



Resumo Expandido

Título da Pesquisa (Português): Desenvolvimento de uma Aplicação de Apoio ao Planejamento de Projetos (ProScenario)		
Título da Pesquisa (Inglês): Development of a Application to Project Planning (ProScenario)		
Palavras-chave: restrição de recursos, planejamento de projetos, escalonamento de projetos, RCPSP, heurísticas		
Keywords: resource constraint, project planning, project scheduling, RCPSP, heuristics		
Campus: Formiga	Tipo de Bolsa: PIBITI/CNPq	Financiador: CNPq
Bolsista(s): Raí Caetano de Jesus		
Professor Orientador: Diego Mello da Silva		
Área de Conhecimento: 3.08.02.00-8: Pesquisa Operacional, 1.03.03.02-2: Engenharia de Software, 1.03.02.02-6: Modelos Analíticos e de Simulação		Edital: 139/2013

Resumo: Este trabalho apresenta brevemente o processo de desenvolvimento do sistema de informação da aplicação ProScenario, cujo objetivo é auxiliar na tomada de decisões de gerentes de projetos durante o planejamento de projetos com restrição de recursos. O protótipo *standalone* implementado até o momento segue a arquitetura de três camadas e possui funcionalidades como: autenticação, recuperação de credenciais, cadastro de usuários, gerenciamento de projetos, gerenciamento de recursos, construção da estrutura analítica do projeto, cadastro de dependências entre as atividades e geração do gráfico de Gantt. O próximo passo é realizar a integração entre as camadas de apresentação e regra de negócio por meio de *Web Services*, e integrar a aplicação ao módulo de tomada de decisão que constrói cronogramas restritos e faz análise de risco no prazo e custo do projeto.

Abstract: This work presents briefly the development of the software ProScenario, an information system that aids project managers to make decision during the projects planning with resource constraints. The present standalone prototype was implemented in three tiers, and has functionalities such as authentication, recovery of credentials, user registration, projects management, resources management, construction of the work breakdown structure, dependences between activities, and Gantt chart generation. The next step is to perform the integration between the decision making module and the application. This feature will help projects managers to create constraint schedules and foreseen risks in costs and time.

INTRODUÇÃO

Concluir projetos dentro das restrições de escopo, custo e tempo é uma tarefa difícil. Para minimizar as dificuldades e ultrapassar os obstáculos impostos por tais restrições durante a execução de um projeto, é necessário o uso de ferramentas, técnicas e habilidades contidas no âmbito da gerência de projetos.

Dentre os diversos sistemas de informação e de apoio à gerência de projetos encontrados na atualidade destacam-se Microsoft Project (<https://products.office.com/pt-br/project/project-and-portfolio-management-software>), Wrike (<https://www.wrike.com/>), ProWorkflow (<http://www.proworkflow.com/>),

Replicon (<http://replicon.com/>), Project Builder (<http://projectbuilder.com.br/>), Artia (<http://artia.com/>), Clarizen (<http://www.clarizen.com/>), TeamWork (<https://www.teamwork.com/>), DeskAway (<http://www.deskaway.com/>), Zoho Projects (<http://www.zoho.com/projects/>), ScrumHalf (<http://myscrumhalf.com/>) entre outros, tanto para a metodologia tradicional (PMBOK, [PMI,2013]) quanto para metodologias ágeis (SUTHERLAND, 2014). Muitos destes sistemas contêm funcionalidades interessantes para um gestor de projetos, como o gerenciamento de tarefas e trabalho colaborativo, e em alguns é possível visualizar um gráfico de Gantt¹ com as tarefas ao longo do tempo. No entanto, nem todos abordam o planejamento de projetos com restrição de recursos (problema conhecido na literatura de otimização combinatória por *Resource Constraint Project Scheduling Problem* – RCPSP [KLEIN, 2000]) ou fazem análise de risco de prazo e custo que ajude na tomada de decisão, sendo necessário pagar mais para usufruir de funcionalidades avançadas já que parte dos sistemas supracitados são pagos. Além do mais, a maioria não constrói cronogramas que atendam ambas as restrições de recursos e de precedência entre as atividades.

