

**MELHORIA E IMPLEMENTAÇÃO, NO ISUTC, DE LIVRATEKA
UMA APLICAÇÃO DE GESTÃO BIBLIOTECÁRIA INTEGRADA
COM DISPOSITIVO BIOMÉTRICO**

Turay Filipe de Melo

Projecto Final de curso

Curso de Licenciatura em Engenharia Informática e de Telecomunicações

Supervisor

Eng. Cameron Ord Smith

Departamento de Tecnologia de Informação e Comunicação

Julho, 2009



or de Transportes e Comunicações

Departamento de Tecnologia de Informação e Comunicação

Licenciatura em Engenharia Informática e de Telecomunicações

Projecto Final de Curso

Tema:

Melhoria e implementação, no ISUTC, de livrateka - uma aplicação de gestão bibliotecária integrada com dispositivo biométrico

Autor:

Supervisor:

(Turay Filipe de Melo)

(Eng. Cameron Smith)

Maputo, Julho de 2009



PDF
Complete

*Your complimentary
use period has ended.
Thank you for using
PDF Complete.*

[Click Here to upgrade to
Unlimited Pages and Expanded Features](#)



MELHORIA E IMPLEMENTAÇÃO NO ISUTC, DE LIVRATEKA, UMA APLICAÇÃO,
BIBLIOTECÁRIA

Turay Filipe de Melo



Dedicatória

Aos meus pais, meus companheiros, professores, Tuahiri e Iara.

Agradecimentos

Aos meus pais, pela educação que me deram, pelo carinho, sempre que precisei. E pelos sermões que me fizeram tão bem quanto aos abraços.

Ao meu irmão e a Iara, por me terem dado apoio quando mais precisei. Agradeço por estarem sempre presentes nos momento mais difíceis e alegres da minha vida.

Ao Eng. Cameron, pela orientação dada, críticas, e pelo facto de ter sido amigo e ao mesmo tempo exigente.

Aos meus amigos, Kiko, Dillon, Estaline, Elisabeth, Victor, Mauro, Néelson, Guerte, e a toda minha turma I51, pelo prazer que tive em ter colegas tão agradáveis.

Ao Narciso, Six, Manhique, S. Langa, Ivo, Eurico e todos os meus colegas de trabalho, muito obrigado pelas dicas de programação e por compartilharem comigo o meu dia-a-dia de trabalho de forma alegre e séria.

Aos meus professores, Enga. Priscila, Prof. Ilal, colegas do LIMEAA, Eduardo, funcionários do ISUTC e a todos que directa ou indirectamente contribuíram para o meu sucesso escolar,

Muito Obrigado!

Turay Melo



Declaração de Honra

Declaro por minha honra que este relatório foi por mim elaborado sob orientação do meu supervisor Eng. Cameron Smith, e as informações nele contidas foram extraídas do material referenciado e reservado as referências bibliográficas.

Turay Filipe de Melo

Resumo.....	11
1 Introdução	xiii
1.1 Definição do problema.....	xiii
1.2 Objectivos do trabalho a Realizar	xiv
1.2.1 Gerais	xiv
1.2.2 Específicos.....	xiv
1.3 Produtos do Projecto	xiv
1.4 Estruturação da dissertação	xv
2 Revisão bibliográfica.....	xv
2.1 Introdução	xv
2.2 Integração de sistemas	xvi
2.2.1 Introdução.....	xvi
2.2.2 Abrangência da Integração de Sistemas de Informação	xvi
2.2.3 Características da integração	xviii
2.2.4 Tipos de Integração.....	xviii
2.2.5 Resumo.....	xix
2.3 O Protocolo LDAP	xx
2.3.1 Historial.....	xx
2.3.2 Definição	xx
2.3.3 Características.....	xxi
2.3.4 Modelos de dados	xxii
2.3.5 Comparação entre LDAP e uma base de dados relacional.....	xxii
2.3.6 Arquitectura do LDAP	xxiii
2.3.7 Operações.....	xxiv
2.3.8 Funcionamento do LDAP.....	xxv
2.3.9 Resumo.....	xxvi
2.4 Single-Sign-On	xxvi
2.4.1 Introdução.....	xxvi

	xxvi
	xxvii
2.4.4	Vantagens do Single-Sign-On	xxvii
2.4.5	Desvantagens	xxvii
2.4.6	Resumo.....	xxviii
2.5	O framework Acegi (Spring Security).....	xxviii
2.5.2	Componentes	xxix
2.5.3	Funcionamento	xxx
2.5.4	Resumo.....	xxxi
2.6	Sistemas Biométricos.....	xxxii
2.6.1	Introdução.....	xxxii
2.6.2	Características de Sistema de reconhecimento de impressão digital	xxxii
2.6.3	Funcionamento	xxxiii
2.7	Modelo de Processo de software	xxxiii
2.7.1	Introdução.....	xxxiii
2.7.2	Modelo em cascata.....	xxxiv
2.7.3	Desenvolvimento evolucionário	xxxiv
2.7.4	Desenvolvimento formal de sistemas	xxxiv
2.7.5	Desenvolvimento orientado a reuso.....	xxxiv
3	Desenvolvimento da aplicação.....	xxxiv
3.1	Introdução	xxxiv
3.2	Arquitetura do sistema	xxxv
3.2.1	Distribuida	xxxv
3.2.2	Em camadas.....	xxxv
3.3	Modelo de desenvolvimento do software	xxxv
3.4	Tecnologias bases para desenvolvimento de livrateka	xxxvi
3.4.1	Introdução.....	xxxvi
3.4.2	Linguagem de programação	xxxvi
3.4.3	Servidor Web.....	xxxvii
3.4.4	Sistema de Gestão de Base de dados	xxxvii

	xxxviii
	xxxix
3.6.1 Computador	xxxix
3.6.2 Leitor de impressão digital	xxxix
3.7 Plano de actividades	xl
3.8 Desenvolvimento das actividades.....	xli
3.8.1 Definição dos casos de uso.....	xli
3.8.2 Desenho da base de dados	xliii
3.8.3 Integração LDAP-Spring.....	xliii
3.8.4 Integração com a biblioteca de SMS	xlvi
3.8.5 Codificação de casos de uso	xlvi
3.8.6 Script para consulta automática de utilizadores com multa	xlvi
3.8.7 Algoritmo de pesquisa de impressões digitais.....	xlvii
3.8.8 Hosting para isunet e para fora	xlvii
3.8.9 Testes da aplicação no ISUTC.....	xlviii
3.8.10 Melhoria resultante das críticas	xlix
3.9 Custos da implementação do projecto	xlix
4 Conclusão.....	li
4.1 Conclusões	li
4.2 Dificuldades enfrentadas.....	li
4.3 Recomendações	lii
4.4 Bilbiografia	liii
4.4.1 Web Sites:.....	liii
4.4.2 Livros	liv
4.4.3 Entrevistas	lv
Anexo 1 ó Reunião com Engenheiro Donis.....	54
Anexo 2 ó Desenho das bases de dados.....	55
Anexo 3 ó Bibliotecas usadas.....	56
Anexo 4 ó Diagramas de casos de uso.....	57

Índice	Eduardo.....	58
Anexo 6 ó Casos de Uso.....		59
Anexo 7 ó Classe UserManager.java.....		61
Anexo 8 ó Classe SecurityUtil.java.....		67
Anexo 9 ó ficheiro spring-config.xml.....		69
Anexo 10 ó ficheiro hopeContext.xml.....		73
Anexo 10 ó Inquérito da Aplicação livrateka.....		77
Anexo 11 ó Carta de pedido para realização de testes.....		78
Anexo 12 ó Minuta de entrada de dados do CEDOC.....		79

Índice de figuras

Ilustração 1 - Redes, middleware e integração de sistemas [L03].....	xvii
Ilustração 2 - Directório de informação do LDAP.....	xxiv
Ilustração 3 - processo de autenticação com Acegi - adaptado do òdiagram 1: Basic Flow for Authentication using Acegiö (W19)	xxxi

ITU ó International Telecommunication Union

ISO ó International Organization For Standardization

DNS ó Domain Name System

DAP ó Direct Access Protocol

IETF ó Internet Engineering Task Force

RFC ó Request For Comment

PDU ó protocol data unit

CEDOC ó Centro de Documentação

SMS ó Short Message Service

API ó Application Programming Interface

EAI ó Enterprise Application Integration

LDAP ó Lightweight Direct Access Protocol

SGBD ó Sistemas de Gestão de Base de dados

DIT ó Directory Information Tree

LAN ó Local Area Network

SSL ó Security Socket Layer

DN ó Distinguished Name

DIT ó Directory Information Tree

UTP - Unshielded Twisted Pair

DHCP ó Dynamic Host Configuration Protocol

NAT ó Network Address Translator

HTTPS ó Hipertext Transfer Protocol Secure

TLS ó Transport Layer Security

SPOF ó Single Point Of Failure

O trabalho descrito nesta dissertação insere-se no âmbito do Projecto de Fim 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 Turay Filipe de Melo e sob supervisão do Dr. Cameron Ord Smith.

O Projecto de Fim de Curso elaborado por Sérgio Henrique Langa em 2008 (L07) consistiu num protótipo para colmatação da necessidade do ISUTC, de um sistema de gestão bibliotecária, apresentando, por um lado, uma boa resposta às necessidades de segurança que os sistemas informáticos exigem, utilizando autenticação via dispositivo biométrico e utilizando o *framework*¹ de segurança Acegiö (Spring-Security)². Contudo, o projecto não foi implementado, nem foi solicitado por parte do ISUTC para posterior melhoria e efectivação.

O trabalho proposto neste documento pretende ajustar o protótipo às necessidades do centro de documentação, permitindo o cadastro e agrupamento de livros de forma mais real. Por um lado, pretende integrar o sistema com o serviço de directório (LDAP) vigente no ISUTC. Por outro, implementar uma série de casos de uso em falta e actividades de rotina automáticas, tais como envio de e-mails, SMS³ e sistemas de *backups*.

Estudou-se o protocolo LDAP e Integração de Sistemas para melhor desenvolver este processo, a medida que se foi realizando a codificação da aplicação, tendo havido sucesso. Registou-se, contudo, um fracasso na integração com SMS, depois de se usar a biblioteca de envio, criada por Eurico da Silva como PFC em 2007(L05), mesmo seguindo à risca o manual de utilização e contactando o autor.

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

² Spring é um framework de apoio a aplicações web robustas escritas na linguagem Java (<http://www.springframework.org/>)

Acegi é o framework oficial do Spring que providencia segurança para as aplicações web. (www.acegisecurity.org/)

³ Short Message Service (SMS) é uma tecnologia que permite a troca de mensagens entre celulares. (L05, p.10)

Introdução

Definição do problema

O Centro de documentação (CEDOC) é um dos sectores fundamentais para assimilação dos conteúdos ministrados nas aulas do ISUTC. Diariamente, alunos, professores, e diversos membros desta e de outras instituições dirigem-se a este centro com o fim de requisitarem diversos tipos de publicações, desde livros, revistas científicas, passando pelos projectos de fim de cursos já elaborados.

O processo de requisição de livros é feito manualmente, onde o bibliotecário regista os dados do aluno e do livro em causa e este aluno assina no acto de levantamento e no de devolução.

Antes do empréstimo, o aluno deve reservar o livro durante a semana. Para fazê-lo, percorre a lista de reservas, observando uma linha por linha, de modo a verificar se o livro pretendido não terá sido reservado. (L07, p.37).

Existe um software de gestão de bibliotecas no CEDOC que se mostra obsoleto.(E01).

A morosidade com que se levanta o livro, e com que se faz a reserva, motivou a dois alunos finalistas do curso de Eng. Informática e de Telecomunicações dos anos anteriores (Rodrigo Canze (2005)(L07, p.45), e Sérgio Henrique Langa(2008)(L07)) a levarem a cabo projectos de fim de curso sobre aplicações inovadoras para gestão de biblioteca.

Contudo, devido a, por um lado as aplicações não estarem acabadas (L07, p.83) e por outro, não ter havido uma plataforma de entendimento entre a escola e os alunos no sentido de se aproveitar os seus protótipos, estes nunca chegaram a ser implementados na integra.

Apesar de implementar ter um sentido lato, é usado nesta dissertação como:

- a) Testado em um ambiente mais credível do que o ambiente do programador;
- b) Alojado num servidor da instituição ou do local para onde o sistema foi idealizado;
- c) Utilizado pela instituição projectada;

Gerais

- Melhorar e implementar o protótipo de modo a torná-lo em condições de utilização efectiva;
- Mostrar a relevância de integração da aplicação com bibliotecas externas;
- Investigar a área de desenvolvimento de software;

Específicos

- Integrar o LDAP na aplicação de gestão bibliotecária;
- Terminar os casos de uso da aplicação;
- Integrar a biblioteca de SMS, proposta no trabalho de Eurico da Silva, na aplicação de gestão bibliotecária;
- Pesquisar e utilizar a biblioteca do Spring para realizar tarefas automáticas e escalonadas, na aplicação;
- Colocar a aplicação a funcionar num dos servidores do ISUTC;
- Documentar o software, realizar testes, e capacitar pessoal na utilização;

Produtos do Projecto

Espera-se neste projecto a produção de:

- Um relatório escrito do projecto;
- Um software funcional para o CEDOC;
- Capacitação de utilizadores e do bibliotecário no usufruto da aplicação;

quatro capítulos. No primeiro capítulo faz-se uma introdução, onde se formula o problema, apresentam-se os objectivos do projecto, os resultados esperados e a estrutura da dissertação.

O segundo capítulo trata da revisão bibliográfica, onde se dá a conhecer alguns conceitos importantes para a percepção do trabalho desenvolvido no terceiro capítulo, que detalha os passos desenvolvidos. O derradeiro capítulo, quarto, é o das conclusões e recomendações do projecto.

Capítulo 2

Revisão bibliográfica

Introdução

Este capítulo fornece fundamentos teóricos para que se entenda melhor os procedimentos seguidos na elaboração do projecto. Falar-se-á da integração de sistemas, o protocolo LDAP, Single Sign On, Sistemas biométricos, Acegi (Spring-Security) e modelos de processamento de software, úteis para autenticação de utilizadores na aplicação.

