atividades).
2. O administrador altera as informações desejadas.	

	<p>3. O sistema valida se os pontos para passar de fase não são maiores do que a pontuação total da fase.</p> <p>O sistema valida se os pontos que o jogador avançado deve atingir para passar de fase são maiores que os pontos que o jogador iniciante deve atingir e os pontos desse são maiores que zero.</p> <p>O sistema valida se a quantidade de vidas é maior que zero.</p> <p>O sistema valida se a duração das atividades é maior ou igual a um e menor ou igual a trinta.</p> <p>Caso as informações estejam inconsistentes ir para <b>Subseção 2.1</b>.</p>
	3. Ir para <i>include</i> “Manter fase escopo”.
	4. Ir para <i>include</i> “Manter fase EAP”.
	5. Ir para <i>include</i> “Manter fase definição de atividades”.
	6. Ir para <i>include</i> “Manter fase sequenciamento”.
	7. Ir para <i>include</i> “Manter fase caminho crítico”.
<b>Subseção 2.1</b>	
<b>Ações do Ator:</b>	<b>Ações do Sistema:</b>
	2.1.1 O sistema informa quais informações estão inválidas.
2.1.2 O administrador corrige as informações.	

A Figura 15 a seguir, representa o diagrama de atividades referente ao caso de uso *Criar Cenário*, a qual mostra o fluxo de interações entre o *Administrador* e o *Sistema*.

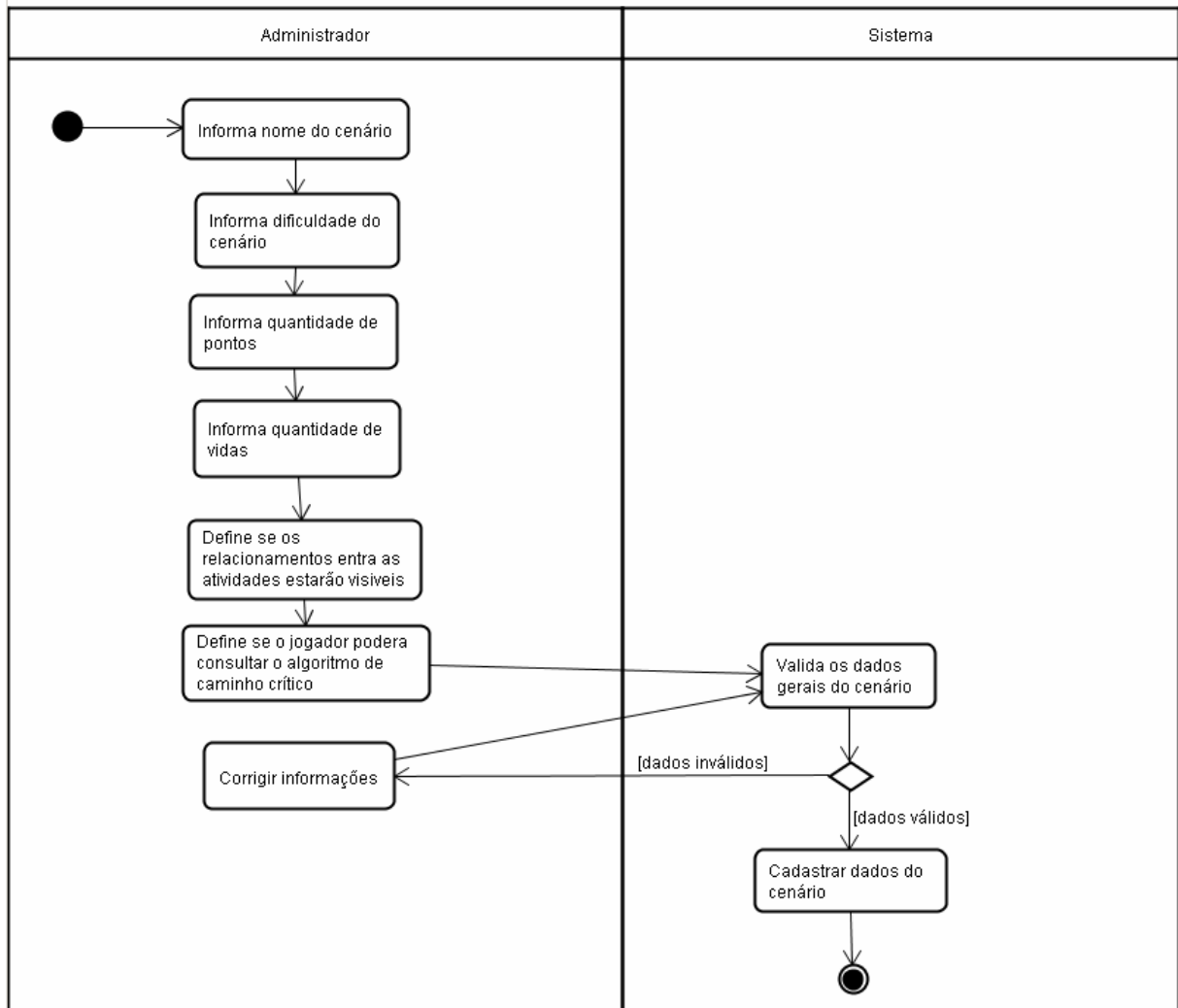


Figura 15 – Diagrama de atividades do caso de uso *Criar Cenário*.

A Figura 16 a seguir, representa o diagrama de atividades referente ao caso de uso *Manter fase de Escopo*, a qual mostra o fluxo de interações entre o *Administrador* e o *Sistema*.

