



**INSTITUTO SUPERIOR DE TRANSPORTES E COMUNICAÇÕES**

## **Melhoria da aplicação de gestão de projectos finais de curso no ISUTC**

**Marcel Danton de Figueiredo Saraiva**

Projecto Final do Curso

**Licenciatura em Engenharia Informática e de Telecomunicações**

Supervisor:

Engº Turay Filipe de Melo

Departamento de Tecnologias de Informação e Comunicação

Julho, 2013



**INSTITUTO SUPERIOR DE TRANSPORTES E COMUNICAÇÕES**

## **Melhoria da aplicação de gestão de projectos finais de curso no ISUTC**

**Marcel Danton de Figueiredo Saraiva**

### **Projecto Final do Curso**

**Licenciatura em Engenharia Informática e de Telecomunicações**

Supervisor:

Engº Turay Filipe de Melo

Departamento de Tecnologias de Informação e Comunicação

**Julho, 2013**



**Melhoria da aplicação de gestão de projectos finais de curso no ISUTC**

## ÍNDICE

AGRADECIMENTOS .....	IV
DEDICATÓRIA .....	V
DECLARAÇÃO DE HONRA .....	VI
ÍNDICE DE TABELAS .....	VII
ÍNDICE DE FIGURAS .....	VIII
LISTA DAS ABREVIATURAS UTILIZADAS .....	X
RESUMO .....	XI
CAPÍTULO 1 - INTRODUÇÃO.....	1
1.1 Justificação do tema.....	1
1.2 Desenho teórico .....	1
1.2.1 Problemática.....	1
1.2.2 Problema de investigação.....	2
1.2.3 Objecto de investigação .....	2
1.2.4 Objectivo geral de investigação .....	2
1.2.4.1 Objectivos específicos de investigação .....	2
1.2.5 Perguntas investigação .....	3
1.3 Metodologia.....	3
1.3.1 Abordagem da investigação .....	3
1.3.2 Desenho da investigação .....	3
1.3.3 Métodos de investigação .....	4
1.3.4 Resultados esperados de investigação.....	4
CAPÍTULO 2 - MARCO TEÓRICO-CONCEITUAL DA INVESTIGAÇÃO.....	5
2.1 Conceitos básicos .....	5
2.1.1 <i>Software</i> .....	5
2.1.2 Tipos de <i>software</i> .....	5
2.1.3 Arquitectura de <i>software</i> .....	6
2.1.4 <i>Web</i> .....	8
2.1.5 Aplicação <i>Web</i> .....	8
2.1.6 Arquitectura de aplicações <i>Web</i> .....	8
2.1.7 Sistema de gestão documental.....	11
2.1.8 Gestão de fluxos de trabalhos (Workflow) .....	12
2.2 Engenharia de <i>software</i> .....	13
2.2.1 Processo de <i>software</i> .....	13

2.2.2 Desenvolvimento ágil.....	14
2.2.4 Modelagem de <i>software</i> .....	15
2.2.5 Qualidade de <i>software</i> .....	15
2.2.6 Norma ISO 12207 .....	16
2.3 Base de dados .....	16
2.3.1 Sistema de gestão de Base de dados.....	17
2.3.2 Modelo Relacional .....	17
CAPÍTULO 3 - MARCO CONTEXTUAL DA INVESTIGAÇÃO .....	18
3.1 Introdução .....	18
3.2 ISUTC.....	18
3.1.1 Estrutura Orgânica do ISUTC .....	19
3.2 Ciclo de vida de PFC.....	20
3.3 Deficiências do actual sistema de gestão de PFC.....	21
3.4 Funcionalidades implementadas na aplicação actual .....	21
CAPÍTULO 4 - METODOLOGIA DE RESOLUÇÃO DO PROBLEMA E APRESENTAÇÃO DE RESULTADOS .....	22
4.1 Introdução .....	22
4.2. Metodologia de desenvolvimento.....	22
4.3 Ferramentas e tecnologias utilizadas .....	22
4.3.1 Ferramentas de modelação.....	22
4.3.2 Tecnologias de desenvolvimento .....	23
4.3.3 Ambientes de trabalho.....	23
4.4 Análise de Requisitos .....	25
4.4.1 Escopo da solução.....	25
4.4.2 Processo de realização do PFC.....	26
4.4.3 Casos de uso implementados .....	27
4.4.4 Casos de uso da solução proposta .....	31
4.4.5 Mapeamento dos casos de uso actuais e propostos .....	33
4.5 ORM e Hibernate .....	34
4.5.1 Implementação do hibernate .....	35
4.7 Protótipo Melhorado.....	35
4.8 Custos .....	43
CAPÍTULO 5 - CONCLUSÕES E RECOMENDACÕES .....	45
5.1 Conclusões.....	45
5.2 Recomendações .....	46

REFERÊNCIAS BIBLIOGRÁFICAS .....	47
BIBLIOGRAFIA .....	49
ANEXOS .....	50

## **AGRADECIMENTOS**

Agradeço em primeiro lugar a Deus, por ter dado-me forças para continuar e para dar este passo final na realização da minha licenciatura.

Aos meus pais Júlio Saraiva e Mária Emília Pereira por terem-me educado e ensinado muito mais do que poderia aprender em algum livro, por terem-me ensinado aspectos importantes da vida como não desistir face aos desafios. A minha família pelo suporte, compreensão e orientações de vida.

A todos os meus colegas de turma, de ano e de trabalho mas em especial aos citados abaixo:

Holden Mariquele, Ze Marra, José Laranjeira, Tuahiri Melo, Gulamo Usmane, Edson Vuma, Leontina Malaze, Yuri Wingester, Dário Mungoi, Erica Bernardo, Lucika Guambe, Anselmo Felix, Mercio Silva, Raquel Raimundo, Belzénia Matsimbe, Belchior Nhantumbo, Armando Chibassicua, Alfredo Chambal, Amandio Sousa, Nazare Dimene, Milton Langa, Cerezo Langa e Victor Esculudes.

Aos meus Professores não só do ISUTC mas também do colégio Kitabu e da Escola 3 de Fevereiro.

Aos meus amigos pela amizade e momentos marcantes.

A alguns nomes marcantes da minha vida profissional nomeadamente: Engº Mujahid Karim, Engº Ivo Abdul, Engº Valter Paruque, Engº Cameron Smith, Engº Micas Rafael, Engº Elton Sixpence e João Leopoldo.

A Filia e ao Dr. Osvaldo Pedro por terem colaborador fortemente para a realização deste trabalho.

Ao meu supervisor Engº Turay Melo pela compreensão acima de tudo, mas também por todos os ensinamentos e paciência que contribuíram bastante para a elaboração deste PFC.

E em geral a todos aqueles que de uma ou de outra forma contribuíram para o meu crescimento profissional e também como pessoa, por terem-me ajudado a chegar onde cheguem e por darem-me a dicas para continuar a crescer.

## **DEDICATÓRIA**

Dedico este trabalho a Deus por ter sido um actor essencial na história da minha vida, por ter colocado desafios quando necessário, por ter-me oferecido somente o que realmente precisei e não o que o pensei precisar, por me ter dado as forças necessárias para enfrentar todas as batalhas da minha vida.

Aos meus Pais por terem esforçado-se tanto para que eu fosse alguém na vida e para que eu enfrentasse o mínimo de dificuldades possíveis na minha vida.

Por terem zangado, negado e criticado quando necessário.

Por terem-me ensinado a fazer todo na minha vida com amor e dedicação.



## **DECLARAÇÃO DE HONRA**

Eu, Marcel Danton de Figueiredo Saraiva declaro por minha honra que o presente Projecto Final do Curso é exclusivamente de minha autoria, não constituindo cópia de nenhum trabalho realizado anteriormente e as fontes usadas para a realização do trabalho encontram-se referidas na bibliografia.

Assinatura: \_\_\_\_\_

## ÍNDICE DE TABELAS

Tabela 1.3 Fases do ciclo de vida de PFC.....	20
Tabela 2.4 Máquina Principal de desenvolvimento .....	24
Tabela 3.4 Máquina que hospeda o LDAP .....	24
Tabela 4.4 Máquina principal de pré-produção.....	25
Tabela 5.4 Requisitos funcionais implementados no protótipo inicial.....	27
Tabela 6.4 Requisitos não-funcionais iniciais.....	28
Tabela 7.4 Requisitos funcionais propostos .....	32
Tabela 8.4 Mapeamento entre os casos de uso.....	33
Tabela 9.4 Custos de desenvolvimento .....	43
Tabela 10.4 Custos de implementação .....	44

## ÍNDICE DE FIGURAS

Figura 1.2 Arquitectura cliente-servidor .....	7
Figura 2.2 Camadas/partes de uma aplicação web. ....	10
Figura 3.2 Arquitectura de uma aplicação web com exemplos .....	11
Figura 4.2 Componentes do sistema de gestão documental.....	12
Figura 5.4 Diagrama de actividades do ciclo de vida do PFC.....	26
Figura 6.4 Tela de autenticação.....	28
Figura 7.4 Tela principal .....	29
Figura 8.4 Submeter tema .....	29
Figura 9.4 Submeter PFC .....	30
Figura 10.4 Consultar PFC.....	30
Figura 11.4 Casos de uso proposto.....	31
Figura 12.4 Tela inicial .....	36
Figura 13.4 Tela de autenticação.....	36
Figura 14.4 Menu principal, item PFC seleccionado .....	37
Figura 15.4 Menu principal, item Pendentes seleccionado .....	37
Figura 16.4 Submeter versão do PFC(pre-projecto) .....	38
Figura 17.4 Submeter versão do PFC (Projecto final) .....	39
Figura 18.4 Notificação de pendente para secretária .....	39
Figura 19.4 Pendentes .....	40
Figura 20.4 Notificação de pendente para GPFC.....	40
Figura 21.4 Menu principal, item GPFC seleccionado .....	41

## Melhoria da aplicação de gestão de PFC no ISUTC

Figura 22.4 Submeter parecer .....	41
Figura 23.4 Atribuir Júri.....	42

## **LISTA DAS ABREVIATURAS UTILIZADAS**

ISUTC	Instituto Superior de Transportes e Comunicações
PFC	Projecto final de curso
GPFC	Gabinete de Projectos final de curso
SGBD	Sistema de gestão de base de dados
MIS	Management Information System
CAD	Computer aided design
MEC	Ministerio de educação e cultura
ISUTC	Instituto Superior de Transportes e Comunicações
PFC	Projecto final de curso
SGBD	Sistema de gestão de base de dados
MIS	Management Information System
CAD	Computer aided design
MEC	Ministerio de educação e cultura
ISUTC	Instituto Superior de Transportes e Comunicações
PFC	Projecto final de curso
ORM	Object-Relational Mapping

## **RESUMO**

O trabalho descrito nesta dissertação insere-se no âmbito do Projecto Final de Curso (PFC), no ramo de Eng. Informática e de Telecomunicações (LEIT).

Incide sobre o Instituto Superior de Transportes e Comunicações (ISUTC), sob proposta de Marcel Danton de Figueiredo Saraiva e sob supervisão do Engº Turay Filipe de Melo.

Em 2011, Stélio Gil dos Santos Zacarias elaborou o seu projecto final de curso com o tema Desenvolvimento de uma aplicação Web para Gestão de Projectos Finais de Curso no ISUTC. Durante a realização do seu projecto Stélio Zacarias conseguiu desenvolver um protótipo da aplicação de gestão de PFC no entanto este protótipo ainda possuía certas lacunas no que esta relacionado com a implementação de funcionalidades críticas como a notificação por email.

Ao longo deste trabalho o autor irá abordar temas importantes das ciências de programação e engenharia de software na tentativa de melhorar o protótipo e consequentemente trazer uma alcançar uma melhor solução para suprir as necessidades de gestão de PFC do ISUTC.

De modo a melhor alinhar a solução de gestão de PFC com os processos seguidos no ISUTC e com o actual ciclo de vida do PFC.

O autor tentará ainda implementar uma framework que possa trazer melhorias para o protótipo no que concerne a conexão com a base de dados.

### **PALAVRAS CHAVE:**

Melhoria da aplicação, Engenharia de software, Gestão de PFC

## CAPÍTULO 1 - INTRODUÇÃO

### 1.1 Justificação do tema

O Instituto Superior de Transportes e Comunicação designado por ISUTC é uma organização que tem como objetivo a formação de quadros superiores, a formação destes quadros superiores geralmente termina com a elaboração e defesa do projeto final de curso designado por PFC. A cada ano o ISUTC tem a necessidade de gerir diversos PFCs dos estudantes finalistas, esta gestão resume-se em gerir o ciclo de vida dos PFCs, ciclo este que está composto por diversas fases que incluem: a entrega em partes do PFC pelo estudante, a avaliação do PFC por parte do Gabinete de PFC designado por GPFC e pelos departamentos técnicos.

O actual processo de gestão de PFC no ISUTC é realizada a partir de um sistema de informação manual, a informatização deste sistema iria aumentar a produtividade, melhorar o cumprimento de prazos e a interação com os estudantes.

Este sistema de informação automatizado poderia:

Centralizar a informação do processo de gestão de PFC

Controlar o ciclo de vida de PFC;

A importância da temática “Melhoria da aplicação Web para gestão de PFC no ISUTC” reside no facto de melhorar o protótipo desenvolvido por Stélio Gil dos Santos Zacarias em 2011 (1) de forma a melhor suprir as necessidades do ISUTC principalmente no que concerne a gestão do ciclo de vida de PFC. Pretende-se ainda melhorar a performance e portabilidade do protótipo com a utilização de *frameworks*, e colocar o sistema de notificação por *email* em funcionamento.

### 1.2 Desenho teórico

#### 1.2.1 Problemática

O actual sistema de gestão de PFC em funcionamento no ISUTC é um sistema manual adaptado, onde a maioria dos processos são manuais no entanto há utilização de algumas ferramentas computadorizadas como planilhas de cálculos, editores de texto para auxiliar estes processos.

---

1 - “Framework é um conjunto de classes que colaboram para realizar uma responsabilidade para um domínio de um subsistema da aplicação” - FAYAD e SCHMIDT

Este sistema possui diversos arquivos de dados que armazenam a mesma informação no entanto durante o processo manual de sincronização de informação pode ocorrer problemas que comprometam a integridade dos dados daí que este sistema careça de uma base de dados centralizada.

Durante “Desenvolvimento de uma aplicação *Web* para gestão de projectos final de curso no ISUTC” Stélio Zacarias desenvolveu um protótipo funcional com o intuito de resolver este problema no entanto esta aplicação carece de testes em ambiente de produção no que está relacionado a gestão do ciclo de vida dos PFC e de reestruturação baseada nas críticas do ISUTC.

O protótipo desenvolvido carece ainda da incorporação do sistema de notificação por *email*.

### **1.2.2 Problema de investigação**

Como reestruturar a aplicação *Web* de gestão de PFC para auxiliar devidamente a gestão do ciclo de vida de PFC.

### **1.2.3 Objecto de investigação**

Processo de melhoria de uma aplicação *Web* para gestão de PFC no ISUTC.

### **1.2.4 Objectivo geral de investigação**

Melhorar a aplicação de gestão de Projecto final de curso para o Instituto Superior de Transportes e Comunicações.

