PUC Minas
A B A K Ó S
Instituto de Ciências Exatas e Informática

Uma proposta de modelagem para o cálculo de reservas de contingência e gerencial em projetos de software usando programação linear inteira estocástica*

A proposal for modeling the calculus of contingency and management reserves in software projects using stochastic linear integer programming

Valésio Pinto[1]
Silvana Bocanegra[2]
Jones Albuquerque[3]

Resumo

Em orçamentos de projetos de software são calculados dois tipos de reservas, de contingência e gerencial, para mitigar riscos previstos e imprevistos, respectivamente. Contudo, a literatura mostra que estes valores são estimados apenas baseados na experiência do gerente em projetos anteriores. Assim, com pouco rigor formal e reais perspectivas de acerto quanto a estimativas em projetos que possuem domínios de aplicação distintos dos anteriores executados pela equipe. Este trabalho tem como principal objetivo utilizar um modelo de programação linear inteira estocástica para gerar formalmente cenários de projeto que possibilitem ao gerente estimar tais reservas de forma numérica e mais rigorosa. Como estudo de caso, foram utilizados dados reais de uma fábrica de software do Porto Digital situado na região metropolitana de Recife-PE em cenários de projetos. O modelo matemático foi implementado e solucionado com o auxílio da ferramenta AIMMS (*Advanced Interactive Multidimensional Modeling System*). Foram utilizados dados de um projeto que já havia sido executado. Os cenários gerados permitiriam não só uma melhor alocação dos recursos reduzindo os custos finais do projeto como possibilitariam calcular estimativas para as reservas de contingência e gerencial de forma mais precisa.

Palavras-chave: Modelagem matemática. Otimização matemática. Programação linear. Riscos em projetos de software. Riscos de reserva contingencial.

*Artigo convidado.

[1]Centro de Estudos Avançados do Recife, Recife - PE, Brasil.
[2]Departamento de Estatística e Informática – Universidade Federal Rural de Pernambuco, Recife - PE, Brasil silvana@deinfo.ufrpe.br.
[3]Departamento de Estatística e Informática – Universidade Federal Rural de Pernambuco, Recife - PE, Brasil joa@deinfo.ufrpe.br.

Abstract

Two types of reserves are computed in software projects to mitigate risks: contingency reserve and management reserve. The contingency reserve is used to manage the identified risks and the management reserve is used to manage the unidentified risks. However, the literature shows that these values are estimated only based on experience of the manager in the previous projects. So, when the projects are from different fields of application which have not been previous executed by the team, the real chances of success are minimized. The main goal of this work is to use a stochastic linear integer programming model to generate formally scenarios that allow the project manager to estimate such reserves numerically and accurately. Real data from a software factory of the Porto Digital (a hub of TICs Business located in the metropolitan area of Recife-PE) has been used as a case-study for the project scenarios. The mathematical model was implemented and solved with a commercial tool called AIMMS (Advanced Interactive Multidimensional Modeling System). It has been used data from a project that had already been executed. The scenarios generated allow a better allocation of resources by reducing the final cost of the project and computing the contingency and management reserves more accurately.

Keywords: Mathematical modeling. Optimization. Linear programming. Risks in software projects. Risk Management Reserve.

1 INTRODUÇÃO

O desenvolvimento de um projeto de software envolve a estimativa de diversas variáveis como escopo, prazo, riscos, capital humano, materiais e equipamentos, prazos, custos. A cada uma das variáveis envolvidas estão associadas incertezas por incluir o fator humano como principal estimador. Especificamente, a metodologia geralmente aplicada na análise de riscos e no cálculo de reservas associadas ainda utiliza poucas ferramentas matemático-computacionais. Em geral, os riscos são identificados, atribui-se uma estimativa numérica da probabilidade e do impacto de cada risco ocorrer e assim são calculados, de forma empírica, valores no orçamento do projeto para sua contenção e prevenção.

As reservas em projetos são divididas em dois tipos: de contingência e gerencial. A reserva de contingência é o valor do orçamento destinado aos riscos que foram identificados durante a análise de riscos do projeto. A reserva gerencial é o valor do orçamento destinado aos imprevistos, riscos que não foram identificados no gerenciamento de riscos. Estas reservas são para mudanças não planejadas de escopo e custo de projeto (PROJECT MANAGEMENT INSTITUTE, 2004).

O uso de ferramentas matemático-computacionais tem sido negligenciado no cálculo de tais estimativas. Uma recente área de pesquisa que utiliza modelos e algoritmos de otimização como ferramenta de apoio na resolução de problemas de engenharia de software é denominada Engenharia de Software baseada em buscas (*Search Based Software Engineering* -SBSE) (HARMAN, 2007). Diversos problemas relacionados ao processo de desenvolvimento do software, como por exemplo, alocação de capital humano (BARRETO, 2005), planejamento de tempo e custo (ANTONIOL; PENTA; HARMAN, 2004), avaliação da qualidade (KHOSHGOFTAAR; YI; SELIYA, 2004), têm sido modelados como problemas de otimização cujas soluções utilizam-se de meta-heurísticas (BERGUESS; LEFLEY, 2001). Além disso, em (ALBUQUERQUE et al., 1999), há um modelo de programação linear inteira estocástica para determinar o particionamento de *Hardware/Software Codesign* envolvendo riscos de projeto, times de desenvolvimento e múltiplas tecnologias. O objetivo deste trabalho é utilizar o modelo proposto em (ALBUQUERQUE et al., 1999) para gerar cenários que auxiliem cálculo de estimativas de reservas em projetos de software.

Este artigo está organizado da seguinte forma. Na Seção 2 é apresentado o ferramental matemático utilizado na modelagem. A representação formal do projeto está descrita na Seção 3 e seu modelo matematico é apresentado na Seção 4. Os cenários gerados e resultados obtidos com diferentes parâmetros encontram-se na Seção 6. Finalizando, na Seção 7, são apresentadas as conclusões deste trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Programação matemática (ou otimização) é um procedimento que tem por objetivo determinar soluções eficientes de problemas reais utilizando modelos matemáticos. Um problema básico de otimização é obtido quando houver a necessidade de maximizar ou minimizar uma função objetivo sujeita a um conjunto de restrições (equações ou inequações) que limitam as variáveis dessa função. Quando a função objetivo e as restrições são lineares tem-se um problema de programação linear.

Em geral, a modelagem de problemas reais envolve parâmetros aos quais estão associadas incertezas, como por exemplo, prazos, custos, produtividade, preços, entre outros. Muitos modelos têm sido utilizados na literatura para tratar com incertezas, dentre eles podemos citar os modelos de programação estocástica (EICHHORN; RöMISCH, 2007), programação robusta (BEN-TAL; GHAOUI; NEMIROVSKI, 2006) e otimização *fuzzy* (LODWICK; KACPRZYK, 2010). Neste trabalho, iremos utilizar um modelo de programação inteira estocástica.

2.1 Modelos de programação linear inteira estocástica

Problemas de programação linear inteira estocástica (BIRGE, 1995) são problemas de programação linear em que as variáveis assumem valores inteiros e reais e os parâmetros são distribuídos sob alguma distribuição de probabilidade. Uma de suas formulações matemáticas pode ser dada por:

$$\begin{aligned} &\min f(x) = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \\ &\text{sujeito a: } Prob\{\sum_{j=1}^{n} a_{ij}x_{ij} \leq b_i\} \geq 1-\alpha_i, \;\; i = 1, 2, \ldots, m. \end{aligned} \tag{1}$$

Os coeficientes da função objetivo c_{ij}, os termos independentes b_i e os coeficientes das restrições a_{ij} são valores distribuídos aleatoriamente; $0 \leq \alpha_i \leq 1$ representa a incerteza de cada restrição ser satisfeita e $x_{ij} \in \{0, 1\}$ são as variáveis de decisão.

A solução deste problema depende fortemente da distribuição das variáveis aleatórias (BISWAL; BISWAL; LI, 1998). Quando estão sendo modelados problemas reais, esta solução muitas vezes não pode ser obtida por métodos tradicionais, devido ao esforço computacional requerido. Muitas técnicas de aproximação têm sido empregadas para resolver o problema (SINHA; HULSURKAR; BISWAL, 2000; GASSMANN, 1998; LAHDELMA; HOKKANEN; SALMINEN, 1998).

