

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

ESCOLA POLITÉCNICA

DEPARTAMENTO DE ENGENHARIA ELETRÔNICA E DE COMPUTAÇÃO

Sistema de Acompanhamento de Projeto

Autor: _____

(Roberto Gomes Gonçalves)

Orientador: _____

(Antonio Cláudio Gómez de Souza)

Examinador: _____

(Sergio Palma da Justa Medeiros)

Examinador: _____

(Marcelo Luiz Drumond Lanza)

DEL

Novembro/2005

Dedicatória

Dedico este projeto aos meus pais, que tanto lutaram para a concretização do meu ensino superior e que agora podem estar colhendo os frutos da educação que me proporcionaram. Considero que sem a ajuda, o suporte, a força, o diálogo e a cobrança deles, eu não teria conseguido terminar o curso e adquirido meu diploma de engenheiro tão desejado desde menino.

Agradecimentos

Agradeço a Deus, meu Pai celestial, por toda a força, inteligência, competência, vontade, determinação e paz que me foi dado durante meu aprendizado no curso de Engenharia Eletrônica e de Computação, o que propiciou a realização deste projeto. Agradeço também a EDS, a empresa que estou trabalhando atualmente, pela oportunidade dada e pelo conhecimento que pude adquirir. Sem essa experiência não poderia estar atuando no projeto em que estou atualmente.

Esta iniciativa consiste no desenvolvimento de um sistema que permitirá que cada líder de projeto (durante o projeto vai ser utilizado a sigla TL para identificar os líderes de projetos) preencha os dados referentes ao seu projeto de uma forma amigável e que, ao terminar, seu respectivo gerente (para identificar o gerente dos líderes de projeto vai ser utilizado a sigla DL) seja automaticamente avisado para que possa avaliar as informações preenchidas.

Depois que todos os TLs (Team Leaders) ligados a um DL (Delivery Leader) terminarem de preencher os status dos seus projetos referentes a uma semana, será a vez do DL preencher a avaliação do seu programa associado (um programa engloba vários projetos) e adicionar comentários que considere relevantes levar ao conhecimento do SCM (gerente de todo o centro de solução). Todas as vezes que um DL finalizar a atualização do seu status semanal, o PSO (Project Support Office – projeto interno) será automaticamente avisado pelo sistema. Neste momento o líder do PSO poderá ler as informações daquele DL e seus TLs.

O PSO poderá questionar os DLs e TLs quanto às informações reportadas e acrescentar comentários para o que foi preenchido pelos mesmos. Ao final, o PSO fará uma avaliação geral do Centro de Solução e incluirá comentários que considere interessante reportar para a ADA (Applications Delivery Americas).

Por último, o SCM será avisado automaticamente da finalização dos comentários do PSO e poderá aprovar, reprovar ou corrigir o que o PSO escreveu. Os grupos internos de apoio (PSO, PPI – grupo de métricas, BM – Business Manager e IT – equipe de infra-estrutura) poderão utilizar o sistema usando o mesmo relacionamento existente entre DLs e TLs. O sistema possibilitará a geração de uma planilha Excel no formato exigido pela ADA que será postada no e-room quinzenalmente pelo PSO.

Todas as informações serão armazenadas num banco de dados. Esta iniciativa segue os “guidelines” para os procedimentos de melhoria de processos e tecnologia e está alinhada com os procedimentos de “Change Requests” tecnológicos. O projeto deve considerar a construção de um protótipo para avaliar melhor se os benefícios estimados serão realmente atingidos.

Palavras-chave:

- Controle de qualidade
- CMMI5
- Banco de Dados
- Framework
- PM – Project Manager
- DL – Delivery Leader
- SCM – Solution Centre Manager

Índice

| | |
|--|--------|
| Parte 1: <i>Introdução</i> | 1 |
| 1.1. Motivação | 1 |
| 1.2. Visão Geral | 1 |
| 1.3. Metodologia | 2 |
| Parte 2: <i>Desenvolvimento</i> | 5 |
| 2.1. O Problema | 5 |
| 2.2. Definições | 10 |
| 2.3. Funcionalidades do Sistema | 11 |
| 2.4. Processo de Desenvolvimento | 12 |
| Parte 3: <i>Resultado</i> | 21 |
| <i>Conclusão</i> | 22 |
| <i>Bibliografia</i> | 25 |
| Apêndice A - Planejamento | 26 |
| Apêndice B - ERS | 36 |
| Apêndice C – Desenho do Sistema | 53 |

Introdução

1.1. Motivação

A idéia deste projeto surgiu de uma iniciativa interna da empresa, sem nenhum patrocínio da corporação. Como estava para ser efetivado na empresa e muitos projetos em J2EE estavam surgindo, decidi como meu gerente a realização deste projeto, onde ambos podíamos ser beneficiados. Eu, que estou querendo aprender J2EE, metodologias de projeto e preciso de uma idéia para o meu projeto final, e a empresa que pode ter um sistema a um custo muito baixo. A idéia inicial é usá-lo no Rio de Janeiro, mas o sistema vai ser portátil para outros centros de solução, como São-Paulo e Florianópolis. Caso o sistema seja bem aceito, ele vai ser utilizado somente pelos diversos níveis de gerência, o que pode me dar uma projeção muito boa na empresa.

1.2. Visão Geral

Esta iniciativa tem como objetivo oferecer, através de um sistema web integrado com banco de dados, um meio dinâmico e amigável para que a gerência dos diversos projetos no centro de solução do Rio de Janeiro reportem o status semanal dos seus projetos e programas, permitindo maior interação entre os envolvidos, eliminação do tempo gasto na consolidação das informações e ainda uma rápida recuperação de dados históricos.

1.3. Metodologia

Na fase de especificação funcional, o processo de desenvolvimento deste projeto, será baseado nos modelos CMMI (*Capability Maturity Model Integration*) de desenvolvimento de software, desenvolvido em conjunto por Corporações da Indústria de Software, pelo governo dos EUA e pelo SEI (*Software Engineering Institute*).

Será adotada, na fase de especificação técnica e design do projeto, uma engenharia de software orientada a objetos utilizando-se dos modelos UML(*Unified Modeling Language*). Através da utilização do software Rational Rose serão feitos os diagramas necessários para um perfeito entendimento do projeto.

O CMMI - Capability Maturity Model Integration foi criado pelo SEI como uma integração e evolução dos modelos: SW-CMM - Capability Maturity Model for Software; SECM - EIA 731 - System Engineering Capability Model, e IPD-CMM - Integrated Product Development CMM. O CMMI é um modelo alinhado com a futura Norma ISO/IEC 15504: Information Technology - Process Assessment e é apresentado em duas representações: uma por estágio (como o CMM, que deverá ser desativado até 2005) e outra contínua (semelhante à 15504).

O projeto CMMI foi desenvolvido para definir um ponto inicial para modelos integrados e aprimorar as melhores práticas para a criação de modelos baseados em lições aprendidas. Estabelecer um framework que possibilite a integração futura de novos modelos com a criação de uma forma associada de avaliação de desempenho e treinamento de produtos. Foi um esforço conjunto: mais de 100 profissionais de aproximadamente 30 empresas envolvidas, Indústria, Governo e Instituto de Engenharia de Software (SEI).

Este processo de certificação tem a premissa do gerenciamento de processos: a qualidade de um sistema é fortemente influenciada pela qualidade do processo usado para produzi-lo, adquiri-lo e mantê-lo. Esta premissa defende que o foco no processo é tão importante quanto o foco no produto.

Abaixo são mostrados os diferentes modelos de CMMI:

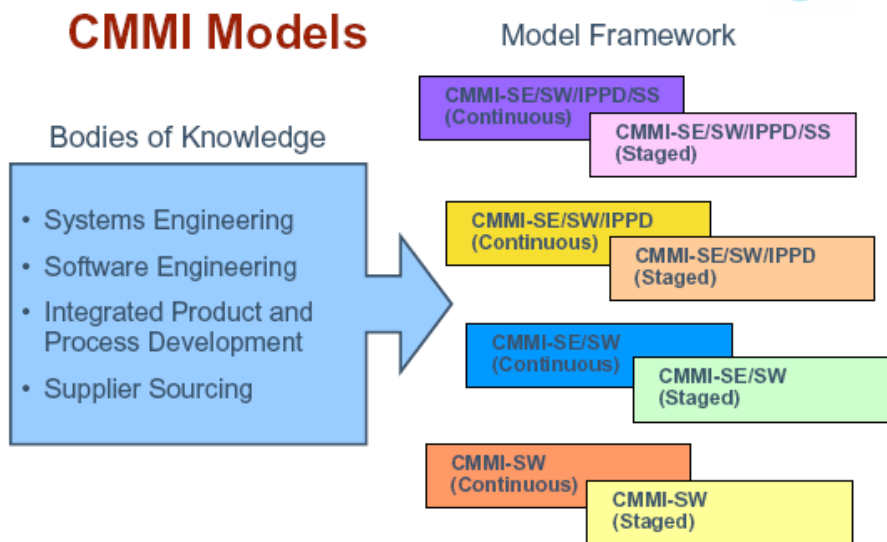
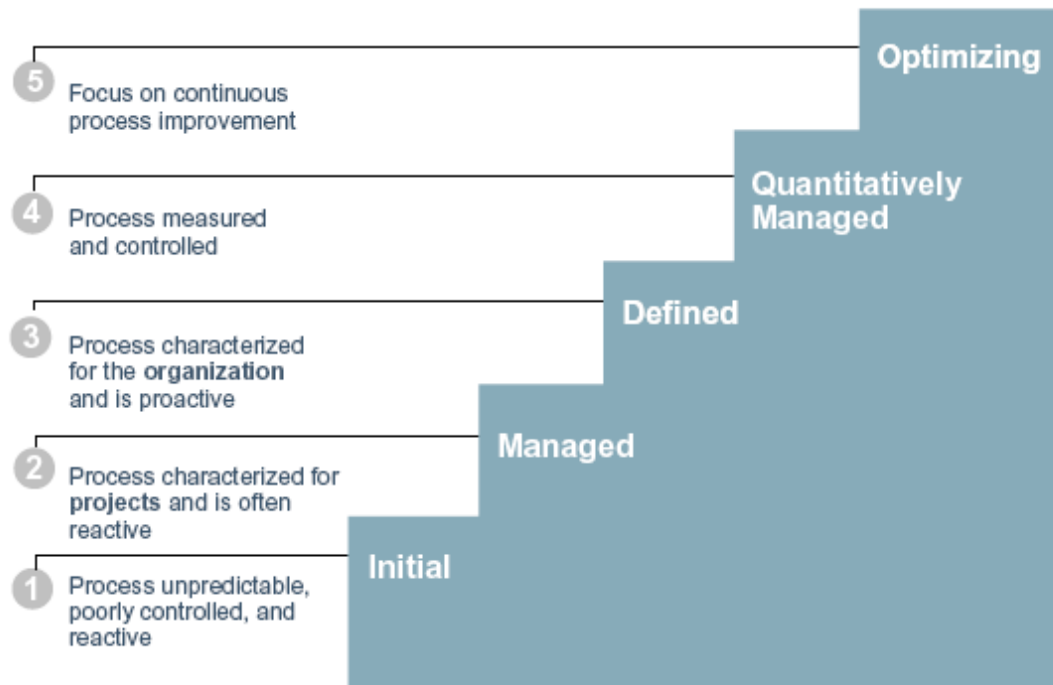


Figura 1 : Modelos CMMI

Representação por etapas:

- Descreve etapas bem definidas de melhoria onde cada uma serve de pré-requisito para a próxima;
- Descreve uma avaliação simples que sumariza o resultado do nível de desempenho e permite comparações dentro e fora da organização (entre seus próprios processos e entre processos de outras organizações);
- Possibilita a migração simples do SW-CMM para o CMMI;
- Permite que organizações escolham uma área de processo específico e façam melhorias relativas a ele;
- Maturity level são as etapas fundamentais do processo contínuo de melhoria através do uso de uma seqüência bem definida de práticas para a melhoria, começando com as práticas básicas de gerenciamento e progredindo através de um caminho estabelecido por níveis sucessivos.

The Maturity Levels



© 2005 by Carnegie Mellon University

CMMI Overview Page 41

Figura 2 : Maturity Levels

1.4 Sumário

A seguir serão mostradas as etapas do desenvolvimento do sistema. A maior parte da documentação está nos apêndices. Toda a documentação dos apêndices foi gerada no decorrer do desenvolvimento da aplicação. Portanto, as fases do desenvolvimento vão ser descritas fazendo referência aos apêndices.

2. Desenvolvimento

2.1. O Problema

A EDS possui um alto grau de maturidade na qualidade dos seus serviços, possuindo CMMI nível 5 e outras certificações, como o ISO. A EDS possui um sistema usado por toda corporação que especifica metodologias para garantir qualidade no desenvolvimento de softwares. Um dos meios para garantir esta qualidade nos serviços é a utilização de uma planilha Excel, que indica o status semanal de cada projeto na empresa. Esta planilha é preenchida por cada centro de solução na América latina e enviada para a EDS LATAM (Latim America) onde todos os dados são analisados. A idéia da construção deste sistema é a substituição desta planilha por um sistema web.

A atual proposta visa melhorar este processo, pois atualmente gasta-se muito tempo colhendo as informações de cada projeto e colocando no formato padrão da corporação. A idéia seria expandir este sistema para outros centros de soluções da empresa.

A compilação de dados seria automatizada pelo sistema e também teria um histórico de avaliações passadas, podendo fazer um acompanhamento detalhado de cada projeto. Este sistema é típico de workflow com atores bem definidos. A estrutura da empresa se apresenta da seguinte forma:

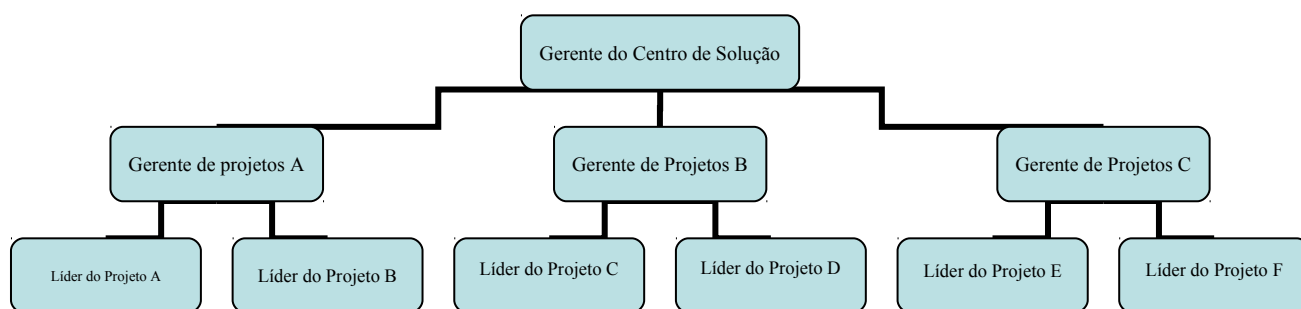


Figura 3 : Hierarquia de Gerência

Podemos perceber que há três níveis de gerenciamento, no primeiro nível estão os pontos focais de cada projeto, seu gerente é responsável por um grupo de recursos de um projeto. No segundo nível estão as pessoas responsáveis por um grupo de projetos, denominado programa, gerenciando um grupo de líderes de projeto. No terceiro nível está o gerente do Centro de Solução. O workflow do status semanal de projeto percorre esses três níveis:

- Primeiro cada líder de projeto vai preencher um formulário, indicando o status atual do projeto. Ao submeter o formulário este será encaminhado para o seu gerente, no segundo nível;
- No segundo nível, baseado nas avaliações dos líderes de projeto, o gerente de programa fará sua avaliação, que por sua vez será encaminhado para o gerente do centro de soluções;
- No último nível o gerente do centro de solução (Rio Solution Centre), fará sua avaliação baseado nas avaliações dos gerentes de programa.

Abaixo está o diagrama de workflow e seus atores:

Fluxo para o processo de Status Report do RioSC

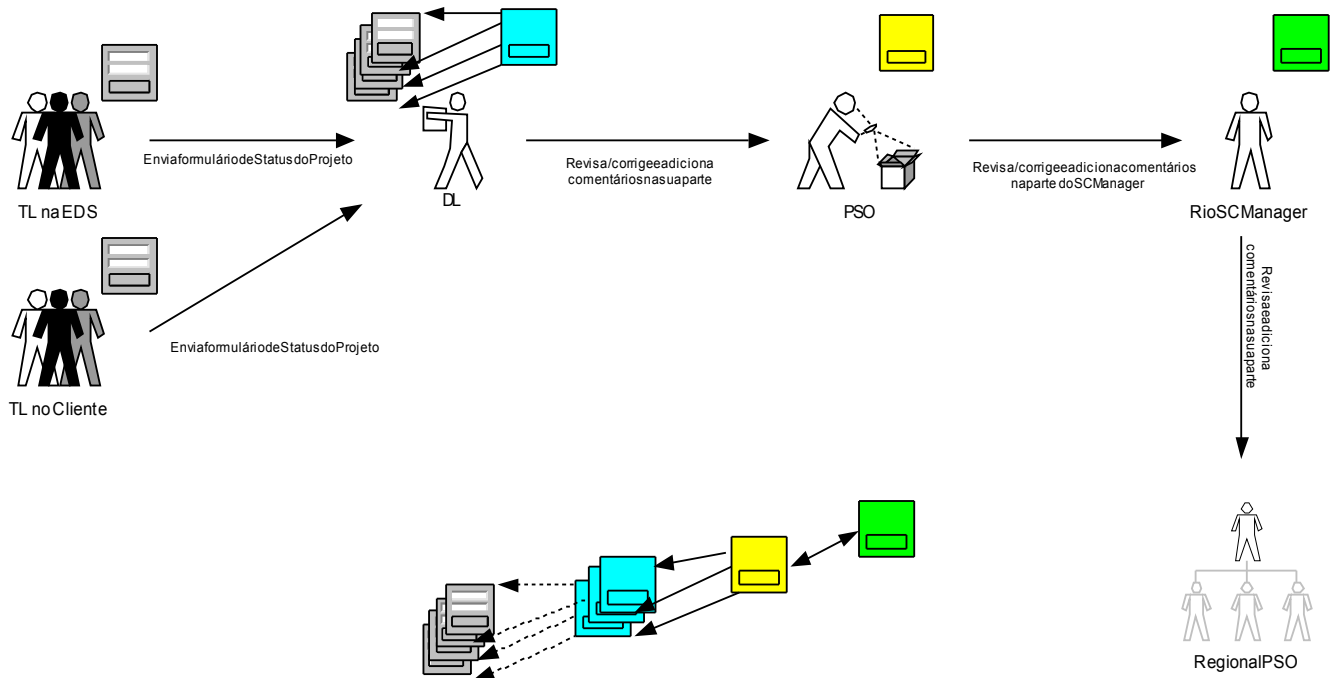


Figura 4 : Workflow do Sistema

Legenda:

TL = Team Leader, líder de projeto, responsável por um projeto apenas.

DL = Delivery Leader, gerente de programa, responsável por vários projetos.

PSO = Project Support Office, responsável por fazer a compilação de dados e a revisão, faz suporte a outros projetos.

RioSC Manager = Gerente do Rio Solution Centre, é responsável por todo o centro de solução.

Regional PSO = Entidade responsável por compilar os dados de avaliação de todos os centros de soluções da América latina.

O formulário usado atualmente para fornecer o status semanal do projeto utiliza os critérios de verde, amarelo e vermelho e cada critério de avaliação está descrito na tabela abaixo:

| | Green | Yellow | Red |
|----------------------------|--|--|---|
| Client Relationship | 1- Client satisfied with the project to date. | 1 -Relationship performing below expectation but recoverable. 2 -Client resources difficult, impacting quality and progress. 3 -Client's expectation of this project requires continuous effort to manage. 4 -Client have reservation with the project - Action plan in place. | 1 -Relationship performing below expectation - help required. 2 -Client resources extremely difficult, serious impact, progress stalled. 3 -Client's expectation of this project is improper or not set. 4 -This project caused major disruption to the Client Business Unit. 5 -Client dissatisfied with projects' progress. |
| Schedule | 1 -Nil project delays experienced. 2 -Project working to a baseline schedule. 3 -All Client commitments and Client Service Request met | 1 -Project delays caused by EDS, Client or Third Party - Action plan in place. 2 -Project working to a tentative baseline schedule - awaiting sign-off. 3 -Possible delay in Client Commitments or Client Service Request. | 1 -EDS, Client or Third Party causing project delay - No Action plan in place. 2 -Project working without an approved schedule. 3 -Delay in meeting Client commitments or Client Service Request. |
| Finance | 1 -Project finances tracking to schedule and BPA | 1 -Some team members are charged less than their level. 2 -BPA under negotiation. 3 -Project budget absorbing contingency. | 1 -Number of Staff higher than Billing. 2 -Unable to recover ODC. 3 -BPA outstanding - not signed-off. 4 -Insufficient budget exist for project. |
| Resource | 1 -Project finances tracking to schedule and BPA 2 -Resources lost due to unforeseen events but manageable. | 1 -Additional unplanned resources added, schedule being reviewed. 2 -Resource allocation is not complete - awaiting for resources to become available. Not critical at this stage. 3 -Resources lost due to unforeseen events - 11% to 25%. 4 -Perceived productivity problems due to training or | 1 -Additional unplanned resources added without changing the schedule. 2 -Resources allocation is not complete - there are two weeks before the project start date. 3 -Resources lost due to unforeseen events - 26% or greater. 4 -Inter team problems exist requiring TM involvement. 5 -Identified productivity problems due to training or inappropriate staff: |

| | | | |
|-------------------|--|--|---|
| | | inappropriate staff: | |
| Solution | 1 -All Issues manageable - Action Plan in place. 2 -Client Business process well defined. 3 -Acceptance criteria sign-off with the Client | 1 -1 to 3 Issues opened for more than 10 days, past its original resolve date. 2 -Client Business process vague. 3 -Acceptance Criteria with client for approval | 1 -4 or more Issues opened for more than 10 days, past its original resolve date. 2 -Client Business process not defined. 3 -No acceptance criteria signed-off with Client |
| Governance | 1 -EDS contract with the Client signed. 2 -Politics between EDS and the Client are manageable or non-evident. 3 -No non-conformances delayed, Less than 5 NC opened. | 1 -Politics between EDS and the Client - extra steps required causing delays. 2 -More the 5 NC opened, on schedule to clear | 1 -EDS contract with the Client remains outstanding - Not signed-off. 2 -Politics between EDS and the Client - progress seriously impeded. 3 -More the 5 NC opened, delayed |
| IT | 1 -Nil IT or Infrastructure problems. | 1 -Security access problem exist 2 -Delays exist in identifying infrastructure for this project | 1 -Delays exist in acquiring Hardware and Software - project impacted. 2 -Insufficient or wrong infrastructure exist. |

Tabela 1 : Critérios de Avaliação

O sistema terá a figura do administrador, que será responsável pelo cadastramento de funcionários e de projetos. O usuário vai se logar no sistema e somente será mostrado o projeto associado a ser avaliado. Então o sistema terá 3 tipos de avaliação:

1. Avaliação de projeto, primeiro nível de avaliação.
2. Avaliação de programa, cujo o programa será composto de vários projetos sendo o segundo nível de aprovação.
3. Avaliação de Centro de Solução, terceiro nível de aprovação.

2.2. Definições

Segue abaixo as definições de alguns termos e siglas que serão usadas no projeto:

- TL :
Team Leader. Líder de projeto, gerente nível 1.
- PM:
Project Manager. Gerente de projeto, possui quase as mesmas atribuições do TL, está no nível 1.
- DL:
Delivery Leader. Gerente de programa, responsável por gerenciar vários projetos, está no nível 2.
- Administrador:
Cabe ao administrador cadastrar funcionários, projetos, gerar relatórios. O administrador pertencerá ao grupo PSO.
- PSO:

Project Support Office, projeto interno que visa dar suporte aos projetos do centro de solução.

- *BPA:*

Business Partner Agreement – Contrato entre a EDS e os clientes.

2.3. Funcionalidades do Sistema

A seguir são apresentadas as regras de negócio que resumem as funcionalidade do sistema e permitem a geração do diagrama de entidades e relacionamento.

- Administrador cadastra usuário (líder de projeto, gerente de projetos, gerente do centro de solução);
- Administrador cadastra projeto;
- Administrador cadastra centro de solução
- Administrador edita usuário;
- Administrador edita projeto;
- Administrador edita centro de solução;
- Líder de projeto avalia projeto;
- Gerente nível 2 (programa) avalia programa;
- Gerente nível 3 (centro de solução) avalia centro de solução;
- As avaliações são editáveis por uma semana;
- As avaliações tem duração de uma semana;
- Gerente nível 2 (programa) pode avaliar projeto também;
- Gerente nível 3 (centro de solução) só avalia o centro de solução;
- O centro de solução tem apenas 1 gerente nível 3;
- Uma ou mais usuários podem avaliar um projeto;
- A mesma pessoa pode avaliar projetos diferentes;
- O gerente nível 2 só pode avaliar um programa;

- Um projeto só pode estar associado a um centro de solução;
- O projeto tem nome, quantidade de recursos, fase atual do projeto, cliente, tipo de contrato, tipo de indústria.
- O usuário tem username, senha, email válido e nome completo;
- Os itens a serem avaliados no projeto são Client Relationship, Schedule, Finance, Resource, Solution, Governance e IT;
- Todos estes critérios de projeto possuem comentários para um maior detalhamento;
- Os itens a serem avaliados no programa são Overall, Delivery, Financial, IT, People, Governance;
- Todos estes critérios de programa possuem comentários para um maior detalhamento;
- Os itens a serem avaliados no centro de solução são Overall, Delivery, Financial, IT, People, Governance;
- Todos estes critérios de avaliação de centro de solução possuem comentários para um maior detalhamento;
- Todos os itens podem assumir 3 valores: Red, Green, Yellow.
- Cada avaliação feita dispara um email para o usuário, para o administrador para o nível de gerência acima confirmando a avaliação;
- O administrador gera relatório semanal.

2.4. Processo de Desenvolvimento

O processo adotado para o desenvolvimento deste software foi o RUP – Rational Unified Process.

