



INSTITUTO SUPERIOR DE TRANSPORTES E COMUNICAÇÕES

**Desenvolvimento de um módulo para gestão de contas de utilizadores no
Instituto Superior de Transportes e Comunicações**

Yuri Jorge Jimes Wingester

Projecto Final do Curso

Licenciatura em Engenharia Informática e de Telecomunicações

Supervisor:

Eng. Micas Rafael

Departamento das Tecnologías de Informação e Comunicação

Julho 2013



INSTITUTO SUPERIOR DE TRANSPORTES E COMUNICAÇÕES

**Desenvolvimento de um módulo para gestão de contas de utilizadores no
Instituto Superior de Transportes e Comunicações**

Yuri Jorge Jimes Wingester

Projecto Final do Curso

Licenciatura em Engenharia Informática e de Telecomunicações

Supervisor:

Eng. Micas Rafael

Departamento das Tecnologías de Informação e Comunicação

Julho 2013



**Desenvolvimento de um módulo para gestão de contas de utilizadores no
Instituto Superior de Transportes e Comunicações.**

Yuri Jorge Jimes Wingester

ÍNDICE

AGRADECIMENTOS	IV
DEDICATÓRIA	V
DECLARAÇÃO DE HONRA	VI
ÍNDICE DE TABELAS	VII
LISTA DAS ABREVIATURAS UTILIZADAS	X
RESUMO	XI
CAPÍTULO 1 - INTRODUÇÃO.....	1
1.1 Introdução	1
1.2 Justificação do tema.....	1
1.3 Desenho teórico	1
1.3.1 Problemática	1
1.3.2 Problema de investigação	2
1.3.3 Objecto de investigação.....	2
1.3.4 Objectivo geral	2
1.3.5 Objectivos específicos	3
1.3.6 Perguntas da investigação.....	3
1.4 Metodologia.....	3
1.4.1 Abordagem da investigação	3
1.4.2 Desenho da investigação	3
1.4.3 Ideia a defender	4
1.4.4 Métodos de investigação	4
1.4.5 Resultados esperados de investigação	4
1.5 Resumo	4
CAPÍTULO 2 - MARCO TEÓRICO-CONCEITUAL DA INVESTIGAÇÃO.....	5
2.1 Introdução	5
2.2 Processo de <i>software</i>	5
2.3 Modelos de processo de <i>software</i>	6
2.4 Modelo Espiral	6
2.5 Engenharia de requisitos.....	8
2.5.1 Tipos de requisitos.....	9
2.5.2 Levantamento ou elicitação de requisitos.....	9

2.6	Arquitectura de <i>software</i>	10
2.7	Técnicas de modelação UML	10
2.7.1	Diagrama de Casos de uso.....	11
2.7.2	Diagrama de classes.....	11
2.7.3	Diagrama de sequência.....	12
2.7.4	Diagrama de Actividades	13
2.8	Protocolo LDAP	15
2.8.1	Modelos de dados	16
2.8.2	Comparação entre LDAP e uma base de dados relacional	17
2.9	Resumo	17
CAPÍTULO 3 - MARCO CONTEXTUAL DA INVESTIGAÇÃO		18
3.1	Introdução	18
3.2	OpenLDAP	18
3.3	Conta de utilizador no ISUTC	19
3.4	Gestão de contas de utilizadores no ISUTC	20
3.5	Processos para a gestão de contas de utilizadores	20
3.5.1	Criação de Contas de utilizadores	20
3.5.2	Actualização de contas de utilizadores	22
3.5.3	Remoção de contas de utilizadores.....	23
3.5.4	Recuperação de senhas	24
3.6	Avaliação dos processos actuais.....	25
3.7	<i>OPUS-College</i>	25
3.8	Resumo	28
CAPÍTULO 4 - METODOLOGIA DE RESOLUÇÃO DO PROBLEMA E		
APRESENTAÇÃO DE RESULTADOS		29
4.1	Introdução	29
4.2	Porque desenvolver um módulo para a gestão de contas de utilizadores e não uma aplicação isolada?	29
4.3	Metodologia de desenvolvimento.....	29
4.3.1	Norma ISO/IEC 12207	29
4.3.2	Fases do projecto	31
4.3.3	Cronograma de desenvolvimento	32
4.4	Implementação do Projecto	34
4.4.1	Integração do desenvolvedor com o sistema <i>OPUS-College</i>	34

4.4.2 Desenvolvimento	35
4.4.3 Testes ao módulo	54
4.4.4 Operação do módulo.....	55
4.4.5 Manutenção do módulo	56
4.5 Estimativa de custos	57
CAPÍTULO 5 - CONCLUSÕES E RECOMENDACÕES	59
5.1 Conclusões.....	59
5.2 Recomendações	59
REFERÊNCIAS BIBLIOGRÁFICAS	61
BIBLIOGRAFIA	63
ANEXOS	64
ANEXO I.....	65
ANEXO II	67

AGRADECIMENTOS

Aos meus pais José Wingester e Khajambibi Issufo, pela educação, carinho, amor e exemplo de dedicação, persistência para o alcance dos objectivos desejados.

Aos meus irmãos, sobrinhos pelo companheirismo e amizade.

Ao meu supervisor Eng. Micas Rafael pelo incentivo, pela confiança, compreensão e acima de tudo pelas críticas.

Aos meus colegas de escola e professores no geral.

A todos que contribuíram directa ou indirectamente para a realização deste trabalho.

Desenvolvimento de um módulo para gestão de contas de utilizadores

DEDICATÓRIA

Aos meus pais, família e amigos.

DECLARAÇÃO DE HONRA

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

Assinatura: _____

ÍNDICE DE TABELAS

Tabela 1 - atributos obrigatórios para a constituição de uma entrada no OpenLDAP do ISUTC.	19
Tabela 2-Lista de universidades em fase de implementação ou com o sistema OPUS-College implementado	26
Tabela 3 - Cronograma de actividades	32
Tabela 4-Requisitos funcionais	35
Tabela 5- Requisitos não funcionais.....	36
Tabela 6 - Estimativa de requisitos para componentes.....	40
Tabela 7 - Resultado dos testes as funcionalidades	54
Tabela 8- Estimativa de custos	58

ÍNDICE DE FIGURAS

Figura 1 - Modelo de processo de software em espiral de Boehm.....	7
Figura 2 - Processo da engenharia de requisitos	8
Figura 3 – Simbología usada nos diagramas de caso de uso.	11
Figura 4 - Simbología usada nos diagramas de classe.	12
Figura 5 - Simbología para a criação de diagramas de sequência.	13
Figura 6 - Simbología para a criação de diagramas de actividades.....	14
Figura 7 - entrada, seus atributos e valores	16
Figura 8 - Directorio de informação do LDAP no ISUTC	18
Figura 9 - Diagrama de actividades para o processo de criação de contas de utilizadores em massa.	22
Figura 10 - Diagrama de actividades para o processo de actualização de contas de utilizadores em massa.	23
Figura 11 - Diagrama de actividades para o processo de remoção de contas de utilizadores em massa.	24
Figura 12- Diagrama de actividades para o processo de recuperação de contas de utilizadores modo individual.....	25
Figura 13 - Processos de negócio OPUS-College	27
Figura 14 - Estrutura da norma ISO/IEC 12207	30
Figura 15 - Diagrama de casos de uso	38
Figura 16 - Padrão MVC	39
Figura 17 - Diagrama de instalação e componentes	40
Figura 18- Diagrama de classes	41
Figura 19- Diagrama de actividades para criação de contas de utilizadores para estudantes a partir do módulo de gestão de contas de utilizadores.....	42
Figura 20-Diagrama de actividades para criação de contas de utilizadores para funcionários a partir do módulo de gestão de contas de utilizadores.....	43
Figura 21-Diagrama de actividades para o processo de actualização de contas de utilizadores a partir do módulo de gestão de contas de utilizadores.....	44
Figura 22 - Diagrama de actividade para remoção de conta sem eliminar registo no OPUS-College.....	45

Figura 23-Diagrama de actividade para remoção de conta a partir da eliminação do registro no OPUS-College	46
Figura 24-Diagrama de actividades para o processo de recuperação de senhas de utilizadores a partir do módulo de gestão de contas de utilizadores.....	47
Figura 25 -Diagrama de sequência para o caso de uso criação de conta de utilizador.....	48
Figura 26- Diagrama de sequência para actualização do ano académico.....	49
Figura 27 - Diagrama de sequência para actualização de informações do estudante.....	50
Figura 28- Diagrama de sequencia para remoção da conta de utilizador sem eliminar o registro no OPUS-College.....	51
Figura 29 - Diagrama de sequencia para remoção da conta de utilizador a partir da eliminação do registro no OPUS-College.....	52
Figura 30 - Diagrama de sequencia para recuperação de senhas	53
Figura 31 - Diagrama de actividade para gestão de pedidos de alteração	56
Figura 32- Diagrama de actividades para disponibilização da aplicação em ambiente de produção	57

LISTA DAS ABREVIATURAS UTILIZADAS

DIT	<i>Directory information Tree</i>
IDE	<i>Integrated Development Environment</i>
ISUTC	Instituto Superior de Transportes e Comunicações
LDAP	<i>Lightweight Directory Access Protocol</i>
LIMEAA	Laboratório informático de métodos de ensino e auto-avaliação
MVC	<i>Model View Controller</i>
OSGi	<i>Open Services Gateway Initiative</i>
OPUS	<i>OPen University Systems</i>
RFC	<i>Request For Comments</i>
SA	Secretaria Académica
SecInf	Sector de Informática
UML	<i>Unified Modeling Language</i>

RESUMO

Este relatório descreve os processos utilizados para o desenvolvimento de um módulo de gestão de contas de utilizadores integrado a aplicação de gestão académica *OPUS-College*.

Neste relatório é evidenciada a solução referenciada acima, pois pretende-se com a mesma melhorar os processos de trabalho existentes para a gestão de contas de utilizadores, reduzindo as actividades manuais existentes e garantindo deste modo a redução do tempo de execução e probabilidade de erros cometidos.

O projecto contou essencialmente com três fases, a primeira reside na identificação do problema e estabelecimento de objectivos, a segunda corresponde a revisão de conceitos relevantes para o alcance dos objectivos, a ultima refere-se a resolução do problema em questão, onde a ultima passou por efectuar a codificação e integração da primeira versão do módulo na aplicação.

PALAVRAS CHAVE:

Ldap, Gestão de contas, *OPUS-College*

CAPÍTULO 1 - INTRODUÇÃO

1.1 Introdução

Este capítulo apresenta uma ligeira descrição a partir da problemática, do actual cenário vivido no contexto deste trabalho, e posteriormente efectua-se a identificação do problema, a determinação dos objectivos e a descrição das metodologias utilizadas para a execução da investigação.

1.2 Justificação do tema

A elaboração deste projecto e os motivos que levam a solucionar o problema em questão, são os seguintes:

- Primeiro, o facto da gestão de contas de utilizadores no ISUTC consistir de processos de trabalho não optimizados, tendo como consequência introdução de erros na execução das tarefas de gestão de contas e tempo de execução elevado.
- Segundo, a implicação prática do mesmo, pois o melhoramento do processo de gestão de utilizadores no ISUTC implica a automatização de actividades como, a criação de contas de utilizadores e a possibilidade de todos os utilizadores da rede de computadores que eventualmente tenham perdido a sua senha, possam a qualquer momento obter uma senha nova sem entrar em contacto com o administrador da rede ou responsável pelo sector informático, o que possibilita maior privacidade e segurança aos utilizadores.

1.3 Desenho teórico

1.3.1 Problemática

O ISUTC possui uma rede de computadores em operação com o objectivo de partilhar os recursos e serviços de rede existentes. Alguns dos serviços exigem que o utilizador contenha uma conta associada a ele, como o serviço de correio electrónico, acesso a *intranet* e a *Internet*.

A gestão de utilizadores no ISUTC resume-se essencialmente em efectuar a criação de novas contas de utilizadores em grande escala e actualizar as contas já existentes, este processo actualmente é executado pelos funcionários do Sector de Informática (SecInf).

Actualmente o SecInf efectua este procedimento de forma semiautomática, onde o mesmo passa pelos seguintes pontos:

- O sector de informática recebe da secretaria académica (SA) ficheiros de tipo texto separados, correspondentes a cada curso, onde em cada um dos ficheiros encontram-se listados os nomes dos estudantes matriculados.
- Posteriormente executa-se um *script*¹ associando o mesmo aos ficheiros correspondentes e alterando a turma e curso directamente no código escrito no *script*.

O responsável por este processo deve efectuá-lo tantas vezes quantas forem o número de turmas, tornando este processo longo e susceptível a erros.

Aliado ao ponto referido acima, está o facto de o ISUTC não conter nenhum mecanismo automatizado de recuperação de senhas para utilizadores que tenham eventualmente perdido ou esquecido a sua, consequentemente todo o utilizador da rede ISUTC quando deseja recuperar a senha deve dirigir-se ao SecInf e expor a situação para que lhe possa ser atribuída uma nova senha.

Visto que a gestão de contas de utilizadores reside essencialmente no facto de efectuar a inserção e actualização de contas de utilizadores em massa, surge a necessidade de uso de um sistema capaz de automatizar por completo este aspecto, sendo o ponto referente a recuperação de senhas também desejável.

1.3.2 Problema de investigação

O presente trabalho visa melhorar os processos de gestão de contas de utilizadores no ISUTC, passando pela automatização dos mesmos. Resultando no seguinte problema, como desenvolver um módulo *web* para automatização dos processos de gestão de contas de utilizadores no ISUTC?

1.3.3 Objecto de investigação

O presente trabalho tem como objecto de investigação, os processos de gestão de contas de utilizadores no ISUTC.

1.3.4 Objectivo geral

Desenvolver um módulo *web* que possibilita a automatização dos processos de gestão de contas de utilizadores ISUTC, visando a melhoria e optimização na execução destes processos.

¹ *Script* é um conjunto de instruções para uma determinada acção que o aplicativo realizará.

1.3.5 Objectivos específicos

- Descrever os procedimentos actuais para a execução dos processos de gestão de contas no ISUTC.
- Identificar os requisitos funcionais e não funcionais.
- Efectuar a modelação do módulo.
- Desenvolver o módulo.
- Testes ao módulo.

1.3.6 Perguntas da investigação

- 1.Quais as limitações dos actuais processos de gestão de contas de utilizadores no ISUTC?
- 2.Como desenhar e desenvolver um módulo capaz de eliminar as limitações existentes?
- 3.Que melhorias verificar-se-ão com o uso do módulo desenvolvido?

1.4 Metodologia

1.4.1 Abordagem da investigação

Este trabalho como já se tem referenciado, visa desenvolver um módulo que possibilita a resolução do problema em questão no ISUTC, passando pela utilização e integração de tecnologias que possibilitam o alcance dos objectivos, onde para tal será necessário um contacto directo entre o pesquisador e o ambiente onde a problemática reside.

Posto isto, este trabalho terá como base de investigação uma abordagem qualitativa, de carácter descritivo, efectuando-se um estudo particular para o caso do ISUTC.

1.4.2 Desenho da investigação

Após a elaboração do desenho teórico, segue-se a fase da pesquisa bibliográfica, onde tem-se como principal objectivo efectuar a consulta e analisar de maneira concisa a informação existente.

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

De seguida procurar-se-á desenvolver o módulo capaz de atender a todos os requisitos funcionais, para tal, primeiro será necessário efectuar a identificação dos mesmos, onde serão efectuadas entrevistas aos membros do SecInf.

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

1.4.3 Ideia a defender

O desenvolvimento de um módulo que possibilite a automatização dos processos de gestão de contas de utilizadores pode melhorar a gestão de contas de utilizadores no ISUTC.

1.4.4 Métodos de investigação

Como método de investigação será usada a investigação empírica.

O qual segundo Audy J. (2006) a pesquisa empírica “significa que o processo de pesquisa deve estar baseado em fenómenos observáveis na realidade”.

Posto isto, este trabalho passará por efectuar entrevistas formalizadas e estruturadas aos membros do sector de informática, os quais vivem a problemática descrita.

Passa também pelo processo de observação do estado actual e da actual forma de execução dos processos de gestão de contas de utilizadores.

Estes pontos possibilitarão a obtenção dos requisitos funcionais do sistema.

No que diz respeito a modelação do sistema, serão utilizadas varias técnicas da *Unified Modeling Language* (UML).

Referente ao desenvolvimento do módulo, a mesma será concebida utilizando o paradigma de programação orientada a objectos tendo em conta os padrões de desenho de *software*, onde o desenvolvedor procura manter-se dentro da norma ISO/IEC 12207 especificamente nos processos fundamentais de desenvolvimento.

1.4.5 Resultados esperados de investigação

Espera-se ao fim desta investigação, a obtenção de um módulo capaz de cumprir com os requisitos funcionais e que consequentemente seja capaz de resolver o problema em questão.

1.5 Resumo

Neste capítulo foram apresentados os aspectos relevantes para a escolha do tema, onde verificou-se que os mesmos residem essencialmente na optimização dos processos de gestão de contas de utilizadores, onde este ultimo aspecto constitui a problemática em questão.

Foi apresentado a metodologia de investigação, onde a abordagem de investigação será qualitativa de carácter descritivo.

CAPÍTULO 2 - MARCO TEÓRICO-CONCEITUAL DA INVESTIGAÇÃO

2.1 Introdução

Esta secção fornece fundamentos teóricos para que se possa entender melhor os procedimentos seguidos na elaboração do projecto, serão apresentados aspectos gerais para facilitar a compreensão sobre o assunto abordado, fundamentos resultantes de uma revisão bibliográfica sobre os principais conceitos.

Desenvolver aplicações ou seja *software* sem antes ter real conhecimento da sua possível constituição é completamente difícil, Pressman R. (1995, pag. 12) define *software* como “instruções que quando executadas, produzem a função e o desempenho desejados.”. Uma análise mais ampla do conceito de *software* é dado por Sommerville I. (2003, pag. 5) afirmando que “software não é apenas o programa mas também toda a documentação associada e os dados de configuração necessários para fazer com que esses programas operem correctamente”.

Este capítulo visa descrever o processo de *software*, dando conhecimento das principais actividades dentro do mesmo, encontra-se especificado o modelo de processos de *software*, concretamente o em espiral, aborda-se também o processo da engenharia de requisitos, a técnica de modelação UML e a essência do protocolo LDAP.

2.2 Processo de *software*

Para que seja possível desenvolver *software* de qualidade, é necessário adoptar um modelo de processo de *software* que irá servir de guia para o desenvolvimento do projecto, destacando-se nestes modelos actividades que fazem parte do processo de *software*, sendo assim faz se necessário conhecer este processo.

Sommerville I. (2003, pag. 7) considera processo de *software* como “um conjunto de actividades e resultados associados que geram um produto de *software*.”. Em geral os profissionais de *software* adoptam uma abordagem sequencial e organizada das actividades no processo de desenvolvimento, uma vez que essa é a maneira mais eficaz de produzir software. Vários autores descrevem as actividades as quais acham fundamentais para o desenvolvimento de *software*, entretanto e segundo Sommerville I. (2003, pag. 7) “há quatro actividades de processo fundamentais comuns a todos os processos de *software*”.

1. Especificação do *software*
2. Desenvolvimento do *software*

3. Validação do *software*

4. Evolução do *software*

Todo o processo deve conter uma abstracção útil capaz de explicar diferentes abordagens do desenvolvimento de *software*, essas abstracções são chamadas de modelos de processo de *software*. (SOUZA O., 2005)

2.3 Modelos de processo de *software*

Segundo Jorge A. e Rafael P. (2008, pag. 9) “... clientes e fornecedores desejam que o *software* produzido atenda a determinados critérios, tais como realizar aquilo para o qual foi solicitado, ser entregue na data prevista”, entretanto muitos projectos de *software* falham em atender a esses critérios.

Jorge A. e Rafael P. (2008, pag. 10) afirmam que “a abordagem informal de desenvolvimento muitas vezes não é suficiente para evitar essas falhas”.

Existem vários modelos de processo de *software* tais como:

- Modelo em cascata
- Prototipagem
- Modelo V
- Modelo em espiral
- Desenvolvimento rápido de aplicações

Segundo Souza O. (2005) “O processo de desenvolvimento deve ser incremental para atender a natureza evolutiva das aplicações *Web*.”. Esta afirmação dada baseia-se no facto de que em aplicações *web* “não é possível especificar totalmente no início do desenvolvimento o que uma aplicação *web* irá conter, porque sua estrutura e funcionalidade evoluem com o tempo”. (Souza O. 2005)

Todos os modelos referenciados acima contêm as suas vantagens e desvantagens, entretanto para este projecto far-se-á uso do modelo em espiral.

2.4 Modelo Espiral

O modelo espiral para a engenharia de *software* foi desenvolvido para abranger as melhores características tanto do ciclo de vida clássico (cascata) como o de prototipagem, acrescentando ao mesmo um novo elemento, a análise de riscos. (Pressman R.1995)

O modelo espiral não representa o processo de *software* como uma sequência de actividades com algum retorno de uma actividade para outra, o processo é representado como uma espiral, onde segundo Sommerville I. (2003, pag. 45) “cada *loop* na espiral representa uma

Desenvolvimento de um módulo para gestão de contas de utilizadores

fase do processo de *software*. Assim, o *loop* mais interno pode estar relacionado a viabilidade do sistema, o *loop* seguinte, a definição de requisitos do sistema, o próximo *loop*, ao projecto do sistema, e assim por diante”.

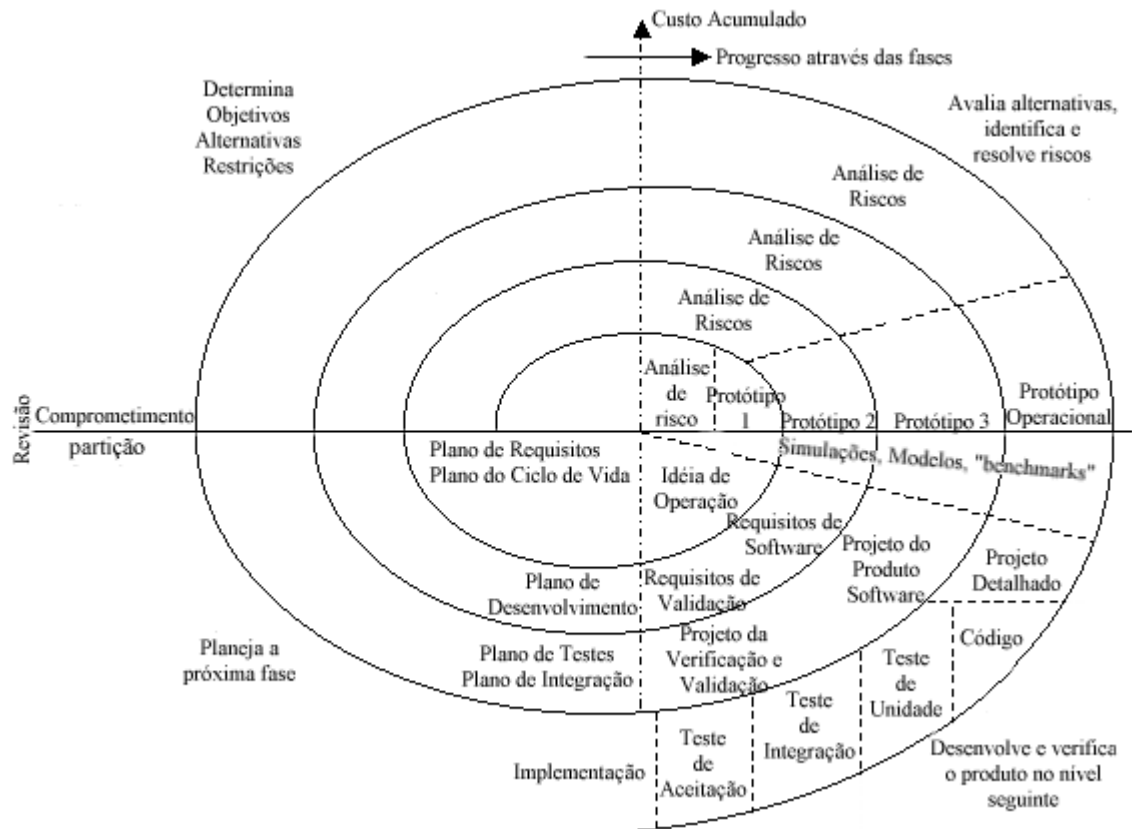


Figura 1 - Modelo de processo de software em espiral de Boehm [Sommerville I., 2003]

Cada volta da espiral é dividida em quatro sectores, descritas por Pressman R. (1995) da seguinte maneira:

- **Definição de objectivos** - São identificadas as restrições para o processo e o produto, os riscos do projecto e, dependendo dos riscos, poderão ser planeadas estratégias alternativas.
- **Avaliação e redução de riscos** – Para cada um dos riscos de projecto identificados, é realizada uma análise detalhada e são tomadas providencias para reduzir os mesmos.
- **Desenvolvimento e validação** – Depois da avaliação dos riscos é escolhido um modelo auxiliar para o desenvolvimento, se forem predominantes os riscos relacionados a *interface* gráfica com o utilizador, o modelo apropriado é a prototipagem, caso o risco maior seja o de integração de sistemas, o modelo apropriado é o modelo em cascata.

