

Escola Superior de Redes RNP



Projetos Especiais

Federação CAFe
Implantação do Provedor
de Identidade





**Federação
CAFe**
**Implantação
do Provedor
de Identidade**



Federação CAFé

Implantação do Provedor de Identidade

Edré Quintão Moreira
Éverton Didoné Foscarini
Gessy Caetano da Silva Junior
Lídia Aparecida O. Alixandrina
Lourival Pereira Vieira Neto
Silvana Rossetto

Rio de Janeiro
Escola Superior de Redes
2011

Copyright © 2011 – Rede Nacional de Ensino e Pesquisa – RNP
Rua Lauro Müller, 116 sala 1103
22290-906 Rio de Janeiro, RJ

Diretor Geral
Nelson Simões

Diretor de Serviços e Soluções
José Luiz Ribeiro Filho

Escola Superior de Redes

Coordenação
Luiz Coelho

Edição
Pedro Sangirardi

Supervisão Técnica
Silvana Rossetto

Equipe ESR (em ordem alfabética)
Alexandre César, Celia Maciel, Cristiane Oliveira, Derlinéa Miranda, Elimária Barbosa, Jacomo Piccolini, Lourdes Soncin, Luciana Batista, Luiz Carlos Lobato, Renato Duarte e Sérgio Souza

Capa, projeto visual e diagramação
Tecnodesign

Versão
1.1.0

Este material didático foi elaborado com fins educacionais. Solicitamos que qualquer erro encontrado ou dúvida com relação ao material ou seu uso seja enviado para a equipe de elaboração de conteúdo da Escola Superior de Redes, no e-mail info@esr.rnp.br. A Rede Nacional de Ensino e Pesquisa e os autores não assumem qualquer responsabilidade por eventuais danos ou perdas, a pessoas ou bens, originados do uso deste material.

As marcas registradas mencionadas neste material pertencem aos respectivos titulares.

Distribuição
Escola Superior de Redes
Rua Lauro Müller, 116 – sala 1103
22290-906 Rio de Janeiro, RJ
<http://esr.rnp.br>
info@esr.rnp.br

Dados Internacionais de Catalogação na Publicação (CIP)

Esta obra é distribuída sob a licença
Creative Commons: Atribuição e Uso Não-Comercial 2.5 Brasil



Grupos de Trabalho RNP (GT-RNP)

Desde 2002, o programa Grupos de Trabalho da RNP (GTs-RNP) promove a formação de parcerias entre a organização e grupos de pesquisa acadêmicos para o desenvolvimento e introdução de aplicações e serviços inovadores na rede operada pela RNP.

A formação de novos GTs é iniciada anualmente por uma Chamada de Propostas, cujas respostas são analisadas por representantes da RNP, da Sociedade Brasileira de Computação e do Laboratório Nacional de Redes de Computadores.

As atividades dos GTs são divididas em duas fases, cada uma com um ano de duração. Na fase 1, cada grupo desenvolve um protótipo para validar a aplicação proposta. Se bem-sucedido, o GT vai para a fase 2, na qual realiza um piloto com apoio de um pequeno grupo de instituições clientes da RNP.

Após estas fases, as propostas de novos serviços, que devem estar alinhadas aos objetivos estratégicos da organização, normalmente passam por uma etapa experimental antes de se transformarem em serviços da RNP. Nesta etapa são envolvidas instituições clientes da organização, que devem atender aos requisitos específicos de cada projeto e colaborar nos ajustes finais dos serviços. Isto é fundamental para agregar as contribuições de usuários finais aos futuros serviços em produção.

Exemplos do sucesso do programa são vários dos serviços hoje disponibilizados pela RNP, além de plataformas utilizadas para gerar produtos que atendem a própria organização, suas instituições clientes e parceiros estratégicos.

Um resultado de trabalhos desenvolvidos no contexto deste programa é a federação CAFé — Comunidade Acadêmica Federada — uma infraestrutura de autenticação e autorização interdomínios voltada para instituições de ensino e pesquisa brasileiras, que forma uma rede de confiança mútua entre elas. Na implantação dessa federação foram aplicadas soluções técnicas e ferramentas

desenvolvidas pelos Grupos de Trabalho de Diretórios (GT-Diretórios), do período 2002 a 2003, de Diretórios para Educação (GT-DIREDU), do período 2003 a 2004, e de Middleware (GT-Middleware), do período 2004 a 2005.

► www.rnp.br/pd/gt

Escola Superior de Redes

A Escola Superior de Redes (ESR) é a unidade da Rede Nacional de Ensino e Pesquisa (RNP) responsável pela disseminação do conhecimento em Tecnologias da Informação e Comunicação (TIC).

A ESR nasce com a proposta de ser a formadora e disseminadora de competências em TIC para o corpo técnico-administrativo das universidades federais, escolas técnicas e unidades federais de pesquisa. Sua missão fundamental é realizar a capacitação técnica do corpo funcional das organizações usuárias da RNP, para o exercício de competências aplicáveis ao uso eficaz e eficiente das TIC.

A ESR oferece dezenas de cursos distribuídos nas áreas temáticas: Administração e Projeto de Redes, Administração de Sistemas, Segurança, Mídias de Suporte à Colaboração Digital e Governança de TI.

A ESR também participa de diversos projetos de interesse público, como a elaboração e execução de planos de capacitação para formação de multiplicadores para projetos educacionais como: formação no uso da conferência web para a Universidade Aberta do Brasil (UAB), formação do suporte técnico de laboratórios do Proinfo e criação de um conjunto de cartilhas sobre redes sem fio para o programa Um Computador por Aluno (UCA).

A metodologia da ESR

A filosofia pedagógica e a metodologia que orienta a realização dos cursos da ESR é baseada na aprendizagem como construção do conhecimento por meio da resolução de problemas típicos da realidade do profissional em formação.

Os resultados obtidos em cursos de natureza teórico-prática são otimizados se o instrutor, auxiliado pelo material didático usado, atuar não apenas como expositor de conceitos e informações, mas principalmente como orientador do aluno na execução de atividades contextualizadas nas situações do cotidiano profissional.

A aprendizagem é entendida como a resposta do aluno ao desafio de situações-problema semelhantes às que são encontradas na prática profissional, que são superadas por meio de análise, síntese, julgamento, pensamento crítico e construção de hipóteses para a resolução do problema, em abordagem orientada ao desenvolvimento de competências.



Dessa forma, o instrutor tem participação ativa e dialógica como orientador do aluno para as atividades em laboratório. Até mesmo a apresentação da teoria no início da sessão de aprendizagem não é considerada uma simples exposição de conceitos e informações. O instrutor busca incentivar a participação dos alunos continuamente.

As sessões de aprendizagem onde se dão a apresentação dos conteúdos e a realização das atividades práticas têm formato presencial e essencialmente prático, utilizando técnicas de estudo dirigido individual, trabalho em equipe e práticas orientadas para o contexto de atuação do futuro especialista que se quer formar.

As sessões de aprendizagem desenvolvem-se em três etapas, com predominância de tempo para as atividades práticas, conforme descrição a seguir:

Primeira etapa: apresentação da teoria e esclarecimento de dúvidas (de 30 a 90 minutos). O instrutor apresenta, de maneira sintética, os conceitos teóricos correspondentes ao tema da sessão de aprendizagem, com auxílio de slides em formato PowerPoint. O instrutor levanta questões sobre o conteúdo dos slides em vez de apenas apresentá-los, convidando a turma à reflexão e participação. Isso evita que as apresentações sejam monótonas e que o aluno se coloque em posição de passividade, o que reduziria a aprendizagem.

Segunda etapa: atividades práticas de aprendizagem (de 60 a 120 minutos). Esta etapa é a essência dos cursos da ESR. A maioria das atividades dos cursos são assíncronas e feitas em duplas de alunos, que seguem o roteiro de atividades proposto na apostila, respeitando seu ritmo. Instrutor e monitor circulam entre as duplas para dirimir dúvidas e oferecer explicações complementares.

Terceira etapa: discussão das atividades realizadas (30 minutos). O instrutor comenta cada atividade, apresentando uma das soluções possíveis para resolvê-la, devendo ater-se às aquelas que geram maior dificuldade e polêmica. Os alunos são convidados a comentar as soluções encontradas e o instrutor retoma tópicos que tenham gerado dúvidas, estimulando a participação dos alunos. O instrutor sempre estimula os alunos a encontrar soluções alternativas às sugeridas por ele e pelos colegas e, caso existam, a comentá-las.

Sobre o curso

O curso foi desenvolvido para auxiliar as instituições no processo de implantação de um provedor de identidade para a Federação Acadêmica Federada (CAFé). O curso tem como objetivo demonstrar o funcionamento de uma infraestrutura de autenticação e autorização federada. Para isso, são apresentadas as ferramentas de software disponíveis para a construção desta infraestrutura, e o modo de integração de uma instituição acadêmica ou de pesquisa à federação CAFé.

A quem se destina

O curso se destina aos técnicos das instituições que pretendem aderir à Comunidade Acadêmica Federada (CAFé) e também aos interessados em saber mais sobre LDAP, esquema brEduPerson, gestão de identidade e Plataforma Shibboleth.

Permissões de uso

Todos os direitos reservados à RNP.

Agradecemos sempre citar esta fonte quando incluir parte deste livro em outra obra.

Exemplo de citação: MOREIRA, Edré Q.; FOSCARINI, Éverton D.; JUNIOR, Gessy C. da Silva; ALIXANDRINA, Lídia A. O.; NETO, Lourival P. V., ROSSETTO, Silvana.

Federação CAFé: Implantação do Provedor de Identidade. Rio de Janeiro: Escola Superior de Redes, RNP, 2011.

Comentários e perguntas

Para enviar comentários e perguntas sobre esta publicação:

Escola Superior de Redes RNP.

Endereço: Av. Lauro Müller 116 sala 1103 – Botafogo Rio de Janeiro – RJ – 22290-906.

E-mail: info@esr.rnp.br

Sobre os autores

Edré Quintão Moreira Bacharel e Mestre em Ciência da Computação pela Universidade Federal de Minas Gerais. Entre 2000 e 2003 participou da implantação do diretório corporativo da UFMG. Possui grande experiência em autenticação federativa com protocolo SAML, tendo atuado como assistente 1 no Grupo de Trabalho Middleware da RNP de 2003 a 2005. Possui grande experiência com a plataforma JEE, tendo se certificado em programação Java em 2001. Em 2009 participou do projeto que deu origem à Federação CAFé. Participou da elaboração e desenvolvimento do sistema EID. Atualmente é membro do Comitê Técnico da Federação CAFé e do Comitê Técnico de Gestão de Identidades da RNP. É também arquiteto de software no Departamento de Ciência da Computação da UFMG.

Éverton Didoné Foscarini Formado Bacharel em Ciência da Computação pela UFRGS, trabalhando como Analista de Suporte no CPD da UFRGS desde 2008. Tem seis anos de experiência como administrador de sistemas Linux, tendo trabalhado principalmente com virtualização de datacenter, servidores de diretório, e-mail, web e de aplicação. No escopo da Federação CAFé, ajudou a definir as metodologias de instalação dos softwares utilizados (Ubuntu, LDAP, Tomcat, Shibboleth etc), criando documentação e roteiros de instalação.



Gessy Caetano da Silva Junior Formado em Física pela Universidade Federal de Minas Gerais, atuando hoje como analista de sistemas para o Laboratório de Computação Científica LCC/CENAPAD da UFMG. Possui grande experiência com protocolo LDAP, administração de servidores Linux/Unix, backup e monitoramento de recursos de rede. Em 2009 participou do projeto que deu origem à Federação CAFe.

Lídia Aparecida O. Alixandrina Bacharel em Sistemas de Informação pela PUC-Minas. Atualmente é Analista de Sistemas na UFMG trabalhando na implantação de diretórios federados no projeto CAFe. Trabalha também no desenvolvimento das ferramentas EID (Export Import Directory), EID2LDAP e pCollecta. Experiência em autenticação federativa com Shibboleth, LDAP, Apache Tomcat, Banco de Dados e Java para Web.

Lourival Pereira Vieira Neto Engenheiro de Computação e Mestre em Informática pela PUC-Rio. Atualmente é consultor da Diretoria de Pesquisa e Desenvolvimento da RNP, membro do Comitê Técnico de Gestão de Identidade da RNP e membro-desenvolvedor da The NetBSD Foundation. Participou da execução e da coordenação do projeto e-AA (Infraestrutura de Autenticação e Autorização Eletrônica), projeto que foi responsável pelo desenvolvimento e implantação da federação CAFe.

Silvana Rossetto Graduou-se em Ciência da Computação na Universidade Federal do Espírito Santo (UFES), em 1998. Cursou o Mestrado em Informática no Programa de Pós-Graduação em Informática da UFES, de 1999 a 2001. Concluiu o Doutorado em Informática pela Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio) em 2006, na área de Sistemas Distribuídos. Realizou o programa de Doutorado Sanduíche no Exterior, entre 2004 e 2005, no Dipartimento di Elettronica e Informazione da Politecnico di Milano. De fevereiro de 2007 a julho de 2009 ocupou o cargo de Professor Adjunto no Departamento de Ciência e Tecnologia da Universidade Federal Fluminense (UFF). Desde agosto de 2009 ocupa o cargo de Professor Adjunto no Departamento de Ciência da Computação, da Universidade Federal do Rio de Janeiro (UFRJ). Nessa universidade, exerce atividades de ensino no Departamento de Ciência da Computação, integra o grupo de pesquisa na área de Redes de Computadores e Sistemas Distribuídos e participa do Programa de Pós-Graduação em Informática (PPGI/UFRJ).



► Sumário

Capítulo 1

Introdução à Federação CAFé	1
Infraestrutura de autenticação e autorização federada	1
O que é uma Federação?	3
Elementos de uma federação	5
Componente adicional de uma federação	6
Provedores de identidade	7
Provedores de serviço	7
Interação entre os elementos de uma federação	8
Exemplos de federações acadêmicas	11
Federação CAFé	11
Visão geral do curso	13

Roteiro de Atividades 1

Introdução à Federação CAFé	15
Atividade 1 – Demonstrar o funcionamento de uma federação	16

Capítulo 2

Revisão de LDAP e esquema brEduPerson	17
Revisão de serviço de diretório e LDAP	17
Serviço de diretório	17
LDAP	18
OpenLDAP	19
Definições LDAP	20
Representação LDIF	31
Comandos de shell e ferramenta gráfica	34
Esquema brEduPerson	38
Modelo de nomes para uso na Federação CAFé	39

Roteiro de Atividades 2

Revisão de LDAP e esquema brEduPerson	43
Atividade 1 – Instalar e configurar um serviço de diretório OpenLDAP	44
Atividade 2 – Editar o arquivo LDIF e executar alterações no diretório	46
Atividade 3 – Utilização de ferramenta gráfica para acesso ao servidor LDAP	48



Capítulo 3

Construindo metadiretórios com EID	49
Visão geral do EID	49
Motivação	49
Metadiretório	50
EID	51
EID e brEduPerson	53
Acesso ao EID	54
Configurações iniciais	56
Definição de classes	57
Configuração de extrações	59
Definição de repositórios	59
Extrações	62
Processos	68
Agendamentos	70

Roteiro de Atividades 3

Construindo metadiretórios com EID	75
Atividade 1 – Instalação do EID e EID2LDAP	76
Atividade 2 – Configuração de um repositório	79
Atividade 3 – Definição de uma extração	79
Atividade 4 – Definição de um processo e seu agendamento	80
Atividade 5 – Limpar o repositório EID	81
Atividade 6 – Reagendar o processo de carga da classe <i>Identificação</i>	82

Capítulo 4

Criando extrações no EID	83
Extração de arquivos texto	83
Resolução de objetos	86
Parâmetros globais	87
Importação incremental	88
Script de conversão	89
Script de conversão – Bean Shell	90
Script de conversão – Java Nativo	91
Algoritmos de unificação	91
Web services	93
Problemas comuns	94

Roteiro de Atividades 4

Criando extrações no EID	95
Atividade 1 – Definição de uma extração de arquivo texto	96
Atividade 2 – Definição de extração para a classe <i>Aluno</i>	98
Atividade 3 – Transformação do campo Sexo	100
Atividade 4 – Importação de login e senha	101
Atividade 5 – Importação Incremental	103



Capítulo 5	
Gestão de pessoas e grupos no EID	105
Gestão manual de pessoas	105
Conciliação de registros	105
Inserção de novas pessoas	109
Alteração de dados via interface	110
Gestão de grupos	113
Inserção, atualização e remoção de grupos	113
Roteiro de Atividades 5	
Gestão de pessoas e grupos no EID	115
Atividade 1 – Conciliação de um registro manualmente	116
Atividade 2 – Registros pendentes para conciliação	116
Atividade 3 – Inserção de uma nova pessoa	117
Atividade 4 – Definição de um grupo	117
Capítulo 6	
Alimentação de diretórios com EID2LDAP	119
Introdução	119
EID2LDAP	119
Características	120
XML do EID	121
XSLT	122
Processamento do LDIF	124
Configuração e uso	124
Acesso a aplicação EID2LDAP	125
Configuração de exportação	127
Inicialização do agente	128
Cadastramento dos servidores	129
Cadastramento do XSLT	131
Definição de agendamento	133
Verificação do log	135
Problemas comuns	137
Roteiro de Atividades 6	
Alimentação de diretórios com EID2LDAP	139
Atividade 1 – Inicialização do agente	140
Atividade 2 – Configuração do servidor LDAP	140
Atividade 3 – Configuração de uma transformação	140
Atividade 4 – Executar teste padrão: leitura no diretório	141
Atividade 5 – Definição de um agendamento	141
Atividade 6 – Desativação e alteração de registros no metadiretório	142

Capítulo 7

Plataforma Shibboleth	145
Introdução	145
Provedor de Identidade (IdP)	147
Provedor de Serviço (SP)	149
WAYF / DS	151
Metadata	151
Funcionamento	152

Roteiro de Atividades 7

Plataforma Shibboleth	165
Atividade 1 – Instalar e configurar o provedor de identidade Shibboleth	166

Capítulo 8

Provedor de identidade na plataforma Shibboleth	177
Principais pontos de configuração	177
Configuração do Apache	178
Configuração do Tomcat	178
Configuração do CAS	178
Configuração do Shibboleth IdP	179

Roteiro de Atividades 8

Provedor de identidade na plataforma Shibboleth	181
Atividade 1 – Validando a instalação e testando a Federação	182

Capítulo 9

Implantação de provedor de identidade a partir de bases de dados relacionais	185
Roteiro de implantação de um provedor de identidade	185
Metodologia adotada	185
Roteiro de atividades	186
Instalar o servidor básico padrão	186
Extrair dados para o metadiretório	187
Instalar o provedor de identidade	188
Entrar na Federação CAFé	189

Roteiro de Atividades 9

Implantação de provedor de identidade a partir de bases de dados relacionais	191
Atividade 1 – Demonstrar o funcionamento da autenticação e envio de atributos	192

Capítulo 10

Implantação de provedor de identidade a partir de um diretório existente	193
Introdução ao Shibboleth-IdP	193
Origem dos dados	194
Análise do cenário	194
Atributos recomendados pela federação	195



Atributos do esquema original	196
Definição dos mapeamentos	198
Renamear atributo	198
Alterar valor de atributo	200
Modificar sequência de atributos	201
Roteiro de Atividades 10	
Implantação de provedor de identidade a partir de um diretório existente	203
Atividade 1 – Renomeando um atributo	204
Atividade 2 – Alterando o valor de um atributo	204
Atividade 3 – Múltiplos atributos	204
Bibliografia	205
Grade curricular da Escola Superior de Redes	206





xiiv

1

Introdução à Federação CAFé

- Infraestrutura de autenticação e autorização federada
- Elementos de uma federação
- Interação entre os elementos de uma federação
- Exemplos de federações acadêmicas
- Federação CAFé
- Visão geral do curso

Infraestrutura de autenticação e autorização federada

- Motivação
 - Disseminação de tecnologias e ferramentas que estimulam o compartilhamento de recursos, informações e serviços inter-institucionais
- Desafio para as instituições
 - Desenvolver ambientes seguros e escaláveis para permitir que a colaboração visionada aconteça de fato

Este curso foi desenvolvido no escopo do projeto e-AA: *Infraestrutura de Autenticação e Autorização Eletrônica*, idealizado e coordenado pela RNP, com a colaboração das instituições: Cefet-MG, UFC, UFF, UFMG e UFRGS. O projeto teve início em julho de 2007 e sua meta principal é criar as condições necessárias para a implantação de uma Federação Acadêmica no Brasil. Uma federação acadêmica envolve instituições de ensino e pesquisa e permite que as pessoas vinculadas a essas instituições compartilhem informações e recursos e tenham acesso a serviços restritos, usando o vínculo institucional como critério básico para esse compartilhamento.



A finalidade deste curso é capacitar o pessoal de TI das instituições de ensino e pesquisa no Brasil para implantar e gerenciar em suas instituições um Provedor de Identidade (componente que mantém e gerencia as informações sobre as pessoas vinculadas a uma instituição) e acoplá-lo à Federação CAFé (Comunidade Acadêmica Federada), criada no escopo do projeto e-AA.

Ao longo do curso serão revisados os conceitos básicos de serviço de diretórios e do protocolo de acesso leve a serviço de diretórios LDAP. Será apresentado o esquema brEduPerson, que define atributos e classes necessários para armazenar informações específicas sobre pessoas e seus vínculos em instituições brasileiras. Juntamente com o esquema brEduPerson, serão apresentados os modelos de informação e de nomes propostos para a organização das informações sobre pessoas em um diretório institucional, o qual servirá de base para a implantação do provedor de identidade em uma instituição.

Na sequência de estudo serão apresentadas as ferramentas de auxílio EID e EID2LDAP, que facilitam o processo de extração de dados de pessoas de bases relacionais e a inclusão desses dados em um diretório LDAP.

A parte central do curso incluirá os passos necessários para implantar um provedor de identidade institucional usando a plataforma Shibboleth e o serviço de diretório LDAP, e o modo de acoplar esse provedor de identidade à Federação CAFé.

- Exemplos de serviços internos:
 - ▀ Cadastro de projetos, matrícula de alunos, registro de notas, compartilhamento de documentos etc.
- Exemplos de serviços externos:
 - ▀ Acesso a bibliotecas digitais, compartilhamento de recursos (ciclos de CPU, espaço de armazenamento), ensino a distância etc.
- Uma federação oferece para as instituições a infraestrutura de autenticação e autorização necessária para interconectar pessoas e compartilhar recursos, informações e serviços

Nesta primeira sessão do curso introduziremos o conceito de federação acadêmica, discutindo os seguintes tópicos:

- Demandas para a implantação de uma infraestrutura de autenticação e autorização inter-institucional;
- Conceito de federação, seus elementos principais e sua arquitetura básica;
- Forma como está sendo projetada a federação acadêmica brasileira CAFé.

Ao final da sessão será apresentada uma visão geral do curso, detalhando os temas que serão abordados em cada uma das sessões seguintes.

O que é uma Federação?

- Tipo de rede de confiança que permite reduzir contratos bilaterais entre usuários e provedores de serviços
- Implementa o princípio de identidade federada:
 - Instituições implementam métodos distintos de autenticação, mantendo a interoperatividade

O crescente avanço das tecnologias de redes de computadores (em particular da internet) e o uso dessas tecnologias para a construção de aplicações que permitem o acesso remoto (e em tempo real) a diferentes serviços, trouxe a necessidade de se criar e manter bases de dados com informações sobre as pessoas que podem acessar esses serviços e definir o nível de privilégio. Essa demanda de reconhecimento e validação de acesso dos usuários aos serviços pode ser sintetizada em duas etapas denominadas **autenticação** e **autorização**.

O cumprimento das etapas de autenticação e autorização como etapas fundamentais para a disponibilização de um serviço implica, normalmente, na necessidade de manutenção de bases de dados com registros sobre os possíveis usuários do serviço. A demanda do lado de quem disponibiliza um serviço é a necessidade de criar e manter suas próprias bases de dados de usuários. Do outro lado, para quem usa os diferentes serviços disponibilizados, a demanda é a necessidade de criar e manter contas (ou cadastros) para cada serviço a que se deseja ter acesso.

O conceito de **federação acadêmica** visa minimizar as demandas dos provedores e dos usuários de serviços disponibilizados por instituições de ensino e pesquisa no que diz respeito à manutenção de informações usadas para autenticação e autorização de acesso a esses serviços. A ideia básica consiste no seguinte: as informações sobre uma pessoa são mantidas em uma única base, gerida por sua instituição de vínculo, cabendo a cada instituição estabelecer seu modelo de **gestão de identidade**, isto é, de que forma informações sobre pessoas são mantidas e atualizadas e quais métodos de autenticação são usados. Os provedores de serviço confiam no modelo de gestão de identidade das instituições e disponibilizam seus serviços para os usuários vinculados a essas instituições, criando assim o princípio de **identidade federada**.



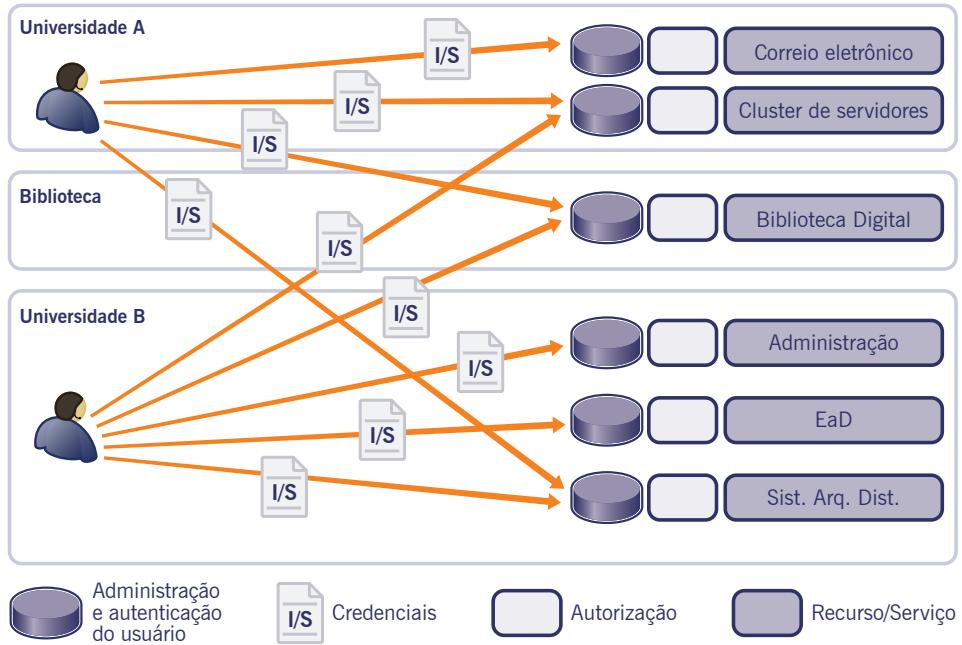
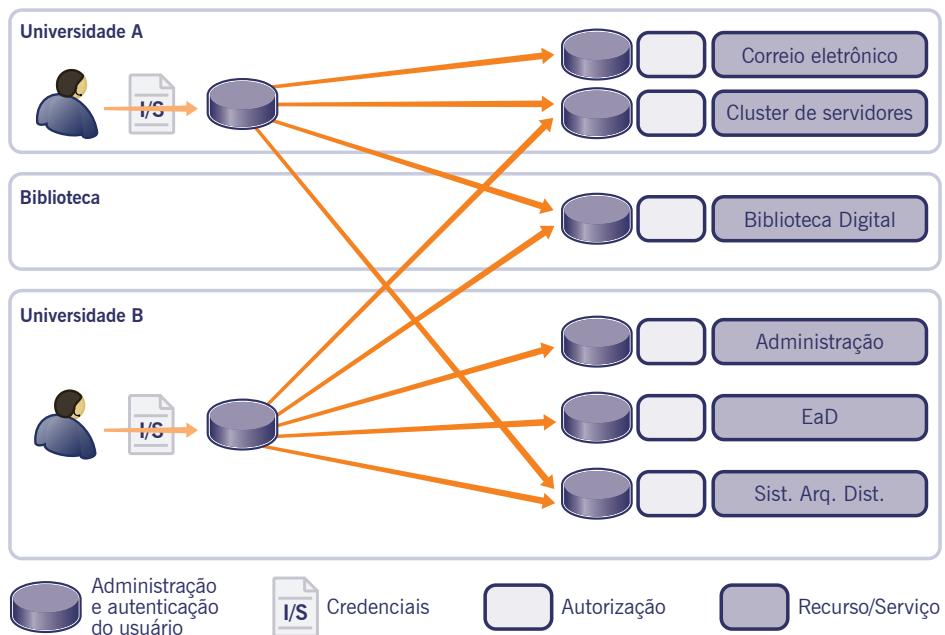


Figura 1.1
Cada serviço mantém informações sobre seus usuários.

As figuras 1.1. e 1.2 (Fonte: SWITCH AAI-Federation) ilustram a diferença entre um modelo usual, onde cada serviço deve manter informações sobre seus possíveis usuários, e um modelo onde as informações sobre os usuários são concentradas e mantidas em um único local.

No primeiro caso, a implementação de cada serviço deve prever um módulo adicional para tratar o registro dos usuários que podem acessá-lo, e cada pessoa precisa ter um cadastro (login/senha) para cada serviço que deseja acessar. No segundo caso, as informações sobre as pessoas são mantidas em um único local, tipicamente a instituição com a qual a pessoa mantém seu vínculo principal, e cada pessoa precisa ter apenas um registro (login/senha); nesse caso, a implementação dos serviços oferecidos não requer o módulo de registro de usuários.

Figura 1.2
Informações sobre os usuários concentradas em local único.



Elementos de uma federação

- Uma federação inclui dois elementos:
 - Provedor de Identidade (IdP)
 - Provedor de Serviço (SP)
- Atores em uma federação:
 - Usuário: deseja usar um recurso protegido
 - Provedor do recurso: aplicação com um SP instalado
 - Instituição do usuário: possui um IdP e um processo interno de autenticação

Uma federação é constituída de dois componentes principais:

- Provedores de identidade – Armazenam e gerenciam as informações sobre pessoas.
- Provedores de serviço – Oferecem serviços restritos para grupos de usuários.

Na arquitetura de uma federação, três atores podem ser distinguidos:

- Usuário – Pessoa vinculada a uma instituição e que deseja acessar um recurso protegido;
- Provedor do recurso – Aplicação associada ao componente provedor de serviço;
- Instituição do usuário – Instituição que mantém o componente provedor de identidade e estabelece um processo interno de autenticação das pessoas vinculadas a ela.

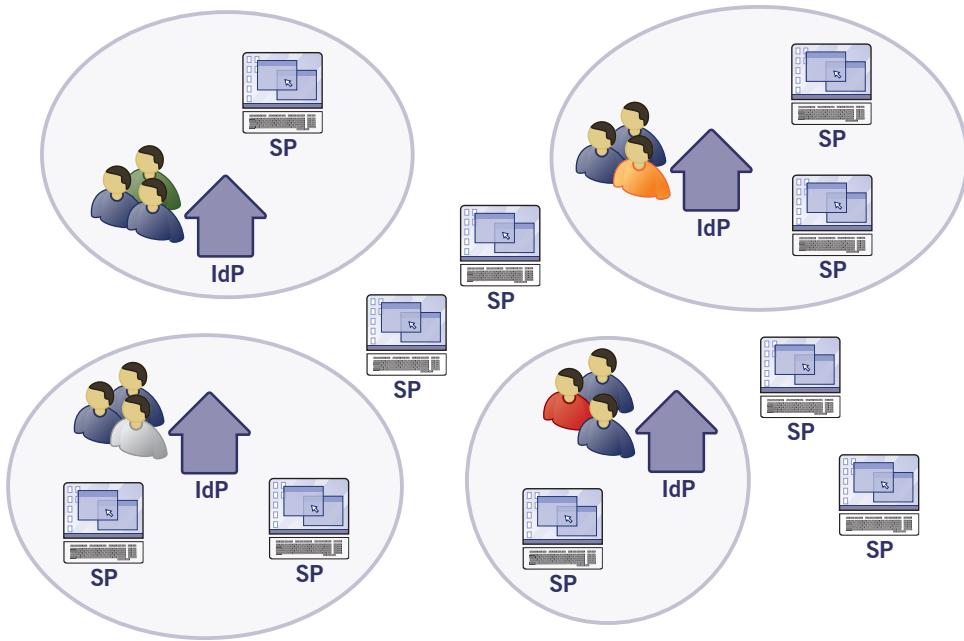


Figura 1.3
Principais componentes de uma federação.

A figura 1.3 apresenta os principais componentes de uma federação e as associações entre eles. Podemos observar que dentro de uma federação é possível definir subgrupos com um provedor de identidade e um ou mais provedores de serviços associados. Essa configuração pode ser usada para os seguintes casos:

- ▲ Serviços internos da instituição, como matrícula de alunos, registro de notas, cadastro de projetos, entre outros exemplos.
- ▲ Serviços externos à instituição, como o caso de bibliotecas digitais, ensino a distância, armazenamento distribuído, entre outros exemplos, podendo ser oferecidos a usuários ligados a diferentes provedores de identidade.

Componente adicional de uma federação

- ▲ Where Are You From (WAYF) / Discovery Service (DS)
 - ▲ Elemento que centraliza as informações sobre provedores de identidade de uma federação

Como um provedor de serviço em uma federação normalmente permite o acesso de usuários de diferentes instituições, um componente adicional é incluído na federação para auxiliar no redirecionamento dos usuários para os seus respectivos provedores de identidade. Esse componente, denominado *Where Are You From* (WAYF), ou *Discovery Service* (DS) a partir do Shibboleth 2.x centraliza as informações sobre os provedores de identidade da federação e suas localizações. Ao ser redirecionado para o WAYF ou DS, o usuário seleciona a sua instituição de origem, e, em seguida, passa a interagir com o seu provedor de identidade para fornecer as suas credenciais.



Provedores de identidade

- Implementam a política interna de gestão de identidade de uma instituição
 - ▲ Atributos dos usuários
 - ▲ Nome, data do vínculo, cargo ocupado, matrícula etc.¹
 - ▲ Método de autenticação
 - ▲ Login/senha, certificados etc.
 - ▲ Identificador único para cada pessoa vinculada à instituição

Os **provedores de identidade** são responsáveis por manter as informações sobre as pessoas vinculadas a uma instituição, incluindo dados pessoais (nome, data de nascimento, CPF, nomes dos pais, sexo, data de nascimento etc.) e vínculos internos (data de admissão, cargo ocupado, número de matrícula, número VoIP etc.). O provedor de identidade estabelece seu método de autenticação interno e deve garantir que cada pessoa da instituição tenha um identificador único.

Provedores de serviço

- Implementam serviços que devem ser disponibilizados para pessoas vinculadas às instituições. Requerem:
 - ▲ Autenticação:
 - ▲ Identificação dos usuários do serviço
 - ▲ Autorização:
 - ▲ Atributos adicionais do usuário que garantem certos privilégios de acesso
- Foco na implementação do serviço, e não na manutenção dos registros dos usuários

Os **provedores de serviço** oferecem serviços de acesso restrito, podendo requisitar ainda privilégios de acesso baseados em informações adicionais sobre os usuários (por exemplo, aluno matriculado em determinado curso, professor coordenador de curso etc.). Na implementação do serviço são definidos os privilégios de acesso e as informações adicionais que serão solicitadas. Não cabe ao provedor de serviço manter essas informações, mas apenas solicitá-las aos provedores de identidade.

Interação entre os elementos de uma federação

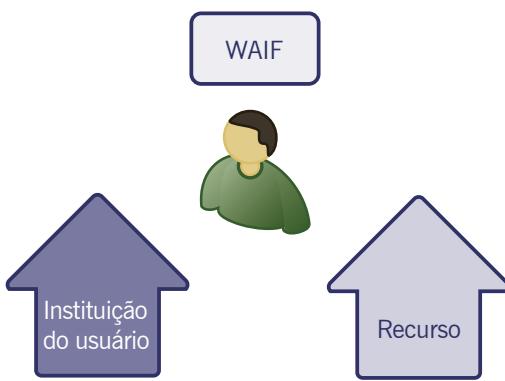


Figura 1.4
Interação entre
elementos de
uma federação.

A interação entre os elementos (atores) de uma federação (figuras 1.4 a 1.8) segue os seguintes passos:

- ▲ Passo 1: usuário faz acesso ao provedor de serviço (SP).
- ▲ Passo 2: o serviço apresenta escolhas fornecidas pelo repositório centralizado WAYF (Where Are You From).
- ▲ Passo 3: o usuário seleciona a sua instituição de origem.
- ▲ Passo 4: o usuário é redirecionado para o seu provedor de identidade (IdP).
- ▲ Passo 5: o IdP autentica o usuário com o método escolhido pela instituição.
- ▲ Passo 6: o SP recebe garantia de autenticação do usuário pelo IdP.
- ▲ Passo 7: se necessário, o SP requisita atributos adicionais desse usuário ao IdP; para garantir a privacidade do usuário, apenas são disponibilizados atributos previamente acordados entre o IdP e o SP.
- ▲ Passo 8: o provedor de serviço decide sobre as autorizações e disponibiliza o serviço para o usuário.

Figura 1.5
Interação entre
elementos de
uma federação.

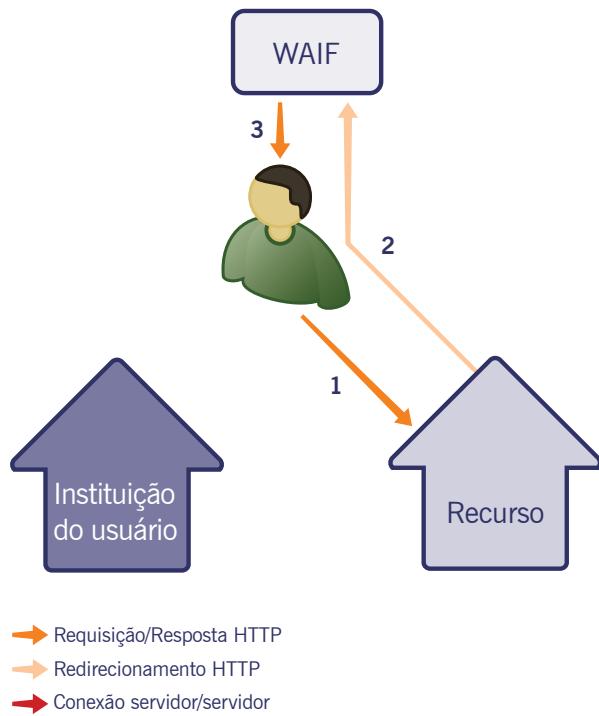
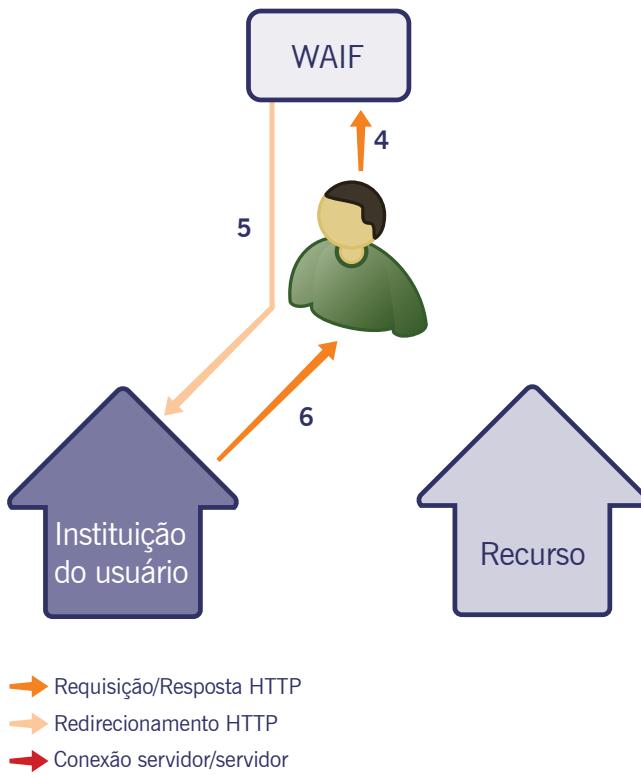


Figura 1.6
Interação entre
elementos de
uma federação.



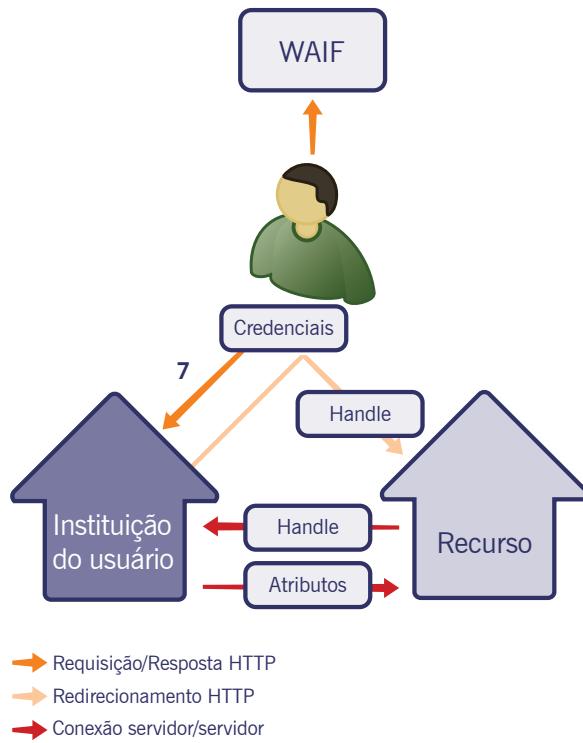


Figura 1.7
Interação entre
elementos de
uma federação.

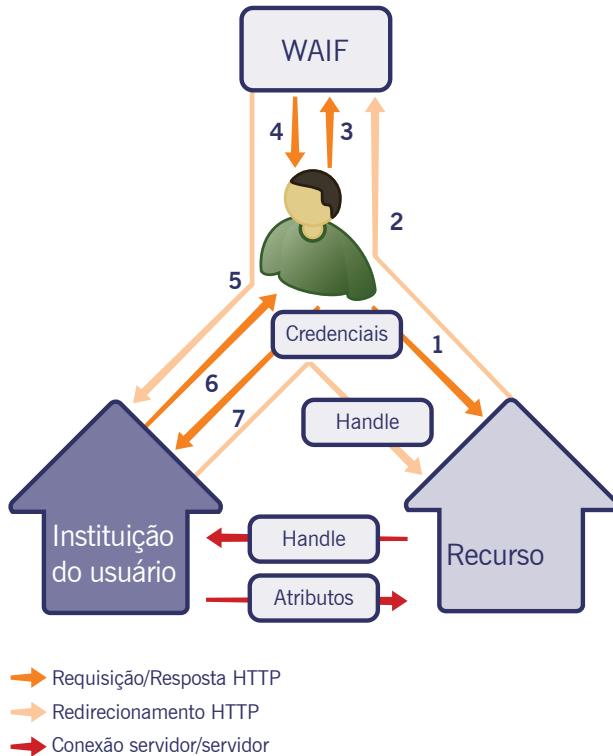


Figura 1.8
Interação entre
elementos de
uma federação.

Fonte das figuras: SWITCH AAI-Federation.

Exemplos de federações acadêmicas

- InCommon
 - Federação nos EUA com 107 instituições e dois milhões de usuários
- Feide
 - Federação na Noruega
- Switch
 - Federação na Suíça
- SDSS
 - Federação no Reino Unido

Federações acadêmicas já são implementadas e mantidas em outros países. Alguns exemplos: InCommon, Feide, Switch e SDSS. Uma tendência natural para o futuro será a junção de federações em **confederações**, ampliando o escopo de serviços disponibilizados aos usuários e o número de possíveis usuários de um serviço para além dos limites geográficos dos países.

Federação CAFé

- Iniciativa da RNP para criar uma Federação Acadêmica no Brasil
 - Projeto iniciado em julho de 2007 envolvendo cinco instituições: UFC, UFMG, UFF, UFRGS e Cefet-MG
- Metodologia adotada:
 - Integrar padrões e soluções de software utilizadas por outras federações
 - Desenvolver ferramentas auxiliares e definir políticas para a federação

No Brasil, os primeiros esforços para a construção de uma federação acadêmica estão resultando na criação da Federação CAFé (Comunidade Acadêmica Federada), cuja meta é congregar todas as universidades e instituições de pesquisa brasileiras. A metodologia adotada para a construção da infraestrutura básica da federação consiste da utilização de padrões e soluções de software já disponíveis e adotadas por outras federações, e da implementação e experimentação de ferramentas auxiliares para apoiar a implantação dos provedores de identidade e de serviço. O projeto de criação da Federação CAFé inclui ainda o estudo, a proposição, a análise e a validação de políticas para regular o funcionamento da federação (requisitos mínimos que provedores de identidade e de serviço deverão cumprir).



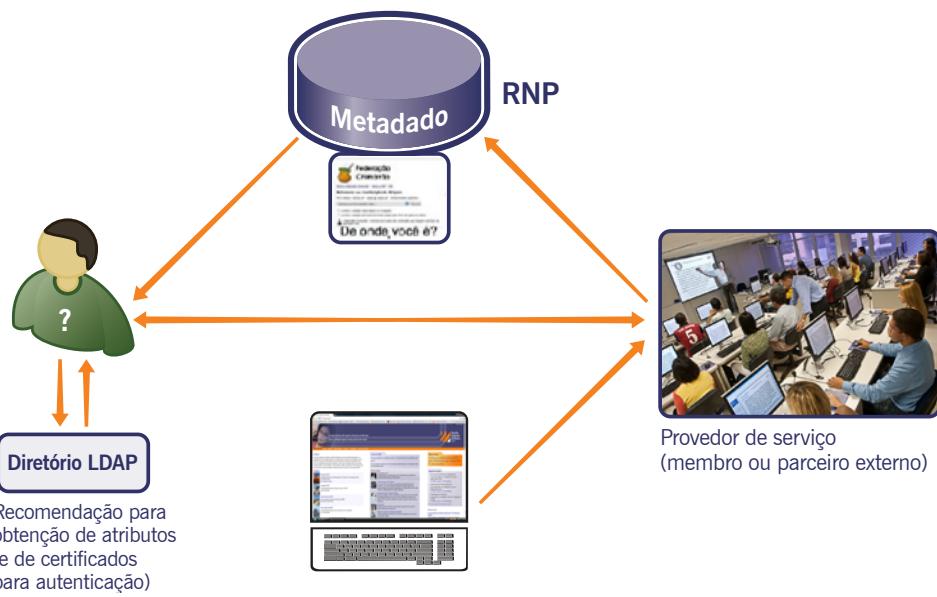


Figura 1.9
Arquitetura
básica da CAFE.

A figura 1.9 mostra a arquitetura básica proposta para a federação CAFE. Inicialmente, o componente WAYF será centralizado e mantido pela RNP. Os provedores de serviço poderão ser implantados nas próprias instituições que compõem a federação (universidades e instituições de pesquisa) ou poderão ser implantados por membros externos (os quais atuam apenas como provedores de serviço).

As políticas definidas para a operação da Federação CAFE deverão estabelecer os critérios para a inclusão de um membro na federação e as obrigações dos provedores de identidade e de serviço, bem como garantir a preservação dos requisitos básicos de privacidade. A recomendação para os provedores de identidade é a utilização de serviço de diretórios para a organização das informações sobre as pessoas vinculadas à instituição.

- Atividades desenvolvidas
 - Definição do esquema brEduPerson
 - Implementação de ferramentas auxiliares e roteiros de implantação
- Atividades em andamento
 - Material de capacitação para desenvolvimento de aplicações federadas (provedores de serviço)
 - Integração de novas instituições à Federação

Visão geral do curso

- Sessão 1: Visão geral sobre a motivação e a metodologia para a construção de uma federação acadêmica no Brasil
- Sessão 2: Revisão de LDAP e esquema brEduPerson
- Sessão 3: Construção de metadiretórios com EID
- Sessão 4: Criação de extrações com EID
- Sessão 5: Gestão de pessoas e grupos no EID
- Sessão 6: Alimentação de diretórios LDAP
- Sessão 7: Provedores de identidade e de serviço na plataforma Shibboleth
- Sessão 8: Configuração de um provedor de identidade na plataforma Shibboleth
- Sessão 9: Implantação de um provedor de identidade a partir de bases de dados relacionais
- Sessão 10: Implantação de um provedor de identidade a partir de um diretório existente

Este curso está organizado em 10 sessões de aprendizagem.

A sessão 1 apresentou uma visão geral sobre a motivação e a metodologia adotadas para a construção de uma federação acadêmica no Brasil.

A sessão 2 fará uma revisão sobre serviço de diretórios e protocolo LDAP e apresentará o esquema brEduPerson, definido para uso com a federação CAFe.

A sessão 3 apresentará a ferramenta EID, uma aplicação web cuja finalidade é auxiliar no processo de migração de dados das bases relacionais de uma instituição para um diretório.

As sessões 4 e 5, também dedicadas à ferramenta EID, mostrarão como configurar e executar as extrações das bases de dados relacionais para o metadiretório definido pelo EID.

A sessão 6 apresentará a ferramenta EID2LDAP, cuja finalidade é levar os dados contidos no metadiretório do EID para o diretório LDAP.

A sessão 7 introduzirá o estudo sobre a plataforma Shibboleth, a solução de software adotada pela federação CAFe para a implementação dos provedores de identidade e de serviço.

A sessão 8 focará no estudo sobre a configuração de um provedor de identidade na plataforma Shibboleth.



As sessões 9 e 10 serão dedicadas à realização de dois casos de uso:

- ▲ Implantação completa de um provedor de identidade a partir de bases de dados relacionais;
- ▲ Implantação completa de um provedor de identidade a partir de um diretório institucional sem o esquema brEduPerson. Nos dois experimentos a ideia é construir uma federação piloto dentro do laboratório.