O problema RCPSP é conhecido ser da classe de problemas NP-Difícil [(ICHIHARA, 2002), (KOLISH; HARTMANN, 1999), (MENDES; GONÇALVES; RESENDE, 2009)], portanto é improvável que exista algum algoritmo determinístico de tempo hábil capaz de calcular cronogramas restritos que minimizem a duração total do projeto (*makespan*). Desta forma, este trabalho apresenta detalhes sobre o desenvolvimento do aplicativo ProScenário que, quando finalizado, será capaz de calcular cronogramas para o RCPSP por meio de heurísticas eficientes em termos de complexidade computacional. A próxima seção apresentará as etapas executadas até o presente momento para o desenvolvimento do aplicativo. Discussões sobre as heurísticas usadas estão fora do escopo deste trabalho.

METODOLOGIA

Inicialmente realizou-se uma revisão de literatura sobre os principais temas envolvidos no projeto, como por exemplo, gerenciamento de projetos, programação orientada à objetos (OOP), arquitetura orientada à serviços (SOA) (HURWITZ; *et. al*, 2009), protocolo simples de acesso a objetos (SOAP) (TIDWELL; SNELL; KULCHENKO, 2001), escalonamento de projetos, escalonamento de projetos com restrição de recursos (RCPSP) e metaheurísticas.

Mais especificamente, o estudo sobre gerenciamento de projetos abordou temas como a levantamento de atividades, estrutura analítica do projeto (EAP/WBS), estimativas de duração de atividades usando estimativa de três pontos, custos e recursos necessários para atividades, escalonamento de projetos usando PERT/CPM, gráfico de Gantt, diagrama de rede, entre outros assuntos. Tais temas foram fundamentais para entender o real objetivo do projeto e como este ajudará no planejamento de projetos usando metodologias tradicionais.

Por se tratar de um projeto de implementação computacional, fez-se necessário um estudo sobre o paradigma orientado a objetos, amplamente utilizado no desenvolvimento de *softwares*. Devido à sua grande popularização no mercado e o fato de ser orientada a objetos, optou-se pela linguagem C# (SHARP, 2013) para implementar a aplicação. Ela oferece recursos que facilitam o desenvolvimento de aplicações, como um

¹ Uma linha do tempo que mostra quando cada atividade começa, por quanto tempo ela continua, e quando termina.

ORM² nativo, facilidades para implementação de arquiteturas orientadas a serviços, entre outros. Além disso, toda a implementação foi feita no ambiente de desenvolvimento integrado (IDE) proprietário da Microsoft, o Visual Studio (JOHNSON, 2014), o qual é disponibilizado gratuitamente graças ao convênio realizado entre o IFMG Câmpus Formiga e o programa *Microsoft DreamSpark*.

O aplicativo ProScenário utiliza a arquitetura cliente-servidor através de *Web Services* (TIDWELL; SNELL; KULCHENKO, 2001) usando SOA e SOAP. A linguagem escolhida para implementação oferece recursos que facilitam o desenvolvimento de aplicações que se comunicam por meio de serviços, como por exemplo o *framework* WCF (*Windows Communication Foundation*) e classes específicas para hospedagem de serviços na forma de *Windows Services* (LOWY, 2010).

A Figura 1 ilustra sob forma de mapa mental os módulos previstos e já implementados para o produto final esperado deste trabalho, que dão uma ideia do escopo abrangido pelo projeto, assim como possíveis funcionalidades do aplicativo que podem ser transformadas em *Web Services*.

