

UNIVERSIDADE DE SÃO PAULO  
ESCOLA POLITÉCNICA

FELIPE MARTINS DA SILVA  
FERNANDO RAFFANI  
JOÃO PAULO SEABRA SANTOS  
MATIAS EDUARDO RECCIUS PUGA

**Desenvolvimento de software ERP, usando SOA e  
BPM, aplicado ao Programa de Educação Continuada da  
Escola Politécnica**

v. 1

São Paulo  
2006

FELIPE MARTINS DA SILVA  
FERNANDO RAFFANI  
JOÃO PAULO SEABRA SANTOS  
MATIAS EDUARDO RECCIUS PUGA

**Desenvolvimento de software ERP, usando SOA e  
BPM, aplicado ao Programa de Educação Continuada da  
Escola Politécnica**

Monografia de conclusão do curso de graduação  
em Engenharia Elétrica ênfase Computação

Área de concentração: Departamento de  
Engenharia de Computação e Sistemas Digitais  
(PCS)

Orientadora: Selma Shin Shimizu Melnikoff

Co-orientador: Leonardo Dominguez Dias

v. 1

São Paulo  
2006

## FOLHA DE APROVAÇÃO

Felipe Martins da Silva  
Fernando Raffani  
João Paulo Seabra Santos  
Matias Eduardo Reccius Puga

Monografia de conclusão do curso apresentada à  
Escola Politécnica da Universidade de São Paulo  
para a obtenção de graduação em Engenharia  
Elétrica ênfase Computação.

Área de concentração: Departamento de  
Engenharia de Computação e Sistemas Digitais  
(PCS)

Banca Examinadora

Prof. Dr. \_\_\_\_\_

Instituição \_\_\_\_\_ Assinatura \_\_\_\_\_

Prof. Dr. \_\_\_\_\_

Instituição \_\_\_\_\_ Assinatura \_\_\_\_\_

Prof. Dr. \_\_\_\_\_

Instituição \_\_\_\_\_ Assinatura \_\_\_\_\_

Prof. Dr. \_\_\_\_\_

Instituição \_\_\_\_\_ Assinatura \_\_\_\_\_

Prof. Dr. \_\_\_\_\_

Instituição \_\_\_\_\_ Assinatura \_\_\_\_\_

## DEDICATÓRIA

Ao ilustre Prof. Dr. Luiz de Queiroz Orsini, pela inspiração e respeito despertados em todos os alunos de engenharia elétrica da escola.



## AGRADECIMENTOS

Os autores agradecem suas famílias e amigos pelo companheirismo em todos os momentos da graduação.

À Prof. Dr. Selma Shin Shimizu Melnikoff, pela atenção e apoio durante o processo de definição e orientação.

Ao engenheiro Leonardo Dominguez Dias, pela disponibilidade em nos orientar durante o dia-a-dia do projeto.

À Escola Politécnica da Universidade de São Paulo, pela oportunidade de realização da graduação em engenharia. ERP (*Enterprise Resource Planning*)

Ao Programa de Educação Continuada da Escola Politécnica da Universidade de São Paulo, por fornecer o caso e dispor os recursos necessários para o desenvolvimento do projeto.

"Somos o que repetidamente fazemos. A excelência, portanto,  
não é um feito, mas um hábito."

Aristóteles

## RESUMO

Silva, F. M.; Raffani, F.; Santos, J. P. S.; Puga, M. E. R. **Desenvolvimento de software ERP, usando BPM e SOA, aplicado ao Programa de Educação Continuada da Escola Politécnica.** 2006. Monografia de conclusão de curso (Graduação) – Escola Politécnica, Universidade de São Paulo, São Paulo, 2006.

Foram estudadas de técnicas para desenvolvimento de *software* de gerenciamento de recursos empresariais (*Enterprise Resource Planning*, ERP). Focou-se nas mais novas tecnologias usadas no desenvolvimento deste tipo de sistema, como a modelagem de processos de negócio (*Business Process Modeling*, BPM), utilizada para formalizar os processos de uma empresa; e a arquitetura orientada a serviços (*Service-Oriented Architecture*, SOA) na organização do *software* visando maior compatibilidade e expansibilidade. Estas técnicas foram aplicadas no desenvolvimento de um sistema a ser usado no Programa de Educação Continuada (PECE) da Escola Politécnica da Universidade de São Paulo (EPUSP). Avaliaram-se as metodologias disponíveis e a eficácia das técnicas no projeto, além das considerações sobre a aplicação em si.

Palavras chave: ERP, SOA, BPM, engenharia de software

## ABSTRACT

Silva, F. M.; Raffani, F.; Santos, J. P. S.; Puga, M. E. R. **Development of ERP software, using BPM and SOA, applied to Programa de Educação Continuada da Escola Politécnica.** 2006. Monograph of course conclusion (Electric Engineering degree) – Escola Politécnica, Universidade de São Paulo, São Paulo, 2006.

Studies were made on ERP (Enterprise Resource Planning) software development techniques. Focused on the newest technologies used in the development of such systems, like the business process modeling (BPM), used to formalize a company's processes; and the service-oriented architecture (SOA) on organizing the software for improved compatibility and expansibility. These techniques were applied on the development of a system to be used in the Programa de Educação Continuada (Continued Education Program, PECE) of the Escola Politécnica da Universidade de São Paulo (Escola Politécnica of the University of São Paulo, EPUSP). The methodologies available and the effectiveness of those techniques on the project were evaluated, as well as considerations on the application itself.

Keywords: ERP, SOA, BPM, software engineering.

## LISTA DE FIGURAS

Figura 1 –	Notação de evento .....	21
Figura 2 –	Notação de atividade .....	22
Figura 3 –	Notação para passagem .....	23
Figura 4 –	Notação para fluxo de seqüência .....	24
Figura 5 –	Notação do fluxo de mensagem.....	24
Figura 6 –	Notação de associação .....	25
Figura 7 –	Notação da área de contenção .....	25
Figura 8 –	Notação para linha divisória .....	26
Figura 9 –	Notação do objeto de dados.....	26
Figura 10 –	Notação para grupo.....	27
Figura 11 –	Notação de anotação.....	27
Figura 12 –	Notação em T de contas contábeis .....	29
Figura 13 –	Método das partidas dobradas de Pacioli .....	30
Figura 14 –	Divisão do valor de uma mensalidade entre contas contábeis usando o método das partidas dobradas.....	33
Figura 15 –	Divisão de uma mensalidade para diferentes recebedores, referenciada pelo seu número do boleto bancário .....	33
Figura 16 –	Dois cursos pagando o rateio de um lanche, usando contas contábeis e o método das partidas dobradas.....	34
Figura 17 –	Dois cursos pagando o rateio de um lanche, referenciando a mesma nota fiscal de saída	34
Figura 18 –	Ciclo de vida de um curso no PECE modelado no nível mais alto .....	41

Figura 19 –	Detalhamento da prospecção de clientes .....	41
Figura 20 –	Detalhamento da solicitação de matrícula, envolvendo diferentes entidades ..	42
Figura 21 –	Diagrama de casos de uso com ator único, o funcionário. ....	44
Figura 22 –	Diagrama de arquitetura .....	45
Figura 23 –	Curso Y empresta para Curso X, Curso X devolve como parte de um pagamento do Curso Y .....	61
Figura 24 –	Pôster do projeto de formatura. ....	72
Figura 25 –	Diagrama de Entidade-Relacionamento do módulo financeiro.....	1
Figura 26 –	Diagrama de classes.....	2
Figura 27 –	Produção de curso.....	1
Figura 28 –	Prospecção de clientes .....	1
Figura 29 –	Cadastro para propaganda .....	2
Figura 30 –	Abertura de cursos abertos .....	3
Figura 31 –	Abertura de cursos corporativos .....	4
Figura 32 –	Execução do curso .....	5
Figura 33 –	Solicitação de matrícula no curso.....	6
Figura 34 –	Oferecimento da disciplina.....	7
Figura 35 –	Troca de disciplina.....	8
Figura 36 –	Conclusão do oferecimento .....	9
Figura 37 –	Contrato e pagamento de professores .....	10
Figura 38 –	Financeiro .....	11
Figura 39 –	Recebimento de receitas .....	12
Figura 40 –	Recebimento de mesalidades.....	13
Figura 41 –	Pagamento de professores .....	14
Figura 42 –	Compra de materiais.....	15

Figura 43 –	Recebimento de cursos corporativos .....	16
Figura 44 –	Balanço final.....	17
Figura 45 –	Cadastrar despesa de curso.....	18

## LISTA DE SIGLAS

BPDs	<i>Business Process Diagrams</i> (diagramas de processo de negócio)
BPM	<i>Business Process Modeling</i> (modelagem de processos de negócio)
BPMI	<i>Business Process Management Initiative</i> (iniciativa de gerenciamento de processos de negócio)
BPMN	<i>Business Process Modeling Notation</i> (notação da modelagem de processos de negócio)
CRM	<i>Customer Relationship Management</i> (gerência de relacionamento com o cliente)
ERP	<i>Enterprise Resource Planning</i> (planejamento de recursos empresariais)
EPUSP	Escola Politécnica da Universidade de São Paulo
FUSP	Fundação de Apoio à Universidade de São Paulo
HTML	<i>Hyper-Text Markup Language</i> (linguagem de marcação de hipertexto)
IIS	<i>Internet Information Services</i> (serviços de informação de Internet)
MBA	<i>Master in Business Administration</i> (mestrado em administração de empresas)
PECE	Programa de Educação Continuada da Escola Politécnica da Universidade de São Paulo
RUSP	Reitoria da Universidade de São Paulo
SOA	<i>Service-Oriented Architecture</i> (arquitetura orientada a serviços)
SQL	<i>Structured Query Language</i> (Linguagem de Consulta Estruturada)
TI	Tecnologia da Informação (em inglês: <i>Information Technology</i> , IT)
UML	<i>Unified Modeling Language</i> (linguagem unificada de modelagem)
XML	<i>Extensible Markup Language</i> (linguagem de marcação extensível)



## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>1</b>
1.1.	OBJETIVO	1
1.2.	DESCRIÇÃO DO PROJETO	1
1.3.	MOTIVAÇÃO	2
1.4.	JUSTIFICATIVA	3
1.5.	ESTRUTURA DO TRABALHO	4
<b>2</b>	<b>PECE</b>	<b>6</b>
2.1.	DESCRIÇÃO	6
2.2.	ESTRUTURA	7
2.2.1.	A UNIDADE DE NEGÓCIO ACADÊMICA	7
2.2.2.	A UNIDADE DE NEGÓCIO FINANCEIRA	8
2.2.3.	INFRA-ESTRUTURA, CENTRAL DE ATENDIMENTO E OUTRAS UNIDADES DE NEGÓCIO	9
<b>3</b>	<b>CONCEITOS UTILIZADOS</b>	<b>10</b>
3.1.	ERP ( <i>ENTERPRISE RESOURCE PLANNING</i> )	10
3.1.1.	DEFINIÇÃO	10
3.1.2.	APLICAÇÃO	11
3.1.3.	EXEMPLOS	12
3.2.	SOA ( <i>SERVICE-ORIENTED ARCHITECTURE</i> )	12
3.2.1.	DEFINIÇÃO	13
3.2.2.	COMPONENTES DA ARQUITETURA ORIENTADA A SERVIÇOS	16
3.2.3.	CONCEITOS DE SERVIÇOS SOB A ÓTICA DA SOA	16

3.2.4.	DESENVOLVIMENTO ORIENTADO A SERVIÇOS	18
3.3.	BPM ( <i>BUSINESS PROCESS MODELING</i> )	20
3.3.1.	DEFINIÇÃO E EXEMPLOS	20
3.3.2.	APLICAÇÃO NO TRABALHO	27
3.4.	CONTABILIDADE	28
3.4.1.	CONCEITOS BÁSICOS	28
3.4.2.	HISTÓRIA	30
3.4.3.	UMA PROPOSTA INOVADORA	31
3.4.4.	VANTAGENS E DESVANTAGENS	34
3.4.5.	CENTRO DE CUSTO	35
<b>4</b>	<b>METODOLOGIA DO PROJETO</b>	<b>37</b>
4.1.	LEVANTAMENTO DE REQUISITOS	38
4.1.1.	MÓDULO ACADÊMICO	39
4.1.2.	MÓDULO FINANCEIRO	39
4.2.	DEFINIÇÃO DOS PROCESSOS	40
4.3.	CASOS DE USO	43
4.4.	ARQUITETURA DO SISTEMA E DESCRIÇÃO DOS SERVIÇOS	44
4.5.	INFRA-ESTRUTURA	47
<b>5</b>	<b>IMPLEMENTAÇÃO E VALIDAÇÃO</b>	<b>48</b>
5.1.	CODIFICAÇÃO	48
5.2.	TESTES	48
5.3.	VALIDAÇÃO	49
5.4.	IMPLANTAÇÃO	50
5.5.	DOCUMENTAÇÃO DO PROJETO	50
<b>6</b>	<b>ANÁLISE DO RESULTADO</b>	<b>50</b>

<b>7</b>	<b>CONSIDERAÇÕES FINAIS</b>	<b>53</b>
7.1.	CONTRIBUIÇÕES DO TRABALHO	53
7.2.	TRABALHOS FUTUROS	55
7.3.	CONCLUSÃO DO GRUPO	56
<b>8</b>	<b>REFERÊNCIAS</b>	<b>59</b>
	MODELAGEM DE EMPRÉSTIMO ENTRE CONTRAS CONTÁBEIS	61
	CASOS DE USO DE CADASTROS	62
	CASOS DE USO DE RELATÓRIOS	70
	<b>APÊNDICE</b>	<b>72</b>
	ACADÊMICO	1
	FINANCEIRO	10

# 1 INTRODUÇÃO

## 1.1. OBJETIVO

Quatro anos haviam se passado desde que os integrantes do grupo ingressaram na Escola Politécnica e, desde então, haviam estudado conceitos de programação básica, estruturada e orientada a objetos, gerência de projetos e desenvolvimento de *softwares* nos laboratórios da Escola. Já se vinha tendo contato com tecnologias modernas em ambientes de desenvolvimento e arquiteturas de *software*, mas os integrantes ainda sentiam falta de um maior aprofundamento em *software* ERP (*Enterprise Resource Planning*). A coesão do grupo veio do intuito comum entre os integrantes, de se aprofundarem nos mesmos conhecimentos e do desejo de se estudar as últimas tecnologias que o mercado utiliza em *software* deste tipo. Desta maneira, o resultado serviria então para concluir sobre sua viabilidade, além de relatar a experiência obtida.

## 1.2. DESCRIÇÃO DO PROJETO

O projeto final escolhido veio a atender todos os requisitos que os integrantes desejavam ter nas duas disciplinas de Projeto de Formatura. O grupo observou que o mercado

estava adotando cada vez mais arquiteturas de sistema orientadas a serviços (*Service-Oriented Architecture*, SOA) e resolveu estudar seu conceito e aplicação. Aliado ao SOA, também foi estudada outra tecnologia nova: a modelagem de processos de negócio (*Business Process Modeling*, BPM). Aproveitou-se que no PECE necessitava de um *software* deste tipo e seria uma oportunidade para o grupo utilizar BPM e SOA em um sistema, mais interessante e desafiador para um dos programas de educação da escola.

Usou-se o BPM e SOA no sistema a ser implantado no Programa de Educação Continuada (PECE), um setor da Escola Politécnica da Universidade de São Paulo (EPUSP) que ministra cursos de extensão universitária em engenharia, descrito em mais detalhes no capítulo 2. Foram escolhidos e modelados pelo grupo duas unidades de negócio do PECE para integrar o novo sistema:

- A unidade de negócio acadêmica onde as principais atividades são relativas aos cadastros de alunos e professores nas disciplinas de cada um dos cursos; e
- A unidade de negócio financeira e o registro de todas as transações da instituição, como os pagamentos dos alunos de cada curso, o pagamento de professores, funcionários, fornecedores e repasses.

O grupo estudou o funcionamento do PECE como um todo e especificou tanto a acadêmica quanto a financeira para compor o ERP. Na codificação final do sistema, no entanto, o grupo focou apenas na parte financeira, sendo a acadêmica desenvolvida internamente pelo PECE em paralelo.

### 1.3. MOTIVAÇÃO

A escolha de se estudar, projetar e desenvolver *software* ERP foi feita face à sua ampla adoção no mercado brasileiro e mundial. Sua capacidade de automação de processos de negócio, sejam eles operacionais ou gerenciais, agregam muito valor às empresas atualmente. O correto planejamento e desenvolvimento de um ERP possibilitam que ele seja expansível e futuramente sejam criadas interfaces com outros sistemas, aumentando o valor agregado à empresa de maneira indefinida. Seu correto planejamento e desenvolvimento são detalhados em 3.2

O conhecimento obtido em um projeto deste tipo é muito útil na formação dos engenheiros, pois a aplicação prática do aprendizado no mercado é imediata. Inclusive alguns integrantes do grupo se ocupavam em empresas de desenvolvimento de *softwares* deste tipo. No entanto, o desafio de concluir o projeto e sua aplicação no PECE por si só já auxiliaram na formação acadêmica e profissional dos integrantes.

Espera-se também que o uso de inovações deste tipo possa eventualmente agregar à academia, servindo como referência para futuros projetos que também venham a se basear em SOA e BPM.

## 1.4. JUSTIFICATIVA

A escolha de SOA para o projeto do PECE, e para qualquer outro sistema ERP, é especialmente interessante em seu aspecto de interface entre diferentes sistemas ou partes de um mesmo sistema, que podem ser chamados de módulos e são descritos em 3.1. Sistemas tradicionais freqüentemente compartilham suas informações em nível de banco de dados relacional ou em classes entidade e seus métodos. A premissa do SOA vai além e permite o

reuso de serviços inteiros, inclusive operando entre diferentes tecnologias, como Java ou .NET.

O BPM, utilizando a notação BPMN descrita em 3.3, facilita o entendimento dos requisitos do sistema, pois permite a modelagem do trabalho das pessoas através de processos. Facilitando o entendimento do time de desenvolvimento e a validação pelos usuários do entendimento dos requisitos feita pelo time de desenvolvimento.

O BPM foi escolhido como uma ferramenta para formalizar os processos da instituição. Sua disposição gráfica e intuitiva, através da BPMN, permite que os conceitos sejam facilmente apresentados e compreendidos pelo cliente e pelo time de desenvolvimento simultaneamente. Em última instância, o BPM serve como garantia para os clientes de que os projetistas modelaram corretamente os processos da empresa cliente.

O grupo buscou com a orientadora uma opinião sobre algumas empresas que se tinha contato e que poderiam ser escolhidas para o projeto. No entanto, a empresa escolhida não foi nenhuma delas, e sim a sugestão dada pela orientadora que se mostrou a mais adequada aos objetivos: foi escolhido o PECE, Programa de Educação Continuada da Escola Politécnica da Universidade de São Paulo, que ministra cursos de extensão universitária. Esta decisão foi também influenciada pela proximidade à Escola, mas decisivo foi o sentimento do grupo de poder retribuir à USP o ensino dado nos 5 anos de graduação.

## 1.5. ESTRUTURA DO TRABALHO

No capítulo 2, a seguir, será descrita a aplicação escolhida, o PECE, em maiores detalhes. Ele mostra a organização e funcionamento dos diferentes unidades de negócio que

compõem o PECE. Também são abordados o sistema atual de gestão do negócio e suas deficiências, para se levantar então os requisitos do sistema ERP a ser desenvolvido.

No capítulo 3 é aprofundada a discussão sobre os temas escolhidos, definindo e ilustrando aplicações de *software* ERP e das tecnologias SOA e BPM. Uma definição de conceitos básicos de contabilidade também é apresentada.

O capítulo 4 mostra a metodologia de especificação que foi escolhida para o desenvolvimento do projeto. São listadas as fases do projeto e como estas foram escolhidas. Cada etapa é detalhada, assim como os seus correspondentes desafios. O capítulo 5 continua descrevendo a metodologia, porém focando nas seguintes etapas do processo de desenvolvimento de software: implementação, codificação, testes, aceitação e documentação do projeto final.

As conclusões acerca do resultado final do projeto são apresentadas no capítulo 6, e as considerações finais, como contribuições do trabalho e trabalhos futuros, são expostas no capítulo 7, que encerra a apresentação do projeto.

Todas as referências utilizadas ao longo do projeto e deste documento estão listadas no capítulo 8, sendo os anexos e os apêndices apresentados no final.



## 2 PECE

### 2.1. DESCRIÇÃO

O Programa de Educação Continuada (PECE) é um braço da diretoria da Escola Politécnica da Universidade de São Paulo (EPUSP). O PECE é responsável pela divulgação do conhecimento gerado na EPUSP através de Cursos de Extensão Universitária e “[...] atua também, como agente patrocinador, na realização de diversos eventos científicos de iniciativa de professores e pesquisadores da EPUSP, bem como na viabilização da vinda de professores estrangeiros, de reconhecimento consagrado pela comunidade científica”<sup>1</sup>. Sua condução é feita pelo diretor e pelo vice-diretor da EPUSP, assim como pelo coordenador geral do PECE.

Os cursos que o PECE oferece são classificados nas categorias presencial ou a distância. Nos cursos presenciais, alunos de diferentes origens se matriculam em um curso (ditos cursos abertos) ou uma empresa reserva uma turma composta só por seus funcionários (ditos cursos fechados). Em ambos os casos, as aulas são ministradas nas dependências da EPUSP ou da empresa, com presença obrigatória. Já nos cursos à distância, as aulas e palestras são gravadas em áudio e vídeo e disponibilizadas pelo PECE junto a outros materiais didáticos, como livros e apostilas. Nestes cursos há interações periódicas do aluno com os docentes através de conversas *on-line* ou troca de mensagens através de correio eletrônico.

---

<sup>1</sup> INSTITUCIONAL, Programa de Educação Continuada, Disponível em: <<http://www.pece.org.br/index.php?ind=home&menu=institucional>>. Acessado em: 15 de novembro de 2006.

Há também a disponibilização de um fórum de discussões. As provas finais e os laboratórios são obrigatoriamente presenciais. Os temas dos cursos estão distribuídos nas diversas áreas da engenharia e são dados em forma de atualizações, especializações, MBAs e difusões tecnológicas tipo *lato sensu*<sup>2</sup>. Cada curso é coordenado e ministrado por professores do seu departamento correspondente da EPUSP.

## 2.2. ESTRUTURA

O PECE é composto, abaixo da coordenação geral, por diversas unidades de negócio. A seguir são descritas as unidades mais importantes para o foco do trabalho.

### 2.2.1. A unidade de negócio acadêmica

A unidade de negócio acadêmica é responsável por manter todos os cadastros de alunos, turmas, professores, disciplinas e cursos do PECE.

<sup>2</sup> “Em sentido amplo” ou “em sentido estreito”, do latim. FERREIRA, A. B. H. Médio Dicionário Aurélio. Editora: Nova Fronteira. 1980.

