

## Redes Neurais Aplicadas na Estimativa de Prazo de Projeto de Software

Beatriz T. Borsoi<sup>1</sup>, Kathya S. L. Colazzos<sup>1</sup>, Rúbia E. O. S. Ascari<sup>1</sup>, Luiz F. Toscan<sup>1</sup>,  
Luiz H. Bossola<sup>1</sup>, Matheus M. Bolo<sup>1</sup>, Matheus M. Arsego<sup>1</sup>

<sup>1</sup> Universidade Tecnológica Federal do Paraná (UTFPR) – Campus Pato Branco  
Via do Conhecimento – Km 01. CEP 85.503-390 – Pato Branco – PR

{beatriz,kathya,rubia}@utfpr.edu.br, luiz.ziulfer@gmail.com, {luizhenrique\_bosso  
la,matheus\_magnusson\_2006}@hotmail.com, matheus\_arsego@yahoo.com.br

**Resumo.** *A realização de estimativa representa uma etapa importante do processo de desenvolvimento de software e fornece subsídios para análise de viabilidade, licitação, contrato com cliente, orçamento, planejamento e acompanhamento do projeto. As conseqüências de estimativas imprecisas podem resultar em perdas significativas, como de contrato. Esses problemas podem evidenciar-se em fábricas de software de pequeno porte e/ou que estão iniciando suas atividades. Assim, este trabalho propõe uma metodologia para auxiliar na elaboração de estimativas de projeto de desenvolvimento de software, baseada na técnica de pontos de função e com uso de redes neurais artificiais que visam substituir o especialista em estimativa.*

### 1. Introdução

A falta de planejamento de projetos de software e de controle adequado faz com que boa parte dos projetos de software incorra em problemas, como prazo extrapolado e custo excedente [1], além do não atendimento aos requisitos definidos. Além disso, está o possível prejuízo financeiro ocasionado por projetos cancelados. Essas afirmativas podem ser respaldadas por Lycett et al. [2] que informam que 80 a 90% dos sistemas de software são entregues com atraso e com custos acima do orçamento original; cerca de 40% dos projetos falham ou são abandonados; menos de 25% dos sistemas integram adequadamente negócio e tecnologia e somente entre 10 a 20% são entregues de acordo com o planejamento de prazo, orçamento e qualidade.

Como forma de minimizar esses problemas está a aplicação de técnicas, métricas, procedimentos, modelos e metodologias para auxiliar no processo de estimativas em projetos de software. O objetivo é fornecer suporte para o planejamento do projeto por meio da definição de custos, prazo e recursos necessários. Contudo, estimar pode representar uma tarefa bastante complexa e com resultados, muitas vezes, imprecisos.

Estimativas corretas são fundamentais para a indústria de software, pois a definição de estimativas inadequadas em projetos de software contribuem para a atual caracterização da produção de software: custo excessivo, atraso, falta de qualidade e não atendimento aos requisitos do usuário [2][6]. Estimativas não realistas podem desgastar a equipe,

gerar problemas de contrato com clientes, determinar perda de oportunidades de mercado, agregar custos ao projeto e ocasionar não atendimento de requisitos, tanto funcionais como de qualidade do software implementado.

O prazo estimado para desenvolvimento do projeto pode ser utilizado para alocar recursos (financeiros, equipamentos, tecnologias, perfis profissionais necessários), determinar esforço (o trabalho necessário) e auxiliar a definir requisitos de contrato ou o valor do produto. É nessa perspectiva que se insere a proposta deste trabalho.

As maneiras de estimar definidas na literatura podem ser difíceis de aplicar e elas podem gerar resultados não adequados, seja pelo não entendimento da maneira de uso dessas formas de estimar, pela falta de material publicado ou de pessoas especializadas no seu uso. Esses problemas podem evidenciar-se ainda mais em fábricas de software de pequeno porte e/ou que estão iniciando as suas atividades.

O problema da estimativa pode ser visto como um problema de regressão, que pode ser resolvido de diversas maneiras, desde técnicas mais clássicas como regressão linear, até técnicas mais recentes como o uso de Redes Neurais Artificiais (RNAs). Redes Neurais Artificiais são inspiradas na estrutura e no funcionamento do cérebro humano [3]. Elas são compostas por unidades de processamento simples (neurônios artificiais) dispostas em uma ou mais camadas e interligadas por conexões geralmente associadas a pesos que armazenam o conhecimento adquirido e ponderam as entradas recebidas pelos neurônios [4]. Pode-se considerar que RNAs representam um sistema capaz de agir de maneira próxima a forma como o humano adquire e representa seu conhecimento.