- **Planeamento** – O projecto é revisto e é tomada uma decisão sobre continuar com o próximo *loop* da espiral, se a decisão for continuar devem ser traçados os passos para a próxima fase.

A grande diferença do modelo em espiral em relação aos outros é a importância que este introduz para a análise de riscos, de maneira informal a análise de riscos é algo de errado que pode acontecer no projecto, como a possibilidade de exceder o prazo.

2.5 Engenharia de requisitos

Um *software* capaz de responder a grande maioria das necessidades dos utilizadores ou do cliente tem grande probabilidade de ser um produto de *software* de sucesso.

Desenvolver *software* sem levar em consideração aspectos como, “o que se pretende fazer”, “quais os aspectos prioritários para o cliente”, torna fictício o processo de concepção.

Segundo Boehm (1981) apud KOURI (2007) “Erros em requisitos detectados apenas depois do *software* implementado são até 20 vezes mais caros de se corrigir que qualquer outro tipo de erro, e até 200 vezes mais custosos do que seriam se corrigidos durante a fase de análise do sistema”.

A fim de evitar estes problemas a engenharia de requisitos introduz um processo com a finalidade de auxiliar as actividades envolvidas na identificação, documentação e evolução dos requisitos. (SOMMERVILLE,1998 apud LIMA & JUNIOR, 2009)

Posto isto faz-se necessário conhecer os processos introduzidos pela engenharia de requisitos.

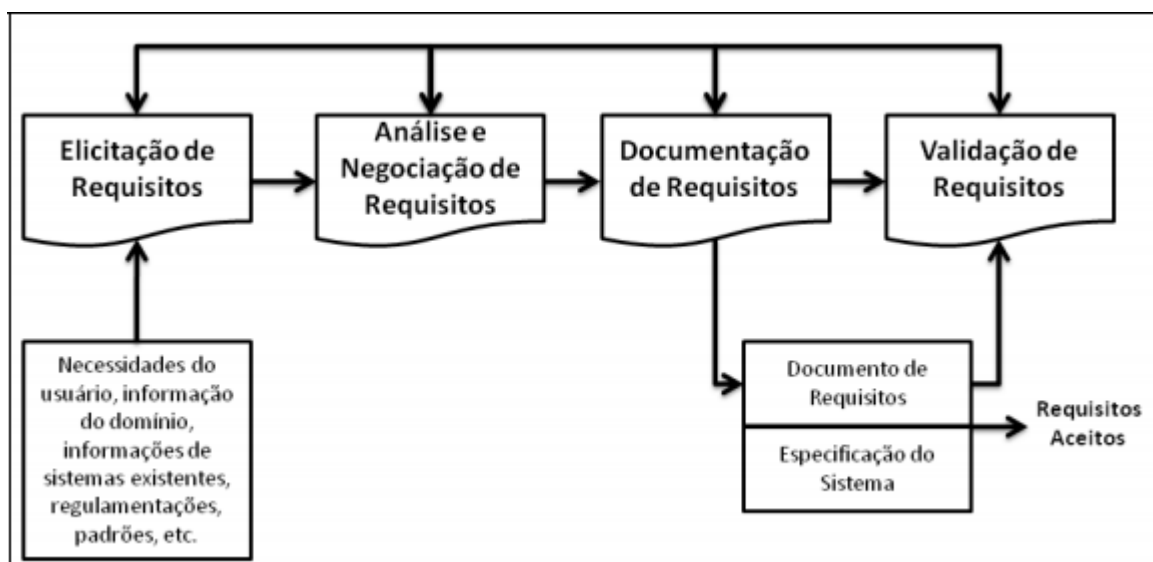


Figura 2 - Processo da engenharia de requisitos [LIMA & JUNIOR, 2009]

A aplicação destes processos é feita na fase inicial do desenvolvimento da aplicação.

É importante também ter conhecimento dos vários tipos de requisitos existentes.

2.5.1 Tipos de requisitos

Segundo Sommerville I. (2007) pode-se dividir os requisitos da seguinte forma:

- **Requisitos de utilizador** – estes devem ser descritos em uma linguagem compreensível pelo utilizador não entrando em detalhes técnicos no que diz respeito ao desenvolvimento da aplicação.
- **Requisitos de sistema** – estes são mais abrangentes e mais detalhados em comparação com os requisitos de utilizador, estes devem definir o que o sistema deve fazer e não como ele deve ser implementado.

Tem-se dentro destes a definição dos requisitos não funcionais e requisitos funcionais.

Requisitos Funcionais

“São declarações de funções que o sistema deve fornecer, como o sistema deve reagir a entradas específicas e como deve se comportar em determinadas situações. Em alguns casos, os requisitos funcionais podem também explicitamente declarar o que o sistema não deve fazer”. (Sommerville I., 2003)

Requisitos não funcionais

“São restrições sobre os serviços ou as funções oferecidos pelo sistema. Entre eles destacam-se restrições de tempo, restrições sobre o processo de desenvolvimento, padrões, entre outros”. (Sommerville I., 2003)

Qualquer processo de desenvolvimento de *software* começa por efectuar o levantamento de requisitos.

De seguida são apresentados alguns aspectos importantes para que esta actividade seja cumprida da melhor maneira.

2.5.2 Levantamento ou eliciação de requisitos

O levantamento de requisito tem como objectivo tentar perceber quais as reais necessidades e restrições do sistema a desenvolver.

Existem varias técnicas que podem ser utilizadas para atingir o objectivo referido, são elas as seguintes:

- Análise da documentação
- Entrevistas
- Questionários
- Observação
- Volumes

- Cenários
- Prototipagem

“É importante também não esquecer que, sempre que possível, se deve utilizar o conhecimento do domínio para guiar a fase de levantamento de requisitos”. (LOPES F. et al, 2005), entende-se por domínio uma área de negócio específica, como por exemplo educação, construção.

Neste projecto, apenas far-se-á uso de uma técnica de entrevistas.

2.6 Arquitectura de *software*

Uma das definições mais clássicas da arquitectura de *software* define o que é o sistema em termos de componentes e o relacionamento entre os componentes. (GARLAN & SHAW,1994), semelhante a mesma, Bass et al. (2003) refere-se a arquitectura de *software* como estruturas que incluem componentes, suas propriedades externas e os relacionamentos entre eles, constituindo uma abstracção do sistema.

Uma boa arquitectura será aquela capaz de corresponder a todos os requisitos funcionais e não funcionais do sistema, isto só será possível se existir um entendimento claro do real ponto de operação da aplicação, para que a aplicação possa responder a todas as necessidades dos utilizadores ou do utente a quem destina-se, é necessário realizar-se da melhor maneira os processos da engenharia de requisitos aliados a arquitectura de *software* para que posteriormente possa ser efectuada a modelação da aplicação, recorrendo a técnicas de modelação UML.

2.7 Técnicas de modelação UML

Os *Software* da actualidade tendem a ser cada vez mais flexíveis, entretanto não estão preparadas para satisfazer todas as necessidades dos potenciais utilizadores.

Posto isto, torna-se necessário recorrer-se a uma linguagem de modelação que facilite a comunicação entre os desenvolvedores e os potenciais utilizadores, pois estes últimos sempre definem as reais necessidades.

Segundo O'Neill e Nunes (2004) a *Unified Modelling Language* (UML) “abre perspectivas para responder ao desafio de desenvolvimento de novos sistemas de informação, cada vez mais complexos, robustos, fiáveis e ajustados as necessidades dos utilizadores.”

O'Neill e Nunes (2004) definem UML como uma “linguagem que utiliza uma notação padrão para especificar, construir, visualizar e documentar sistemas de informação orientados por objectos.”

Pela simplicidade dos conceitos utilizados na UML, esta facilita o desenvolvimento de sistemas de informação, pois a mesma é capaz de integrar os aspectos de natureza organizacional e de natureza tecnológica. Pelo facto de utilizar um conjunto de símbolos padronizados, a UML funciona como um meio de comunicação entre os diversos elementos envolvidos no processo de desenvolvimento, utilizadores, gestores e a própria equipa de desenvolvimento.

2.7.1 Diagrama de Casos de uso

“O diagrama de caso de uso representa, segundo a perspectiva do utilizador, o que o sistema deve efectuar. Poder-se-á dizer que tem como objectivo representar os requisitos funcionais do sistema, recorrendo para tal aos conceitos de caso de uso e de actor”. (LOPES F. et al, 2005)

Estes diagramas utilizam as seguintes abstracções de modelação:

- **Actor** – é o utilizador que interage com o sistema, pode ser uma pessoa ou outro sistema.
- **Caso de uso** – qualquer sequência de acções que os utilizadores realizam no sistema.
- **Relações (*include*, *extend* e *generalização*)** – a relação *include* significa que um determinado caso de uso utiliza a funcionalidade existente em outro caso de uso. A relação *extend* usado quando existe um comportamento opcional que deve ser incluído num caso de uso. A generalização é usada quando existe um caso de uso que é um caso particular de outro.

A representação gráfica segue-se abaixo:

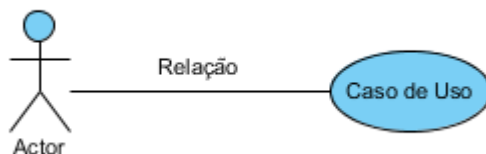


Figura 3 – Simbologia usada nos diagramas de caso de uso. [adaptado de O'NEILL & NUNES,2004]

2.7.2 Diagrama de classes

Segundo Flower M. (2005) um diagrama de classes “descreve os tipos de objectos presentes no sistema e os vários tipos de relacionamentos estáticos existentes entre eles.”

O'Neill e Nunes (2004) descrevem o conceito de classe da seguinte maneira, “a criação de um modelo de classes resulta de um processo de abstracção através do qual se identificam os

objectos (entidades e conceitos) relevantes no contexto que se pretende modelar e se procura descrever características comuns em termos de propriedades (atributos) e de comportamento (operações). A essa descrição genérica designa-se por classe.”

Um diagrama de classes é composto pelos seguintes elementos:

- **Classes** – é a representação de um conjunto de objectos que partilham os mesmos atributos e comportamentos. (Lopes F. et al , 2005)
- **Objectos** – um objecto é uma entidade ou conceito existente no contexto de modelação (mundo real). Que é relevante incorporar no modelo de informação. (O’NEILL & NUNES, 2004)
- **Relações de associação e generalização** – No diagrama de classe as relações representam a ligação entre classes.

“A generalização é um caso especial no diagrama de classes, que demonstra a noção de superclasse na perspectiva de uma relação ‘pai e filho’ “. (O’NEILL & NUNES, 2004)

- **Multiplicidade** – indica quantos objectos participam na relação.

Segue-se a simbologia para a criação dos diagramas de classe.

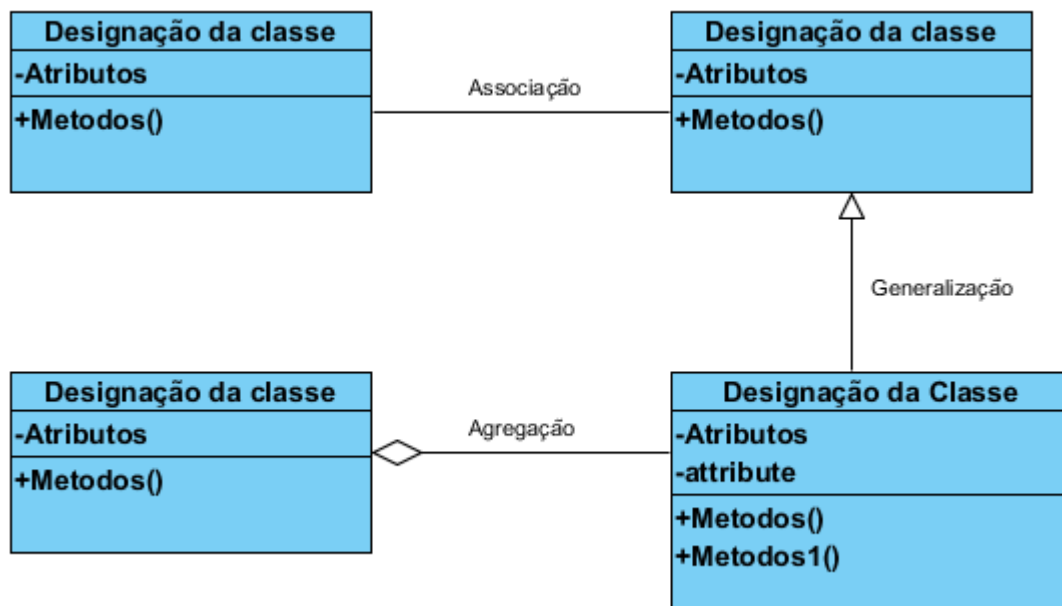


Figura 4 - Simbologia usada nos diagramas de classe. [adaptado de O’NEILL & NUNES,2004]

2.7.3 Diagrama de sequência

Segundo O’Neill e Nunes (2004) “diagrama de sequência é um diagrama de interacção que realça a ordem das mensagens”.

São três os objectos de modelação contemplados por este diagrama:

- Actor.
- Classes.
- Mensagens.

“As mensagens trocadas entre objectos representam a invocação de um serviço (operação) disponibilizado por um objecto, com o objectivo de despoletar uma acção ou actividade”. (O’NEILL & NUNES, 2004)

O actor é representado com o mesmo símbolo dos diagramas de caso de uso, as classes são representadas com um rectângulo no topo do diagrama com uma linha a tracejado por baixo, chamada linha de vida do objecto.

Segue-se a simbologia para a criação dos diagramas de sequência.

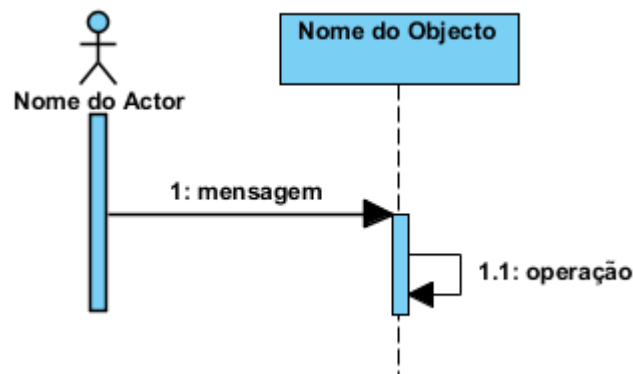


Figura 5 - Simbologia para a criação de diagramas de sequência.[adaptado de O’NEILL & NUNES,2004]

2.7.4 Diagrama de Actividades

O diagrama de actividades é um diagrama UML utilizado para modelar o aspecto comportamental dos processos.

“O Diagrama de actividades constitui um elemento de modelação simples, mas eficaz para descrever fluxos de trabalho numa organização ou para detalhar operações de uma classe, incluindo comportamentos que possuam processamento paralelo.” (O’NEILL & NUNES, 2004)

Um diagrama de actividades pode ser constituído pelos seguintes componentes:

- Linhas verticais de responsabilidade.
- Actividades.
- Símbolos de comportamento condicional.
- Processamento paralelo

Linhas verticais de responsabilidade

Através da utilização das linhas verticais de responsabilidade, é possível descrever quais são os objectos responsáveis.

Actividades

Em um diagrama de actividades é necessário identificar a actividade inicial, esta actividade é definida para identificar o início do diagrama, e é representada por um círculo preenchido a preto.

Existe também a actividade operacional, é descrita graficamente por um rectângulo de lados arredondados com um identificador. Uma actividade permite descrever uma acção ou conjunto de acções que são realizadas quando a actividade se inicia, durante o seu decurso normal, e quando termina. (O'NEILL & NUNES, 2004)

Para identificar uma actividade final de um fluxo de trabalho utiliza-se um círculo a preto, limitado com uma circunferência.

Comportamento condicional

Em um fluxo de actividades podem existir caminhos alternativos, onde faz se uso de um nó de decisão ou diamantes de decisão, tem uma entrada e duas saídas.

Segue-se a simbologia básica dos diagramas de actividades.

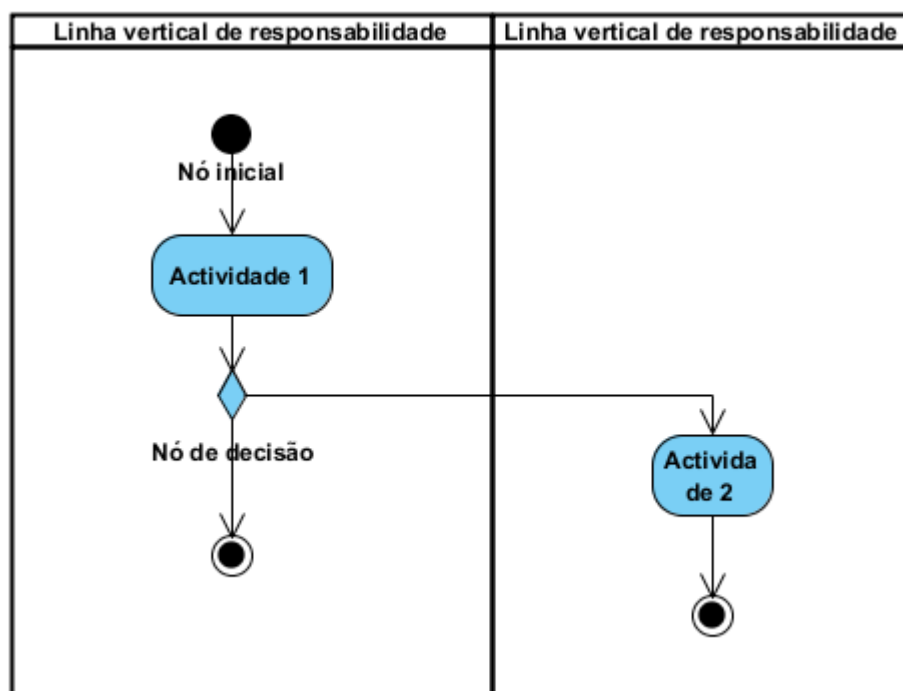


Figura 6 - Simbologia para a criação de diagramas de actividades.[adaptado de O'NEILL & NUNES,2004]

2.8 Protocolo LDAP

Para falarmos sobre o protocolo LDAP (*Lightweight Directory Access Protocol*) precisamos antes ter conhecimento do conceito de directório, pode-se encontrar vários exemplos de directórios, tais como, páginas amarelas, catálogos de endereços, lista telefónica e outros, estes directórios caracterizam-se por disponibilizar uma grande quantidade de informação para que possa ser consultada. Deste modo, toda a informação que é constantemente consultada e não permanentemente modificada, pode ser armazenada num directório.

Trigo C. e Carrara L. (2010) referem-se a um directório como “um serviço de armazenamento hierárquico de informações com o objectivo principal de facilitar a pesquisa e a recuperação destas informações.”

LDAP foi definido na sua primeira versão no *Request for comments*² (RFC) 1487 da especificação da *internet Engineering Task Force*³ (IETF) e na sua terceira e actual versão denominada LDAPv3 no RFC 4510.

LDAP é um protocolo utilizado para actualizar e pesquisar directórios, define a linguagem de consulta as informações, mas também é o protocolo de rede que transporta as requisições e respostas do cliente para o servidor. (BUTCHER M., 2007)

Segundo Howes T. et al (1997) o LDAP foi originalmente desenvolvido como um cliente para o padrão X.500, este define o protocolo de acesso a directório (DAP), para que os clientes pudessem usar quando estivessem em contacto com servidores que disponibilizem serviços de directórios.

Segundo Trigo C. e Carrara L.(2010) “o X.500 foi baseado no modelo de referencia OSI, e que dentre outros problemas, era bastante difícil de ser implementado correctamente, resultando em aplicações complexas e lentas.”

O LDAP surge como uma alternativa ao protocolo DAP para o acesso aos serviços de directórios baseados no padrão X.500.

“O LDAP é baseado no modelo cliente-servidor. Um ou mais servidores LDAP contem os dados criando a árvore de Directório LDAP. Um cliente LDAP conecta-se a um servidor, geralmente pela porta padrão 389, e faz uma requisição. O servidor responde ou exhibe um ponteiro para onde o cliente pode conseguir a informação (tipicamente, outro servidor LDAP).” (CHAVES,2010)

² RFC é um documento que descreve os padrões de cada protocolo da *internet*.

³ IETF é uma comunidade internacional ampla e aberta preocupada com a evolução da arquitectura da *internet*.

O LDAP define quatro modelos de dados, descritos de seguida.

2.8.1 Modelos de dados

Modelo de informação – Fornece a estrutura das informações armazenadas no directório LDAP. “O modelo de informação do LDAP está centrado na ideia de entrada. Uma entrada é um conjunto de atributos que é inserido no directório, geralmente para representar algum objecto do mundo real.” (ISQUIERDO G., 2001)

Cada atributo contém um tipo de dado e um ou mais valores.

Toda a entrada deve ter um atributo *objectClass*, onde está definido o conjunto de atributos obrigatórios, que devem estar presentes.

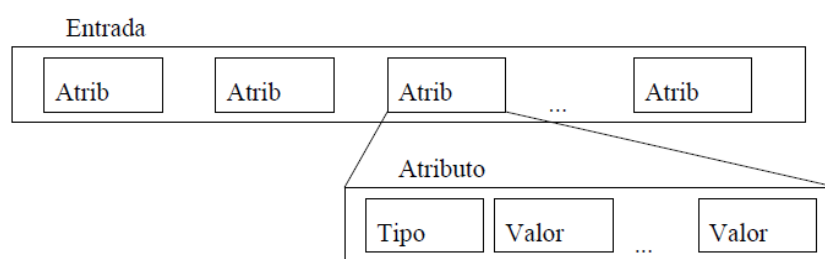


Figura 7 - entrada, seus atributos e valores [ISQUIERDO G.,2001]

Modelo de nomes - Descreve a organização das entradas no directório e como eles são identificados, as entradas estão dispostas de forma hierárquica, num formato de árvore. Segundo Ughini C. (2004) “ árvore é denominada DIT (*Directory information Tree*). As entradas estão dispostas na DIT de acordo com seu *distinguished name* (DN), que serve como identificador único. Um DN é uma sequência de *relative distinguished names* (RDN) que juntos apontam para uma única entrada. ”

Alguns dos atributos definidos em LDAP são:

- **c** (*countryName*) – Guarda o nome do país.
- **cn** (*commonName*) – Guarda nome de objectos.
- **dc** (*DomainCompany*) – Guarda o nome de dominio do DNS.
- **o** (*organizationName*) – Guarda nome de uma organização.
- **ou** (*organizationalUnitName*) – Guarda nome de uma unidade na organização.