O Processo Unificado proposto pela Rational (Rational Unified Process – RUP) foi criado para apoiar o desenvolvimento orientado a objetos, fornecendo uma forma sistemática para se obter reais vantagens no uso da Linguagem de Modelagem Unificada (Unified Modeling Language – UML). De fato, ele não é exatamente um processo: é uma infra-estrutura genérica de processo que pode ser especializada para

uma ampla classe de sistemas de software, para diferentes áreas de aplicação, tipos de organização, níveis de competência e tamanhos de projetos.

O RUP está fundamentado em três princípios básicos: orientação a casos de uso, arquitetura e iteração. Ele é dito dirigido a casos de uso, pois são os casos de uso que orientam todo o processo de desenvolvimento. Com base no modelo de casos de uso, são criados uma série de modelos de análise, projeto e implementação, que realizam estes casos de uso. É centrado em arquitetura, pois defende a definição de um esqueleto para a aplicação (a arquitetura), a ganhar corpo gradualmente ao longo do desenvolvimento. Finalmente, o RUP é iterativo e incremental, oferecendo uma abordagem para particionar o trabalho em porções menores ou mini-projetos. Esses três conceitos são igualmente importantes. A arquitetura provê a estrutura para guiar o desenvolvimento do sistema em iterações, enquanto os casos de uso definem as metas e conduzem o trabalho de cada iteração. O ciclo de vida adotado no RUP é tipicamente evolutivo. Contudo, uma forma de organização em fases é adotada para comportar os ciclos de desenvolvimento, permitindo uma gerência mais efetiva de projetos complexos. Ao contrário do tradicionalmente definido como fases na maioria dos modelos de ciclo de vida. O RUP define: planejamento, levantamento de requisitos, análise, projeto, implementação e testes, como “fluxo de atividades” e as fases são ortogonais a este fluxo:

- Conceção: nesta fase, é estabelecido o escopo do projeto e suas fronteiras, determinando os principais casos de uso do sistema. Esses casos de uso devem ser elaborados com a precisão necessária para se proceder a estimativas de prazos e custos. As estimativas devem ser globais para o projeto como um todo e detalhadas para a fase seguinte. Assim, a ênfase nesta etapa recai sobre o planejamento e, por conseguinte, é necessário levantar requisitos do sistema e preliminarmente analisá-los. Ao término dessa fase, são examinados os objetivos do projeto para se decidir sobre a continuidade do desenvolvimento;
- Elaboração: o propósito desta fase é analisar mais refinadamente o domínio do problema, estabelecer uma arquitetura de fundação sólida, desenvolver um plano de projeto para o sistema a ser construído e eliminar os elementos de projeto que oferecem maior risco. Embora o processo deva sempre acomodar alterações, as atividades da fase de elaboração asseguram que os requisitos, a arquitetura e os planos estão suficientemente estáveis e que os

riscos estão suficientemente mitigados, de modo a se poder prever com precisão os custos e prazos para a conclusão do desenvolvimento.

- Construção: durante esta fase, um produto completo é desenvolvido de maneira iterativa e incremental, para que esteja pronto para a transição à comunidade usuária.
- Transição: nesta fase, o software é disponibilizado à comunidade usuária. Após o produto ter sido colocado em uso, naturalmente surgem novas considerações que vão demandar a construção de novas versões para permitir ajustes do sistema, corrigir problemas ou concluir algumas características que foram postergadas. É importante realçar que dentro de cada fluxo, um conjunto de iterações, envolvendo planejamento, levantamento de requisitos, análise, projeto e implementação e testes, é realizado. Contudo, de uma iteração para outra e de uma fase para a próxima, a ênfase sobre as várias atividades muda, como mostra a figura 1, em que a cor preta indica grande ênfase, enquanto a cor branca indica muito pouca ênfase. Na fase de concepção, o foco principal recai sobre o entendimento dos requisitos e a determinação do escopo do projeto (planejamento e levantamento de requisitos). Na fase de elaboração, o enfoque está na captura e modelagem dos requisitos (levantamento de requisitos e análise), ainda que algum trabalho de projeto e implementação seja realizado para prototipar a arquitetura, evitando certos riscos técnicos. Na fase de construção, o enfoque concentra-se no projeto e na implementação, visando evoluir e recheiar o protótipo inicial, até obter o primeiro produto operacional. Finalmente, a fase de transição concentra-se nos testes, visando garantir que o sistema possui o nível adequado de qualidade. Além disso, usuários devem ser treinados, características ajustadas e elementos esquecidos adicionados.

| | Levantamento de Requisitos | Análise | Projeto | Implementação | Testes |
|------------|----------------------------|---------|---------|---------------|--------|
| Concepção | | | | | |
| Elaboração | | | | | |
| Construção | | | | | |
| Transição | | | | | |

Figura 5 : Ênfase principal em cada uma das fases

Processo de Engenharia de Software na visão do RUP



Figura 6 : Ciclo de vida RUP

Um processo é um conjunto de passos ordenados com a intenção de atingir uma meta. Em engenharia de software, a meta é criar um software ou aperfeiçoar um existente; em engenharia de processos, a meta é desenvolver ou aperfeiçoar um processo. No RUP, eles são organizados em um conjunto de disciplinas para posteriormente definirem os fluxos de trabalho e outros elementos do processo.

Em termos de modelagem de negócios, o processo de desenvolvimento de software é um processo de negócios, e o Rational Unified Process (RUP) é um processo de negócios genérico para engenharia de software orientada a objetos. Ele

descreve uma família de processos de engenharia de software relacionados, que compartilham uma estrutura comum, uma arquitetura de processos comum. Ele proporciona uma abordagem disciplinada para a atribuição de tarefas e de responsabilidades dentro de uma organização de desenvolvimento. Sua meta é garantir a produção de software de alta qualidade que atenda às necessidades dos usuários, dentro de uma programação e um orçamento previsíveis. O RUP captura muitas das melhores práticas do desenvolvimento de software moderno, de forma que possam ser adaptadas para uma grande variedade de projetos e de organizações.

Quando um sistema de software é desenvolvido começando do zero, o desenvolvimento é o processo de criação de um sistema a partir dos requisitos. Porém, depois que os sistemas tiverem tomado forma (ou seja, tiverem passado pelo ciclo de desenvolvimento inicial), os desenvolvimentos subsequentes serão o processo de adaptação do sistema aos requisitos novos ou modificados. Isso se aplica durante todo o ciclo de vida do sistema.

No processo de desenvolvimento do sistema de acompanhamento de projeto, cada fim de uma iteração foi representado por um documento. Todos esses documentos estão no apêndice. Segue abaixo um resumo das fases do projeto. É apenas um resumo já que estas fases estão descritas com detalhes na documentação de apêndice.

2.4.1 Planejamento Inicial

Neste fase foi feita uma abordagem generalista do sistema como um todo. Nela foi apresentado o propósito do sistema bem como suas motivações e principais características. Além disso foram abordados as tecnologias que serão usadas no desenvolvimento do sistema (Patterns e técnicas de Engenharia de Sistemas em geral) bem como o escopo e cronograma do projeto. Este foi o cronograma inicial que está na documentação de planejamento.

| Cronograma de Atividades | | | | | | | | | | | | | | | | |
|----------------------------|-------|---|---|---|-------|---|---|---|------|---|---|---|-------|---|---|---|
| Semana: | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| Mês: | março | | | | abril | | | | maio | | | | junho | | | |
| Definição do scopo inicial | | | | | | | | | | | | | | | | |
| Planejamento do sistema | | | | | | | | | | | | | | | | |

[illegible]

Tabela 2 : Cronograma Inicial

O cronograma inicialmente foi seguido até o mês de Junho, pois tinha reservado algumas horas do dia no trabalho para a criação do sistema. Já em Julho fui deslocado para outro projeto no qual fui efetivado pela empresa e não tive mais tempo disponível para a finalização do sistema. Por isso algumas fases não foram completadas como o teste de integração e os testes de validação com os usuários. Foi criada uma versão inicial do sistema. Seguem no apêndice os detalhes da documentação do planejamento inicial da aplicação.

2.4.2 Requisitos

Neste fase foram feitas definições relativas aos objetivos, perspectivas e funções do produto que foi desenvolvido. Também especificou-se as características dos usuários, público alvo, restrições de utilização, definição de siglas e termos utilizados. Além disso foram apresentados alguns requisitos funcionais do projeto como interfaces de comunicação (com usuários, hardware e outros sistemas), requisitos de desempenho e as restrições que foram oferecidas. A seguir podemos ver o diagrama de casos de uso do sistema, mostrando as principais funcionalidades do sistema. Este diagrama está no documento de especificação de requisitos de software contido nos apêndices.

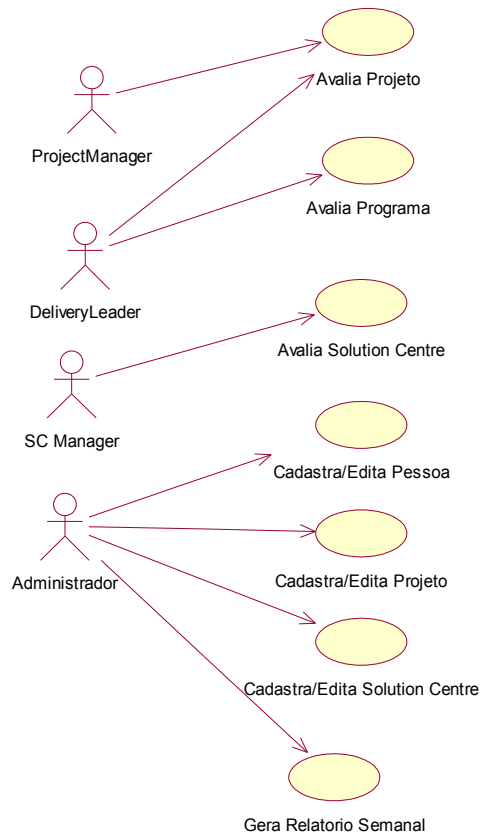


Figura 7 : Casos de Uso

Atores

- **Project Manager** : O nível mais baixo de gerência, está relacionado por projeto. É a pessoa responsável pelas avaliações semanais de cada projeto.
- **Delivery Leader**: Está acima do Project Manager, está associado a vários projetos. É a pessoa responsável pelas avaliações de programa.
- **SC Manager**: É a pessoa responsável por todo o centro de solução, vai ser a pessoa que vai fazer a avaliação do centro de solução.
- **Administrador**: É a pessoa responsável por fazer a manutenção do sistema, bem como o cadastro de pessoas, projeto, centro de solução. Esta função vai estar diretamente ligada ao PSO (Project Support Office) – Grupo interno de suporte a todos os outros projetos.

Descrição dos Casos de Uso

- **Avalia Programa:** O use case Avalia Programa vai permitir que os usuários (Delivery Leader) façam as avaliações de programa em uma determinada semana.
- **Avalia Projeto:** O use case Avalia Projeto vai permitir que os usuários (Program manager e Delivery Leader) façam as avaliações de projeto em uma determinada semana.
- **Avalia Solution Centre:** O use case Avalia Solution Centre vai permitir que os usuários (SC Managers) façam as avaliações de Solution Centre em uma determinada semana.
- **Cadastra/Edita Pessoa:** Este use case vai realizar as operações de cadastro e edição de pessoa (Gerentes).
- **Cadastra/Edita Projeto:** Este use case vai realizar as operações de cadastro e edição de projeto.
- **Cadastra/Edita Solution Centre:** Este use case vai realizar as operações de cadastro e edição de solution centre.
- **Gera Relatório Semanal:** Use case responsável pela geração de relatório semanal sobre as avaliações feitas.

Nesta fase foi apresentado os diagramas de classe e de sequência da aplicação que estão descritos no apêndice.

2.4.3 Análise e Design

Este é a fase que especifica exatamente como o sistema foi implementado. Nela foi mostrado os módulos, processos concorrentes e a composição do modelo de dados que fazem parte do projeto. Nesta fase foi definido como o sistema está arquitetado e funcionando.

Abaixo se encontra a arquitetura final do sistema:

ARQUITETURA - MVC

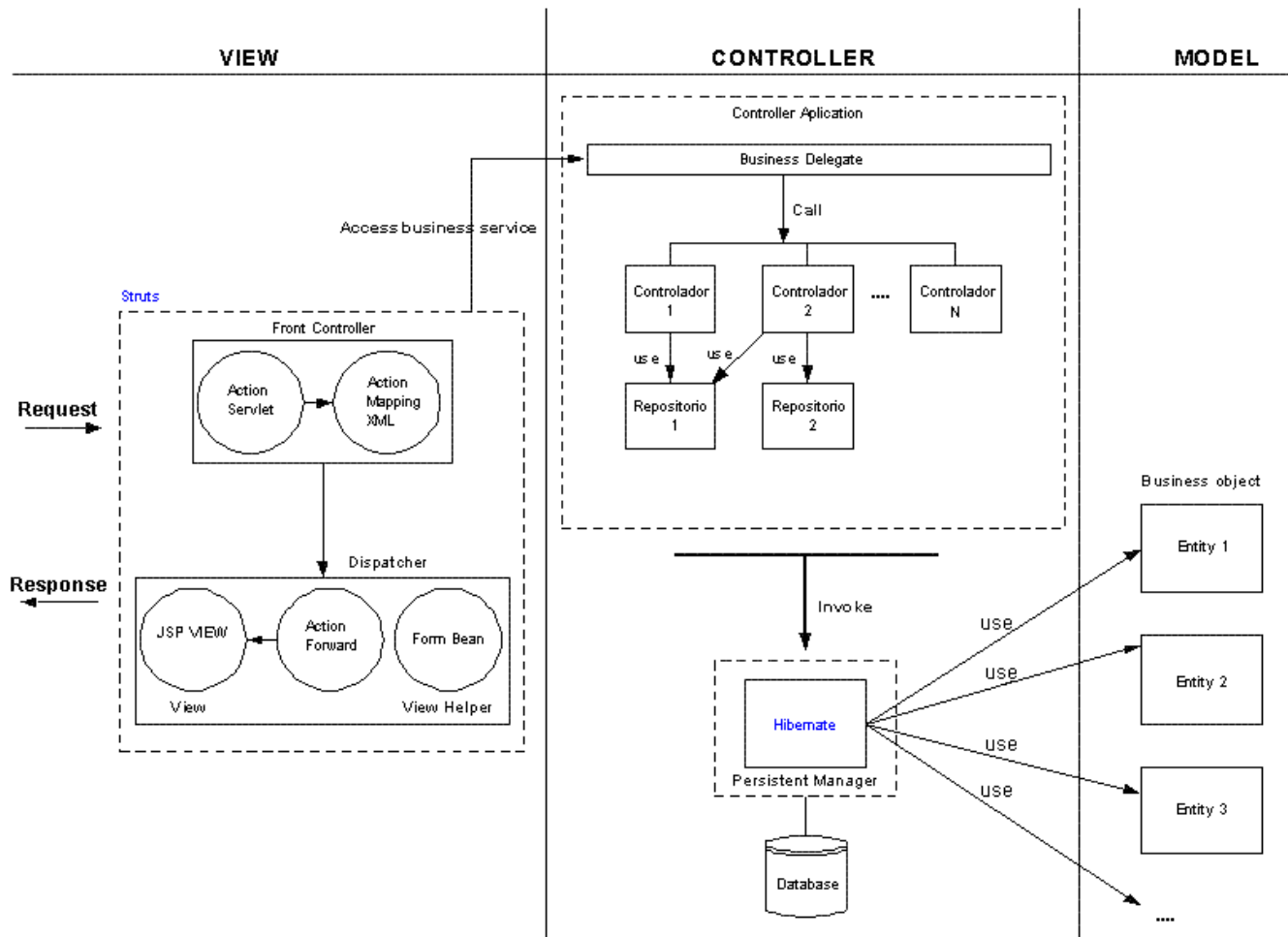


Figura 8 : Arquitetura Final do Sistema

A arquitetura com certeza foi um dos pontos fortes do sistema. A divisão em camadas foi muito bem feita. A utilização do framework struts acelerou bastante o desenvolvimento do software junto com o uso do hibernate. Maiores detalhes sobre estes frameworks se encontram na documentação de projeto no apêndice. Todas os módulos foram mostradas usando diagramas de sequência ou de classes, juntamente com as telas de navegação.

Resultados

Os resultados obtidos pelo sistema de acompanhamento de projeto foram muito bons. Apesar de ainda não estar em produção, a sua primeira versão foi muito bem aceita pelos gerentes que tiveram a chance de testá-lo. O fluxo de mensagens enviadas por email funcionou muito bem e é um modo de lembrar os esquecidos. Outro grande resultado foi a interface que foi muito bem aceita pelos gerentes que testaram. A possibilidade de visualizar as avaliações de mais de um projeto, para aqueles que gerenciam mais de um projeto, foi muito elogiado devido a sua praticidade. A navegação com um simples click do mouse nas avaliações antigas foi outro ponto forte. A aplicação não teve custo nenhum, já que as ferramentas de desenvolvimento ou eram gratuitas, na maioria (eclipse, struts, hibernate, mySQL) ou então tinham contrato com a EDS que foi o caso do Rational Rose.

Esta iniciativa gerou resultados muito satisfatórios para a minha carreira na empresa também. Com o conhecimento obtido neste projeto pude ter a oportunidade de entrar em outro grande projeto de desenvolvimento, já que a maioria dos projetos na empresa infelizmente são de suporte à produção. Pude ter a minha contratação na empresa efetuada. A única funcionalidade que ficou faltando, e com certeza será gerada em uma futura versão, foi a geração de relatórios. A princípio seria usado uma API de Java chamada POI que exporta dados para planilhas Excel ou então para documentos no formato Word. Infelizmente não foi possível fazer esta funcionalidade pela falta de tempo. Estava bastante curioso para aprender o uso desta API.

Conclusão

- Uso do Jboss como servidor de aplicação

Servidores de aplicação J2EE estão ganhando um grande impulso devido à produtividade no desenvolvimento de aplicações corporativas distribuídas que eles permitem, e devido à facilidade que têm no aproveitamento de sistemas legados e bases de dados relacionais em novas aplicações. Estes servidores estão se tornando a infraestrutura da nova geração de aplicações corporativas. Em um cenário em que os executivos de TI têm cada vez maiores restrições orçamentárias, mas também enormes pressões para resultados cada vez maiores, uma alternativa de servidor de aplicação open source de grande qualidade caiu como uma luva. Assim como o Linux, o Apache e outros softwares open source, o JBoss está se tornando parte fundamental nas decisões de TI das grandes corporações.

- Uso do framework Struts

As aplicações web normalmente são componentes críticos da missão da empresa. Como sempre as equipes de desenvolvimento precisam construir aplicações em tempo recorde, mas têm que construí-las corretamente, de forma a não dificultar a sua manutenção e que a performance não seja comprometida.

O Framework Struts não é uma tecnologia em específico, mas sim um conjunto destas, que possibilita o desenvolvimento de aplicações WEB , utilizando-se do modelo MVC-2. Este modelo é implementado no Struts através de um servlet que controla a aplicação, o ActionServlet, sendo este o responsável pelo controle do fluxo entre as páginas JSP e as diversas camadas da aplicação. É implementado ainda o padrão MVC/Layers usando ActionForwards e ActionMappings para manter as decisões do fluxo do controle fora da camada de apresentação.

Esta definição bem clara das camadas em uma aplicação permite a distribuição do trabalho, de forma que o designer pode desenvolver as páginas JSP's sem se preocupar com o restante da aplicação, e a sua integração será bastante simples através da utilização das taglibs do próprio Struts.

Por se tratar de um conjunto de tecnologias, o aprendizado do Struts pode tornar-se bastante dispendioso. Outro ponto que pode dificultar o aprendizado deste framework é a nomenclatura não muito clara dos componentes do Struts.

Apesar destas dificuldades, os pontos positivos e a predominância deste framework, são fatores motivantes para seu aprendizado e sua utilização:

- O Struts é performático, sua utilização melhora o desempenho das aplicações WEB;
- Os componentes ActionForm minimizam o código de subclasse e encurtam as hierarquias de subclasses;
- As bibliotecas de tags Struts fornecem uma funcionalidade com finalidade geral;
- Os componentes do Struts são reutilizáveis pela aplicação;
- As estratégias de localização do Struts reduzem a necessidade de JSPs redundantes;
- O Struts é designado com uma arquitetura aberta;
- O Struts é leve. Todo o framework é composto apenas por cinco bibliotecas de tags e cinco pacotes básicos;
- O Struts é compatível com os padrões;
- O Struts é de fonte aberta e bem documentada.

- Uso do Hibernate

O Hibernate é um framework incrível, que facilita o desenvolvimento de aplicações que acessam bancos de dados, fazendo com que o programador se preocupe mais com o seu modelo de objeto e seus comportamentos, do que com as tabelas do banco de dados. O Hibernate também evita o trabalho de escrever dúzias de código repetido para fazer as mesmas coisas, como “inserts”, “selects”, “updates” e “deletes” no banco de dados, além de ter duas opções para se montar buscas complexas em grafos de objetos e ainda uma saída para o caso de nada funcionar, usar SQL. Além de mecanismo de mapeamento objeto/relacional, o Hibernate também pode trabalhar com um sistema de cache das informações do banco de dados, aumentando ainda mais a performance das aplicações. O esquema de cache do Hibernate é complexo e totalmente extensível, existindo diversas implementações possíveis, cada uma com suas próprias características. E junto com isso, ele também tem um gerenciador de versões próprio.

Entretanto, ele não é a solução perfeita para todos os problemas, aplicações que fazem uso intenso de stored procedures, triggers, user defined functions e outras extensões específicas dos bancos de dados, normalmente não vão se beneficiar do mecanismo oferecido pelo Hibernate. Além disso, aplicações que tem uma alta frequência de “movimentos em massa”, como seqüências de “inserts”, “updates” e “deletes” terminariam por perder desempenho, graças à instanciação e carregamento de diversos objetos na memória.

- Uso da ferramenta Rational Rose

Após utilizar maneira básica a Ferramenta de Engenharia de Software ROSE, pudemos observar a versatilidade de uso do modelamento de casos que a ferramenta apresenta.

Com pouca interação e facilidade de uso, é possível construir modelos básicos para a iniciação na modelagem UML. A orientação à modelagem unificada enfatiza o raciocínio, o planejamento e o controle de estados, tirando o foco da programação do código, o que preza nosso raciocínio e não a linguagem de programação.

- Pessoal

Foi muito importante fazer este projeto, aprendi bastante sobre metologias de desenvolvimento de software orientado a objeto e pude ter um conhecimento aprofundado no paradigma de orientação a objeto. Aprendi sobre padrões de projeto, como uso do MVC como padrão de arquitetura e outros padrões de design como o Business Delegate, Singleton, Controller. A infra-estrutura de teste foi toda feita por mim. Ou seja, configurei o JBOSS em um servidor da empresa e coloquei a aplicação no ar. Foi importante saber a importância do XML. Todos os frameworks utilizados fizeram uso de XML. Foi importante também conhecer a UML, um padrão para desenvolvimento de sistema com orientação a objeto. Todo o desenvolvimento feito na ferramenta Rational Rose da IBM foram utilizados diagramas UML.

Bibliografia

- I. Roger S. Pressman, **Engenharia de Software**, Ed. McGraw-Hill Interamericana do Brasil Ltda, 5ª edição 2002
- II. CMMI Models
 - <http://www.sei.cmu.edu/cmmi/cmmi.html> 10/08/2005
- III. Srikanth Shenoy and Nithin Mallya, [Struts Survival Guide: Basics to Best Practices](#), [ObjectSource](#) © 2004
- IV. Darren Broemmer, [J2EE Best Practices: Java Design Patterns, Automation, and Performance](#), [John Wiley & Sons](#) © 2003
- V. [Using MySQL](#), [SkillSoft Press](#) © 2003
- VI. Java API
 - <http://developers.sun.com/> 10/08/2005
- VII. Hibernate
 - www.hibernate.org/ 10/08/2005

UFRJ - Universidade Federal do Rio de Janeiro
Escola Politécnica
Departamento de Engenharia Eletrônica e de Computação
04 de Abril de 2005

Planejamento

Aluno:

Roberto Gomes Gonçalves (rgg19812000@yahoo.com.br)

Orientador:

Prof.: Antônio Cláudio Gómez de Sousa

Título:

Sistema Web de Status de Projeto – Status Report

1. Resumo:

A idéia de fazer este projeto surgiu na empresa da qual faço parte atualmente, a EDS (Eletrônica Data System do Brasil), que atua na área de tecnologia da informação. A EDS possui um alto grau de maturidade na qualidade dos seus serviços, possuindo CMMI nível cinco. Um dos meios para garantir esta qualidade nos serviços é a utilização de uma planilha Excel, que indica o status semanal de cada projeto na empresa. A idéia da construção deste sistema veio desta planilha, pois ao invés da utilização desta planilha existiria um sistema na web.

A minha proposta é melhorar este processo, pois atualmente se gasta muito tempo colhendo as informações de cada projeto e colocando no formato padrão da corporação. A idéia seria expandir este sistema para outros centros de soluções da empresa que utilizam esta mesma planilha.