A unidade de negócio acadêmica ainda recebe pedidos esporádicos da Central de Atendimento respondendo a pedidos de alunos de notas de uma disciplina ou a emissão de comprovantes de frequência e atestados de matrícula. Além dessas tarefas mencionadas ela faz também a matrícula dos novos alunos nos cursos e nas disciplinas, muda a grade horária de alunos, faz o controle de frequência e de notas. Ao final de um curso, providencia os certificados de cada aluno com as disciplinas cursadas e o histórico de frequências e notas.

Antes do projeto, todo o controle era feito por bases de dados distintas e não consolidadas, o que prejudica a coleta de qualquer tipo de informação solicitada.

### **2.2.2. A unidade de negócio financeira**

A unidade de negócio financeira do PECE mantém todo o histórico de transações financeiras de entrada e saída. É responsável pela divisão dos pagamentos e pelo rateio de custos entre os cursos, pelo pagamento de fornecedores, pelos repasses às outras fundações (FUSP, departamentos e RUSP) e pelo pagamento de salários dos professores e dos funcionários. A unidade também lida com os investimentos que o próprio PECE faz. Rotineiramente, a unidade de negócio financeira gera relatórios das movimentações financeiras do PECE como um todo (razão geral), assim como relatórios específicos para cada curso. Estes relatórios são apresentados à coordenação do PECE para controle interno.

A unidade de negócio utilizava um sistema contábil baseado em contas contábeis de entrada de recursos, de balanço de cada curso, de rateios diversos, entre muitas outras. O sistema usava a mesma base de dados da unidade de negócio acadêmica e também não supria todas as necessidades da unidade de negócio financeira. Muitas planilhas em Excel eram

usadas para apontamentos outros e, mesmo dentro da base de dados, os registros de alunos e cursos não eram utilizados em conjunto pelas duas unidades de negócio, causando muito retrabalho de cadastro de cursos, turmas, alunos e professores.

Esta unidade estava sobrecarregada de contas contábeis de difícil sumarização, fazendo com que relatórios esporádicos ou que fossem diferentes dos moldes atuais demorassem muito para serem gerados. Havia uma grande carência de maior transparência no controle das transações financeiras do PECE.

### **2.2.3. Infra-estrutura, central de atendimento e outras unidades de negócio**

A unidade de negócio de infra-estrutura lida com a alocação de salas, materiais e todo tipo de infra-estrutura necessária para o andamento dos cursos do PECE. Ele deve manter o registro das salas disponíveis e ser capaz de alocar e desalocar as turmas conforme a disponibilidade das salas, além de manter o controle dos acordos de uso das salas com os departamentos da EPUSP concedentes.

A central de atendimento é composta pelos funcionários que atendem os telefonemas, correios e pessoas que vão fisicamente ao PECE. Eles anotam os pedidos e repassam internamente para os departamentos responsáveis, gerando ordens de processo que são monitoradas por uma gerente.

Também é a unidade de negócio de tecnologia da informação, que assiste os funcionários com suas máquinas, gerencia a rede interna de computadores, mantém as páginas da Internet e é responsável pela visão estratégica do PECE em relação a adoção de novos sistemas, equipamentos e tecnologias. Esta unidade trabalhou em conjunto com o grupo no

desenvolvimento do módulo acadêmico do *software* ERP desenvolvido. Outras atividades compõem o PECE, como as jurídicas para a formalização de contratos entre alunos, empresas e professores.

### 3 CONCEITOS UTILIZADOS

#### 3.1. ERP (*ENTERPRISE RESOURCE PLANNING*)

##### 3.1.1. Definição

Sistemas ERP (*Enterprise Resource Planning*, gerenciador de recursos corporativos) são constituídos por *software* e/ou *hardware* genericamente responsáveis por gerenciar todos os recursos de uma empresa de maneira integrada <sup>3</sup>. Eles são compostos por conjuntos de funcionalidades específicas que atendem às necessidades de cada um dos departamentos da empresa, chamados módulos do ERP. A característica modular dos ERPs permitem que eles sejam criados, expandidos e substituídos de forma incremental, aumentando muito sua vida útil na empresa. Os módulos, apesar de estarem logicamente separados entre si, têm seu funcionamento integrado.

<sup>3</sup> KOCH, K. The ABCs of ERP: Getting started with Enterprise Resource Planning. Enterprise Resource Planning Research Center. Disponível em: <<http://www.cio.com/research/erp/edit/erpbasics.html>>. Acessado em: 30 de novembro de 2006.

A chave do funcionamento integrado dos ERPs é o compartilhamento de informações entre os módulos, normalmente explicitado no uso de uma base de dados comum e compartilhada. Assim como as empresas são de diferentes portes e complexidade de recursos, processos e controles, os ERPs também podem ser mais ou menos complexos e poderosos.

Seu funcionamento é imprescindível nas grandes empresas, pois permitem consultas em tempo real de informações muitas vezes complexas e dispostas em diferentes sistemas da empresa. Em empresas menores, seu uso beneficia a organização dos processos e no gerenciamento. Os sistemas ERP podem ter sua implementação nas empresas de apenas uma localidade ou podem estar distribuídos e interconectando diferentes locais.

### 3.1.2. Aplicação

Os ERPs são frequentemente usados para os diferentes tipos de gerenciamento da empresa. Para tal, alguns exemplos de módulos que compõem os ERPs são <sup>3</sup>:

- Financeiro;
- Produção;
- Estoques;
- Cadeia de fornecimento;
- Infra-estrutura;
- Recursos humanos;
- CRM (*Customer Relationship Management*);
- Etc.

Apesar de inicialmente ter sido desenvolvido para a gerência geral da empresa, os ERPs atualmente têm agregado módulos que cada vez mais modelam departamentos específicos. Com isso, a gerência média da empresa também vem se beneficiando dos recursos dos ERPs.

Os ERPs são vendidos como:

- Produtos de prateleira, requerendo da empresa a configuração dos módulos para atender os processos da empresa;
- Totalmente personalizados, sendo desenvolvido o sistema especificamente para a empresa;
- Um misto de funcionalidade de prateleira e módulos personalizados.

### 3.1.3. Exemplos

Dentre as empresas famosas de fornecimento de soluções de ERP de prateleira estão a SAP (23% de participação no mercado brasileiro), a Oracle (16%) e a Microsiga e RM (24%)

<sup>4</sup>. Já empresas que utilizem ERPs são muitas e em quase todos os ramos do mercado, como indústrias, bancos, prestadores de serviços, redes de supermercados, aeroportos, etc.

## 3.2. SOA (*SERVICE-ORIENTED ARCHITECTURE*)

<sup>4</sup> DECISION REPORT. Nasce a gigante nacional das soluções de gestão. Disponível em: <<http://www.decisionreport.com.br/publicue/cgi/cgilua.exe/sys/start.htm?infoid=231&sid=17>>. Acesso em: 30 de novembro de 2006.

### 3.2.1. Definição

A arquitetura orientada a serviços (*Service-Oriented Architecture*, SOA) é um modelo de arquitetura de sistemas computacionais de intercâmbio de informações do tipo cliente-servidor. Segundo Martucci, ela consiste em aplicações clientes acessando um sistema por terminais de interface e trocando informações com o servidor que detém o repositório de informações (informação verbal) <sup>5</sup>.

A SOA foi definida pela primeira vez em 1996 pela Gartner <sup>6</sup> mas, mesmo dez anos após a primeira definição, ainda há muita confusão sobre uma definição concreta. A SOA tornou-se uma arquitetura muito utilizada no mercado de desenvolvimento de *software*, porém, ainda segundo Natis <sup>6</sup>, “tentativas de uso da SOA causarão grandes sucessos e grandes falhas em projetos de software”. Isso porque a SOA é uma arquitetura baseada no uso de serviços, frequentemente utilizando *web-services*, e determinar a separação de serviços e regras de negócio é um trabalho difícil, uma vez que teoricamente qualquer regra de negócio pode ser implementada por *web-services*.

Ainda no conceito de Natis <sup>6</sup>, a SOA começa com a definição das interfaces do sistema e constrói a topologia do sistema como uma topologia de interfaces, implementação e chamada de interfaces.

---

<sup>5</sup> Informação fornecida pelo Prof. Moacyr Martucci Jr, em 2006, durante curso PCS-2554 Engenharia da Informação, Escola Politécnica da USP.

<sup>6</sup> SERVICE-ORIENTED ARCHITECTURE SCENARIO. Produzido por Yefim V. Natis. Disponível em: <<http://www.gartner.com/resources/114300/114358/114358.pdf>>. Acesso em: 13 de novembro de 2006.



Ele ainda acrescenta que a arquitetura orientada a serviços seria mais bem nomeada como arquitetura orientada a interfaces. Segundo o modelo de referência para a arquitetura SOA, OASIS, a SOA é definido como um paradigma para as organizações na utilização de competências que estão distribuídas em diferentes sistemas proprietários.

A arquitetura orientada a serviços segue três conceitos, ainda segundo o Modelo de Referência para Arquitetura Orientada a Serviço (OASIS) versão 1.0, que contribuem para a implementação de um sistema que permita a interface com outros sistemas.

O primeiro conceito é a visibilidade entre os sistemas que necessitam de competências que não possuem e os sistemas que possuem essas competências. Para que isso seja possível, existem regras bem definidas que estabelecem o contrato da comunicação entre os dois sistemas, versando sobre segurança, restrições, aspectos técnicos e mecanismos para acesso e resposta. As descrições devem ser acessíveis aos sistemas que necessitam as competências.

O segundo conceito é a interação, que é a atividade de uso da competência em si, feita basicamente pela troca de mensagens entre os dois sistemas e, conseqüentemente, é a troca de informações entre os sistemas. A interação está sempre ligada a um contexto de execução, que é o conjunto de elementos que forma o caminho entre o sistema com necessidades e o sistema com competências para suprir as necessidades.

O terceiro conceito é o efeito no mundo real que a interação tem. Esse efeito geralmente está associado à troca de informações, mas pode estar ligado também a alteração de estado dos sistemas. O SOA é adequado a serviços que demandam crescimento, promovem reuso, interoperabilidade, desenvolvimento incremental e adição posterior de novas regras de negócio, segundo Natis <sup>6</sup>.

Ainda segundo Natis, a interoperabilidade não é total, nem o sistema pode ser totalmente independente de plataformas e tecnologias na grande maioria dos casos <sup>6</sup>.

Nas organizações, a arquitetura orientada a serviços torna-se, num nível mais abstrato, uma mudança de paradigma na empresa ao promover crescimento, reuso e interoperabilidade. Esta arquitetura altera a forma como os sistemas são planejados, podendo ser fragmentados, realizados em plataformas diferentes, usando modelos de desenvolvimento diferentes sem que se perca a possibilidade da interação entre os sistemas. O SOA também cria a cultura de levantamento de possíveis interfaces que o sistema possa precisar no futuro, fazendo com que o desenvolvimento do sistema não mais fique apenas focado nos requisitos internos do sistema que podem ser implementados.

Ainda sobre os reais benefícios da utilização de SOA, muito se discute se o uso da arquitetura orientada a serviços não é apenas uma moda e por isso está sendo adotada nas grandes organizações. Segundo Pissello <sup>7</sup>, os reais benefícios do uso do SOA são de duas naturezas:

- Benefícios financeiros: redução de custo em integração com outros sistemas, reuso de aplicações, redução dos riscos do projeto, que estão relacionados a interfaces e reuso, melhorando também a qualidade do projeto em si. Outro benefício é o aumento da taxa de uso dos servidores, uma vez que as aplicações podem ser executadas em um ambiente virtualizado de hardware, através de uma arquitetura cliente/servidor.
- Benefícios de negócio: melhoria de produtividade com automação dos processos, aumento da velocidade com que os projetos podem ser desenvolvidos.

<sup>7</sup> IS THERE REAL BUSINESS VALUE BEHIND THE HYPE OF SOA? Produzido por: Tom Pissello, Alinean. Disponível em: <<http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=9001155&pageNumber=1>>. Acesso em: 13 de novembro de 2006.

Portanto, os processos são melhorados e há maior agilidade no caso de mudanças pelo uso de arquiteturas virtuais e serviços de interação entre os componentes.

Esses benefícios tangíveis e substanciais justificam o uso do SOA.

### **3.2.2. Componentes da arquitetura orientada a serviços**

A arquitetura orientada a serviços é caracterizada por uma camada no modelo de arquitetura cliente-servidor por camadas. Os sistemas computacionais de informação utilizam comumente a arquitetura de três camadas para o seu desenvolvimento: apresentação, negócios e dados. A nova camada inserida pela SOA é responsável pelos serviços de interface do sistema com os outros sistemas e outros módulos do mesmo sistema.

Este sistema gerenciador de serviços, além de dar suporte à execução dos serviços, ainda orquestra o funcionamento deles de acordo com os fluxos de negócio implementados pelo sistema, a fim de que os serviços sejam utilizados de forma coerente e gerando os resultados esperados.

### **3.2.3. Conceitos de serviços sob a ótica da SOA**

Para uma eficiente implantação de um sistema utilizando a arquitetura orientada a serviços, é necessário entender<sup>17</sup>:

- O que realmente é um serviço, isto é, de que ele é constituído;
- Como os serviços se relacionam entre si e com os demais componentes da aplicação.

Os serviços herdam muitas características dos outros elementos constituintes de um sistema, os objetos e componentes. Entretanto, os serviços também integram elementos dos fluxos de trabalho e processos de negócio, uma vez que são modelados justamente para implementar partes dos processos de negócios <sup>17</sup>.

As propriedades necessárias aos serviços é que eles tenham um acoplamento muito fraco, se possível, nenhum acoplamento. Isto significa que os usuários do serviço não precisam se moldar a ele: o serviço pode estar implementado em qualquer linguagem e ser executado em qualquer plataforma diferente da do usuário e, mesmo assim, deve funcionar.

Cada processo de negócio deve ter os seus serviços bem definidos e o domínio de atuação desses serviços não deve ser maior que o processo de negócio em si. Os serviços não devem ser executados atravessando processos de negócio como, por exemplo, começar no processo de cadastro de cliente e terminar na geração de relatórios. Eles também não devem ser usados com diferentes finalidades pelos processos, isto é, cada processo usando uma parte do serviço. Neste segundo caso, o serviço deveria ser quebrado para satisfazer cada serviço a apenas uma tarefa.

Essa característica se deve a dois principais fatores que podem estar presentes na construção de um sistema. O primeiro fator é a sobrecarga do sistema causada pelo acionamento de serviços que, por sua vez necessitam de outros serviços, e acabam por acessar diversos serviços em cascata, gerando sobrecarga indevida do sistema. O segundo fator é que, pela natureza dos serviços de interagir com outros sistemas, um serviço que um provedor de serviços fornece a outros sistemas pode deixar de ser fornecido, sendo necessária a sua substituição nos sistemas clientes desses serviços por outros serviços equivalentes. Nesse

caso, serviços que fossem acoplados ao serviço descontinuado deixarão de ser válidos e terão de ser reformulados.

Os serviços que estão relacionados a um mesmo domínio de negócios, interdependentes ou que implementem funcionalidades do mesmo processo, devem ser coesos. Essa coesão deve ser observada na comunicação entre os serviços, utilizando mensagens e protocolos de comunicação entre serviços padronizados. Esses serviços também devem apresentar coesão lógica por estarem agrupados no mesmo fluxo de atividade, para que a camada gerenciadora de serviços possa implementar o fluxo de atividade de forma objetiva.

No nível dos processos, estes devem realizar apenas uma tarefa do negócio e conter serviços necessários apenas para esta tarefa, para que o sistema apresente a característica de modularidade. Apesar de poder conter componentes reusáveis em sua estrutura, os processos nunca devem ser usados para implementar diferentes atividades. Sendo assim, os processos delimitam os domínios de coesão e interdependência dos serviços. Serviços que compõem processos diferentes não podem ser interdependentes para não ocasionar um acionamento em cascata de processos durante a execução de uma única atividade dentro do sistema.

#### **3.2.4. Desenvolvimento orientado a serviços**

O Desenvolvimento Orientado a Serviços, além de utilizar a Arquitetura Orientada a Serviços, versa sobre as etapas de desenvolvimento do software. O Desenvolvimento Orientado a Serviços <sup>8</sup> é comumente desenvolvido baseado na metodologia do Processo Unificado (*Unified Process*).

A característica principal do Desenvolvimento Orientado a Serviços é focar uma sessão no gerenciamento dos serviços, prevendo o gerenciamento destes durante todo o ciclo de vida do projeto. Este inclui as atividades de <sup>8</sup>:

- Identificação de serviços;
- Projeto da camada de serviços do sistema;
- Desenvolvimento dos serviços;
- Instalação dos serviços; e
- Manutenção.

Com isso, o projeto demanda o desenvolvimento sobre uma plataforma que dê suporte a gerenciamento, instalação e execução de serviços. Atualmente as plataformas de desenvolvimento e execução de software incluem suporte a serviços, como o Framework .NET 2.0 da Microsoft® <sup>9</sup>, que será utilizado neste projeto.

O ciclo de vida dos serviços contém as fases de análise, projeto, construção, teste, instalação, execução e monitoramento. Nas fases de análise e projeto, a captura de requisitos e a modelagem do sistema prevêm uma etapa de identificação das interfaces do sistema que constituirão serviços. Já as demais etapas focam o desenvolvimento dos serviços, teste, instalação e uma plataforma que suporte a execução e o monitoramento do funcionamento dos serviços junto ao software.

---

<sup>8</sup> PAPAOGLOU, M. P. Service-Oriented Design and Development Methodology. Disponível em: <<http://indscience.metapress.com/index/8K167G3E5A6U5APV.pdf>>. Acessado em: 13 de novembro de 2006.

<sup>9</sup> SHODJAI, P. Serviços da Web e a plataforma Microsoft. Disponível em: <[http://www.microsoft.com/brasil/msdn/Tecnologias/webservices/microsoft\\_plataform.msp](http://www.microsoft.com/brasil/msdn/Tecnologias/webservices/microsoft_plataform.msp)>. Acessado em: 13 de novembro de 2006.

### 3.3. BPM (*BUSINESS PROCESS MODELING*)

#### 3.3.1. Definição e exemplos

A notação da modelagem de processos de negócio BPMN (*Business Process Modeling Notation*) é uma das especificações concebida pelo BPMI (*Business Process Management Initiative*). A BPMI é uma organização independente, dedicada a desenvolver especificações para o gerenciamento de processos de negócio <sup>10</sup>, vindo a complementar iniciativas como o J2EE e SOAP, que permitiram a convergência de infra-estruturas legadas para a computação orientada a processos, além de outras como ebXML, RosettaNet, BizTalk, WSDL, UDDI, tpaML, e E-Speak que fornecem suporte a essa metodologia. O BPMI teve início com a empresa Intalio, o principal vendedor de soluções de BPM <sup>11</sup>, e associou organizações distintas para desenvolver processos de negócio os mais complexos possíveis. Além do BPMN, a BPMI fornece suporte a outros padrões da indústria com a mesma finalidade, como BPEL e BPEL4People. O BPMI.org iniciou em agosto de 2000 e contava com dezesseis empresas de *software* e consultorias, e dispõe de uma comunidade aberta para a entrada de outras companhias, organizações ou mesmo indivíduos.

---

<sup>10</sup> BPMI. Disponível em: < <http://www.bpmi.org/>>. Acessado em: 30 de novembro de 2006.

<sup>11</sup> INTALIO. Disponível em: < <http://www.intalio.com/company/profile/>>. Acessado em: 30 de novembro de 2006.

Como diz a própria sigla, BPMN é uma notação padronizada para modelagem de processos de negócio. A primeira versão de sua especificação foi lançada para o público em maio de 2004 <sup>12</sup>.

Seu principal objetivo era prover uma notação que fosse prontamente entendível tanto pelos usuários e gerentes do negócio, como pelos analistas de negócio responsáveis por criar a visão inicial dos processos, assim como pelo pessoal técnico (desenvolvedores) responsável por implementar a tecnologia que atuará nesses processos. Em suma, o BPMN foi concebido visando unificar a visão dessas diferentes frentes sobre o negócio, criando uma interligação entre o projeto do processo e sua implementação.

A notação é implementada através dos chamados BPDs (*Business Process Diagrams*, diagramas de processo de negócio), compostos por diversos tipos de elementos gráficos <sup>13</sup>, enumerados nos itens a seguir. Exemplos de diagramas elaborados, para maior entendimento, podem ser encontrados nas figuras de número 27 a 45.

#### 3.3.1.1. Evento



Figura 1 – Notação de evento

---

<sup>12</sup> WHITE, S. A. Introduction to BPMN. Disponível em: <<http://www.bpmn.org/Documents/Introduction%20to%20BPMN.pdf>>. Acessado em: 30 de novembro de 2006.

<sup>13</sup> STORELLI, F. O. Método de aplicação do SA-CMM em uma abordagem RM-ODP e BPMN. Disponível em: <[http://www.cqgp.sp.gov.br/downloads/Versao\\_Conclusao22-D\\_Correcao%20\\_V8.pdf](http://www.cqgp.sp.gov.br/downloads/Versao_Conclusao22-D_Correcao%20_V8.pdf)>. Acessado em: 30 de novembro de 2006.



A palavra evento deriva do latim *eventus*, que denota um acontecimento <sup>14</sup>. De fato, no contexto aqui visto, um evento significa algo que ocorreu no decorrer de um processo de negócio. Todo acontecimento tem a sua causa (chamada de gatilho ou *trigger*) e o seu efeito (resultado). Os três símbolos da figura 1 designam, respectivamente, um evento inicial, intermediário ou final.

#### 3.3.1.2. Atividade



Figura 2 – Notação de atividade

O retângulo com bordas arredondadas da Figura 2 ilustra uma atividade, que é um termo genérico que designa um trabalho realizado. Uma atividade que pode ser decomposta em outras atividades é dita não-atômica, sendo atômica caso contrário. Ela pode ser ainda dos tipos sub-processo (atividade que pode ser decomposta em outro processo, constituído por outras atividades) e tarefa (atividade que não é decomposta em outras). Para diferenciar um sub-processo, adota-se um sinal de adição centralizado sobre a aresta inferior do retângulo.

---

<sup>14</sup> PRIBERAM. Dicionário eletrônico. Disponível em: <<http://www.priberam.pt/>>. Acessado em: 30 de novembro de 2006.

#### 3.3.1.3. Passagem (*Gateway*)

É representada pelo losango ilustrado na figura 3. Aplica-se para controlar uma convergência ou divergência numa seqüência de um fluxo. Dessa forma, é um elemento que implica uma decisão (equivalentemente a um bloco de decisão, num fluxograma tradicional), bem como um desvio, uma ramificação, ou uma intercalação de caminhos. Marcas internas indicam seu tipo de comportamento.

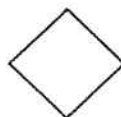


Figura 3 – Notação para passagem

#### 3.3.1.4. Fluxo de seqüência

Representada por este símbolo de vetor simples, o fluxo de seqüência conecta duas atividades, indicando a seqüência em que elas são executadas no processo (figura 4).



Figura 4 – Notação para fluxo de seqüência

#### 3.3.1.5. Fluxo de mensagem

Ilustrado na figura 5 (seta tracejada com um pequeno círculo em seu início), é utilizado para representar a troca de mensagens entre dois *pools*, que representam entidades de negócio, e serão descritos em 3.3.1.7.