1

Roteiro de Atividades Introdução à Federação CAFe

Tópicos e conceitos

- Autenticação e autorização federada
- Elementos de uma federação
 - ▲ Provedores de identidade
 - ▲ Provedores de serviço
 - ▲ Where Are You From (WAYF)
- Arquitetura básica
- Exemplos de federações acadêmicas
- Federação CAFe
- Visão geral do curso e seus objetivos.

Competências técnicas desenvolvidas

- Compreender o funcionamento de uma infraestrutura de autenticação e autorização federada.

Tempo previsto para as atividades

- 30 minutos

Servidor de sala de aula

- Provedor de Serviço e Provedor de Identidade configurados



Atividade 1 – Demonstrar o funcionamento de uma federação

Acesse serviços provados na máquina do instrutor:

1. Abra um browser e acesse a URL indicada pelo instrutor;
2. Escolha o provedor de identidade;
3. Informe as credenciais de identificação;
4. Acesse o serviço disponibilizado.



2

Revisão de LDAP e esquema brEduPerson

- Revisão de serviço de diretório e LDAP
 - Serviço de diretório, LDAP e OpenLDAP
 - Modelos LDAP
 - Representação LDIF
 - Comandos de shell e ferramenta gráfica
- Esquema brEduPerson
 - Modelos de informação e de nomes propostos para utilização com o esquema brEduPerson

Revisão de serviço de diretório e LDAP

Serviço de diretório

- Banco de dados especializado para localizar, gerenciar, administrar e organizar objetos e recursos de rede;
- Unificação de informações de pessoas e serviços;
- Banco de informações distribuídas;
- Mecanismo de busca flexível;
- Espaço de nomes homogêneo;
- Serviço padronizado.

Nesta sessão serão revisados os conceitos gerais sobre diretório com o uso do protocolo LDAP e a utilização do esquema brEduPerson para criar um modelo de dados mais adequado para as instituições brasileiras.

Um diretório é uma lista de informações sobre objetos arranjados em uma ordem que fornece detalhes sobre cada objeto. Exemplos comuns são listas telefônicas e



catálogos de livros. Para a lista telefônica, os objetos listados são pessoas. Os nomes são organizados em ordem alfabética e endereço e número de telefone são os detalhes fornecidos sobre cada pessoa.

Em termos computacionais, um diretório é um banco de dados especializado, também chamado de repositório de informação, guardando informações ordenadas e de tipo definido sobre objetos. Uma característica especial dos diretórios é que eles são acessados (lidos ou pesquisados) muito mais frequentemente do que atualizados (escritos). Como diretórios devem ser capazes de suportar grandes volumes de requisições de leitura, são tipicamente otimizados para acessos de leitura. O acesso de escrita deve ser limitado a administradores de sistema ou ao proprietário de cada parte da informação. Um banco de dados relacional, por outro lado, precisa suportar aplicações, como aplicações bancárias e de reservas aéreas, relativamente com grandes volumes de atualização.

Diretórios permitem que usuários ou aplicações encontrem recursos que tenham características necessárias para uma tarefa em particular. Por exemplo, um diretório de usuários pode ser utilizado para procurar um endereço de e-mail ou número de fax.

Os termos “páginas brancas” e “páginas amarelas” algumas vezes são utilizados para descrever o modo como um diretório é usado. Se o nome de um objeto (pessoa, impressora etc) é conhecido, suas características (número de telefone, páginas por minuto) podem ser encontradas, em processo similar a procurar um nome nas páginas brancas de uma lista telefônica. Se o nome de um objeto é desconhecido, o diretório pode ser pesquisado por uma lista de objetos que possuem certas características. Diretórios guardados em um computador são muito mais flexíveis que uma lista telefônica, pois podem ser pesquisados por critérios específicos, não apenas por um conjunto de categorias pré-definidas.

Deste modo, um serviço de diretório é toda infraestrutura capaz de disponibilizar a informação contida no diretório. Esta infraestrutura é representada por softwares, hardwares, processos e políticas utilizadas para acessar e administrar a informação.

LDAP

- Lightweight Access Directory Protocol, ou seja, protocolo leve de acesso a diretórios
- Especificado inicialmente em 1993 na RFC 1487
- Simplificação do Directory Access Protocol (DAP) para acesso a diretórios X.500
- Funciona sobre protocolos orientados à conexão
- Arquitetura Cliente/Servidor

O LDAP define um protocolo de mensagens utilizado por clientes e servidores de diretório. O protocolo utiliza diferentes mensagens, como por exemplo requisição de bind, que pode ser enviada do cliente ao servidor LDAP no início da conexão, ou operações de busca, utilizadas para pesquisar por uma entrada específica no diretório.



Trata-se de um padrão aberto que define um método para acessar e atualizar informações em um diretório, que tem ganhado ampla aceitação como um método de acesso a diretórios da internet, tornando-se estratégico dentro das intranets. LDAP define um protocolo de comunicação, isto é, define o transporte e o formato das mensagens utilizadas por um cliente para acessar informações em um diretório de tipo X.500. O LDAP não define o diretório; quando as pessoas falam sobre o diretório LDAP, referem-se à informação guardada que pode ser encontrada pelo protocolo LDAP.

Todos os servidores LDAP compartilham características básicas, desde que estejam baseados no padrão proposto pelas Requests For Comments (RFC). Entretanto, devido a diferenças de implementação, eles não são completamente compatíveis.

LDAP foi desenvolvido como uma alternativa leve em relação ao DAP, requerendo recursos mais leves e o protocolo TCP/IP, mais popular que o protocolo de camadas OSI. LDAP também simplifica algumas operações X.500 e omite as características mais exóticas.

A primeira versão do LDAP foi definida em X.500 Lightweight Access Protocol (RFC 1487), que foi substituído pelo Lightweight Directory Access Protocol (RFC 1777). LDAP refinou ideias presentes em protocolos anteriores, sendo uma implementação mais neutra e de complexidade reduzida, servindo para encorajar o desenvolvimento de aplicações com suporte a diretórios.

OpenLDAP

- Implementação open source de LDAP v3
- Independente de plataforma
- Mecanismos fortes de autenticação SASL
- Confidencialidade e integridade de dados com uso do protocolo SSL/TLS
- Internacionalização através do uso do Unicode
- Orientações e continuação
- Revelação de esquemas
- Controles e operações estendidas

Existem muitas implementações de servidores de diretórios, muitas das quais incompatíveis entre si. Na maioria dos casos, as implementações de servidores são concebidas para servir a determinado software e possuem restrições de uso ou características exóticas, como é o caso do MS Active Directory ou IBM Lotus Domino. O OpenLDAP, implementação mantida pela Fundação OpenLDAP, é um servidor LDAP de código aberto e de uso geral, ou seja, não agrega nenhum outro serviço que não tenha relação com a administração do diretório. Fundado em 1998, o projeto OpenLDAP foi baseado em uma implementação de servidor LDAP feita pela Universidade de Michigan. Deste modo, o OpenLDAP foi escolhido como



servidor de diretório para o projeto e-AA, sendo instalado através dos scripts que estão disponíveis na página do projeto:

- <http://wiki.rnp.br/display/cafewebsite/Procedimentos+de+entrada+na+CAFe>

Definições LDAP

- Modelos LDAP
 - Descrevem as informações que podem ser armazenadas no diretório e o que pode ser feito com elas
- Esquemas LDAP
 - Definem a estrutura de uma entrada em um diretório e os atributos que podem ser inseridos nela

Os quatro modelos básicos definidos pelo LDAP (Informação, Nomes, Funcional e Segurança) permitem descrever por completo a operação de um serviço de diretório: que informações podem ser armazenadas e o que pode ser feito com elas.

O modelo de Informação define o tipo de informação que pode ser armazenada em um diretório LDAP, enquanto o modelo de Nomes define como a informação pode ser organizada e referenciada no diretório LDAP. O modelo funcional descreve as operações que podem ser realizadas nos dados presentes no diretório e, por fim, o modelo de segurança recomenda o uso de autenticação e mecanismos de controle do acesso aos dados.

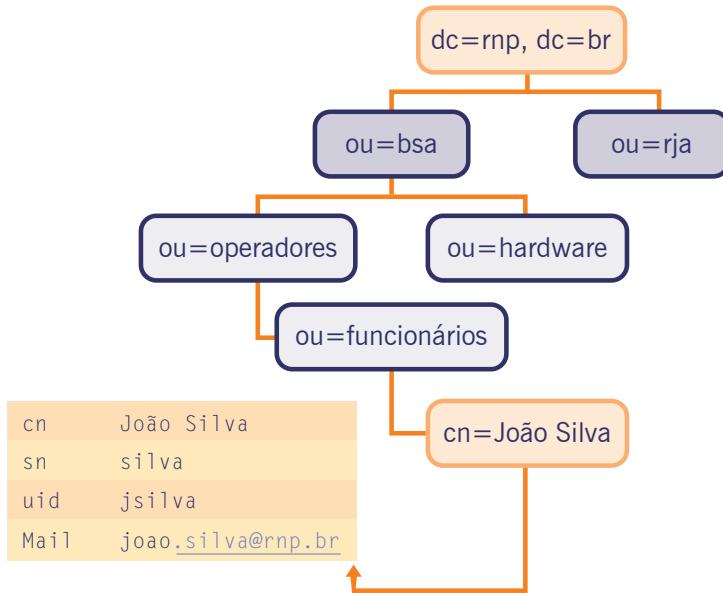
- Modelo de informação
 - Descreve a estrutura da informação no diretório LDAP
 - Unidades básicas de informação são objetos chamados de entradas
 - Entradas são compostas por uma coleção de atributos
 - Entradas são dispostas em estrutura de árvore chamada Directory Information Tree (DIT)

A unidade básica de informação guardada no diretório é chamada de entrada. Entradas representam objetos de interesse no mundo real, como pessoas, servidores ou organizações. Entradas são compostas de coleções de atributos que contêm informações sobre o objeto. Todo atributo tem um tipo e um ou mais valores. O tipo do atributo está associado com uma sintaxe que especifica o tipo de valor que pode ser gravado. Por exemplo, uma entrada deve ter um atributo, e a sintaxe associada ao tipo do atributo deve especificar os valores possíveis para este atributo. Em adição, na definição dos dados que podem ser guardados como os valores de um atributo, uma sintaxe de atributo também define como estes valores se comportarão durante pesquisas e outras operações. Alguns atributos possuem apelidos (alias) que podem ser utilizados como os nomes reais dos mesmos. Por exemplo, commonName e cn representam o mesmo atributo, sendo cn um alias para commonName.



Vínculos podem ser associados com tipos de atributos para limitar o número de valores que podem ser guardados em um atributo ou para limitar o tamanho total de um valor. Por exemplo, um atributo que contém uma imagem poderia ser limitado ao tamanho de 10 KB para prevenir o uso demasiado de espaço de armazenamento; ou um atributo usado para guardar um número de CPF pode ser limitado a um único valor.

Figura 2.1
Modelo de informação DIT.



Entradas são organizadas em forma de estrutura de árvore invertida, chamada DIT ou árvore de informação do diretório. O modelo de nome define como estas entradas são identificadas unicamente, o que reflete a estrutura vista na figura 2.1.

- Modelo de informação
 - Entradas (Objetos)
 - ▲ Cada entrada possui um nome único (DN)
 - ▲ Em geral, toda entrada utiliza uma classe abstrata, pelo menos uma estrutural, e pode possuir classes auxiliares
 - ▲ Possuem apenas atributos definidos nas classes de objetos
 - Classes de objetos
 - ▲ Definem quais atributos são opcionais e obrigatórios
 - ▲ Podem ser abstratas, estruturais ou auxiliares
 - ▲ Podem herdar propriedades de outras classes

Uma classe de objetos (*objectclass*) é um termo LDAP que denota um tipo de objeto representado por uma entrada do diretório ou registro. Alguns tipos de objetos típicos são *person*, *organization*, *organizationUnit*, *domainComponent* e *groupOfNames*. Há também classes de objetos que definem relações entre objetos, tal como a classe de objeto *top*, que estipula que um objeto pode ter objetos subordinados a ele, em uma estrutura hierárquica de árvore.

Uma classe de objetos é declarada como abstrata, estrutural ou auxiliar. Uma classe de objeto abstrata é usada como modelo para criação de outras classes. Uma entrada do diretório não pode ser instanciada por uma classe de objeto abstrata. Entradas do diretório são instanciadas por classes de objetos estruturais. Uma classe de objetos auxiliar fornece um método para estender classes estruturais sem mudar a definição do esquema desta classe estrutural. Deste modo, uma classe auxiliar não pode ser a única a instanciar uma entrada do diretório. É obrigatório que em uma entrada do diretório haja ao menos uma classe estrutural.

Classes de objetos LDAP definem conjuntos de atributos padrões que são listados como atributos obrigatórios (MUST) e atributos opcionais (MAY). Diferentes classes podem prescrever alguns atributos que se sobrescrevem, ou são redundantes com atributos de outras classes. Muitas classes de objetos são definidas em uma ordem hierárquica, onde uma classe é dita herdeira de outra classe superior. Considere o objeto LDAP, que é definido com as classes de objetos:

- ▶ objectclass: top
- ▶ objectclass: person
- ▶ objectclass: organizationalPerson
- ▶ objectclass: inetOrgPerson
- ▶ objectclass: posixAccount

A ordem mostrada para as classes de objetos acima indica uma relação hierárquica entre estas classes, mas não necessariamente. A classe *top* está no topo da hierarquia. Muitas outras classes que não são subordinadas a nenhuma outra classe têm *top* como classe superior. A classe *person* é subordinada de *top* e requer que os atributos *cn* e *sn* sejam populados, permitindo vários outros atributos opcionais. A classe *organizationalPerson* é uma subclasse de *person*, portanto uma classe herdeira, assim como a classe *inetOrgPerson*.

Como exemplo, a classe *posixAccount* é subordinada à classe *top* e requer que os atributos *cn* e *uid*, dentre outros, sejam populados. Perceba que isso se sobreponhe aos requerimentos para *cn* da classe *person*. Isto significa que temos que guardar o atributo *cn* duas vezes? Não, ambas as classes requerem a presença de um atributo *cn*. Não é possível adicionar atributos sem valor ou apenas preenchidos com espaço, não havendo restrição em relação ao valor contido ou existência de uma exclusividade de atributos em relação às classes.

Os métodos de definição de classe de objetos para LDAPv3 são descritos nas RFCs 2251 e 2252. A forma genérica de definição de classes de objetos é mostrada abaixo:

Modelos de informação: Classes de objetos

```
objectclass ( <OID da classe de objeto>
              [ "NAME" <nome da classe de objetos> ]
              [ "DESC" <Descrição da classe de objeto> ]
              [ "OBSOLETE" ] )
```



```
[ “SUP” <OID da classe de objeto ancestral> ]
[ ( “ABSTRACT” | “STRUCTURAL” | “AUXILIARY” ) ]
[ “MUST” <atributos obrigatórios> ]
[ “MAY” <atributos opcionais> ]
)
```

Cada classe de objeto começa com uma sequência de números delimitados por pontos. Estes números são referenciados como OID (Object Identifier); WHSP é uma abreviação de “white space” e apenas indica a necessidade de um espaço. Depois do OID está o nome da classe (NAME) seguido por uma descrição (DESC). Se a classe é subordinada a outra, a classe superior (SUP) é listada. Finalmente, a definição da classe de objetos especifica os atributos obrigatórios (MUST) e os opcionais (MAY).

Modelos de informação: Classes de objetos

```
objectclass ( 2.5.6.6 NAME ‘person’
              SUP top STRUCTURAL
              MUST ( sn $ cn )
              MAY ( userPassword $ telephoneNumber $ seeAlso
                    $ description ) )
```

Como mais um exemplo, suponha que uma classe chamada *person* foi definida incluindo um atributo *surname*. A classe de objeto *organizationalPerson* poderia ser definida como uma subclasse de *person*. A classe *organizationalPerson* teria os mesmos atributos da classe *person* e poderia adicionar outros atributos, como *title*. A classe de objetos *person* pode ser chamada de superior da classe *organizationalPerson*.

Modelo de informação: Atributos

```
attributetype ( <OID do atributo>
                [ “NAME” <nome do atributo> ]
                [ “DESC” <descrição do atributo> ]
                [ “OBSOLETE” ]
                [ “SUP” <OID da classe ancestral> ]
                [ “EQUALITY” <regra de comparação> ]
                [ “ORDERING” <regra de comparação> ]
                [ “SUBSTR” <regra de comparação> ]
                [ “SYNTAX” <OID da sintaxe> ]
                [ “SINGLE-VALUE” ]
                [ “COLLECTIVE” ]
                [ “NO-USER-MODIFICATION” whsp ]
                [ “USAGE” whsp attributeUsage ] )
```

Tudo que a classe de objetos faz é definir os atributos, ou o tipo de itens de dados contidos em um tipo de objeto. A definição de atributos é independente da definição de classe de objetos. Alguns exemplos são atributos típicos como *cn* (common name), *sn* (surname), *givenName*, *mail*, *uid* e *userPassword*. Como as classes de objetos, os atributos são definidos com OIDs únicos, com cada atributo contendo também um único número OID ligado a ele.



Uma classe de objeto instancia os atributos, permitindo que sejam utilizados de forma consistente nas entradas do diretório. A definição de atributos é independente da definição de uma classe de objetos.

Na definição de um atributo, há opções como SUP, OBSOLETE, SINGLE-VALUE, COLLECTIVE, NO-USER-MODIFICATION e USAGE. As demais opções devem ser fornecidas na definição. Mesmo o uso de regras de comparação dependerá de cada definição. Atributos com a opção SINGLE-VALUE não podem ter mais de um valor nas entradas. NO-USER-MODIFICATION é geralmente usado em atributos controlados ou de uso exclusivo do servidor do serviço de diretório.

Modelo de informação: Atributos

```
attributetype ( 2.5.4.20 NAME 'telephoneNumber'  
    DESC 'RFC2256: Telephone Number'  
    EQUALITY telephoneNumberMatch  
    SUBSTR telephoneNumberSubstringsMatch  
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.50{32} )
```

O atributo *telephoneNumber* é definido com um OID único, um nome e uma breve descrição. O nome é um apelido para o OID. Os valores que podem ser associados a este atributo são descritos pela sintaxe 1.3.6.1.4.1.1466.115.121.1.50{32}, que aceita números, hífens e espaços e no máximo até 32 caracteres.

- Padrão IANA para OIDs
 - ▲ Cada atributo e classe de objeto possui um único identificador OID, registrado na IANA.
- <http://www.iana.org>
 - ▲ Para criar novos atributos e classes de objetos é preciso requisitar o cadastro da instituição junto à IANA.
 - ▲ A RNP adquiriu o OID 1.3.6.1.4.1.15996, e novos atributos e objetos podem ser numerados a partir dele.

Cada elemento de um esquema é identificado por um OID (Object Identifier). Para evitar ambiguidades e estabelecer uma padronização para a codificação desses identificadores, os OIDs são registrados por uma autoridade específica, a IANA (Internet Assigned Numbers Authority).

O sistema de numeração de objetos é hierárquico e a IANA garante que um OID será usado por um objeto apenas.

- Modelo de informação: Exemplos de sintaxes
 - ▲ Booleano: 1.3.6.1.4.1.1466.115.121.1.7
 - ▲ DN: 1.3.6.1.4.1.1466.115.121.1.12
 - ▲ Caractere UTF-8: 1.3.6.1.4.1.1466.115.121.1.15
 - ▲ Inteiro: 1.3.6.1.4.1.1466.115.121.1.27
 - ▲ Caractere numérico: 1.3.6.1.4.1.1466.115.121.1.36

- ▶ Endereço postal: 1.3.6.1.4.1.1466.115.121.1.41
- ▶ Áudio: 1.3.6.1.4.1.1466.115.121.1.4
- ▶ Certificado: 1.3.6.1.4.1.1466.115.121.1.8
- ▶ JPEG: 1.3.6.1.4.1.1466.115.121.1.28

A RFC 2252 define um conjunto de sintaxes que podem ser usadas com o LDAP-v3 e as regras pelas quais os valores dos atributos definidos por meio dessas sintaxes são representados para serem transmitidos via protocolo LDAP. Destacamos aqui alguns exemplos de sintaxes de atributos.

Tabela 2.2
Exemplos de regras de comparação.

Nome	Tipo	Descrição
BooleanMatch	equality	Boleana
CaselgnoreMatch	equality	Não diferencia maiúsculas e minúsculas
CaselgnoreOrderingMatch	ordering	Não diferencia maiúsculas e minúsculas
CaselgnoreSubstringsMatch	substrings	Não diferencia maiúsculas e minúsculas
CaseExactMatch	equality	Diferencia maiúsculas e minúsculas
NumericStringOrderingMatch	ordering	Numérico

A RFC 2798 descreve um conjunto de regras de casamento para uso com o LDAP-v3. Três tipos de comparação podem ser usados:

- ▶ Igualdade (equality);
- ▶ Ordenação (ordering);
- ▶ Concatenação (substring).

Destacamos aqui alguns exemplos de regras de casamento para cada um dos tipos de comparação.

- ▶ Modelo de nomes
 - ▶ Entradas são nomeadas de acordo com sua posição na DIT
 - ▶ DNs são formados por Relative Distinguished Names (RDN) que tem a forma: <nome do atributo> = <valor>
 - ▶ Enquanto DNs identificam unicamente uma entrada no diretório, RDNs fazem o mesmo dentro de um nível do diretório

Entradas são arranjadas dentro da DIT baseada em seus DNs. Um DN é um nome único que identifica sem ambiguidades uma única entrada *single*. DNs são feitos de sequências de RDNs (Relative Distinguished Name), ou nome distinto relativo.



Cada RDN em um DN corresponde a um ramo em uma DIT saindo da raiz até a entrada do diretório. Cada RDN é derivado de atributos de entradas de diretório. De forma simplificada, um RDN tem a forma <nome do atributo> = <valor>. Um DN é composto de uma sequência de RDNs separados por vírgulas.

Entradas em um diretório LDAP são identificadas por seus nomes. As características destes nomes são:

- ▲ Eles têm duas formas, uma representação por cadeias de caracteres e uma URL.
- ▲ Eles têm uma sintaxe uniforme.
- ▲ Limite do espaço de nomes não é evidente.

Um componente de um nome é chamado de Relative Distinguished Name (RDN), que representa o ponto dentro da hierarquia do espaço de nomes. RDNs são separados e concatenados usando uma vírgula (,). Cada RDN é de um tipo definido. RDNs podem ser multi-valorados: atributo = valor + atributo = valor.

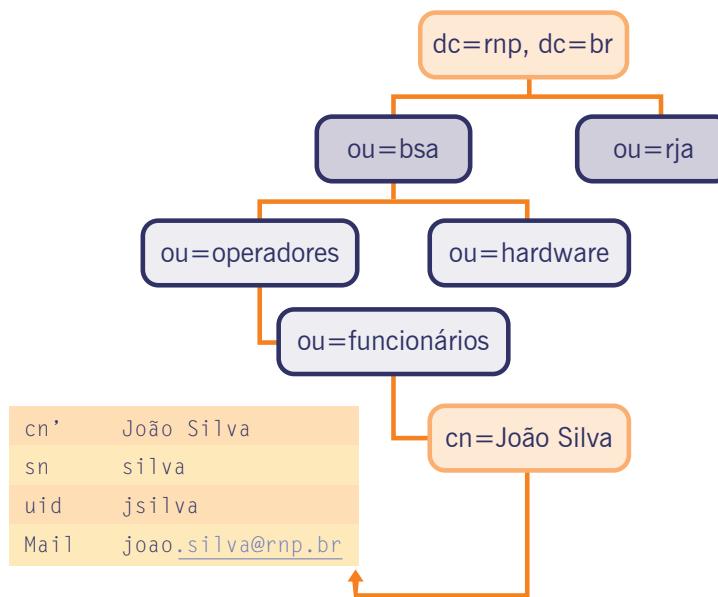
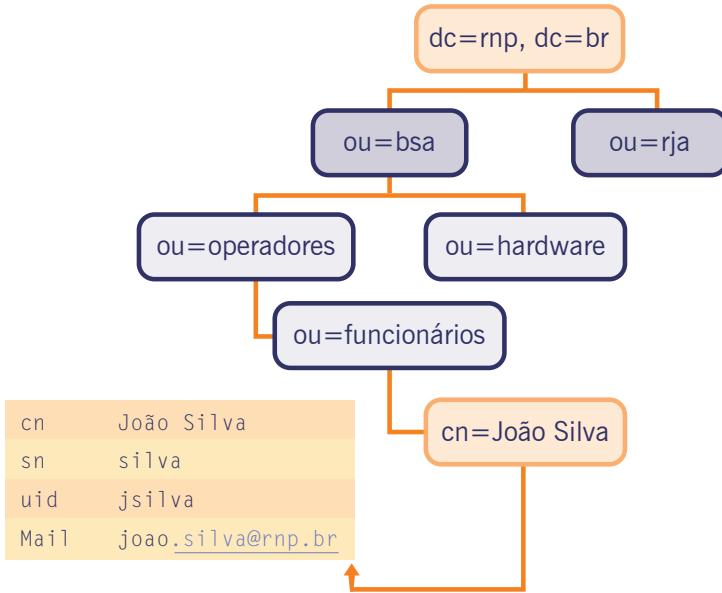


Figura 2.3
Modelo de nomes:
DN e RDN.

Em síntese, as entradas de um diretório são dispostas de forma hierárquica, onde o DN de uma entrada indica a localização de uma entrada dentro da DIT. DNs são formados por RDNs, que são na realidade os DNs separados por vírgula das entradas anteriores, contando-se da raiz da DIT até a entrada em questão. Cada entrada recebe como RDN um atributo ou uma soma de atributos com seus respectivos valores.

`cn=João Silva,ou=funcionarios,ou=operadores,
ou=bsa,dc=rnp,dc=br`

Figura 2.4
Modelo de
nomes:
Representação
por strings.



A sintaxe exata para nomes é definida na RFC 2253. Os exemplos seguintes são DNs válidos escritos na forma de string:

cn=Joao Silva,dc=RNP,dc=BR

Este é um nome contendo três RDNs:

ou=operadores + ou=funcionarios,ou=BSA,o=RNP

Novamente há três RDNs; porém, o primeiro RDN é multi-valorado:

cn=Joao Silva,ou=operadores\,BSA,dc=RNP,dc=br

Usando-se barra invertida (\), tem-se um caractere de escape para utilizar vírgula (,), igual (=) e demais caracteres especiais na formação dos RDNs:

ou=Antes\Depois,o=Teste,c=br

Este é um exemplo em que o valor contém o caractere de retorno (ODH).

Para definição mais detalhada sobre a forma de string de DNs, consulte a RFC 2253.

ldap://servidor/cn=João Silva,ou=funcionarios,
ou=operadores,ou=bsa,dc=rnp,dc=br?uid

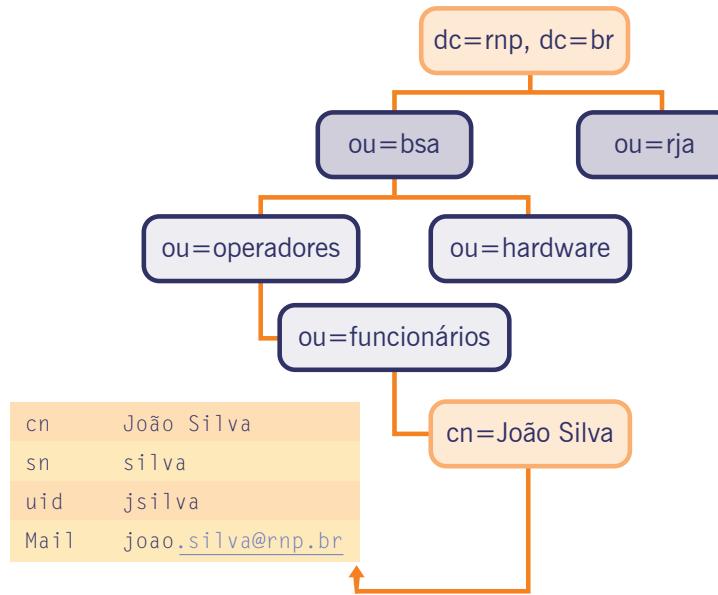


Figura 2.5
Modelo de
nomes:
Representação
por URL.

O formato da URL LDAP tem a forma geral:

ldap://<host>:<porta>/<caminho>,

onde <caminho> tem a forma :

<dn>[?<atributos>[?<escopo>?<filtro>]]]

O <dn> é um nome distinto LDAP (DN) usando a representação em string. O <atributo> indica os atributos que devem ser retornados da entrada ou entradas. Se for omitido, todos os atributos serão retornados. O <escopo> especifica o escopo da busca a ser feita. O escopo pode ser uma entrada, um nível, entrada e filhos imediatos, ou uma sub-árvore inteira. O filtro especifica o filtro de busca a ser aplicado às entradas dentro do escopo especificado durante a busca. O formato de URL permite a clientes de internet, como navegadores web, terem acesso direto ao protocolo LDAP, e consequentemente ao diretório.

- Modelo funcional
 - Três categorias de operações que podem ser realizadas em LDAPv3
 - ▲ Autenticação:
 - ▲ bind
 - ▲ unbind
 - ▲ abandon
 - ▲ Pesquisa:
 - ▲ search
 - ▲ compare
 - ▲ Atualização:
 - ▲ add
 - ▲ modify
 - ▲ delete
 - ▲ modifyRDN

O modelo funcional LDAP é composto por três categorias de operações que podem ser feitas contra um servidor LDAPv3:

- ▲ Autenticação – Operações de *Bind*, *Unbind* e *Abandon* usadas para conectar a um servidor LDAP ou desconectar-se dele, estabelecer direitos de acesso e proteger a informação.
- ▲ Pesquisa – *Search* e *Compare* para pesquisar ou comparar entradas de acordo com o critério especificado.
- ▲ Atualização – *Add* para adicionar uma entrada, *Delete* para excluí-la, *Modify* para modificá-la e *ModifyRDN* para modificar seu RDN.
- ▲ Comparação – A operação de comparação é utilizada para verificar as entradas que têm um atributo com determinado valor. Se a entrada tem o valor, a operação *Compare* retorna VERDADEIRO; caso contrário, retorna FALSO.

- ▲ Modelo funcional

- ▲ Pesquisa
 - ▲ Base
 - ▲ Escopo
 - ▲ Filtro de busca
 - ▲ Atributos para retornar
 - ▲ Limites

A operação mais comum é a de pesquisa, bastante flexível e com algumas opções mais complexas, permitindo a um cliente pedir que o servidor LDAP pesquise através de alguma porção da DIT, procurando informações de acordo com o critério especificado, listando os resultados. Não há distinção entre ler e listar. A pesquisa pode ser muito geral ou específica. Ela permite especificar um ponto de início dentro da DIT, a profundidade da busca, os atributos que uma entrada deve ter para ser considerada compatível e os atributos que devem ser retornados e ainda se os valores destes atributos devem ser retornados ou não.

Para realizar uma busca ou pesquisa, os seguintes parâmetros devem ser especificados:

- ▲ Base – Um DN que define o ponto de início da busca, chamado de objeto base. O objeto base é um nó dentro da DIT.
- ▲ Escopo – Especifica a profundidade da busca iniciada do objeto base dentro da DIT. Há três escolhas: *baseObject*, *singleLevel* e *wholeSubtree*. Se *baseObject* é especificado, somente o objeto base é examinado. Se *singleLevel* é especificado, somente as entradas filhas do objeto base são examinadas. Já com *wholeSubtree*, o objeto base e todos seus descendentes são examinados.
- ▲ Filtro de busca – Especifica o critério ao qual uma entrada deve se encaixar para que seja retornada na pesquisa.



- Atributos para serem retornados – Seleciona os atributos que devem ser retornados das entradas que se encaixam no critério de busca.
- Limites – Limitação do número de entradas retornadas.

Operador	Exemplo
&	(&(cn=joao)(sn=silva))
	((uid=joao)(uid=silva))
!	(!(uid=joao))
=	gidNumber=100
~=	sn~=silv
>=	uidNumber>=5000
<=	Sn<=silva
*	*

Tabela 2.6
Filtros de busca
(atributo
operador valor).

Um filtro de busca define a qual critério uma entrada deve encaixar-se para ser retornada em uma pesquisa. O componente básico de um filtro de busca é um valor de atributo na forma:

Filtro: <Atributo> <operador> <valor>

Filtros de busca podem ser combinados com operadores lógicos para formar filtros mais complexos. A sintaxe para combinar filtros é:

(“&” ou “|” (filtro1) (filtro2) ...)(“!” (filtroN))

Operadores:

= igualdade

>= maior igual

<= menor igual

~= aproximação

=* quaisquer caracteres

Operações de autenticação são usadas para estabelecer e finalizar uma sessão entre um cliente e um servidor LDAP. A sessão pode estar segura em vários níveis, desde uma sessão anônima insegura (uma sessão autenticada na qual o cliente identifica-se por fornecer uma senha) até sessão criptografada com mecanismos SASL ou SSL.

- Bind – Inicia uma sessão LDAP entre um cliente e um servidor. Permite ao cliente identificar-se ao servidor;



- ▶ Unbind – Termina uma sessão cliente-servidor;
- ▶ Abandon – Permite ao cliente pedir ao servidor que cancele uma operação.

Representação LDIF

- ▶ LDAP Data Interchange Format
 - ▶ Descrição de conjunto de entradas
 - ▶ Descrição de sentenças de atualização

LDAP Data Interchange Format (LDIF) é um formato de gerenciamento de informação que, como o nome sugere, significa formato LDAP de alteração de informação. Este formato permite manipular facilmente grandes quantidades de informação. A forma básica de uma entrada LDIF é:

```
dn: <nome distinto>
<atributo>: <valor>
<atributo>: <valor>
```

Uma linha pode ser continuada, começando uma nova linha com um caractere de espaço ou tabulação:

```
dn: cn=Jorge, ou=lcc, o=ufmg, c=br
```

Atributos multi-valorados são especificados em linhas separadas:

```
cn: João Silva
cn: João
```

Se o valor de um atributo contém um caractere que não esteja na codificação US-ASCII ou comece com um espaço ou dois-pontos (:), o valor do atributo é seguido por um duplo dois-pontos (::) e codificado em uma notação em base64. Entretanto, é sempre possível usar a codificação UTF-8 para suportar a internacionalização.

Existem duas construções para um LDIF:

- ▶ Descrição de conjuntos de entradas;
- ▶ Descrição de sentenças de atualização.

Descrição de conjunto de entradas:

```
dn: <distinguished name>
<attrdesc>: <attrvalue>
<attrdesc>: <attrvalue>
<attrdesc>:: <base64-encoded-value>
<attrdesc>:< <URL>
...

```



Um LDIF cuja estrutura é a de conjuntos de entradas contém todas as informações das entradas nele contidas, isto é, todos os atributos e seus respectivos valores estão presentes em cada uma de suas entradas:

```
dn: o=rnp
objectclass: top
objectclass: organization
o: RNP
description: Rede Nacional de Ensino e Pesquisa
```

```
dn: ou=esr
objectClass: top
objectclass: organizationalUnit
ou: ESR
description: Escola Superior de Redes
```

Com este tipo de LDIF, quando uma entrada é modificada, a entrada é sobreescrita, isto é, todas as informações da entrada no diretório são substituídas pelas informações no LDIF quando é feita uma operação de atualização. Os atributos que não existem no LDIF, mas que existem na entrada do diretório serão apagados quando for realizada a operação de atualização. As operações com este tipo de LDIF são similares a sobreescriver um arquivo de um sistema operacional por outro arquivo; as informações do arquivo antigo deixam de existir, dando lugar a novas informações. Esta estrutura de LDIF é importante ao carregar ou fazer uma cópia do diretório inteiro e adicionar uma nova entrada.

Descrição de conjunto de entradas:

```
dn: cn=Joao Silva
,dc=rnp,dc=br
objectclass: top
objectclass: person
cn: Joao Silva
sn: Silva
cn:: IGJ1Z2lucyB3aXRoIGEgc3BhY2U=
cn:< file:///tmp/arquivo
```

Já um LDIF estruturado em sequências de atualização contém apenas as informações relevantes para as modificações necessárias a uma entrada do diretório. Comparado ao tipo anterior, onde o foco está em operações realizadas nas entradas como um todo, em um LDIF o tipo de sequências de atualização permite realizar modificações em um único atributo de uma entrada. Sua forma básica é:

```

dn: <nome distinto>
changeType: <Tipo da operação>
<operação>: <atributo>
<atributo>: <valor>
-
<operação>: <atributo>
<atributo>: <valor>
-
...

```

Observe que em todos os tipos de LDIF as entradas são separadas por uma linha em branco e, para um LDIF de sequências de atualização, cada operação em um atributo diferente deve ser separada por uma linha contendo um hífen (-).

Descrição de sentenças de atualização:

```

dn: <distinguishedname>
changetype: <[modify|add|delete|modrdn]>
<[modify|add|delete|modrdn]>: <attributetype>
<attrdesc>: <value1>
...
-
<[modify|add|delete|modrdn]>: <attributetype>
<attrdesc>: <value1>
<attrdesc>: <value2>
...
-
dn: cn=Joao Silva,dc=rnp,dc=br
changetype: add
objectclass: person
objectclass: inetorgperson
cn: Joao
cn: Joao Silva
sn: Silva
dn: cn=Joao Silva,dc=rnp,dc=br
changetype: modify
add: givenName
givenName: jo
givenName: Joao
-
replace: description
description: Funcionario Joao

```



Comandos de shell e ferramenta gráfica

- Principais clientes LDAP por linha de comando
 - ▀ *ldapadd* <opções> -f <arquivo LDIF>
 - ▲ Adiciona entradas nos diretórios
 - ▀ *ldapmodify* <opções> -f <arquivo LDIF>
 - ▲ Altera os dados no diretório, seja modificando entradas ou adicionando-as
 - ▀ *ldapdelete* <opções> <lista de DNs | -f arquivo>
 - ▲ Exclui entradas do diretório
 - ▀ *ldapsearch* <opções> <filtro de busca>
 - ▲ Realiza buscas no diretório de acordo com critérios específicos

Os comandos listados acima fazem parte da distribuição do OpenLDAP. Estes comandos shell são utilizados com o uso de argumentos que configuram a operação que se deseja realizar no diretório. O *ldapadd* é na realidade um *ldapmodify* com o argumento *-a* indicando adição de entradas. Para o *ldapadd* os parâmetros mais comuns são um usuário com permissão de escrita, uma senha e um arquivo LDIF contendo as entradas a serem adicionadas no diretório. No caso do *ldapmodify*, o que muda é que o arquivo LDIF contém os dados que devem ser modificados, seja por operação de exclusão ou adição de novos dados, ou apenas através da substituição de valores de atributos. O *ldapdelete* também precisa de um usuário com permissões de escrita, e o arquivo que é passado como parâmetro contém uma lista de DNs que devem ser excluídos do diretório. Esta lista pode ser passada na linha de comando. Por fim, *ldapsearch* requer os parâmetros listados anteriormente e o resultado da busca está sujeito a permissões de acesso ao diretório para o usuário utilizado.

As principais opções utilizadas nos comandos shell são os seguintes parâmetros:

1. -x: informa ao comando para utilizar bind simples, não utilizando SASL.
2. -D: define qual será a identidade utilizada para realizar a operação.
3. -W: retorna o prompt para que a senha da identidade indicada com o parâmetro -D seja digitada.
4. -f: lê um arquivo no formato LDIF contendo as operações a serem realizadas no diretório.

Apache Directory Studio é um cliente LDAP feito em uma plataforma Eclipse e possuindo uma série de plugins. O ApacheDS é uma ferramenta completa para ser utilizada em qualquer servidor LDAP; LDAP Browser permite não apenas mostrar os dados como também criar, modificar, editar e remover entradas.

A figura 2.7 mostra a tela inicial do ApacheDS. Para utilizá-lo como cliente LDAP, basta ir ao menu LDAP e configurar a conexão com um servidor.

Exemplos de comandos shell:

```
ldapadd -x -H ldap://servidor.ldap -D  
"cn=admin,dc=curso,dc=ldap" -W -f arquivo.ldif  
ldapmodify -x -D "cn=admin,dc=esr,dc=rnp,dc=br" -W -f arquivo.  
ldif  
ldapsearch -x -D "cn=admin,dc=esr,dc=rnp,dc=br" -W -b  
"dc=curso,dc=ldap" uid=00123456  
ldapdelete -x -D "cn=admin,dc=esr,dc=rnp,dc=br" -W "uid=dijkstra,  
ou=people,dc=esr,dc=rnp,dc=br"
```

Figura 2.7
Ferramenta
gráfica: Apache
Directory Studio.



Escolhendo nova conexão, uma nova tela é aberta (figura 2.8), onde é possível configurar o nome de conexão, o servidor LDAP a ser conectado, a porta de acesso e o uso de protocolos de segurança.

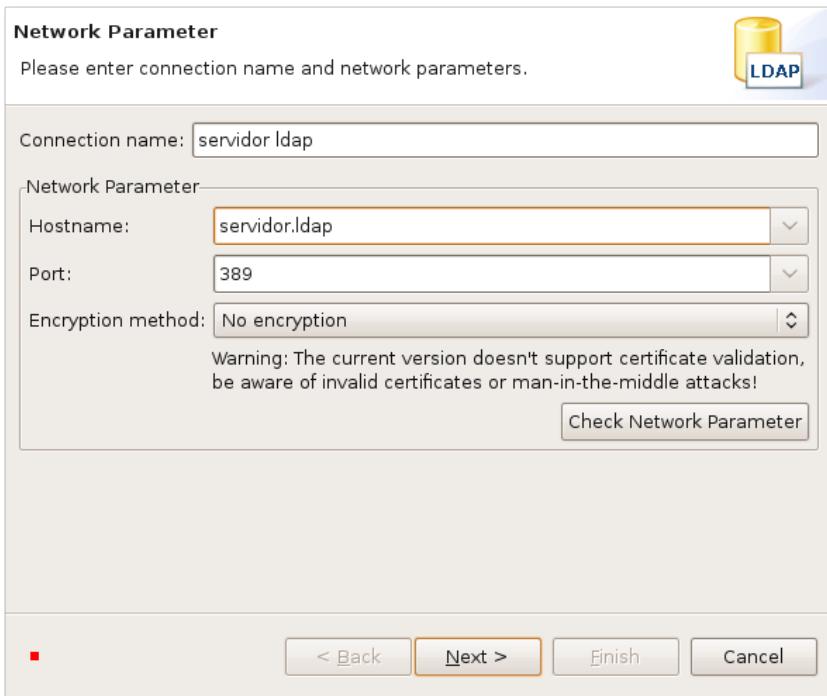


Figura 2.8
Conexão com o servidor LDAP.

A tela mostrada na figura 2.9 permite configurar as opções de acesso, ou seja, usuário e senha de acesso ao servidor LDAP.

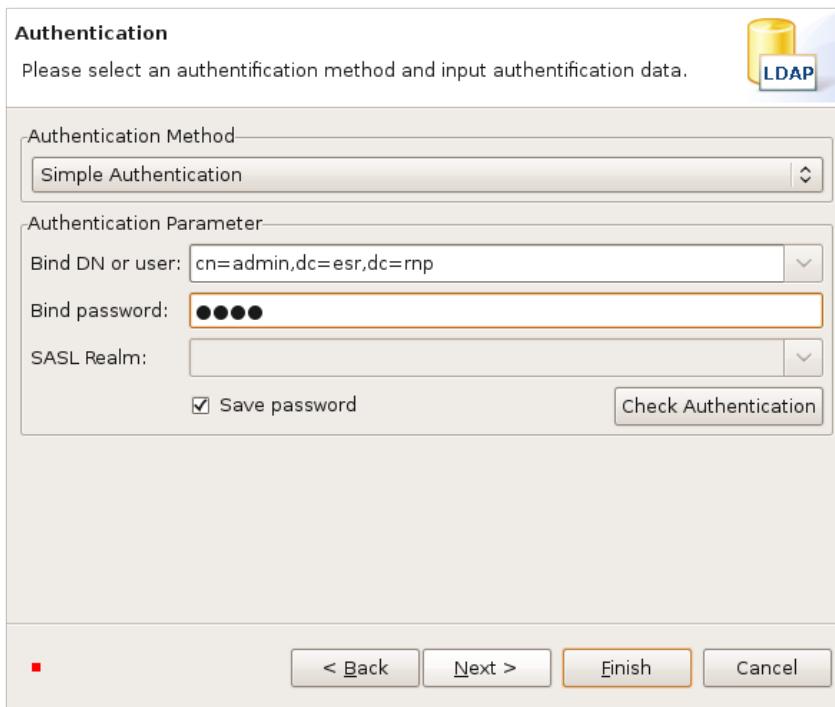
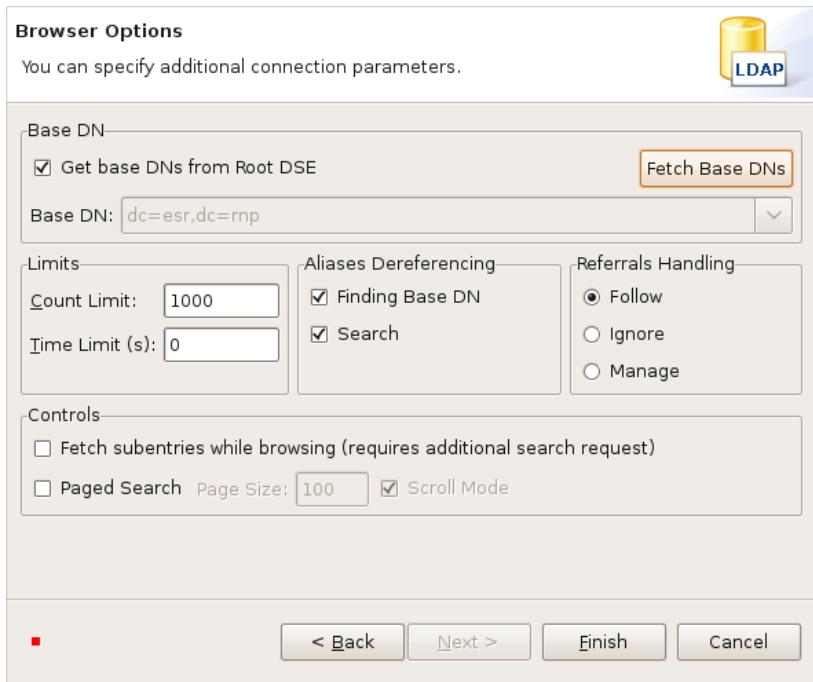


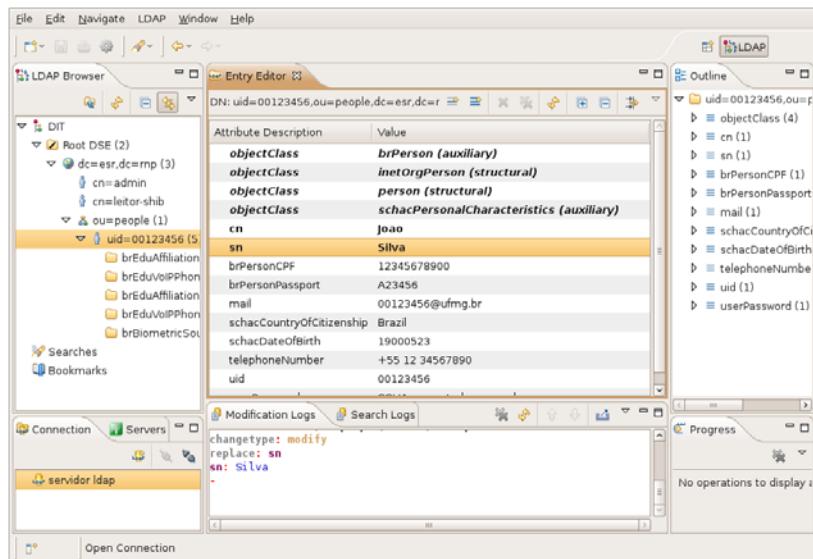
Figura 2.9
Administrador da base LDAP.

Figura 2.10
Opções de navegação no diretório.



Com a opção “Fetch Base DNs” é possível obter o DN da base LDAP apenas consultando o servidor.

Figura 2.11
Tela principal do ApacheDS.



A figura 2.11 mostra o ambiente padrão da função browser do ApacheDS, que possibilita navegar pelo diretório LDAP e executar com simplicidade modificações nos dados. Pode-se perceber as abas descritas como “LDAP Browser”, “Entry Editor” e “Modification Logs”, que são úteis na administração de alguns dados e para a visualização de informações no diretório.

Esquema brEduPerson

- Classes do esquema brEduPerson
 - ▀ Esquema proposto para membros de instituições de ensino superior no Brasil
 - ▀ Divide-se em:
 - ▲ Informações gerais sobre qualquer cidadão
 - ▲ Informações gerais sobre membros de uma instituição
 - ▲ Informações específicas sobre funcionários e alunos
- Relacionamentos modelados em estrutura hierárquica

O esquema brEduPerson é uma proposta de esquema LDAP para participantes de instituições de ensino superior no Brasil. Ele armazena informações específicas para a realidade do país, tais como: informações genéricas de qualquer cidadão brasileiro (CPF, entre outras), informações gerais sobre os membros de uma instituição (e-mail, cargo, entre outros), além de informações específicas sobre os funcionários e alunos destas instituições.