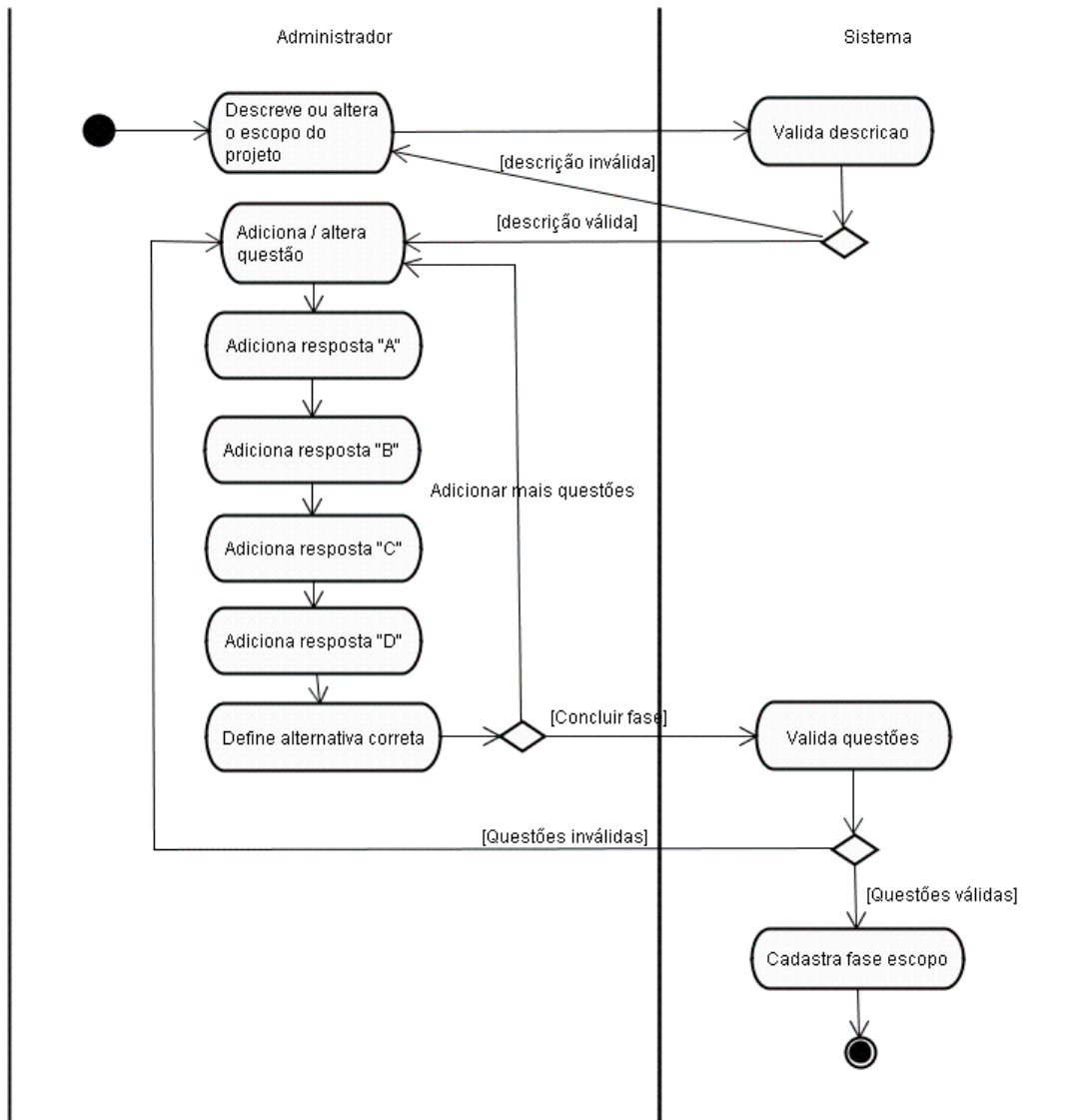


Figura 16 – Diagrama de atividades do caso de uso *Manter fase de Escopo*.

A Figura 17 a seguir, representa o diagrama de atividades referente ao caso de uso *Manter fase de EAP*, a qual mostra o fluxo de interações entre o *Administrador* e o *Sistema*.

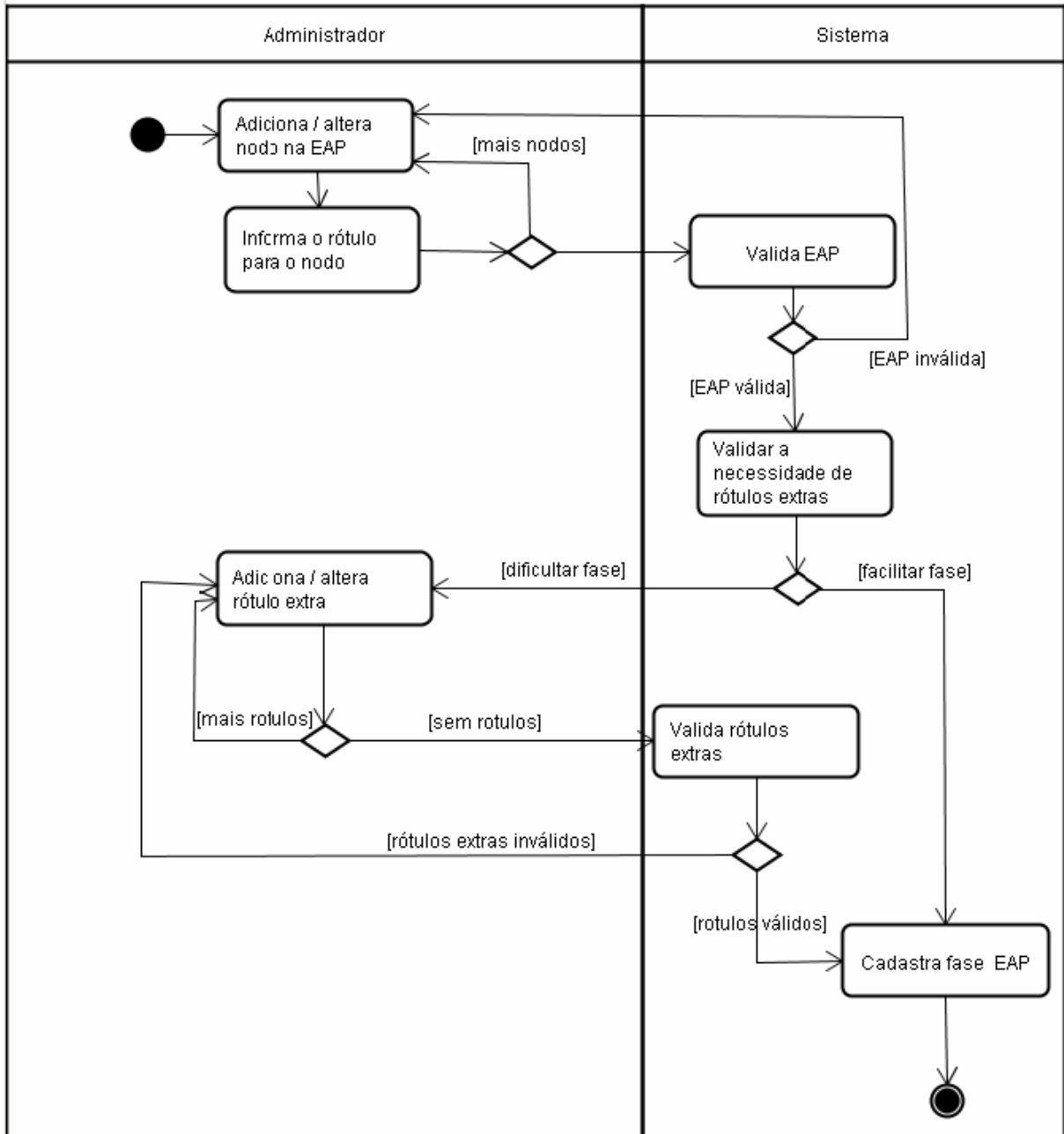


Figura 17 – Diagrama de atividades do caso de uso *Manter fase de EAP*.

A Figura 18 a seguir, representa o diagrama de atividades referente ao caso de uso *Manter fase de definição de atividades*, a qual mostra o fluxo de interações entre o *Administrador* e o *Sistema*.