Modelo Funcional - Define quais operações podem ser realizadas nas informações armazenadas no directório LDAP.

As operações básicas são:

- **Add** – Adiciona uma nova entrada.
- **Modify** – Modifica uma entrada.

- **Delete** – Apaga uma entrada.
- **Compare** – Testa se uma entrada tem determinado valor como atributo.

Modelos de segurança – Descreve como as informações no directório LDAP podem ser protegidas de acessos não autorizados.

2.8.2 Comparação entre LDAP e uma base de dados relacional

Chaves T. (2010) refere-se a esta comparação da seguinte maneira:

- No LDAP as informações são armazenadas em um directório, ou seja, em uma estrutura em árvore, optimizada para operações de pesquisa e leitura, e com baixo desempenho para operações de escrita e actualização.
- Nos bancos de dados relacionais as informações são armazenadas em tabelas, ideais para grandes volumes de dados e execução de transacções complexa.
- No LDAP não é possível relacionar directamente dois ou mais atributos, pois é uma base de dados estruturada hierarquicamente, ao contrário de uma base de dados relacional.

2.9 Resumo

Neste capítulo foram abordados os conceitos que apoiam o alcance dos objectivos propostos, descreveu-se o conjunto de actividades que possibilitam gerar um produto de *software*.

Verificou-se que a partir de uma arquitectura de *software* bem definida e aliada a técnicas de obtenção e percepção de requisitos, e posterior modelagem a partir de técnicas de modelação UML, tem-se um grande caminho percorrido para obter um produto de *software* que corresponde aos requisitos funcionais e não funcionais desejados pelo cliente.

Abordou-se também o protocolo LDAP, dando-se ênfase aos modelos de dados que este contém e efectuou-se uma comparação entre o LDAP e uma base de dados relacional.

CAPÍTULO 3 - MARCO CONTEXTUAL DA INVESTIGAÇÃO

3.1 Introdução

O resultado do desenvolvimento de *software* é a entrega de um produto que satisfaça os requisitos definidos anteriormente, e que devem satisfazer ao utilizador final, produzindo o resultado esperado pelo mesmo.

Posto isto, é fundamental que o desenvolvedor conheça o contexto a qual será implementado o *software*, tendo total domínio dos processos existentes, conhecendo os passos fundamentais para execução dos mesmos e também as suas limitações, para possibilitar o melhoramento. Neste capítulo é feita a descrição do actual modo de execução dos processos de gestão de contas de utilizadores, resultante de entrevistas feitas aos funcionários do SecInf do ISUTC, procurando sempre identificar as limitações existentes, começa-se por conhecer a implementação do LDAP no ISUTC, o OpenLDAP.

3.2 OpenLDAP

Referir que todos os utilizadores da rede de computadores do ISUTC actualmente encontram-se registados em um servidor de base de dados o qual esta instalado o serviço de directórios LDAP, concretamente o OpenLDAP.

OpenLDAP é um *software* livre de código aberto que implementa o protocolo LDAP, o projecto de desenvolvimento do OpenLDAP teve início em 1998 por Kurt Zeilenga e posteriormente Howard Chu juntou-se ao projecto. (BUTCHER M., 2007)

De todas as funcionalidades que o OpenLDAP possui, a que é de principal destaque no contexto do ISUTC é a capacidade de oferecer a autenticação aos utilizadores, fazendo uso da sua base de dados. Faz-se uso da centralização, pois a autenticação de todos os serviços de rede se concentram em uma única árvore de informações.

De seguida é demonstrada a estrutura do directório de informação do LDAP no ISUTC.

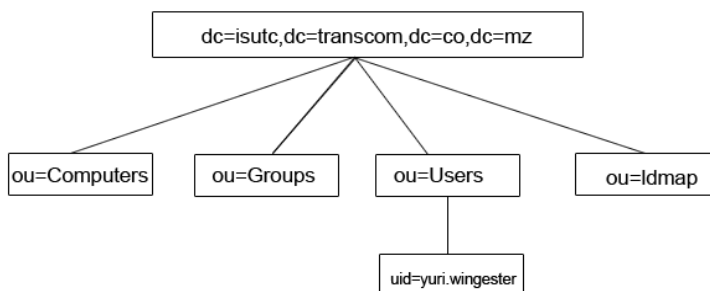


Figura 8 - Directório de informação do LDAP no ISUTC

Desenvolvimento de um módulo para gestão de contas de utilizadores

O dn para entrada representada acima, seria:

uid=yuri.wingester, ou=Users, dc=isutc,dc=transcom,dc=co,dc=mz.

No ISUTC uma entrada de utilizador no LDAP é constituída pelos seguintes atributos:

Tabela 1 - atributos obrigatórios para a constituição de uma entrada no OpenLDAP do ISUTC.

Atributo	Descrição
uid	Id do utilizador, corresponde a primeiro.utilimoNome no ISUTC
Cn	CommonName, correspondente ao no do individuo
Sn	Surname, corresponde ao último nome do individuo
User name	Nome do utilizador
userPassword	Senha do utilizador
departmentNumber	Corresponde a turma do estudante utilizador
displayName	Nome visualizado.
Email	Email do utilizador
gidNumber	Id do grupo que o utilizador esta incluso
sambaSID	Valor que possibilita verificar se o utilizador pode autenticar em um certo dominio.

3.3 Conta de utilizador no ISUTC

É importante antes de mais conhecer o que realmente significa uma conta de utilizador no contexto do ISUTC.

Uma conta de utilizador no ISUTC segundo os funcionários entrevistados é uma identificação atribuída ao utilizador, e que consiste numa entrada no LDAP, esta conta tem a função de apoiar aos serviços existentes no ISUTC, como o serviço de correio electrónico, o serviço de ftp, o acesso as estações de trabalho e o serviço de apoio a aprendizagem Moodle.

3.4 Gestão de contas de utilizadores no ISUTC

A gestão de contas de utilizadores no ISUTC significa segundo o Eng. Micas Rafael, manipular contas existentes no domínio, que possibilitam aceder as estações de trabalho e os serviços implementados, como o acesso a *Internet*, serviço de ftp e etc.

Segundo os funcionários do SecInf entrevistados, a gestão de utilizadores reside essencialmente na execução dos seguintes processos:

- Criação de contas de utilizadores.
- Actualização de contas de utilizadores.
- Remoção de contas de utilizadores.
- Redefinir as senhas de utilizadores sempre que necessário.

A redefinição de senhas após um eventual esquecimento, torna-se um aspecto importante devido ao número considerável de utilizadores da rede ISUTC que dirigem-se ao SecInf apresentando este desejo.

3.5 Processos para a gestão de contas de utilizadores

Os processos existentes no ISUTC para efectuar a gestão de contas de utilizadores seguem dois modos de operação, que são os seguintes:

Modo em massa - Execução do processo para um número elevado de utilizadores.

Modo individual - Execução do processo para utilizadores de maneira individual.

De Seguida são descritos os processos utilizados para efectuar a gestão de contas de utilizadores no ISUTC nestes dois modos de operação.

3.5.1 Criação de Contas de utilizadores

O processo de criação de contas de utilizadores começa no acto da matrícula do aluno, os funcionários da SA são responsáveis por efectuar a matrícula dos alunos.

Após efectuado este processo, e após o início das aulas, a SA é responsável por elaborar vários ficheiros em quantidade proporcional ao número de turmas criadas, este ficheiro deve conter o nome completo de cada estudante, número de estudante e a turma associada.

De seguida este ficheiro é enviado ao SecInf, onde é dado continuidade ao processo de criação de contas.

Desenvolvimento de um módulo para gestão de contas de utilizadores

O funcionário do SecInf quando executa este processo em modo em massa, começa por manipular todos os nomes existentes nos ficheiros, sendo esta manipulação o aspecto que mais tempo consome na execução deste processo.

A manipulação efectuada pelo funcionário do SecInf resume-se essencialmente nos seguintes aspectos referidos de seguida:

- Remoção dos caracteres especiais existentes nos nomes.

Este processo é concluído com a execução de um *script* escrito em *shell scripting*⁴, o funcionário do SecInf deve efectuar os seguintes passos para concluir este processo:

- Editar o *script* associando este a turma que está a ser inserida.
- Executar o *script*.
- Guardar em um ficheiro o resultado da execução do *script*, que consiste no nome de utilizador e a sua respectiva senha, isto para cada estudante.

Importante referir que este processo é repetitivo, vezes igual ao número de turmas existentes.

Em modo individual o funcionário do SecInf deve aceder a aplicação *php ldap admin* e a partir da mesma introduzir uma entrada no LDAP com os atributos necessários e com a informação correcta do utilizador.

O processo de criação em massa pode ser definido pelo diagrama abaixo.

⁴ *Shell Script* é uma linguagem de desenvolvimento de *script*.

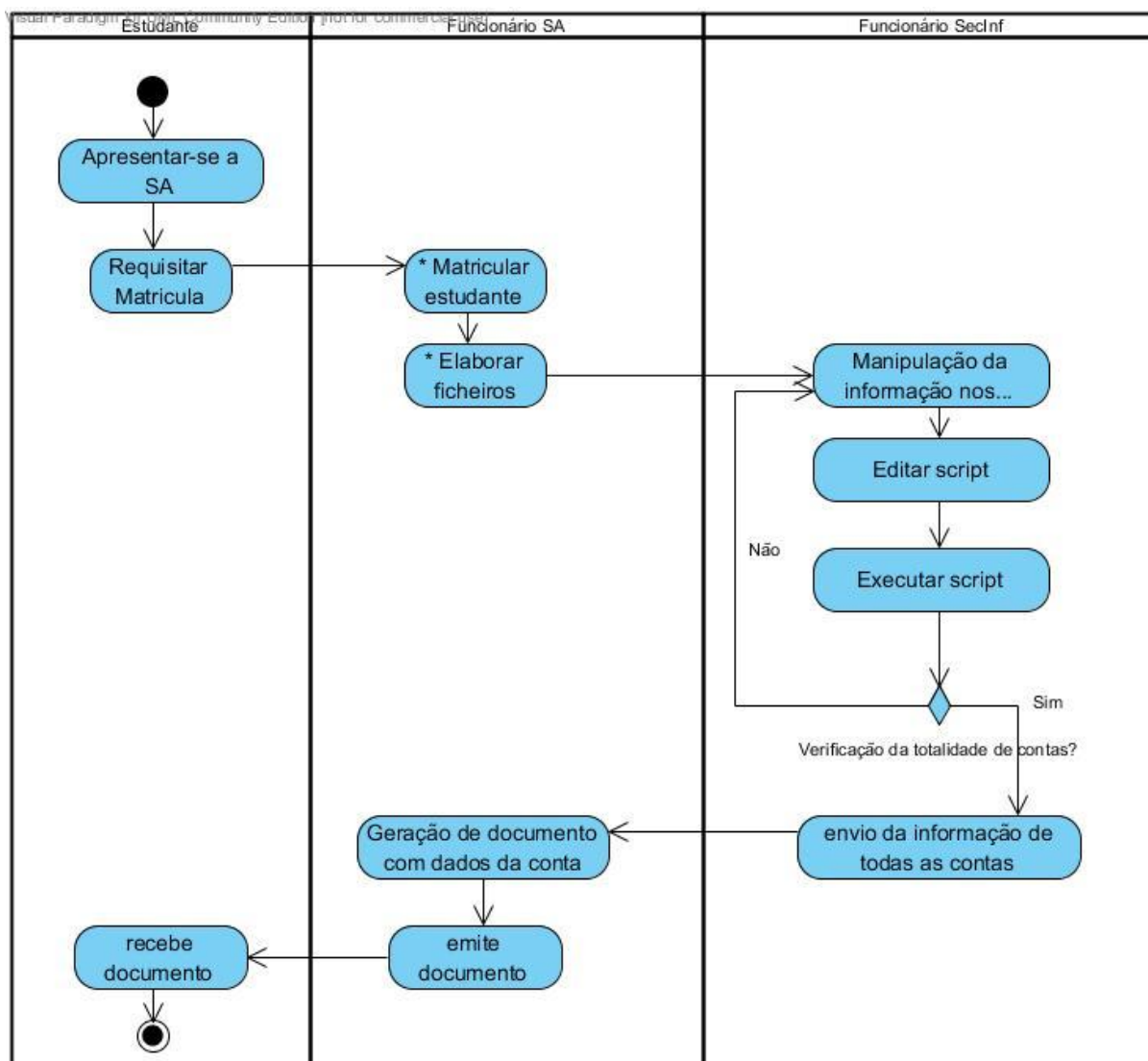


Figura 9 - Diagrama de actividades para o processo de criação de contas de utilizadores em massa.

3.5.2 Actualização de contas de utilizadores

No início de cada ano existe a necessidade de transitar todos os estudantes do ano académico em que estavam inseridos para o ano seguinte, caso tenham aprovado. Este processo tem início também na SA, onde os funcionários são responsáveis por elaborar ficheiros, analogamente para o processo de criação de contas de utilizadores, contendo a informação necessária.

Após a recepção destes ficheiros por parte dos funcionários do SecInf, estes devem novamente manipular a informação analogamente a criação de contas de utilizadores, e posteriormente executar um *script* responsável por efectuar a actualização da turma do estudante, efectuando a alteração do valor existente no atributo *departmantNumber* da entrada no LDAP.

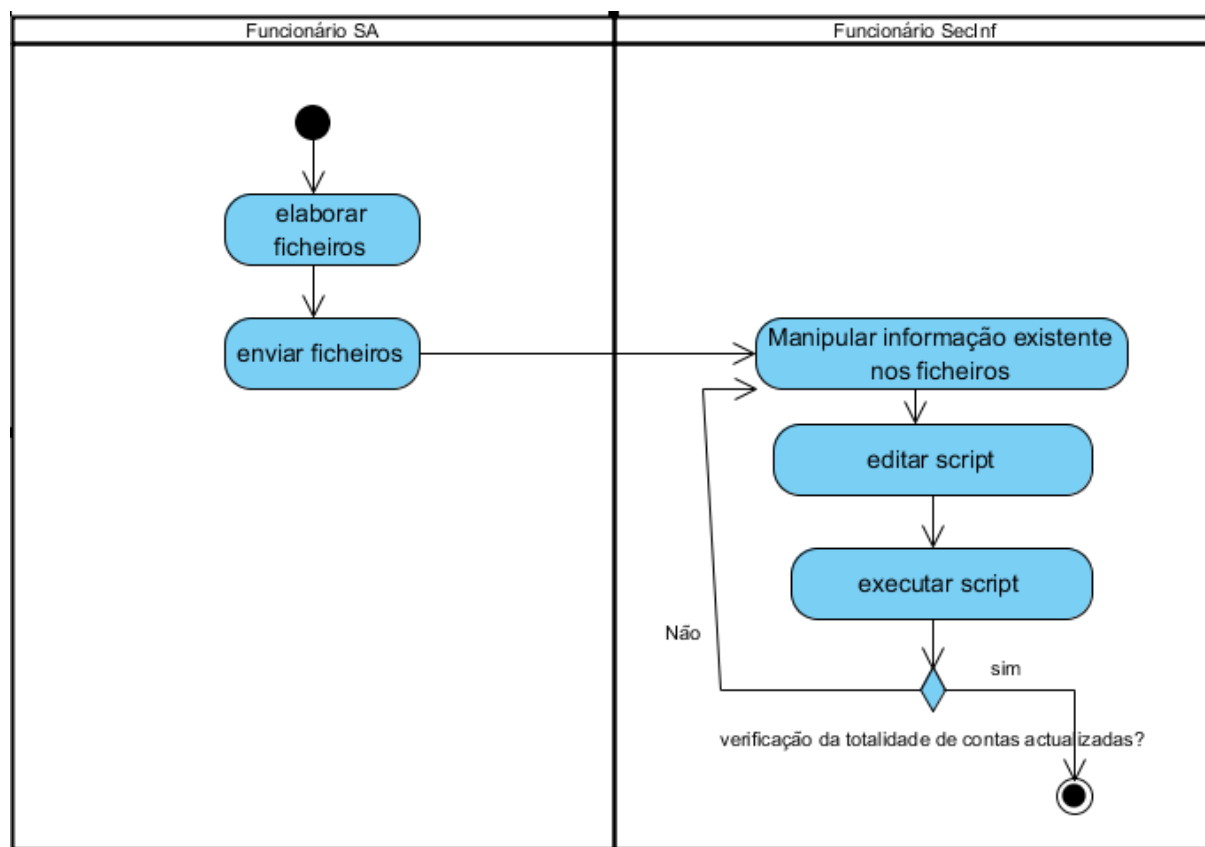


Figura 10 - Diagrama de actividades para o processo de actualização de contas de utilizadores em massa.

Sempre que é cometido um erro nos dados do utilizador relativamente a conta, o funcionario do SecInf deve efectuar a alteração dirigindo-se a aplicação *php ldap admin* e a partir da mesma alterar o necessario.

3.5.3 Remoção de contas de utilizadores

Devido as políticas de segurança, ao facto do ISUTC conter uma capacidade de armazenamento de informação baixa para o número de utilizadores que possui e o nível de utilização dos seus serviços, os gestores da rede vêm-se obrigados a efectuar a remoção da conta dos estudantes e profissionais que já não fazem parte da instituição.

Este processo é feito segundo os funcionários do SecInf seguindo um padrão de tempo, onde o funcionário tem apenas uma semana para retirar toda a sua informação e o estudante tem direito a usar a conta por mais um ano, após este período a conta é completamente eliminada.

A SA é responsável por enviar um ficheiro tipo texto contendo os nomes e as turmas dos estudantes e um outro ficheiro contendo os nomes dos funcionários os quais deve-se eliminar a conta.

Desenvolvimento de um módulo para gestão de contas de utilizadores

Neste processo o funcionário executa um script de eliminação associado aos ficheiros que contêm os nomes dos utilizadores a eliminar.

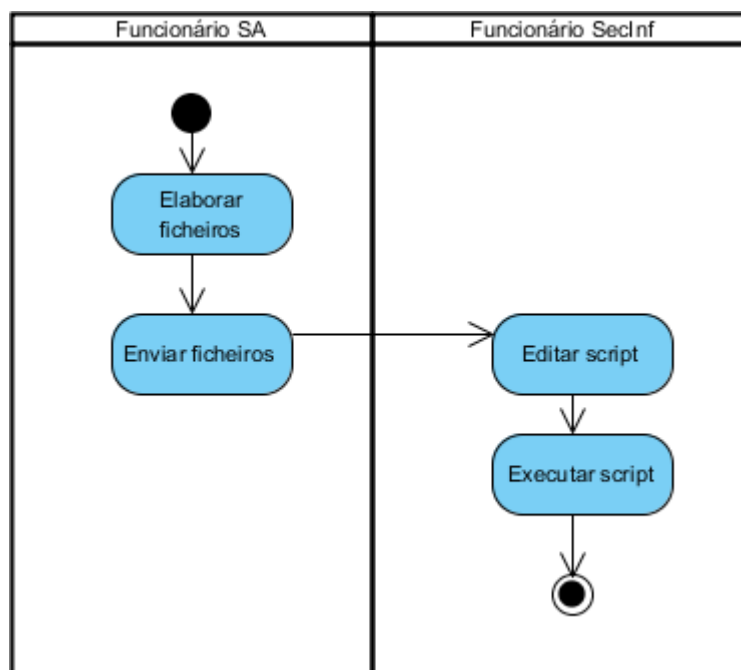


Figura 11 - Diagrama de actividades para o processo de remoção de contas de utilizadores em massa.

3.5.4 Recuperação de senhas

Este processo apenas é executado após eventual esquecimento da actual senha do utilizador.

Neste actual processo o utilizador deve dirigir-se a secretaria académica expondo a sua actual situação, o funcionário da SA deve exigir as informações básicas do utilizador e a partir do seu actual sistema de gestão escolar denominado Platão identifica-lo e imprimir os dados da conta do utilizador, concretamente o nome do utilizador e a primeira senha atribuída a essa conta.

Posteriormente o utilizador deve dirigir-se ao SecInf, e requisitar que a senha da sua conta seja alterada para a primeira atribuída a mesma imprimida pela SA.

De seguida é obrigação do utilizador efectuar a troca da senha.

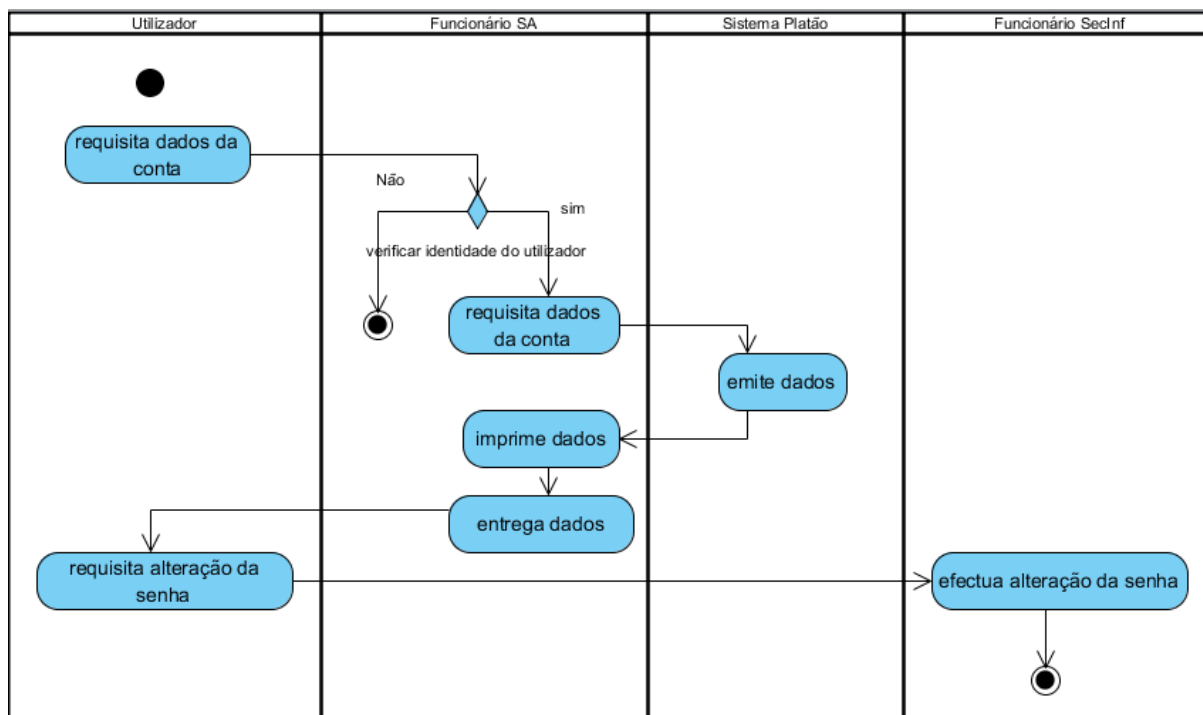


Figura 12- Diagrama de actividades para o processo de recuperação de contas de utilizadores modo individual.

3.6 Avaliação dos processos actuais

São descritos de seguida as actuais limitações do modo de execução destes processos:

- Primeiramente, todos os processos executados em massa são efectuados executando um *script* que deve ser alterado sempre que seja necessário utiliza-lo.
- Segundo, no processo de criação de contas de utilizadores, há necessidade de manipular os nomes existentes nos ficheiros, resulta em uma alta probabilidade de existência de erros nos mesmos, e exige do funcionário uma capacidade de concentração extremamente elevada.
- Esta manipulação torna estes processos muito longos, devido a grande quantidade de utilizadores.
- Existe um tempo de espera para o estudante obter as credenciais da sua conta.
- No processo de recuperação de senhas, o facto do utilizador ter que entrar em contacto com vários intervenientes para recuperar a sua senha, poem em causa a segurança da informação existente nos vários serviços, como o serviço de email.