Figura 5 – Notação do fluxo de mensagem

#### 3.3.1.6. Associação

É representada por uma linha pontilhada, que pode aparecer em BPDs com ou sem seta (figura 6). Como diz o nome, aplica-se para associar artefatos aos objetos de fluxo desejados.

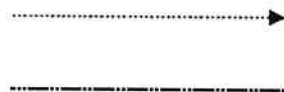


Figura 6 – Notação de associação

### 3.3.1.7. Área de Contenção (*Pool*)



Figura 7 – Notação da área de contenção

Um pool é representado por um retângulo, e corresponde ao conjunto de atividades de um participante do processo (entidade do negócio) (figura 7). Seu nome aparece na vertical, no canto esquerdo do retângulo. Sua forma serve para isolar o fluxo de atividades referentes a cada participante das demais entidades, que aparecerão fechados em outros *pools*.

### 3.3.1.8. Linha divisória (*Lane*)

A linha divisória aplicada no meio do *pool* serve para segmentar e separar em departamentos as atividades realizadas em cada entidade do negócio (figura 8).



Figura 8 – Notação para linha divisória

### 3.3.1.9. Objeto de dados

Os objetos de dados são representados por este ícone em formato de página da figura 9, e são ligados através de associações às atividades, representando dados de entrada ou saída para as mesmas.



Figura 9 – Notação do objeto de dados

### 3.3.1.10. Grupo

Tem o objetivo de representar o agrupamento de atividades, contornando-as com o símbolo da figura 10. Sua notação se assemelha ao das atividades, porém, desenhada com segmentos traço-ponto, em nada altera o fluxo do processo. Esse agrupamento aparece apenas

para fins de documentação e análise (geralmente para apontar notas válidas para um grupo de atividades).



Figura 10 – Notação para grupo

#### 3.3.1.11. Anotação

É utilizada quando o projetista precisa fornecer informações extras em forma de texto para os usuários do BPD.

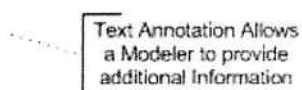


Figura 11 – Notação de anotação

### 3.3.2. Aplicação no trabalho

Os recursos de BPMN foram aplicados no projeto em sua fase inicial, para desenhar os processos do PECE a serem trabalhados. Seu uso se mostrou bastante eficaz por parte do

grupo, pois o esforço de entendimento da notação foi pequeno, dada a inteligibilidade da mesma, em contraponto à utilidade da mesma, pois ela se mostrou fundamental no sentido de formalizar o conhecimento obtido informalmente através de entrevistas com os funcionários do PECE.

Após a elaboração dos diagramas, os mesmos foram validados pela equipe de tecnologia do PECE. Também para essa etapa eles se mostraram facilitadores, pois a condensação de todos os dados recebidos na visão dos BPDs reduziu o trabalho do cliente, tendo sido uma tarefa pouco dispendiosa para o cliente repassar para o grupo incoerências observadas nos processos, a serem corrigidas. Outro ganho com os BPDs veio durante o próprio processo de confecção dos mesmos, pois este fomentou o grupo na criação da visão sistemática dos processos.

Por fim, os BPDs também fazem parte do produto de entrega, já que poderão ser de utilidade para futuras equipes que porventura dêem continuidade ao projeto do sistema.

### 3.4. CONTABILIDADE

Para entender e conseguir modelar o módulo financeiro do ERP do PECE, faz-se necessária uma apresentação da Contabilidade e como foi o seu uso no projeto.

#### 3.4.1. Conceitos básicos

É chamado de conta contábil todo conjunto de transações contábeis relativas a uma categoria de gastos, seja no Ativo, no Passivo ou no Patrimônio Líquido. Sua notação tradicional é o Razonete em T <sup>15</sup>, uma representação gráfica na qual são separados os débitos (dinheiro que entra na conta) e os créditos (dinheiro que sai da conta) em cada lado, como na

**Erro! A origem da referência não foi encontrada.** As contas são usadas para obter o balanço após cada operação financeira.

TÍTULO	
DÉBITO	CRÉDITO

Figura 12 – Notação em T de contas contábeis

As contas contábeis são usadas a fim de se verificar o balanço ao se comparar o total creditado e debitado. Notações tradicionais usam sinais diferentes para valores creditados e debitados, de modo que o balanço seja dado pela somatória de todas as transações.

Dentro das organizações e empresas são feitas muitas transações tanto externamente quanto internamente. Para controlar todas elas, são usadas diversas categorias de transações, como receitas de clientes, despesas administrativas e serviços de terceiros, por exemplo. Para cada categoria são usadas contas contábeis para acompanhar o balanço.

<sup>15</sup> IUDÍCIBUS, S. de. **Contabilidade Introdutória**. Editora Atlas S.A., 8ª edição. São Paulo, 1995.



Para a modelagem de uma transação, entre duas ou mais contas, é usado o método das partidas dobradas. Este método foi desenvolvido por Luca Pacioli no século XV, e “dá início a uma nova fase para a Contabilidade como disciplina adulta, além de desabrochar a Escola Contábil Italiana”<sup>16</sup>. Ele especifica que, para cada débito, é gerado ao mesmo tempo um crédito de igual valor. Tome-se o exemplo de uma conta ceder 300 dinheiros para uma outra conta 2. Duas transações de 300 no total, uma debitando 300 da primeira conta e outra creditando 300 na outra conta, como mostra a figura 13.

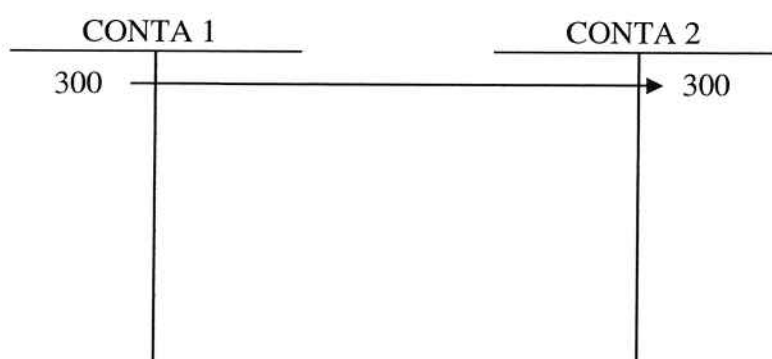


Figura 13 – Método das partidas dobradas de Pacioli

No PECE tem-se que, por exemplo, todas as entradas das mensalidades dos alunos entram em uma conta e da mesma conta são retirados valores para outras contas relativas a operações internas, como cursos, material, gastos administrativos, etc.

### 3.4.2. História

<sup>16</sup> MARION, José Carlos. **Contabilidade Básica**. Editora Atlas, 7ª edição.

Este tipo de organização contábil teve sua origem histórica quando cada conta contábil era armazenada em livros referentes a cada divisão contábil, chamados de livros razão <sup>16</sup>. Assim, para fazer uma consulta, por exemplo, ao fluxo de caixa de gastos com os salários dos funcionários, estes estariam todos centralizados em um só livro. Esta organização do fluxo contábil é muito mais rápida e prática para buscas manuais de transações de uma dada categoria.

No entanto, com o advento da informática e dos bancos de dados, uma consulta a centenas de milhares de transações é praticamente instantânea, aumentando muito o poder de busca em grandes agrupamentos de dados. Para tal, basta uma correta modelagem da base de dados usando chaves primárias eficientes, geralmente numéricas e incrementais.

Tais facilidades são usadas para auxiliar a contabilidade de modo a eliminar o tempo de busca manual e da geração dos livros contábeis, de modo que todas as transações foram inseridas no banco de dados e a pesquisa a uma transação, usando o conceito de conta contábil, é feita pelo número-chave da tal conta. Na prática, o uso de contas contábeis é feito apenas para identificar um conjunto de transações contábeis de um dado tipo, por exemplo, todas as de entrada de recursos vindos de clientes.

### **3.4.3. Uma proposta inovadora**

O grupo, após estudar as origens da contabilidade e sua evolução, entendeu que o uso de contas contábeis não se faz necessário em um razão geral, isto é, em uma listagem de transações de um dado período: o razão é orientado a fluxo de caixa, e o sistema poderia ser

orientado a transações. No lugar de se fazer a busca usando a chave das contas, propõe-se filtrar todas as transações usando diretamente os conceitos que formam a conta contábil. Assim, usando o mesmo exemplo do item anterior, se antes todas as entradas de clientes eram lançadas em uma conta com o número 1302, uma consulta às transações era feita no banco de dados a partir deste número.

Se forem criadas categorias de transações e classificações para os diferentes envolvidos em cada transação, pode-se fazer a mesma consulta filtrando por transações de entrada com o credor do tipo cliente. Esta modelagem orientada a fluxo de caixa direto centraliza todos os lançamentos em uma única conta, com a vantagem de maior simplicidade da base de dados e da menor curva de aprendizado em consultas financeiras, pois os filtros passam a ser intuitivos. Como opção, podem ser criadas chaves para conjuntos de filtros, de modo a compatibilizar o 1302 com os filtros de entrada de clientes.

Existe também o caso de uma transação de entrada no PECE servir de entrada em diversas outras contas, como por exemplo, um pagamento de mensalidade servir de entrada para o pagamento do curso em si e dos gastos administrativos. O modelo propõe que sejam criadas diferentes transações de entrada para cada credor, todas na categoria de pagamento de mensalidade, e referenciando um mesmo documento. No caso, o documento referenciado que une as duas transações é o boleto bancário pago pelo aluno. As figuras Figura 14 – e Figura 15 – ilustram como se modela este caso com contas contábeis e com o modelo proposto.

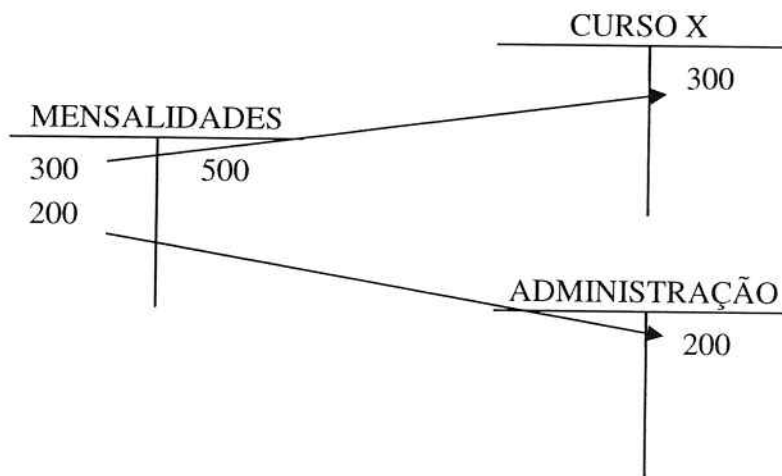


Figura 14 – Divisão do valor de uma mensalidade entre contas contábeis usando o método das partidas dobradas

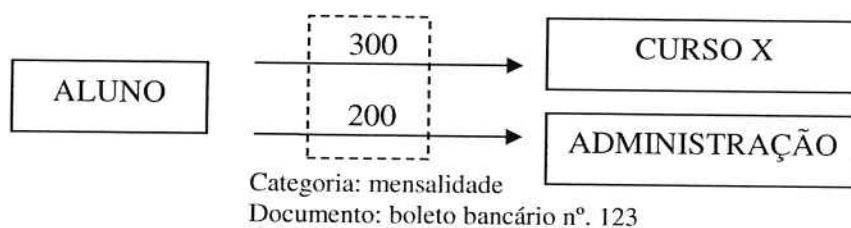


Figura 15 – Divisão de uma mensalidade para diferentes recebedores, referenciada pelo seu número do boleto bancário

Inversamente, existe o caso de saídas de diferentes contas entrarem em uma só, como por exemplo, o rateio dos custos de lanches entre os cursos.

A proposta do modelo é análoga, sugerindo a criação de transações individuais entre cada curso e o fornecedor, todas na categoria de pagamento de lanche e referenciando um mesmo documento, no caso, uma nota fiscal de saída. As figuras Figura 16 – e Figura 17 – ilustram o caso.

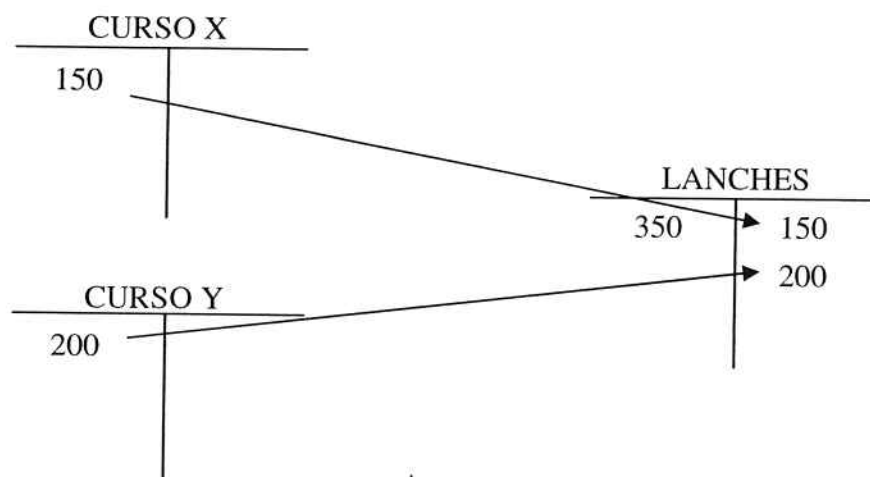


Figura 16 – Dois cursos pagando o rateio de um lanche, usando contas contábeis e o método das partidas dobradas

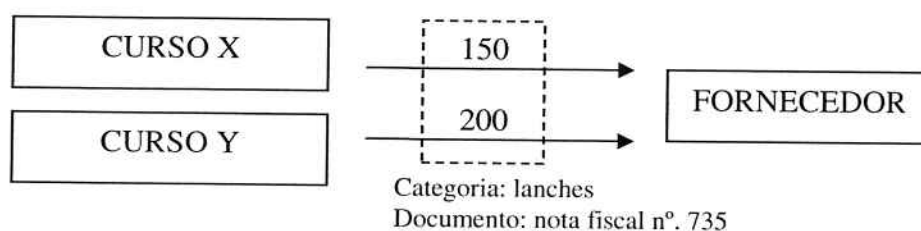


Figura 17 – Dois cursos pagando o rateio de um lanche, referenciando a mesma nota fiscal de saída

### 3.4.4. Vantagens e Desvantagens

O modelo apresentado consegue reproduzir a mesma movimentação financeira da contabilidade tradicional. Os relatórios de balanço financeiro (razão geral) que se podem obter dos dois modelos são iguais. No entanto, acredita-se que o uso do modelo se justifica pela facilidade de apresentação e operação. O modelo contém uma abordagem mais direta usando

envolvidos e categorias de transação, o que torna cada transação mais intuitiva e palpável. A organização em contas contábeis pode ser muito abstrata para se apresentar a um leigo, e a modelagem computacional mais complexa. Já o modelo proposto aumenta significativamente a transparência das operações financeiras, e a operação com o sistema é beneficiada tanto na inserção de novas transações quanto por desempenho das consultas em uma única tabela com todas as transações, evitando processamento dobrado de consulta a diferentes contas.

A maior desvantagem do modelo é a possível resistência à sua adoção dado que a contabilidade tradicional é usada há séculos, e o método das partidas dobradas já está amplamente difundido no mercado e na academia. O uso de um novo modelo tenta reinventar um padrão *de facto*, que já é jargão entre os profissionais do ramo. Para pessoas que estão habituadas à organização tradicional, esta mudança de paradigma pode ser confusa.

Porém, para o usuário final, isso será transparente, pois o sistema lhe dará relatórios análogos aos dados caso as partidas dobradas fossem utilizadas.

#### **3.4.5. Centro de custo**

No PECE, é chamada de centro de custo a identificação de um envolvido de uma transação. Assim, por exemplo, quando um professor recebe seu salário referente às aulas de um dado curso, o centro de custo é um código que identifica o curso responsável pela transação. Com base nos conceitos anteriores, tal código é compatível com a identificação do pagador envolvido em uma transação.

Na modelagem proposta, tais envolvidos estão identificados com chaves próprias, que são as mesmas do cadastro acadêmico dos cursos. Assim, a base de dados financeira está ligada à base acadêmica, tornando a base de dados unificada e o ERP integrado.

## 4 METODOLOGIA DO PROJETO

Como explicado na seção 3.2, o SOA insere novos componentes em um sistema cliente-servidor de três camadas. Logo, a primeira alternativa lógica para o desenvolvimento de um sistema utilizando esta arquitetura é a utilização de uma metodologia de desenvolvimento de sistemas consagrado, como o Processo Unificado (*Unified Process*), utilizando a linguagem UML (*Unified Modeling Language*) para representação dos componentes do sistema, assim como os diversos diagramas previstos pelo Processo Unificado <sup>17</sup>. Em particular, a versão 2.0 da UML contém conceitos e mecanismos para lidar com blocos lógicos do sistema que não são os componentes regulares da orientação a objetos, através dos quais os serviços podem ter sua representação adaptada.

Outra abordagem utilizada no desenvolvimento de sistemas utilizando a arquitetura orientada a serviços é a utilização da notação BPMN para o modelamento dos processos da organização para a qual se pretende desenvolver o sistema computacional para automação dos processos. Essa abordagem permite a identificação de serviços por todo fluxo de processos de negócio, facilitando a separação dos serviços de forma a atender os requisitos de coesão e baixo acoplamento. Ao utilizar essa abordagem, faz-se necessária a utilização de uma outra metodologia para complementar a modelagem do sistema, podendo ser o Processo Unificado.

---

<sup>17</sup> Stojanovic, Z; Dahanayake, A.; Sol, Z. Modeling and Design of Service-Oriented Architecture. In: 2004 IEEE International Conference on Systems, Man and Cybernetics Modeling and Design of Service-Oriented Architecture. The Hague, 2004.



Essa metodologia é utilizada no desenvolvimento do sistema de ERP do PECE por aproveitar a vantagem de utilização da linguagem BPMN para a definição dos serviços e depois a complementação pelo Processo Unificado por ser uma metodologia amplamente utilizada e já de conhecimento dos desenvolvedores do sistema.

#### 4.1. LEVANTAMENTO DE REQUISITOS

Para o levantamento dos requisitos do sistema ERP do PECE foram feitas entrevistas com os funcionários de diferentes unidades de negócios e os resultados foram discutidos como o gerente de TI e co-orientador do grupo, o engenheiro Leonardo. As entrevistas começaram no dia 11 de julho de 2006 com funcionárias da unidade de negócio acadêmico e da central de atendimento. No dia 13 do mesmo mês, foi entrevistado mais um funcionário da unidade de negócio acadêmica, assim como o próprio Leonardo. A partir do dia 17 de agosto, entrevistas foram feitas com frequência na unidade de negócio financeira, sempre esclarecendo os procedimentos que se fazia na época e os problemas que o sistema antigo apresentava.

As entrevistas seguiram os conceitos do Documento de Elaboração especificado no SERVPRO<sup>18</sup>, isto é, usando conceitos de uma planilha que categoriza as atividades, seus elementos de contato e seus determinantes de qualidade.

<sup>18</sup> SANTOS, L. C.; VARVAKIS, G. **SERVPRO**: uma técnica para gestão de operações de serviços. São Paulo: Universidade de São Paulo, 2002. (Revista Produção v. 12 n. 1)

#### **4.1.1. Módulo Acadêmico**

O módulo acadêmico do sistema ERP do PECE deve conter todos os cadastros de alunos, professores, disciplinas e cursos, havendo a possibilidade de se criarem turmas para os oferecimentos das disciplinas. Também é necessário que se gerem listas de presença e, posteriormente, guarde a presença individual em cada aula. O módulo também deve quantificar as aulas lecionadas em um dado período, sendo esta informação usada então pelo módulo financeiro no cálculo da remuneração do professor. Adicionalmente, o sistema deve ter a flexibilidade de incluir as disciplinas escolhidas pelo aluno, assim como permitir a prévia requisição da troca de alguma disciplina em cursos que tenham essa opção.

As notas de cada aluno, em cada disciplina, devem ser inseridas no sistema e este deve gerar relatórios de histórico escolar a qualquer momento do curso, assim como deve gerar o histórico escolar final a ser impresso no certificado. Todos os relatórios, certificados e diplomas devem poder ser exportados para Microsoft Excel ou Word, para serem impressos em papel timbrado padronizado do PECE ou da USP.

#### **4.1.2. Módulo Financeiro**

O módulo financeiro deve registrar todas as transações financeiras do PECE entre as entidades envolvidas, a data, o valor e a finalidade do gasto. O módulo deve ainda usar as

mesmas informações cadastradas pelo módulo acadêmico, eliminando-se a duplicidade de dados e os possíveis erros que possam decorrer.

Assim como o módulo acadêmico, o financeiro deve gerar relatórios a partir dos dados cadastrados. Relatórios periódicos gerados devem ser gerados, como o balanço mensal por curso e o balanço bancário do PECE. Entretanto, a maior funcionalidade exigida é a geração simples e rápida de relatórios esporádicos, a qual não era atendida pelo sistema anterior.

Os funcionários também perdiam muito tempo no lançamento manual de transações de repasse às instituições controladoras do PECE. Existe a necessidade de o sistema ser capaz de gerar as transações referentes aos repasses automaticamente, tornando necessário que se cadastrem regras de repasses em função do tipo de transação que for inserida. O módulo também deve se comunicar com o sistema do Banco Real, onde o PECE tem conta corrente, para a geração automática dos boletos de pagamentos dos alunos.

## 4.2. DEFINIÇÃO DOS PROCESSOS

Definida a metodologia a ser usada e entrevistados os funcionários do PECE, o grupo passou a formalizar o aprendizado na forma de diagramas BPMN, descrita em 3.3, denota caixas para as tarefas e as setas indicam o andamento do processo. Losangos são usados para desvios condicionais, cujas condições são explicitadas no texto dentro deles. Os retângulos que envolvem os processos são as entidades, como o PECE ou um aluno, por exemplo.

A metodologia especifica que a modelagem seja *top-down*, isto é, que comece do nível mais abrangente e generalista, para então explodir os agrupamentos de tarefas de maneira hierárquica em novos diagramas. Os agrupamentos estão notados na forma de um símbolo de

adição, “+”, envolto em um retângulo dentro de uma tarefa. A Figura 18 – ilustra o nível mais alto da unidade de negócio acadêmica, isto é, o ciclo de vida de cada curso e tarefas que foram detalhadas em outros diagramas.

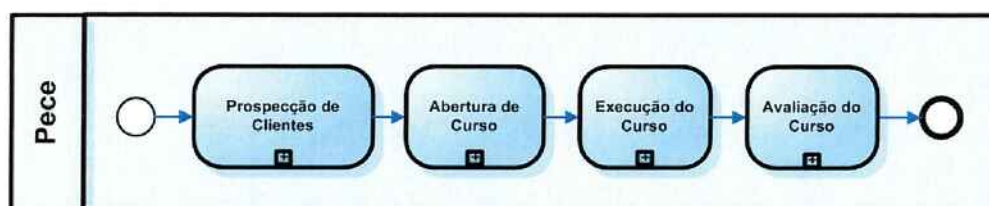


Figura 18 – Ciclo de vida de um curso no PECE modelado no nível mais alto

Tomando por exemplo o agrupamento de tarefas “Prospecção de Clientes” e explodindo em um novo diagrama, tem-se a Figura 19 – .