#### **1.2.4.1 Objectivos específicos de investigação**

Com o presente projecto pretende-se:

- Analisar o actual sistema de gestão de PFC;
- Identificar e eliminar as lacunas da aplicação *Web* de gestão de Projectos finais de Curso no que está relacionado com a gestão de ciclo de vida de PFC;
- Implementar a integração com correio eletrónico de forma a habilitar as notificações;



- Implementar uma *framework* de persistência de forma a melhorar a aplicação;

### 1.2.5 Perguntas investigação

As perguntas de investigação para o seguinte projecto seriam:

- Como identificar lacunas em um sistema de informação?
- Como melhorar um sistema de gestão de PFC?
- Que norma utilizar na reengenharia de uma aplicação *Web*?
- Como incorporar o sistema de notificação por *email* ?
- Como integrar uma *framework* de aplicação a uma aplicação *Web*?

## 1.3 Metodologia

### 1.3.1 Abordagem da investigação

A abordagem metodológica a ser seguida na elaboração deste projecto, vai ser uma pesquisa bibliográfica e de campo, em que se irá identificar os problemas a melhorar no campo e posteriormente após fruto de uma pesquisa bibliográfica com vista a identificar ferramentas adequadas para resolver o problema irá seguir-se uma metodologia experimental na reestruturação da aplicação.

### 1.3.2 Desenho da investigação

O desenho da investigação a ser seguido no presente trabalho, será composto por diversas fases.

Primeiramente far-se-á uma investigação bibliográfica e de campo com objectivo de efectuar a consulta e analisar de maneira concisa a informação existente.

Posto isto, seguir-se-á a fase de elaboração do marco teórico, o qual serão abordados os principais conceitos teóricos que contribuirão na percepção do objecto de estudo e das ferramentas para resolução do problema em questão.

---

3 - Persistência é o termo associado a uma ação que consiste em manter em meio físico recuperável, como banco de dados ou arquivo, de modo a garantir a permanência das informações de um determinado estado de um objeto lógico.

De seguida far-se-á uma análise do sistema actual e da aplicação Web desenvolvida de forma a identificar os problemas e lacunas relacionadas com a gestão do ciclo de vida dos Projectos finais de curso.

Posto isto, procurar-se-á reestruturar a aplicação Web desenvolvida de forma a melhor adequa-la a gestão do ciclo de vida dos Projectos finais de curso.

Por fim, será elaborado o relatório final, onde serão apresentadas as conclusões e possíveis recomendações.

### **1.3.3 Métodos de investigação**

A metodologia de investigação usada será empírica, usando os seguintes as etapas:

- Entrevistas aos profissionais da área, no ISUTC;
- Consultas de material bibliográfico relacionado com a matéria;
- Análise do protótipo da aplicação
- Utilização de ferramentas de desenvolvimento de aplicação.

### **1.3.4 Resultados esperados de investigação**

- Com este projecto pretende-se:
- Melhorar o protótipo da aplicação de gestão de PFC de forma a torna-lo mais adequada para a gestão do ciclo de vida de PFC e melhorar a sua performance e qualidade através da utilização de um *framework* na camada de persistência<sup>3</sup> da aplicação;
- Consolidar os conhecimentos pessoais de desenvolvimento de aplicações *Web*.

## **CAPÍTULO 2 - MARCO TEÓRICO-CONCEITUAL DA INVESTIGAÇÃO**

### **2.1 Conceitos básicos**

#### **2.1.1 *Software***

Segundo PRESSMAN (1995) “ Software é: instruções (programas de computador) que, quando executadas, produzem a função e o desempenho desejados; estruturas de dados que possibilitam que os programas manipulem adequadamente a informação; e documentos que descrevem a operação e o uso dos programas. ”

Segundo SOMMERVILLE (2003) “*Software* não é apenas o programa mas também toda a documentação associada e os dados de configuração necessários para fazer com que esses programas operem correctamente. Um sistema de *software*, usualmente, consiste em uma série de programas separados, arquivos de configuração que são utilizados para configurar esses programas, documentação do sistema, que descreve a estrutura desse sistema, e documentação do usuário, que explica como utilizar o sistema e, no caso dos produtos de *software*, sites *Web* para os usuários fazerem o download das informações recentes sobre o produto.”

É importante definir-se o conceito de *software* visto que o grande fulcro deste projecto será exactamente o software de gestão de projectos finais de curso. Em resumo pode-se afirmar que um software é conjunto de instruções e elementos auxiliares que definem a forma de realizar uma determinada função.

#### **2.1.2 Tipos de *software***

Segundo PRESSMAN (1995) “*Desenvolver categorias genéricas para as aplicações de software é uma tarefa um tanto difícil. À medida que a complexidade do software cresce, desaparece a clara divisão em compartimentos. As seguintes áreas de software indicam a amplitude das aplicações potenciais:*

- *Software básico;*
- *Software em tempo real;*
- *Software embutido;*
- *Software científico e de engenharia;*
- *Software educativo;*
- *Software de inteligência artificial;*
- *Software de gestão empresarial;*
- *Software SIG – Sistemas de informações gerenciais;*
- *Software SAD – Sistemas de apoio a decisões;*
- *Software EIS – Sistemas de informações executivas;”*

### **2.1.3 Arquitectura de *software***

Uma das definições mais clássicas da arquitectura de *software* define o que é o sistema em termos de componentes e o relacionamento entre os componentes. (GARLAN & SHAW,1994)

Semelhante a mesma, BASS et al. (2003) refere-se a arquitectura de *software* como estruturas que incluem componentes, suas propriedades externas e os relacionamentos entre eles, constituindo uma abstracção do sistema.

Para VAROTO (2002) “a arquitetura de *software* é colocada como uma ferramenta para lidar com a complexidade do *software* e enfatizam que arquitetura deve satisfazer os requisitos funcionais e não funcionais do sistema.”

Efectuar uma boa arquitectura no processo de desenvolvimento de *software* é extremamente importante.

Como a engenharia de requisitos, a arquitectura de *software* também define um processo.

“O processo de arquitectura de *software* compreende as actividades complementares que são necessárias para a construção do software, entendendo que tais actividades servirão para realizar o projecto (*design*), implementar e evoluir o sistema. O produto arquitectural

representa o resultado final das actividades que acontecem em cada etapa do processo de definição de uma arquitectura de *software*.” (VAROTO,2002)

Uma boa arquitectura será aquela capaz de corresponder a todos os requisitos funcionais e não funcionais do sistema, isto só será possível se existir um entendimento claro do real ponto de operação da aplicação.

Para que a aplicação possa responder a todas as necessidades dos utilizadores ou do utente a quem destina-se a mesma, é necessário realizar-se da melhor maneira os processos da engenharia de requisitos aliados a arquitectura de *software* para que posteriormente possa ser efectuada a modelação da aplicação, recorrendo a técnicas de modelação UML.

### 2.1.3.1 Arquitectura cliente-servidor

Segundo COULOURIS et al (2007)“é a arquitectura mais citada quando os sistemas distribuídos são discutidos. Historicamente ela é a mais importante e continua sendo amplamente empregada. A figura abaixo ilustra a estrutura simples na qual os processos clientes interagem com processos servidores, localizados em distintos computadores hospedeiros, para acessar os recursos compartilhados, que estes gerenciam.

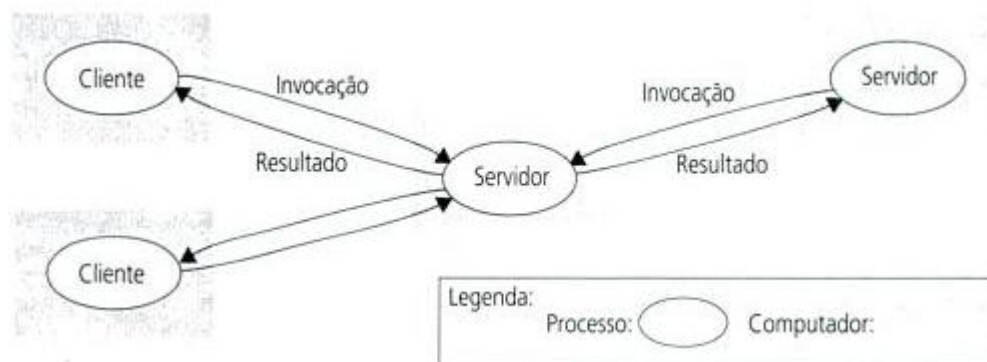


Figura 1.2 Arquitectura cliente-servidor

Fonte: George Coulouris, *Sistemas distribuídos: Conceitos e Projecto*

Os servidores podem, por sua vez, ser clientes de outros servidores, conforme a figura indica. Por exemplo, um servidor web é frequentemente um cliente de um servidor de arquivos local que gerência os arquivos nos quais as páginas *web* estão armazenadas. Os servidores web, e a maioria dos outros serviços Internet, são clientes do serviço DNS, que mapeia nomes de domínio Internet a endereços de rede (IP). Outro exemplo relacionado à web diz respeito aos

mecanismos de busca, os quais permitem aos usuários pesquisar resumos de informações disponíveis em páginas web, em *sites* de toda a Internet. Esses resumos de informações disponíveis em páginas web, em *sites* de toda a Internet.”

#### **2.1.4 Web**

Segundo SHELLY et al. (2011) *Web* ou *World Wide Web* é uma parte da Internet que suporta multimédia e que consiste em uma coleção de documentos conectados. Para suportar a multimédia a Web utiliza o protocolo HTTP (*Hypertext Transfer Protocol*), que consiste em um conjunto de regras para a troca de textos, gráficos, som, vídeo e os outros ficheiros multimédia. Os documentos conectados ou páginas de informação na *Web* são também designados por *Web Pages* ou página *Web*. Devido ao suporte a texto, gráficos, som, e vídeo, uma página *Web* pode incluir qualquer um destes elementos de multimédia. A *Web* está em constante mudança e consiste de bilhões de páginas *Web*.

#### **2.1.5 Aplicação Web**

Segundo SHKLA e ROSEN (2003), uma aplicação web é algo mais que um simples *website*.

É uma aplicação cliente servidor que usa um Web Browser como o seu programa cliente, e que realiza um serviço interativo através da conexão com os servidores através da Internet ou Intranet.

Um Website simplesmente apresenta conteúdo de ficheiros estáticos. Uma aplicação Web apresenta conteúdo dinâmico adaptado com base nos parâmetros do pedido, acções do utilizador e considerações de segurança.

#### **2.1.6 Arquitectura de aplicações Web**

Segundo BARISH (2002), antes de falar-se de arquitectura de aplicações *Web* é necessário perceber-se alguns conceitos.

Uma transação em uma aplicação *Web* é um diálogo composto por pedidos e respostas que corresponde a um único comportamento lógico da aplicação, isto é, o pedido realizado pelo

cliente leva a invocação da lógica da aplicação do lado do servidor que irá eventualmente responder ao cliente.

Existe uma diferença entre simples pedidos e transações, um clique em um *link* de uma página *Web* estática é um pedido que não necessita de lógica da aplicação para obter a resposta e consequentemente não é considerada uma transação.

Uma sessão consiste no uso da aplicação por um cliente durante um período de tempo. Sessões são compostas por uma ou várias transações definidas. Da mesma forma transações correspondem a séries de operações na aplicação, sessões correspondem a séries de transações. No entanto diferentemente das transações, uma sessão pode consistir numa lista de transações sem relação lógica.

Toda aplicação tem requisitos que especificam as funcionalidades que ela deve suportar. Aplicações *Web* não são diferentes neste aspecto, no entanto também devemos observar para dois aspectos que são a interface e a gestão de dados.

Os requisitos da lógica do negócio são os mais importantes de qualquer aplicação *Web*. Em resumo se os requisitos da lógica do negócio não são cumpridos a aplicação tem pouco valor.

No que refere aos requisitos da lógica do negócio existem algumas observações gerais que devem ser seguidas.

Primeiramente, o código da aplicação que corresponde directamente a práticas manuais do negócio é tipicamente designado por lógica do negócio. Assim sendo a lógica de negócio é um conjunto de operações necessárias para providenciar um serviço. Na sua forma mais básica, esta lógica é um conjunto de funções, cada composta por uma sequência de paços com um propósito.

Em segundo, a lógica do negócio deve ser dinâmica e customizável, deve existir a possibilidade de substitui-la ou altera-la sem reconstruir toda a aplicação. O facto da lógica do negócio ser customizável pode exigir que ela ou os seus parâmetros sejam armazenados em uma base de dados.

Quanto a gestão de dados ou persistência, podemos considerar que a lógica do negócio são tubos de uma aplicação *Web* e os dados associados a aplicação representam a água que passa pelos tubos.

Os dados na verdade não fazem parte da lógica do negócio, no entanto a sua existência dá sentido à lógica do negócio e evidencia o seu valor.

A gestão dos dados está relacionada com a confiabilidade, segurança e acesso eficiente dos dados. É desejável a capacidade de armazenar grandes volumes de dados, acessá-los rapidamente, e de relacionar uns com os outros. Base de dados são os mecanismos primários usados para garantir os requisitos de gestão de dados. Elas permitem a modelação de dados (representação) e a sua persistência (confiabilidade); suportam transações de dados (integridade e ordenação); oferecem segurança; e geralmente são rápidas (eficiência).

Utilizadores acessam a aplicação através de *Web Browsers*, telefones normais, telefones celular, PDAs. Tipicamente todas as funcionalidades da aplicação estão acessíveis a partir dos *Web Browsers*, no entanto apenas partes específicas da aplicação podem estar disponíveis em outros terminais.



Figura 2.2 Camadas/partes de uma aplicação web.