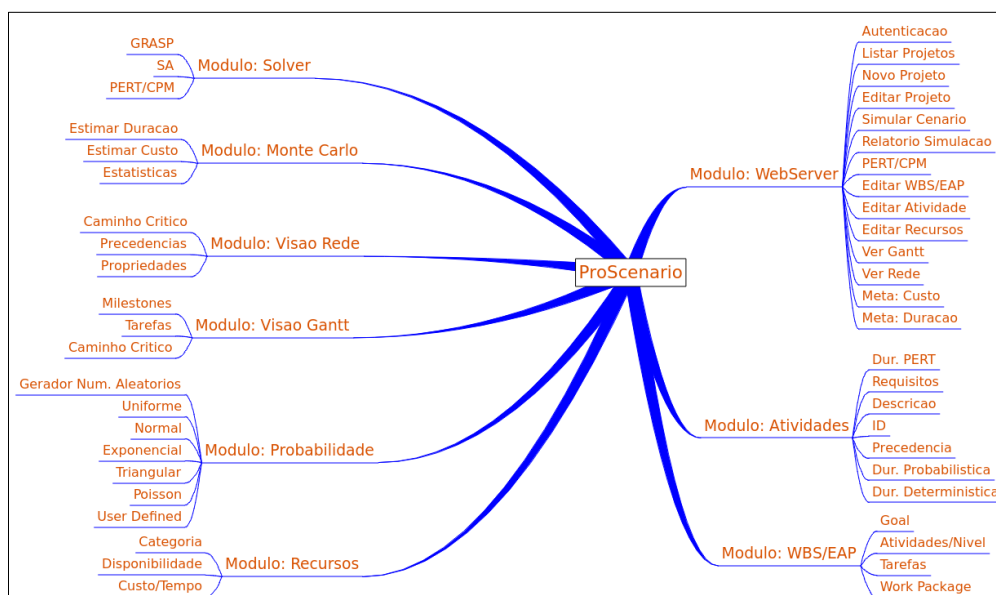


Figura 1 - Mapa Mental com os módulos previstos para o aplicativo ProScenário

Toda a modelagem computacional da aplicação foi realizada em partes. Primeiro realizou-se a modelagem das classes; em seguida, fez-se a modelagem do banco de dados e, por fim, a modelagem dos serviços, cuja qual ainda está em desenvolvimento. O aplicativo ProScenário utilizará o banco de dados PostgreSQL (<http://www.postgresql.org/>), uma vez que o mesmo é eficiente e oferece recursos robustos de forma gratuita.

O desenvolvimento do aplicativo segue o padrão de arquitetura de três camadas (apresentação, regra de negócio e acesso a dados [FOWLER; *et al.*, 2003]). A camada de apresentação contém as classes responsáveis pela interface gráfica da aplicação. Na camada de regra de negócio estão as classes responsáveis por toda a lógica da aplicação, como as heurísticas para geração de cronogramas, cálculos para estimativas de custos e duração de atividades, cálculo do PERT/CPM, entre outras. Por fim, a camada

² Mapeamento objeto-relacional (ou ORM, do inglês *Object-Relational Mapping*). Permite aos desenvolvedores trabalhar com dados relacionais usando classes e objetos específicos de domínio.

de acesso a dados contém as classes responsáveis pela persistência e acesso aos mesmos. Nesta última, todo o trabalho é realizado pelo ORM nativo da plataforma .NET, o Entity *Framework* (MUELLER, 2013).

O primeiro protótipo do aplicativo contém funcionalidades limitadas, abrangendo os seguintes módulos conforme a Figura 1: Módulo Recursos, Módulo Atividades, Módulo WBS/EAP e Módulo Visão Gantt. Toda a interface gráfica do protótipo foi desenvolvida com os recursos do *Windows Forms*³. Ainda, classes de serialização e desserialização foram implementadas de modo a realizar o salvamento e carregamento dos dados do projeto, permitindo que o usuário salve seu projeto localmente ou carregue um já salvo. Além do mais, para a construção do gráfico de Gantt, utilizou-se o componente gráfico gratuito *.NET Winforms Gantt Chart Control* (disponível em: <http://ganttchart.codeplex.com/>).

A aplicação *server-side* será hospedada em um servidor Dell PowerEdge T420 com processadores Intel Xeon E5-2430, 2.5 GHz, 15 Mb de cache, com 6 núcleos/12 threads, 12 Gb, disco rígido de 1Tb SATA. A configuração inclui processadores *multi-core* para dar suporte a simulação de cenários via múltiplas *threads* de execução (isto é, um cenário por núcleo) de forma a reduzir o tempo de resposta ao usuário visto que diferentes cenários variando disponibilidade de recursos serão simulados para o mesmo projeto.

