



Universidade Federal
do Rio de Janeiro

Escola Politécnica

SIGECON

SISTEMA DE GERENCIAMENTO CONTÁBIL

Felipe Mayer Gonçalves

Projeto de Graduação apresentado ao Curso de Engenharia Eletrônica e de Computação da Escola Politécnica, Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Engenheiro.

Orientador: Sergio Palma da Justa Medeiros

Rio de Janeiro
Agosto de 2013

SIGECON
SISTEMA DE GERENCIAMENTO CONTÁBIL

Felipe Mayer Gonçalves

PROJETO DE GRADUAÇÃO SUBMETIDO AO CORPO DOCENTE DO CURSO DE ENGENHARIA ELETRÔNICA E DE COMPUTAÇÃO DA ESCOLA POLITÉCNICA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE ENGENHEIRO ELETRÔNICO E DE COMPUTAÇÃO.

Examinado por:

Prof. Sergio Palma da Justa Medeiros, Ph. D.

Prof. José Arthur da Rocha, M. Sc.

Prof. Sérgio Barbosa Villas-Boas, Ph. D.

RIO DE JANEIRO, RJ – BRASIL
AGOSTO DE 2013

Gonçalves, Felipe Mayer

SiGeCon: Sistema de Gerenciamento Contábil/ Felipe Mayer Gonçalves –Rio de Janeiro: UFRJ/ Escola Politécnica, 2013.

IX, 40p.: il.; 29,7 cm.

Orientador: Sergio Palma da Justa Medeiros

Projeto de Graduação – UFRJ/ Escola Politécnica/ Curso de Engenharia Eletrônica e de Computação, 2013.

Referência Bibliográfica: p. 38-39

1. Contabilidade. 2. COPPETEC. 3. UFRJ. 4. Sistema Web. 5. Pós-Graduação *Latu-Sensu*. I. Sergio Palma da Justa Medeiros II. Universidade Federal do Rio de Janeiro, Escola Politécnica, Curso de Engenharia Eletrônica e de Computação. III. Título.

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

Escola Politécnica – Departamento de Eletrônica e de Computação

Centro de Tecnologia, bloco H, sala H-217, Cidade Universitária

Rio de Janeiro – RJ CEP 21949-900

Este exemplar é de propriedade da Universidade Federal do Rio de Janeiro, que poderá incluí-lo em base de dados, armazenar em computador, microfilmear ou adotar qualquer forma de arquivamento.

É permitida a menção, reprodução parcial ou integral e a transmissão entre bibliotecas deste trabalho, sem modificação de seu texto, em qualquer meio que esteja ou venha a ser fixado, para pesquisa acadêmica, comentários e citações, desde que sem finalidade comercial e que seja feita a referência bibliográfica completa.

Os conceitos expressos neste trabalho são de responsabilidade do autor e do orientador.

Aos meus pais e meu irmão.

AGRADECIMENTO

Em primeiro lugar, agradeço à minha família, que sempre esteve do meu lado, me apoiando, me motivando, dando forças, além de proporcionar todas as oportunidades que eu tive. Meus pais me proporcionaram o melhor colégio onde eu poderia estudar, o Colégio Santo Inácio, que foi essencial para minha formação pessoal e profissional, além de me preparar muito bem para ser aprovado na Universidade Federal do Rio de Janeiro, no curso de Engenharia Eletrônica.

Ao meu irmão, Guilherme, que sempre foi o meu melhor amigo e companheiro. Esteve ao meu lado sempre que precisei, e também quando eu não precisava. Brigas entre irmãos sempre existem, mas era ótimo ver a total instabilidade entre brigas e risos.

À minha namorada, Sofia, que me deu forças e me cobrava a respeito do andamento do projeto. Tenho muito a agradecer a ela por toda a felicidade e crescimento proporcionados nesses últimos anos de faculdade.

Aos meus amigos e amigas da faculdade, em especial à Caroline Rivera, que me deu total apoio no desenvolvimento deste projeto, e aguentando meu desespero; e aos três Lucas, Noel, Leo e Arthur, grandes amigos e companheiros desta jornada.

Agradeço a todos os meus professores, tanto do CSI, quanto da UFRJ, que foram muito importantes para minha formação pessoal e profissional. Várias dicas e ensinamentos foram valiosíssimos, sempre!

Ao professor Sergio Palma, meu orientador, por apresentar toda a visão de negócios dentro da engenharia, e do uso de ferramentas de computação e eletrônica voltados para a administração.

Ao povo brasileiro e a todos os demais professores e funcionários da Universidade Federal do Rio de Janeiro que contribuíram para a minha formação. Este projeto é uma pequena forma de retribuir o investimento e confiança em mim depositados.

Resumo do Projeto de Graduação apresentado à Escola Politécnica/ UFRJ como parte dos requisitos necessários para a obtenção do grau de Engenheiro Eletrônico e de Computação.

SiGeCon
Sistema de Gerenciamento Contábil

Felipe Mayer Gonçalves

Agosto/2013

Orientador: Sergio Palma da Justa Medeiros

Curso: Engenharia Eletrônica e de Computação

A COPPETEC, fundação criada para apoiar a realização de projetos de desenvolvimento tecnológico, realiza diversos pagamentos e contabiliza recebimentos diariamente. Esses pagamentos são, muitas vezes, requeridos por meio de relatórios que cada coordenador de projeto envia à instituição, a qual providenciará a compra de produtos, ou o pagamento a algum serviço, ou professor. Essas providências correspondem aos pedidos que são preenchidos, manualmente, no programa Microsoft Word. Os registros armazenados correspondem a arquivos texto, não havendo, portanto, um sistema automatizado capaz de reunir e integrar os dados referentes aos gastos. As informações de receitas de cada projeto são disponibilizadas, mas não são apresentadas de maneira clara e acessível aos coordenadores.

A partir da observação das frágeis características observadas nesse processo de contabilização adotado na COPPETEC surgiu a possibilidade de desenvolvimento do SiGeCon, que é um sistema que possibilitará a reunião de algumas informações contábeis desses projetos e a automatização do processo de geração de pedidos de pagamento.

Palavras-Chave: Contabilidade, COPPETEC, UFRJ, Sistema Web, Pós-Graduação.

Abstract of Undergraduate Project presented to POLI/UFRJ as a partial fulfillment of the requirements for the degree of Engineer.

SiGeCon
Management Accounting System

Felipe Mayer Gonçalves

August/2013

Advisor: Sergio Palma da Justa Medeiros

Course: Electronic and Computer Engineering

The COPPETEC foundation, created to support the implementation of technological development projects, performs a variety of payments and receipts daily. These payments are often ordered through reports sent by each project coordinator to the institution, which will provide any product purchase, and any account or salary payment required. The order reports are filled manually, using Microsoft Word. The records are stored within the text files and there isn't any automatic system to gather and integrate data related to the expenses. The income informations for each project are available, but are not presented clearly to the coordinators.

Considering the observations of the fragility from this contabilization process, adopted by COPPETEC, emerged the possibility to develop the SiGeCon, witch correspond to a system that will be able to gather some accounting information, and will automate the process of generating payment requests.

Key-words: Accounting, COPPETEC, UFRJ, Web System, post graduation.

SIGLAS

SiGeCon	– Sistema de Gerenciamento Contábil
DEL	– Departamento de Engenharia Eletrônica
UFRJ	– Universidade Federal do Rio de Janeiro
COPPE	– Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia
COPPETEC	– Fundação Coordenação de Projetos, Pesquisas e Estudos Tecnológicos
CSS	– <i>Cascade Sytle Sheet</i>
HTML	– <i>Hypertext Markup Language</i>
IDE	– <i>Integrated Development Environment</i>
ORM	– <i>Object-Relational Mapper</i>
SQL	– <i>Structured Query Language</i>
UML	– <i>Unified Modeling Language</i>
URL	– <i>Uniform Resource Locator</i>
XML	– <i>Extensible Markup Language</i>
XLS	– <i>Excel Binary File Format</i>
RAP	– Relatório de Acompanhamento de Projeto
RPP	– Relatório de Pedido de Pagamento
CPF	– Cadastro de Pessoa Física
CNPJ	– Cadastro Nacional de Pessoa Jurídica

Sumário

INTRODUÇÃO	1
1.1 - TEMA	1
1.2 - DELIMITAÇÃO	1
1.3 - JUSTIFICATIVA	1
1.4 - OBJETIVOS	2
1.5 - METODOLOGIA	3
1.6 - DESCRIÇÃO	3
FUNDAMENTOS TEÓRICOS	5
2.1 - ENGENHARIA ELETRÔNICA E DE COMPUTAÇÃO	5
2.2 - PROGRAMAÇÃO	6
2.3 - ESTRUTURA DE DADOS	7
DEMANDAS E DESAFIOS	9
3.1 - CONTEXTUALIZAÇÃO	9
3.2 - PROBLEMAS	9
3.3 - DEMANDAS	10
PROJETO DO SISTEMA	12
4.1 - REQUISITOS DO SISTEMA	12
4.2 - CASOS DE USO	13
4.2.1 - CADASTROS	14
4.2.2 - RELATÓRIO DE PEDIDOS DE PAGAMENTO	20
4.2.3 - EXTRATO	21
4.3 - FERRAMENTAS & SOFTWARES	22
4.4 - IMPLEMENTAÇÃO E DESENVOLVIMENTO	25
4.4.1 - MODELAGEM DE DADOS & CLASSES	26
4.4.2 - DJANGO	27
4.4.3 - HTML & CSS	31
4.5 - TESTES DO SISTEMA	34
4.6 - PROPOSTAS FUTURAS	34
CONCLUSÕES	36
5.1 - AVALIAÇÃO DAS FERRAMENTAS	36
5.2 - APRENDIZADO	37
5.3 - CONSIDERAÇÕES FINAIS	37
BIBLIOGRAFIA	38
APÊNDICE - RELATÓRIO DE PEDIDO DE PAGAMENTO	40

Lista de Figuras

Figura 1: Tela de cadastro de despesas, com exemplo de cadastro.	15
Figura 2: Tela de cadastro de despesas padrão, mostrando a lista dos tipos de despesas possíveis.....	16
Figura 3: Tela de cadastro de receitas do projeto, com um exemplo de como aparecem os clientes.	16
Figura 4: Tela de cadastro de projetos, com exemplo.	17
Figura 5: Tela de Cadastro de MBA.....	17
Figura 6: Tela de cadastro de agências bancárias.....	18
Figura 7: Tela de cadastro de pessoas físicas e jurídicas, com exemplo de cadastro de pessoa física.....	18
Figura 8: Tela de cadastro de pessoas físicas e jurídicas, com exemplo de pessoa jurídica.....	19
Figura 9: Tela de cadastro de usuários.	19
Figura 10: Tela de edição de usuários.	20
Figura 11: Tela de emissão de segunda via de despesas.	21
Figura 12: Tela de visualização de extratos.....	22
Figura 13: Imagem do Microsoft Excel exibindo o extrato emitido em XLS.....	22
Figura 14: Imagem do aplicativo XCode, utilizado no desenvolvimento do projeto, exibindo o código da página padrão.	23
Figura 15: Imagem da página ADMIN do Django, utilizada para modificações diretas no banco de dados.....	24
Figura 16: Modelo do banco de dados gerado no MySQLWorkbench.....	26
Figura 17: Sistema exibindo lista com todos os bancos do Brasil.....	30
Figura 18: Sistema exibindo lista com os quatro tipos de despesa padrão.....	30
Figura 19: Sistema exibindo lista com os três tipos de pagamento de despesas.	31
Figura 20: Sistema exibindo lista com os Estados do Brasil.....	31
Figura 21: Sistema na página inicial de extrato. Pode-se visualizar o padrão de tela sob todas páginas do sistema.....	32
Figura 22: Código do arquivo ‘ <i>default.py</i> ’, <i>template</i> de todas as páginas do sistema....	33

Lista de Tabelas

Tabela 1: Descrições dos casos de uso; e os respectivos casos de uso dos quais são dependentes.	13
--	----

Capítulo 1

Introdução

1.1 – Tema

O tema do trabalho consiste no estudo e implementação de um sistema de gerenciamento contábil. A partir do desenvolvimento desse sistema pretende-se automatizar, agilizar e integrar o controle de despesas e faturamento da área responsável por ministrar cursos específicos em uma instituição de ensino de nível superior.