Fonte: <http://flavioaf.wordpress.com/2010/02/25/tutorial-entendendo-java-para-web-parte-1/>



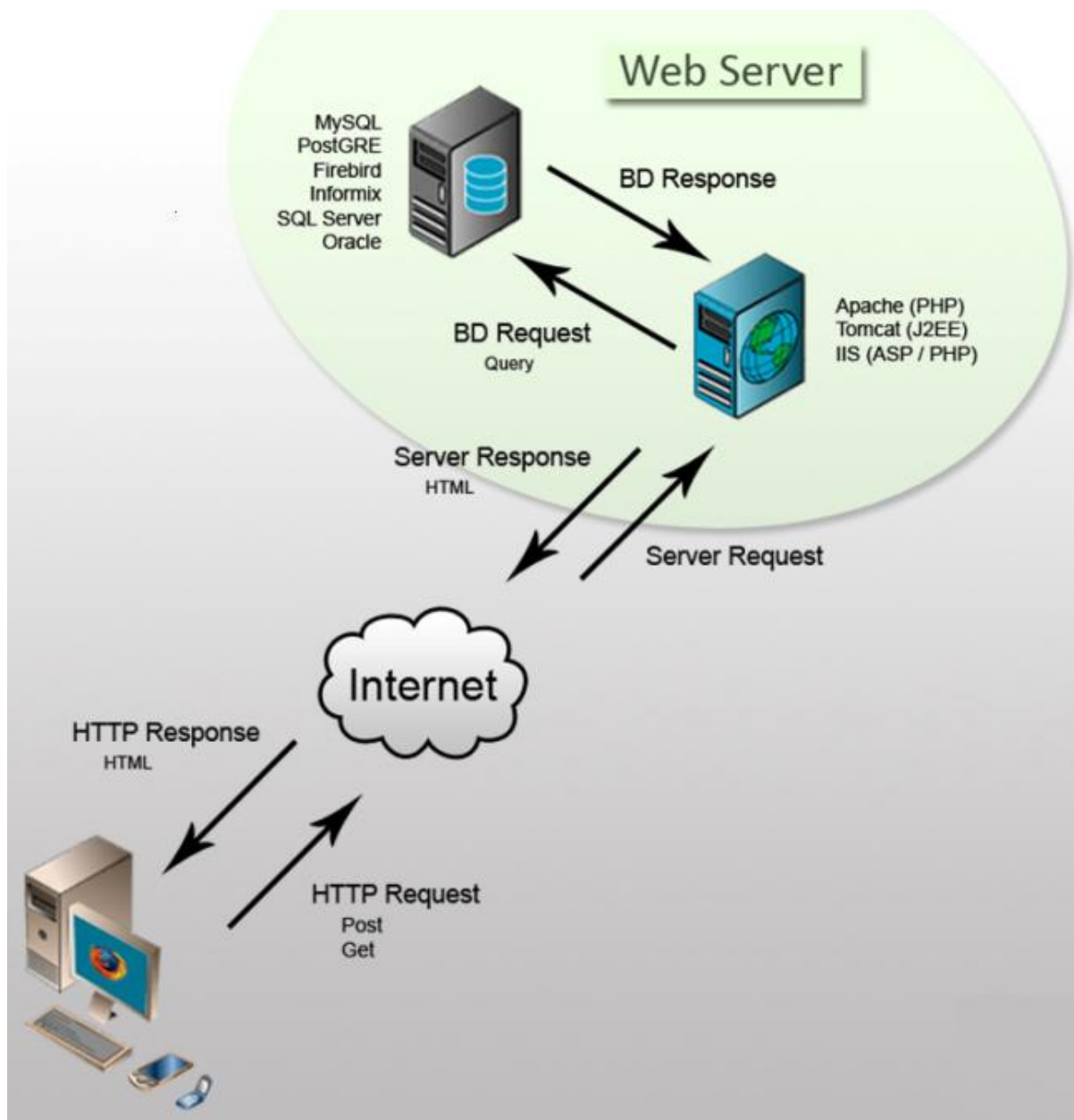


Figura 3.2 Arquitectura de uma aplicação web com exemplos

Fonte: <http://flavioaf.wordpress.com/2010/02/25/tutorial-entendendo-java-para-web-parte-1/>

### 2.1.7 Sistema de gestão documental

Segundo Wikipedia (2013) “Sistema de gestão documental é um sistema de computador ou um conjunto de programas de computador usado para controlar e armazenar documentos electrónicos. É geralmente dotado da capacidade de manter controlo das diferentes versões modificados por diferentes utilizadores (controle de histórico). O termo possui alguns conceitos em comum com sistemas de gestão de conteúdo. É geralmente visto como um componente de sistemas de gestão empresarial de conteúdos.”

Segundo ADAM (2007) “A maioria dos sistemas comerciais de gestão eletrónica de documentos, possuem certas funcionalidades que tem relação com áreas como fluxos de trabalho, colaboração, gestão de registos, arquivação. Isto acontece porque os vendedores de

*software* tem a tendência de direcionar os seus produtos para uma audiência que geralmente incorpora outras funcionalidades que são necessárias em paralelo com as funcionalidades principais dos sistemas de gestão documental.”

Document Repository
Integration with Desktop Applications
Check-In and Check-Out
Versioning
Auditing
Security
Classification and Indexing
Search and Retrieval

Figura 4.2 Componentes do sistema de gestão documental

Fonte: Azad Adam, *Implementing Electronic Document and Record Management Systems*

### 2.1.8 Gestão de fluxos de trabalhos (Workflow)

Segundo CRUZ (2004) “Ferramenta que tem por finalidade automatizar processos, racionalizando-os e consequentemente aumentando sua produtividade por meio de dois componentes implícitos: organização e tecnologia.”

Outras definições são:

*Workflow* é a tecnologia que ajuda a automatizar as políticas e procedimentos numa organização.

*Workflow* é um conjunto de ferramentas que possibilita análise proactiva compressão e automação de actividades e tarefas baseadas em informação. ”

## **2.2 Engenharia de *software***

Engenharia de *software* é um conceito importantíssimo no contexto de desenvolvimento e manutenção de software.

Segundo REZENDE (2005) “Engenharia de *software* é a metodologia de desenvolvimento e manutenção de sistemas modulares, com as seguintes características: processo (roteiro) dinâmico, integrado e inteligente de soluções tecnológicas; adequação aos requisitos funcionais do negócio do cliente e seus respectivos procedimentos pertinentes; efetivação de padrões de qualidade, produtividade e efetividade em suas atividades e produtos; fundamentação na tecnologia da informação disponível, viável, oportuna e personalizada; planejamento e gestão de atividades, recursos, custos e datas. ”

Para SOMMERVILLE (2003) a engenharia de *software* envolve questões técnicas e não-técnicas, tais como a especificação do conhecimento, técnicas de projecto e implementação, conhecimento, técnicas de projecto e implementação, conhecimentos dos factores humanos pelo engenheiro de software e ainda, gestão de projectos.

### **2.2.1 Processo de software**

Segundo SOMMERVILLE (2003) “ Um processo de *software* é um conjunto de actividades e resultados associados que levam à produção de um produto de software. Esse processo pode envolver o desenvolvimento de *software* desde o início, embora, cada vez mais, ocorra o caso de um *software* novo ser desenvolvido mediante a expansão e a modificação de sistemas já existentes. ”

Segundo PRESSMAN (1995) “Os métodos de engenharia de software proporcionam os detalhes de como fazer para construir o *software*. Os métodos envolvem um amplo conjunto de tarefas que incluem: planeamento e estimativa de projecto, análise de requisitos de *software* e de sistemas, projecto da estrutura de dados, arquitectura de programa e algoritmo de processamento, codificação, teste e manutenção. Os métodos da engenharia de *software* muitas vezes introduzem uma notação gráfica ou orientada à linguagem especial e introduzem um conjunto de critérios para a qualidade do *software*. ”

### 2.2.2 Desenvolvimento ágil

Segundo AMBLER (2002) “A Modelagem Ágil (MA) é uma metodologia baseada na prática para modelagem e documentação eficazes de sistemas baseados em software. A metodologia MA é um conjunto de práticas guiadas por princípios e valores para profissionais de software aplicarem em seu dia a dia. A MA não é um processo prescritivo. Em outras palavras, não define procedimentos detalhados sobre como criar um determinado tipo de modelo. Em vez disso, fornece conselhos sobre como ser um modelador eficiente. A MA(Hocl, 1999) mistura o *caos* de praticas simples de modelagem com a ordem inerente a artefactos de modelagem de software. A MA não significa menos modelagem; na verdade, muitos desenvolvedores acharão que estão fazendo mais modelagem MA do que antes. Pense na MA como uma arte, não uma ciência.

Por que queremos ser eficientes na modelagem? Porque ela é uma parte importante de qualquer processo de desenvolvimento. Processos ágeis como a Programação eXtrema (XP – *eXtreme Programming*) (Beck, 2000), SCRUM (Beedle e Schwaber, 2001), Método dinâmico de desenvolvimento de sistemas (*dynamic system development model*) (stapleton, 1997) incluem atividades de modelagem. Sim, até a XP inclui técnicas de modelagem, como histórias de usuários, modelos CRC (*Class Responsibility Collaborator*) e esboços. Ao contrário do que os detratores da XP dirão, está não abandona a modelagem. Ao contrário, ela minimiza o trabalho de modelagem, usando uma perspectiva de projecto na qual você desenvolve seus testes antes de desenvolver o código. Isso o força a pensar em como você construirá o *software* antes de realmente fazê-lo, exactamente como a modelagem tradicional. A XP cumpre alguns dos objectivos da modelagem, compreendendo o que você está construindo e, por isso, requer menos modelagem. Não há nada errado nisso. Os processos prescritivos de *software* também incluem atividades de modelagem. No caso do processo unificado, três das seis disciplinas básicas de processo focalizam a modelagem, e meu próprio OOSP tem uma etapa de projecto chamada simplesmente de Modelo.”

### 2.2.3 Técnicas de Orientação a Objecto (OO)

Segundo MARTIN & ODELL (1996), as técnicas OO mudam a visão que os analistas têm do mundo. Em vez de pensarem em processos e na sua decomposição, eles pensam em objectos e no comportamento destes. O objecto internamente pode ser complexo, porém o analista não

precisa entender esta complexidade e sim, saber como se comporta tal objecto e como utilizá-lo.

Segundo RUMBAUGH (1994), a modelagem e o projecto baseado em objectos são um novo modo de estudar problemas com utilização de modelos fundamentados em conceitos do mundo real. A estrutura básica é o objecto, que combina a estrutura e o comportamento dos dados em uma única entidade. Os modelos baseados em objectos são úteis para a compreensão de problemas, para a comunicação com os peritos em aplicações, para modelar empresas, preparar documentação e projectar programas e bancos de dados.

As vantagens da OO são:

- Maior facilidade para reutilização de códigos e por consequência do projecto;
- Utilização de um padrão único durante todo o processo de criação do *software*;
- Maior adequação à arquitectura cliente /servidor;
- Ciclo de vida mais longo para os sistemas;
- Menor custo para o desenvolvimento e manutenção de sistemas.

#### **2.2.4 Modelagem de *software***

Segundo LOBO (2008) “Na criação de produtos são utilizados modelos, normalmente com o uso de símbolos e desenhos técnicos, que tem o objetivo de representar todas as suas características relevantes. Em cada engenharia, existe uma forma técnica de utilizar símbolos e diagramas para criar modelos da peça, do produto ou do sistema a ser construído. Na modelagem de *software* não seria diferente, através de diagramas são construídos modelos para possibilitar a comunicação entre as equipes, além de permitir materiais (documentação/diagramas) necessários para as fases posteriores do projeto, principalmente, a fase de codificação.

Para exemplificar alguns modelos de *software* mais utilizados ao longo dos anos, podemos citar os fluxogramas na programação estruturada, os diagramas de fluxo de dados na análise estruturada e a atual UML – Linguagem de Modelagem Unificada para modelagem de sistemas com orientação a objetos. ”

#### **2.2.5 Qualidade de *software***

A qualidade de *software* é um ponto bastante crítico na engenharia de software.

Segundo VENECIAN (2013), um *software* de qualidade é:

- Fácil de usar;
- Funciona corretamente;
- É de fácil manutenção;
- E mantém a integridade dos dados mesmo havendo falhas;

Então o que é a qualidade de *software* ?

Para VENECIAN (2013) qualidade de *software* é um conjunto de características a serem satisfeitas em um determinado grau, de modo que o software satisfaça às necessidades de seus usuários.

Existem vários aspetos a tomar em consideração para garantir a qualidade de *software*, um destes aspetos é o uso de normas internacionais a quando do desenvolvimento do *software*.

### **2.2.6 Norma ISO 12207**

Para VENECIAN (2013) a ISO 12207 é uma norma internacional que descreve em detalhes os processos, actividades e tarefas que envolvem o fornecimento, desenvolvimento, operação e manutenção de *software*.

Esta norma prevê uma serie de processos que segundo VENECIAN (2013) são:

- Processos fundamentais: inicio e execução do desenvolvimento, operação ou manutenção do *software* durante o seu ciclo de vida.
- Processos de apoio: auxiliam e contribuem para o sucesso e a qualidade do projecto de *software*.

Processos organizacionais: implementam uma estrutura constituída de processos de ciclo de vida e pessoal associado, melhorando continuamente a estrutura e os processos.

## **2.3 Base de dados**

Base de dados é um conceito importantíssimo no contexto das tecnologias de informação e comunicação e ainda mais importante no contexto de *software*.

Segundo Ramakrishnan (2008) “*Base de dados é uma coleção de dados que, tipicamente, descreve as actividades de uma ou mais organizações relacionadas.*”

### **2.3.1 Sistema de gestão de Base de dados**

Os bancos de dados por si só já possuem uma organização dos dados no entanto existe também a necessidades de utilizar sistemas adequados para a manipulação destes dados.

Segundo Ramakrishnan (2008) *“Um sistema de gerenciamento de banco de dados, ou SGBD, é um software projectado para auxiliar a manutenção e utilização de vastos conjuntos de dados. A necessidade de tais sistemas, assim como seu uso, tem crescido rapidamente. A alternativa para não usar um SGBD é armazenar os dados em arquivos e escrever código específico do aplicativo para gerenciá-los. O uso de um SGBD tem diversas vantagens importantes.”*

### **2.3.2 Modelo Relacional**

O modelo de dados relacional é o modelo mais usado na actualidade, mas antes de falar deste modelo existe uma necessidade de descrever o que são modelos de dados.

Segundo Ramakrishnan (2008) *“Um modelo de dados é uma coleção de construtores de alto nível para descrição dos dados que ocultam vários detalhes de baixo nível do armazenamento. Um SGBD possibilita que um usuário defina os dados a serem armazenados em termos de modelo de dados. A maioria dos sistemas de gerenciamento de banco de dados actuais baseia-se no modelo de dados relacional.”*

Segundo Costa (2007) *“Em um SGBD Relacional, os dados são armazenados em tabelas. As tabelas são materializações das relações. Dizemos que as relações possuem atributos e tuplas, e as tabelas possuem colunas e linhas.”*

## **CAPÍTULO 3 - MARCO CONTEXTUAL DA INVESTIGAÇÃO**

### **3.1 Introdução**

O ISUTC é uma IES privada, em actividade desde 2000, a 8ª mais antiga no país.

Privilegia, como áreas de actuação, as temáticas dos Transportes e Comunicações e suas envolventes. Uma escola eminentemente de Engenharia, que atribui também prioridade à área de Gestão, pelo papel que esta desempenha, em intervenção horizontal, em todos os sectores, segundo ISUTC (2013).

Ainda segundo ISUCT (2013) no último ano da Licenciatura todos os estudantes devem realizar um Projecto Final de Curso - PFC, que poderá ser, ou não, a última actividade curricular a realizar.

### **3.2 ISUTC**

O ISUTC foi instituído pela Transcom, SA. Arrancou com as Licenciaturas no ano lectivo 2000-01, precedido de um Semestre Zero no início de 2000.

Em 2000 arrancaram as Licenciaturas em Eng<sup>a</sup> Informática e de Telecomunicações - LEIT e em Gestão e Finanças - LGF. Em 2001 arrancou a Licenciatura em Eng<sup>a</sup> Civil e de Transportes - LECT e, em 2006 a de Eng<sup>a</sup> Mecânica e de Transportes - LEMT.

O ISUTC assume como sua Missão "... a criação e difusão da ciência, da cultura e da tecnologia, exercidas nos domínios do estudo, da docência, da investigação e da prestação de serviços, em harmonia com os desígnios da identidade nacional e do desenvolvimento da comunidade nacional e internacional."

O ISUTC é vocacionado para a formação de quadros superiores nas áreas de conhecimento ligadas aos transportes e comunicações e suas envolventes, incluindo as tecnologias, a gestão e a economia, em particular as tecnologias da informação e comunicação, os transportes, a logística e distribuição, a gestão.