- Classes de objetos e atributos
 - ▀ brPerson
 - ▲ brPersonCPF, brPersonPassport
 - ▀ brEduPerson
 - ▲ brEduAffiliationType, brEntranceDate, brExitDate, brEduAffiliation
 - ▀ brBiometricData
 - ▲ brCaptureDate, brBiometricSource, brBiometricData
 - ▀ brEduVoIP
 - ▲ brEduVoIPalias
 - ▲ brEduVoIPtype
 - ▲ brEduVoIPadmin
 - ▲ brEduVoIPcallforward
 - ▲ brEduVoIPaddress
 - ▲ brEduVoIPexpiryDate
 - ▲ brEduVoIPbalance
 - ▲ brEduVoIPcredit
 - ▲ brEduVoIPphone

O esquema brEduPerson define quatro classes de objetos:

- brPerson (com atributos gerais sobre pessoas);
- brEduPerson (com atributos comuns para pessoas em universidades);
- brBiometricData (com atributos sobre dados biométricos);
- brEduVoIP (com atributos sobre telefones VoIP).



Modelo de nomes para uso na Federação CAFe

- Necessidade de refletir na base de dados o fato de uma mesma pessoa desempenhar diferentes papéis dentro da sua instituição ou possuir mais de um número VoIP, cada um com suas características, ou armazenar dados biométricos de fontes distintas.
- Exemplos:
 - O mesmo aluno em mais de um curso, com data de ingresso e código do curso distintos
 - Um professor exercendo diferentes funções em períodos determinados
 - ▲ Coordenação de curso
 - ▲ Direção de unidade

Ao definir o modelo de nomes a ser usado em instituições de ensino e pesquisa, é necessário tratar a questão do modelamento de relacionamentos entre conjuntos de informações.

Devemos capturar na base de dados, por exemplo, o fato de uma mesma pessoa poder desempenhar diferentes papéis dentro da instituição. Exemplos: um aluno matriculado em mais de um curso, um professor desempenhando diferentes funções, com cada uma delas associada a uma data de ingresso e de saída, entre outras informações.

Para modelar esses relacionamentos, estudamos algumas alternativas e optamos pelo uso de uma solução hierárquica, que será descrita a seguir.

- Modelo proposto:
 - O item principal – pessoa de uma instituição – será tratado como um container abaixo do qual aparecerão nós com as informações relacionadas
 - ▲ Vínculos distintos com a instituição. Exemplos:
 - ▲ Professor, aluno, funcionário
 - ▲ Telefones VoIP
 - ▲ Fontes biométricas

Os nós em um diretório LDAP formam uma árvore. Cada nó, independentemente de ser pai de algum outro nó na árvore, é uma entrada com suas próprias informações (atributos).

Esses nós são por vezes chamados de “containers” na terminologia LDAP.

O item principal (em nosso exemplo, uma pessoa com inserção em instituição de ensino e/ou pesquisa) com o qual se deseja relacionar as demais informações, será tratado como um container, abaixo do qual aparecerão nós com as informações relacionadas.



Por exemplo, abaixo da entrada que descreve dados básicos de uma pessoa, podemos ter entradas descrevendo vínculos, como professor e aluno.

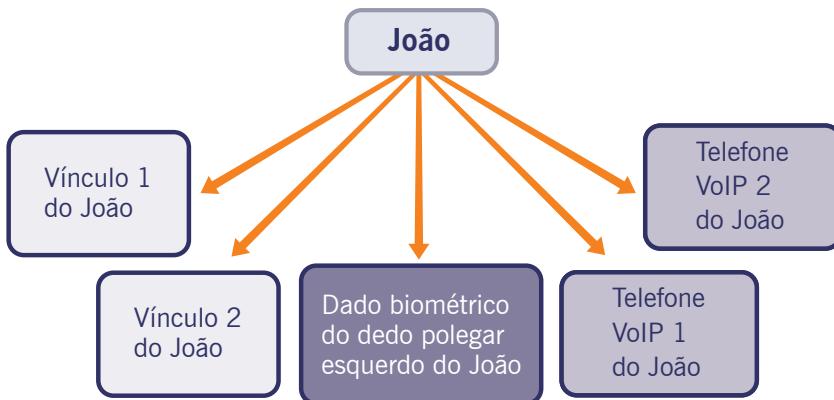


Figura 2.12
Entradas
descrevendo
vínculos.

Esta solução tem como vantagem a manutenção da possibilidade de recuperação da informação em uma única consulta, os tipos originais dos atributos, e não serem criadas classes e atributos artificiais.

Como desvantagem, temos uma árvore cuja topologia é ditada por relacionamentos, o que pode causar confusão por não ser a maneira tradicional de desenhar uma topologia.

Exemplos de entradas:

```

dn: uid=silvana,ou=people,dc=uff,dc=br
objectClass: person
objectClass: inetOrgPerson
objectClass: brPerson
objectClass: schacPersonalCharacteristics
uid: silvana
brcpf: 12345678900
brpassport: A23456
schacCountryOfCitizenship: Brazil
telephoneNumber: +55 22 81389199
cn: Silvana
userPassword: *****
  
```

```

dn: braff=1,uid=silvana,ou=people,dc=uff,dc=br
objectclass: brEduPerson
braff: 1
brafftype: faculty
brEntranceDate: 20070205
dn:braff=2,uid=silvana,ou=people,dc=uff,dc=br
  
```

```

objectclass: brEduPerson
braff: 2
brafftype: student
brEntranceDate: 20070205
brExitDa
te: 20080330

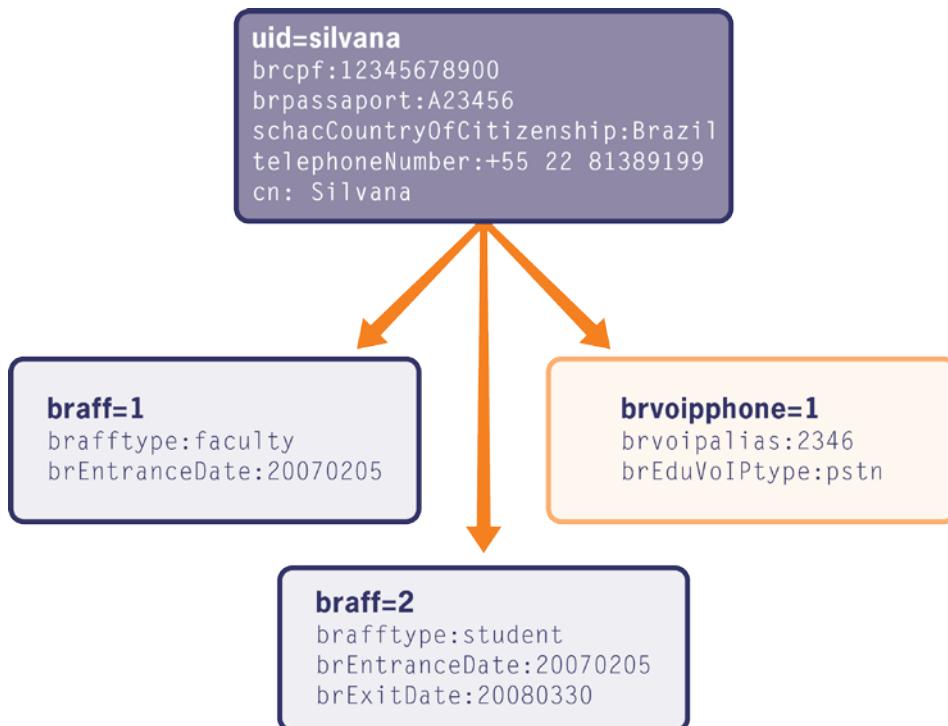
```

```

dn:brvoipphone=1,uid=silvana,ou=people,dc=uff,dc=br
objectclass: brEduVoIP
brvoipphone: 1
brvoipalias: 2346
brEduVoIPtype: pstn
brEduVoIPadmin:uid=admin,ou=people,dc=uff,dc=br

```

Figura 2.13
Exemplos de entradas.



Referências

- Documento “Proposta de Esquema brEduPerson: Federação CAFé: setembro de 2008”, disponível no site do projeto e-AA.
- Esquemas:
brEduPerson-20080917-0.0.6.schema
schac-20061212-1.3.0.schema
RFC 2252
RFC 2798
LDAP(v.3): Attribute Syntax Definitions
LDAP(v.3): Matching Rules

2

Roteiro de Atividades Revisão de LDAP e esquema brEduPerson

Tópicos e conceitos

- Protocolo LDAP
- Esquema brEduPerson

Competências técnicas desenvolvidas

- Instalação de um servidor LDAP
- Utilização de clientes LDAP

Tempo previsto para as atividades

- 40 – 60 minutos

Servidor de sala de aula

- Os arquivos para instalação do sistema encontram-se na máquina virtual presente na área de trabalho, pasta */opt/treinamento*. Ver também pasta Treinamento no desktop do Windows.



Atividade 1 – Instalar e configurar um serviço de diretório OpenLDAP

O projeto e-AA fornece um roteiro detalhado para instalação de todos os softwares necessários para que a sua instituição faça parte da federação CAFe.

Execute os comandos passo a passo para a instalação do diretório LDAP na sua máquina virtual (MV). Para facilitar abra um terminal SSH, copie e cole os comandos.

1. Logue-se na MV como *sudo*:

```
sudo su -
```

2. Faça a atualização dos pacotes:

```
apt-get update
```

3. Faça a instalação do pacote *slapd*:

```
debconf-set-selections <<-EOF
```

```
slapd    slapd/no_configuration boolean true
```

```
EOF
```

```
apt-get -y install slapd
```

4. Pare o serviço de LDAP (é normal dar erro, pois ainda não foi configurado):

```
/etc/init.d/slapd stop
```

5. Faça a cópia dos arquivos de configuração:

```
cp /opt/treinamento/ldap/slapd /etc/default/slapd
```

```
cp /opt/treinamento/ldap/slapd.conf /etc/ldap/slapd.conf
```

```
cp /opt/treinamento/ldap/ldap.conf /etc/ldap/ldap.conf
```

```
cp /opt/treinamento/ldap/DB_CONFIG /var/lib/ldap/DB_CONFIG
```

```
cp /opt/treinamento/ldap/eduperson.schema /etc/ldap/schema/  
eduperson.schema
```

```
cp /opt/treinamento/ldap/breduperson.0.0.6.schema /etc/ldap/  
schema/breduperson.0.0.6.schema
```

```
cp /opt/treinamento/ldap/schac-20061212-1.3.0 /etc/ldap/  
schema/schac-20061212-1.3.0
```

6. Após fazer a cópia dos arquivos, deve-se atentar para a necessidade de fazer breves alterações em alguns dos arquivos, conforme segue:

- **/etc/ldap/slapd.conf**: deve-se substituir as ocorrências de **\${HOSTNAME}** pelo IP da máquina. Deve-se substituir ainda as ocorrências de **\${RAIZ_BASE_LDAP}** pelo valor correspondente à raiz da base LDAP de sua instituição, como por exemplo:

```
dc=instituicao,dc=br.
```



- **/etc/ldap/ldap.conf**: deve-se substituir as ocorrências de **\${RAIZ_BASE_LDAP}** pelo valor correspondente à raiz da base LDAP, como por exemplo:
dc=instituicao,dc=br.

Obs: para substituir no editor de texto VIM pode-se utilizar o seguinte comando:
:%s/palavra_a_ser_substituida/nova_palavra/g

Exemplo:

```
:%s/ ${RAIZ_BASE_LDAP}/dc=ufmg,dc=br/g
```

7. Geração de certificado SSL para LDAP: antes de executar este comando troque as ocorrências de **SUBSTITUIR_IP_MAQUINA** pelo IP da sua MV.

```
sed -e 's/HOSTNAME_FQDN/'SUBSTITUIR_IP_MAQUINA'/ -i /opt/treinamento/openssl.cnf
```

```
openssl genrsa -out /etc/ldap/SUBSTITUIR_IP_MAQUINA.key 2048  
-config /opt/treinamento/openssl.cnf
```

```
openssl req -new -key /etc/ldap/SUBSTITUIR_IP_MAQUINA.key  
-out /etc/ldap/SUBSTITUIR_IP_MAQUINA.csr -batch -config /opt/treinamento/openssl.cnf
```

```
openssl x509 -req -days 730 -in /etc/ldap/SUBSTITUIR_IP_MAQUINA.csr -signkey /etc/ldap/SUBSTITUIR_IP_MAQUINA.key -out /etc/ldap/SUBSTITUIR_IP_MAQUINA.crt
```

8. Inicialize o LDAP através do comando:

```
/etc/init.d/slapd start
```

9. Carga Inicial de Dados: o LDAP que foi instalado encontra-se vazio, ou seja, não há nenhum elemento em sua base de dados. Agora iremos fazer a carga inicial de dados na base LDAP.

Para isso edite o arquivo *popula.sh* que se encontra no */opt/treinamento*, alterando o valor da variável *RAIZ_BASE_LDAP* para o valor informado no passo 5: *dc=<SUINSTUIÇÃO>,dc=br*

Não se esqueça de salvar. Para abrir e editar o arquivo digite:

```
vim /opt/treinamento/popula.sh
```

10. Execute o script através das seguintes linhas de comando:

```
/etc/init.d/slapd stop
```

```
sh /opt/treinamento/popula.sh  
/etc/init.d/slapd start
```

11. Execute os seguintes comandos para instalar utilitários para manipulação do LDAP:

```
apt-get install ldap-utils  
/etc/init.d/slapd restart
```

Agora o LDAP já está instalado na sua MV.

Atividade 2 – Editar o arquivo LDIF e executar alterações no diretório

Nas atividades a seguir, quando for requisitada utilize a senha do usuário admin do LDAP: 1234

1. Crie um arquivo *atividade2.ldif* contendo os dados abaixo e substituindo o que estiver entre <> por valores personalizados:

```
dn: uid=<LOGIN>,ou=people,dc=<SUAINSTITUICAO>,dc=br  
objectClass: person  
objectClass: inetOrgPerson  
objectClass: brPerson  
objectClass: schacPersonalCharacteristics  
uid: <LOGIN>  
brcpf: 12345678900  
brpassport: A23456  
schacCountryOfCitizenship: Brazil  
telephoneNumber: +55 22 81389199  
mail: <EMAIL>  
cn: <NOME>  
sn: <SOBRENOME>  
userPassword: <SENHA>  
schacDateOfBirth: <DATA NASC. : YYYYMMDD>  
schacGender: 10
```



2. Carregue o arquivo *atividade2.ldif* no diretório:

```
ldapadd -f atividade2.ldif -x -D "cn=admin,dc=<SUAINSTITUICAO>,dc=br" -W
```

Neste comando os parâmetros são:

- x

Informa ao *ldapadd* que utilize operação de bind simples.

- D <DN>

Especifica um DN para realizar o bind.

-W

Mostra o prompt para digitar a senha do DN especificado com a opção *-D*.

- f <arquivo>

Especifica um arquivo LDIF cujos dados serão adicionados ao diretório.

3. Verifique a inserção dos dados no diretório substituindo <LOGIN> e <SUAINSTITUICAO> pelo valor associado no item 1.

```
ldapsearch -x -D "cn=admin,dc=<SUAINSTITUICAO>,dc=br" -W -b dc=<SUAINSTITUICAO>,dc=br "uid=<LOGIN>"
```

No comando acima, além dos parâmetros utilizados no item anterior há também:

- b

Especifica uma base para começar a busca; deve ser um DN da base LDAP.

"uid=<LOGIN>"

Filtro de busca que seleciona as entradas que se encaixam no critério especificado.

4. Remova a entrada adicionada ao diretório no item 1:

```
ldapdelete "uid=<LOGIN>,ou=people,dc=<SUAINSTITUICAO>,dc=br" -x -D "cn=admin,dc=<SUAINSTITUICAO>,dc=br" -W
```

5. Verifique a remoção da entrada repetindo o comando do item 3.



Atividade 3 – Utilização de ferramenta gráfica para acesso ao servidor LDAP

1. Clique no ícone do Apache Directory Studio que se encontra no seu desktop para executá-lo:
 - 1.1. Escolha no menu a opção LDAP;
 - 1.2. Clique em *New Connection*.
2. Entre com os dados do servidor LDAP:
 - 2.1. Preencha o nome da conexão;
 - 2.2. Preencha o IP do servidor LDAP (IP da sua MV);
 - 2.3. Clique em *Check Network Parameter*.
3. Para os parâmetros de autenticação siga os seguintes passos:
 - 3.1. Preencha o campo *Bind DN ou User* com “cn=admin,dc=<**SUA_INSTITUICAO**>,dc=br”;
 - 3.2. Em *Bind password* entre com a senha do usuário *admin*, senha: 1234;
 - 3.3. Clique em *Check Authentication*.
4. Configure o DN da base:
 - 4.1. Clique em *Fetch Base DNS*;
 - 4.2. Clique em *Finish*;
 - 4.3. Feche a aba *Welcome*;
 - 4.4. Na tela LDAP Browser procure pelo usuário cujo uid=00123456.
5. Importe um arquivo LDIF com o ApacheDS:
 - 5.1. Abra o arquivo *people.ldif* (que se encontra na área de trabalho) e edite-o trocando <**SUA_INSTITUICAO**> pela sigla da sua instituição. Salve o arquivo.
 - 5.2. No menu *File* escolha a opção *Open File*;
 - 5.3. Clique em *Browse* localizado acima do arquivo e escolha o nome da conexão;
 - 5.4. Para importar o LDIF, clique na seta verde (*Execute LDIF*) ao lado do botão *Browse* e observe a importação das entradas;
 - 5.5. Verifique se as entradas foram importadas.



3

Construindo metadiretórios com EID

- ▶ Visão geral do EID
 - ▶ Motivação
 - ▶ Metadiretórios
 - ▶ Export Import Directory (EID)
 - ▶ EID e brEduPerson
- ▶ Configuração de extrações
 - ▶ Definição de repositório
 - ▶ Definição de extrações
 - ▶ Definição de processos
 - ▶ Agendamentos

Visão geral do EID

Motivação

- ▶ Ao contrário de pequenos diretórios, diretórios grandes não podem ser gerenciados manualmente.
- ▶ O desenvolvimento de um integrador a partir do zero tem um custo muito alto.
- ▶ Solução: integração com sistemas já existentes.

Esta terceira sessão do curso apresentará conceitos gerais sobre metadiretórios, além de demonstrar como construir metadiretórios com a ferramenta Export Import Directory (EID), detalhando a definição de repositórios, extrações, processos e agendamentos.

Diretórios que possuem um baixo fluxo de pessoas ou poucas dezenas de cadastros podem ser facilmente gerenciados pela inclusão e exclusão manual de registros.



Diretórios com muitos usuários e com comportamento mais dinâmico demandam um esforço maior de manutenção, o que praticamente inviabiliza seu gerenciamento manual. Este é o caso de diretórios acadêmicos, onde entram e saem centenas (ou milhares) de pessoas todos os semestres.

Entretanto, os processos que implicam a modificação do diretório já existem e, em geral, são registrados formalmente em algum sistema, como é o caso de ingresso e formatura de alunos, aposentadoria de professor ou técnico etc. Desta forma, é possível aproveitar essas informações e integrar a manutenção do diretório com esses processos.

O desenvolvimento de extractores para o cenário específico de uma organização pode ser muito alto, porém a ideia central é sempre a mesma: consolidar os dados para a construção do diretório. O objetivo do Export Import Directory (EID) é facilitar a integração de dados de diversos sistemas para construir um metadiretório e, por fim, um ou mais diretórios.

Metadiretório

- ▲ Base de dados intermediária para construção do diretório
- ▲ Modelo independe do esquema final do diretório

De acordo com o Burton Group (<http://www.burtongroup.com/>), um metadiretório é uma junção de esquemas e atributos de diferentes repositórios em uma visão comum. O metadiretório ideal permite a um administrador fazer alterações em um repositório e prover a atualização da informação em todos os diretórios ligados a ele.

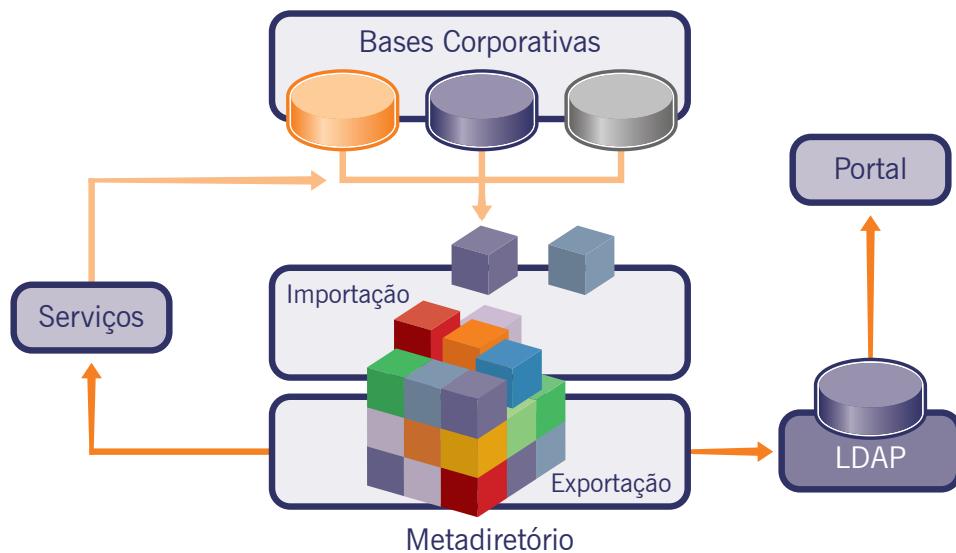


Figura 3.1
Fluxo de
informações em
metadiretório.

A figura 3.1 demonstra o fluxo de informações em um metadiretório: os dados das bases corporativas serão importados para o metadiretório, e do metadiretório os dados podem ser exportados para o LDAP e utilizados para autenticação em um portal, por exemplo.

EID

- Desenvolvido pelo Grupo São Tomé da UFMG
- Recursos da RNP
 - GTs diretório
 - Projeto e-AA
- Recursos da SESu/MEC
 - Projeto PingIFES

O EID foi desenvolvido pelo Grupo São Tomé da UFMG para ser utilizado no projeto Infraestrutura de Autenticação e Autorização (e-AA), que tem como objetivo principal implantar um serviço experimental de autenticação e autorização federativa para as instituições de ensino e pesquisa.

- Export Import Directory Tool
 - Ferramenta para facilitar a construção e manutenção de metadiretórios
 - Extensão do PCollecta
 - Integrado aos processos administrativos já consolidados
 - Atualização contínua dos dados

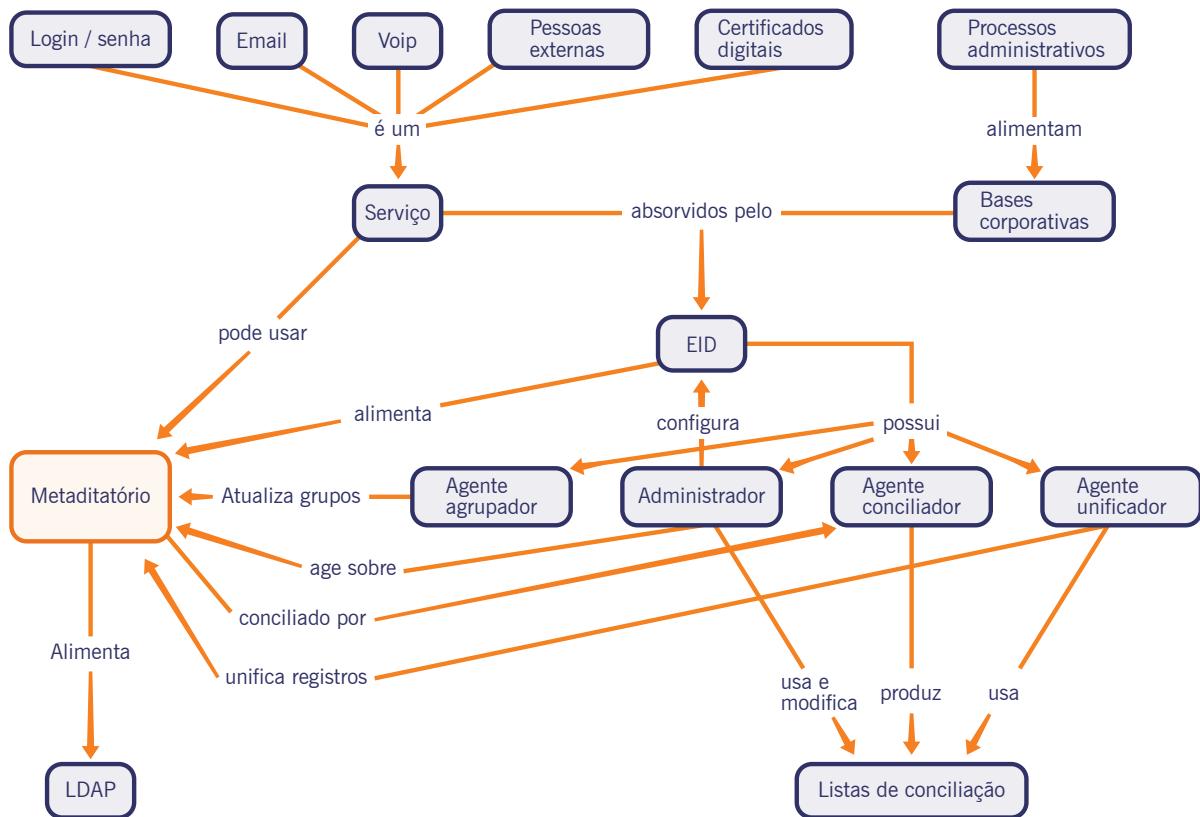
O EID foi desenvolvido tendo por base a ferramenta PCollecta, uma ferramenta de Extração, Transformação e Carga (ETL), utilizada pelas instituições de ensino superior para alimentação do modelo de dados (PingIFES) definido pelo MEC. O EID é integrado aos processos administrativos já consolidados pelas instituições e possibilita a atualização contínua dos dados importados das bases corporativas.

- Importação por conexão direta nas bases institucionais
- Exposição dos dados via web services
 - Dados expostos como XML
 - Pode ser usado por diversas aplicações clientes

O EID pode conectar-se diretamente às bases de dados institucionais, desde que seja possível utilizar conectores JDBC para tal.

Os dados importados são associados a pessoas, e os registros completos dessas pessoas podem ser facilmente recuperados utilizando uma interface web service disponibilizada pelo EID.





A figura 3.2 mostra que vários serviços, como VoIP, e-mail e certificados digitais podem ser também incorporados ao metadiretório. O metadiretório, por sua vez, é alimentado através do EID pelas bases corporativas mantidas pelos processos administrativos da organização.

Figura 3.2
Serviços
incorporados ao
metadiretório.

- ▲ Estrutura dos dados
 - Grupos e pessoas são tipos de objetos
 - Objetos possuem um identificador global chamado Global Unique Identifier (GUID)
 - Dados são incorporados a pessoas e grupos pela implementação de classes
 - Estrutura semelhante a um diretório LDAP

O EID utiliza o conceito de Objeto (EidObject) para representar as informações que armazena. São considerados objetos: pessoas e definições de grupos. Um objeto é uma entidade que possui um identificador único e um conjunto de atributos, sendo a unidade mínima de armazenamento de informações.

Os atributos são mapeamentos nome-valor, onde o valor possui um tipo ou domínio definido. Os nomes e os tipos dos atributos são especificados em entidades denominadas classes.

As classes são definições de agrupamentos de atributos. Cada classe pode ser considerada uma definição de um tipo de dado composto.

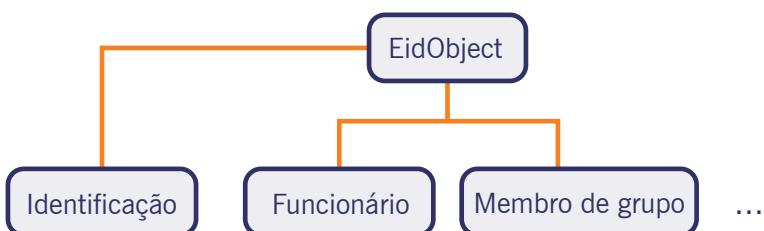
Denominamos de *instanciação* da classe o processo de atribuição de valores aos atributos definidos pela classe e sua associação a um objeto. Um objeto pode estar associado a várias instâncias de uma mesma classe ou de classes diferentes, mas não aos atributos individualmente. O usuário da ferramenta é livre para definir as classes que atendem às suas necessidades.

Todo objeto possui um identificador global, denominado GUID, gerado automaticamente pela ferramenta, que o identifica unicamente em todo sistema. Esse atributo é definido por uma classe especial denominada *EidObject*.

- Estrutura dos dados
 - Toda classe registrada gera uma tabela
 - As instâncias de classes são vinculadas via *EidObject*

Toda classe criada na aplicação gera uma tabela no banco de dados. A classe *EidObject* se relaciona com as demais classes do sistema, denominadas *EidClasses*. Qualquer classe definida pelo usuário é uma *EidClass*. Essas classes agregam a um objeto EID seus atributos específicos.

Figura 3.3
Classe
EidObject.



EID e brEduPerson

- Classes fornecidas pelo grupo e-AA
 - Identificação
 - Conta
 - E-mail
 - Endereço
 - Telefone
 - Professor
 - Técnico
 - Aluno
 - Biometria
- Definem os atributos necessários para brEduPerson
- Conversão pré-configurada das classes para LDIF
- Outras classes podem ser definidas



O EID não está limitado a classes específicas (exceto pela exigência das classes *Identificação*, *Grupo* e *MembroDeGrupo*), de forma que classes podem ser definidas a critério da organização utilizadora.

Com o intuito de facilitar a implantação da federação, o grupo e-AA fornece algumas classes que podem ser usadas para alimentar diretórios LDAP sem nenhuma configuração adicional. A razão disso é que já existe uma conversão pré-configurada para a ferramenta EID2LDAP, como veremos adiante. As classes fornecidas pelo grupo e-AA definem os atributos necessários para brEduPerson.

Outras classes podem ser definidas pela própria organização, para suprir suas necessidades. Estas modificações certamente deverão ser também refletidas na conversão utilizada pelo EID2LDAP para que as informações fluam automaticamente para o diretório.

Acesso ao EID

- Devem existir um ou mais administradores
- Responsabilidades:
 - Definir classes
 - Definir repositórios de origem
 - Configurar as extrações
 - Agendar as extrações
 - Gestão manual de pessoas
 - Gestão manual de grupos
- Administrador responsável pela configuração
- Usuários definidos em arquivo XML (padrão)

O EID pode ser acessado através da URL: <http://<máquina>:8080/eid>.

O usuário administrador deverá definir as classes necessárias à instituição utilizadora, fazer as configurações necessárias para a realização de extrações de dados de outras fontes para alimentar o metadiretório, além de fazer a gestão manual de pessoas e grupos.

Na instalação padronizada fornecida, as classes recomendadas para o brEduPerson são instaladas automaticamente.

Para acesso à aplicação devem ser definidos um ou mais administradores.

A autenticação do EID pode ser feita com vários tipos de bases de usuários (arquivo XML, banco relacional, LDAP etc.), que são configuradas no servidor de aplicação (Tomcat).

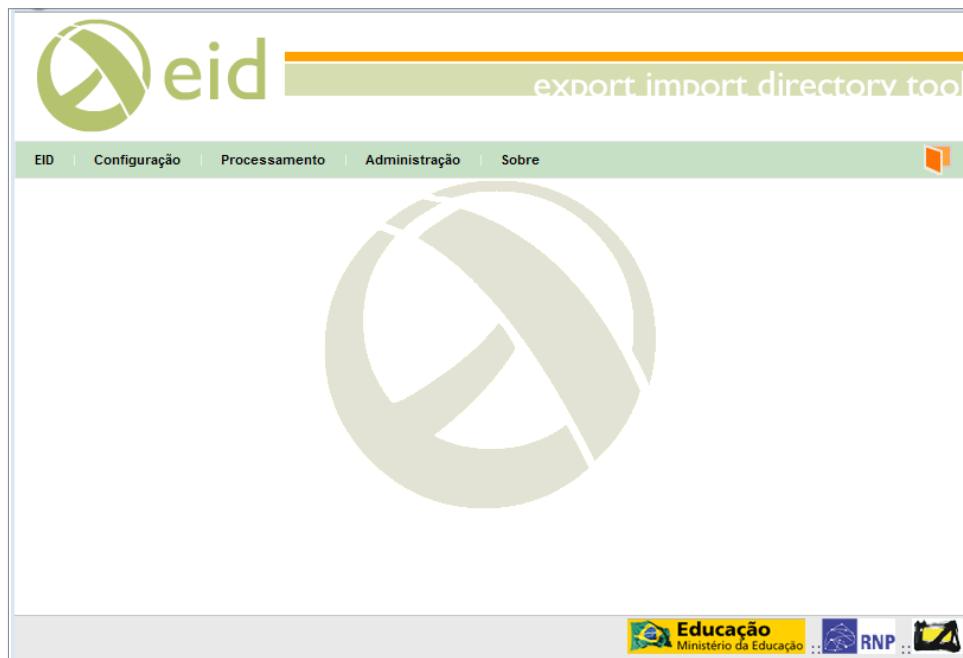


A distribuição utilizada configura os usuários no arquivo *tomcat-users.xml*. O login e senha de um administrador são definidos no momento da instalação.

Figura 3.4
Tela de login.



Figura 3.5
Tela inicial.



A figura 3.5 apresenta a tela inicial do EID; a grande maioria dos comandos está localizada na parte superior da tela, que disponibiliza menus e botões. Em algumas telas os botões podem ser encontrados em outras posições, o que é mais comum nos casos onde a tela demanda a inclusão de uma lista de itens.

- ▲ O menu *EID* dá acesso às funcionalidades de gestão de pessoas, grupos e classes, além de opções de conciliação.
- ▲ O menu *Configuração* possibilita configurar os repositórios, extrações, processos, parâmetros globais e ainda a opção de importar e exportar configuração de processos.
- ▲ O menu *Processamento* dá acesso ao agendamento de processos, resultado de processamento e controle do agente que escalona os processos.
- ▲ O menu *Administração* dá acesso à consulta de mapeamentos dos sistemas e também à consulta a repositórios de dados cadastrados.

Configurações iniciais

- Diretório de instalação do EID
- Classes

O EID compila código Java dinamicamente para cada nova classe definida. O código e as classes compiladas são colocados no diretório WEB-INF da aplicação, motivo pelo qual é necessário configurar este caminho no sistema. Para realizar esta configuração, acesse o menu *EID* e escolha a opção *Configuração*. Nesta tela deverá ser informado o caminho para o diretório WEB-INF do EID, diretório localizado dentro do Tomcat no qual sua aplicação está sendo executada.

Em seguida, devem ser definidas as classes que serão alimentadas, muito embora novas classes possam ser definidas posteriormente.

A distribuição padrão já configura previamente o diretório de instalação e as classes que serão utilizadas no decorrer do curso.

O EID confia na existência de três classes básicas para a conciliação de registros e criação de agrupamentos, que são as seguintes classes:

- Identificação – Dados básicos de identificação pessoal;
- Grupo – Definição de critérios de agrupamento;
- MembroDeGrupo – Associação de pessoas a grupos.

Definição de classes

Na versão atual do sistema, a alteração na definição de uma classe não é suportada, podendo produzir erros. A figura 3.6 mostra a tela de listagem de classes definidas no sistema.

Figura 3.6
Definição de classes.

	Nome	Descrição	Visualizar	Alterar	Excluir Registros
<input type="checkbox"/>	Aluno	Dados relativos a um aluno			
<input type="checkbox"/>	Conta	Define os atributos para dados biométricos			
<input type="checkbox"/>	Convidado	Dados relativos a um vínculo no IdP Convidados			
<input type="checkbox"/>	Email	Dados de e-mail de um indivíduo			
<input type="checkbox"/>	Endereco	Define os atributos para construção de endereços			
<input type="checkbox"/>	Grupo	Define a consulta para manutenção do grupo			
<input type="checkbox"/>	Identificacao	Define os atributos para identificação de indivíduos			
<input type="checkbox"/>	MembroDeGrupo	Define os campos para mapeamento de pessoas em grupos			
<input type="checkbox"/>	Professor	Dados relativos a um professor na instituição			
<input type="checkbox"/>	Tecnico	Dados relativos a um profissional técnico na instituição			
<input type="checkbox"/>	Telefone	Define os atributos para telefones de contato			

Educação
 RNP

De uma forma geral, todas as telas do sistema apresentam uma caixa de seleção, que é usada para selecionar registros para exclusão, um comando para visualizar os detalhes de cada registro (representado pela lupa) e um comando para editar os dados do registro (representado pelo lápis/caderno).

Para a tela de gestão de classes existe ainda o comando de excluir registros. Este comando promove a exclusão de todos os registros da classe em questão. Um caso especial é o da classe *Identificação*, que só pode ser removida após não existirem outras classes preenchidas.

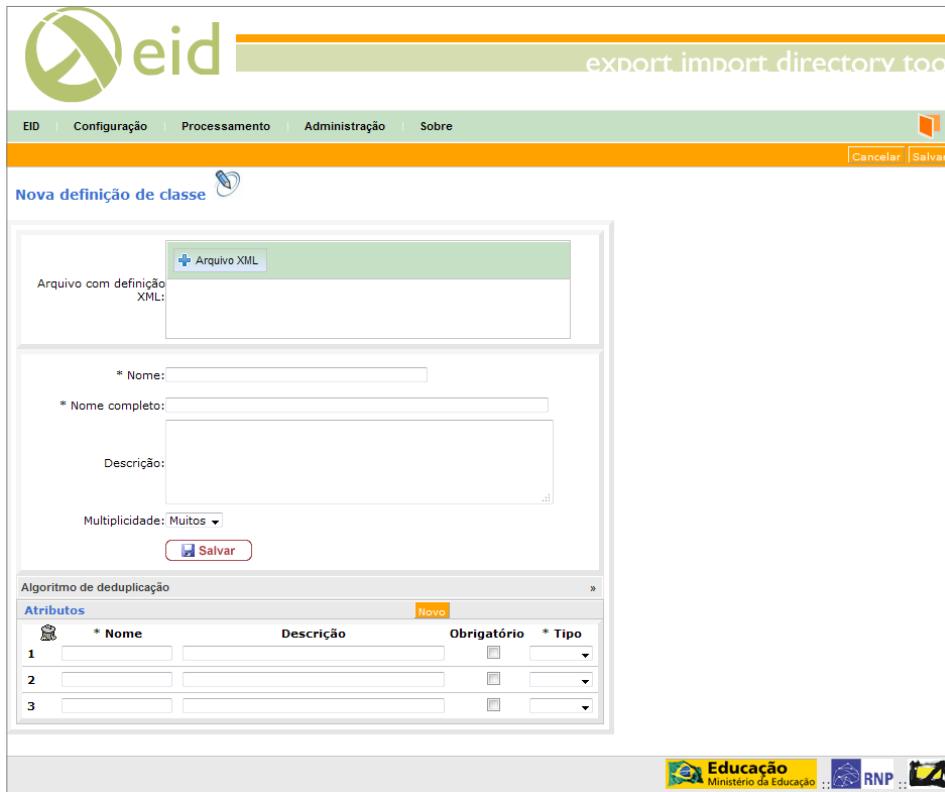


Figura 3.7
Nova definição
de classe.

Na tela para definição de uma classe, um arquivo contendo a definição XML da classe deve ser inserido por upload através do botão *Arquivo XML*.

Os campos *Nome*, *Nome completo*, *Descrição*, *Multiplicidade* e o detalhe de Atributos serão lidos do arquivo e preenchidos automaticamente pelo sistema.

O painel “Algoritmo de deduplicação” define a classe Java responsável pela deduplicação (unificação, junção, descarte) das instâncias dessa classe.

Este algoritmo pode ser cadastrado via upload ou inserção manual do conteúdo e deve implementar a classe IclassUnifier; também é possível apenas especificar o nome completo incluindo o caminho da classe caso ela esteja disponível no EID.

Caso não seja informado um algoritmo de deduplicação, o EID utiliza um algoritmo padrão para fazer a mesclagem dos atributos (br.ufmg.lcc.eid.model.unifier.DefaultSingleInstanceUnifier), caso a classe permita apenas uma instância por objeto, ou um algoritmo padrão para adicionar a instância a uma lista (br.ufmg.lcc.eid.model.unifier.DefaultMultipleInstanceUnifier), caso a classe permita várias instâncias.

Configuração de extrações

- Envolve
 - Repositórios
 - ETCs
 - Processos
 - Agendamentos

A configuração de extrações passa pelo cadastro de repositórios, definição de extrações, processos de extrações e agendamento dos processos.

Definição de repositórios

- Fontes ou destinos de dados
- Destino fixo:
 - Base de dados do EID (Metadiretório)
- Fontes são as bases institucionais
 - Bancos relacionais
 - Arquivos texto
- Necessário driver JDBC ou ODBC
- Driver JDBC deve estar disponível no diretório *lib* do Tomcat

Antes que sejam definidas as extrações, é necessário que sejam definidas as fontes de dados, considerando que o destino é sempre único: o metadiretório gerenciado pelo EID.

As fontes são os bancos de dados institucionais que o alimentarão, como as bases do RH, sistema acadêmico de graduação ou pós-graduação, planilhas etc.

O EID trabalha com diversos tipos de bancos de dados. O pré-requisito é a existência de um driver JDBC ou ODBC (para fazer ponte) para o EID. Também é possível importar arquivo em formato CSV, com campos separados por ponto-e-vírgula, tabulação, vírgula, suspenso(#) e barra vertical (|).

O driver JDBC deve estar presente no diretório *lib* do Tomcat no momento de sua inicialização para que seja reconhecido. Deve-se ter atenção especial para a versão do driver; consulte as instruções do fornecedor do banco para saber a versão mais adequada para uma determinada versão de banco.

A definição de repositórios é acessada pelo menu *Configuração/Repositório de Dados*.



Figura 3.8
Tela de administração de repositórios.

A figura 3.8 exibe a tela de administração de repositórios, onde é possível exibir, alterar ou remover repositórios cadastrados no sistema, ou ainda cadastrar novos repositórios. O repositório EID é configurado automaticamente no roteiro de instalação fornecido pelo projeto, e deve sempre ter o nome “Metadiretório” para o correto funcionamento do sistema. Os repositórios da organização devem ser configurados nesse ponto para que as extrações possam ser configuradas.

Figura 3.9
Campos para cadastro de banco de dados relacional.

Para cadastrar um novo repositório, ação o comando *Novo* na tela de Administração de Repositórios, e será apresentada a tela para escolha do tipo de

Repositório, que pode ser Arquivo CSV ou Banco de Dados Relacional.

De acordo com a escolha é exibida a tela para cadastro dos dados de conexão.

A figura 3.9 exibe os campos para cadastro de um repositório do tipo Banco de Dados Relacional.

- ▲ Os campos *Nome* e *Descrição* são utilizados para uma melhor identificação do repositório na interface.
- ▲ Em especial, os campos *URL* e *Driver* devem seguir a especificação do fabricante. Clicando no ícone ao lado do campo URL é exibida uma janela pop-up com exemplos de URLs e Drivers para diversos bancos de dados.
- ▲ Os campos *Usuário* e *Senha* indicam as credenciais a serem utilizadas para comunicação com o banco. Lembrando que por questões de segurança a senha nunca é exibida e o campo fica em branco.
- ▲ O Painel Versão do Banco de Dados pode ser preenchido com o nome da Tabela e campo do banco que contém a versão do mesmo ou ainda com o número de versão diretamente no campo *Versão* (manual).
- ▲ Após preencher todos os campos obrigatórios é possível testar a conexão com o banco, através do botão *Testar Conexão*.

Figura 3.10

Inclusão de arquivo CSV.



Se o tipo do Diretório for Arquivo CSV, os campos *Nome*, *Descrição* e *Diretório* devem ser informados de acordo com a figura 3.10. O campo *Diretório* deve apontar para o diretório no servidor local que conterá os arquivos.

Extrações

- ▲ Regras de conversão entre fonte e destino de dados
- ▲ Parâmetros podem ser usados como constantes nos SQLs e scripts

O próximo passo é a definição de uma extração de dados.

- ▲ Cada extração define a fonte de dados propriamente dita e a relaciona com uma tabela de destino;
- ▲ A regra de conversão e compatibilização de tipo também é definida aqui, mapeando os campos de entrada nos campos de saída;
- ▲ É possível a utilização de parâmetros globais nas extrações para denotar valores constantes no momento do processamento da extração.

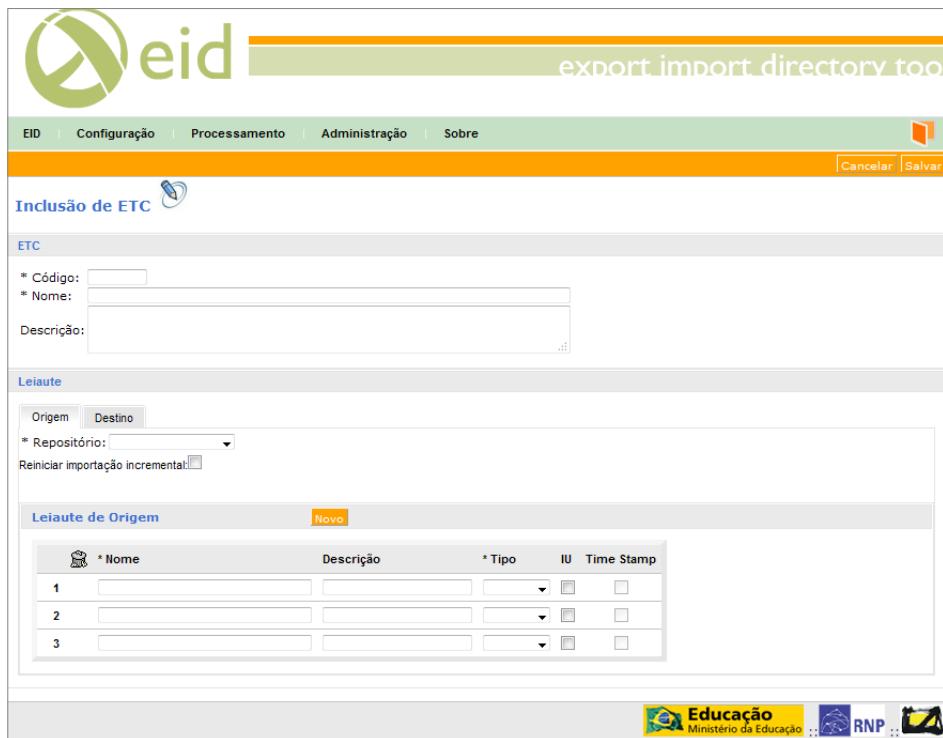
Código	Nome	Descrição	Clonar	Visualizar	Alterar
9	ETC Alunos	Etc importação de Alunos			
8	ETC Identificação	Etc para importar Pessoas			
10	ETC Usuários CSV	ETC Usuários CSV			

Figura 3.11
Administração
de extrações.

Extrações são conhecidas no sistema como ETC (Extração, Transformação e Carga). A tela para administração de extrações é acessada pelo menu *Configuração/ETC* (ver figura 3.11).

- ▲ O comando *Novo* permite a definição de uma nova extração (ou ETC), discutida em detalhes a seguir.
- ▲ O comando *Alterar* permite editar uma extração já configurada no sistema.
- ▲ O comando *Clonar* permite realizar uma cópia da ETC escolhida apenas com os campos *Código* e *Nome* vazios para serem redefinidos.
- ▲ O comando *Visualizar* permite exibir os dados da ETC em estado de somente leitura.
- ▲ É possível ainda excluir uma ETC cadastrada no sistema; para isso, selecione o item que se deseja excluir e clique no botão *Excluir*.

Figura 3.12
Tela de cadastro
de ETC.



Ao acionar o novo comando a tela para cadastro de uma ETC é exibida (ver imagem 3.12). O cadastro de ETC é dividido em três partes para facilitar a inserção dos dados: a parte Superior apresenta os campos *Código*, *Nome* e *Descrição*, que são campos descritivos da extração, e as abas *Leiaute de Origem* e *Leiaute de Destino* que serão detalhadas a seguir.

- Leiaute de origem pode ser:
 - Um SQL qualquer sobre o repositório de origem
 - Um arquivo texto presente no diretório
 - Deve definir um Identificador Único (IU)
 - É possível definir um campo como *time stamp* para importação incremental

A aba *Leiaute de origem* do cadastro de ETC define os campos que serão extraídos do repositório de origem, que podem ser definidos por um SQL ou mapeamento dos campos de um arquivo CSV.

- Para bancos de dados relacionais, o EID descobre dinamicamente os nomes e tipos, montando a lista de campos disponíveis para importação.
- Para arquivos texto, os campos devem ser cadastrados um a um.
- É obrigatória a definição de um Identificador Único (IU) para os registros importados. Este identificador pode ser composto, sendo utilizado para conciliação e referência a registros previamente importados.

- É possível definir também um campo como *time stamp* para possibilitar a importação incremental de registros. Este campo pode ser uma data de atualização dos registros no repositório de origem ou ainda um número sequencial, que é incrementado a cada alteração nos dados.

Leiaute

Origem Destino

* Repositório: Banco Academico ▾

```
select Id, Nome, Sexo, Nascimento, Documento, TipoDocumento, NomePai,
NomeMae FROM Pessoas
```

* SQL:

Leiaute

Reiniciar importação incremental:

Leiaute

Origem Destino

* Repositório: Arquivos ▾

* Arquivo de Origem: pessoas.txt

* Separador Decimal: Ponto * Separador Campos: Ponto e vírgula ▾

* Codificação Caracteres: UTF-8 * Formato da Data: dd/MM/yyyy

Reiniciar importação incremental:

No leiaute de origem (ver imagem 3.13) é escolhido o repositório de onde os dados serão extraídos; dependendo do tipo de repositório escolhido (Banco de Dados Relacional ou Arquivo CSV), os campos para preenchimento são customizados.