1.2 – Delimitação

O foco do Projeto é o desenvolvimento de um software para ambiente Web que consiga gerenciar de forma simples e automática as despesas e receitas de cursos de Pós-Graduação *Latu-Sensu* da UFRJ. Além do controle financeiro, pretende-se também que o sistema seja capaz de gerar relatórios para o acompanhamento de lançamentos passados e futuros para auxiliar no planejamento financeiro dos cursos mencionados.

O projeto será limitado à UFRJ e aos departamentos que trocam informações e relatórios com a COPPETEC envolvendo movimentações financeiras. O produto aqui apresentado foi desenvolvido como sendo um sistema web, tendo como características principais uma fácil visualização e usabilidade por parte dos responsáveis pela administração dos cursos de Pós-Graduação.

1.3 – Justificativa

Atualmente, todo o trabalho relacionado à contabilidade dos cursos de Pós-Graduação é feito manualmente. A justificativa para o desenvolvimento deste Projeto de Graduação é implementar um sistema que automatize e integre toda a movimentação financeira dessa área do Departamento de Eletrônica.

O diálogo com a Fundação COPPETEC é feito através de relatórios em papel, trocados entre a Fundação e o departamento responsável pelos projetos. O primeiro tipo de relatório é referente a despesas, representado por pedidos de pagamento de despesas necessárias ao desenvolvimento dos projetos. Esses relatórios são preparados manualmente através do preenchimento de formulários, os quais são impressos, assinados pelo responsável dos cursos e, então, enviados à COPPETEC, que efetua o pagamento.

Os relatórios enviados pela Fundação para o departamento são semelhantes a extratos bancários. Esses documentos apresentam histórico de pagamentos, de recebimentos e o saldo disponível referente a um determinado projeto.

O fato de todo o processo ser manual implica em dificuldades no desenvolvimento de novas atividades e na criação de novos cursos. O tempo demandado poderia ser reduzido drasticamente com a implementação de um sistema que armazenasse as informações referentes a despesas e receitas, e que gerasse os relatórios a serem enviados para a Fundação. Outra facilidade prevista com a adoção desse sistema seria o melhor planejamento e aprimoramento dos cursos.

Além das dificuldades mencionadas, há várias outras provenientes do canal de comunicação existente com a COPPETEC, o qual não só é primitivo e limitado, como também, ineficiente. Pretende-se desenvolver esse sistema para a redução dessas dificuldades e para a ampliação da análise contábil dos cursos de Pós-Graduação do DEL/UFRJ.

1.4 – Objetivos

O objetivo geral do Projeto é desenvolver um software de gestão contábil que atenda às necessidades dos responsáveis pelos cursos de Pós-Graduação Lato-Sensu do DEL/UFRJ.

Os objetivos específicos consistem no desenvolvimento de um sistema Web (1), que seja seguro (2), que proporcione uma melhor gestão de despesas (3) e receitas (4), que seja capaz de gerar relatórios que atendam às necessidades dos responsáveis pelos cursos (5) e que sirvam para a formalização de pedidos de pagamento de despesas para a COPPETEC (6).

1.5 – Metodologia

O trabalho ora apresentado seguirá uma série de práticas e técnicas de desenvolvimento estudadas na Engenharia de Software. Para a implementação do projeto será utilizado o conceito de Programação Orientada a Objetos. A partir da adoção desse conceito, o sistema será modelado utilizando-se UML – Unified Modeling Language.

O processo de desenvolvimento do software pretendido seguirá uma estratégia incremental. Para dar continuidade a esse desenvolvimento, outras etapas são importantes para suprir as demais necessidades relacionadas a administração dos cursos de Pós-Graduação, que não serão implementadas nesta fase do projeto. O processo será incremental, pois pretende-se, inicialmente, implementar um sistema simples que atenda a um número mínimo de necessidades e, de acordo com o andamento do projeto, será estudada a implementação de novas funcionalidades. Cada parte do sistema deverá ser implementada, testada, demonstrada aos clientes e, após sua aprovação, implantada no sistema, gerando uma nova versão.

Considerando que este Projeto de Graduação corresponde à primeira versão de um sistema que poderá ter utilização comercial, e que haverá a necessidade de manutenção e aprimoramento do sistema, deverá ser produzida uma documentação bem detalhada de seu código. Essa documentação facilitará o trabalho de outros desenvolvedores que se propuserem a dar manutenção e/ou continuidade na expansão do Sistema de Gerenciamento Contábil dos cursos de Pós-Graduação.

1.6 – Descrição

No Capítulo 2 (dois) serão discutidos os fundamentos teóricos por trás do projeto. Serão apresentados os conhecimentos adquiridos ao longo do curso de Engenharia Eletrônica e de Computação, os quais foram aplicados durante todo o desenvolvimento do trabalho. Alguns aspectos da programação implementada e da estrutura de dados referentes ao projeto terão suas características específicas explicadas.

O Capítulo 3 (três) apresentará o contexto no qual o trabalho está inserido. Serão explicitados a realidade, os problemas e as demandas do Projeto de Pós-Graduação do Departamento de Eletrônica, assim como as necessidades das secretárias e dos coordenadores responsáveis.

Sendo considerados os problemas e definidas as demandas, no capítulo 4 será definido o escopo do projeto, apresentadas as ferramentas e softwares utilizados e serão explicitados os requisitos do sistema. Tendo todas as informações referentes ao projeto, e as ferramentas a serem utilizadas, bem definidas, serão relatadas as etapas de implementação e desenvolvimento do Projeto, além dos testes necessários para comprovar a confiabilidade do sistema. Por último, serão descritas as próximas etapas para algum possível desenvolvedor que queira incrementar as funcionalidades do projeto.

Finalmente, no Capítulo 5 será apresentada a conclusão do trabalho, onde serão realizadas avaliações a respeito do projeto, das ferramentas utilizadas, além de todo o aprendizado por trás do desenvolvimento do projeto. Por último, serão feitas algumas considerações finais.

Capítulo 2

Fundamentos Teóricos

2.1 – Engenharia Eletrônica e de Computação

Durante o curso de Engenharia Eletrônica e de Computação foram ministradas diversas disciplinas, tanto da área de eletrônica quanto da de computação, embora o foco deste trabalho esteja voltado para a área de computação. Nesta seção serão apontadas as matérias que foram fundamentais para a formação do conhecimento necessário ao desenvolvimento do Projeto de Graduação. Os créditos serão apresentados em ordem cronológica e sua importância será explicada em cada parágrafo.

A primeira disciplina cursada, da área de computação, corresponde a Computação I, que constituiu o primeiro contato com os conceitos de lógica de programação. A linguagem inicialmente ministrada foi a linguagem Pascal, que tem como uma das principais características facilitar o aprendizado e o treino na aplicação de algoritmos.

Em seguida foi cursada a disciplina Computação II, na qual foi estudada a linguagem C ANSI, com a utilização do ambiente UNIX. O aprendizado desse tipo de sistema operacional foi fundamental na elaboração do sistema, uma vez que o Projeto de Graduação foi previsto para funcionar nesse tipo de ambiente de desenvolvimento. Nessa disciplina, o treinamento no uso de algoritmos e estruturas de dados recebeu maior ênfase. A terceira matéria abordou mais profundamente esse último tipo de treino, visto que os créditos eram referentes a Algoritmos e Estruturas de Dados.

A quarta matéria cursada foi a de Linguagens de Programação, na qual foi desenvolvido o conhecimento a respeito de Orientação a Objetos utilizando C++. A orientação a objetos foi muito utilizada no desenvolvimento deste Projeto que contava com diferentes classes separadas por tipo de cadastro e de informação armazenada. Essas classes serviram de base para a geração do banco de dados e sua estrutura, por meio do *Framework Django*, que gerenciava o banco.

Por último, foi cursada a disciplina de Banco de Dados. O professor ensinava e priorizava a qualidade da modelagem do banco de dados, para manter uma alta

qualidade do mesmo, a fim de evitar problemas no futuro. Embora a base de dados não tenha sido gerada por meio de *Queries* e, portanto, não tendo sido necessário o estudo de um banco de dados específico, essa matéria foi muito importante para a modelagem das classes, que se assemelha muito ao modelo do banco.

2.2 – Programação

O desenvolvimento de sistemas, principalmente aqueles voltados para a *Web*, tem sido realizado por meio de *Frameworks*, que facilitam e agilizam o desenvolvimento de um projeto. Esses *Frameworks* são ferramentas que facilitam a programação e se utilizam de classes, bibliotecas e funções genéricas, comuns no desenvolvimento de sistemas, e que tem como base alguma linguagem de programação, servindo de plataforma.

O atual Projeto foi desenvolvido no *Framework Django*, que utiliza linguagem *Python*. Como referência para o aprendizado de ambos, tanto da linguagem, quanto da plataforma, foi utilizado o livro *Python e Django* (Santana & Galesi, 2010). Além do livro, foi muito utilizada a página *Web* de referência do próprio *Django Project* (Django Software Foundation, 2013) que provê tutoriais e guias, além de fóruns para discussão.

Django é um *Framework Open Source* para o desenvolvimento de aplicações *Web*, escrito e baseado em *Python*. Permite o desenvolvimento rápido de aplicações *web* de “alto desempenho e elegância” (Django Project, 2013). A ferramenta é baseada no conceito *DRY (Don't repeat yourself)*, definido como “*Every piece of knowledge must have a single, unambiguous, authoritative representation within a system.*” (Hunt & Thomas, 1999), ou seja, cada conhecimento deve ser representado sem ambiguidade, de forma singular no sistema.