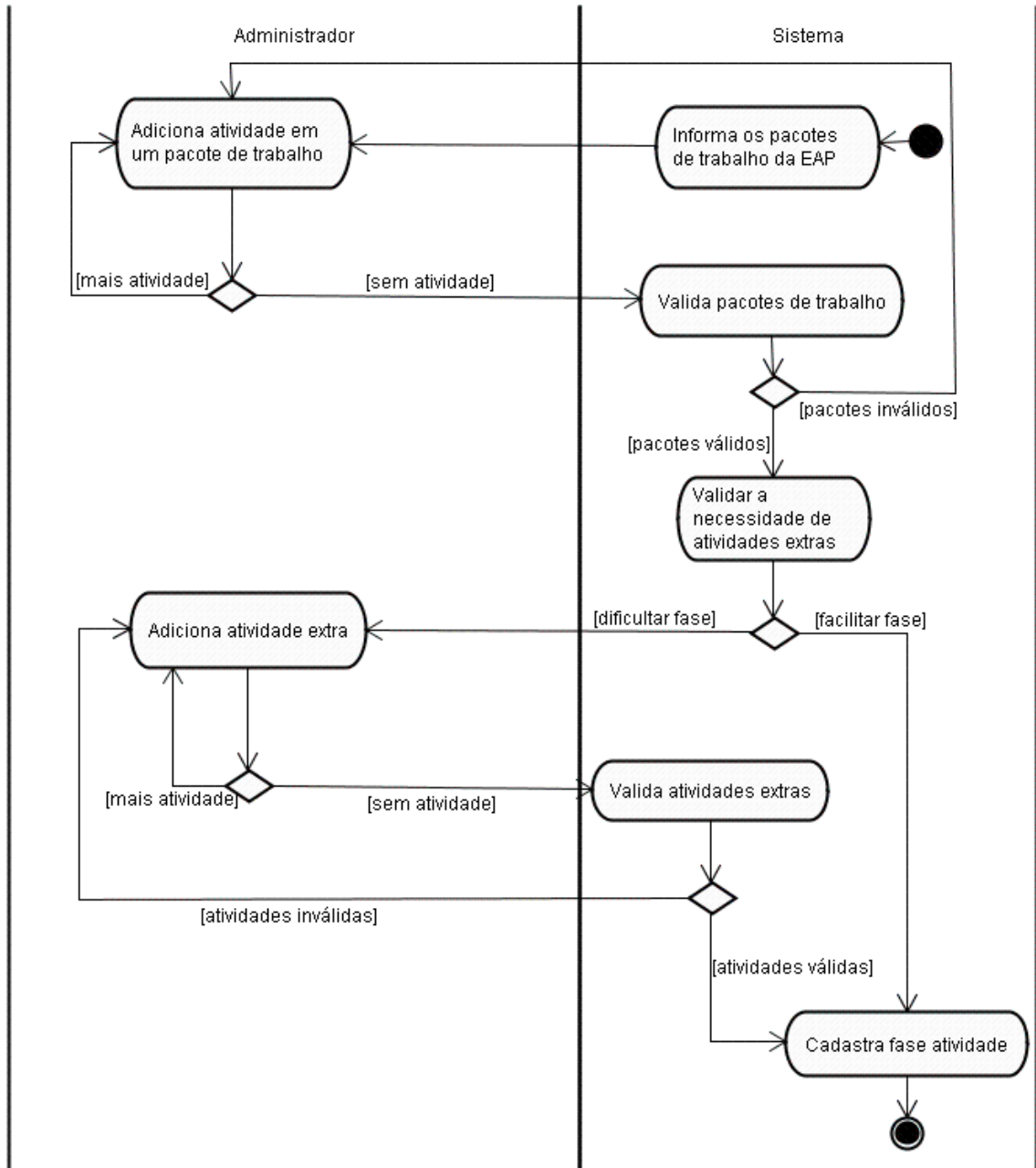


Figura 18 – Diagrama de atividades do caso de uso *Manter fase de definição de atividades*.

A Figura 19 a seguir, representa o diagrama de atividades referente ao caso de uso *Manter fase de sequenciamento*, a qual mostra o fluxo de interações entre o *Administrador* e o *Sistema*.

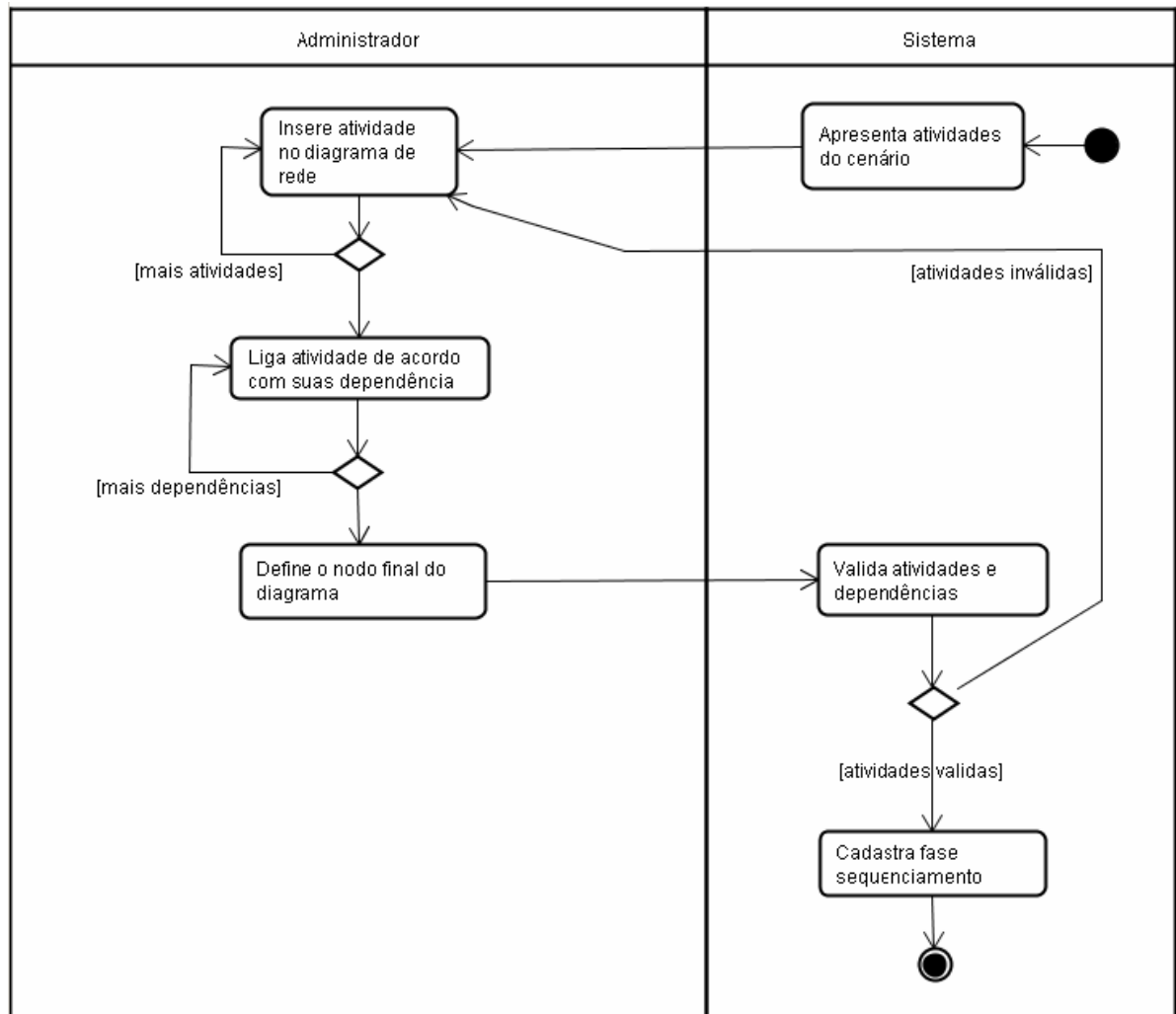


Figura 19 – Diagrama de atividades do caso de uso *Manter fase de sequenciamento*.

A Figura 20 a seguir, representa o diagrama de atividades referente ao caso de uso *Manter fase de caminho crítico*, a qual mostra o fluxo de interações entre o *Administrador* e o *Sistema*.