Os conceitos aqui introduzidos serão melhor compreendidos na secção 3.8, quando for necessário aplicá-los, porém, para não tornar a leitura um mistério, pode-se afirmar, em traços gerais, que em conformidade com os objectivos específicos:

- A leitura sobre integração de sistemas, LDAP, bem como de Spring Security é útil para apoiar no processo de integração da aplicação com o LDAP;
- O Single Sign On, por sua vez, torna-se útil para perceber os mecanismos, vantagens e riscos de existir a senha única nas aplicações de empresas, que é o que acontece com o ISUTC e portanto com livrateka;

Introdução

Nas empresas, há a necessidade de trocar dados e processos sem ter que fazer mudanças bruscas nas aplicações⁴ ou fontes de dados⁵. (L02, p.8).

Por vezes, um sistema torna-se obsoleto ou já não responde as actuais demandas da empresa, na totalidade. A empresa deve comparar o custo de migrar de um sistema para o outro, com o custo de integrá-los. Abandonar tecnologias ou sistemas nem sempre é tão fácil: por vezes as empresas gastam somas avultadas para a compra, instalação e personalização de sistemas. Tempo é dispendido no treino de utilizadores e muitas vezes, a própria forma de trabalhar de uma organização muda ou passa a vigorar em função de sistemas informáticos. Torna-se difícil desfazer-se de um investimento desta natureza. (L01, p.1).

Da mesma forma que trazer uma nova tecnologia pode ser dispendioso. Existe o gasto referente a aquisição do sistema, o tempo investido no treino do pessoal e adaptação às metodologias de funcionamento da empresa. O investimento, em alguns casos, torna-se mais alto do que o investimento na aplicação existente. (L01, p.1).

Em informática, integração de sistemas de informação (EAI em inglês), refere-se a partilha não restrita de dados e processos de negócio entre quaisquer fontes de dados e aplicações interligadas dentro da empresa ou entre empresas. (L02, p.8).

Esta partilha de dados dá origem a um sistema de informação (SI), que designa um conjunto de aplicações que partilham dados entre si. L03

Abrangência da Integração de Sistemas de Informação

No campo de EAI⁶, existem muitas formas de abranger a integração de SI⁷:

Geografia da integração

Quanto a geografia de integração, a Integração de Sistemas de informação classifica-se em: Integração no computador e Integração na empresa.

⁴ Sistemas de informática.

⁵ Fontes de informação digital, como por exemplo: bases de dados ou ficheiros de computadores.

⁶ EAI ó Enterprise Application Integration ó Equivalente a integração de sistemas de informação dentro da empresa ou entre empresas;

⁷ SI ó sistema de informação

ina, sendo um mainframe, servidor departamental,

computador pessoal, dispositivo móvel, máquina de análises, balança, telemóvel ou relógio; L03

Integração na empresa

Quando os SI residem em máquinas diferentes (não partilham a mesma memória), mas estão ligados através de uma rede de banda larga segura e fiável, de pequena latência; L03

Integração entre empresas

Quando os SI estão ligados ocasionalmente, quando existem restrições de segurança e/ou quando são geridos por entidades distintas; L03.

Redes e middleware

Nesta classificação, a integração é dividida em níveis: Redes, Middleware e Integração de sistemas de informação.

Integração de SI
Middleware
Redes

Ilustração 1 - Redes, middleware e integração de sistemas [L03]

Redes

Hardware e software necessários para troca de dados entre aplicações residentes em computadores distintos.

Middleware

Software que permite a comunicação entre duas aplicações. L03. Um middleware é o elemento intermédio entre duas aplicações essencialmente em termos de definir o fluxo de dados de negócio. Existe uma obrigatoriedade das aplicações que pretendem estabelecer a comunicação possuírem plataformas diferentes, armazenamentos de dados heterogéneos, e mesmo redes e protocolos de comunicação heterogéneos [W08], embora possam ocorrer no mesmo computador. L03.

A figura está em níveis, porque é assim que estes elementos se relacionam. O middleware usa as redes, enquanto que a integração de SI usa middleware; L03.

Facilidade

As tecnologias devem ser fáceis de perceber e aprender, como os produtos devem ser fáceis de utilizar, manter e alterar; L03

Transparência

O programador e/ou utilizador não deve ser obrigado a conhecer os detalhes de baixo nível, um conceito que também varia conforme o nível da integração. L03;

Aplicabilidade

As tecnologias devem resolver uma grande variedade de problemas, ou resolver apenas um problema específico, mas de forma inequívoca. L03;

Fiabilidade

Os sistemas devem falhar pouco e quando falham, devem demorar o mínimo tempo possível para recuperar o estado normal; L03.

Uma rede é fiável se a probabilidade de trocar dados com outra aplicação noutra computador for muito elevada.

Performance

Baseia-se na quantidade de dados trocados por segundo. Em redes, a largura de banda expressa esta capacidade.

Segurança

Inclue a autenticação e autorização de utilizadores (AAA em ingles, ãAccess, Authentication and Authorizationö), para acederem a determinados recursos e protecção para evitar acesso não autorizado.

Gestão

É necessário fazer a gestão de funcionamento no dia a dia de qualquer sistema.

Em redes, existem, a título de exemplo, os softwares Unicenter e Tivoli para gerir redes TCP/IP. L03.

Tipos de Integração

A integração de sistemas comporta cinco tipos de integração de sistemas, que são orientados a: dados, métodos, interfaces, portais e processos.[L03].

dos de uma base de dados (geralmente relacional) e colocação noutra base de dados, usando comando SQL.

Orientado a métodos

Consiste em obter um serviço remoto de uma aplicação, permitindo o acesso directo da lógica das aplicações do SI, como se um sistema de informação fosse uma extensão do outro. É usado entre linguagens de programação como COBOL, Basic ou Java e permite que se partilhe a lógica e dados sem ter que replicá-los, impedindo versões inconsistentes;

Orientado às interfaces

Utiliza interfaces com o utilizador (GUI)⁸, como ponto de entrada no sistema de informação. Simula-se o comportamento de um utilizador tanto para inserir como para obter dados.

Orientado aos portais

A integração orientada aos portais tira partido do facto da maioria das aplicações recentemente construídas já terem uma interface Web, que é relativamente fácil integrar a nível de interface com utilizador. Neste tipo de integração, um website é constituído por diversos websites.

Orientado a processos

Integração baseada nos processos permite aos SI comunicarem-se entre si ao nível do negócio, não ao nível tecnológico. Por exemplo, um sector de uma empresa recebe uma apólice e quer transmitir a outro sector. Pode-se digitalizar este documento e enviá-lo pela rede, para o outro sector.

Resumo

- I. A integração de sistemas permite que um sistema sirva como *input* de dados para outro sistema e vice-versa se necessário, permitindo partilha;
- II. A integração está condicionada ao mínimo de falhas, que os sistemas em causa devem ter, e a facilidade de entendimento que se deve obter nelas, sendo que o programador não necessita conhecer detalhes de baixo nível na maior parte dos casos para integrá-las.
- III. É crucial que se identifique de forma clara, o objectivo da integração;
- IV. É necessário garantir a segurança durante a integração de sistemas, e quanto melhor for a capacidade da rede, mais rápido e com maior qualidade vão trocar dados, bastando que se faça a gestão do processo de integração no dia-a-dia;

⁸ Graphical User Interface ó interface gráfica do utilizador ó representa o layout da página ou do programa que interage com o utilizador;

ção: de dados (base de dados), orientada a métodos (uma a), orientada a interfaces (a interface gráfica é o ponto de entrada), orientada a portais (um website é um aglomerado de websites) e orientada a processos, onde a integração é ao nível de negócio, não tecnológico);

O Protocolo LDAP

Historial

No fim dos anos 70 e inícios dos anos 80, a ITU (União Internacional das Telecomunicações) começou a trabalhar na padronização de e-mails, dando origem as recomendações X.400. [W01]. Desenvolvida em parceria com a ISO, a série de X.400 especifica os protocolos a usar quando se troca mensagens electrónicas.[W02]

Esta padronização precisava de um directório de nomes e outras informações, que pudessem ser acedidas de forma hierárquica, de modo similar ao DNS (domain name system).

Esta procura por um directório de acesso global para redes, levou a ITU a desenvolver a série de recomendações X.500, mais concretamente as recomendações X.519, que definem o DAP (direct access protocol), protocolo de acesso a directório, que é o protocolo de acesso a um serviço de directório⁹, em rede.

No início da década de 1990, a IETF (Internet Engineering Task Force) procurou desenvolver um padrão que permitisse aceder a serviço de directório, sem ter que utilizar o overhead¹⁰ do protocolo, e começou a desenvolver o LDAP (Lightweight Directory Access Protocol), desenhado para providenciar quase todas as funcionalidades das recomendações X.519, mas usando o protocolo TCP/IP, enquanto ainda permitia a interconexão com directórios baseados no protocolo X.500. O mapeamento com base no protocolo X.500 e a interconexão com este protocolo, fazem parte do RFC¹¹ que a IETF produziu para LDAP [W02] . Actualmente este protocolo está na versão LDAPv3.

Definição

Em termos técnicos, LDAP é um protocolo que define o método para o qual a informação é acedida. Define também a forma como os dados são representados no serviço de directório

⁹ Directório é estrutura para guardar arquivos. Arquivos ou ficheiros são registos que são guardados numa certa regra estrutural, geralmente dada pela extensão do ficheiro.

¹⁰ Durante a transmissão, nem todos os bits são reservados para informação. Os de overhead são reservados para roteamento da mensagem, descrever o conteúdo da mensagem, e outras necessidades.

¹¹ documento que descreve os padrões de cada protocolo da internet previamente a serem considerados um padrão.

como os dados são carregados (importados) e como são

Características

São vários aspectos que caracterizam o LDAP[W11]:

Desenhado para propósito geral

Usa um esquema de definição orientado a objectos, baseado em herança, que permite extensão para qualquer uso necessário;

Simplicidade do protocolo

Mais simples de implementar e trabalhar do que o DAP. É suportado pela maioria das linguagens de programação, como C, Java e Perl e pela maioria dos sistemas operativos, nomeadamente Solaris, GNU/Linux, Microsoft Windows, e o Mac OS.

Arquitectura distribuida

Com o uso de replicação de dados, é possível replicar todo ou parte de um directório LDAP para locais separados fisicamente, o que permite que os dados tenham alta disponibilidade.

Em 2008, no ISUTC, houve um problema relacionado com o servidor de LDAP. Os alunos não conseguiam aceder as suas contas de e-mail, nem aos computadores da escola. Com esta função, este cenário pode ser evitado; (E05), (E06).

Segurança

Com a versão actual do LDAP (v3), surgiram melhorias significativas. Existem três aspectos básicos de segurança em um directório: acesso, autenticação e autorização (AAA, ver secção 2.5.1.1ö).

Para acesso seguro, o LDAP suporta o Transport Layer Security (TLS), que criptografa toda a comunicação entre cliente e servidor. Para autenticação, o LDAP suporta a Simple Authentication and Security Layer (SASL), que permite que o cliente e servidor negociem um método de autenticação (seguro), estando a autorização a cargo de ACL (Access Control Lists ó listas de controle de acesso);

Padrão aberto

Pode ser utilizado por qualquer desenvolvedor, companhia, ou administrador, sem receio de protocolos proprietários;

[487, de Julho de 1993), contribui para aumentar o leque
[V03]

- **Mapeamento durante o serviço de transporte** ó este protocolo foi desenhado para correr em protocolos orientados a conexão, com transporte seguro;
- **TCP (Protocolo de Controle de Transmissão)** ó As implementações do servidor correndo em TCP devem providenciar escuta na porta 389;

Modelos de dados

LDAP define quatro modelos de dados que serão listados em seguida:

1. **Modelo de informação** ó este define como a informação é representada no LDAP.
2. **Modelo de denominação** ó define o caminho completo para nomear um atributo. Por exemplo, o modelo de denominação do utilizador do administrador do ldap do isutc é:
cn=admin,dc=isutc,dc=transcom,dc=co,dc=mz
3. **Modelo funcional** ó é o modelo que permite que se faça leituras, pesquisas, escritas ou modificações no LDAP.
4. **Modelo de segurança** ó pode-se atribuir um nível de autoridade a cada utilizador, permitindo identificar quem tem permissão para verificar ou modificar que atributo no serviço de directório.
[W01]

Comparação entre LDAP e uma base de dados relacional

- LDAP é caracterizada por ser do género ãescreve uma vez, lê muitasõ, enquanto que uma base de dados relacional é caracterizada por poder estar sempre a ser actualizada sem complicações. Normalmente é usado para informação que pode crescer, mas não alterar muito.
- Para se conectar ao LDAP, basta se conectar ao servidor, não importa se a versão de implementação é openLdap ou e-Directory, enquanto que numa base de dados relacional

- LDAP não é relacional. É possível identificar dados univocamente, tal e qual nas bases de dados relacionais, mas não é possível estabelecer uma relação entre os dados;

Arquitectura do LDAP

O LDAP possui um espaço de nomes, que é o modo de identificar univocamente cada entrada. Este espaço de nomes é hierárquico, como sugere o padrão X.500. As entradas estão organizadas num DIT (Directory Information Tree), consoante o seu nome distinto (DN ó Distinguished Name). O DN é o nome que identifica univocamente cada entrada. Cada DN é constituído por um RDN (Relative Distinguished Name), que representa cada ramo da árvore de localização. [W04]

Os tipos de atributos guardam informação do utilizador. [w10]. Existem vários tipos de atributos em LDAP:

c ó (countryName) ó é o atributo para guardar o nome do país;

cn ó (commonName) ó contém nomes de um objecto. Se for de uma pessoa, provavelmente é do nome completo da pessoa;

dc ó (domainComponent) ó contém normalmente o nome do domínio de dns;

givenName ó nome de um utilizador, sem apelido;

o ó (organizationName) ó contém nomes de uma organização;

ou ó (organizationalUnitName) ó contém nomes de uma unidade na organização;

sn ó (surname) ó um nome para o apelido da pessoa;

uid ó (userid) ó contém o nome do utilizador quando faz o login nos computadores;

userPassword ó contém password dos utilizadores;

¹² Sistema de Gestão de Base de dados

para humanos, com descrição sobre o objecto;

displayName ó nome preferencial ao mostrar uma entrada;

gidNumber ó número inteiro que identifica univocamente um grupo num domínio administrativo;

departamentNumber ó identifica um departamento dentro de uma organização;

mail ó endereço de e-mail dentro da entrada na organização;

uidNumber ó número inteiro que identifica univocamente um utilizador dentro no dominio administrativo;

Por exemplo, a arquitectura do LDAP da instituição é:

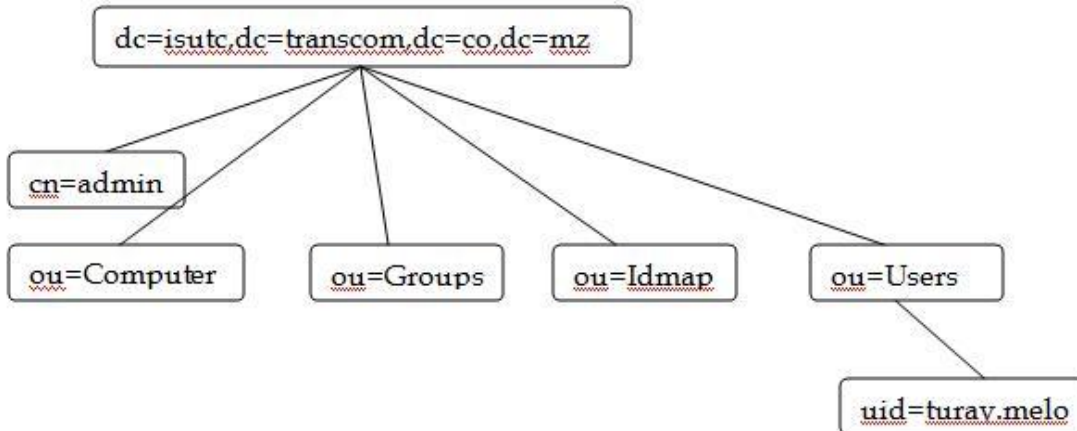


Ilustração 2 - Directório de informação do LDAP

O dn neste caso de turay.melo seria:

dc=isutic,dc=transcom,dc=co,dc=mz,ou=Users,uid=turay.melo

Operações

Conjunto de acções que o cliente pode fazer em relação ao servidor.