Python é uma linguagem interpretada e dinâmica (Python Software Foundation, 2013), isto significa que não é necessário um compilador que gere o código binário. Sua execução é feita por meio de um software que interpreta os comandos. Essa característica faz dos aplicativos desenvolvidos em *Python* serem altamente portáveis para diversas máquinas, sendo necessário, apenas, o interpretador. Essa é uma linguagem de alto nível (de abstração), sendo mais próximo à linguagem humana e mais distante da de máquina.

Durante todo o processo de desenvolvimento e teste, houve vários problemas e dúvidas que, na maioria, foram resolvidos com buscas na internet. Um portal muito utilizado como referência para a resolução desses problemas durante todo o processo, desde a instalação dos componentes até as últimas etapas de programação, foi o *Stackoverflow* (Stackoverflow, 2013).

O banco de dados utilizado foi o MySQL, que possui fácil integração com a plataforma do *Django*. Embora tenha utilizado o banco de dados, não foi necessário profundos estudos a respeito do mesmo. O conhecimento necessário para a adoção de um banco de dados, para qualquer sistema, pode ser resumido em modelagem de dados e sintaxe da linguagem específica. Tendo cursado a disciplina Banco de Dados e sem precisar escrever as *Queries*, a facilidade para utilização foi decisiva na escolha do mesmo.

Nas atividades de desenvolvimento de software, uma ferramenta muito importante é a de controle de versões. Esse tipo de ferramenta é utilizada para evitar problemas que podem surgir durante as fases de alterações do sistema. Na ocorrência de alterações que prejudiquem muito um sistema, essas ferramentas permitem que todos os arquivos voltem a estados anteriores. Em projetos que envolvem múltiplos desenvolvedores, esse tipo de ferramenta também facilita o desenvolvimento simultâneo.

Em desenvolvimento de projetos de programação, ferramentas de controle de versões são muito importantes caso haja a necessidade de desfazer alguma alteração que venha a gerar erros no sistema. Para tal, foi utilizado o GIT (Git Hub, 2013), ferramenta que possui diversos tutoriais na Web. Para interagir por meio de uma interface gráfica, foi utilizado um software para Mac OSX, chamado *SourceTree* (Atlassian, 2013). Com esse aplicativo não há a necessidade de escrita de linhas de comando que consomem muito tempo desnecessariamente.

2.3 – Estrutura de Dados

A estrutura de dados foi implementada utilizando-se o *Framework Django*, que faz Mapeamento Objeto Relacional do sistema, atuando como um Sistema Gerenciador de Banco de Dados. Nessa plataforma, são criadas as classes, chamadas “*models*”. As classes são criadas, como ‘filhas’ (herança) da classe “*models.Model*”. São, então,

definidos e especificados os atributos de cada classe, assim como o tipo de campo correspondente do banco de dados. Nas classes também são especificadas as conexões entre classes, através de “*ForeignKeys*”.

O MySQL, utilizado no sistema, é um Banco de Dados Relacional, sendo essa a denominação de um conceito abstrato que define maneiras de armazenar, manipular e recuperar dados estruturados unicamente na forma de tabelas. Esse tipo de Banco de Dados é o mais comum atualmente, sendo implementado nos softwares da Oracle, IBM DB2 (IBM, 2013), MySQL, PostgreSQL, entre outros.

Como o *Django* cuida de toda a comunicação do sistema com o banco de dados, não foi necessária a configuração ou execução de *Queries* específicas, sendo necessária, apenas, sua instalação na máquina que roda o sistema.

O diagrama com a estrutura de dados e sua implementação serão apresentados no quarto capítulo, na seção de Implementação e Desenvolvimento.

Capítulo 3

Demandas e Desafios

3.1 – Contextualização

Este projeto está inserido nas atribuições do Departamento de Engenharia Eletrônica e de Computação (DEL) da Universidade Federal do Rio de Janeiro (UFRJ). Uma dessas atribuições é a de ministrar cursos de Pós-Graduação Latu-Sensu, relacionados às áreas de Computação e Eletrônica. Os profissionais responsáveis por esses cursos são: Sergio Palma da Justa Medeiros e José Arthur da Rocha, professores desse departamento.

Esses cursos já são oferecidos pelo departamento há vários anos. Nesse mesmo período, tem sido observados problemas ligados à gestão contábil da Pós-Graduação. Parte desses problemas se deve à não aplicação de boas práticas de gestão de projetos pela Fundação COPPETEC. Essa Fundação é a responsável pela gerência financeira e pelos pagamentos e recebimentos relacionados à maioria dos cursos de Pós-Graduação da COPPE, bem como da UFRJ. Relacionam-se esses problemas às dificuldades de comunicação característica da COPPETEC, no que se refere aos relatórios financeiros por ela encaminhados, e à gestão daqueles recebidos.

Como forma de organizar todas as limitações verificadas nessa primeira avaliação do problema, foi idealizado o Projeto SiGeCon, cujo objetivo é proporcionar a automatização do processo de gestão contábil nesse departamento.

O SiGeCon – Sistema de Gerenciamento Contábil – é um sistema Web que visa facilitar a integração dos dados referentes a receitas e despesas financeiras, para facilitar o trabalho das secretárias do departamento, responsáveis por redigir os relatórios, e facilitar o planejamento financeiro dos cursos, feito pelos coordenadores.

3.2 – Problemas

Atualmente, como já foi explicado, a comunicação com a COPPETEC é feita por meio de Relatórios de Pedido de Pagamentos, que são enviados para essa instituição. Por outro lado, os Relatórios de Acompanhamento de Projeto são, por sua vez, a forma de comunicação dela para os departamentos. Enquanto os pedidos são encaminhados de forma física, em papel, o segundo tipo é enviado em arquivo de Excel, no formato de um extrato, com informações de clientes, de taxas pagas, pagamentos e recebimentos.

O primeiro problema identificado é a redação do Relatório de Pedido de Pagamentos, que é feita no Word, sendo necessário o preenchimento dos campos manualmente. Há um gasto de tempo desnecessário com a escrita do arquivo, além de seu armazenamento. Esse é feito em arquivos separados em pastas referentes às datas dos pedidos. Há ineficiência associada, tanto no preparo manual, quanto na posterior procura por uma respectiva despesa, e na análise das transações.

Por outro lado, o arquivo enviado pela COPPETEC, embora contenha as informações referentes aos pagamentos e recebimentos, estas não são mostradas de uma maneira clara e objetiva. Há muitas taxas, tanto pagas quanto recebidas, que apresentam baixo valor, que não interessam ao departamento, mas que “poluem”, o RAP.

Um outro problema é o tipo de arquivo adotado para a geração do Relatório de Pedido de Pagamentos, correspondentes a arquivos RTF, cujos campos, que não são funcionalmente agradáveis de se preencher, ainda apresentam grande facilidade de perda de formatação geral do documento.

Observando esses problemas foi possível definir o sistema que será melhor explicado adiante.

3.3 – Demandas

Tendo observado os problemas no processo real, fica mais fácil de se identificar e enumerar as demandas e necessidades de ambas as instituições.

A primeira demanda, mais imediata, está ligada aos Relatórios de Pedido de Pagamento. Como mencionado, os relatórios são preenchidos no programa Microsoft Word, manualmente. Muitas vezes, as informações são repetidas, havendo, então, desperdício de tempo com trabalho repetido. Além disso, como são armazenados em arquivos, o trabalho administrativo se mostra prejudicado, exigindo maior tempo e trabalho para as análises.

A proposta de atendimento dessa primeira demanda será definida como sendo a criação de um sistema que gere, rapidamente, os relatórios, facilitando o trabalho das secretárias. Esse sistema deverá ter um banco de dados que armazene as informações de forma estruturada, visando otimizar dados comuns tais como: pessoas e empresas favorecidas, para quem são direcionados os pagamentos; como também, suas informações bancárias. Além disso, há um campo para que o valor a ser pago seja escrito por extenso, o que exige mais tempo das secretárias, além de ser uma tarefa mecanizada. O sistema, portanto, também precisa converter o número, escrito em dígitos, para sua forma por extenso.

A segunda demanda, ligada a concatenação das despesas, é a emissão de uma planilha com as despesas, podendo ser filtradas por data. Nela, deverão ser relacionados os seguintes campos: projeto, data da despesa, descrição da despesa (ou transação) e o respectivo valor.

Tendo ambas as demandas relacionadas anteriormente, uma demanda de menor importância seria a inclusão das receitas do projeto, relacionadas aos seus clientes – os estudantes que cursam o MBA. Com a inclusão das receitas, a planilha deverá integrar essa informação na emissão do relatório.

Uma última demanda diz respeito ao RAP (Relatório de Acompanhamento de Projeto), enviado pela COPPETEC. Esse relatório inclui as receitas e despesas dos projetos, junto ao fluxo de caixa do projeto. O problema desse relatório é a quantidade de informação existente, além de sua formatação inapropriada e inconsistente, que dificulta o trabalho de análise sem a clareza das informações.

Essas são, portanto, as quatro principais demandas do sistema, que visam facilitar o trabalho dos gestores e das secretárias do Programa de MBA da Engenharia Eletrônica.

Capítulo 4

Projeto do Sistema

4.1 – Requisitos do Sistema

Com os problemas e demandas estabelecidas, será feito um levantamento de algumas funcionalidades necessárias no sistema.

O primeiro requisito é a emissão de um RPP, automaticamente. Esse requisito foi satisfeito através da criação de um arquivo do Excel, com formatação similar ao da COPPETEC. O sistema, então, precisa conseguir ler o arquivo modelo e gerar um novo, com os campos preenchidos, que possa ser enviado ao usuário para impressão.

Um outro requisito diz respeito ao acesso por parte dos funcionários. Como são algumas as pessoas que precisam ter acesso ao sistema, este deve ser implementado como um sistema Web. Como tal, precisa ser seguro, sendo necessários usuário e senha – que deve ser criptografada – para acesso ao banco de dados.

O sistema deve conseguir escrever, automaticamente, números por extenso. Esse requisito é necessário ao preenchimento do RPP.

Os usuários devem ser capazes de cadastrar diversos tipos de informação, que serão utilizadas para o preenchimento do RPP. O cadastro deve ser feito de forma a minimizar o tempo gasto, e deve ser integrado.

O sistema deve ser capaz de exportar um extrato em arquivo Excel para os gestores dos programas de Pós-Graduação. Esse arquivo deve conter informações referentes ao projeto, às datas de pagamento e recebimento, à descrição da transação financeira, além do valor movimentado.

Todos os requisitos mencionados anteriormente foram implementados. Outros, porém, poderiam ser implementados para melhorar o sistema. Os requisitos que não foram implementados são:

- Importação do RAP para inclusão e comparação de dados; e
- Visualização e edição de todos os cadastros do Banco de Dados.