Figura 3.13
Leiaute de origem.

Para Banco de Dados Relacional o campo SQL deve ser informado.

O comando *Leiaute* monta a lista de campos encontrados automaticamente.

Pra arquivos CSV:

- Arquivo de Origem: nome do arquivo CSV.
- Separador decimal: indica o caractere utilizado como separador decimal em campos numéricos no arquivo texto.
- Separador Campos: indica o caractere utilizado como separador dos campos do arquivo de texto.
- Codificação de caracteres: utilizada para interpretação correta durante a leitura de arquivos texto.

- Formato da data: indica o formato da data.

Quando o Repositório de Origem for Arquivo Texto, o leiaute deverá ser montado manualmente, adicionando linhas através do comando novo. A opção *Reiniciar Importação Incremental* pode ser usada para zerar os valores da Importação incremental de uma determinada ETC.

	*Nome	Descrição	*Tipo	IU	Time Stamp
1	id	Id	Inteiro	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2	nome	Nome	Texto	<input type="checkbox"/>	<input type="checkbox"/>
3	sexo	Sexo	Texto	<input type="checkbox"/>	<input type="checkbox"/>
4	nascimento	Nascimento	Data	<input type="checkbox"/>	<input type="checkbox"/>
5	documento	Documento	Texto	<input type="checkbox"/>	<input type="checkbox"/>
6	tipodocumento	TipoDocumento	Texto	<input type="checkbox"/>	<input type="checkbox"/>
7	nomepai	NomePai	Texto	<input type="checkbox"/>	<input type="checkbox"/>
8	nomemae	NomeMae	Texto	<input type="checkbox"/>	<input type="checkbox"/>

Figura 3.14
Leiaute de origem.

O Leiaute do Origem é exibido após o preenchimento do campo SQL e acionamento do comando *Leiaute*, caso o tipo de repositório seja banco de dados relacional, ou pela inserção dos campos um a um, através do acionamento do comando *Novo*, caso o repositório seja do tipo arquivo CSV (ver imagem 3.14).

- O campo *Nome* indica a identificação do campo na origem. Este identificador será também utilizado para referenciá-lo no mapeamento para o destino.
- O campo *Tipo* indica o tipo dos dados. No caso de arquivos texto, o tipo deve ser sempre *Texto*.
- IU define os campos utilizados como identificadores únicos para o registro. Devem ser definidos com cautela para evitar erros durante a importação, como conciliação incorreta de registros.
- O campo *Time Stamp* é utilizado para identificar o campo responsável pela marcação de atualização do registro. Este campo só fica habilitado quando o tipo é igual a Inteiro ou Data.

- Leiaute de destino
 - Repositório de destino é o Metadiretório
 - Sempre será uma classe definida pelo EID
 - Scripts em Java ou Bean Shell podem ser usados para conversão de dados
 - Registros são atualizados pela chave na importação

Na aba *Leiaute de Destino* é definida a tabela que receberá os dados e o mapeamento dos campos da origem para os campos dessa tabela.

- ▲ O destino sempre será uma tabela previamente definida por uma classe do EID.
- ▲ É possível utilizar scripts de conversão mais sofisticados, escritos em Java ou Bean Shell, para transformação de dados de origem para o destino.
- ▲ No momento da importação, registros que já foram importados são identificados automaticamente. Existe a opção da atualização dos dados do registro importado.
- ▲ A atualização é feita com base no identificador único definido (IU).



No *Leiaute de Destino* o repositório selecionado deve ser sempre o Metadiretório.

A Tabela de Destino é a tabela que será alimentada. É necessário que a tabela da classe *Identificação* seja a primeira a ser alimentada, pois os demais dados serão vinculados às pessoas previamente importadas.

A opção *Atualizar registros existentes*, quando selecionada, promove a atualização do registro em questão, caso ele já exista na base de destino. Em caso da não seleção, o registro é descartado na reimportação.

No painel de configurações avançadas é possível definir um *Filtro de Conciliação*, um script Java que pode ser utilizado para consultar o banco de destino e optar pela importação, atualização ou descarte do registro.

O botão *Leiaute* constrói a lista de campos disponíveis na tabela de destino, assim como no leiaute de origem.

Figura 3.15
Leiaute de destino.

Leiaute de Destino									Novo
	* Nome	Descrição	* Tipo	Tam.	Dec.	Atualizável?	Campo Fonte	Script	
1	<input type="checkbox"/> cpf	cpf	Texto	255		<input checked="" type="checkbox"/>	Documento		Excluir
2	<input type="checkbox"/> dataNascimento	dataNascimento	Data			<input checked="" type="checkbox"/>	Nascimento		Excluir
3	<input type="checkbox"/> nomeCompleto	nomeCompleto	Texto	255		<input checked="" type="checkbox"/>	Nome		Excluir
4	<input type="checkbox"/> nomeMae	nomeMae	Texto	255		<input checked="" type="checkbox"/>	NomeMae		Excluir
5	<input type="checkbox"/> nomePai	nomePai	Texto	255		<input checked="" type="checkbox"/>	NomePai		Excluir
6	<input type="checkbox"/> nomeSolteiro	nomeSolteiro	Texto	255		<input checked="" type="checkbox"/>	Nome		Excluir
7	<input type="checkbox"/> sexo	sexo	Texto	255		<input checked="" type="checkbox"/>	Sexo		Excluir

Objeto referenciado

- Figura 3.16 Os pontos fundamentais no *Leiaute de destino* estão em *Campo Fonte* e *Script*.
- Leiaute de destino. O *Campo Fonte* é definido no *Leiaute de origem* e será mapeado diretamente para o campo de destino.
- O *Script* pode ser utilizado para um tratamento desse campo. Neste caso, o *Campo Fonte* deve ser deixado vazio.
- Podem existir diversas ETCs carregando a mesma classe, o que é de grande utilidade para carga de tabelas a partir de repositórios diferentes.

Leiaute de Destino									Novo
	* Nome	Descrição	* Tipo	Tam.	Dec.	Atualizável?	Campo Fonte	Script	
1	<input type="checkbox"/> codigoCapes	codigoCapes	Texto	255		<input checked="" type="checkbox"/>	CodInepCapes		Excluir
2	<input type="checkbox"/> codigoInep	codigoInep	Texto	255		<input checked="" type="checkbox"/>	CodInepCapes		Excluir
3	<input type="checkbox"/> matricula	matricula	Texto	255		<input checked="" type="checkbox"/>	CodDiscente		Excluir
4	<input type="checkbox"/> nivelCurso	nivelCurso	Texto	255		<input checked="" type="checkbox"/>	Nivel		Excluir
5	<input type="checkbox"/> nomeCurso	nomeCurso	Texto	255		<input checked="" type="checkbox"/>	Nome		Excluir

Objeto referenciado

6	<input type="text"/> eid_object_guid	<input type="text"/> eid_object_guid	Texto	21		<input type="checkbox"/>	ETC Identificação		<input type="text"/> CodPessoa	
---	--------------------------------------	--------------------------------------	-------	----	--	--------------------------	-------------------	--	--------------------------------	--

- Figura 3.17 Para a extração de todas as classes no *Leiaute de Destino*, excetuando-se *Identificação*, deve ser informado o GUID do objeto referenciado no painel *Objeto referenciado*.
- Leiaute de destino. O *Campo Fonte* (resultado do script) deve resolver o valor que foi utilizado como IU para a classe *Identificação*. Outra possibilidade é resolver diretamente o GUID do objeto.

Processos

- Processos definem:
 - ▲ Conjunto de extrações a serem executadas
 - ▲ Ordem da execução
 - ▲ Outras configurações mais detalhadas

Depois de definidas as extrações (ETCs), é necessário associar a ETC a um processo e agendar a execução do mesmo.

Processos são agrupamentos de ETCs executadas juntas, isto é, em um mesmo agendamento. As ETCs em um mesmo processo são executadas de forma sequencial, em uma ordem definida no processo.

The screenshot shows a web-based application for managing processes. At the top, there's a logo with a green 'e' and 'id' text, followed by the title 'export import directory tool'. Below the title is a navigation bar with links: 'EID', 'Configuração', 'Processamento', 'Administração', and 'Sobre'. To the right of the navigation bar are buttons for 'Novo', 'Excluir', 'Pesquisar', and 'Imprimir'. A search bar labeled 'Pesquisa de Processo' is present. The main area is divided into sections: 'Argumentos da Pesquisa' (Search Arguments) with fields for 'Nome:' and 'Descrição:', and 'Resultados da Pesquisa' (Search Results). The results section displays one item found, with columns for 'Número de itens encontrados: 1', 'Processo', 'Descrição', 'Visualizar', and 'Alterar'. The first result row contains a checkbox, the name 'Processo Eid', and icons for 'Visualizar' and 'Alterar'. At the bottom of the page, there are logos for 'Educação Ministério da Educação' and 'RNP'.

Figura 3.18
Tela de
administração
de processos.

A figura 3.18 apresenta a tela de administração de processos, que pode ser acessada pelo menu *Configuração/Processos*. Nela é possível visualizar os processos cadastrados no sistema, alterá-los ou ainda cadastrar um novo processo.

Figura 3.19
Tela de cadastro

Inclusão de Processo

Processo

* Nome: Processo Identificação

Descrição:

* Modo: Não Interromper Interromper Processamento

* N° Tentativas: 1 * Intervalo entre Tentativas: 1 Minutos

Itens de Processo

	ETC	* Intervalo de Commit	* Máximo de Erros	* Ordem	Mudar Ordem
1	ETC Identificação	500	0	1	↓
2	ETC Usuários CSV	500	0	2	↑

Acionando o botão *Novo* a tela de cadastro de processo é exibida (ver figura 3.19).

- Um nome deve ser informado para o processo.
- A opção *Modo* indica a ação que deve ser tomada caso alguma das ETCs listadas não seja finalizada com sucesso. Se for escolhido *Interromper*, as ETCs seguintes à causadora do erro não são processadas. No caso de *Não interromper*, as ETCs seguintes são processadas, independentemente de haver erro.
- *Número de tentativas* indica o número máximo de vezes que o sistema tentará estabelecer conexão com os repositórios utilizados em cada extração antes de abortar o processamento.
- *Intervalo entre tentativas* indica o tempo de espera entre duas tentativas sucessivas.
- As ETCs devem ser especificadas no painel *Itens do processo*. Acessando o botão *Novo* uma janela pop-up é exibida com as ETCs disponíveis para o cadastro. Deve-se selecionar as ETCs clicando no check box e acionar o botão *Selecionar*, então o pop-up é fechado e as ETCs são inseridas no painel *Itens de processo*.
- O *Intervalo de commit* indica o número de registros inseridos em cada transação. Um número muito alto pode sobrecarregar o banco (muitos registros para *commit* no log), enquanto que um número muito baixo pode comprometer a performance; 500 é um número razoável, que pode ser ajustado de acordo com o banco utilizado e a capacidade da máquina.
- *Máximo de erros* determina o número máximo de erros que a ETC suporta sem ser abortada e, consequentemente, finalizar seu processamento com código de erro. Esta opção é interessante, pois é sabido que existem inconsistências no banco de origem e que os registros que geram inconsistências devem ser descartados.
- *Ordem* indica a ordem de processamento das ETCs no processo. É possível alterar a ordem clicando nas setinhas disponíveis para cima ou para baixo.

Agendamentos

- Definem para o processo:
 - ▲ Horário de importação
 - ▲ Frequência de repetição

Uma vez definido o processo, ele deve ser agendado. Só pode existir um agendamento para cada processo; não é aconselhável que uma mesma ETC participe de processos distintos que possam rodar em paralelo. Um agendamento de processo definirá o horário para executar a importação e sua frequência de repetição.

	Processo	Situação	Próxima Execução	Número de Processamento	Visualizar	Alterar
<input type="checkbox"/>	Processo Eid	Aguardando	03/05/2011	1		

Figura 3.20
Tela de
agendamentos.

A tela de agendamentos pode ser acessada pelo menu *Processamento/Agendamento* (ver figura 3.20). Cada novo agendamento de um processo ganha um *Número de Processamento*. É com este número que o usuário terá o controle do número de vezes que o agendamento foi executado e acompanhar o resultado do processamento na tela *Resultado de Processamento*.

Figura 3.21
Tela de cadastro
de agendamento
de processo.

Ao acionar o comando *Novo*, é exibida a tela de *Cadastro de Agendamento de Processo* (ver figura 3.21). Nesta tela deve-se escolher o processo a ser agendado, o tipo de repetição, e também definir a partir de qual item (ETC) o processo deverá iniciar e terminar, e a data e hora da próxima execução. A caixa *Processar Agora* pode ser marcada caso o processamento deva ter início imediato.

O Campo *Resultado de Processamento* exibe o resultado do processamento agendado através de uma janela pop-up. Em casos onde a fonte de dados é muito demandada por outras aplicações, pode-se definir no painel *Horários permitidos para processamento* os horários nos quais a importação é permitida, bastando informar os intervalos de início e fim.

Pesquisa de Resultado de Processamento

Argumentos da Pesquisa

Processo: Data Início: Excluir Pesquisar Imprimir

Resultados da Pesquisa

Permitir auto-recarga da página
Número de itens encontrados: 1

Coleta	Processo de Importação	Organização Fonte	Data Início	Data de Término	Situação	Número de Processamento	Visualizar
	Processo Eid		03/05/2011 15:40:59	03/05/2011 15:41:02	Erros Encontrados	1	

Educação Ministério da Educação ... RNP ...

Figura 3.22
Tela de resultado de processamento.

Após a execução de um processo é possível visualizar seu resultado de processamento. A tela *Resultado de Processamento* é acessada através do menu *Processamento/Resultado de Processamento* (ver figura 3.22). Para facilitar a busca pelos registros, pode-se filtrar os resultados de processamentos tanto pelos processos de interesse quanto pela data de execução. No caso de não serem especificados esses parâmetros, todos os resultados são exibidos.

Figura 3.23
Visualização de resultado de processamento.

Visualização de Resultado de Processamento

Processo

Processo: Processo Eid
Data Início: 03/05/2011 15:40:59
Data de Término: 03/05/2011 15:41:02
Situação: Erros Encontrados
Número de Processamento: 1
Organização Fonte:

Permitir auto-recarga da página

03/05/2011 15:40:59 >> Iniciando o processo
03/05/2011 15:40:59 Iniciando processo de exclusão de dados.
03/05/2011 15:41:00 O item ETC Usuários CSV não configurado para exclusão no processo. Nada a excluir.
03/05/2011 15:41:00 O item ETC Alunos não configurado para exclusão no processo. Nada a excluir.
03/05/2011 15:41:00 O item ETC Identificação não configurado para exclusão no processo. Nada a excluir.
03/05/2011 15:41:00 Finalizado o processo de exclusão de dados.
03/05/2011 15:41:00 Iniciando a importação de registros.
03/05/2011 15:41:00 Iniciando processo de importação de dados para o item ETC Identificação
03/05/2011 15:41:01 Total de registros no repositório de origem: 5.
03/05/2011 15:41:01 INFO - 5 registros processados, sendo 5 registros inseridos, 0 registros conciliados, 0 registros descartados, 0 registros atualizados, 0 erros de inclusão e 0 erros de atualização, a 18 Reg/seg. Tempo restante: 0h 0m 0s.
03/05/2011 15:41:01 Item ETC Identificação finalizado.
03/05/2011 15:41:01 Iniciando processo de importação de dados para o item ETC Alunos
03/05/2011 15:41:01 Total de registros no repositório de origem: 10000.
03/05/2011 15:41:01 Erro lendo origem de dados: Tipo incorreto no layout, campo AnoIngresso. No layout está Inteiro, mas no repositório é java.lang.String..
03/05/2011 15:41:01 Item ETC Alunos finalizado.

Educação Ministério da Educação ... RNP ...

Ao acionar o botão *Visualizar* na tela de *Pesquisa de Resultados de Processamentos*, é exibida a tela mostrada na figura 3.23. Por esta interface é possível observar detalhes do processamento, incluindo mensagens de erro durante a importação, o que permite a identificação de registros causadores de problemas e de ETCs configuradas incorretamente.

Por padrão, durante a importação a tela é recarregada automaticamente, apresentando o seu progresso.



3

Roteiro de Atividades Construindo metadiretórios com EID

Tópicos e conceitos

- Visão geral
 - ▲ Metadiretórios
 - ▲ EID
 - ▲ EID e brEduPerson
- Configuração de extrações
 - ▲ Definição de repositórios
 - ▲ Definição de extrações
 - ▲ Definição de processos
- Agendamento de processos

Competências técnicas desenvolvidas

- Criação de extrações de bases de dados relacionais e alimentação do metadiretório.

Tempo previsto para as atividades

- 40 – 60 minutos

Servidor de sala de aula

- Os arquivos para instalação do sistema encontram-se na máquina virtual presente na área de trabalho, pasta */opt/treinamento*



Atividade 1 – Instalação do EID e EID2LDAP

Instale EID e EID2LDAP na máquina virtual presente em sua estação de trabalho.

Conecte-se com o SSH na VM, que já possui instalados o Tomcat, Java e MySQL. Faremos a configuração necessária para instalar o EID.

Configurações no Tomcat:

1. Para desabilitar a execução segura do Tomcat, deve-se editar o arquivo `/etc/default/tomcat6`. Dentro deste arquivo faça a seguinte alteração na linha que contém `#TOMCAT6_SECURITY=yes` para:

```
TOMCAT6_SECURITY=no
```

Ainda no mesmo arquivo acrescente a seguinte linha:

```
JAVA_OPTS="-XX:MaxPermSize=512M -Xmx512M -Duser.timezone=America/Sao_Paulo -Duser.language=pt -Duser.country=BR -Djava.library.path=$JARO_WINKLER_DIR -Dfile.encoding=UTF-8"
```

2. Os drivers para conexão com os bancos de dados se encontram em `/opt/treinamento`. Copie os drivers de banco para `/usr/share/java/` com o comando a seguir:

```
cp /opt/treinamento/mysql-connector-java-3.1.8-bin.jar /usr/share/java
```

```
cp /opt/treinamento/usr/share/java/postgresql-8.4-702.jdbc3.jar /usr/share/Java
```

3. É necessário ainda fazer a criação de alguns links simbólicos para o conector.

Para tanto, execute as linhas de comando a seguir:

```
ln -sf /usr/share/java/mysql-connector-java-3.1.8-bin.jar /usr/share/tomcat6/lib/
```

```
ln -sf /usr/share/java/mysql-connector-java-3.1.8-bin.jar /var/lib/tomcat6/lib/
```

```
ln -sf /usr/share/java/postgresql-8.4-702.jdbc3.jar /var/lib/tomcat6/lib/
```

```
ln -sf /usr/share/java/postgresql-8.4-702.jdbc3.jar /usr/share/tomcat6/lib/
```

4. Sabendo-se que a instalação padrão do Tomcat via `apt-get` não possui o arquivo `tomcat-dbcp.jar` (necessário para algumas aplicações), deve-se baixá-lo e colocá-lo na pasta `lib` do Tomcat. Para tanto, execute o seguinte comando:

```
wget -t 3 -T 10 -c http://pacotes.ufrrgs.br/ubuntu/hardy/installacao-java-tomcat/tomcat-dbcp.jar -O /usr/share/tomcat6/lib/tomcat-dbcp.jar
```



- Para permitir que um usuário do Tomcat faça login no EID, edite o arquivo `/etc/tomcat6/tomcat-users.xml`, deixando-o como está abaixo. Substitua `{SENHA_EID}` pela senha que será usada ao logar no EID.

```
<tomcat-users>
    <role rolename="manager"/>
    <user username="eid" password="{SENHA_EID}" roles="manager"/>
</tomcat-users>
```

- Inicialize o Tomcat através do seguinte comando:

```
/etc/init.d/tomcat6 start
```

- Por fim, para testar se o mesmo está funcionando corretamente, através do browser, acesse o endereço `http://ip_do_servidor:8080` e verifique se a mensagem “It works!” é exibida.

Configurações de banco de dados no MySQL:

- É necessário fazer a criação das bases de dados que serão utilizadas pelo EID e EID2LDAP. As informações são armazenadas em bases MySQL. Para criar as bases execute a linha de comando a seguir:

```
echo "create database eid; create database pcollecta; create database eid2ldap" | mysql -uroot -proot
```

- Os arquivos para popular as bases estão disponíveis na pasta `/opt/treinamento`. Faça a carga no banco de dados através dos comandos a seguir:

```
mysql -uroot -proot eid < /opt/treinamento/eid-1.3.0.sql
mysql -uroot -proot pcollecta < /opt/treinamento/pcollecta-1.3.0.sql
mysql -uroot -proot eid2ldap < /opt/treinamento/eid2ldap-1.1.1.sql
```

Instalando o EID e EID2LDAP:

- Na pasta `/opt/treinamento` estão os arquivos WAR da versão mais recente do EID (1.3.0) e do EID2LDAP (1.1.1). Execute os comandos a seguir para criar os diretórios, e descompacte os arquivos WAR dentro deles. O EID se encontra disponível também no Sourceforge: <http://sourceforge.net/projects/eid/files/>

```
mkdir /opt/eid-1.3.0/
unzip /opt/treinamento/eid.war -d /opt/eid-1.3.0
mkdir /opt/eid2ldap-1.1.1/
unzip /opt/treinamento/eid2ldap.war -d /opt/eid2ldap-1.1.1/
```



- Crie a variável de ambiente JARO_WINKLER_DIR através do comando a seguir:

```
export JARO_WINKLER_DIR=/opt/eid-1.3.0/lib  
export JAVA_HOME="/usr/lib/jvm/java-6-sun"  
export JRE_HOME="/usr/lib/jvm/java-6-sun"  
export CATALINA_HOME="/usr/share/tomcat6"  
export TOMCAT_HOME="/usr/share/tomcat6"  
echo 'export JARO_WINKLER_DIR="/opt/eid-1.3.0/lib"' >> /etc/  
profile
```

- Crie a pasta referenciada pela variável de ambiente JARO_WINKLER_DIR através do comando a seguir:

```
mkdir -p $JARO_WINKLER_DIR
```

- Para proceder à compilação do algoritmo JARO WINKLER, execute as linhas de comando a seguir:

```
cd /opt/eid-1.3.0/WEB-INF/classes/br/ufmg/lcc/eid/model/  
conciliator  
make compile
```

- Copie os arquivos *eid.xml* e *eid2dap.xml* para */etc/tomcat6/Catalina/localhost/* com os comandos abaixo:

```
cp /opt/treinamento/eid.xml /etc/tomcat6/Catalina/localhost  
cp /opt/treinamento/eid2dap.xml /etc/tomcat6/Catalina/localhost
```

Atribua as respectivas permissões à pasta do EID e reinicie o Tomcat através dos comandos abaixo:

```
chown -R tomcat6:tomcat6 /opt/eid-1.3.0/  
/etc/init.d/tomcat6 restart
```

- Acesse a aplicação através do browser: http://IP_VM:8080/eid. Logue com o user "eid" e a senha informada no arquivo *tomcat-users.xml* (Passo 5: configurando o Tomcat). Acesse o menu Configuração > Repositório de dados e defina o usuário e a senha do repositório Metadiretório (user: *root* e senha: *root*). Em seguida teste a conexão com o banco de dados através do botão *Testar conexão*.
- Acompanhe os logs em */var/log/tomcat6/catalina.{DATA_ATUAL}.log* e */var/log/tomcat6/localhost.{DATA_ATUAL}.log*.

Atividade 2 – Configuração de um repositório

Configure um repositório do tipo Banco de Dados Relacional no EID (servirá como fonte de dados). O banco se encontra no servidor.

1. Acesse o menu *Configuração/Repositório de Dados*;
2. Acione o comando *Novo* para definir um novo repositório;
3. Escolha o tipo do Repositório como Banco de Dados Relacional.
4. Forneça os campos necessários:
 - 4.1. *Nome*: Repositório Acadêmico
 - 4.2. *Descrição*: Repositório de testes do curso EID
 - 4.3. *URL*: Ao clicar no ícone ao lado do campo uma janela pop-up é exibida com exemplos de URLs e drivers.
→ **USAR Banco Mysql**: jdbc:mysql://localhost:3306/acadêmico
Banco Postgresql: jdbc:postgresql://servidor:5432/academico (onde *servidor* deve ser substituído pelo IP da máquina onde o banco está instalado).
 - 4.4. Driver:
→ **USAR Banco Mysql**: com.mysql.jdbc.Driver
Banco Postgresql: org.postgresql.Driver
 - 4.5. *Usuário*: root
 - 4.6. *Senha*: root
 - 4.7. No painel *Versão do Banco de Dados*, insira o valor 1.0 no campo *Versão* (manual).
5. Acione o comando *Testar Conexão*;
6. Acione o comando *Salvar*.

Atividade 3 – Definição de uma extração

Crie uma extração para retirar informações da tabela *Pessoas* e alimentar a classe *Identificação*.

1. Acesse o menu *Configuração/ETC*;
2. Acione o comando *Novo*;
3. Na guia *ETC*, especifique:
 - 3.1. *Nome*: Extração de pessoas do sistema acadêmico
 - 3.2. *Descrição*: Extração de dados de pessoas a partir do sistema acadêmico

4. Na guia *Leiaute de Origem*:
 - 4.1. *Repositório*: Repositório Acadêmico
 - 4.2. *SQL*: `select * from Pessoas`
 - 4.3. Acione o comando *Leiaute*
 - 4.4. Selecione o campo *Id* como identificador único (IU)
 - 4.5. Selecione o campo *data_atualizacao* como TimeStamp
5. Na guia *Leiaute de Destino*:
 - 5.1. *Tipo de Script*: Bean Shell
 - 5.2. *Tabela de Destino*: Identificação
 - 5.3. Acione o comando *Leiaute*
 - 5.4. *Atualizar Registros Existentes*: marcar a caixa
 - 5.5. No painel *Leiaute de Destino dos Dados*:
 - Mapeie os campos de origem para o destino
 - Marque para remoção os campos que não serão mapeados
 - 5.6. Acione o comando *Salvar*.

Atividade 4 – Definição de um processo e seu agendamento

Crie um processo que inclua a extração definida anteriormente e o agende para ser executado de imediato, sem repetições.

1. Acesse o menu *Configuração/Processo*;
2. Acione o comando *Novo*;
3. Preencha os campos:
 - 3.1. *Nome*: Processo de extração Acadêmico
 - 3.2. *Descrição*: Processo de extração de dados do sistema acadêmico
 - 3.3. *Modo*: selecione *Interromper Processamento*
 - 3.4. *Número de tentativas*: 1
 - 3.5. *Intervalo entre tentativas*: 1
 - 3.6. No painel *Itens de processo*:
 - Acione o botão *Novo* e selecione a ETC: Extração de pessoas do sistema acadêmico, e em seguida clique no botão *Selecionar*
 - *Intervalo commit*: 500
 - *Número de erros*: 0

4. Acione *Salvar*
5. Acesse o menu *Processamento/Agendamento*
6. Acione o comando *Novo*
7. Selecione:
 - 7.1. *Processo:* Processo de extração Acadêmico
 - 7.2. *Tipo de repetição:* Nenhum
 - 7.3. *Item de início:* Extração de pessoas do sistema acadêmico
 - 7.4. *Finalizar no item:* Extração de pessoas do sistema acadêmico
 - 7.5. *Próxima execução:* marcar *Processar agora*
8. Acione o comando *Salvar*
9. Observe o resultado acessando o ícone *Resultado de processamento* ou o menu *Processamento/Resultado de processamento*.
10. Depois de alguns minutos acesse o menu *EID/Gestão e pessoas* para visualizar as pessoas importadas.

Atividade 5 – Limpar o repositório EID

Faça a limpeza dos dados de todas as tabelas do banco EID.

1. Abra o phpMyAdmin acessando o endereço: http://IP_Servidor/phpmyadmin (onde **IP_Servidor** deve ser substituído pelo IP da máquina onde o EID foi instalado).
2. Informe usuário/senha do MySQL: root/root.
3. No canto superior esquerdo da tela, selecione o banco do EID.
4. Clique na aba SQL e cole o seguinte SQL:

```
DELETE FROM eid.TBL_SVC_ALUNO;  
DELETE FROM eid.TBL_SVC_CONTA;  
DELETE FROM eid.TBL_SVC_EMAIL;  
DELETE FROM eid.TBL_SVC_ENDERECO;  
DELETE FROM eid.TBL_SVC_PROFESSOR;  
DELETE FROM eid.TBL_SVC_TECNICO;  
DELETE FROM eid.TBL_SVC_TELEFONE;  
DELETE FROM eid.TBL_SVC_GRUPO;  
DELETE FROM eid.TBL_SVC_IDENTIFICACAO;
```

```
DELETE FROM eid.TBL_EID_CLASS;  
DELETE FROM eid.TBL_MAPPING;  
DELETE FROM eid.TBL_MATCH;  
DELETE FROM eid.TBL_EID_OBJECT;  
DELETE FROM eid.TBL_EXTERNAL_SOURCE;  
DELETE FROM pcollecta.PC_KEY_MAPPING;  
UPDATE `pcollecta`.`PC_ETL` SET `FINAL_TIME_STAMP` = null;  
UPDATE `pcollecta`.`PC_ETL` SET `INITIAL_TIME_STAMP` = null;
```

5. Clique no botão *Executar*.

Atividade 6 – Reagendar o processo de carga da classe *Identificação*

Altere o *Processo de extração Acadêmico* para ser executado novamente.

1. Acesse o menu *Processamento/Agendamento*;
2. Acione o comando *Alterar* para o agendamento do *Processo de extração Acadêmico*;
3. Próxima execução: marcar *Processar agora*;
4. Acionar o comando *Salvar*;
5. Observe o resultado acessando o menu *Processamento/Resultados de processamento*.

4

Criando extrações no EID

Sumário

- Extração de arquivos texto
- Resolução de objeto EID
- Parâmetros globais
- Importação incremental
- Scripts de conversão
- Algoritmos de unificação
- Web services
- Problemas comuns

Extração de arquivos texto

- EID importa arquivos CSV (Comma-Separated Value)
 - ▲ Informações complementares não mantidas em bancos de dados

A quarta sessão do curso apresentará algumas funcionalidades avançadas que podem ser utilizadas na configuração de extrações, entre elas: extrações de arquivos texto, uso de parâmetros globais, importação incremental e uso de scripts de conversão.

O EID é capaz de importar dados de arquivos CSV, além de bancos de dados relacionais. Arquivos CSV são arquivos separados por ponto-e-vírgula ou tabulação (o Excel exporta arquivos neste formato). Em algumas situações este recurso é útil, principalmente em casos onde a informação é mantida em planilhas externas aos sistemas utilizados na organização.





Figura 4.1
Repositório.

Para realizar uma extração em um arquivo CSV é necessário cadastrar um repositório do tipo arquivo de texto CSV e informar no campo *Diretório* o caminho onde os arquivos se encontram. No caso de servidores Linux/Unix, o caminho do diretório é *case sensitive*. Outro ponto que deve ser salientado é que o repositório é um local que possui vários conjuntos de dados, portanto não insira neste campo o diretório seguido pelo nome do arquivo, mas somente o diretório. O nome do arquivo será definido na configuração da extração mais adiante. É importante lembrar também que este diretório se refere a um local na máquina que executa o EID, e que o usuário com o qual o Tomcat foi iniciado deve possuir acesso de leitura ao diretório e aos arquivos que serão importados.

- **Leiaute de origem**
 - Arquivo de Origem define o nome do arquivo
 - Permite a seleção da codificação
 - Configuração de separador decimal e separador de campos
 - Não cria leiaute automático

Para extrações de arquivos texto, o campo *Arquivo de Origem* deve ser preenchido com o nome do arquivo do qual os dados serão extraídos. Em servidores Linux/Unix o nome é *case sensitive*.

É possível escolher a codificação de caracteres do arquivo original, lembrando sempre que a codificação do banco do EID é ISO-8859-1. A escolha correta é de suma importância para a interpretação correta dos caracteres acentuados.

O *Separador Decimal* indica o caractere utilizado para separar casas decimais (vírgula ou ponto).

O *Separador Campos* indica o caractere utilizado para separar as colunas do arquivo, podendo ser ponto-e-vírgula, vírgula, barra vertical (|), sustenido (#) ou tabulação.

O campo *Formato da Data* deve descrever o formato das datas no arquivo.

Figura 4.2
ETC: Leiaute de origem (arquivo CSV).

#	* Nome	Descrição	* Tipo	IU	Time Stamp
1	id	id	Inteiro	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2	login	login	Texto	<input type="checkbox"/>	<input type="checkbox"/>
3	senha	senha	Texto	<input type="checkbox"/>	<input type="checkbox"/>

A figura 4.2 mostra o leiaute de origem de uma extração em um repositório do tipo arquivo de texto CSV. Quando estamos definindo uma extração para arquivos CSV, diferentemente de bancos relacionais, não se pode definir um SQL e nem é possível a construção automática do leiaute, uma vez que não existem metadados que descrevem os campos.

Os nomes de cada campo devem ser informados manualmente, não podendo haver espaço entre palavras, sempre na forma de caracteres ASCII de a-z, A-Z ou 0-9 (exceto no início do identificador), sem acentuação.

O tipo dos campos é sempre texto, podendo ser convertido para os tipos corretos no momento da configuração do destino, com o uso dos *scripts de mapeamento* ou *conversão* que serão explicados adiante. Também aqui é possível determinar um identificador único, utilizado na conciliação automática.

Caso o arquivo possua algum campo indicador da data de atualização dos registros é possível utilizar a importação incremental, através do campo *Time Stamp*.

O comando *Novo*, no painel de *Leiaute de Origem de Dados*, adiciona novas linhas nesse painel para configuração de novas colunas.

- Leiaute de destino
 - Idêntico ao de bancos de dados relacionais
 - Conversão de tipos feita por scripts Java ou Bean Shell



O leiaute de destino continua sendo feito da mesma forma que em bancos relacionais.

- Os tipos podem ser convertidos via script de mapeamento.

Resolução de objetos

- Objetos vinculados via GUID
- Importação de *Identificação* cria os objetos
- Instâncias de novas classes devem resolver o GUID
- EID possibilita resolução automática

Ainda no leiaute de destino, o campo *eid_object_guid* deve indicar o GUID para vinculação da instância da classe com o objeto.

A importação da classe *Identificação* promove a criação de novos objetos no metadiretório; em geral ela é utilizada como referência para a vinculação de instâncias de outras classes ao objeto criado.

A vinculação é feita por um mapeamento, como exemplificado a seguir:

- Importando-se um registro de *Identificação* da pessoa X da origem, é gerado um mapeamento da chave primária escolhida para a extração de X para o GUID do objeto criado no metadiretório;
- Na base de origem, os demais dados da pessoa X (endereço, dados de aluno etc.) certamente possuirão algum tipo de relacionamento com o registro de identificação, podendo fazer parte do registro na mesma tabela ou fazer uma referência a ele via chave estrangeira (fk);
- Considerando ser Y um registro com dados referentes a X, no momento da importação devemos indicar para o sistema a qual objeto ele deve ser associado. Isto é feito indicando-se a extração que carregou X e o campo da chave estrangeira que relaciona Y com X na origem. Com base nesse campo, o EID é capaz de consultar o mapeamento e descobrir o GUID do objeto ao qual Y deve ser relacionado.

Figura 4.3
Leiaute de destino.

Leiaute de Destino								Novo
	 * Nome	Descrição	* Tipo	Tam.	Dec.	Atualizável?	Campo Fonte	Script
1	<input type="text" value="codigoCapes"/>	<input type="text" value="codigoCapes"/>	Texto ▾	255		<input checked="" type="checkbox"/>	CodInepCapes ▾	
2	<input type="text" value="codigoInep"/>	<input type="text" value="codigoInep"/>	Texto ▾	255		<input checked="" type="checkbox"/>	CodInepCapes ▾	
3	<input type="text" value="matricula"/>	<input type="text" value="matricula"/>	Texto ▾	255		<input checked="" type="checkbox"/>	CodDiscente ▾	
4	<input type="text" value="nivelCurso"/>	<input type="text" value="nivelCurso"/>	Texto ▾	255		<input checked="" type="checkbox"/>	Nivel ▾	
5	<input type="text" value="nomeCurso"/>	<input type="text" value="nomeCurso"/>	Texto ▾	255		<input checked="" type="checkbox"/>	Nome ▾	
Objeto referenciado								
6	<input type="text" value="eid_object_guid"/>	<input type="text" value="eid_object_guid"/>	Texto ▾	21		<input type="checkbox"/>	ETC Identificação ▾	CodPessoa ▾

A figura 4.3 apresenta o leiaute de destino dos dados, onde o objeto referenciado deve ser informado. Para todas as classes sua informação é obrigatória, com exceção da classe *Identificação*.

Parâmetros globais

- ▲ Constantes
 - ▲ Consultas
 - ▲ Script de conciliação
 - ▲ Script de conversão

Outra funcionalidade a ser explorada nas extrações está relacionada aos parâmetros globais. Parâmetro global é um mecanismo utilizado pelo EID para definição de constantes que podem ser utilizadas nas extrações.

- ▲ Utilizado como constante em consultas, scripts de conciliação ou scripts de mapeamento.

Figura 4.4
Administração de parâmetros globais.

	* Nome	* Valor	Alterar
1	<input type="text" value="DataInicial"/>	<input type="text" value="01/01/2011"/>	
2	<input type="text" value="Valor"/>	<input type="text" value="10"/>	

Os parâmetros globais são definidos no menu *Configuração/Parâmetros Globais* (ver figura 4.4). Todos os parâmetros devem ter um nome e um valor. O nome serve como identificador, não podendo haver, portanto, espaço entre as palavras ou caracteres especiais.

Os parâmetros são sempre tratados como sendo do tipo *string*. Estes parâmetros funcionam por substituição; nos pontos onde são referenciados, seu valor é inserido antes do início do processamento sempre com a sintaxe `#{nome_do_parâmetro}`.

Vale a pena lembrar que a substituição é direta. Assim, nos casos onde o parâmetro é tratado como valor numérico, basta colocar `#{nome_do_parâmetro}` e, onde é tratado como string, as aspas (simples ou duplas, dependendo do caso) devem ser utilizadas, como em `'#{nome_do_parâmetro}'`.



Abaixo um exemplo de consulta que utiliza parâmetros globais, considerando que o banco realiza automaticamente a conversão de string para data:

```
Select * from Pessoas where dataNascimento >= '#{DataInicial}'
and dataNascimento <= '#{DataFinal}'
```

Importação incremental

- Reimportação com atualização de registros implica em reconciliação e é:
 - ▀ Computacionalmente cara
 - ▀ Desnecessária em casos onde o registro não foi alterado
- A importação incremental minimiza o problema

O metadiretório deve refletir o dinamismo da organização. Isso implica a importação de dados não importados anteriormente e também a atualização de outros já importados.

Uma forma de se fazer este processo é selecionar a opção *Atualizar registros existentes* na definição da ETC, que força com que todos os registros importados anteriormente sejam atualizados em uma reimportação. Novos registros são inseridos naturalmente.

A consequência da atualização de todos os registros é que o EID é obrigado a trabalhar novamente sobre todos os objetos afetados, pois não é possível saber, a priori, se o registro sofreu alterações na origem ou não. Isto pode ser melhorado com o uso de importações incrementais, que podem alterar o escopo das consultas a cada execução, desde que haja alguma informação no banco de origem que permita a distinção dos registros alterados dos não alterados (uma coluna com carimbo de tempo, por exemplo).

	* Nome	Descrição	* Tipo	IU	Time Stamp
1	<input type="checkbox"/> Id	Id	Inteiro	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2	<input type="checkbox"/> Nome	Nome	Texto	<input type="checkbox"/>	<input type="checkbox"/>
3	<input type="checkbox"/> Sexo	Sexo	Texto	<input type="checkbox"/>	<input type="checkbox"/>
4	<input type="checkbox"/> Nascimento	Nascimento	Data	<input type="checkbox"/>	<input type="checkbox"/>
5	<input type="checkbox"/> Documento	Documento	Texto	<input type="checkbox"/>	<input type="checkbox"/>
6	<input type="checkbox"/> TipoDocumento	TipoDocumento	Texto	<input type="checkbox"/>	<input type="checkbox"/>
7	<input type="checkbox"/> NomePai	NomePai	Texto	<input type="checkbox"/>	<input type="checkbox"/>
8	<input type="checkbox"/> data_atualizacao	Time Stamp	Data	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Figura 4.5
Time Stamp
no leiaute
de origem.

Para utilizar importação incremental automática do EID é necessário que a base de origem tenha algum campo que funcione como carimbo de tempo dos registros atualizados.

No leiaute de origem da ETC este campo deve ser identificado com a marcação de Time Stamp, como na figura 4.5. O EID armazenará internamente o maior valor já importado e sempre que for executar novamente a ETC irá atualizar ou inserir apenas os registros alterados ou novos.

A opção *Reiniciar Importação Incremental*, quando marcada, zera todos os campos de importação incremental da ETC, fazendo com que, na próxima execução, todos os registros sejam importados do repositório de origem.

Além da importação incremental automática, o campo *Time Stamp* pode ser útil em processos que agruparão ETCs dependentes. Quando o campo *Time Stamp* é marcado, o sistema cria duas variáveis internas: *InitialTimeStamp* e *FinalTimeStamp*. Essas variáveis poderão ser acessadas pelo usuário no SQL de origem de outra ETC como parâmetro de consulta, limitando os registros selecionados na base de origem. Para ter acesso a essas variáveis, deve-se usar a seguinte sintaxe: `#{{ETL.NOME_ETL.INITIAL_OU_FINAL}}`. Exemplo:

```
SELECT a.*  
FROM aluno a, pessoas p  
WHERE a.idPessoa = p.id and  
p.DATA_ATUALIZACAO_REGISTRO > #{{ETL.Etc de pessoas.INITIAL}}  
AND p.DATA_ATUALIZACAO_REGISTRO < #{{ETL.Etc de pessoas.FINAL}};
```

Neste exemplo, o SQL parametrizado impede que sejam selecionados registros de alunos referenciando pessoas ainda não carregadas pela ETC pessoas.

Script de conversão

- Possibilita o tratamento do dado da origem
- Complementa as possibilidades do SQL
- Código Java
- Campo *Origem* é disponibilizado como variável
- Campo *Fonte* tem prioridade sobre o script

Outra funcionalidade a ser explorada na configuração de uma ETC é o *script de conversão* ou *mapeamento*. Cada campo do *Leiaute de origem* pode ser tratado antes de ser inserido no destino. Este tratamento é feito via código Java, onde pode ser utilizada toda sua funcionalidade (como expressões regulares, tratamento de datas etc.). Para utilização de script, o *Campo Fonte* no *Leiaute de destino* deve ser deixado em branco; caso seja preenchido, o script não será executado e o valor do *Campo Fonte* será atribuído ao registro. Pode-se optar por dois tipos de scripts:



Bean Shell ou Jana Nativo. O código do script deve ser inserido acionando-se o comando *Script* na linha equivalente ao campo.

Caso a escolha seja Bean Shell, deve ser implementado o método com assinatura *public void execute()*, onde o valor calculado deve ser colocado na variável *result*, e o acesso às variáveis do leiaute do origem feito apenas pelo seu nome. Caso seja utilizado Java nativo, não é necessário utilizar o método, sendo o acesso às variáveis do leiaute de origem feito de forma idêntica ao acesso aos parâmetros globais: *# {nome_variável}*.

Script de conversão – Bean Shell

```
String result = null;
public void execute() {
    if (senha != null) {
        result = senha.substring(1, 4);
    } else{
        result=null;
    }
}
```

Um exemplo de script para atribuir o valor de uma substring ao campo *Senha* em Bean Shell é apresentado a seguir. Para acessar as variáveis do leiaute de origem apenas utilize o nome. É necessário utilizar o método *execute()*.

```
String result;
public void execute() {
    result = null;
    if (senha != null) {
        result = senha.substring(1, 4);
    }
}
```

O código exemplifica como pegar apenas parte da string *senha* do repositório de origem para ser o valor atribuído ao resultado no Metadiretório EID. Assim como este script, vários outros podem ser desenvolvidos de acordo com a necessidade de transformação dos dados. Alguns destes podem ser encontrados na seção FAQ do Wiki da Federação CAFé.



Script de conversão – Java Nativo

```
if (#{$senha} != null){
    result =#${senha}.substring(1, 4);
} else{
    result=null;
}
```

O mesmo exemplo de script pode ser visto em Java Nativo, em que para acessar as variáveis do leiaute de origem é necessário usar `#${nome_variável}` (não se usa o método `execute()`).

Algoritmos de unificação

- Critérios de mesclagem de instâncias
- Dois algoritmos pré-definidos
 - Instância única
 - Múltiplas instâncias
- Cada classe pode ter seu próprio algoritmo
- Algoritmos próprios devem ser adicionados à aplicação EID

O EID usa algoritmos de unificação para mesclar instâncias de classes para um dado objeto. É esse algoritmo que define os critérios para preservação de um dado atributo em detrimento de outro ou mesmo o descarte de determinada instância de classe.

O EID disponibiliza dois algoritmos padrões: um para conciliação de instâncias únicas, onde os atributos de duas ou mais instâncias são mesclados em uma instância final, e outro para instâncias múltiplas, onde todas as instâncias são preservadas em uma lista.

É permitida a definição do algoritmo a ser utilizado por cada classe. Não informar esse algoritmo implica a utilização de um dos algoritmos padrões.

Novas implementações podem ser dadas e devem implementar a interface `IClassUnifier` e ser disponibilizadas na aplicação EID.



Nome da Classe de Unificação:

Upload de Classe JAVA de Unificação:

- + Arquivo JAVA
- BrEduPersonContaUnifier.java
- Done
- Limpar**

Definição do algoritmo:

```
package br.ufmg.lcc.eid.services.unifier;
import br.ufmg.lcc.eid.commons.EidClassHelper;
import br.ufmg.lcc.eid.commons.EidConstantes;
import br.ufmg.lcc.eid.commons.EidException;
import br.ufmg.lcc.eid.dto.ClassDef;
import br.ufmg.lcc.eid.dto.EidClass;
public class BrEduPersonContaUnifier implements IClassUnifier {
    /**
     * Precedence constants
     */
    private enum Precedences {
        FIRSRT_PRECEDES, SECOND_PRECEDES, NO_PRECEDENCE,
        ABSOLUTELY_NO_PRECEDENCE
    }
    /**
     * Active status for students
     */
    private static final Set<String> STUDENT_ACTIVE_STATUS = new
    HashSet<String>();
    static {
        STUDENT_ACTIVE_STATUS.add("normal");
    }
}
```

Na figura 4.6 vemos o painel *Algoritmo de deduplicação* da tela de definição de classe.

Nesta tela há três maneiras para informar o algoritmo que irá fazer a unificação:

1. Informando o nome completo do algoritmo de Unificação no campo *Nome da Classe de Unificação*. Esta opção é válida quando o algoritmo de unificação já está disponível compilado no *classpath* do Tomcat: /diretório_tomcat/webapps/eid/WEB-INF/classes/.
2. Através de upload de um arquivo Java, clicando no botão + *Arquivo JAVA*, logo em seguida no botão *Upload* que é exibido, e então a classe é carregada e exibida conforme a imagem 4.6.
3. Digitando ou colando o conteúdo do algoritmo nos campos específicos.

Depois de cadastrar o algoritmo de unificação salve a definição de classes.

Figura 4.6
Algoritmo de deduplicação.

Web services

- Clientes podem usufruir dos registros conciliados
- Web services possibilitam uma forma mais adequada de acesso aos dados
 - Independente de linguagem ou plataforma
 - Abstração do modelo de dados
 - Objetos EID expostos como XML
- <http://servidor:porta/eid/services/EidService?wsdl>
 - Não pede autenticação
 - Deve ser protegido com firewall ou autenticação SSL

O EID disponibiliza um web service para exportação e consulta de dados, o que facilita o acesso por aplicações que utilizem tecnologias diversas. O web service serve de base também para outras ferramentas de exportação. Um exemplo é a ferramenta denominada EID2LDAP, que exporta os dados do EID para servidores LDAP.

```
<eid-object type="person" guid="EHBBCXKA-YLHXBAAA"
serial="148048">

  <attributes class="Identificacao" id="52347">

    <attribute name="nomeCompleto" source="Etc exemplo 1"
key="134"><![CDATA[ZACARIAS SILVA]]></attribute>

    <attribute name="nomeSolteiro"><![CDATA[]]></attribute>

    <attribute name="cpf" source="Etc exemplo 1" key="134"><![CDATA[03392002698]]></attribute>

  </attributes>

  <attributes class="Email" id="72201">

    <attribute name="email"><![CDATA[zeca@mail.com]]></
attribute>

  </attributes>

</eid-object>
```

O uso de web services foi escolhido por abstrair os clientes do modelo de dados do EID.

- Os objetos são entregues como documentos XML auto-contidos.
- Outra vantagem é a independência de plataformas dos clientes do EID, que podem ser implementadas em outras linguagens além de Java.
- O serviço não está protegido, o que pode ser feito via configuração de SSL autenticado para a URL e firewall.



A descrição dos serviços no formato WSDL pode ser acessada na URL <http://localhost:8080/eid/services/EidService?wsdl>, onde *localhost* deve ser substituído pelo endereço da máquina onde o EID está instalado. Ao se carregar o EID no Tomcat, o web service é automaticamente iniciado.