Extraindo dos seus estatutos (Decreto já citado), o ISUTC promove:

- A formação de técnicos e cientistas ao nível da graduação e pós-graduação nas áreas da sua vocação;



- A investigação científica (...) incidindo principalmente na resolução de problemas relevantes para a sociedade moçambicana;
- A ligação estreita à actividade económica, organizacional e social (...)
- O intercâmbio científico, tecnológico e cultural com instituições nacionais e estrangeiras (...)

A **Visão** do ISUTC é a de “... tornar-se uma instituição de ensino superior líder, no país, nas áreas em que realiza formação e investigação e presta serviços, com particular destaque para as da engenharia e da gestão, ocupando a posição de melhor escola superior de engenharia do país”.

### **3.1.1 Estrutura Orgânica do ISUTC**

Órgãos da actual Estrutura Orgânica do ISUTC:

- Reitoria;
- Conselho Universitário;
- Direcção de Programas de Graduação;
- Direcção de Programas de Pós-Graduação;
- Direcção de Programas de Formação Contínua;
- Laboratório Informático de Métodos de Ensino-Aprendizagem-Avaliação – LIMEAA;
- Sector de Informática;
- Serviços Administrativos;

O gabinete de projectos finais de curso ou GPFC que é responsável pela gestão dos PFC é pertence a direcção de programas de graduação ou DPG.

### 3.2 Ciclo de vida de PFC

O ciclo de vida do PFC é composto pelas seguintes fases:

Tabela 1.3 Fases do ciclo de vida de PFC

Fonte: Auto, 2013

Fase	Descrição	Actores
1.	Inscrição na cadeira de introdução ao PFC	Estudante Secretaria académica
2.	Entrega do pré-projecto ao docente de Intro. Ao PFC	Estudante Docente de Intro. ao PFC
2.1	Avaliação do tema e dos pontos cruciais do pré-projecto	Docente de Intro. ao PFC
3.	Entrega dos capítulos 1 e 2 do PFC ao GPFC através da secretaria académica	Secretaria académica GPFC
3.1	Avaliação dos capítulos pelo GPFC e pelos departamentos técnicos	GPFC Departamento técnico
4.	Entrega do PFC ao GPFC através da secretaria académica	Secretaria académica GPFC
4.1	Avaliação do PFC pelos departamentos técnicos	GPFC Departamento técnico
4.2	Proposta do júri	GPFC Departamento técnico
4.3	Alocação do oponente e avaliação do PFC por este	GPFC

		Oponente
5.	Marcação da defesa	GPFC

### 3.3 Deficiências do actual sistema de gestão de PFC

O actual sistema de gestão de PFC apresenta as seguintes deficiências:

- Falta de segurança no que esta relacionado com a integridade da informação sobre a gestão de PFC, isto pois a informação é armazenada em ficheiros excel e word que são mais susceptíveis a alteração de dados por parte de pessoal não autorizado;
- Falta de segurança no armazenamento dos PFCs, os PFCs são somente armazenados fisicamente e uma perda das cópias físicas;
- Falta de mecanismos de alerta que melhorariam o controlo dos estados do PFC como por exemplo a verificação de estudantes sem supervisor;
- Não armazenamento de cópias dos PFCs em formato eletrónico que melhoraria a disponibilidade dos PFCs.

### 3.4 Funcionalidades implementadas na aplicação actual

Durante a elaboração do seu PFC Stelio Zacarias desenvolveu o protótipo da aplicação de gestão de PFCs, neste protótipo foram implementadas as seguintes funcionalidades:

- Submeter, actualizar, consultar e apagar temas;
- Submeter, actualizar, consultar e apagar PFCs;
- Verificar os alunos supervisionados por um supervisor;

## **CAPÍTULO 4 - METODOLOGIA DE RESOLUÇÃO DO PROBLEMA E APRESENTAÇÃO DE RESULTADOS**

### **4.1 Introdução**

Neste capítulo serão apresentados os resultados desta investigação bem como as teorias e ferramentas utilizados para alcançar os resultados.

### **4.2. Metodologia de desenvolvimento**

Durante a resolução deste problema que consistiu em melhorar o protótipo da aplicação de gestão de PFC foi utilizada a metodologia de desenvolvimento ágil em parceria com a linguagem de modelação *UML*.

A metodologia de desenvolvimento ágil foi optada devido ao foco desta metodologia na comunicação entre a equipe de desenvolvimento e a parte interessada ou *stakeholders*. Outro motivo que levou a escolha é o tamanho de equipa de desenvolvimento que neste projecto foi composta por apenas um elemento, e visto que a metodologia de desenvolvimento ágil é a mais adequada para equipas pequenas ou medias esta tornou-se uma das mais adequadas.

Para auxiliar a modelação da solução foi utilizada a linguagem de modelação UML, mais especificamente os seguintes diagramas:

- Diagrama de actividades;
- Diagrama de caso de uso;

Os dois diagramas forão utilizados durante a fase de levantamento de requisitos.

### **4.3 Ferramentas e tecnologias utilizadas**

Ao longo do trabalho foram utilizadas diversas ferramentas para auxiliar a investigação

#### **4.3.1 Ferramentas de modelação**

Para a modelação UML, isto é, mais em concreto para criação dos diagramas UML foi utilizada a ferramenta Visual Paradigm for UML Community Edition na versão 10.2 (VP-UML).

VP-UML é uma ferramenta UML CASE que permite criar diagramas UML, realizar a modelação de dados, de entre outras funções relacionados com o levantamento de requisitos e modelação.

Apesar do VP-UML possuir funções necessárias para a modelação de dados optou-se por utilizar o MySQL Workbench na versão 5.2 para a modelação de dados devido a integração

com o SGBD MySQL o que aceleraria a integração na alteração da base de dados através das mudanças realizadas nos diagramas.

### 4.3.2 Tecnologias de desenvolvimento

O protótipo desenvolvido por Stélio Zacarias foi desenvolvido com base nas seguintes tecnologias:

- Linguagem de programação de alto nível, orientada a objectos – Java na versão 1.6;
- SGBD relacional - MySQL na versão 5.0;
- Servidor de aplicação Web - Apache Tomcat na versão 5.5;
- Framework de aplicação – ZK na versão 5.0.9;
- Framework de aplicação – Spring na versão 2.5;
- Framework de aplicação – Spring Security na versão 2.0;
- Ambiente de desenvolvimento integrado – Eclipse na versão 3.2

O autor optou por manter e a maioria das tecnologias optando apenas por fazer o *upgrade* das seguintes ferramentas:

- MySQL para a versão 5.1;
- Apache Tomcat para a versão 6.0;
- Eclipse para a versão 4.2;

O autor optou ainda por acrescentar a seguinte ferramenta:

- Framework de aplicação - *Hibernate*;

Posteriormente serão apresentadas as razões para acrescentar esta tecnologia.

### 4.3.3 Ambientes de trabalho

Ao longo da investigação foram utilizados dois ambientes de trabalho principais nomeadamente o ambiente de desenvolvimento e o ambiente de pré-produção.

#### 4.3.3.1 Ambiente de desenvolvimento

O ambiente de desenvolvimento utilizado ao longo da realização do projecto é composto por duas máquinas com as seguintes características:

Tabela 2.4 Máquina Principal de desenvolvimento

Fonte: Auto, 2013

<b>Máquina principal de desenvolvimento</b>	
Tipo de Máquina	Física
Hardware	Processador: Intel Core I5-2400 3.1 Ghz, 4 núcleos Memória: 8 GB Espaço em disco: 500 GB
Sistema Operativo	Windows 7 64 bits
Principais softwares	VMware Workstation 9 JDK 1.6 Eclipse 3.2 GitHub 1.0.52 VP-UML 10.2 MySQL 5.1 Apache Tomcat 6.0

Tabela 3.4 Máquina que hospeda o LDAP

Fonte: Autor, 2013

<b>Máquina que hospeda o LDAP</b>	
Tipo de Máquina	Virtual
Hardware	Processador: Intel Core I5-2400 3.1 Ghz, 2 núcleos Memória: 1 GB Espaço em disco: 40 GB
Sistema Operativo	Ubuntu 10.04 32 bits
Principais softwares	OpenLdap 2.4 PhpLdapAdmin 1.2

#### 4.3.3.2 Ambiente de pré-produção

O ambiente de desenvolvimento utilizado ao longo da realização do projecto é composto por duas máquinas com as seguintes características:

Tabela 4.4 Máquina principal de pré-produção

Fonte: Autor, 2013

Máquina principal de pré-produção	
Tipo de Máquina	Física
Hardware	Processador: Intel Atom N570 1.67 GHz, 2 núcleos Memória: 2 GB Espaço em disco: 30 GB
Sistema Operativo	Ubuntu 10.04 32 bits
Principais softwares	JDK 1.6 MySQL 5.1 Apache Tomcat 6.0 OpenLdap 2.4 PhpLdapAdmin 1.2

#### 4.4 Análise de Requisitos

Para melhorar o protótipo é necessário antes de tudo foi necessário identificar quais eram as necessidades do ISUTC no que refere a gestão de PFCs e quais são as actuais capacidades do protótipo.

##### 4.4.1 Escopo da solução

O escopo desta solução limita-se ao processo que decorre na cadeira de PFC que tem o seu início quando o estudante manifesta o seu interesse em realizar esta cadeira após a aprovação na cadeira de introdução ao PFC. O processo que decorre na cadeira de PFC tem como principal actor o GPFC de forma que este também seja o principal *stakeholder* desta solução.

#### 4.4.2 Processo de realização do PFC

Após uma serie de entrevistas com o GPFC foi possível perceber como ocorre o processo de realização do PFC, isto é, como decorre o processo durante a realização da cadeira de PFC.

Abaixo apresenta-se o diagrama que apresenta o actual processo que decorre durante a realização do PFC:

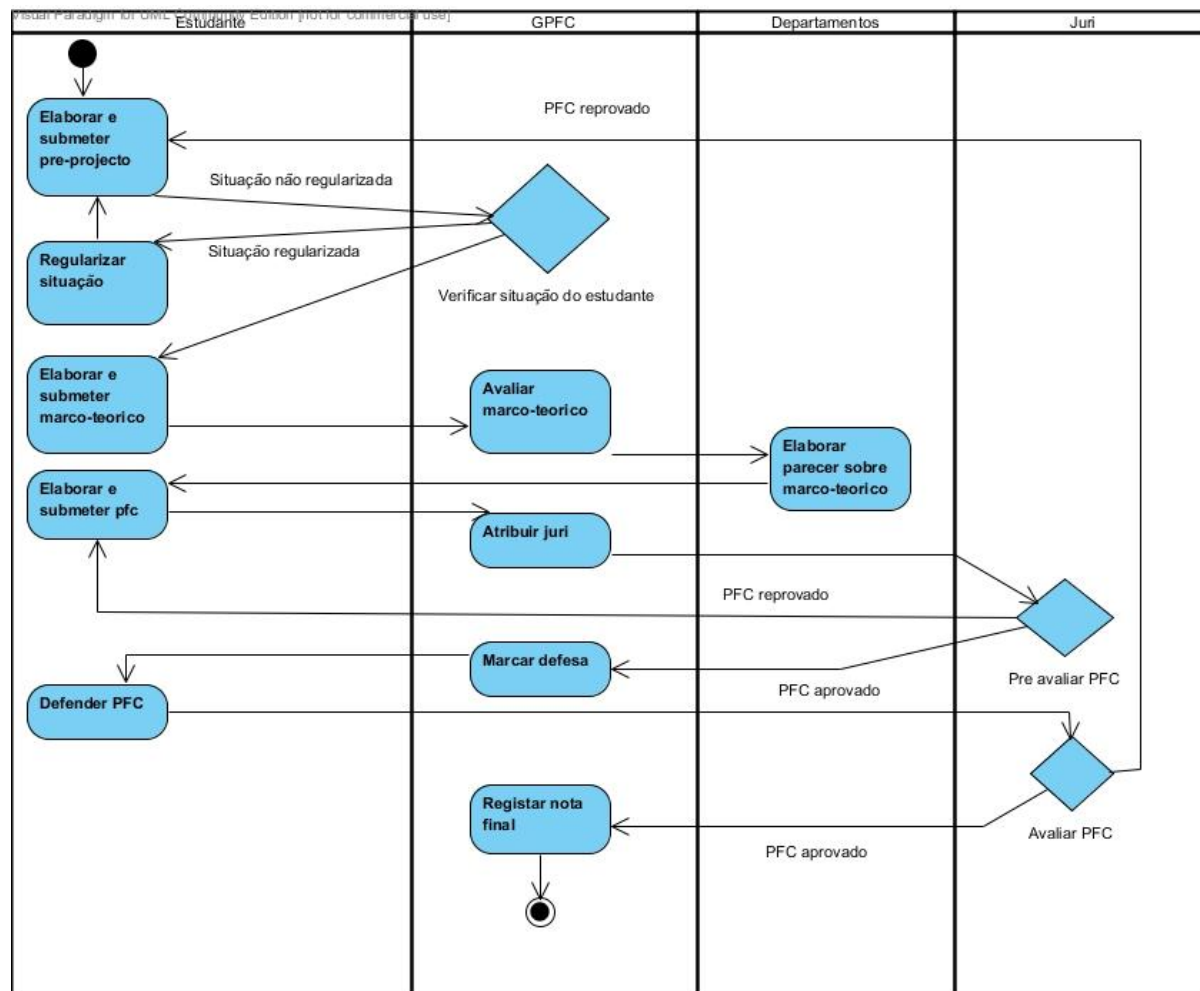


Figura 5.4 Diagrama de actividades do ciclo de vida do PFC

Fonte: Autor, 2013

Este processo compõem a maior parte do ciclo de vida de PFC e é basicamente composto por uma serie de fases onde o estudante vai elaborando e submetendo versões do seu PFC ao GPFC, que em parceria com os departamentos e posteriormente com o Júri composto pelo supervisor, oponente e presidente vão avaliando e encaminhando o estudante ao longo da realização do seu PFC, este ciclo tem o seu termino a quando da aprovação do estudante após a sua defesa.



É importante salientar que no momento da realização do PFC de Stélio Zacarias o processo de realização do PFC era relativamente diferente e que o processo descrito no diagrama acima é o novo processo que entrou em uso no ano actual, por estes motivos o autor não apresentou o diagrama anterior.

#### 4.4.3 Casos de uso implementados

Durante a realização do seu projecto final de curso Stélio Zacarias desenvolveu um protótipo que cumpre com os requisitos funcionais.

Tabela 5.4 Requisitos funcionais implementados no protótipo inicial

Fonte: Stélio Zacarias, 2011

Requisitos funcionais	Actor	Descrição
UC1: Submeter tema	Estudante	Permite ao estudante submeter um tema pertinente para a realização do projecto, onde este pode propor ou não um supervisor
UC2: Consultar tema	Todos	Consultar os temas existentes e o bem como o estado de cada um, e dependendo dos privilégios pode actualizar os temas.
UC3: Apagar tema	GPFC	Permite apagar tema submetido por um estudante
UC4: Actualizar tema	GPFC	Permite actualizar tema submetido por um estudante
UC5: Submeter PFC	Estudante	Submeter o documento no formato electrónico para a devida avaliação, sendo que o documento da fase anterior é substituído pelo documento da presente fase.
UC6: Consultar PFC	Todos	Consultar os PFC existentes e o bem como o estado de cada um, e dependendo dos privilégios pode actualizar os temas.
UC7: Apagar PFC	GPFC	Permite apagar PFC submetido por um estudante
UC8: Actualizar PFC	GPFC	Permite actualizar PFC submetido por um estudante
UC9: Criar relatório	Todos	Permite criar relatório com base na consulta de PFC

Para além desses requisitos funcionais foram ainda implementados os seguintes requisitos não-funcionais.