2.2 Aproximação por valor esperado

A aproximação pelo valor esperado é uma técnica comumente utilizada para solucionar tais problemas. Esta técnica consiste em transformar o problema em um determinístico equivalente substituindo os parâmetros probabilísticos por seus valores esperados (STOUGIE; VLERK, 1997). Considerando que a probabilidade da soma das variáveis aleatórias é equivalente à soma das probabilidades das variáveis aleatórias, o problema (1) pode ser reescrito da seguinte forma:

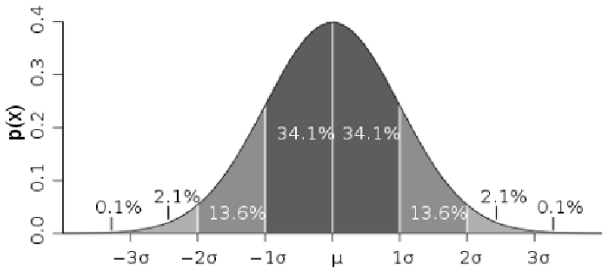
$$\begin{aligned} &\min f(x) = \sum_{i=1}^{m}\sum_{j=1}^{n} Prob\{c_{ij} \geq 1-\alpha_o\}x_{ij} \\ &\text{sujeito a: } \sum_{j=1}^{n} Prob\{a_{ij} \geq 1-\alpha_i\}x_{ij} \leq b_i, \;\; i = 1, 2, \ldots, m. \end{aligned} \tag{2}$$

Neste trabalho, estamos usando a função de distribuição normal. Considere a seguinte tripla (m, M, c), em que m e M são respectivamente os valores esperados mínimo e máximo de uma dada métrica e c é o grau de confiança destes valores ($c = 1$ é considerado baixo; $c = 2$, médio e $c = 3$, alto). Em geral, o grau de confiança é obtido baseado em dados históricos. Para cada tripla pode ser calculada uma probabilidade de que a métrica seja atingida, usando uma função de distribuição de probabilidade, como, por exemplo, a distribuição normal. A função de probabilidade da distribuição normal com média μ e variancia σ^2 é descrita por

$$f(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\Big[-\frac{(x-\mu)^2}{2\sigma^2}\Big], \tag{3}$$

em que $x; -\infty \leq x \leq \infty$ representa o valor esperado da métrica, $\mu = \frac{m+M}{2}; -\infty \leq \mu \leq \infty$ é o valor médio esperado e $\sigma = \frac{M-m}{c}; \sigma \geq 0$ é o desvio padrão definido pelo grau de confiança c. Na Figura 1 pode ser observado que conforme o grau de confiabilidade aumenta, maior é a probabilidade de x pertencer ao intervalo $[m, M]$.

Figura 1 – Probabilidade de x pertencer a diferentes intervalos da distribuição normal;$\sigma = 1$ (preto) $\sigma = 2$ (cinza escuro) $\sigma = 3$ (cinza claro).



Para exemplificar o funcionamento da aproximação pelo valor esperado, considere uma

restrição do problema (1) ($i = 1$; $n = 2$ e $\alpha_i = 0.5$) :

$$Prob\{a_{11}x_{11} + a_{12}x_{12} \leq b_1\} \geq 0.5\ , \tag{4}$$

onde x_{11} e x_{12} são variáveis de decisão, $a_{11} = (\mu_{a_{11}}, \sigma_{a_{11}})$ e $a_{12} = (\mu_{a_{12}}, \sigma_{a_{12}})$ são variáveis aleatórias sob alguma distribuição de probabilidade. Essa restrição pode ser aproximada por

$$\mu_{a_{11}}x_{11} + \mu_{a_{12}}x_{12} \leq b_1.$$

Estendendo a aproximação para cenários de projetos (ALBUQUERQUE, 2002), e considerando a_{11} and a_{12} variáveis independentes, a equação (4) pode ser aproximada por

$$(\mu_{a_{11}} + \mathsf{F}^{-1}(1-\alpha)\sigma_{a_{11}})x_{11} + (\mu_{a_{12}} + \mathsf{F}^{-1}(1-\alpha)\sigma_{a_{12}})x_{12} \leq b_1,$$

em que $\mathsf{F}^{-1}(1-\alpha)$ é a inversa da função de distribuição normal de uma variável aleatória.

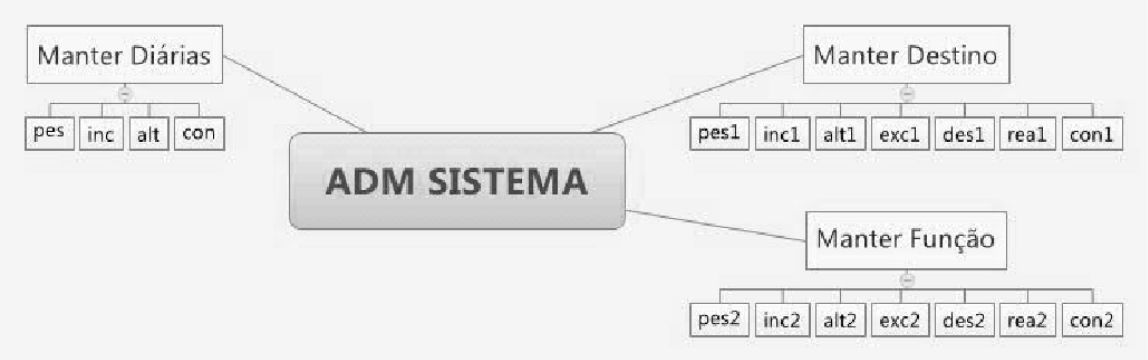
3 REPRESENTAÇÃO FORMAL DO PROJETO

Por motivo de confidencialidade, o nome da fábrica de software e o sistema em tela utilizado neste trabalho são designados genericamente como sendo fábrica de software e sistema ADM. Em alguns casos os artefatos foram mostrados parcialmente ou com pequenas alterações.

O sistema ADM é baseado no processo de arquitetura matricial que divide o ciclo de desenvolvimento em fases (que por sua vez são divididas em iterações) e fluxos. Este estudo detalha um segmento de iteração durante a fase de desenvolvimento.

A modelagem do sistema a ser desenvolvido é composta de três visões: a estática (chamada neste trabalho de Diagrama Hierárquico), a funcional e a dinâmica. A visão estática é atendida pela modelagem de dados ou objetos, a visão funcional é atendida pela modelagem das funções e fluxos de dados, e a visão dinâmica é atendida, em muitas abordagens metodológicas, pela modelagem de estado dos objetos. A modelagem de dados ou objetos descreve a estrutura de entidades ou objetos do sistema, identificando as classes, os relacionamentos entre as classes, os atributos e, em orientação a objetos, também as operações das classes (FUENTES; VALLECILLO, 2004). A modelagem funcional descreve as transformações que os dados sofrem no interior do sistema. A notação gráfica mais utilizada é o clássico diagrama de fluxo de dados.

Figura 2 – Visão hierárquica do sistema ADM.



3.1 Diagrama hierárquico

A visão hierárquica do sistema ADM pode ser visto como um conjunto de objetos composto por outros objetos, formando uma relação de dependência. Cada objeto contém métodos de acesso (operações ou interface) que executam diferentes funcionalidades previstas para esse objeto. Neste sentido, a visão hierárquica pode ser interpretada como um modelo orientado a objetos, onde as relações de dependência entre objetos são obtidos através de hierarquia.

Qualquer objeto na estrutura hierárquica pode ser subdividido em refinamentos sucessivos ou durante sua execução, o que pode requerer a atribuição de novos objetos para as mesmas equipes. Do ponto de vista formal a visão hierárquica pode ser representada por um grafo $G(M, R)$, onde M é o conjunto de vértices que representa os objetos do projeto, e R é o conjunto de arestas que representa uma composição sucessiva de um objeto. Nesta visão, cada dois objetos, os quais não estão na relação transitiva, uns dos outros, são assumidos como concorrentes (ALBUQUERQUE et al., 1999).

