Brazilian Journal of Development

Estudo de caso no contexto da Engenharia de Software: SGCOPEX

Case study in the context of Software Engineering: SGCOPEX

Recebimento dos originais: 20/12/2018
Aceitação para publicação: 21/01/2019

José Damião de Melo
Mestre em Modelagem Computacional de Conhecimento pela Universidade Federal de Alagoas
Instituição: Instituto Federal de Educação, Ciência e Tecnologia de Sergipe
Endereço: Av. Padre Airton Gonçalves Lima, 1140 - São Cristóvão, Itabaiana - SE, Brasil
E-mail: damiaomelo@gmail.com

Mikaele Costa Mendonça
Técnica em Manutenção e Suporte em Informática pelo Instituto Federal de Sergipe
Instituição: Instituto Federal de Educação, Ciência e Tecnologia de Sergipe
Endereço: Av. Padre Airton Gonçalves Lima, 1140 - São Cristóvão, Itabaiana - SE, Brasil
E-mail: mikaelemendonca@gmail.com

Wanderson Roger Azevedo Dias
Pós-Doutor em Computação Ciência da Computação pela Universidade Federal de Sergipe
Instituição: Instituto Federal de Educação, Ciência e Tecnologia de Sergipe
Endereço: Av. Padre Airton Gonçalves Lima, 1140 - São Cristóvão, Itabaiana - SE, Brasil
E-mail: wradias@gmail.com

RESUMO
No contexto deste artigo, os métodos da engenharia de software foram o framework para o desenvolvimento de um protótipo de sistema web responsável pelo gerenciamento das atividades de Pesquisa e Extensão em um Instituto Federal. Para o desenvolvimento da pesquisa foi usado o modelo de processo incremental. O produto foi colocado on-line em em versão Alfa e validado pelo especialista do domínio quanto aos requisitos e pelo usuário gestor quanto a usabilidade e eficácia, sua adoção em nível institucional poderá ampliar a capacidade gerencial da atividade de pesquisa e extensão da instituição alvo.

Palavras-Chave: Modelagem.Prototipagem. Processo de Desenvolvimento. Engenharia de Software.

ABSTRACT
In the context of this article, software engineering methods were the framework for the development of a prototype web system responsible for the management of research and extension activities at a Federal Institute. For the development of the research the incremental process model was used. The product was put online in an Alpha version and validated by the domain expert regarding the requirements and the user manager regarding usability and effectiveness, its adoption at the institutional level could increase the managerial capacity of the research activity and extension of the target institution.

Keywords: Modeling. Prototyping. Development Process. Software Engineering.

1 INTRODUÇÃO

Brazilian Journal of Development

A engenharia de software é essencial no desenvolvimento de sistemas ao possibilitar que o analista se aproprie e utilize metodologias assertivas para projetos de software. Se insere no contexto das pesquisa para o desenvolvimento do SGCOPEX (Sistema de Gerenciamento da Coordenadoria de Pesquisa e Extensão) ao possibilitar a automatização do gerenciamento dos projetos vinculados as Coordenadorias de Pesquisa e Extensão (COPEX) do Instituto Federal de Sergipe (IFS).

Foi intenção deste projeto que o SGCOPEX não somente se apresente como um software responsável pelo gerenciamento de projetos de pesquisa e extensão mas também possa vir a se tornar parte de um processo gerencial escalável, que possa evoluir gradual e permanentemente. É norteado segundo metodologias preconizadas pela engenharia de software, as quais permitem uma melhor organização do processo de desenvolvimento, foi condição da ação ainda obter um resultado final de alta qualidade a um baixo custo e de acordo com os requisitos fornecidos pelo cliente.

Desta forma, o objetivo deste artigo é apresentar o relato do estudo deste caso de desenvolvimento de uma aplicação web, seguindo os preceitos da engenharia de software, destacando o processo, métodos e ferramentas utilizadas na modelagem e desenvolvimento do SGCOPEX.

O artigo está organizado da seguinte forma: a Seção 2 apresenta uma revisão teórica, afim, de ambientar o estudo corrente, destacando a Engenharia de Software; Modelos de Processos; Prototipação de Software; Linguagem de Modelagem Unificada e a Modelagem Conceitual; A Seção 3 apresenta a modelagem conceitual e a estratégia de percurso para o desenvolvimento do SGCOPEX e a Seção 4 apresenta as conclusões e possibilidades para trabalhos futuros.

2 REVISÃO TEÓRICA

Segundo Pressman (2011) “um software consiste em: (i) instruções (programas de computador) que quando executados fornecem características, funções e desempenhos desejados; (ii) estrutura de dados que possibilitam aos programas manipular informações adequadamente; e (iii) informações descritivas as quais descrevem a operação e o uso dos programas”.