Tabela 6.4 Requisitos não-funcionais iniciais

Fonte: Stélio Zacarias, 2011

Requisitos não-funcionais	Descrição
RNF1: Autenticar utilizador	Permite que utilizadores cadastrados no LDAP autenticuem-se na aplicação de forma que a autenticação na aplicação seja completamente baseada no LDAP.
RNF2: Verificar permissões	Permite que somente os utilizadores com as devidas permissões no LDAP tenham acesso a certas funcionalidades.
RNF3: Notificar utilizador por email	Permite a notificação por email de diferentes utilizadores quando necessário.

#### 4.4.3.1 Screenshots iniciais da aplicação

Para melhor apresentar a situação inicial da aplicação o autor irá apresentar *screenshoots* das principais funcionalidades da aplicação.

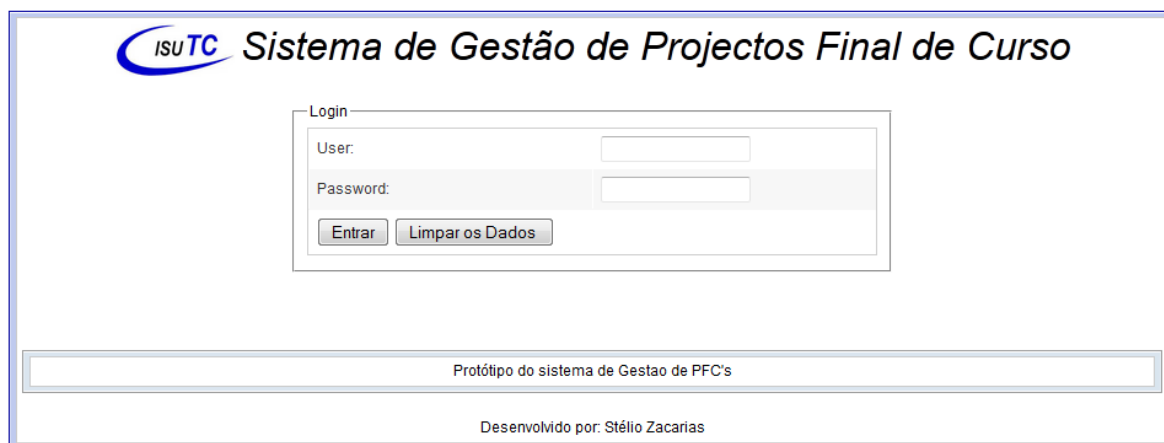


Figura 6.4 Tela de autenticação

Fonte: Autor, 2013

A imagem acima apresenta a tela inicial da aplicação que tem a função de permitir que os utilizadores autenticuem-se na aplicação, é importante salientar que a autenticação é realizada a partir do LDAP.

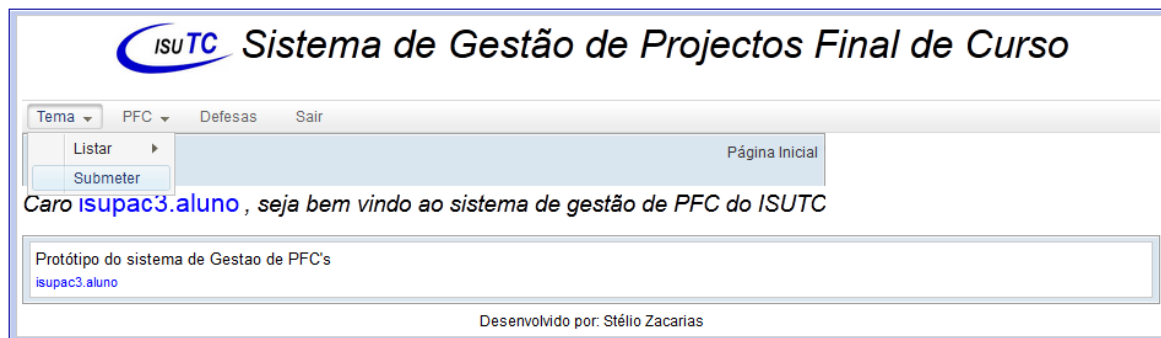


Figura 7.4 Tela principal

Fonte: Autor, 2013

Após uma autenticação com sucesso o utilizador terá acesso a página inicial com o menu que permitirá com que este acesse as restantes funcionalidades, como por exemplo submeter tema.

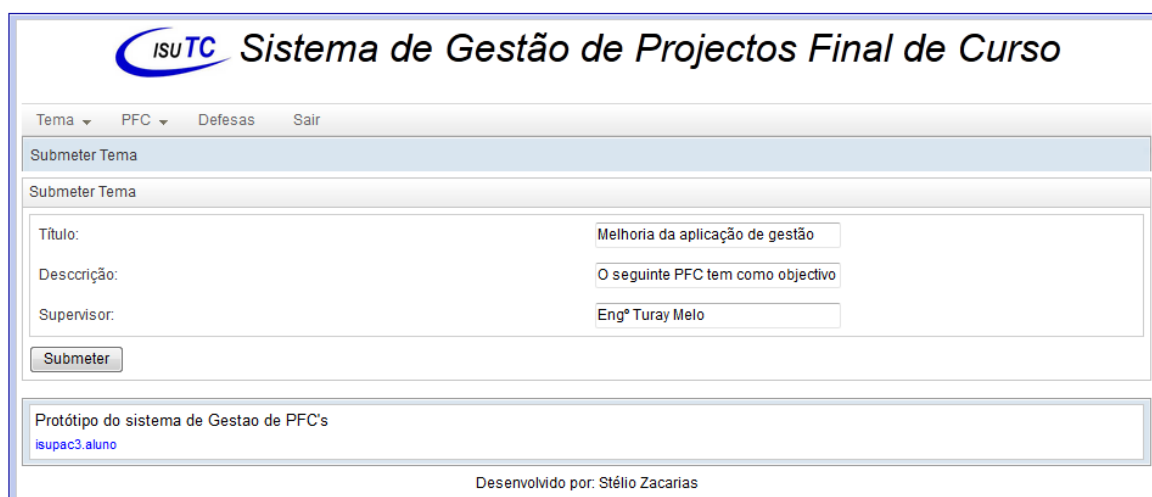


Figura 8.4 Submeter tema

Fonte: Autor, 2013

Uma das funcionalidades mais importantes da aplicação é a funcionalidade que permite submeter temas que é apresentada acima, esta funcionalidade esta somente disponível para estudantes.

## Melhoria da aplicação de gestão de PFC no ISUTC

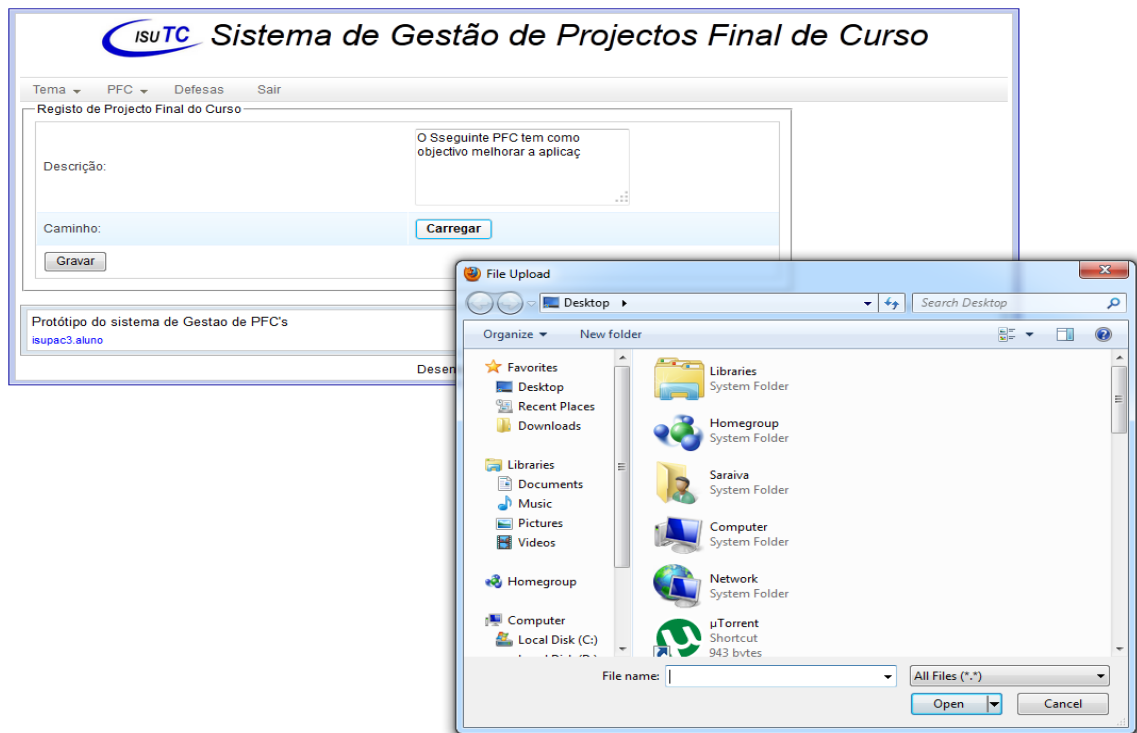


Figura 9.4 Submeter PFC

Fonte: Autor, 2013

Em seguida é apresentada a funcionalidade que permite submeter o PFC, um factor crucial desta funcionalidade é a possibilidade de colocar o ficheiro contendo o PFC em anexo.

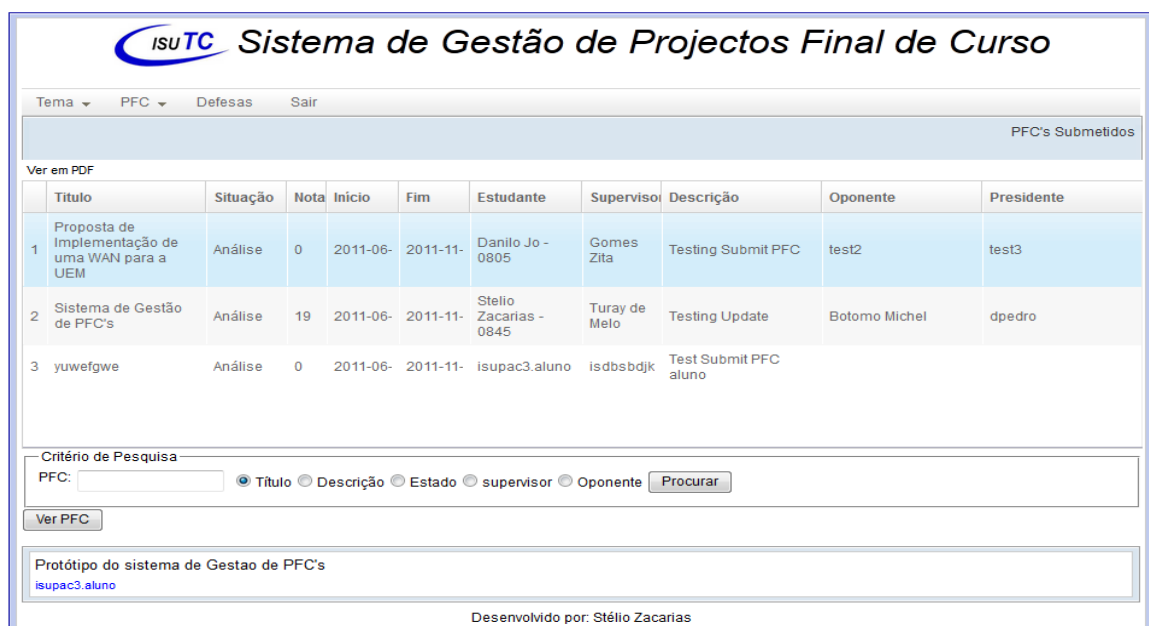


Figura 10.4 Consultar PFC

Fonte: Autor, 2013

O GPFC tem a possibilidade verificar a lista dos PFC's submetidos através da funcionalidade acima.

#### 4.4.4 Casos de uso da solução proposta

Após uma análise das necessidades e do processo de realização de PFC o autor chegou a conclusão que a aplicação com os seguintes casos de uso poderia satisfazer o ISUTC.

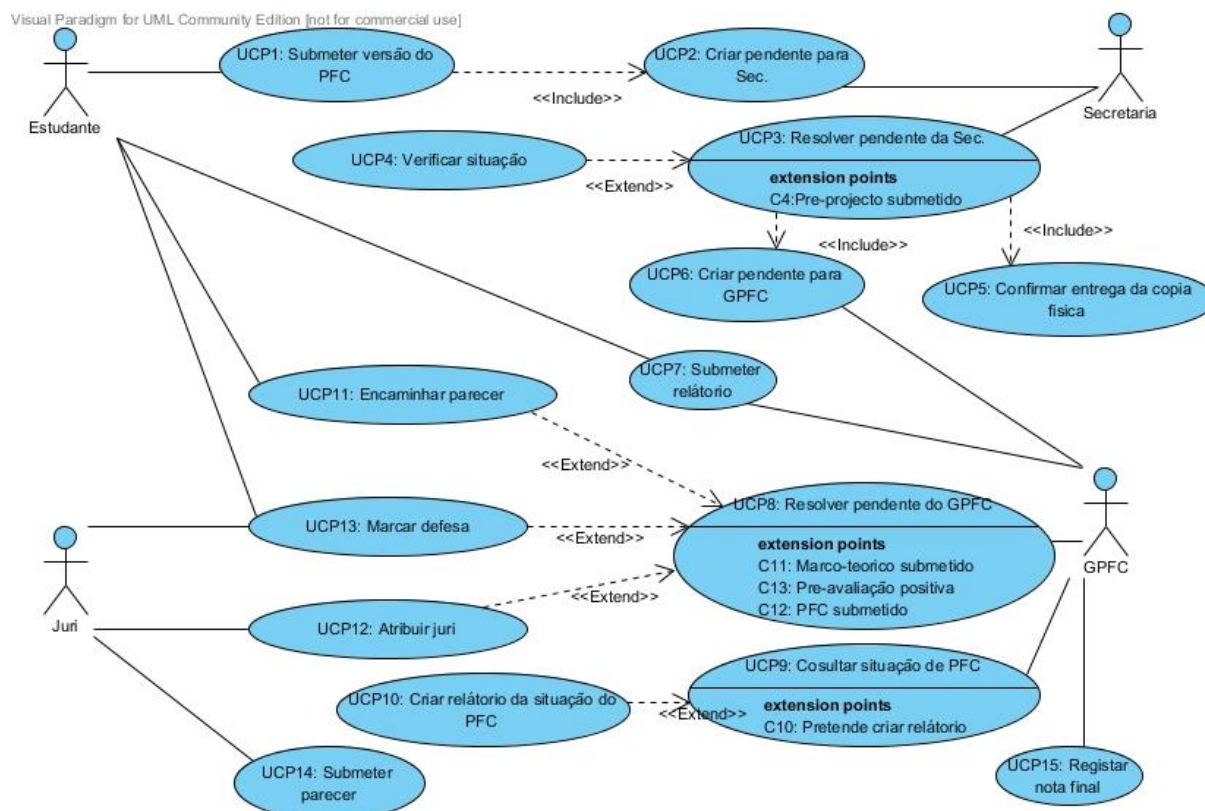


Figura 11.4 Casos de uso proposto

Fonte: Autor, 2013

Abaixo é apresentada uma tabela descritiva dos casos de uso da solução proposta e a sua respectiva relação com os actores.