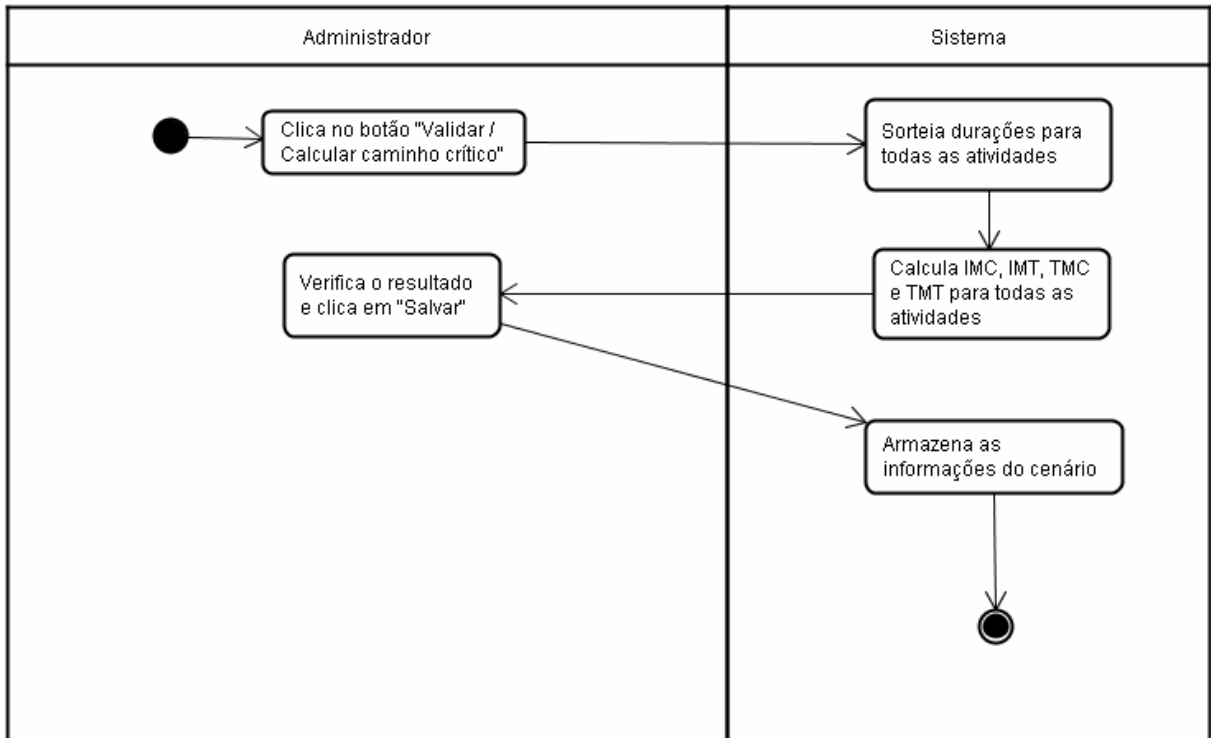


Figura 20 – Diagrama de atividades do caso de uso *Manter fase de caminho crítico*.

O diagrama a seguir (Figura 21), representa os casos de uso relacionados ao ator *Jogador* no módulo *Jogo* do sistema.

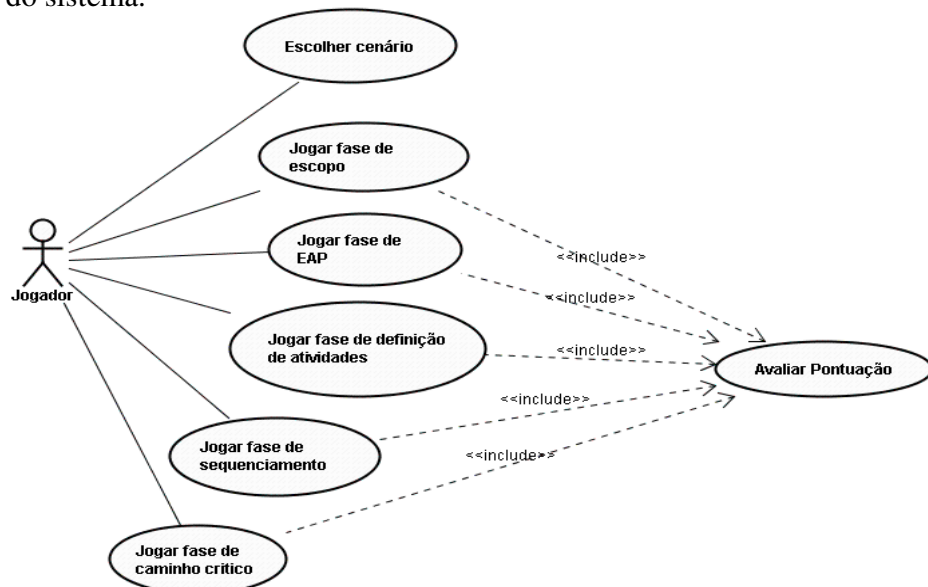


Figura 21 – Casos de uso do Jogo

Em relação aos casos de uso, foram descritos: *Escolher cenário*, *Jogar fase de escopo*, *Jogar fase de EAP*, *Jogar fase de definição atividades*, *Jogar fase de sequenciamento*, *Jogar fase de caminho crítico* e *Avaliar pontuação*.

#### Caso de Uso 14 – Escolher cenário

ID do Caso de Uso:	14
Nome do Caso de Uso:	Escolher cenário
Criado por:	Eric Kieling e Ricardo Rosa
Data de Criação:	20/05/2006
Última Atualização Realizada por:	Eric Kieling
Data da Última Atualização:	15/11/2006
Atores:	Jogador
Propósito:	O jogador deve escolher um cenário para jogar a partir da lista de cenários cadastrados no sistema.
Pré-condições:	1. Jogador logado. 2. Cenário cadastrado.
Pós-condições:	1. Jogador começa a jogar o cenário escolhido.

#### Fluxo Básico de Eventos

Ações do Ator (Jogador):	Ações do Sistema:
	1. O sistema apresenta a lista de cenários cadastrados. A lista contém o nome, resumo e a dificuldade de cada cenário.
2. A partir destas informações o jogador seleciona qual cenário ele deseja jogar.	
	3. O sistema apresenta a primeira fase do cenário (ver caso de uso <i>Jogar fase escopo</i> ).

#### Caso de Uso 15 – Jogar fase de escopo

ID do Caso de Uso:	15
Nome do Caso de Uso:	Jogar fase de escopo
Criado por:	Eric Kieling e Ricardo Rosa
Data de Criação:	20/05/2006
Última Atualização Realizada por:	Eric Kieling
Data da Última Atualização:	15/11/2006

Atores:	Jogador
Propósito:	O jogador deve ler a descrição do escopo do cenário e responder as questões de múltipla escolha.
Pré-condições:	1. Jogador logado. 2. Fase cadastrada.
Pós-condições:	1. Jogador passa para a próxima fase ou perde vida e continua na mesma fase.