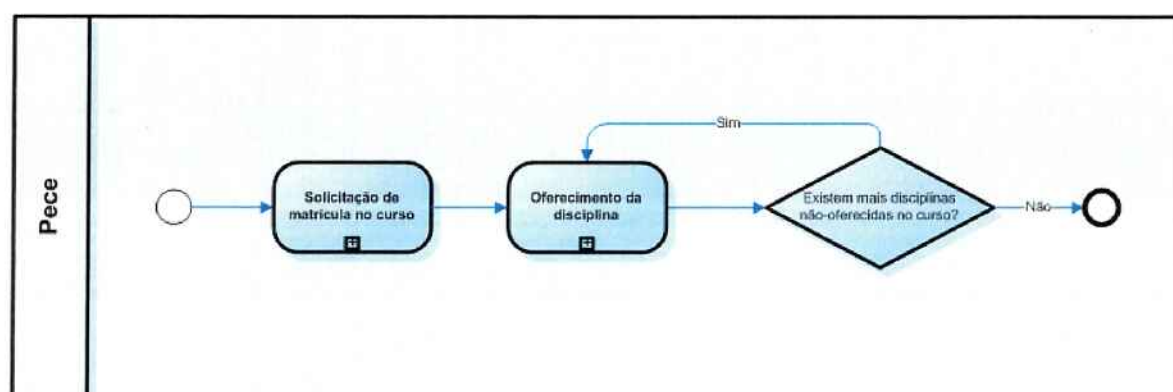


Figura 19 – Detalhamento da prospecção de clientes

A Figura 19 – exemplifica como um agrupamento de tarefas pode ser explodido em um novo diagrama, fazendo com que o conjunto de diagramas crie uma árvore. Explodindo

novamente um agrupamento de tarefas, desta vez usando um agrupamento da Figura 19 – , obtém-se a Figura 20 – , em que diferentes entidades interagem entre si.

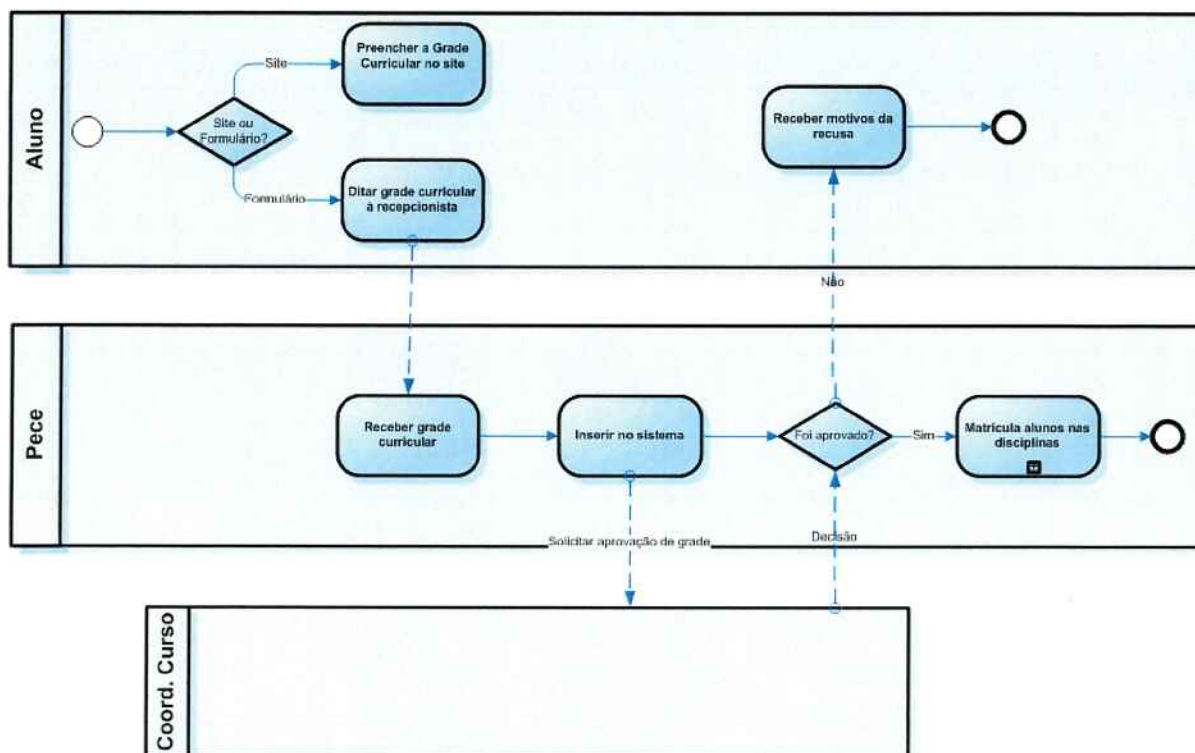


Figura 20 – Detalhamento da solicitação de matrícula, envolvendo diferentes entidades

Assim, explodindo sucessivamente agrupamentos de tarefas, chegou-se ao detalhamento julgado como suficiente pelo grupo. O critério usado para decidir quando parar de detalhar foi baseado no valor que o novo detalhamento agregaria à compreensão do processo e ao projeto do sistema. Todos os processos mapeados pelo grupo estarão listados no apêndice de BPMs para referência.

### 4.3. CASOS DE USO

Os casos de uso foram especificados pelo grupo principalmente para o pacote financeiro, pois o desenvolvimento do módulo acadêmico foi feito internamente pelo PECE. Dentro do financeiro, os casos de uso foram divididos nas categorias cadastros e relatórios, que serão listados a seguir.

O diagrama de casos de uso é mostrado na figura Figura 21 – a seguir. Suas descrições estão em anexo.



Figura 21 – Diagrama de casos de uso com ator único, o funcionário.

#### 4.4. ARQUITETURA DO SISTEMA E DESCRIÇÃO DOS SERVIÇOS



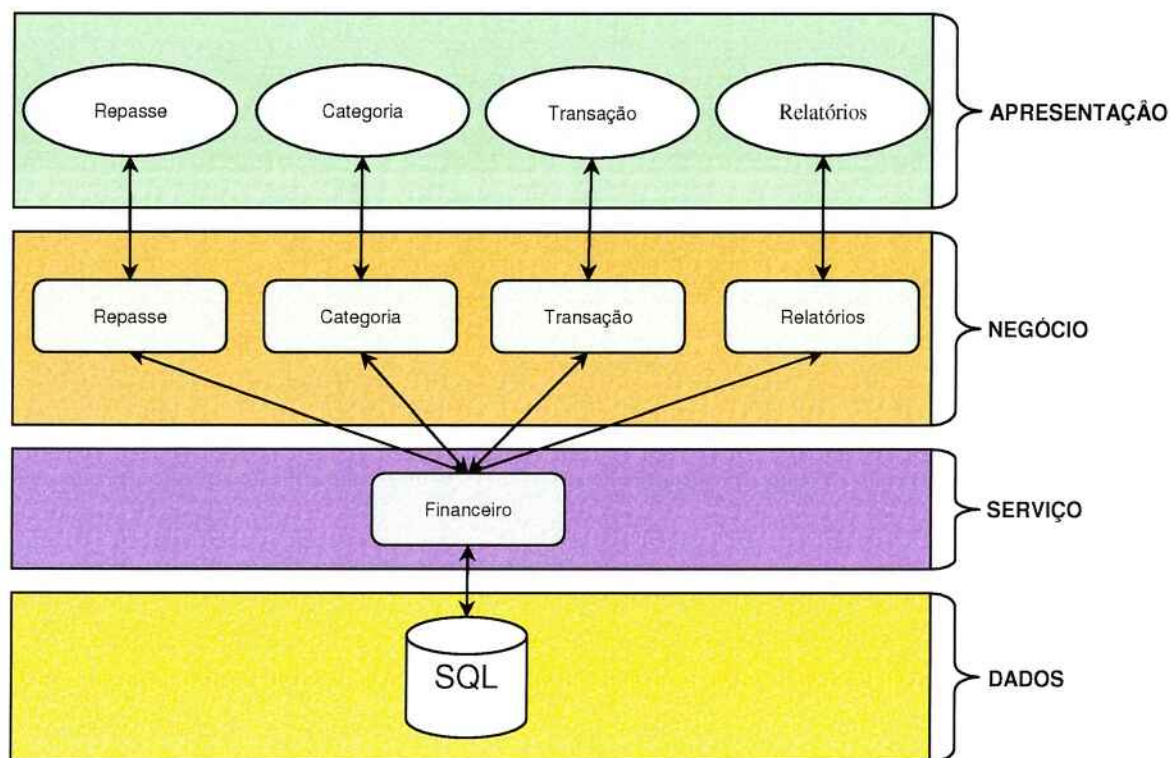


Figura 22 – Diagrama de arquitetura

O sistema desenvolvimento se baseia no diagrama de arquitetura em quatro camadas exibida na figura 22. Na camada de apresentação, responsável por expor a lógica de negócio ao usuário e possibilitar a interação do usuário com o sistema, encontra-se, por exemplo, a tela de cadastro de repasse, a tela de cadastro de transação e as telas de relatórios. Resume-se, portanto, basicamente aos arquivos de extensão *.aspx*.

A camada de negócio, responsável pela manipulação dos valores inseridos pelo usuário na camada de apresentação agrupa, por exemplo, os dados inseridos na tela Cadastro de Fornecedor em um objeto (no caso, o objeto Fornecedor) o qual é enviado para o *web-service* de cadastramento de fornecedor.

A camada de dados apresenta todos os comandos de consulta e inserção ao banco de dados relacional em SQL, como, por exemplo, a consulta de todos os repasses existentes na tabela Repasse ou o comando de inserção na tabela Fornecedor.



A camada de serviço, implementada em *web-services*, é responsável pela comunicação das estações clientes com o servidor, além da interoperabilidade do sistema PECE com outros sistemas. De modo a atender os casos de uso especificados, foi concluído que os *web-services* deverão realizar basicamente três ações:

- Inserir novo cadastro
- Atualizar um cadastro existente
- Buscar/selecionar dados de um cadastro já existente

Inicialmente, foi cogitada a opção de criar apenas um *web-service* por entidade para atender essas três ações. Logo isso se tornou inviável, uma vez que cada uma dessas ações é específicas e, ao ter um *web-service* tão complexo, aumentaria o custo e trabalho de uma possível manutenção ou expansão que sistema venha a receber no futuro. Inclusive, ao construir apenas um método (*web-service*) para todas as ações, mas isso desrespeitaria regras de boas praticas de programação que foram adotadas no início do projeto, uma vez que o método não estará representando apenas um caso de uso, mais sim três. Decidiu-se então que cada entidade teria três *web-services*, cada uma abrangendo uma das ações mencionadas.

Essa decisão atendeu basicamente as necessidades de inserção/atualização/busca no banco de dados. O próximo passo definiu quais *web-services* deveriam ser criados para disponibilizar os dados que seriam escolhidos pelo usuário para realizar os cadastros. Por exemplo, ao realizar o cadastro de repasse, é necessário informar o pagador, o credor e o receptor. Esses são dados a serem buscados do cadastro de alunos ou fornecedores ou professores ou curso.

Os *web-services* definidos até então não atendem completamente essas consultas, uma vez que para estes casos é necessário exibir apenas informações básicas do cadastro, e os *web-services* de busca já definidos trazem inúmeras informações (a maioria desnecessária). Inclusive, seria de grande importância ter disponível um serviço que disponibilizará os

envolvidos cadastrados no PECE para uma possível expansão do sistema (por exemplo, comunicação do sistema com o banco no qual o PECE é correntista). Viu-se que devem ser criados *web-services* que busquem todos os envolvidos, utilizando como argumento o tipo do envolvidos (fornecedor, aluno, professor, ou curso).

Por último, analisaram-se as necessidades de serviços para os relatórios. Por serem pesquisas específicas, ficou decidido que será criado um *web-services* para cada relatório. Como foi entregue somente um relatório de fluxo de caixa, foi criado somente um *web-service*. O relatório é implementado na forma de *web-service* pelo fato de poder ser integrado com o Microsoft® Excel, um software externo ao sistema.

#### 4.5. INFRA-ESTRUTURA

O ERP foi desenvolvido em plataforma ASP.NET e codificado em Visual Basic.NET 2.0, atendendo a especificação feita pela gerência de TI do PECE. Sua interface é uma página HTML e seu acesso pelos funcionários é feito através de qualquer navegador de Internet, como o Microsoft Internet Explorer ou o Mozilla Firefox, por exemplo.

Assim, o sistema apesar de estar logicamente dividido nas 3 camadas tradicionais (interface, negócios e banco de dados) mais a nova camada de interface do SOA, ele até o momento da implementação estava todo concentrado em um único servidor, chamado desenvolvimento. Este executa o servidor de páginas de Internet Microsoft IIS e o banco de dados Microsoft SQL Server 2000.

## 5 IMPLEMENTAÇÃO E VALIDAÇÃO

### 5.1. CODIFICAÇÃO

A codificação começou com o módulo acadêmico desenvolvido pela gerência de TI do PECE. Em outubro de 2006, quando começou a fase de codificação do módulo financeiro, já haviam sido desenvolvidas as funcionalidades de autenticação e alguns dos cadastros acadêmicos. Já haviam sido criadas todas as suas telas, assim como o *framework*, isto é, as classes e telas que formam a base do sistema. Assim, o grupo teve a etapa de codificação restrita em escopo e em prazo.

A linguagem usada foi ASP.NET e Visual Basic.NET 2.0 em notação polonesa, compilada em Visual Studio 2005, todos da Microsoft, seguindo a especificação da gerência de TI do PECE. Foi utilizada a orientação a objetos e algumas boas práticas como arquivos de configuração dos comandos de conexão centralizados e em XML, por exemplo. Foram usados tipos comuns e as referências aos nomes das tabelas usaram constantes, além da implementação de classes de entidade e tratamentos de exceções usando try/catch.

### 5.2. TESTES

O código foi testado sem o auxílio de ferramentas de teste unitário automatizado, isto é, programas que rodam trechos de código a cada nova versão armazenada no repositório. Por necessidades de prazo, adotou-se a metodologia de testes tradicional: na medida em que se codificava, eram testados exemplos no código ou nas telas para a validação dos procedimentos. Ao término de cada grupo de funcionalidades, todas eram testadas para verificar se o desenvolvimento das últimas não alterou o funcionamento das primeiras.

### 5.3. VALIDAÇÃO

O código, depois de desenvolvido, passou por validação da gerência de TI do PECE. As diversas funcionalidades foram testadas uma a uma seguindo uma planilha de testes desenvolvida para tal. A validação teve foco principalmente no cumprimento dos casos de uso especificados e acordados entre o grupo e o PECE.

Assim como os casos de uso do módulo financeiro, a planilha abrange:

- O cadastro de dados primários: testa a inserção, edição e exclusão de categorias, fornecedores e dados bancários dos cursos;
- O cadastro de dados secundários, que usam os primários: mesmas operações para o cadastro de repasses e das transações financeiras.
- O relatório de transações financeiras: teste de cada um dos filtros e da combinação deles, assim como as funcionalidades de exportação para Excel e impressão.

Somente validado o sistema através da planilha de testes ele poderá ser considerado como validado formalmente.

#### 5.4. IMPLANTAÇÃO

Por motivos de prazo, a implantação do sistema não pôde ser feita pelo grupo. A gerência de TI do PECE, que acompanhou todo o desenvolvimento do projeto no módulo financeiro e que desenvolveu o módulo acadêmico, será responsável por colocá-lo em operação, fazer eventuais modificações, fazer a manutenção e expansão do sistema.

#### 5.5. DOCUMENTAÇÃO DO PROJETO

O manual do usuário entregue pelo grupo ao PECE refere-se às funcionalidades e ao uso do módulo financeiro desenvolvido. Ele aborda os mesmos princípios teóricos do modelo de contabilidade adotado e que foram descritos neste trabalho, além do manual de uso das telas do sistema. Informações técnicas como a configuração dos comandos de configuração e conexão ao servidor de banco de dados, armazenados em arquivos XML, também serão descritos e exemplificados.

O manual do módulo acadêmico, com os mesmos tópicos apresentados no manual do módulo financeiro, também será desenvolvido internamente pelo PECE, assim como o módulo em si.

### 6 ANÁLISE DO RESULTADO

O projeto visou o estudo de tecnologias recentes no desenvolvimento de sistemas ERP, para então implementá-las no PECE. O estudo de BPM se mostrou relevante para o grupo, já que seus integrantes não conheciam a notação, e conseguiram ficar a par da sua utilidade dentro de um projeto. Porém o processo de confecção de todos os diagramas e sua validação acabou por consumir mais tempo que o esperado, pois eram muitos processos a serem modelados. Este período poderia ter sido canalizado para atividades posteriores do projeto. Percebeu-se, porém, que a não construção dos diagramas aumentaria o esforço de equipes posteriores a entender os processos. Concluímos então quanto aos diagramas, que foi um esforço a mais do grupo que terá um resultado a longo prazo.

Quanto ao estudo da SOA, este foi bastante proveitoso. O grupo pôde perceber com o projeto a diferença entre saber o conceito de serviço, que é o nível no qual o grupo se encontrava antes do projeto, para implementar na prática a Arquitetura Orientada a Serviços. Este será um conhecimento a ser levado para o mercado de trabalho, pois a SOA está sendo bastante difundida nas empresas. Outro ponto observado é que a filosofia da orientação a serviços não deverá, nos momentos iniciais que sucedem a entrega do produto final, ser utilizado pelo PECE. O projeto prevê uma futura integração com outros sistemas, da USP e externos, como sistemas bancários, e acredita-se que quando essa necessidade se concretizar, essa característica do sistema poderá ser utilizada em sua plenitude.

A parte de contabilidade empregada no projeto mostrou-se o ponto mais surpreendente do trabalho. Quando os integrantes resolveram elaborar o desenvolvimento de ERP como projeto de formatura, esse tópico não estava previsto na abordagem. Fez-se necessário um estudo dos conceitos envolvidos, dada a insipiência dos componentes do grupo quanto ao tema. E o resultado obtido mostrou-se acima das expectativas, não só por ter agregado conhecimento ao grupo, mas principalmente pelo grupo ter proporcionado ao projeto uma abordagem diferente da convencional no tratamento das transações contábeis e sua

modelagem em banco de dados. Esse último ganho foi encarado com entusiasmo, pois não de início não se esperavam, no projeto, inovações em áreas que não fossem da área de Engenharia de Software.

Enfim, destacaram-se como aspectos positivos do trabalho a profundidade nos estudos da SOA e de contabilidade. Porém, o grupo aponta, como principal ponto que deixou um tanto a desejar, a implantação do sistema. Quando da elaboração do cronograma, o planejamento realizado acabou por conceder prazos mais alargados para etapas anteriores, em detrimento desta última. Acabou-se obtendo, com o pouco tempo para sua realização, um resultado que não condizia com as expectativas nesse aspecto.

## 7 CONSIDERAÇÕES FINAIS

### 7.1. CONTRIBUIÇÕES DO TRABALHO

O projeto trouxe contribuições importantes ao grupo no que concerne às partes teóricas e práticas da Engenharia de Software, complementando a formação na área, em conjunto os estudos realizados nas disciplinas regulares do curso de Engenharia Elétrica – Ênfase Computação. Os estudos realizados pelo grupo foram fundamentais para o entendimento de novas metodologias de desenvolvimento de software e características do processo em si de desenvolvimento.

O uso do BPM para a modelagem dos processos do PECE mostrou-se útil principalmente para a correta modelagem de processos de negócio já existentes e que estão sendo automatizados pelo sistema computacional desenvolvido. O uso do BPM permitiu a modelagem correta dos fluxos de atividade dos processos, melhorando a eficiência da captura de requisitos funcionais, deixando menor margem para ambigüidades. A validação dos processos com os usuários tornou-se fácil pela característica visual dos diagramas gerados e o processo de automatização ficou mais objetivo, pois se puderam definir as tarefas a serem automatizadas e verificar que elas são transparentes para o usuário final do sistema.

Uma vantagem adicional do uso do BPM para a modelagem é que este evidencia os processos que possuem interface com outros sistemas e que sejam candidatos a serem implementados através de web-services, o que facilita a implementação da Arquitetura Orientada a Serviços (SOA).



A desvantagem percebida do uso do BPM é que ele demanda mais tempo do que a captura normal de requisitos que é consolidada e validada mais rapidamente com os clientes; entretanto, a captura normal de requisitos não mostra para o cliente onde esses requisitos serão satisfeitos dentro do sistema.

Em relação à arquitetura do sistema, a SOA provou-se muito útil para sistemas modulares, que promovem reuso de componentes e que possuem interface com outros sistemas, sendo, os serviços, a forma de automatizar os processos que envolvem troca de mensagens entre sistemas distintos. Pelo estudo realizado das diferentes arquiteturas disponíveis, concluiu-se que não há arquitetura de sistema computacional específica para o desenvolvimento de sistemas ERP, mas a SOA é a arquitetura que possui mais características que atendem as necessidades de um sistema ERP. A SOA é uma arquitetura cliente-servidor de n camadas e que possibilita grande reuso de componentes através dos serviços e desenvolvimento modular, que é uma característica do sistema ERP, que é dividido em módulos de acordo com as divisões internas do negócio no qual está sendo implantado o ERP. A SOA também se mostra uma arquitetura com grande futuro pelo potencial de fazer os sistemas diferentes se comunicarem, o que é a grande limitação da arquitetura tradicional de três camadas do desenvolvimento de software; por isso o aprendizado da mesma é muito importante.

No que concerne ao desenvolvimento do módulo financeiro em si, o estudo de contabilidade proporcionou um maior entendimento do funcionamento da contabilidade em geral. Tal estudo permitiu o desenvolvimento de um sistema mais simples de cadastro de transações financeiras, gerando relatórios financeiros consistentes e com a mesma estrutura dos relatórios financeiros gerados anteriormente, resolvendo os problemas atuais da contabilidade do PECE.

Pessoalmente, o projeto também teve o papel fundamental de ser um projeto real, no qual o grupo lidou pela primeira vez com clientes reais. O projeto impôs a captura correta dos requisitos dos clientes e a transformação destes em um sistema que automatiza o trabalho feito atualmente, dando mais agilidade e conforto para a realização das tarefas que compõem os processos de negócio do cliente. No caso do projeto, o cliente é o PECE, uma instituição ligada à Escola Politécnica e, conseqüentemente, à Universidade de São Paulo. Isto causou uma satisfação pessoal por colaborar com a Escola que contribuiu tanto com a formação profissional e pessoal dos integrantes do grupo.

## 7.2. TRABALHOS FUTUROS

O ERP integrou as informações de duas unidades de negócios do PECE, e muito se pode obter ainda deste novo sistema. No entanto, outras unidades de negócio ainda poderão ser modeladas e incorporadas ao sistema, aumentando ainda mais o escopo do ganho de produtividade.