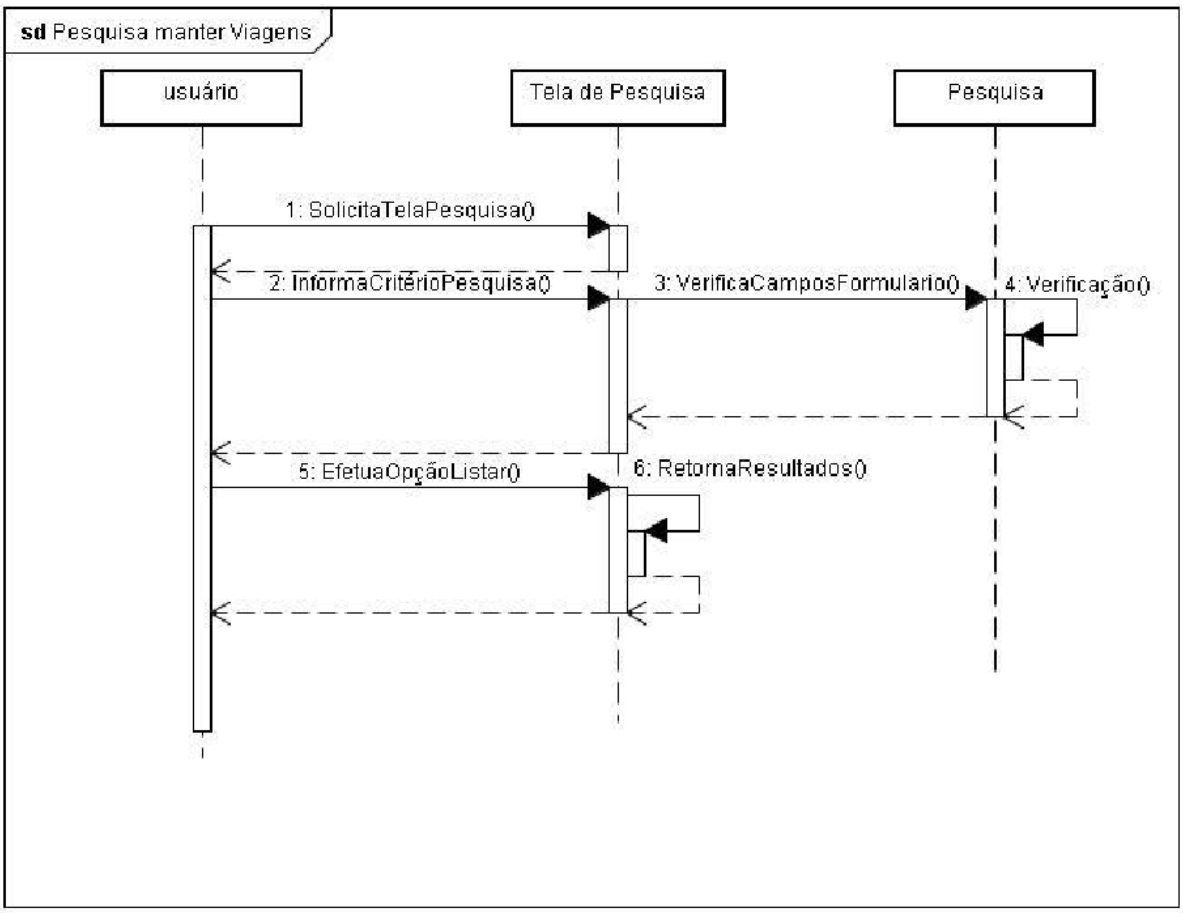
Na Figura 2 é possível visualizar os casos de uso da Figura 5 numa abstração em forma de casos de uso: Manter Diárias, Manter Destino e Manter Função. Fazendo a subdivisão dos objetos abstraem-se os métodos. Ao todo, o sistema ADM é composto dos seguintes casos de uso: *Manter Diárias* com 4 objetos e 12 métodos; *Manter Destino* com 7 objetos e 30 métodos e *Manter Função* com 7 objetos e 25 métodos. No total, tem-se 18 objetos e 67 métodos.

3.2 Diagrama sequencial

Consiste em caminhos sequenciais executados pelas interfaces de acesso aos objetos (ALBUQUERQUE et al., 1999). No sistema ADM, a ênfase está na organização estrutural dos objetos que enviam e recebem mensagens mostrando o fluxo de atividades. A Figura 3 representa a visão sequencial do método (pes) do objeto (Manter Diárias).

1. O caso de uso inicia quando o usuário seleciona ”manter classe de função” e solicita a opção;
2. O sistema apresenta tela de filtro de pesquisa;
3. O usuário informa o critério de pesquisa a ser utilizado;
4. O sistema exibe uma lista;
5. erro: Origem da referência não encontrado que satisfaça aos parâmetros de pesquisa informado;
6. O usuário seleciona uma das opções apresentadas;
7. O sistema aciona o subfluxo correspondente à seleção do usuário.

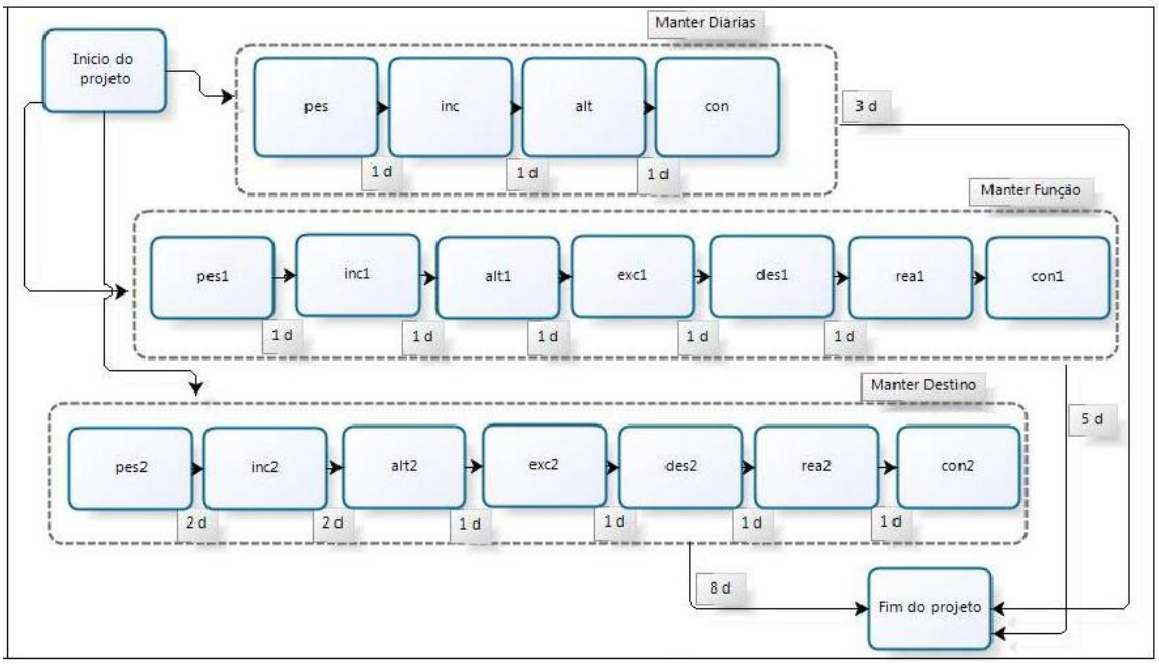
Figura 3 – Diagrama sequencial do sistema ADM para o método “pes” do objeto “Manter Diárias”.



3.3 Diagrama de desenvolvimento

Este diagrama destaca a configuração, em tempo de execução, dos componentes do sistema ADM. A Figura 4 apresenta o grafo de desenvolvimento do sistema ADM com distribuição do tempo para cada objeto.

Figura 4 – Grafo de desenvolvimento do sistema ADM.



3.3.1 Diagrama de casos de uso

O Diagrama de Casos de Uso é definido no Processo Unificado da Rational (*Rational Unified Process* - RUP) (JACOBSON; BOOCH; RUMBAUGH, 1999) e dá apoio ao desenvolvimento orientado a objetos e utiliza uma Linguagem de Modelagem Unificada (*Unified Modeling Language* - UML) para a descrição de sistemas. O RUP está fundamentado em três princípios básicos: orientação a casos de uso, arquitetura e iteração. São os casos de uso que orientam todo o processo de desenvolvimento. Com base no modelo de casos de uso, são criados os modelos de análise, projeto e implementação, que instanciam estes casos de uso. A Figura 5 representa o relacionamento de comunicação do ator com o sistema ADM.

Uma boa prática de modelagem de sistemas é agrupar operações tipo CRUD (sigla do inglês que significa: *Create*, *Retrieve*, *Update* e *Delete*) em casos de uso do tipo "Manter", que segue um padrão bem estabelecido na indústria, variando apenas as regras de negócio (CHEESMAN; DANIELS, 2000). A Figura 6 mostra o indicador de esforço da equipe no desenvolvimento dos casos de uso, ao longo do projeto. A cada ciclo de desenvolvimento, um conjunto

Figura 5 – Ator associado aos casos de uso.

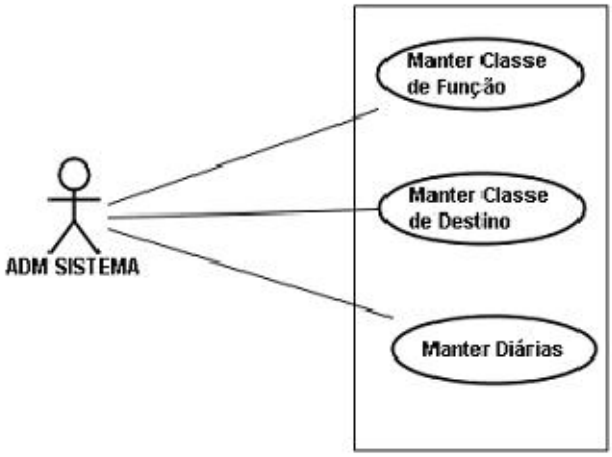


Figura 6 – Indicador de esforço da equipe.