<b>Fluxo Básico de Eventos</b>	
<b>Ações do Ator (Jogador):</b>	<b>Ações do Sistema:</b>
1. O jogador lê a descrição do escopo do cenário.	
2. O jogador informa uma alternativa correta para cada uma das questões.	
	3. O sistema valida se todas as questões possuem respostas. Caso as informações estejam inconsistentes vá para <b>Subseção 3.1</b> .
	4. O sistema corrige as respostas do jogador e calcula sua pontuação. Avaliação: <ul style="list-style-type: none"> <li>• Se ele responder corretamente a questão ele ganha X pontos;</li> <li>• Se ele errar a questão ele não ganha pontos.</li> <li>• X é igual a 1000 / numero de questões.</li> </ul>
	5. Ir para o caso de uso “Avaliar pontuação”.
	6. O sistema cadastra o resultado no banco de dados.
<b>Subseção 3.1</b>	
<b>Ações do Ator:</b>	<b>Ações do Sistema:</b>
	3.1.1 O sistema informa quais questões não foram respondidas.
3.1.2 O jogador corrige as informações.	

### **Caso de Uso 16 – Jogar fase de EAP**

ID do Caso de Uso:	16
--------------------	----

Nome do Caso de Uso:	Jogar fase de EAP
Criado por:	Eric Kieling e Ricardo Rosa
Data de Criação:	20/05/2006
Última Atualização Realizada por:	Ricardo Rosa
Data da Última Atualização:	15/11/2006
Atores:	Jogador
Propósito:	O jogador deve escolher na lista de rótulos quais destes fazem parte da EAP, associando-os com os nodos da EAP que estará montada na tela.
Pré-condições:	1. Jogador deve ter passado pela fase de escopo. 2. Fase cadastrada.
Pós-condições:	1. Jogador passa para a próxima fase ou perde uma vida e continua na mesma fase.

<b>Fluxo Básico de Eventos</b>	
<b>Ações do Ator (Jogador):</b>	<b>Ações do Sistema:</b>
1. O jogador analisa a lista de rótulos, para descobrir quais fazem parte da EAP.	
2. O jogador arrasta os rótulos escolhidos para cada nodo da EAP.	
	3 . O sistema valida se todos os nodos da EAP foram preenchidos. Caso as informações estejam inconsistentes ir para <b>Subseção 3.1</b> .
	4. O sistema corrige as respostas do jogador e calcula sua pontuação. Avaliação: <ul style="list-style-type: none"> <li>• Se ele colocar um item no lugar certo e este item tiver as ligações corretas ele ganha X pontos;</li> <li>• Se o item que ele colocou estiver ligando os nodos certos ele ganha 0.5X pontos e se os nodos que ligam ele estiverem corretos ele ganha 0.5X pontos;</li> <li>• Se ele colocar um item que não pertence à EAP ele não ganha pontos.</li> </ul>
	5. Ir para o caso de uso “Avaliar pontuação”.
	6. O sistema cadastra o resultado no banco de dados.

<b>Subseção 3.1</b>	
<b>Ações do Ator:</b>	<b>Ações do Sistema:</b>
	3.1.1 O sistema informa quais nodos não foram preenchidos.
3.1.2 O jogador corrige as informações.	

### **Caso de Uso 17 – Jogar fase de definição de atividades**

ID do Caso de Uso:	17
Nome do Caso de Uso:	Jogar fase de definição de atividades
Criado por:	Eric Kieling e Ricardo Rosa
Data de Criação:	22/05/2006
Última Atualização Realizada por:	Eric Kieling
Data da Última Atualização:	15/11/2006
Atores:	Jogador
Propósito:	O jogador deve vincular atividades aos pacotes de trabalho da EAP.
Pré-condições:	1. Jogador logado. 2. Fase cadastrada.
Pós-condições:	1. Jogador passa para a próxima fase ou perde uma vida e continua na mesma fase

<b>Fluxo Básico de Eventos</b>	
<b>Ações do Ator (Jogador):</b>	<b>Ações do Sistema:</b>
	1. O sistema informa os pacotes da EAP e uma lista de possíveis atividades.
2. O jogador analisa a lista de atividades, para descobrir quais estão vinculadas aos pacotes de trabalho da EAP.	
3. O jogador vincula as atividades escolhidas aos pacotes de trabalho.	
	4. O sistema corrige as respostas do jogador e calcula sua pontuação. Avaliação: <ul style="list-style-type: none"> <li>• Se ele vincular uma atividade ao pacote correto ele ganha X pontos;</li> <li>• Se ele vincular uma atividade que pertence ao escopo, mas ao pacote errado, ele ganha X/4 pontos;</li> </ul>

	<ul style="list-style-type: none"> <li>• Se ele vincular uma atividade que não pertence ao escopo ele perde X pontos.</li> <li>• X é igual a 1000 / número de atividades corretas</li> </ul>
	5. Ir para o caso de uso “Avaliar pontuação”.
	6. O sistema cadastra o resultado no banco de dados.

### Caso de Uso 18 – Jogar fase de sequenciamento

ID do Caso de Uso:	18
Nome do Caso de Uso:	Jogar fase de sequenciamento
Criado por:	Eric Kieling e Ricardo Rosa
Data de Criação:	22/05/2006
Última Atualização Realizada por:	Ricardo Rosa
Data da Última Atualização:	15/11/2006
Atores:	Jogador
Propósito:	O jogador deve associar as atividades ao diagrama de sequenciamento de atividades e informar as suas dependências.
Pré-condições:	1. Jogador logado. 2. Fase cadastrada.
Pós-condições:	1. Jogador passa para a próxima fase ou o jogador perde uma vida e continua na mesma fase.

### Fluxo Básico de Eventos

Ações do Ator (Jogador):	Ações do Sistema:
	1. O sistema apresenta a lista de atividades que foram utilizadas na fase anterior e monta a estrutura do diagrama de sequenciamento de atividades (apenas os retângulos vazios).
2. O jogador associa as atividades nos retângulos do diagrama de sequenciamento de atividades.	
3. O jogador liga as atividades conforme suas dependências.	
	4. O sistema valida se todas as atividades da lista de atividades foram utilizadas e se cada atividade tem no mínimo uma seta entrando e uma seta saindo. Exceção para o

	<i>start</i> (só sai seta) e <i>finish</i> (só entra seta). Caso as informações estejam inconsistentes vá para <b>Subseção 4.1</b> .
	5. O sistema corrige as respostas do jogador e calcula sua pontuação. Avaliação: <ul style="list-style-type: none"> <li>• Se ele colocar um item no lugar certo e este item tiver as ligações corretas ele ganha X pontos;</li> <li>• Se o item que ele colocou estiver ligando os nodos certos ele ganha 0.5X pontos e se os nodos que ligam ele estiverem corretos ele ganha 0.5X pontos;</li> <li>• Se ele um nodo com mais ligações que este deveria ter ele perde (0.5X/numLigações) pontos.</li> <li>• Se ele associar a atividade no nível errado ele não ganha pontos;</li> </ul>
	6. Ir para o caso de uso “Avaliar pontuação”.
	7. O sistema cadastra o resultado no banco de dados.
<b>Subseção 4.1</b>	
<b>Ações do Ator:</b>	<b>Ações do Sistema:</b>
	4.1.1 O sistema informa quais informações estão inválidas.
4.1.2 O jogador corrige as informações.	

#### Caso de Uso 19 – Jogar fase de caminho crítico

ID do Caso de Uso:	19
Nome do Caso de Uso:	Jogar fase de caminho crítico
Criado por:	Eric Kieling e Ricardo Rosa
Data de Criação:	22/05/2006
Última Atualização Realizada por:	Ricardo Rosa
Data da Última Atualização:	15/11/2006
Atores:	Jogador
Propósito:	O jogador deve calcular a primeira data de início (PDI), última data de início (UDI), primeira data de término (PDT) e última data de término (UDT) de cada atividade e marcar as atividades que fazem parte do caminho crítico.