A compilação de dados seria automatizada pelo sistema e também teria um histórico de avaliações passadas, podendo fazer um acompanhamento detalhado de cada projeto. Este sistema é típico de workflow com atores bem definidos. A estrutura da empresa se apresenta da seguinte forma:

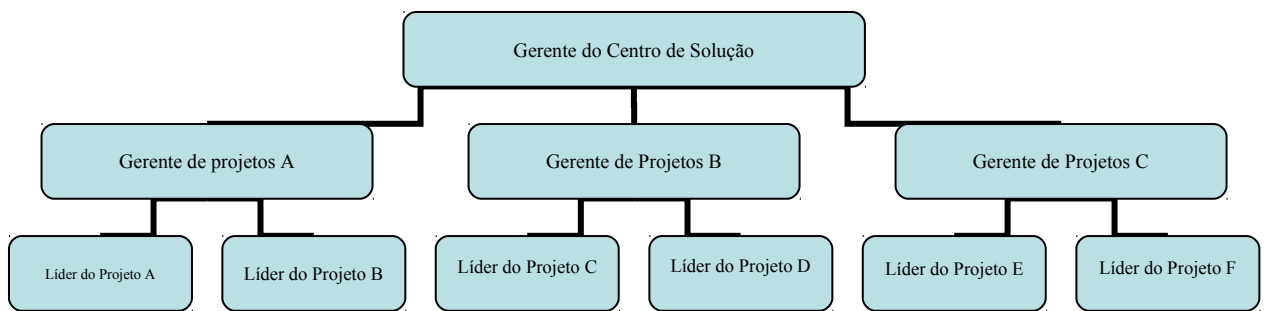


Figura 1: Estrutura da Gerência

Podemos perceber que há três níveis de gerenciamento, no primeiro nível estão os pontos focais de cada projeto, esta pessoa ela gerencia um grupo de recursos de um projeto. No segundo nível estão as pessoas responsáveis por um grupo de projetos, gerenciando um grupo de líderes de projeto. No terceiro nível está o gerente do Centro de Solução. O workflow do status semanal de projeto percorre esses três níveis:

- Primeiro cada líder de projeto vai preencher um formulário, indicando o status do atual projeto, ao submeter o formulário este será encaminhado para o seu gerente, para o segundo nível;
- No segundo nível, baseado nas avaliações dos líderes de projeto, o gerente de projetos fará sua avaliação, este então será encaminhado para o gerente do centro de soluções;
- No último nível o gerente do centro de solução (Rio Solution Centre), fará sua avaliação baseada nas avaliações dos gerentes de projetos.

Abaixo está o diagrama de workflow e seus atores:

Fluxo para o processo de Status Report do RioSC

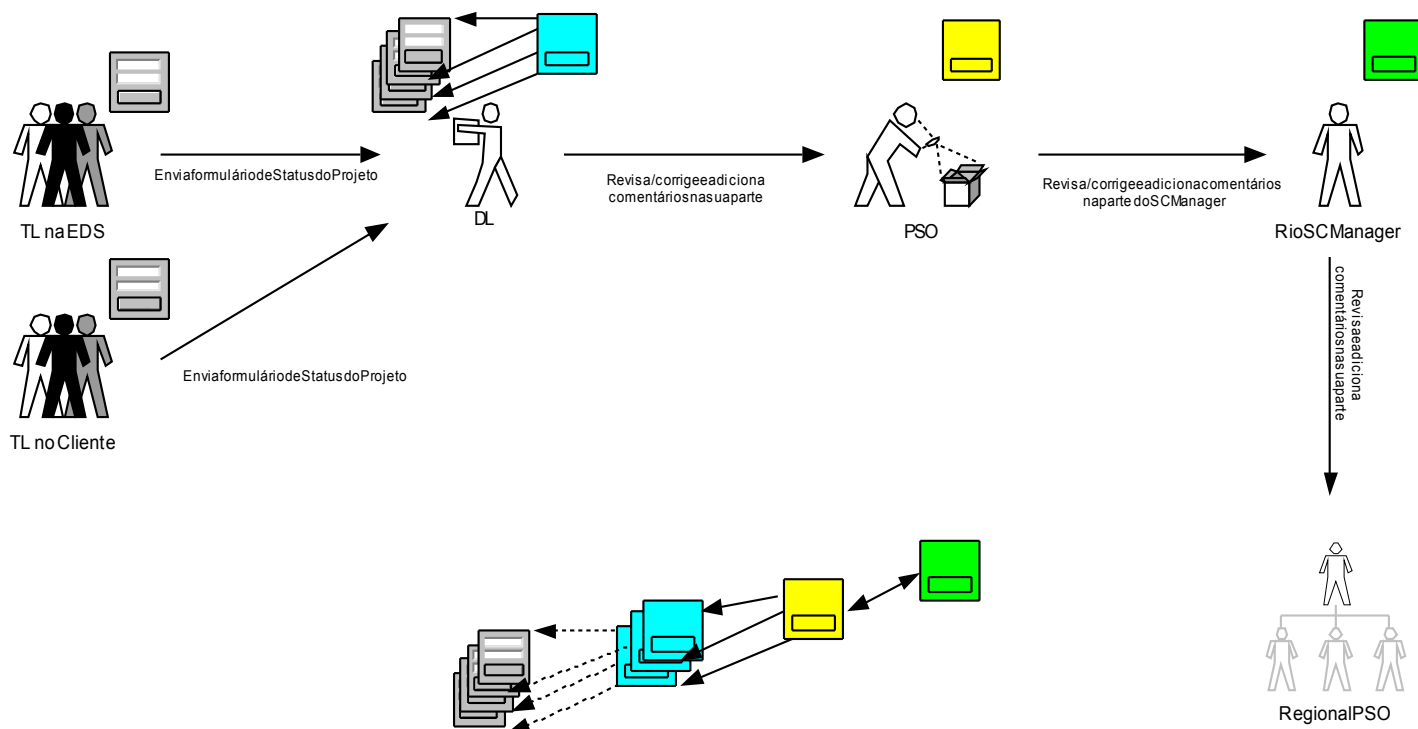


Figura 2: Diagrama de workflow

Legenda:

TL = Team Leader, líder de projeto, responsável por um projeto apenas.

DL = Delivery Leader, gerente de projetos, responsável por vários projetos.

PSO = Project Support Office, responsável por fazer a compilação de dados e a revisão, faz suporte a outros projetos.

RioSC Manager = Gerente do Rio Solution Centre, é responsável por todo o centro de solução.

Regional PSO = Entidade responsável por compilar os dados de avaliação de todos os centros de soluções da América latina.

O formulário inicial para fornecer o status semanal do projeto usado atualmente na planilha usa os critérios de verde, amarelo e vermelho e cada critério de avaliação está descrito na tabela abaixo:

| | Green | Yellow | Red |
|--|-------|--------|-----|
|--|-------|--------|-----|

| | | | |
|----------------------------|---|---|--|
| Client Relationship | 1- Client satisfied with the project to date. | 1 -Relationship performing below expectation but recoverable. 2 -Client resources difficult, impacting quality and progress. 3 -Client's expectation of this project requires continuous effort to manage. 4 -Client have reservation with the project - Action plan in place. | 1 -Relationship performing below expectation - help required. 2 -Client resources extremely difficult, serious impact, progress stalled. 3 -Client's expectation of this project is improper or not set. 4 -This project caused major disruption to the Client Business Unit. 5 -Client dissatisfied with projects' progress. |
| Schedule | 1 -Nil project delays experienced. 2 -Project working to a baseline schedule. 3 -All Client commitments and Client Service Request met | 1 -Project delays caused by EDS, Client or Third Party - Action plan in place. 2 -Project working to a tentative baseline schedule - awaiting sign-off. 3 -Possible delay in Client Commitments or Client Service Request. | 1 -EDS, Client or Third Party causing project delay - No Action plan in place. 2 -Project working without an approved schedule. 3 -Delay in meeting Client commitments or Client Service Request. |
| Finance | 1 -Project finances tracking to schedule and BPA | 1 -Some team members are charged less than their level. 2 -BPA under negotiation. 3 -Project budget absorbing contingency. | 1 -Number of Staff higher than Billing. 2 -Unable to recover ODC. 3 -BPA outstanding - not signed-off. 4 -Insufficient budget exist for project. |
| Resource | 1 -Project finances tracking to schedule and BPA 2 -Resources lost due to unforeseen events but manageable. | 1 -Additional unplanned resources added, schedule being reviewed. 2 -Resource allocation is not complete - awaiting for resources to become available. Not critical at this stage. 3 -Resources lost due to unforeseen events - 11% to 25%. 4 -Perceived productivity problems due to training or inappropriate staff: | 1 -Additional unplanned resources added without changing the schedule. 2 -Resources allocation is not compete - there are two weeks before the project start date. 3 -Resources lost due to unforeseen events - 26% or greater. 4 -Inter team problems exist requiring TM involvement. 5 -Identified productivity problems due to training or inappropriate staff: |
| Solution | 1 -All Issues manageable - Action Plan in place. 2 -Client Business process well defined. 3 -Acceptance criteria sign-off with the Client | 1 -1 to 3 Issues opened for more than 10 days, past its original resolve date. 2 -Client Business process vague. 3 -Acceptance Criteria with client for approval | 1 -4 or more Issues opened for more than 10 days, past its original resolve date. 2 -Client Business process not defined. 3 -No acceptance criteria signed-off with Client |

| | | | |
|-------------------|--|--|---|
| Governance | 1 -EDS contract with the Client signed. 2 -Politics between EDS and the Client are manageable or non-evident. 3 -No non-conformances delayed, Less than 5 NC opened. | 1 -Politics between EDS and the Client - extra steps required causing delays. 2 -More the 5 NC opened, on schedule to clear | 1 -EDS contract with the Client remains outstanding - Not signed-off. 2 -Politics between EDS and the Client - progress seriously impeded. 3 -More the 5 NC opened, delayed |
| IT | 1 -Nil IT or Infrastructure problems. | 1 -Security access problem exist 2 -Delays exist in identifying infrastructure for this project | 1 -Delays exist in acquiring Hardware and Software - project impacted. 2 -Insufficient or wrong infrastructure exist. |

Tabela 1: Critérios de avaliação

O sistema terá a figura do administrador, que será responsável pelo cadastramento de funcionários e de projetos. O usuário vai se logar no sistema e somente será mostrado o projeto associado a ser avaliado. Então o sistema terá três tipos de avaliação:

4. Avaliação de projeto, primeiro nível de avaliação.
5. Avaliação de programa, cujo programa será composto de vários projetos sendo o segundo nível de aprovação.
6. Avaliação de Centro de Solução, terceiro nível de aprovação.

2. Descrição do Projeto

2.1. Documentação

Para nos auxiliar no desenvolvimento desse sistema, utilizaremos o processo de produção de software baseado na disciplina Engenharia de Software cujo professor é o Antônio Cláudio – Analise, Desenho, Construção e Integração de Sistemas. O objetivo desta documentação é descrever detalhadamente todo o processo de construção do sistema, desde o seu planejamento até a produção.

Para tanto desenvolveremos e apresentaremos os seguintes documentos:

2.1.1. Documento de Especificação do Sistema

Neste documento (o de atual leitura) faremos uma abordagem generalista do sistema como um todo. Nele apresentaremos o propósito do sistema bem como suas motivações e principais características. Além disso abordaremos as tecnologias que serão usadas no desenvolvimento do sistema (Patterns e técnicas de Engenharia de Sistemas em geral) bem como o escopo e cronograma do projeto e onde cada membro da equipe irá atuar.

2.1.2. Documento de Planejamento do Sistema

Neste documento apresentaremos a finalidade do plano, escopo e objetivos do projeto, estimativas de tempo de desenvolvimento e de custo do software, estratégias para o gerenciamento de riscos do projeto, cronograma temporal detalhado, recursos necessários ao projeto, organização da equipe e mecanismos para o controle de qualidade e acompanhamento das mudanças no decorrer do desenvolvimento.

2.1.3. Documento de Especificação de Requisitos de Software

Neste documento faremos definições relativas aos objetivos, perspectivas e funções do produto que esta sendo desenvolvido. Também especificaremos as características dos usuários, publico alvo, restrições de utilização, definição de siglas e termos utilizados. Além disso, apresentaremos alguns requisitos funcionais do projeto como interfaces de comunicação (com usuários, hardware e outros sistemas), requisitos de desempenho e as restrições que serão oferecidas.

2.1.4. Documento do Projeto de Software

Este é o documento que especifica exatamente como o sistema será implementado. Nele mostraremos os módulos, processos concorrentes e a composição do modelo de dados que fazem parte do projeto. Ele será a melhor referência para o caso de atualizações futuras ou para aqueles que desejem compreender como o sistema está arquitetado e funcionando.

2.2. Arquitetura para o Desenvolvimento do Sistema

Abordaremos nessa sessão como será montada a base de trabalho para desenvolvermos o sistema. Mostraremos que nos basearemos em tecnologias sustentadas pelos paradigmas mais modernos de desenvolvimento de software, com o objetivo de implementar um sistema de altíssima manutenibilidade e qualidade.

2.2.1. Ferramentas que serão utilizadas

Para o desenvolvimento do sistema utilizaremos a linguagem Java em seu ultimo *release* o 1.5 (J2SE Development Kit e J2SE Runtime Environment – <http://java.sun.com>) com completo suporte a tecnologia web J2EE™.

Utilizaremos a IDE Eclipse para facilitar e integrar o desenvolvimento com outras ferramentas (<http://www.eclipse.org>). O Eclipse é uma IDE à qual desenvolvedores de todo mundo estão dedicados a gerar novas plugins com o objetivo de integrar o desenvolvimento Java a varias outras ferramentas. Em um patamar utópico desenvolveríamos todo o sistema usando apenas esta IDE e os plugins que emulassem outras ferramentas em seu interior.

Precisamos também de um Web Server capaz de executar classes Java, Java Server Pages e que também seja um EJB Container para que nosso sistema possa ser acessado de qualquer browser independente de sistema. Para esse fim escolhemos o Tomcat Web Server, um servidor open source desenvolvido pelo Apache Jakarta Project.

Ainda está sendo estudada a utilização de um framework para o desenvolvimento do sistema, mas o padrão que utilizarei será o MVC (Model-View-Controller), tentando ao máximo separar a lógica de negócio da interface do sistema. Todas as ferramentas utilizadas são gratuitas.

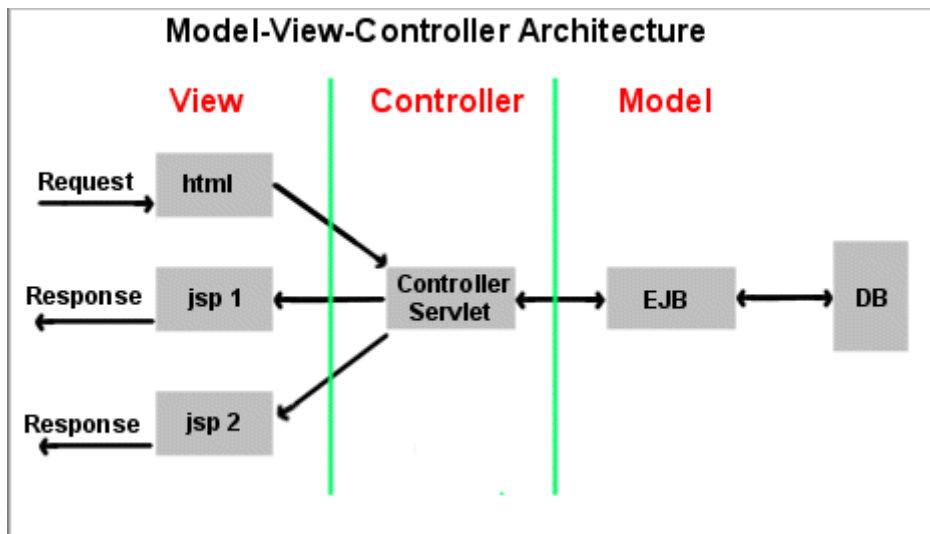


Figura 3: Padrão MVC – adotado pelo atual sistema

2.2.2. Banco de Dados (Portabilidade)

O Sistema de Gerenciamento de Banco de Dados (SGBD) escolhido foi o MySQL, por apresentar algumas características como:

- Utilizar o GNU Automake, Autoconf, e Libtool para portabilidade;
- Funciona em diversas plataformas;
- Um sistema de alocação de memória muito rápido e baseado em processos (threads);
- Suporte pleno às cláusulas SQL `GROUP BY` e `ORDER BY`. Suporte para funções de agrupamento (`COUNT()`, `COUNT(DISTINCT ...)`, `AVG()`, `STD()`, `SUM()`, `MAX()` e `MIN()`);
- Um sistema de privilégios e senhas que é muito flexível, seguro e que permite verificação baseada em estações/máquinas. Senhas são seguras porque todo o tráfego de senhas é criptografado quando você se conecta ao servidor;
- Joins muito rápidas usando uma multi-join de leitura única otimizada;
- Ser gratuito;

2.3. Regras de Negócio (Versão 1)

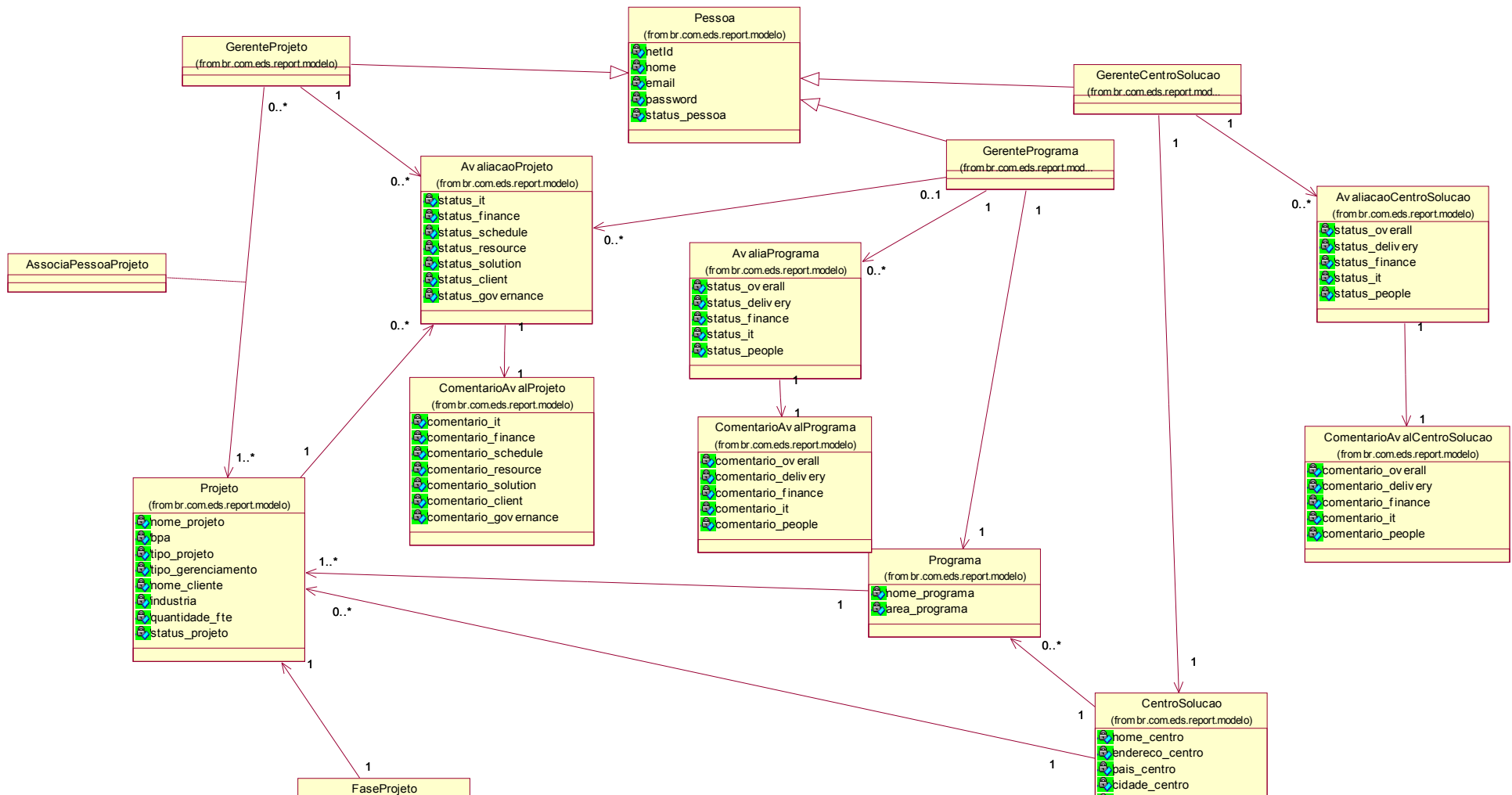
As regras de negócio são frases (com sujeito, verbo e predicado) a respeito do sistema que nos ajudam a identificar suas principais funcionalidades. Através dessas regras podemos criar o modelo de dados do sistema e entender como implementar as funcionalidades.

- Administrador cadastra usuário (líder de projeto, gerente de projetos, gerente do centro de solução);
- Administrador cadastra projeto;

- Administrador cadastra centro de solução
- Administrador edita usuário;
- Administrador edita projeto;
- Administrador edita centro de solução;
- Líder de projeto avalia projeto;
- Gerente nível dois (projetos) avalia programa;
- Gerente nível três (centro de solução) avalia centro de solução;
- As avaliações são editáveis por uma semana;
- As avaliações têm duração de uma semana;
- Gerente nível dois (projetos) pode avaliar projeto também;
- Gerente nível três (centro de solução) pode avaliar projeto e programa;
- O centro de solução tem apenas um gerente nível três;
- Uma ou mais pessoas podem avaliar um projeto;
- A mesma pessoa pode avaliar projetos diferentes;
- O gerente nível dois só pode avaliar um programa;
- O gerente nível três só pode avaliar um centro de solução;
- Um projeto só pode estar associado a um centro de solução;
- O projeto tem nome, quantidade de recursos, fase atual do projeto, cliente, tipo de contrato, tipo de indústria.
- O usuário tem username, senha, email válido e nome completo;
- Os itens a serem avaliados no projeto são Client Relationship, Schedule, Finance, Resource, Solution, Governance e IT;
- Os itens a serem avaliados no programa são Overall, Delivery, Financial, IT, People, Governance;
- Os itens a serem avaliados no centro de solução são Overall, Delivery, Financial, IT, People, Governance;
- Todos os itens podem assumir três valores: Red, Green, Yellow.
- Para cada item existem uns comentários, para maior detalhamento;
- Cada avaliação (nível um, dois e 3) terá um histórico sobre quantas semanas consecutivas está no mesmo status, por exemplo: um projeto está com duas semanas no mesmo status, Green, o sistema tem que mostrar ao usuário o número 2, mostrando que está a duas semanas neste mesmo status.
- ..

2.4. **Modelo de Classes**

Neste momento após identificar as regras de negócio, foi construído o primeiro modelo conceitual do sistema, com as regras de negócio foi identificados às classes do modelo e seus relacionamentos, esta é uma parte muito importante do sistema.



3. Cronograma de Trabalho

| Cronograma de Atividades | | | | | | | | | | | | | | | | |
|---|-------|---|---|---|-------|---|---|---|------|---|---|---|-------|---|---|---|
| Semana: | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| Mês: | março | | | | abril | | | | maio | | | | junho | | | |
| Definição do scopo inicial | ■ | ■ | | | | | | | | | | | | | | |
| Planejamento do sistema | | ■ | ■ | | | | | | | | | | | | | |
| Modelamento do sistema | | | ■ | | | | | | | | | | | | | |
| Especificação de requisitos do software | | | | ■ | | | | | | | | | | | | |
| Codificação | | | | | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | | | | |
| Refinamento do sistema | | | | | | | | | | | | | | | | |
| Teste de Integração | | | | | | | | | | | | | | | | |
| Teste de Validação | | | | | | | | | | | | | | | | |
| Documentação | | | | | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |

4. Bibliografia Inicial

- [1] Kotonya, Gerald e Sommerville, Ian, “Requirements Engineering Processes and Techniques”; [John Wiley & Sons \(UK\)](#) © 1998 (282 pages)
- [2] Buschmann, Frank, “Pattern-Oriented Software Architecture: A System of Patterns, Volume 1”; [John Wiley & Sons \(UK\)](#) © 1996 (467 pages).
- [3] The Apache Jakarta Project, “*The Apache Jakarta Tomcat 5.5 Servlet/JSP Container*”; <http://jakarta.apache.org/tomcat/tomcat-5.5-doc/index.html>; 2005.
- [4] The Apache Software Foundation, “*Struts User/Developer Guide*”; <http://struts.apache.org/userGuide/index.html>; 2005.
- [5] MySQL AB, “*MySQL Reference Manual*”; <http://dev.mysql.com/doc/>; 2005.
- [6] Gold, Russell, Hammel, Thomas e Snyder, Tom, “Test Driven Development: A J2EE Example”; [Apress](#) © 2005 (296 pages).

5. Processador de Texto

Como processador de texto para escrever o texto do trabalho final utilizaremos o Microsoft Word XP.

ESPECIFICAÇÃO DE REQUISITO DE SOFTWARE

1. Introdução

1.1. Finalidade:

A finalidade deste documento, que contém a especificação de requisitos de software é descrever os requisitos desse sistema para que ele seja desenvolvido de acordo com o que foi planejado pelos seus idealizadores.

Este documento é destinado aos desenvolvedores do sistema, aos que realizarão a manutenção do sistema e aos que realizarão possíveis atualizações no sistema.

1.2. Escopo:

A criação deste sistema tem como objetivo automatizar todo o procedimento semanal que antes era feito através de uma planilha Excel, onde havia um grande trabalho de compilação de dados fornecidos de cada projeto. O objetivo deste sistema seria alcançar outros *solution centres* que utilizam o mesmo procedimento de acompanhamento de status semanal, centralizando todo este procedimento. Hoje existe uma grande perda de tempo, cada líder de projeto preenche o status semanal que é enviado em seguida para os gerentes, que fazem as suas avaliações baseadas nas avaliações de seus líderes de projetos. Em seguida é feita a avaliação do *solution centre* baseada nas avaliações dos gerentes. Pode-se perceber que o nível de detalhamento vai diminuindo à medida que vamos subindo na hierarquia da empresa. Por fim vai todos estes dados são compilados em uma única planilha, correspondente ao *solution centre* do Rio de Janeiro. Esta planilha é enviada em um padrão específico para a EDS LATAM que é responsável por todas as operações da América latina.

Esse sistema será desenvolvido para a Web. Os usuários serão os líderes de projetos, os gerentes de projetos e os gerentes dos centros de soluções. Os usuários vão submeter às informações de status semanal do seu respectivo projeto, programa e centro de solução.

1.3. Referências:

Utilizamos como referência para este documento o Manual de Especificação de Requisitos de Software encontrado na homepage do professor Antônio Cláudio do departamento de Engenharia Eletrônica e de Computação da UFRJ. Esta norma de ERS é baseada na norma ANSI/IEEE 830 simplificada.