Problemas comuns

- Dados inconsistentes no banco
- Carga da classe *Conta*
- Usuário que sobe o Tomcat deve ter permissão na pasta *webapps* do EID

Algumas situações podem levar à presença de dados inconsistentes na base do metadiretório, que se apresentam no log do Tomcat (*catalina.out*) da seguinte forma:

```
188853 ERROR [Eid thread] br.ufmg.lcc.eid.controller.  
EidServletContextListener      - Error processing conciliation  
  
br.ufmg.lcc.eid.common.EidException: Error retrieving object:  
org.hibernate.InstantiationException, Cannot instantiate  
abstract class or interface: br.ufmg.lcc.eid.dto.EidClass  
  
at br.ufmg.lcc.eid.common.EidException.  
eidErrorHandler(EidException.java:46)  
  
at br.ufmg.lcc.eid.model.EidFacade.runConciliator(EidFacade.  
java:62)  
  
at br.ufmg.lcc.eid.controller.EidServletContextListener$EidTh  
read.run(EidServletContextListener.java:39)  
  
at java.lang.Thread.run(Thread.java:619)
```

Essa situação pode ser corrigida com o script disponibilizado no site do projeto.

Uma dúvida constante diz respeito à carga da classe *Conta*, em particular ao campo *algoritmoSenha*. Esse campo deve ser preenchido com o algoritmo que foi utilizado para calcular a senha do usuário, caso não esteja em texto plano (SHA, MD5, CRYPT etc.).

Para senhas codificadas em base64, independente do algoritmo utilizado para o hash, o valor do campo deve ser base64, e para senhas em texto plano o campo não deve ser alimentado. A alimentação incorreta impossibilitará a autenticação dos usuários, que é o sintoma deste problema.



4

Roteiro de Atividades Criando extrações no EID

Tópicos e conceitos

- Extração de arquivos texto
- Resolução de objeto EID
- Parâmetros globais
- Importação incremental
- Scripts de conversão
- Algoritmos de unificação

Competências técnicas desenvolvidas

- Criação de extrações avançadas para alimentação do metadiretório.

Tempo previsto para as atividades

- 30 minutos

Atividade 1 – Definição de uma extração de arquivo texto

Abra um navegador e acesse o EID para configurar o repositório:

1. Acesse o menu *Configuração/Repositório de Dados*;
2. Acione o comando *Novo* para definir um novo repositório;
3. Selecione o Tipo do Repositório Arquivo CSV.
4. Forneça os campos necessários:
 - 4.1. *Nome*: Repositório de arquivos CSV
 - 4.2. *Descrição*: Repositório de dados externos aos sistemas
 - 4.3. *Diretório*: /treinamento
5. Acione o comando *Salvar*.

Crie uma extração para carregar a classe *Identificação* a partir do arquivo texto *novasPessoasComCpf.txt*.

1. Acesse o menu *Configuração/ETC*.
2. Acione o comando *Novo*.
3. Na guia *ETC*, especifique:
 - 3.1. *Nome*: Extração de pessoas do arquivo CSV
 - 3.2. *Descrição*: Extração de dados de pessoas a partir de arquivo CSV
4. Na guia *Leiaute de Origem*:
 - 4.1. *Repositório*: Repositório de arquivos CSV
 - 4.2. *Objeto de origem*: novasPessoasComCpf.txt
 - 4.3. *Separador Decimal*: vírgula
 - 4.4. *Separador Campos*: ponto e vírgula
 - 4.5. *Codificação Caracteres*: UTF-8
 - 4.6. *Formato da data*: dd/MM/yyyy
 - 4.7. No painel *Leiaute de Origem de Dados*, defina os campos *id*, *nome*, *sexo*, *nascimento* e *CPF* para equivaler aos campos presentes no arquivo texto.

Acione o comando *Novo* deste painel para adicionar novos itens, se necessário. Ordem dos campos do arquivo: identificador único para os registros, nome completo, sexo, data de nascimento (formato dd/mm/aaaa) e CPF.

- 4.8. Selecione o campo *Id* como identificador único (IU).



5. Na guia *Leiaute de Destino*:
 - 5.1. Tipo Script: Bean Shell
 - 5.2. Tabela de Destino: identificação
 - 5.3. Atualizar Registros Existentes: marcar a caixa
 - 5.4. Acione o comando *Leiaute*
 - 5.5. No painel Leiaute de Destino dos Dados:
 - Mapeie os campos de origem para o destino.
 - Marque para remoção os campos que não serão mapeados.
 - Crie um script para converter o campo *dataNascimento*. Deixe o campo fonte em branco e preencha o campo *Script* com o código abaixo:

```
java.util.Date result = null;
public void execute() {
    if (nascimento != null){
        java.text.SimpleDateFormat formatador = new java.text.
SimpleDateFormat("dd/MM/yyyy");
        result = formatador.parse(nascimento);
    }
}
```

6. Acione o comando *Salvar*.

Crie um processo que inclua a extração definida anteriormente e o agende para ser executado de imediato, sem repetições.

7. Acesse o menu Configuração/Processos.
8. Acione o comando *Novo*.
9. Preencha os campos:
 - 9.1. *Nome*: Processo de extração de CSV
 - 9.2. *Descrição*: Processo de extração de dados de arquivo CSV
 - 9.3. *Modo*: selecione *Interromper Processamento*
 - 9.4. *Número de tentativas*: 1
 - 9.5. *Intervalo entre tentativas*: 1



10. No painel *Itens de processo*:

10.1. Clique no botão *Novo*, selecione a ETC “Extração de pessoas do arquivo CSV”, acione o botão *Selecionar*.

10.2. *Intervalo commit*: 500

10.3. *Número de erros*: 0

11. Acione *Salvar*.

12. Acesse o menu *Processamento/Agendamento*.

13. Acione o comando *Novo*.

14. Selecione:

14.1. *Processo*: Processo de extração de CSV

14.2. *Tipo de repetição*: nenhum

14.3. *Item de Início*: Processo de extração de CSV

14.4. *Finalizar no Item*: Processo de extração de CSV

14.5. *Próxima execução*: marcar *Processar agora*

15. Acione o comando *Salvar*.

16. Observe o resultado acessando o menu *Processamento/Resultado de processamento*.

Atividade 2 – Definição de extração para a classe Aluno

Configure uma extração para a classe aluno, extraindo dados das tabelas *Discente* e *Curso*.

1. Acesse o menu *Configuração/ETC*.

2. Acione o comando *Novo*.

3. Na guia *ETC*, especifique:

3.1. *Nome*: Extração de alunos do sistema acadêmico

3.2. *Descrição*: Extração de dados de alunos a partir do sistema acadêmico

4. Na guia *Leiaute de Origem*:

4.1. *Repositório*: Repositório Acadêmico

SQL:

```
SELECT d.CodDiscente, d.CodCurso, d.CodPessoa,      d.CodTurno,d.
AnoIngresso, d.CodIngresso, c.Nome, c.CodInepCapes, c.Nivel, c.
Modalidade, c.Formato FROM Discente d, Cursos c WHERE d.CodCurso
= c.CodCurso
```

4.2. Acione o comando *Leiaute*.



- 4.3. Selecione os campos *CodDiscente*, *CodCurso* e *CodPessoa* como identificador único (IU).
 5. Na guia *Leiaute de Destino*:
 - 5.1. *Tipo de Script*: Bean Shell
 - 5.2. *Tabela de Destino*: Aluno
 - 5.3. Acione o comando *Leiaute*.
 - 5.4. *Atualizar Registros Existentes*: marcar a caixa
 - 5.5. No painel *Leiaute de Destino dos Dados*
 - Mapeie os campos de origem para o destino.
 - No campo *eid_object_guid*:
 - ETC para FK: Extração de pessoas do sistema acadêmico
 - Campo *Fonte*: CodPessoa
 - Marque para remoção os campos que não serão mapeados.
 6. Acione o comando *Salvar*.
- Modifique o *Processo de extração Acadêmico* para incluir a extração definida anteriormente e o agende para ser executado de imediato, sem repetições.
1. Acesse o menu *Configuração/Processos*.
 2. Altere o *Processo de extração Acadêmico*.
 3. No painel *Itens de processo* acione o comando *Novo* e adicione:
 - 3.1. *ETC*: Extração de alunos do sistema acadêmico
 - 3.2. *Intervalo commit*: 500
 - 3.3. *Número de erros*: 0
 4. Acione *Salvar*.
 5. Acesse o menu *Processamento/Agendamento*.
 6. Altere o agendamento do *Processo de extração Acadêmico*.
 7. Selecione:
 - 7.1. *Iniciar no item*: Extração de alunos do sistema acadêmico
 - 7.2. *Item de início*: Extração de alunos do sistema acadêmico
 - 7.3. *Próxima execução*: marcar *Processar agora*
 8. Acione o comando *Salvar*.
 9. Observe o resultado acessando o menu *Processamento/Resultado de processamento*.

Atividade 3 – Transformação do campo Sexo

Modifique a extração de arquivo CSV da classe *Identificação* de forma que o campo de destino Sexo assuma os valores masculino ou feminino.

Altere a ETC de Identificação do arquivo CSV:

1. Acesse o menu *Configuração/ETC*.
2. Altere a *Extração de pessoas* do arquivo CSV.
3. Na guia *Leiaute de Destino*:

3.1. Selecione o campo fonte do campo Sexo como *vazio*.

3.2. Crie um script para converter o campo Sexo com o código:

```
String result = null;  
  
public void execute() {  
  
    if (sexo != null) {  
  
        if (sexo.equals("masculino")) {  
  
            result = "M";  
  
        } else if (sexo.equals("feminino")) {  
  
            result = "F";  
  
        }  
  
    }  
  
}
```

3.3. Acione o botão *Confirmar*.

3.4. Acione o comando *Salvar*.

Altere o agendamento:

4. Acesse o menu *Processamento/Agendamento*.
5. Altere o agendamento do *Processo de extração de CSV*.
6. Selecione:
 - 6.1. *Tipo de repetição*: Nenhum
 - 6.2. *Próxima execução*: marcar *Processar agora*
7. Acione o comando *Salvar*.
8. Observe o resultado acessando o menu *Processamento/Resultado de Processamento*.



Atividade 4 – Importação de login e senha

Crie uma extração para carregar a classe *Conta* a partir do arquivo texto *usuarios.txt*.

1. Acesse o menu *Configuração/ETC*.
2. Acione o comando *Novo*.
3. Na guia *ETC*, especifique:
 - 3.1. *Nome*: Extração de usuários do arquivo CSV
 - 3.2. *Descrição*: Extração de dados de usuários a partir de arquivo CSV
4. Na guia *Leiaute de Origem*:
 - 4.1. *Repositório*: Repositório de arquivos CSV
 - 4.2. *Objeto de origem*: usuarios.txt
 - 4.3. *Separador Decimal*: vírgula
 - 4.4. *Separador Campos*: ponto e vírgula
 - 4.5. *Codificação Caracteres*: UTF-8
 - 4.6. *Formato data*: dd/MM/yyyy
 - 4.7. Defina manualmente, no painel *Leiaute de Origem de Dados*, os campos *id*, *login* e *senha* para equivalerem aos campos presentes no arquivo texto.
Acione o comando *Novo* deste painel para adicionar novos itens, se necessário. Ordem dos campos do arquivo: identificador único para os registros, login e senha.
 - 4.8. Selecione o campo *Id* como *identificador único (IU)*.
5. Na guia *Leiaute de Destino*:
 - 5.1. *Tipo do Script*: Bean Shell
 - 5.2. *Tabela de Destino*: Conta
 - 5.3. Acione o comando *Leiaute*
 - 5.4. *Atualizar Registros Existentes*: marcar a caixa
 - 5.5. No painel *Leiaute de Destino dos Dados*:
 - Mapeie o campo *Login* de origem para o destino.
 - Marque para remoção o campo *Domínio* que não será mapeado.
 - Crie um script para extrair o campo *algoritmoSenha*. Utilize o código no campo *Script*:

```
String result = null;  
  
public void execute() {  
  
    if (senha != null) {
```



```

        result = senha.substring(1, 4);

    }

}

```

Após inserir o código na janela pop-up, clique no botão *Confirmar*.

- ▲ Crie um script para extrair o campo *Senha*. Utilize o código:

```

String result = null;

public void execute() {

    if (senha != null) {

        result = senha.substring(5);

    }

}

```

6. No *Painel Objeto* referenciado:

- 6.1. ETC para FK: Extração de pessoas do sistema acadêmico
- 6.2. Campo Fonte: id
- 6.3. Senha criptografada no arquivo usuários.txt para todos os usuários: esr

7. Acione o comando *Salvar*.

Altere o processo de extração de arquivos texto definido anteriormente, adicione a nova ETC e agende para ser executado de imediato, sem repetições.

8. Acesse o menu *Configuração/Processos*.

9. Altere o processo *Processo de extração de CSV*.

No painel *Itens de processo* clique em *Novo* e adicione:

- ▲ *ETC*: Extração de usuários do arquivo CSV
- ▲ *Intervalo commit*: 500
- ▲ *Número de erros*: 0

10. Acione *Salvar*.

11. Acesse o menu *Processamento/Agendamento*.

12. Altere o agendamento do processo *Processo de extração de CSV*.

- 12.1. *Tipo de repetição*: Nenhum
- 12.2. *Próxima execução*: marcar *Processar agora*
- 12.3. *Item de início*: Extração de usuários do arquivo CSV
- 12.4. *Finalizar no item*: Extração de usuários do arquivo CSV

13. Acione o comando *Salvar*.

14. Observe o resultado acessando o menu *Processamento/Resultado de Processamento*.

Atividade 5 – Importação Incremental

Na ETC de Pessoas do sistema acadêmico o campo *Time Stamp* no leiaute de origem foi marcado, o que significa que esta ETC irá fazer importações incrementais da segunda vez em diante quando for executada, importando apenas os registros que sofreram alterações na base de origem, tendo como base o campo *data_atualizacao*, que foi marcado como campo *Time Stamp* no leiaute de origem da ETC.

1. Acesse o phpMyAdmin da sua máquina, banco acadêmico, tabela *Pessoas* e altere o *Nome* adicionando alguma letra e a *data_atualizacao* de algum registro para a data de hoje.
2. Acesse o menu *Processamento/Agendamento de processo*.
3. Altere o agendamento do processo *Processo de extração Acadêmico*.
 - 3.1. Tipo de repetição: Nenhum
 - 3.2. Próxima execução: marcar Processar agora
 - 3.3. Item de início: Extração de pessoas do sistema acadêmico
 - 3.4. Finalizar no item: Extração de pessoas do sistema acadêmico
4. Acione o comando *Salvar*.
5. Observe o resultado acessando o menu *Processamento/Resultado de Processamento*. Verifique que todos os registros foram descartados por não terem sofrido alterações na base de origem, exceto o registro que sofreu a alteração.



5

Gestão de pessoas e grupos no EID

Sumário

- Gestão manual de pessoas
 - ▲ Conciliação de registros
 - ▲ Inserção de novas pessoas
 - ▲ Alteração de dados via interface
- Gestão de grupos
 - ▲ Inserção e atualização de grupos
 - ▲ Remoção de grupos

Gestão manual de pessoas

- Forma de manipulação dos registros via GUID
 - ▲ Conciliação manual
 - ▲ Inclusão, alteração e remoção de pessoas

Esta quinta sessão do curso apresentará as funcionalidades da gestão manual de pessoas e grupos. A ferramenta EID, além das funções de exportação e importação de dados, também possui a funcionalidade de gestão manual de pessoas e gestão de grupos.

A gestão manual de pessoas possibilita a conciliação de registros sugeridos pelo sistema ou duplicidades encontradas pelo administrador, além de inclusão, atualização e ativação/desativação de pessoas no metadiretório.

Conciliação de registros

- O EID procura conciliar automaticamente
- Outros casos não detectados podem ter conciliação forçada



Conciliação é o processo de identificação de objetos duplicados provenientes de fontes de dados diferentes. Objetos duplicados são registros separados que referenciam uma mesma entidade real. O principal problema de se ter objetos duplicados é a possível existência de atributos com valores divergentes. Após a identificação deve ser feita uma resolução dos conflitos.

O EID procura conciliar pessoas automaticamente. Ele utiliza o algoritmo Jaro Winkler, que faz um cálculo baseado em distância entre strings para detectar registros duplicados. Para realizar esta conciliação, ele leva em conta os dados *nomeCompleto*, *nomePai*, *nomeMae*, CPF, data Nascimento e sexo.

Em situações mais adversas, o administrador pode também forçar a conciliação de registros, selecionando-os diretamente pela interface do sistema.

- ▶ Processo assíncrono executado a cada 2 minutos
- ▶ Todo registro reimportado é reconciliado
 - ▶ Conciliação é direta
- ▶ Registros atualizados são marcados no metadiretório

O algoritmo de conciliação é executado de forma assíncrona a cada 2 minutos, consolidando todas as modificações pendentes.

Para uma extração configurada corretamente, a reimportação de registros causa sua atualização no EID, marcando os registros como pendentes. A conciliação, em uma próxima execução, tratará esses registros e refletirá as alterações no registro final. Essa reconciliação é mais barata que a primeira, dado que o grupo de registros a serem conciliados já seja conhecido.

Vale observar que uma reimportação desnecessária pode implicar uma maior demora do EID em refletir a alteração no registro consolidado. Por este motivo, é aconselhável que reimportações sejam incrementais, atualizando apenas os registros que tenham sido realmente alterados na fonte.

O EID mantém um controle sequencial que possibilita o monitoramento de registros conciliados.

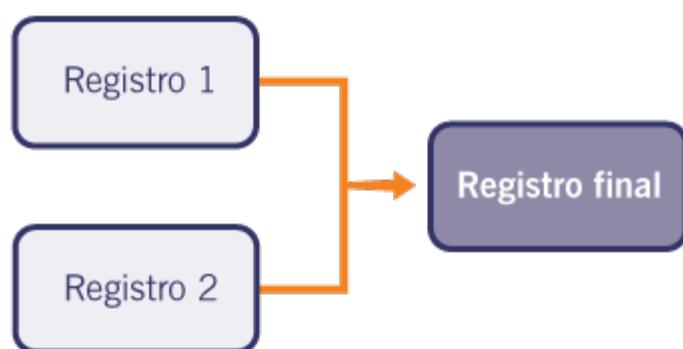


Figura 5.1
Monitoramento
de registros
conciliados.

Por questões de implementação, os registros importados não são alterados no processo de conciliação; eles são sempre mantidos no banco, em um estado diferenciado.

O algoritmo de conciliação gera um novo registro equivalente a cada conjunto de registros conciliados, sendo formado pela unificação dos vários registros iniciais. Essa primeira conciliação é custosa, pois exige a verificação de um grande conjunto de registros do banco.

Figura 5.2
Funcionalidades de conciliação.

The screenshot shows the EID interface with the 'Conciliação' (conciliation) menu selected. The main area displays a table titled 'Resultado da pesquisa' (search results) with 13 items. The columns are 'GUID', 'Identificacao' (with sub-columns 'nomeCompleto', 'sexo', and 'dataNascimento'), and actions 'Conciliar', 'Visualizar', and 'Atualizar'. The first two rows of data are:

GUID	Identificacao			Conciliar	Visualizar	Atualizar
	nomeCompleto	sexo	dataNascimento			
NUD-AMAGWWFA-GFLVBAAA	NEILA URBINA EDGARD ALYSSON D'ALVA	F	1899-12-30 00:00:00.0			
NUD-AKAZENXA-NKLVBAAA	NEILA URBINA DURVAL EDGARD ABALEM	F	1899-12-30 00:00:00.0			

A figura 5.2 apresenta a interface que dá acesso às funcionalidades de conciliação, interface acessada através do menu *EID/Conciliação*. São listados todos os registros julgados como possivelmente conciliáveis pelo sistema, onde é possível optar por descartar a sugestão ou efetivar a conciliação.

Figura 5.3
Nova conciliação.

Se a sugestão for acatada, o EID promoverá a fusão dos registros em um único registro final, caso contrário serão gerados registros independentes para cada objeto listado.

The screenshot shows the EID interface with the 'Nova conciliação' (new conciliation) menu selected. On the left, there's a sidebar with 'Atributos visíveis' (visible attributes). The main area has a search panel with fields for 'Classe' (Identificacao), 'Classe de atributos' (nomeCompleto), and 'Valor do atributo' (%AMADO%). A 'Pesquisar' button is present. Below it is a table titled 'Resultado da pesquisa' with 13 items. The columns are 'GUID', 'Identificacao' (with sub-columns 'nomeCompleto', 'sexo', and 'dataNascimento'), and 'Selecionar' (checkboxes). The first two rows of data are:

GUID	Identificacao			Selecionar
	nomeCompleto	sexo	dataNascimento	
AAHPSGX-A-FXLVBAAA	AMADO SAMARA LILIAN WELTON	M	1972-12-25 00:00:00.0	<input checked="" type="checkbox"/>
SYGZZNVA-MTLVBAAA	AMADO AMARO BAUAB HESS	M	1969-01-03 00:00:00.0	<input checked="" type="checkbox"/>

Ainda na tela de conciliação, ao acionar o comando *Novo* é exibida a interface para definição de conciliação forçada.

Nesta interface, o comando *Adicionar* pode ser utilizado para localizar um registro e adicioná-lo à lista; *Remover* promove a remoção de um registro da lista; *Conciliar* coloca o conjunto de registros na fila de conciliação e *Cancelar* cancela a definição da conciliação.

Os registros selecionados serão mesclados em um único registro final, sendo mantido o GUID lexicograficamente menor. Os demais serão descartados.

Pesquisa de pessoas

- ▲ Localização por valores de atributos de qualquer classe
- ▲ Curinga '%' pode ser utilizado
- ▲ Pode-se selecionar os atributos que serão exibidos

Na pesquisa de pessoas o EID possibilita a pesquisa pelo atributo de qualquer uma de suas classes. A busca exibe, por padrão, apenas o GUID dos registros. Outras informações podem ser observadas selecionando-se os atributos das classes de interesse.

Figura 5.4
Tela de gestão
de pessoas.

A figura 5.4 apresenta a tela de gestão de pessoas, que pode ser acessada pelo menu *EID/Gestão de Pessoas*.

Na aba *Parâmetros* devem ser definidos os parâmetros de busca. No campo *Classe* deve-se selecionar a classe que contém o atributo a ser pesquisado.

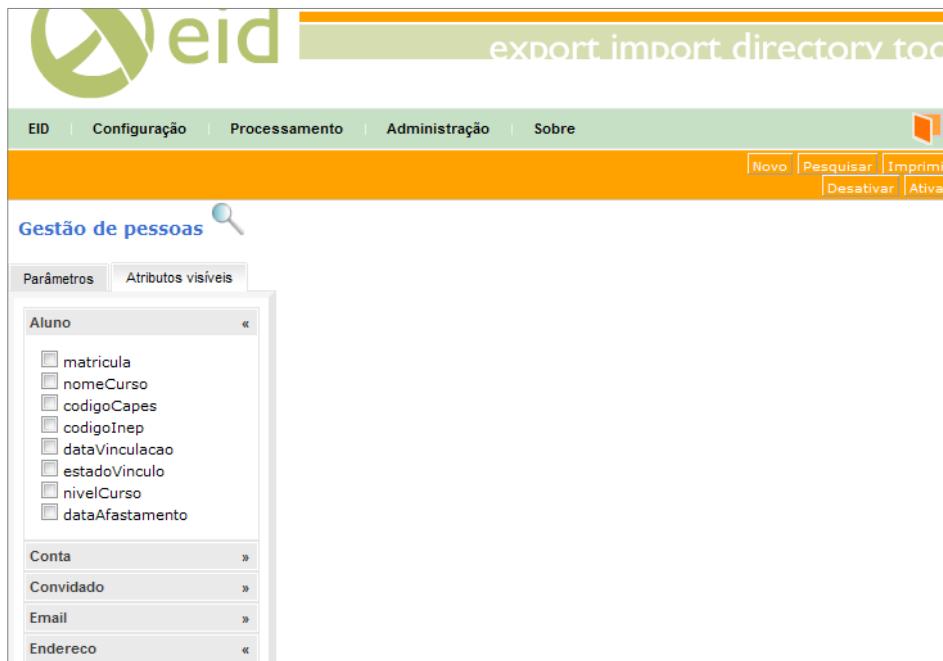
O campo *Atributo* apresenta os atributos definidos para a classe em questão. O valor desejado deve ser informado no campo *Valor do atributo*.

Preenchidos os dados, o comando *Pesquisar* efetua a busca, apresentando os dados na parte inferior da tela. Os dados de uma pessoa específica podem ser observados clicando-se na lupa na linha do registro, ou alterados acionando-se o botão de alteração de sua linha. Ambos os comandos levam a outra interface que será discutida adiante.

Como exemplo, os critérios a seguir serão usados para retornar a relação de todas as pessoas que tenham nome completo iniciado por José e terminado com Silva:

- ▲ Classe: Identificação
- ▲ Atributo: NomeCompleto
- ▲ Valor: José%Silva

Figura 5.5
Atributos visíveis.



Por padrão, o EID exibe apenas o GUID dos objetos encontrados. A figura 5.5 exibe a aba *Atributos visíveis*, onde é possível selecionar os atributos que serão exibidos, clicando-se nas caixas equivalentes aos nomes das classes. Ao marcá-las, o atributo estará visível no resultado apresentado na parte inferior da tela.

Inserção de novas pessoas

- ▲ Forma de inserção de pessoas não existentes nas bases corporativas
- ▲ Passam a compor o metadiretório

O EID possibilita a inserção de pessoas externas às bases corporativas. Uma vez incluídas, estas passam a fazer parte do metadiretório, participando também das conciliações.

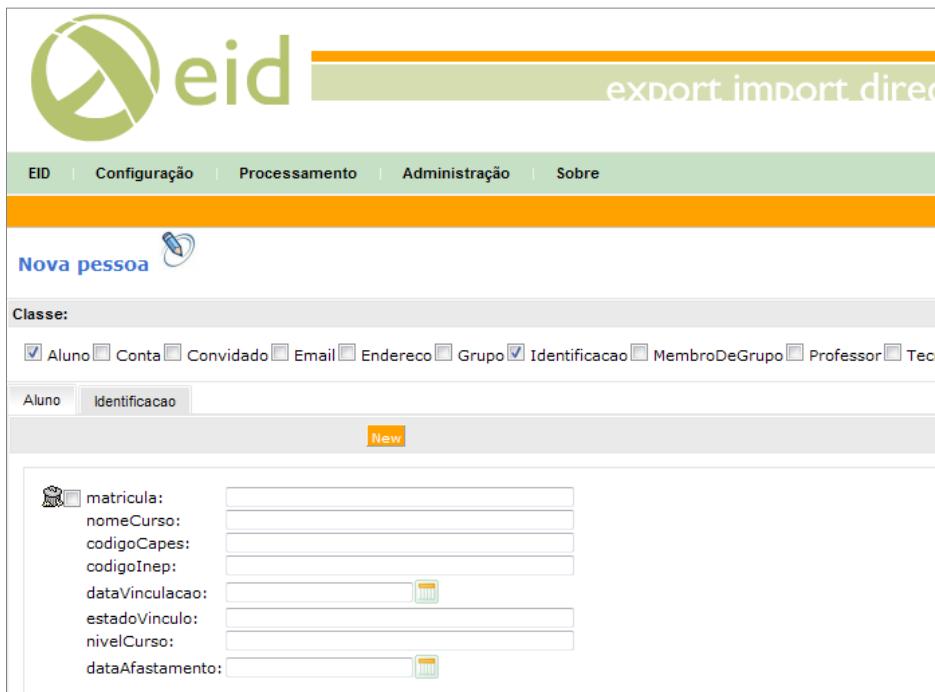


Figura 5.6
Tela de gestão
de pessoas.

A figura 5.6 apresenta a tela de gestão de pessoas que pode ser acessada pelo menu *EID/Gestão de Pessoas*.

- ▲ O comando *Novo* exibe a interface para definição dos dados da pessoa.
- ▲ É necessário selecionar as classes a serem instanciadas e preencher os campos para cada instância.
- ▲ A classe *Identificação* deve ser sempre selecionada para o correto funcionamento do sistema.
- ▲ Depois de preenchidos os dados, o comando *Salvar* deve ser acionado.

Alteração de dados via interface

- ▲ Correção de dados
- ▲ Atribuição de instâncias de classes

Figura 5.7
Tela de gestão
de pessoas.

nomeCompleto:	PAULINO JAUARA GLEIBER GROSSI
cpf:	88856743256
numeroIdentidade:	M7708291
orgaoEmissorIdentidade:	SSP
ufIdentidade:	MG
passaporte:	
estadoCivil:	SOLTEIRO
sexo:	F
nacionalidade:	
dataNascimento:	28/09/1971
cidadeNascimento:	BELO HORIZONTE
estadoNascimento:	MG
paisNascimento:	Brasil
nomeMae:	PICCININI NEILA
nomePai:	CLEO CASULA GLEIBER GROSSI

Dados de uma pessoa podem ser alterados pelo administrador, muito embora esta não seja a forma recomendada: o ideal é que a alteração seja feita na fonte. É possível, também, a atribuição de instâncias de classes a pessoas, opção útil para classes gerenciadas manualmente e com instâncias para poucos usuários (atributos de serviços mais específicos).

A tela de gestão de pessoas pode ser acessada pelo menu *EID/Gestão de Pessoas*. Deve-se, primeiramente, localizar a pessoa que terá seus dados modificados. Esta pesquisa pode ser feita conforme explicado na seção *Pesquisa de registros*.

Novas instâncias podem ser atribuídas através da seleção das classes de interesse, e dados podem ser alterados pela edição dos valores dos atributos das instâncias existentes.

Efetuadas as alterações, o comando *Salvar* deve ser acionado. O registro editado será então marcado como pendente para reconciliação.



Forçar Reunificação

- Botão Reunificar
 - ▀ Ao ser acionado marca o registro como pendente para reunificação

Muitas vezes deseja-se atualizar um registro no LDAP. O botão *Reunificar*, criado para isso, refaz a unificação para determinado registro fazendo com que seu *serialNumber* seja incrementado e consequentemente fique marcado como atualizado para ser exportado novamente para o LDAP.

Desativação de pessoas

- Pessoas não são removidas, mas marcadas como inativas
 - ▀ Não são expostas pelo EID
 - ▀ Elimina complicações em reimportação
- Podem ser reativadas
- Conciliação e atualização de dados continuam sendo realizadas

Resultado da pesquisa:			
Número de itens retornados:1			
	GUID	Visualizar	Atualizar
<input checked="" type="checkbox"/>	UXFJPTAA-TJQCBAAA		

Figura 5.8
Tela de visualização de pessoa.

Registros de pessoas não são removidos, mas marcados como inativos. Essa estratégia elimina problemas relativos à reimportação de registros, o que poderia ocasionar o reaparecimento da pessoa.

Figura 5.9
Registro marcado como inativo.

Registros inativos não são expostos pelo EID, a não ser que sejam requisitados por uma função específica. Estando os registros inativos, as atualizações feitas nos registros originais continuam refletidas no registro final. Em caso de reativação, os dados do registro já refletem a situação atual dos registros originais.

A desativação de registro é feita na tela principal de *Gestão de pessoas*.

Os registros de interesse devem ser localizados e marcados. O comando *Desativar* promove sua desativação.

A reativação pode ser feita marcando-se o registro e acionando-se o comando *Ativar*.

Pessoas inativas são localizadas normalmente na interface de pesquisa, porém não disponibilizam função de visualização ou edição.

Gestão de grupos

- Grupo é um tipo especial de objeto EID
- Realiza gestão automática de membros
- Critérios definidos como consulta HQL
- Atualizados diariamente
- Relacionamento do grupo com as pessoas
- Relacionamento das pessoas com o grupo

A ferramenta EID disponibiliza uma forma simples de criar agrupamentos, tanto de pequenos quanto de grandes grupos (professores da universidade, alunos da disciplina *Cálculo 1*).

Os critérios são definidos como consulta Hibernate Query Language (HQL) e executados periodicamente, procurando manter o grupo atualizado.

Relacionamentos do grupo com as pessoas e seus atributos são criados, indicando pertinência a grupos.

 A atualização de grupos pode ser forçada via interface.

Inserção, atualização e remoção de grupos

- Inclusão, alteração e remoção feitos como na gestão de pessoas

As operações com grupos são feitas da mesma forma que na gestão de pessoas. A interface pode ser acessada pelo menu *EID/Gestão de Grupos*.



5

Roteiro de Atividades Gestão de pessoas e grupos no EID

Tópicos e conceitos

- Gestão manual de pessoas
 - ▲ Conciliação de registros
 - ▲ Inserção de novas pessoas
 - ▲ Alteração de dados via interface
- Gestão de grupos
 - ▲ Inserção e atualização de grupos
 - ▲ Remoção de grupos

Competências técnicas desenvolvidas

- Resolução de conciliações via interface, inserção manual de pessoas e gestão de grupos no EID. metadiretório.

Tempo previsto para as atividades

- 40 – 50 minutos

Atividade 1 – Conciliação de um registro manualmente

Selecione registros do banco de dados e force a conciliação:

1. Acesse o menu *EID/Conciliação*.
2. Acione o comando *Novo*.
3. Acione o comando *Adicionar*.
4. Preencha o campo *Classe* com *Identificação*.
5. Preencha o campo *Classes de Atributos* com *NomeCompleto*.
6. Preencha o campo *Valor do atributo* com *JOSE FI%* e clique em *Pesquisar*.
7. Na aba *Atributos visíveis* selecione *Identificação* e marque o campo *nomeCompleto*.
8. Volte para a aba *Parâmetros* e selecione o registro *JOSE FELISBINO*, marcando-o e clicando no botão *Selecionar*.
9. Acione novamente o comando *Adicionar*.
10. Preencha o campo *Classe* com *Identificação*.
11. Preencha o campo *Classes de Atributos* com *NomeCompleto*.
12. Preencha o campo *Valor do atributo* com *JOSE FE%* e clique em *Pesquisar*.
13. Na aba *Atributos visíveis* selecione *Identificação* e marque o campo *nomeCompleto*.
14. Volte para a aba *Parâmetros* e selecione o registro *JOSE FELISBINO* marcando-o e clicando no botão *Selecionar*.
15. Acione o comando *Conciliar*.
16. Acesse o menu *EID/Gestão de pessoas*.
17. Pesquise por *JOSE F%* e observe os dados da pessoa.

Atividade 2 – Registros pendentes para conciliação

Conciliar ou excluir da conciliação os registros que o EID não teve certeza de que eram da mesma pessoa:

1. Acesse o menu *EID/Conciliação*.
2. É exibida uma lista com todos os registros que ficaram pendentes para conciliação.
3. Nos *Parâmetros visíveis* marque a classe *Identificação* e selecione: *nomeCompleto*, *sexo*, *nomePai*, *nomeMãe*.

4. Pesquise por usuários duplicados e faça a conciliação clicando no ícone *Conciliar*.
5. Pesquise por usuários que não são os mesmos, mas estão agrupados para conciliar. Exclua da conciliação clicando no checkbox abaixo da lixeira eacionando o botão *Excluir*.

Atividade 3 – Inserção de uma nova pessoa

Faça a inserção manual de uma nova pessoa via interface.

1. Acesse o menu *EID/Gestão de pessoas*.
2. Acione o comando *Novo*.
3. Selecione as classes *Identificação* e *Conta*.
4. Preencha os dados da aba *Identificação* e também da aba *Conta*.
 - 4.1. Identificação:
 - Nome completo: Maria Silva Souza
 - CPF: 12345678900
 - Data de nascimento: 23/01/1985
 - 4.2. Conta:
 - login: msilva
 - senha: esr
5. Acione o comando *Salvar*.

Atividade 4 – Definição de um grupo

Faça a definição de um grupo no EID.

1. Acesse o menu *EID/Gestão de grupos*.
2. Acione o comando *Novo*.
3. Selecione a classe *Grupo*.
4. Informe o nome *Alunos de Arquitetura*.
5. Como critério, coloque a seguinte consulta:


```
select a.eidObject from Aluno a where a.nomeCurso
= 'ARQUITETURA'
```
6. Acione o comando *Salvar*.

6

Alimentação de diretórios com EID2LDAP

Sumário

- Introdução
 - EID2LDAP
 - Características
- Arquitetura
 - XML do EID
 - XSLT
 - Processamento do LDIF
- Configuração e uso
- Problemas comuns

Introdução

EID2LDAP

- Busca informações de diretório armazenadas em um servidor EID e as transfere para servidores LDAP

Esta sexta sessão do curso apresentará a ferramenta EID2LDAP. A introdução trata de suas características e sua arquitetura (XML do EID, XSLT e processamento LDIF); por fim são vistas as configurações e alguns exemplos de uso da ferramenta, incluindo os problemas comuns.

O EID2LDAP é uma ferramenta que acessa o servidor EID via web service, transforma os registros para o formato LDIF compatível com o servidor LDAP de destino e transfere as informações.



Características

- ▲ Permite o agendamento periódico da exportação
 - ▲ Em cada exportação, são atualizados apenas os registros modificados/inseridos/apagados desde a última exportação
- ▲ Acessa o EID via Web Service
- ▲ Utiliza a marcação XSLT para especificar a transformação dos dados para o formato LDAP Data Interchange Format (LDIF)
 - ▲ O XSLT é fornecido pelo usuário e deve gerar um LDIF compatível com o esquema do LDAP de destino

Assim como o EID, a ferramenta EID2LDAP permite o agendamento periódico das exportações; em cada exportação são atualizados apenas os registros modificados/inseridos/desativados desde a última importação.

Como a estrutura do LDAP é flexível, ao exportar é necessário conhecê-la. O Extensible Stylesheet Language Transformations (XSLT) introduz flexibilidade no EID2LDAP, permitindo ao usuário definir como se dará o mapeamento entre os dados do EID e o formato do LDAP. Logo, para realizar a exportação, três conhecimentos são necessários:

- ▲ O formato do EID (padrão);
- ▲ O formato do LDAP (específico);
- ▲ A linguagem XSLT.

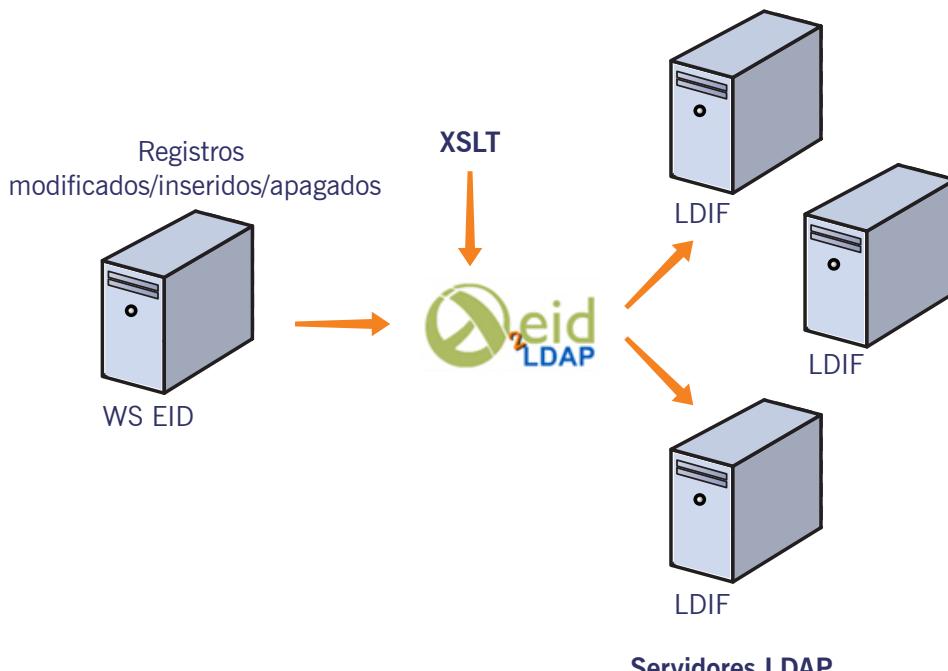


Figura 6.1
Arquitetura.

A exportação dos dados se inicia quando o algoritmo de transformação determina se o tempo de agendamento foi alcançado:

- EID2LDAP acessa o EID via Web Service.
- Busca registros modificados/inseridos/desativados.
- Transforma os registros no formato LDIF.
- Envia o LDIF aos servidores LDAP.
- Faz um novo agendamento caso o modo de repetição esteja açãoado.
- Registros são requisitados e processados de 100 em 100.
- Todos os registros são transformados em LDIF e depois enviados.
- Caso ocorra erro, o processamento será interrompido.
- O próximo agendamento reiniciará a partir da série de registros em que o erro ocorreu.

Importante: o que foi enviado nesse intervalo ao LDAP (antes do erro) não será desfeito, mas reescrito na próxima iteração.

A seguir serão detalhados o XML do EID, o modo de especificar o XSLT, e a forma como é realizada a transformação para LDIF.

XML do EID

- Contém informações sobre:
 - Pessoas e seus atributos


```
<eid-object type="person">....</eid-object>
```
 - Grupos


```
<eid-object type="group">...</eid-object>
```
 - Membros do grupo


```
<member> <eid-object>...</eid-object>....</member>
```
 - Pessoas e grupos desativados


```
<eid-object type="person" removed="true">
```

O XML fornecido pelo EID carrega as informações sobre as pessoas e os grupos. São buscados apenas os objetos novos, alterados ou excluídos.

Não há marcação no XML para indicar o atributo alterado, nem para diferenciar um objeto novo de um alterado. Sempre é enviado todo o conteúdo do objeto.

Na desativação, o atributo do objeto *removed* é marcado como *true*.

```
<!--Pessoa ou Grupo -->
<eid-object type="person" guid="EHBBCXKA-YLHXBAAA"
serial="148048">
```



```

<!-- Classe -->
<attributes class="Identificacao" id="52347">
    <attribute name="nomeCompleto"
        key="03812882698"><![CDATA[ZACARIAS SILVA]]></
attribute>
    <attribute name="nomeSolteiro"><![CDATA[]]></attribute>
    <attribute name="cpf"><![CDATA[0121222222]]></attribute>
        </attributes>
<attributes class="Email" id="72201">
    <attribute name="email"><![CDATA[zeca@mail.com]]></
attribute>
    </attributes>
</eid-object>
<!--Membros de Grupos -->
<member>
    <eid-object>...</eid-object>
</member>

```

No XML do EID, as várias classes existentes para a pessoa são recuperadas em elementos *attributes* e seus atributos dispostos em elementos *attribute*, contendo nome e valor de cada um.

XSLT

- ▲ Transformações necessárias:
 - ▲ Marcação para inserção de pessoas e grupos
 - ▲ Marcação para exclusão de registros e grupos
 - ▲ Marcação para adição de pessoas
 - ▲ Lembre-se de gerar o mesmo Domain Name (DN) na adição e na exclusão

O XSLT controla a transformação do XML no LDIF que será enviado ao LDAP.

O LDIF gerado determina as operações que serão aplicadas no LDAP (Inserção/Exclusão/Alteração).

O XSLT deve tratar os tipos de informações enviadas pelo EID, que são:

- ▲ Inserção de pessoas, membros de grupo e grupos (como a alteração não é especificada, deve ser tratada como inserção);
- ▲ Exclusão de pessoas, grupos e membros de grupos;
- ▲ O processo de alteração é tratado de forma automática pelo EID2LDAP, descrito em “Processamento do LDIF”.

Inserção de registros:

```
<xsl:template match="/">
<xsl:apply-templates select="eid-object[@type='person',
and (not(attribute::removed) or @removed='false'))"
mode="person" />
</xsl:template>
<xsl:template match="eid-object" mode="person">
dn: cn=<xsl:value-of select="@guid" />, dc=lcc, dc=ufmg,
dc=br
changetype: add
objectclass: person
cn: <xsl:value-of select="@guid" />
sn: <xsl:value-of select="@guid" />
</xsl:template>
```

O XSLT é utilizado para formatar o LDIF que será enviado para o LDAP. De acordo com as informações contidas nos dados provenientes do EID, o XSLT especifica o mapeamento de cada um dos atributos para os atributos LDAP, bem como a operação a ser feita (add/delete/modify).

Exclusão de registros (registros marcados com removed="true"):

```
<xsl:template match="/">
<xsl:apply-templates select="eid-object[@type='person',
and @removed='true']" mode="removed" />
</xsl:template>
<xsl:template match="eid-object" mode="removed">
dn: cn=<xsl:value-of select="@guid" />, dc=lcc,
dc=ufmg, dc=br
changetype: delete
</xsl:template>
```

O XSLT apresentado ilustra o uso do atributo *removed* com o valor igual ao de *true*.

Inclusão de membro em grupo:

```
<xsl:template match="/">
<xsl:apply-templates select="eid-object[@type='group']"
mode="group" />
<xsl:apply-templates select="member"/>
</xsl:template>
<xsl:template match="eid-object" mode="group">
dn: cn=<xsl:value-of select="@guid" />, dc=lcc, dc=ufmg,
dc=br
changetype: add
objectclass: groupOfNames
cn: <xsl:value-of select="@guid" />
</xsl:template>
```



```
<xsl:template match="member">
  member: cn=<xsl:value-of select="eid-object/@guid" />,
  dc=lcc, dc=ufmg, dc=br
</xsl:template>
```

A marcação `<member>` não existe no EID, sendo inserida pelo EID2LDAP para agrupar e indicar os *EIDObjects* que são membros do grupo.

A marcação `<eid-object type="group">` deve ser gerada para criar o LDIF com o *objectclass groupOfNames*.

Processamento do LDIF

- ▲ Entradas sem a operação definida (Add/Modify/Delete) ou com a operação *Add* são tratadas como operações de adição
- ▲ Se o registro já existir no LDAP (identificado pelo DN gerado no LDIF):
 - ▲ O LDIF é modificado para aplicar operações de alteração
 - ▲ Apenas os *objectClasses* representados no LDIF serão substituídos no LDAP
- ▲ Entradas especificadas com a operação *Delete* são propagadas para todos os registros na sub-árvore da entrada
- ▲ Outras operações são aplicadas de forma inalterada

No momento da exportação, caso o registro já exista no LDAP (identificado pelo DN gerado no LDIF), o LDIF é modificado para aplicar operações de alteração (*modify*) no registro do LDAP.

Apenas os *objectClasses* representados no LDIF serão substituídos no LDAP, isto é, os *objectClasses* no LDAP passarão a ter os atributos com os mesmos valores do EID, enquanto outros *objectClasses* permanecerão com seus atributos inalterados. Isto possibilita que outras aplicações alimentem diretamente o diretório sem a necessidade de passar pelo EID.

Configuração e uso

- ▲ Acesso
 - ▲ Tela de login
 - ▲ Tela inicial
- ▲ Configurações
 - ▲ Servidores LDAP
 - ▲ Transformações
 - ▲ Agendamentos

A seguir serão apresentadas algumas interfaces da aplicação e exemplos de uso.



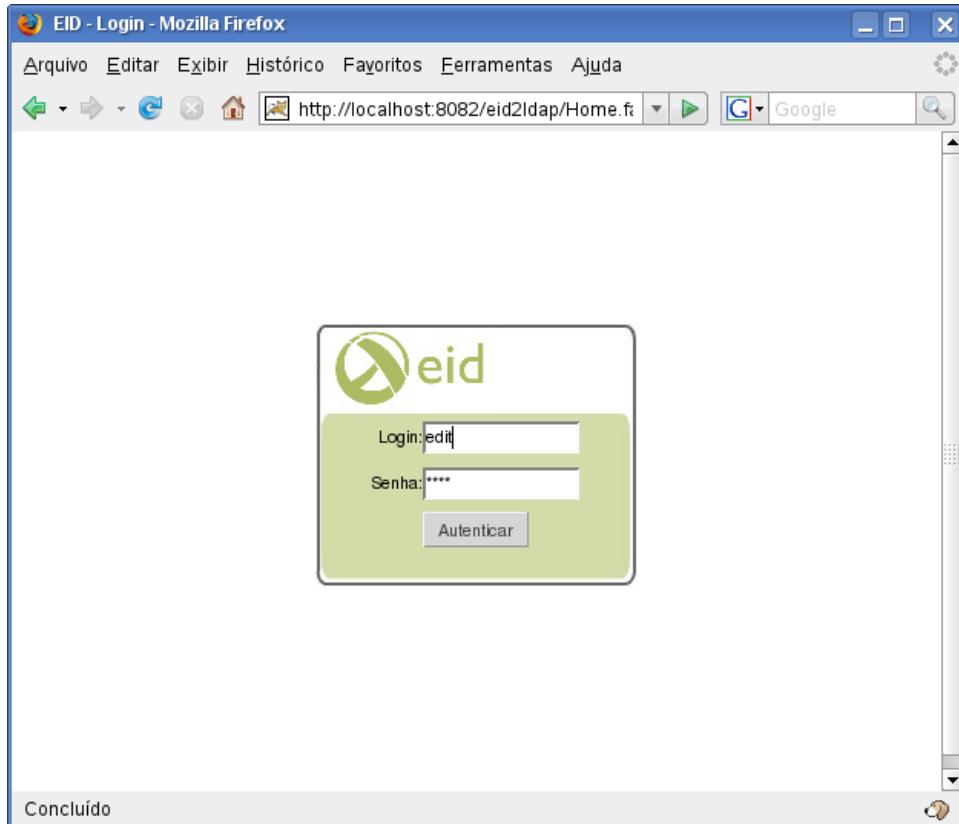
Acesso a aplicação EID2LDAP

- Para acessar a aplicação:
 - <http://nomeservidor:8080/eid2ldap>
 - nomeservidor:
 - Nome da máquina onde o EID2LDAP foi instalado

Após a instalação da aplicação, para acessá-la basta abrir um browser e redirecioná-lo para: <http://nomeservidor:8080/eid2ldap>.

Onde *nomeservidor* deve ser substituído pelo nome da máquina onde o EID2LDAP está instalado.

Figura 6.2
Tela de login.



A figura 6.2 apresenta a tela de login: o sistema define apenas um papel, o de administrador. Várias pessoas podem desempenhar este papel.

A autenticação do usuário é delegada ao Tomcat, podendo ser feita em arquivo texto, banco de dados, LDAP etc.