As aplicações/softwares para a Web já foram somente um conjunto de arquivos de texto interconectados que apresentavam informações quase sempre através de texto e raramente informações gráficas, ainda assim, de forma limitada. Todavia, após o aparecimento da Web 2.0, elas se tornaram sofisticados ambientes computacionais que não apenas fornecem recursos especializados, funções computacionais e conteúdo para o usuário final, como também estão integradas a banco de dados coorporativos e aplicações comerciais [Pressman, 2011].

Já Sommerville (2003) afirma que a engenharia de software é responsável por todos os aspectos da produção de software, iniciando na especificação de sistema até a manutenção desse sistema depois que ele já está em operação. É uma disciplina que tem como meta apresentar técnicas de desenvolvimento de softwares com alta qualidade a um baixo custo.

Na visão de Pressman (2011), a engenharia de software é formada por processo, métodos (práticas) e ferramentas as quais permitem construir sistemas dentro do prazo e com qualidade (ver Figura 1).

A engenharia de software é, portanto, o conjunto de processos, métodos e feramentas que podem ser representadas na forma de camadas (Figura 1), comprometida com a qualidade.

Com o aperfeiçoamento contínuo de processos é possível obter abordagens cada mais efetivas. Na engenharia de software a camada fundamental é a de processos, pois é ela que possibilita o desenvolvimento de software de forma racional e dentro do prazo, mantendo sempre o foco na qualidade.

O processo de software é a base para o controle do gerenciamento de projetos, no qual estabelece o contexto onde serão aplicados os métodos técnicos que serão produzidos os produtos derivados do software, tais como: (i) modelos dos documentos; (ii) dados e etc. Também são estabelecidos marcos, a qualidade é garantida e as mudanças são geridas de forma apropriada. Além disso, a metodologia a ser desenvolvida para a entrega efetiva de tecnologia de engenharia de software também é definida na etapa do processo (Pressman, 2011; Sommervile, 2003).

2.1 FERRAMENTAS

Tem como função fornecer apoio automatizado ou semi-automatizado aos métodos e aos processos. As ferramentas nos permite visualizar os detalhes de “como fazer” na construção de um software. Atualmente existem diversas ferramentas para sustentar cada método, conforme (Pressman, 2011; Rezende, 2005), tais como: (i) starUML (StarUML, 2015); (ii) DBDesigner (FabForce.net, 2015); (iii) PHP (PHP, 2015) dentre outras.

3 MÉTODOS

São responsáveis por fornecerem os detalhes técnicos para a construção do software, ou seja, o roteiro do desenvolvimento do software. Os métodos da engenharia de software são baseados num conjunto de princípios básicos que regem as áreas da engenharia de software, especificamente para o desenvolvimento e contém as atividades de modelagem e outras técnicas descritivas, conforme (PRESSMAN, 2011).

De acordo com Sommerville (2003), “um método na engenharia de software é uma abordagem estruturada para o desenvolvimento de software, cujo objetivo é facilitar a produção de software de alta qualidade, apresentando uma boa relação custo-benefício”.

3.1 PROCESSO DE SOFTWARE

Um processo de software corresponde a um conjunto de passos formados por atividades, métodos, práticas e transformações parcialmente ordenados, os quais são usados para desenvolver e manter software e produtos associados (Sommerville, 2003).

Conforme Pressman (2011), processo de software é um conjunto de atividades de trabalho, ações e tarefas realizadas quando algum artefato de software deve ser criado. O processo não é uma prescrição rígida de como desenvolver um software. Ao contrário, é uma abordagem adaptável que possibilita a equipe de software realizar o trabalho de selecionar e escolher o conjunto apropriado de ações e tarefas.

As atividades, ações e tarefas são alocadas dentro de uma modelo ou metodologia o qual determina seu relacionamento com o processo e seu relacionamento entre elas. A metodologia de processo fornece a base para um processo de engenharia de software completo, através de um pequeno número de atividades estruturais e um conjunto de atividades de apoio que são aplicadas a todo o processo. Essas atividades são aplicáveis a todos os projetos de software, independente de complexidade ou tamanho (Pressman, 2011).

3.2 MODELOS DE PROCESSOS EM ENGENHARIA DE SOFTWARE

O modelo de processo de software é uma representação abstrata do processo. Ele é responsável por descrever de forma simplificada um processo de software, o qual é apresentado a partir de uma perspectiva especifica. (Sommerville, 2003).

Analisando os modelos de processo existentes e citados acima, foi definido para o desenvolvimento do sistema web SGCOPEX o modelo de desenvolvimento incremental, parte de um conjunto de requisitos iniciais, e tendo-os como base, é projetada a arquitetura completa do sistema.

Através dessa arquitetura e da agregação dos demais requisitos, o sistema vai sendo desenvolvido, indicando este modelo como ideal quando se deseja fornecer rapidamente um determinado conjunto funcional aos usuários para após esse fornecimento, refinar e expandir sua funcionalidade em versões de software posteriores. Assim, o modelo incremental aplica-se de forma linear e escalonada ao decorrer do tempo, onde cada sequência linear é responsável por gerar um incremento (Pressman, 2011; Garcia, 2013).