3.7 OPUS-College

OPUS-College (OPen University Systems) é um sistema de informação baseado em *web* para o registo e consulta de informações sobre estudantes, planos de estudo, membros da equipa

envolvida no processo de formação académica e informação da organização académica no geral. (SIMONS ED, 2011)

O sistema *OPUS-College* abrange uma ampla gama de funcionalidades sobre questões académicas, desde a administração até funções de registo e actualização dos processos dos estudantes, o ISUTC encontra-se neste momento em fase de implementação deste sistema, onde pretende-se até ao fim do ano de 2013 ter-se o sistema em produção, para efectuar a substituição do actual sistema de gestão escolar, o Platão.

Este sistema é de código aberto, e traz consigo todas as vantagens que um sistema com código aberto proporciona, é desenvolvido com recurso a JEE⁵, faz uso do sistema de gestão de base de dados PostgreSQL, contem uma arquitectura modular orientada a serviços, é um sistema 100 % baseado em web, é facilmente extensível devido a arquitectura modular do sistema, as funcionalidades do *OPUS-College* podem ser facilmente estendidas com a integração dos diferentes módulos, segundo Simons Ed.(2011) este sistema é desenvolvido por vários especialistas internacionais (Holanda, Áustria, Moçambique) e todos os módulos desenvolvidos pelos participantes são colocados disponíveis gratuitamente para toda a comunidade.

OPUS-College foi inicialmente desenvolvido como parte de um projecto de desenvolvimento para as universidades Moçambicanas, projecto este fundado pela NUFFIC, organização não-governamental holandesa de cooperação para o desenvolvimento da universidade.

Segundo Simons Ed.(2011) este sistema hoje é implementado ou esta em fase de implementação em várias universidades Moçambicanas e também da Zâmbia, de seguida são listadas as universidades que fazem uso deste sistema.

Tabela 2-Lista de universidades em fase de implementação ou com o sistema *OPUS-College* implementado

Universidade	Localização
Universidade Eduardo Mondlane (UEM)	Moçambique
Universidade católica de Moçambique (UCM)	Moçambique
Universidade Pedagógica (UP)	Moçambique
Universidade Mussa Bin Bique (UMBB)	Moçambique

⁵ JEE é uma plataforma de programação na linguagem de programação Java.

Desenvolvimento de um módulo para gestão de contas de utilizadores

Instituto Superior de Transportes e Comunicações (ISUTC)	Moçambique
University of Zambia (UNZA)	Zâmbia
Copperbelt University (CBU)	Zâmbia

Dependendo do contexto a qual o sistema está a ser implementado, existe sempre a necessidade de adequar o sistema aos processos de trabalho existentes, entretanto ao desenvolver-se o *OPUS-College* foram definidos processos de negócio que o sistema devia cobrir.

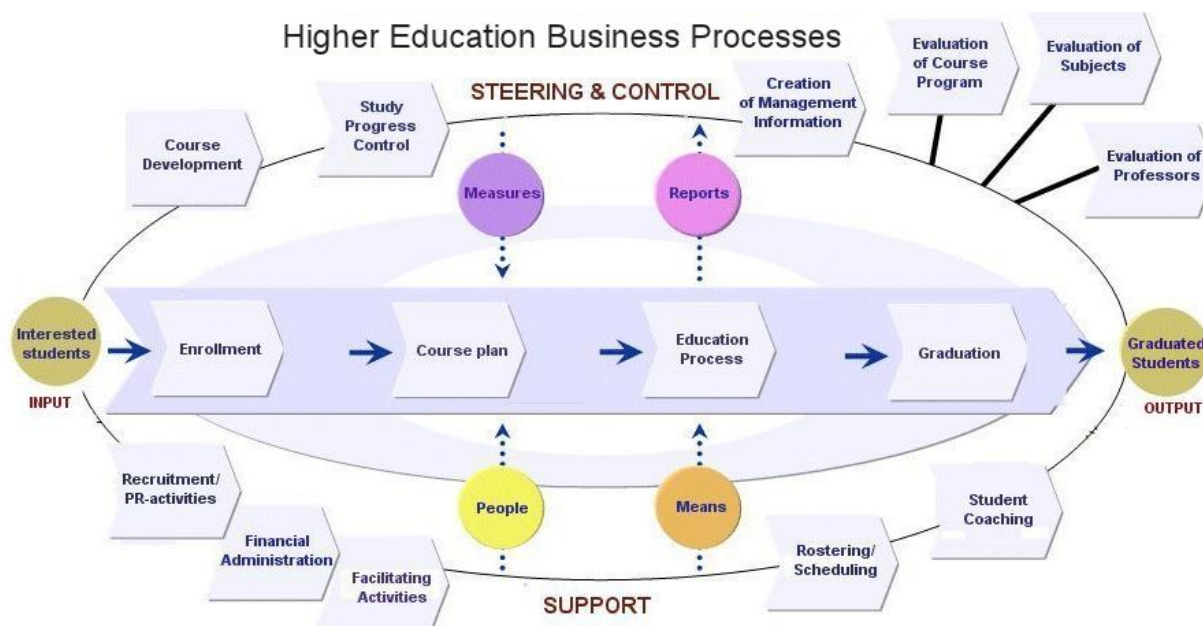


Figura 13 - Processos de negócio *OPUS-College* [SIMONS ED., 2011]

Segundo SIMONS ED. (2011), a primeira versão do *OPUS-college*, lançada em agosto de 2011 era composta pelos seguintes módulos:

- Um *core module* denominado *College*, comum a todas instituições que implementam o *OPUS-College*.
- Módulos opcionais – Que podem ser especificados e adaptados as necessidades e situações de um país ou mesmo instituição individual, actualmente os seguintes módulos adicionais estão disponíveis e em permanente melhoramento.
 - *Online Registration* – Módulo que permite aos estudantes interessados em ingressar em certa instituição académica, efectuarem o registo pela Internet.

Desenvolvimento de um módulo para gestão de contas de utilizadores

- *Reports Module* – Módulo que contem todas as funções de saída do opus-college.
- *Scholarship Module* – Registo de informações sobre bolsas de estudo dos estudantes.
- *Fees Module* – Registo de informações sobre cobranças.
- *Accommodation Module* – Registo de informações sobre o alojamento dos estudantes.
- *Accommodation Fees Module* - Registo de informações financeiras sobre o alojamento dos estudantes.

Todos estes módulos adicionais podem ser modificados para adaptação ao contexto em que esta a ser implementado o *OPUS-College*.

3.8 Resumo

Neste capítulo efectuou-se a contextualização do problema, descreveu-se a implementação do protocolo LDAP no ISUTC, o OpenLdap, demonstrou-se a estrutura do directório LDAP implementada, e detalhou-se os atributos que constituem uma entrada no LDAP.

Verificou-se que uma conta de utilizador no ISUTC é uma identificação atribuída ao utilizador da rede, onde esta conta tem a função de auxiliar a todos outros os serviços de rede existentes.

Descreveu-se os processos que fazem parte da gestão de contas de utilizadores no ISUTC, identificou-se as suas principais limitações, e descreveu-se o sistema que encontra-se em fase de implementação no ISUTC, o *OPUS-College*.

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

4.1 Introdução

Este capítulo apresenta os processos seguidos para o desenvolvimento do módulo, desde a obtenção dos requisitos funcionais até a implementação dos mesmos, mostrando posteriormente o resultado dos testes executados.

Após ter-se verificado as limitações existentes no actual modo de execução dos processos de trabalho para a gestão de contas de utilizadores no ISUTC, toda a modelação e desenvolvimento do módulo será no sentido de eliminar no máximo as limitações actuais.

4.2 Porque desenvolver um módulo para a gestão de contas de utilizadores e não uma aplicação isolada?

O grande motivo que impulsiona o desenvolvimento de um módulo para o sistema *OPUS-College* e não de uma aplicação, é a possibilidade de efectuar a centralização das funções de gestão de contas de utilizadores no sistema de gestão de registo académico.

Devido ao facto da aplicação *OPUS-College* conter uma arquitectura modular orientada a serviços torna-se possível desenvolver um módulo para a gestão de contas de utilizadores, evitando deste modo o desenvolvimento de uma aplicação, obtendo assim vantagens que são resultantes da integração de sistemas, como:

- **Comunicação** – Efectuar a dinamização da comunicação entre departamentos, concretamente no ISUTC, a SA e o SecInf.
- **Redução de tempo, esforços e erros na execução de tarefas** – Efectuando a integração do módulo com o sistema *OPUS-College* pode permitir um melhor desempenho do SecInf, e evita a duplicação de esforços na inserção de informação, concretamente na criação de contas de utilizadores.

4.3 Metodologia de desenvolvimento

Para o desenvolvimento deste projecto foi seguida uma abordagem evolutiva, optando-se pelo modelo de processo de desenvolvimento em espiral, anteriormente descrito neste trabalho, procurou-se seguir as macro actividades definidas por este modelo aliadas a certos processos fundamentais e de apoio definidos na norma ISO/IEC 12207.

4.3.1 Norma ISO/IEC 12207

Esta norma estabelece uma estrutura comum para os processos de ciclo de vida de *software*, com terminologia bem definida que pode ser referenciada pela indústria de *software*, contém

Desenvolvimento de um módulo para gestão de contas de utilizadores

processos, actividades e tarefas que podem ser aplicadas durante a aquisição de um produto ou serviço de software, e durante o desenvolvimento, operação e manutenção de produtos de software. (BERNARDI, 2003)

Esta norma agrupa as actividades que podem ser executadas durante o ciclo de vida de *software* em cinco processos fundamentais, oito processos de apoio e quatro processos organizacionais. (BERNARDI, 2003)

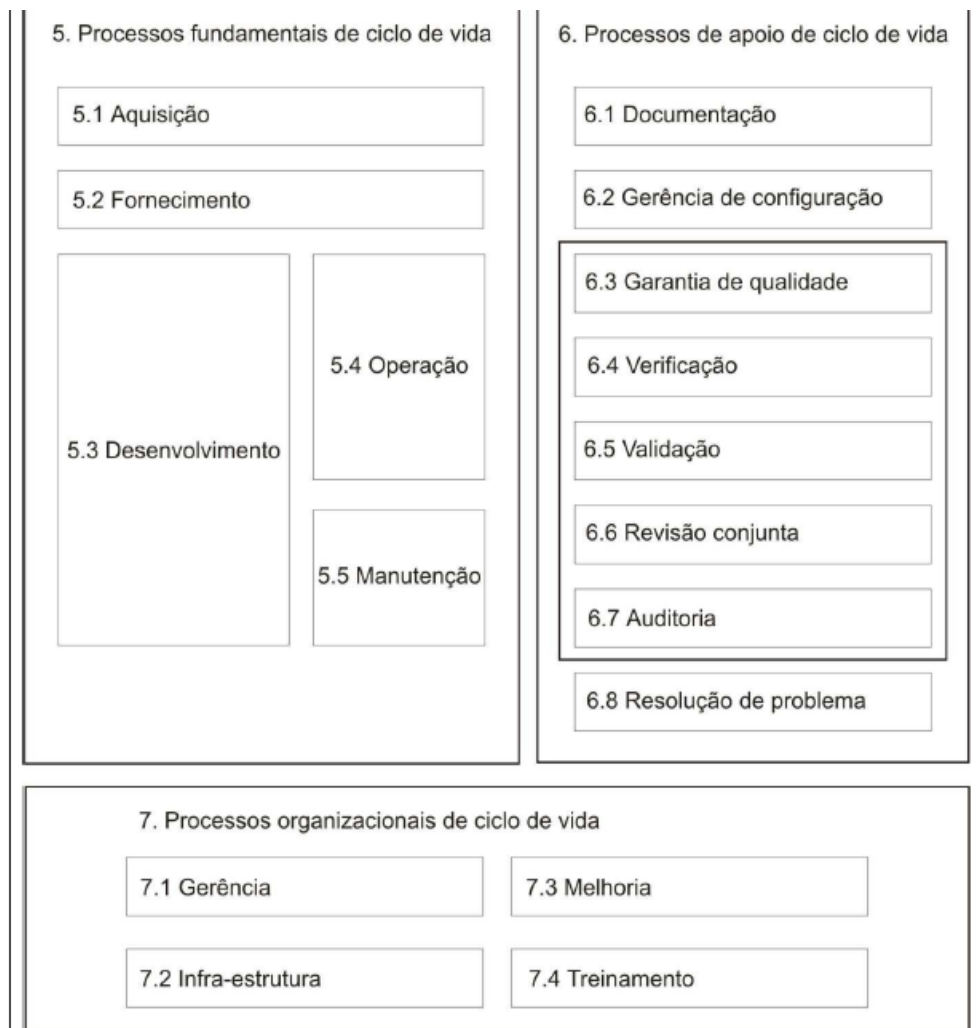


Figura 14 - Estrutura da norma ISO/IEC 12207 [Bernardi, 2003]

Esta norma é ainda formada por uma sessão de anexos, onde estão inclusos os seguintes:

- Processo de adaptação.
- Processo para adaptação.
- Orientações sobre Processos e organizações.
- Bibliografia.

4.3.2 Fases do projecto

Fase 1 – Integração do desenvolvedor com o sistema *OPUS-College*

Nesta primeira fase, e uma vez que o projecto tem o objectivo de desenvolver um módulo integrado a aplicação *OPUS-College* para gestão de contas de utilizadores, corresponde sobretudo a uma fase de aprendizagem das tecnologias, da forma como estão implementados o modelo de dados e o código existente na aplicação *OPUS-College*.

Fase 2 – Desenvolvimento do módulo

Esta fase é um processo fundamental do ciclo de vida, e consiste essencialmente das seguintes actividades:

- Especificação e validação de requisitos.
- Definição de tecnologias a usar.
- Definição de ferramentas de suporte.
- Análise de riscos.
- Modelação de casos de uso.
- Desenho da arquitectura do *software*.
- Desenho dos diagramas de classe, actividade e de sequência.
- Codificação de casos de uso.

Fase 3 – Testes as funcionalidades

Esta fase passa pela execução das seguintes actividades:

- Testes as funcionalidades.
- Escrita do relatório de testes.

Fase 4 - Operação do módulo

Esta fase corresponde na definição de actividades que devem ser seguidas de maneira periódica para garantir a operação do *software*.

Fase 5 – Manutenção do módulo

Esta fase consiste da seguinte actividade:

- Definição de um processo de manutenção.

4.3.3 Cronograma de desenvolvimento

Tabela 3 - Cronograma de actividades

Fase	Actividade	Maio				Junho				Julho			
		Semanas				Semanas				Semanas			
		1	2	3	4	1	2	3	4	1	2	3	4
1	Procura e Leitura da documentação do Opus-College												
1	Instalação do Opus-College em ambiente de desenvolvimento												
1	Alterações básicas e simulação de ambiente de desenvolvimento												
2	Especificação e validação de requisitos												
2	Definição de tecnologias a usar												
2	Definição de ferramentas de suporte												
2	Análise de riscos												
2	Modelação de casos de uso												
2	Desenho da arquitectura de												

Desenvolvimento de um módulo para gestão de contas de utilizadores

	<i>software</i>												
2	Desenho de diagramas de classe, actividade e de sequência												
2	Codificação de casos de uso críticos												
2	Criação da primeira versão da aplicação												
2	Disponibilização da aplicação em ambiente de testes												
3	Testes aos casos de uso críticos												
2	Desenvolvimento dos casos de uso menos críticos												
3	Testes aos casos de uso menos críticos												
3	Elaboração do relatório de testes												
5	Resolução dos problemas encontrados												
3	Elaboração do relatório de testes												
2	Elaboração do Manual de												

	instalação												
2	Elaboração do Manual de uso												

4.4. Implementação do Projecto

4.4.1 Integração do desenvolvedor com o sistema *OPUS-College*

Esta fase consistiu na execução das seguintes actividades:

1. Procura e leitura da documentação existente para Aplicação *OPUS-College*.

A documentação referente ao *OPUS-College* foi obtida na página oficial do projecto *OPUS-College*, documentação referente aos seguintes pontos:

- Apresentação da aplicação.
- Manual de utilizador.
- Manual de instalação do ambiente de desenvolvimento.
- Manual de instalação do ambiente de produção.
- Diagramas de caso de uso.
- Arquitectura da aplicação.
- Modelo de dados.

2. Montagem de ambiente de desenvolvimento para a aplicação *OPUS-College*.

Esta fase consistiu na instalação dos seguintes aplicativos:

- *Java Development Kit 6*.
- *IDE Eclipse Galileo*.
- Sistema de gestão de base de dados - PostgreSQL 9.0
- Gestor de Base de dados - *pgAdmin 1.12*
- Servidor de aplicações - *Apache Tomcat 6.0.3*

E posterior importação para o IDE dos projectos que constituem a versão 4.1 do *OPUS-College*, de seguida efectuou-se a eliminação dos problemas relacionados a importação de bibliotecas e dependências entre módulos.

3. Alterações básicas na aplicação e posterior simulação do ambiente de produção.

Esta fase resumiu-se em efectuar pequenas alterações nas telas do *OPUS-College* e instalar a aplicação alterada num ambiente de produção simulado, isto para garantir que o

desenvolvedor estava capacitado para introduzir alterações mais robustas, como a introdução de um módulo novo.

4.4.2 Desenvolvimento

Esta constitui a fase crítica para a implementação do projecto, passando pelas actividades descritas abaixo.

1. Especificação e validação de requisitos.

O levantamento de requisitos foi resultado da entrevista feita ao funcionarios do SecInf (Anexo I). De seguida seguiu-se com a fase de negociação, posteriormente efectuou-se a documentação e validação.

O documento final da especificação de requisitos pode ser visto no Anexo II.

O resumo desta especificação pode ser vista nas tabelas abaixo.

Requisitos funcionais

Tabela 4-Requisitos funcionais

Requisito		Descrição	Prioridade
RF01	Criação de conta de utilizador	Após efectuar-se a matrícula de um novo estudante o módulo integrado na aplicação <i>OPUS-College</i> deve criar automaticamente uma conta de utilizador no ISUTC, criando uma entrada no directório LDAP.	Máxima
RF02	Actualização de contas de utilizador	O módulo integrado na aplicação <i>OPUS-College</i> deve possibilitar a actualização de contas de utilizadores a partir da alteração dos dados do <i>OPUS-College</i> .	Máxima
RF03	Eliminação de conta de utilizador	O módulo integrado na aplicação <i>OPUS-College</i> deve possibilitar a eliminação de contas de utilizadores.	Mínima
RF04	Recuperação de senhas de utilizadores	A aplicação deve conter um mecanismo que possibilite a	Máxima

		recuperação da senha do utilizador que eventualmente tenha perdido a senha actual, não obrigando o utilizador a entrar em contacto com o administrador de rede.	
--	--	---	--

Requisitos Não funcionais

Tabela 5- Requisitos não funcionais

Requisito	Descrição
RNF01- Autenticação centralizada	A aplicação <i>OPUS-College</i> deve permitir ao estudante efectuar a autenticação com a conta anteriormente criada, autenticando-se com a conta do directório LDAP.
RNF02- Níveis de autorização	A aplicação <i>OPUS-College</i> deve carregar o nível de autorização dos utilizadores segundo os grupos definidos no LDAP.

2. Definição de Tecnologias e plataformas usadas

O produto de *software* aqui proposto é uma extensão a aplicação *OPUS-College*, onde a mesma foi desenvolvida em cima das seguintes tecnologias e plataformas:

- Sistema de gestão de base de dados
 - *PostgreSQL*
- Biblioteca de Persistência
 - *Ibatis*
- Linguagem de programação
 - *Java*
- Tecnologias de desenvolvimento
 - *JavaServer Pages – jsp*
 - *Java Platform, Enterprise Edition - JEE*
 - *Spring*

Todo o módulo será desenvolvido fazendo uso das mesmas tecnologias utilizadas no desenvolvimento do sistema *OPUS-College*, isto porque as mesmas garantem o alcance dos requisitos funcionais aqui descrito e proporciona ao desenvolvedor menor nível de dificuldade para efectuar a integração do módulo com o resto da aplicação.

3. Definição das ferramentas de suporte

Foram utilizadas as seguintes ferramentas de suporte a modelação, desenvolvimento e administração:

- *Visual Paradigm for UML 10.1 Community edition* - Ferramenta de modelação, com suporte a modelação UML, BPMN e outros.
- *Eclipse Galileo* - Utilizado como IDE para codificação dos casos de uso.
- *pgAdmin 1.12* – Ferramenta utilizada para efectuar a administração e desenvolvimento da base de dados PostgreSQL.

4. Análise de riscos

Documentação desactualizada – A documentação disponível do sistema *OPUS-College* não corresponde em vários aspectos a realidade do sistema, o que traduz-se em um aumento da probabilidade de erros introduzidos na aplicação pelo desenvolvedor e consequente aumento no tempo de execução do projecto.

5. Diagrama de casos de uso

O resumo dos diagramas de casos de uso pode ser vista na imagem abaixo, entretanto no documento de especificação de requisitos em anexo, contem uma descrição detalhada do fluxo de execução de cada caso de uso.

Desenvolvimento de um módulo para gestão de contas de utilizadores

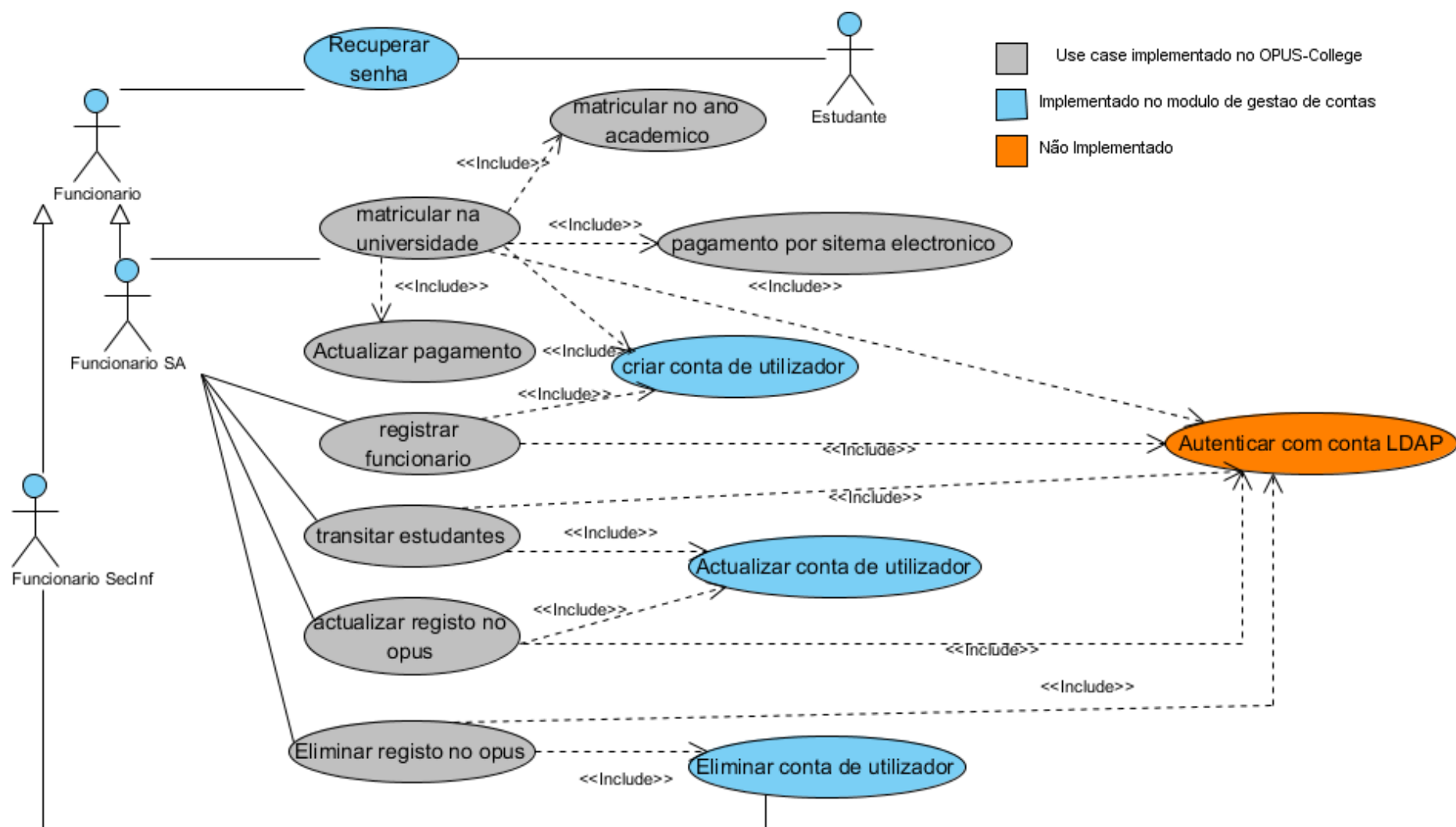


Figura 15 - Diagrama de casos de uso

6. Desenho da Arquitectura de *software*

Padrão de desenvolvimento usado

O modelo de desenvolvimento utilizado para a produção deste módulo é o MVC (*Model-View-Controller*).