O colectivo de operações são apresentadas em seguida: [W12]

Bind ó autentica o utilizador e indica a versão do LDAP;

or devolva um conjunto de entradas que tenham relação

Modify ó permite que o servidor modifique a informação de utilizador;

Add operation ó permite que um utilizador solicite que a sua informação seja introduzida no LDAP;

Delete operation ó permite que um utilizador solicite ao serviço para apagar uma entrada sua;

Modify DN Operations ó permite que o utilizador peça ao serviço de RDN para modificar a localização da sua entrada ou mover uma sub-árvore da DIT (Directory Information Protocol).

Compare operation ó permite que se compare um valor introduzido com um valor específico do directório.

Abandon Operation ó permite que se cancele uma operação não terminada;

Unbind ó termina a sessão LDAP;

Funcionamento do LDAP

1. O cliente estabelece a sessão com o servidor LDAP. Este passo é conhecido como *õbindingõ*. O cliente especifica o nome do host ou o endereço IP do servidor e a porta onde este está à escuta. (Normalmente porta 389)
2. O cliente pode fornecer um user name e password para se autenticar perante o servidor (através de SSL¹³), ou estabelecer uma sessão anónima com permissões por defeito. Podem também ser estabelecidas sessões com métodos de segurança mais fortes, como encriptação de dados.
3. O cliente realiza então operações sobre os dados. O LDAP fornece operações de leitura e de actualização. Também permite pesquisas sobre determinados critérios (filtros). A pesquisa é uma operação frequente em LDAP. Um utilizador pode especificar que parte do directório pesquisar e que informação quer de volta.
4. Quando o cliente acaba, fecha a sessão (unbinding); [W04]

¹³ SSL- Secure Socket Layer ó software que constrói comunicação segura entre dois sockets[L04]. Sockets são os programas responsáveis pela interligação ou comunicação de outros programas na internet. (W18);

es a reter:

- I. O LDAP define como os dados são armazenados num serviço de directório;
- II. É compatível com Java, foi criado para um propósito geral, permite uma réplica do servidor e o acesso e autorização são feitos com camadas seguras;
- III. É organizado numa estrutura hierárquica, onde o modelo de denominação dá o DN (distinguished name), ou caminho completo;
- IV. É constituído por uma série de espaços de nome, sendo que os mais importantes são o uid (userId), o userPassword e o uidNumber.
- V. Como foi dito, é um protocolo que é mais utilizado para inserir uma vez e ler muitas, pelo que as operações mais utilizadas seriam então bind, search, e unbind;

Single-Sign-On

Introdução

O ISUTC tem várias aplicações e sistemas que precisam de autenticação, para que os utilizadores possam usufruir dos seus serviços. Desde o login aos computadores da instituição, ao uso do servidor de e-mail da mesma, passando pela aplicação ISUPAC3 ou mesmo o simples acesso a internet é feito mediante a introdução de uma senha e password. Se cada um destes sistemas tivesse a sua própria autenticação, cada utilizador teria que fixar cerca de quatro passwords diferentes caso não utilizasse o mesmo. Se com um sistema de autenticação único, existem alunos que acorrem ao sector de informática a reportarem a perda de password, com estes sistemas isolados, este número aumentaria drasticamente.

Por outro lado, os administradores das aplicações teriam que ter cada um um sistema de autenticação que seria ou uma base de dados, ou um servidor de autenticação. Estas bases de dados e servidores, teriam informação redundante.[W16]

Definição

Single Sign-On (SSO) é um mecanismo no qual, a mesma autenticação é utilizada para aceder todos os computadores e sistemas da rede de uma empresa ou de onde o SSO é implementado.

mas single-sign-on: [W17]

- **Server-based Intranet Single-Sign-On** ó permite que se integrem vários tipos de aplicações heterogéneas, onde os utilizadores tem um par de credenciais (username e password) para todas as aplicações dentro da empresa, introduzindo estas credenciais cada vez que se precise de aceder a uma das aplicações internas.
- **Windows Integrated Single-Sign-On** ó permite que se introduza o par de credenciais uma única vez na rede, e a partir do momento em que o utilizador é autenticado, pode aceder a todas outras aplicações internas sem precisar de se autenticar novamente;

Vantagens do Single-Sign-On

- **Aumenta a produtividade do utilizador** ó utilizadores não precisam fixar muitos usernames nem passwords, o que faz com que helpdesk receba poucas solicitações para reiniciar passwords para os utilizadores que esquecem; Torna-se mais cómodo ainda para o utilizador no caso de Windows Integrated Single-Sign-On, pois o utilizador digita uma única vez as suas credenciais e utiliza todas as aplicações sem ter que voltar a escrevê-las;
- **Aumenta a produtividade do desenvolvedor de aplicações** ó no caso de Windows Integrated Single-Sign-On, o desenvolvedor de aplicações não precisa de se preocupar com autenticação, pois a partir do momento em que um request (solicitação) vem acompanhado de um username, significa que esse utilizador já foi autenticado;
- **Administração simplificada** ó apenas tem que manter um servidor de autenticação. Outro factor é que as actualizações aos dados dos utilizadores são feitos somente em um único local. [W16]

Desvantagens

- **Abandono do ambiente** ó um utilizador mal intencionado pode ganhar acesso a um computador de um utilizador que sai do seu ambiente de trabalho sem fazer o logout. Embora seja um problema de segurança a nível geral, torna-se pior quando se trata de SSO do tipo Windows Integrated Single-Sign-On, principalmente. No caso Server-based

nuado porque o utilizador só ganha acesso a um dos
z com que somente um dos ambientes seja comprometido

de cada vez;

- **Single point of failure (SPOF)** ó A autenticação é centralizada, o que faz com que em caso de falha deste sistema, toda a instituição fique paralizada; [W16]. A situação referida no ponto 2.3.3.3 é um caso real desta desvantagem;

Resumo

- I. Single-sign-On basea-se em utilizar mesmas credenciais para todos programas;
- II. Com esta metodologia, os utilizadores não esquecem a todo o momento da senha, contudo, dada a centralização da autenticação, quando há falha deste sistema, todos os sectores colapsam. Daí a necessidade de existirem servidores replicados.

O framework Acegi (Spring Security)

Introdução

O Acegi ou *Spring Security* é um *framework* que é usado para segurança de aplicações J2EE¹⁴, construídas utilizando Spring. Foi criada em 2003, tendo posteriormente sido englobado no portfólio de Spring e surgiu Spring Security, um sub-projecto oficial.

Nos sistemas de segurança, a par do **Acesso** que representa a habilidade de conectar-se a um serviço, normalmente existem duas operações chave, autenticação e autorização. **Autenticação** que é o processo de estabelecimento do *õprincipalö*, ou seja, do utilizador, dispositivo ou sistema que irá realizar algum tipo de operação na aplicação. **Autorização** é o processo de decisão se é permitido que o *principal* realize ou não uma operação. Estes aspectos são comuns a sistemas de segurança, formando a sigla (AAA) e não específicos a Acegi, como foi visto na secção õ2.2.3.6ö e õ2.3.3.4ö.

O spring security autentica diversos tipos de tecnologias, desde bases de dados relacionais, não relacionais, LDAP e muito mais. Quando nenhum dos mecanismos satisfaz as necessidades do utilizador, permite que este escreva os seus próprios mecanismos de autenticação. Esta capacidade é utilizada na integração de sistemas, com sistemas obsoletos, que não seguem nenhum padrão de segurança.[W07]

¹⁴ Módulo da linguagem Java com bibliotecas que servem para facultar aplicações que corram no servidor

enciar a segurança no modo como um *principal* se
 exemplo assegurar que os pedidos do utilizador cheguem
 somente sobre HTTPS¹⁵, para proteger passwords de eavesdropping¹⁶ ou utilizadores do ataque
 ãman-in-the-middleö, no qual um cracker¹⁷ se posiciona no meio de dois utilizadores que se
 comunicam, e vai se passando por cada um deles e controlando a comunicação. Ou por exemplo,
 para se assegurar que quem está a realizar um request a aplicação é um humano e não um
 processo automático. Acegi utiliza õchannel securityö para esta funcionalidade [W07]. Contudo,
 estas funcionalidades não foram exploradas a nível deste projecto, devido ao facto de o interesse
 nesta ferramenta por parte do proponente centrar-se na autenticação e autorização que esta
 fornece em relação ao LDAP.

Spring Security é compatível com Java *runtime*¹⁸ 1.3, funcionando igualmente com versões mais
 avançadas como 5.0 ou 6.0. Possui um documento de referencia [W07], que fornece a
 informação necessária para a sua implementação.

Componentes

Existem certos procedimentos e objectos que são comuns na autenticação de todas as
 tecnologias suportadas por Acegi. Estes são chamados shared components.

Shared Components

O objecto fundamental do Acegi é o *SecurityContextHolder*. É onde são armazenados os
 contextos de segurança da aplicação, incluindo detalhes do *principal* que se encontra a utilizar a
 aplicação. Acegi utiliza um objecto *Authentication* para representar esta informação do *principal*.
 Normalmente pesquisa-se este objecto *Authentication*, com um código pré-definido pelo Acegi.

Outro objecto importante é o *SecurityContext*, que guarda informações sobre os requests não
 específicos ao *principal*.

O *principal* é um objecto que pode ser obtido a partir do objecto *Authentication*. Muitas vezes, o
principal pode ser um objecto concreto *UserDetails*. Outras vezes, como por exemplo, depois de
 passar pelo LDAP, o *principal* é um objecto específico *LdapUserDetailsImpl*.

Outro componente fulcral no processo de autenticação Acegi é o *UserDetailsService*. Esta é uma
 interface que tem um método que recebe um String de username, do utilizador que pretende

¹⁵ Implementação de HTTP segura, sobre uma camada SSL ou TLS

¹⁶ Técnica de hacking que se baseia na violação de confidencialidade, permitindo a leitura não autorizada de
 mensagens

¹⁷ Indivíduo que pratica quebra de sistema de segurança, de forma ilegal e sem ética

¹⁸ Ambiente que, com a máquina virtual de java (JVM), interpreta um código java

etails. É utilizada para construir um objecto
SecurityContextHolder.

O *Authentication* possui o método *getAuthorities()* que devolve um *GrantedAuthority*, representando um array de conjunto de autoridades que são concedidas ao *principal*. Estas autoridades são designadas por roles.

O último componente partilhado é o *HttpSessionContextIntegrationFilter*. Este é responsável por colocar o *SecurityContext* no *HttpSession*, para que este fique guardado entre dois requests. Spring Security recomenda que não se interaja directamente com o *HttpSession* e que se use o *SecurityContextHolder* ao invés, por motivos de segurança [W07].

Funcionamento

Autenticação

Em seguida se ilustra um diagrama do processo de autenticação de Acegi:

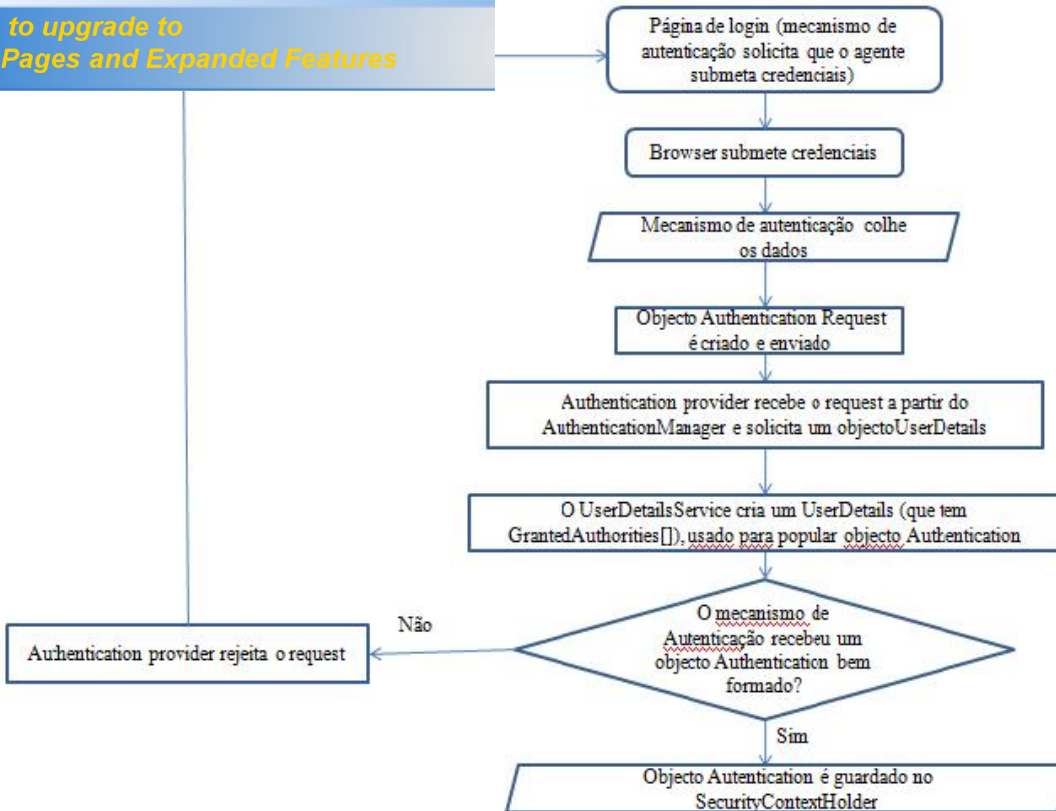


Ilustração 3 - processo de autenticação com Acegi - adaptado do [diagrama 1: Basic Flow for Authentication using Acegi](#) (W19)

Autorização

No processo de autorização, o sistema lê os roles (papéis que o *principal* pode desempenhar, ou níveis de acesso) guardados no `GrantedAuthorities[]` do *principal* autenticado e com base neles decide quais as acções que o utilizador pode ou não desempenhar.