Este trabalho tem como objetivo principal propor uma forma de aplicação de redes neurais artificiais para apoiar uma metodologia baseada em pontos de função proposta em Borsoi et al. [5] para a definição de prazo de projetos de desenvolvimento de software. As redes neurais substituiriam o conhecimento do especialista em estimativas.

## **2. Estimativa de Prazo de Desenvolvimento de Software**

Estimar, em Engenharia de Software, consiste em determinar (prever) prazo, recursos e esforço necessários para desenvolver um projeto de software [7]. Reinaldo e Filipakis [8] complementam com determinar o tamanho do produto a ser desenvolvido e o custo à organização. O tamanho do software, medido por meio das suas funcionalidades, pode fornecer suporte para definir os custos, o prazo e o esforço (trabalho) necessários para implementá-lo. O prazo pode ser usado para definir o cronograma, os custos, o planejamento e a alocação de recursos.

De acordo com Hazan [9], o responsável pelas estimativas deve analisar os requisitos para garantir a qualidade do produto e então estimar o tamanho do projeto de software, a partir do qual é calculado o esforço necessário que determina as estimativas de prazo e de custo. Assim, a partir do tamanho do projeto é possível obter as outras estimativas, de forma a identificar as necessidades de recursos e de pessoal, estimar produtividade e qualidade e quantificar o impacto de mudanças no projeto.

Existem diversas técnicas descritas na literatura para auxiliar na realização de

estimativas. A técnica escolhida como base para a proposta deste trabalho é pontos de função [10], que é orientada à funcionalidade do software. Assim, o processo de definição de requisitos tem papel fundamental. É por meio deles que as necessidades dos usuários são mapeadas para as características e funcionalidades de um software. Contudo, os requisitos devem ser definidos em uma linguagem que seja compreendida pela equipe de projeto. Para que, assim, a estimativa possa ser mais real.

Neste trabalho pontos de função é utilizada na fase de requisitos [7], pois acredita-se que quanto mais cedo, em termos de ciclo de vida, a estimativa é realizada mais útil ela poderá ser para definir o cronograma do projeto, o contrato com o cliente, o planejamento, estimar custos e recursos necessários.

## 2.1 Pontos de Função

Pontos de função, criada na década de 1970 por Allan Albrecht da IBM [11], é uma técnica utilizada para medir sistemas do ponto de vista dos seus usuários pela quantificação da funcionalidade solicitada e fornecida. Para Vazquez, Simões e Albert [12] requisito é uma condição, característica ou capacidade determinada no universo das necessidades do negócio do usuário, que deve ser atendida por um sistema na forma de aspectos funcionais e não funcionais. As funcionalidades dizem “o que” será entregue ao usuário ou “o que” o sistema fará.

Pontos de função é uma técnica caracterizada como modelo algoritmo, regulamentada pelo International Function Point Users Group (IFPUG), e oficializada pelo padrão ISO/IEC 20926 [13]. Os cinco principais componentes de uma aplicação sendo medida e estimada são: entradas da aplicação, saídas da aplicação, arquivos lógicos mantidos pela aplicação, consultas e interfaces da aplicação e com outras aplicações. Esses componentes definem os pontos de função não ajustados. Valores de ajuste de complexidade podem ser utilizados para adequar a contagem. São fatores que podem influenciar no desenvolvimento do projeto e não são considerados na obtenção dos pontos de função. A FPCPM [10] indica 14 fatores de ajuste.

Várias extensões e complementos têm sido propostos ao modelo de pontos de função. Dentre esses estão as adaptações para software orientado a objetos. Schooneveldt [14] trata classes como objeto e serviços fornecidos a clientes como transações. Para Whitmire [15] cada classe é considerada como um arquivo interno e as mensagens enviadas através da fronteira do sistema são tratadas como transações. Sneed [16] propôs pontos de objeto como medidas para estimar tamanho de software orientado a objetos. Pontos de objeto são derivados da estrutura de classes, de mensagens e de processos ou casos de uso e ajustados por meio de fatores de ajuste.