Alguns trabalhos futuros que podem ser realizados no PECE, no contexto deste projeto de formatura são listados a seguir:

- O controle de pedidos feitos pela central de atendimento: formalização dos pedidos feitos por pessoas à central de atendimento e que são repassadas para as unidades de negócios internas. Poderia ser feito um cadastro de pedidos com a data, o requisitante e a descrição, podendo os funcionários marcar o pedido como atendido e à gerência acompanhar o andamento.

- Desenvolvimento do módulo de infra-estrutura: para cadastrar e acompanhar a alocação dos recursos materiais e a alocação das salas de aula do PECE.
- Integração das informações acadêmicas com a página na Internet do PECE: permite que os alunos, depois de validado o acesso, tenham disponíveis informações acadêmicas, podendo gerar automaticamente comprovantes de frequência e de notas.
- Formatação dos relatórios: os relatórios gerados com o serviço desenvolvido poderiam ter sua formatação adequada ao rodapé e ao logotipo do PECE. Assim, os relatórios resultantes poderiam ser impressos diretamente da interface gráfica do ERP já com formatação de papel timbrado oficial.
- Desenvolvimento da interface com o banco em que o PECE é correntista: poderia ser criada para atualizar automaticamente as informações referentes aos pagamentos dos alunos no banco de dados do ERP. Isto pouparia o trabalho manual dos funcionários, aumentando a produtividade.
- Geração de demonstrativos contábeis: poderia se expandir o módulo financeiro para que fossem gerados demonstrativos contábeis para o PECE, como a demonstração dos resultados do exercício e o balanço patrimonial, utilizando os dados e movimentações financeiras cadastradas.

### 7.3. CONCLUSÃO DO GRUPO

O grupo, pela primeira vez, teve contato com o desenvolvimento de um software com cliente real, e se dedicou fortemente às fases iniciais do desenvolvimento. Estas foram desde a

proposta geral de sistema (um software do tipo ERP utilizando a SOA como arquitetura do sistema), o modelo BPM para se levantar os processos e validar com os funcionários responsáveis pelas etapas dos processos. Para que isso fosse possível, o grupo se aprofundou bastante no estudo dos temas, principalmente na parte de arquitetura de software até que se encontrasse uma arquitetura que satisfizesse os requisitos do sistema pretendido, sendo que esta foi a SOA.

O grupo também se dedicou bastante na etapa de levantamento de requisitos para conhecimento do sistema, realizando várias entrevistas com os clientes e aprendendo bastante sobre captura de requisitos na prática. Houve grande preocupação para entender os requisitos, principalmente quanto ao módulo financeiro do sistema a ser desenvolvido, por tratar de contabilidade, uma área na qual o grupo teve pouco conhecimento, demandando um estudo detalhado sobre o assunto. Tal estudo acabou gerando uma nova forma de se realizar o cadastro das transações financeiras, simplificando o trabalho dos funcionários da contabilidade por meio da automatização do sistema e gerando os mesmos relatórios com os quais os funcionários comumente trabalham.

A parte do desenvolvimento do sistema, principalmente a que concerne à codificação do software, foi prejudicada pela falta de tempo necessário para a tarefa, principalmente pelo fato de usar uma linguagem não familiar aos integrantes do grupo e ao excesso de trabalhos das outras disciplinas, principalmente próximo ao encerramento do ano e deste projeto. Esses trabalhos de outras disciplinas impedem uma dedicação maior do que o projeto de formatura merece, tendo em vista a contribuição deste em relação aos outros para a formação dos alunos.

O apoio dos orientadores foi fundamental. O auxílio geral prestado pela orientadora Profa. Dra. Selma Shin Shimizu Melnikoff, orientando sobre os aspectos gerais do projeto foi crucial para o bom andamento do projeto, auxiliando-nos a atingir os objetivos propostos. O

co-orientador engenheiro Leonardo Dominguez Dias auxiliou muito o grupo acompanhando o desenvolvimento em detalhes, orientando atividades, revisando os trabalhos e fornecendo o suporte necessário para o desenvolvimento do projeto. Ambos estiveram muito presentes e disponíveis para esclarecimentos sobre o projeto em si e também sobre a disciplina de Projeto de Formatura.

O desempenho da maioria dos componentes do grupo foi bem equilibrado, tendo feito suas participações nas etapas cruciais do projeto e na divisão das tarefas, nas quais um ou mais membros se aprofundavam no cumprimento das tarefas, gerando um paralelismo e permitindo que as tarefas fossem realizadas dentro do prazo estabelecido. A organização e a comunicação dos componentes do grupo permitiram uma grande sincronização do andamento do projeto, não havendo aspectos a corrigir neste sentido. No entanto, um integrante se manteve afastado na maioria das atividades do projeto, contribuindo pouco para sua realização, estando presente apenas na maior parte das reuniões.

## 8 REFERÊNCIAS

BPMI. Disponível em: < <http://www.bpmi.org/>>. Acessado em: 30 de novembro de 2006.

DECISION REPORT. Nasce a gigante nacional das soluções de gestão. Disponível em: <<http://www.decisionreport.com.br/publique/cgi/cgilua.exe/sys/start.htm?infoid=231&sid=17>>. Acesso em: 30 de novembro de 2006.

FERREIRA, A. B. H. Médio Dicionário Aurélio. Editora: Nova Fronteira. 1980.

Informação fornecida por Moacyr Martucci Jr, em 2006, durante curso PCS-2554 Engenharia da Informação, Escola Politécnica da USP.

INSTITUCIONAL, Programa de Educação Continuada, Disponível em: <<http://www.pece.org.br/index.php?ind=home&menu=institucional>>. Acessado em: 15 de novembro de 2006.

INTALIO. Disponível em: < <http://www.intalio.com/company/profile/>>. Acessado em: 30 de novembro de 2006.

IS THERE REAL BUSINESS VALUE BEHIND THE HYPE OF SOA? Produzido por: Tom Pisello, Alinean. Disponível em: <<http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=9001155&pageNumber=1>>. Acesso em: 13 de novembro de 2006.

IUDÍCIBUS, S. de. Contabilidade Introdutória. Editora Atlas S.A., 8ª edição. São Paulo, 1995.

KOCH, K. The ABCs of ERP: Getting started with Enterprise Resource Planning. Enterprise Resource Planning Research Center. Disponível em: <<http://www.cio.com/research/erp/edit/erpbasics.html>>. Acessado em: 30 de novembro de 2006.

MARION, José Carlos. Contabilidade Básica. Editora Atlas, 7ª edição.

PAPAZOGLU, M. P. Service-Oriented Design and Development Methodology. Disponível em: <<http://inderscience.metapress.com/index/8K167G3E5A6U5APV.pdf>>. Acessado em: 13 de novembro de 2006.

PRIBERAM. Dicionário eletrônico. Disponível em: <<http://www.priberam.pt/>>. Acessado em: 30 de novembro de 2006.

SANTOS, L. C.; VARVAKIS, G. SERVPRO: uma técnica para gestão de operações de serviços. São Paulo: Universidade de São Paulo, 2002. (Revista Produção v. 12 n. 1)

SERVICE-ORIENTED ARCHITECTURE SCENARIO. Produzido por Yefim V. Natis. Disponível em: <<http://www.gartner.com/resources/114300/114358/114358.pdf>>. Acesso em: 13 de novembro de 2006.

SHODJAI, P. Serviços da Web e a plataforma Microsoft. Disponível em: <[http://www.microsoft.com/brasil/msdn/Tecnologias/webservices/microsoft\\_plataform.msp](http://www.microsoft.com/brasil/msdn/Tecnologias/webservices/microsoft_plataform.msp)>. Acessado em: 13 de novembro de 2006.

Stojanovic, Z; Dahanayake, A.; Sol, Z. Modeling and Design of Service-Oriented Architecture. In: 2004 IEEE International Conference on Systems, Man and Cybernetics Modeling and Design of Service-Oriented Architecture. The Hague, 2004.

STORELLI, F. O. Método de aplicação do SA-CMM em uma abordagem RM-ODP e BPMN. Disponível em: <[http://www.cqgp.sp.gov.br/downloads/Versao\\_Conclusao22-D\\_Correcao%20\\_V8.pdf](http://www.cqgp.sp.gov.br/downloads/Versao_Conclusao22-D_Correcao%20_V8.pdf)>. Acessado em: 30 de novembro de 2006.

WHITE, S. A. Introduction to BPMN. Disponível em: <<http://www.bpmn.org/Documents/Introduction%20to%20BPMN.pdf>>. Acessado em: 30 de novembro de 2006.

## ANEXOS

### MODELAGEM DE EMPRÉSTIMO ENTRE CONTRAS CONTÁBEIS

No caso de um envolvido não poder efetuar uma transação, porque seu balanço se tornará negativo, usou-se o artifício de um terceiro envolvido participar na transação na como sendo um credor. Assim, a transação tem um pagador e um receptor, e é acrescida de um credor que empresta dinheiro para o pagador. Ao se fazer o balanço do pagador, constará a dívida com o credor, que deverá ser paga posteriormente por meio de uma outra transação. A Figura 23 – ilustra o exemplo.

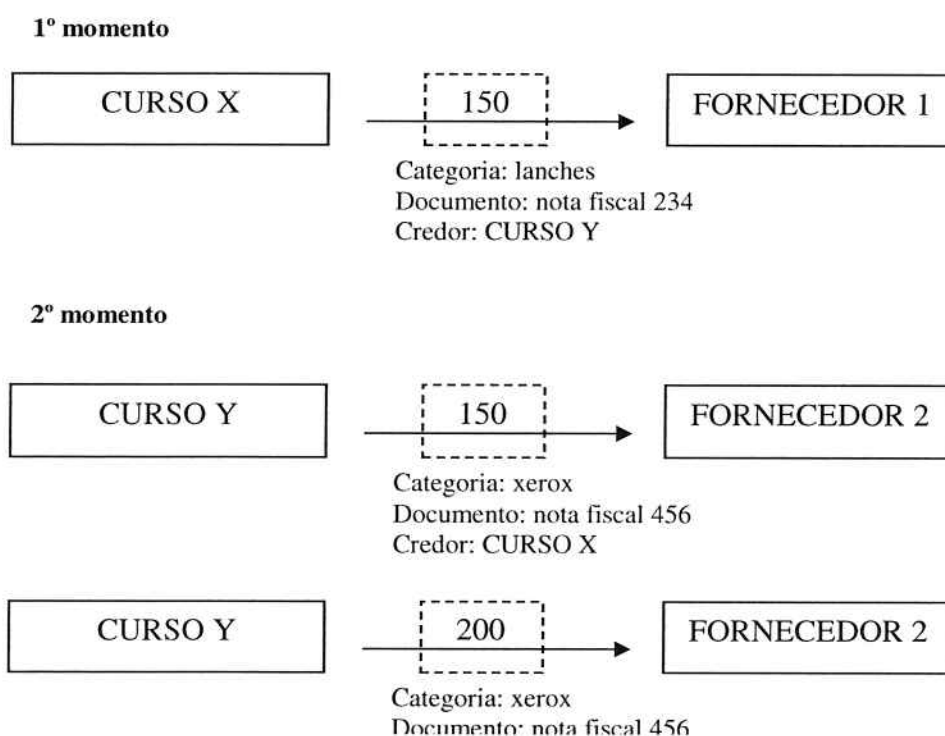


Figura 23 – Curso Y empresta para Curso X, Curso X devolve como parte de um pagamento do Curso Y



## CASOS DE USO DE CADASTROS

### **Caso de Uso 1: Inserção de dados financeiros dos cursos**

**Descrição:** O sistema deve inserir os dados bancários dos cursos, sendo estes iguais ou não ao da fundação controladora.

**Ator:** Funcionário.

**Evento Iniciador:** ator clica em inserir dados financeiros do curso.

**Pré-Condição:** Funcionário autenticado no sistema

#### **Seqüência de eventos:**

- a) O sistema lista todos os cursos cadastrados na tela.
- b) Ator escolhe o curso desejado, e o sistema mostra as informações bancárias do curso.  
Se não houver cadastro, mostra os campos em branco.
- c) O ator pode preencher os dados (nome do banco, números da agência e da conta corrente) manualmente, ou selecionar uma das fundações que o sistema copia os dados nos campos.
- d) Ator salva os dados, o sistema armazena o cadastro.

**Pós Condição:** Cadastro armazenado no sistema.

**Extensões:** Nenhuma.

**Inclusões:** Nenhuma.

### **Caso de Uso 2: Edição de dados financeiros dos cursos**

**Descrição:** O sistema deve permitir que se editem os dados bancários dos cursos.

**Ator:** Funcionário.

**Evento Iniciador:** ator clica em editar dados financeiros do curso.

**Pré-Condição:** -

**Seqüência de eventos:**

- a) O sistema lista todos os cursos cadastrados na tela.
- b) Ator escolhe o curso desejado, e o sistema mostra as informações bancárias do curso.  
Se não houver cadastro, o sistema avisa.
- c) O ator pode editar os dados (nome do banco, números da agência e da conta corrente) manualmente, ou selecionar uma das fundações que o sistema copia os dados nos campos.
- d) Ator salva os dados, o sistema armazena o cadastro.

**Pós Condição:** Cadastro editado armazenado no sistema.

**Extensões:** Nenhuma.

**Inclusões:** Nenhuma.

### **Caso de Uso 3: Inserção de cadastro de fornecedores**

**Descrição:** O sistema deve ser capaz de cadastrar uma lista de fornecedores do PECE.

**Ator:** Funcionário.

**Evento Iniciador:** ator clica em inserir cadastro de fornecedores.

**Pré-Condição:** -

**Seqüência de eventos:**

- a) Ator insere dados cadastrais do fornecedor: nome, CNPJ, endereço, etc.
- b) Ator clica em cadastrar.

**Pós Condição:** Sistema armazena cadastro do fornecedor.

**Extensões:** Sistema verifica se não há redundância de CNPJ. Caso haja, sistema avisa e não insere os dados.

**Inclusões:** Nenhuma.

#### **Caso de Uso 4: Edição de cadastro de fornecedores**

**Descrição:** O sistema deve ser capaz de editar um cadastro existente de fornecedores do PECE na lista.

**Ator:** Funcionário.

**Evento Iniciador:** ator clica em editar cadastro de fornecedores.

**Pré-Condição:** fornecedor já cadastrado.

**Seqüência de eventos:**

- a) O sistema apresenta uma lista de fornecedores.
- b) Ator escolhe um fornecedor.
- c) Os dados cadastrais do fornecedor escolhido são disponibilizados para edição.
- d) Após editar, o ator manda gravar.

**Pós Condição:** Sistema armazena cadastro editado do fornecedor.

**Extensões:** Nenhuma.

**Inclusões:** Nenhuma.

#### **Caso de Uso 5: Inserção de cadastro de repasse**

**Descrição:** O sistema deve ser capaz de inserir no cadastro um repasse associado a um conjunto de envolvidos. Exemplo: repasse dos 13% da receita para a RUSP e EPUSP, ICMS, ISS, etc.

**Ator:** Funcionário.

**Evento Iniciador:** ator seleciona inserir cadastro de repasse.

**Pré-Condição:** 2 ou mais envolvidos cadastrados.

**Seqüência de eventos:**

- a) O ator deve escolher o pagador, o receptor e o credor de uma lista de envolvidos dependendo do tipo (aluno, professor, fornecedor, instituição). O pagador pode ser especificado por sua categoria.
- b) Ator clica em cadastrar.
- c) Sistema cadastra a despesa para o curso.

**Pós Condição:** Repasse cadastrado no sistema.

**Extensões:** Nenhuma.

**Inclusões:** Nenhuma.

### **Caso de Uso 6: Edição de cadastro de repasse**

**Descrição:** O sistema deve ser capaz de editar no cadastro um repasse associado a um conjunto de envolvidos.

**Ator:** Funcionário.

**Evento Iniciador:** ator seleciona editar cadastro de repasse.

**Pré-Condição:** repasse já cadastrado.

**Seqüência de eventos:**

- a) O sistema apresenta uma lista dos repasses cadastrados.
- b) Ator escolhe um e clica em editar.
- c) Sistema apresenta os campos do repasse (envolvidos, descrição e ponderador) para o ator editar.
- d) Ator edita campos e manda salvar.

**Pós Condição:** Cadastro editado salvo no sistema.

**Extensões:** Nenhuma.

**Inclusões:** Nenhuma.

### **Caso de Uso 7: Inserção de cadastro de categorias de transações**

**Descrição:** O sistema deve ser capaz de cadastrar categorias de transações contábeis em forma de árvore.

**Ator:** Funcionário.

**Evento Iniciador:** ator clica em inserir cadastro de categorias de transações

**Pré-Condição:** 2 ou mais envolvidos, 1 ou mais repasses cadastrados, caso sejam necessários.

#### **Seqüência de eventos:**

- a) O sistema apresenta o primeiro nível da árvore de categorias.
- b) O ator pode selecionar uma categoria existente ou navegar em subníveis de categorias abaixo.
- c) Ator preenche o nome da nova categoria. O sistema apresenta uma lista de repasses cadastrados. Opcionalmente, o ator pode inserir repasses relacionados a esta categoria e manda gravar.
- d) Sistema grava a nova categoria.

**Pós Condição:** Nova categoria de transação cadastrada e pronta para ser usada nas transações.

**Extensões:** Nenhuma.

**Inclusões:** Nenhuma.

**Caso de Uso 8: Edição de cadastro de categorias de transações**

**Descrição:** O sistema deve ser capaz de editar as categorias de transações contábeis já cadastradas.

**Ator:** Funcionário.

**Evento Iniciador:** ator clica em editar um cadastro de categoria de transações.

**Pré-Condição:** categoria de transação já cadastrado.

**Seqüência de eventos:**

- a) O sistema apresenta o primeiro nível da árvore de categorias.
- b) O ator pode selecionar uma categoria existente ou navegar em subníveis de categorias abaixo.
- c) Ator edita o nome e os repasses de uma categoria existente e manda gravar.
- d) Sistema sobrescreve o nome da categoria.

**Pós Condição:** Categoria de transação cadastrada, como novo nome e pronta para ser usada nas transações.

**Extensões:** Nenhuma.

**Inclusões:** Nenhuma.

**Caso de Uso 9: Remoção de cadastro de uma categoria de transações**

**Descrição:** O sistema deve ser capaz de remover as categorias de transações contábeis já cadastradas.

**Ator:** Funcionário.

**Evento Iniciador:** ator clica em remover um cadastro de categoria de transações.

**Pré-Condição:** categoria de transação já cadastrado.

**Seqüência de eventos:**

- a) O sistema apresenta o primeiro nível da árvore de categorias.
- b) O ator pode selecionar uma categoria existente ou navegar em subníveis de categorias abaixo.
- c) O ator manda remover. Somente categorias folha da árvore serão removidas.
- d) Sistema confirma a operação com o ator e depois remove a categoria escolhida.

**Pós Condição:** Categoria de transação removida. Todas as transações históricas da categoria removida são automaticamente atribuídas a sua categoria pai.

**Extensões:** Nenhuma.

**Inclusões:** Nenhuma.

#### **Caso de Uso 10: Inserção de cadastro de tipo de documento**

**Descrição:** O sistema guarda o cadastro de tipos de documentos relacionados a uma transação. Ex: documentos anexos a transação como nota fiscal, recibo, etc.

**Ator:** Funcionário.

**Evento Iniciador:** ator seleciona inserir cadastro de tipo de documento.

**Pré-Condição:** Nenhuma.

**Seqüência de eventos:**

- a) Sistema lista todos os tipos de documentos já cadastrados.
- b) Ator preenche nome do novo tipo de documento e manda gravar.
- c) Sistema verifica se o nome já existe. Se não existir, cadastra o novo tipo.

**Pós Condição:** Novo tipo de documento cadastrado.

**Extensões:** Nenhuma.

**Inclusões:** Nenhuma.

**Caso de Uso 11: Edição de cadastro de tipo de documento**

**Descrição:** O sistema deve ser capaz de editar o cadastro de tipos de documentos relacionados a uma transação.

**Ator:** Funcionário.

**Evento Iniciador:** ator seleciona editar cadastro de tipo de documento.

**Pré-Condição:** Nenhuma.

**Seqüência de eventos:**

- a) Sistema lista todos os tipos de documentos já cadastrados.
- b) Ator escolhe um tipo, edita o nome e manda gravar.
- c) Sistema verifica se o nome já existe. Se não existir, guarda o cadastro do tipo.

**Pós Condição:** Cadastro do tipo de documento editado.

**Extensões:** Nenhuma.

**Inclusões:** Nenhuma.

**Caso de Uso 12: Inserção de cadastro de transação contábil**

**Descrição:** O sistema guarda uma transação contábil referente a uma categoria entre 2 envolvidos, com a opção de haver 1 credor.

**Ator:** Funcionário.

**Evento Iniciador:** ator seleciona inserir cadastro de transação contábil.

**Pré-Condição:** Uma ou mais categorias de transação cadastradas, 2 ou mais envolvidos cadastrados.

**Seqüência de eventos:**



- a) O ator deve escolher: uma categoria de transação; o pagador, receptor e o credor de uma lista de envolvidos dependendo do tipo (aluno, professor, fornecedor, instituição), a data e o valor.
- b) Ator clica em cadastrar.
- c) Sistema cadastra a despesa para o curso.

**Pós Condição:** Despesa cadastrada referente ao curso selecionado. Despesas de repasse automaticamente criadas para os envolvidos, contendo o número da transação original no campo descrição.

**Extensões:** Nenhuma.

**Inclusões:** Nenhuma.

## CASOS DE USO DE RELATÓRIOS

**Caso de Uso 13:** Geração de relatório de transações contábeis.

**Descrição:** O sistema deve ser capaz de gerar relatórios de transações customizados, usando os campos que compõem cada transação como filtro: pagador, receptor, credor, categoria, data, etc. Estes podem ser exportados para o MS Excel.

**Ator:** Funcionário.

**Evento Iniciador:** ator seleciona gerar relatório de transações contábeis.

**Pré-Condição:** Uma ou mais transações cadastradas.

**Seqüência de eventos:**

- a) Sistema apresenta os campos que compõem cada transação, listando períodos, categorias, datas, valores, envolvidos, etc. Cada campo está inicialmente sem nenhuma seleção.
- b) Ator escolhe quais campos ele quer usar como filtro e quais ele vai deixar sem nenhuma seleção, e manda gerar o relatório.
- c) Sistema apresenta na tela o relatório desejado, com a opção de exportar para uma planilha Excel.

**Pós Condição:** Relatório impresso ou exportado.

**Extensões:** Nenhuma.

**Inclusões:** Nenhuma.

# APÊNDICE

## Projeto de Formatura – Turmas 2006



### PCS - Departamento de Engenharia de Computação e Sistemas Digitais

#### Engenharia Elétrica – Ênfase Computação

**Tema: Desenvolvimento de software ERP (Enterprise Resource Management), usando BPM e SOA, aplicado ao PECE**

O grupo desde o início decidiu estudar as últimas tecnologias de desenvolvimento de softwares ERP, para então desenvolver um sistema que fosse futuramente implantado em uma empresa. O PECE foi escolhido por necessitar deste mesmo tipo de sistema e também pela proximidade à escola, além de ser uma forma de retribuição do grupo à EPUSP.