RESULTADOS E DISCUSSÕES

Até o presente momento, a aplicação em desenvolvimento conta com 25 soluções do Visual Studio (aproximadamente 9517 linhas de código). A Tabela 1 resume as soluções implementadas, descrevendo seu papel na aplicação bem como a quantidade de classes implementadas e a qual camada pertence. A camada de apresentação possui 16 classes implementadas (uma por interface, totalizando 4643 linhas de código).

Tabela 1 - Resumo das soluções contidas na aplicação

Solução	Descrição	# Classes	Camada
AccountService	Contém os contratos e implementação do serviço responsável pelo gerenciamento de contas.	1	Negócio
DB	Contém os métodos para persistência dos dados (utilizando o ORM Entity).	2	Dados
Distribution	Implementa as distribuições probabilísticas para geração de variáveis aleatórias, utilizadas pela aplicação no módulo de simulação.	8	Negócio
ExceptionHandler	Contém classes que manipulam exceções geradas pela aplicação.	3	Negócio
Heuristics	Gerencia as características e comportamentos das heurísticas utilizadas no escalonamento das atividades.	1	Negócio
HumanReadable	Faz conversões de datas em formato discreto para formato de data. Ex: DD/MM/YYYY	1	Negócio
LoginService	Contém os contratos e implementação do serviço de <i>login</i> .	1	Negócio
LogService	Utilizada pela camada de apresentação para simular o funcionamento do serviço que fará o armazenamento de LOG's no banco de dados.	1	Negócio
Models	Contém as classes que são utilizadas pelo Entity para mapeamento objeto-relacional.	24	Dados Apresentação
PriorityRules	Implementa as heurísticas construtivas baseadas em regras de prioridade.	16	Negócio
ProjectService	Contém os contratos e implementação do serviço responsável pelo gerenciamento de projetos do usuário.	1	Negócio
ProScenario	Contém as classes que representam os objetos de um projeto. Utilizada pela camada de apresentação.	16	Negócio
ProScenarioCollections	Implementa estruturas de dados específicas para a aplicação, como por exemplo a fila de prioridade de atividades.	1	Negócio
ProScenarioMainForm	Contém as classes referentes às interfaces gráficas da aplicação.	16	Apresentação
QueryService	Utilizada pela camada de apresentação para simular o funcionamento do serviço que fará comunicação com o banco de dados.	1	Negócio
RandomProscenario	Implementa geradores de números aleatórios, atualmente contém o	2	Negócio

³ Recurso contido no *.NET framework* e presente na linguagem C# para o desenvolvimento de formulários em ambientes Windows.

	Mersenne Twister.		
Run	Realiza testes das heurísticas e da aplicação em modo console.	3	*
Schedule	Contém uma série de utilitários para métodos de geração de cronograma.	2	Negócio
Security	Implementa a segurança da aplicação, como criptografia, checagem de credenciais, etc.	1	Negócio
SecurityService	Utilizada pela camada de apresentação para simular o funcionamento do serviço que fará a segurança da aplicação.	1	Negócio
Solvers	Implementa os principais métodos para geração de cronograma, como o PERT/CPM e SGS Serial.	6	Negócio
WCFAccount	Contém as classes responsáveis pela hospedagem e instalação do serviço	2	Negócio
WCFLLogin	Contém as classes responsáveis pela hospedagem e instalação do serviço que faz o <i>login</i> do usuário.	2	Negócio
WCFFProjects	Contém as classes responsáveis pela hospedagem e instalação do serviço que faz o CRUD dos projetos do usuário.	2	Negócio
XML_Handler	Implementa a serialização e desserialização dos dados.	2	Negócio
Total de classes implementadas (nas três camadas)			116

As Figuras 2, 3 e 4 são exemplos de algumas das interfaces implementadas no protótipo. A Figura 2 é referente à construção da estrutura analítica do projeto, onde é possível observar a duração estimada do mesmo através dos rótulos dos nós. Já a Figura 3 mostra a interface responsável pelo cadastro das dependências entre as atividades. Por fim, a Figura 4 exemplifica o gráfico de Gantt gerado a partir das relações entre as atividades, no qual pode-se observar o cronograma PERT/CPM calculado.