A importação do RAP não pôde ser feita por problemas técnicos – de formatação e falta de informações – dos arquivos enviado pela COPPETEC aos departamentos.

A visualização e edição dos dados não foi implementada por já haver uma ferramenta do Django – ADMIN – com essa funcionalidade. O problema dessa ferramenta se refere à entrada de dados, que pode gerar erros no sistema, se não for feita cautelosamente.

4.2 – Casos de Uso

Os casos de uso do sistema, aqui implementados, estão relacionados na tabela a seguir. Todos os casos de uso são comuns à todos os tipos de “pessoas” que utilizam o sistema, havendo, apenas, um tipo de usuário. Outros casos de uso ainda podem ser implementados, visando melhorar a experiência dos usuários. Muitos desses casos, porém, podem ser realizados com a ferramenta ADMIN, oferecida pelo *Django*. Essa ferramenta será melhor explicada posteriormente.

Todos os casos de uso devem ser intuitivos para que o sistema cumpra sua função de facilitar e agilizar a elaboração dos relatórios, além do cadastro dos dados.

Código	Descrição	Dependências
1	Realizar <i>Login</i>	3, 4, 9,
2	Realizar <i>Logout</i>	1, 3, 4, 9,
3	Cadastrar Usuário	9,
4	Editar Usuário	3, 9,
5	Cadastrar MBA	
6	Cadastrar Agência	
7	Cadastrar Despesa Padrão	
8	Cadastrar Projeto	5, 6,
9	Cadastrar Pessoa	6,
10	Cadastrar Despesa do Projeto	5, 7, 9,
11	Cadastrar Receita	8, 9,
12	Emitir Relatório de Pedido de Pagamento	10,
13	Emitir Segunda Via de RPP	12,
14	Visualizar Extrato	10, 11,
15	Emitir Extrato em XLS	10, 11,

Tabela 1: Descrições dos casos de uso; e os respectivos casos de uso dos quais são dependentes.

Para utilizar o aplicativo, deve-se possuir um usuário cadastrado. Após sua instalação em alguma máquina, o sistema fornecerá um usuário padrão, com perfil de administrador: ROOT, com senha TOORROOT, utilizado em sua inicialização, sem que haja a necessidade de uso da ferramenta ADMIN (de controle de banco de dados).

O “*Super User*” Root deverá ser utilizado nos primeiros *Logins* no sistema e no cadastro dos primeiros usuários. Após a utilização do sistema, visando manter sua segurança, deverá ser feito o *Logout*, para que outras pessoas, sem autorização, não possam utilizá-lo. Após a inclusão de novos usuários, deve-se alterar a senha do administrador ROOT, ou até mesmo desativá-lo, uma vez que a senha é padrão, e está presente nesse trabalho, de acesso aberto ao público.

Este caso de uso é um padrão em sistemas web, aplicados no início de seu uso para que possam prover segurança aos dados e usuários.

Pode-se dividir os casos de uso restantes em Cadastros, Emissão de Relatórios de Pedidos de Pagamento e Emissão de Extrato. A seguir será feita uma breve descrição de cada caso de uso dentro desses grupos, e serão exibidas as respectivas telas.

4.2.1 – Cadastros

Na seção de Implementação desse Projeto, serão melhor explicadas todas as classes existentes no projeto, assim como seus atributos e inter-relações. No momento, serão apresentadas as telas de cadastro de algumas dessas classes, uma vez que nem todas devem ou precisam ter objetos criados por usuários.

O sistema permite o cadastro das seguintes classes:

- Despesas (Pedidos de Pagamento);
- Despesas Padrão;
- Receitas;
- Projetos;
- MBAs;
- Pessoas (Físicas ou Jurídicas) – Coordenadores – Favorecidos – Clientes;
- Agências Bancárias; e de
- Usuários.

A seguir, estão relacionadas as telas de cadastro de todas essas classes.

Nas telas, pode-se verificar que existem botões pretos com o nome da maioria dessas classes. Ao pressionar algum botão, o sistema abre a respectiva página de

cadastro, com o formulário de cadastro. Em alguns casos, com o ponteiro do mouse sobre o botão, aparece uma lista relacionando que tipos de objetos são necessários a cada classe. Isso facilita o trabalho, visto que, para que uma opção seja inserida nas listas dos formulários, essa deve ter sido registrada anteriormente.

Uma observação importante a ser feita diz respeito aos formulários. Em todos, quando um campo está sendo indicado com letra em negrito, significa que o campo é obrigatório. Caso contrário, seu preenchimento não é necessário. Uma exceção a essa regra é o cadastro de pessoas no que diz respeito ao cadastro de contas e agências, pois há uma interdependência entre ambas as informações.

Todos os campos são validados quando é executado o cadastro do formulário. Caso haja algum erro ou algum dado inválido, será interrompido o pedido e uma mensagem de erro será apresentada. Caso contrário, aparecerá uma mensagem de sucesso e o formulário será apagado. Uma exceção a essa regra é o formulário de cadastro de despesa, que não apresenta mensagem de sucesso, mas oferece o download do RPP preenchido.

The image shows a web browser window displaying the 'SiGeCon' system. The page title is 'Cadastro de Despesas'. The form contains the following fields and values:

Valor:	248.50
Localização do Bem:	
Número da Nota Fiscal:	
Projeto:	14115 - Eletrônica e Computação
Despesas com:	Material de Consumo
Favorecido:	Universidade Informática e Telefonia - Conta: 7518-3, 3642-8, Banco do Brasil S.A.
Tipo de Pagamento:	Depósito em conta corrente

A 'Cadastrar' button is located at the bottom right of the form area.

Figura 1: Tela de cadastro de despesas, com exemplo de cadastro.



Figura 2: Tela de cadastro de despesas padrão, mostrando a lista dos tipos de despesas possíveis.



Figura 3: Tela de cadastro de receitas do projeto, com um exemplo de como aparecem os clientes.



Figura 4: Tela de cadastro de projetos, com exemplo.



Figura 5: Tela de Cadastro de MBA



Figura 6: Tela de cadastro de agências bancárias.



Figura 7: Tela de cadastro de pessoas físicas e jurídicas, com exemplo de cadastro de pessoa física.



Figura 8: Tela de cadastro de pessoas físicas e jurídicas, com exemplo de pessoa jurídica.



Figura 9: Tela de cadastro de usuários.



Figura 10: Tela de edição de usuários.

4.2.2 – Relatório de Pedidos de Pagamento

Esse caso de uso está ligado ao cadastro das Despesas de Projeto. Após o cadastro das despesas, além de inserir as informações no Banco de Dados, o sistema gera um arquivo XLS no formato utilizado pela COPPETEC, usado para o requerimento de pagamentos, com todas as informações preenchidas nos campos necessários. Outra funcionalidade oferecida é a emissão de segunda via dessas despesas.

O sistema oferece o download do arquivo para a máquina do usuário. O arquivo disponibilizado está pronto para: ser impresso, carimbado, assinado, e enviado para a Fundação. Uma Cópia não preenchida de cada formulário – um para pessoa física e outro para pessoa jurídica – segue em anexo, junto a um exemplo preenchido de despesa.

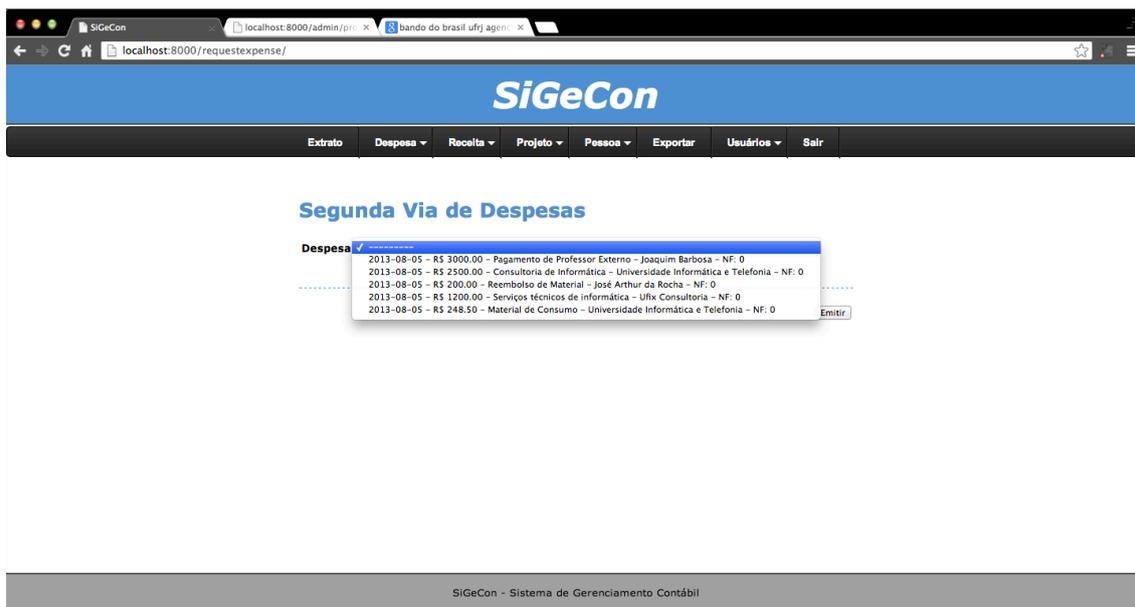


Figura 11: Tela de emissão de segunda via de despesas.

4.2.3 – Extrato

Ao entrar no sistema, a página inicial apresenta o Extrato Geral, ordenado por número do projeto, do menor para o maior; seguido pela data, começando na mais antiga e terminando na mais recente. Esse extrato é uma tabela simples, mas com os quatro campos necessários. Uma desvantagem dessa tabela é a de não ser possível trabalhar sobre ela e, por isso, há uma opção para a emissão de um arquivo do Microsoft Excel, para que o usuário possa aproveitar melhor as tabelas e as informações existentes.

Em ambos os casos, tanto no exibido na tela, quanto na emissão do “xls”, há a possibilidade de filtragem dos dados. A primeira opção é filtrar os dados pelo projeto ao qual está vinculado. A segunda, é filtrar pela data das receitas e das despesas, estabelecendo os limites iniciais e / ou finais do período desejado. Ambos os filtros podem ser aplicados ao mesmo tempo.