A missão do sistema era integrar e automatizar os departamentos acadêmico e financeiro do PECE, substituindo sistemas legados e demorados procedimentos manuais que eram usados a anos. Todos os cursos, alunos e professores deveriam estar cadastrados em bases de dados que também fossem utilizadas pelo departamento financeiro. O trabalho foi separado em dois, em que o grupo modelou os dois departamentos mas só desenvolveu o financeiro, tendo sido o acadêmico desenvolvido internamente.

O grupo começou pesquisando sobre a notação BPM (Business Process Modeling), que seria usada para formalizar a modelagem que foi feita dos processos do instituto. A seguir, entrevistas com os funcionários foram feitas para entender o funcionamento dos departamentos como um todo. Criaram-se os diagramas dos processos, sendo estes mais detalhados nos departamentos que o grupo focou.

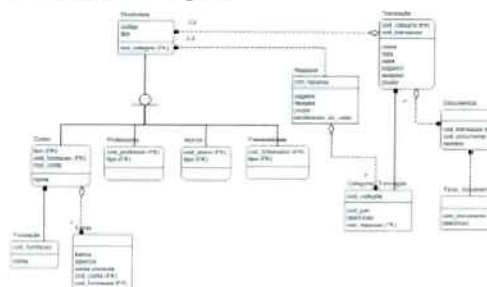
Passou-se então a estudar a arquitetura orientada a serviços (SOA) em diversos artigos e fontes acadêmicas, e como esta arquitetura se aplica em sistemas ERP. Mais que isso, era necessário buscar uma metodologia que abordasse a separação da lógica de negócios em web-services de modo formal.

Definida a separação ideal dos serviços e adotando uma metodologia de desenvolvimento específica, começou a prototipação das telas e a sua integração no sistema base que vinha sendo desenvolvido em paralelo internamente pelo PECE.

**Integrantes:** Felipe Martins da Silva  
Fernando Raffani  
João Paulo Seabra Santos  
Matias Eduardo Reccius Puga

**Professor Orientador:** Selma Shin Shimizu Melnikoff  
**Co-orientador:** Leonardo Dominguez Dias

Foi criado o Diagrama de Entidade-Relacionamento do departamento financeiro, que incluía inovações na própria maneira do sistema armazenar e apresentar as transações financeiras. As tabelas do banco de dados foram criadas em seguida.



O sistema foi então codificado usando Visual Basic e ASP.NET 2.0, linguagens que já vinham sendo adotadas pelo PECE no desenvolvimento do módulo acadêmico. Foram criadas as telas definitivas e implantada a lógica de negócios dos cadastros básicos, das transações e dos relatórios financeiros, estes utilizando o componente Crystal Reports.

Por fim, o sistema foi testado e implantado até o fim do período, tendo os integrantes escrito não só a documentação da disciplina mas também o manual do usuário para o PECE. O sistema final foi adotado pelos funcionários em seguida, e abrem-se muitas possibilidades de expansões futuras do sistema baseadas no mesmo core que foi desenvolvido pelo grupo no projeto de formatura.

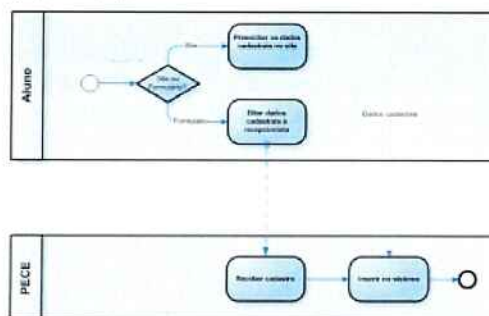


Figura 24 – Pôster do projeto de formatura.

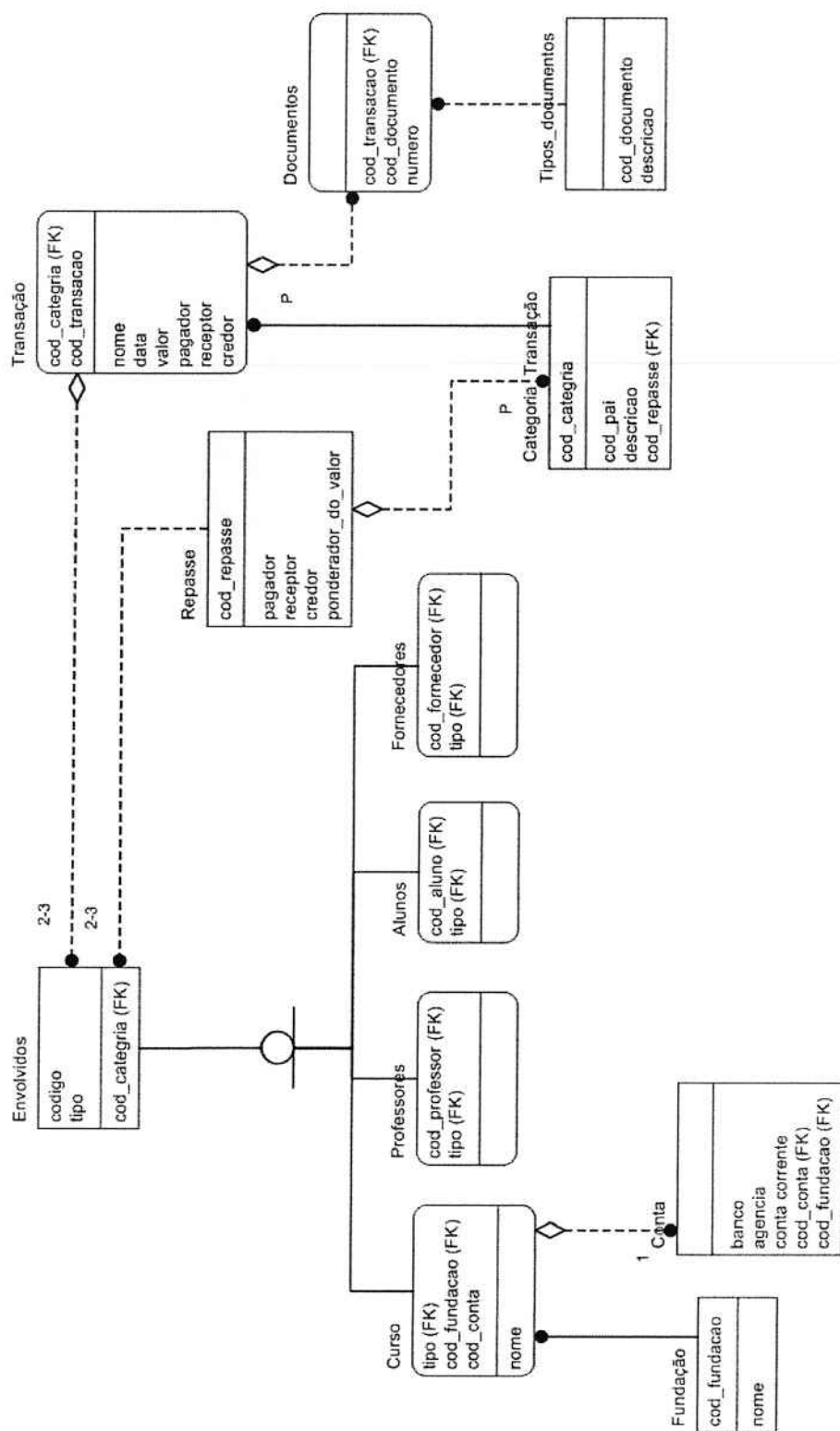


Figura 25 – Diagrama de Entidade-Relacionamento do módulo financeiro.

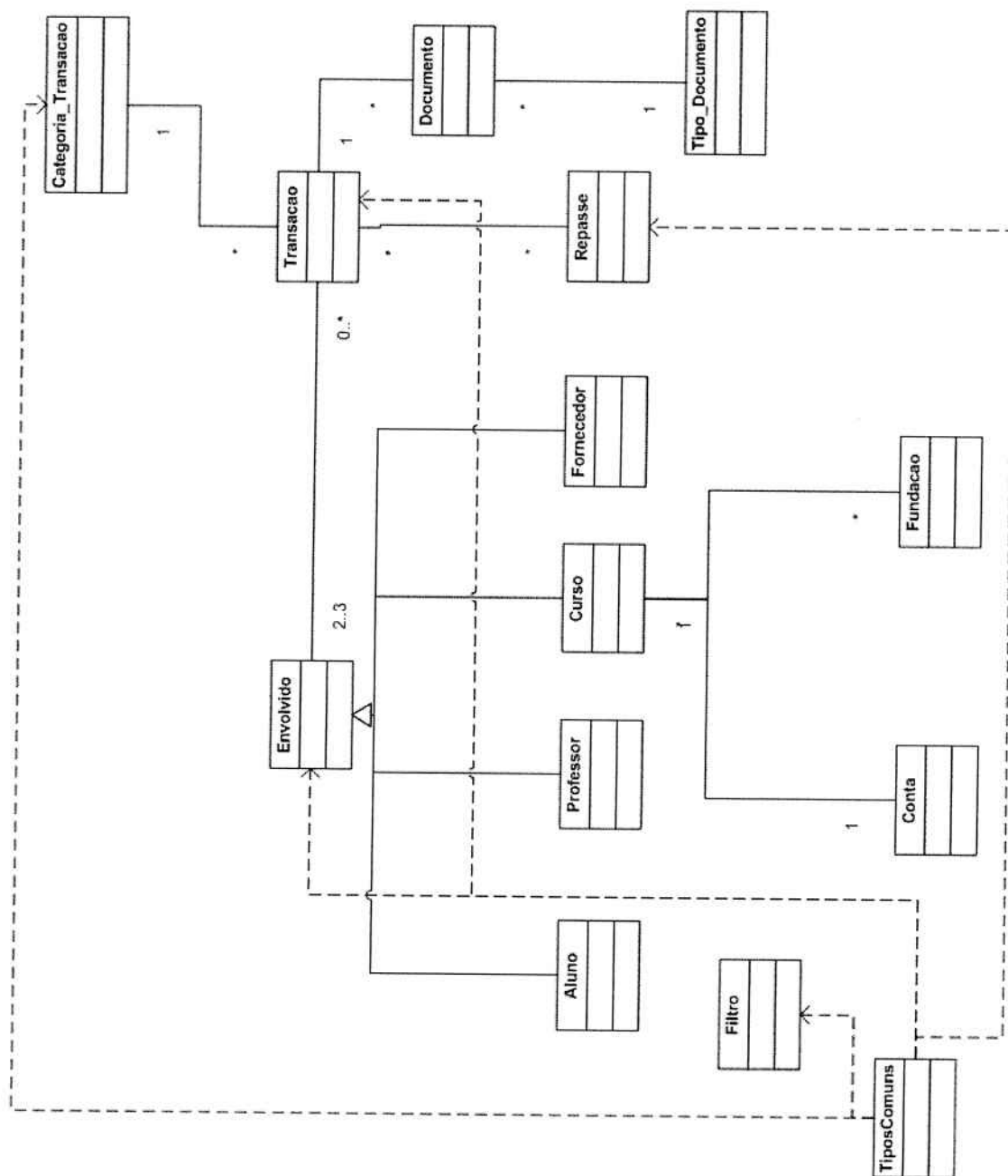
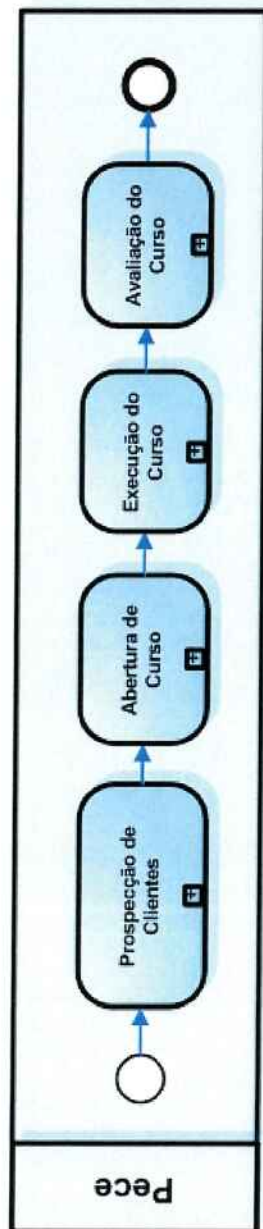


Figura 26 – Diagrama de classes




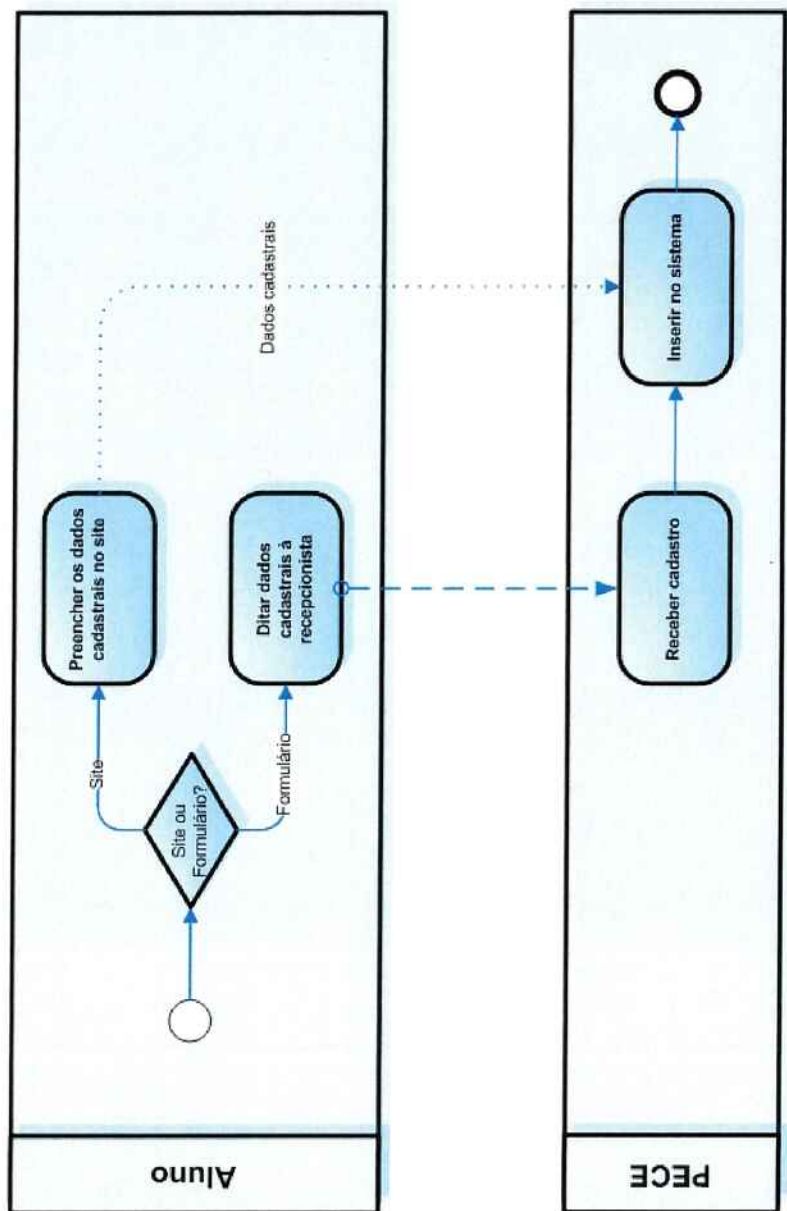
Produção de Curso - PECE	Autor:	Leonardo D. Dias	Revisor:	Matias Reclus	 <b>Escola Politécnica da USP</b> <b>PECE</b> Programa de Educação Continuada
	versão:	1.1	criado:	22/06/2006	
Área: Acadêmica e Financeira	status:	pr' validação	Modificado:	27/07/2006	

Figura 27 – Produção de curso





Cadastro para Propaganda - PECE		Autor: versão:	Márcia Rescius 1.0	Revisor: criado:	04/07/2006	<div>Escola Politécnica da USP</div> <div>PECE</div> <div>Programa de Educação Continuada</div>
Área: Acadêmica		status:	p/ validação	modificado		