1.4. Resumo:

Este documento pretende mostrar como está estruturado o modelo de dados que será implementado para realização do sistema. A especificação dos requisitos de software tem como principal função esclarecer o funcionamento do sistema através das seguintes especificações, visto que estamos construindo o sistema utilizando o paradigma orientado a objeto:

- Casos de uso;
- Diagrama de Atividade;
- Diagrama de Seqüência;
- Diagrama de Classes;

Estes diagramas e modelo podem ser encontrados na seção Requisitos Específicos no item Requisitos Funcionais. O documento apresenta também uma perspectiva e funções do produto, características dos usuários, principais restrições e pressupostos, descrições das principais interfaces, requisitos de desempenho, restrições do projeto e principais atributos a que se refere o projeto.

2. Descrição Geral

2.1. Perspectiva do Produto

O sistema vai ser uma ferramenta interna da corporação, logo não vai objetivar nenhuma forma de lucro e sim agilizar e melhorar todo o procedimento atual e com isto economizar em horas trabalhadas dos funcionários e ter uma ferramenta melhor de acompanhamento semanal de projeto. Já que vamos usar uma base de dados, pode ser uma importante fonte de informação contendo o histórico de cada projeto.

2.2. Funções do Produto

Lista das Principais Funcionalidades do Sistema:

- Cadastrar Usuário;
- Editar Usuário;
- Cadastrar Projeto;
- Editar Projeto;
- Cadastrar Centro de Solução;
- Editar Centro de Solução;
- Avaliação de projeto;
- Avaliação de programa;
- Avaliação de Centro de Solução;
- Enviar mensagens de avisos;
- Gerar relatório semanal.

Descrição das Principais Funcionalidades do Sistema:

O Cadastro de Usuários é feito pelo administrador do sistema, pois este sistema não vai ser aberto a qualquer funcionário da empresa e sim para a liderança de projetos, os gerentes de projetos e o gerente do Centro de Solução. O usuário vai fornecer uma senha para entrar no sistema e seu email. Será também fornecido o tipo de usuário, ou seja, líder de projeto, gerente de projetos ou gerente do Centro de Solução.

A Edição do Usuário também é feita pelo administrador, visto que um usuário poderá no decorrer do funcionamento do sistema alcançar uma posição melhor na empresa, subindo na hierarquia, ou então alguma mudança no email.

O Cadastro de projeto também vai ser feito pelo administrador do sistema, possivelmente o administrador vai receber um formulário contendo as informações

específicas do projeto. Neste formulário vai ser indicado o PM/TL (Project Manager / Team Leader) que pode assumir mais de uma pessoa e indicará o DL (Delivery Leader – Gerente de Programa) que só pode ser uma pessoa.

A Edição de Projeto vai ocorrer quando houver alguma mudança na liderança do projeto ou então mudança na fase do projeto, na quantidade de recursos e outros atributos do projeto.

O Cadastro de Centro de Solução será feito pelo administrador do sistema. Inicialmente apenas um centro de solução será cadastrado, mas no atual modelo permite-se a utilização de mais de um centro de solução. No cadastro, que pode ser feito através de um envio de um formulário para o administrador do sistema, será definido quem vai ser o Solution Centre Manager (Gerente do Centro de Solução).

A Edição de Centro de Solução será feita quando houver alguma mudança nos atributos do centro de solução como endereço, telefone ou então uma mudança na gerência do centro de solução.

A Avaliação de Projeto vai ser feita pelos gerentes associados na hora do cadastramento de projeto (PM ou TL), podendo ser feita por mais de um. Mas apenas uma avaliação por projeto pode ser feita na semana. A avaliação de projeto pode ser modificada dentro de um período de uma semana.

A Avaliação de Programa vai ser feita pelo gerente associado na hora do cadastramento de projeto (DL), podendo ser feita apenas por uma pessoa. Apenas uma avaliação pode ser feita na semana. A avaliação de programa pode ser modificada dentro de um período de uma semana. A avaliação de programa vai ser baseada nas avaliações de projeto, pois lembrando um programa é constituído de vários projetos. Seria uma avaliação menos detalhada englobando vários projetos.

A Avaliação de Centro de Solução vai ser feita pelo gerente associado na hora do cadastramento do centro de solução (SC Manager), podendo ser feita apenas por uma pessoa. Apenas uma avaliação pode ser feita na semana. A avaliação de centro de solução pode ser modificada dentro de um período de uma semana. A avaliação de centro de solução vai ser baseada nas avaliações de programa, pois o centro de solução é constituído de vários programas que são constituídos de vários projetos. Seria uma avaliação menos detalhada englobando todo o centro de solução.

Enviar Mensagens de Aviso seria feito automaticamente pelo sistema notificando a liderança quando as avaliações fossem feitas. Essas mensagens seriam enviadas por email.

Gerar relatório semanal vai ser feita pelo administrador do sistema contendo todas as avaliações de projeto, programa e centro de solução. Ainda está sendo pensado na utilização de alguma API para gerar uma planilha no formato Excel.

2.3. Características do Usuário

Como se trata de uma ferramenta interna, de uso da corporação, o usuário tem bastante conhecimento em informática, por se tratar de uma empresa focada na área de TI. Mas o processo de preenchimento das avaliações tem que ser de uma maneira bem amigável e fácil, pois se trata de uma mudança de procedimento que antes era feito através de uma planilha Excel e agora vai ser feita na web. O usuário tem que estar satisfeito com a mudança, para facilitar esta migração.

2.4. Restrições

O sistema não possui restrições de hardware para o usuário a não ser acesso a web. É necessário ao usuário possuir conta de e-mail para realizar comunicações sobre o andamento dos processos. Recomenda-se a utilização do browser Internet Explorer, caso contrário algumas funções de Javascript e do Estilo CSS poderão ser comprometidas.

Não há restrições de software para o sistema.

3. Requisitos Específicos

3.1. Requisitos Funcionais

São os requisitos que explicam como o sistema será implementado e desenvolvido. Para este sistema utilizaremos o Projeto Orientado a Objeto para descrever as funcionalidades e requisitos do software.

Nas sessões a seguir apresentaremos os Casos de Uso, Diagrama de Atividade, Diagrama de Sequência, Diagrama de Classes e o MER (Modelo de Entidades de relacionamento).

3.2. Casos de Uso

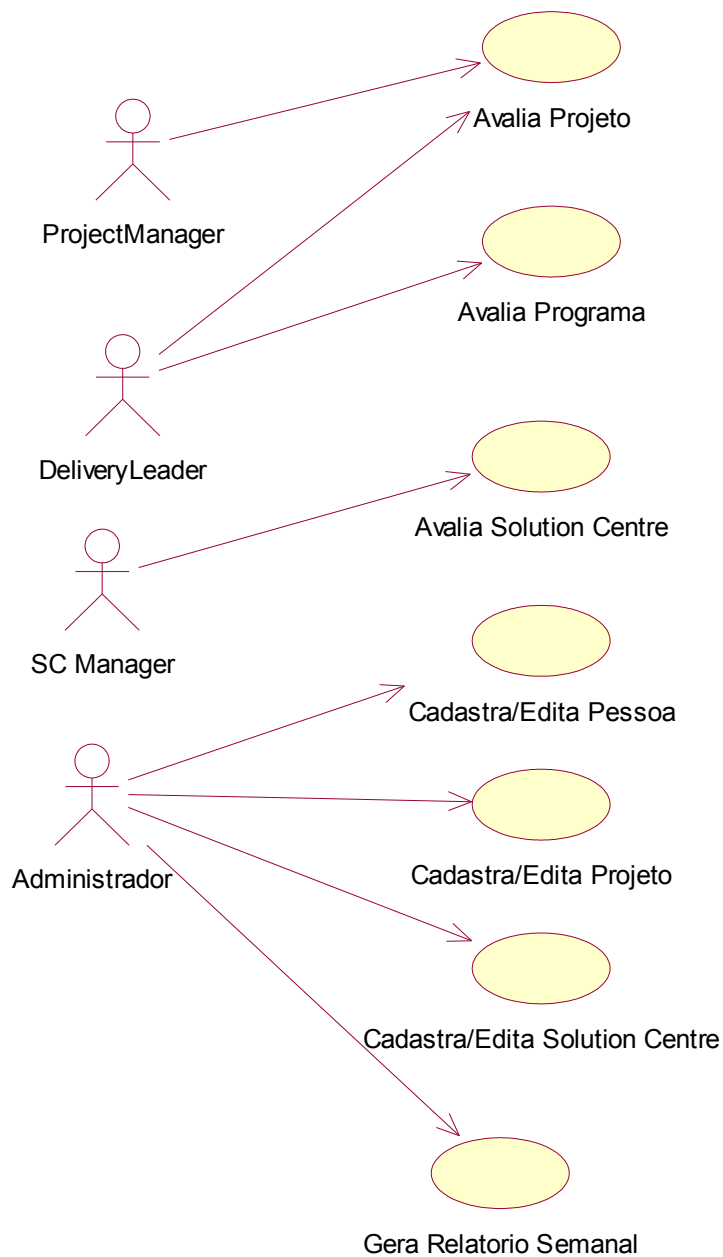


Figura 9 : Casos de Uso

3.2.1. Atores

- **Project Manager** : O nível mais baixo de gerência, está relacionado por projeto. É a pessoa responsável pelas avaliações semanais de cada projeto.
- **Delivery Leader**: Está acima do Project Manager, está associado a vários projetos. É a pessoa responsável pelas avaliações de programa.

- **SC Manager:** É a pessoa responsável por todo o centro de solução, vai ser a pessoa que vai fazer a avaliação do centro de solução.
- **Administrador:** É a pessoa responsável por fazer a manutenção do sistema, bem como o cadastro de pessoas, projeto, centro de solução. Esta função vai estar diretamente ligada ao PSO (Project Support Office) – Grupo interno de suporte a todos os outros projetos.

3.2.2. Casos de Uso

- **Avalia Programa:** O use case Avalia Programa vai permitir que os usuários (Delivery Leader) façam as avaliações de programa em uma determinada semana.
- **Avalia Projeto:** O use case Avalia Projeto vai permitir que os usuários (Program manager e Delivery Leader) façam as avaliações de projeto em uma determinada semana.
- **Avalia Solution Centre:** O use case Avalia Solution Centre vai permitir que os usuários (SC Managers) façam as avaliações de Solution Centre em uma determinada semana.
- **Cadastra/Edita Pessoa:** Este use case vai realizar as operações de cadastro e edição de pessoa (Gerentes).
- **Cadastra/Edita Projeto:** Este use case vai realizar as operações de cadastro e edição de projeto.
- **Cadastra/Edita Solution Centre:** Este use case vai realizar as operações de cadastro e edição de solution centre.
- **Gera Relatório Semanal:** Use case responsável pela geração de relatório semanal sobre as avaliações feitas.

3.3. Diagrama de Atividade

O objetivo deste diagrama é mostrar como funciona o workflow de avaliações. Primeiramente os Project Managers (Gerente de Projetos) vão fazer as avaliações de projeto. Os Delivery Leaders (Gerente de Programa) vão ser notificados sobre as avaliações de seus projetos associados ao programa e vão fazer as avaliações de programa baseado nas avaliações de projeto. E por último o gerente do centro de solução (Solution Centre Manager) vai ser notificado sobre as avaliações dos gerentes de programa e vai fazer sua avaliação do centro de solução baseado nas avaliações de programa.

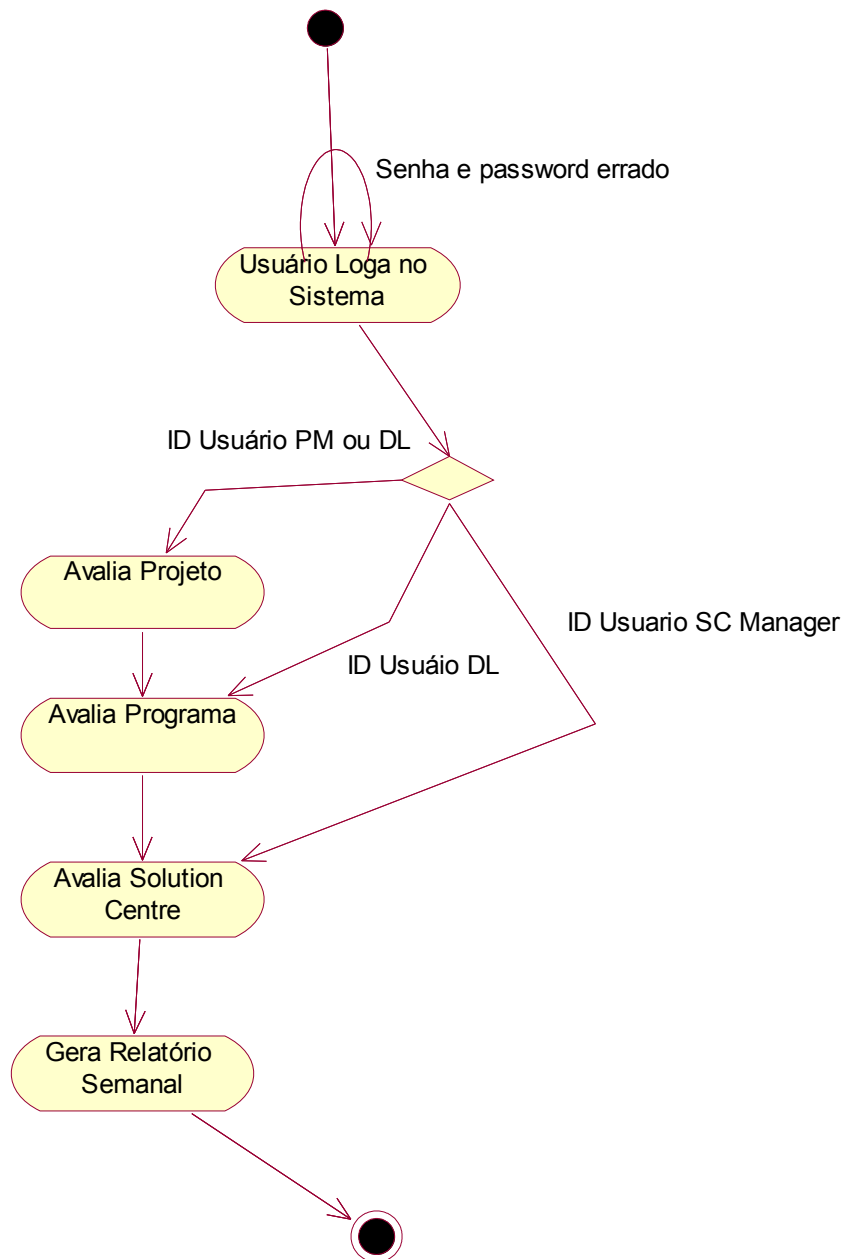


Figura 10 : Diagrama de Atividades

3.4. Diagrama de Classes

Como foi informado no planejamento o sistema a ser criado vai ser feito utilizando o framework Struts. Este framework ajuda bastante no desenvolvimento de sistemas na web. Sua principal característica é isolar ao máximo as camadas de View, Controller e Model. O desenvolvedor mantém seus esforços na lógica do negócio. Abaixo vamos ver o diagrama de classes do Struts:

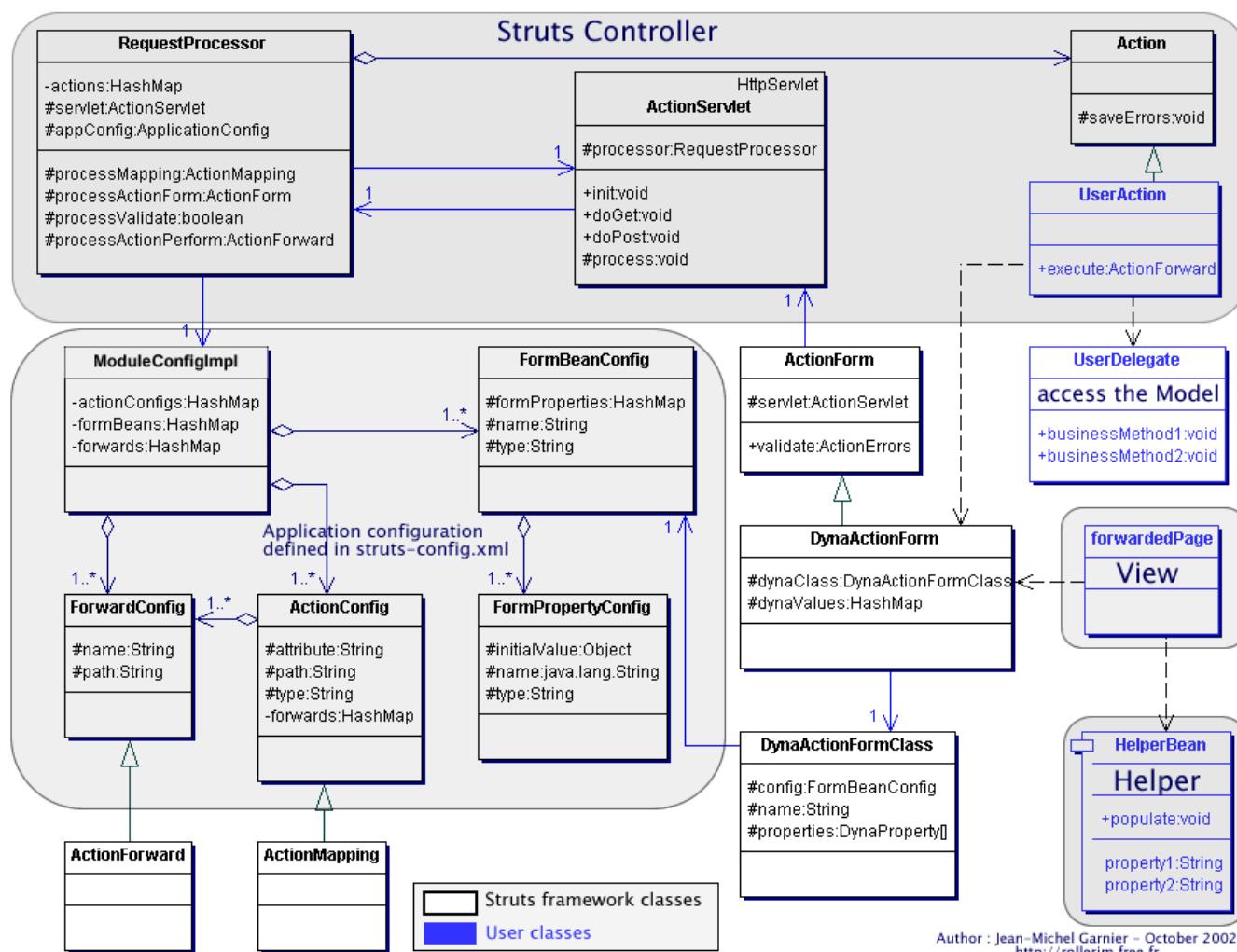


Figura 11 : Diagrama de Classes Struts

Podemos reparar por esse diagrama de classes que o desenvolvedor vai construir apenas as classes que estão em azul. Essas classes vão ter a lógica para acessar o nosso modelo de dados. Abaixo está o diagrama de classes da nossa lógica do negócio, que vai acessar a nossa base de dados:



Figura 12 : Diagrama de Classes

- **Action:** Toda vez que um usuário vai submeter informações nas páginas esta classe vai ser invocada e partir daí toda lógica do sistema vai ser executada.
- **BusinessDelegate:** Esta classe vai apresentar os métodos da lógica do negócio. Ela que vai chamar as classes de controle.
- **ControladorUtil:** Classe de controle para atividades que são comuns como inclusão, edição deleção. Vai conter a lógica para chamar as classes do repositório que vão conter informações da base de dados.

- **ControladorAvaliacao:** Classe de controle para atividades específicas do sistema. Vai conter a lógica para chamar as classes do repositório que vão conter informações da base de dados.
- **RepositorioUtil:** Classe que vai se comunicar com a nossa base de dados. Cada tabela vai ser representada por um objeto.
- **Repositorio:** Classe base que contém métodos de inclusão, edição, consulta que vão acessar o nosso banco de dados através da nossa camada de persistência implantada pelo hibernate.

3.5. Diagrama de Seqüência

O diagrama de seqüência é importante, pois mostra os relacionamentos entre objetos de uma forma seqüencial, no tempo. Como foi informado antes existe o diagrama de seqüência padrão do framework struts. Depois é mostrado o diagrama de seqüência da lógica de negócio.

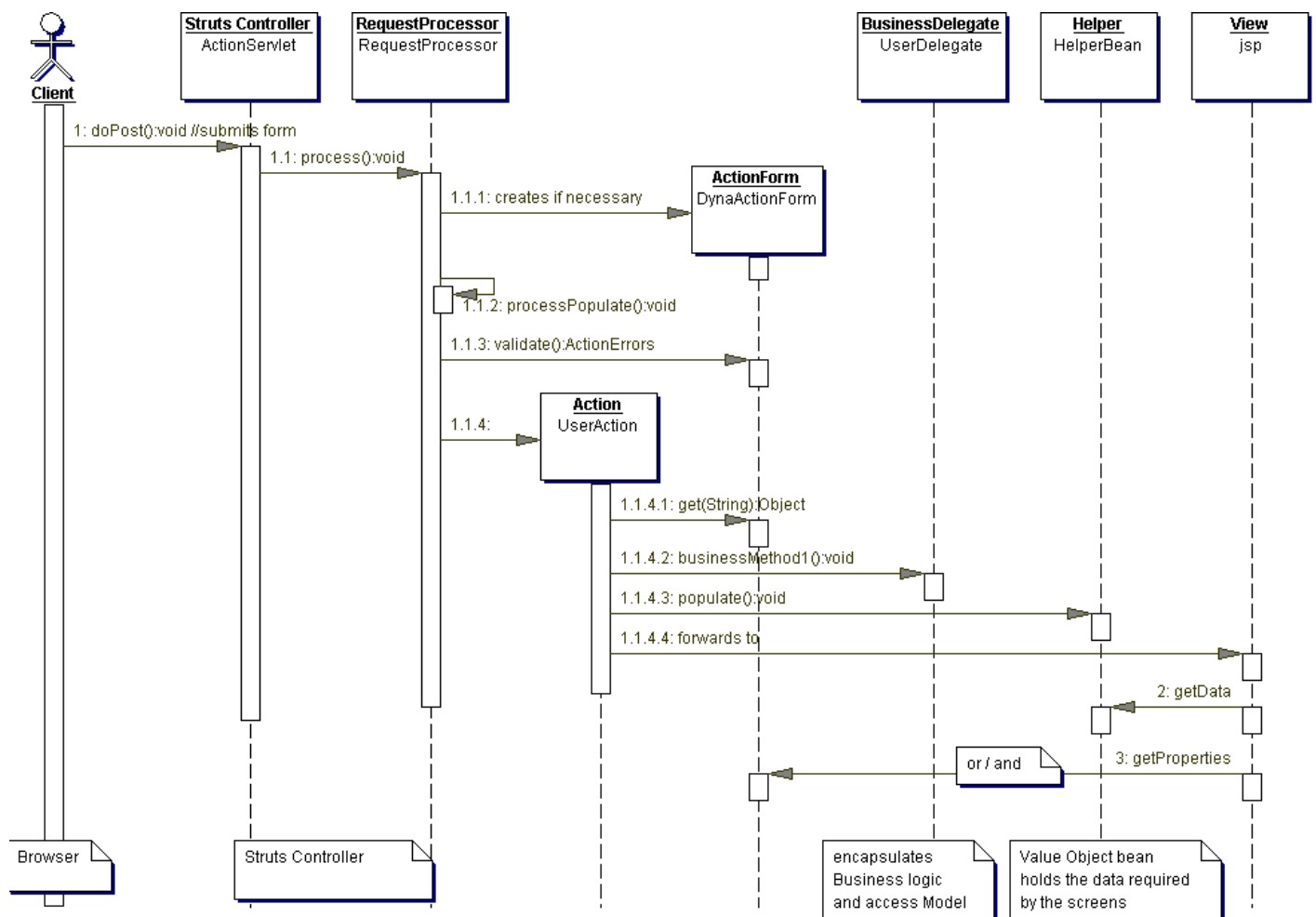


Figura 13 : Diagrama de Seqüência Struts

Agora vai ser mostrado o diagrama de seqüência gerado a partir de quando é acessada a classe BusinessDelegate. É mostrados as operações de inclusão de pessoa, projeto e centro de solução.

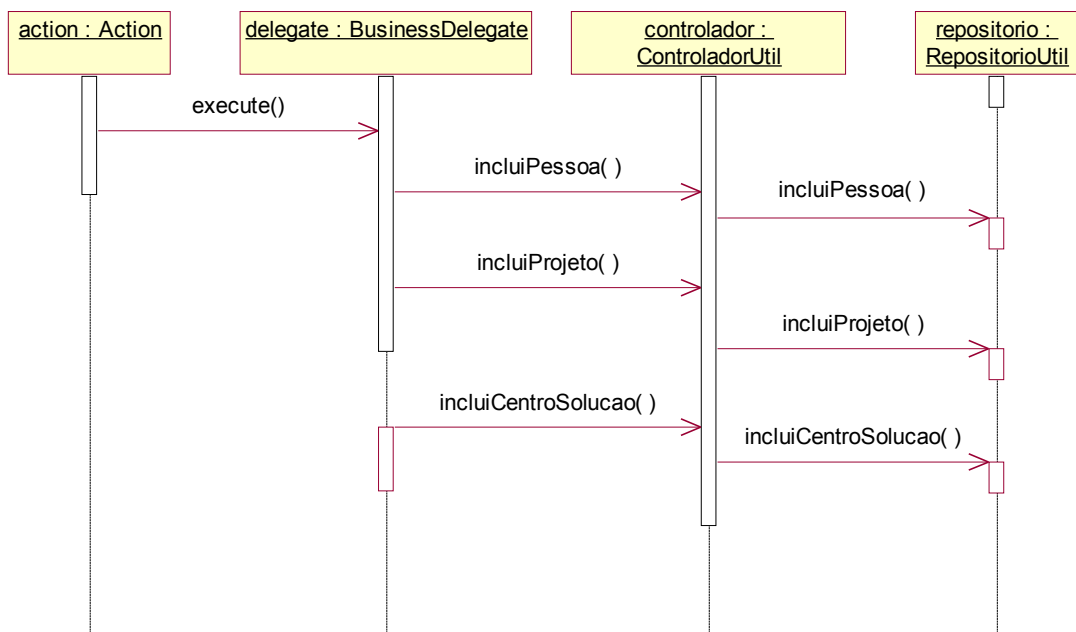


Figura 14 : Diagrama de Seqüência Business Delegate

No diagrama de seqüência abaixo é mostrados as operações de avaliação de projeto e programa e centro de solução.