Pré-condições:	1. Jogador logado. 2. Fase cadastrada.
Pós-condições:	1. Jogador passa para a próxima fase ou perde uma vida e continua na mesma fase

<b>Fluxo Básico de Eventos</b>	
<b>Ações do Ator (Jogador):</b>	<b>Ações do Sistema:</b>
	1. O sistema apresenta o diagrama de sequenciamento de atividades.
2. O jogador calcula a primeira data de início (PDI), última data de início (UDI), primeira data de término (PDT) e última data de término (UDT) de cada atividade.	
3. O jogador marca todas as atividades que fazem parte do caminho crítico.	
	4. O sistema valida se todas as atividades possuem os campos PDI, UDI, PDT e UDT informados. O sistema também verifica se o caminho crítico foi informado. Caso as informações estejam inconsistentes vá para <b>Subseção 4.1</b> .
	5. O sistema corrige as respostas do jogador e calcula sua pontuação. Avaliação: <ul style="list-style-type: none"> <li>• Se ele acertar o cálculo dos campos FOLGA, PDI, UDI, PDT e UDT ele ganha X pontos, cada campo vale X/5;</li> </ul>
	6. Ir para o caso de uso “Avaliar pontuação”.
	7. O sistema cadastra o resultado no banco de dados.
<b>Subseção 4.1</b>	
<b>Ações do Ator:</b>	<b>Ações do Sistema:</b>
	4.1.1 O sistema informa quais informações estão inválidas.
4.1.2 O jogador corrige as informações.	

### **Caso de Uso 20 – Avaliar pontuação**

ID do Caso de Uso:	20
Nome do Caso de Uso:	Avaliar pontuação

Criado por:	Eric Kieling e Ricardo Rosa
Data de Criação:	03/06/2006
Última Atualização Realizada por:	Eric Kieling
Data da Última Atualização:	15/11/2006
Atores:	Sistema
Propósito:	O sistema avalia a pontuação do jogador no final de cada fase. Dependendo da pontuação o jogador perde vida ou passa para a próxima fase.
Pré-condições:	1. Fase finalizada.
Pós-condições:	1. Jogador passa para a próxima fase ou perde vida e continua na mesma ou o jogo acaba.

### Fluxo Básico de Eventos

#### Ações do Sistema:

1 – O sistema mostra a pontuação do jogador, após este finalizar uma fase.

2 – O sistema compara a pontuação do jogador (p1) com a pontuação necessária para passar de fase (p2), considerando o nível do jogador, se  $p1 \geq p2$  ver **Subseção 2.1**, senão ver **Subseção 2.2**.

3 – O sistema informa ao jogador se ele passa para a próxima fase ou continua na mesma ou o jogo acaba.

#### Subseção 2.1

##### Ações do Sistema:

2.1.1 Caso a pontuação do jogador for igual à pontuação máxima da fase ele ganha uma vida extra.

2.1.2 O sistema mostra a resposta correta da fase para o jogador, antes deste passar de fase.

#### Subseção 2.2

##### Ações do Sistema:

2.2.1 Caso a quantidade de vida do jogador for igual a zero o sistema informa que ele perdeu e mostra a mensagem de “Game Over”, senão o jogador perde uma vida e continua na mesma fase.

O diagrama a seguir (Figura 22), representa os casos de uso relacionados ao ator *Jogador* no módulo *Tutorial* do sistema.

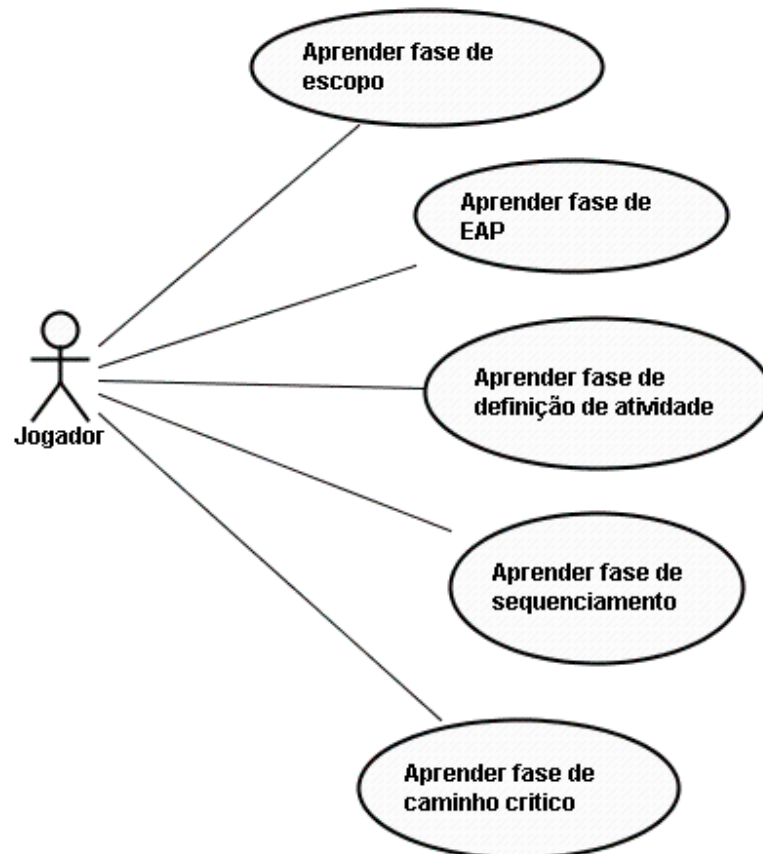


Figura 22 – Casos de uso do Tutorial

Foram descritos os casos de uso *Aprender fase de escopo*, *Aprender fase de EAP*, *Aprender fase de definição de atividades*, *Aprender fase de sequenciamento* e *Aprender fase de caminho crítico*.

#### Caso de Uso 21 – Aprender fase de escopo

ID do Caso de Uso:	21
Nome do Caso de Uso:	Aprender fase de escopo
Criado por:	Eric Kieling e Ricardo Rosa
Data de Criação:	15/06/2006
Última Atualização Realizada por:	Eric Kieling
Data da Última Atualização:	15/11/2006
Atores:	Jogador
Propósito:	O jogador deve aprender a jogar na fase de escopo através de um exemplo pratico.

Pré-condições:	1. Jogador logado.
Pós-condições:	1. Jogador passa para a próxima fase do tutorial.

Fluxo Básico de Eventos	
Ações do Ator (Jogador):	Ações do Sistema:
	1. O sistema apresenta a definição de escopo de projeto e descreve de maneira sucinta o objetivo desta fase e como funciona a sua pontuação.
2. O jogador seleciona a opção “avançar”.	
	3. O sistema apresenta a mesma tela do caso de uso <i>Jogar fase</i> escopo, contendo a descrição do cenário exemplo juntamente com as perguntas relacionadas a ele.
4. O jogador seleciona a opção “avançar”.	
	5. O sistema marca as opções corretas de cada uma das questões.
6. O jogador seleciona a opção “avançar” para ir para a próxima fase do tutorial.	

#### Caso de Uso 22 – Aprender fase de EAP

ID do Caso de Uso:	22
Nome do Caso de Uso:	Aprender fase de EAP
Criado por:	Eric Kieling e Ricardo Rosa
Data de Criação:	15/06/2006
Última Atualização Realizada por:	Eric Kieling
Data da Última Atualização:	16/06/2006
Atores:	Jogador
Propósito:	O jogador deve aprender a jogar na fase de EAP através de um exemplo pratico.
Pré-condições:	1. Jogador logado.
Pós-condições:	1. Jogador passa para a próxima fase do tutorial.