Figura 29 – Cadastro para propaganda



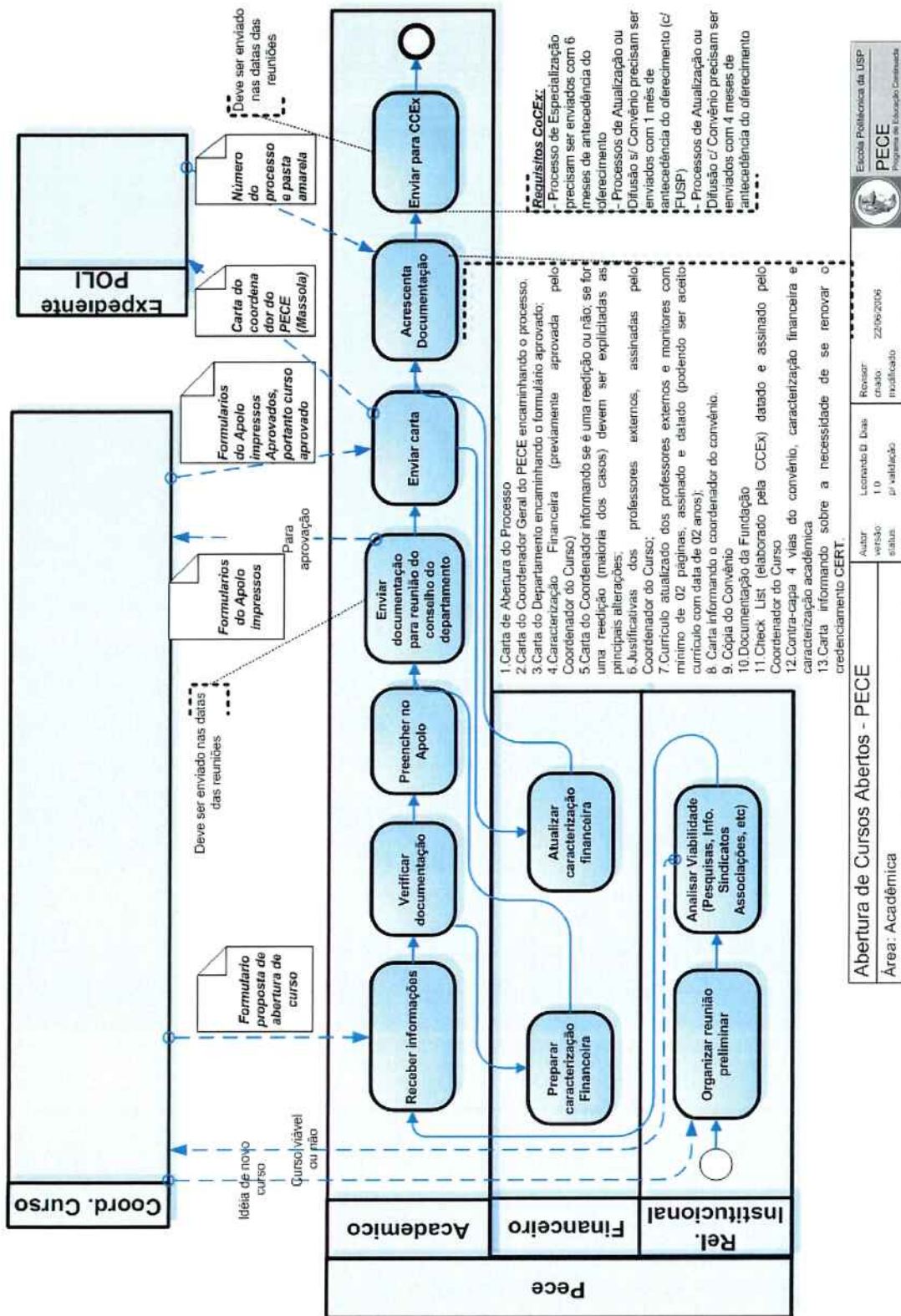


Figura 30 – Abertura de cursos abertos

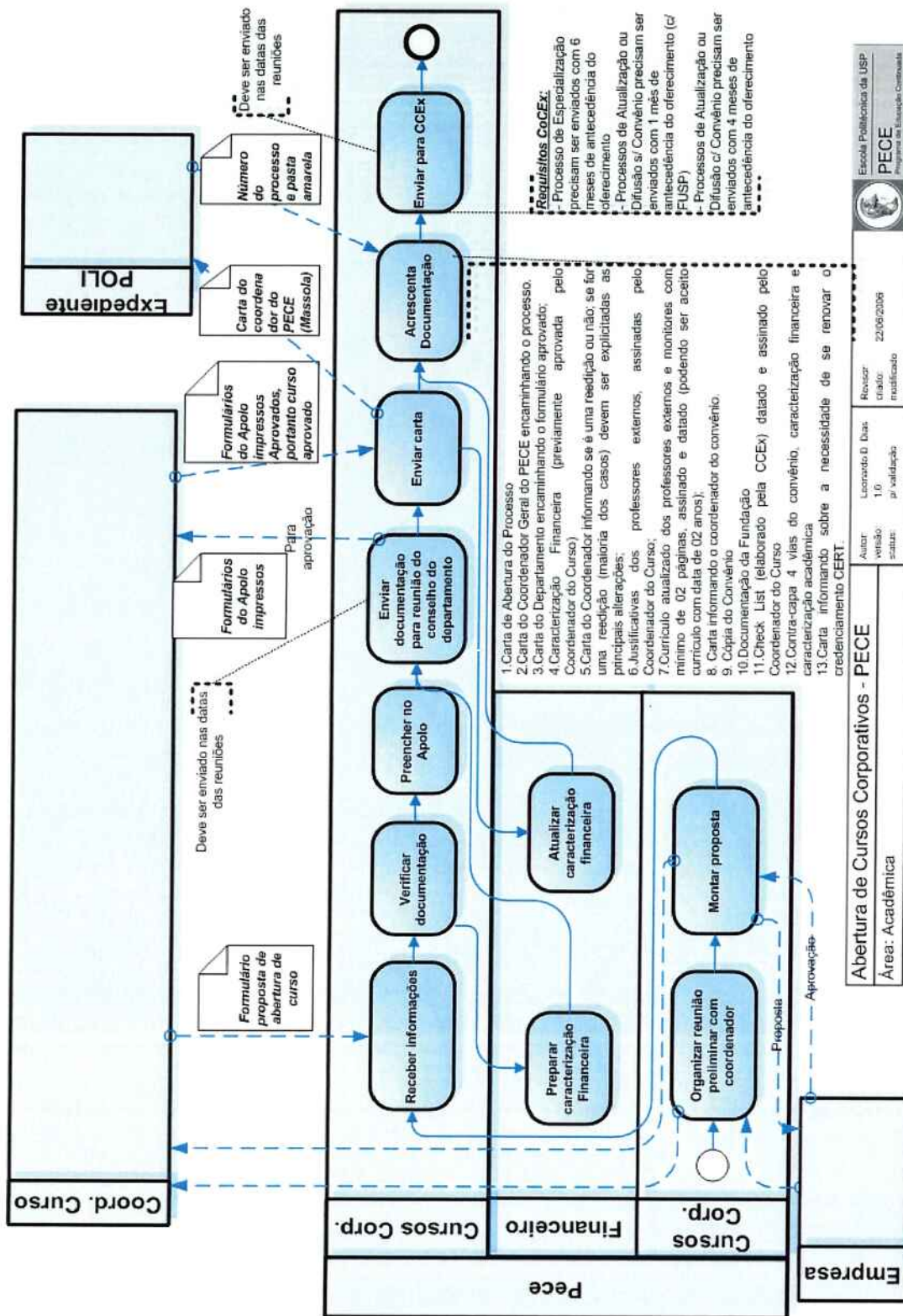
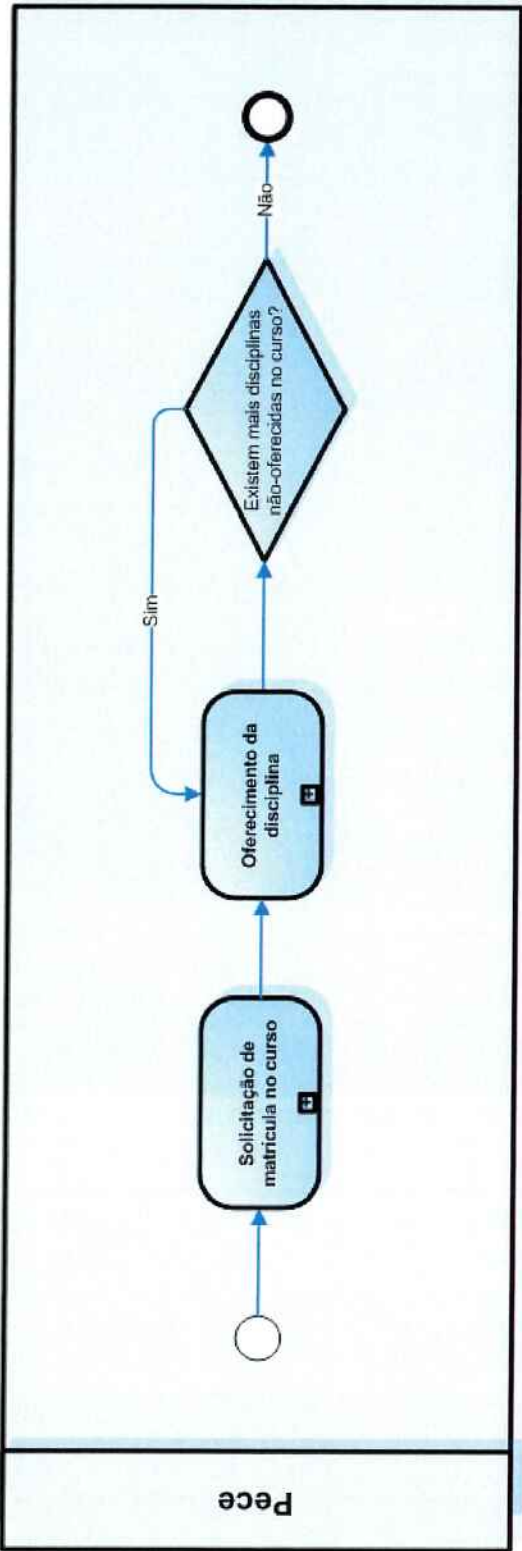
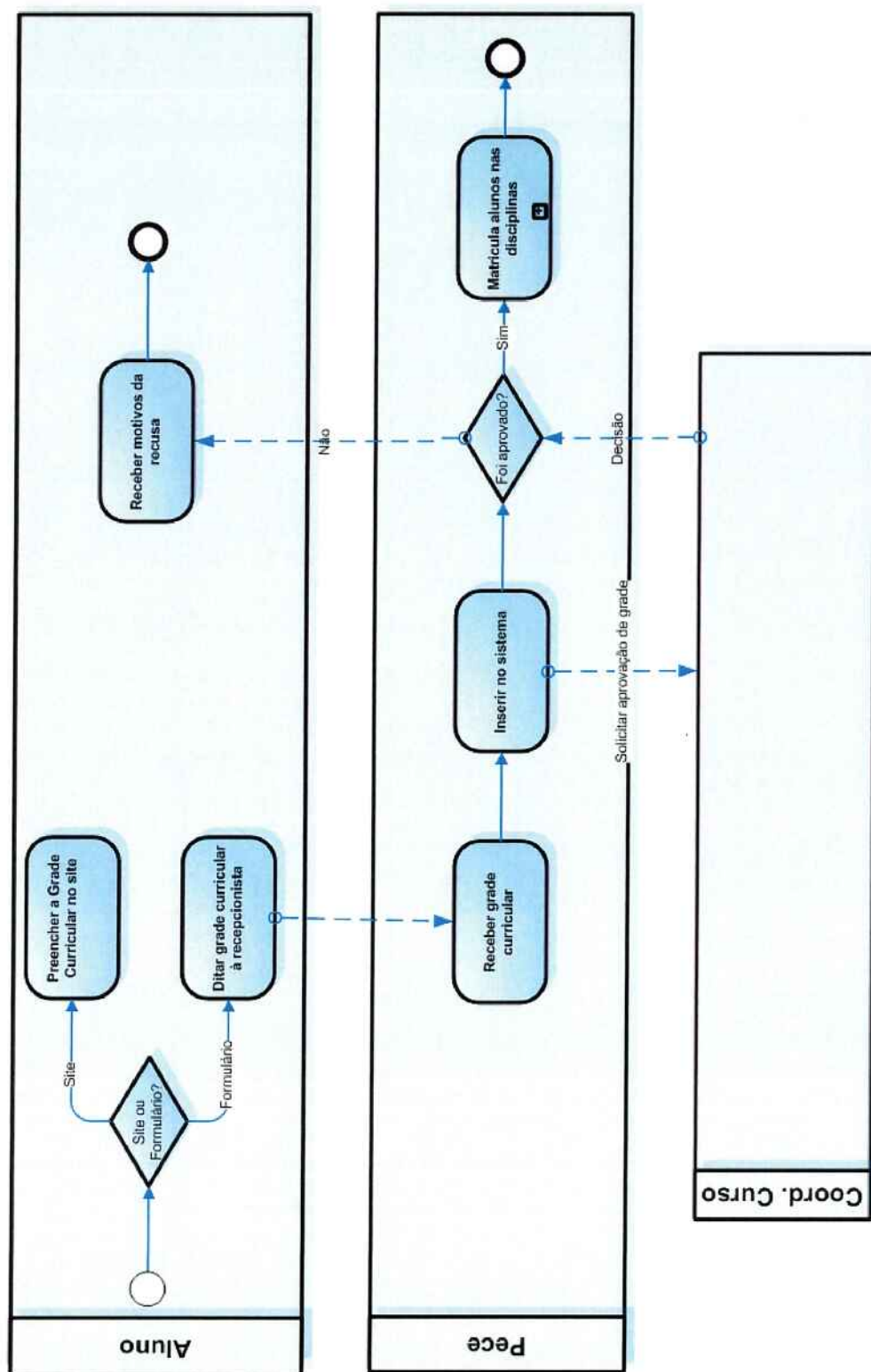


Figura 31 – Abertura de cursos corporativos



Execução do Curso - PECE		<div> <div> <div>Escola Politécnica da USP</div> <div>PECE</div> <div>Programa de Educação Continuada</div> </div> <div> </div> </div>	
Área: Acadêmica	<div> <div>Autor:</div> <div>Matias Reclus</div> </div> <div> <div>versão:</div> <div>1.0</div> </div> <div> <div>status:</div> <div>p/ validação</div> </div>	<div> <div>Revisor:</div> <div>criado:</div> <div>modificado:</div> </div> <div> <div>27/07/2006</div> </div>	<div> <div>Escola Politécnica da USP</div> <div>PECE</div> <div>Programa de Educação Continuada</div> </div> <div> </div>

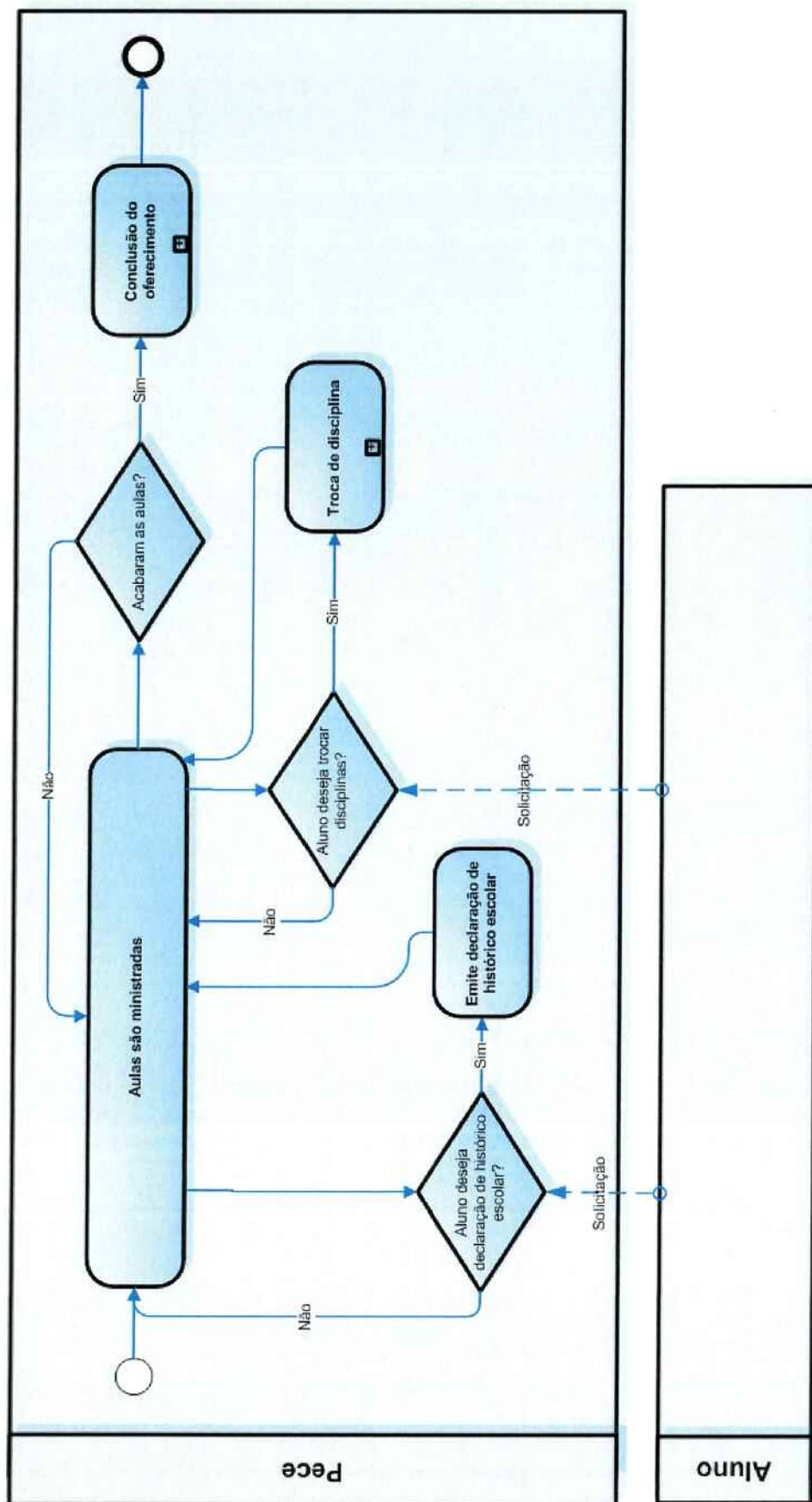
Figura 32 – Execução do curso



Solicitação de Matrícula no Curso - PECE		Autor: versão: status:	Mais Revisão 1.0 p/ validação	Revisor: criado: modificado:	04/07/2006
Área: Acadêmica					
		Escola Politécnica da USP			
		PECE			
		Programa de Educação Continuada			

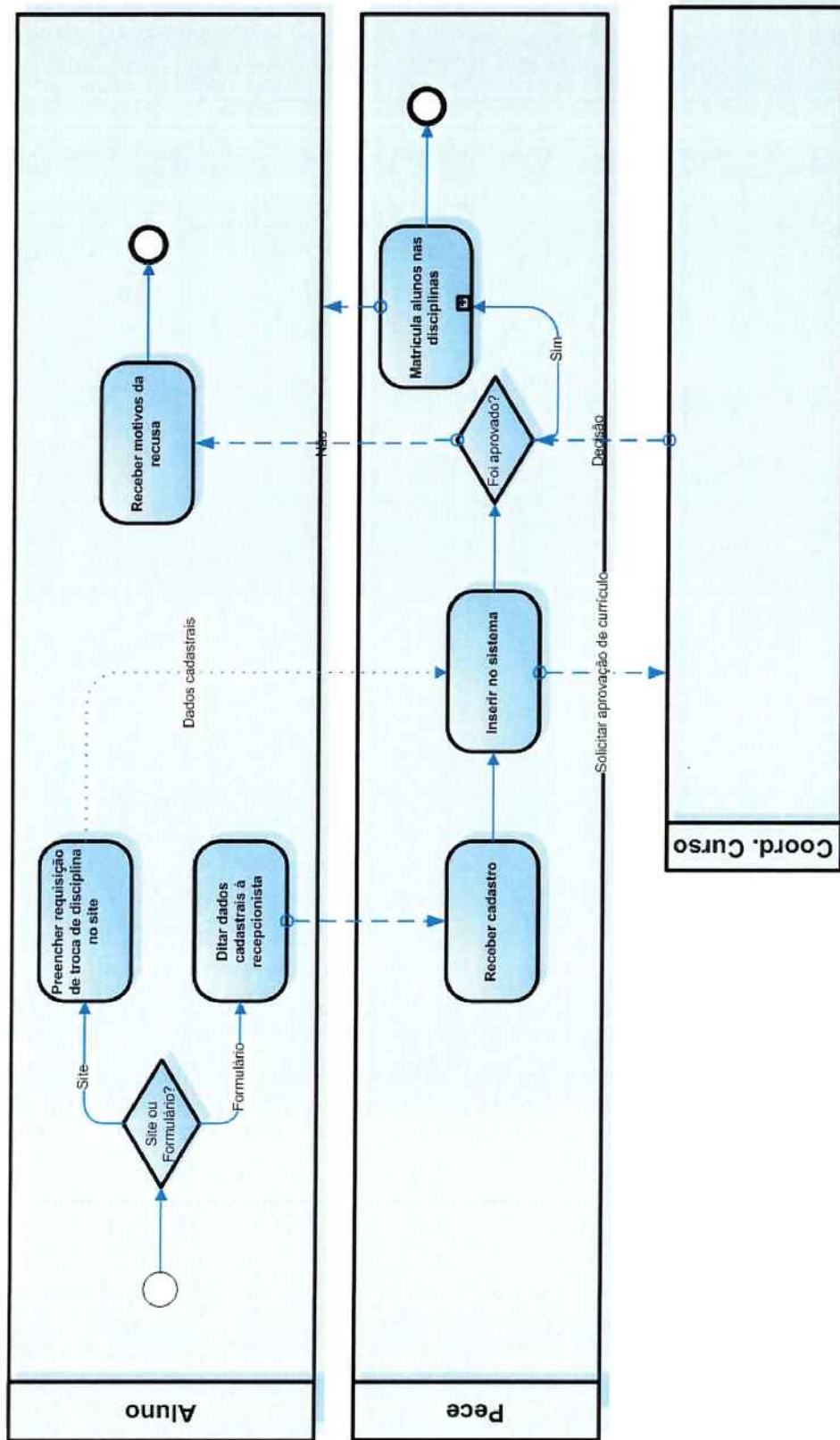
Figura 33 – Solicitação de matrícula no curso





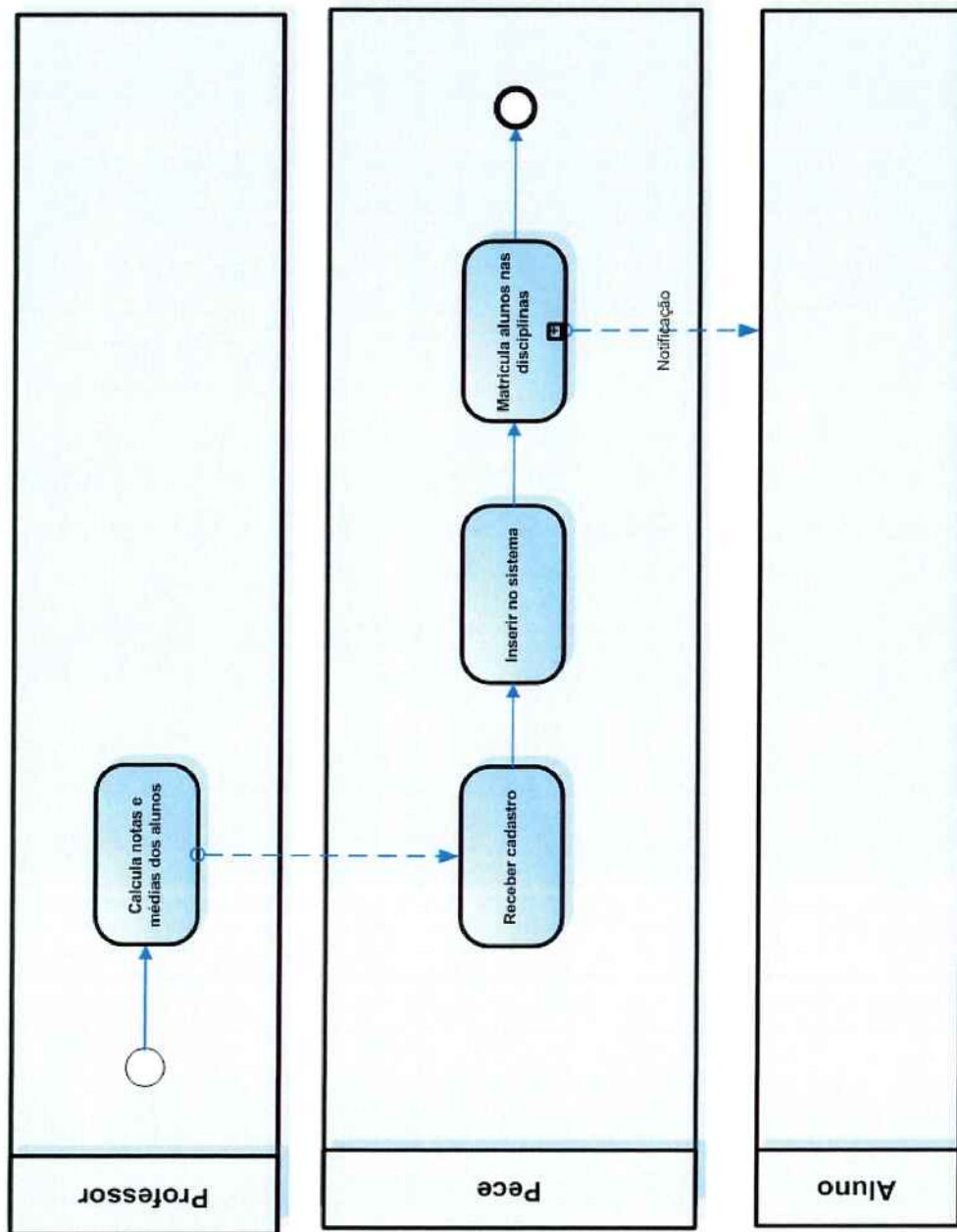
Oferecimento da Disciplina - PECE		Revisor: 27/07/2006		<div> <div>Escola Politécnica da USP</div> <div>PECE</div> <div>Programa de Educação Continuada</div> </div>
Área: Acadêmica		Manuais Revisão: 1.0	criado/modificado	

Figura 34 – Oferecimento da disciplina



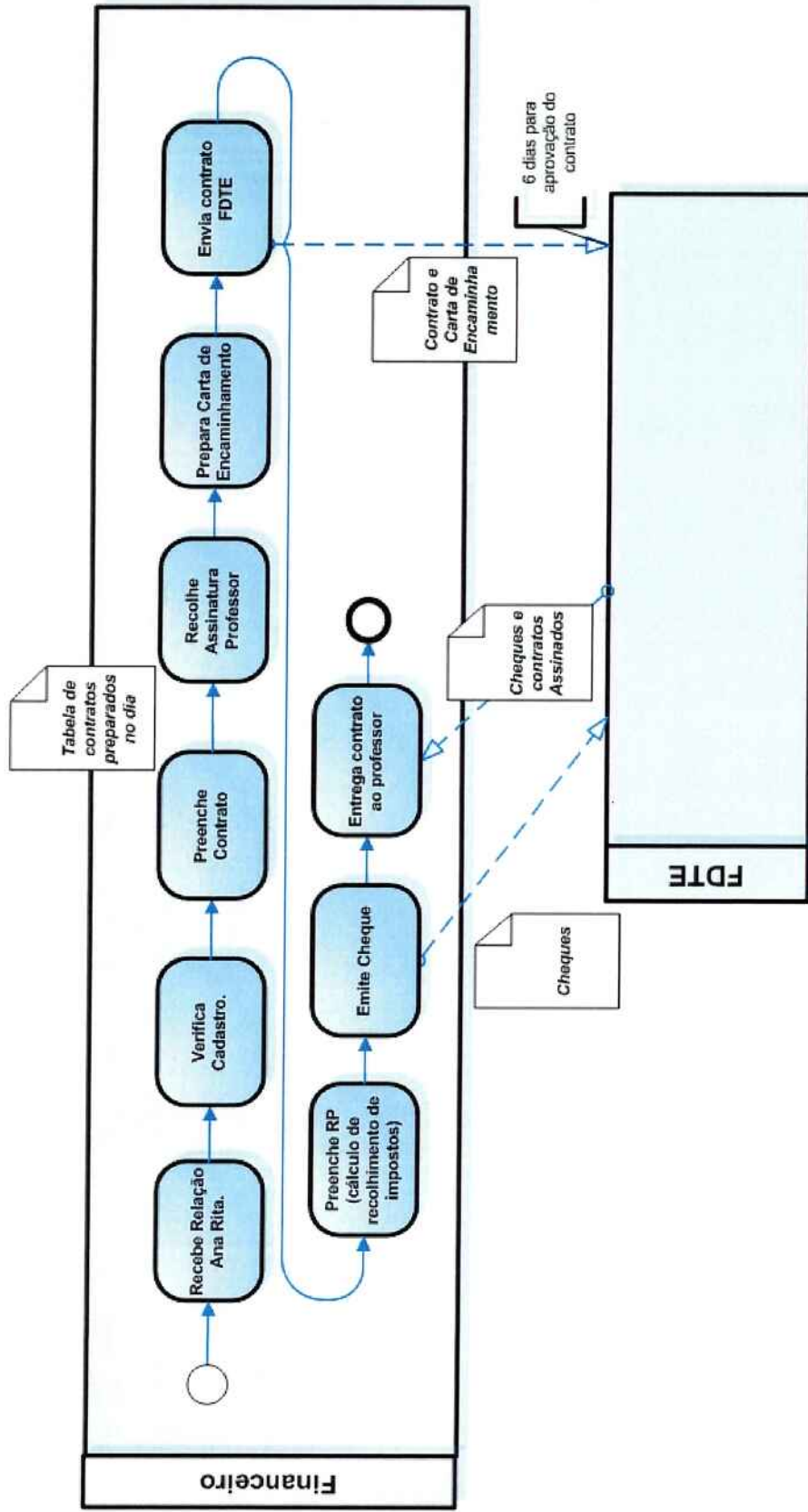
Troca de Disciplina - PECE		Autor: Felipe Martins	Revisor: Márcio Rencis	Escola Politécnica da USP PECE
Área: Acadêmica		versão: 1.1 status: p validação	criado: 04/07/2006 Modificado: 27/07/2006	