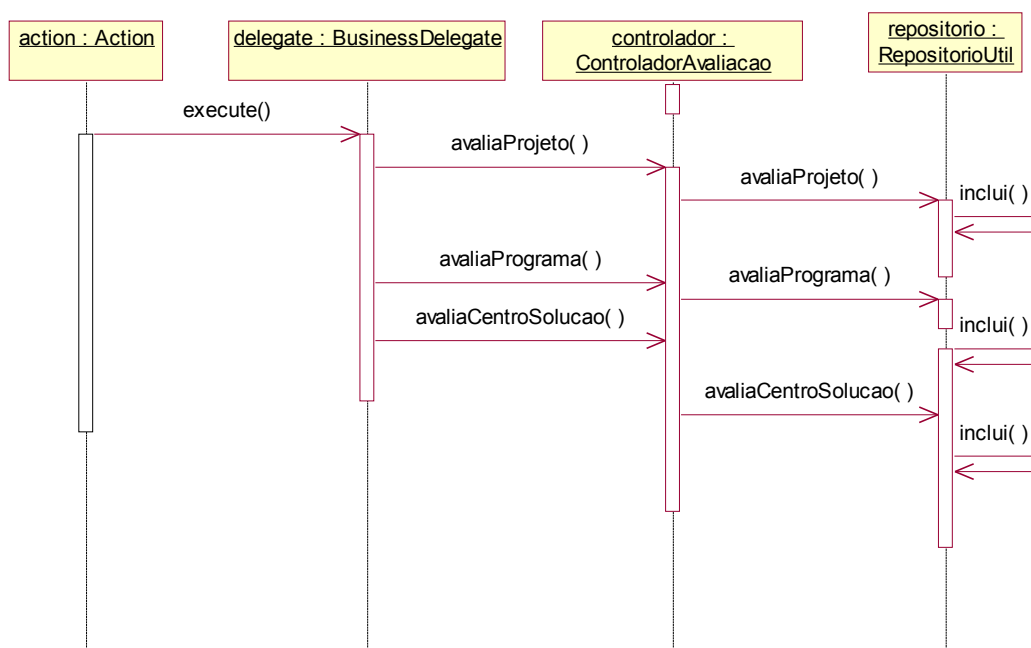


Figura 15 : Diagrama de Seqüência Inclusão Avaliação

3.6. Diagrama de Classe do Modelo

Abaixo se encontra o modelo do sistema. As classes do repositório vão acessar o banco de dados através de uma camada de persistência. Este isolamento foi feito com o Hibernate. O Hibernate faz todo o mapeamento das tabelas, construindo objetos a partir delas, inclusive os relacionamentos são mantidos. Então se fizermos alguma violação no modelo a camada de persistência vai tratar o erro não permitindo tais ações. Quando por exemplo for feita uma consulta no banco de dados, é feito através de métodos de classes do hibernate. Não vai ser necessário implementar scripts de SQL. A grande vantagem é que há um total isolamento com o tipo de banco de dados utilizado. Na página seguinte é mostrado o diagrama de classes do sistema. Estas classes representam as tabelas e seus relacionamentos.

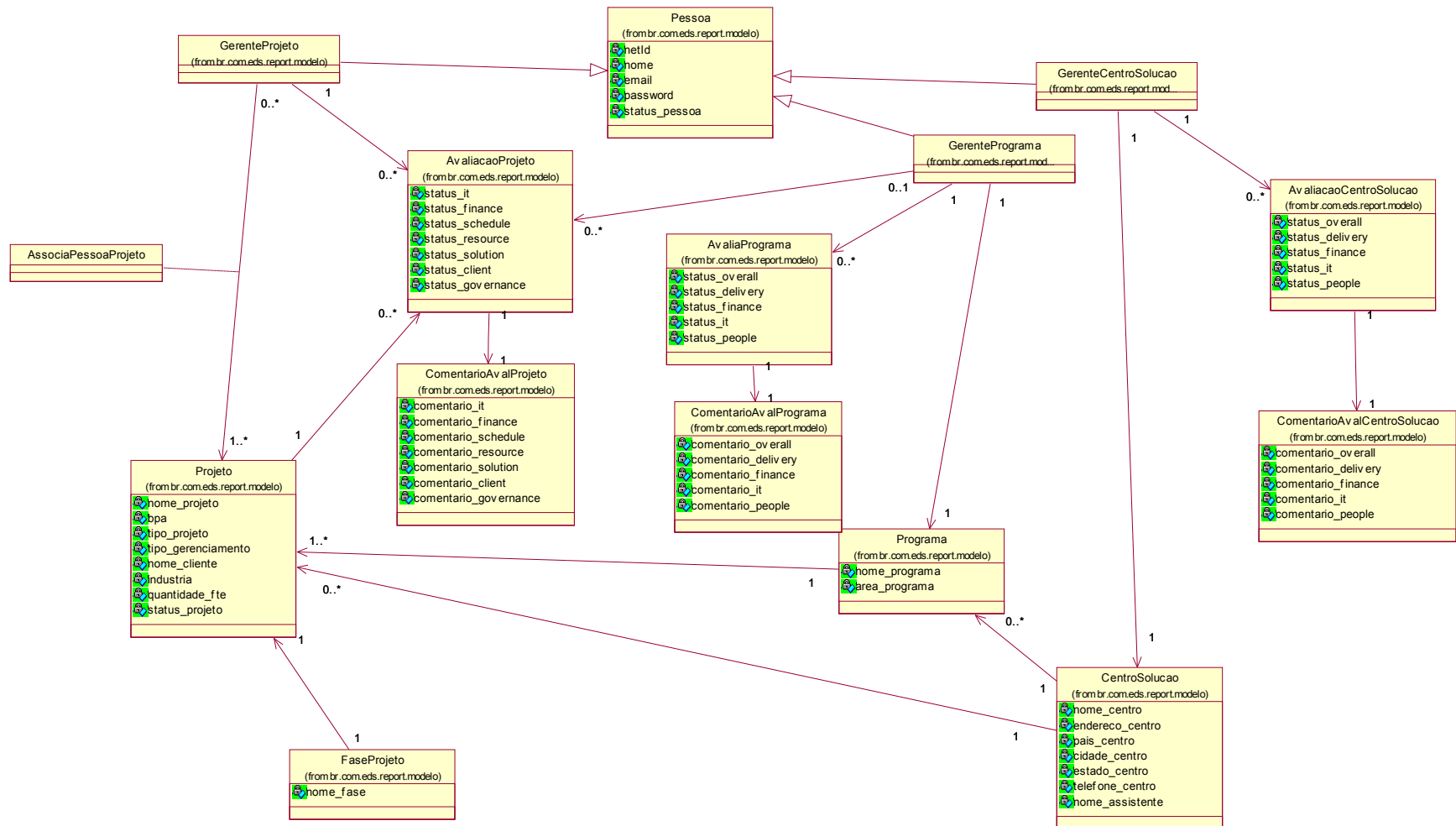


Figura 16 : Diagrama de Classes

3.7. Interfaces Externas

3.7.1. Interfaces de Usuário

As interfaces dos Usuários estão descritas na documentação de projeto.

3.7.2. Interfaces de Software

Não há interfaces de Software neste projeto.

3.7.3. Interfaces de Comunicação

A Struts Framework é um projeto open source mantido pela Apache Software Foundation. É uma implementação do design pattern MVC (Model-View-Controller) para aplicações Java com Internet. O objetivo do pattern MVC é separar de maneira clara a camada de apresentação (View) da camada de Negócio (Model).

A arquitetura MVC - Model-View-Controller (Modelo-Visualização-Controle) é um padrão que separa de maneira independente o Modelo, que representa os objetos de negócio (Model) da camada de apresentação, que representa a interface com o usuário ou outro sistema (View); e o Controle de fluxo da aplicação (Controller).

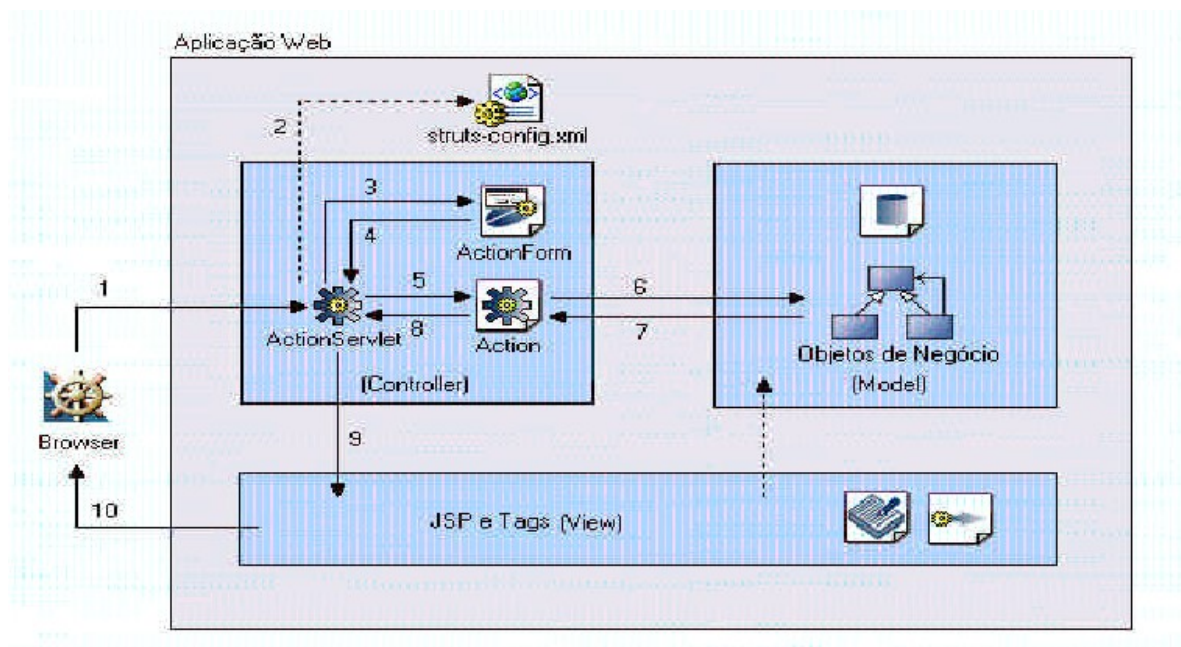


Figura 17 : Arquitetura Struts

3.7.3.1. Fluxo do Struts

1. O usuário faz uma solicitação através de uma url no browser. Ex: `http://localhost:8080/cadastro/listUsers.do`. Note que no final da url tem um `.do` que será usado para invocar (na verdade mapear) o servlet controller da struts.
2. Se for a primeira solicitação que o container recebeu para esta aplicação, ele irá invocar o método `init()` da `ActionServlet` (controller da Struts) e irá carregar as configurações do arquivo `struts-config.xml` em estruturas de dados na memória. Vale lembrar que esta passagem só será executada uma única vez, pois nas solicitações subsequentes, a servlet consulta estas estruturas na memória para decidir o fluxo a ser seguido.
3. Baseado no fluxo definido no arquivo `struts-config.xml`, e que neste momento já se encontra carregado em estruturas na memória, o `ActionServlet` identificará qual o `ActionForm` (classe para a validação dos dados) irá invocar. A classe `ActionForm` através do método `validate` irá verificar a integridade dos dados que foram recebidos na solicitação que vem do browser.
4. O controle da aplicação é retomado pelo `ActionServlet`, que verifica o resultado da verificação do `ActionForm`. Se faltou alguma coisa (campo não preenchido, valor inválido, etc), o usuário recebe um formulário html (geralmente o mesmo que fez a solicitação), informando o motivo do não atendimento da solicitação, para que o usuário possa preencher corretamente os dados para fazer uma nova solicitação. Se não faltou nenhuma informação, ou seja, todos os dados foram enviados corretamente, o controller passa para o próximo passo (`Action`).
5. O `ActionServlet`, baseado no fluxo da aplicação (estruturas já carregadas em memória) invoca uma classe `Action`. A classe `Action` passará pelo método `execute` que irá delegar a requisição para a camada de negócio.
6. A camada de negócio irá executar algum processo (geralmente popular um bean, ou uma coleção). O resultado da execução deste processo (objetos já populados) será usado na camada de apresentação para exibir os dados.
7. Quando o controle do fluxo da aplicação votar ao `Action` que invocou o processo da camada de negócio será analisado o resultado, e definido qual o mapa adotado para o fluxo da aplicação. Neste ponto, os objetos que foram populados na camada de negócio serão "atachados" como atributos na seção do usuário.
8. Baseado no mapeamento feito pelo o `Action`, o Controller faz um `forward` para o JSP para apresentar os dados.
9. Na camada de apresentação (`View`), os objetos que foram setados como atributos da sessão do usuário serão consultados para montar o html para o browser.
10. Chega o html da resposta requisitada pelo usuário.

3.8. Requisitos de Desempenho

Inicialmente o sistema não vai sofrer uma carga grande de visitas, pois está limitado apenas ao uso de pessoas ligadas à gerência de projetos.

3.9. Atributos

O sistema pretende estabelecer uma interface que seja amigável e de fácil utilização para que os usuários possam tomar decisões rápidas e seguras. Porém ao mesmo tempo tentaremos criar um design que seja atrativo e que possibilite uma melhor visualização do sistema.

Um outro ponto que vale a pena abordar é a questão na manutenibilidade do sistema. O projeto tem como objetivo desenvolver uma ferramenta onde seus aplicativos funcionem de uma forma totalmente independente, isto é, tanto o Design quanto os acessos ao Bando de Dados e as implementações das Funções do sistema serão realizados de forma isolada, garantido sua manutenibilidade. Faremos isso implementando o Pattern MVC (Model View Controller) para garantir o isolamento das camadas.

Universidade Federal do Rio de Janeiro – UFRJ
Departamento de Engenharia Eletrônica e de Computação - DEL

Documento de Projeto do Sistema

Alunos:

Roberto Gomes Gonçalves (rgg81@ig.com.br)

Orientador:

Prof.: Antônio Cláudio Gómez de Sousa

1. Introdução

1.1 Finalidade

A finalidade deste documento que contem o Projeto do Sistema é descrever como o sistema será implementado, apresentar as técnicas de divisão de projeto que serão utilizadas bem como mostrar o desenho do projeto através de ferramentas de UML (Universal Markup Language).

Este documento é destinado aos desenvolvedores, revisores e àqueles que manterão o sistema.

1.2 Escopo

O sistema de acompanhamento de projeto semanal visa substituir uma planilha no formato Excel, que era preenchida pela gerência da empresa, para um sistema web. As vantagens para tal mudança são grandes, entre elas o histórico das informações vai permitir fazer uma análise mais detalhada sobre o andamento dos projetos, a geração

automática de relatórios e a possível utilização do sistema para outros centros de solução.

1.3 Definições e Acronismos

Algumas siglas que serão utilizadas adiante:

- **ERS**: Documento de Especificação de Requisitos de Software.
- **BC**: Business Component – Componente de Negócio.
- **JSP**: Java Server Pages.
- **JDBC**: Java Database Connector.
- **SGBD**: Sistema de Gerência de Banco de Dados.

Referências

No restante do documento faremos utilização de conceitos já definidos anteriormente nos documentos de **Planejamento e Especificação de Requisitos do Sistema**.

2. Arquitetura do Sistema

Nesta sessão descreveremos o projeto da arquitetura do sistema a qual usaremos como base para implementar todos os processos descritos nos casos de uso.

Abaixo se encontra uma visão geral da arquitetura do projeto. Percebe-se aqui a utilização do padrão MVC para separação de camadas (utilizando o framework Struts), os design patterns Business Delegate e Controller e a utilização do Hibernate para fazer a camada de persistência de dados. A grande vantagem de utilizar esses padrões é na fácil manutenção do sistema e também da grande reutilização de código.

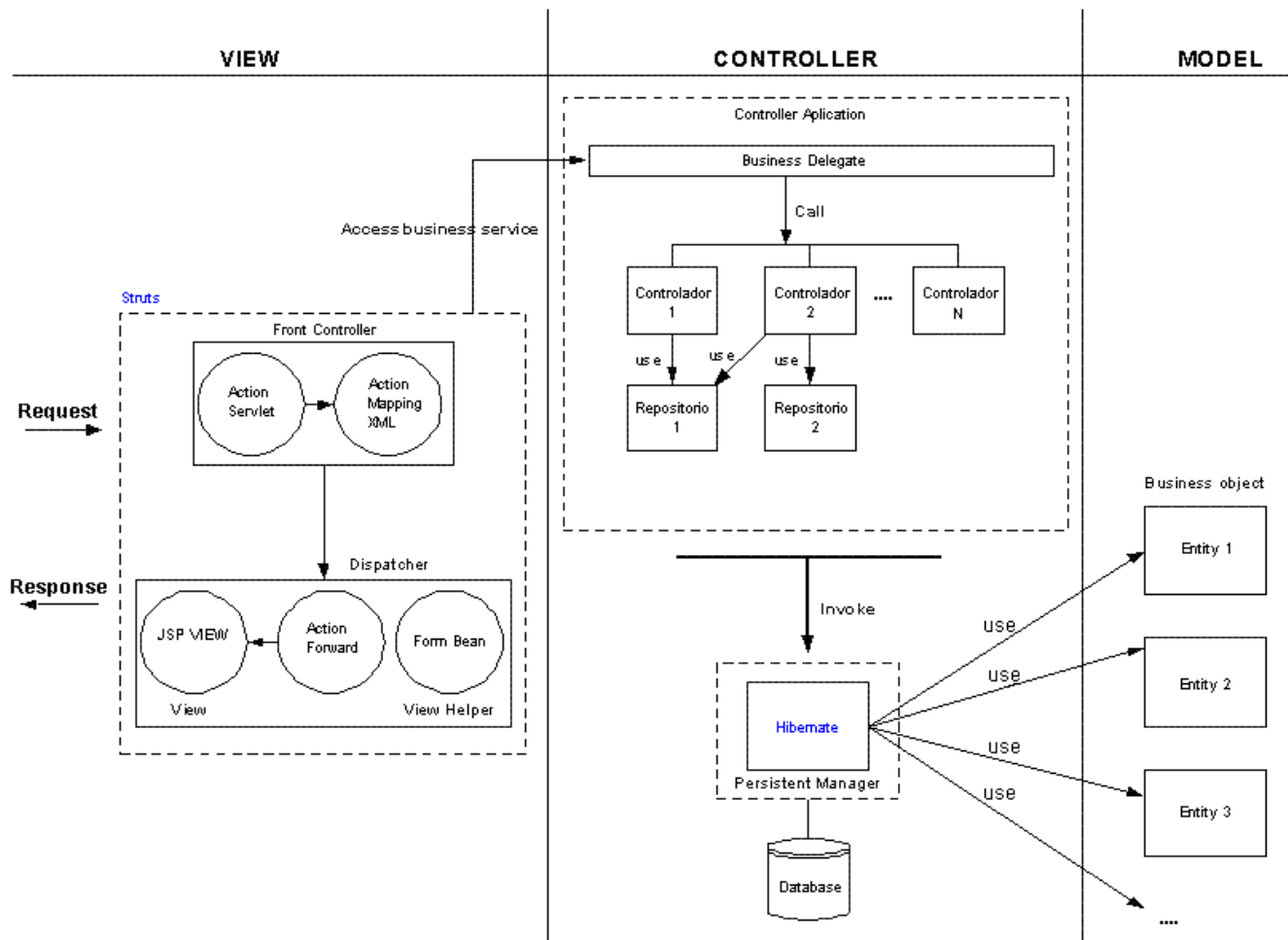


Figura 18 : Arquitetura Geral do Sistema

2.1 Descrição dos Componentes da Arquitetura

A seguir serão descritos os componentes que estão envolvidos com um quadrado tracejado na figura da arquitetura;

- Struts;
- Controller Application;
- Persistent Manager.

2.1.1 Struts

O Action Servlet é utilizado para interceptar todas as requisições. No arquivo struts-config.xml existem mapeamentos entre as classes actions com endereços urls para facilitar a manutenção do código. A seguir segue um exemplo, podemos ver que o usuário ao entrar com o endereço `www.localhost:8080/status_report/Login.do` a action que vai interceptar o request vai ser a classe `LoginAction` do pacote `br.com.eds.report.actions` :

```
<action path="/Login"
  type="br.com.eds.report.actions.LoginAction"
  scope="session"
  validate="true"
  name="LoginForm"
  input="/jsp/dadosInvalidos.jsp">
  <exception key="errors.netid.invalid" path="/jsp/dadosInvalidos.jsp" type="br.com.eds.report.excecoes.ObjetoInexistenteException"/>
  <exception key="errors.password.invalid" path="/jsp/dadosInvalidos.jsp" type="br.com.eds.report.excecoes.PasswordInvalidoException"/>
  <forward name="success" contextRelative="true" path="/Home.do"/>
</action>
```

Figura 19 : Mapeamento Classe/Endereço

O Form Bean é mapeado em um arquivo xml também, neste arquivo os campos são identificados e as devidas validações são feitas (outra grande vantagem do struts é a fácil validação dos campos de formulários). Podemos ver na figura acima, no mapeamento das Actions, é descrito também a validação dos dados nas propriedades `validate` e `name` – nome do formulário. Abaixo se encontra o mapeamento dos nomes dos formulários com as classes correspondentes.

```
<form-beans>
  <form-bean name="LoginForm" type="br.com.eds.report.form.LoginForm"/>
  <form-bean name="EditaPessoaForm" type="br.com.eds.report.form.EditaPessoaForm"/>
  <form-bean name="EditaProjetosForm" type="br.com.eds.report.form.EditaProjetosForm"/>
  <form-bean name="EditaSCForm" type="br.com.eds.report.form.EditaSCForm"/>
  <form-bean name="ConsultaPessoaForm" type="br.com.eds.report.form.ConsultaPessoaForm"/>
  <form-bean name="CadastroPessoaForm" type="br.com.eds.report.form.CadastroPessoaForm"/>
  <form-bean name="CadastroProjetoAssignForm" type="br.com.eds.report.form.CadastroProjetoAssignForm"/>
  <form-bean name="CadastroSCForm" type="br.com.eds.report.form.CadastroSCForm"/>
  <form-bean name="AvaliaProjetosForm" type="br.com.eds.report.form.AvaliaProjetosForm"/>
  <form-bean name="AvaliaProgramaForm" type="br.com.eds.report.form.AvaliaProgramaForm"/>
</form-beans>
```

Figura 20 : Mapeamento Form Bean

Quando o usuário submete informações para o servidor como já foi mostrado no diagrama de seqüência geral na ERS, as Actions interceptam a requisição e fazem o devido processamento da informação. Após a lógica do negócio ser executada, a Action vai acionar a página de resposta que está mapeado no arquivo de configuração do struts (struts-config.xml). Pode ser visto na figura 2 na tag forward. Outra grande facilidade do struts é o tratamento de exceções. Não precisamos nos preocupar em fazer o tratamento no código através de try e catch. Nas Actions quando houver alguma exceção, de novo o struts-config.xml é acionado (pode-se perceber que este arquivo é o coração do struts). No mapeamento das Actions existe uma tag exception que identifica o tipo de exceção e faz o direcionamento para a devida página de erro.

2.1.2 Controller Application

O objetivo de fazer essa divisão de camadas do componente Controller Application é fazer um maior desacoplamento da camada de negócios com a camada **View**. Desta forma fica mais transparente para a camada **view** o seu acesso aos dados do sistema. A classe BusinessDelegate vai conter interfaces que vão ser implementadas pelas classes Controladoras e estas vão acessar o componente persistent manager para ter acesso à base de dados do sistema. Fazendo esta divisão de camadas não vai ser necessário utilizar nenhuma lógica na camada view para ter acesso aos dados, isolando toda a dificuldade de acesso.

2.1.3 Persistent Manager

O Hibernate é uma ferramenta de mapeamento objeto/relacional para Java. Ela transforma os dados tabulares de um banco de dados em um grafo de objetos definido pelo desenvolvedor. Usando o Hibernate, o desenvolvedor se livra de escrever muito do código de acesso a banco de dados e de SQL que ele escreveria não usando a ferramenta, acelerando a velocidade do seu desenvolvimento de uma forma fantástica. Mas o framework não é uma boa opção para todos os tipos de aplicação. Sistemas que fazem uso extensivo de stored procedures, triggers ou que implementam a maior parte da lógica da aplicação no banco de dados, contando com um modelo de objetos “pobre” não vai se beneficiar com o uso do Hibernate. Ele é mais indicado para sistemas que

contam com um modelo rico, onde a maior parte da lógica de negócios fica na própria aplicação Java, dependendo pouco de funções específicas do banco de dados.

O Hibernate utiliza um arquivo de configuração chamado hibernate.cfg.xml. Neste arquivo são configurados o endereço do servidor do banco de dados, o tipo de dialeto que o hibernate vai traduzir para uma linguagem universal. Essa é uma das grandes vantagens do hibernate, pois ele possibilita que seja feita uma mudança, por exemplo, passar de um SQL Server para um Oracle apenas alterando o arquivo de configuração, sem mudanças no código. Abaixo está um trecho do arquivo xml exemplificando o que foi dito:

```
<!-- Property dialect: The classname of a Hibernate Dialect - enables certain
platform dependent features. -->
<property name="dialect">net.sf.hibernate.dialect.SQLServerDialect</property>

<!-- Property connection: Pass the necessary information to
DriverManager.getConnection(). -->
<property name="connection.driver_class">net.sourceforge.jtds.jdbc.Driver</property>
<property name="connection.url">jdbc:jtds:sqlserver://brrjads01:1433/status_report</property>
<property name="connection.username">szf1f3</property>
<property name="connection.password">szf1f3</property>
```

Figura 21 : Hibernate Configuration File

Cada tabela do modelo de dados é mapeado em uma classe Java e um respectivo arquivo xml contendo o mapeamento dos atributos dos objetos com as colunas da tabela. Isto é feito de forma automática com o uso de Xdoclet. Esta poderosa ferramenta permitiu a geração automática dos mapeamentos das classes. Xdoclet é uma extensão do javadoc, permiti colocar comandos nas classes na forma de comentários que são lidos pela ferramenta. Abaixo está um exemplo do uso de xdoclet:

```
/**
 * @hibernate.id generator-class="assigned" type="java.lang.String" column="netID" length="6" unsaved-value="null"
 */
public String getNetID() {
    return netID;
}
```

Figura 22 : Hibernate com Xdoclet

2.2 Infra-estrutura do projeto

A seguir vão ser mostradas as classes de infra-estrutura do projeto, que incluem as funcionalidades de controle de sessão, envio de emails, conexão com o banco de dados, e log do sistema.