		% Complete	Atividade	Work	Duration
107	✓	**100%**	**Casos de Uso**	**768 hrs**	**30,75 days**
108	✓	100%	Preparação de ambiente + treinamento equipe	138 hrs	5,75 days
109	✓	100%	CDU003 Manter Diárias	24 hrs	3 days
110	✓	100%	CDU001 Manter Classe de Função	40 hrs	5 days
111	✓	100%	CDU002 Manter Classe de Destino	64 hrs	8 days

tratável de casos de uso é considerado. O padrão "manter" do sistema ADM foi utilizado para documentar os requisitos de manutenção que são caracterizados por operações de Inclusão, Consulta, Alteração e Exclusão, Reativar e Desativar.

3.3.2 Especificação dos objetos e métodos

Embora exista grande variedade de representações disponíveis, o método proposto emprega um estilo informal. As informações dos casos de uso são descritas e mapeadas em forma de fluxos. A estratégia consiste em identificar o número de objetos e métodos, quantificando o número de sub-fluxos e o número de atributos dos sub-fluxos, respectivamente. A Tabela 1 ilustra esta relação com os objetos instanciados.

Tabela 1 – Relação dos objetos instanciados.

Caso de uso	Sub-Fluxos	objetos	métodos
Manter diárias	pesquisar	pes	met1
Manter diárias	incluir	inc	met2 met3 met4
Manter diárias	alterar	alt	met5 met6 met7 met8 met9 met10
Manter diárias	consultar	con	met11 met12
Manter função	pesquisar	pes1	met13
Manter função	incluir	inc1	met14 met15 met16 met17 met18 met19 met20
Manter função	alterar	alt1	met21 met22 met23 met24 met25 met26 met27
Manter função	consultar	con1	met28 met29 met30 met31 met32
Manter função	desativar	des1	met33 met34 met35 met36
Manter função	reativar	rea1	met37 met38 met39 met40
Manter função	excluir	exc1	met41 met42
Manter destino	pesquisar	pes2	met43
Manter destino	incluir	inc2	met44 met45 met46
Manter destino	alterar	alt2	met47 met48 met49 met50 met51 met52
Manter destino	consultar	con2	met53 met54 met55 met56 met57
Manter destino	desativar	des2	met58 met59 met60 met61
Manter destino	reativar	rea2	met62 met63 met64 met65
Manter destino	excluir	exc2	met66 met67

4 O MODELO MATEMÁTICO

O modelo matemático utilizado neste projeto foi proposto originalmente em (ALBUQUERQUE et al., 1999) para modelar projetos colaborativos de *Hardware/Software Codesign*. Este modelo pertence a categoria de problemas de programação linear inteira estocástica.

A notação usada na modelagem está descrita na Tabela 2. Os índices i são usados para representar os objetos, m para os métodos, j para as equipes, p para os caminhos nos grafos e k para semanas. As restrições do modelo são apresentadas em duas classes: probabilísticas e determinísticas.

- **Restrições Probabilísticas**: Cada parâmetro probabilístico c_{ij}, d_{mj}, t_{ij} e λ_{ij} é representado por uma tripla (m, M, c), onde "m" é o valor mínimo; "M", o valor máximo e "c", o grau de confiança.

  1. Custo: O custo total do projeto, definido como a soma dos custos de todos os objetos, não deve exceder o custo máximo desejado, com um determinado grau de tolerância (α_{custo}),

$$Prob\left\{\sum_{\substack{i \in Objetos,\\ j \in Equipes}} c_{ij}x_{ij} \leq C\right\} \geq 1 - \alpha_{custo}. \tag{5}$$

  2. Tempo de execução: para cada caminho (p) no grafo SG, a soma dos tempos de execução estimados (d_{mj}) dos métodos m implementados pelas equipes j deve ser menor que o tempo máximo desejado (DI_p) com uma dada probabilidade de certeza ($1 - \alpha_{caminho}$), ou seja:

$$\forall P \in CaminhosE, j \in Equipes,$$
$$Prob\left\{\sum_{(p,i,m)\in P} d_{mj}x_{ij} \leq DI_p\right\} \geq 1 - \alpha_{caminho}. \tag{6}$$

  3. Tempo de desenvolvimento: para cada caminho (p) no grafo DG, a soma dos tempos de desenvolvimento estimados (t_{ij}) para as equipes j implementarem os objetos i neste caminho, deve ser menor que o tempo máximo desejado de desenvolvimento (DD_p), com um determinado grau de confiança (α_{des}), ou seja:

$$\forall P \in CaminhosD, j \in Equipes,$$
$$Prob\left\{\sum_{(p,i_1,i_2)\in P} t_{i_1 j}x_{i_1 j} \leq DD_p\right\} \geq 1 - \alpha_{des}. \tag{7}$$

Tabela 2 – Símbolos, parâmetros e variáveis utilizadas na formulação do problema.

Símbolos	**Descrição**
$Objetos$	conjunto de objetos.
$Metodos_i$	conjunto de métodos pertencente a cada objeto i.
$Equipes$	conjunto de equipes de desenvolvimento.
$Semanas$	conjunto ordenado de semanas (ou qualquer outra métrica para tempo).
$CaminhosD$	conjunto ordenado de dependências de desenvolvimento (objetos) definidos em cada caminho "objeto inicial - objeto final" no grafo de desenvolvimento (denotado DG).
$CaminhosE$	conjunto ordenado de dependências de execução (métodos) definidos em cada caminho de "metódo inicial - método final" no grafo sequencial (denotado SG).
C	custo desejado para o projeto.
DI_p	tempo de execução desejado para o caminho p no grafo SG.
DD_d	tempo de desenvolvimento desejado para o caminho d no grafo DG.
Λ_j	carga máxima da equipe j.
Parâmetros	**Descrição**
c_{ij}	custo estimado para a equipe j implementar o objeto i (US$, area, KLOC, ...).
d_{mj}	tempo de execução estimado do método m implementado pela equipe j ($ms, \mu s, \ldots$).
t_{ij}	tempo de desenvolvimento estimado para equipe j implementar o objeto i ($horas, semanas, \ldots$).
λ_{ij}	carga estimada do time j para implementar o objeto.
Variáveis	**Descrição**
x_{ij}	variável binária que assume o valor 1 se o objeto i é implementado pela equipe j.
γ_{ijk}	variável binária que assume o valor 1 se na semana k, o time j inicia a implementação do objeto i.
Γ_{ijk}	variável binária que assume o valor 1 se na semana $k-1$, o time j termina a implementação do objeto i.
a_{ijk}	variável binária que assume o valor 1 se na semana k, a equipe j está implementando o objeto i.

4. Carga de trabalho da equipe: a carga de trabalho da equipe j é a soma das cargas (λ_{ij}) associadas com todas os objetos i implementados por esta equipe. Esta carga não pode exceder a carga máxima da equipe (Λ) com um grau de tolerância definido (α_{carga})

$$\forall j \in Equipes, k \in Semanas,$$
$$Prob\left\{\sum_{i \in Objetos} \lambda_{ij} a_{ijk} \leq \Lambda_j\right\} \geq 1 - \alpha_{carga}. \quad (8)$$

em que,

$$a_{ijk} = \begin{cases} \gamma_{ijk} - \Gamma_{ijk} & \text{se } k = 1 \\ a_{ij(k-1)} + \gamma_{ijk} - \Gamma_{ijk} & \text{se } k > 1. \end{cases}$$

Na primeira semana $(k = 1)$ cada equipe j pode iniciar ou não implementação de um objeto i. Nas semanas seguintes, a equipe j pode iniciar a implementação de um objeto i ($\gamma_{ijk} = 1$) ou já estar implementando um objeto ($a_{ij(k-1)} = 1$) e ainda não ter terminado ($\Gamma_{ijk} = 0$).