A técnica de pontos de função é questionada sobre o significado da quantificação de pontos de função. Pressman [7] ressalta que ponto de função é apenas uma indicação numérica. Para Peters e Pedrycz [17] o modelo de pontos de função ignora as tecnologias de redução do esforço como ambientes integrados de desenvolvimento de software, descrições de projetos executáveis, bibliotecas de reuso e orientação a objetos. Outra desvantagem apontada é que na contagem não são consideradas particularidades

da equipe de desenvolvimento ou mesmo da complexidade algorítmica do sistema.

Apesar dessas desvantagens, a técnica pontos de função foi escolhida como base para este trabalho porque a referência de contagem são os requisitos do sistema. E, ainda, por poder ser utilizada no início do processo de desenvolvimento e por não ter uso vinculado a linguagens de programação específicas. Os aspectos indicados como não abrangidos por essa técnica, são considerados na proposta deste trabalho, seja como fatores definidos ou de ajuste. O cálculo do prazo estimado é realizado com base nos requisitos do sistema. Esses requisitos são utilizados pelas redes neurais para prever o prazo necessário para implementá-los. O valor de ajuste é definido por requisitos relacionados ao ambiente, às tecnologias e à equipe.

### **3. Redes Neurais**

As RNAs são sistemas paralelos distribuídos compostos por unidades de processamento simples (neurônios artificiais) que calculam determinadas funções matemáticas (normalmente não-lineares). Tais unidades são dispostas em uma ou mais camadas e interligadas por um grande número de conexões, geralmente unidirecionais. Na maioria dos modelos essas conexões estão associadas a pesos, os quais armazenam o conhecimento adquirido pelo modelo e servem para ponderar a entrada recebida por cada neurônio da rede [4].

Em RNAs, o procedimento usual na solução de problemas passa inicialmente por uma fase de aprendizagem, em que um conjunto de exemplos é apresentado para a rede, a qual extrai as características necessárias para representar a informação fornecida. Essas características são utilizadas posteriormente para gerar respostas ao problema [4]. As RNAs são usualmente organizadas em camadas compostas de unidades de processamento (neurônios), conectadas por canais de comunicação pelos quais transitam dados numéricos. Os dados são apresentados à rede por meio de uma camada de entrada, que se comunica a uma ou mais camadas ocultas onde ocorre o processamento por meio de um sistema de conexões ponderadas. A camada de saída apresenta o resultado do processamento [18].

O uso de uma RNA para a solução de um determinado problema consiste no ajuste de seus pesos, o que é feito em uma primeira fase de treinamento. Treinar uma rede significa ajustar a sua matriz de pesos ( $W_i$ ) de forma que o vetor de saída coincida com certo valor desejado para cada vetor de entrada. O treinamento supervisionado necessita de pares de entrada e saída que são denominados de conjunto de treinamento. No treinamento, o vetor de entrada é aplicado e a saída da rede calculada. Comparando a resposta da rede com o vetor de saída esperado, o erro é calculado. Com base nesse erro, os pesos são ajustados. O processo de treinamento é repetido até que o erro para o conjunto de treinamento alcance um valor de limite previamente determinado [3].

Um dos benefícios das RNAs se refere ao tratamento de um problema clássico de Inteligência Artificial que é a representação de um universo no qual as estatísticas mudam com o tempo [19]. Osório [20] destaca que as RNAs podem ser aplicadas a diferentes tipos de tarefas, tais como: o reconhecimento de padrões (exemplo

reconhecimento de faces humanas), a classificação (como o reconhecimento de caracteres), a transformação de dados (compressão de informações, por exemplo), a predição (previsão de séries temporais, como as cotações da bolsa de valores ou diagnóstico médico) e o controle de processos e a aproximação de funções (com aplicações na área da robótica).

O problema da estimativa é apenas mais um entre vários problemas do mundo real que podem ser resolvidos com uso de conceitos de inteligência computacional. Em vez de usar um especialista humano para a obtenção das estimativas, usa-se um sistema computacional que possa substituí-lo. Uma das técnicas inteligentes que pode resolver o problema das estimativas são as RNAs [4]. Contudo, para que estimativas confiáveis sejam obtidas utilizando RNAs é indispensável que elas sejam adequadamente implementadas e treinadas. O conjunto de dados de treinamento e de testes deve ser amplo e confiável. Os testes servem para verificar os dados obtidos como resultado do treinamento.