Fluxo Básico de Eventos	
Ações do Ator (Jogador):	Ações do Sistema:

	1. O sistema apresenta a definição de EAP e descreve de maneira sucinta o objetivo desta fase e como funciona a sua pontuação.
2. O jogador seleciona a opção “avançar”.	
	3. O sistema apresenta a mesma tela do caso de uso <i>Jogar fase EAP</i> , contendo uma EAP sem rótulos e a lista de rótulos do cenário exemplo.
4. O jogador seleciona a opção “avançar”.	
	6. O sistema associa os rótulos à EAP.
6. O jogador seleciona a opção “avançar” para ir para a próxima fase do tutorial.	

### Caso de Uso 23 – Aprender fase de definição de atividades

ID do Caso de Uso:	23
Nome do Caso de Uso:	Aprender fase de definição de atividades
Criado por:	Eric Kieling e Ricardo Rosa
Data de Criação:	15/06/2006
Última Atualização Realizada por:	Eric Kieling
Data da Última Atualização:	16/06/2006
Atores:	Jogador
Propósito:	O jogador deve aprender a jogar na fase de definição de atividades através de um exemplo pratico.
Pré-condições:	1. Jogador logado.
Pós-condições:	1. Jogador passa para a próxima fase do tutorial.

### Fluxo Básico de Eventos

Ações do Ator (Jogador):	Ações do Sistema:
	1. O sistema apresenta o conceito de atividade e como ela está relacionada com os pacotes de trabalho da EAP. O sistema também descreve de maneira sucinta o objetivo desta fase e como funciona a sua pontuação.

2. O jogador seleciona a opção “avançar”.	
	3. O sistema apresenta a mesma tela do caso de uso <i>Jogar fase definição de atividades</i> , contendo a EAP da fase anterior e uma lista de atividades que podem ser associadas aos pacotes de trabalho da EAP.
4. O jogador seleciona a opção “avançar”.	
	6. O sistema associa as atividades aos pacotes de trabalho da EAP.
6. O jogador seleciona a opção “avançar” para ir para a próxima fase do tutorial.	

#### **Caso de Uso 24 – Aprender fase de sequenciamento**

ID do Caso de Uso:	24
Nome do Caso de Uso:	Aprender fase de sequenciamento
Criado por:	Eric Kieling e Ricardo Rosa
Data de Criação:	15/06/2006
Última Atualização Realizada por:	Eric Kieling
Data da Última Atualização:	16/06/2006
Atores:	Jogador
Propósito:	O jogador deve aprender a jogar na fase de sequenciamento de atividades através de um exemplo prático.
Pré-condições:	1. Jogador logado.
Pós-condições:	1. Jogador passa para a próxima fase do tutorial.

#### **Fluxo Básico de Eventos**

<b>Ações do Ator (Jogador):</b>	<b>Ações do Sistema:</b>
	1. O sistema apresenta o conceito de sequenciamento de atividades e descreve de maneira sucinta o objetivo desta fase e como funciona a sua pontuação.
2. O jogador seleciona a opção “avançar”.	
	3. O sistema apresenta a mesma tela do caso de uso <i>Jogar fase sequenciamento</i> , contendo o diagrama de sequenciamento de atividades vazio.

4. O jogador seleciona a opção “avançar”.	
	6. O sistema associa as atividades às caixas vazias e utiliza setas para criar dependências entre elas.
6. O jogador seleciona a opção “avançar” para ir para a próxima fase do tutorial.	

### Caso de Uso 25 – Aprender fase de caminho crítico

ID do Caso de Uso:	25
Nome do Caso de Uso:	Aprender fase de caminho crítico
Criado por:	Eric Kieling e Ricardo Rosa
Data de Criação:	15/06/2006
Última Atualização Realizada por:	Eric Kieling
Data da Última Atualização:	16/06/2006
Atores:	Jogador
Propósito:	O jogador deve aprender a jogar na fase caminho crítico através de um exemplo pratico.
Pré-condições:	1. Jogador logado.
Pós-condições:	1. Jogador passa para a próxima fase do tutorial.

### Fluxo Básico de Eventos

Ações do Ator (Jogador):	Ações do Sistema:
	1. O sistema apresenta o conceito de caminho crítico e descreve de maneira sucinta o objetivo desta fase e como funciona a sua pontuação.
2. O jogador seleciona a opção “avançar”.	
	3. O sistema apresenta a mesma tela do caso de uso <i>Jogar fase caminho crítico</i> , contendo o sequenciamento das atividades da fase anterior e os campos para cálculo do caminho crítico.
4. O jogador seleciona a opção “avançar”.	
	6. O sistema preenche os campos PDI, UDI, PDT e UDT para todas as atividades e explica como esses cálculos foram realizados. O sistema mostra qual é o caminho crítico.



- *qtdVidas*: quantidades de vidas inicial que o jogador recebe quando começa um novo jogo;
- *pontosIniciante*: pontuação mínima para que um jogador de nível iniciante possa passar de fase;
- *pontosAvancado*: pontuação mínima para que um jogador de nível avançado possa passar de fase;
- *pontosFase*: pontuação máxima que um jogador pode fazer em uma fase do cenário.
- *resumo*: uma breve explicação do assunto do cenário.

A classe *Fase* possui os seguintes atributos:

- *nome*: nome da fase;
- *explicação*: explicação de como jogar a fase, para ajudar o entendimento pelo jogador.

A classe *FaseEscopo* possui os seguintes atributos:

- *descricao*: descrição textual do escopo;
- *questoes*: lista de questões.

A classe *FaseEAP* possui os seguintes atributos:

- *arvore*: árvore que monta a EAP;
- *rotulosAdicionais*: lista de rótulos adicionais para aumentar a dificuldade do cenário.

A classe *FaseAtividade* possui os seguintes atributos:

- *arvore*: referência da árvore construída na classe *FaseEAP*, onde as atividades são adicionadas aos pacotes desta;
- *atividadesAdicionais*: lista de atividades adicionais para aumentar a dificuldade do cenário.

A classe *FaseSequenciamento* possui os seguintes atributos:

- *listaAtividades*: lista de atividades para o sequenciamento.

A classe *FaseCaminhoCritico* possui os seguintes atributos:

- *listaAtividades*: lista de atividades para montar o caminho crítico.

A classe *Questao* possui os seguintes atributos:

- *nome*: nome da questão;

A classe *Alternativa* possui os seguintes atributos:

- *texto*: texto da alternativa;
- *correta*: diz qual é a alternativa correta.

A classe *Arvore* possui os seguintes atributos:

- *nodos*: nodos da árvore

A classe *Nodo* possui os seguintes atributos:

- *nome*: nome do nodo (rótulo);
- *listaAtividades*: lista de atividades caso o nodo seja uma folha (pacote).

A classe *Atividade* possui os seguintes atributos:

- *nome*: nome da atividade;
- *duracao*: duração que a atividade tem;
- *associacoes*: em que outras atividades ela está ligando.