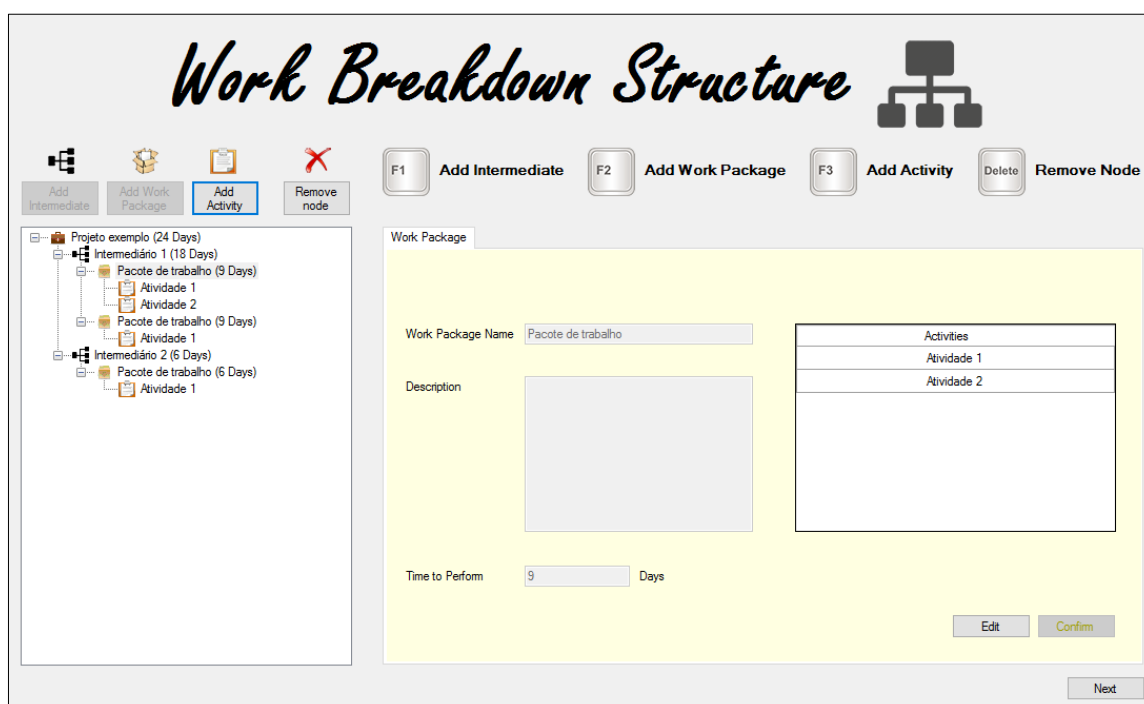


Figura 2 - Interface para construção da WBS

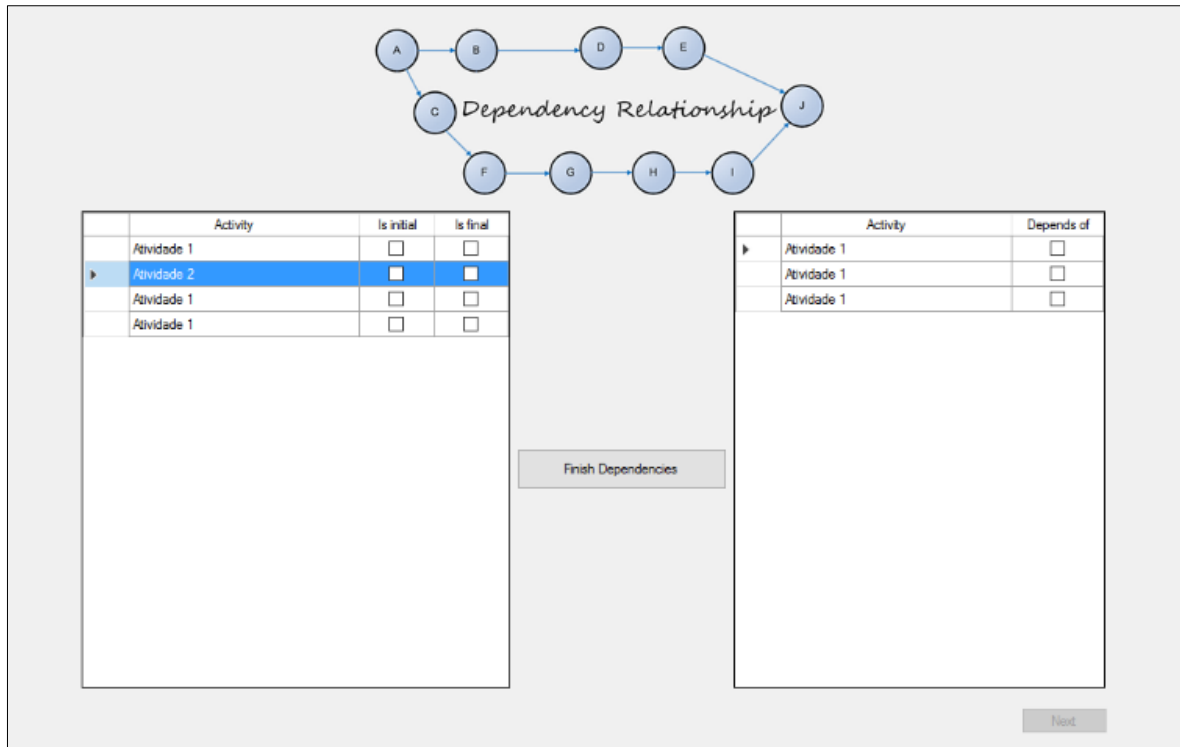


Figura 3 - Interface para cadastro das dependências entre as atividades

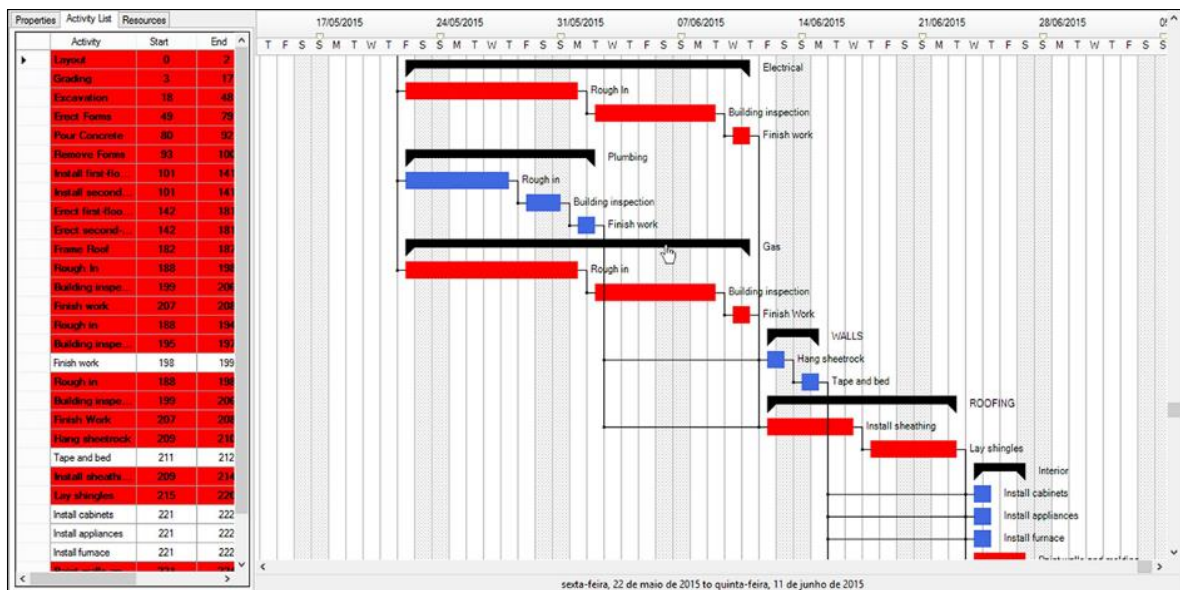


Figura 4 - Interface para visualização do gráfico de Gantt

Além destas interfaces, outras já foram implementadas, contemplando funcionalidade como: *login*, recuperação de credenciais, cadastro de usuários, gerenciamento de projetos e gerenciamento de recursos. Além disso, algumas funcionalidades já foram convertidas em serviços, como por exemplo o *login* e algumas operações básicas realizadas no banco de dados, como busca e inserção de novos usuários e projetos. A integração das interfaces com os serviços encontra-se em fase de testes, estando parcialmente implementadas.

CONCLUSÕES

De acordo com os resultados apresentados, boa parte das camadas de apresentação, regra de negócio e acesso aos dados da aplicação já foram desenvolvidas. Até o término do projeto será desenvolvido o restante do módulo *WebServer* (ver Figura 1), fazendo a total separação entre as camadas cliente-servidor. Outra