O uso de redes neurais artificiais em estimativa de projetos de software é exemplificado pelos trabalhos de Tronto, Silva e Sant'Anna [21], Attarzadeh e Ow [22], Tadayon [23], Ajitha et al. [24] e García et al. [25].

Tronto, Silva e Sant'Anna [21] comparam redes neurais artificiais preditivas e modelos baseados em regressão, constatando que redes neurais artificiais são efetivas na geração de estimativa de esforço em projetos de software.

Attarzadeh e Ow [22] propõem uma rede neural artificial e Tadayon [23] uma máquina de aprendizado adaptativo baseada em uma rede neural dinâmica para estimar o custo de software com base no modelo de custo construtivo (COCOMO). Esses trabalhos agregam características desejáveis de redes neurais, como a habilidade de aprendizagem e boa interpretabilidade, mantendo as fundamentações desse modelo.

Ajitha et al. [24] desenvolveram uma rede neural artificial usando pontos de caso de uso para gerar estimativas de tamanho de software, destacando também as vantagens na utilização de redes neurais artificiais com relação aos diversos procedimentos existentes para geração de estimativas. García et al. [25] propõem uma metodologia de otimização para procurar o melhor modelo neural aplicável na avaliação de esforço em projetos de software, permitindo extrair um conjunto de fatores conhecidos em estágios iniciais do desenvolvimento, a fim de garantir a adequação do modelo de rede neural proposto e otimizar o desempenho em relação ao tempo e a exatidão das estimativas.

#### **4. Uso de Redes Neurais na Estimativa de Prazo de Software**

Neste trabalho, é apresentada uma proposta de aplicação de redes neurais para estimar prazo de desenvolvimento de projetos de software. O uso de redes neurais decorre de elas empregarem técnicas de aproximação de funções por regressão não linear, aproximando-se da forma como um especialista realiza estimativas. Isso porque o prazo não tem aumento linear proporcional ao número de requisitos de entrada.

As redes neurais utilizam como entrada os requisitos do sistema a ser desenvolvido e o o

tempo padrão para implementar cada tipo de requisito. Esse tempo é definido pelas redes treinadas. O treinamento está baseado em padrões de tempo informados por especialistas. As entradas para as redes utilizam uma tabela de fatores definidores de prazo (Quadro 1). Os fatores que correspondem a esses agrupamentos e as respectivas quantidades associadas estão em Borsoi et al. [5].

Grupos	Fatores definidores de prazo – Quantidade
Manutenção de dados	1. Simples (inclusão, exclusão, alteração, consulta): a) até 5 campos; b) 6 a 15 campos; c) mais de 15 campos.
	2. Complexa (com validação, referências cruzadas, campos calculados, buscas inclusive com filtros): a) até 5 campos; b) 6 a 15 campos; c) mais de 15 campos.
Geração de relatórios	1. Simples (listagem de cadastros): a) 1 tabela;
	2. Complexo (campos calculados, joins, unions, subselects, filtros, agrupamentos, ordenação, referência cruzada, gráficos): a) 1 tabela; b) 2 a 5 tabelas; c) mais de 5 tabelas.
	3. Com geração de arquivos externos de dados: a) 1 tabela; b) 2 a 5 tabelas; c) mais de 5 tabelas.
Interação com periféricos	1. Acesso a sistemas externos (validação em banco de dados de outro sistema, interface/protocolo de comunicação): 1 tabela; b) 2 a 5 tabelas; c) mais de 5 tabelas.
	2. Interação com dispositivos como leitores de biometria, código de barras e etc. Obtenção de dados de dispositivos (sensores): 1 tabela; b) 2 a 5 tabelas; c) mais de 5 tabelas.
	3. Envio de comandos para periféricos: 1 tabela; b) 2 a 5 tabelas; c) mais de 5 tabelas.
Processamento	1. Validação de acesso (login)
	2. Cálculos com ou sem consulta a banco de dados, com conversão de dados.: a) 1 cálculo; b) 2 a 5 cálculos; c) mais de 5 cálculos
	3. Cálculos para atender a legislação, com conversão de dados: a) 1 cálculo; b) 2 a 5 cálculos; c) mais de 5 cálculos
	4. Processamento interno complexo, lógico e matemático extensivo: a) 1 processamento; b) 2 a 5 processamentos; c) mais de 5 processamentos.
	5. Segurança como SSL ou código para criptografia.
	6. Requisitos não funcionais relevantes: restrições de desempenho, memória, portabilidade, níveis de acesso: a) 1 requisito; b) 2 a 4 requisitos; c) mais de 5 a 8 requisitos; d) mais que 8 requisitos.
	7. Bancos de dados distintos com peculiaridades na manipulação de cada banco.: a) 1 tabela; b) 2 a 5 tabelas; c) mais de 5 tabelas.