## **5. Recursos Necessários**

### **5.1 Recursos Necessários**

Para a elaboração e desenvolvimento da ferramenta proposta neste Trabalho de Conclusão, serão necessários os seguintes recursos:

1. Computadores Pentium IV 2.0 GHz ou superior com resolução 1024x768;
2. Sistema operacional Microsoft Windows XP;
3. JSDK 1.5;
4. Eclipse Platform 3.2 ou superior;
5. Jude Community;
6. Microsoft Office 2003;
7. Microsoft Project.

## 6. Conclusões

Com o passar do tempo os projetos de software cresceram muito em complexidade e a sua taxa de sucesso diminuiu drasticamente. O gerenciamento de projetos é muito importante para garantir que os projetos atinjam os seus objetivos dentro do prazo, utilizando os recursos disponíveis e com a qualidade desejada.

O PMI definiu, através do PMBOK, diversas técnicas e padrões para a execução do projeto, facilitando o gerenciamento de projetos. Foram criados 44 processos que estão divididos em nove áreas de conhecimento em gerenciamento de projetos que permitem que o início, o planejamento, execução, o controle e o encerramento do projeto sejam executados de maneira eficaz.

Este trabalho de conclusão foi desenvolvido devido à percepção de que por um lado existe uma demanda cada vez maior pelo aprendizado na área de gerenciamento de projetos, mas por outro lado a grande maioria dos treinamentos existentes tem como público alvo profissionais experientes que buscam certificações, como a de *Project Management Professional* (PMP), ou são treinamentos puramente teóricos.

Existem disciplinas em diversos cursos de graduação que abordam o gerenciamento de projetos. Estas ensinam gerência de projetos para um público que geralmente não possui nenhuma experiência na área, e devido à falta de livros e softwares didáticos, esta tarefa torna-se mais difícil e cansativa.

Por este motivo, neste trabalho de conclusão foi desenvolvida a ferramenta *Planager*, que busca facilitar o aprendizado de gerência de projetos, principalmente em disciplinas de graduação, por alunos que possuem pouca ou nenhuma experiência na área. Esta ferramenta foi desenvolvida na forma de um jogo para que os usuários possam aprender de uma maneira mais prática e interativa do que as encontradas na literatura.

Como a ferramenta possui um módulo de criação de cenários, além do jogo propriamente dito, os professores podem encorajar os alunos mais experientes a participar do jogo no papel de administradores criando novos cenários, pois isto ajuda a aprimorar os mesmos conceitos de gerenciamento de projetos utilizados durante o jogo, mas de forma mais aprofundada.

Mas é importante salientar que a criação de novos cenários é um processo bastante desafiador. O usuário deve possuir bons conhecimentos no planejamento de projetos e deve conseguir definir muito bem o escopo do cenário. Caso a descrição do escopo não for bem escrita, as fases do jogo poderão possuir mais de uma resposta e então o sistema de pontuação

não conseguirá pontuar o jogador de forma justa. O objetivo é tornar a ferramenta mais interativa tanto para o aluno quanto para o professor.

Foi limitado o escopo da ferramenta devido ao cronograma pouco flexível deste trabalho de conclusão. A ferramenta não pretende simular todos os processos do gerenciamento de projetos. Ela tem como objetivo auxiliar o usuário a praticar alguns dos processos de planejamento das áreas de gerência de tempo e gerência de escopo, mas é facilmente expansível para que outros processos de outras áreas de conhecimento em gerenciamento de projetos também possam ser incluídos através de novos módulos.

A principal dificuldade encontrada para projetar e desenvolver este jogo foi a subjetividade existente nos processos de gerenciamento de projetos. É muito difícil desenvolver um software que seja capaz de avaliar atividades tão subjetivas como a criação de uma EAP ou a definição das atividades que estão no escopo de um projeto. Por este motivo, é importante ressaltar a necessidade de descrever bem o escopo do projeto e o resto do cenário.

Para viabilizar o jogo foi necessário escolher quais são os processos de gerenciamento de projetos que poderiam ser avaliados automaticamente e quais são as adaptações que deveriam ser feitas para desenvolvê-los. A fase de caminho crítico é a única fase do jogo que não possui subjetividade, pois o cálculo do caminho crítico é um processo matemático. As principais adaptações que foram realizadas são:

1. Estrutura da EAP pré-definida (jogador só pode associar elementos).
2. Atividades pré-definidas (jogador só pode associá-las aos pacotes de trabalho).
3. Utilização de dependências de término para início no seqüenciamento de atividades.
4. Utilização de questões de múltipla escolha para avaliar o escopo do projeto.

O refinamento da modelagem e da documentação, assim como o desenvolvimento de todos os casos de uso descritos no trabalho de conclusão I foi realizado durante o trabalho de conclusão II.

Devido a limitações de tempo para o desenvolvimento do jogo, não foi possível desenvolver a interface gráfica de acordo com os padrões utilizados em jogos convencionais, mas existem planos para continuar o desenvolvimento da ferramenta, assim utilizando-a como apoio nas disciplinas de gerência de projetos de software da FACIN/PUCRS. As principais alterações para uma nova versão são:

1. Aperfeiçoar a interface gráfica do jogo;
2. Utilizar o retorno dos alunos de graduação para complementar as fases;

3. Cadastrar diversos cenários na ferramenta;
4. Criar novas fases que utilizem os outros processos do PMBOK.

Em relação ao uso da ferramenta, é importante destacar que num primeiro momento, a ferramenta foi desenvolvida pensando em ser utilizada em disciplinas com foco em gerência de projetos de software. Entende-se que, por apresentar os conceitos de forma genérica, e permitir a criação de qualquer cenário, é possível utilizá-la com foco na gerência de projetos como um todo e não apenas em projetos de desenvolvimento de software.

Por fim, e não menos importante, o jogo desenvolvido neste trabalho foi apresentado para alunos da disciplina de Gerência de Projetos de Software do curso de Bacharelado em Sistemas de Informação da FACIN e eles mostraram bastante interesse na utilização desta ferramenta para praticar os conceitos aprendidos em aula. Além disso, o jogo foi apresentado para três profissionais da área de Gerência de Projetos, que da mesma forma julgaram ser um facilitador para o ensino de gerência de projetos. Isto serviu como estímulo para a continuação do desenvolvimento após a entrega do trabalho.

Em relação a publicação de resultados, este trabalho participa do ciclo 2006 do PBQP Software (Programa Brasileiro de Qualidade e Produtividade em Software), concorrendo ao prêmio Dorgival Brandão Junior de Qualidade e Produtividade. Adicionalmente, pretende-se escrever e submeter artigos científicos para eventos nacionais e internacionais, divulgando os resultados alcançados e contribuindo com a comunidade acadêmica e empresarial.

## 7. Referências Bibliográficas

[BETZ, 1995] BETZ, J. A, Computer Games: Increases learning in an interactive multidisciplinary environment - Journal of Educational Technology Systems.

[GRAMIGMA, 1994] GRAMIGMA, M.R.M.. *Jogos de empresa*. São Paulo: Makron Books, 1994.

[NEAL, 1990] NEAL, L. Implications of computer games for system design – Proceedings of INTERACT '90 Elsevier, North Holland.

[PMI, 2004] Um Guia do Conjunto de Conhecimentos em Gerenciamento de Projetos - PMBOK. PMI, 2004.

[SCHWALBE, 2002] SCHWALBE, K. *Information Technology. Project Management*, Cambridge, MA: Course Technology, 2002.

[STANDISH GROUP, 1995] CHAOS report, 586 Olde Kings Highway. Dennis, MA 02638, USA, 1995.