



INFORMAÇÕES GERAIS DO TRABALHO

Título do Trabalho: Desenvolvimento de um Software de Apoio ao Planejamento de Projetos (ProScenario)

Autor (es): Diego Mello da Silva (orientador), Mário Luiz Rodrigues Oliveira (co-orientador), Raí Caetano de Jesus (bolsista), Natanael Ramos (bolsista)

Palavras-chave: restrição de recursos, planejamento de projetos, escalonamento de projetos, RCPSP, heurísticas

Campus: Formiga

Área do Conhecimento (CNPq): 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

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 implementado 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 (EAP), cadastro de dependências entre as atividades e geração do gráfico de Gantt.



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 Project¹, Wrike², ProWorkflow³, Replicon⁴, Project Builder⁵, Artia⁶, Clarizen⁷, TeamWork⁸, DeskAway⁹, Zoho Projects¹⁰, ScrumHalf¹¹ 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 Gantt¹² 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 é capaz de calcular cronogramas para o RCPSP por meio de heurísticas eficientes em termos de complexidade computacional. Tal software contém funcionalidades que permitem ao gestor organizar as atividades do projeto em uma estrutura analítica hierárquica, determinar as precedências existentes entre as atividades, especificar e alocar recursos e custos por atividade, simular o cronograma por meio de metaheurísticas e escaloná-lo com o auxílio do algoritmo *Schedule Generator Scheme* (SGS). A arquitetura da solução prevê que o *software* seja hospedado como um servidor de aplicação, disponível na internet e acessado por gerentes de projeto de qualquer lugar através do módulo cliente. A próxima seção apresentará as etapas executadas para o desenvolvimento do

¹ <https://products.office.com/pt-br/project/project-and-portfolio-management-software>

² <https://www.wrike.com/>

³ <http://www.proworkflow.com/>

⁴ <http://replicon.com/>

⁵ <http://projectbuilder.com.br/>

⁶ <http://artia.com/>

⁷ <http://www.clarizen.com/>

⁸ <https://www.teamwork.com/>

⁹ <http://www.deskaway.com/>

¹⁰ <http://www.zoho.com/projects/>

¹¹ <http://myscrumhalf.com/>

¹² Linha do tempo que mostra quando cada atividade começa, por quanto tempo ela continua, e quando termina.



aplicativo durante a vigência do projeto. Discussões sobre as heurísticas como motor de decisão do *software* usadas estão fora do escopo deste resumo.

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 ORM¹³ 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*. A Figura 1 ilustra sob forma de mapa mental os módulos 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*.

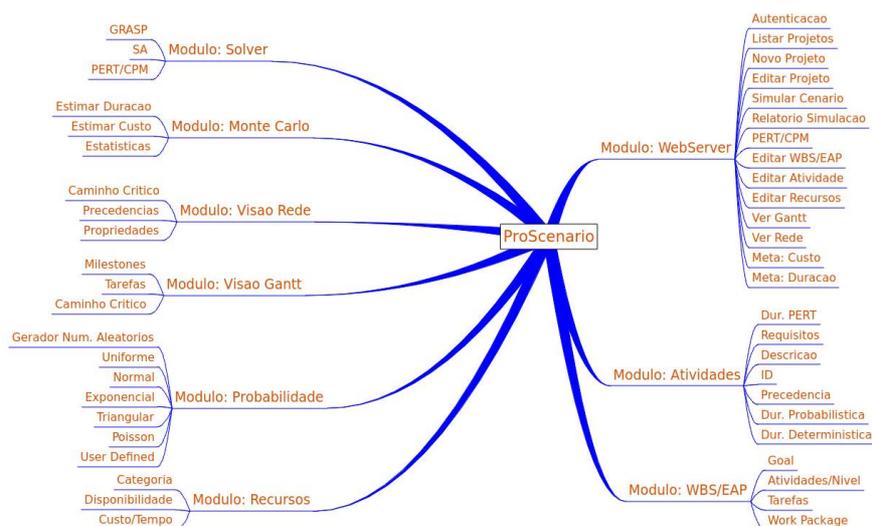


Figura 1 - Mapa Mental com os módulos previstos para o aplicativo ProScenário

¹³ Mapeamento objeto-relacional (ou ORM, do inglês *Object-Relational Mapping*). Permite aos desenvolvedores trabalhar com dados relacionais usando classes e objetos específicos de domínio.