Brazilian Journal of Development

Num processo de desenvolvimento incremental, os interessados (clientes) identificam através de um esboço as funções a serem fornecidas pelo sistema. Depois é definida uma série de estágios de entrega. Cada estágio fornece um subconjunto de funções do sistema. Após a identificação dos incrementos, os requisitos para as funções que serão entregues no primeiro incremento são definidos em detalhes e o mesmo é desenvolvido utilizando o processo de desenvolvimento mais apropriado (Sommerville, 2003).

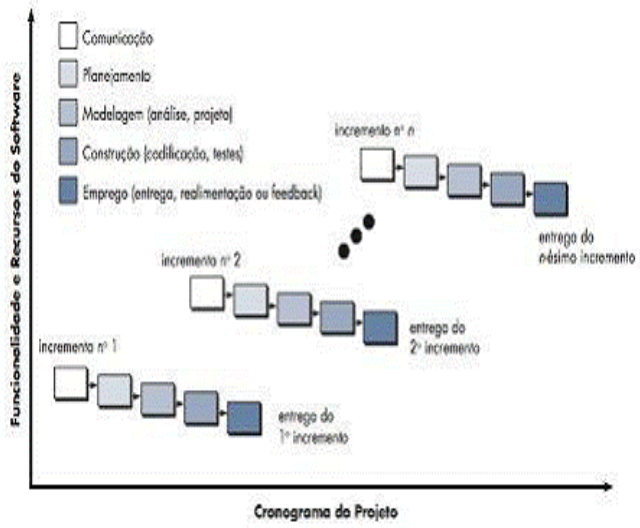


Figura 1. Processo - Modelo de Desenvolvimento Incremental (Pressman, 2011).

3.3 PROTOTIPAÇÃO DE SOFTWARE

É uma versão inicial do sistema a qual é utilizada para conhecer mais sobre os problemas e suas possíveis soluções. A prototipação permite que os usuários melhorem a especificação dos requisitos. Além desse benefício, desenvolver um protótipo de sistema pode ter outras vantagens, tais como:

Possíveis equívocos podem ser identificados à medida que as funções do sistema são apresentadas;

A equipe de desenvolvimento pode encontrar requisitos incompletos e/ou inconsistentes desenvolvendo o protótipo;

Um sistema em operação, embora limitado é capaz de mostrar a viabilidade e a utilidade da aplicação para a gerência.

A prototipação evolucionária inicia com um sistema relativamente simples que é a implementação dos requisitos mais importantes para o usuário, fornecendo assim um sistema funcional aos usuários finais. Esse sistema será alterado e ampliado assim que novos requisitos vão sendo descobertos. Dessa forma, ele se torna o sistema requerido.

Para a engenharia de software a prototipação é essencial no processo do projeto da interface. Pois, as interfaces possuem uma natureza dinâmica e as descrições textuais e os diagramas não são

bons para exprimir os requisitos da interface com o usuário. Por conseguinte, a prototipação evolucionária com o envolvimento do usuário final é uma maneira sensata de desenvolver interfaces gráficas com o usuário destinadas a sistemas de software.

3.4 UML (LINGUAGEM DE MODELAGEM UNIFICADA)

A UML segundo Fowler (2004), é uma família de notações gráficas, apoiadas por um metamodelo único, que ajuda na descrição e no projeto de sistemas de software, particularmente daqueles construídos utilizando o estilo Orientado a Objeto (OO).

Já para Melo (2010), é uma linguagem para especificação, visualização, construção e documentação de artefatos de sistemas de software.

Os criadores da UML, Booch, Rumbaugh e Jacobson (2005) acrescentam: "A UML proporciona uma forma padrão para a preparação de planos de arquitetura de projetos de sistemas, incluindo aspectos conceituais tais como: processos de negócio e funções do sistema, além de itens concretos como as classes escritas em determinada linguagem de programação, esquemas de banco de dados e componentes de software reutilizáveis".

A UML é uma linguagem independente de processo, sendo assim pode ser usada em qualquer tipo de processo. Com sua estrutura é possível conduzir a criação e leitura de seus modelos, mas não determina quais ou quando esses modelos devem ser criados, pois essa é uma responsabilidade do processo de desenvolvimento (MELO, 2010).

Um diagrama de caso de uso, no contexto a UML, fornece uma visão geral de todos os casos de uso e como estão relacionados, ou seja, uma visão geral das funcionalidades do sistema (PRESSMAN, 2011).

Os casos de uso ajudam no entendimento dos requisitos funcionais de um sistema. Porque ele mostra a interação entre os usuários e o sistema definindo os passos necessários para atingir um objetivo especifico (FOWLER, 2004; PRESSMAN, 2011).

Segundo Melo (2010), um caso de uso é descrito, de forma compreensível tanto para a equipe de desenvolvimento como para os clientes, como sendo uma sequência de ações as quais representam um cenário principal "perfeito" e cenários alternativos, os quais tem como objetivo demonstrar o comportamento de um sistema ou parte dele através de interações com os atores.