Resumo

Os componentes mais importantes de Acegi são:

- I. *SecurityContextHolder*, providencia qualquer tipo de acesso ao *SecurityContext*.
- II. *SecurityContext*, para guardar um *Authentication* ;

- III. *HttpSessionContextIntegrationFilter*, para guardar o *SecurityContext* entre dois requests, utilizando o *HttpSession*.
- IV. *Authentication*, representa o *principal* numa forma específica de Acegi Security.
- V. *GrantedAuthority*, guarda as permissões do *principal*.
- VI. *UserDetails*, providencia informação suficiente para construir o objecto *Authentication*;
- VII. *UserDetailsService*, cria um *UserDetails* quando recebe o username;

Sistemas Biométricos

Introdução

A biometria, no âmbito informático, consiste no estudo de características físicas ou comportamentais de uma pessoa, como forma de identificá-la univocamente.

Os sistemas biométricos podem-se basear em características físicas ímpares do Homem: os olhos (a retina), a face, a palma da mão, as impressões digitais, as veias, ou em alguns comportamentos, como a voz ou a maneira de andar. (L07, p.4).

Livrateka é um sistema biométrico que se baseia na leitura de impressão digital, e é utilizada no âmbito de uma biblioteca universitária. A biometria é usada também na identificação criminal, controlo de acesso, segurança de instalações, etc.

Características de Sistema de reconhecimento de impressão digital

Existem muitos tipos de sistemas de reconhecimento, já mencionados anteriormente. Em seguida apresentar-se-á as características dos sistemas de impressão digital, que foi o tipo utilizado neste projecto: (L07, p.5).

la imagem da impressão digital e a sua identificação é um

- **Baixo custo** ó custo de aquisição de sensores de leitura são acessíveis (o utilizado nesta aplicação custou 40 USD, que seriam adicionados a 35 USD de licença por computador, traduzindo-se em 75 USD);
- **Mais difundido** ó em relação aos outros tipos mencionados;
- **Muito confiável** ó oferece níveis de segurança altos, uma vez que o padrão de impressão digital é único para cada indivíduo (taxa de falhas na identificação é 1:1000, pelo que o ISUTC, com registo de todos os estudantes e professores no activo, não registaria falhas, teoricamente);

Funcionamento

Sistemas biométricos funcionam com reconhecimento de padrões. Deve-se registar o utilizador previamente, de modo a que o sistema compare a autenticidade. Existem dois módulos de funcionamento, sendo o segundo antecedido do primeiro: (L07, p.4)

- **Módulo de registo** ó as características físicas são extraídas por um sensor biométrico e processadas pelo sistema, gerando um modelo, que é guardado normalmente numa base de dados;
- **Módulo de autenticação** ó no momento de acesso, o sensor recolhe novamente as características do utilizador, gera um novo template e compara com o anterior, ou anteriores, no caso de se terem registado vários;

Modelo de Processo de software

Introdução

Um modelo de processamento de software é uma representação abstracta de um processo de software. (L09, p.36). Esses modelos genéricos podem explicar diferentes abordagens do desenvolvimento de software. Em seguida, apresentar-se-á alguns dos modelos de processo, também conhecido por paradigmas de processo:

...ificação, desenvolvimento do projecto, implementação, validação e manutenção, como fases separadas e fundamentais ao processo. (L09, p.37).

Desenvolvimento evolucionário

Tem como base a ideia de desenvolver uma implementação inicial, expor o resultado ao comentário do utilizador e fazer seu aprimoramento por meio de muitas versões, até que tenha sido desenvolvido. (L09, p.39). Ao invés das actividades citadas na secção 2.7.2, a especificação, desenvolvimento e validação são executados concorrentemente para gerar um retorno rápido.

Desenvolvimento formal de sistemas

É uma abordagem do desenvolvimento de software que tem algo em comum com o modelo em cascata, mas cujo processo se baseia na transformação matemática formal de uma especificação do sistema em um programa executável. (L09, p.40).

Desenvolvimento orientado a reuso

O reuso é comum no projecto, embora de forma informal. Neste modelo, o reuso é formalizado com uma ampla base de componentes reutilizáveis e com alguma infra-estrutura de integração para estes componentes. (L09, p.41).

Capítulo 3

Desenvolvimento da aplicação

Introdução

Dada a particularidade de este projecto ser a continuidade e consolidação do projecto de Sérgio Langa referido no ponto 1.1, o desenvolvimento da aplicação esteve sempre conectado com os passos seguidos no projecto anterior, e havendo nesta secção, comparações entre as duas aplicações.

Inicialmente apresentar-se-á a arquitectura do sistema. Em seguida, o modelo de desenvolvimento do software. Após isto, mostrar-se-á a escolha das tecnologias e apresentar-se-á o plano de actividades inicial e a comparação com o que aconteceu na prática, e por fim far-se-á a comparação entre os casos de uso desenvolvidos agora e por S. Langa. O último ponto deste capítulo representará o custo do projecto.

a arquitectura do sistema é distribuída e em camadas.

Distribuída

Na aplicação existem vários subsistemas invisíveis a um utilizador normal, a trabalharem como um único, como é o caso do dispositivo biométrico que tem tratamento no cliente e o resto da aplicação tem tratamento no servidor. Outro caso é a autenticação com LDAP que é tratada noutro servidor.

Em camadas

Interface com utilizador

Trata dos aspectos da visualização e interacção com utilizador.

Modelo de negócio

Trata do funcionamento do sistema.

Acesso a dados

Trata do acesso, criação e actualização da base de dados

Estas três camadas dão a separação explícita do sistema, permitindo que o programador defina claramente o trabalho a realizar em cada uma das camadas.

Modelo de desenvolvimento do software

Devido ao facto de se estar a fazer uma melhoria a um projecto elaborado, e pelo facto das alterações terem sido feitas tanto a nível da especificação, quanto a nível de desenvolvimento, validação e ter-se pretendido implementar a fase de evolução, o trabalho efectuado neste projecto não teve uma forma de desenvolvimento sequencial. Dos modelos oficiais de processo de software especificados no ponto 2.7, nenhum foi adoptado na íntegra neste projecto.

Em algumas fases (secção 3.8.4, por exemplo) utilizou-se o desenvolvimento orientado a reuso, mas na maior parte dos casos, o desenvolvimento do projecto aconteceu, em termos de engenharia de software, sem seguir um modelo, embora o proponente tivesse um diagrama de actividades (secção 3.7).

Introdução

Nesta secção definir-se-á a estrutura base para o desenvolvimento. Só a partir deste ponto é que advém as tecnologias citadas no ponto 3.5.

Linguagem de programação

Java 1.6

Linguagem de programação de alto-nível, para qualquer Sistema Operativo, orientada a objectos. Compila-se uma vez e corre-se quantas quizer. Pode ser implementada sem custos de licenciamento. Muito utilizada para desenvolvimento de aplicações distribuídas. Tem suporte a construção de interface Web, acesso a base de dados, modelo de segurança, entre outros, no seu modelo Java Enterprise Edition (vulgo Java EE. (W21, W22, W23).

Java EE é um conjunto de tecnologias construídas em Java Standard Edition (a versão padrão) que visam reduzir o custo e complexidade de desenvolvimento de aplicações que correm do lado do servidor web. Fornecem uma boa plataforma para desenvolvimento de aplicações robustas. (W24).

Foi escolhida esta linguagem, devido as seguintes características:

a) Antecedentes

- É O início da aplicação desenvolvido por S. Langa foi criado com base na tecnologia java;
- É A aplicação possui um applet, no módulo de autenticação de utilizadores, que foi construído em java e obrigatoriamente a integração entre esse módulo e a aplicação teria que ser feita em java, já que Applets só correm em Java. (W25).

b) Aspectos técnicos

Para além dos aspectos técnicos referenciados na definição, pode-se ressaltar

- É A linguagem java é bem documentada;
- É Tem licença para uso gratuito GPL (W26);

c) Pessoais

concedido para elaboração de um projecto desta natureza, bem por ser utilizada pelo proponente a 4 anos;

Servidor Web

Tomcat 5.5.20

Servidor de aplicações java para web. É utilizado mesmo em ambiente de produção, dada a sua robustez. É um web container, ou seja, possui classes de apoio a programação web. Apoia a linguagem Java em aspectos de segurança web, e no desenvolvimento, com tecnologias como Java Server Pages (JSP) e Servlets. Também pode funcionar como servidor web, recebendo requests HTTP. Foi utilizado nesta aplicação como o servlet container e servidor web;

Foi escolhido este servidor devido ao facto de ser o servidor mais usado para Java, para além de possuir um módulo de autenticação para LDAP (W27), embora como se pode ver na secção 3.8.3, a autenticação não foi feita utilizando tomcat.

Actualmente encontra-se na versão 6.0.20. (W28). Utilizou-se a versão 5.5.20, pelo falta de tempo para investigar as novidades da versão supracitada, bem como possíveis incompatibilidades que podia ter com java 1.6. Para além deste motivo, pesou o facto de S. Langa ter iniciado o projecto com este servidor (reaproveitamento de tecnologias).

Sistema de Gestão de Base de dados

Mysql 5.0.24

Este é um sistema de gestão de base de dados (SGBD). É gratuito, usa a linha de comando e é um dos mais usados actualmente. Permite a conexão com a linguagem de programação java.

Para além do sistema já ter sido implementado com mysql anteriormente, outros factores que motivaram a escolha foram o preço benefício que traz em relação a oracle (livre), embora oracle seja mais robusta (W29). O mesmo factor contribue para escolha em relação a SQL server, pois este software também é proprietário (W30).

Fez-se a comparação de mysql com estas duas bases de dados pelo facto de serem as mais populares, segundo dados de 2004 (não se pesquisou mais pelo facto de não ser o objectivo do projecto). (W31).

Foi utilizada para guardar os dados referentes a atributos de livros, os templates de impressão digital, entre outros.

Outras tecnologias

VmWare

Este programa permite que se corra várias máquinas virtuais numa única máquina e física, permitindo a troca de recursos desse pequeno computador com muitos outros ambientes.

Máquinas virtuais diferentes podem correr diferentes sistemas operativos e várias aplicações diferentes, no mesmo computador. [W05]

OpenLDAP 2.2

OpenLDAP é um software open source de implementação do Lightweight Direct Access Protocol [W13]. Open source significa um software que engrandece o poder da distribuição livre (disponibilizando o código fonte das aplicações) e do código aberto e transparente, para que haja modificações, mas mantendo a integridade do autor. Foi criado pela OSI (Open Source Initiative).[W14].

phpLdapAdmin

phpLDAPAdmin é um cliente LDAP baseado na web. Tem o interface gráfico e permite que se façam os pedidos para o servidores [W15].

Eclipse 3.2

IDE¹⁹ de código aberto para construção de programas de computador. É compatível com a linguagem Java e foi utilizado no desenvolvimento desta aplicação como ambiente de programação.

¹⁹ *Integrated Development Environment* ou **Ambiente Integrado de Desenvolvimento**, é um programa de computador que reúne características e ferramentas de apoio ao desenvolvimento de *software* com o objectivo de agilizar este processo.

iva, pertencente a tecnologia Ajax, oferecendo uma interface gráfica web mais aprazível e com pouca codificação. Foi o framework utilizado para construir o interface gráfico da aplicação (todas as janelas da aplicação).

Spring Framework 2.08

Um framework de apoio ao J2EE do Java. Foi utilizado neste projecto para apoiar o acesso a base de dados e a pesquisa de utilizadores no LDAP, para envio de e-mails, escalonamento de funções e outras;

Acegi (Spring Security) 1.7

Utilizada para o modo de autenticação com LDAP. (Ver definição na secção 2.5)

Hardware

Computador

- marca HP, processador de 2GHz, Memória 2GB, Espaço 240GB;
- Sistema operativo Windows Vista Service pack 1, e Linux Ubuntu 8.04, incorporado numa máquina virtual (secção 3.5.1.1).

Leitor de impressão digital

- marca Microsoft FingerPrint Reader, Resolução 500dpi; [L07]

Plano de actividades

Tabela 1 - plano de actividades para o projecto

Actividades		Março				Abril				Maio					Junho				Julho	
		Semana				Semana				Semana					Semana				Semana	
		1	2	3	4	1	2	3	4	1	2	3	4	5	1	2	3	4	1	
1	Definição de casos de uso																			
2	Desenho da base de dados																			
3	Integração LDAP . Spring																			
4	Relatório da integração																			
5	Integração com biblioteca de SMS																			
6	Relatório da integração																			
7	Codificação use cases																			
8	Relatório da codificação																			
9	tempo para pontos em atraso																			
12	hosting para isunet e para fora																			
10	Testes da aplicação no ISUTC																			
11	Relatório dos testes																			
13	Melhoria resultante das criticas e teste																			
14	Entrega																			

	Em progresso
	Plano inicial
	Feito
	Não conseguiu

Como se pode observar, o decurso das actividades foi completamente diferente do que se tinha planeado. Actividades como desenho de base de dados, estavam a ser continuamente alteradas. A integração com a biblioteca de SMS demorou bastante devido aos erros que não se conseguiram ultrapassar. A codificação dos casos de uso atrasou-se devido as dificuldades nas outras actividades. Já o hosting é uma questão que ainda está pendente devido a aceitação ou rejeição do ISUTC em utilizar o software.