MVC é actualmente considerado um padrão de projecto utilizado na engenharia de *software*. Este modelo determina a separação de uma aplicação em três camadas lógicas. A camada *Model* formada por entidades que representam os dados da aplicação. A camada *View* tem por objectivo realizar a apresentação destes dados e capturar os eventos do utilizador. A camada *Controller* faz a ligação entre o *Model* e a *View*, realiza o tratamento dos eventos, actua sobre a camada *Model* e altera a *View* para representar a nova forma dos dados. (Tiago Barros et al., 2001)

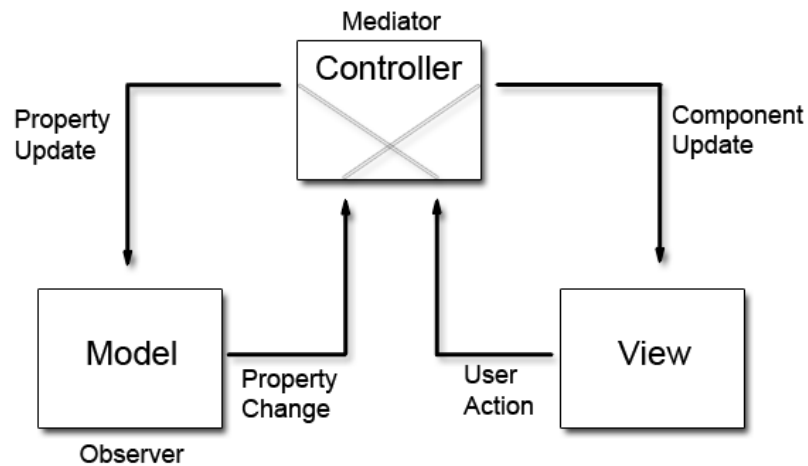


Figura 16 - Padrão MVC [Tiago Barros et al, 2001]

Diagrama de instalação e componentes

Faz-se necesario demonstrar a forma como os componentes físicos estão organizados e como os componentes de software estão distribuidos, estabelecendo a relação entre os componentes físicos e os lógicos.

A definição deste diagrama ajuda a efectuar uma manutenção mais eficiente e até mesmo reduzir ou eliminar por completo o impacto negativo da substituição de componentes antigos por novos.(O'NEILL & NUNES, 2004)

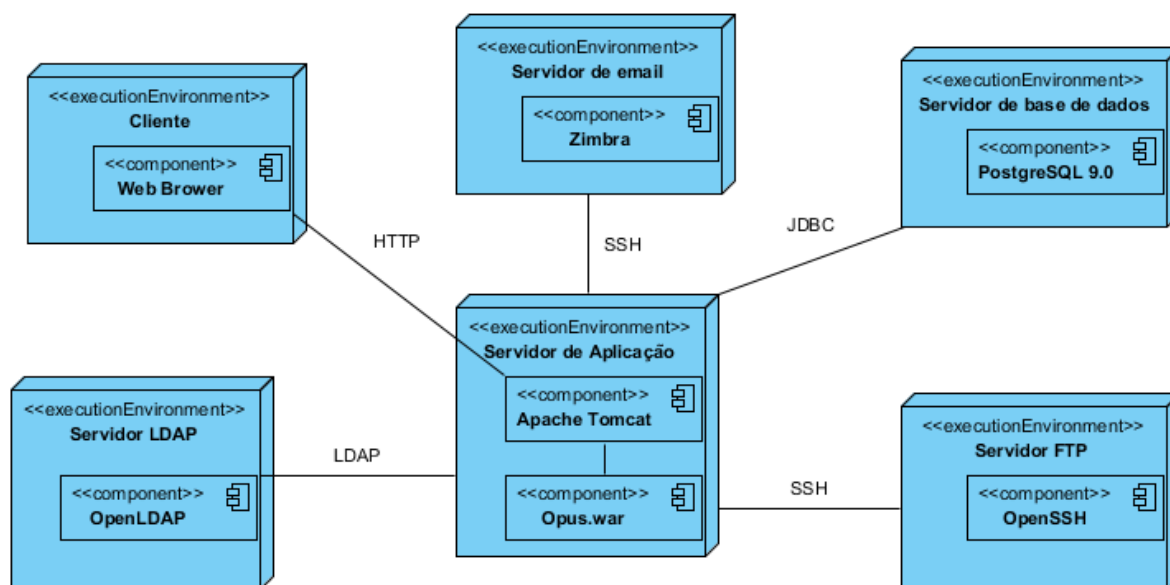


Figura 17 - Diagrama de instalação e componentes

Importa referir que a pesar de este diagrama retratar um ambiente distribuído, não significa que os serviços devem estar deste modo, isto é para demonstrar que o modulo esta preparado para funcionar no ambiente demonstrado acima, nada impede de existir dois ou mais serviços no mesmo servidor. De seguida é mostrada a especificação necessária do servidor de aplicações em ambiente de produção.

Tabela 6 - Estimativa de requisitos para componentes

Componente	Especificação mínima
Servidor de aplicação	Processador - Intel Core i3 Espaço em disco - 260 GB

Desenvolvimento de um módulo para gestão de contas de utilizadores

	Memoria Ram - 4 GB.
Plataforma de execução	Linux server (kernel version 2.6) / Windows 2003 Server
Maquina Cliente	Windows XP - IE 6.0, Firefox 1.5 Linux (kernel version 2.6) - Firefox 1.5

7. Desenho de diagramas

Diagrama de classes

O modelo de classes utilizado na implementação do módulo é descrito abaixo, e o mesmo foi desenhado com o objectivo de suportar os requisitos funcionais descritos.

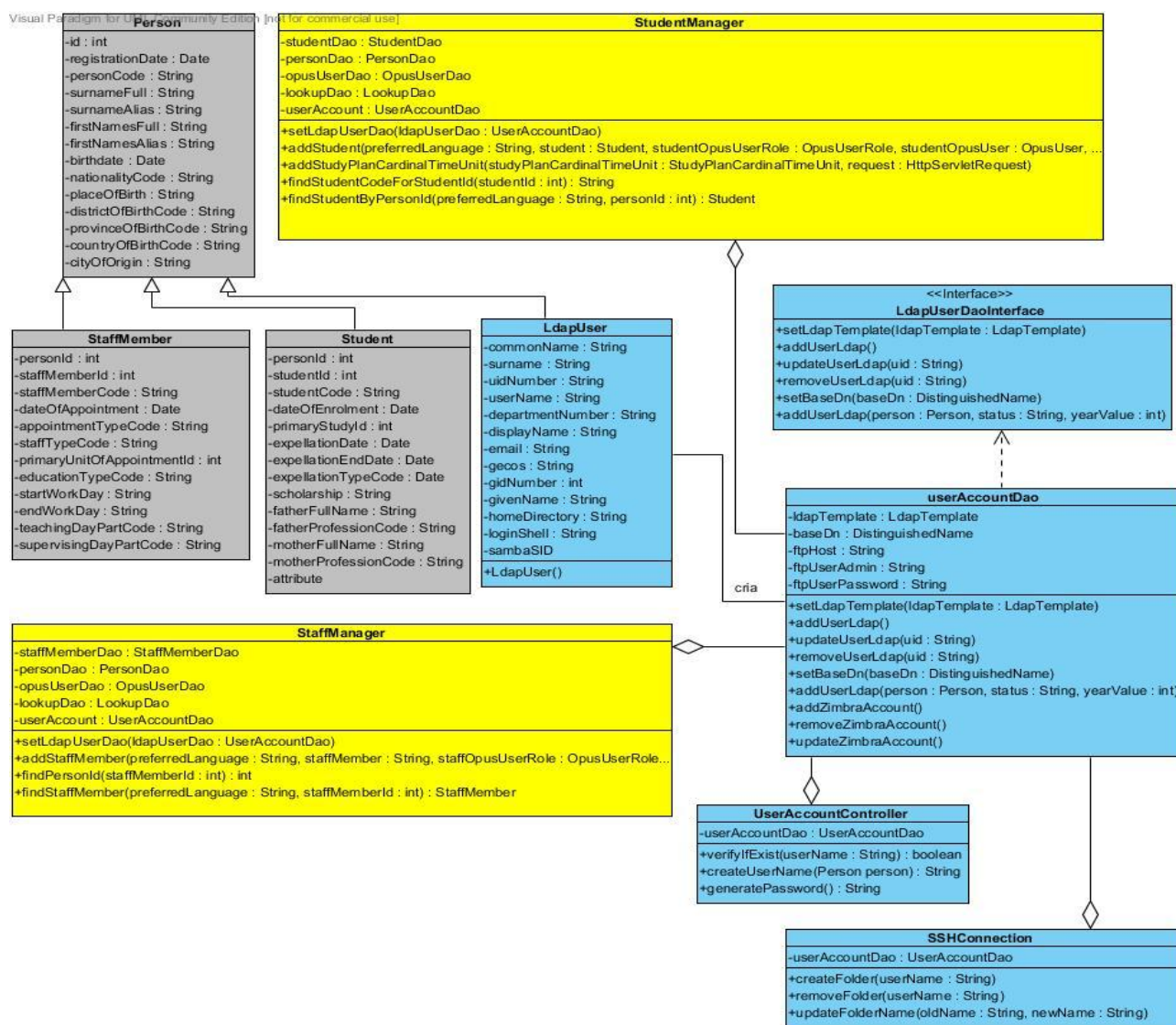


Figura 18- Diagrama de classes

Diagramas de actividades

De seguida são representados os diagramas de actividades, descrevendo o actual processo de execução das actividades a partir do módulo de gestão de contas de utilizadores.

Criação de contas de utilizadores

Caso 1: Criação da conta de estudante

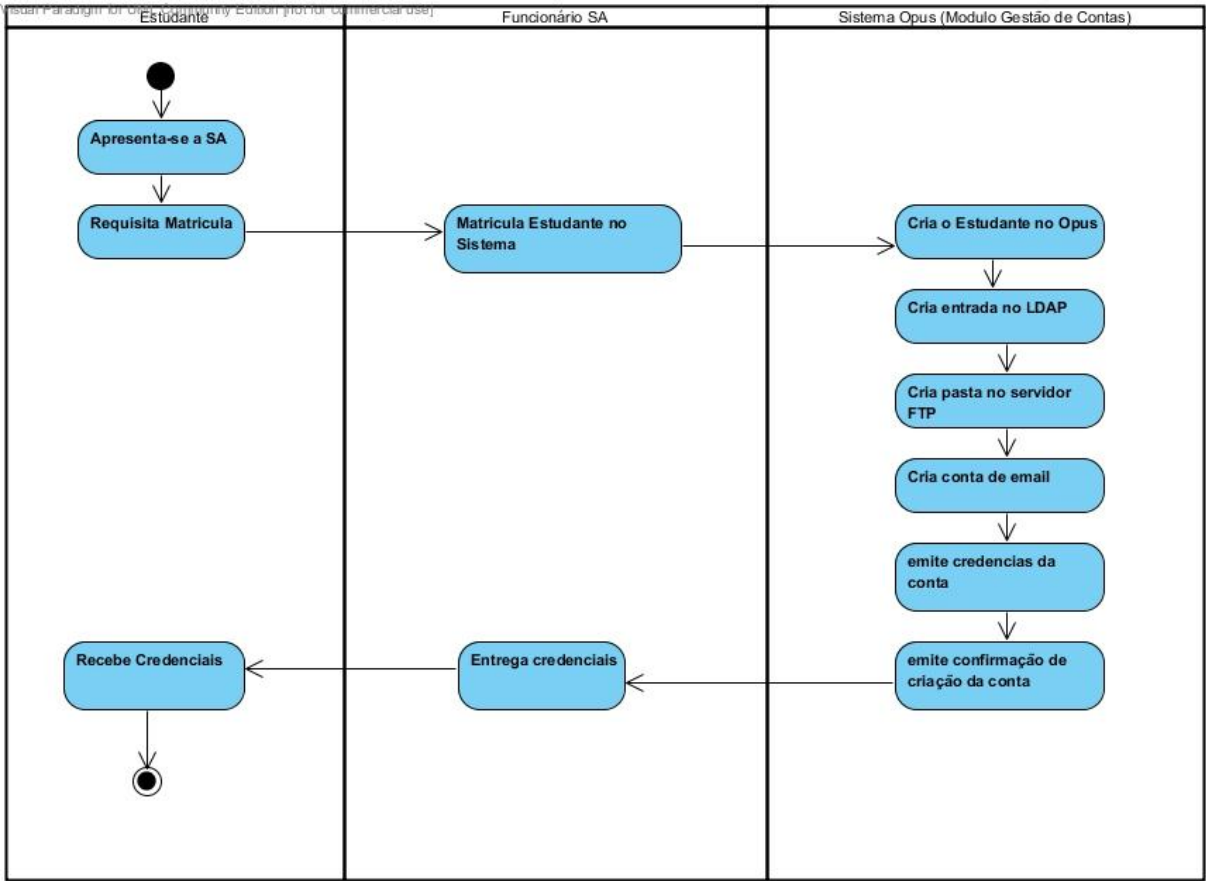


Figura 19- Diagrama de actividades para criação de contas de utilizadores para estudantes a partir do módulo de gestão de contas de utilizadores

Caso 2: Criação da conta do funcionario.

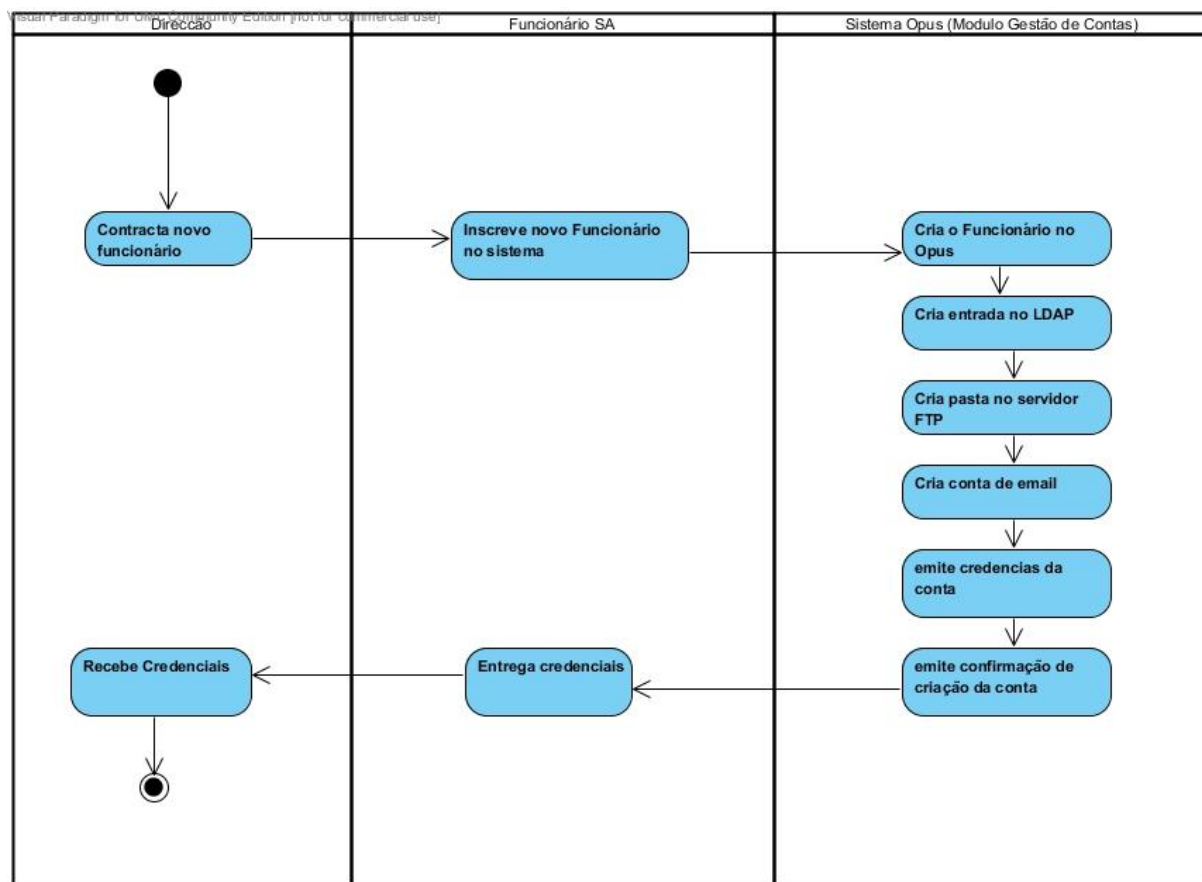


Figura 20-Diagrama de actividades para criação de contas de utilizadores para funcionários a partir do módulo de gestão de contas de utilizadores

Importa referir que a SA é responsável e fica responsável por criar a conta de utilizador do funcionário pois a mesma funciona como um sector de serviços administrativos.

Melhoramento introduzido

Após o desenho destes dois diagramas, pode-se efectuar uma comparação com o diagrama descrito no capítulo 3 sobre a criação de contas de utilizadores, e reparar que houve claramente uma melhoria no processo de trabalho existente.

São eliminadas as actividades manuais existentes como elaborar ficheiro, manipular informação no ficheiro, editar *script* e executar *script*, que eram anteriormente desempenhada pelos funcionarios da SA e do SecInf, eliminando estas actividades introduz-se uma redução de tempo de execução do processo e eliminação a probabilidade de erro humano.

O estudante recebe as credenciais da sua conta quase instantaneamente.

Actualização de conta de utilizadores

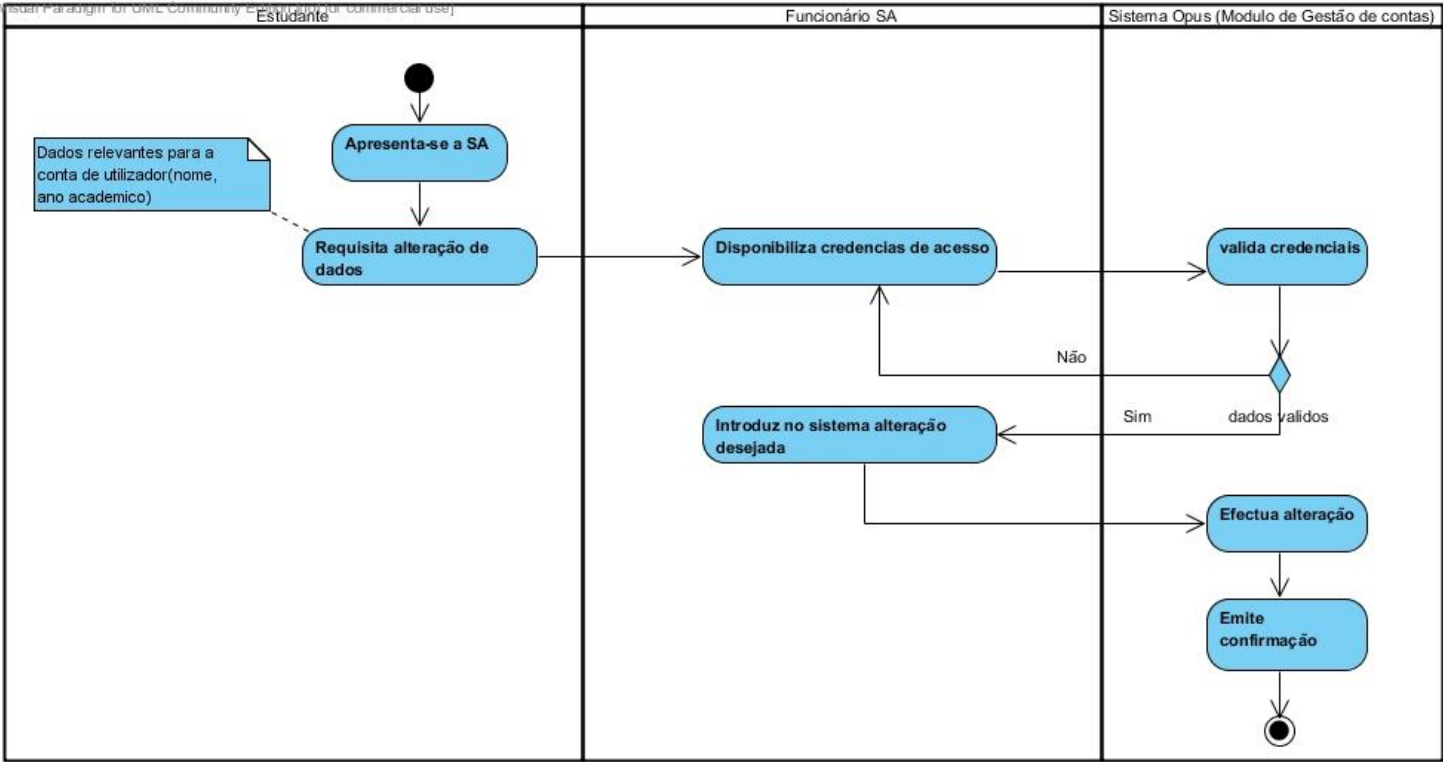


Figura 21-Diagrama de actividades para o processo de actualização de contas de utilizadores a partir do módulo de gestão de contas de utilizadores

Melhoramento introduzido

Este processo tem inicio com o estudante que normalmente identifica algum erro nos seus dados ou que em cada inicio de ano deseja efectuar a renovação da matricula, posteriormente segue-se com as outras actividades.

Este processo passa deste modo a ser executado em modo individual e não em modo massa, elimina-se todas actividades manuais e consequentemente introduz-se uma melhoria no tempo de execução do processo.

Remoção de contas de utilizadores

A remoção passa a conter dois cenarios, primeiro eliminar a conta de utilizador sem eliminar o registo no sistema *OPUS-College*. Segundo, eliminar o registo no *OPUS-College* e eliminar a conta de utilizador.