O aplicativo ProScenario 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). 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 ProScenario utiliza o banco de dados PostgreSQL¹⁴, 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 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 protótipo de aplicação *cliente* do aplicativo foi desenvolvido com 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*¹⁵. 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*¹⁶. A aplicação server foi desenvolvida em linguagem C#, e comunica-se com a aplicação cliente através de arquitetura orientada a serviços (SOA). Dentre os serviços implementados destacam-se AccountService, que contém contratos e implementação do serviço responsável pelo gerenciamento de contas de usuários; o LoginService, que contém contratos e implementação do serviço de login; o ProjectService, que contém contratos e implementação do serviço de gerenciamento de projetos do usuário; por fim o serviço ScenarioService, que contém contratos e implementação do serviço responsável pela simulação do cronograma com base nas metaheurísticas para o RCPS e escalonamento SGS. A aplicação é hospedada como um serviço do Windows.

A aplicação *server-side* foi hospedada durante a fase de testes e validação 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 são simulados para o mesmo projeto.

RESULTADOS E DISCUSSÕES:

¹⁴ <http://www.postgresql.org/>

¹⁵ Recurso do *.NET framework* para o desenvolvimento de formulários em ambientes Windows.

¹⁶ <http://gantchart.codeplex.com/>



Até o presente momento, a aplicação em desenvolvimento conta com 26 soluções do Visual Studio (aproximadamente 12517 linhas de código em 118 arquivos diferentes). A Tabela 1 resume as soluções implementadas, descrevendo seu papel na aplicação bem como a quantidade de arquivos implementados e a qual camada pertence.

Tabela 1 - Resumo das soluções contidas na aplicação

Solução	Descrição	Arquivos	Linhas	Camada
AccountService	Contém os contratos e implementação do serviço responsável pelo gerenciamento de contas.	2	52	Negócio
DB	Contém os métodos para persistência dos dados (utilizando o ORM Entity).	1	1309	Dados
Distribution	Implementa as distribuições probabilísticas para geração de variáveis aleatórias, utilizadas pela aplicação no módulo de simulação.	9	330	Negócio
ExceptionHandler	Contém classes que manipulam exceções geradas pela aplicação.	3	53	Negócio
Heuristics	Gerencia as características e comportamentos das heurísticas utilizadas no escalonamento das atividades.	8	1144	Negócio
Interfaces	Contém interfaces para uso geral na aplicação.	4	61	Apresentação
HumanReadable	Faz conversões de datas em formato discreto para formato de data. Ex: DD/MM/YYYY	1	112	Negócio
LoginService	Contém os contratos e implementação do serviço de <i>login</i> .	2	44	Negócio
PriorityRules	Implementa as heurísticas construtivas baseadas em regras de prioridade.	16	725	Negócio
ProjectService	Contém os contratos e implementação do serviço responsável pelo gerenciamento de projetos do usuário.	2	116	Negócio
ProScenario	Contém as classes que representam os objetos de um projeto. Utilizada pela camada de apresentação.	22	2365	Negócio
ProScenarioCollecti ons	Implementa estruturas de dados específicas para a aplicação, como por exemplo a fila de prioridade de atividades.	2	227	Negócio
ProScenarioMainFo rm	Contém as classes referentes às interfaces gráficas da aplicação.	15	3006	Apresentação
PSP2Proscenari o	Converte instâncias da biblioteca PSPLib para formato compatível com a aplicações.	1	141	Negócio
Random	Implementa geradores de números aleatórios, atualmente contém o Mersenne Twister.	2	227	Negócio
Run	Realiza testes das heurísticas e da aplicação em modo console.	3	92	*
Schedule	Contém uma série de utilitários para métodos de geração de cronograma.	2	69	Negócio
ScenarioService	Contém os contratos e implementação do serviço responsável pela geração e escalonamento do cronograma.	2	135	Negócio
Security	Implementa a segurança da aplicação, como criptografia, checagem de credenciais, etc.	1	95	Negócio
Solvers	Implementa os principais métodos para geração de cronograma, como o PERT/CPM e SGS Serial.	6	854	Negócio
WCFAccount	Contém as classes responsáveis pela hospedagem e instalação do serviço	3	141	Negócio
WCFLogin	Contém as classes responsáveis pela hospedagem e instalação do serviço que faz o <i>login</i> do usuário.	3	143	Negócio
WCFProjects	Contém as classes responsáveis pela hospedagem e instalação do serviço que faz o CRUD dos projetos do usuário.	3	153	Negócio
WCFScenarios	Contém as classes responsáveis pela instalação e hospedagem do serviço ScenariosService.	3	155	Negócio
XML_Handler	Implementa a serialização e desserialização dos dados.	2	768	Negócio
TOTAL		118	12517	

A Figura 2 exemplifica a interface de saída da aplicação apresentando o gráfico de Gantt gerado a partir das relações entre as atividades, no qual pode-se observar o cronograma PERT/CPM calculado.