Desenvolvimento das actividades

Nesta secção falar-se-á do processo de desenvolvimento das tarefas referidas no plano de actividades. Não se fará um a exploração exaustiva sobre os passos efectuados, remetendo o código da aplicação para o anexo.

Definição dos casos de uso

Apesar de ser considerado no sentido lato definição de casos de uso, neste ponto analisou-se os casos de uso implementados por S. Langa e os requisitos funcionais e não funcionais necessários. Os diagramas de casos de uso que se apresentarão a diante, servem para representar requisitos. [L08]

Descrição do sistema proposto

Pretende-se autenticar o protótipo de S. Langa ao servidor LDAP, permitindo que os utilizadores submetam as mesmas credencias das outras aplicações da instituição. Pretende-se também que os casos de uso já implementados se ajustem a planilha de preenchimento de dados para livros, utilizada no CEDOC (ver anexo).

Actores

Antes de se abordar os requisitos funcionais e não funcionais, um elemento importante a analisar são os actores. Representam uma entidade externa que interage com o sistema[L08].

Requisitos funcionais

Descrevem o que o sistema faz ou é esperado que faça. Representa as acções e a interacção do sistema com os actores[L08]. Em seguida far-se-á uma comparação entre os requisitos funcionais implementados neste projecto e no de S. Langa. Em anexo se pode observar uma descrição mais extensiva dos mesmos e mesmo das bibliotecas utilizadas.

Tabela 2 - comparação de Requisitos funcionais entre estágios da aplicação

Requisito funcional	Aplicação desenvolvida por Sérgio Langa	Continuação da aplicação desenvolvida por Turay Melo
UC1 - Registar utilizador	Implementado	Retirado
UC2 - Registar autor	Implementado	Mantido
UC3 - Registar editora	Implementado	Mantido
UC4 - Registar livro	Implementado	Alterado
UC5 - Registar cópia de	Implementado	Alterado

	Implementado	Alterado
UC7 - Reservar livro	Implementado	Alterado
UC8 - Registrar estante	Não implementado	Implementado
UC9 - Requisitar livro	Implementado	Alterado
UC10 - Devolver livro	Implementado	Alterado
UC11 - Notificar utilizador	Não implementado	Implementado

Requisitos não funcionais

Estes descrevem a qualidade com que o sistema deverá fornecer os requisitos funcionais. Abrange medidas de desempenho como tempo de resposta, volume de dados ou condições de segurança [L08]. Em seguida apresenta-se novamente uma comparação entre os dois estágios da aplicação, quanto a este tópico:

Tabela 3 - Comparação de Requisitos não funcionais estágios da aplicação

Requisito não funcional	Aplicação desenvolvida por Sérgio Langa	Continuação da aplicação desenvolvida por Turay Melo
Autenticação na tela de entrada ao sistema	Implementado, fazendo autenticação a uma base de dados mysql;	Alterado, fazendo autenticação ao LDAP;
Logging	Não implementado	Implementado
Autenticação pelo dispositivo biométrico	Implementado	Mantido
Script para consulta automática de utilizadores com multa;	Não Implementado	Implementado
Algoritmo de pesquisa de impressões digitais	Implementado	Mantido ²⁰

²⁰ Mantido por falta de tempo, mas necessitando de análise (ver 3.8.7)

Diagrama de casos de uso

Em anexo se ilustrará o diagrama de casos de uso, actual e o diagrama de casos de uso de S. Langa. Os principais casos de uso se mantiveram, contudo o Registrar utilizador foi retirado e foram implementados mais dois use cases: o Registrar estante e o Notificar aluno.

Desenho da base de dados

O desenho da base de dados sofreu algumas alterações. Removeu-se a tabela dos utilizadores, pois agora os utilizadores estão registados no LDAP.

Introduziu-se também algumas tabelas:

- subtype;
- type;
- class;

Em anexo pode-se constatar o esquema da base de dados de S. Langa e o actual.

Integração LDAP-Spring

Resumo de passos realizados

- É Instalação de vmware (W05);
- É Instalação de Ubuntu (W32);
- É Cópia do programa openLdap existente no ISUTC (sector do limeaa);
- É Cópia do arquivo de directório replicado do LDAP (do limeaa);
- É Download da biblioteca log4j-1.2.9.jar e referência no classpath (W34) ;
- É Configuração do ficheiro log4j.properties (W35);
- É Utilização da biblioteca acegi-security-1.0.6.jar e configuração do ficheiro hopeContext.xml para autenticação com LDAP (em anexo);
- É Download da biblioteca spring-ldap-1.3.0.jar e colocação no classpath da aplicação; (W33)

Nesta actividade, foram utilizados os conhecimentos sobre LDAP, nomeadamente modelo de dados e tipos de atributos como ou, uid, cn, etc, para possibilitar a pesquisa (search), de atributos no LDAP e codificação dos métodos fulcrais para autenticação:

- É *UserDetails* *loadUserByUsername(String username)* - classe *UserSecurityManager* e *UserManager* em anexo;
- É *User* *getUserLoggedIn()* - classe *SecurityUtil* em anexo;

al criada pelo Vmware é que a sua rede suporta três

- Bridged(a máquina virtual é vista como um outro computador na rede, com IP obtido via DHCP);
- NAT (a máquina virtual se conecta ao computador host, que por sua vez se conecta à rede);
- Host-Only (a máquina virtual apenas se conecta ao host). [W06]

Durante o desenvolvimento de livrateka, utilizou-se o modo Bridged e o modo NAT. Desta forma consegue-se pôr a comunicar o sistema operativo windows com o sistema operativo linux e deste modo, aceder ao servidor LDAP na máquina virtual Ubuntu a partir do windows que é onde a aplicação se encontra.

Resultados

- A autenticação foi implementada com sucesso;

Dificuldades

- A grande dificuldade enfrentada foi na percepção do mecanismo de autenticação para LDAP via framework Acegi. Este impasse deveu-se ao facto de não se ter percebido que havia mecanismos comuns para todos os tipos de tecnologias (base de dados relacional ou LDAP) e devido a não se ter encontrado um exemplo de implementação que se adequasse a situação, na internet;
- Teve que se fazer instalações e procedimentos intermédios bastante morosos e não previstos inicialmente, para se chegar aos resultados finais; Desses inclui-se a instalação de VMware, sistema operativo Ubuntu;

Introdução

Foi feita uma tentativa de integração da aplicação com a biblioteca de envio de SMS proposta por Eurico da Silva no seu projecto de curso [L05], com vista a notificar aos estudantes com multa, que devem regularizar a situação.

Pré-requisitos

1. Celular com GSM modem
2. Software do celular
3. JDK 1.5
4. Java(tm) Communications API (/dependencies) (L05, anexo)

Teste do GSM Modem

Foram feitos os testes em conformidade com o guia de instalação (L05,anexo), com um telefone Nokia N70, e Nokia N73.

Utilização da biblioteca

1. Copia-se as bibliotecas sms2gsm.jar e comm.api para a aplicação, e instala-se o ficheiro comm.properties.
2. Instancia-se uma classe da biblioteca EasySMS e chama-se o método sendSMS utilizando como atributo o número da porta. [L05]

Resultado

O sistema responde dizendo sempre que a port não foi encontrada, mesmo após o teste de port ter tido um resultado positivo. Foi contactado o autor da biblioteca, mesmo assim não foi possível perceber o motivo da falha do sistema (E02).

Conclusão

- Não se conseguiu implementar a biblioteca na aplicação;

Introdução

Não se vai alongar em relação aos casos de uso, pois podem ser consultadas as alterações em anexo e na aplicação que se apresenta. Alteraram-se 11 requisitos funcionais, como se pode constatar no ponto 3.8.1.3. As codificações foram feitas maioritariamente utilizando a linguagem java, coadjuvada de Spring (para a lógica), o framework zk (para interface web) , estando a pesquisa e guarda de dados a cargo de mysql e LDAP.

Contudo, torna-se relevante falar dos requisitos não funcionais comunicados no ponto 3.8.1.4. Dos quatro analisados, o desenvolvimento de ãAutenticação na tela de entrada ao sistemaõ foi referido no ponto 3.8.3. Não se fez nada em relação a ãAutenticação na tela de entrada ao sistemaõ, pelo que não há nada a reportar, havendo, contudo para õScript para consulta automática de utilizadores com multaõ e ãAlgoritmo de pesquisa de impressões digitaisõ .

Script para consulta automática de utilizadores com multa

Introdução

Existem, em muitas aplicações, requisitos para escalonamento de tarefas para diversas acções, desde pesquisas a bases de dados, relatórios da actividade do processador do computador, e outras. Para realizar estas tarefas, muitos sistemas utilizam CRON, que é um sistema para apoio destas actividades, compatível com Unix/Linux. Neste trabalho, utilizou-se mais uma vez o framework Spring. (L06, p. 246).

Resumo de passos realizados

- Download da biblioteca mail-1.3.jar e activation-1.1.jar (W36) e referência no classpath;
- Colocação das bibliotecas citadas na pasta /Java/jre6/lib/ext;
- Download da biblioteca quartz-1.6.2.jar; (W37) e referência no classpath;
- Configuração da tarefa no ficheiro spring-config.xml; (L06, p. 246);
- Criação da classe RealEmailManager que contém os métodos para envio de e-mail;

Resultados

- Foi possível testar no ambiente do programador, a implementação de envio automático de e-mails a uma certa hora;
- Foi implementado o método para consulta de alunos com multa, mas ainda não foi comprovado;

loçada as bibliotecas na pasta destinada a bibliotecas do servidor web *õtomcatõ*, com vista a permitir que todas as aplicações (externas ao projecto em causa) pudessem também usufruir destas. Tal facto criou conflito e a aplicação não iniciava, dando sempre erro, informando que não conseguia encontrar classes específicas das bibliotecas.

Algoritmo de pesquisa de impressões digitais

Apesar de mantido (secção 3.8.1.4), foi constatado que o algoritmo utilizado na identificação de impressões digitais (secção 2.6.3, *õmódulo de autenticaçãoõ*) é de busca sequencial, pelo que no momento de pesquisa de impressões digitais para comparação, o sistema leva um a um template para análise, estando estes templates num vector de n posições.

Segundo Koffman e Wolfgang, (L10, p.109), o tempo total de execução deste algoritmo é directamente proporcional a dimensão do vector (que tem n posições), o que resulta num caso de performance $O(n)$ (L10, p.111).

õOõ é uma notação que se usa para analisar a performance de algoritmos (o seu tempo de resposta) (L10, p.111). A performance de algoritmos vai decrescendo se o algoritmo classificar-se como $O(1)$, $O(\log n)$, $O(n)$, $O(n \log n)$, $\mathcal{O}(2^n)$, $\mathcal{O}(n!)$, $\mathcal{O}(n^n)$ ou $O(n!)$, respectivamente. $O(1)$ acontece quando é um algoritmo que encontra o valor pretendido sempre a primeira vez. É impossível para o nosso caso.

Koffman e Wolfgang sugerem que se use o algoritmo de pesquisa binária, para se conseguir a performance $O(\log n)$. (L10, p.409).

Conclusão

- Foi analisado o algoritmo, criticou-se, propoz-se a solução, mas não se implementou;

Hosting para isunet e para fora

Ficou acordado que a aplicação ficaria alojada no servidor de preparação do ISUTC com IP (192.168.0.80), para efeiros de teste. Este é um servidor onde se alojam aplicações em fase de teste. (E03).

Recebeu-se ainda a informação que para se alojar a aplicação no servidor de produção, ter-se-ia que: (E04)

- Terminar a fase de testes;

io da aplicação;

- Formar-se uma comissão de jús para deliberar a viabilidade de livrateka no ISUTC;
- Ter um encontro com o Eng. Donis para deliberar o servidor a utilizar para a aplicação;
- Alojara aplicação;

Conclusão

- A aplicação foi alojada no servidor de preparação;

Testes da aplicação no ISUTC

Sessão 1

Foi realizada uma sessão de testes (E07), envolvendo o proponente e o bibliotecário, com vista a registrar livros.

Resultados:

- Foram registados 20 livros;
- O bibliotecário teve formação para utilização do software;

Conclusão

- A sessão de treinos ocorreu com sucesso;

Sessão 2

Foi realizada uma sessão de testes (E08) no CEDOC, envolvendo o proponente, 5 alunos e o bibliotecário.

Resultados

- Três alunos requisitaram dois livros cada um;
- Um aluno requisitou quatro livros;
- No momento em que se ia gravar a impressão digital, ocorreu um erro: o leitor de impressão digital não conseguia encontrar o endereço web para escrita;

Conclusão

- A sessão não foi realizada com sucesso;
- Um aluno requisitou quatro livros, o que devia ser alterado, já que no CEDOC só se pode levantar dois por aluno;

se uma investigação e resolveu-se o problema descrito na secção 3.8.9.2.1

Possível causa da falha

Ao fazer o deploy²¹ da aplicação no servidor, alterou-se o contexto da aplicação, de forma que o url para aceder, deixasse de ser 192.168.0.80:8080/libprint e passasse a ser <http://192.168.0.80:8080/livrateka>.