Os atores são responsáveis por executar os casos de usos que estão representados graficamente através de formas ovais e estão conectados aos atores ou a outros casos de uso através de linhas, ou seja, o relacionamento (PRESSMAN, 2010).

O diagrama de classe é provavelmente o mais utilizado e um dos mais importantes da UML. É responsável por definir a estrutura das classes utilizadas pelo sistema, determinando os atributos e

os métodos que cada classe possui. Além de estabelecer como as classes se relacionam e trocam informações entre si (GUEDES, 2011).

Os diagramas de classe são usados para modelar a estrutura estática de um sistema. Estes modelos são designados por "vista do desenho estático do sistema" (Silva e Videira, 2001).

Os elementos que constituem um diagrama de classe são:

Classe: é a descrição de um objetos que partilham os mesmo atributos, operações, relações e a mesma semântica. Uma classe pode ser algo tangível ou uma abstração do engenheiro de software;

Relações: é responsável pela ligação entre os elementos e é representado graficamente por um determinado tipo de linha. As mais importantes são dependência, generalização e associação, nas quais estão descritas a seguir:

Relação de dependência: indica que a alteração em um elemento pode afetar outro elemento que a usa, mas não necessariamente o oposto. É representado por uma linha tracejada;

Relação de generalização: é uma relação entre um elemento geral e um elemento mais específico. É representado por uma linha dirigida com um triângulo na ponta;

Relação de associação: é uma relação estrutural que especifica que objetos de uma classe estão ligados a objetos de outra classe. É representado por uma linha cheia.

Multiplicidade: é o número de instâncias em que uma classe pode se relacionar (através de associação) com uma única instância da(s) outras(s) classe(s) participantes(s);

Instâncias: é uma manifestação concreta de uma abstração. Sendo um conjunto de operações que pode ser aplicado e que tem um estado que registra os efeitos das operações realizadas;

Objeto: é uma instância de uma classe, onde é herdada todos os atributos e métodos definidos na própria classe e possui uma representação de execução própria, a qual pode-se designar como estado;

Operações: os objetos podem efetuar operações definidas nas suas classes;

Estado: o estado de um objeto é dado pelos valores assumidos pelo conjunto de atributos de um objeto.

3.5 MODELAGEM CONCEITUAL

É uma representação e/ou uma descrição da realidade do ambiente do problema, constituindo uma visão global dos principais dados e relacionamentos (estruturas de informação).

Ao utilizar a modelagem conceitual com a técnica de Entidade-Relacionamento, obtém-se os esquemas conceituais sobre o negócio para o qual o projeto está sendo desenvolvido.

Brazilian Journal of Development

A abordagem Entidade-Relacionamento representa o ponto central no projeto conceitual de um sistema. O objetivo da modelagem de dados é representar de forma única, não redundante e resumida os dados de uma aplicação em banco de dados.

4 MODELAGEM E DESENVOLVIMENTO DO SGCOPEX

Para o desenvolvimento do sistema *web* SGCOPEX (Sistema de Gerenciamento da Coordenadoria de Pesquisa e Extensão) foi utilizado o Modelo de Desenvolvimento Incremental. Este modelo foi escolhido devido ao objetivo de fornecer um conjunto funcional, ou seja, um protótipo, ao cliente (coordenador da COPEX) que após análise e validação do SGCOPEX, permitisse o refinamento e a expansão de suas funcionalidades em cada interação. A seguir, são descritas em detalhes as fases que compuseram o desenvolvimento do sistema SGCOPEX.

4.1 ESPECIFICAÇÃO

Para a extração dos requisitos foi utilizada a técnica de entrevista, aplicada ao ex-coordenador da COPEX, o qual forneceu os principais requisitos que o sistema deveria conter. Os requisitos obtidos com a entrevista foram documentados com uma ferramenta de edição de texto e passaram a fazer parte da documentação do projeto.

4.2 PROJETO

Na elaboração do projeto, foram utilizados dois diagramas UML, sendo um comportamental e outro estrutural, representados respectivamente pelos diagramas de caso de uso e de classe .

Vale ressaltar que o diagrama de caso de uso fornece uma descrição das funções do software através dos casos de uso e mostra a relação deles com os atores que os executam. Já o diagrama de classe, fornece a estrutura que o sistema deve possuir através das classes, dos atributos que elas contém e dos métodos que executam.

Além disso, são estabelecidos os relacionamentos entre elas para mostrar como deve ocorrer as trocas das informações no sistema. A Tabela 1 apresenta todos os casos de uso do SGCOPEX.

As funções representadas pelos casos de uso no diagrama, estão todas contidas no documento de projetos dos requisitos do sistema web SGCOPEX.