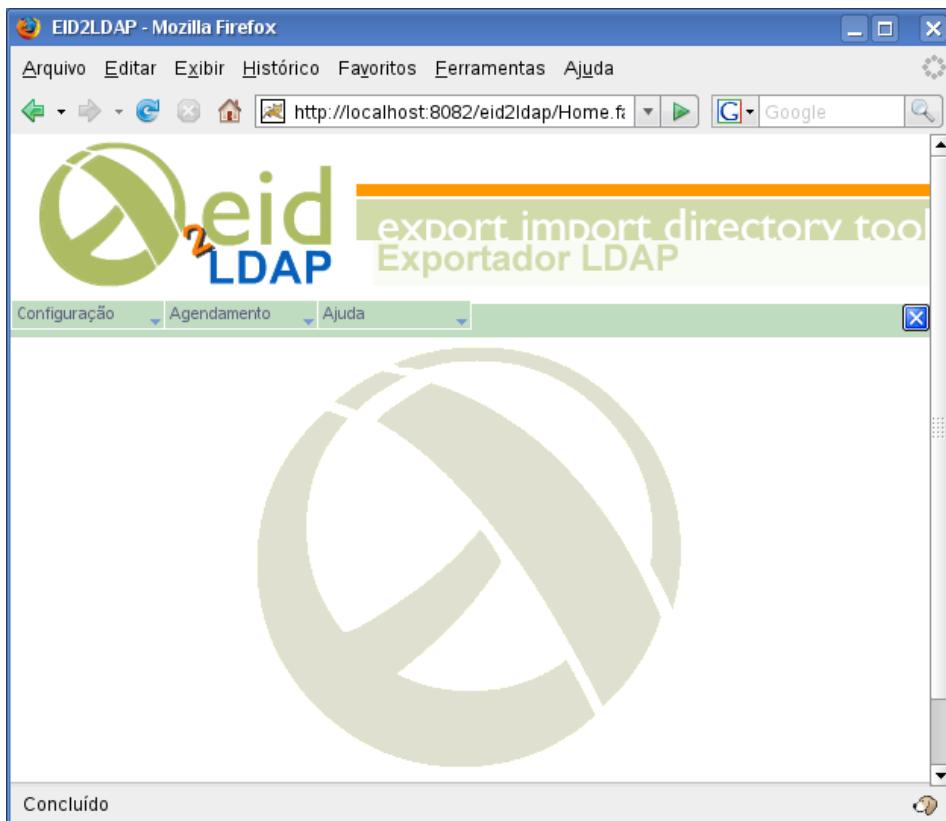


Figura 6.3
Tela inicial.

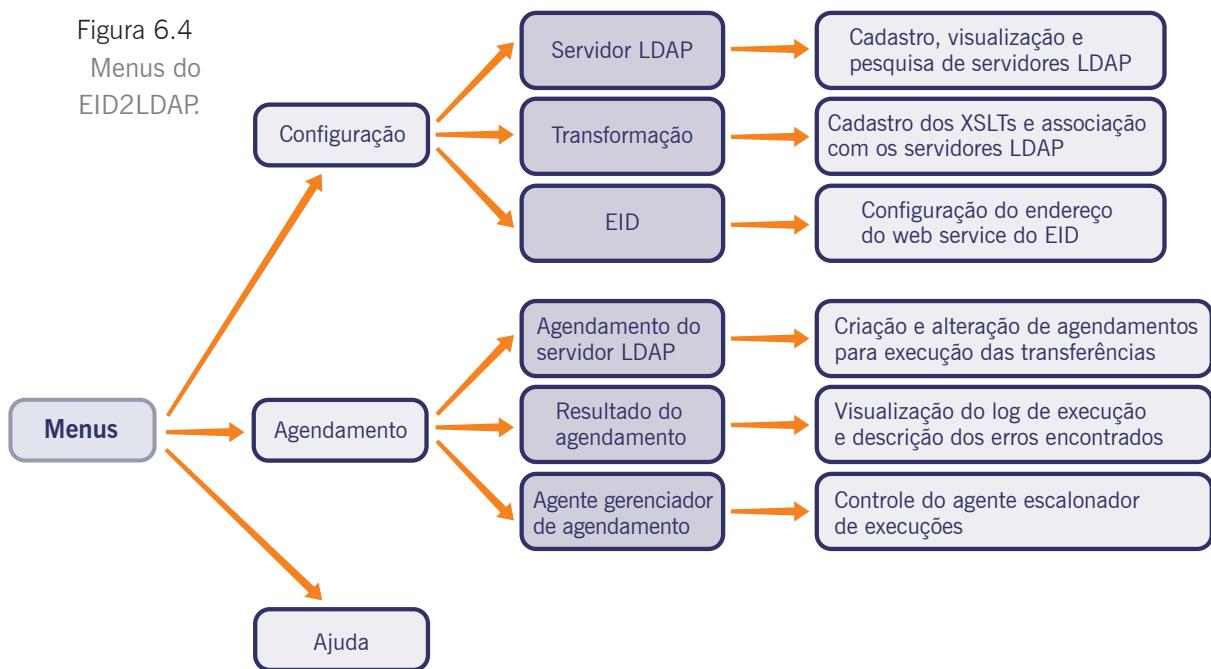
Na tela inicial o EID2LDAP apresenta três menus por onde são acessadas as funcionalidades do sistema: *Configuração*, *Agendamento*, *Ajuda* e um ícone azul localizado na parte superior direita da janela que finaliza a aplicação.

Menus

Os menus do EID2LDAP se organizam da seguinte forma:

- ▲ *Menu Configuração/Servidor LDAP* – Tela de pesquisa, visualização, alteração e cadastro de servidores LDAP.
- ▲ *Menu Configuração/Transformação* – Tela de cadastro dos XSLTs e associação com os servidores LDAP.
- ▲ *Menu Configuração/EID* – Tela para configuração do endereço do web service do EID.
- ▲ *Menu Agendamento/Agendamento Servidor LDAP* – Criação e alteração de agendamentos para execução das transferências.
- ▲ *Menu Agendamento/Resultado Agendamento* – Visualização do log de execução, descrição dos erros encontrados durante a execução das transferências.
- ▲ *Agendamento/Agente Gerenciador de Agendamento* – Controle do agente escalonador de execuções.

Figura 6.4
Menus do EID2LDAP.



Configuração de exportação

- Resumo
 - ▀ Inicialização do agente, se estiver parado
 - ▀ Cadastramento dos servidores LDAP
 - ▀ Cadastramento dos XSLTs e associação aos LDAPs
 - ▀ Criação do agendamento e definição dos LDAPs de destino
 - ▀ Verificação do log de processamento

Para configurar uma exportação de dados do servidor EID para um servidor LDAP via EIDLDAP, os seguintes passos devem ser executados:

1. Acesso ao *Menu Agendamento/Agente Gerenciador de Agendamento* e inicialização do agente, caso ele esteja parado.
2. Acesso ao *Menu Configuração/Servidor LDAP* e cadastramento dos servidores LDAP para onde se deseja exportar os dados.
3. Acesso ao *Menu Configuração/Transformação* e cadastramento dos XSLTs e associação aos respectivos LDAPs para realizar a correta transformação dos dados.
4. Acesso ao *Menu Agendamento/Agendamento Servidor LDAP* e criação do agendamento e definição dos LDAPs de destino.
5. Acesso ao *Menu Agendamento/Resultado Agendamento* e verificação do log de processamento.

Inicialização do agente



Figura 6.5
Agendamento/
Agente
Gerenciador de
Agendamento.

A figura 6.5 exibe a tela de *Gerenciador de Agendamentos*. O agente escalonador é responsável por verificar e iniciar a execução dos agendamentos. Ele está desabilitado após a instalação; se estiver parado, nenhum agendamento é iniciado. Quando iniciado, começa todos os agendamentos “atrasados”.

Para iniciar/parar o agente, basta acionar o botão e observar a mensagem ATIVO ou INATIVO.

Cadastramento dos servidores

Figura 6.6
Configuração/
Servidor LDAP.



A figura 6.6 mostra a tela de *Administração de LDAP*, que lista todos os LDAPs cadastrados. O comando *Novo* aciona a interface de definição de um novo servidor LDAP; o comando *Visualizar* dá acesso ao registro no modo de visualização e o comando *Alterar* exibe o registro no modo de edição.

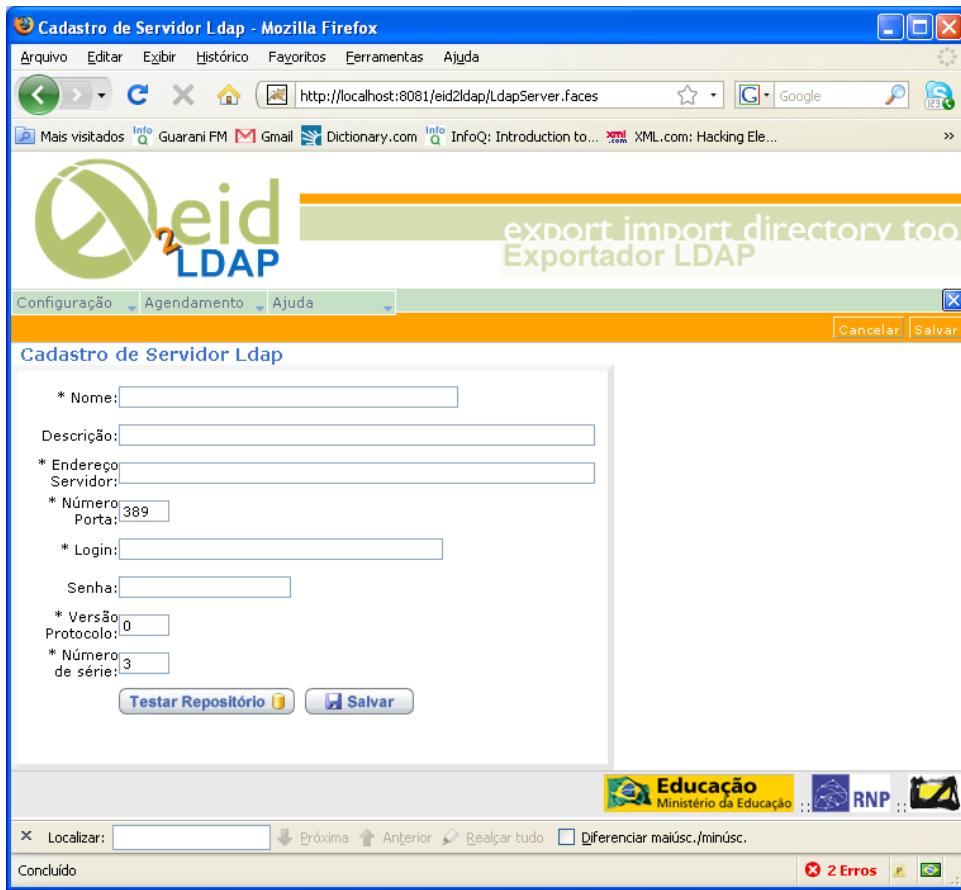


Figura 6.7
Cadastro de servidor LDAP.

A tela da figura 6.7 é exibida após o acionamento do botão *Novo* na tela *Administração de LDAP*. Nesta tela são definidos os dados necessários para o estabelecimento da conexão com o servidor LDAP.

- ▲ Os campos *Nome* e *Descrição* definem os dados utilizados para identificação do servidor nas outras partes do sistema.
- ▲ *Endereço do servidor* indica a URL do servidor em questão, incluindo o protocolo (*ldap://* ou *ldaps://*).
- ▲ *Número da porta* indica a porta em que o servidor escuta.
- ▲ *Usuário* e *Senha* definem os dados do usuário de conexão. Em *Usuário* deve ser especificado o DN completo, e não apenas o login.
- ▲ *Versão do protocolo* indica a versão do protocolo LDAP que será utilizada na comunicação.
- ▲ *Número de série* apresenta o número de série do último registro EID processado pelo EID2LDAP.



Cadastramento do XSLT

Figura 6.8
Configuração/
Transformação.



A tela da figura 6.8 apresenta a interface de *Administração de Arquivos XSLT*, que lista todas as transformações cadastradas.

Ao acionar o botão *Novo* a tela de *Cadastro de Arquivos XSLT* é exibida.

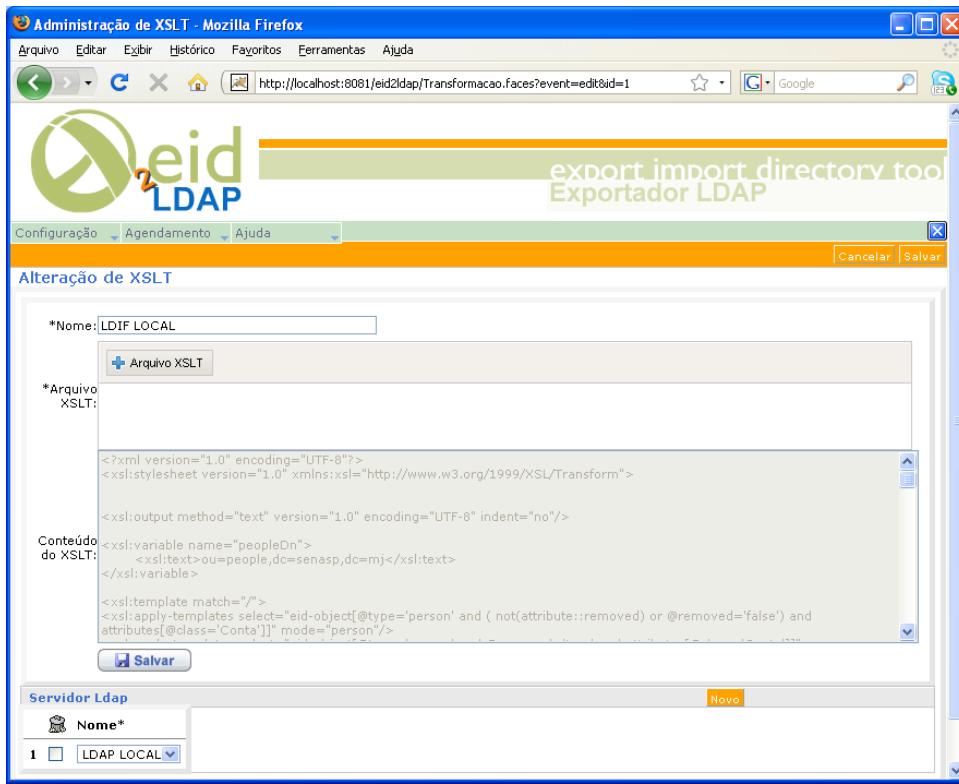


Figura 6.9
Tela de
cadastro XSLT.

A figura 6.9 apresenta a tela de cadastro de XSLT, responsável pelo cadastro do XSLT e associação com o LDAP.

- ▲ Como o XSLT é específico ao formato usado no LDAP, deve ser associado ao LDAP.
- ▲ Como vários LDAPs podem ter a mesma estrutura, um mesmo XSLT pode ser cadastrado para mais de um LDAP.

Definição de agendamento

Figura 6.10
Agendamento/
Agendamento
Servidor LDAP.



A tela *Administração de Agendamentos* permite visualizar e editar os agendamentos cadastrados no sistema. Ela lista todos os agendamentos cadastrados e o estado dos mesmos, que pode ser:

- ▲ Finalizado;
- ▲ Aguardando;
- ▲ Em execução.

Não é possível cancelar um agendamento durante a sua execução.

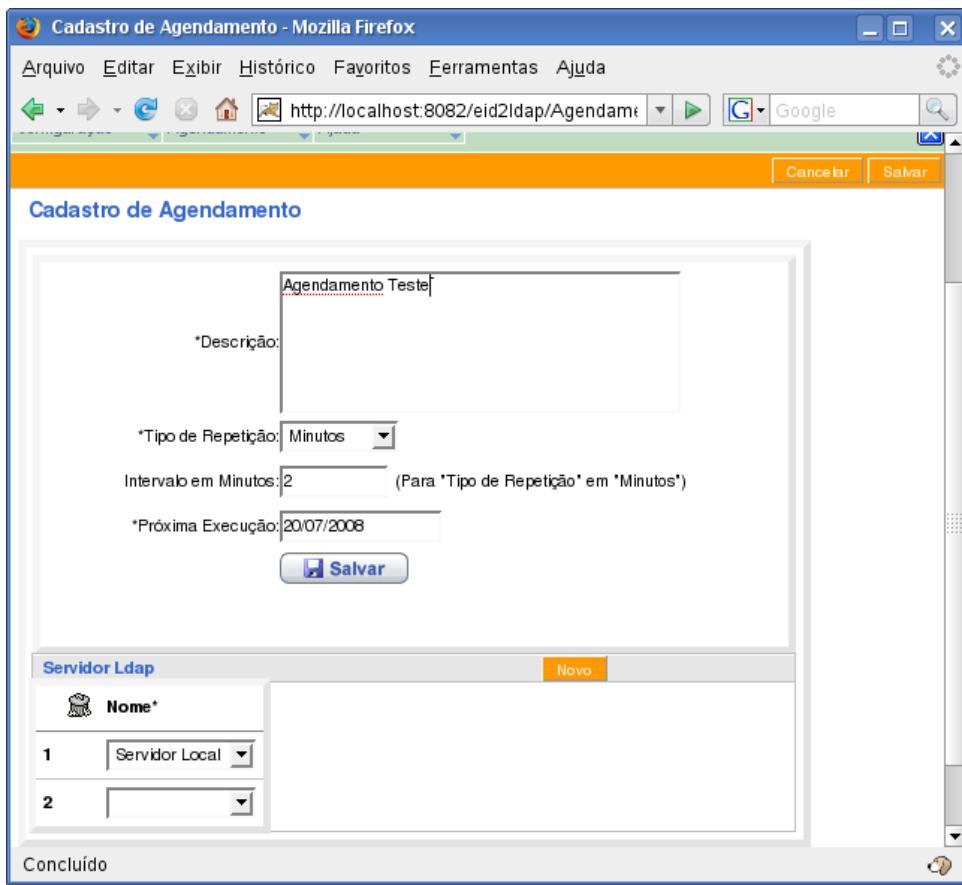


Figura 6.11
Interface para
cadastro de
agendamento.

A tela da figura 6.11 apresenta a interface para cadastro de um agendamento, exibida quando é acionado o botão *Novo* da tela de *Administração de Agendamentos*.

O critério para o início da execução de um agendamento é se a data do agendamento é menor que a atual.

- ▲ O campo *Tipo de Repetição* indica como será o incremento no agendamento da próxima execução: diário, semanal, mensal etc.
- ▲ O campo *Intervalo em minutos* somente será utilizado se o tipo de repetição for em minutos.
- ▲ O campo *Próxima Execução* indica a data em que será iniciada a execução do primeiro agendamento.
- ▲ Os LDAPs a serem atualizados com esta configuração são definidos no painel *Servidor LDAP*.

Verificação do log

Figura 6.12
Agendamento/
Resultado do
Agendamento.

	Descrição do Agendamento	Data Início	Data Fim	Situação	Número do Processamento	Visualizar
<input type="checkbox"/>	Agendamento Teste	25/06/2008 13:53	25/06/2008 13:53	FINISHED_ERRORS	4	
<input type="checkbox"/>	Agendamento Teste	25/06/2008 13:56	25/06/2008 13:56	FINISHED	5	

A figura 6.12 apresenta a tela *Resultado de Agendamento*, que exibe dados sobre os agendamentos em execução ou já executados. Cada execução gera uma entrada. São informadas as datas de início e término da execução, o número do processamento que indica quantas vezes o agendamento foi executado e a situação, que pode ser:

- ▲ FINISHED: Execução finalizada com sucesso;
- ▲ FINISHED_ERRORS: Execução finalizada com erro.

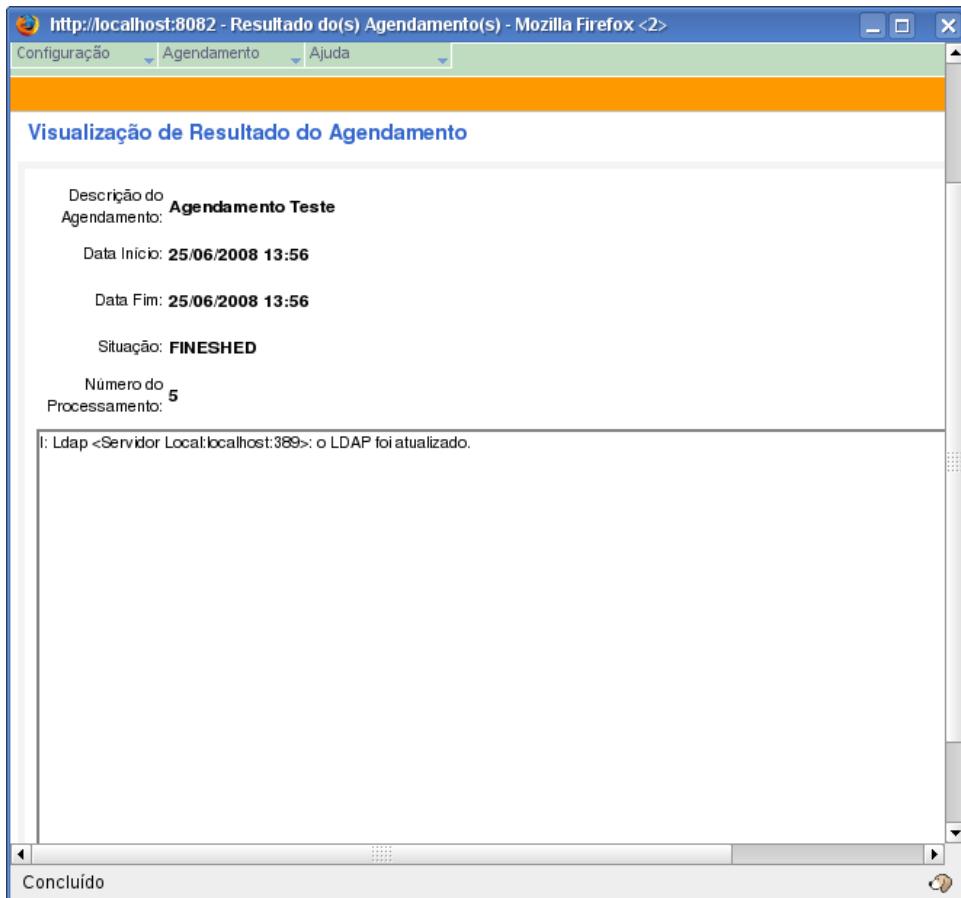


Figura 6.13
Visualização de resultado de agendamentos.

A tela da figura 6.13 exibe a interface de *Visualização de Resultado de Agendamentos*, que é acessada através do botão *Visualizar* da tela *Resultado do Processamento*. Nela é possível visualizar informações detalhadas sobre a execução. Se algum erro ocorreu é detalhado nesta tela.

Problemas comuns

- Erros de sintaxe
 - ▀ Em função de dados malformados importados das fontes
- Solução
 - ▀ Correção do dado na fonte, seguida por sua reimportação
 - ▀ Utilização de scripts de conversão

O LDAP é bastante rígido quanto à sintaxe de alguns atributos, como *mail*, *telephoneNumber* etc.

Durante a exportação podem ocorrer erros dessa natureza em função de dados malformados importados das fontes.

A solução mais adequada é a correção do dado na fonte, seguida por sua reimportação. Na impossibilidade de fazê-lo, pode-se também utilizar scripts de conversão no *leiaute de destino* da ETC, criando-se regras de validação. Algumas regras estão disponibilizadas na seção FAQ do site do projeto, como validação de e-mail e CPF.



6

Roteiro de Atividades Alimentação de diretórios com EID2LDAP

Tópicos e conceitos

- Introdução
 - ▲ EID2LDAP
 - ▲ Características
- Arquitetura
 - ▲ XML do EID
 - ▲ XSLT
 - ▲ Processamento do LDIF
- Configuração e uso

Competências técnicas desenvolvidas

- Mapeamento de dados do metadiretório para diretório LDAP, escalonamento de atualizações.

Tempo previsto para as atividades

- 40 minutos

Servidor de sala de aula

- Os arquivos para instalação do sistema encontram-se no servidor.



Atividade 1 – Inicialização do agente

Abra o aplicativo EID2LDAP no browser através da URL: http://<IP_VM>:8080/eid2ldap.

Inicie o agente escalonador de processos, acessando o menu *Agendamento/Agente Gerenciador de Agendamento* e acionando o comando *Iniciar* para iniciá-lo.

Atividade 2 – Configuração do servidor LDAP

Configure um servidor LDAP local:

1. Acesse o menu *Configuração/Servidor LDAP*.
2. Acione o comando *Alterar* do LDAP local.
3. Altere os dados de conexão para seu servidor LDAP local deixando-os como abaixo:
 - ▲ Nome: LDAP local
 - ▲ Descrição: Servidor LDAP local
 - ▲ Endereço Servidor: IP da sua VM
 - ▲ Número Porta: 389
 - ▲ Login: cn=admin,dc=<instituição>,dc=br (ex. **cn=admin,dc=ufmg,dc=br**)
 - ▲ Senha: 1234
 - ▲ Versão Protocolo: 3
 - ▲ Número de série: -1
4. Acione o comando *Salvar*.

Atividade 3 – Configuração de uma transformação

Configure uma transformação e associe-a ao servidor LDAP local. Para tanto:

1. Acione o menu *Configuração/Transformação*.
2. Acione o comando *Alterar* para modificar a transformação **brEduPerson** já cadastrada.
3. Informe um nome para a transformação.
4. No campo *Conteúdo* do XSLT, no arquivo exibido, substitua **\${RAIZ_BASE_LDAP}** pelo DN da raiz do diretório (dc=<instituição>,dc=br)
5. No detalhe *Servidor LDAP*, o servidor LDAP configurado na Atividade 2 deve estar selecionado.
6. Acione o comando *Salvar*.



Atividade 4 – Executar teste padrão: leitura no diretório

Execute o teste padrão para leitura no metadiretório:

1. Verificar carga da classe *Conta*:

Utilizando um navegador web, acesse a URL a seguir, trocando <servidor> pelo endereço do servidor EID:

```
http://<servidor>:8080/eid/services/EidService/
getGuids?condition=select%20c.eidObject.stringID%20
from%20Conta%20c%20where%20c.eidObject.unifiedDomain%20
%3D%20true%20and%20c.login%20!=%20null%20and%20c.
eidObject.serialNumber%20%3E%20(select%20max(e.
serialNumber)- 1000%20from%20EidObject%20e%20where%20e.
unifiedDomain%20%3D%20true)
```

2. Observe o resultado de busca e se a página exibida assemelha-se ao trecho:

```
<ns:getGuidsResponse>
<ns:return>CIVZAGRA-CXJFBAAA</ns:return>
<ns:return>KHWRXWEA-CXJFBAAA</ns:return>
<ns:return>MEMJJEJA-DXJFBAAA</ns:return>
<ns:return>OYFQQYMA-CXJFBAAA</ns:return>
<ns:return>QACX0EDA-DXJFBAAA</ns:return>
<ns:return>QGEDIIFA-BXJFBAAA</ns:return>
</ns:getGuidsResponse>
```

Atividade 5 – Definição de um agendamento

Agende a atualização do diretório LDAP.

1. Acesse o menu *Agendamento/Agendamento Servidor LDAP*.
2. Acione o comando *Novo* e configure os parâmetros do agendamento, de forma que o LDAP seja atualizado.
 - 2.1. Informe uma *Descrição* para o agendamento.
 - 2.2. Informe o *Tipo de Repetição* como NENHUM.
 - 2.3. Deixe o campo *Intervalo em minutos* em branco.



- 2.4. No campo *Próxima Execução* informe data e hora atual, no formato: dd/mm/aaaa hh:mm.
- 2.5. No campo *Máximo de erros* informe 0.
- 2.6. No campo *Nome do Servidor LDAP* informe o LDAP cadastrado.
- 2.7. Acione o comando *Salvar*.
- 2.8. Aguarde alguns minutos até a importação ser realizada com sucesso; para verificar acesse o menu *Agendamento/Resultado Agendamento*.
- 2.9. Observe os dados no LDAP através do Apache DirectoryStudio ou utilizando o seguinte comando no Linux:

```
# ldapsearch -x -D "cn=admin,dc=<nome_da_instituição>,dc=br" -W
```

Atividade 6 – Desativação e alteração de registros no metadiretório

Os procedimentos a seguir fazem com que as alterações no metadiretório sejam refletidas no LDAP.

1. Acesse o EID http://IP_VM:8080/eid, menu *EID/Gestão de Pessoas*.
2. Preencha os parâmetros de pesquisa com:
 - 2.1. *Classe*: Conta
 - 2.2. *Classe de Atributos*: login
 - 2.3. *Valor do atributo*: usuario1
 - 2.4. Clique em *Pesquisar*.
3. Selecione o usuário para ser desativado clicando no check box abaixo do ícone de lixeira e no botão *Desativar* na barra de menus.
4. Preencha os parâmetros de pesquisa novamente com:
 - 4.1. *Classe*: Conta
 - 4.2. *Classe de Atributos*: login
 - 4.3. *Valor do atributo*: usuario2
5. Clique no ícone *Atualizar* do registro pesquisado, e vá para a aba *Identificação*. Altere a data de nascimento para 01/01/1990.
6. Clique em *Salvar*.
7. Acesse o EID2LDAP http://IP_SERVIDOR:8080/eid2ldap, menu *Agendamento/Agendamento Servidor LDAP*.
8. Clique no botão *Alterar* e em seguida no botão *Salvar* forçando com que a exportação seja executada novamente.



9. Aguarde alguns segundos até a importação ser realizada com sucesso; para verificar acesse o menu *Agendamento/Resultado Agendamento*.
10. Observe os dados no LDAP através do Apache DirectoryStudio, e verifique que o *usuário1* foi removido do LDAP, já que foi marcado como *Desativado* no metadiretório através do EID. Já o registro do *usuário2* teve sua data de nascimento alterada para 01/01/1990. As alterações feitas no metadiretório foram refletidas no LDAP após a exportação dos dados via EID2LDAP.

7

Plataforma Shibboleth

- ▶ Introdução
- ▶ Provedor de Identidade (IdP)
- ▶ Provedor de Serviço (SP)
- ▶ WAYF (**W**here **A**re **Y**ou **F**rom?) / DS (**D**iscovery **S**ervice)
- ▶ Metadata
- ▶ Funcionamento
- ▶ Atividades

Introdução

- ▶ O que é Shibboleth?
 - ▶ Terminologia
 - ▶ Palavra que distingue pessoas de um grupo das pessoas de outro grupo
 - ▶ Origem bíblica
 - ▶ Diferenciação entre as tribos dos efraimitas e dos gileaditas

Nesta sessão apresentaremos o Shibboleth, um sistema de autenticação e autorização via web, descreveremos os seus componentes típicos (Provedor de Identidade, Provedor de Serviço, WAYF e Metadata) e demonstraremos o seu funcionamento.

O termo “shibboleth” denota uma palavra usada para distinguir pessoas de um grupo das pessoas de outro. A origem deste termo remete ao Velho Testamento (Juízes, 12: 1-15), onde foi usado para distinguir entre membros de duas tribos semitas, os gileaditas e os efraimitas, que travaram uma grande batalha. Os gileaditas, vencedores, bloquearam as passagens do Jordão para evitar que os efraimitas sobreviventes pudessem escapar. As sentinelas exigiam que todo passante dissesse “shibboleth”; como os efraimitas não tinham o fonema /x/ em seu dialeto,



só conseguiam pronunciar “sibboleth” (com /si/ na primeira sílaba), sendo assim reconhecidos e executados.

- O que é Shibboleth?
 - Projeto de *middleware* da Internet2
 - SAML (**S**ecurity **A**ssertion **M**arkup **L**anguage)
 - Padrão definido pela OASIS (**O**rganization for the **A**dvancement of **S**tructured **I**nformation **S**tandards)
 - Acesso federado
 - Autenticação
 - Autorização
 - SSO (**S**ingle **S**ign-**O**n)

O Shibboleth é um projeto da Internet2 Middleware Initiative, que consiste na implementação de padrões amplamente utilizados para autenticação e autorização federada via web, principalmente o SAML (Security Assertion Markup Language) criado pela OASIS (Organization for the Advancement of Structured Information Standards). Além disso, o Shibboleth possibilita que o usuário acesse diferentes aplicações web, autenticando-se apenas uma vez (Single Sign-On) em sua instituição de origem.

- Componentes:
 - Provedor de Identidade (IdP)
 - Provedor de Serviço (SP)
 - WAYF (**W**here **A**re **Y**ou **F**rom?) / **D**iscovery **S**ervice
 - Metadata

O Shibboleth é composto majoritariamente pelos provedores de identidade e de serviço, que proveem, respectivamente, autenticação e autorização. Contudo, uma federação Shibboleth geralmente apresenta dois componentes adicionais: serviço de WAYF (Where Are You From?) ou Discovery Service (Shibboleth 2.x), usados para localizar o provedor de identidade de um usuário, e serviço de Metadata, usado para concentrar as informações dos provedores pertencentes à federação.

- Por que Shibboleth?
 - Desenvolvido para tratar os seguintes desafios:
 - Múltiplas senhas requeridas para múltiplas aplicações
 - Escalabilidade no gerenciamento de múltiplas aplicações
 - Problemas de segurança associados ao acesso de serviços de terceiros
 - Privacidade
 - Interoperabilidade dentro e entre organizações

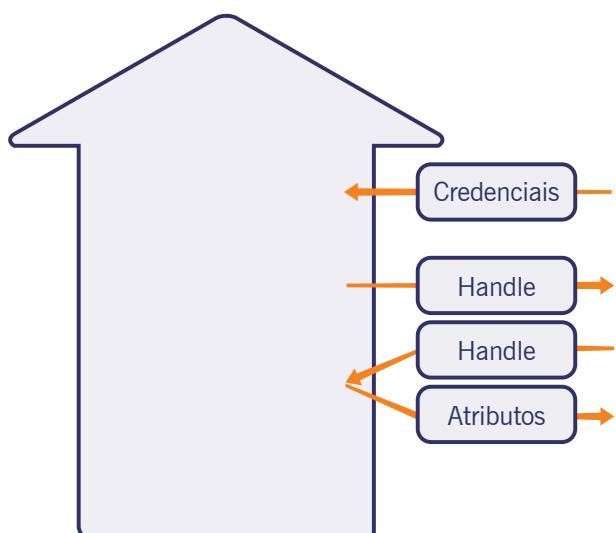


- ▲ Liberdade de escolha das tecnologias de autenticação para as instituições
- ▲ Controle de acesso efetuado a partir dos provedores de serviço
- ▶ Aplicações compatíveis:
 - ▲ Google Apps
 - ▲ Media Wiki
 - ▲ Moodle, Joomla, Drupal
 - ▲ Blackboard
 - ▲ ProQuest
 - ▲ Confluence
 - ▲ Microsoft DreamSpark
 - ▲ MAMS (Austrália)
 - ▲ SIR (Espanha)
 - ▲ SURFnet Federation (Holanda)
 - ▲ SWAMID (Suécia)
 - ▲ SWITCHaaI (Suíça)
 - ▲ UK Federation (RU)
 - ▲ WAYF (Dinamarca)
 - ▲ CAFé (Brasil)

Provedor de Identidade (IdP)

- ▶ Identidade
 - ▶ Autenticação
 - ▲ Web SSO
 - ▶ Atributos

Figura 7.1
Provedor de
Identidade.



O provedor de identidade é responsável por fornecer a autenticação e os atributos do usuário, possibilitando que o provedor de serviço faça a autorização ao recurso. A autenticação e a entrega de atributos são realizadas da seguinte forma: o usuário envia as suas credenciais, que são devidamente verificadas pelo provedor de identidade; o provedor de identidade envia um *handle* para o provedor de serviço, atestando que o usuário foi autenticado; o provedor de serviço envia este *handle* para o provedor de identidade, solicitando a entrega de atributos referentes ao usuário em questão; e, por fim, o provedor de identidade envia esses atributos para o provedor de serviço.

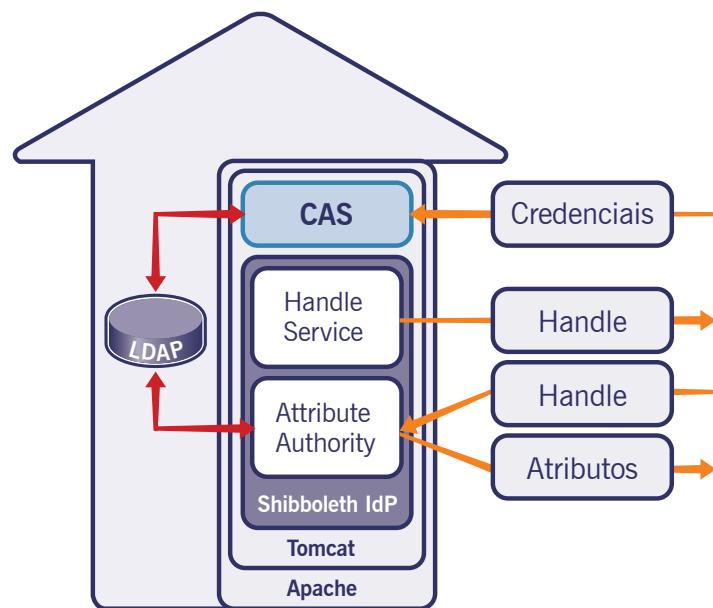


Figura 7.2
Shibboleth
Identity Provider.

- Identidade
 - ▀ Autenticação
 - ▲ Web SSO
 - ▀ Atributos
- Shibboleth Identity Provider
 - ▀ Handle Service
 - ▀ Attribute Authority
- CAS
 - ▀ Web SSO
- LDAP
 - ▀ Autenticação
 - ▀ Atributos

A instalação padrão de um provedor de identidade da federação CAFe é composta por três elementos principais:

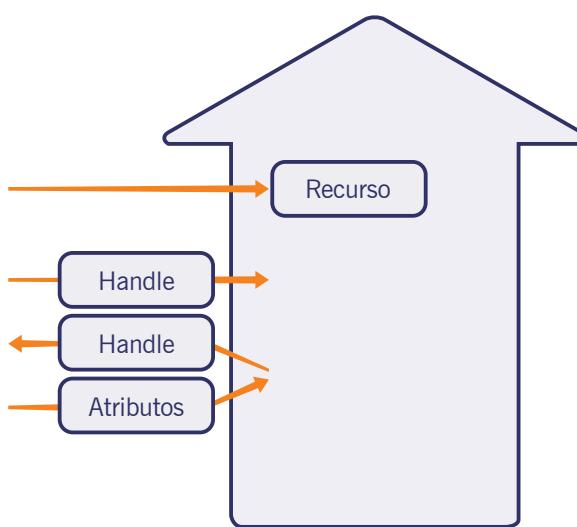
- ▲ Shibboleth Identity Provider – Serviço de *middleware*, responsável por intermediar a autenticação e o envio de atributos.
- ▲ Central Authentication Service – Serviço de autenticação web Single Sign-On, responsável pela interface de autenticação com o usuário.
- ▲ OpenLDAP – Servidor de diretório, responsável por armazenar os atributos dos usuários e validar as suas credenciais.

Além disso, é importante ressaltar que o Shibboleth IdP pode trabalhar com outros servidores de autenticação e atributos.

Provedor de Serviço (SP)

- ▲ Serviço
- ▲ Recurso
- ▲ Autorização

Figura 7.3
Service Provider
(SP).



O provedor de serviço (service provider – SP) é responsável por fazer a autorização do usuário e disponibilizar o acesso ao recurso, através da autenticação e dos atributos disponibilizados pelo provedor de identidade. A autorização e o acesso ao recurso são realizados da seguinte forma: o usuário solicita o acesso ao recurso; o provedor de serviço solicita que ele se autentique no provedor de identidade da sua instituição; o provedor de identidade envia um *handle* atestando a autenticação do usuário; o provedor de serviço envia o *handle* para o provedor de identidade solicitando os seus atributos; e, por fim, o provedor de serviço processa a autorização baseado nos atributos do usuário e disponibiliza o acesso ao recurso.

- ▲ Serviço
- ▲ Recurso
- ▲ Autorização

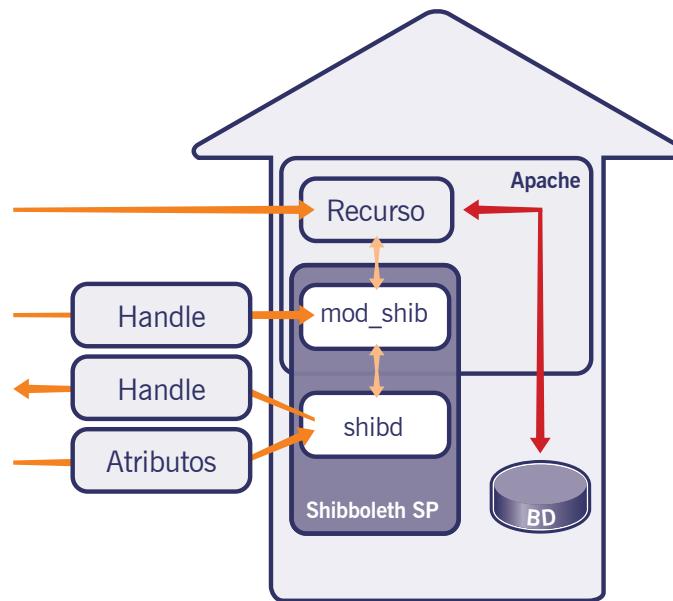


Figura 7.4
Shibboleth
Service Provider.

- ▲ Shibboleth Service Provider
 - ▲ mod_shib
 - ▲ Módulo do Apache
 - ▲ shibd
 - ▲ Daemon

A instalação padrão de um provedor de serviço da federação CAFE é baseada no Shibboleth Service Provider, que, por sua vez, é composto por dois elementos:

- ▲ mod_shib – Módulo do Apache, responsável por controlar a autorização e o acesso ao recurso.
- ▲ shibd – Daemon, responsável por intermediar a solicitação de autenticação e de atributos.

Além disso, é importante ressaltar que o Shibboleth SP pode trabalhar com o servidor HTTP Microsoft IIS.

WAYF / DS

- ▶ De onde você é?
- ▶ Qual é o seu provedor de identidade?

O serviço de WAYF (Where Are You From?) é responsável por identificar o provedor de identidade do usuário. Quando o usuário tenta acessar um recurso disponibilizado por um provedor de serviço da federação, ele é redirecionado para o WAYF para que possa indicar o seu provedor de identidade e proceder corretamente com a autenticação. A partir da versão 2.x o Shibboleth disponibiliza o Discovery Service (DS) que é similar ao WAYF. Ele utiliza informações do cookie no browser para armazenar a instituição do usuário.

Metadata

- ▶ Arquivo de configuração
 - ▶ SAML Metadata (schema) + Extensões Shibboleth
 - ▶ Compartilhado entre os provedores da federação

O serviço de Metadata é apenas um arquivo de configuração padronizado e compartilhado entre os provedores de identidade e de serviço da federação.

- ▶ Metadados
 - ▶ Relacionamento de confiança entre provedores
 - ▶ Certificados
 - ▶ Chaves públicas
 - ▶ Informações para a comunicação entre provedores
 - ▶ IDs
 - ▶ URLs
 - ▶ Protocolos

Através deste arquivo é estabelecida a relação de confiança entre os provedores da federação, utilizando certificados digitais ou chaves públicas. Além disso, o arquivo de metadados disponibiliza as informações relevantes para a comunicação entre os provedores, como identificadores, URLs e protocolos utilizados.



Funcionamento

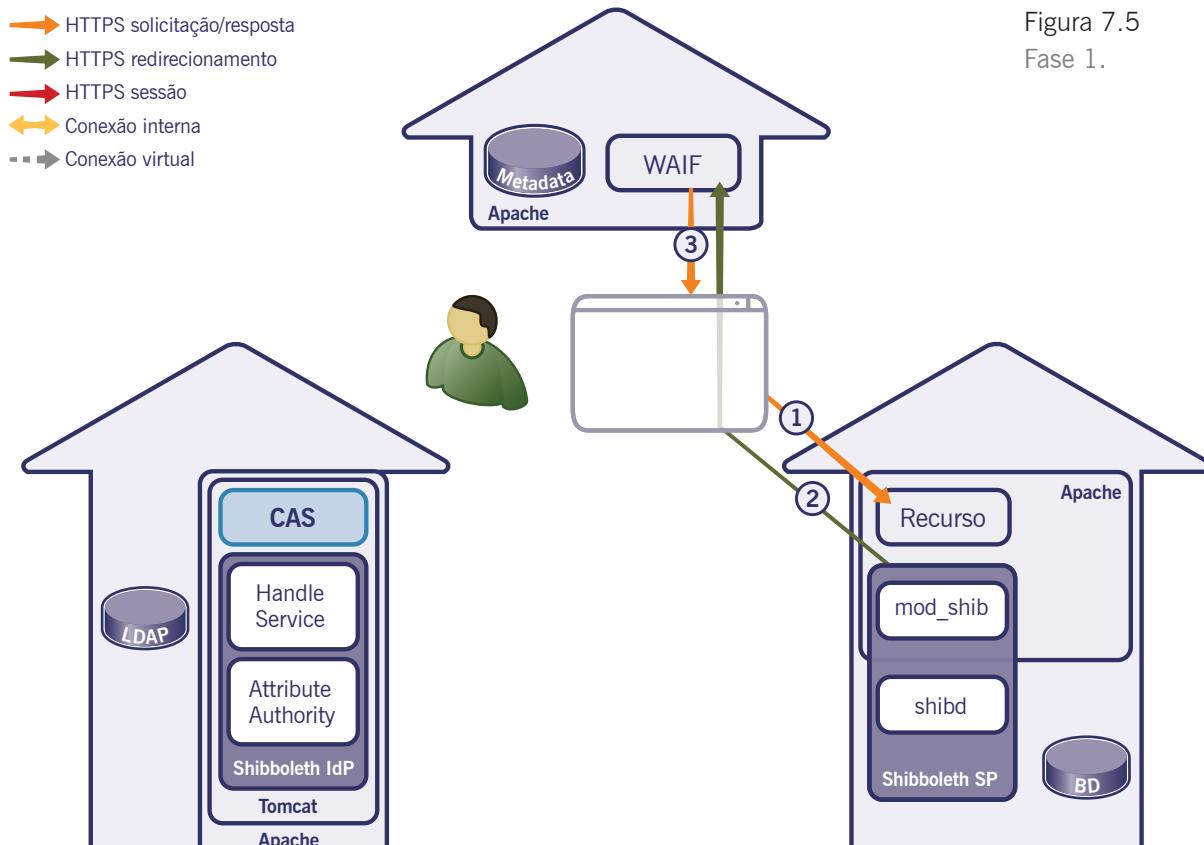


Figura 7.5
Fase 1.

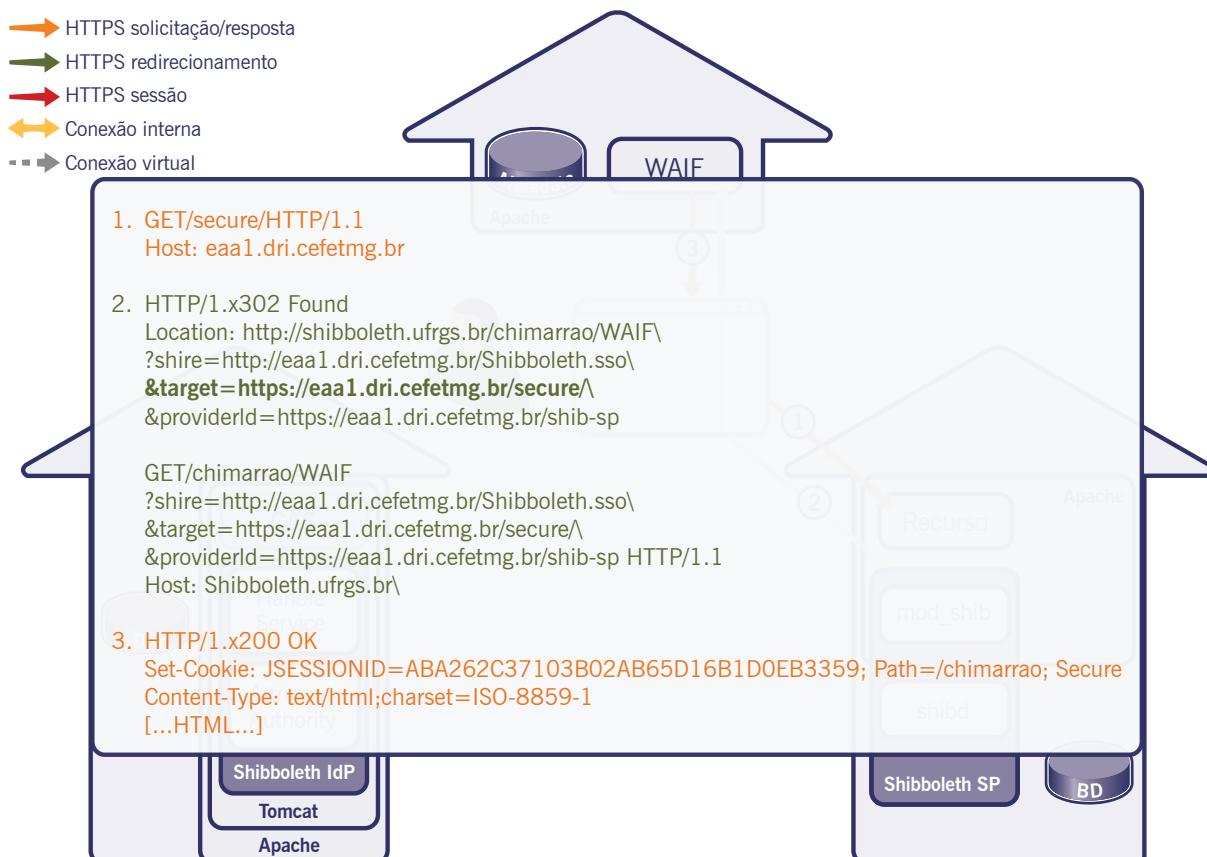
* Essa demonstração foi baseada no Expert Demo da SWITCHaaI.