2.2.1 Controle de Sessão

A seguir é mostrado o diagrama de seqüência para criação do session bean:

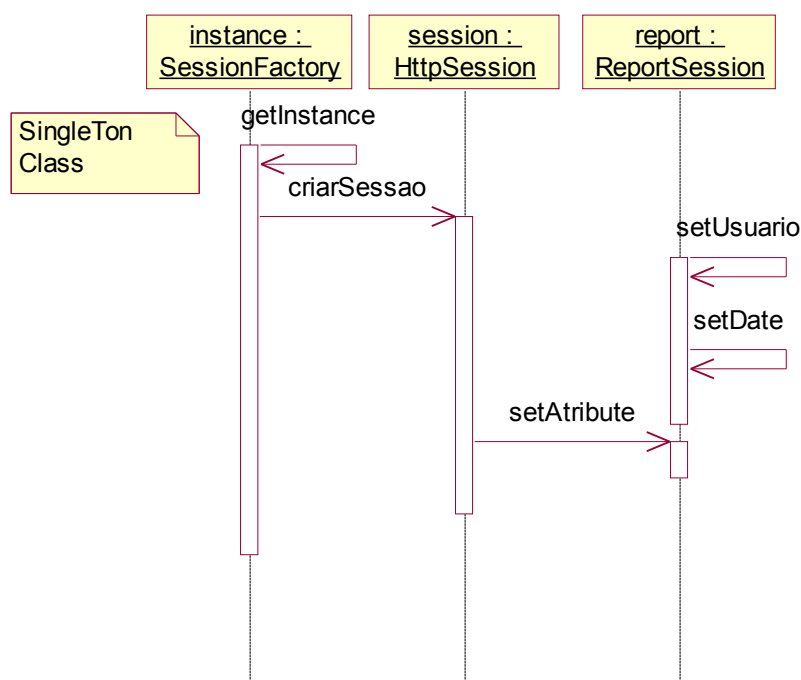


Figura 23 : Criação de um session bean

A classe SessionFactory é uma singleton ou seja o seu construtor é private, só é possível criar uma instância da classe enquanto estiver valendo uma sessão. Após a sessão ser criada, o bean report do tipo ReportSession popula seus atributos com as

informações específicas de cada usuário. Após isso o bean report é armazenado na sessão. De qualquer Action é possível carregar este session bean.

Agora vai ser mostrado o diagrama de seqüência para validação de uma sessão:

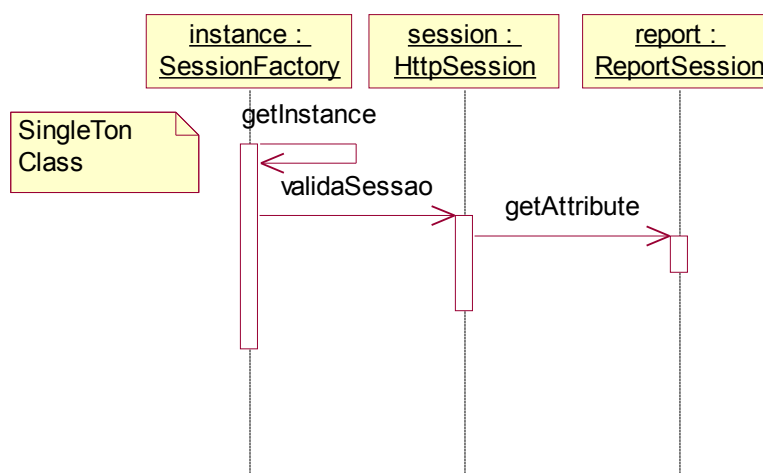


Figura 24 : Validação de Sessão

A operação `validaSessao` confere se existe uma sessão existente e se ainda é válida. Isto impede que usuários não cadastrados tentem acessar as páginas e também impede de o usuário tentar acessar a página fora do fluxo normal do sistema. A operação `getAttribute` recupera as informações do usuário. Pode-se por exemplo fazer uma saudação ao usuário em cada página. Estas informações de sessão são acessíveis nas páginas JSPs.

2.2.2 Envio de emails

Como foi dito anteriormente uma importante funcionalidade deste sistema é o envio de mensagens. Sempre que alguma avaliação for feita, é disparada uma mensagem de aviso informando ao grupo de suporte, ao usuário (confirmando a avaliação) e ao nível acima de gerência. Por exemplo: imaginemos que o líder do projeto da General Motors faz a sua avaliação semanal, logo após uma série de mensagens serão enviadas. Uma mensagem ao grupo de suporte (PSO), uma ao usuário e outra para o Delivery Leader (gerência nível 2 – gerencia vários subprojetos da GM). Assim o DL (Delivery Leader) poderá acompanhar quando as avaliações forem feitas.

Quando todas as avaliações dos projetos que o DL gerencia forem feitas ele poderá fazer a avaliação do programa.

Para implementar esta funcionalidade foi usada uma API de Java chamada JavaMail. Esta API apresenta classes prontas que facilitam bastante o desenvolvimento deste tipo de funcionalidade. Abaixo se encontra a hierarquia de classes desta API:

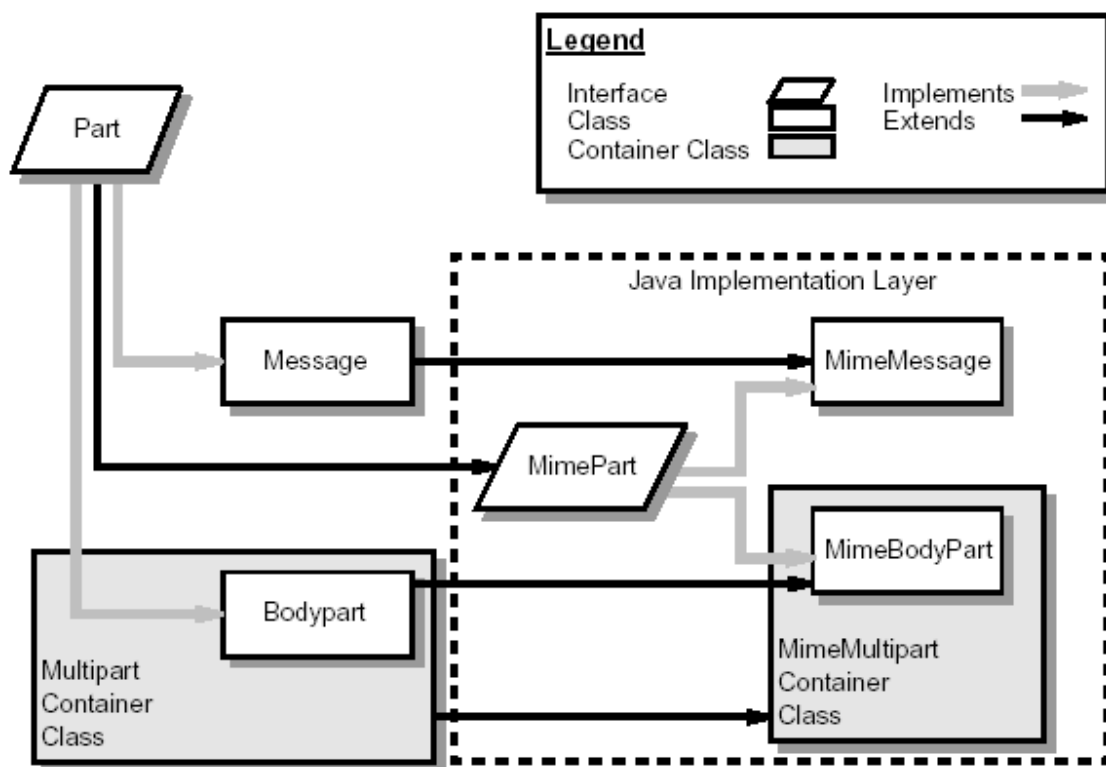


Figura 25 : JavaMail API

Para fazer uso deste framework é necessário seguir alguns passos básicos que estão listados abaixo:

- Criar uma instância da classe Message;
- Definir quem será o remetente, o destinatário e o conteúdo da mensagem (assunto, corpo). Todos são atributos da classe Message;
- Criar um objeto do tipo Session para fazer a autenticação do usuário, e também definir o servidor e o tipo de transporte da mensagem;
- Enviar a mensagem.

2.2.3 Conexão com o banco de dados

Para fazer a conexão com o banco de dados foi adotado o hibernate. Ele vai ser utilizado como *persistent manager*. Agora vai ser mostrado as classes envolvidas no processo de conexão com o banco de dados.

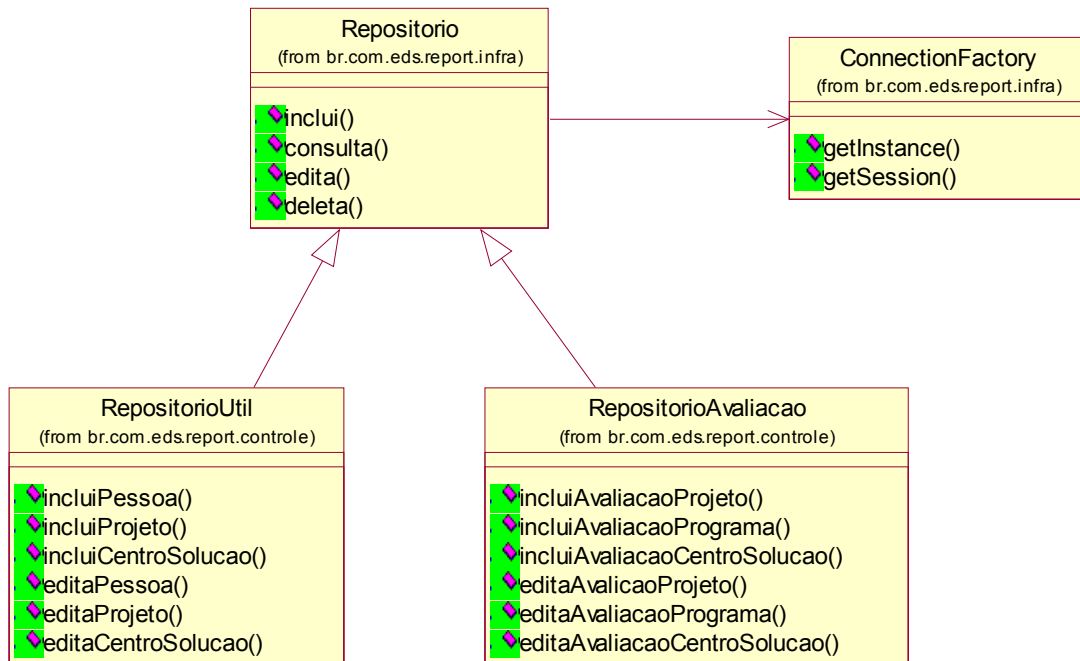


Figura 26 : Diagrama de classes conexão BD

A classe **ConnectionFactory** vai ser a responsável pela leitura do arquivo de configuração do hibernate e devolver um objeto de conexão com o banco de dados. Sendo sucedido este processo, os métodos da classe **Repositorio** vão utilizar este objeto de conexão para fazer operações no banco de dados. Estas operações incluem edição, inclusão, consulta e deleção. São as operações mais simples. As classes **RepositorioAvaliacao** e **RepositorioUtil** vão ser responsáveis por fazer operações mais complexas pois vão exigir o uso de uma ou mais operações. Por exemplo para incluir uma avaliação de projeto é necessário incluir em duas tabelas, uma de associação e outra de avaliação de projeto.

2.2.4 log do sistema

Hoje em dia é muito comum utilizar uma API de logging de mensagens como o Log4j para indicar os comportamentos de uma aplicação. O Log4j é um projeto open source distribuído pela Apache Software License e foi portado por autores independentes a C, C++, Python, ruby, Eiffel.

Utilização do log4j:

- 1 – Configurar o Log4j para o seu ambiente. O Log4j oferece diversas maneiras de configuração.
- 2 – Em toda classe que você deseje utilizar logging, recupere um objeto Logger invocando o método `Logger.getLogger()` passando uma String, normalmente o nome da classe completo. O objeto Logger é geralmente declarado como `static final`.

Logger

Uma das principais vantagens de qualquer API de logging sobre a simples `System.out.println`, reside na possibilidade de desabilitar certas declarações de log enquanto outras permitidas são exibidas.

O logger é o núcleo do processo de logging.

No Log4j existem 5 níveis normais de logger disponíveis:

- `static Level DEBUG`: o nível DEBUG designa informações úteis para debugar uma aplicação.
- `static Level INFO`: o nível INFO designa informações que focalizam no progresso da aplicação.
- `static Level WARN`: o nível WARN designa situações potencialmente prejudiciais.
- `static Level ERROR`: o nível ERROR designa eventos de erro que poderão permitir que a aplicação continue funcionando.
- `static Level FATAL`: o nível FATAL designa eventos de erro severos que presumivelmente poderão interromper a aplicação.
- Os métodos ALL e OFF são intencionados somente para o gerenciamento.

No sistema de acompanhamento de projeto todas as exceções serão registradas em um arquivo de LOG usando o Log4j.

3.1 Decomposição

Nesta sessão abordaremos como será feita e no que será baseada a divisão do sistema em módulos menores a fim de possibilitar desenvolvimento concorrente pela equipe.

3.1.1 Decomposição em Módulos

A decomposição dos módulos foi baseada no diagrama de casos de uso mostrado na Especificação de Requisitos de Software. Neste ponto vai ser mostrado com detalhes a implementação de cada caso de uso. Vai ser mostrado também todo o comportamento do sistema quando o usuário se *loga*. Apesar da simplicidade da ação, o usuário digitar uma senha e o netid, o sistema tem que identificar a *role* do usuário no sistema e carregar as informações específicas para cada tipo de usuário. Serão mostrados as telas, um diagrama de seqüência representando a ordem temporal das ações que levam a conclusão de cada caso de uso. Algumas funcionalidades da arquitetura (criação de sessão, conexão com o banco de dados) não vão ser detalhadas neste ponto pois já foram mostradas nos itens acima.

3.1.2 Módulo Lajem

3.1.2.1 Descrição da Funcionalidade

Este é o ponto inicial do sistema, o usuário enviará informações do seu netid (código universal identificador do empregado) e sua senha previamente cadastrada no sistema. Então informações específicas do usuário são mostradas em sua homepage. Esta é a página principal do sistema. Nela é possível acessar todas as funcionalidades

permitidas para cada tipo de usuário. Ou seja não é possível para um usuário comum cadastrar uma pessoa ou um projeto. Só o administrador poderá ter acesso a essas funcionalidades. O sistema suportará quatro tipos de usuários: gerentes ao nível de projeto, gerente de programa (englobando vários projetos), gerente do centro de solução e o administrador do sistema.

3.1.2.2 Protótipo das Telas

Tela de Login

Nesta tela o usuário entrará no sistema.

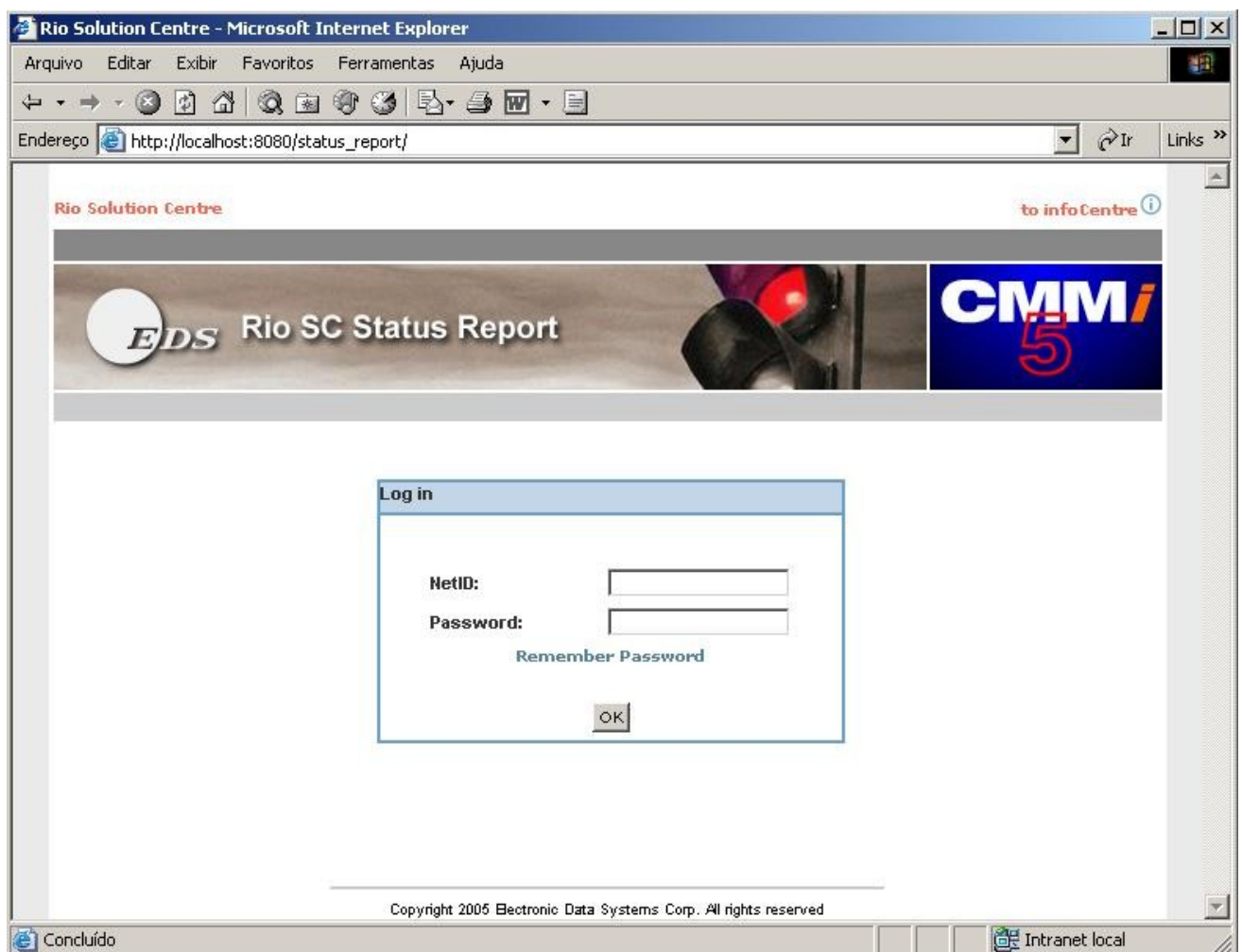


Figura 27 : Tela de Login

Definições dos Campos da Tela

| Nome do Campo | Descrição do Campo | Regra de Exibição |
|----------------------|--|--------------------------|
| NetID | Identificação de 6 dígitos alfanuméricos. | Sempre |
| Password | Senha cadastrada pelo usuário. | Sempre |
| Remember Password | Ao ser selecionado envia a senha por email | Sempre |

Tela Inicial para gerentes de projetos

Nesta tela o gerente de projeto vai poder ter uma visão geral de suas atividades no sistema. Ele vai poder ter acesso a todos os projetos em que ele é gerente. Vai poder visualizar suas avaliações antigas e atuais e no caso de projetos em que existam mais de um gerente associado, vai poder visualizar a avaliação do outro gerente.

Rio Solution Centre Hello, Joaquim Manuel to info Centre

Home Register Edit Evaluate Report Help

EDS Rio SC Status Report **CMM**

Evaluations to Do

Evaluations Made

General Motors

Start Finish

02/10/05 08/10/05

Projects Evaluations

General Motors

Edit

Project

People

General Motors

Evaluated by: Joaquim Manuel

| Criteria | Status |
|------------|--------|
| IT | G |
| Schedule | G |
| Finance | Y |
| Resource | R |
| Governance | G |
| Solution | Y |
| Client | G |

Last Evaluations Made

General Motors

Evaluated by: Joaquim Manuel

| Criteria | Status |
|------------|--------|
| IT | G |
| Schedule | G |
| Finance | G |
| Resource | G |
| Governance | G |
| Solution | G |
| Client | G |


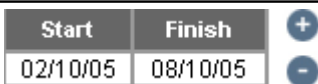
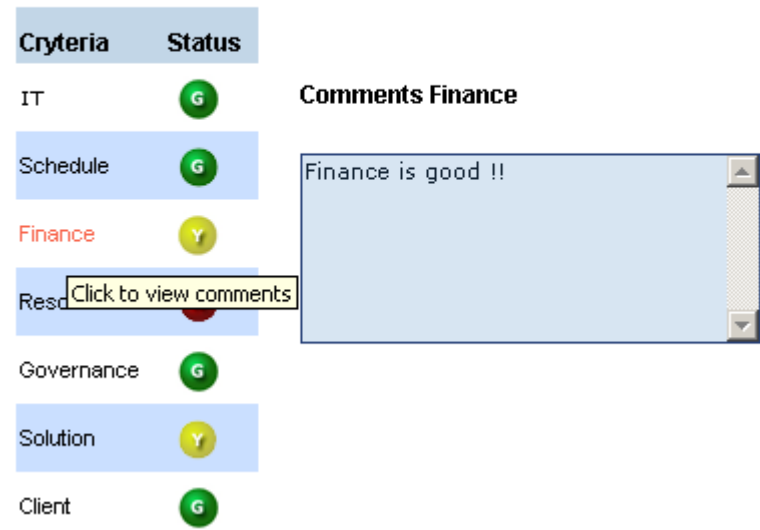
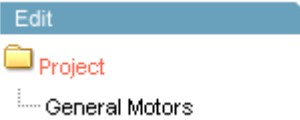

Copyright 2005 Electronic Data Systems Corp. All rights reserved

Figura 28: Homepage gerente nível 1

No caso mostrado acima o usuário Joaquim Manuel só é gerente do projeto General Motors. Manuel já fez a sua avaliação da semana. A página é dividida em 4 frames. O primeiro está no topo, o segundo e o terceiro nas laterais esquerda e direita e o frame central. O frame central mostra as avaliações feitas na semana atual e na semana anterior contendo também uma tabela indicando o período das avaliações. Já o

frame esquerdo indica as avaliações que estão pendentes e as avaliações feitas (representados pelas gifs **Evaluations to Do** e **Evaluations Made**). Percebe-se que a gif **Evaluations to Do** está vazia pois todas as avaliações já foram feitas. Abaixo da gif **Evaluations Made** está o nome do projeto com um lápis do lado, indicando que ele ainda pode editar a sua avaliação (apenas clicando sobre o lápis). No frame direito é mostrada também a avaliação que foram feitas e as que faltam fazer, apesar de parecer redundante neste caso vai ser útil para os gerentes de programa. A gif **Edit** mostra as opções que o usuário pode editar.

Definições dos Botões/Links da Tela

| Botão / Link | Ação desempenhada |
|---|---|
|  | O lápis leva para a página de edição da avaliação. |
|  | O “+” quando é pressionado adianta uma semana e o “-” atrasa. |
| <p>General Motors Evaluated by: Joaquim Manuel</p>  | Cada critério de avaliação é um link que leva para a mesma página. Mas ao ser pressionado aparece o comentário da avaliação deste critério. No caso ao lado foi pressionado o critério Finance . |
|  | A pasta representada pelo nome project expande quando o usuário clica sobre ela, mostrando todos os projetos gerenciados pelo usuário caso queira editá-los. |
|  | Ao clicar sobre o link representado pela gif ao lado, é possível alterar o nome, email e password do usuário. |

Tela Inicial para gerentes de programa

Na tela inicial do gerente de programa (gerente nível 2) aparecem todos os projetos que são pertencentes ao programa e as avaliações feitas pelos gerentes de projetos. Podemos perceber pela figura abaixo que as avaliações dos projetos AirDNS e General Motors já foram realizadas faltando somente à avaliação do programa. Outra diferença é a permissão de cadastrar pessoas para os gerentes de nível 2.

Rio Solution Centre Hello, Roberto Gomes to infoCentre

Home Register Edit Evaluate Report Help

EDS Rio SC Status Report **ISO 9001:2000 CERTIFIED**

Currents Evaluations Made

Evaluations to Do

- Program Evaluation

Evaluations Made

- AirDNS
- General Motors

Start **Finish** **+** **-**

02/10/05 08/10/05

Program Evaluations

- AirDNS ✓
- General Motors ✓

Register

- People

Edit

- Project
- People

AirDNS

Evaluated by: Roberto Gomes

| Criteria | Status |
|------------|--------|
| IT | G |
| Schedule | R |
| Finance | R |
| Resource | G |
| Governance | G |
| Solution | Y |
| Client | Y |

Comments Schedule

Schedule are bad

Figura 29 : Homepage Gerente de Programa

Tela Inicial para gerente do centro de solução

As diferenças nesta tela inicial é que o gerente do centro de solução tem a visão das avaliações dos gerentes de programa, como pode ser visto abaixo na gif **DL Evaluations**, onde aparece à avaliação do DL (gerente do programa – Delivery Leader) Roberto Gomes já efetuada. Outra diferença é a possibilidade do gerente do centro de solução poder registrar um projeto no sistema.

The screenshot displays the 'Rio Solution Centre' interface. At the top, there's a navigation bar with links: Home, Register, Edit, Evaluate, Report, and Help. Below this is a header section with the 'EDS' logo and the title 'Rio SC Status Report'. The main content area is divided into several sections:

- Evaluations to Do:** A sidebar on the left with a link to 'Solution Centre Evaluation'.
- Currents Evaluations Made:** A central section showing a date range from 02/10/05 to 08/10/05.
- Solution Centre Evaluation:** A section titled 'Evaluated by: Alexandre Thomaz' containing a table of criteria and status, and a 'Comments Overall' text area.
- DL Evaluations:** A sidebar on the right showing 'Roberto Gomes' with a checkmark, and links to 'Register', 'Edit', 'Project', and 'People'.
- Delivery Leaders Evaluations:** A section titled 'Roberto Gomes' containing a table of criteria and status, and a 'Comments Overall' text area.