Tipo do Caso de Uso	**Agente**
Efetuar Login	Usuário
Cadastrar	Usuário
	Projeto

Brazilian Journal of Development

	Edital
Consultar	Usuário
	Projeto
	Edital
Alterar	Usuário
	Projeto
	Edital
Excluir	Usuário
	Projeto
	Edital

Tabela 1. Tabela dos Casos de Uso do sistema (elaborado pelos autores).

A Tabela 2 apresenta as classes e métodos contidos na modelagem do SGCOPEX.

Classes	**Métodos**
Usuário	
Docente	Incluir Usuário
	Alterar Usuário
	Excluir Usuário
	Pesquisar Usuário
	Incluir Projeto
	Alterar Projeto
	Excluir Projeto
	Pesquisar Projeto
	Incluir Edital
	Alterar Edital
	Excluir Edital
	Pesquisar Edital
Discente	Consultar Usuário
	Consultar Projeto
Projeto	
Edital	

Tabela 2. Classes e Métodos do Diagrama de Classe (elaborado pelos autores).

Brazilian Journal of Development

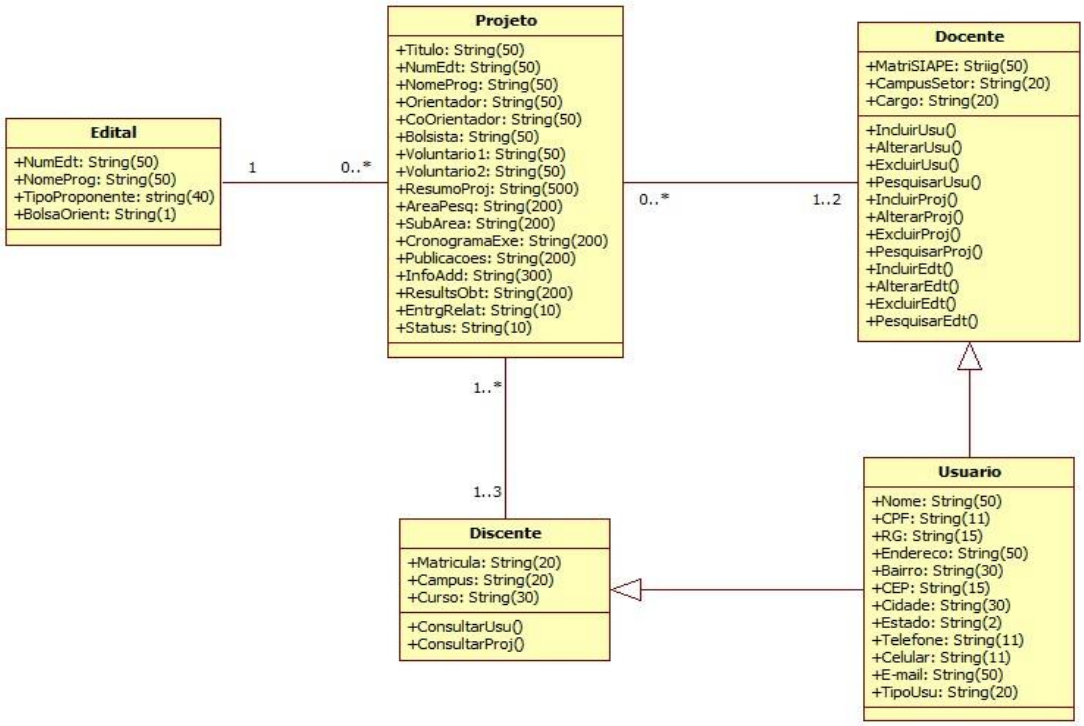


Figura 2. Diagrama de Classes do SGCOPEX

O diagrama de classes é apresentado na figura x elencando todas as classes, atributos, métodos e relacionamentos do sistema SGCOPEX.

Para o projeto do banco de dados também foi consultado o documento de requisitos e o diagrama de classes, para uma melhor visão da estrutura do sistema. Nesta etapa da execução do projeto foi desenvolvido o Diagrama de Entidade-Relacionamento.

Para a criação e modelagem deste diagrama foi utilizado a ferramenta DBDesign [FabForce.net, 2015]. O Diagrama de Entidade-Relacionamento está representado na figura 4., contém as entidades, relacionamentos e todos os campos presentes no banco de dados do sistema SGCOPEX.

Para o projeto da interface do sistema, a princípio foram desenvolvidos dois *layouts*, que foram apresentados ao ex-coordenador da COPEX, no qual após análises pautadas em conceitos de "interface homem-máquina" definiu como o *layout* para o sistema SGCOPEX o modelo exibido na figura 3.