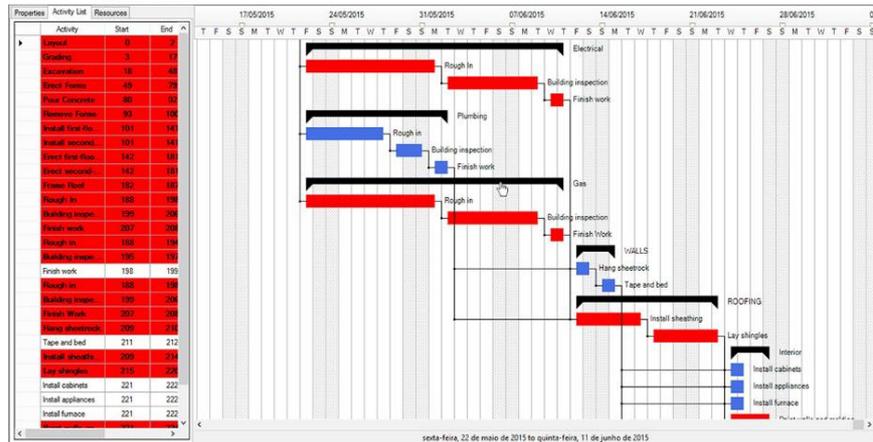


Figura 2 - Interface para visualização do gráfico de Gantt

Além destas interfaces, outras foram implementadas, contemplando funcionalidade como: *login*, recuperação de credenciais, cadastro de usuários, gerenciamento de projetos e gerenciamento de recursos. Além disso, funcionalidades 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.

CONCLUSÕES:

De acordo com os resultados apresentados, muito do previsto nas camadas de apresentação, regra de negócio e acesso aos dados da aplicação foram desenvolvidas no decorrer do projeto. O *software* ProScenario buscou resolver o problema de escalonamento com recursos restritos de forma a beneficiar toda a comunidade de gerentes de projetos, em especial aqueles que lidam com uma quantidade restrita de recursos tal como é o caso de empresas de pequeno porte orientadas a serviços. Novos contratos podem ser negociados considerando (i) o prazo factível de entrega dos serviços, sem atrasos e prejuízo com multas e (ii) o custo real associado com o projeto considerando as limitações de sua força de trabalho, como por exemplo, em tamanho de equipe. O software ProScenario contém muitas funcionalidades importantes para planejar projetos restritos, como é o caso do planejamento de atividades utilizando EAP, escalonamento de projetos com PERT/CPM, escalonamento de projetos restritos com heurísticas para o RCPSP, entre outros. Muitas funcionalidades de interesse não foram desenvolvidas, no entanto, em virtude de prazo e complexidade de implementação. Isto posto, sugere-se como trabalhos futuros a implementação das seguintes ações para tornar o ProScenario mais completo:

- No módulo WebServer: criação do serviço de simulação de Monte Carlo e hospedagem dos serviços em domínio público.
- No módulo Probabilidade e Monte Carlo: implementação da simulação de Monte Carlo para geração de diferentes cenários de cronogramas restritos, e não apenas em cronogramas PERT;
- Na interface gráfica: implementar a interface que exibirá todos os dados gerados pela simulação de Monte Carlo sobre cronogramas restritos;
- Na validação da implementação e hospedagem da aplicação: criação e execução de testes estruturados para validação da aplicação e, uma vez validada, realizar a hospedagem desta em domínio público.



- Na implementação da interface gráfica para exibição da rede PERT: criação de uma interface que exiba as atividades e suas respectivas relações de precedência sob a forma de diagrama.
- Na implementação da camada de segurança nos serviços: atualmente os serviços implementados compartilham dados na forma de texto plano. Para garantir a segurança dos dados na comunicação cliente-servidor, deve-se realizar a implementação de uma camada de segurança para criptografar os dados durante a comunicação.
- Na mensuração da performance das metaheurísticas GRASP e Simulated Annealing: implementaram-se versões das metaheurísticas citadas para execução no ambiente multithreading. Assim, faz-se necessária a análise de viabilidade de uso destas versões, levando em conta a qualidade das soluções encontradas e o tempo de execução.

REFERÊNCIAS BIBLIOGRÁFICAS:

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.

Participação em Congressos, publicações e/ou pedidos de proteção intelectual:

O projeto foi apresentado no V Jornada de Educação, Ciência e Tecnologia do IFMG Campus Formiga (Nov2014) sob a forma de painel.