Fase 1: solicitação de acesso ao recurso e redirecionamento do usuário

1. O usuário inicia o browser e acessa a URL referente ao recurso: <https://eaa1.dri.cefetmg.br/secure>.
2. Como o usuário ainda não está autenticado, o servidor web responde com um redirecionamento HTTP para o servidor WAYF (<http://shibboleth.ufrgs.br>). Como o WAYF precisa saber o provedor de serviço que o usuário está tentando acessar, as informações são enviadas como parâmetros GET.
3. O WAYF responde ao browser com uma página para o usuário selecionar a sua instituição de origem.

Figura 7.6

Fase 1.





Federação Chimarrão

[Sobre a Federação Chimarrão](#) : [Sobre a RNP](#) : [FAQ](#)

Selecione sua instituição do Origem

Para acessar o serviço em 'eaal.dri.cefetmg.br' você precisa se autenticar.

Lembrar a seleção nesta sessão do navegador.
 Lembrar a seleção permanentemente e passar pelo WAYF de agora em diante.

 A federação Chimarrão é mantida para testes das instituições que desejam participar da federação Café.

Figura 7.7
Fase 2.

Fase 2: seleção da instituição de origem

Na página do WAYF, o usuário seleciona a sua instituição de origem, ou seja, o seu provedor de identidade. Essa seleção é armazenada por cookies de sessão no browser do usuário.



Federação Chimarrão

[Sobre a Federação Chimarrão](#) : [Sobre a RNP](#) : [FAQ](#)

Selecione sua instituição do Origem

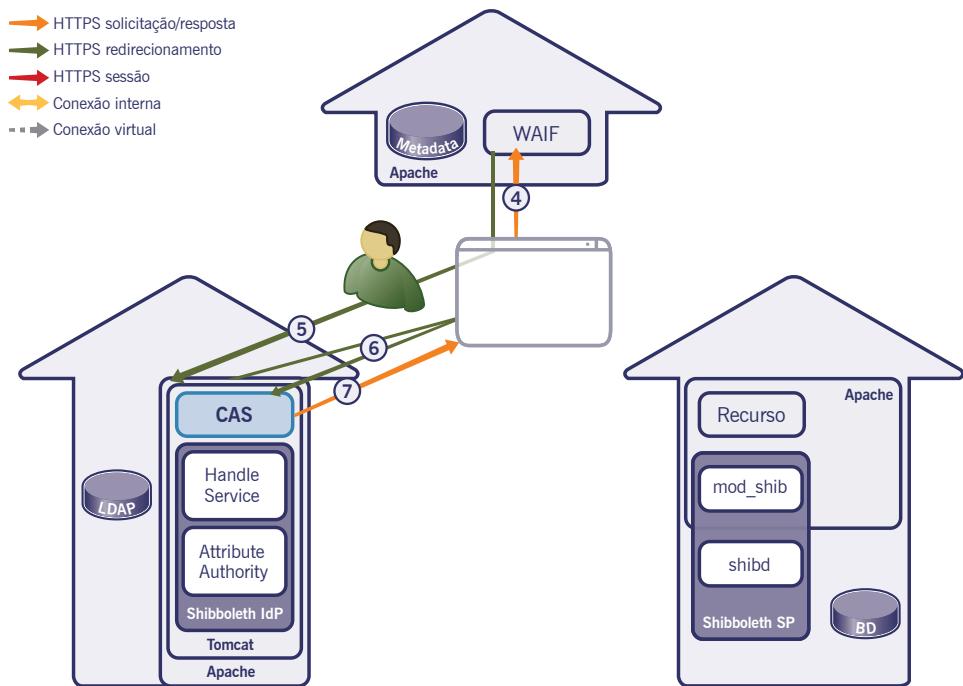
Para acessar o serviço em 'eaal.dri.cefetmg.br' você precisa se autenticar.

Selecionar sua Universidade de origem ...
CEFET-MG Centro Federal de Educação Tecnológica de Minas Gerais
RNP - Rede Nacional de Ensino e Pesquisa
UFC - Universidade Federal do Ceará
UFF - Universidade Federal Fluminense
UFMG - Universidade Federal de Minas Gerais
UFRGS - Universidade Federal do Rio Grande do Sul
UFV - Universidade Federal de Viçosa
UFPE - Universidade Federal de Pernambuco

em diante.
desejam participar

Figura 7.8
Fase 2.

Figura 7.9
Fase 3.



Fase 3: autenticação do usuário na sua instituição de origem

4. O usuário envia a seleção da sua instituição de origem a partir de uma requisição HTTP.
5. Após o envio da requisição do usuário, o WAYF responde com um redirecionamento HTTP para o provedor de identidade do usuário. Os cookies são habilitados para lembrar a escolha do usuário para o checkbox “Lembrar a seleção nesta sessão do navegador”, ou seja, o cookie estará disponível somente durante a sessão atual do browser. O browser do usuário, então, envia uma requisição HTTP para o Shibboleth Handle Service da sua instituição de origem.
6. Como o usuário ainda não está autenticado, o servidor web, protegendo o acesso ao Handle Service, redireciona o browser para o sistema de autenticação Single Sign-On (CAS).
7. O sistema de autenticação Single Sign-On envia a página de login para o browser e habilita os seus cookies.



Figura 7.10
Fase 3.

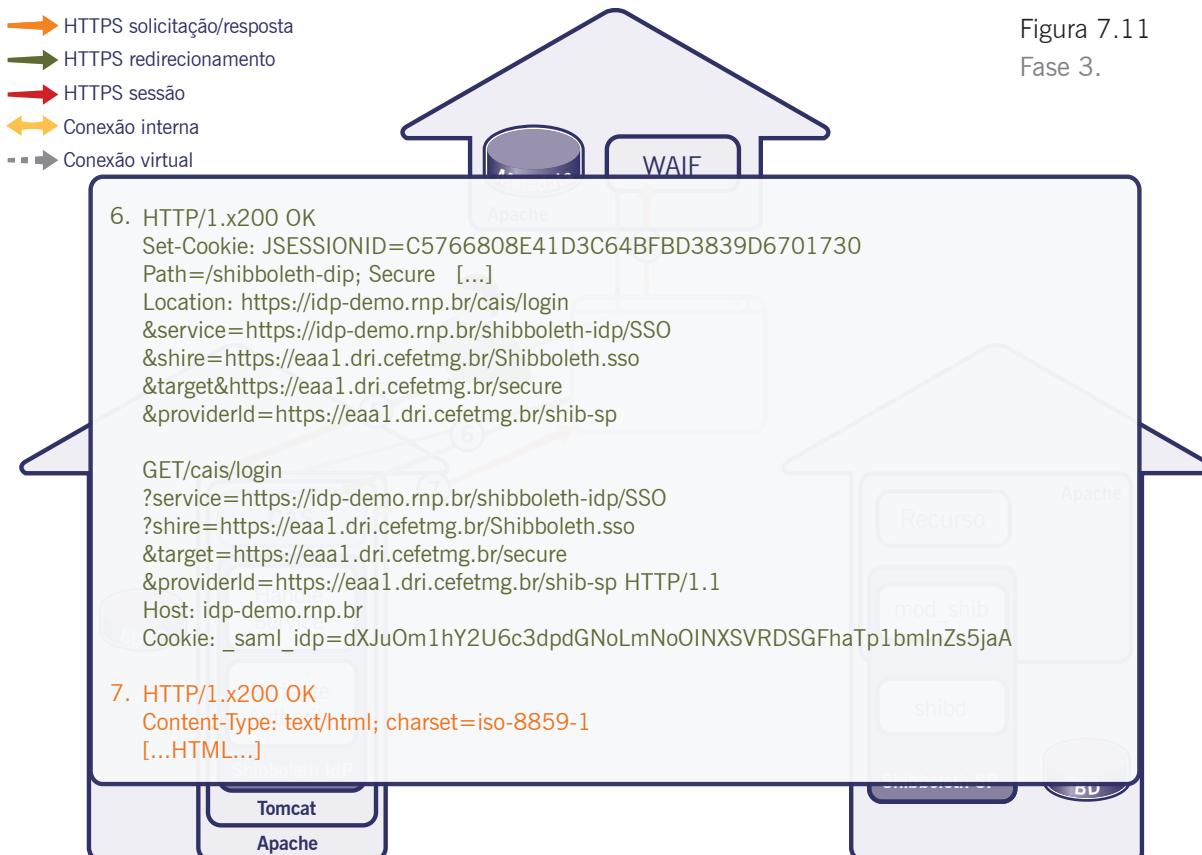


Figura 7.11
Fase 3.

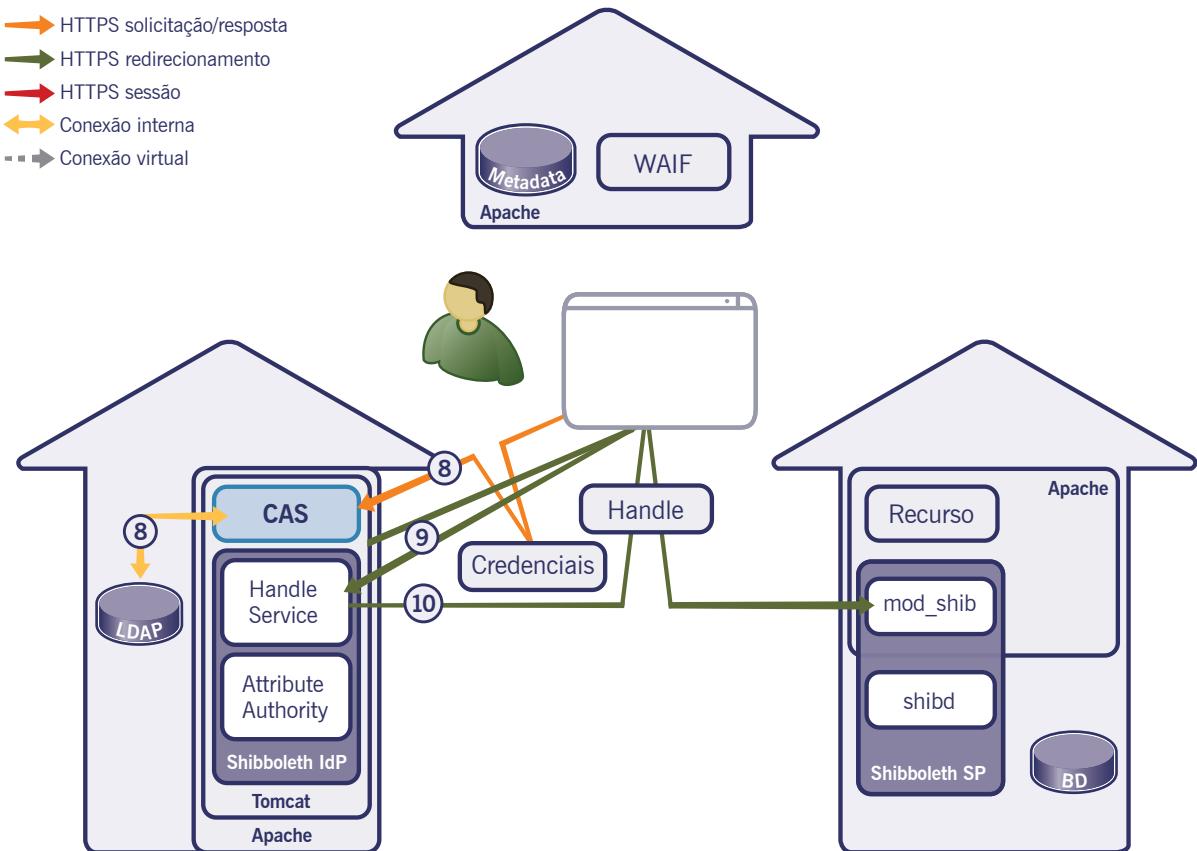
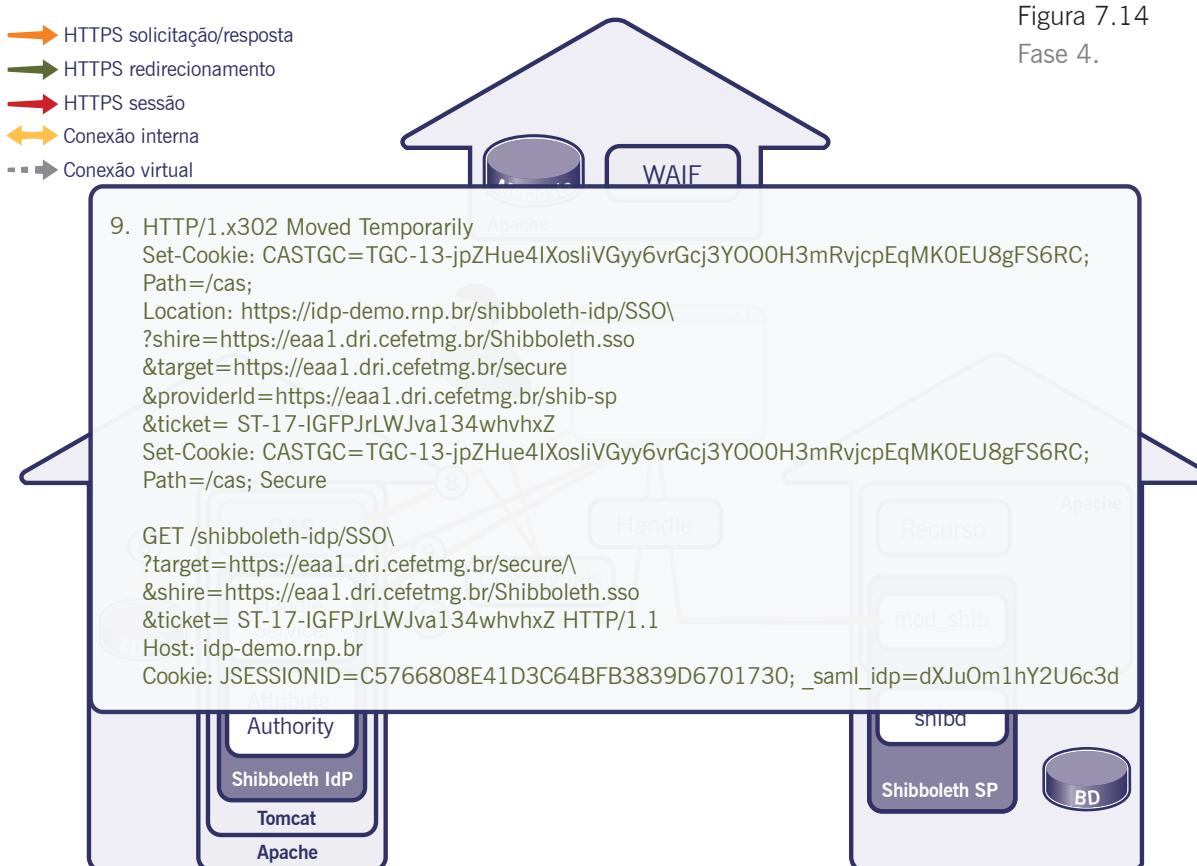
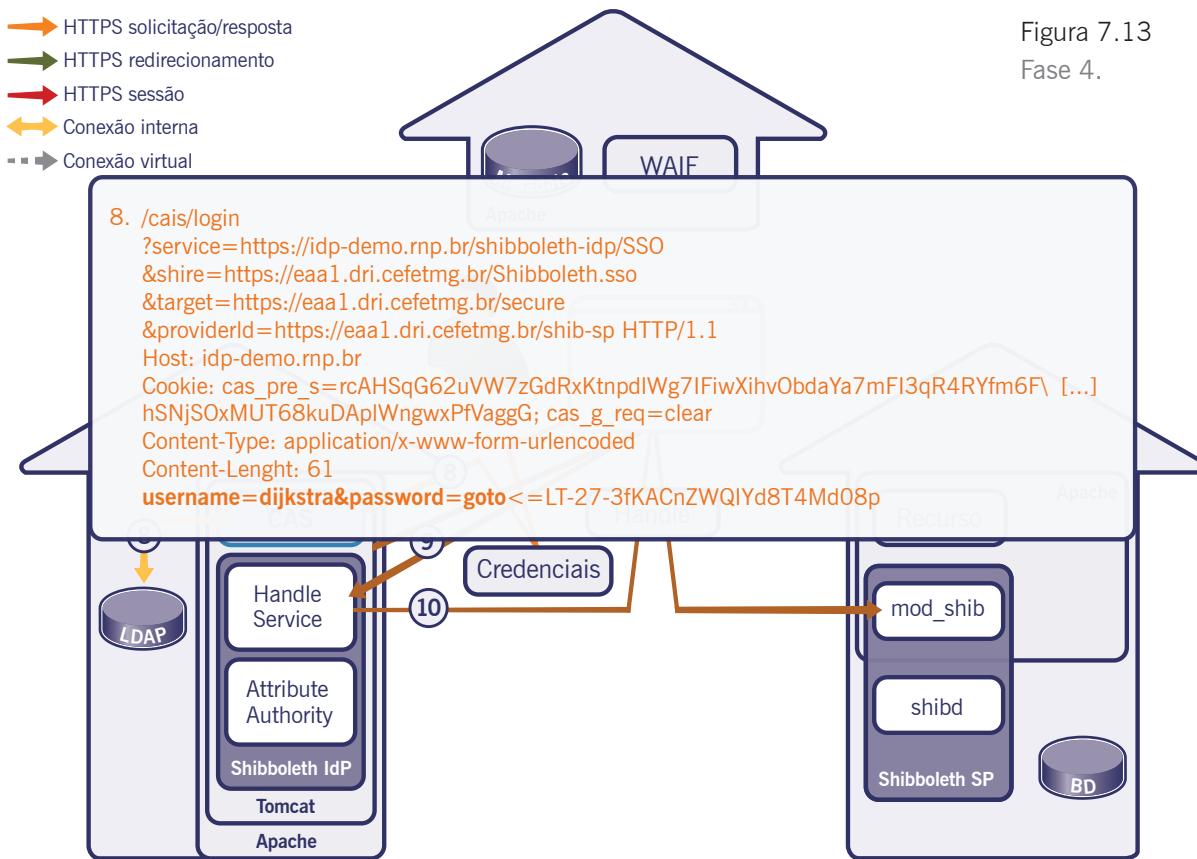


Figura 7.12

Fase 4. Fase 4: acesso ao recurso

- Uma vez que o usuário disponibiliza as suas credenciais — nome de usuário ‘dijkstra’ e senha ‘goto’, neste exemplo —, o browser envia uma nova solicitação para o sistema de autenticação (CAS). O sistema de autenticação, que é independente do Shibboleth, verifica as credenciais do usuário através do diretório LDAP.
- Após o sucesso da autenticação, o browser recebe um pedido de redirecionamento e cookies para enviar ao Handle Service do Shibboleth IdP.
- Baseado nos cookies, o Shibboleth IdP sabe que o usuário foi devidamente autenticado. Então, o Handle Service cria um handle para o usuário. Esse handle é embarcado em um *hidden form* que é enviado pelo browser para o provedor de serviço.



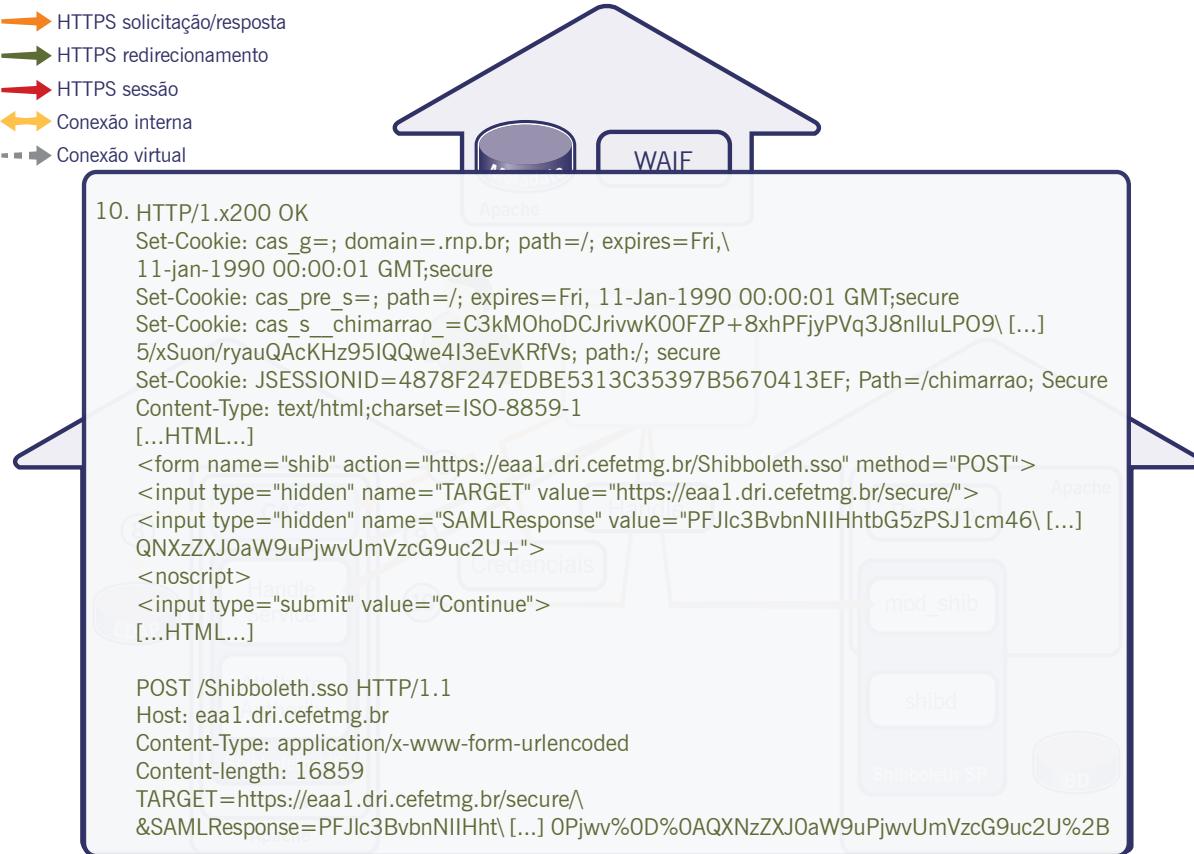


Figura 7.15

Fase 4.

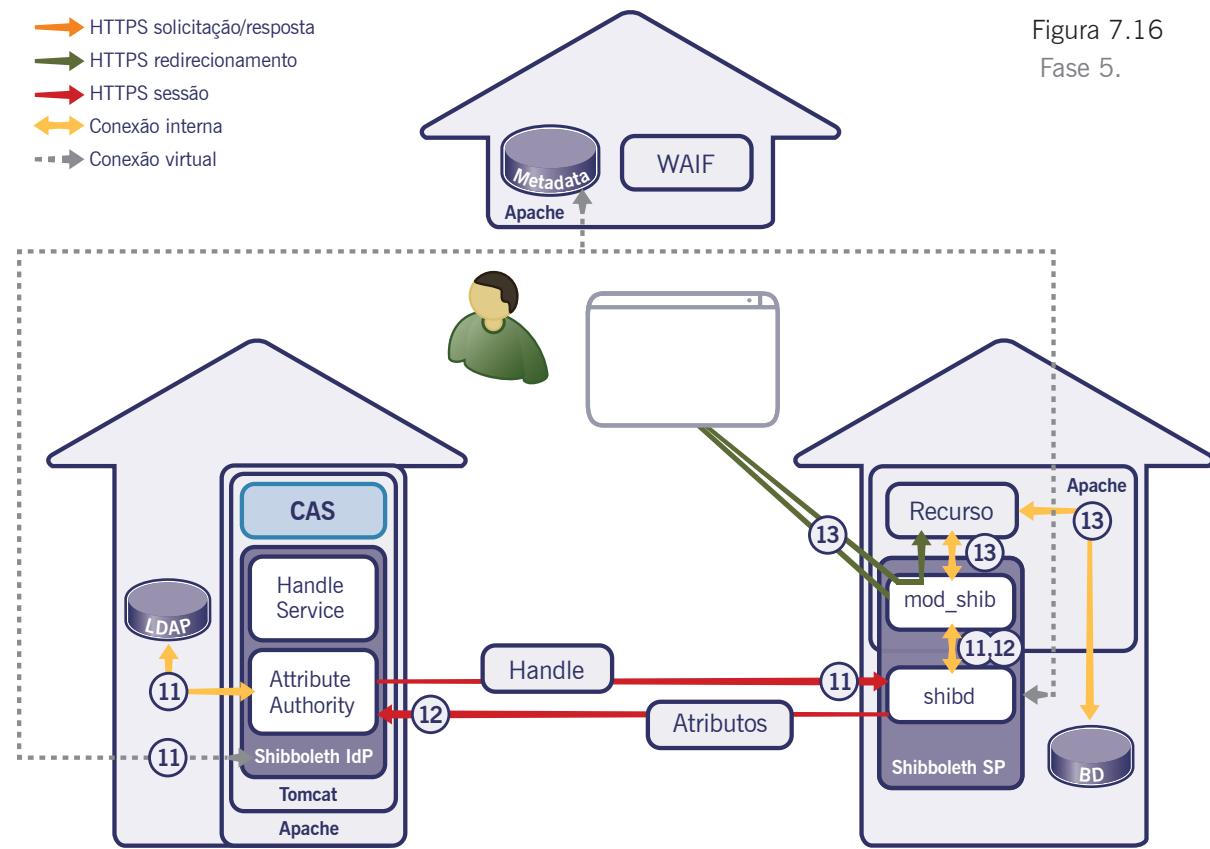


Figura 7.16
Fase 5.

Fase 5: solicitação de atributos

Para decidir se o usuário está autorizado a acessar o recurso, o *mod_shib* examina as regras de acesso do Shibboleth. O seguinte fragmento do arquivo de configuração do Apache habilita o acesso a qualquer usuário da federação com uma sessão válida:

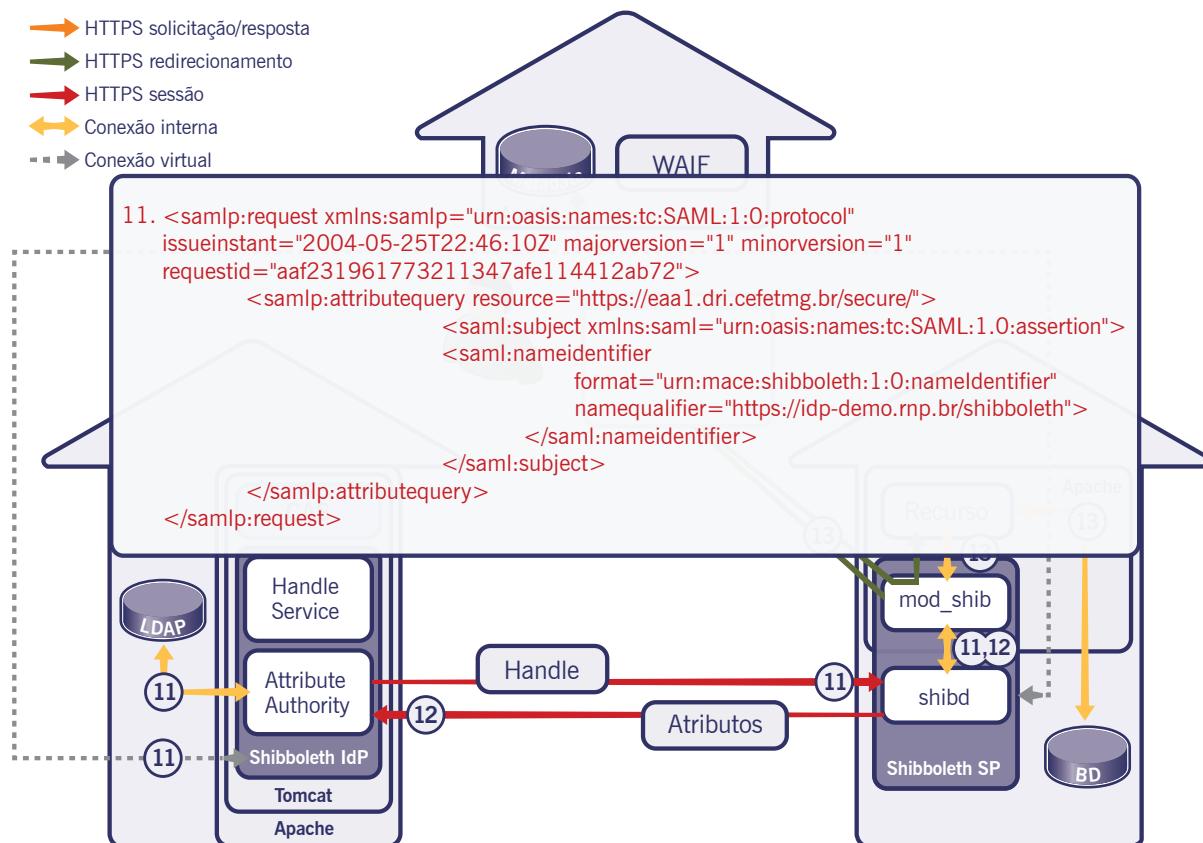
```
<Directory /var/www/secure>
    AuthType shibboleth
    ShibRequireSession On
    require valid-user
</Directory>
```

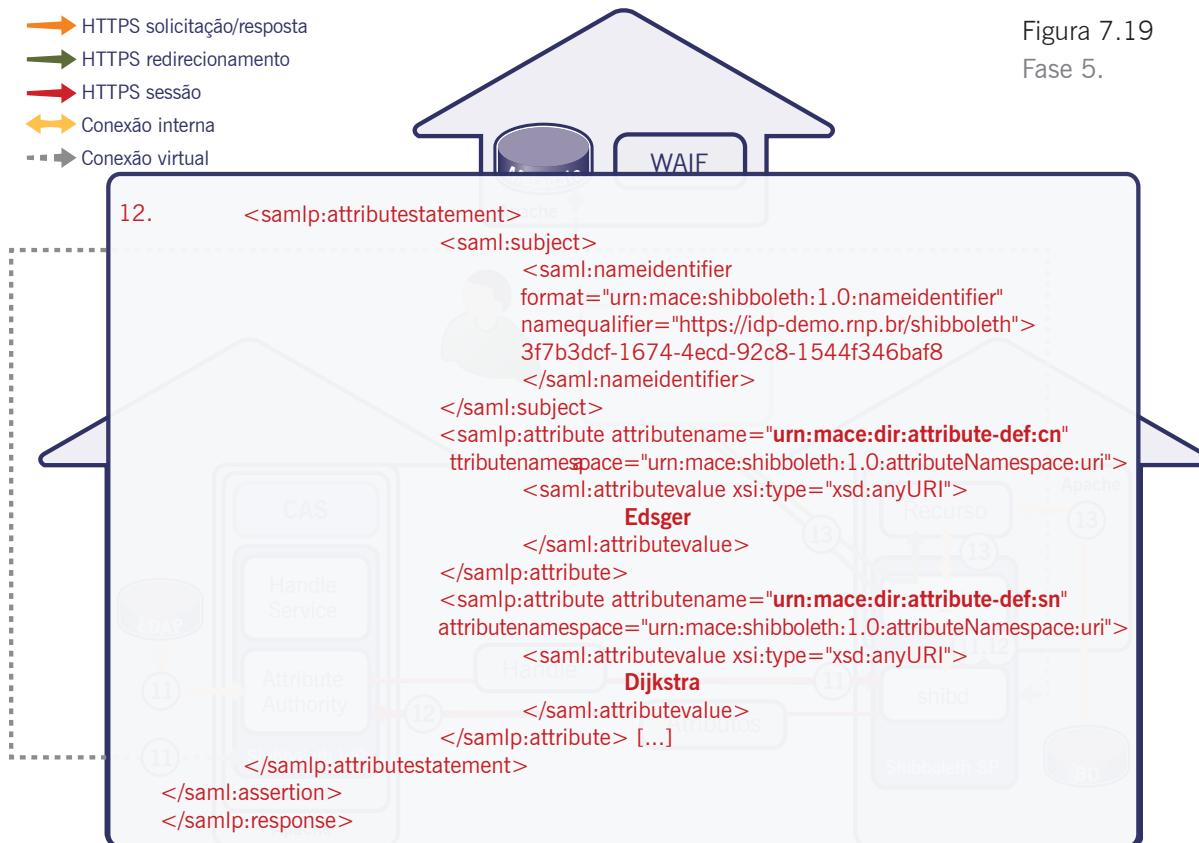
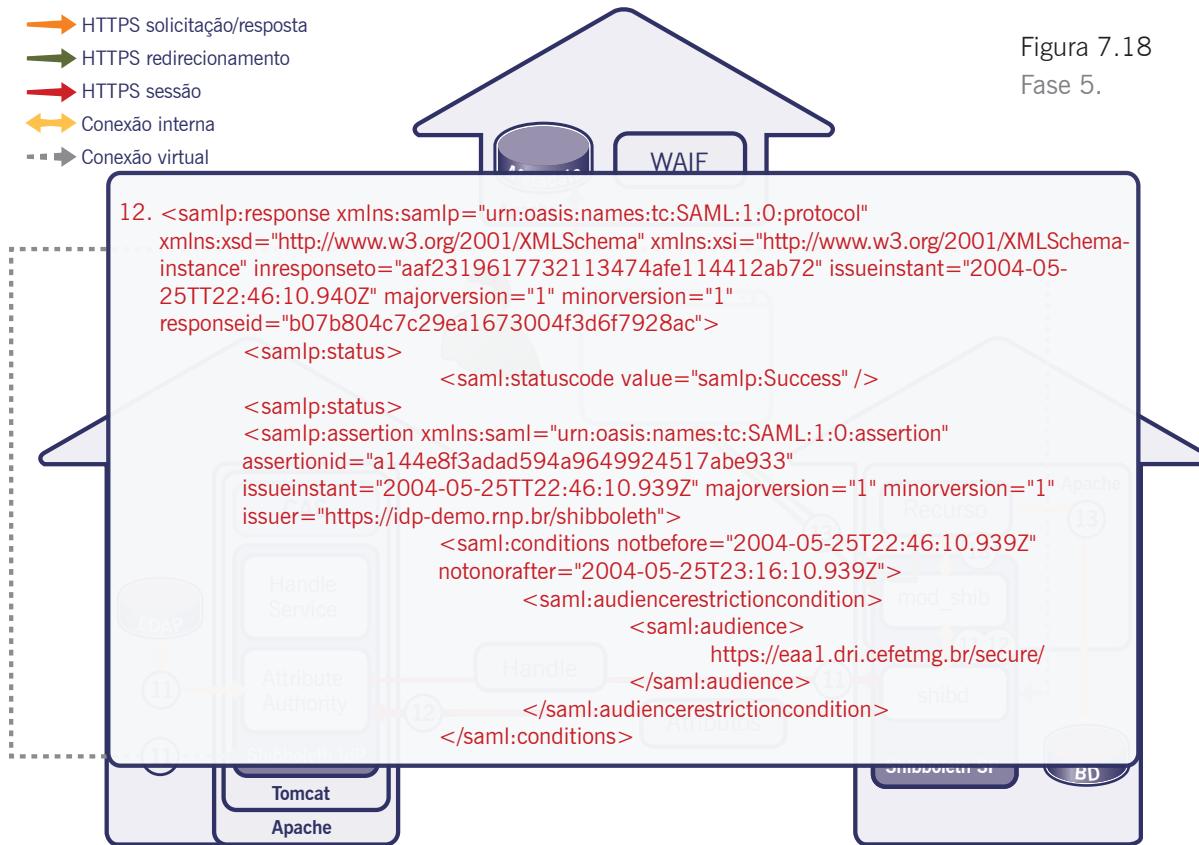
11.O Shibboleth SP, então, solicita ao provedor de identidade todos os atributos disponíveis para o usuário associado ao *handle* recebido no passo anterior.

12. Após a sessão HTTPS ser estabelecida entre o shibd e o Attribute Authority do Shibboleth IdP, o Attribute Authority verifica a identidade do SP com base no certificado enviado pelo shibd. Uma vez que o Attribute Authority recebe a solicitação de atributos, ele verifica se o *handle* é o mesmo gerado pelo Handle Service no passo 10; caso isso seja verdade, ele sabe a que usuário o *handle* se refere. Então, ele verifica o Attribute-Filter, arquivo XML responsável pelas regras que determinam quando um atributo de um determinado usuário pode ser enviado para um determinado provedor de serviço. Após esta verificação, o Attribute Authority envia para o provedor de serviço todos os atributos permitidos de acordo com o arquivo *attribute-filter.xml*.

13. Finalmente, o usuário recebe um cookie de sessão Shibboleth e é redirecionado para o recurso. Os atributos enviados pelo provedor de identidade são disponibilizados para aplicação web pelo *mod_shib*, na forma de variáveis de ambiente do servidor web. Desta forma, o recurso pode usar esses atributos para prover um nível de autorização mais granular, além de possibilitar funcionalidades extras na aplicação, com base nestes atributos.

Figura 7.17
Fase 5.





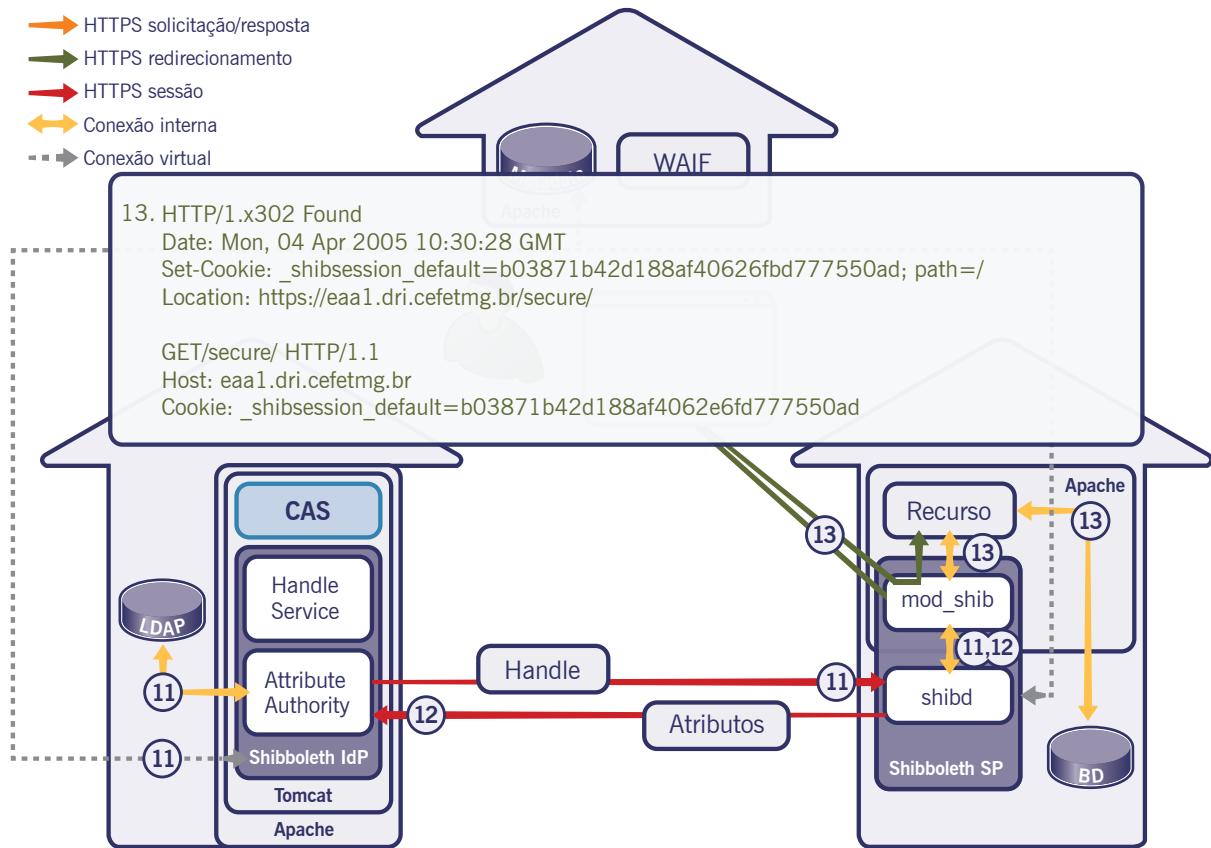


Figura 7.20

Fase 5.

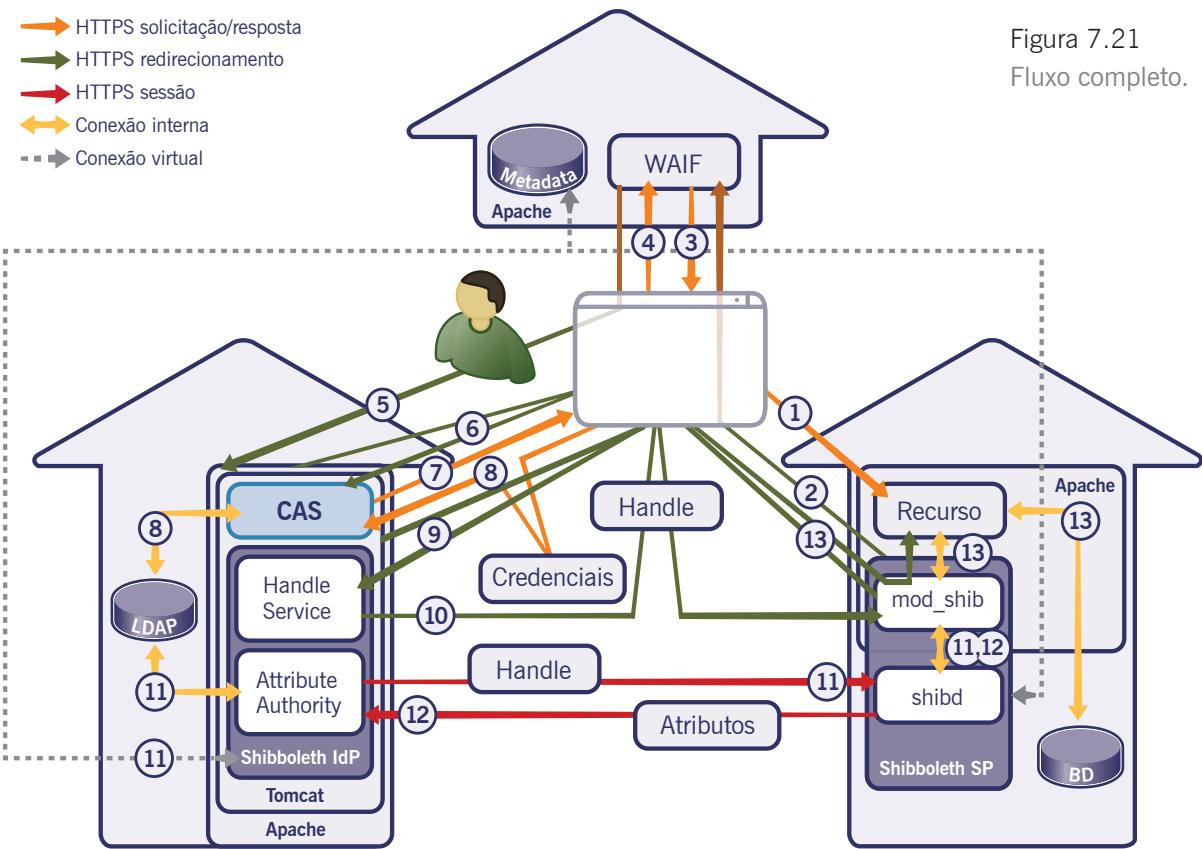


Figura 7.21
Fluxo completo.

7

Roteiro de Atividades Plataforma Shibboleth

Tópicos e conceitos

- Provedor de Identidade (IdP)
- Provedor de Serviço (SP)
- Where Are You From? (WAYF)
- Metadata
- Funcionamento

Competências técnicas desenvolvidas

- Instalação do provedor de identidade
- Configuração manual do provedor de identidade
- Solicitação e instalação de certificado

Tempo previsto para as atividades

- 2 horas

Servidor de sala de aula

- Máquina Virtual que se encontra na área de trabalho. Arquivos em */opt/treinamento*.



Atividade 1 – Instalar e configurar o provedor de identidade Shibboleth

Para instalar o Shibboleth, siga os passos abaixo. Java e Tomcat já estão instalados na máquina virtual (virtual machine – VM).

Instalar Apache:

```
apt-get update  
apt-get install apache2 libapache2-mod-jk
```

Configurar Java, Tomcat e Apache

- As seguintes configurações devem ser feitas para que o Tomcat execute o Shibboleth-IDP:

- Edite */etc/java-6-sun/security/java.security* e adicione as linhas 9 e 10 listadas abaixo:

```
security.provider.7=org.jcp.xml.dsig.internal.dom.XMLDSigRI  
security.provider.8=sun.security.smartcardio.SunPCSC  
..  
security.provider.9=edu.internet2.middleware.shibboleth.  
DelegateToApplicationProvider  
security.provider.10=org.bouncycastle.jce.provider.  
BouncyCastleProvider
```

- Edite */etc/tomcat6/server.xml* para definir que receberá conexões HTTPS na porta 8443, e coloque na mesma seção dos outros conectores (próximo da linha 85 do arquivo original).

Utilize exatamente a configuração abaixo. O arquivo citado em *keystoreFile* e *truststoreFile* será criado adiante.

```
<Connector port="8443"  
          maxHttpHeaderSize="8192"  
          maxSpareThreads="75"  
          scheme="https"  
          secure="true"  
          clientAuth="want"  
          SSLEnabled="true"  
          sslProtocol="TLS"  
          keystoreType="PKCS12"  
          keystoreFile="/opt/shibboleth-idp/credentials/idp.p12"
```

```

        keystorePass="changeit"
        truststoreFile="/opt/shibboleth-idp/credentials/
idp.p12"
        truststorePass="changeit"
        truststoreAlgorithm="DelegateToApplication"/>

```

- Copie o arquivo para auto-deploy do Shibboleth IdP para o Tomcat. O instalador do Shibboleth-IDP deixará o arquivo *idp.war* disponível no caminho descrito em docBase, e o Tomcat o instalará durante a sua inicialização.

```
cp /opt/treinamento/idp/idp.xml    /etc/tomcat6/Catalina/
localhost
```

- Altere o arquivo */opt/treinamento/idp/idp-SSO* (arquivo de virtualhost para o portal de autenticação). O portal de autenticação pode utilizar um certificado SSL diferente do que será fornecido nos metadados da federação. Substitua as variáveis pelo IP da VM.

```
<VirtualHost SUBSTITUIR_IP:443>
    ServerName SUBSTITUIR_IP
    ServerSignature Off
    SSLEngine          on
    SSLCertificateKeyFile /etc/ssl/private/chave-
apache.key
    SSLCertificateFile     /etc/ssl/certs/certificado-
apache.crt
    DocumentRoot /var/www/vazio/
```

```
<Directory /var/www/vazio/>
    Options -Indexes -FollowSymLinks -MultiViews
    AllowOverride None
    Order deny,allow
    Deny from all
</Directory>
```

```
CustomLog /var/log/apache2/access-idp-443.log combined
LogLevel warn
ErrorLog /var/log/apache2/error-idp-443.log
</VirtualHost>
```



5. Após alterar copie o arquivo para o Apache:

```
cp /opt/treinamento/idp/idp-SSO /etc/apache2/sites-available
```

6. Copie o arquivo *idp.conf* com o conteúdo abaixo para configurar a ligação entre o Apache e o Tomcat:

```
cp /opt/treinamento/idp/idp.conf /etc/apache2/conf.d/
```

Conteúdo do arquivo:

```
JkWorkersFile    /etc/libapache2-mod-jk/workers.properties  
JkShmFile       /var/run/apache2/jk-runtime-status  
JkLogFile        /var/log/apache2/mod_jk.log  
JkLogLevel       info  
JkMount          /idp/* ajp13_worker
```

7. Comandos para finalizar a configuração do Apache:

```
mkdir /var/www/vazio/  
a2dissite default  
a2ensite idp-SSO  
a2enmod ssl
```

8. Adicione certificados de servidor à cadeia de certificados confiáveis do Java:

```
cd /etc/ssl/certs/  
wget http://pacotes.ufrgs.br/softwares/certs/ds.chimarrao.  
cafe.rnp.br.crt  
wget http://pacotes.ufrgs.br/softwares/certs/ds.cafe.rnp.br.crt  
wget http://pacotes.ufrgs.br/softwares/certs/ACRaizdaICPEDU.crt  
wget http://pacotes.ufrgs.br/softwares/certs/ACUFSC.crt  
wget http://pacotes.ufrgs.br/softwares/certs/AC_SSL_UFSC.crt  
keytool -importcert -keystore /etc/java-6-sun/security/cacerts \  
-alias ds.chimarrao.cafe.rnp.br -file /etc/ssl/certs/  
ds.chimarrao.cafe.rnp.br.crt \  
-storetype JKS -storepass changeit -noprompt  
keytool -importcert -keystore /etc/java-6-sun/security/  
cacerts \  
-alias ds.cafe.rnp.br -file /etc/ssl/certs/ds.cafe.  
rnp.br.crt \  
-storetype JKS -storepass changeit -noprompt
```

```
keytool -importcert -keystore /etc/java-6-sun/security/
cacerts \
    -alias ACRaizdaICPEDU -file ACRaizdaICPEDU.crt \
    -storetype JKS -storepass changeit -noprompt

keytool -importcert -keystore /etc/java-6-sun/security/
cacerts \
    -alias ACUFSC -file ACUFSC.crt \
    -storetype JKS -storepass changeit -noprompt

keytool -importcert -keystore /etc/java-6-sun/security/
cacerts \
    -alias AC_SSL_UFSC -file AC_SSL_UFSC.crt \
    -storetype JKS -storepass changeit -noprompt
```

Baixar e instalar o Shibboleth-IDP e bibliotecas Java

O Shibboleth-IDP está disponível no site da Internet2.