Custos da implementação do projecto

Os custos de implementação são:

Item	Descrição	Custo
Microsoft FingerPrint Reader	Leitor de impressão digital usado para registo e verificação de impressão digital dos utilizadores	40 USD
Licença de Griaule Biometrics SDK	Licença necessária para poder usar legalmente o SDK	35 USD/ computador
ZK Framework (Licença Standard)	Licença necessária para usar o ZK, sob licença comercial e não GPL.	100 USD
Acegi-Security	FrameWork necessário para implementar segurança no sistema.	Livre
Spring Framework	FrameWork necessário para desenvolvimento da aplicação	Livre
Tomcat	Servidor da aplicação necessário para funcionar o sistema.	Livre
Java	Linguagem de programação e software necessário para funcionar a aplicação	Livre

²¹ instalação da aplicação em um servidor de aplicações;

	cessário para o funcionar	Livre
Eclipse	IDE de desenvolvimento da aplicação.	Livre
1 computador	O computador utilizado pelo bibliotecário fica ocupado pelos alunos para impressão. Seria necessário mais um;	1000 USD

Custo: 1175 USD

Precisar-se-ia de dispender mais tempo na elaboração do sistema somente. Pelos casos de uso em falta, estima-se que um programador III precisaria de mais ou menos 100 horas de programação, dando:

Programação	Trabalho de um programador III durante 100 horas para terminar os casos de uso	417 USD
-------------	--	---------

Neste caso o custo total seria: **1192 USD**

Capítulo 4

Conclusões

- Não foi possível integrar a aplicação com a biblioteca de SMS, mesmo tendo-se falado com o autor e tentando-se efectuar experiências no ambiente deste;
- O protótipo foi integrado com LDAP e está alocado no ambiente de testes;
- realizaram-se duas sessões de testes, tendo sido registados 20 livros e tendo 4 alunos reservado 10 livros;
- A utilização de bibliotecas, exceptuando a de SMS, foi realizada com sucesso, e a sua importância é extrema, tendo sido provada na secção 3.7.6, com a criação de um script automático para envio de e-mails;
- O bibliotecário está capacitado a operar o sistema e 4 alunos também;
- Incrementou-se o conhecimento na área de desenvolvimento de software, dada a elevada documentação que teve de ser feita, e estudo de alguns capítulos e modelos de engenharia de software;

Dificuldades enfrentadas

- A maior dificuldade enfrentada na realização do trabalho residiu na falta de capacidade de planeamento de trabalho em relação ao tempo disponível para a realização do projecto.
- Outra grande dificuldade encontrada foi na percepção do mecanismo de autenticação para LDAP via framework Acegi. Este impasse deveu-se ao facto de não se ter percebido que havia mecanismos comuns para todos os tipos de tecnologias (base de dados relacional ou LDAP);

projecto de fim de curso de S. Langa para se perceber na integra algumas características iniciais do sistema, como a arquitectura, ou a montagem do ambiente de programação. É importante frisar que o projecto anterior estava mais virado para o desenvolvimento da aplicação e a sua integração com o dispositivo biométrico, enquanto que o projecto corrente preocupa-se mais com alteração de casos de uso, implementação de novos casos de uso e com integração com LDAP. Daí o motivo de não ter sido feita uma análise exaustiva a arquitectura do sistema, ou no sistema actual de gestão da biblioteca, pois, enquanto S. Langa parte do sistema manual para desenvolver um software protótipo, este projecto parte desta realidade para uma software ajustado às necessidades do ISUTC.

- Recomenda-se que se conceda uma extensão ao período para a realização deste projecto, de modo a que o mesmo seja concluído, dado que os quatro meses que a instituição fornece, aliados a intensa avaliação das disciplinas que são leccionadas em paralelo, mostram-se insuficientes para o término de um projecto deste nível. Respeitando o período inalterável, concedido para o projecto de fim de curso, e dada a motivação do proponente em terminar, solicita-se que este prolongamento de tempo seja encarado como uma actividade extra;
- Recomenda-se igualmente que o ISUTC leccione a cadeira de Gestão de Projectos antes do período do projecto de fim de curso, pois este parece, a primeira vista, ser crucial para planificação de um projecto melhor, principalmente em termos de definição clara das acções possíveis de implementar no período disponibilizado;

Web Sites:

- W01 - <http://www.zytrax.com/books/ldap/ch2/>
- W02 - <http://en.wikipedia.org/wiki/X.400>
- W03 - <http://www.ietf.org/rfc/rfc1487.txt?number=1777>
- W04 ó http://marco.uminho.pt/~osg/CC-I-LESI/ficha5/LDAP_pt.pdf
- W05 - <http://www.vmware.com/technology/virtualization.html>
- W06 - <http://pt.wikipedia.org/wiki/VMware>
- W07 ó <http://www.acegisecurity.org/reference.html>
- W08 - <http://www.middleware.org/>
- W09-<http://msmvps.com/blogs/richardwu/archive/2007/06/15/hyperterminal-for-vista.aspx>
- W10 - <http://tools.ietf.org/html/rfc4519>
- W11 - <http://www.onlamp.com/pub/a/onlamp/2001/08/16/ldap.html?page=2>
- W12 - <http://tools.ietf.org/html/rfc4511>
- W13 - <http://www.openldap.org/>
- W14 ó <http://www.opensource.org>
- W15 - http://phpldapadmin.sourceforge.net/wiki/index.php/Main_Page
- W16 - <http://www.ibm.com/developerworks/web/library/wa-singlesign/>
- W17 - <http://msdn.microsoft.com/en-us/library/aa745042.aspx>
- W18 - <http://www.scribd.com/doc/7473169/Aula-12-Sockets>, página 2
- W19 - [http://en.wikipedia.org/wiki/Acegi_security_framework_\(Java\)](http://en.wikipedia.org/wiki/Acegi_security_framework_(Java))
- W20 - <http://wgserver.sigmanet.com.br/users//jeraa/educacao/pg18fg30.jpg>
- W21 - <http://java.sun.com/docs/white/langenv/Intro.doc2.html#334>
- W22 - <http://www.java.com/en/about/>



PDF Complete

*Your complimentary
use period has ended.
Thank you for using
PDF Complete.*

[Click Here to upgrade to
Unlimited Pages and Expanded Features](#)

(linguagem de programa%C3%A7%C3%A3o)

- W24 - <http://java.sun.com/javase/technologies/index.jsp>
- W25 - <http://java.sun.com/applets/>
- W26 - <http://www.gnu.org/copyleft/gpl.html>
- W27 - <http://tomcat.apache.org/tomcat-4.1-doc/realms-howto.html>
- W28- <http://tomcat.apache.org/download-60.cgi>
- W29 - <http://qa.techinterviews.com/q/20060813003217AA6PGxN>
- W30 - http://searchsqlserver.techtarget.com/tip/0,289483,sid87_gci1196735,00.html
- W31 - http://www.davidtan.org/oracle-vs-mysql-vs-postgresql/pt_BR/
- W32 - <https://help.ubuntu.com/community/Installation>
- W33 - <http://www.springsource.com/download/>
- W34 - <http://logging.apache.org/log4j/1.2/download.html>
- W35 - <http://logging.apache.org/log4j/1.2/manual.html>
- W36 - www.java.sun.com
- W37 - www.opensymphony.com

Livros

- L01 ó SHARMA, Raul, STEARNS, Beth, NG, Tony, J2EE Connector Architecture and Enterprise Application Integration, Prentice Hall, 2001;
- L02 - DAVID S. Linthicum, Enterprise Application Integration, Addison-Wesley, 1999;
- L03- SILVA, Miguel Mira da, integração de sistemas de informação, 1 edição, Lisboa: FCA- editora informática, 2003
- L04 óTANENBAUM, Andrew S., 4ª edição, Brasil, Editora Campus, 2003;
- L05 - DA SILVA, projecto final de curso ó õDesenvolvimento de uma componente de envio de short message service (sms) para aplicações web.õ, Maputo: ISUTC, 2007;

L07 - LANGA, Sérgio, projecto final de curso - ãIntegração de dispositivos biométricos com aplicações - Caso de estudo: Integração de sensor de leitura de Impressão Digital com aplicação de gestão bibliotecária, Maputo: ISUTC, 2008;

L08 ó NUNES, M., O'NEIL, H., Fundamental de UML, 4ª edição, FCA-editora de informática, 2004;

L09 - SOMMERVILLE, Ian, Engenharia de Software, 6ª edição, Addison-Wesley, 2003;

L10 ó KOFFMAN, Elliot B., WOLFGANG, Paul A. T., Objects, Abstraction, Data Structures and Design Using Java, International Edition, Wiley, 2005;

Entrevistas

E01 ó Eduardo Jovo, 11 de Março, CEDOC, entrevista com o autor;

E02 ó Eng. Eurico da Silva, 13 de Abril, Maputo, programação com o autor;

E03 ó Eng. Carlos Donis, 20 de Maio, Sec-Inf, entrevista com o autor;

E04 ó Eng. Carlos Donis, 2 de Julho, Sec-Inf, entrevista com o autor;

E05 ó Eng. Elton Sixpence, 24 de Julho, LIMEAA, entrevista com o autor;

E06 ó Eng. Rafael Manhique, 25 de Julho, Maputo, troca de e-mails;

E07 ó Eduardo Jovo, 29 de Julho, CEDOC, sessão de testes;

E08 ó Staline Sofiano, Marcel Saraiva, Cassamo Dulobo, Filinto Vazula, Eduardo Jovo, 2 de Julho, CEDOC, sessão de testes;

ANEXO I ó Reunião com Engenheiro Donis

Acta da Reunião

Realizou-se uma reunião no dia 20 de Maio de 2009 no ISUTC, uma reunião com o objectivo de discutir a possibilidade de alojamento da aplicação em algum dos servidores do ISUTC.

Participantes:

- Turay Melo ó proponente do PFC;
- Eng. Carlos Donis ó sector de Informática;
- Eng. Elton Sixpence ó sector do LIMEAA;

Resultado da reunião:

Ficou acordado que, por ser esta uma aplicação ainda em estado de testes, deveria ser alocada no servidor de preparação do ISUTC (192.168.0.80).

ANEXO II ó desenhos das bases de dados

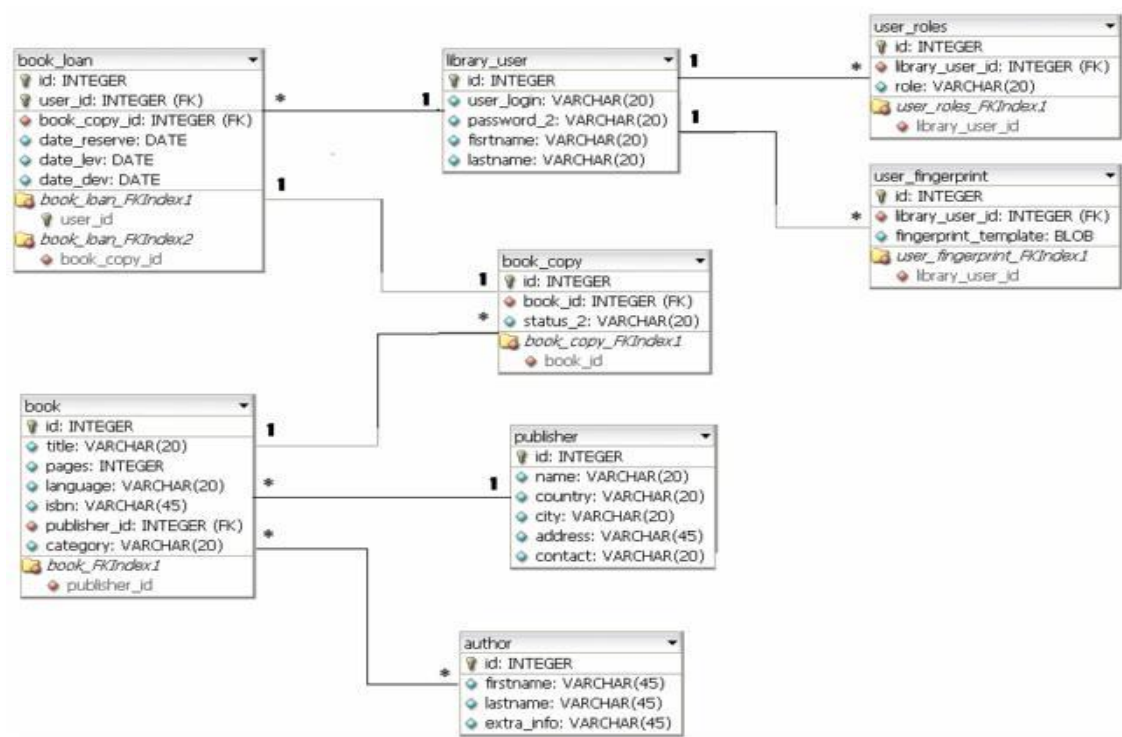


Figura 1 - Base de dados desenvolvida por Sérgio Langa

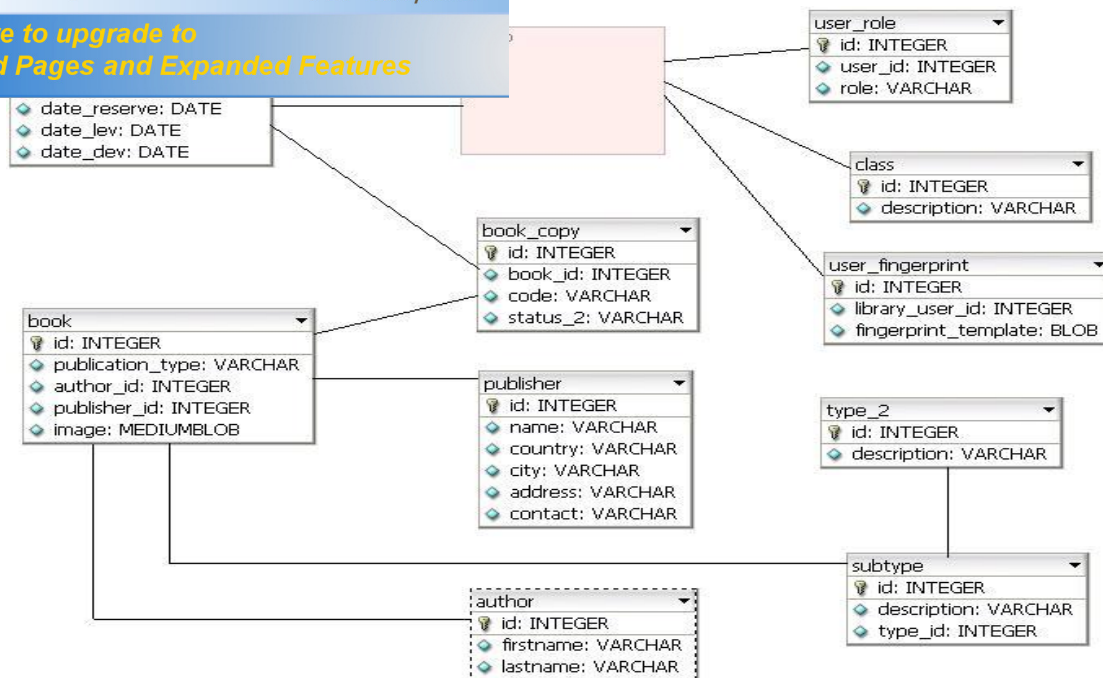


Figura 2 - desenho da base de dados desenvolvida por Turay Melo

ANEXO 6 bibliotecas utilizadas

Bibliotecas úteis	Aplicação desenvolvida por Sérgio Langa	Continuação da aplicação desenvolvida por Turay Melo
Bibliotecas para acesso a base de dados/ldap	Spring-jdbc.jar, que vem com o jar completo spring.2.08.jar	ldapbp-1.0.jar , spring-ldap-1.3.0.RELEASE-all.jar e spring-jdbc.jar
Biblioteca de envio de e-mail	spring-support.jar, que vem embutido no full spring.jar;	spring-support.jar, mail-1.3.jar e activation-1.1.jar colocadas no jre/lib/ext
Bibliotecas para escalonamento de tarefas	spring-support.jar, que vem embutido no full spring.jar;	spring-support.jar, quartz-1.6.2.jar;
Bibliotecas para logging	Nenhuma	log4j-1.2.9.jar