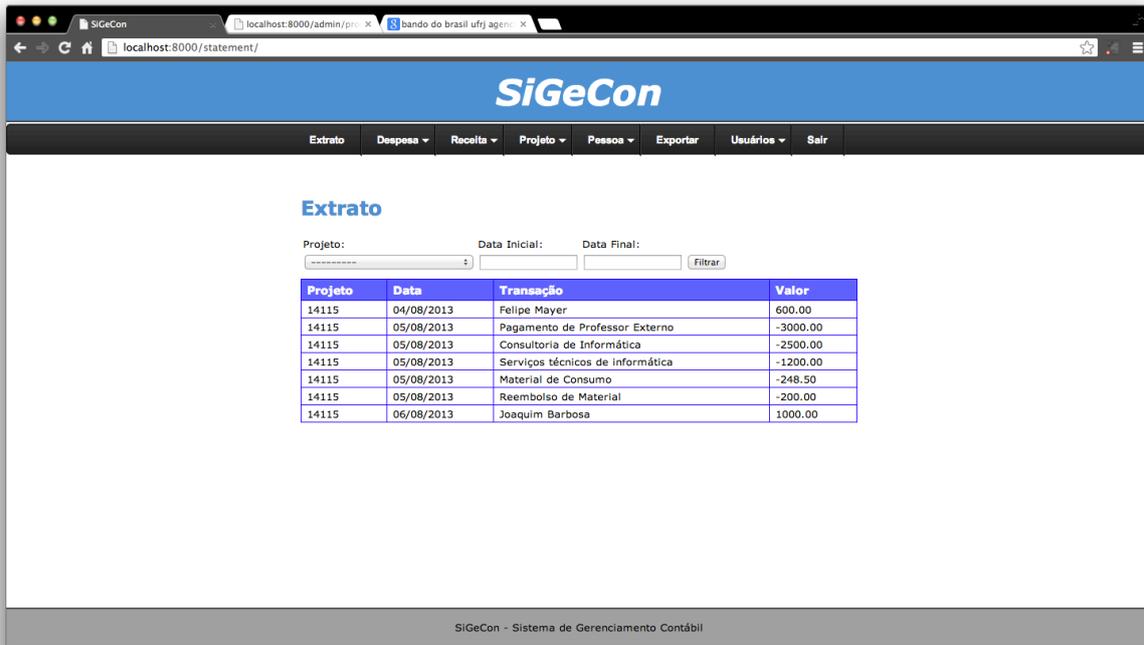


Figura 12: Tela de visualização de extratos.

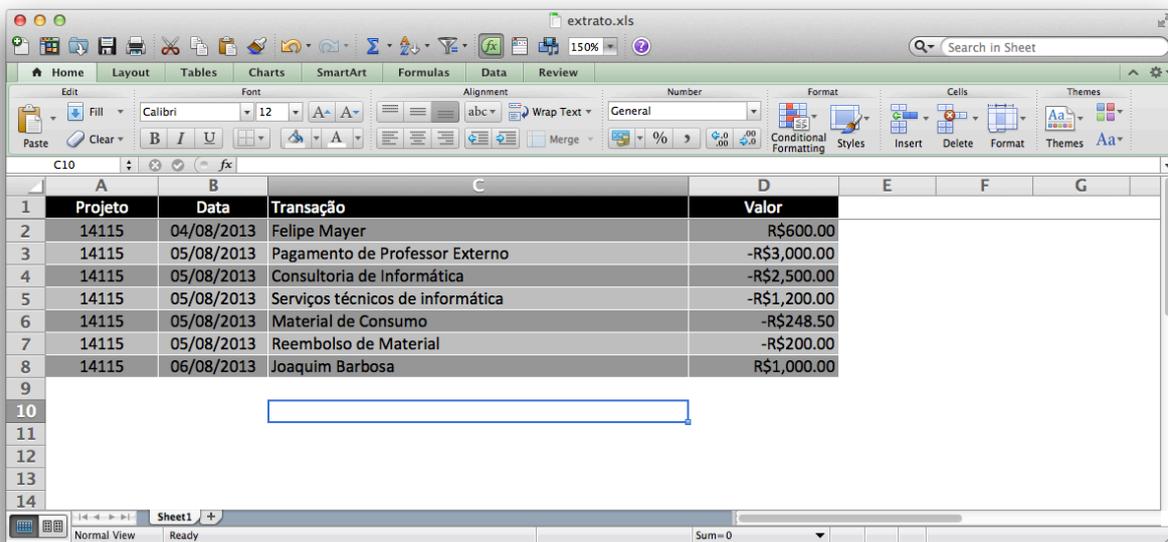


Figura 13: Imagem do Microsoft Excel exibindo o extrato emitido em XLS.

4.3 – Ferramentas & Softwares

Nessa seção será feito o levantamento dos softwares utilizados no desenvolvimento do projeto, considerando todos os softwares auxiliares.

O sistema foi inteiramente desenvolvido em um computador com o sistema operacional Mac OSX (Apple, 2013), e a plataforma XCode de desenvolvimento. Esse ambiente de programação foi escolhido, não só por ser o utilizado no cotidiano do autor, como também pelo fato da programação ser compatível com sistemas *Linux* (Linux, 2013) – software *OpenSource* –, o que não geram custos extras, além de serem muito utilizados no Departamento de Eletrônica e em seus servidores. Outro forte motivo foi a escolha do *Framework Django*, que não apresenta compatibilidade direta com ambientes Microsoft Windows (Microsoft, 2013).

Por ser uma aplicação Web, usuários de máquinas com os três tipos de Sistema Operacional poderão acessar o portal SiGeCon sem restrições.

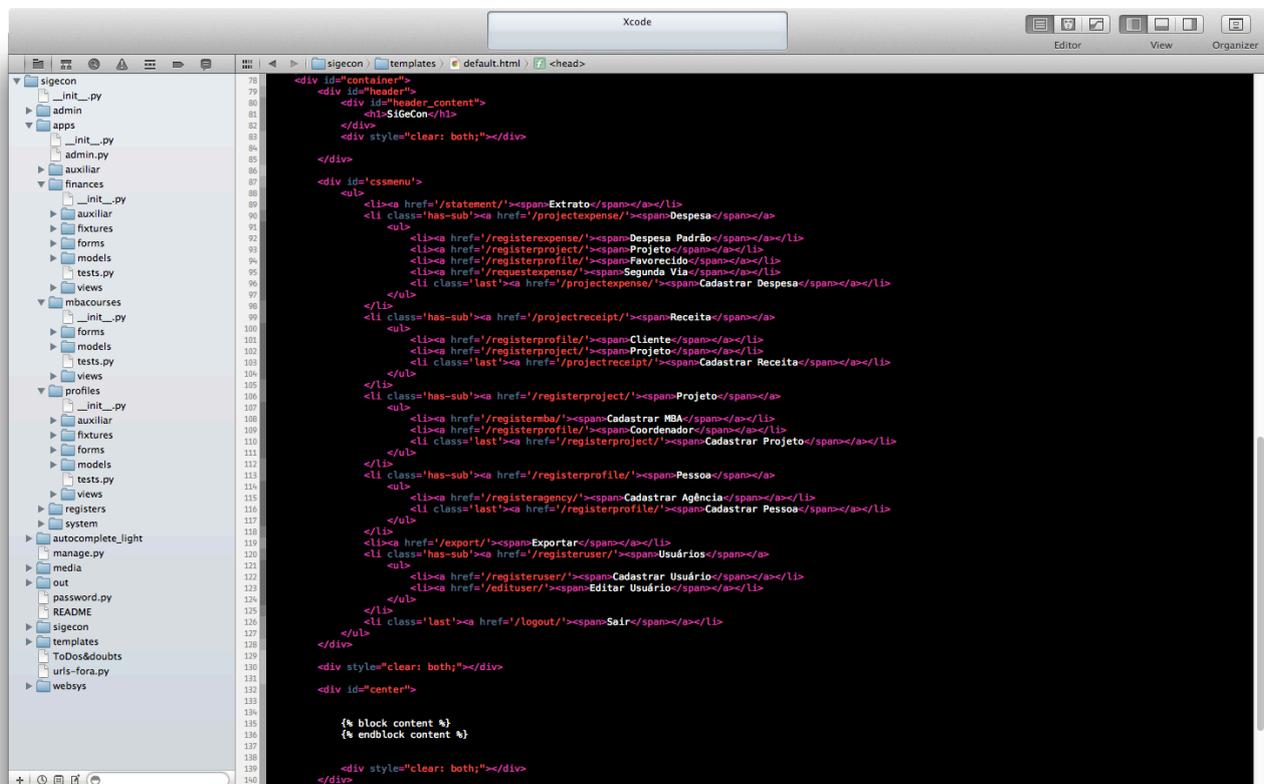


Figura 14: Imagem do aplicativo XCode, utilizado no desenvolvimento do projeto, exibindo o código da página padrão.

Como já mencionado, o Framework e a linguagem de programação utilizados foram o *Django* e *Python*. A plataforma oferece ferramentas genéricas voltadas para o desenvolvimento de sistemas Web. Outra funcionalidade é a interação com Bancos de Dados, sem a restrição de um banco específico. Podem ser utilizados diferentes Bancos

de Dados, com apenas uma pequena alteração nas configurações iniciais do sistema, feita por meio da alteração de menos de 10 linhas no arquivo “*settings.py*”. A portabilidade que esse Framework oferece e suas bibliotecas, portanto, foram decisivos para sua escolha no desenvolvimento do Projeto de Graduação.

O Banco de Dados inicialmente escolhido foi o PostgreSQL (The PostgreSQL Global Development Group, 2013), que apresentou problemas de instalação na máquina do autor, embora haja a opção de sua instalação no Mac OSX. Esse banco oferece mais recursos, e é considerado melhor do que o utilizado: MySQL (Oracle Corporation, 2013). A maior facilidade de instalação deste último e do MySQLWorkbench (Oracle Corporation, 2013) – ferramenta gráfica para controle do Banco de Dados – foi importante na sua escolha, como também, sua vasta utilização no meio acadêmico, havendo maior e mais completa documentação, além de soluções, existentes na internet.

O Django oferece, também, a ferramenta ADMIN, para controle direto dos objetos no Banco de Dados. A partir dela, pode-se inserir, editar e deletar registros do banco, recurso muito utilizado nos testes do sistema para verificação do cadastro de dados.