### Quadro 1. Funcionalidades do sistema (agrupamento de requisitos)

Fonte: Borsoi et al. (2010) [5]

A aplicação sugerida de redes neurais artificiais é para automatizar a definição dos tempos padrão para cada requisito do sistema. Dessa forma, o conhecimento do responsável por gerar estimativas, o especialista, seria transmitido às redes neurais em sua fase de treinamento. Essas redes treinadas podem generalizar as respostas apresentadas pelo especialista, sendo capazes de gerar estimativas de prazo para situações diferentes das utilizadas como treinamento.

Considera-se necessário implementar uma rede neural para cada um dos quatro grupos que compõe a tabela de fatores definidores de prazo, de forma a diminuir a complexidade da própria rede. Assim, os tempos apresentados pelas redes neurais distintas serão somados para gerar o tempo total estimado para o desenvolvimento de um projeto. A partir de vários conjuntos de entrada (requisitos) e saída (tempo estimado), cada rede neural é treinada até que ela obtenha o conhecimento necessário para simular a prática e conhecimento do especialista para gerar estimativas.

A Figura 1 representa como são as entradas, o processamento (que ocorre em uma ou mais camadas ocultas) e a saída da rede neural. Essa figura é utilizada como modelo

para a proposta que está no Quadro 1. Nesse quadro, a coluna “Grupos” indica cada uma das quatro redes utilizadas e a na coluna 2 estão as entradas para cada uma das redes. Para cada uma dessas entradas é informada a quantidade, por exemplo, quantas manutenções de dados simples há para o sistema objeto de estimativa de prazo. Na camada oculta é realizado o processamento, a estimativa de prazo propriamente dita. Como saída de cada uma das redes neurais que implementada cada um dos quatro grupos de requisitos obtém-se:

- a)  $\sum 1$  dos tempos para manutenção de dados;
- b)  $\sum 2$  dos tempos para geração de relatórios;
- c)  $\sum 3$  dos tempos para interação com periféricos;
- d)  $\sum 4$  dos tempos para processamento.

Na fase de treinamento de cada uma das quatro redes, o processamento é realizado com base em conhecimento do especialista. Após a fase de treinamento, a definição do prazo tem como base o conhecimento adquirido pela rede neural a partir desse treinamento. Para o cálculo da estimativa de prazo não ajustado é feito o somatório dos prazos obtidos em cada subgrupo, que é a saída de cada uma das quatro redes neurais:

$$\text{Tempo não ajustado} = \sum 1 + \sum 2 + \sum 3 + \sum 4$$

Para que as redes possam estimar é necessário indicar a quantidade de cada uma das entradas (os requisitos). Isso será feito à semelhança do sistema que foi desenvolvido para estimar o prazo de projeto de software com base no conhecimento do especialista (tela principal na Figura 1). As redes neurais, após devidamente treinadas, substituirão o especialista na indicação dos tempos padrão para cada um dos fatores definidos. O campo “Tempo” indicado no formulário da Figura 1 será fornecido pelas redes neurais.

Dados Simples		Dados Complexos	
Até 5 campos:	3	Tempo:	20
De 6 até 15 campos:	3	Tempo:	25
Mais de 15 campos:	2	Tempo:	40
Até 5 campos:	2	Tempo:	50
De 6 até 15 campos:	1	Tempo:	60
Mais de 15 campos:	1	Tempo:	90

**Figura 1. Interface do aplicativo para estimativa de prazo**

O tempo estimado pelas redes é ajustado por meio da aplicação de fatores modificadores

(última aba da Figura 1), que consideram o contexto em que o projeto será desenvolvido. Esses fatores (expostos no Quadro 2) têm o objetivo de ajustar, reduzir ou aumentar, o prazo obtido com base nos atributos definidores de prazo. Esses fatores podem influenciar na implementação, mas não estão diretamente relacionados ao sistema que será produzido e que é objeto da estimativa de prazo obtida pelas redes.