4.3 IMPLEMENTAÇÃO

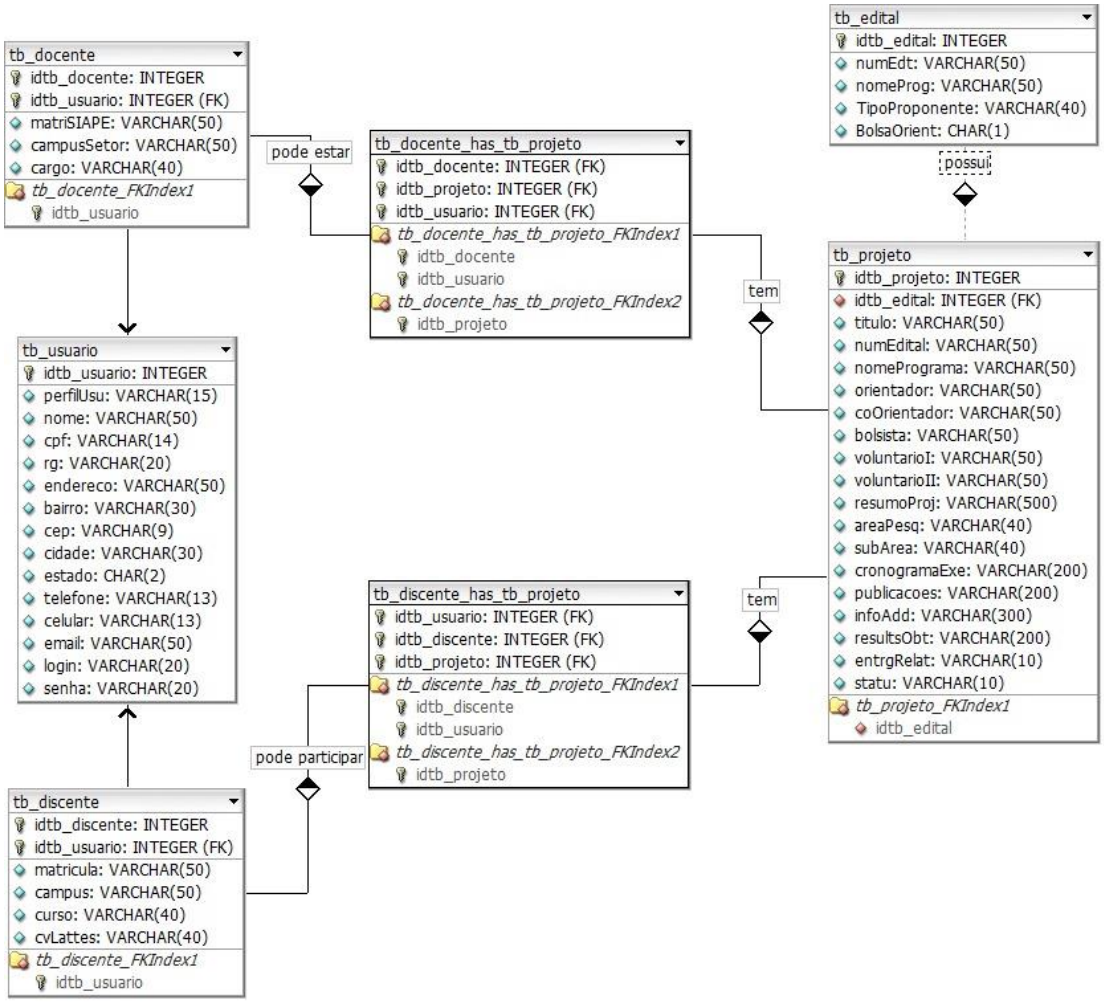
Para a implementação do sistema web SGCOPEX foram utilizadas as seguintes linguagens de programação:

HTML (*HyperText Markup Language*): é uma linguagem que tem como principal característica a possibilidade de se interligar a outros documentos web, uma vez que o hipertexto é o tipo de

conteúdo inserido em um documento para a web (Silva, 2008);

CSS (C*ascading Style Sheet*): são folhas de estilo em cascata que servem para adicionar os estilos aos documentos web (Silva, 2008);

Figura 3 - Layout de interface do SGCOPEX



Brazilian Journal of Development

Figura 4. Diagrama E-R do SGCOPEX

PHP (Hyper*text Preprocessor)*: é uma das linguagens mais utilizadas na web. A principal diferença em relação às outras linguagens é a capacidade que o PHP tem de interagir com o mundo web, transformando totalmente os websites que possuem páginas estáticas (Niederauer, 2011);

JavaScript: é uma linguagem desenvolvida para rodar no lado do cliente, isto é, a interpretação e o funcionamento da linguagem dependem de funcionalidades hospedadas no navegador do usuário. Isso é possível pois existe um interpretador JavaScript hospedado no navegador (Silva, 2010);

SQL (S*tructured Query Language*): é uma linguagem de pesquisa declarativa padrão para banco de dados relacional (base de dados relacional) (Prates e Niederauer, 2005).

A figura 5 apresenta a implementação da página responsável pelo "Cadastro de Usuário" utilizada no sistema SGCOPEX. Para a estilização das páginas, ou seja, o desenvolvimento da interface com o usuário, foram desenvolvidos scripts CSS e linkados as páginas do sistema SGCOPEX. Após a implementação e os scripts CSS, foram obtidas as telas do protótipo para validação da interface juntamente com o usuário validador.