Tabela 7.4 Requisitos funcionais propostos

Fonte: Autor, 2013

Requisito funcional	Actor	Descrição
UCP1: Submeter versão de PFC	Estudante	Permite que o estudante submeter as diferentes versões do PFC que actualmente são o pre-projecto, marco-teórico e projecto final.
UCP2: Criar pendente para secretária	Estudante	Cria um pendente e notifica a secretária de que deve tomar alguma acção em relação a uma versão do PFC que foi submetida.
UCP3: Resolver pendente da secretária	Secretária	Permite resolver o pendente para que o processo do estudante em causa possa avançar
UCP4: Verificar situação	Secretária	Permite a verificação académica e financeira da situação do estudante
UCP5: Confirmar entrega da cópia física	Secretária	Permite confirmar que o estudante para além de submeter a cópia eletrónica submeteu também a cópia física da versão do PFC
UCP6: Criar pendente para GPFC	Secretária	Cria um pendente e notifica ao GPFC de que deve tomar alguma acção em relação a uma versão do PFC que foi submetida.
UCP7: Submeter relatório	Estudante	Permite que o estudante submeta o relatório informando ao GPFC como esta a decorrer o seu PFC
UCP8: Resolver pendente de GPFC	GPFC	Permite resolver o pendente para que o processo do estudante em causa possa avançar
UCP9: Consultar situação de PFC	GPFC	Permite listar todos os estudantes que estão inscritos na cadeira de PFC e a situação de cada um
UCP10: Criar relatório da situação	GPFC	Permite criar um relatório, contendo a lista de estudantes que estão inscritos na cadeira de PFC e a situação de cada
UCP11: Encaminhar parecer	GPFC	Permite que o GPFC encaminhe o parecer que os departamentos enviam sobre o marco-teórico

UCP12: Atribuir júri	GPFC	Permite o registo do júri atribuído ao estudante
UCP13: Marcar defesa	GPFC	Permite o registo da defesa e envio da notificação ao Júri e ao Estudante
UCP14: Submeter parecer	Júri	Permite que o Júri submeta o parecer sobre o PFC e a respectiva nota.
UCP15: Registrar nota final	GPFC	Permite que o GPFC registe a nota final do estudante e publique o seu PFC se for o caso

#### 4.4.5 Mapeamento dos casos de uso actuais e propostos

Abaixo é apresentado o mapeamento entre os casos de uso actuais e os propostos, isto é, a relação entre os casos implementados no protótipo e os casos de uso da solução idealizada pelo autor.

Tabela 8.4 Mapeamento entre os casos de uso

Fonte: Autor, 2013

Requisito no protótipo melhorado	Requisito no protótipo desenvolvido	Relação entre os requisitos
UCP1: Submeter versão de PFC	UC1: Submeter tema UC5: Submeter PFC	Incorporados
UCP2: Criar pendente para secretária		Acrescentado
UCP3: Resolver pendente da secretária		Acrescentado
UCP4: Verificar situação		Acrescentado
UCP5: Confirmar entrega da cópia física		Acrescentado
UCP6: Criar pendente para GPFC		Acrescentado
UCP7: Submeter relatório		Alterado
UCP8: Resolver pendente de GPFC		Acrescentado

UCP9: Consultar situação de PFC	UC2: Consultar Tema UC6: Consultar PFC	
UCP10: Criar relatório da situação	UC9: Criar relatório	Alterado
UCP11: Encaminhar parecer		Acrescentado
UCP12: Atribuir juri		Acrescentado
UCP13: Marcar defesa		Acrescentado
UCP14: Submeter parecer		Acrescentado
UCP15: Registrar nota final		Acrescentado
RNFP1: Autenticar utilizador	RNF1: Autenticar utilizador	Mantido
RNFP2: Verificar permissões	RNF2: Verificar permissões	Alterado
RNFP3: Notificar utilizador por email	RNF3: Notificar utilizador por email	Incorporado

#### 4.5 ORM e Hibernate

No protótipo desenvolvido é utilizado um SGBD Relacional no entanto a linguagem de programação utilizada é uma linguagem orientada a objectos. Existem certos problemas na compatibilidade entre estes. A maioria dos problemas estão centralizados na dificuldade de mapear uma tabela do modelo relacional a um objecto.

Para tal existem *frameworks Object-Relational Mapping* (ORM).

O *Hibernate* é uma *framework ORM* open-source implementada em Java. O autor optou por utilizar o *hibernate* na aplicação devido as seguintes razões:

- Produtividade – pois ele remove a necessidade de escrever consultas SQL dando a possibilidade do programador focar-se mais na logica de negócio da aplicação;
- Manutenção – ajuda a reduzir as linhas de código tornando a aplicação mais fácil de perceber e consequentemente de manter.
- Portabilidade – Devido a forma de funcionar do *hibernate* que permite a modificação do SGBD necessitando apenas de pequenas configurações.



#### 4.5.1 Implementação do hibernate

Durante a implementação do *hibernate* o autor deparou-se com duas opções no que refere ao mapeamento objecto-tabela. O mapeamento objecto-tabela é um conceito importantíssimo quando se fala de ORM e pode dizer-se que é o núcleo dos ORM. Este mapeamento consiste em interligar a classe de objectos a uma determinada tabela da base de dados de forma que a aplicação saiba em que atributo do objecto deve ser armazenado o valor de um atributo da tabela.

No *hibernate* existem duas formas de fazer estes mapeamentos que são:

- *XML Mappings* – nesta forma o mapeamento é realizado através da utilização de ficheiros XML;
- *Annotations* – nesta forma o mapeamento é realizado através da adição de anotações especiais ao código das classes que representam entidades.

O autor optou em utilizar as *annotations* de modo a:

- Reduzir o número de ficheiros XML na aplicação;
- Facilitar a compreensão da relação entre objectos e tabelas da base de dados.
- De forma a evitar possíveis erros que possam ocorrer caso um administrador de sistema altere o ficheiro XML de mapeamento sem alterar o código.

#### 4.7 Protótipo Melhorado

Após a fase de desenvolvimento o autor foi capaz de implementar a maioria dos casos de uso. Abaixo são apresentados os *screenshots* das principais funcionalidades.

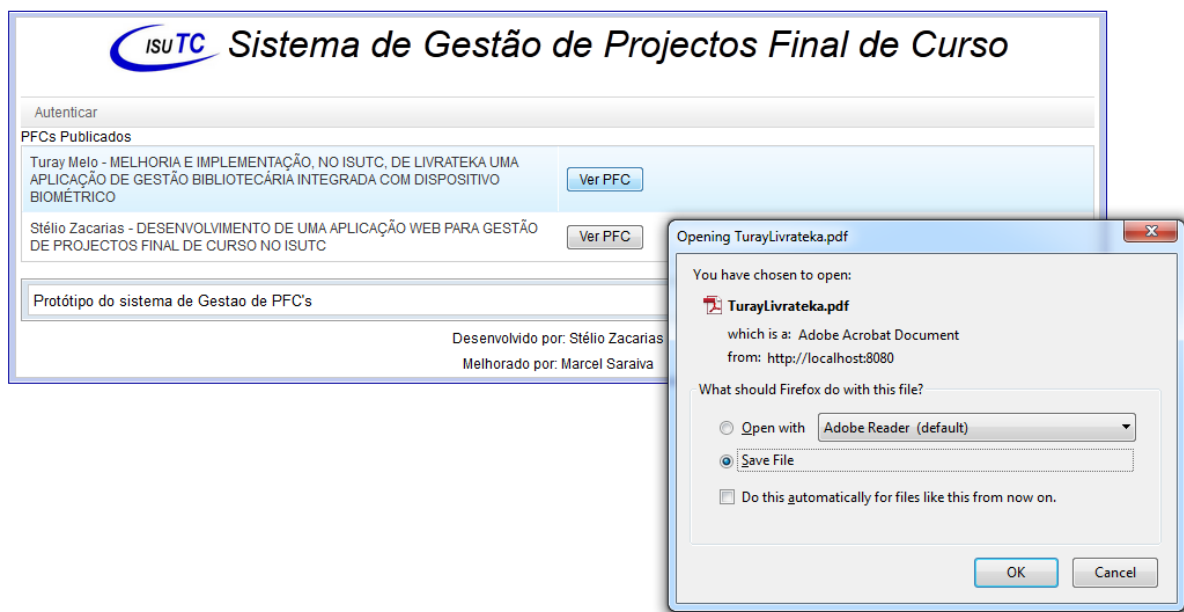


Figura 12.4 Tela inicial

Fonte: Autor, 2013

A primeira alteração foi a criação de uma nova página inicial que contém uma lista de PFCs publicados pelo GPFC. O GPFC poderá optar por publicar ou não o PFC ao registar a sua nota final após a defesa com sucesso do PFC. É importante salientar que não é necessário autenticar-se para ter acesso a esta tela inicial de forma a que o publico em geral possa ter aceso a esta tela e que ela possa ser usada para divulgar os melhores PFCs realizados no ISUTC.

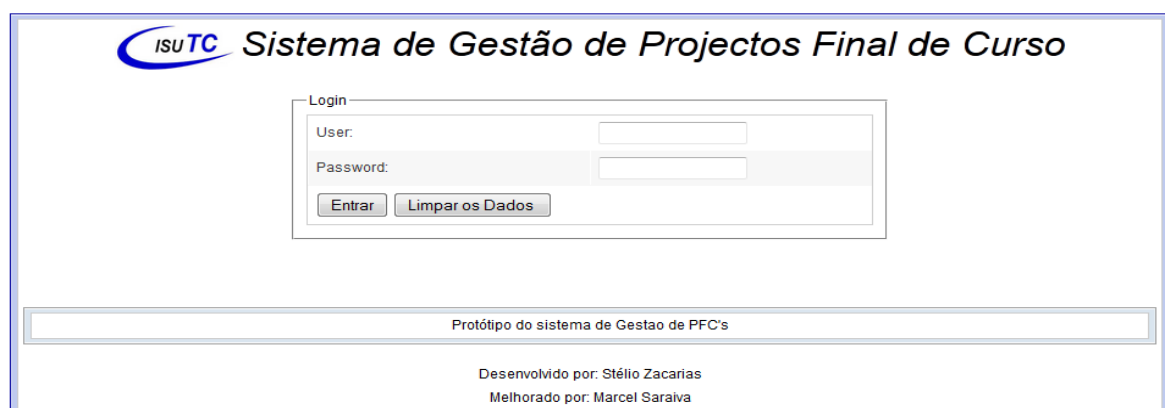


Figura 13.4 Tela de autenticação

Fonte: Autor, 2013

A funcionalidade de autenticação não sofreu modificações visto que o autor considerou a sua implementação adequada para as necessidades da aplicação.

Caso o utilizador consiga autenticar-se com sucesso, ele será encaminhado a página principal contendo o menu para as restantes aplicações.

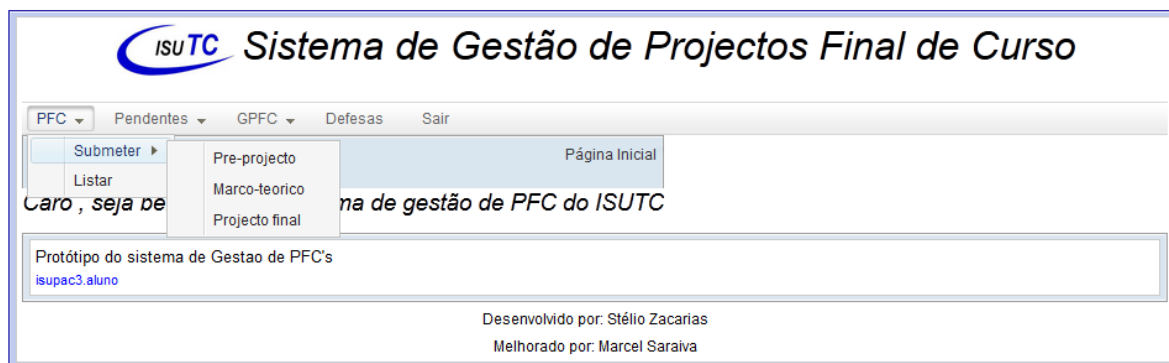


Figura 14.4 Menu principal, item PFC seleccionado

Fonte: Autor,2013

Na imagem acima pode verificar o menu principal e o item PFC que possui as opções:

- Submeter pré-projecto;
- Submeter marco-teórico;
- Submeter projecto final;
- Listar PFC;

As funcionalidades que permitem submeter as diferentes versões do PFC não estarão disponíveis em simultâneo, elas serão disponibilizadas de acordo com o avanço do estudante no processo.

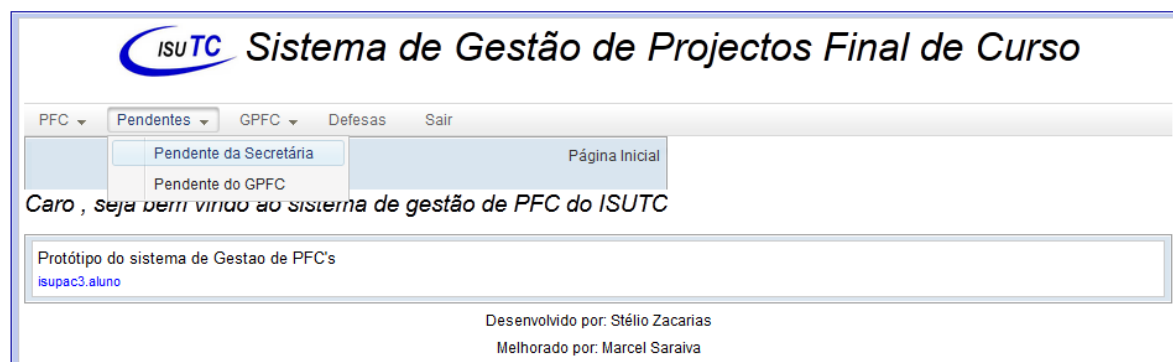


Figura 15.4 Menu principal, item Pendentes seleccionado

Fonte: Autor, 2013

Acima apresenta-se o menu principal com o item Pendentes seleccionado que possui as funcionalidades: pendentes da secretária e pendentes do GPFC, estas funcionalidades estarão disponíveis a secretária e ao GPFC respectivamente, estas funcionalidades são importantes pois para que o processo de realização do PFC do estudante possa avançar não basta que ele submete as versões do PFC a secretária deve confirmar que ele também submeteu a copia física e posteriormente o GPFC deve dar o devido seguimento ao processo.