pendência que está sendo endereçada consiste na integração das heurísticas no módulo de decisão com a aplicação ProScenario, a qual consumirá os serviços responsáveis pela construção e simulação de cenários restritos, gerando cronogramas que minimizem o tempo total do projeto e ao mesmo tempo respeitem as dependências entre as atividades e as restrições de recursos impostas.

Espera-se com a finalização da aplicação disponibilizar a comunidade de gerentes de projetos uma ferramenta mais completa para lidar com o planejamento de projetos com limitação de recursos, realidade comum em empresas de pequeno porte orientadas a serviços.

REFERÊNCIA BIBLIOGRÁFICA

FOWLER, M. *et al.* **Padrões de Arquitetura de Aplicações Corporativas**. Pearson Education 2003.

HURWITZ, J. *et al.* **Service Oriented Architecture For Dummies**. 2.ed. Wiley Publishing, 2009.

ICHIHARA, J. A. O Problema da Programação de Projetos com Restrição de Recursos (Resource-Constrained Project Scheduling Problem). **XXII Encontro Nacional de Engenharia de Produção**, Curitiba, out. 2002.

KLEIN, R. **Scheduling of Resource-Constrained Projects**. Springer, 2000.

JOHNSON, B. **Professional Visual Studio 2013**. Wrox, 2014.

KOLISCH, R., HARTMANN, S. Heuristic Algorithms for Solving the Resource-Constrained Project Scheduling Problem: Classification and Computational Analysis. *In*: WEGLARZ, J. (Ed.). **Project scheduling: Recent models, Algorithms and Applications**. Kluwer, 1999. cap.7, p.147-178.

LOWY, J. Programming WCF Services: **Mastering WCF and the Azure AppFabric Service Bus**. 3.ed. O'Reilly Media, 2010.

MENDES, J. J. M.; GONÇALVES, J. F.; RESENDE, M. G. C. R. A Random Key Based Genetic Algorithm for the Resource Constrained Project Scheduling Problem. **Computers & Operations Research**, v. 36, n. 1, p. 92-109, 2009.

MUELLER, J. P. **Microsoft ADO.NET Entity Framework Step by Step**. Microsoft Press, 2013.

PMI. **Um Guia do Conhecimento em Gerenciamento de Projetos (Guia PMBOK®)**. 5.ed. Project Management Institute, 2013.

SHARP, J. **Microsoft Visual C# 2013 Step by Step**. Microsoft Press, 2013.

SUTHERLAND, J. **Scrum: The Art of Doing Twice the Work in Half the Time**. Crown Business, 2014.

TIDWELL, D.; SNELL, J.; KULCHENKO, P. **Programming Web Services with SOAP**. O'Reilly Media, 2001.