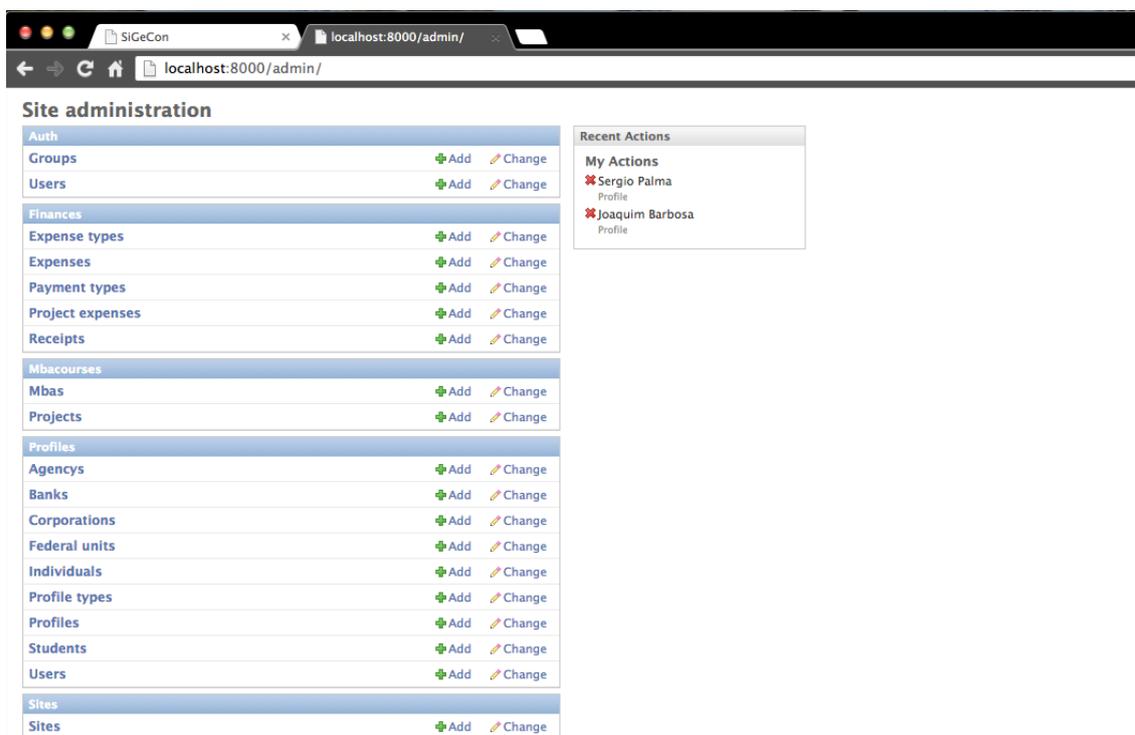


Figura 15: Imagem da página ADMIN do Django, utilizada para modificações diretas no banco de dados.

As ferramentas e softwares apresentados anteriormente são aplicativos complexos, e plataformas de desenvolvimento. A seguir, serão expostas algumas

ferramentas – bibliotecas e funções auxiliares - de *Django* e *Python*, já prontas, que foram utilizadas no desenvolvimento do Projeto.

O sistema precisa trabalhar com arquivos do tipo XLS, do Microsoft Excel. Para conseguir lidar com esse tipo de arquivo, foram utilizados os seguintes pacotes de ferramentas para Python: XLRD, XLWT e XLUTILS (Simplistix, 2013), que possuem módulos para leitura, gravação e cópia de arquivos do Excel. Essa ferramenta foi essencial para gerar os Relatórios do Projeto.

Para facilitar a criação e formatação desses arquivos, foram criados os formulários, não preenchidos e já formatados, com o Microsoft Excel (Microsoft, 2013). Foram deixados numa pasta de arquivos *templates* (não confundir com os *templates* dos *Django*). O sistema, quando executando a emissão de algum relatório, abre o arquivo com o formulário original e vazio; copia para uma nova planilha, criada na memória; altera os campos adicionando as informações do Banco de Dados; e, finalmente, envia o Relatório de Pedido de Pagamento para o usuário.

Além de não haver a necessidade de trabalho de formatação em linha de comando, as mudanças de formatação futuras são mais fáceis e intuitivas de serem feitas. Para tal, pode-se, simplesmente, alterar a formatação do arquivo *template*. O código também fica menos complexo e sua execução, mais ágil.

Na emissão do Relatório de Pedido de Pagamento é necessária a escrita, por extenso, de números referentes aos valores de pagamento. Para essa conversão, foi utilizada uma rotina pronta (Santos, Cervi, Vital, & Pontes, 2013), que precisou ser modificada, por causa de alguns erros em sua implementação.

Da mesma fonte anterior, foram retiradas duas rotinas para a validação e formatação de CPF e CNPJ (Python Brasil, 2013). Em todos os arquivos contendo essas partes extraídas da internet há uma indicação no cabeçalho de sua fonte e um aviso de que os códigos foram copiados ou copiados e modificados.

4.4 – Implementação e Desenvolvimento

Nessa seção será melhor explicado como o sistema funciona, a modelagem de dados por trás do sistema, e como foi feita a programação do sistema. Na tópico do *Django*, serão melhor explicadas as classes e seus atributos, além de suas inter-relações.

Pedido de Pagamento. Por outro lado, a *Profile* armazena as *‘pessoas’* (empresas ou indivíduos) envolvidas e relacionadas ao programa de MBA.

Outras informações serão dadas mais adiante.

4.4.2 – Django

Uma das primeiras instruções dos tutoriais do Django ensina a criar os aplicativos, ou módulos, do sistema. O SiGeCon é constituído por três módulos básicos: *‘finances’*, *‘mbacourses’*, e *‘profiles’*.

Todos os módulos incluem as classes (*models*), formulários (*forms*), funções criação de páginas Web (*views*), registros iniciais do Banco de Dados (*fixtures*) necessários para o funcionamento do sistema. Os módulos *‘finances’* e *‘profiles’* contam, também, com pastas auxiliares, com algumas funções que podem ser utilizadas por quaisquer classes. Alguns exemplos são a verificação de CPF e CNPJ, e a geração dos relatórios.

A tabela a seguir faz a divisão das classes apresentadas na seção anterior, referente a modelagem de dados, entre os três módulos.

Profiles	Finances	MBACourses
Agency	Expense	MBA
Bank	ExpenseType	Project
FederalUnit	PaymentType	
ProfileType	ProjectExpense	
Profile	Receipt	
User		

Desse conjunto, podem ser destacadas quatro classes especiais: *‘Bank’*, *‘ExpenseType’*, *‘FederalUnit’* e *‘PaymentType’*. Todas essas classes já tem seus objetos criados e inseridos no Banco de Dados no momento da instalação do sistema.

As Unidades Federativas do Brasil existem e já estão definidas. Não constituem registros que precisam de alteração constante. O mesmo pode se dizer dos bancos instalados no país. Por isso, não há opção de cadastro ou alteração de ambas as classes.

No caso dos Tipos de Pagamento e de Despesa, são opções preestabelecidas do Relatório de Pedido de Pagamento. Caso haja alteração no formulário, ou os gestores queiram novas possibilidades, os objetos dessas classes precisarão ser atualizados.

Caso haja a necessidade de modificação dessas classes com a inserção, alteração ou remoção de algum objeto, a mesma deverá ser feita através da ferramenta ADMIN. Deve-se lembrar que qualquer alteração feita com essa ferramenta exige muito cuidado para que erros não sejam criados no sistema.

No final dessa seção, encontram-se imagens com as listas geradas a partir desses objetos e classes.

A Classe ‘*MBA*’ apenas armazena o nome do Programa da UFRJ, no qual o projeto de MBA está inserido, emitente do requerimento de pagamento. O nome registrado nessa classe será o utilizado para o preenchimento do campo “Programa” no RPP. Para o Departamento de Eletrônica, normalmente, haverá apenas um objeto MBA, com nome ‘POLI’.

A classe ‘*Project*’ armazena os dados referentes a um Projeto de Pós-Graduação. Nela são armazenados os seguintes atributos: o número do projeto, normalmente de cinco dígitos; o nome, por exemplo, “Curso de Especialização MBA Engenharia de Software – 22^a”; a data de início e de conclusão do projeto. Armazena, também, os códigos de objetos (Foreign Keys) das seguintes classes: ‘MBA’ – informando em que Programa está inserido –, e ‘Profile’ – registrando o coordenador do Projeto.

A classe ‘*Agency*’, registra a agência bancária de alguma pessoa, tanto física, quanto jurídica. Nela são armazenados os dados referentes ao número da agência (no formato “1234-5”), ao banco e Unidade Federativa de origem, e ao seu próprio nome.

As agências são campos presentes no cadastro de pessoas, e seus dados são imprescindíveis na emissão do RPP, caso a opção “Depósito em conta corrente” seja selecionada. Nesse caso, o sistema acusará um erro de Pagamento Indisponível.

A classe ‘*Profile*’ é a responsável pelo armazenamento das diferentes pessoas envolvidas nos projetos de Pós-Graduação, podem ser de diferentes tipos, armazenados na classe ‘*ProfileType*’, tais como aluno, professor, ou, até mesmo, pessoas jurídicas. Essa classificação é feita de acordo com o papel exercido no programa. As outras informações armazenadas são: seu nome completo; o tipo de pessoa, física ou jurídica, sendo necessária a comprovação através do Cadastro Nacional, podendo ser CPF ou CNPJ; além da agência e conta bancária. Ambos os últimos são opcionais, mas um não pode ser cadastrado sem o outro.

A classe ‘*User*’ é a responsável pelo gerenciamento de usuários. Para a criação de um usuário, deve-se criar um objeto ‘*Profile*’, que estará associado a conta. Apenas alguns tipos de ‘*ProfileTypes*’ tem a permissão de serem cadastrados como usuários do

sistema. Os tipos são: “Coordenador”, “Professor”, “Administrador” e “Secretária”. Há uma *flag* para relacionar usuários ativos ou não, caso haja a necessidade de retirar o acesso de algum funcionário, ou paralisar o sistema momentaneamente. Por último, e mais importante, são armazenados o nome de usuário (*Login*) e a senha, utilizados para o acesso ao SiGeCon.

A classe ‘*Receipt*’ representa um fluxo de caixa positivo, normalmente, advindo de algum cliente (estudante) da Pós-Graduação. São armazenados a referência para o cliente – objeto ‘*Profile*’ –; a data de entrada da Receita; o projeto que recebe os recursos, além do valor da fatura.

A classe ‘*Expense*’ armazena os tipos de despesa padrão. A descrição armazenada é utilizada para o preenchimento do campo “Despesas com:” do RPP, enquanto o “Tipo de despesa” se refere à opção a ser marcada (com um “X”) no Formulário.

Por último, e mais importante, a classe ‘*ProjectExpense*’ é a responsável pelo armazenamento das informações dos diversos Relatórios de Pedido de Pagamento. O formulário a ser preenchido no sistema é a versão automatizada do RPP. Os dados registrados são: valor da despesa, caso não haja valor padrão; localização do bem; número da nota fiscal de alguma aquisição; o projeto de Pós-Graduação; a despesa padrão; a pessoa favorecida (física ou jurídica); e o tipo de pagamento.