Solution Centre Evaluation Data:

| Criteria | Status |
|-----------|--------|
| IT | R |
| Overall | G |
| Financial | R |
| Delivery | G |
| People | G |

Comments Overall: fdsa

Delivery Leaders Evaluations Data (Roberto Gomes):

| Criteria | Status |
|-----------|--------|
| IT | Y |
| Overall | Y |
| Financial | Y |
| Delivery | G |
| People | R |

Comments Overall: asdfdsa

Figura 30 : Gerente do centro de solução

3.1.2.3 Diagramas de Seqüência

Diagrama de Seqüência do Login gerente nível 1

Abaixo está ilustrado o diagrama de seqüência de Login, mas para o caso de um gerente de projeto, nível 1. Pode-se observar que a classe BusinessDelegate é quem vai fazer toda a lógica de negócio do sistema. Todo o acesso ao banco de dados é feito através desta classe. Quando o gerente nível 1 acessar o sistema, primeiramente os dados de login e password são validados. Feita a validação é feito uma consulta sobre as avaliações feitas e não feitas durante a semana que o usuário acessou o sistema. Essas informações serão disponibilizadas para a página **Home.jsp**.

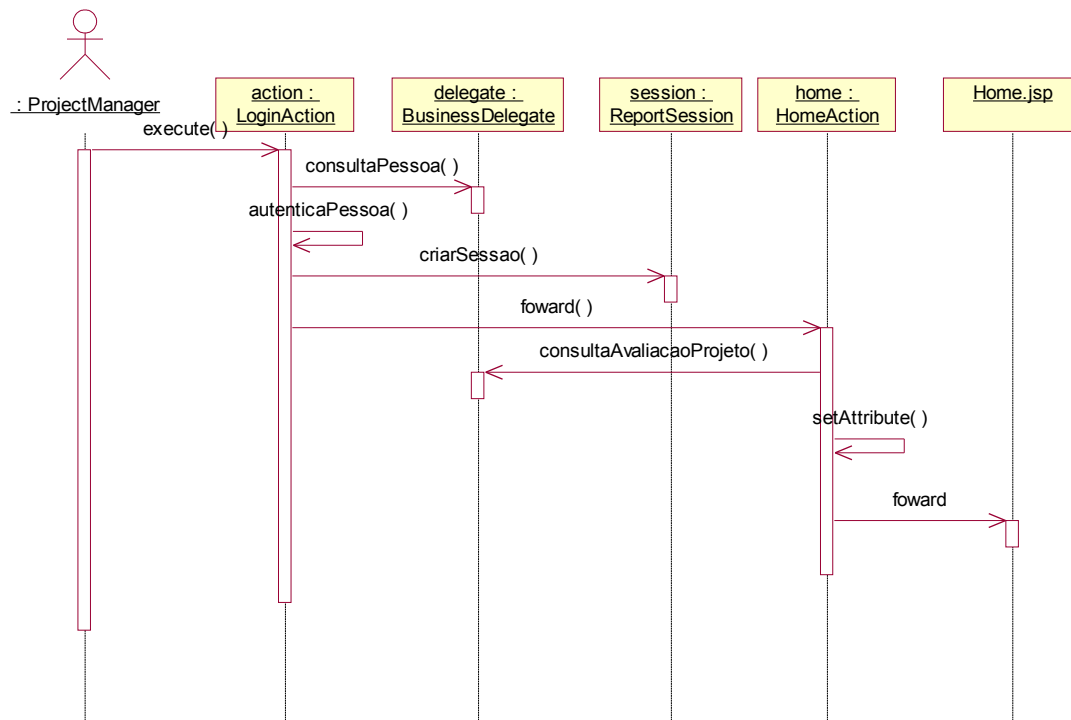


Figura 31 : Diagrama de Seqüência Lajem

Diagrama de Seqüência do Lajem gerente nível 2

A deferência quando um gerente de programa acessa o sistema é na busca pelas avaliações. Enquanto o gerente de projeto só pode avaliar projetos, o gerente de programa pode fazer avaliações de projeto e de programa. São carregadas as avaliações de todos os projetos associados ao programa. A diferença está na utilização da classe BusinessDelegate.

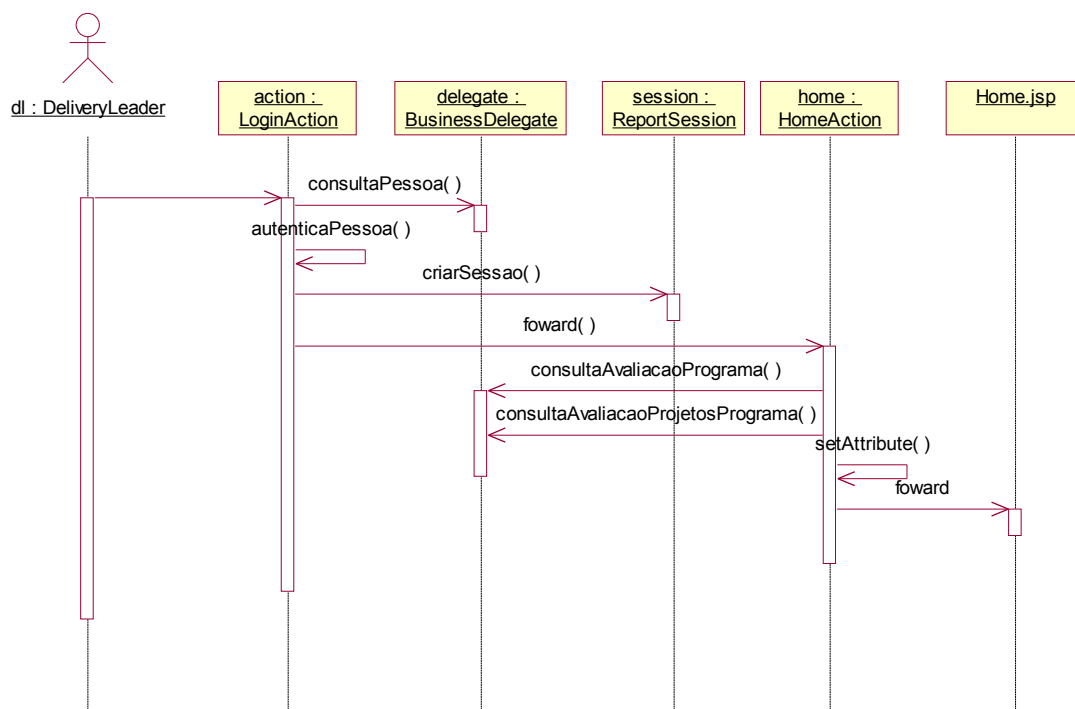


Figura 32 : Diagrama Seqüência Login nível 2

Diagrama de Seqüência do Login gerente nível 3

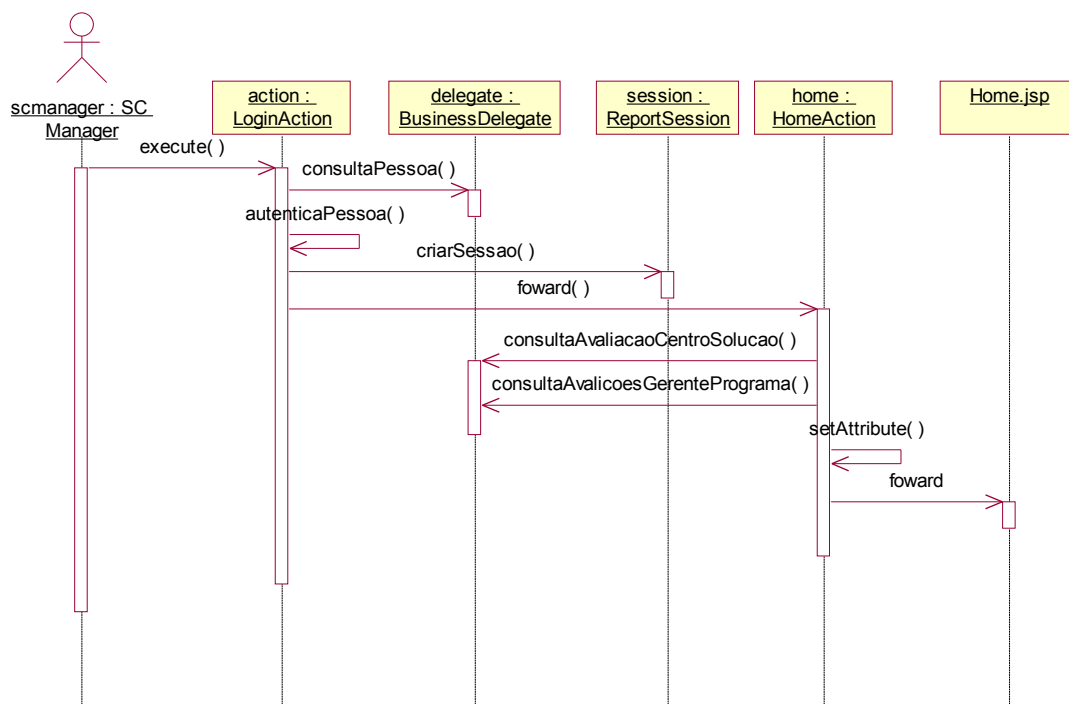


Figura 33 : Diagrama de Seqüência Login nível 3

3.1.3 Módulo Avaliações – casos de uso Avalia Projeto, AvaliaPrograma e Avalia Centro de Solução.

3.1.3.1 Descrição da Funcionalidade

Esta é a principal funcionalidade do sistema. Como já foi mostrado no documento ERS, existe um workflow que deve ser seguido. Começa com as avaliações de projeto, depois as avaliações de programa são feitas a partir de uma análise das avaliações dos projetos associados ao programa. Em seguida quando todas as avaliações de programa forem feitas é então feito à avaliação do centro de solução.

3.1.3.2 Fluxo das Avaliações

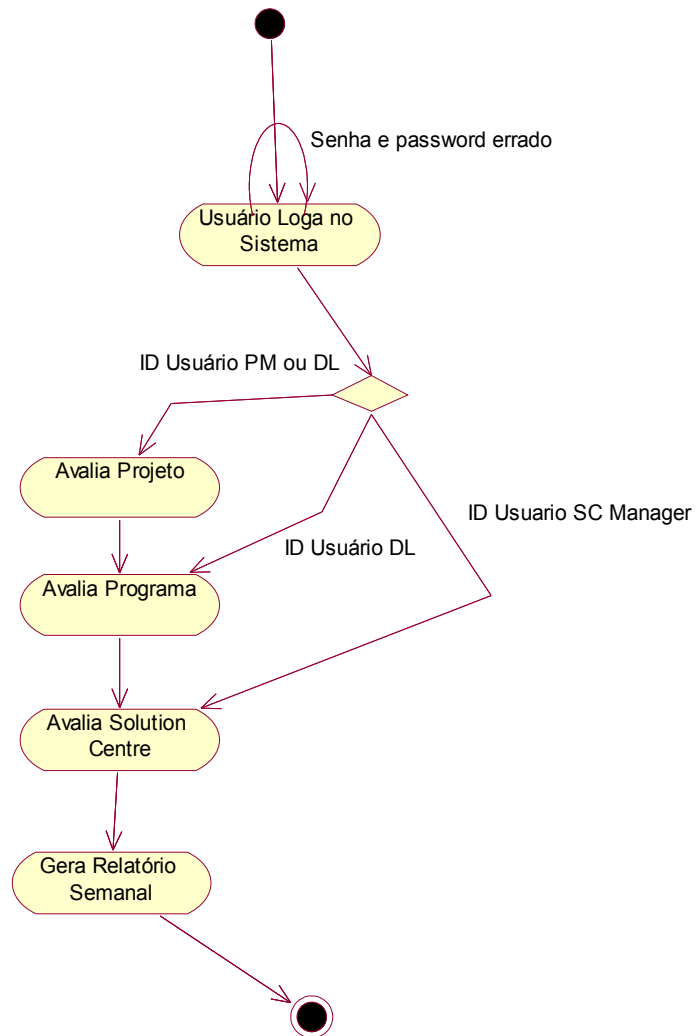


Figura 34 : Fluxo das Avaliações

Protótipo das Telas

Tela de Avaliação de Projeto

Para a melhor visualização, esta tela foi dividida em duas partes. A primeira é uma visão mais detalhada da tela, mas ficaram de fora os outros campos. A segunda é uma visão de toda a tela, mas com uma definição piorada. É uma tela simples que contém todas as informações que devem ser submetidas semanalmente para a avaliação do projeto. Todos os campos são obrigatórios. Caso exista uma avaliação feita na semana anterior ela é carregada como default. Neste caso abaixo é a primeira avaliação.

The screenshot shows the 'Rio SC Status Report' web application. At the top, there is a header with 'Rio Solution Centre', a user greeting 'Hello Joaquim Manuel, 14 de Setembro de 2005', and a 'to info Centre' link. Below the header is a navigation bar with links: Home, Register, Edit, Evaluate, Report, and Help. The main content area has a title 'EDS Rio SC Status Report' and a background image of a traffic light. On the left, there is a sidebar with 'Evaluation Criterias' including Client, Resources, Finance, IT, Solution, Schedule, and Governance. On the right, there is a section for 'General Motors' with 'Evaluations to Do' including General Motors. The main content area displays two evaluation sections: 'Client' and 'Finance'. Each section has three radio buttons labeled G (Green), Y (Yellow), and R (Red). The 'Client' section has a 'Comments' text area below the radio buttons. The 'Finance' section also has a 'Comments' text area below the radio buttons.

Figura 35 : Tela Avaliação de projeto perto

Rio Solutions Center Hello Joaquim Mansur, 14 de Setembro de 2005 [No InfoCenter](#)

Rio SC Status Report

General Motors

Client

Comments

Finance

Comments

Resources

Comments

IT

Comments

Solution

Comments

Schedule

Comments

Governance

Comments

submit

Copyright 2005 Electronic Data Systems Corp. All rights reserved.

Figura 36 : Tela Avaliação de Projeto

Definições dos Botões/Links da Tela

| Botão / Link | Ação desempenhada |
|--------------|-------------------------------------|
| Submit | A avaliação é submetida no sistema. |

| | |
|---|--|
| <div data-bbox="323 210 628 241">Evaluation Criteria</div> <ul style="list-style-type: none"> ➤ Client ➤ Resources ➤ Finance ➤ IT ➤ Solution ➤ Schedule ➤ Governance | <p>Estes são links que apontam para a mesma página, mas posicionam a tela para o campo desejado.</p> |
|---|--|

Tela de Avaliação de Programa

A tela de avaliação de programa possui algumas diferenças com a tela de avaliação de projeto. As avaliações de programa são baseadas nas avaliações dos projetos associados ao programa, portanto é necessário a visualização destas avaliações neste processo. Pode ser visto no lado esquerdo da tela os projetos associados ao programa, AirDNS e General Motors.

Os critérios de avaliação estão com o fundo pintado de acordo com seu status.

Rio Solution Centre Hello Roberto Gomes, 5 de Outubro de 2005 to infoCentre ⓘ

Home Register Edit Evaluate Report Help

EDS Rio SC Status Report

Evaluation Criteria

- Overall
- Delivery
- Financial
- IT
- People

Program Evaluation

Overall

☐ ☒ **G** ☒ **Y** ☐ **R**

Comments

asdfsda

Delivery

☒ ☐ **G** ☐ **Y** ☐ **R**

Comments

asdfd

Program Evaluations

- AirDNS
 - Client
 - Schedule
 - Resource
 - Governance
 - Finance
 - Solution
 - IT
- General Motors
 - Client
 - Schedule
 - Resource
 - Governance
 - Finance
 - Solution
 - IT

Figura 37 : Tela Avaliação de programa

Rio Solution Centre

Hello Roberto Gomes, 5 de Outubro de 2005

to infoCentre

Home

Register

Edit

Evaluate

Report

Help

EDS

Rio SC Status Report

Evaluation Criteria

Overall

Delivery

Financial

IT

People

Program Evaluation

Overall

G

Y

R

Comments

asfdsa

Delivery

G

Y

R

Comments

asdfd

Financial

G

Y

R

Comments

asdf

IT

G

Y

R

Comments

asdfd

People

G

Y

R

Comments

asdfd

submit

Program Evaluations

AirDNS

Client

Schedule

Resource

Governance

Finance

Solution

IT

General Motors

Client

Schedule

Resource

Governance

Finance



















Solution

IT

Copyright 2005 Electronic Data Systems Corp. All rights reserved

Figura 38 : Tela de Avaliação de programa inteira

Definições dos Botões/Links da Tela

| Botão / Link | Ação desempenhada |
|---|---|
| Submit | A avaliação é submetida no sistema. |
|  <ul style="list-style-type: none"> ❖ Overall ❖ Delivery ❖ Financial ❖ IT ❖ People | Estes são links que apontam para a mesma página, mas posicionam a tela para o campo desejado. |
|  <ul style="list-style-type: none">  AirDNS <ul style="list-style-type: none">  Client  Schedule  Resource  Governance  Finance  Solution  IT  General Motors <ul style="list-style-type: none">  Client  Schedule  Resource  Governance  Finance  Solution  IT | Estas são avaliações de projetos pertencentes ao programa. As cores de fundo dos critérios de avaliação representam o seu status. |

Tela de Avaliação de Centro de Solução

Esta tela é bem parecida com a de avaliação de programa, com a diferença de que sua avaliação vai ser baseada nas avaliações de programa feitas pelos Delivery Leaders – Gerente de programa, nível 2. Então ao invés de aparecer no canto esquerdo da tela as avaliações de projetos pertencentes ao programa vai aparecer as avaliações dos DLs de todo o centro de solução. Na tela abaixo aparece o nome do DL Roberto Gomes.

Rio Solution Centre
Hello Alexandre Thomaz, 5 de Outubro de 2005
to info Centre ⓘ

Home
Register
Edit
Evaluate
Report
Help

EDS
Rio SC Status Report

Evaluation Criterias

- Overall
- Delivery
- Financial
- IT
- People

Solution Centre Evaluation

Overall

☒ G
☐ Y
☐ R

Comments

fdsa

Delivery

☒ G
☐ Y
☐ R

Comments

asdf

Financial

☐ G
☐ Y
☒ R

Comments

fdsa

IT

☐ G
☐ Y
☒ R

Comments

asdf

People

☒ G
☐ Y
☐ R

Comments

bvcbn

submit

Program Evaluations

Roberto Gomes

- Overall
- Delivery
- Financial
- IT
- People

Copyright 2005 Electronic Data Systems Corp. All rights reserved

Figura 39 : Tela de Avaliação de Centro de Solução

Diagrama de Seqüência de Avaliação de Projeto

Neste diagrama de seqüência podemos perceber a grande vantagem de utilizar um framework como struts. O struts possui uma superclasse chamada ActionForm, cujo todos os form beans da aplicação estendem ela. Esta funcionalidade permite que todos os dados de avaliação de projeto submetidos pelo gerente de projeto estejam armazenado no formbean AvaliaProjetosForm. Isto permite um fácil acesso aos dados de avaliação do gerente de projeto. As classes que representam o modelo de dados utilizam todas as informações dos formbeans para então ser utilizado pela classe BusinessDelegate. Esta classe será a responsável pela inserção dos dados no banco de dados.

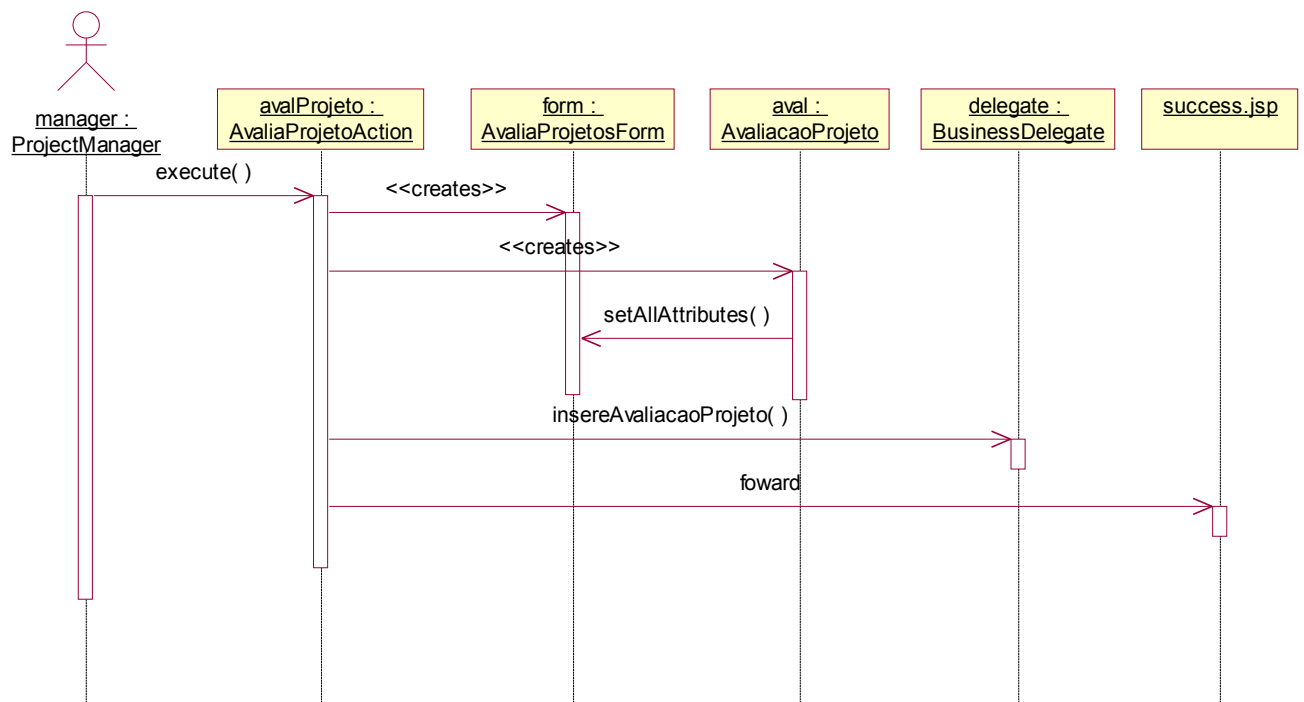


Figura 40 : Diagrama de Seqüência Avaliação de Projeto

Diagrama de Seqüência de Avaliação de Programa

As diferença deste diagrama de seqüência com o de avaliação de projeto estão nas classes Action, formbean e modelo. Mas a estrutura é a mesma. Percebe-se a grande reutilização de código.

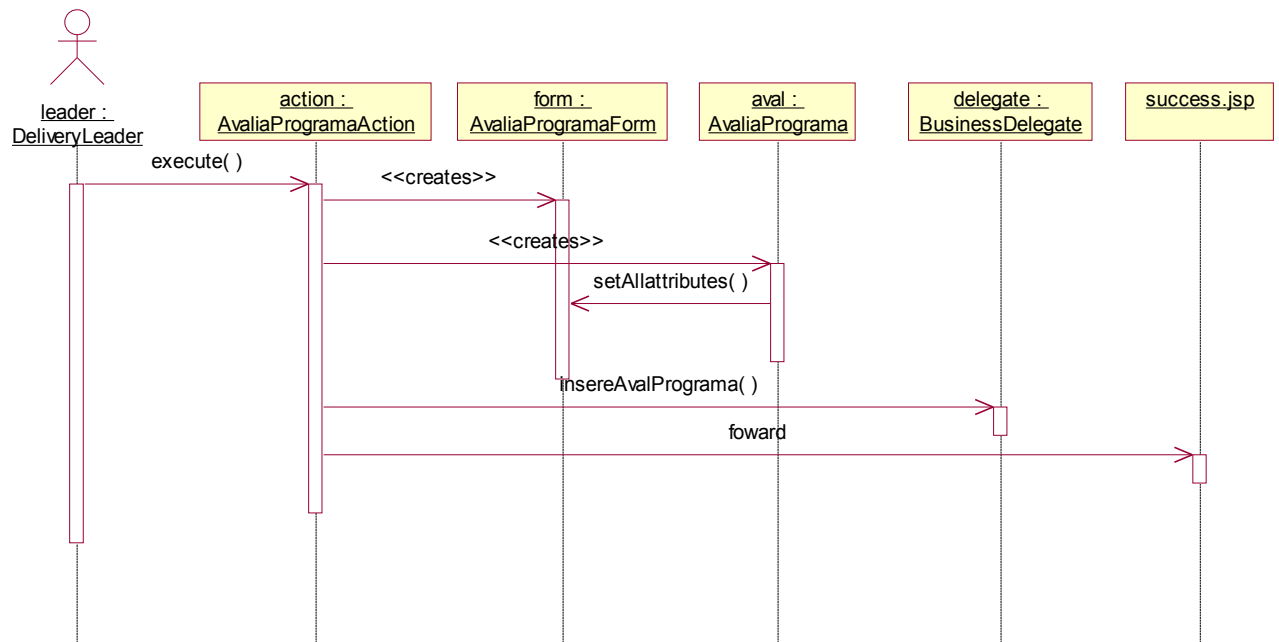


Figura 41 : Diagrama de Seqüência Avaliação de Programa

Diagrama de Seqüência de Avaliação de Centro de Solução

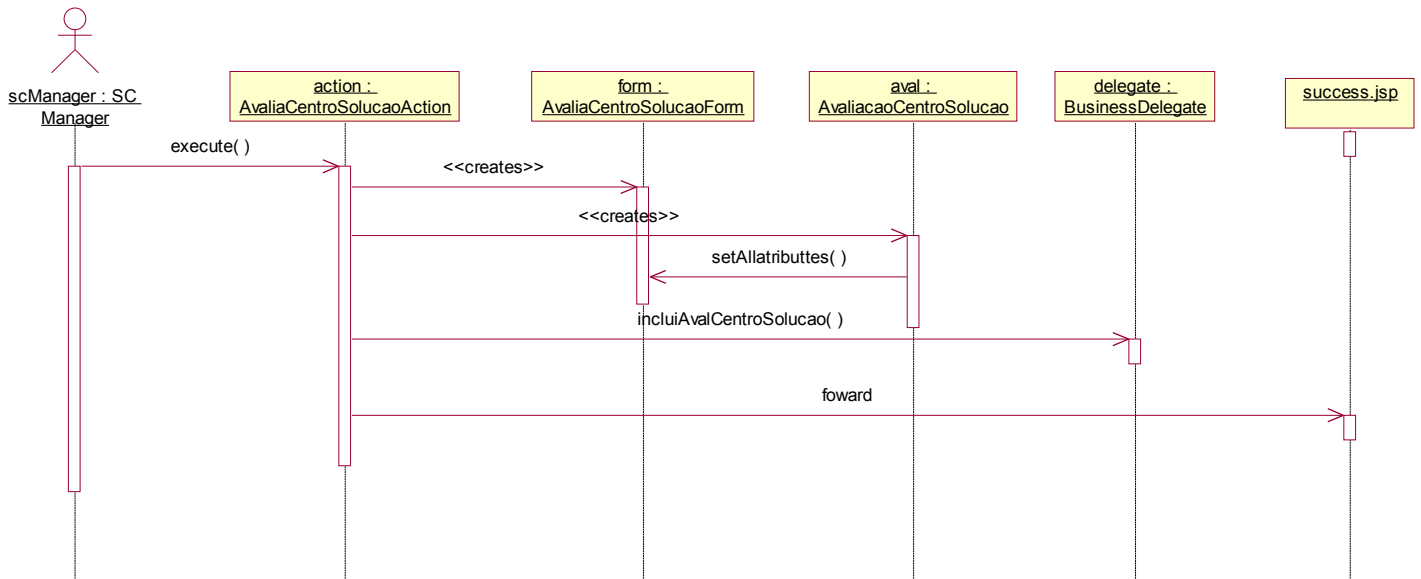


Figura 42 : Diagrama Seqüência Avaliação Centro Solução

3.1.4 Módulo Cadastro – caso de uso Cadastro de projeto, Cadastro de pessoas e Cadastro de Centro de Solução.

3.1.4.1 Descrição da Funcionalidade

Esta funcionalidade permite a cadastramento de pessoas, projetos e centros de soluções. Esta funcionalidade não está disponível para qualquer usuário. Os gerente de projetos não podem cadastrar nada, já os gerentes de programa – Delivery Leader – podem cadastrar pessoas. Só está disponível a funcionalidade para cadastramento de projetos ao administrador e ao gerente do centro de solução.