5. Prazo de entrega de objetos: a semana de entrega de cada objeto i implementado pela equipe j (Γ_{ijk}) deve ser maior que o tempo estimado para desenvolvimento (t_{ij}) adicionado à semana inicial (γ_{ijk}) com uma dada probabilidade de certeza (α_{ass}):

$$\forall i \in Objetos, j \in Equipes,$$
$$Prob\left\{ \sum_{k \in Semanas} k\Gamma_{ijk} \geq t_{ij}x_{ij} + \sum_{k \in Semanas} k\gamma_{ijk} \right\} \geq 1 - \alpha_{ass}.$$

- **Restrições Determinísticas**:

1. Cada objeto i deve ser implementado somente por uma equipe j:

$$\forall i \in Objetos, \sum_{j \in Equipes} x_{ij} = 1$$

2. Tempo único para inicio e fim de atividades: Cada objeto i implementado pela equipe j começa em apenas uma semana inicial (γ_{ijk}) e termina em apenas uma semana final (Γ_{ijk}):

$$\forall i \in Objetos, j \in Equipes \ ,$$
$$\sum_{k \in Semanas} \gamma_{ijk} = x_{ij};$$
$$\sum_{k \in Semanas} \Gamma_{ijk} = x_{ij}.$$

3. Desenvolvimento sequencial: deve ser respeitada o ordem de desenvolvimento dos objetos. Considere i_1 e i_2 dois objetos que devem ser implementados sequencialmente. Para cada aresta (p, i_1, i_2) de um caminho $p \in CaminhosD$ no grafo DG, o desenvolvimento do objeto i_2 poderá começar ao menos na semana seguinte ao objeto i_1 ter sido implementado.

$$\forall (p, i_1, i_2) \in CaminhosD,$$
$$\sum_{\substack{j \in Equipes, \\ k \in semanas}} k\gamma_{i_2jk} - \sum_{\substack{j \in Equipes, \\ k \in semanas}} k\Gamma_{i_1jk} \geq 1$$

Nesta formulação foram propostas as quatro funções objetivo descritas a seguir. Cada

uma delas prioriza um determinado fator para auxiliar o gerente de projetos na alocação das equipes e no uso das tecnologias.

- Minimizar o custo de desenvolvimento (equação (5));
- Minimizar o tempo de execução (equação (6));
- Minimizar o tempo de desenvolvimento (equação (7)) ;
- Minimizar a carga de trabalho das equipes (equação (8)).

Quando o gerente seleciona o objetivo, a respectiva restrição será convertida na função objetivo. A solução do problema irá retornar a alocação "objeto: equipes x tecnologias", que satisfaz as restrições e atinge o objetivo selecionado.

5 METODOLOGIA DE SOLUÇÃO

As informações necessárias para construção do modelo foram obtidas nas diversas entrevistas realizadas com o gerente de projeto e com as equipes de desenvolvimento do sistema ADM. Quando as informações não puderam ser obtidas, usamos dados hipotéticos sempre seguindo uma distribuição gaussiana simples em torno da média. O sistema ADM é composto por 18 objetos (ver grafo de desenvolvimento apresentado na Figura 4) que podem ser implementados por qualquer uma das três equipes de desenvolvimento. Na Tabela 3, é apresentado um exemplo das estimativas de tempo de desenvolvimento feitas pelas equipes para implementar cada objeto do sistema. As colunas $Equipe_c$ representam o intervalo de tempo (m,M) estimado pela equipe. A essas estimativas estão associados graus de confiança: $c = 1$ representa pouca confiabilidade (equipe com pouco experiência: maturidade baixa); $c = 2$ representa confiabilidade média e $c = 3$ representa alta confiabilidade (equipe experiente: maturidade elevada).

O modelo matemático foi implementado na linguagem de modelagem matemática da ferramenta de otimização AIMMS (GETTING...,). Esta ferramenta pode ser obtida gratuitamente para propósitos acadêmicos e já vem integrada com diversos pacotes de otimização (CPLEX, MINOS, XPRESS, entre outros). Além disso, a ferramenta também está integrada com uma biblioteca que possibilita a construção de interfaces amigáveis para usuários com pouca habilidade técnica.

Utilizamos a aproximação pelo valor esperado, descrita na Seção 2.2, para transformar o problema probabilístico em um determinístico equivalente. Diversos experimentos foram realizados para avaliar o impacto, no projeto, da variação de parâmetros como custo, tempo de desenvolvimento, carga de trabalho das equipes e, assim, analisar cenários para se estimar as reservas de Contingência e Gerencial a partir desta análise. As Figuras 7, 8 e 9 apresentam respectivamente exemplos das telas do AIMMS obtidas durante a modelagem, a inserção

Tabela 3 – Exemplo de estimativas de tempo de desenvolvimento (em horas) dadas por 3 equipes ($Equipe_1$ - imatura, $Equipe_2$ - maturidade média, $Equipe_3$ madura) para implementar os objetos do sistema.

Objetos	$Equipe_1$	$Equipe_2$	$Equipe_3$
pes	[2.0,4.0]	[2.5,3.5]	[0.75,1.25]
inc	[2.0,4.0]	[3.5,4.5]	[1.5,2.5]
alt	[2.0,4.0]	[1.5,2.5]	[1.25,2.75]
con	[2.0,4.0]	[1.5,2.5]	[0.75,1.25]
pes1	[1.0,3.0]	[2.5,3.5]	[1.25,2.75]
inc1	[0.5,1.5]	[0.5,1.5]	[0.75,1.25]
alt1	[2.0,4.0]	[1.5,2.5]	[1.25,2.75]
exec1	[3.0,5.0]	[3.0,5.0]	[2.5 3.5]
des1	[1.0,3.0]	[2.5,3.5]	[1.25,2.75]
rea1	[2.0,4.0]	[2.0,4.0]	[1.25,2.75]
con1	[1.0,3.0]	[1.5,2.5]	[0.75,1.25]
pes2	[3.0,5.0]	[1.5,2.5]	[2.5 3.5]
inc2	[2.0,4.0]	[1.5,2.5]	[0.75,1.25]
alt2	[2.0,4.0]	[3.0,5.0]	[1.25,2.75]
exec2	[3.0,5.0]	[2.0,4.0]	[2.5 3.5]
des2	[2.0,4.0]	[3.0,5.0]	[3.5 4.5]
rea2	[1.0,3.0]	[0.5,1.5]	[0.75,1.25]
con2	[3.0,5.0]	[1.5,2.5]	[0.75,1.25]

dos parâmetros e solução de um cenário (mimizando o tempo de desenvolvimento) do sistema ADM.

Dentre os testes realizados, foram selecionados os parâmetros e funções objetivo que tiveram maior influência no custo do projeto. Com a variação do conjunto de dados selecionados foram gerados cenários para auxiliar o gerente na composição dos custos e no cálculo da estimativa de reservas.

Figura 7 – Modelagem do sistema ADM no AIMMS.