Caso 1: Eliminar conta de utilizador sem eliminar registo no *OPUS-College*

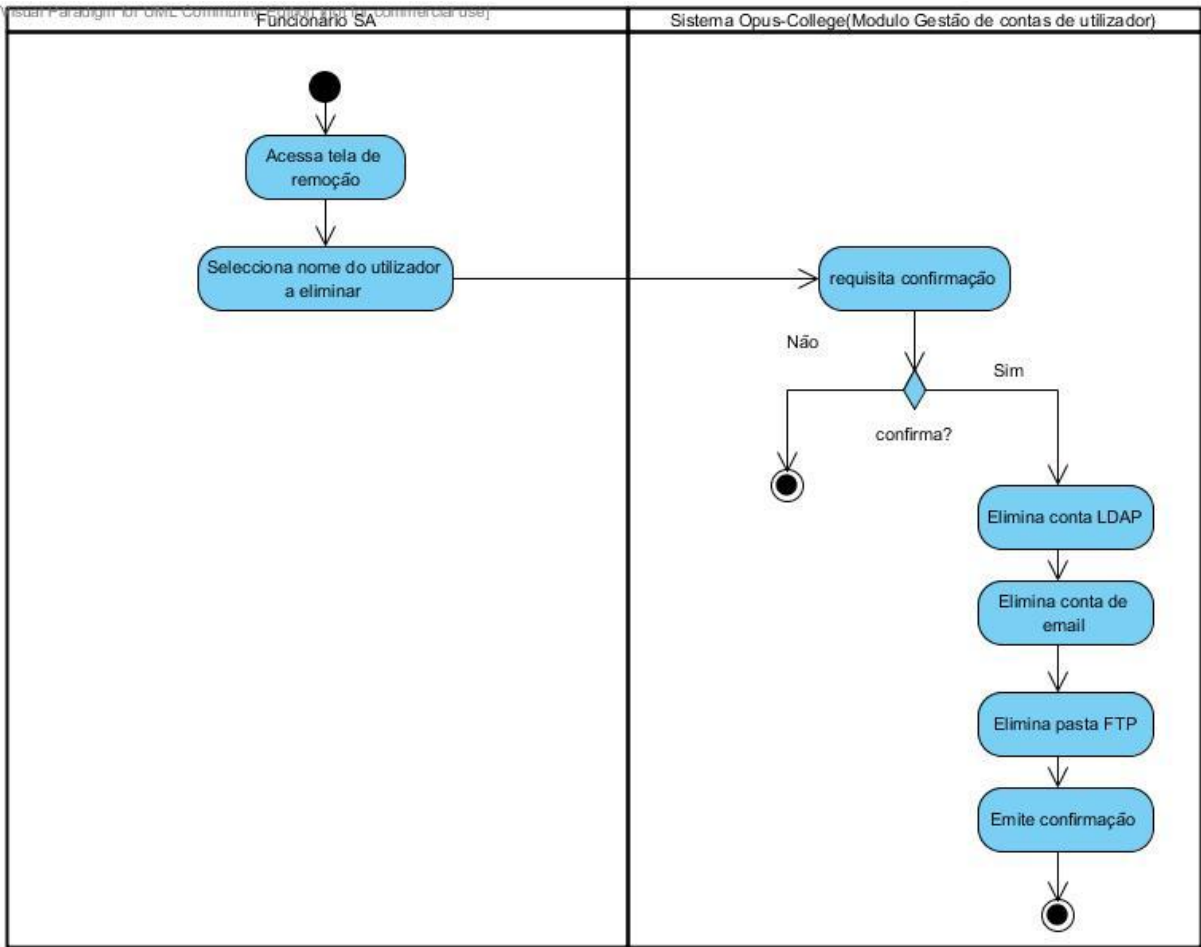


Figura 22 - Diagrama de actividade para remoção de conta sem eliminar registo no *OPUS-College*

Caso 2: Eliminação de conta a partir da eliminação do registo no *OPUS-College*

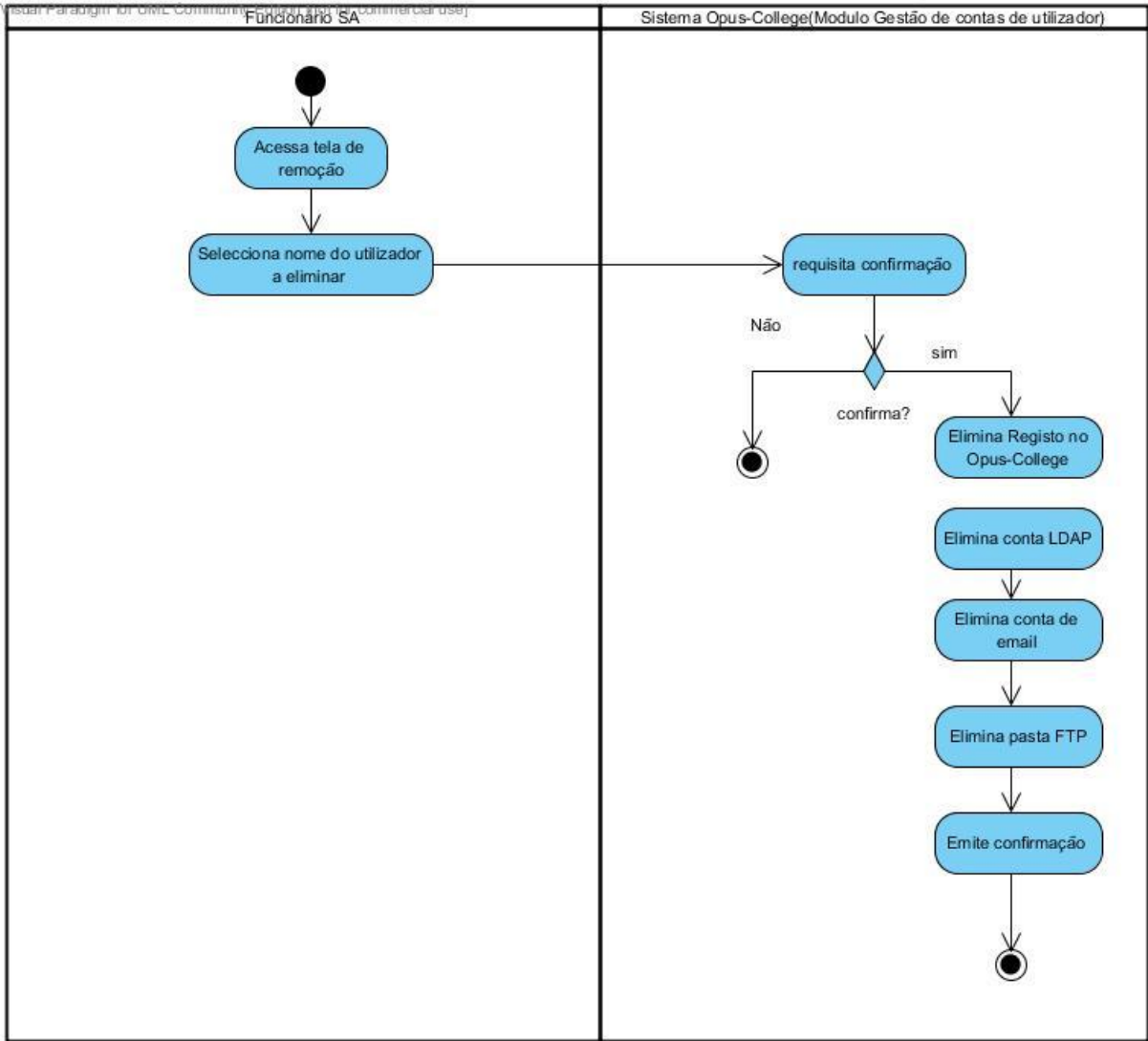


Figura 23-Diagrama de actividade para remoção de conta a partir da eliminação do registo no *OPUS-College*

Recuperação de senhas

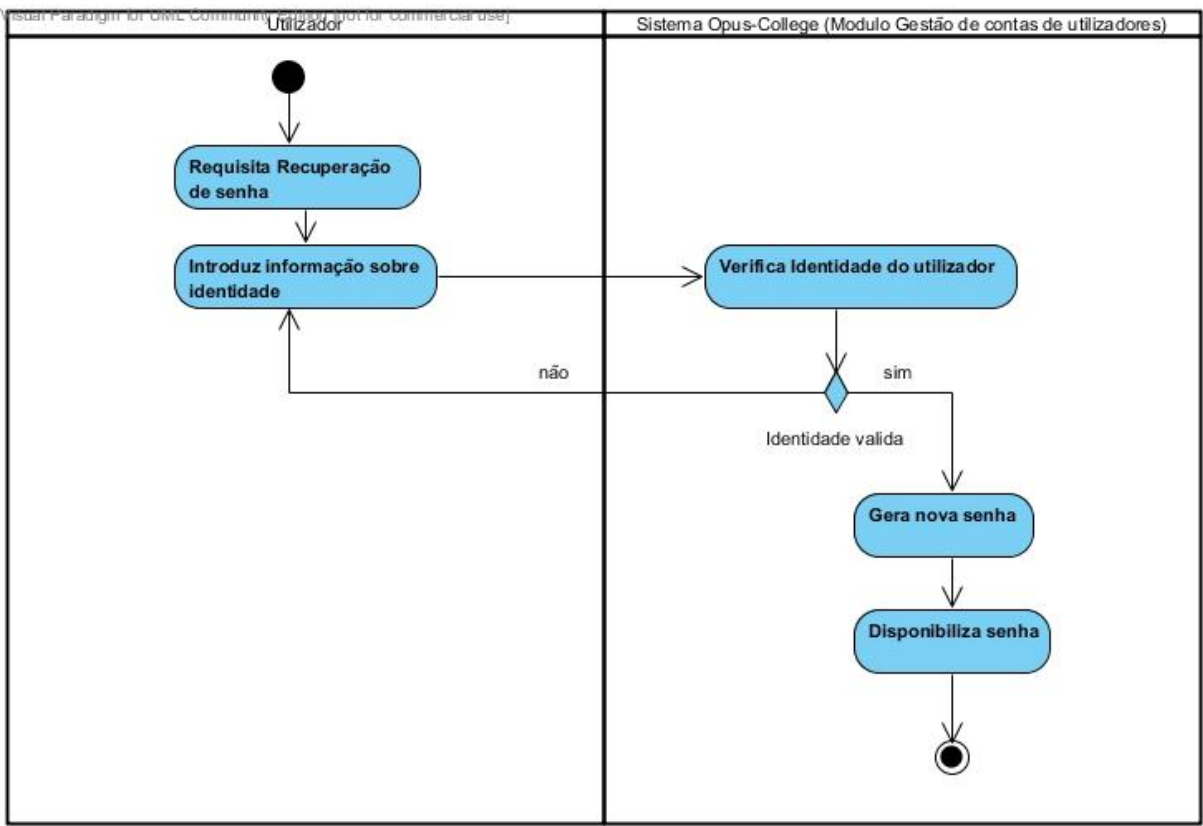


Figura 24-Diagrama de actividades para o processo de recuperação de senhas de utilizadores a partir do módulo de gestão de contas de utilizadores

Melhoramento introduzido neste processo

Este processo é melhorado eliminando-se todo o contacto do utilizador com intervenientes humanos, garantindo deste modo a confidencialidade da sua informação e melhorando o tempo de execução do processo.

Diagramas de sequência

De seguida são mostrados os diagramas de sequência para os casos de uso.

Nota: Os diagramas de sequência foram desenhados ignorando toda a sequência de objectos referente a autenticação.

Criação de conta de utilizador

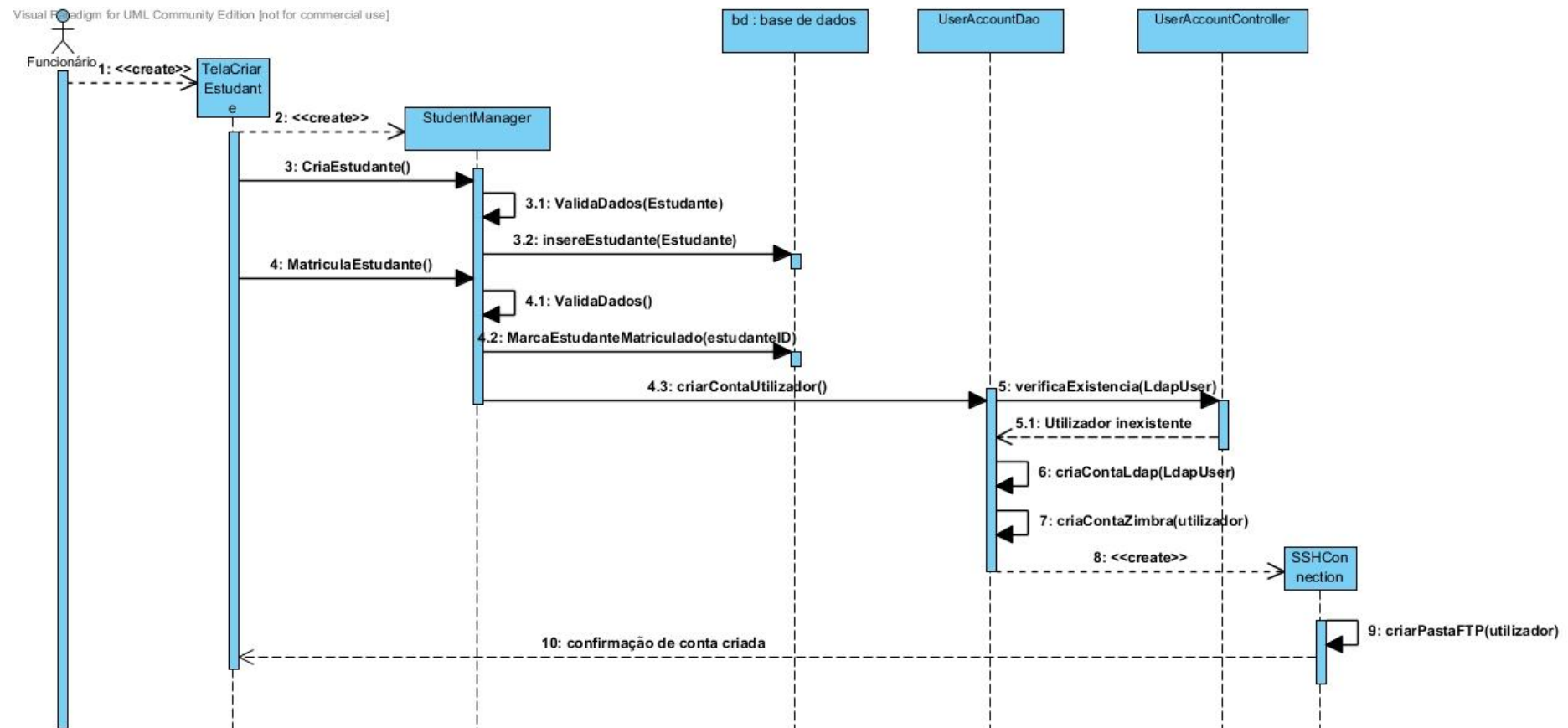


Figura 25 -Diagrama de sequência para o caso de uso criação de conta de utilizador

Desenvolvimento de um módulo para gestão de contas de utilizadores

Este diagrama foi desenhado para a criação de contas de utilizadores referente a estudantes, para o caso de funcionários, efectua-se a substituição do objecto *studentManager* pelo objecto *staffMemberManager*.

Actualização de conta de utilizador

Caso 1: Transição do ano académico dos estudantes.

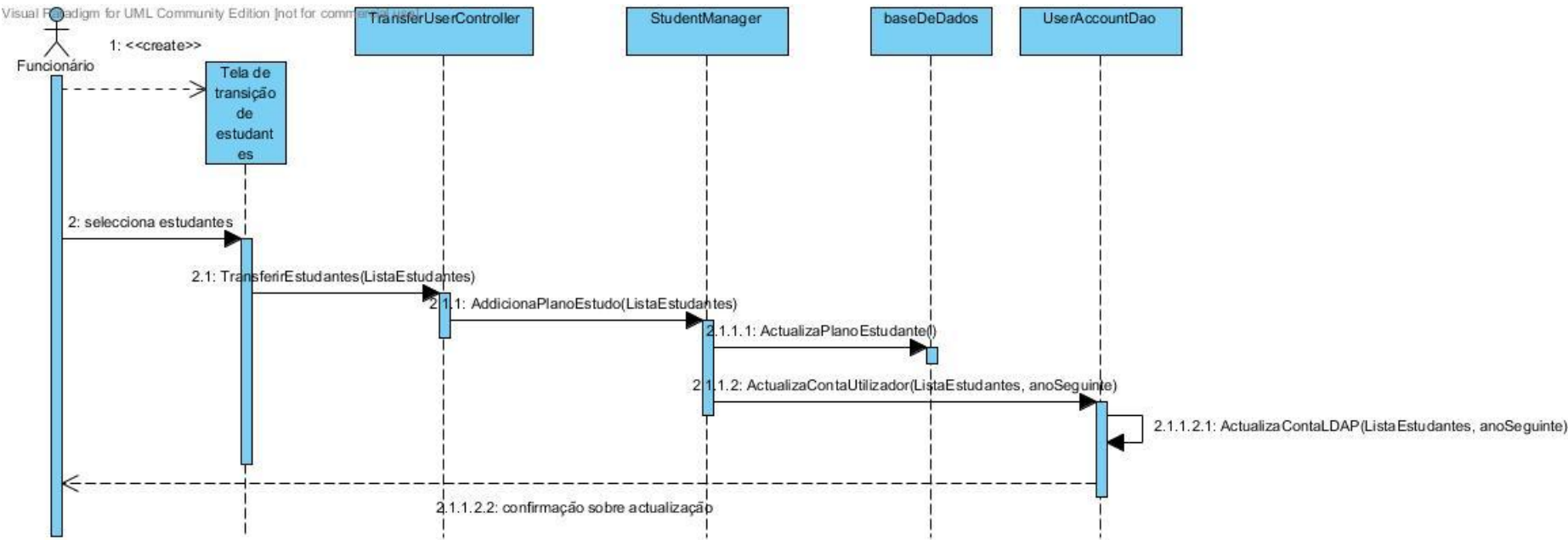


Figura 26- Diagrama de sequência para actualização do ano académico

Caso 2: Actualização de informações sobre estudante

Este caso refere-se a situações em que deve-se alterar informações básicas sobre o estudante, como o nome ou mesmo o nome de utilizador, a mesma situação acontece para o funcionario e deve-se efectuar a substituição do objecto *StudentManager* pelo objecto *staffMemberManager*.

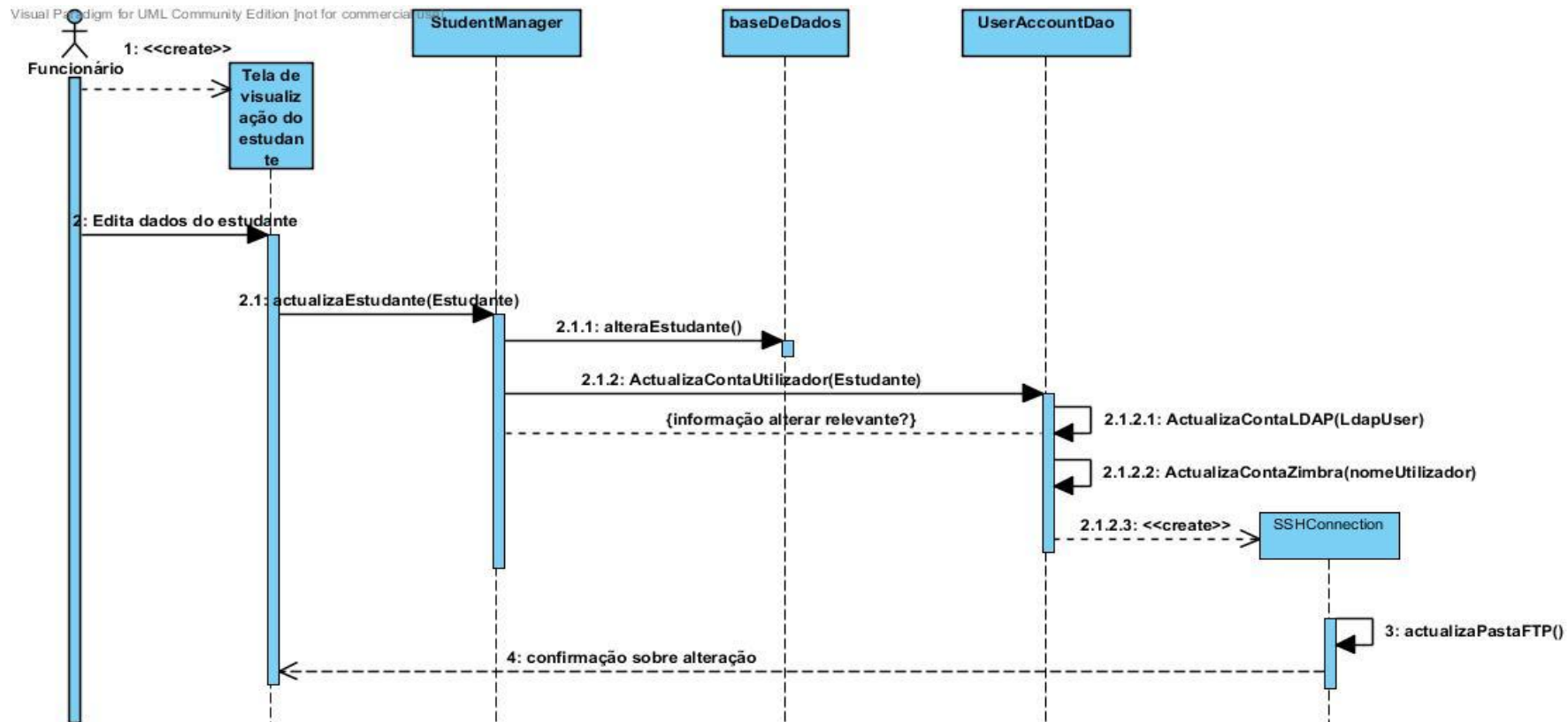


Figura 27 - Diagrama de sequência para actualização de informações do estudante

Remoção da conta de utilizador

Caso 1: Eliminar conta de utilizador sem eliminar registo no *Opus-College*.

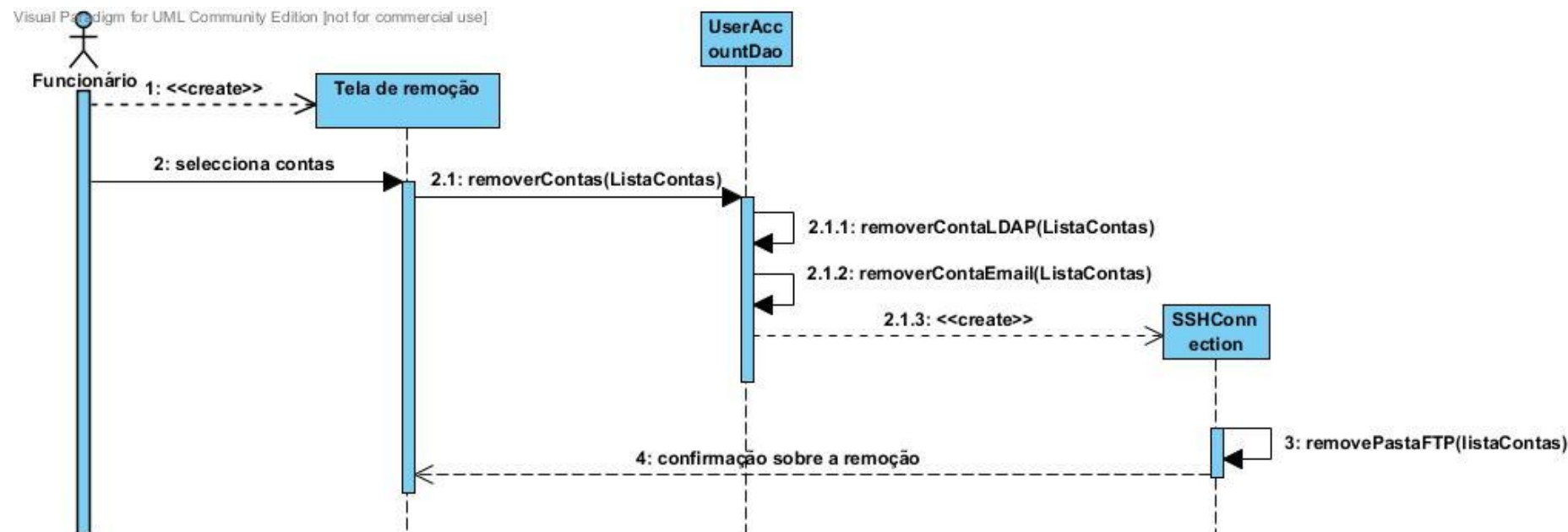


Figura 28- Diagrama de sequencia para remoção da conta de utilizador sem eliminar o registo no *OPUS-College*.