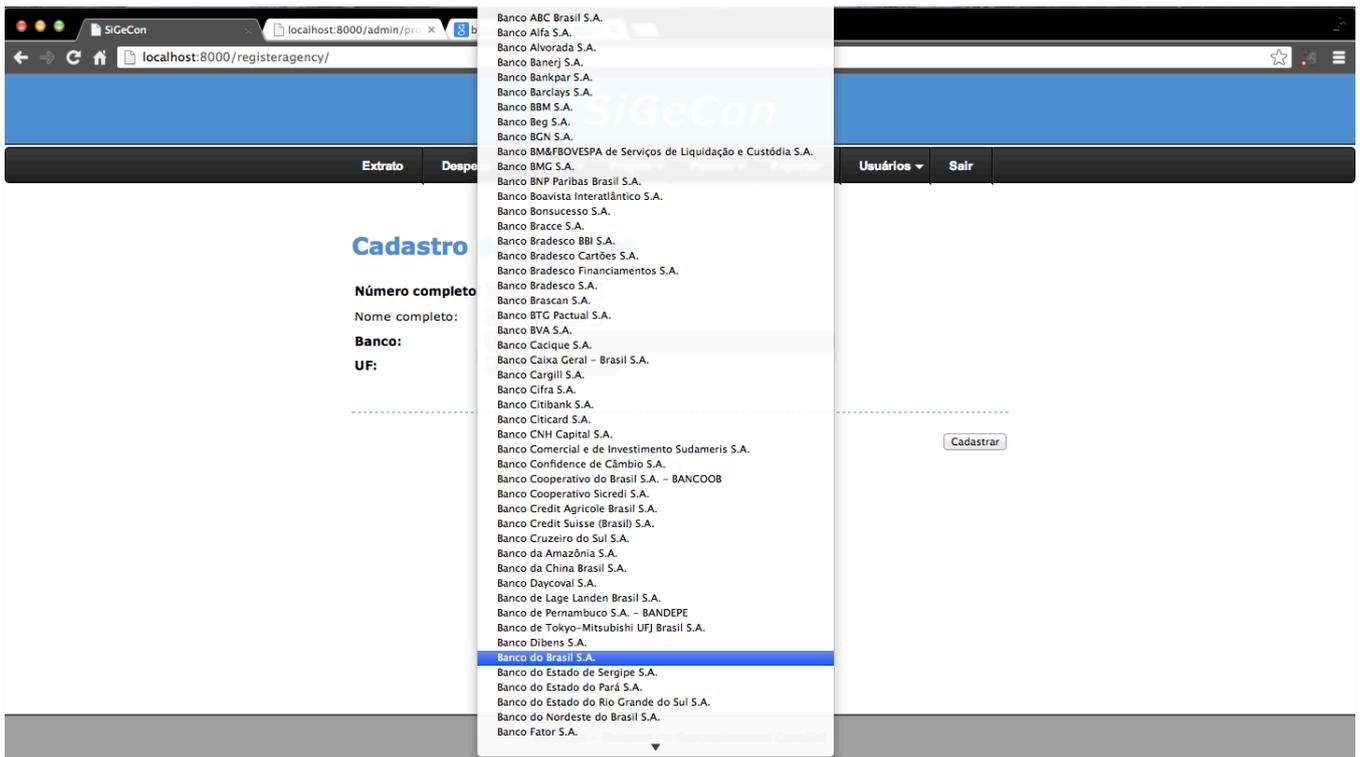


Figura 17: Sistema exibindo lista com todos os bancos do Brasil.



Figura 18: Sistema exibindo lista com os quatro tipos de despesa padrão.

SiGeCon

Extrato Despesa Recelita Projeto Pessoa Exportar Usuários Sair

Cadastro de Despesas

Valor:

Localização do Bem:

Número da Nota Fiscal:

Projeto:

Despesas com:

Favorecido:

Tipo de Pagamento:

- Depósito em conta corrente
- Pagamento através de Boleto Bancário
- Pagamento em cheque

Figura 19: Sistema exibindo lista com os três tipos de pagamento de despesas.

SiGeCon

Extrato Despesa Recelita Projeto Pessoa Exportar Usuários Sair

Cadastro de Agências

Número completo:

Nome completo:

Banco:

UF:

- Acre
- Alagoas
- Amazonas
- Amapá
- Bahia
- Ceará
- Distrito Federal
- Espírito Santo
- Goiás
- Maranhão
- Minas Gerais
- Mato Grosso do Sul
- Mato Grosso
- Pará
- Paraíba
- Pernambuco
- Piauí
- Paraná
- Rio de Janeiro
- Rondônia
- Rio Grande do Norte
- Roraima
- Rio Grande do Sul
- Santa Catarina
- Sergipe
- São Paulo
- Tocantins

SiGeCon - Sistema de Gerenciamento Contábil

Figura 20: Sistema exibindo lista com os Estados do Brasil.

4.4.3 – HTML & CSS

Por ser um sistema Web, a criação de páginas em HTML é necessária. Para tal, o *Django* trabalha alterando alguns campos dos arquivos “*templates*”, escritos em HTML. O componente do *Django* que executa essa atividade é o *Template System*.

Visando facilitar e modularizar a formatação das páginas, foram utilizados arquivos CSS básicos, que são responsáveis pela estilização e organização do conteúdo das páginas.

Segue, abaixo, a tela inicial do sistema, apresentando o extrato. Nela, pode-se visualizar o padrão estilístico do sistema. Em seguida, encontra-se parte do código do arquivo “*default.html*” – que constitui a base da página - , necessário à formatação da página. O Código está sendo exibido para mostrar a simplificação de um HTML com a ajuda do CSS.

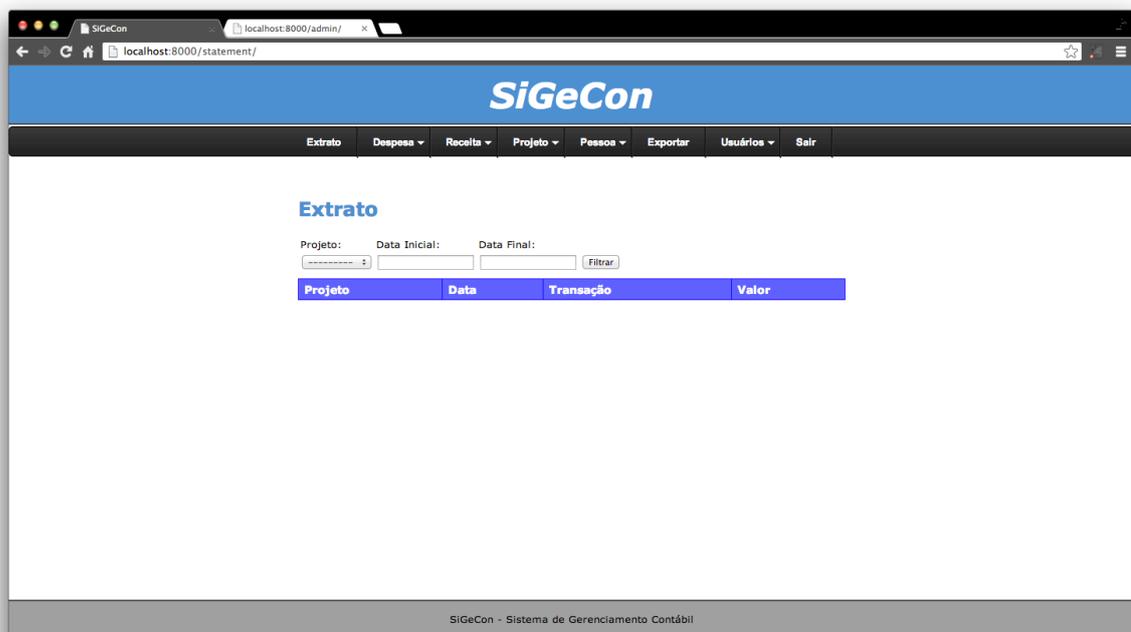


Figura 21: Sistema na página inicial de extrato. Pode-se visualizar o padrão de tela sob todas páginas do sistema.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pt-br" lang="pt-br">
<head>
<title>SiGeCon</title>
<link rel="stylesheet" type="text/css" href="{{ MEDIA_URL }}css/inside_layout.css"/>
<link rel="stylesheet" type="text/css" media="screen" href="{{ MEDIA_URL }}css/redmond/jquery-
ui-1.8.16.custom.css"/>
<link rel="stylesheet" type="text/css" media="screen" href="{{ MEDIA_URL }}css/ui.jqgrid.css"/>
{% block style %} {% endblock style %}