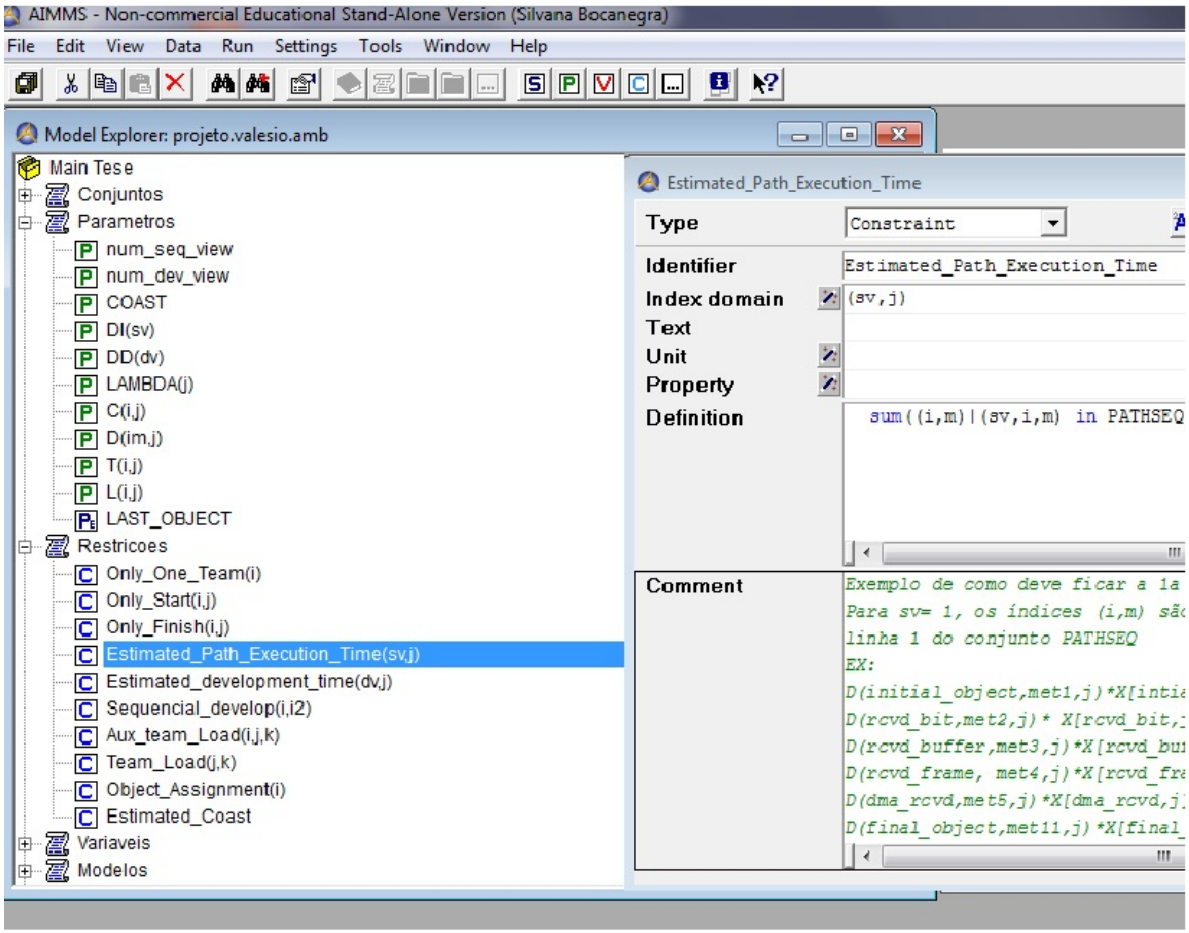
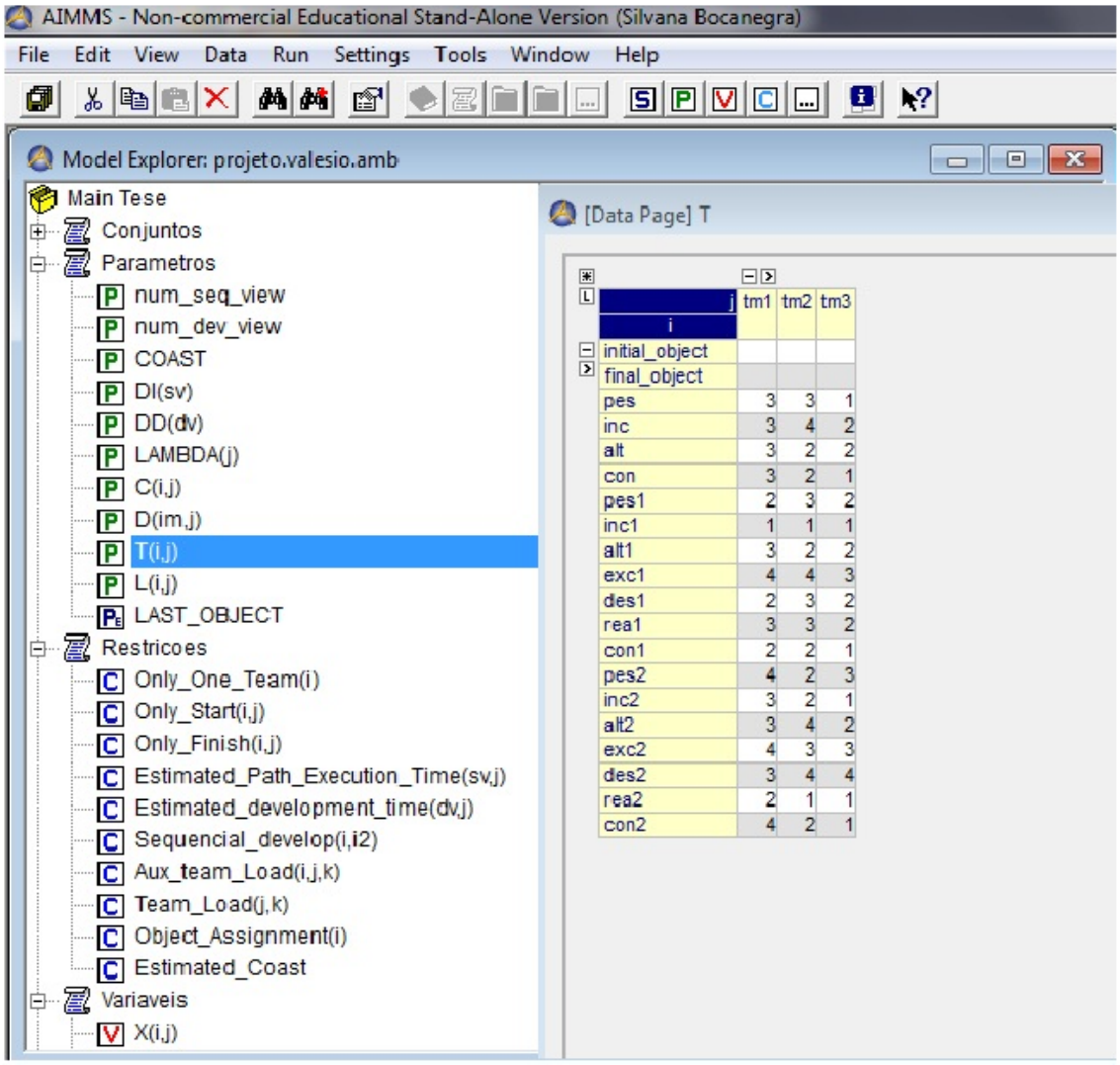
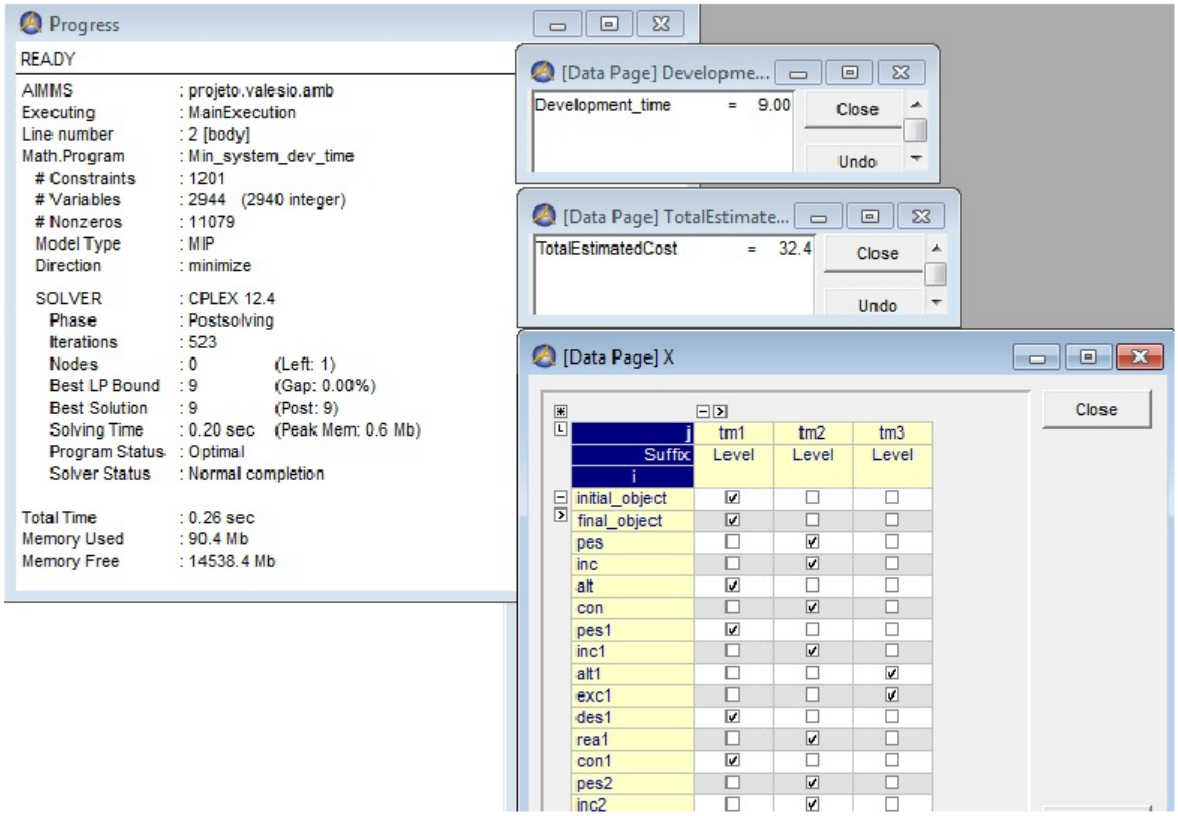


Figura 8 – Dados referentes ao tempo de desenvolvimento.



i \ j	tm1	tm2	tm3
initial_object			
final_object			
pes	3	3	1
inc	3	4	2
alt	3	2	2
con	3	2	1
pes1	2	3	2
inc1	1	1	1
alt1	3	2	2
exc1	4	4	3
des1	2	3	2
rea1	3	3	2
con1	2	2	1
pes2	4	2	3
inc2	3	2	1
alt2	3	4	2
exc2	4	3	3
des2	3	4	4
rea2	2	1	1
con2	4	2	1

Figura 9 – Solução obtida minimizando tempo de desenvolvimento.