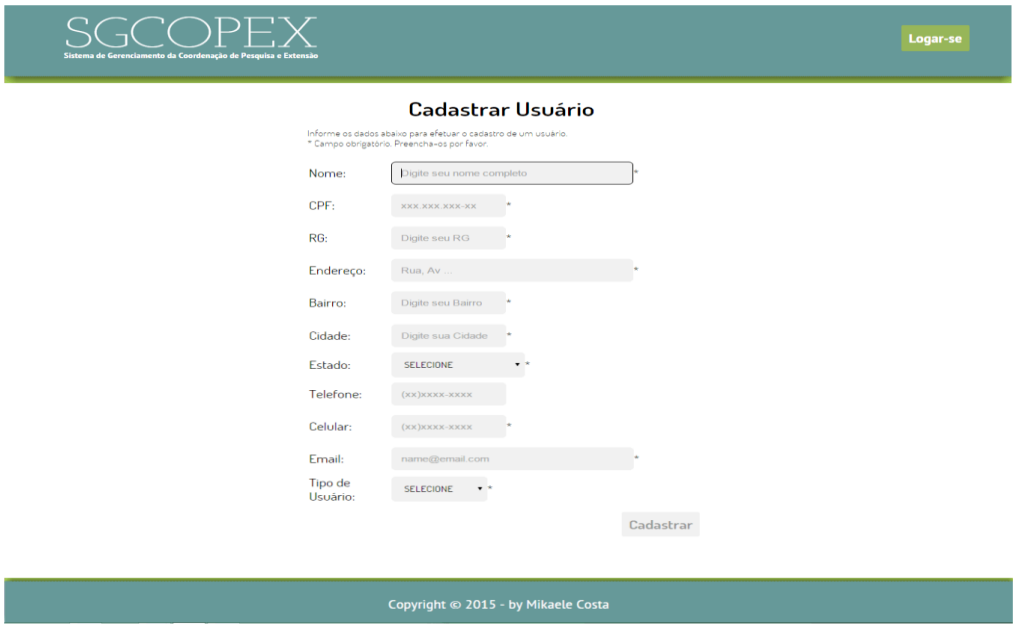


Figura 5. Página Cadastrar usuário do SGCOPEX

4.4 VALIDAÇÃO

Para a validação do protótipo do sistema web SGCOPEX, foi apresentado ao coordenador da COPEX, IFS, campus Itabaiana, com a execução de protocolo de testes diversos a exemplo de seu

próprio cadastro de usuário, *logon* e *logoff* no sistema, além de outros testes de usabilidade, e navegações internas pelo sistema, visando aferir o grau de funcionalidade da solução.

Também foi aplicado ao coordenador da COPEX, um questionário para obter a avaliação dele sobre o protótipo desenvolvido. Com respeito aos dados obtidos no questionário verificamos o grau de utilidade do sistema para a COPEX, em relação ao gerenciamento dos projetos de pesquisa e extensão, analisando assim a possibilidade de resolução de pendências ou dificuldades que os pesquisadores (docentes e/ou bolsistas e voluntários) pudessem ter junto ao(s) projeto(s). Possibilitou ainda uma verificação acerca do acesso quantitativo de projetos pertencentes ao campus e até mesmo a obtenção de meta-dados referentes aos projetos.

Após a etapa de avaliação, entendemos que o sistema, do ponto de vista do especialista do domínio, validador de suas funcionalidades atendeu as necessidades específicas da coordenação, inicialmente definido no escopo do projeto como cliente para fins de controle e definição de requisitos, além de apresentar uma interface e usabilidade compatível com os demais sistemas *web*.

5 CONCLUSÕES E TRABALHOS FUTUROS

Com adoção de métodos, tecnicas e ferramentas da engenharia de software foi possível desenvolver um sistema *web* que atende as necessidades do cliente (COPEX). Por conseguinte, o objetivo deste projeto foi obtido com êxito pois, o protótipo do SGCOPEX foi validado pelo especialista do domínio como sendo um sistema útil e que atende as necessidade da Coordenadoria de Pesquisa e Extensão do Instituto Federal de Sergipe, campus Itabaiana, em particular.

Com a análise das respostas da validação inferimos que o sistema é útil para a COPEX, pois organiza o gerenciamento dos projetos de pesquisa e extensão, agilizando assim as pendências ou dificuldades que os pesquisadores (docentes e/ou bolsistas e voluntários) tenham junto ao(s) projeto(s). Além de permitir acesso rápido a dados estatísticos no quantitativo de projetos gerenciados ou até mesmo na obtenção de dados mais particulares referentes aos mesmos, entre outros benefícios gerados exclusivamente pelo sistema web SGCOPEX, entre eles acesso descentralizado, multi-usuário e plataforma WEB.