Caso 2: Eliminação de conta a partir da eliminação do registo no *OPUS-College*.

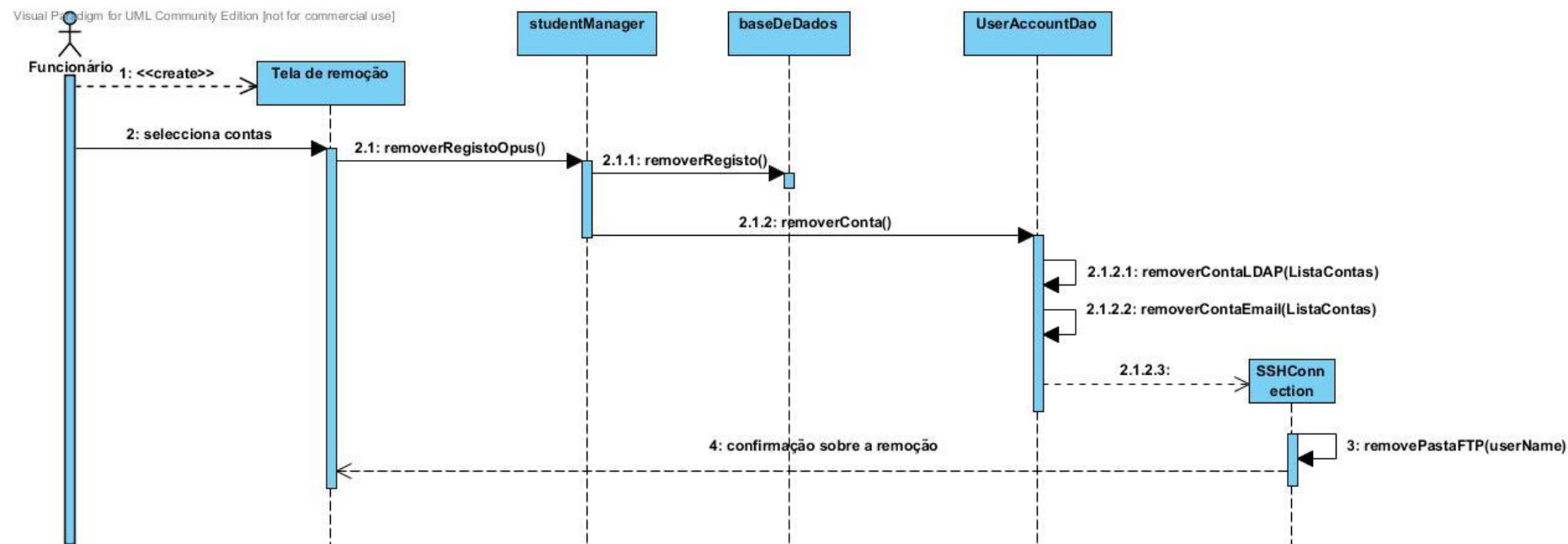


Figura 29 - Diagrama de sequencia para remoção da conta de utilizador a partir da eliminação do registo no *OPUS-College*

Recuperação de senha

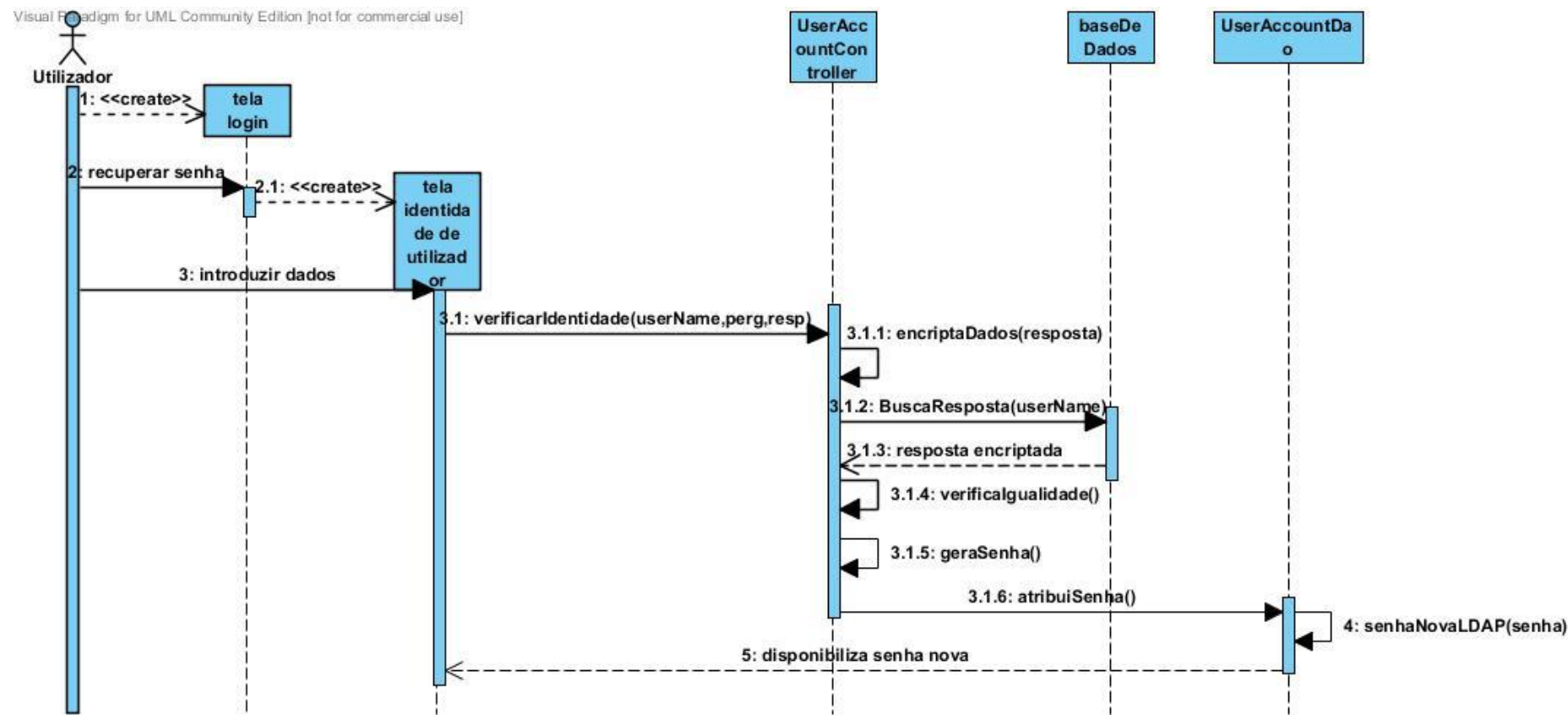


Figura 30 - Diagrama de sequencia para recuperação de senhas

4.4.3 Testes ao módulo

Esta secção do projecto consite na elaboração das seguintes actividades:

1. Testes as funcionalidades

Os testes as funcionalidades contou com a participação de dois funcionários do ISUTC, concretamente um funcionário do SecInf (Ted Fernandes) para validação da acção do módulo referente as contas de utilizador no ISUTC, e um analista de testes, funcionário do LIMEAA (Belzenia Matsimbe).

2. Resultado dos testes

Tabela 7 - Resultado dos testes as funcionalidades

Caso de uso	Estado	Descrição do problema
Criação de contas	Funcional (Incompleta)	A aplicação efectua a criação de contas, entretanto apenas faz a introdução da entrada no LDAP e da pasta FTP. Não efectua a criação da conta de <i>email</i> zimbra.
Remoção de contas a partir da eliminação do registo no <i>OPUS-College</i> .	Funcional	
Actualização de contas a partir da actualização do registo no <i>OPUS-College</i>	Funcional	
Recuperação de senhas	Funcional	

4.4.4 Operação do módulo

Esta fase consiste no conjunto das seguintes actividades sendo as mesmas funções do funcionário do SecInf:

- Garantia de disponibilidade do *software*.
- Procedimentos de rotinas de *backups* e de restauração.
- Assegurar a formação contínua de funcionalidades existentes e novas para os utilizadores finais.
- Monitorização do ficheiro de *logs*, para identificação prévia de problemas.
- Definição e actualização de um manual de operações do *software*.
- Suporte sempre que necessário ao utilizador final.

4.4.5 Manutenção do módulo

A definição deste processo vai consistir no conjunto das seguintes subactividades.

- **Definição de processo para gestão de alterações**

Definir a sequência de acções capazes de priorizar, categorizar, documentar, aceitar ou rejeitar as modificações sugeridas.

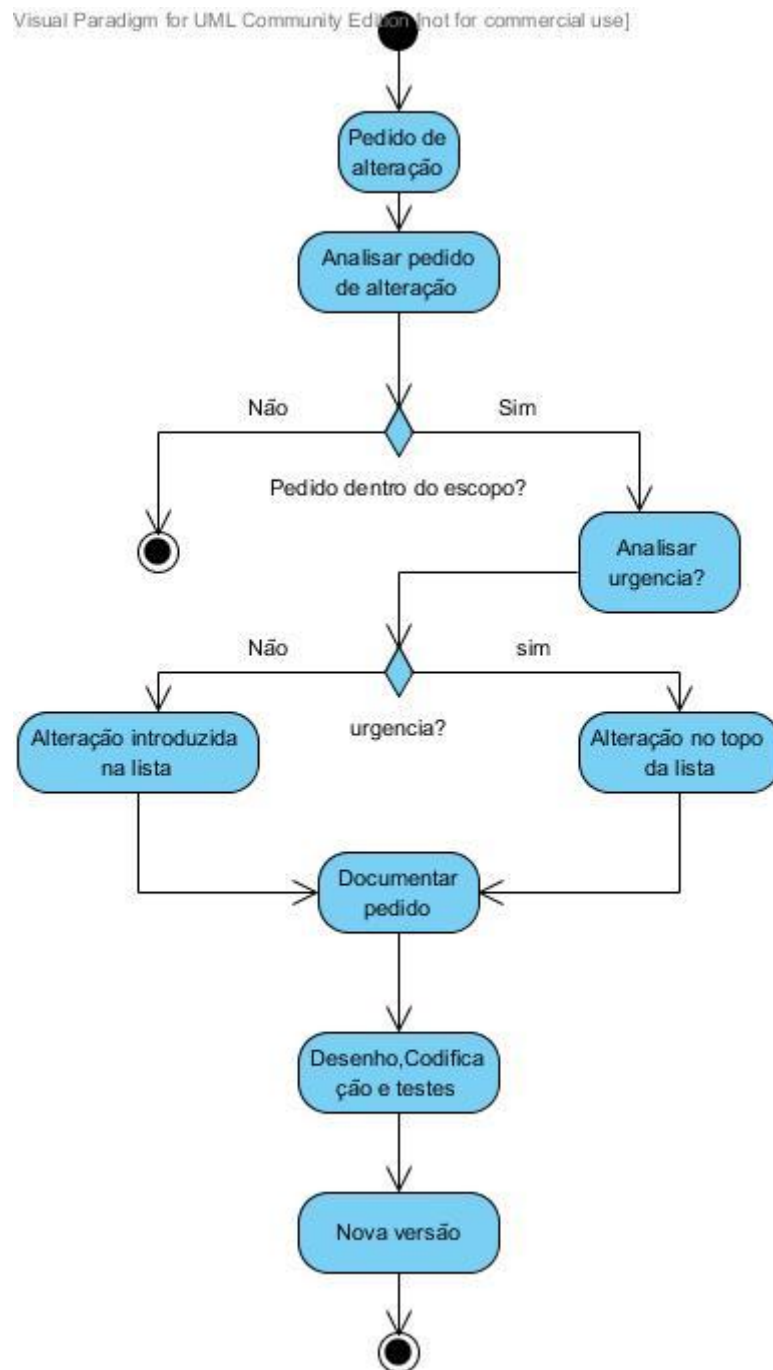


Figura 31 - Diagrama de actividade para gestão de pedidos de alteração

- **Definição de processo de Transição**

Definir uma sequência de acções controladas e coordenadas onde o sistema é transferido do ambiente de desenvolvimento para o ambiente de produção.



Figura 32- Diagrama de actividades para disponibilização da aplicação em ambiente de produção

- **Definição de um acordo de nível operacional**

Um acordo feito entre o departamento de desenvolvimento e os departamentos que fazem uso da aplicação. Este acordo é necessário para garantir que a entrega das versões da aplicação contendo as alterações sejam feitas dentro dos prazos estimados.

4.5 Estimativa de custos

De seguida é efectuada uma estimativa dos custos do projecto, para possibilitar a implementação e estabilização em produção do módulo descrito.

Os custos foram elaborados segundo a tabela de salários do LIMEAA.

Desenvolvimento de um módulo para gestão de contas de utilizadores

Tabela 8- Estimativa de custos

	Descrição	Custo
Custo de desenvolvimento	Estima-se que um Programador de nível II precisaria de 12 horas semanais durante 6 meses para desenvolver todos os casos de uso.	$94 \text{ Mt/h} \times 24 \text{ Semanas} \times 12 \text{ h}$ $\cong 27.072 \text{ Mt}$
Custo de implementação	Estima-se que somente o ISUTC contém todos os outros componentes.	35.000 Mt
Custo de Manutenção	A manutenção implica manter o programador e submeter o módulo a testes contínuos durante 2 meses.	
	• Programador	$94 \text{ Mt/h} \times 8 \text{ Semanas} \times 12 \text{ h}$ $\cong 9.024 \text{ Mt}$
	• Testador	$79 \text{ Mt/h} \times 8 \text{ Semanas} \times 12 \text{ h}$ $\cong 7.584 \text{ Mt}$
	Total	78.680 Mt

CAPÍTULO 5 - CONCLUSÕES E RECOMENDACÕES

5.1 Conclusões

O projecto teve como propósito desenvolver um módulo de gestão de contas de utilizadores integrado a aplicação *OPUS-College*, foi possível numa primeira fase analisar-se os principais conceitos necessários para apoiar o alcance dos objectivos propostos.

Referente aos objectivos da investigação, foi feita a descrição dos actuais processos de gestão de contas de utilizadores, informação esta adquirida a partir de entrevistas aos membros do SecInf, onde foi possível identificar as principais limitações que estes possuem, as mesmas residem essencialmente no tempo de execução dos processos, na probabilidade de ocorrência de erros, na falta de privacidade existente no processo de recuperação de senhas.

Foi possível a partir das mesmas entrevistas identificar os requisitos funcionais e não funcionais, verificou-se que os requisitos criação de contas e actualização e recuperação de senhas são os de maior prioridade.

Verificou-se o melhoramento que pode ser atingido em todos os processos de trabalho após implementação do módulo, onde este resume-se essencialmente na eliminação de actividades manuais e consequentemente em ganho de tempo de execução e redução da probabilidade de erros cometidos.

Efectuou-se a modelação de todos os requisitos funcionais, posteriormente a codificação dos casos de uso funcionais e de seguida os testes, onde neste ultimo verificou-se que o caso de uso criação de conta de utilizador permanece incompleto, pois existe a necessidade de efectuar-se a criação da conta de *email*.

5.2 Recomendações

Tendo em conta que esta solução introduz um melhoramento nos processos de trabalho para a gestão de contas de utilizador no ISUTC, surge a necessidade de efectuar-se trabalhos e estudos futuros em cima desta solução.

Recomenda-se um estudo detalhado referente a custos de desenvolvimento, manutenção e implementação.

No caso de desejo de introduzir este módulo em produção é recomendável que sejam desenvolvidos planos de desenvolvimento e planos de testes, executados por profissionais qualificáveis para tal, isto com vista a estabilização do módulo.

Desenvolvimento de um módulo para gestão de contas de utilizadores

Recomenda-se que o desenvolvimento contínuo do módulo seja acompanhado por um profissional experiente na área de desenvolvimento de sistemas.

Recomenda-se que o sistema *OPUS-College* seja implementado em um ambiente completamente seguro, deste modo garantindo a segurança do módulo em questão.

REFERÊNCIAS BIBLIOGRÁFICAS

1. AUDY, J. Metodologia de pesquisa em sistemas de informação e engenharia de *software*. Disponível em: <<http://pt.scribd.com/doc/6943465/Metodologia-de-Pesquisa>> Acesso 09 Outubro de 2012.
2. BASS, L.; CLEMENTS, P.; KAZMAN, R. *Software Architecture in Practice*. 3^a ed. New York: Addison-Wesley, 2003.
3. BERNARDI, A. *Software* de auxilio a implantação da norma ISO/IEC 12207: Processos do ciclo de vida do software. Dissertação de Bacharelato – Universidade Regional de Blumenau – Blumenau, 2003.
4. BUTCHER, M. *Mastering OpenLDAP*: Configuring, Securing, and Integrating Directory Services. 1^a ed. Birmingham: Packt, 2007.
5. CHAVES, T. R. Estudo de caso: Autenticação IEEE 802.1X Baseada no protocolo *radius* e serviço de directório LDAP aplicado a rede GIGAUFOPNET. Dissertação de Bacharel – Universidade Federal – Ouro Preto, 2010.
6. CARRARA, L.; TRIGO, C. Autenticação centralizada com *OpenLDAP*. Disponível em: <<http://pt.scribd.com/doc/100023057/tcc-2011-11-18>> Acesso 15 Março de 2012.
7. ISQUIERDO G. Integração do Serviço de Directório LDAP com o Serviço de Nomes CORBA. Dissertação de Mestrado – Universidade de São Paulo – São Paulo. 2001.
8. PRESSMAN, R. *Engenharia de software*. 3^a ed. São Paulo: Makron Books, 1995.
9. FOWLER, M. **UML essencial**: um breve guia para a linguagem padrão de modelagem de objectos. 3^a ed. Porto Alegre: Bookman, 2005.
10. GARLAN, D.; SHAW, M. An Introduction to software Architecture. Disponível em: <http://www.cs.cmu.edu/afs/cs/project/vit/ftp/pdf/intro_softarch.pdf> Acesso em 12 de Novembro de 2012.
11. KOURI, M. G. Definição de requisitos para um sistema de monitoramento de veículos no transporte rodoviário de cargas. Dissertação de Mestrado – Universidade de São Paulo – São Paulo, 2007.
12. LIMA, C.D; JUNIOR, J. P. Relacionando Engenharia de Requisitos a Engenharia de Software Orientada a Modelos. Disponível em: <http://www.cin.ufpe.br/~in1020/arquivos/monografias/2009_1/carlos_josias.pdf> Acesso em 12 Novembro de 2012.

13. LOPES, F.; MORAIS M.; CARVALHO J. **Desenvolvimento de sistemas de informação**. 2ª ed. FCA editora. 2005.
14. O'NEILL, H.; NUNES, M. **Fundamental de UML**. 2ª ed. Lisboa: FCA editora, 2004.
15. SOMMERVILLE, I. **Software Engineering**. 8ª ed. New York: Addison-Wesley, 2007.
16. SOMMERVILLE, I. **Engenharia de software**. 6ª ed. São Paulo: Addison-Wesley, 2003.
17. SOUZA O. Processos de Apoio ao Desenvolvimento de Aplicações Web. Dissertação de Mestrado – Instituto de ciências matemáticas e de computação- São Carlos. 2005.
18. SIMONS, ED. **Opus-college: Student information And Registration system**. Disponível em <<http://www.opus-college.net/>> acesso em: 19 de Maio de 2013.
19. BARROS, T.; SILVA, M.; ESPINOLA, E. *State MVC* : Estendendo o padrão MVC para o uso no desenvolvimento de aplicações para dispositivos móveis. Disponível em: <http://www.cesar.org.br/site/files/file/SMVC_01_0104.pdf> Acesso 15 Junho de 2013.
20. UGHINI C. Uma Proposta para Utilização de LDAP na Rede do Instituto de Informática. Dissertação de Bacharel – Universidade Federal – Porto alegre. 2004.

BIBLIOGRAFIA

1. CRISTINA B. Avaliação de processos de *software* – Um estudo de caso em XP e ICONIX. Dissertação de Mestrado – Universidade Federal de Santa Catarina – Florianópolis, 2002.
2. OLIVEIRA, M.; MAÇADA, A.; GOLDONI, V. Análise da aplicação do método Estudo de caso na área de Sistema de Informação. Disponível em:
<[www.ea.ufrgs.br/professores/acgmacada/pubs/enanpad Mirian macada 2006.pdf](http://www.ea.ufrgs.br/professores/acgmacada/pubs/enanpad%20Mirian%20macada%202006.pdf)>
Acesso 10 Outubro de 2012.
3. PILONE, D. **UML**: Pocket Reference. Califonia: O'reilly, 2003.
4. VAROTO, A. C. Visões em arquitectura de *software*. Dissertação de mestrado – Universidade de São Paulo – São Paulo, 2002.

ANEXOS

ANEXO I

5. Entrevistas aos funcionários do sector de informática

Entrevista 1

No dia 27 de Março de 2013 das 14:00 – 15:00 Horas foi feita uma entrevista estruturada com perguntas abertas ao Eng. Micas Rafael membro do sector de informática do Instituto Superior de Transportes e Comunicações.

Nota: todos os resultados são situações indirectas do entrevistado.

Participantes

- Eng. Micas Rafael
- Yuri wingester

Resultados

1. O que é gestão de contas de utilizadores no ISUTC?

Significa gerir contas existentes no nosso domínio, que possibilitam aceder as estações de trabalho e os serviços de rede por nos implementados, como o acesso a internet, serviço de FTP, etc.

2. Que processos fazem parte da gestão de contas de utilizadores?

Os processos que fazem parte da gestão de contas de utilizadores no ISUTC são:

- Criação de contas
- Gestão de campos das contas (Actualização)
- Remoção

No aspecto de remoção o funcionário após retirar-se do ISUTC tem uma semana para obter toda a informação que deseja na sua conta de utilizador, caso contrário tudo será eliminado, no caso de um estudante, o mesmo tem um ano para efectuar este processo.

No aspecto de actualização, muitas das vezes é somente actualizado o campo department Number do LDAP o qual refere a turma em que o estudante esta inserido.

3. O que significa uma conta de utilizador no ISUTC?

A conta de utilizador no ISUTC é uma identificação atribuída ao utilizador, consiste em, criação de conta de acesso as estações de trabalho, correio electrónico e acesso ao serviço ftp.

4. Como pensa que a automação destes processos pode ajudar?

Introduz menos tempo na execução do trabalho, mais eficaz.

Entrevista 2

No dia 28 de Março de 2013 das 15:00 – 16:00 Horas foi feita uma entrevista estruturada com perguntas abertas ao Eng. Vanessa Mabunda membro do sector de informática do Instituto Superior de Transportes e Comunicações.

Nota: todos os resultados são situações indirectas do entrevistado.

Participantes

- Eng. Vanessa Mabunda
- Yuri wingester

Resultados

1. O que é gestão de contas de utilizadores no ISUTC?

Gestão de utilizadores reside essencialmente nos seguintes pontos:

- Criação de contas de domínio
- Conta de *email*
- Actualização das mesmas

O que mais se faz é a recuperação de senhas das contas.

2. Que processos fazem parte da gestão de contas de utilizadores?

Os processos que fazem parte da gestão de contas de utilizadores no ISUTC são:

- Criação de conta no *OpenLdap*
- Criar pasta de ftp
- Criar conta de *email*

3. O que significa uma conta de utilizador no ISUTC?

É um identificador que permite o acesso aos serviços de rede no ISUTC.

4. Como pensa que a automação destes processos pode ajudar?

Redução de tempo.