6 GERAÇÃO E ANÁLISE DE CENÁRIOS PARA OBTENÇÃO DAS RESERVAS DE CONTINGÊNCIA E GERENCIAL

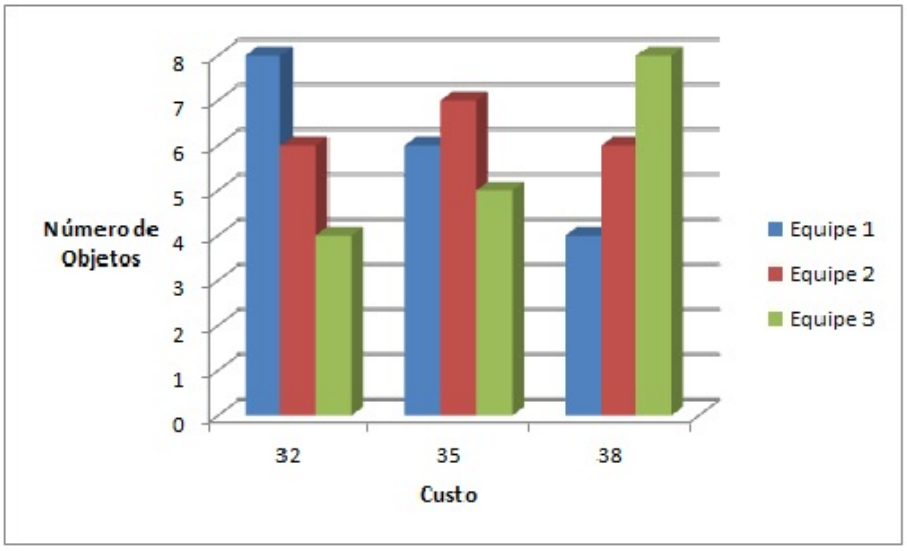
Para avaliar o uso do modelo no problema de estimativas de reservas de contingência e gerencial, utilizaram-se, como objetivo, neste estudo, as funções que minimizam o custo, o tempo de execução e o tempo de desenvolvimento. Nos três cenários, apresentados aqui, foi utilizado o caso médio de risco: $(1 - \alpha) = 0.7$ e assumiu-se que em tempo infinito todo os recursos são esgotados, sempre.

6.1 Minimização de tempo de desenvolvimento variando o custo do projeto

A Figura 10 representa o caso em que o tempo de desenvolvimento foi minimizado considerando diferentes custos. Para cada custo (que serão considerados cenários), foi gerado o particionamento da equipe buscando o menor tempo de desenvolvimento. Para este cenário, observou-se que se o custo associado ao projeto for menor que 32, não há soluções viáveis dentro do prazo estabelecido. Podemos observar, pela figura, que o aumento do custo (que pode ser interpretado como relaxação no orçamento do projeto) proporciona que equipes mais experientes sejam alocadas para desenvolver uma maior quantidade de objetos do sistema. Assim, observa-se na alocação apresentada na Figura 10, que no cenário $Custo = 32$, a equipe 3 é alocada pelo modelo a apenas 4 objetos para implementar, a equipe 2, a 6 objetos e a equipe 1 é alocada a 8 objetos. Já no cenário $Custo = 38$, observa-se que a equipe 3 é alocada a 8 objetos para implementar, a equipe 2, a 6 objetos e a equipe 1, a apenas 4 objetos. Isto porque como as equipes são mais experientes, o tempo de desenvolvimento é menor, porém normalmente equipes mais experientes são mais caras. Também observou-se que para custos maiores que 38 não há variação na alocação das equipes. Assim, observa-se que há uma faixa de solução viável e variável entre 32 e 38 unidades de custo (utilizadas aqui, por exemplo, como dezenas de milhares de reais) com risco associado de 70% de certeza.

Para o cálculo da reserva de contingência, o gerente utiliza desta faixa variável, entre 32 e 38 unidades de custo para executar o projeto. Como o risco de certeza adotado nestes cenários é de 70%, executa-se o modelo com risco de certeza ($(1 - \alpha) = 1$), ou seja 100% de certeza de sucesso. De posse dos resultados obtidos para risco $(1 - \alpha) = 1$, a margem entre o cenário escolhido com certeza 70% e o valor mínimo calculado com risco de certeza de 100% é a reserva de contingência. Assim, isto fornece a margem de custo correta para a garantia da completa execução do projeto dentro de todas as restrições previstas para o sistema. Observa-se que se o cenário para risco $(1 - \alpha) = 1$ não apresentar solução viável, adota-se risco de certeza menor até se obter solução viável. Este é um processo iterativo e exaustivo nas margens de risco definidas para a análise. Observa-se, também, que quando se adota risco de certeza de 100% no

Figura 10 – Minimizando tempo de desenvolvimento com variação no custo.



cálculo dos cenários de um projeto não se faz necessário ter reserva de contingência, pois todos os riscos previstos foram contemplados.

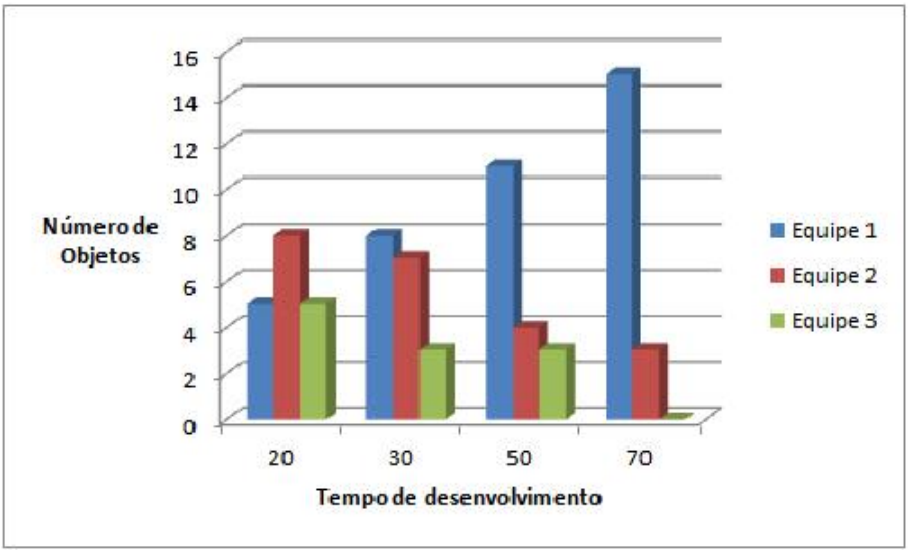
Para o cálculo da reserva gerencial, tomam-se os cenários para custos acima do limiar ótimo obtido. No exemplo, o limiar é o cenário de valor 38 (para risco de certeza 70%). Os cenários acima de 38 fornecem folga de projeto por definição, pois não oneram a alocação da equipe, apenas de recursos. Assim, a reserva gerencial é dada pela diferença entre este limiar e o custo mínimo definido para o projeto, desde que maior que 32 (neste exemplo). Para se ter uma reserva gerencial mais abrangente considera-se esta mesma análise para os cenários obtidos com risco de certeza $(1 - \alpha) = 1$. Observa-se, ainda, que para se ter 100% de certeza, considera-se sempre recursos infinitos disponíveis.

6.2 Minimização do tempo de execução variando o tempo de desenvolvimento

Por tempo de execução entende-se a latência entre a solicitação do serviço e o tempo de sistema de resposta. A Figura 11 representa o caso em que se minimiza o tempo de execução, fixa-se o custo em 35 e considera-se diferentes tempos de desenvolvimento. Para cada cenário em que se varia o tempo de desenvolvimento, obtém-se uma partição da equipe. Observa-se que quando o tempo de desenvolvimento foi menor que 20, não houve solução viável, pois há limitação da carga de trabalho das equipes. Conforme relaxa-se a restrição do tempo de desenvolvimento, aumenta-se o número do objetos alocados às equipes menos experientes. O que é natural, pois se há mais tempo, a implementação pode ser feita por equipes menos experientes e a um custo menor. Para tempos de desevolvimento maiores que 70 não há variação na alocação das equipes, indicando um limiar ótimo de alocação.

Assim, observa-se que as reservas de contingência e gerencial também podem ser cal-

Figura 11 – Minimizando tempo de execução, fixando custos e variando tempo de desenvolvimento.



culadas para este cenário do mesmo modo que para o cenário descrito na Seção 6.1.