Tabela Anexo 1 - Bibliotecas utilizadas na programação



PDF
Complete

*Your complimentary
use period has ended.
Thank you for using
PDF Complete.*

[Click Here to upgrade to
Unlimited Pages and Expanded Features](#)

ANEXO ó Casos de Uso

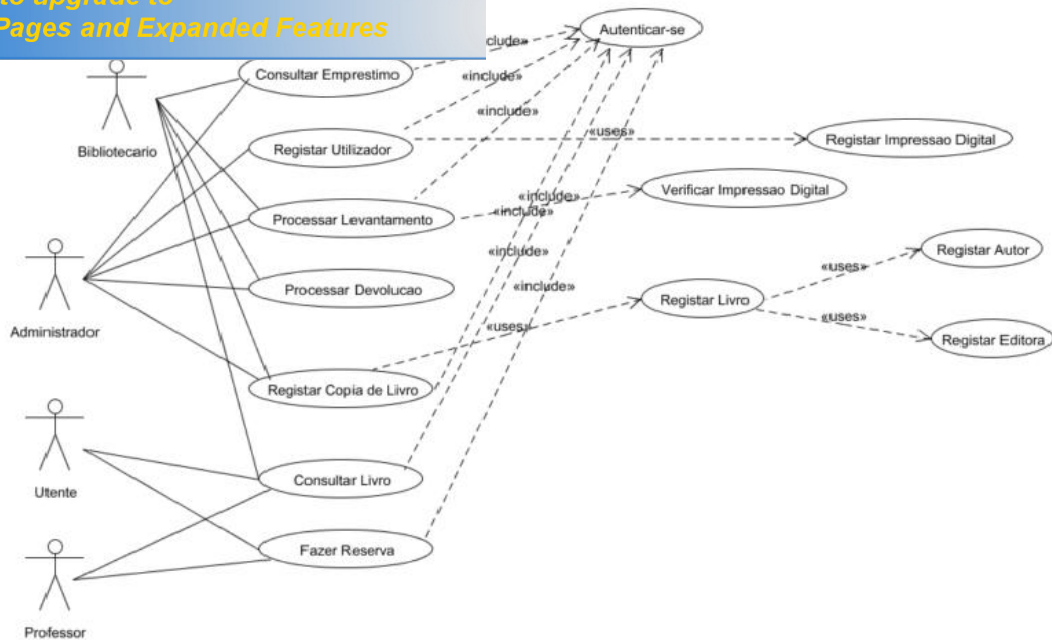


Figura 3 - diagrama de casos de uso por Sérgio Langa

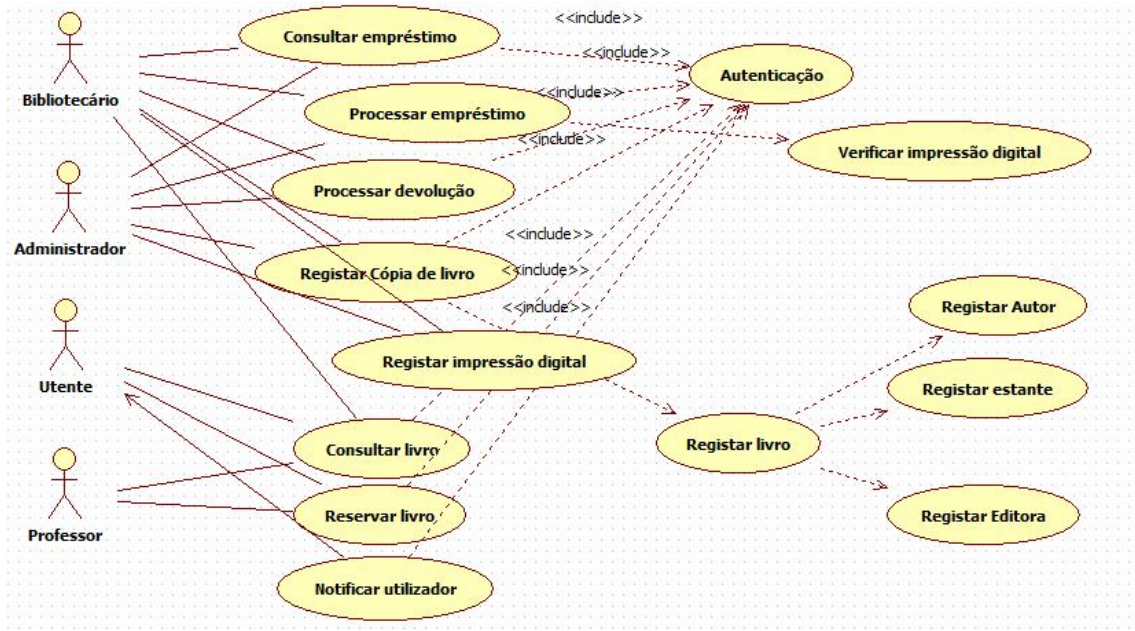


Figura 4 - diagram de casos de uso desenvolvidos por Turay Melo

Anexo

Acta de Reunião

No dia 11 de Maio de 2009, das 13h ó 14h, ocorreu um encontro para definição de uma forma de agrupamento de livros por estante, no CEDOC.

Participantes:

- Turay Melo
- Eduardo Jovo

Resultado:

Ficou concordado que se haveria de agrupar os livros consoante o tipo e a subclasse a que o livro pertence, seguindo a forma de agrupar nas prateleiras do cedoc.

ANEXO 6 CASOS DE USO

1. Caso de Uso Cancelar Reservas:

Âmbito:

É necessário permitir que os utilizadores, após efectuarem reservas na biblioteca, possam cancelá-las.

Actores:

Actor	Caso de Uso
Utilizador	Cancela a reserva

Descrição Estruturada:

Cancelar Reserva (Cenário Principal)	
Pré Condição	O utilizador já efectuou uma reserva
Descrição	<ol style="list-style-type: none"> 1. O use case começa quando o utilizador selecciona opção de cancelar reserva. 2. Aparece uma lista de livros reservados. 3. O utilizador escolhe o livro ou livros a cancelar. 4. O sistema retira da lista e apaga da base de dados de reserva
Caminho alternativo	A qualquer momento o utilizador pode abandonar esta janela de cancelamento da reserva.
Pós-Condição	O livro fica no estado livre, para que outro utilizador possa reservar

2. Alteração ao caso de uso "Consultar livros":

mas de registar livros, haverá uma divisão de tipos de
a utilizando estes filtros.

3. Alteração do caso de uso ãRegisto de livrosö:

- De acordo com o bibliotecário, para registo de livros no ISUTC, segue-se uma planilha. Esta será apresentada em anexo.

4. Calcular multas:

- É necessário calcular-se as multas para os alunos com livros atrasados. No momento da devolução de um livro, deve-se calcular a multa, com base no número de dias em atraso e de acordo com uma taxa fixa definida pelo ISUTC.

5. script para alterar o estado dos livros:

- É necessário que o sistema altere automaticamente o estado dos livros que foram requisitados, mas não foram levantados, logo na segunda-feira.

6. Enviar notificações para os utilizadores com multas:

Âmbito:

É necessário que o sistema tenha a capacidade de, utilizando os meios vigentes na instituição (e-mail) e os que se pretende integrar neste projecto, envie notificações em relação a multa a que os utilizadores com entregas atrasadas serão submetidos.

Actores

Actor	Caso de Uso
Sistema	Envia notificação

Descrição estruturada:

Condição	O utilizador fez um empréstimo de um livro e não devolveu a tempo
Descrição	1. O use case começa quando se atinge a data de entrega de livro 2. O sistema após calcular a multa, envia uma notificação via e-mail ou via SMS

ANEXO - Classe UserManager.java

```
package mz.co.slanga.library.manager;

import java.util.ArrayList;

import java.util.List;

import java.util.Map;


import javax.naming.NamingException;

import javax.naming.directory.Attribute;

import javax.naming.directory.Attributes;


import mz.co.slanga.library.bean.User;


import org.acegisecurity.GrantedAuthority;

import org.acegisecurity.GrantedAuthorityImpl;

import org.acegisecurity.userdetails.UserDetails;

import org.acegisecurity.userdetails.UsernameNotFoundException;
```



```
ception;  
  
import org.springframework.dao.EmptyResultDataAccessException;  
  
import org.springframework.ldap.core.AttributesMapper;  
  
  
public class UserManager extends AbstractManager {  
  
    Logger logger=Logger.getLogger(UserManager.class);  
  
    public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException,  
DataAccessException {  
  
        List<User> users = (List<User>)_ldap.search("ou=Users", "(uid=" + username + ")", new  
UserMapperLdap());  
  
        if(users.size()==0)  
        {return null;}  
  
        User ldapUser=users.get(0);  
  
        logger.debug("Consegui apanhar o utilizador "+ldapUser.getUsername()+" no metodo  
loadUserByUsername");  
  
        //Buscar permissoes do utilizador  
  
        setRoleAndFingerPrint(ldapUser, ldapUser.getStudentNumber());  
  
        return ldapUser;  
    }  
  
    //insere o role e verifica se o utilizador tem fingerprint gravada  
  
    public void setRoleAndFingerPrint(User ldapUser, int id)
```

```

try{

    //TODO: no forum de Spring sugerem que ao invés de try-catch, se use o metodo query. So que
    usa-se com um ResultSeExtractor. Ver como

    role=(String)_jdbc.queryForObject("SELECT role FROM user_role WHERE user_id='"+id+"'",
String.class);

}

catch(EmptyResultDataAccessException e)

{

    role="UTENTE"; //inicialmente não vamos colocar todos os utilizadores com role. Só o
bibliotecário e os professores

}

ldapUser.setHasFingerPrint( UserHasFingerPrint(id) );

GrantedAuthority[]authoritiesLdap={new GrantedAuthorityImpl(role)};

ldapUser.setAuthorities(authoritiesLdap);

}

//busca todos utilizadores de uma certa turma

public List<User> getUsersFromClass(String selectedClass) throws SecurityException

{

    List<User> users= (List<User>)_ldap.search("ou=Users", "(departmentNumber=" + selectedClass
+ ")", new UserMapperLdap());

    for(int i=0;i<users.size();i++) //Buscar permissões de cada utilizador

    {

        setRoleAndFingerPrint(users.get(i), users.get(i).getStudentNumber());

    }

}

```

```
//selecciona todas as turmas gravadas

public List<String> getAllClasses()

{

    List<Map> listOrderedMap = _jdbc.queryForList( " SELECT * FROM class");

    List<String> classes=new ArrayList<String>();

    for( Map map : listOrderedMap )

    {        String id = String.valueOf( map.get("id") );

            classes.add(id);

    }

    return classes;

}

//vai buscar um utilizador no LDAP tenha um certo id - o id antes pertencia a uma tabela. Agora
referenciará ao uidNumber do LDAP

public User getUserWithId(int id)

{

    //Buscar Utilizador no LDAP

    User user=(User)_ldap.search("ou=Users", "(uidNumber=" + id + ")", new
    UserMapperLdap()).get(0);

    //Buscar permissões do utilizador

    setRoleAndFingerPrint(user, id);

}
```

```

public User getUserWithFingerPrintId(Integer referenceTemplateId)
{
    //vai ver que utilizador tem relacao com esse templateID

    int uidNumber =Integer.parseInt((String)_jdbc.queryForObject("SELECT library_user_id FROM
user_fingerprint WHERE id='"+referenceTemplateId+"'", String.class));

    //Buscar Utilizador no servidor LDAP

    User user=(User)_ldap.search("ou=Users", "(uidNumber=" + uidNumber + ")", new
UserMapperLdap()).get(0);

    //Buscar permissoes do utilizador

    setRoleAndFingerPrint(user, referenceTemplateId);

    return user;
}

//verifica se um dado utilizador tem ou nao um registo de fingerprint

public boolean UserHasFingerPrint(int id)
{
    //TODO: Mudar Query

    List result =_jdbc.queryForList("SELECT id FROM user_fingerprint WHERE library_user_id="+id);

    if(result.size(>0) //pode ter mais do que uma impressao digital
    {
        return true;
    }
}

```

```
//=====dasses privadas
=====\\

//uma classe que representa um AttributesMapper - Spring fornece a interface e serve para apoiar a setar
atributos

private class UserMapperLdap implements AttributesMapper
{
    public Object mapFromAttributes(Attributes attrs) throws NamingException
    {
        String id = (String)attrs.get("cn").get();
        String fullName = (String)attrs.get("displayName").get();
        int studentNumber = Integer.parseInt((String)attrs.get("uidNumber").get()); //TODO
        String classId = attrAsString(attrs, "departmentNumber");
        String email = attrAsString(attrs, "mail");
        User user = new User(id, fullName, studentNumber, classId, email);
        return user;
    }
}

private String attrAsString(Attributes attrs, String attrName) throws NamingException
{
    Attribute attr = attrs.get(attrName);
    return attr == null ? null : (String)attr.get();
}}
```



PDF
Complete

*Your complimentary
use period has ended.
Thank you for using
PDF Complete.*

[Click Here to upgrade to
Unlimited Pages and Expanded Features](#)