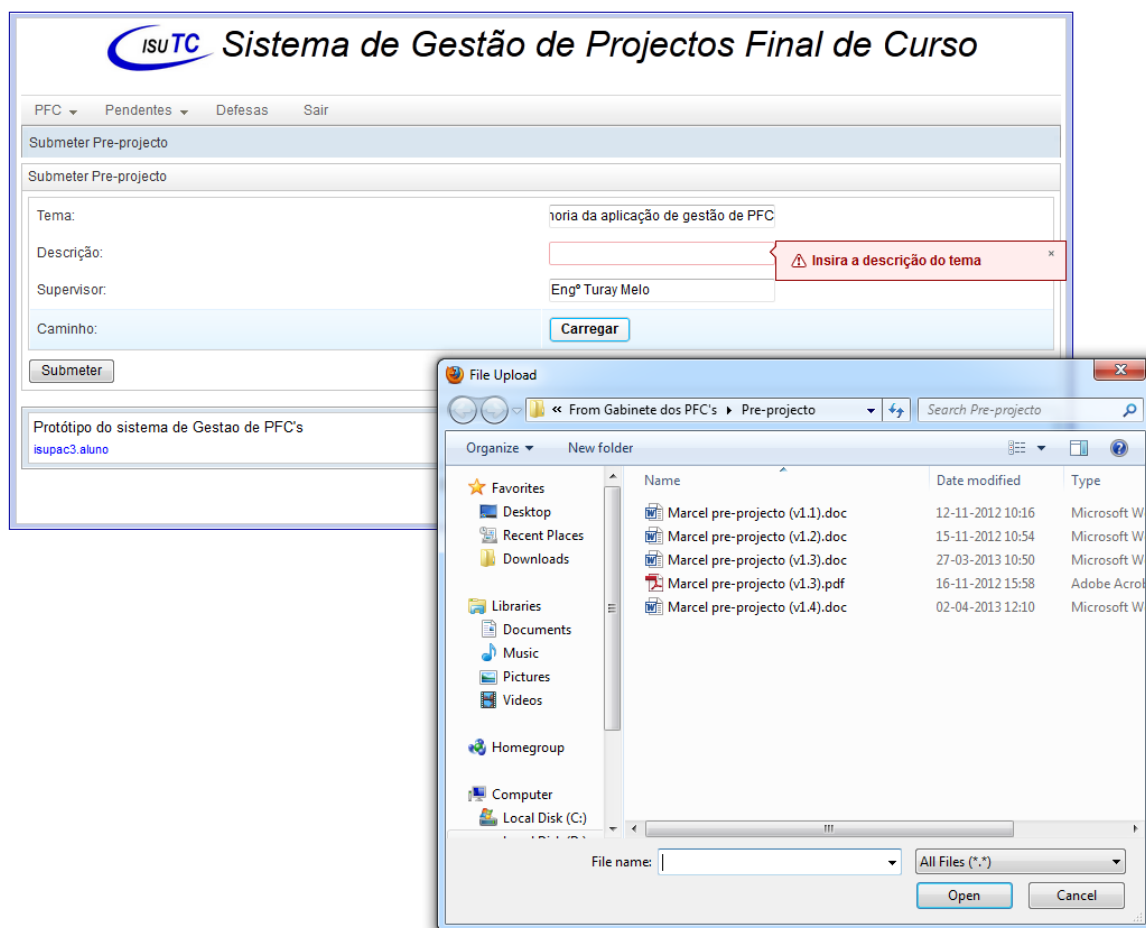


Figura 16.4 Submeter versão do PFC(pre-projecto)

Fonte: Autor, 2013

A funcionalidade de submeter versão de PFC surge como uma junção das funcionalidades submeter tema e submeter PFC, no entanto esta possui a particularidade de alterar-se ligeiramente dependendo da versão do PFC que está a ser submetida.

Caso o utilizador esteja a submeter o pré-projecto este é também obrigado a submeter o tema.

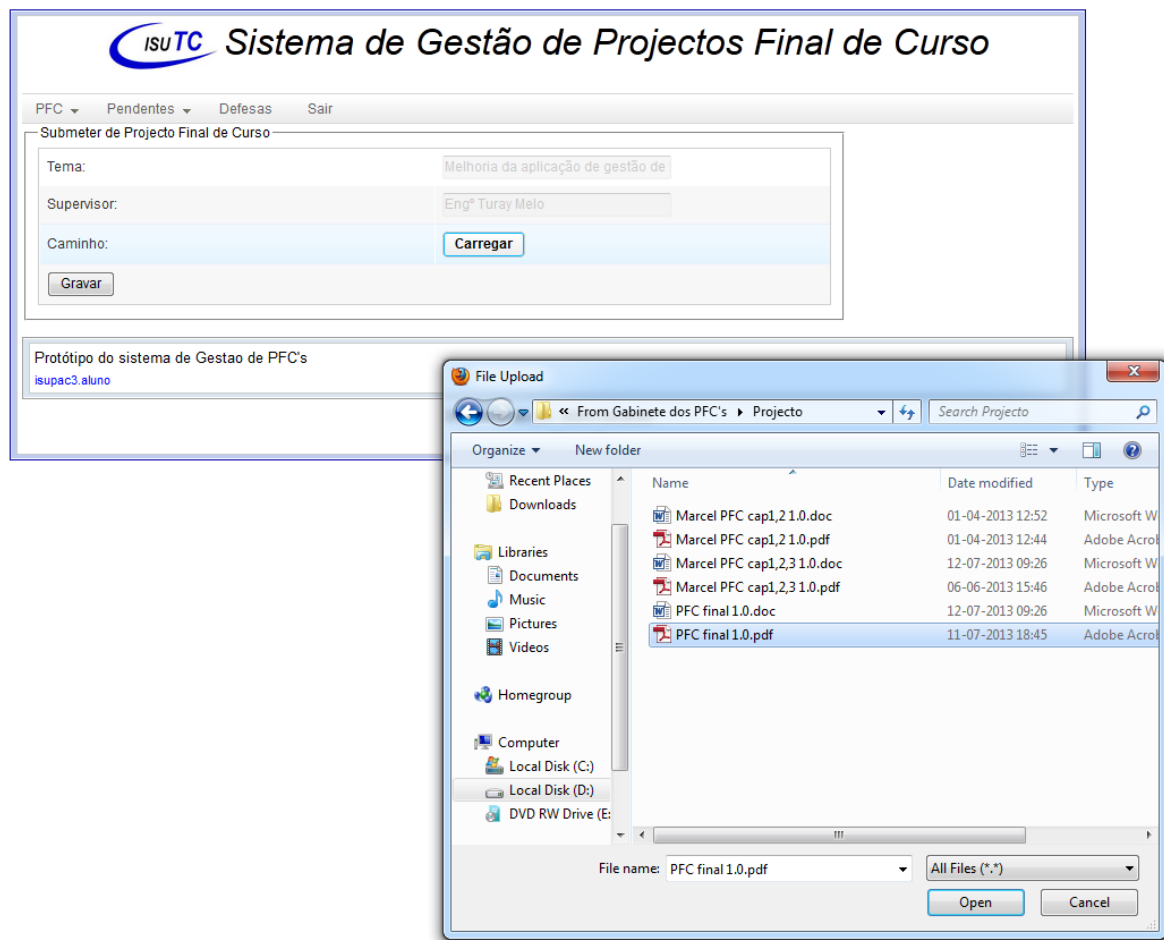


Figura 17.4 Submeter versão do PFC (Projecto final)

Fonte: Autor, 2013

A mesma funcionalidade apresenta um comportamento ligeiramente diferente quando o utilizador esta a submeter o marco-teórico ou a versão final do PFC.

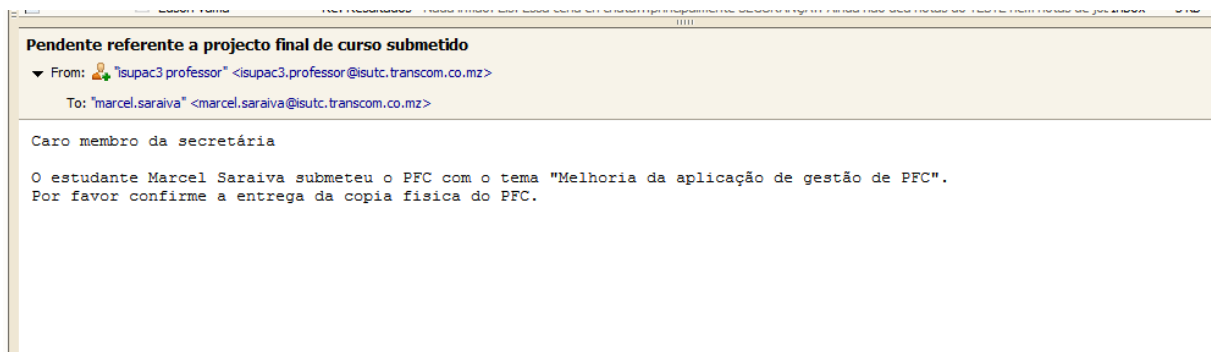


Figura 18.4 Notificação de pendente para secretária

Fonte: Autor, 2013

Ao submeter qualquer versão do PFC é criado um pendente para a secretaria que é responsável por confirmar a entrega da copia física da versão do PFC e caso o estudante esteja a submeter o pré-projecto é também responsável por confirmar que a situação académica e financeira do estudante esteja regularizada.

Acima é apresentado um email enviado automaticamente pela aplicação como forma de notificar a secretária que deve resolver o pendente da notificação em causa.

Abaixo é apresentado a tela de pendentes da secretária que permite que os membros da secretária resolvam os pendentes existentes.



Figura 19.4 Pendentes

Fonte: Autor, 2013

Quando o membro da secretaria resolve o pendente é criado um novo pendente para o GPFC que deve resolve-lo adequadamente de acordo com a versão do PFC que foi submetida.

Na imagem abaixo apresenta-se um *email* onde é solicitado ao GPFC que atribua o Júri após o estudante ter submetido a versão final do PFC.

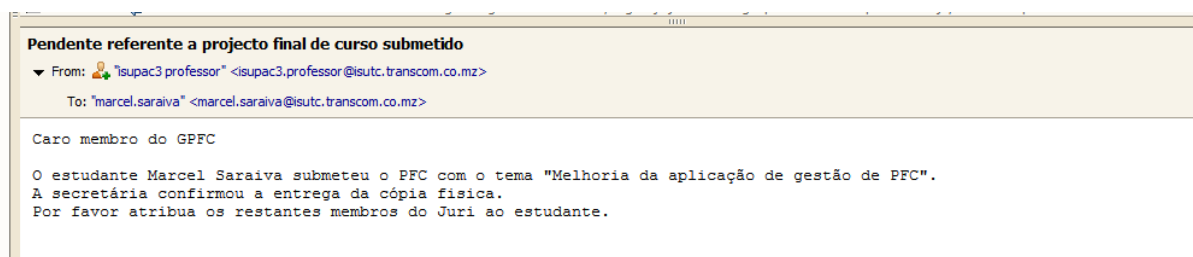


Figura 20.4 Notificação de pendente para GPFC

Fonte: Autor, 2013

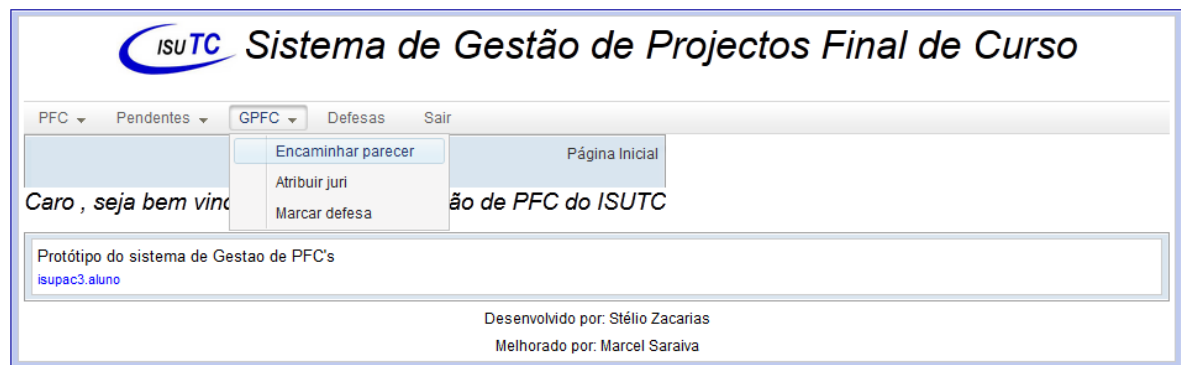


Figura 21.4 Menu principal, item GPFC seleccionado

Fonte: Autor, 2013

Acima verificar o menu principal com o item GPFC seleccionado este item dá acesso ao GPFC para as principais funções a realizar dependendo da versão do PFC que foi submetida.

A funcionalidade encaminhar parecer é usada quando um estudante submete o marco-teórico como pode ver na imagem abaixo:

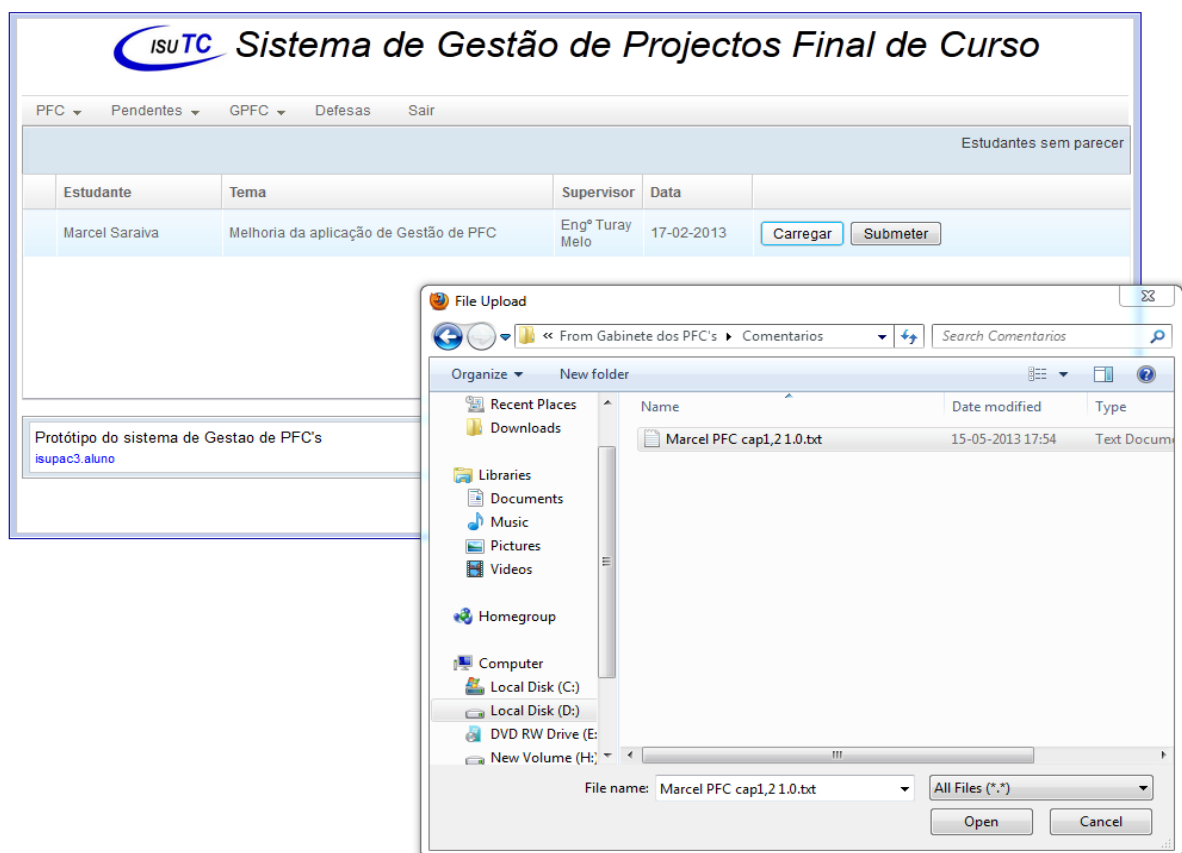


Figura 22.4 Submeter parecer

Fonte: Autor, 2013

## Melhoria da aplicação de gestão de PFC no ISUTC

A funcionalidade Atribuir Juri é usada após o estudante submeter a versão final do projecto.