{% block script %} {% endblock script %}
</head>
<body>
<div id="container">
<div id="header">
<div id="header_content">
<h1>SiGeCon</h1>
</div>
<div style="clear: both;"></div>
</div>
<div id="cssmenu">
<ul>
<li><a href="/statement/"><span>Extrato</span></a></li>
<li class="has-sub"><a href="/projectexpense/"><span>Despesa</span></a>
<ul>
<li><a href="/registerexpense/"><span>Despesa Padrão</span></a></li>
<li><a href="/registerproject/"><span>Projeto</span></a></li>
<li><a href="/registerprofile/"><span>Favorecido</span></a></li>
<li><a href="/requestexpense/"><span>Segunda Via</span></a></li>
<li class="last"><a href="/projectexpense/"><span>Cadastrar Despesa</span></a></li>
</ul>
</li>
<li class="has-sub"><a href="/projectreceipt/"><span>Receita</span></a>
<ul>
<li><a href="/registerprofile/"><span>Cliente</span></a></li>
<li><a href="/registerproject/"><span>Projeto</span></a></li>
<li class="last"><a href="/projectreceipt/"><span>Cadastrar Receita</span></a></li>
</ul>
</li>
<li class="has-sub"><a href="/registerproject/"><span>Projeto</span></a>
<ul>
<li><a href="/registermba/"><span>Cadastrar MBA</span></a></li>
<li><a href="/registerprofile/"><span>Coordenador</span></a></li>
<li class="last"><a href="/registerproject/"><span>Cadastrar Projeto</span></a></li>
</ul>
</li>
<li class="has-sub"><a href="/registerprofile/"><span>Pessoa</span></a>
<ul>
<li><a href="/registeragency/"><span>Cadastrar Agência</span></a></li>
<li class="last"><a href="/registerprofile/"><span>Cadastrar Pessoa</span></a></li>
</ul>
</li>
<li><a href="/export/"><span>Exportar</span></a></li>
<li class="has-sub"><a href="/registeruser/"><span>Usuários</span></a>
<ul>
<li><a href="/registeruser/"><span>Cadastrar Usuário</span></a></li>
<li><a href="/edituser/"><span>Editar Usuário</span></a></li>
</ul>
</li>
<li class="last"><a href="/logout/"><span>Sair</span></a></li>
</ul>
</div>
<div style="clear: both;"></div>
<div id="center">
{% block content %}
{% endblock content %}
<div style="clear: both;"></div>
</div>
<div id="footer">
<div id="footer_content">
<p>SiGeCon - Sistema de Gerenciamento Contábil</p>
</div>
</div>
</div>
</body>
</html>

```

Figura 22: Código do arquivo ‘default.py’, template de todas as páginas do sistema.

4.5 – Testes do Sistema

Os testes do sistema foram feitos à medida que o projeto estava sendo desenvolvido. A cada nova classe, função ou demais alterações, muitas das funcionalidades e classes eram testadas, uma vez que uma pequena alteração no código de um aplicativo pode levar outros componentes a exibirem erros.

Os principais testes envolveram:

- Login e Logout do sistema;
- Validação da entrada de dados dos formulários;
- Validação da filtragem de dados para a criação de listas;
- Verificação quanto ao funcionamento das páginas do sistema;
- Verificação da inclusão dos dados, de forma correta, no banco de dados;
- Emissão de Relatórios de Pedido de Pagamento;
- Emissão de Extrato em arquivo XLS;
- Verificação dos relatórios, para constatar possíveis erros.

Na conclusão será feita uma consideração a respeito dessa etapa do projeto.

4.6 – Propostas Futuras

A Proposta futura mais imediata é a implementação de uma função que importe o Relatório de Acompanhamento de Projeto (RAP), e compare com o banco de dados. O sistema deveria extrair do RAP as informações necessárias para a população do banco de dados, após a comparação com objetos do Banco de Dados, para verificar se os pagamentos foram efetuados e se as receitas foram contabilizadas.

Essa etapa não foi implementada pela dificuldade de se extrair as informações do RAP que apresenta poucas informações, por exemplo, sobre clientes, o que pode prejudicar o funcionamento do sistema, com campos obrigatórios não preenchidos, na atual versão. Com a inclusão de dados incompletos em objetos do banco de dados, o sistema apresentaria muitos erros. Uma melhor conversa e negociação com a COPPETEC poderia ser feita para que se arranjasse uma maneira de integrar os sistemas ou simplesmente de incluir mais informações no RAP, facilitando a sua importação para o sistema. Essa possível conversa poderia facilitar o trabalho dos profissionais dos departamentos da Escola Politécnica, e também da própria Fundação.

Outra proposta para o futuro é oferecer o sistema a outros departamentos que oferecem MBAs e enfrentam os mesmos problemas de gestão das informações contábeis. Outros programas poderão se beneficiar do sistema, reduzindo trabalhos e aprimorando as informações gerenciais.

Outra função não desenvolvida no sistema, relacionada aos registros, foi a possibilidade de edição e de remoção de objetos. Essa etapa não foi implementada por algumas razões, mas num futuro, seria interessante sua implementação, visando, novamente, a melhor experiência dos usuários do sistema. A primeira razão é o fato do *Django* disponibilizar uma página (ferramenta ADMIN – *url/admin*) que permite que essas alterações sejam feitas no banco de dados. Embora não sejam validados os campos dos formulários, essas modificações deveriam ser feitas com cautela, para que não gere problemas e as informações erradas não sejam passadas aos gestores – que constitui o segundo motivo.

Adentrando no sistema, uma alteração que poderia facilitar a busca por pessoas e despesas seria a modificação de alguns campos “*ChoiceField*” em lista para “*Text boxes*” com “*Auto-complete*”. Existe um pacote para *Django* que oferece essa comodidade, mas várias pequenas e específicas validações seriam necessárias para sua introdução no sistema, embora a usabilidade poderia ser incrivelmente elevada, caso novas funções de edição fossem incluídas.

Capítulo 5

Conclusões

5.1 – Avaliação das Ferramentas

A combinação do Framework *Django* com a linguagem *Python* se mostrou muito eficiente e de fácil usabilidade e agilidade na execução dos programas. Simplificou e facilitou, também, a utilização do banco de dados, que pode deixar o código mais complexo e trabalhoso. Um dos pontos fundamentais, porém, foi a rapidez da elaboração dos códigos e da vasta facilitação do desenvolvimento de páginas Web.

Em relação ao banco de dados, não é possível obter uma avaliação significativa, uma vez que há integração total com o Banco de Dados por parte do *Django*. A avaliação que pode ser feita é que, nesse quesito, ambos satisfizeram todas as necessidades do Projeto.

A vasta documentação online das ferramentas, as soluções para os erros encontrados e a quantidade de amigos que conhecem o Framework foram muito importantes para o desenvolvimento desse Projeto de Graduação.

O GIT se mostrou muito importante para o controle de versões e para o desenvolvimento de aplicativos. Sua facilidade, principalmente quando utilizando uma interface gráfica, e seu poder de controlar e “fundir” (*merge*) versões diferentes, foi essencial.

O Site do Python Brasil mostrou como é importante a utilização de um padrão difundido de programação e a busca por ferramentas já prontas, para que não seja necessário o retrabalho de algumas partes do desenvolvimento de sistemas Web. Esse portal apresenta muitas informações a respeito do *Django* e muitos utilitários já prontos, que facilitam ainda mais o desenvolvimento de aplicativos.

Em geral, a experiência com ferramentas gratuitas, softwares livres e de código aberto, foi muito importante no desenvolvimento do sistema, e pôde-se observar a quantidade e qualidade do que já existe no mundo OpenSource, o que incentiva ainda mais a utilização desse tipo de ferramenta.

5.2 – Aprendizado

O tempo alocado para o aprendizado de *Python* e de *Django* antes da implementação do sistema foi curto. Durante o desenvolvimento, a documentação da linguagem e do framework foram vastamente consultadas. A prática da programação é a melhor forma de aprender, mas necessita de uma boa documentação, existente para ambos. Isso mostra como uma documentação bem especificada e atualizada é essencial para um projeto.

Infelizmente, em muitos casos, não só de programação, mas em diversos outros tipos de projetos, não são redigidas boas documentações que facilitem o trabalho. Desenvolver um projeto com base em uma documentação atualizada, organizada e sólida, portanto, mostrou-se uma boa prática para a implementação de sistemas e outros aplicativos.

Outro aprendizado foi a necessidade de uma metodologia sólida para o teste de aplicativos e sistema complexos. Os testes desse sistema foram feitos, como mencionado, junto com o desenvolvimento e implementação do mesmo, mas não havia uma rotina de testes. Com uma metodologia bem definida, e com rotinas automatizadas, o trabalho repetitivo poderia ter sido poupado.

O maior problema dos testes está relacionado ao fato de que pequenas mudanças em uma parte do projeto podem afetar, negativamente, outra parte. Isso se mostrou muito presente, próximo à finalização do Projeto.

Fica, portanto, o aprendizado do valor da organização dos códigos; de uma boa documentação em conjunto com uma programação visualmente amigável, seguindo boas práticas de programação; e com rotinas e metodologia de testes bem definidos.

5.3 – Considerações Finais

O Projeto de Graduação foi uma excelente oportunidade de aprendizagem para o autor. Foram vividas diversas dificuldades relacionadas ao acompanhamento de projetos; à falta de recursos, principalmente, de tempo; e ao desenvolvimento de um sistema inteiro, do início ao fim.

Tudo isso foi essencial à conclusão de uma graduação em engenharia, para que os alunos consigam ter alguma experiência antes de entrar no mercado de trabalho.

Bibliografia

- Hunt, A., & Thomas, D. (1999). *The Pragmatic Programmer: From Journeyman to Master* (1st Edition ed.). Addison-Wesley Professional.
- Apple. (2013). *Mac OSX Mountain Lion*. Acesso em 2013, disponível em Apple: <http://www.apple.com/osx/>
- Atlassian. (2013). <http://sourcetreeapp.com/>. Acesso em 2013, disponível em SourceTree: <http://sourcetreeapp.com/>
- Django Software Foundation. (2013). *Django Project*. Acesso em 2013, disponível em Django: <https://www.djangoproject.com/>
- Git Hub. (2013). *GIT*. Acesso em 2013, disponível em <http://git-scm.com/>
- IBM. (2013). *IBM DB2 database software*. Acesso em 2013, disponível em IBM: <http://www-01.ibm.com/software/data/db2/>
- Linux. (2013). Acesso em 2013, disponível em Linux: <http://www.linux.org/>
- Microsoft. (2013). *Microsoft Excel for Mac*. Acesso em 2013, disponível em Microsoft: <http://www.microsoft.com/mac/excel>
- Microsoft. (2013). *Microsoft Windows*. Acesso em 2013, disponível em Microsoft: <http://windows.microsoft.com/en-US/windows/home>
- Oracle Corporation. (2013). Acesso em 2013, disponível em My SQL: <http://www.mysql.com/>
- Oracle Corporation. (2013). *MySQL Workbench*. Acesso em 2013, disponível em MySQL: <http://www.mysql.com/products/workbench/>
- Python Brasil. (2013). *Python CookBook*. Acesso em 2013, disponível em Python Brasil: <http://www.python.org.br/wiki/CookBook>
- Python Brasil. (2013). *Receita: Verificador de CNPJ*. Acesso em 2013, disponível em Python CookBook: <http://www.python.org.br/wiki/VerificadorDeCnpj>
- Python Software Foundation. (2013). *Python*. Acesso em 2013, disponível em Python: <http://www.python.org/>
- Santana, O., & Galesi, T. (2010). *Python e Django: Desenvolvimento ágil de aplicações web*. São Paulo, São Paulo: Novatec Editora.
- Santos, F. W., Cervi, G. H., Vital, L. B., & Pontes, M. A. (2013). *Numero Para Palavras Portugues*. Acesso em 2013, disponível em Python CookBook: <http://www.python.org.br/wiki/NumeroParaPalavrasPortugues>

Simplistix. (2013). *Working with Excel Files in Python*. Acesso em 2013, disponível em Python-Excel: <http://www.python-excel.org/>

Stackoverflow. (2013). *Stackoverflow*. Acesso em 2013, disponível em <http://stackoverflow.com/>

The PostgreSQL Global Development Group. (2013). Acesso em 2013, disponível em PostgreSQL: <http://www.postgresql.org/>

Apêndice A

Relatório de Pedido de Pagamento

Ficha de Pedido de PAGAMENTO de PESSOA JURÍDICA para materiais ou serviços (anexar nota fiscal)		
Nº de Protocolo (COPPETEC):		
Data de Entrega:		
Programa:	POLI	Nº do Projeto 14.000
Coordenador do Projeto - Nome:	José Arthur da Rocha	
Valor do Pagamento:	R\$ 248,50	
(por extenso)	Duzentos e quarenta e oito reais e cinquenta centavos	
Para despesas com:	Material de Consumo	
<input checked="" type="checkbox"/> Material de Consumo	<input type="checkbox"/> *Material Permanente	<input type="checkbox"/> *Equipamento <input type="checkbox"/> Serviços de Terceiros
*Localização do Bem:		
Despesa pré-empenhada na COPPETEC?	<input type="checkbox"/> Sim	<input type="checkbox"/> Não
Favorecido:	Empresa Universidade Consultoria Ltda.	
CNPJ (antigo CGC):	12.345.678/0001-50	
<input type="checkbox"/>	Pagamento em cheque	
<input type="checkbox"/>	Pagamento através de boleto bancário	
<input checked="" type="checkbox"/>	Depósito em conta corrente *(preencha abaixo)	
Banco:		
Nome:	- Banco do Brasil S.A.	
Número:	- 001	
Agência:		
Nome:	-	
Número Completo:	- 1234-8	
Conta Corrente nº :	- 1982-3	
Estado:	- Rio de Janeiro	
Ao assinar este pedido o Coordenador do projeto assume a responsabilidade de entregar junto com o mesmo toda relevante documentação referente aos gastos acima detalhados de acordo com a regulamentação vigente da Fundação COPPETEC.		
Assinatura do Coordenador		