ANEXO ó A classe SecurityUtil.java

```
package mz.co.slanga.library.util;
```

```
import javax.naming.NamingException;
```

```
import javax.naming.directory.Attribute;
```

```
import javax.naming.directory.Attributes;
```

```
import javax.naming.directory.BasicAttributes;
```

```
import mz.co.slanga.library.manager.AbstractManager;
```

```
import org.acegisecurity.GrantedAuthority;
```

```
import org.acegisecurity.GrantedAuthorityImpl;
```

```
import org.acegisecurity.context.SecurityContextHolder;
```

```
import org.acegisecurity.userdetails.UserDetails;
```

```
import org.acegisecurity.userdetails.Idap.LdapUserDetailsImpl;
```

```
import org.springframework.dao.EmptyResultDataAccessException;
```

```
/**
```

```
*
```

```
* @author Sérgio Langa e Turay Melo
```

```
* esta classe serve de apoio a questões de segurança
```

```
*/
```

```
public class SecurityUtil extends AbstractManager{
```

```
public SecurityUtil()
```

```
{
```

```
}
```

```
/**
```

```
* @author Sérgio Langa e Turay Melo
```

```
* @return devolve um utilizador que já submeteu credenciais válidas ao sistema, e que já está logado ao sistema
```

```
* @throws NamingException
```

```
*/
```

```

Object obj=SecurityContextHolder.getContext().getAuthentication().getPrincipal();

if(obj instanceof UserDetails)
{
    LdapUserDetailsImpl realUser=(LdapUserDetailsImpl)obj;

    BasicAttributes attrs=(BasicAttributes)realUser.getAttributes();

    String id = (String)attrs.get("cn").get();

    String fullName = (String)attrs.get("displayName").get();

    int studentNumber = Integer.parseInt((String)attrs.get("uidNumber").get()); //TODO

    String classId = attrAsString(attrs, "departmentNumber");

    String email = attrAsString(attrs, "mail");

    User user = new User(id, fullName,studentNumber, classId, email);

    GrantedAuthority[]authoritiesLdap={new GrantedAuthorityImpl(getRole(studentNumber))};

    user.setAuthorities(authoritiesLdap);

    return user;
}

return null;
}

/**
 * @author Turay Melo
 * @param id - id do utilizador que procuramos o role

```


que pretendemos

```
private String getRole(int id)
{
    String role="";
    try{
        //TODO: no forum de Spring sugerem que ao invés de try-catch, se use o metodo query. So que
        usa-se com um ResultSeExtractor. Ver como fazer
        role=(String)_jdbc.queryForObject("SELECT role FROM user_role WHERE user_id='"+id+"'",
String.class);
    }
    catch(EmptyResultDataAccessException e)
    {
        role="UTENTE"; //inicialmente não vamos colocar todos os utilizadores com role. Só o
        bibliotecário e os professores
    }
    return role;
}
```

```
private String attrAsString(Attributes attrs, String attrName) throws NamingException
{
    Attribute attr = attrs.get(attrName);
    return attr == null ? null : (String)attr.get();
}
```

/**

- * Verifica se um dado utilizador tem uma certa permissao
- * @param user
- * @param authority

```
public boolean hasAuthority(User user, String authority)
{
    for(GrantedAuthority auth: user.getAuthorities())
    {
        if( auth.getAuthority().equals(authority) )
        {
            return true;
        }
    }
    return false;
}
}
```

ig-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN"
"http://www.springframework.org/dtd/spring-beans.dtd">
<beans>
    <!-- 1.1. Configuracao para a DB -->
    <bean id="connPool" class="org.apache.commons.dbcp.BasicDataSource">
        <property name="url"
value="jdbc:mysql://localhost:3306/library?zeroDateTimeBehavior=round"/>
        <property name="driverClassName" value="com.mysql.jdbc.Driver"/>
        <property name="username" value="library"/>
        <property name="password" value="library"/>
        <property name="maxActive" value="20"/>
        <property name="maxIdle" value="10"/>
        <property name="maxWait" value="5000"/>
    </bean>

    <!-- 1.1. DB -->
    <bean id="jdbc" class="org.springframework.jdbc.core.JdbcTemplate">
        <constructor-arg ref="connPool"/>
    </bean>

    <bean id="ldapContextSource"
class="org.springframework.ldap.core.support.LdapContextSource">
        <property name="url" value="ldap://192.168.197.128:389"/>
        <property name="base" value="dc=isutc,dc=transcom,dc=co,dc=mz" /> <!--
queries camecam daqui -->
        <property name="anonymousReadOnly" value="true"/>
    </bean>

    <bean id="ldap" class="org.springframework.ldap.core.LdapTemplate">
        <constructor-arg ref="ldapContextSource" />
    </bean>

    <!-- 2.1 configurar os Managers -->
    <bean id="userManager" class="mz.co.slanga.library.manager.UserManager">
        <property name="jdbcTemplate" ref="jdbc"/>
        <property name="ldapTemplate" ref="ldap"/>
    </bean>
    <bean id="bookManager" class="mz.co.slanga.library.manager.BookManager">
        <property name="jdbcTemplate" ref="jdbc"/>
    </bean>
    <bean id="bookLoanManager"
class="mz.co.slanga.library.manager.BookLoanManager">
        <property name="jdbcTemplate" ref="jdbc"/>
    </bean>
    <bean id="fingerPrintManager"
class="mz.co.slanga.library.manager.FingerPrintManager">
        <property name="jdbcTemplate" ref="jdbc"/>
    </bean>

    <!--3.1 Beans Usado na Seguranca-->
```

```
mz.co.slanga.library.util.SecurityUtil">
    ref="jdbc"/>
```

```
<!--4.1 Beans usados para escalonamento de tarefas -->
<bean id="reminderEmailJobDetail"
class="org.springframework.scheduling.quartz.MethodInvokingJobDetailFactoryBe
an">
<property name="targetObject" ref="emailManager"/>
<property name="targetMethod" value="sendMail"/>
</bean>

<bean id="reminderEmailJobTrigger"
class="org.springframework.scheduling.quartz.CronTriggerBean">
    <property name="jobDetail" ref="reminderEmailJobDetail"/>
    <property name="cronExpression" value="0 30 16 ? * 5"/>
</bean>

<bean class="org.springframework.scheduling.quartz.SchedulerFactoryBean">
    <property name="triggers">
        <list>
            <ref bean="reminderEmailJobTrigger"/>
        </list>
    </property>
</bean>

<bean id="mailSender"
class="org.springframework.mail.javamail.JavaMailSenderImpl">
    <property name="host" value="mail.transcom.co.mz"/>
    <property name="username" value="turay.melo"/>
    <property name="password" value=" "/>
</bean>

<bean id="emailManager"
class="mz.co.slanga.library.manager.RealEmailManager">
    <property name="mailSender" ref="mailSender"/>
    <property name="mailMessage" ref="mailMessage"/>
</bean>

<bean id="mailMessage" class="org.springframework.mail.SimpleMailMessage">
    <property name="from" value="turay.melo@isutc.transcom.co.mz"/>
    <property name="subject" value="Voce tem multa no Cedoc"/>
    <property name="text" value="PFC - este e so um teste... se conseguires
receber... fico contente"/>
</bean>

</beans>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN"
"http://www.springframework.org/dtd/spring-beans.dtd">

<!--
    Ficheiro adaptado da configuração de uma aplicação sample que acegi
    disponibiliza
                                denominada acegi-security-samples-contacts-1.0.7.war
-->

<beans>

    <!-- ===== FILTER CHAIN ===== -->

    <!-- if you wish to use channel security, add
    "channelProcessingFilter," in front
        of "httpSessionContextIntegrationFilter" in the list below -->
    <bean id="filterChainProxy"
class="org.acegisecurity.util.FilterChainProxy">
        <property name="filterInvocationDefinitionSource">
            <value>
                CONVERT_URL_TO_LOWERCASE_BEFORE_COMPARISON
                PATTERN_TYPE_APACHE_ANT

/**=httpSessionContextIntegrationFilter,logoutFilter,authenticationProcessing
Filter,exceptionTranslationFilter
                <!--tinha ,filterInvocationInterceptor -->
            </value>
        </property> <!--estes são filtros que sempre devem aparecer -->
    </bean>

    <!-- ===== AUTHENTICATION ===== -->

    <bean id="authenticationManager"
class="org.acegisecurity.providers.ProviderManager">
        <property name="providers">
            <!-- sempre tem que ter um autentication manager. Para jdbc é o
daoAuthenticationManager
            que te permite instanciar uma classe userDetailsService, para melhor
manipulares as consultas na base de dados -->
            <list>
                <ref local="ldapAuthenticationProvider"/>
            </list>
        </property>
    </bean>

    <!-- configurações exclusivas do ldap -->
    <bean id="initialDirContextFactory"
class="org.acegisecurity.ldap.DefaultInitialDirContextFactory">
```

```

;9/dc=isutc,dc=transcom,dc=co,dc=mz"/>

name="managerDn"><value>cn=admin,dc=isutc,dc=transcom,dc=co,dc=mz</value></pr
operty>
  <property name="managerPassword"><value>limeaa</value></property>
  <!-- <property name="managerPassword"><value>DFE-855</value></property> -
->
  </bean>

  <bean id="ldapAuthenticationProvider"
class="org.acegisecurity.providers.ldap.LdapAuthenticationProvider">
  <constructor-arg> <!-- como se pode ver, esta forma de autenticação é
fazendo bind como manager -->
    <bean
class="org.acegisecurity.providers.ldap.authenticator.BindAuthenticator">
    <constructor-arg><ref
local="initialDirContextFactory"/></constructor-arg>
    <!-- <property
name="userDnPatterns"><list><value>uid={0},ou=people</value></list></property>
> -->
    <property
name="userDnPatterns"><list><value>uid={0},ou=Users</value></list></property>
    </bean>
  </constructor-arg>
  <constructor-arg>
    <bean
class="org.acegisecurity.providers.ldap.populator.DefaultLdapAuthoritiesPopul
ator">
    <constructor-arg><ref
local="initialDirContextFactory"/></constructor-arg>
    <constructor-arg><value>ou=Users</value></constructor-arg> <!--
*****NAO SEI QUAL E O INICIAL DIRCONTEXT estava ou=groups***** -->
    <property name="groupRoleAttribute"><value>ou</value></property>
    </bean>
  </constructor-arg>
</bean>

  <!-- Automatically receives AuthenticationEvent messages -->
  <bean id="loggerListener"
class="org.acegisecurity.event.authentication.LoggerListener"/>

  <bean id="httpSessionContextIntegrationFilter"
class="org.acegisecurity.context.HttpSessionContextIntegrationFilter">
  </bean>

  <bean id="logoutFilter" class="org.acegisecurity.ui.logout.LogoutFilter">
    <constructor-arg value="/login.jsp"/> <!-- URL redirected to after
logout -->
    <constructor-arg>
      <list>
        <bean
class="org.acegisecurity.ui.logout.SecurityContextLogoutHandler"/>
      </list>
    </constructor-arg>

```

```
<!-- ===== HTTP REQUEST SECURITY ===== -->

<bean id="exceptionTranslationFilter"
class="org.acegisecurity.ui.ExceptionTranslationFilter">
  <property name="authenticationEntryPoint"><ref
local="authenticationProcessingFilterEntryPoint"/></property>
</bean>

<bean id="authenticationProcessingFilter"
class="org.acegisecurity.ui.webapp.AuthenticationProcessingFilter">
  <property name="authenticationManager"><ref
bean="authenticationManager"/></property>
  <property
name="authenticationFailureUrl"><value>/login.jsp?login_error=1</value></prop
erty> <!-- PAGINA DE ERRO DE LOGIN -->
  <property name="defaultTargetUrl" value="/zul/home.zul"/>
  <property
name="filterProcessesUrl"><value>/j_acegi_security_check</value></property>
</bean>

<bean id="authenticationProcessingFilterEntryPoint"
class="org.acegisecurity.ui.webapp.AuthenticationProcessingFilterEntryPoint">
  <property name="loginFormUrl"><value>/login.jsp</value></property><!--
PAGINA DE LOGIN -->
  <property name="forceHttps"><value>>false</value></property>
</bean>

<!--
  <bean id="httpRequestAccessDecisionManager"
class="org.acegisecurity.vote.AffirmativeBased">
  <property
name="allowIfAllAbstainDecisions"><value>>false</value></property>
  <property name="decisionVoters">
    <list>
      <ref bean="roleVoter"/>
    </list>
  </property>

  <property name="decisionVoters">
    <list>
      <ref local="roleVoter"/>
    </list>
  </property>
</bean>
-->

  <!-- Note the order that entries are placed against the
objectDefinitionSource is critical.
  The FilterSecurityInterceptor will work from the top of the list
down to the FIRST pattern that matches the request URL.
  Accordingly, you should place MOST SPECIFIC (ie a/b/c/d.*)
expressions first, with LEAST SPECIFIC (ie a/.*) expressions last
```



PDF
Complete

Your complimentary
use period has ended.
Thank you for using
PDF Complete.

[Click Here to upgrade to
Unlimited Pages and Expanded Features](#)

```
onInterceptor"
    <property name="authenticationManager"><ref
    </property>
    <property name="accessDecisionManager"><ref
    </property>
    <property name="httpRequestAccessDecisionManager"><ref
    </property>
    <property name="objectDefinitionSource">
        <value>
            CONVERT_URL_TO_LOWERCASE_BEFORE_COMPARISON
            \A/secure/super.*\Z=ROLE_WE_DONT_HAVE
            \A/secure/.*\Z=ROLE_SUPERVISOR,ROLE_TELLER
        </value>
    </property>
</bean> *****EXPERMINTAR ELIMINAR ESTA PARTE
*****
```

BASIC Regular Expression Syntax (for beginners):

\A means the start of the string (ie the beginning of the URL)
 \Z means the end of the string (ie the end of the URL)
 . means any single character
 * means null or any number of repetitions of the last expression
 (so .* means zero or more characters)

Some examples:

```
Expression: \A/my/directory/.*\Z
Would match: /my/directory/
              /my/directory/hello.html
```

```
Expression: \A/.*\Z
Would match: /hello.html
              /
```

```
Expression: \A/.*/secret.html\Z
Would match: /some/directory/secret.html
              /another/secret.html
Not match:  /anothersecret.html (missing required /)
```

```
-->
</beans>
```