Segundo o validador, o sistema atende as necessidades específicas da coordenação, além de apresentar uma interface e usabilidade compatível com os demais sistemas web e também cumpre com o seu objetivo. O especialista do domínio ainda ressaltou da importância do sistema ser web, onde permite estar disponível na internet contribuindo com os possíveis usuários, mesmo quando não estiverem presentes no campus.

Permitindo assim, que seja executado o trabalho da coordenação, independente a localização física.Como trabalhos futuros, sugere-se a implementação de novos módulos, tais como: pesquisas

mais específicas, relatórios mais detalhados incluindo gráficos, além do plano de manutenção contínua do software visando a evolução do sistema para que o mesmo continue atendo as necessidades da COPEX.

Finalmente, demonstra-se que o uso dos preceitos da engenharia de software permitiu desenvolver um sistema web que atendeu o escopo inicial das necessidades do cliente e por conseguinte, o objetivo deste projeto de pesquisa foi obtido com êxito pois, o protótipo do SGCOPEX foi validado pelo especialista e pelo atual agente executivo, como sendo um sistema útil e que atende as necessidade da Coordenadoria de Pesquisa e Extensão no Instituto Federal de Sergipe.

Como trabalhos futuros, sugere-se a implementação de novos módulos, tais como: pesquisas mais específicas, relatórios mais detalhados incluindo gráficos evolucionarios e series temporais, além do plano de manutenção contínua do software visando a evolução do sistema em versão Beta em produção, para que o mesmo possa atender as necessidades dinamicas da COPEX e ampliar a capacidade operacional da instituição alvo.

REFERÊNCIAS

Boulic, R. and Renault, O. (1991) “3D Hierarchies for Animation”, In: New Trends in Animation and Visualization, Edited by Nadia Magnenat-Thalmann and Daniel Thalmann, John Wiley & Sons ltd., England.

Bezerra, Eduardo. Princípios de Análise e Projeto de Sistemas com UML. – São Paulo, Brasil: Editora Elsevier-Campus, 2006, 279p.

Dall’Oglio, Pablo. PHP Programando com Orientação a Objetos. – São Paulo, Brasil: Editora NOVATEC, 2009, 571p.

FabForce.net. DBDesigner 4 and MySQL. Disponível em: <http://www.fabforce.net/dbdesigner4/>. Acessado em 12 de setembro de 2015.

Fowler, Martin. UML Essencial - Um Breve Guia para a Linguagem-Padrão de Modelagem de Objetos. – São Paulo, Brasil: Editora Bookman, 2004, 160p.

Garcia, Luíz Fernando F. Engenharia de Software I. – Canoas, Brasil: Editora da ULBRA, 2013, 101p.

Guedes, Gilleanes T. A. UML 2 - Uma Abordagem Prática. – São Paulo, Brasil: Editora NOVATEC, 2011, 488p.

Melo, Ana Cristina. Desenvolvendo Aplicações com UML 2.2 – do Conceitual à Aplicação. – São Paulo, Brasil: Editora Brasport, 2010, 317p.

Machado, Felipe; Abreu, Maurício. Projeto de Banco de Dados - Uma Visão Prática. – São Paulo, Brasil: Editora Érica, 2004, 292p.

Niederauer, J. Desenvolvendo Websites com PHP. – São Paulo, Brasil: Editora NOVATEC, 2ª edição, 2011, 304p.

Pádua, Wilson. Engenharia de Software. – São Paulo, Brasil: LTC, 2009, 260p.

PHP. Disponível em: <https://php.net/>. Acessado em 12 de setembro de 2015.

Prates, Rubens; Niederauer, Juliano. MySQL 5 – Guia de Consulta Rápida. – São Paulo, Brasil: Editora NOVATEC, 2005, 112p.

Pressman, Roger S. Engenharia de Software - Uma Abordagem Profissional. – Porto Alegre, Brasil: Editora AMGH, 2011, 773p.

Rezende, Denis Alcides. Engenharia de Software e Sistemas de Informação. – São Paulo, Brasil: Editora Brasport, 2005, 313p.

Rumbaugh, James; Booch, Grandy; Jacobson, Ivar. UML - Guia do Usuário. – São Paulo, Brasil: Editora Campus, 2005, 463p.

Silva, Maurício Samy. Criando Sites com HTML – Sites de Alta Qualidade com HTML e CSS. – São Paulo, Brasil: Editora NOVATEC, 2008, 427p.

Silva, Maurício Samy. JavaScript - Guia do Programador. – São Paulo, Brasil: Editora NOVATEC, 2010, 608p.

Silva, Alberto; Videira, Carlos. UML, Metodologias e Ferramentas CASE. – Portugal, Editora Centro Atlântico, 2001, 545p.

Sommerville, Ian. Engenharia de Software. – São Paulo, Brasil: Editora PEARSON, 2003, 353p.

StarUML. Disponível em: <http://staruml.io/>. Acessado em 20 de agosto de 2015.