6.3 Minimização do custo variando o tempo de desenvolvimento

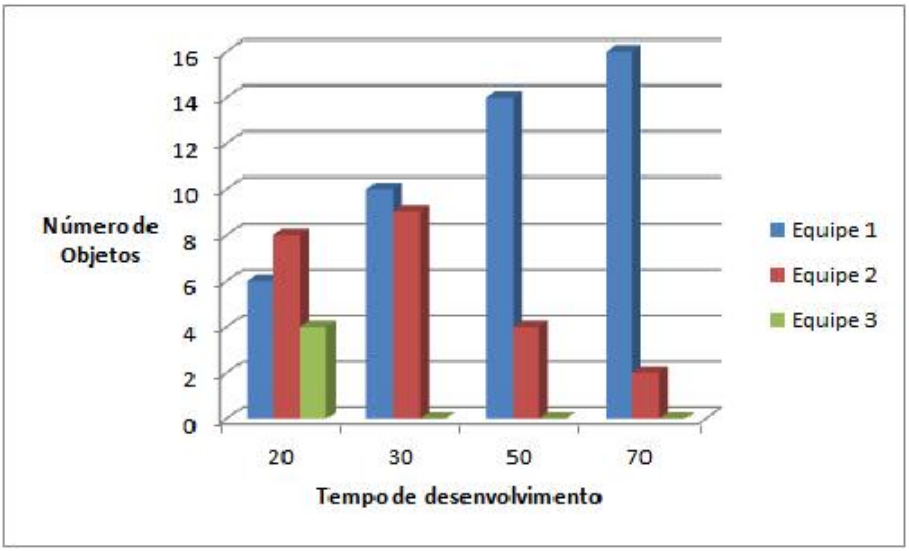
Na Figura 12 está ilustrado o caso em que o objetivo é a minimização dos custos. Também foram considerados diferentes cenários, que representam variações no tempo de desenvolvimento. Quando o custo é priorizado, as equipes mais experientes (que são mais caras) participam da alocação somente quando o limite de tempo de execução é uma restrição. Nos cenários testados, observados na Figura 12, a $equipe_3$, que é a mais experiente, só é alocada a objetos quando o tempo de desenvolvimento estiver limitado em 20. Quando houve folga no tempo de desenvolvimento, equipes mais imaturas foram priorizadas na alocação, pois equipes mais imaturas tendem a desenvolver sistemas em mais tempo.

Assim, observa-se também neste cenário que as reservas de contingência e gerencial podem ser calculadas para este cenário do mesmo modo que para o cenário descrito na Seção 6.1.

7 Conclusões

O modelo apresenta uma proposta para o cálculo de reservas de contingência e gerencial de forma rigorosa baseado em resolução de problemas de programação linear estocástica com aproximação determinística. Isso possibilita a resolução do problema por ferramentas disponíveis no mercado, algumas delas livres de custo dependendo do tamanho do problema, como ilustrado no exemplo apresentado. Além disso, como ilustrado nos exemplos de cenários ana-

Figura 12 – Minimizando custo com variação no tempo de desenvolvimento.



lisados, este cálculo fornece ao gerente do projeto diversos cenários de execução do projeto tornando a resolução de problemas do dia a dia do projeto menos caótica e mais determinística.

Como restrições, pode-se citar a geração exaustiva de cenários para se encontrar os cenários de risco $(1 - \alpha) = 1$, ou menor, para o cálculo das reservas, pois pode-se não obter solução viável para instâncias próximas de 1. Inicialmente, pode-se apresentar como um fator impeditivo se for durante a execução do projeto ou quando realmente o projeto não apresenta solução viável para nenhum dos cenários. Por outro lado, observa-se que esta fase de planejamento das reservas normalmente é realizada nas iterações iniciais de um projeto e, portanto, tem-se tempo para experimentar exaustivamente os cenários até obter-se uma solução viável.

8 AGRADECIMENTOS

Este trabalho foi (parcialmente) suportado pelo Instituto Nacional de Ciência e Tecnologia para Engenharia de Software - INES, financiado pelo CNPq e FACEPE, processos $573964/2008 - 4$ e $APQ - 1037 - 1.03/08$, respectivamente.

REFERÊNCIAS

ALBUQUERQUE, Jones et al. System-level partitioning with uncertainty. In: **Proceedings of the seventh international workshop on Hardware/software codesign**. New York, NY, USA: ACM, 1999. (CODES '99), p. 198–202. ISBN 1-58113-132-1. Disponível em: <http://doi.acm.org/10.1145/301177.301531>.

ALBUQUERQUE, J. O. **A System-level design model for Hardware/Software codesign.** 2002. Tese (Doutorado) — Universidade Federal de Minas Gerais, Departamento de Ciência da Computação, Belo Horizonte.

ANTONIOL, G.; PENTA, M. Di; HARMAN, M. A robust search-based approach to project management in the presence of abandonment, rework, error and uncertainty. In: TENTH INTERNATIONAL SYMPOSIUM ON (SOFTWARE METRICS, 2004). **Proceedings of the...** [S.l.], 2004. p. 172 – 183. ISSN 1530-1435.

BARRETO, Ahilton Silva. **Alocação ótima de recursos humanos a projetos de software**. 2005. Dissertação (Mestrado) — Universidade Federal do Rio de Janeiro, Instituto Alberto Luiz Coimbra de Pós-graduação e Pesquisa de Engenharia, Rio de Janeiro.

BEN-TAL, A.; GHAOUI, L. El; NEMIROVSKI, A. Foreword: special issue on robust optimization. **Math. Program.**, v. 107, n. 1-2, p. 1–3, 2006.

BERGUESS, C.J.; LEFLEY, M. Can genetic programming improve software effort estimation? **Information and Software Technology**, v. 43, n. 14, p. 863–873, 2001.

BIRGE, J. R. Models and model value in stochastic programming. **Annals of Operations Research**, v. 59, p. 1–18, 1995.

BISWAL, M.P.; BISWAL, N.P.; LI, Duan. Probabilistic linear programming problems with exponential random variables: A technical note. **European Journal of Operational Research**, v. 111, p. 589–597, 1998.

CHEESMAN, J.; DANIELS, J. **UML Components: A Simple Process for Specifying Component-Based Software**. Addison-Wesley, 2000. Disponível em: <http://www.syntropy.co.uk/umlcomponents/>.

EICHHORN, A.; RöMISCH, W. Stochastic integer programming: Limit theorems and confidence intervals. **Math. Oper. Res.**, INFORMS, Institute for Operations Research and the Management Sciences (INFORMS), Linthicum, Maryland, USA, v. 32, n. 1, p. 118–135, fev. 2007. ISSN 0364-765X. Disponível em: <http://dx.doi.org/10.1287/moor.1060.0222>.

FUENTES, L.; VALLECILLO, A. An introduction to UML profiles. **UPGRADE, The European Journal for the Informatics Professional**, v. 5, n. 2, p. 5–13, 2004.

GASSMANN, H. I. Modelling support for stochastic programs. **Annals of Operations Research**, v. 82, p. 107–137, 1998.

GETTING better results by making smarter decision. Paragon Decision Technology. Disponível em: <http://www.aimms.com/>.

HARMAN, M. The current state and future of search based software engineering. In: WORKSHOP (Ed.). **Proceedings of the...** Minneapolis, USA: IEEE Computer Society Washington, 2007. p. 1–15.

JACOBSON, I.; BOOCH, G.; RUMBAUGH, J. **The Unified Development Process**. New York: Addison-Wesley, 1999.

KHOSHGOFTAAR, T. M.; YI, L.; SELIYA, N. A multiobjective module-order model for software quality enhancement. **IEEE Transactions on Evolutionary Computation**, v. 8, n. 6, p. 593–608, 2004.

LAHDELMA, R.; HOKKANEN, J.; SALMINEN, P. SMAA - Stochastic multiobjective acceptability analysis. **European Journal of Operational Research**, v. 106, p. 137–143, 1998.

LODWICK, W. A.; KACPRZYK, J. (Ed.). **Fuzzy Optimization - Recent Advances and Applications**. New York: Springer, 2010. (Studies in Fuzziness and Soft Computing, v. 254). ISBN 978-3-642-13934-5.

PROJECT MANAGEMENT INSTITUTE. [S.l.]: **A Guide To The Project Management Body Of Knowledge (PMBOK Guides)**, 2004. ISBN 193069945X, 9781933890517.

SINHA, S. B.; HULSURKAR, S.; BISWAL, M. P. Fuzzy programming approach to multiobjective stochastic programming problems when b(i)'s follow joint normal distribution. **Fuzzy Sets and Systems**, v. 109, n. 1, p. 91–96, Jan. 2000.

STOUGIE, L.; VLERK, M. H. van der. **Annotated Bibliographies**. In: COMBINATORIAL OPTIMIZATION. New York: John Wiley & Sons, 1997. cap. Stochastic Integer Programming, p. 478–536.