Grupo	Fator
Relacionados ao produto (o sistema a ser desenvolvido)	Reuso de artefatos
	Confiabilidade das operações realizadas pelo sistema.
	Integração com sistemas existentes
Relacionados à equipe	Conhecimento e experiência da equipe em análise e projeto
	Conhecimento e experiência da equipe nas tecnologias usadas
	Mudanças na equipe
Relacionados ao negócio	O conhecimento da equipe do negócio a que se refere o sistema
Relacionados ao uso de ferramentas, métodos e modelos	Uso de processos
	Uso de modelos de artefatos

**Quadro 2. Fatores modificadores de prazo**

A proposta deste trabalho está sendo desenvolvida com a criação de redes neurais do tipo *Multi Layer Perceptron* em linguagem de programação C, baseada nos fundamentos apresentados em Ludwig Junior e Montgomery [19]. Após sua conclusão, pretende-se utilizar o sistema desenvolvido em fábricas de software de pequeno porte, com áreas de atuação distintas, a fim de averiguar a eficácia da proposta, ou mesmo do modelo de rede neural adotado.

## 5. Conclusão

A base das RNAs está na sua capacidade de aprender por meio de um conjunto de exemplos e extrapolar esse aprendizado para dados não conhecidos. Assim, elas podem substituir o especialista, após estarem devidamente treinadas com base em conhecimento do especialista. Esse treinamento se refere à rede aprender para determinadas entradas quais as possíveis saídas e gerar saídas corretas para outras entradas, ainda que deva ser considerado o escopo do problema e da solução nos quais ela foi treinada. Nessa perspectiva, a proposta deste trabalho visa utilizar o conhecimento do especialista para capacitar redes neurais a realizarem a atividade de estimativa de prazo de projeto de software com base nos requisitos do sistema e tendo a fundamentação conceitual da técnica de pontos de função.

A solução proposta substituirá um sistema que atualmente realiza o cálculo do prazo com base na indicação de prazo por especialista para cada um dos fatores definidos a partir de pontos de função. Nesse sistema, o prazo é ajustado utilizando lógica *fuzzy*. Os fatores definidores de prazo são combinados em regras para melhor representar a influência de cada fator. Na solução proposta por meio deste trabalho, as redes neurais substituirão o especialista na definição do prazo não ajustado. E lógica *fuzzy* poderá, ainda, ser utilizada para o fator de ajuste.

O algoritmo para implementar as redes já foi testado, restando treiná-las. Para isso é necessário obter dados de diversos especialistas e verificar como o algoritmo se

comporta considerando os diferentes tempos informados pelos especialistas. Diversos dados já foram coletados e verificou-se que eles são consideravelmente divergentes. Um estudo será realizado para verificar se os especialistas têm considerado a influência de fatores modificadores na indicação do tempo para implementar os requisitos.

As redes neurais têm capacidade de aprendizado a partir de dados com ruídos e até contraditórios, mas é necessário verificar e validar adequadamente os resultados da fase de treinamento. Assim, o treinamento realizado será do tipo supervisionado, em que será fornecido o conjunto de exemplos com as entradas e respectivas respostas desejadas.

### **Agradecimentos**

A Fundação Araucária do Estado do Paraná, Brasil, pelo suporte financeiro ao projeto.