ANEXO II

Controle de versões

Versão	Data	Autor	Descrição
1.1	06-31-2013	Yuri Wingester	Definição de casos de uso prioritários
1.0	05-13-2013	Yuri Wingester	Criação do documento original

1. Introdução

1.1 Propósito

Este documento tem como propósito efectuar a especificação dos requisitos contemplados para a execução do PFC com o tema desenvolvimento de um módulo para gestão de contas de utilizadores no Instituto Superior de Transportes e Comunicações.

Este documento é direccionado a todo o leitor que tem a necessidade de perceber o módulo a ser desenvolvido e ao desenvolvedor do mesmo.

1.2 Escopo

O produto de *software* esperado no final deste projecto final de curso, é um módulo integrado a aplicação destinada a gestão escolar no ISUTC, opus-college. Este módulo será projectado para melhor os processos de gestão de contas de utilizadores no ISUTC, efectuando a automatização destes processos.

1.3 Definições, acrónimos e abreviações

Nota: Os termos são descritos no contexto deste documento.

Termo	Descrição
ISUTC	Instituto superior de transportes e comunicações
PFC	Projecto final de curso
RF	Requisito Funcional
RNF	Requisito Não funcional
Essencial	Indica que o requisito é imprescindível para o funcionamento do sistema.
Importante	indica que o requisito não é essencial para o funcionamento do sistema, contudo seu funcionamento sem implementação do requisito, se torna insatisfatório.
Desejável	Indica que o requisito não compromete as funcionalidades básicas do sistema.
SA	Secretaria Academica
SecInf	Sector Informatico

1.4 Referências

1. IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.
2. Entrevista aos funcionários do sector de informática.

a. Organização do documento

O documento é dividido em três capítulos, onde primeiramente encontra-se a descrição geral, este é constituído pelos seguintes subcapítulos, características dos utilizadores e restrições, no primeiro são abordados aspectos gerais sobre os possíveis utilizadores do módulo a ser desenvolvido, e no segundo possíveis restrições.

Posteriormente são descritos os requisitos funcionais e não funcionais para o módulo em questão.

De seguida, encontra-se a descrição dos aspectos que deverão ser documentados, no processo de desenvolvimento do módulo e posteriormente ao mesmo.

3. Descrição Geral

a. Perspectivas do produto

Ao fim da execução deste trabalho, pretende-se obter um módulo que seja possível integrar na aplicação *Opus-College* e que esteja capacitado para prover soluções aos requisitos aqui descritos.

b. Funções do produto

Caso de uso: Criação de conta de utilizador.

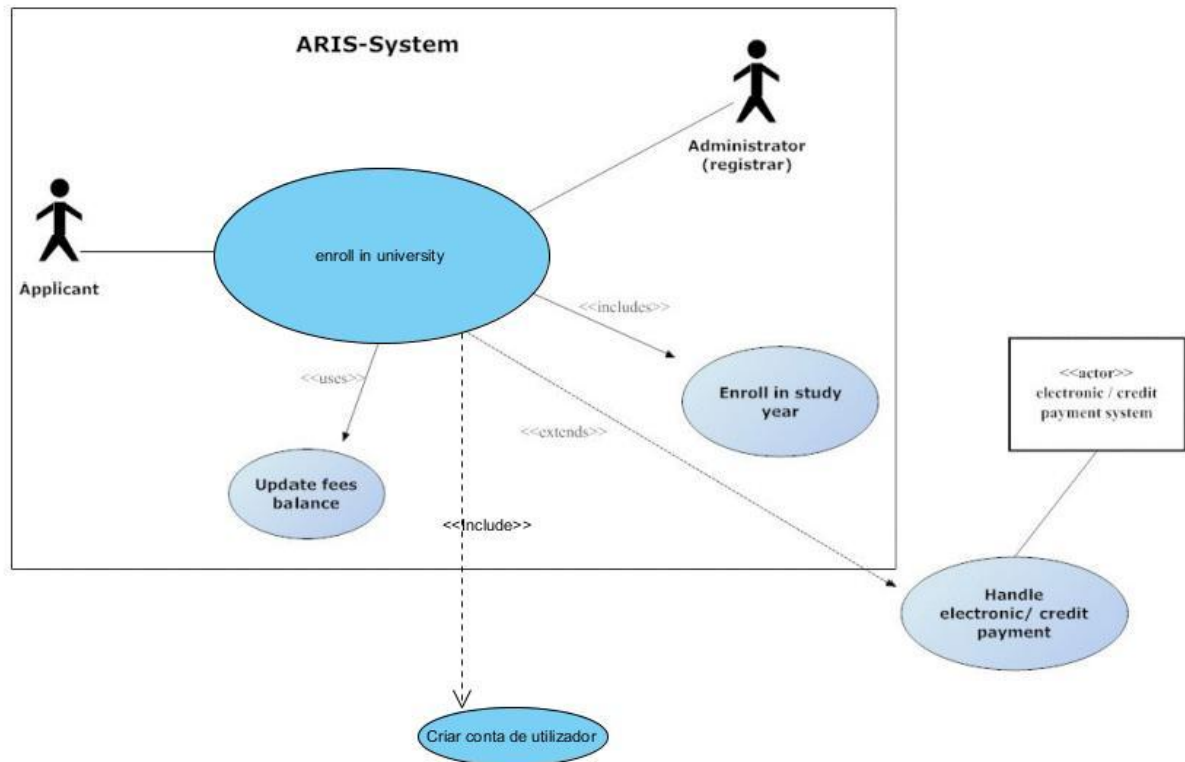
Descrição: Após efectuar-se a matrícula de um novo estudante ou funcionário o módulo integrado na aplicação *Opus-College* deve criar automaticamente uma conta de utilizador no ISUTC.

Cenário 1: Adição de conta de estudante

NOTA: (Adaptação do diagrama de caso de uso *Enroll in University* da aplicação opus-college, adição do caso de uso “Criar conta de utilizador” como um *include*)

Visual Paradigm for UML Community Edition [not for commercial use]

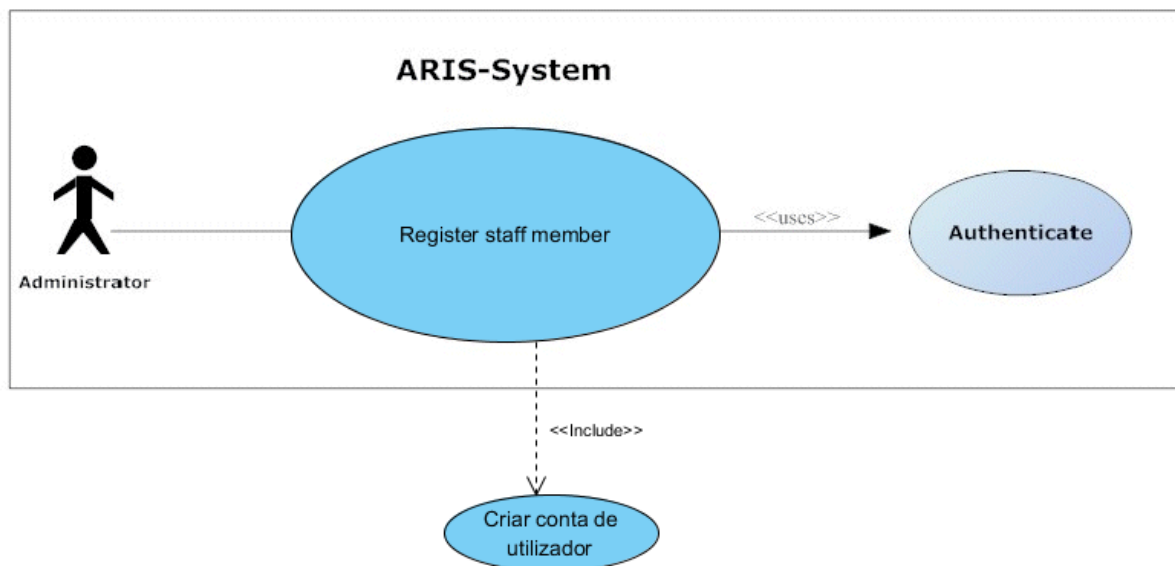
Use Case ENR1: Enroll in University



Cenário 2 : Adição de conta de funcionário

NOTA: Adaptação do use case “*register staff member*”

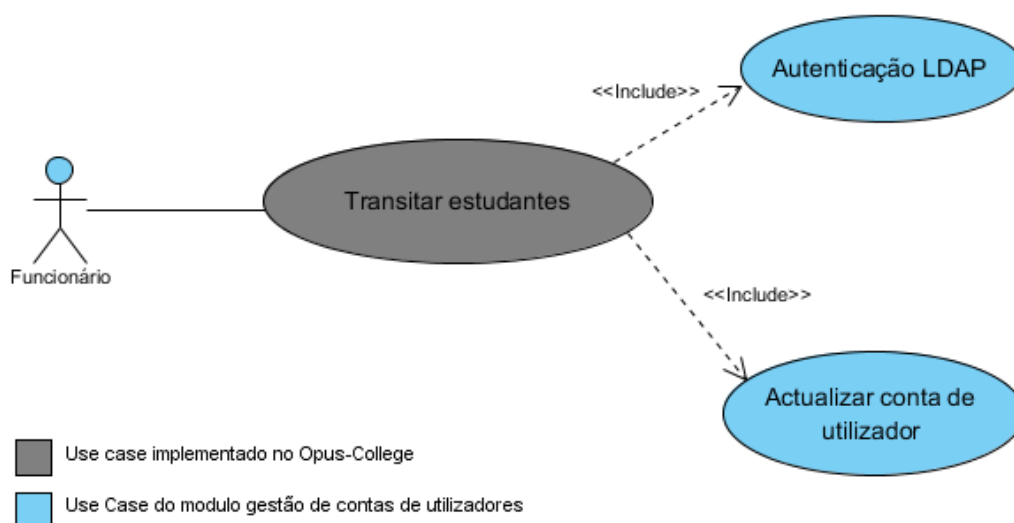
Use Case FA2: Register Staff Member



Caso de uso: Actualização de contas de utilizadores

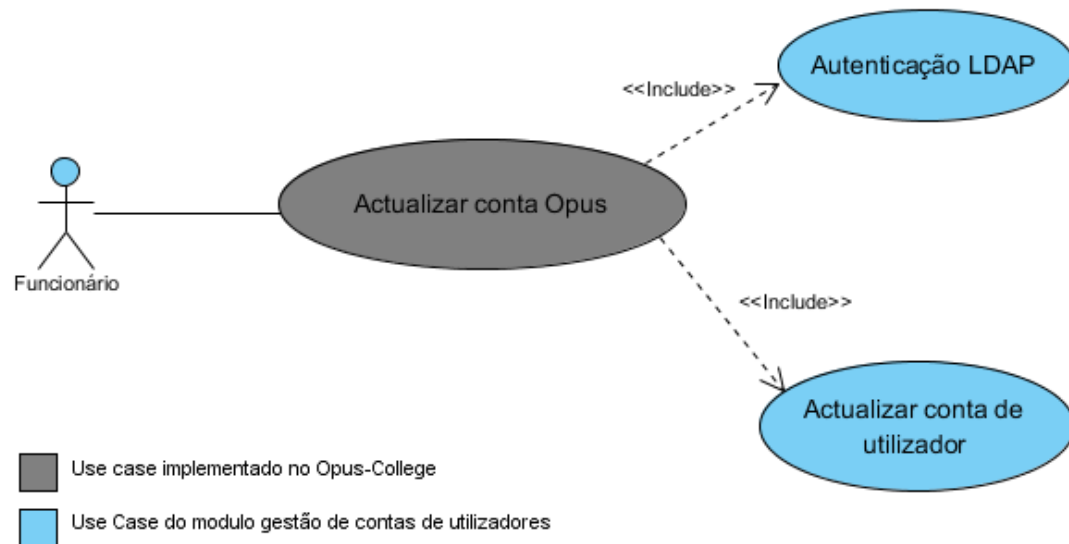
Descrição: O módulo integrado na aplicação opus-college deve possibilitar a actualização de contas de utilizadores a partir da alteração de dados relevantes no sistema Opus-College, e a partir da transição dos anos académicos.

Cenário 1: Transição do ano académico do estudante



Cenário 2: Actualização de dados relevantes para conta de utilizador

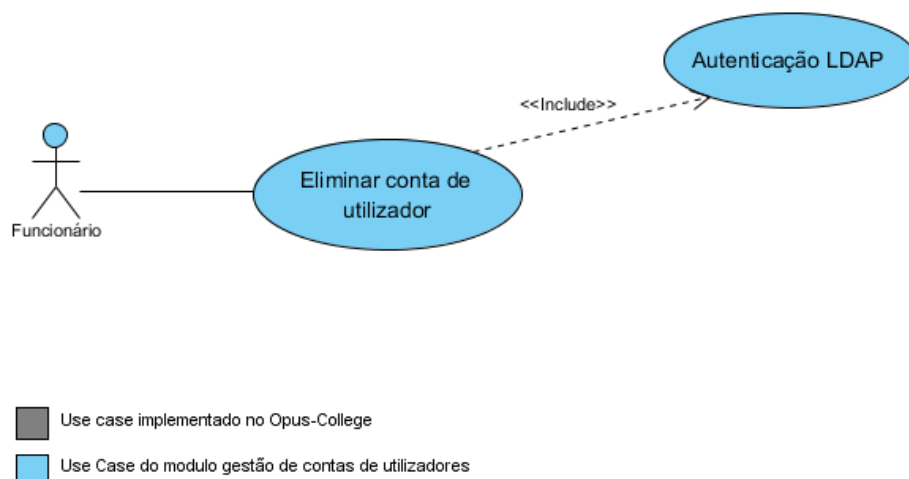
Diagrama:



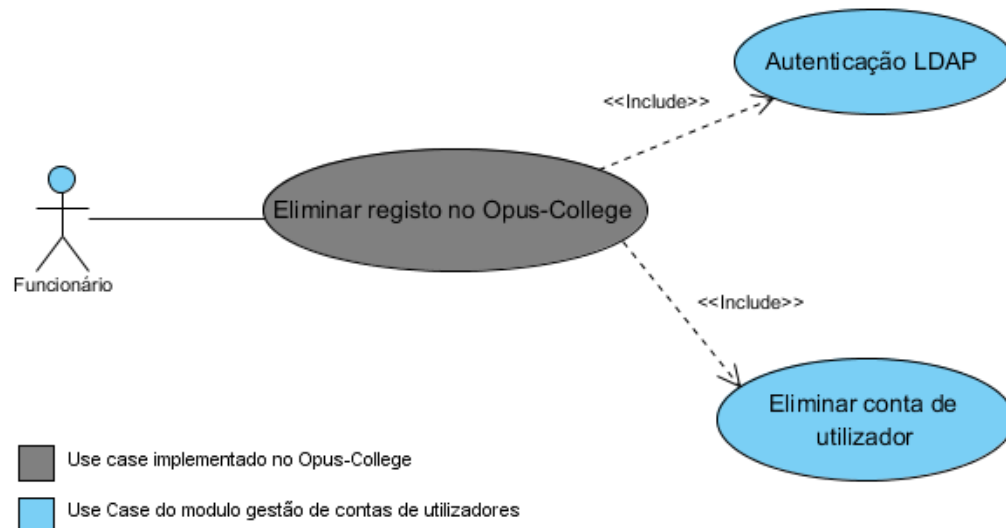
Caso de uso: Eliminação de contas de utilizadores

Descrição: O módulo integrado na aplicação opus-college deve possibilitar a eliminação de contas de utilizadores, em dois cenários, primeiro somente a eliminação da conta de utilizador e a eliminação do registo no opus e posterior eliminação da conta de utilizador.

Cenário 1: Eliminação da conta de utilizador sem eliminar o registo no opus-college.



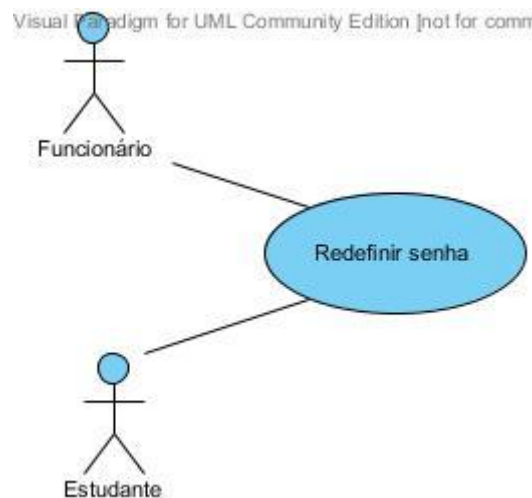
Cenário 2: Eliminação do registo no Opus-College e remoção implícita da conta de utilizador



Caso de uso: Recuperação de senhas

Descrição: A aplicação deve conter um mecanismo que possibilite a redefinição da senha do utilizador que eventualmente tenha perdido a senha actual, não obrigando o utilizador a entrar em contacto com o administrador de rede.

Diagrama:



c. Características dos utilizadores

A aplicação Opus-College será utilizada por um perfil de utilizadores de nível médio-alto, consequentemente este módulo será utilizado pelos mesmos.

Actores

Funcionário SA – Responsável por efectuar a utilização do sistema de gestão académica opus-college, onde são inseridos e matriculados os estudantes e funcionários do ISUTC.

Funcionário SecInf – Responsável por efectuar a gestão de contas de utilizadores, efectuando a execução das funcionalidades, actualização e remoção de contas de utilizadores.

Estudante e Funcionário no geral – Fazem uso do sistema de gestão escolar, e são responsáveis por activar o mecanismo para a recuperação de senhas de utilizadores, em caso de eventual perda da senha actual.

Nível de conforto com sistemas informáticos dos actores: Alto nível de conhecimento na óptica do utilizador.

d. Restrições

Tempo de execução – O tempo para a execução da primeira fase do projecto não possibilita a programação de todos os requisitos aqui descritos, posto isto, serão definidos os requisitos prioritários para este projecto final de curso.

4. Requisitos

a. Interfaces de *software*

Nome	Número de versão	Relação com o produto de <i>software</i> a ser desenvolvido
OpenLdap		Manter o único ponto de autenticação existente no ISUTC, a partir da introdução de entradas no <i>OpenLdap</i> com os atributos certos, formando a conta de utilizador.
Postgres		Entidades armazenadas neste sistema de gestão de base de dados.
SMTP		Envio de <i>email</i> para utilizador que esta a ser criado.
SSH		Comunicação com os servidores de <i>email</i> e de FTP.

b. Requisitos Funcionais

Requisito		Prioridade
RF01	Criação de conta de utilizador	Essencial
RF02	Actualização de contas de utilizador	Essencial
RF03	Eliminação de conta de utilizador	Importante
RF04	Recuperação de senhas de utilizadores	Essencial

Descrição estruturada dos requisitos funcionais

RF01 – Criação de conta de utilizador	
Pré-condição	O Funcionário SA é um utilizador valido no sistema.
Descrição	<ol style="list-style-type: none">1. O caso de uso começa quando o utilizador selecciona a opção registar estudante/ Ou registar funcionário.2. O Funcionário SA preenche os dados do estudante/funcionário, preenchendo o <i>email</i> usado pelo estudante/funcionário actualmente, caso tenha.3. O Funcionário SA matricula o estudante no curso requerido.4. Sistema cria entrada no LDAP.5. Sistema cria pasta no servidor de FTP.6. Sistema cria conta de <i>email</i>.7. O sistema deve enviar um <i>email</i> com os dados da conta criada.8. Caso o estudante não tenha uma conta de <i>email</i>, o sistema deve imprimir os dados.9. Sistema mostra tela de confirmação.
Pós-condição	A conta de utilizador deve ser criada

RF02 – Actualização de conta de utilizador	
Cenário 1: Transitar estudantes	
Pré-condição	O Funcionário é um utilizador valido no sistema.
Descrição	<ol style="list-style-type: none">1. O caso de uso começa quando o utilizador selecciona a tela transitar estudantes.2. O utilizador deve seleccionar os estudantes que deseja transitar.3. Utilizador clica em transitar estudantes.4. Sistema transita estudantes de um ano académico para outro.5. Sistema actualiza entrada no LDAP, identificando as contas com o ano académico actual.6. Sistema mostra tela de confirmação.
Pós-condição	Contas de utilizador actualizadas, referente ao ano académico do estudante.

RF02 – Actualização de conta de utilizador	
Cenário 2: Actualização de dados relevantes para conta de utilizador	
Pré-condição	O Funcionário é um utilizador valido no sistema.
Descrição	<ol style="list-style-type: none">1. O caso de uso começa quando o utilizador selecciona visualizar dados do estudante.2. O utilizador altera os dados.3. Sistema verifica se dados alterados são relevantes para a conta.4. Sistema efectua alteração de dados na entrada LDAP.5. Sistema efectua alteração de dados na conta de email.6. Sistema efectua alteração de dados na pasta FTP.7. Sistema mostra tela de confirmação.
Pós-condição	Conta de utilizador deve ser actualizada.

RF03 – Eliminação de conta de utilizador Eliminação da conta de utilizador sem eliminar o registo no opus-college.	
Pré-condição	O Funcionário SecInf é um utilizador valido no sistema.
Descrição	<ol style="list-style-type: none"> 1. O caso de uso começa quando o utilizador selecciona a opção eliminar conta. 2. O funcionário faz a busca das contas de utilizadores a eliminar a partir do critério escolhido (nome, turma). 3. O funcionário selecciona a conta ou as contas a eliminar. 4. Confirma a necessidade de eliminação de contas. 5. Sistema remove entrada no LDAP. 6. Sistema remove pasta e ficheiros dos respectivos utilizadores no servidor FTP. 6. Sistema remove a conta de <i>email</i>. 7. Sistema mostra tela de confirmação.
Pós-condição	Conta de utilizador deve ser removida.

RF03 – Eliminação de conta de utilizador Eliminação do registo no Opus-College e remoção implícita da conta de utilizador	
Pré-condição	O Funcionário SA é um utilizador valido no sistema.
Descrição	<ol style="list-style-type: none"> 1. O caso de uso começa quando o utilizador selecciona a opção eliminar registo no Opus-College. 2. Sistema requisita confirmação. 3. Sistema elimina registo no Opus-College. 4. Sistema remove entrada no LDAP. 5. Sistema remove pasta e ficheiros dos respectivos utilizadores no servidor FTP. 6. Sistema remove a conta de <i>email</i>. 7. Sistema mostra tela de confirmação.
Pós-condição	Conta de utilizador deve ser removida.

RF04 – Recuperação de senhas de utilizador	
Pré-condição	
Descrição	<ol style="list-style-type: none">1. O caso de uso começa quando o utilizador selecciona a opção recuperação de senha.2. Sistema requisita dados para a confirmação da identidade do utilizador.3. O utilizador disponibiliza os dados.4. Sistema valida os dados.5. Caso sejam validos, mostra tela para a redefinição da senha de utilizador.6. Utilizador introduz nova senha.6. Sistema efectua a alteração da senha, modificando a entrada no LDAP.7. Sistema mostra tela de confirmação.
Pós-condição	Senha de utilizador deve ser alterada.

c. Requisitos não funcionais

Requisito	Descrição
RNF01- Autenticação centralizada	A aplicação Opus-College deve permitir ao estudante efectuar a autenticação com a conta anteriormente criada, autenticando-se com a conta do directório LDAP.
RNF02- Níveis de autorização	A aplicação Opus-College deve carregar o nível de autorização dos utilizadores segundo os grupos definidos no LDAP.

5. Documentação

a. Manual de instalação

Deve ser definido um manual de instalação, que possibilite o uso do módulo desenvolvido.

Este manual deve ser escrito e detalhado com a linguagem mais simples possível, permitindo a instalação deste modulo por um técnico informático médio.

Desenvolvimento de um módulo para gestão de contas de utilizadores

b. Manual de uso

Deve ser definido um manual de uso para o módulo que será integrado a aplicação de gestão escolar opus-college.