3.1.4.2 Protótipo das Telas

Tela de Cadastro de Pessoa

Rio Solution Centre
Hello, Roberto Gomes
to info Centre

Home
Register
Edit
Evaluate
Report
Help


Rio SC Status Report


Edit
Project
AirDNS
General Motors
People

Register
People

Full Name:
NetID: Email:
Password: Level:

submit

Copyright 2005 Electronic Data Systems Corp. All rights reserved

Figura 43 : Cadastro Pessoa

Definições dos Campos da Tela

| Nome do Campo | Descrição do Campo | Regra de Exibição |
|---------------|--|-------------------|
| Full Name | Nome todo do usuário. | Sempre |
| NetID | Código Alfanumérico de 6 dígitos que identifica o usuário em toda a EDS. | Sempre |
| Email | Email da EDS do usuário. | Sempre |
| Password | Senha para o usuário entrar no sistema. | Sempre |
| Level | Nível do usuário: 1 – Gerente do projeto 2 – Gerente de programa 3 – Gerente de Centro de Solução 4 – Administrador do sistema | Sempre |

Definições dos Botões/Links da Tela

| Botão / Link | Ação desempenhada |
|--------------|---|
| Submit | Submete as informações para o sistema cadastrando com sucesso ou não o usuário. |

Primeira Tela de Cadastramento de projeto

Register Project

Project Name:

Managed/Non Managed: ▼

Management Type according to BPA: ▼

Client: Industry:

BPA: Project type: ▼

Solution Center FTEs: Contractor FTEs:

Solution Centre: ▼ Phase: ▼

[Add New?](#)

Figura 44 : Primeira Tela de Cadastramento de Projeto

Definições dos Campos da Tela

| Nome do Campo | Descrição do Campo | Regra de Exibição |
|----------------------------------|---|-------------------|
| Project Name | Nome todo do projeto. | Sempre |
| Managed/Non Managed | Se o projeto é gerenciado aqui no Rio de Janeiro ou pelo cliente. | Sempre |
| Management Type according to BPA | Muitos projetos são gerenciados por pessoas aqui do Rio de Janeiro, mesmo no contrato dizendo o contrário. Este campo define se no contrato o projeto é gerenciado ou não por pessoas aqui do Rio de Janeiro. | Sempre |
| Client | Nome do Cliente | Sempre |
| Industry | Tipo de indústria do projeto | Sempre |
| BPA | Número do contrato assinado com o cliente. | Sempre |
| Project Type | Se o projeto é de suporte a produção ou desenvolvimento. | Sempre |
| Solution Centers FTE | Quantidade de recursos alocados no projeto contrados por CLT. | Sempre |
| Contractor FTE | Quantidade de recursos contratados como pessoa jurídica. | Sempre |

| | | |
|-----------------|---|--------|
| Solution Centre | Centro de solução associado ao projeto. | Sempre |
| Phase | Fase atual do projeto | Sempre |

| Botão / Link | Ação desempenhada |
|--------------|---|
| Next | Avança para a segunda tela de exibição. |
| reset | Reseta todas as informações. |

Segunda Tela de Cadastramento de projeto

Rio Solution Centre Hello, admin to infoCentre

Home Register Edit Evaluate Report Help

EDS Rio SC Status Report CMMi 5

Edit Register

Project

AirDNS

General Motors

People

Solution Center

Rio Solution Centre

Project

People

Project

Solution Center

Names Found:

admin

Joaquim Manuel

Alexandre Thomaz

Fabio Puentes

select

Names Selected:

Roberto Gomes

List of DLs:

Roberto Gomes

submit

Copyright 2005 Electronic Data Systems Corp. All rights reserved

Figura 45 : Segunda Tela de Cadastramento de Projeto

Definições dos Campos da Tela

| Nome do Campo | Descrição do Campo | Regra de Exibição |
|----------------|--|-------------------|
| Names Found | São todos os nomes resultantes da busca. | Sempre |
| Names Selected | São os nomes selecionados pelo usuário para a gerência do projeto. | Sempre |
| List of DLs | Seleção do gerente de programa, o DL (Delivery Leader) | Sempre |

| Botão / Link | Ação desempenhada |
|--------------|---|
| buscar | Busca todos os nomes que contenham as palavra digitada no campo. |
| select | Ao ser pressionado seleciona a pessoa para a gerência do projeto. Movendo o nome para lista da caixa representada por Names Selected. |
| submit | Submete todos os dados concluindo o cadastramento de projeto. |

Tela de Cadastramento de Centro de Solução

Rio Solution Centre Hello, admin to infoCentre

Home Register Edit Evaluate Report Help

EDS Rio SC Status Report

Evaluations to Do
Evaluations Made

Register Solution Center

Solution Center Name:

Address:

City: State:

Country: Phone:

Fax: Assistant Phone:

Administrative Assistant:

Solution Center Manager:

Register
✦ People
✦ Project
✦ Solution Center

Edit
✦ Project
✦ People
✦ Solution Center

Copyright 2005 Electronic Data Systems Corp. All rights reserved

Figura 46 : Cadastramento de Centro de Solução

Definições dos Campos da Tela

| Nome do Campo | Descrição do Campo | Regra de Exibição |
|----------------------|--------------------------------------|-------------------|
| Solution Center Name | Nome todo do centro de solução. | Sempre |
| Address | Endereço do centro de solução. | Sempre |
| City | Cidade do centro de solução. | Sempre |
| State | Estado do centro de solução | Sempre |
| Country | País do centro de solução | Sempre |
| Phone | Telefone de contato para o centro de | Sempre |

| | | |
|--------------------------|---|--------|
| | solução. | |
| Assistant Phone | Telefone da Assistente do gerente do centro de solução. | Sempre |
| Administrative Assistant | Nome da assistente do gerente do centro de solução. | Sempre |
| Solution Center Manager | Gerente de todo o centro de solução. | Sempre |

| Botão / Link | Ação desempenhada |
|--------------|---|
| submit | Submete todos os dados concluindo o cadastramento de centro de solução. |

3.1.4.3 Diagramas de Seqüência

Diagrama de Cadastro de Pessoa

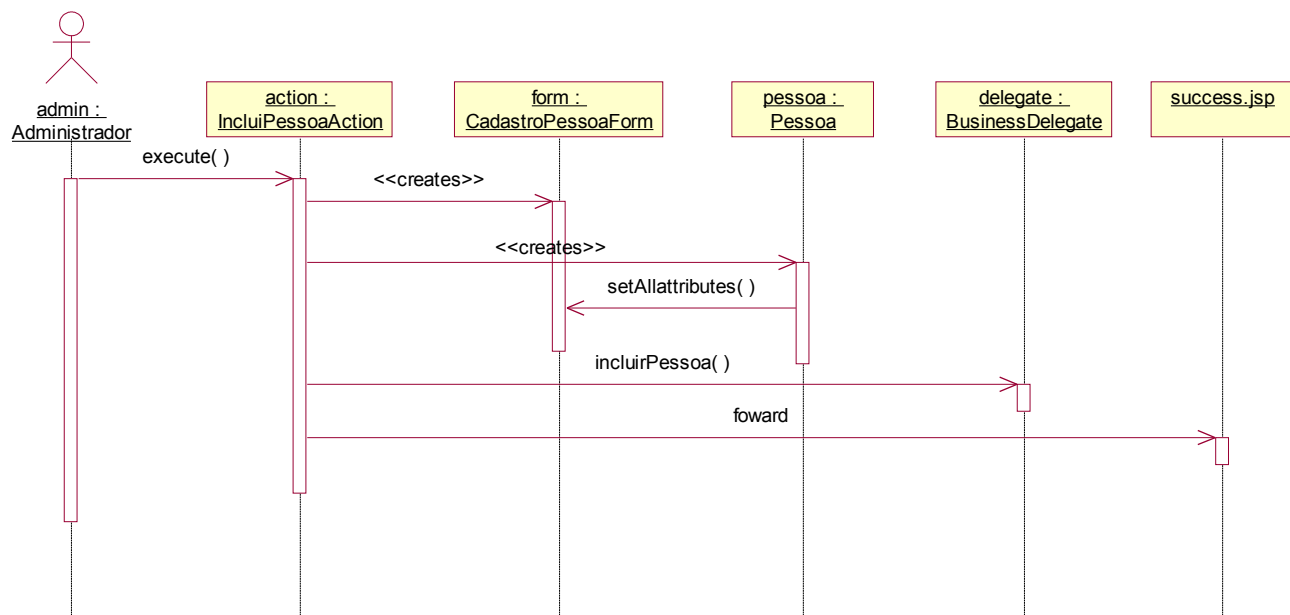


Figura 47 : Diagrama de Seqüência de Cadastro de Pessoa

Diagrama de Cadastro de Projeto

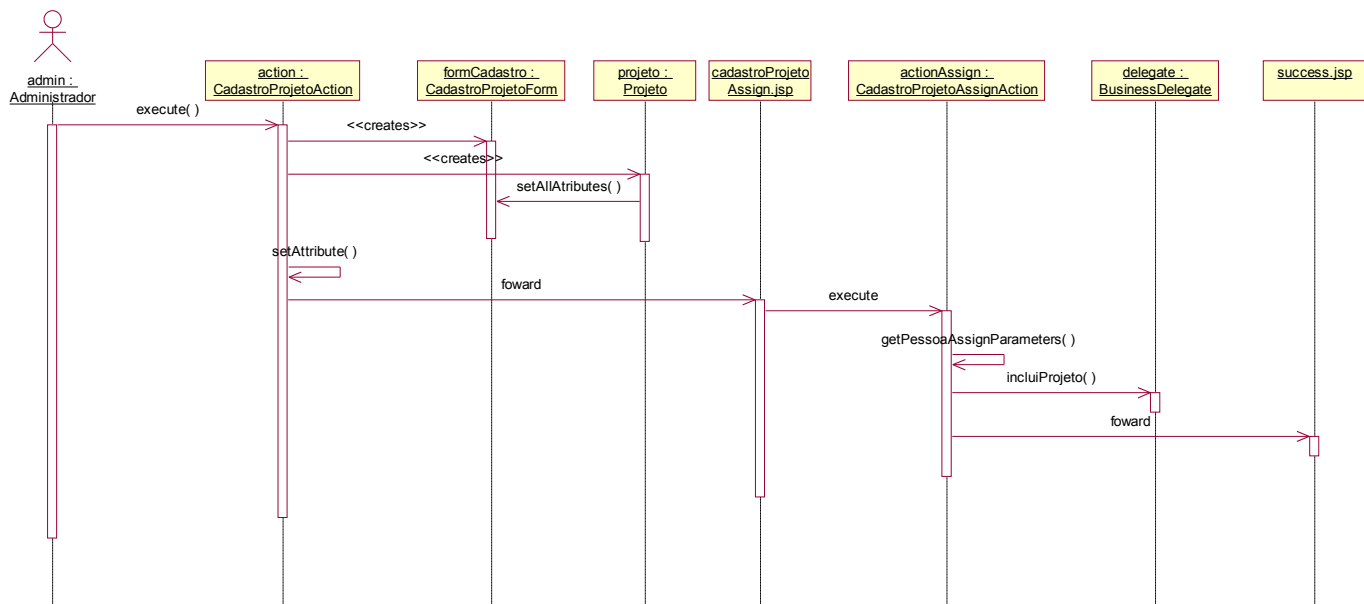


Figura 48 : Diagrama Seqüência Cadastro Projeto

Diagrama de Cadastro de Centro de Solução

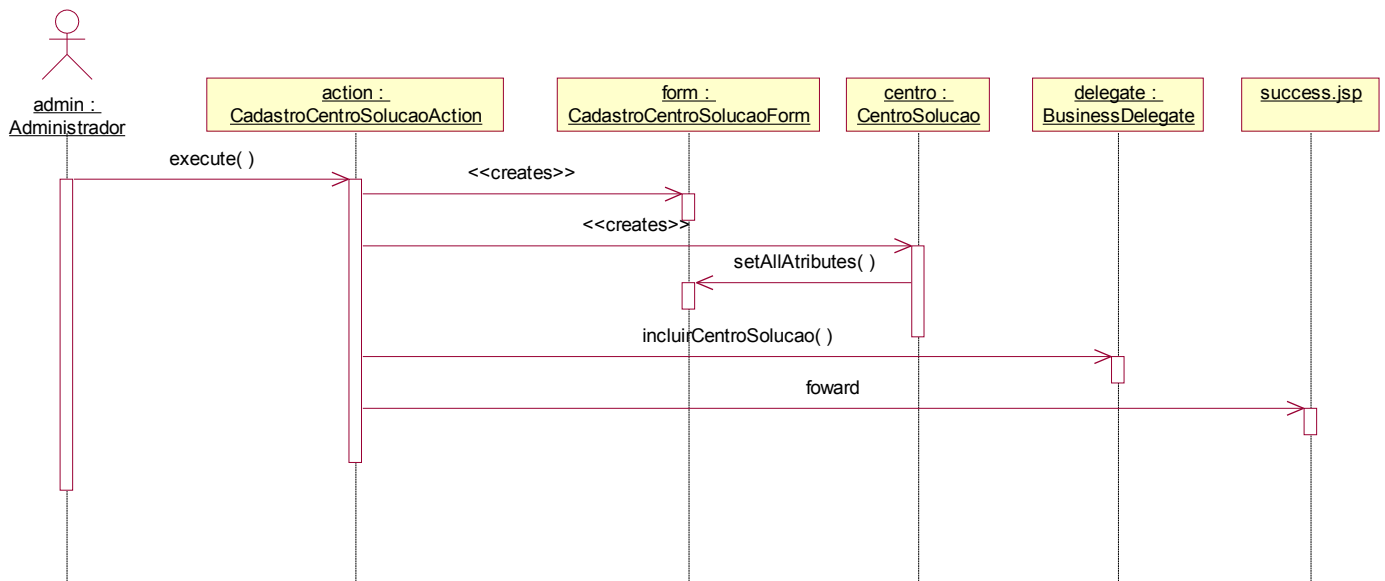


Figura 49 : Diagrama Seqüência Cadastro Centro de Solução

3.1.5 Módulo Edição

Neste módulo vai ser apresentado apenas os diagramas de seqüência devido às telas serem as mesmas. A única diferença é que os campos vêm preenchidos com as informações a serem editadas. Em relação ao diagrama de seqüência a diferença está no método usado na classe BusinessDelegate.

Diagrama Edita Cadastro Centro Solução

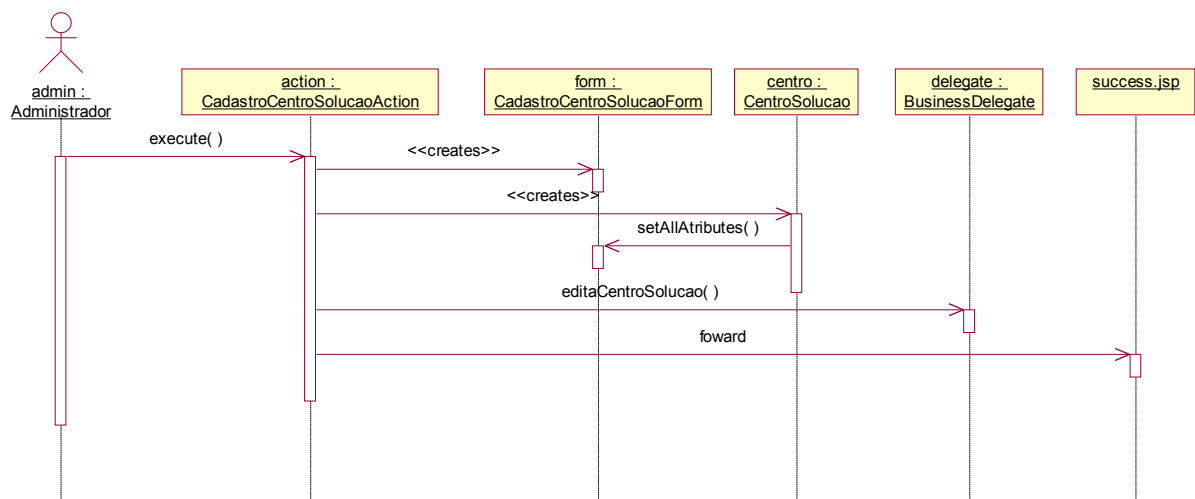


Diagrama Edita Cadastro Projeto

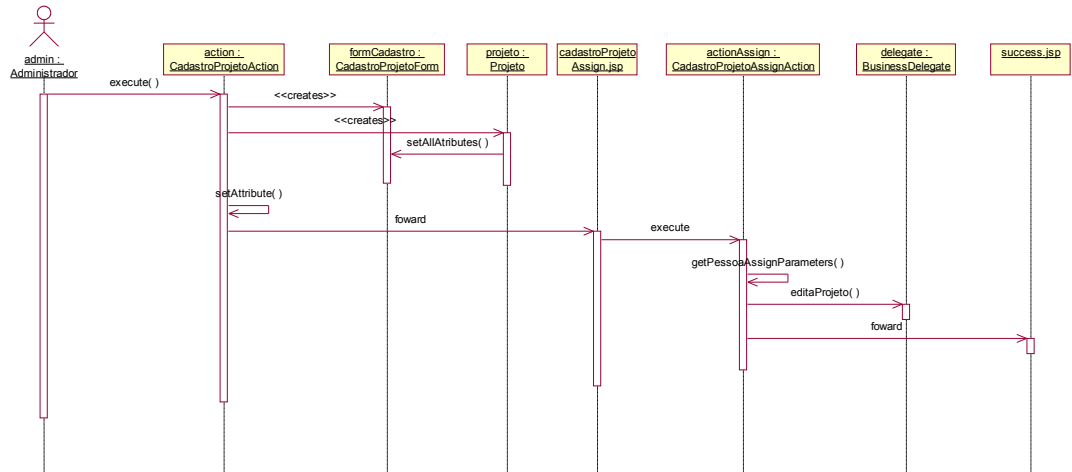


Diagrama Edita Cadastro Pessoa

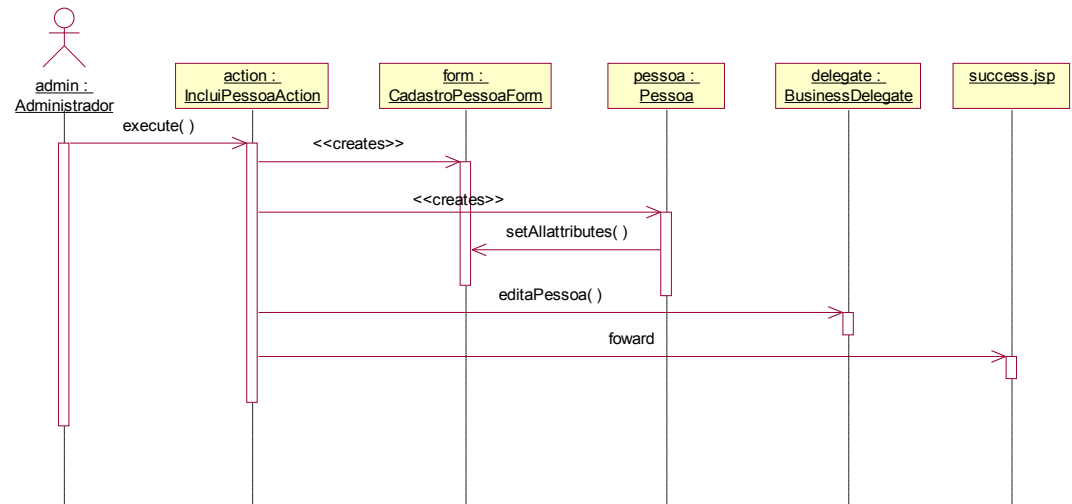


Diagrama Edita Avaliação de Projeto

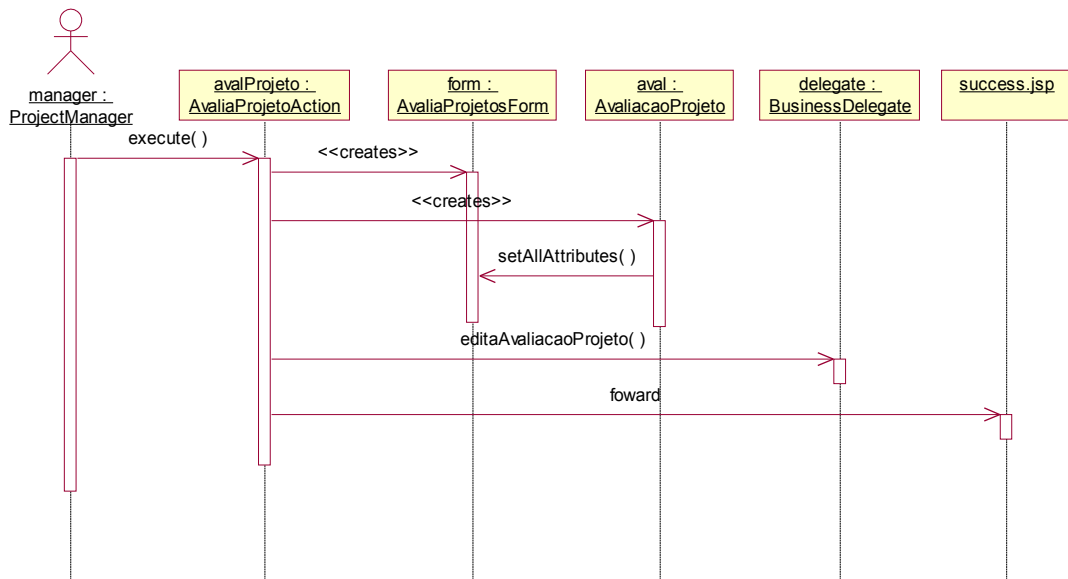
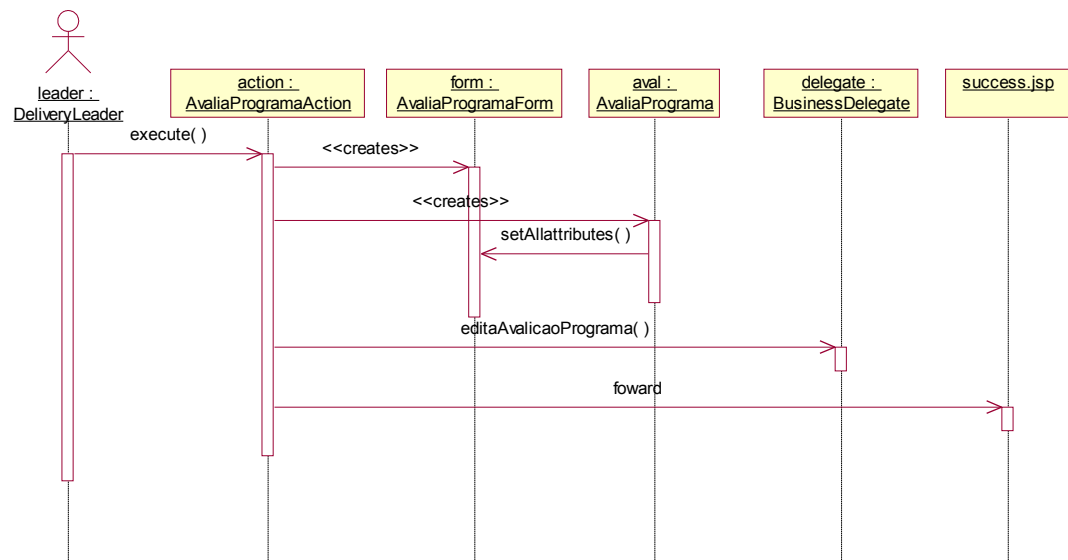


Diagrama Edita Avaliação de Programa



Edita Avaliação de Centro de Solução