Por defeito a aplicação completa os dados do supervisor que já foi submetido anteriormente, o GPFC tem a função de apresentar o Presidente e o oponente.

The screenshot displays the ISUTC Sistema de Gestão de Projectos Final de Curso interface. At the top, there is a navigation bar with links: PFC, Pendentes, GPFC, Defesas, and Sair. A dropdown menu is open for 'GPFC', showing the option 'Atribuir juri'. Below this, a table lists projects with columns for Estudante, Tema, Supervisor, and a button to 'Atribuir juri'.

Estudante	Tema	Supervisor	
Yuri Wingester	Desenvolvimento de um modulo para gestão de contas	Engº Micas Rafael	<a href="#">Atribuir juri</a>
Marcel Saraiva	Melhoria da aplicação de Gestão de PFC	Engº Turay Melo	<a href="#">Atribuir juri</a>

Below the table, the details for the selected project 'Yuri Wingester - Desenvolvimento de um modulo para gestão de contas' are shown. It includes fields for Supervisor (Engº Micas Rafael), Email do Supervisor (micas.rafael@isutc.transcom.co.mz), Oponente, Email do Oponente, Presidente, and Email do Presidente. A 'Gravar' button is present. At the bottom, there is a section for the 'Protótipo do sistema de Gestao de PFC's' with a link to 'isupac3.aluno'. The footer credits 'Desenvolvido por: Stélio Zacarias' and 'Melhorado por: Marcel Saraiva'.

Figura 23.4 Atribuir Júri

Fonte: Autor, 2013



## 4.8 Custos

Ao longo do projecto foi necessário desembolsar certos valores que podem ser apresentados no orçamento abaixo:

Tabela 9.4 Custos de desenvolvimento

Fonte: Autor, 2013

Actividade	Tempo (Hr/homem) [A]	Recursos [B]			Custo = A + B (MT)
		Humanos	Materiais	Financeiros (MT)	
Revisão de bibliografias	70	1	Livros, Internet	5000	5000
Treinamento em tecnologias	40	1	Internet, Computador Pessoal	30000	30000
Entrevistas	6	5	Bloco de notas para anotações	2000	2000
Testes da aplicação	20	5	Licenças de aplicações, Computador servidor	50000	50000
Codificação	100	1	Computador pessoal (adquirido durante treinamento), Computador Servidor (adquirido durante testes da aplicação) Programador I	7800	7800
Levantamento de requisitos e desenho da aplicação	50	1	Programador II	4650	4650
Total					52450

Os fundos necessários para a realização do projecto foram financiados pelo autor.

Caso o ISUTC esteja interessado em implementar a aplicação será necessário realizar um certo investimento. Este investimento pode ser descrito na tabela abaixo:

Tabela 10.4 Custos de implementação

Fonte: Autor, 2013

Actividade	Tempo (Hr/homem) [A]	Recursos [B]			Custo = A + B (MT)
		Humanos	Materiais	Financeiros (MT)	
Aquisição do servidor	6	1	Computador servidor	30000	30000
Instalação e configuração do ambiente	15	1	-	7500	7500
Instalação e configuração da aplicação	5	5	-	2500	2500
Formação dos utilizadores	20	5	Sala apropriada, projector	10000	10000
Total					50000

É colocado ainda no primeiro anexo o manual resumido de instalação da aplicação.

## CAPÍTULO 5 - CONCLUSÕES E RECOMENDACÕES

### 5.1 Conclusões

Em jeito de conclusão dizer que a realização deste trabalho foi de grande valia para o autor que teve a oportunidade de consolidar boa parte dos conhecimentos aprendidos ao longo da sua formação principalmente no que esta relacionada com as cadeiras de programação, engenharia de *software*, estrutura e base de dados e gestão de projectos.

A realização deste projecto deu ainda ao autor a oportunidade de trabalhar com uma aplicação bem mais complexa do que as habituais encontradas ao longo da sua formação. Uma grande mais-valia desta aplicação foi a integração com o LDAP e a utilização do mesmo para a autenticação e autorização dos utilizadores levando esta a um nível de complexidade bem mais próximo das aplicações encontradas no mercado.

A implementação do *hibernate* na aplicação demonstrou-se como uma das actividades mais complexas realizadas ao longo da investigação, mas que na visão do autor foi de extrema importância académica e também técnica no que concerne a qualidade e a performance da aplicação, que inicial trará o inconveniente de exigir que futuros programadores tenham conhecimento na *framework* mas que a longo prazo deverá trazer melhorias na produtividade devido a redução do trabalho necessário para realizar consultas a base de dados.

A implementação do *hibernate* do mapeamento objecto-tabela usando *annotations* demonstrou-se eficiente por evitar o acréscimo de mais ficheiros de configuração XML, mas também por apresentarem a relação entre objecto e tabela de uma forma bastante perceptível na visão do autor.

A implementação da notificação por *email* foi realizada com sucesso tendo sido um acréscimo que trará grande impacto na utilidade da aplicação.

O protótipo melhorado esta mais próximo de uma representação de um *workflow* que de uma aplicação de gestão baseada em CRUD (*create read update delete*) e este era o objectivo do autor que ao estudar o processo de gestão de PFC e o ciclo de vida de PFC chegou a conclusão que uma aplicação que seguisse um *workflow* seria mais adequada para a realidade da gestão de PFC.

Caso o ISUTC esteja interessado em implementar e/ou a continuar a desenvolver o protótipo o autor encontra-se disponível para apoiar o ISUTC mediante um acordo satisfaça ambas partes.

## **5.2 Recomendações**

Recomenda-se que o ISUTC:

- Antecipe a cadeira de gestão de projectos, que demonstrou-se uma grande mais-valia na realização do projecto mas que no entanto foi lecionada já a meio da realização do projecto, caso esta fosse lecionada no início ou antes da realização do projecto ela teria um impacto positivo no que concerne a planificação e execução do projecto final de curso;
- Sugere-se ainda que o ISUTC adeque melhor os métodos de avaliação a projectos como este contendo uma forte componente prática e que acabam trazendo como produto final não só o relatório mas também protótipos de aplicações desenvolvidas pelos estudantes;
- O protótipo seja melhorado também no que esta relacionado com as notas do PFC, neste trabalho o autor não focou-se neste aspecto que também é importante.
- A integração com a secretária académica seja verificada, inicialmente o autor não apercebeu-se da importância desta na gestão de PFC mas nas fases finais do projecto o autor percebeu que esta também é um importante autor na gestão de PFC, daí que acredite que seja necessário melhorar este aspecto.
- O ISUTC ou algum estudante continue a desenvolver o protótipo e de especial atenção a integração entre a aplicação de gestão de PFC e aos sistemas de gestão académica usado pelos restantes departamentos do ISUTC;

## REFERÊNCIAS BIBLIOGRÁFICAS

- 1 - ZACARIAS, Stélio Gil dos Santos, projecto final de curso – “Desenvolvimento de uma aplicação web para gestão de projectos final de curso no ISUTC”, Maputo: 2011;
- 2 - QML Zhishu - Computer Applications and Software, 2006 - en.cnki.com.cn
- 3 - SOMMERVILLE, I. Engenharia de software. 6ª Edição, Addison Wesley, São Paulo. 2003
- 4 - PRESSMAN, R. S. Engenharia de software. 3ª Edição, MAKRON Books Editora Ltda, São paulo . 1995
- 5 - Luis Amaral, João Varajão, Luis Alfredo Martins do Amaral, João Eduardo Quintela
- 6 -Varajão. Planeamento de sistemas de informação. 3ª Edição, FCA – Editora de Informática. 2000
- 7 – Shkla, L. e Rosen, R. , Web Application Architecture, John Wiley and Sons. 2003
- 8 – Shelly B., Woods D., Dorin W., HTML,XHTML, and CSS Comprehensive, 6ª Edição – Course Technology. 2011
- 9 – Coulouris G. et al, Sistemas Distribuídos. Conceitos e Projecto, 4ª edição – Concepts and Design. 2007
- 10 – Cruz T., Worflow II: A tecnologia que revolucionou processos, E-papers Serviços Editoriais Lda – 2004
- 11 – Adam A., Implementing Electronic Document and Record Management Systems, Auerbach Publications – 2007
- 12 - BASS, L.; CLEMENTS, P.; KAZMAN, R. Software Architecture in Prattice. 3ª Edição. New York: Addison-Wesley, 2003.
- 13 - VAROTO, A. C. Visões em arquitectura de software. Dissertação de mestrado – Universidade de São Paulo – São Paulo, 2002.
- 14 – REZENDE, D. Engenharia de software e sistemas de informação. 3ª Edição. Brasport Livros e Multimidia Ltda - 2005
- 15 – Venecian L. [http://olaria.ucpel.tche.br/venecian/lib/exe/fetch.php?media=qs\\_intro2.pdf](http://olaria.ucpel.tche.br/venecian/lib/exe/fetch.php?media=qs_intro2.pdf) (acesso: 6-05-2013)

16 – Venecian L.

[http://olaria.ucpel.tche.br/venecian/lib/exe/fetch.php?media=qs\\_iso12207\\_15504.pdf](http://olaria.ucpel.tche.br/venecian/lib/exe/fetch.php?media=qs_iso12207_15504.pdf) (acesso: 6-05-2013)

17 – Lobo, E. Cursos de Engenharia de Software. Universo dos Livros Editora Ltda. São Paulo – 2008

18 – ISUTC. <http://www.transcom.co.mz/isutc>. (acesso: 8-05-2013)

19 - Ramakrishnan R., Gehrke J., Sistemas de gerenciamento de banco de dados, 3ª edição - McGraw-Hill. 2008

20 – Costa R., SQL Guia Prático, 2ª edição – BRASPORT livros e multimídia Ltda. 2007

## **BIBLIOGRAFIA**

- 1 - Sara M., em  
[http://www.notapositiva.com/trab\\_estudantes/trab\\_estudantes/tic/tic\\_trabalhos/sistgestbd.htm](http://www.notapositiva.com/trab_estudantes/trab_estudantes/tic/tic_trabalhos/sistgestbd.htm)  
(acesso: 4-09-2012).
- 2 - GARLAN, D.; SHAW, M. An Introduction to software Architecture. Disponível em:  
<[http://www.cs.cmu.edu/afs/cs/project/vit/ftp/pdf/intro\\_softarch.pdf](http://www.cs.cmu.edu/afs/cs/project/vit/ftp/pdf/intro_softarch.pdf)> Acesso em 12 de  
Novembro de 2012.
- 3 - Conallen, J. Building web applications with uml. 2º Edição, Addison – Wesley. 2003
- 4 - MARTIN, J. e ODELL, J. Análise e Projeto Orientados a Objeto. São Paulo,  
Makron Books, 1995.
- 5 - RUMBAUGH, J.; BLAHA, M.; PREMERLANI, W; EDDY, F. e LORENSEN, W, .  
Modelagem e Projetos Baseados em Objetos. Rio de Janeiro, 1994.
- 6 - Teorey, T. et All . Database Design- Know It All. Elsevier: USA, 2009.

## ANEXOS

### Manual de Instalação da aplicação

#### 1. Introdução

O presente documento tem como função descrever resumidamente os passos necessários para a instalação e configuração da aplicação.

Este manual é direccionado a administradores de sistemas que tenham experiencia básica em Linux, MySQL e aplicações J2EE

#### 2. Passos da instalação

##### 2.1 Instalação e configuração inicial do sistema

O primeiro passo é preparar a máquina que deverá hospedar a aplicação esta máquina pode ser física ou virtual mas aconselha-se que ela tenha os seguintes requisitos mínimos:

- Processador *single-core* com 2.5 GHz
- 1.5 GB de memória
- 45 GB de espaço em disco.

Instale o Sistema Operativo na máquina e faça as configurações de rede necessárias de forma que a máquina tenha conexão com o servidor LDAP e os clientes. Quanto ao sistema Operativo aconselha-la o uso do Ubuntu na distribuição 10.04 no entanto a aplicação é também compatível com as versões 8.04 e 12.04 do Ubuntu, Windows XP e Windows 7.

Instale o JRE (Java Runtime Environment) e configure correctamente as variáveis de ambiente. É importante que seja usada a versão 1.6 para evitar incompatibilidades.

Instale o SGBD MySQL aconselhe-se que usa a versão 5.1.

Instale o Apache Tomcat na versão 6.0



## 2.2 Instalação e configuração da aplicação

### Aceda ao MySQL

E execute os comandos abaixo para criar a base de dados, carregar os dados nela e criar um utilizador para acessar a base de dados:

```
Create database pfc;  
Use pfc;  
source D:\Repository\SaraivaRepository\pfc\data_base\dump\pfc_full.sql  
grant all privileges on pfc.* to username@localhost identified by 'password';
```

Copie o ficheiro *pfc.war* para o diretório *TOMCAT\_HOME/webapps*.

Faça o extract dos *pfc.war* e edite o ficheiro: *spring-manager-pfc.xml*

```
<bean id="ldapContextSource"  
class="org.springframework.ldap.core.support.LdapContextSource">  
    <property name="url" value="ldap://192.168.1.101:389" />  
    <property name="base" value="dc=isutc,dc=transcom,dc=co,dc=mz" />  
    <property name="anonymousReadOnly" value="true"/>  
</bean>
```

Altere o hostname do servidor de Ldap que no exemplo acima é 192.168.1.101.

```
<bean id="dataSource" class="com.mchange.v2.c3p0.ComboPooledDataSource"  
destroy-method="close">  
    <property name="driverClass" value="com.mysql.jdbc.Driver" />  
    <property name="jdbcUrl" value="jdbc:mysql://localhost:3306/pfc" />  
    <property name="user" value="username" />  
    <property name="password" value="password" />  
</bean>
```

Altere também o username e a password de acesso a base de dados.

Edite o ficheiro: *spring-security-pfc.xml*

```
<security:ldap-server id="ldap" url="ldap://192.168.1.101:389"/>
```

Altere o hostname do servidor de Ldap que no exemplo acima é 192.168.1.101.

Por fim arranque o tomcat e poderá acessar a aplicação através do endereço abaixo:

*Http://<hostname>:8080/pfc*