### **Referências**

- [1] Pech, G., “Como gerenciar na prática projetos de desenvolvimento de software,” 2002. <<http://www.bfpug.com.br>>, março de 2011,11.
- [2] Lycett, M., Macredie, R. D., Patel, C., Paul, R. J. “Migrating agile methods to standardized development practice,” IEEE Computer Society, jun., 2003, p. 79-85.
- [3] Haykin, S., Redes neurais: princípios e práticas. 2 ed. Bookman, 2007.
- [4] Braga, A. P. Carvalho, A. P. L. F.; Ludermir, T. B. Redes neurais artificiais – teoria e aplicações. 2a ed. Rio de Janeiro: LTC, 2011.
- [5] Borsoi, B. T.; Linares, K. S. C.; Ascari, R. E. O. S.; Rocha, L. G.; Toscan, L. F.; Bolo, M. M., “Fatores definidores e modificadores em estimativa de prazo de projeto de software,” Semana de Inovações em Sistemas de Informação (XI SIS2INFO), 2010.
- [6] Bragança, A.; Machado, R. J., “Model driven development of software product lines,” 6th International Conference on the Quality of Information and Communications Technology (QUATIC 2007), IEEE Computer Society, 2007, p. 199-203.
- [7] Pressman, R., Engenharia de software, 6a ed. Rio de Janeiro: McGraw-Hill, 2006.
- [8] Reinaldo, W. T.; Filipakis, C. D., “Estimativa de tamanho de software utilizando APF e a abordagem NESMA,” XI Encontro de Estudantes de Informática do Tocantins, 2009, p. 151-160.
- [9] Hazan, C., “Análise de pontos de função - uma aplicação nas estimativas de tamanho de projetos de software,” Engenharia de Software Magazine, No. 2, 2009, p. 25-31.
- [10] FPCPM. Function point counting practices manual, ver. 4.1.1. International Function Point Users Group, 1999. <<http://www.ifpug.org>>, dezembro 2011, 08.
- [11] Albrecht, A. J., “Measuring application development productivity,” Joint Application Development Symposium, 1979, p. 83-92.
- [12] Vazquez. C. E. Simões, G. S. Albert, R. M., Análise de pontos de função. Medição, estimativas e gerenciamento de projetos de software, 8ª ed. São Paulo: Érica, 2008.
- [13] ISO/IEC 20926:2009, Software and systems engineering - Software measurement.

IFPUG functional size measurement method 2009, ISO/IEC, 2009.

[14] Schooneveldt, M., "Measuring the size of object oriented system," 2nd Australian Conference on Software Metrics. Australian Software Metrics Association, 1995, p. 168-177.

[15] Whitmire, S., "Applying function points to object-oriented software," Software Engineering Productivity Handbook, McGraw-Hill, 1993, p. 229-244.

[16] Sneed, H., "Estimating the costs of object-oriented software," Software Cost Estimation Seminar, 1995, p. 35-45.

[17] Peters, J. F.; Pedrycz, W., Engenharia de software. São Paulo: Campus, 2001.

[18] German, G. W. H.; Gahegan, M. N., "Neural network architectures for the classification of temporal image sequences," Computers Geosciences, Vol. 22, No. 9, 1996, p. 969-979.

[19] Ludwig Junior, O.; Montgomery, E., Redes neurais - fundamentos e aplicações com programas em C. Rio de Janeiro: Editora Ciência Moderna Ltda. 2007.

[20] Osório, F., "Redes neurais artificiais: do aprendizado ao aprendizado Artificial," I Fórum de Inteligência Artificial, 1999. <<http://osorio.wait4.org/oldsite/IForumIA/fia99.pdf>>, junho 2011, 22.

[21] Tronto, I. F. B.; Silva, J. D. S. S. Sant'anna, N., "Comparison of artificial neural network and regression models in software effort estimation," International Joint Conference on Neural Networks (IJCNN'2007), 2007, p. 771-776.

[22] Attarzadeh, I.; OW, S. H., "Proposing a new software cost estimation model based on artificial neural networks," 2nd International Conference on Computer Engineering and Technology (ICCET 2010), IEEE Computer Society, 2010, p. 487-491.

[23] Tadayon, N., "Neural network approach for software cost estimation," International Conference on Information Technology: Coding and Computing (ITCC'05), 2005 p. 815-818.

[24] Ajitha, S.; Kumar, T. V. S.; Geetha, D. E.; Kanth, K. R., "Neural network model for software size estimation using use case point approach," 5th International Conference on Industrial and Information Systems (ICIIS 2010), 2010, p. 372-376.

[25] García, A.; González, I.; Colomo, R.; López, J.L.; Ruiz, B., "Methodology for software development estimation optimization based on neural networks," IEEE Latin America Transactions, Vol. 9, No. 3, jun. 2011, p. 391-405.