Figura 35 – Troca de disciplina



Conclusão do Oferecimento - PECE		Revisor:	27/07/2006	<div> <div>Escola Politécnica da USP</div> <div>PECE</div> <div>Programa de Educação Continuada</div> </div>
Área: Acadêmica		criado: status:	modificado	

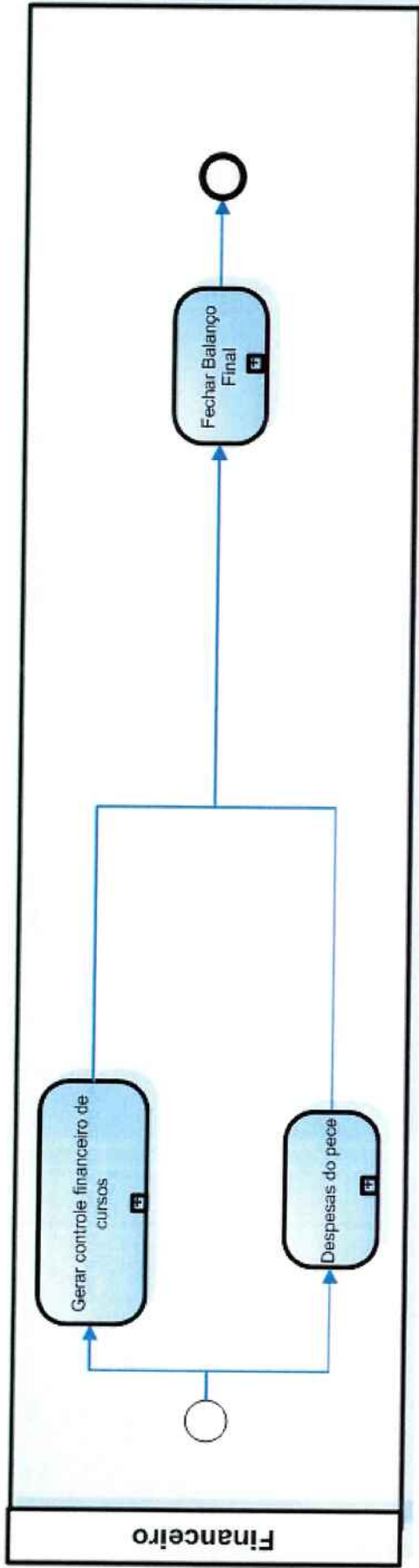
Figura 36 – Conclusão do oferecimento



Contrato e pagamento de professores - PECE		Autor: Paulo Elias	Revisor: Leonardo D. Dias	Escola Politécnica da USP	
Área: Financeiro		versão: 1.1	criado: 02/05/2006	PECE	
		status: pr validação	Modificado: 22/05/2006	Programa de Educação Continuada	

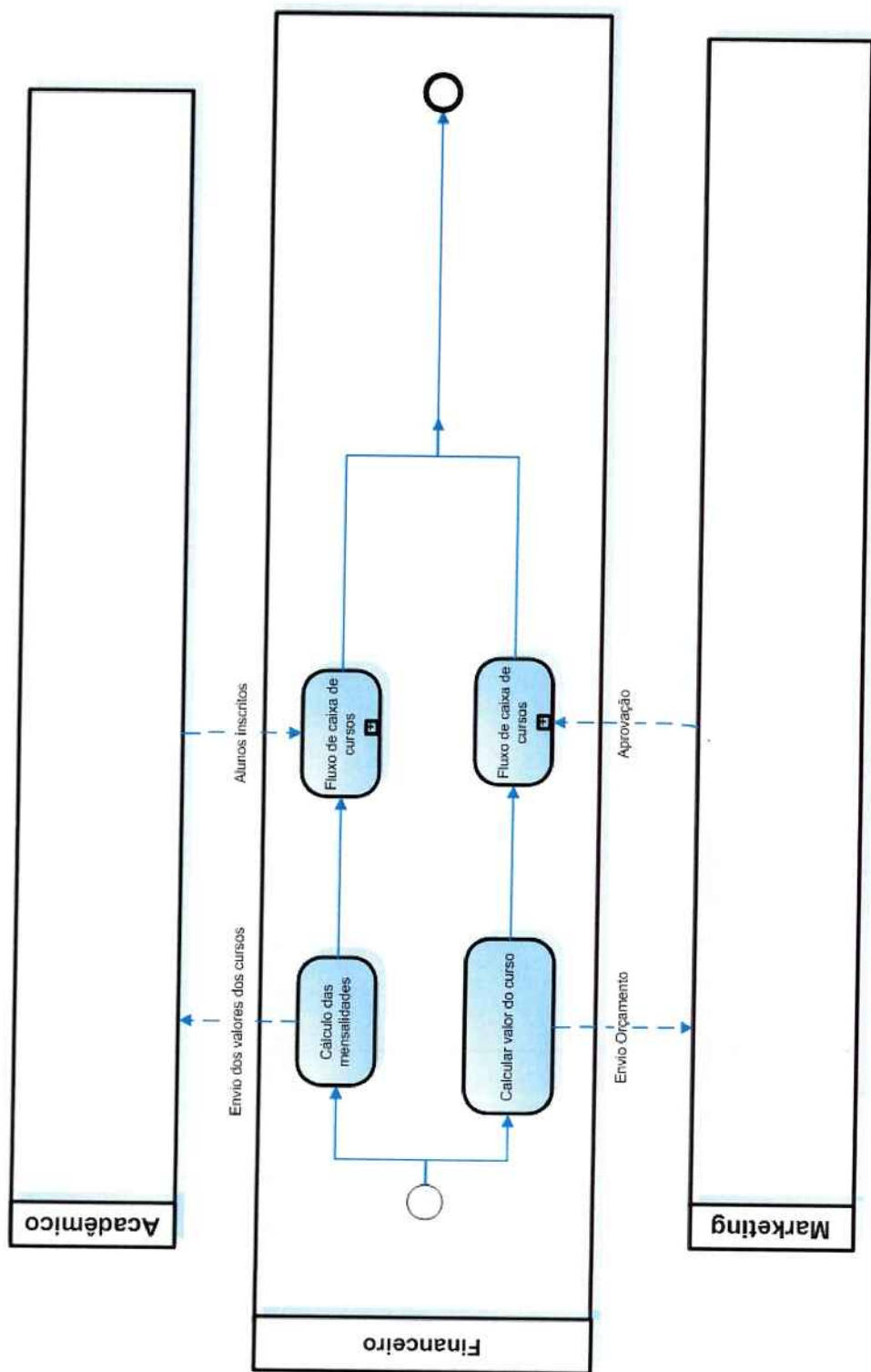
Figura 37 – Contrato e pagamento de professores





Financeiro - PECE		Escola Politécnica da USP	
Área - Financeira		PECE	
		Programa de Educação Continuada	
		Revisor:	
		criado: 27/07/2006	
		modificado:	
		Autor:	
		Fernando Raffani	
		versão: 1.0	
		status: p' validação	

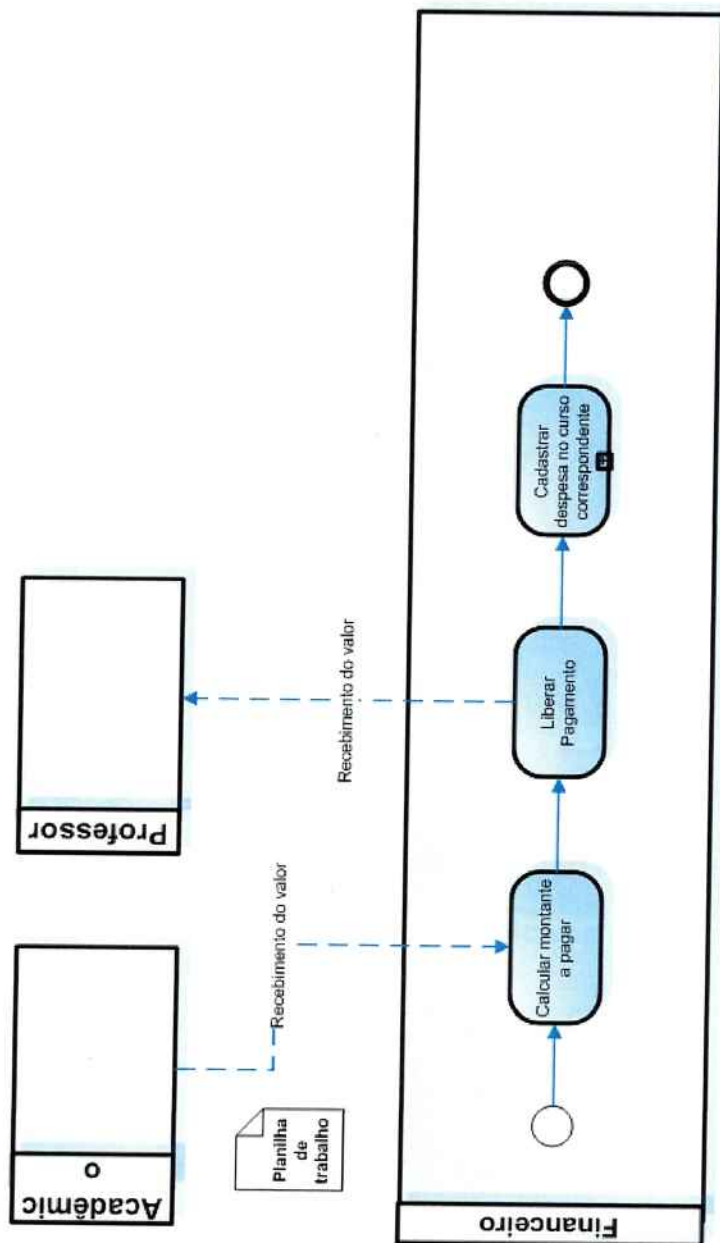
Figura 38 – Financeiro



Recebimento de Receitas - PECE		Autor:	Fernando Ruffini	Revisar:	criado:	27/07/2006	 <b>PECE</b> Programa de Educação Continuada
Área - Financeira		versão:	1.4	pré validação	criado:	modificação	

Figura 39 – Recebimento de receitas






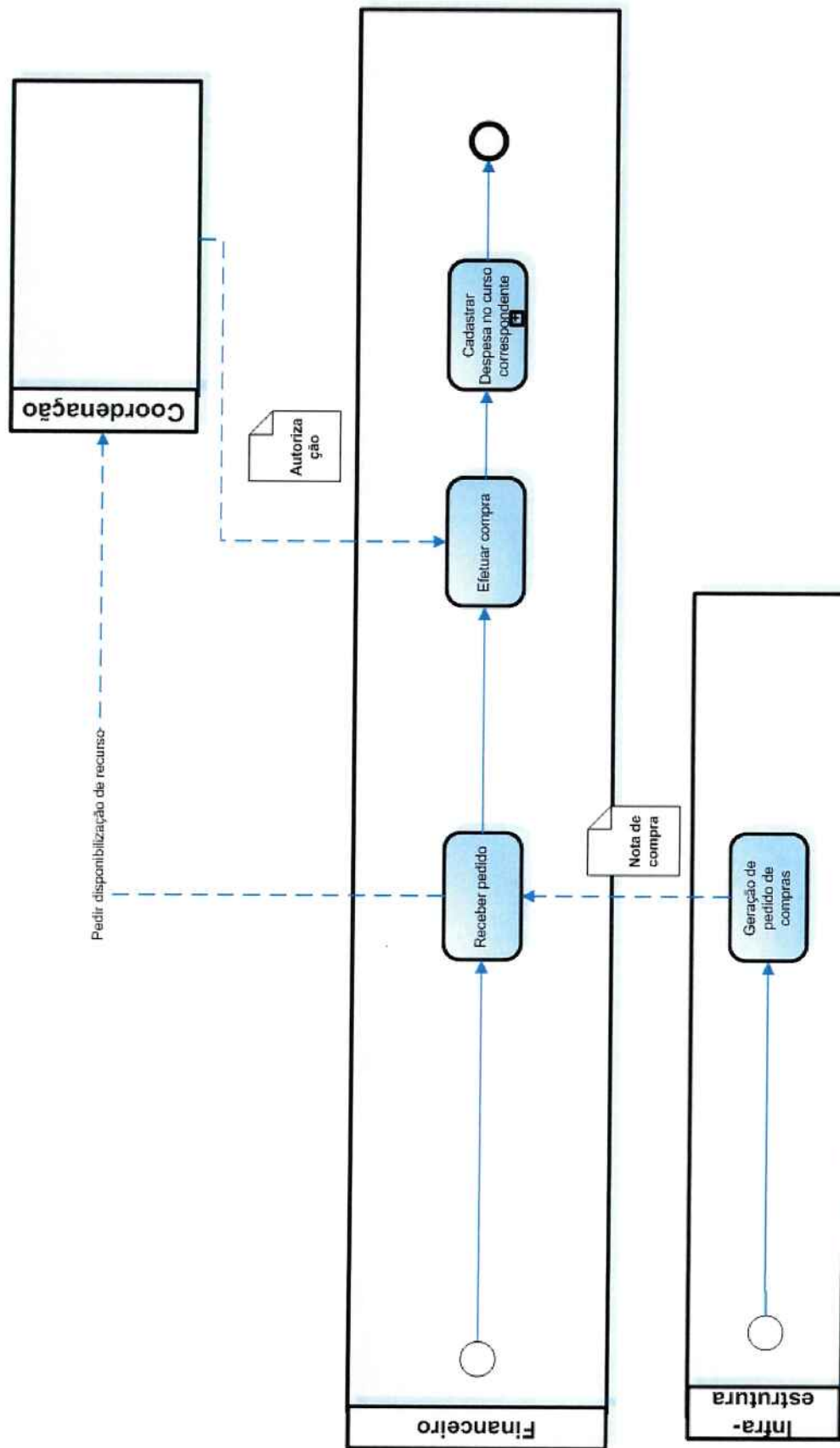
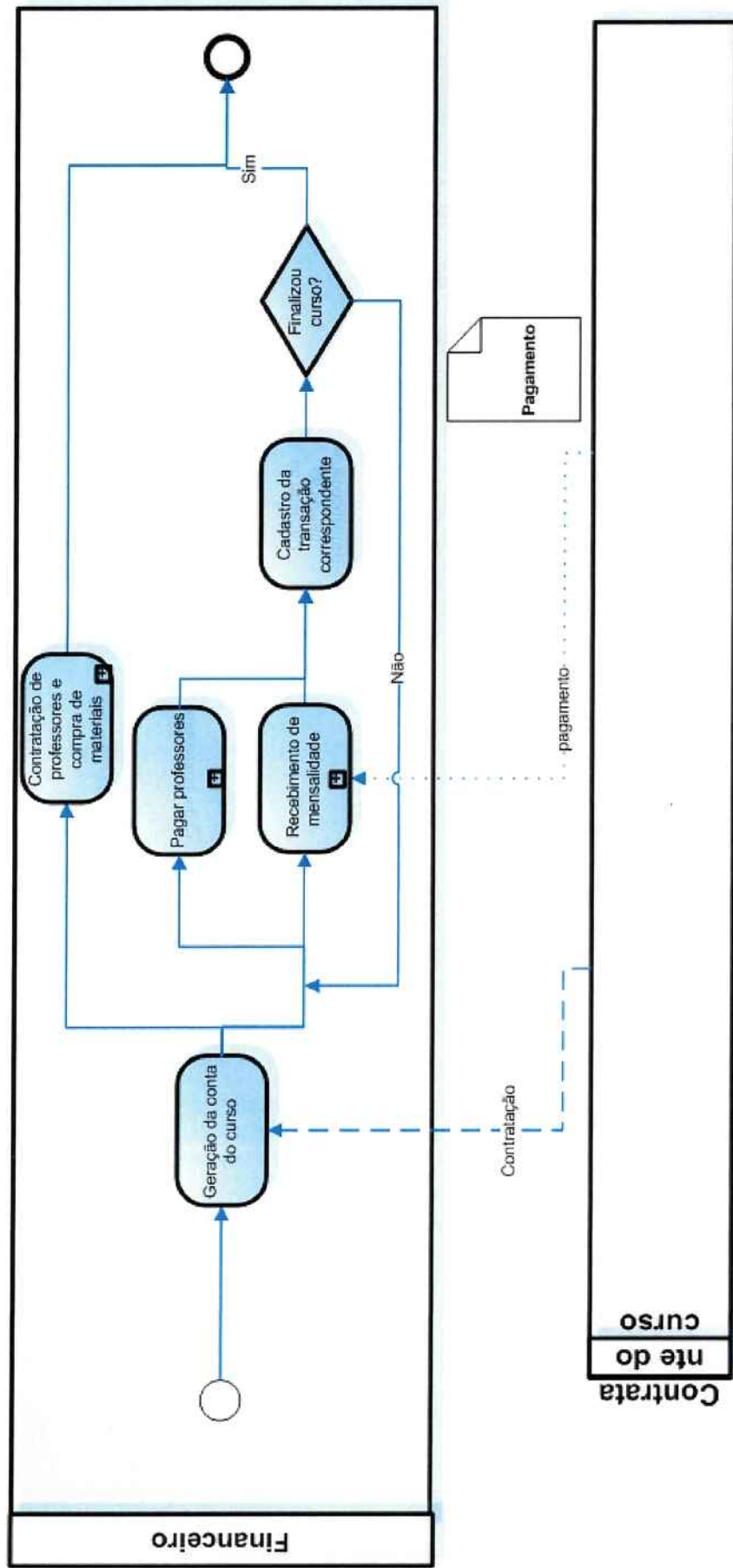
Pagamento de Professores - PECE		Autor: Fernando Raffalli	Revisor: 27/07/2006	 Escola Politécnica da USP PECE Programa de Educação Continuada
Área - Financelira		versão: 1.4 status: p/ validação	criado: modificado	

Figura 41 – Pagamento de professores



Compra de Materiais - PECE		<div> <div>Escola Politécnica da USP</div> <div>PECE</div> <div>Programa de Educação Continuada</div> </div>	<div> <div>Revisor:</div> <div>criado: 27/07/2006</div> <div>modificado:</div> </div>
Área - Financeira	<div> <div>Autor:</div> <div>versão: 1.4</div> <div>status: pr validação</div> </div>		

Figura 42 – Compra de materiais




Recebimento de cursos corporativos - PECE		Revisor:	27/07/2006	 Escola Politécnica da USP PECE Programa de Educação Continuada
Área - Financeira		autor:	1.4	
		versão:	1.4	
		status:	em validação	

Figura 43 – Recebimento de cursos corporativos

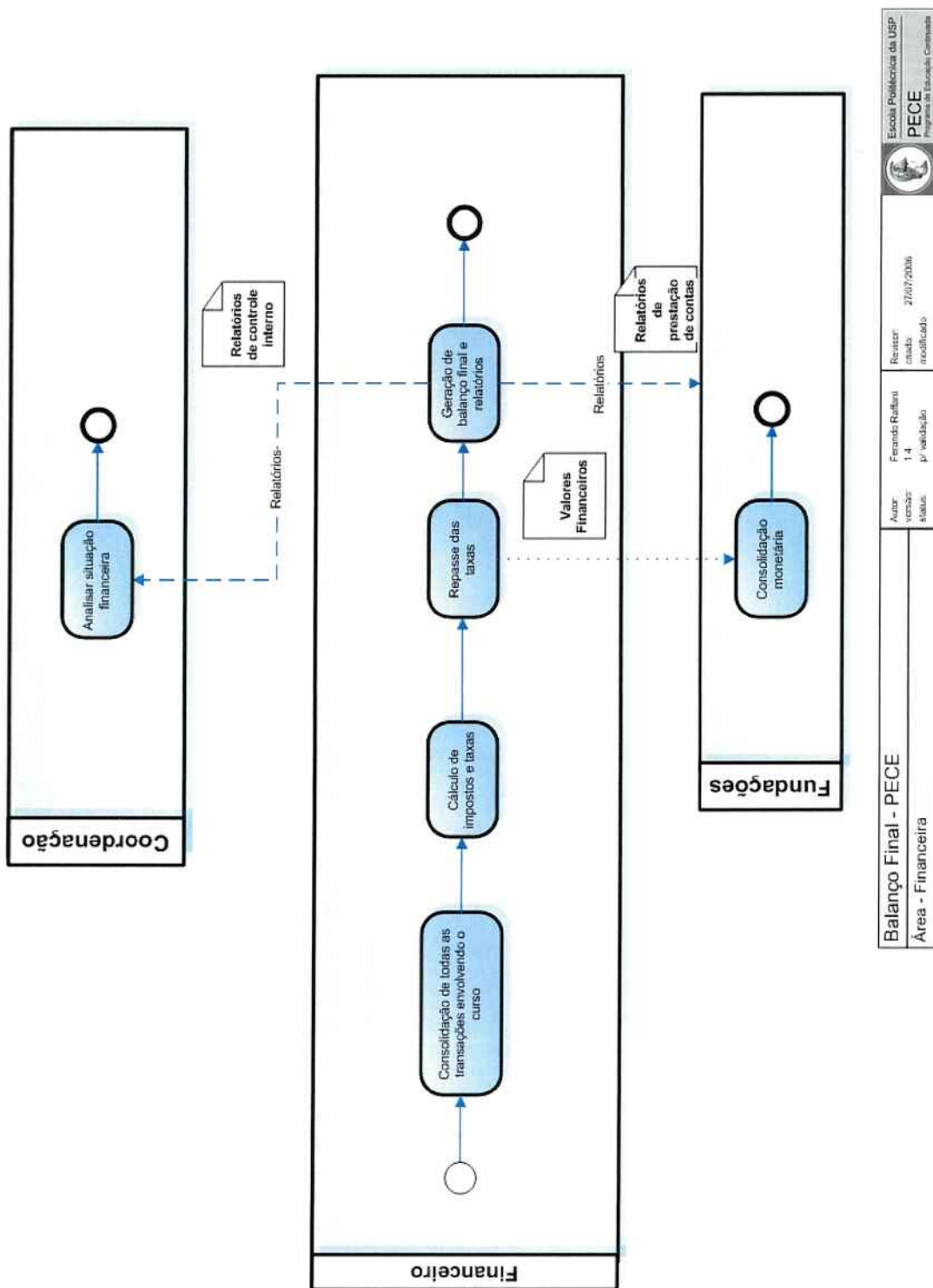


Figura 44 – Balanço final

Balanco Final - PECE		Autor: Fernando Rullani	Revisor: 27/07/2006	Escola Politécnica da USP PECE Programa de Engenharia de Computação
Área - Financeira		versão: 1.4 status: p/ validação	criado: modificado	