Substitua o IP da sua VM no lugar de SUBSTITUIR_IP, copie a sequência de comandos abaixo e cole no terminal:

```
export JAVA_HOME="/usr/lib/jvm/java-6-sun"
cd /root/
wget http://pacotes.ufrgs.br/softwares/shibboleth-
identityprovider-2.1.5-bin.zip
wget http://pacotes.ufrgs.br/softwares/bcprov-jdk16-144.jar
cp bcprov-jdk16-144.jar /usr/lib/jvm/java-6-sun/jre/lib/
unzip shibboleth-identityprovider-2.1.5-bin.zip
cd shibboleth-identityprovider-2.1.5/
cp -r endorsed /usr/share/tomcat6/
cp lib/shibboleth-jce-1.1.0.jar /usr/lib/jvm/java-6-sun/jre/lib/
ext/
cat > src/installer/resources/install.properties -<<EOF
idp.home=/opt/shibboleth-idp
idp.home.input=/opt/shibboleth-idp
idp.hostname=SUBSTITUIR_IP
idp.hostname.input=SUBSTITUIR_IP
idp.keystore.pass=changeit
```



```
EOF
./install.sh
chown tomcat6:tomcat6 /opt/shibboleth-idp/logs/
chown tomcat6:tomcat6 /opt/shibboleth-idp/metadata/
```

Configuração do Shibboleth IdP

1. Edite o arquivo */opt/shibboleth-idp/conf/handler.xml*, comente a seção referente ao *RemoteUser* e habilite a seção do *UsernamePassword*. O arquivo final deve ter a seguinte configuração:

```
<!-- Login Handlers -->
<!--
<LoginHandler xsi:type="RemoteUser">

<AuthenticationMethod>urn:oasis:names:tc:SAML:2.0:ac:classes:unspecified
</AuthenticationMethod>

</LoginHandler>
-->

<!-- Username/password login handler -->
<LoginHandler xsi:type="UsernamePassword">
    jaasConfigurationLocation="file:///opt/shibboleth-idp/conf/login.config">
        <AuthenticationMethod>urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport</AuthenticationMethod>
    </LoginHandler>
```

2. Edite */opt/shibboleth-idp/conf/relying-party.xml* para substituir o bloco correspondente à tag *MetadataProvider id="URLMD"* pelo bloco abaixo:

```
<MetadataProvider id="URLMD" xsi:type="FileBackedHTTPMetadataProvider"
    xmlns="urn:mace:shibboleth:2.0:metadata"
    metadataURL="http://idp.shib.curso/Shibboleth.sso/Metadata"
    backingFile="/opt/shibboleth-idp/metadata/sp-metadata.xml">
    <MetadataFilter xsi:type="ChainingFilter"
        xmlns="urn:mace:shibboleth:2.0:metadata">
        <MetadataFilter xsi:type="EntityRoleWhiteList"
            xmlns="urn:mace:shibboleth:2.0:metadata">
            <RetainedRole>samlmd:SPSSODescriptor</RetainedRole>
            </MetadataFilter>
        </MetadataFilter>
    </MetadataFilter>
</MetadataProvider>
```



- Copie o arquivo de configuração *attribute-filter.xml* e o metadata:

```
cp /opt/treinamento/idp/attribute-filter.xml /opt/shibboleth-idp/conf
cd /opt/shibboleth-idp/metadata/
wget http://idp.shib.curso/Shibboleth.sso/Metadata
cp Metadata sp-metadata.xml
chmod 777 sp-metadata.xml
```

- Copie o seguinte arquivo com o comando:

```
cp /opt/treinamento/idp/attribute-resolver.xml /opt/shibboleth-idp/conf
```

Em seguida, edite o arquivo */opt/shibboleth-idp/conf/attribute-resolver.xml* substituindo:

SUBSTITUIR_IP – pelo IP da sua VM.

SUBSTITUIR_INSTITUICAO – pela sigla da sua instituição usada na instalação do LDAP.

- Configuração da autenticação LDAP:

Substitua os seguintes valores pelos dados de seu servidor no arquivo abaixo.

Edite o arquivo em */opt/shibboleth-idp/conf/login.config*:

```
ShibUserPassAuth {
    edu.vt.middleware.ldap.jaas.LdapLoginModule required
        host="SUBSTITUIR_SERVIDOR_LDAP:389"
        base="SUBSTITUIR_BASE_DN"
        ssl="false"
        userField="uid"
        serviceUser="SUBSTITUIR_USUARIO_LEITOR_SHIB"
        serviceCredential="SUBSTITUIR_SENHA_LEITOR_SHIB"
        subtreeSearch="false";
};
```

SUBSTITUIR_SERVIDOR_LDAP – endereço IP do host que contém o LDAP.

SUBSTITUIR_BASE_DN – ramo da árvore que contém os usuários:
“ou=people,dc=SUBSTITUIR_INSTITUICAO,dc=br”



SUBSTITUIR_USUARIO_LEITOR_SHIB – usuário com direito de leitura na base LDAP: “cn=leitor-shib, dc=SUSTITUIR_INSTITUICAO,dc=br”

SUBSTITUIR_SENHA_LEITOR_SHIB – senha do usuário de leitura: 00123456

Certificados SSL

6. Antes de gerar as chaves criptográficas e os certificados SSL é preciso preparar a configuração do OpenSSL. O comando do bloco abaixo cria esse arquivo. Substitua o IP na primeira linha. Depois copie e cole todo o bloco no terminal.

HOSTNAME_FQDN=**SUBSTITUIR_IP**

```
cat >/opt/shibboleth-idp/credentials/openssl.cnf <<-EOF
[ req ]
default_bits = 2048 # Size of keys
string_mask = nombstr # permitted characters
distinguished_name = req_distinguished_name
x509_extensions = v3_ca
[ req_distinguished_name ]
countryName = Nome do país (código de 2 letras)
countryName_min = 2
countryName_max = 2
stateOrProvinceName = Unidade da Federacao (por extenso)
localityName = Nome do municipio (por extenso)
0.organizationName = Nome da universidade/instituicao
organizationalUnitName = Departamento da universidade/
instituicao
emailAddress = Endereco de email da administracao
emailAddress_max = 40
commonName = Nome completo do host (incluindo o dominio)
commonName_max = 64
commonName_default = $HOSTNAME_FQDN
# Default values for the above, for consistency and less typing.
# Variable name    Value
#-----#
#0.organizationName_default =
```



```

# organizationalUnitName_default = CPD
#localityName_default = Porto Alegre
#stateOrProvinceName_default = Rio Grande do Sul
countryName_default = BR
[ usr_cert ]
basicConstraints= CA:FALSE
extendedKeyUsage      = serverAuth, nsSGC, msSGC
[ ssl_server ]
basicConstraints= CA:FALSE
keyUsage = digitalSignature, keyEncipherment
nsCertType = server
nsComment           = “OpenSSL Certificate for SSL Web
Server”
[ v3_req ]
basicConstraints= CA:FALSE
keyUsage = digitalSignature, keyEncipherment
extendedKeyUsage     = serverAuth, nsSGC, msSGC
[ v3_ca ]
basicConstraints= CA:FALSE
keyUsage = digitalSignature, keyEncipherment
extendedKeyUsage     = serverAuth, nsSGC, msSGC
EOF

```

7. Certificado para Shibboleth-IDP:

Copie e cole na janela de terminal os comandos seguintes (um a um) de acordo com as instruções abaixo:

- ▲ No quarto comando, informe os seguintes dados:
 - ▲ Confirme o código do país (BR)
 - ▲ Unidade da federação (seu estado)
 - ▲ Cidade
 - ▲ Instituição (preferencialmente preencha com a sigla)
 - ▲ Departamento da instituição
 - ▲ Confirme se o hostname está correto (IP do host que será o IDP)



- ▲ No quinto comando, informe a senha “changeit”. A senha está cadastrada no arquivo `/etc/tomcat6/server.xml` e o tomcat6 precisará dela para abrir o keystore que está sendo gerado.

```
cd /opt/shibboleth-idp/credentials/  
rm -f idp*  
  
openssl genrsa 2048 -config openssl.cnf > idp.key  
  
openssl req -new -x509 -nodes -days 1095 -sha1 -key idp.key  
-set_serial 00 -config openssl.cnf > idp.crt  
  
openssl pkcs12 -export -in idp.crt -inkey idp.key -out idp.  
p12 -name idp -caname selfsigned
```

8. Certificado para Apache:

Esse certificado será exibido para o usuário/browser quando o portal de autenticação for acessado. Ainda a partir do mesmo diretório dos comandos acima, execute os seguintes comandos (um a um):

```
openssl genrsa 2048 -config openssl.cnf > /etc/ssl/private/  
chave-apache.key  
  
openssl req -new -x509 -nodes -days 1095 -sha1 -key /etc/ssl/  
private/chave-apache.key -set_serial 00 -config openssl.cnf >  
/etc/ssl/certs/certificado-apache.crt  
  
chown root /etc/ssl/private/chave-apache.key /etc/ssl/certs/  
certificado-apache.crt  
  
chmod 640 /etc/ssl/private/chave-apache.key
```

9. Corrigir o arquivo de metadados local:

O arquivo de metadados do servidor Shibboleth-IDP local `/opt/shibboleth-idp/metadata/idp-metadata.xml` foi gerado com um certificado SSL auto-gerado na instalação do Shibboleth-IDP. Esse certificado precisa ser substituído pelo que foi gerado no passo anterior da instalação.

- ▲ Liste o conteúdo do arquivo do certificado que foi auto-assinado.

```
cat /opt/shibboleth-idp/credentials/idp.crt
```

- ▲ Copie o conteúdo entre as linhas BEGIN CERTIFICATE e END CERTIFICATE.

- Edite o arquivo de metadados */opt/shibboleth-idp/metadata/idp-metadata.xml* e exclua o certificado incorreto. Há duas ocorrências desse certificado no arquivo, ambas dentro das seguintes tags *xml*:

```
<KeyDescriptor>
  <ds:KeyInfo>
    <ds:X509Data>
      <ds:X509Certificate>
        INSIRA_AQUI_O_CONTEUDO_DO_ARQUIVO_DO_CERTIFICADO
      </ds:X509Certificate>
    </ds:X509Data>
  </ds:KeyInfo>
</KeyDescriptor>
```

 Atenção ao substituir o certificado, pois pode ocorrer de faltar alguma parte dele.
Após copiar e colar, confira se todo o conteúdo foi colado.

10. Envie para o instrutor o arquivo de Metadata que está localizado em:

opt/shibboleth-idp/metadata/idp-metadata.xml

Substitua o final do seu IP no comando abaixo. Quando solicitada informe a senha *sysadmin*:

```
scp /opt/shibboleth-idp/metadata/idp-metadata.xml sysadmin@  
idp.shib.curso:/home/sysadmin/SUBSTITUIR_FINAL_IP.xml
```

11. Reinicie o Tomcat e o Apache para que as configurações sejam recarregadas:

```
/etc/init.d/apache2 restart  
/etc/init.d/tomcat6 restart
```

12. Acesse o seguinte endereço para verificar se o IdP está no ar: http://SUBSTITUIR_IP_VM/idp/profile>Status.

13. Verifique se uma página em branco com a palavra OK é exibida.

8

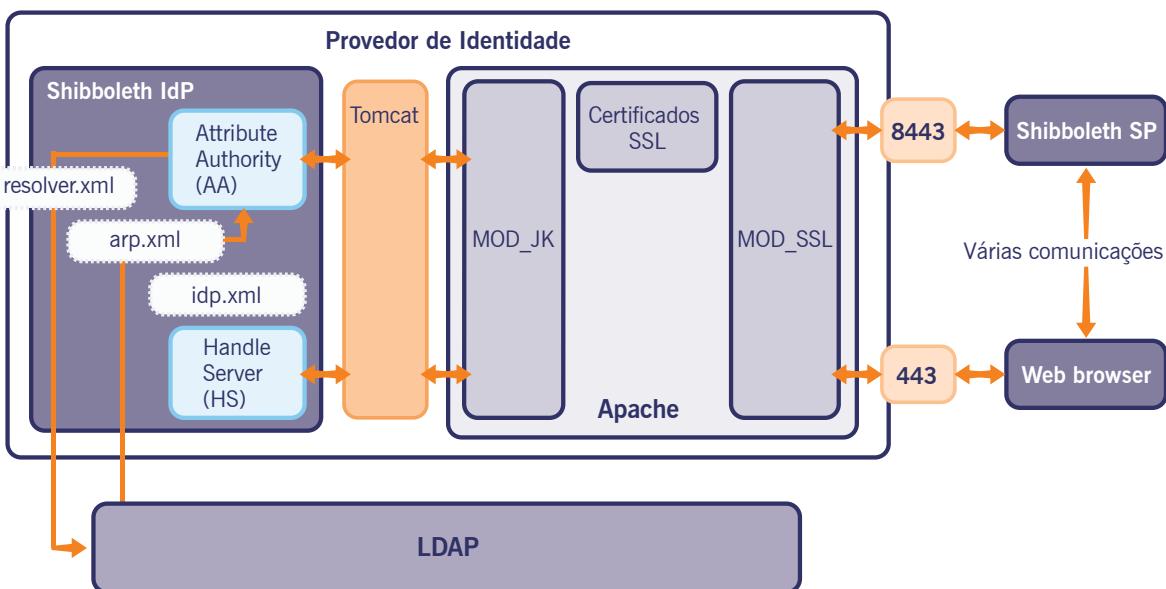
Provedor de identidade na plataforma Shibboleth

Sumário

- Principais pontos de configuração:
 - ▀ Apache
 - ▀ Tomcat
 - ▀ CAS
 - ▀ Shibboleth IdP

Figura 8.1
Configuração
de provedor
de identidade.

Principais pontos de configuração



Nesta sessão apresentaremos os principais pontos de configuração de um provedor de identidade na plataforma Shibboleth.

Configuração do Apache

- ▶ Virtual Hosts
 - ▶ AA – Attribute Authority
 - ▶ SSO – CAS e Handle Server
- ▶ mod_ssl
 - ▶ Certificado, Chave e Autoridade Certificadora
 - ▶ Autenticação mútua entre o AA e o shibd
 - ▶ Exigência do certificado do shibd e repasse para o Shibboleth IdP
- ▶ mod_jk
 - ▶ Redirecionamento para o Tomcat

Na configuração do Apache são criados dois Virtual Hosts:

- ▶ AA, porta 8443 – Responsável pela comunicação entre o Attribute Authority e o Provedor de Serviço.
- ▶ SSO, porta 443 – Responsável pela comunicação entre o browser do usuário e o Handle Server e o CAS.

Além disso, são habilitados os seguintes módulos:

- ▶ mod_ssl – Responsável por criptografar a comunicação com o provedor de identidade.
- ▶ mod_jk – Responsável por redirecionar as requisições para o Tomcat.

Configuração do Tomcat

- ▶ Conector AJP 1.3
 - ▶ Redirecionamento do Apache
 - ▶ Desabilitar os demais conectores

Na configuração do Tomcat é necessário apenas habilitar o conector AJP 1.3, responsável pelo redirecionamento das requisições do Apache.

Configuração do CAS

- ▶ CAS Server
 - ▶ CAS Generic Handler
 - ▶ Acesso ao diretório LDAP para a autenticação do usuário
 - ▶ Exibição das páginas de login
- ▶ CAS Client
 - ▶ Java CAS Client
 - ▶ Redirecionamento do Shibboleth IdP para o CAS Server



A configuração do CAS é dividida em duas etapas:

- CAS Generic Handler – Responsável pela exibição da página de login e validação da autenticação na base de usuários. Essa configuração é realizada no momento da sua instalação no Tomcat. É necessário indicar o usuário responsável pelas consultas ao diretório e o endereço de redirecionamento para o Shibboleth IdP. Além disso, é possível personalizar a exibição das páginas de login.
- Java CAS Client – Responsável pelo redirecionamento das requisições do Shibboleth IdP para o CAS Generic Handler. Essa configuração é realizada dentro da instalação do Shibboleth IdP. É necessário indicar somente o endereço de redirecionamento para o CAS Generic Handler.

Configuração do Shibboleth IdP

- Identificação e configuração básica do provedor:
 - ▲ /opt/shibboleth-idp/conf/relying-party.xml
- Resolução dos atributos:
 - ▲ /opt/shibboleth-idp/conf/attribute-resolver.xml
- Liberação dos atributos:
 - ▲ /opt/shibboleth-idp/conf/attribute-filter.xml
- Provedores de serviço autorizados:
 - ▲ /opt/shibboleth-idp/metadata/<federação>-metadata.xml

O Shibboleth IdP possui quatro principais arquivos para configuração do serviço. Através destes arquivos é possível especificar detalhadamente como o provedor de identidade irá atuar na federação. A seguir o detalhamento de cada um deles.

- relying-party.xml
 - ▲ Identificador do provedor na federação
 - ▲ Credenciais: certificado e chave
 - ▲ URL do Attribute Authority
 - ▲ Tratadores de protocolos (Protocol Handlers)
 - ▲ Nível de detalhamento dos logs
 - ▲ Caminho dos outros arquivos de configuração

O arquivo *relying-party.xml* é responsável pela identificação e configuração básica do provedor de identidade. Nele são indicadas as credenciais do provedor, os tratadores de protocolos e a URL do Attribute Authority, entre outras informações necessárias para que o provedor possa se identificar e se comunicar corretamente dentro da federação. Além disso, são indicados o nível de detalhamento dos logs e os caminhos para os demais arquivos de configuração.



- ▶ attribute-resolver.xml
 - ◀ Conector para a base de atributos de usuários (LDAP ou SQL)
- ▶ DataConnectors
 - ◀ Credenciais do usuário leitor da base de atributos
- ▶ AttributeDefinition
 - ◀ Definições de atributos

O arquivo *attribute-resolver.xml* é responsável pela definição das regras de resolução de atributos. Nele são configurados os parâmetros de acesso à base de dados ou ao diretório, e o mapeamento dos atributos. O mapeamento pode ser feito diretamente, através da simples declaração do atributo, ou pode ser definido pelo usuário através de scripts personalizados.

- ▶ attribute-filter.xml
 - ◀ Regras de liberação de atributos
 - ◀ Liberação de atributos por SP
 - ◀ Liberação de atributos por Federação
 - ◀ Liberação de todos atributos sem restrição

A Política de Liberação de Atributos é configurada através do arquivo *attribute-filter.xml*, onde é possível filtrar a liberação de atributos de acordo com quem os requisita.

Pode-se, por exemplo, liberar todos os atributos caso o requisitante seja um SP membro da Federação CAFE, ou ainda liberar ou não determinado atributo para determinado SP.

- ▶ metadata.xml
 - ◀ Disponibilizado pela federação
 - ◀ Informações relevantes sobre os provedores
 - ◀ Confiança
 - ◀ Certificados
 - ◀ Chaves públicas
 - ◀ Comunicação
 - ◀ IDs
 - ◀ URLs
 - ◀ Protocolos

O arquivo de metadata é disponibilizado pela federação e a sua função é estabelecer a relação de confiança entre os provedores. Nesse arquivo são indicadas, para o provedor de identidade, as informações pertinentes sobre os provedores de serviço (e vice-versa). Desta forma, garante-se a segurança e a autenticidade na comunicação entre os provedores.



8

Roteiro de Atividades Provedor de identidade na plataforma Shibboleth

Tópicos e conceitos

- Principais pontos de configuração de um provedor de identidade na plataforma Shibboleth

Competências técnicas desenvolvidas

- Principais pontos de configuração de um provedor de identidade na plataforma Shibboleth.
- Testes no ambiente configurado.

Tempo previsto para as atividades

- 40 minutos

Servidor de sala de aula

- Máquina virtual com Shibboleth IdP já instalado pelo aluno.

Atividade 1 – Validando a instalação e testando a Federação

1. Valide uma instalação através do aacli.sh, script que simula a requisição de atributos do IdP por um SP. Siga os passos abaixo para testar o IdP que acabou de instalar:

```
cd /opt/shibboleth-idp/bin  
./aacli.sh --configDir=/opt/shibboleth-idp/conf  
--principal=00123456
```

Depois de alguns segundos será retornado um trecho de mensagem SAML com os atributos requeridos, que deve ser semelhante ao seguinte:

```
<?xml version="1.0" encoding="UTF-  
8"?><saml2:AttributeStatement xmlns:saml2="urn:oasis:names:tc  
:SAML:2.0:assertion">  
  
    <saml2:Attribute FriendlyName="cn" Name="urn:oid:2.5.4.3"  
    NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">  
  
        <saml2:AttributeValue xmlns:xs="http://www.w3.org/2001/  
        XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-  
        instance" xsi:type="xs:string">Joao</saml2:AttributeValue>  
  
    </saml2:Attribute>  
  
    <saml2:Attribute FriendlyName="mail" Name="urn:  
    oid:0.9.2342.19200300.100.1.3" NameFormat="urn:oasis:names:tc  
    :SAML:2.0:attrname-format:uri">  
  
        <saml2:AttributeValue xmlns:xs="http://www.w3.org/2001/  
        XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-  
        instance" xsi:type="xs:string">00123456@ufmg.br</  
        saml2:AttributeValue>  
  
    </saml2:Attribute>  
  
    <saml2:Attribute FriendlyName="sn" Name="urn:oid:2.5.4.4"  
    NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">  
  
        <saml2:AttributeValue xmlns:xs="http://www.w3.org/2001/  
        XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-  
        instance" xsi:type="xs:string">Silva</saml2:AttributeValue>  
  
    </saml2:Attribute>  
  
</saml2:AttributeStatement>
```

2. Simulando uma federação:

- 2.1. Acesse via browser o seguinte serviço protegido pelo Shibboleth SP no servidor do instrutor: <http://idp.shib.curso/homologa/atributos>. Os certificados serão exibidos, e você será redirecionado para o WAYF/DS, onde deverá escolher o seu Provedor de Identidade (IdP) para se autenticar.
 - 2.2. Após escolher seu próprio IdP, você será redirecionado para se autenticar no IdP instalado na sua máquina. Informe o UID e a senha do usuário que inseriu no LDAP na segunda sessão do curso. Caso não se lembre, acesse o Apache Directory Studio e altere a senha de algum usuário.
 - 2.3. Após ser autenticado você irá visualizar os atributos fornecidos pelo seu IdP para a aplicação *Homologa*.
3. Verifique nos logs do sistema as asserções SAML trocadas entre IdP e SP.

Localização dos arquivos de logs do IdP: */opt/shibboleth-idp/logs*

Arquivos para configuração de níveis de log: */opt/shibboleth-idp/conf/logging.xml*



9

Implantação de provedor de identidade a partir de bases de dados relacionais

- Roteiro de implantação de um provedor de identidade
 - Metodologia adotada
 - Roteiro de atividade

Roteiro de implantação de um provedor de identidade

Metodologia adotada

- Dividir a tarefa de implantação do provedor de identidade em atividades ou etapas distintas
- Construir scripts para auxiliar na execução das atividades definidas
- Elaborar testes intermediários ao final de cada atividade/etapa

Nesta sessão do curso apresentaremos o roteiro completo de implantação de um Provedor de Identidade (IdP), reproduzindo o roteiro disponível na web e destinado às instituições que não possuem serviço de diretório em operação, isto é, as fontes de informação sobre os membros da instituição são mantidas em bases de dados relacionais. Discute-se inicialmente a metodologia adotada para a definição do roteiro, e, em seguida, cada uma das atividades ou etapas do roteiro é detalhada.

A implantação de um provedor de identidade interinstitucional constitui-se de uma sequência de etapas que precisam ser cumpridas por qualquer instituição. Entretanto, existem detalhes internos de configuração em cada etapa que podem variar de uma instituição para outra. Assim, a metodologia adotada para auxiliar na implantação dos IdPs foi a elaboração de um roteiro de atividades, com a inclusão de passos intermediários de verificação, e a implementação de ferramentas de auxílio para facilitar a execução e o acompanhamento das etapas de implantação dos provedores de identidade.



Roteiro de atividades

- ▶ Instalar o servidor básico padrão
- ▶ Instalar o diretório com o esquema brEduPerson
- ▶ Extrair dados para o metadiretório
- ▶ Alimentar o diretório a partir do metadiretório
- ▶ Instalar o provedor de identidade
- ▶ Entrar na Federação CAFé

O roteiro de atividades proposto parte da instalação e configuração do sistema operacional da máquina que será dedicada ao IdP. Ele inclui a instalação de um servidor de diretório onde serão armazenadas as informações dos membros da instituição acessadas pelo IdP. O roteiro segue com a etapa de extração das informações das bases de dados relacionais da instituição e o armazenamento dessas informações no servidor de diretório. Por fim, o software do IdP é instalado e suas informações de configuração são remetidas para o gerente da federação.

Instalar o servidor básico padrão

- ▶ Instalar o sistema operacional Ubuntu
- ▶ Configurar o ambiente
- ▶ Instalar o diretório com o esquema brEduPerson
 - ▶ Instalar o OpenLDAP com o esquema brEduPerson incluído
 - ▶ Executar teste padrão de escrita e leitura no diretório

O roteiro proposto recomenda a utilização da distribuição Ubuntu para instalar os sistemas operacionais nas máquinas nas quais os provedores de identidade serão instalados. Essa distribuição foi escolhida por disponibilizar de forma nativa os pacotes Java 6, requisito de software necessário para executar as ferramentas de extração de dados e a carga do servidor de diretório (EID e EID2LDAP), e ainda o próprio software que implementa o provedor de identidade (Shibboleth-IdP). Além disso, o Ubuntu 10.04 LTS (Lucid) terá suporte para atualizações de segurança até abril de 2015.

Utilize o Ubuntu Server versão 10.04, disponível em:

- ▶ PC (Intel x86): <http://br.archive.ubuntu.com/releases/lucid/ubuntu-10.04.2-server-i386.iso>
- ▶ 64-bit PC (AMD64): <http://br.archive.ubuntu.com/releases/lucid/ubuntu-10.04.2-server-amd64.iso>

Instale o Ubuntu Server executando as configurações sugeridas no roteiro disponível no site do projeto: <http://wiki.rnp.br/pages/viewpage.action?pagId=41616299>



Após a instalação do sistema operacional e a configuração básica da máquina, o passo seguinte é a instalação do servidor de diretório LDAP. As seguintes instalações e configurações são efetuadas nessa etapa:

- Instalação e configuração do slapd (servidor LDAP);
- Criação de um usuário de teste na base LDAP (uid=00123456,ou=people,dc=dominio,dc=br);
- Liberação das portas 389 e 636 no firewall;
- Criação de chaves SSL para o LDAP (armazenadas em /etc/ldap/).

Extrair dados para o metadiretório

- Instalar o EID e o EID2LDAP
- Configurar as extrações
- Executar teste padrão de acesso ao metadiretório
- Alimentar o diretório a partir do metadiretório
 - Configurar a exportação
 - Executar o teste padrão de leitura no diretório

Após a instalação do servidor de diretório, o próximo passo é a instalação das ferramentas EID e EID2LDAP. Essas ferramentas são utilizadas para auxiliar na extração das informações sobre os integrantes das bases de dados relacionais e na inclusão das mesmas no servidor de diretório. Tais ferramentas requerem a instalação dos seguintes softwares: JDK 1.6, Tomcat 6.0.13, MySQL 5.0.51x e phpMyAdmin 2.11.3.

Os requisitos de hardware mínimos são: 1 Gb de memória RAM (recomendado 2 Gb), 240 Mb livres para as aplicações (Tomcat, MySQL, EID e EID2LDAP), 50 Mb livres para as bases de dados (pode variar em função do tamanho das bases de dados relacionais de origem).

Para utilizar a ferramenta EID é necessário ter acesso às bases de dados da instituição de onde serão extraídas as informações sobre as pessoas. Essa ferramenta cria uma base de dados MySQL intermediária denominada metadiretório. Para configurar as extrações e criar o metadiretório, os seguintes passos são necessários:

- Identificar as bases dos dados que serão utilizados para alimentar o sistema (base de alunos de graduação, base de alunos de pós-graduação, base de recursos humanos etc.);
- Verificar as classes e atributos necessários para o esquema brEduPerson (classes EID recomendadas para brEduPerson já vêm configuradas na instalação via roteiro);
- Cadastrar as fontes de origem (normalmente requer acesso autorizado).



A utilização da ferramenta EID cria o metadiretório (base de dados MySQL intermediária) e carrega para o metadiretório as informações sobre as pessoas vinculadas à instituição que devem ser adicionadas ao diretório LDAP, que será acessado pelo IdP. O passo seguinte consiste em utilizar a ferramenta EID2LDAP para transferir os dados do metadiretório para o diretório LDAP.

O roteiro de instalação disponibilizado define:

- ▲ O endereço do web service do EID em [http://localhost:8080/eid/services/
EidService?wsdl](http://localhost:8080/eid/services/EidService?wsdl)
- ▲ O servidor LDAP em localhost;
- ▲ A transformação para LDIF no padrão brEduPerson;
- ▲ O metadiretório do EID em localhost;
- ▲ As classes EID recomendadas para o brEduPerson.

Instalar o provedor de identidade

- ▲ Instalar o Shibboleth IdP
- ▲ Enviar metadados para a Federação Chimarrão
- ▲ Executar a aplicação de teste

Uma vez carregado o diretório que será acessado pelo provedor de identidade, o passo seguinte é a instalação do próprio provedor de identidade. O roteiro recomenda a instalação do software Shibboleth IdP (versão 2.x), o qual requer os seguintes softwares adicionais: Tomcat, Apache2 e OpenSSL.

O tutorial disponibilizado requer o endereço do servidor LDAP que será acessado pelo IdP e as seguintes configurações:

- ▲ Cadastramento da senha do usuário leitor-shib na base LDAP;
- ▲ Configuração do Apache para utilizar o módulo *mod_jk*;
- ▲ Criação de certificados SSL para o Apache e o Shibboleth-IdP.

Após a instalação do IdP, o próximo passo do roteiro é a integração do IdP instalado com uma federação de teste (Federação Chimarrão). Para isso, é necessário enviar para o gerente dessa federação os metadados gerados na execução das etapas anteriores. Os metadados servem para informar aos demais participantes da federação quais são os servidores reconhecidos e confiáveis.

O roteiro de instalação do Shibboleth-IdP faz a geração da chave criptográfica e de um certificado SSL auto-assinado para o servidor. A chave pública deste certificado é parte integrante dos metadados do servidor.



Entrar na Federação CAFe

- Solicitar ou gerar certificado
- Instalar o certificado na máquina
- Migrar as configurações
- Enviar metadados para a Federação CAFe
- Executar aplicação de teste

A entrada na federação de CAFe é a última etapa da implantação do provedor de identidade interinstitucional. Uma vez que todas as etapas anteriores foram cumpridas com sucesso, é necessário dispor de um certificado assinado por autoridade certificadora reconhecida pelos demais membros da federação.

De posse do certificado, basta migrar as configurações necessárias e enviar a nova versão dos metadados para o gerente da Federação CAFe.



9

Roteiro de Atividades Implantação de provedor de identidade a partir de bases de dados relacionais

Tópicos e conceitos

- Roteiro para implantação de um provedor de identidade a partir de bases de dados relacionais.

Competências técnicas desenvolvidas

- Executar o roteiro completo de instalação de um provedor de identidade usando os scripts e ferramentas do projeto e-AA.

Tempo previsto para as atividades

- 30 – 60 minutos

Tempo previsto para as atividades

- Provedor de serviço com aplicação homologada na máquina do instrutor.
- Provedores de Identidade instalados nas máquinas dos alunos.

Atividade 1 – Demonstrar o funcionamento da autenticação e envio de atributos

1. Iniciar visualização dos arquivos de log dos provedores de identidade e de serviço.

Localização dos arquivos de log do IdP: /opt/shibboleth-idp/logs

Localização dos arquivos de log do SP: /opt/shibboleth-sp-2.2/var/log

- 1.1. Abrir terminais (SSH) no Provedor de Serviço:

SSH: sysadmin@idp.shib.curso

Senha: sysadmin

- 1.2. Visualizar (tail -f) os arquivos de log indicados pelo instrutor.

2. Acessar recurso web:

- 2.1. Abrir o browser e acessar a URL indicada pelo instrutor.

- 2.2. Verificar o redirecionamento para o WAYF no arquivo de log do Shibboleth SP.

3. Selecionar provedor de identidade no WAYF.

4. Autenticar-se com uma conta inválida:

- 4.1. Enviar credenciais inválidas (login e senha) para o provedor de identidade.

5. Autenticar-se com uma conta válida: enviar credenciais válidas (login e senha) para o provedor de identidade.

- 5.1. Verificar o redirecionamento e envio do *handle* para o provedor de serviço no arquivo de log do Shibboleth IdP.

6. Visualizar o recurso web:

- 6.1. Verificar recebimento do *handle* no arquivo de log do Shibboleth SP.

- 6.2. Verificar solicitação de atributos no arquivo de log do Shibboleth SP.

- 6.3. Verificar envio de atributos no arquivo de log do Shibboleth IdP.

- 6.4. Verificar recebimento de atributos no arquivo de log do Shibboleth SP.

10

Implantação de provedor de identidade a partir de um diretório existente

- ▶ Introdução ao Shibboleth-IdP
 - ▶ Shibboleth-IdP
 - ▶ Origem dos dados
- ▶ Análise do cenário
- ▶ Atributos requisitados pela federação
 - ▶ Atributos do esquema original
- ▶ Definição dos mapeamentos
 - ▶ Renomear atributos
 - ▶ Alterar valor de atributo
 - ▶ Modificar sequência de atributos

Introdução ao Shibboleth-IdP

- ▶ Shibboleth-IdP (Identity Provider)
 - ▶ Disponibiliza os atributos de uma base de dados local para a federação
 - ▶ Efetua o mapeamento dos atributos e controle de acesso

Esta décima sessão do curso irá apresentar as configurações necessárias para que uma instituição que está utilizando uma base LDAP com schema já definido possa compatibilizar-se com os requisitos da federação CAFé. Serão apresentados os procedimentos necessários para criar o arquivo de configuração que define o mapeamento realizado pelo Shibboleth-IDP entre o schema LDAP da instituição e os atributos que devem ser compartilhados com os provedores de serviço da federação.



O Shibboleth-IdP (Identity Provider) faz a busca dos atributos dos usuários na base de dados e apresenta-os de forma organizada e padronizada para os provedores de serviço federados. Para executar essa operação ele acessa a base de dados da instituição utilizando os chamados “Conectores”, realizando uma busca dos atributos de determinado usuário e mapeando-os para que sejam visualizados na federação.

Origem dos dados

- Base LDAP
- Banco de dados relacional
- Arquivos de texto
- Conector personalizado (utilizando Java)

Os “conectores” podem fazer essa busca em diversas origens, entre elas bases de dados relacionais, diretórios LDAP, arquivos texto e ainda conectores personalizados, que podem ser definidos caso a caso.

Para efetuar o mapeamento entre uma base LDAP existente e que não utiliza o schema brEduPerson utilizaremos operações referentes ao conector LDAP e também operações personalizadas.

Análise do cenário

- O objetivo do mapeamento é compatibilizar os requisitos da federação com os atributos existentes na base LDAP
- Definido no arquivo *attribute-resolver.xml*
- Operações básicas:
 - Renomear atributo
 - Modificar valor de atributo
 - Modificar valor de sequência de atributos

A configuração dos conectores é definida no arquivo:

“/opt/shibboleth-idp/conf/attribute-resolver.xml”

```
<resolver:DataConnector id="myLDAP" xsi:type="LDAPDirectory"
xmlns="urn:mace:shibboleth:2.0:resolver:dc"

    ldapURL="ldap://" baseDN="ou=people,dc=ufmg,dc=br" principal="cn=admin,dc=ufmg,dc=br"

...
</resolver:DataConnector>
```



As operações básicas são as seguintes:

- ▲ Renomear atributo;
- ▲ Modificar valor de atributo;
- ▲ Modificar valor de sequência de atributos.

Atributos recomendados pela federação

- ▲ Conjunto mínimo de atributos recomendados pela federação CAFé para identificar um usuário:
 - ▲ eduPersonPrincipalName
 - ▲ mail
 - ▲ cn
 - ▲ sn
 - ▲ brEduAffiliation
 - ▲ brEntranceDate
 - ▲ brExitDate

Os atributos recomendados pela federação CAFé são os seguintes:

- ▲ eduPersonPrincipalName – Identificação única do usuário no escopo da instituição (código de usuário@domínio_da_instituição).
- ▲ mail – Endereço de e-mail.
- ▲ cn – Primeiro nome.
- ▲ sn – Restante do nome.
- ▲ brEduAffiliation – Número de ordem do vínculo do usuário. Se houver mais de um vínculo eles devem ser fornecidos em ordem crescente. Este atributo serve como índice para os demais atributos de informações sobre vínculos.
- ▲ brEduAffiliationType – Tipo do vínculo do usuário. Se houver mais de um vínculo eles devem ser fornecidos na ordem definida em brEduAffiliation.
- ▲ brEntranceDate – Data de entrada para cada vínculo. Se houver mais de um vínculo as datas de entrada devem ser fornecidas na ordem definida em brEduAffiliation.
- ▲ brExitDate – Data de saída para cada vínculo. Se houver mais de um vínculo as datas de entrada devem ser fornecidas na ordem definida em brEduAffiliation. A inexistência dessa informação indica que o usuário ainda tem vínculo ativo.



Os atributos listados acima permitem que cada usuário seja identificado corretamente junto à sua instituição e que sejam verificados os vínculos ativos ou inativos que ele possui. Diferentes provedores de serviço poderão requisitar mais ou menos informações sobre um usuário para permitir o uso de seu serviço.

Atributos do esquema original

► Atributos presentes no esquema UFRGS

► Esses atributos serão mapeados para os atributos recomendados pela federação CAFé

uid	: "00123456"
cn	: "JOAO DA SILVA"
sn	: "JOAO DA SILVA"
ufrgsTipoVinculo	: "01: aluno-graduacao"
ufrgsDataAfastamento	: "01: "
ufrgsDataIngresso	: "01: 01/03/2002"

Os atributos já existentes no LDAP de sua instituição refletem as necessidades para a identificação de seus usuários, e devem ser completos o suficiente para suprir os atributos requisitados pela federação CAFé. Segue uma listagem contendo os atributos de um usuário da UFRGS, com os seus respectivos valores, como se encontra na base LDAP:

```
dn: uid=00112389,ou=People,dc=ufrgs,dc=br
objectClass: CourierMailAccount
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
objectClass: ufrgs
cn: JOAO DA SILVA
gidNumber: 100000
homeDirectory: /export/home/0/0/1/2/3/00123456
sn: JOAO DA SILVA
uid: 00123456
displayName: JOAO DA SILVA
gecos: JOAO DA SILVA
loginShell: /bin/false
mail: 00123456@ufrgs.br
mailbox: /export/home/0/0/1/2/3/00123456/Maildir
```



quota: 1048576000S
shadowLastChange: 1
shadowMax: 99999
shadowWarning: 7
ufrgsCategoriaFuncional: 01: ANALISTA DE TECNOLOGIA DA INFORMACAO
ufrgsCategoriaFuncional: 02:
ufrgsCategoriaFuncional: 03:
ufrgsCodCurso: 01:
ufrgsCodCurso: 02: 305
ufrgsCodCurso: 03: 64
ufrgsCodTipoVinculo: 01: 1
ufrgsCodTipoVinculo: 02: 4
ufrgsCodTipoVinculo: 03: 6
ufrgsCurso: 01:
ufrgsCurso: 02: CIÊNCIA DA COMPUTAÇÃO
ufrgsCurso: 03: COMPUTAÇÃO
ufrgsDataAfastamento: 01:
ufrgsDataAfastamento: 02: 23/12/2005
ufrgsDataAfastamento: 03:
ufrgsDataIngresso: 01: 31/07/2008
ufrgsDataIngresso: 02: 12/03/2001
ufrgsDataIngresso: 03: 08/03/2006
ufrgsRamal: 5000
ufrgsTipoVinculo: 01: Tecnico-Administrativo
ufrgsTipoVinculo: 02: Aluno de graduacao
ufrgsTipoVinculo: 03: Aluno de mestrado academico
userPassword:: e1NTSEF9SGM2VF1waW

Definição dos mapeamentos

- ▶ Renomear atributo
 - ▶ uid => brEduAffiliationType
- ▶ Alterar o valor de um atributo
 - ▶ uid + "@ufrgs.br" => mail
- ▶ Modificar valor de sequência de atributos
 - ▶ atrib="01: aluno", atrib="02: bolsista" => aluno;bolsista

Analizando esta listagem, é possível fazer o seguinte mapeamento:

- ▶ eduPersonPrincipalName – Utilizar o atributo “uid” no escopo “ufrgs.br”.
- ▶ mail – Utilizar atributo “mail” já existente.
- ▶ cn – Utilizar a primeira parte do string “cn”.
- ▶ sn – Utilizar o restante do string “cn”.
- ▶ brEduAffiliation – Enviar os valores de “ufrgsTipoVinculo” na ordem correta de acordo com o índice. É preciso remover o índice.
- ▶ brEntranceDate – Modificar o valor de “ufrgsDataIngresso” para se adequar ao padrão AAAMMDD e enviar ordenadamente, removendo o índice.
- ▶ brExitDate – Similar ao “brEntranceDate”, utilizando o atributo “ufrgsDataAfastamento”.

Renamear atributo

A renomeação de um atributo é feita usando apenas uma relação de dependência entre um atributo que deve ser buscado na base LDAP e a definição de outro atributo com o novo nome.

Propriedades da tag *SimpleAttributeDefinition*:

- ▶ id (obrigatório) – Nome do atributo.
- ▶ sourceName (opcional) – Indica o nome do atributo origem cujo valor será copiado.
- ▶ allowEmpty (opcional) – Permite que o valor do atributo seja nulo. Valores possíveis: “true” ou “false”.
- ▶ smartScope (opcional) – Adiciona escopo ao atributo, usualmente o domínio da instituição. O escopo é adicionado como uma propriedade do atributo. Ex.: “ufrgs.br”.

Propriedade da tag *AttributeDependency*:

- ▶ requires (obrigatório) – Indica o nome do atributo que deve ser buscado na base LDAP para satisfazer a dependência.



Exemplo: faz a busca do atributo original na base LDAP e depois o renomeia.

```
<SimpleAttributeDefinition  
    id="urn:mace:dir:attribute-def:telephoneNumber"  
    sourceName="urn:mace:dir:attribute-def:ufrgsRamal">  
    <AttributeDependency  
        requires="urn:mace:dir:attribute-def:ufrgsRamal" />  
</SimpleAttributeDefinition>  
  
<SimpleAttributeDefinition  
    id="urn:mace:dir:attribute-def:uid">  
    <DataConnectorDependency requires="directory"/>  
</SimpleAttributeDefinition>  
  
<SimpleAttributeDefinition  
    id="urn:mace:dir:attribute-def:eduPersonPrincipalName"  
    sourceName="urn:mace:dir:attribute-def:uid"  
    smartScope="ufrgs.br">  
    <AttributeDependency  
        requires="urn:mace:dir:attribute-def:uid" />  
</SimpleAttributeDefinition>  
  
<SimpleAttributeDefinition  
    id="urn:mace:dir:attribute-def:uid">  
    <DataConnectorDependency requires="directory"/>  
</SimpleAttributeDefinition>  
  
<SimpleAttributeDefinition  
    id="urn:mace:dir:attribute-def:eduPersonPrincipalName"  
    sourceName="urn:mace:dir:attribute-def:uid"  
    smartScope="ufrgs.br">  
    <AttributeDependency  
        requires="urn:mace:dir:attribute-def:uid" />  
</SimpleAttributeDefinition>
```



Alterar valor de atributo

A alteração do valor de um atributo pode ser feita utilizando código Java definido no arquivo *attribute-resolver.xml*. O trecho do programa é armazenado na tag “Scriptlet” e compilado em tempo de execução pelo Shibboleth-IdP, durante a carga do Tomcat.

A resolução de atributos é feita através do JNDI (Java Naming and Directory Interface), que é a implementação de um conector do LDAP com o Java. É possível buscar qualquer atributo da base LDAP para efetuar a modificação do seu valor.

O código abaixo busca o valor do atributo “ufrgsTipoVinculo” e armazena-o em um objeto da classe “Attribute”:

- ▶ Attributes atributos = dependencies.getConnectorResolution("directory");
- ▶ Attribute tipoAfiliacao = atributos.get("ufrgsTipoVinculo").

A classe “Attribute” é um vetor de atributos, conforme definido pelo JNDI. Mais informações podem ser encontradas na sua documentação: <http://java.sun.com/products/jndi/1.2/javadoc/>

Para os atributos mono valorados é possível acessar diretamente o valor obtido. O código abaixo busca o primeiro item do vetor, armazenando-o em “valorAtr”.

```
String valorAtr = tipoAfiliacao.get(0)
```

Após qualquer alteração efetuada no valor ele deve ser enviado para o Shibboleth-IdP, que se encarregará de passá-lo para o provedor de serviço da federação. O comando seguinte apresenta o processamento de um atributo e o retorno da variável “valorAtr” para o Shibboleth-IdP, que contém o valor original do atributo convertido para maiúsculas:

```
Attributes atributos = dependencies.getConnectorResolution("directory");
Attribute tipoAfiliacao = atributos.get("ufrgsTipoVinculo");
String valorAtr = tipoAfiliacao.get(0);
valorAtr.toUpperCase();
resolverAttribute.addValue( valorAtr );

<ScriptletAttributeDefinition
    id="urn:mace:dir:attribute-def:cn">
    <DataConnectorDependency requires="directory"/>
    <Scriptlet>
        <![CDATA[
            Attributes atributos = dependencies.getConnectorResolution("directory");
            Attribute cn = atributos.get("cn");
            resolverAttribute.addValue( cn.get(0).substring(0,cn.get(0).indexOf(" ")));
        ]]>
    </Scriptlet>
```



```
</ScriptletAttributeDefinition><ScriptletAttributeDefinition id="urn:mace:dir:attribute-def:cn">
    <DataConnectorDependency requires="directory"/>
    <Scriptlet>
        <![CDATA[
            Attributes atributos =
            dependencies.getConnectorResolution("directory");
            Attribute cn = atributos.get("cn");
            resolverAttribute.addValue(
            cn.get(0).substring(0,cn.get(0).indexOf(" ")));
        ]]>
    </Scriptlet>
</ScriptletAttributeDefinition>
```

 A alteração de valores é efetuada através de código Java inserido na definição de um atributo.

Modificar sequência de atributos

Quando um atributo é multivalorado (como os vínculos de um usuário com uma instituição) é preciso iterar através do vetor “Attribute”, que é retornado da busca no LDAP. O código abaixo apresenta uma iteração no vetor “Attribute” (1), com posterior transferência dos valores para um vetor de strings (2) e a ordenação do vetor (3).

Após efetuadas as modificações nos valores dos atributos, eles devem ser retornados em sequência, com o cuidado de retornar a string “null” caso o valor do atributo seja vazio.

O código abaixo apresenta a modificação de uma sequência de atributos multivalorados do schema UFRGS. O vetor é ordenado de acordo com o índice presente no valor dos atributos “ufrgsDataAfastamento”(1), e posteriormente o resultado é devolvido no formato de data esperado pela federação CAFé (2), mantendo a ordem dos atributos de acordo com o índice original:

```
<ScriptletAttributeDefinition
    id="urn:mace:dir:attribute-def:brEduAffiliationType">
    <DataConnectorDependency requires="directory"/>
    <Scriptlet>
        <![CDATA[
            Vector vect_atributos;
            vect_atributos = new Vector();
            Attributes atributos =
            dependencies.getConnectorResolution("directory");
            Attribute tipoAfiliacao = atributos.
```



```
get("ufrgsTipoVinculo");
for (int i = 0; tipoAfiliacao != null && i <
tipoAfiliacao.size(); i++)
vect_atributos.add(tipoAfiliacao.get(i));
String[] array_atributos = (String [])vect_atributos.toArray(
new String[0] );
java.util.Arrays.sort(array_atributos);

for (int i = 0; i < array_atributos.length; i++ )
if(array_atributos[i].length() > 0)
resolverAttribute.addValue(
array_atributos[i].substring(4) );
Else
    resolverAttribute.addValue(<>null<>);

]]>
</Scriptlet>
</ScriptletAttributeDefinition>

<ScriptletAttributeDefinition id=>urn:mace:dir:attribute-
def:brEduAffiliationType">
<DataConnectorDependency requires="directory"/>
<Scriptlet>
<![CDATA[
    Vector vect_atributos;
    vect_atributos = new Vector();
    Attributes atributos = dependencies.getConnectorRe
solution("directory");
    Attribute tipoAfiliacao = atributos.
get("ufrgsTipoVinculo");
        for (int i = 0; tipoAfiliacao != null && i <
tipoAfiliacao.size(); i++)
            vect_atributos.add(tipoAfiliacao.get(i));
        String[] array_atributos = (String [])vect_
atributos.toArray( new String[0] );
        java.util.Arrays.sort(array_atributos);
        for (int i = 0; i < array_atributos.length; i++ )
            if(array_atributos[i].length() > 0)
                resolverAttribute.addValue( array_
atributos[i].substring(4) );
            else
                resolverAttribute.addValue("null");
]]>
</Scriptlet>
</ScriptletAttributeDefinition>
```



10

Roteiro de Atividades Implantação de provedor de identidade a partir de um diretório existente

Tópicos e conceitos

- Configuração do Shibboleth-IdP
- Análise de um schema LDAP
- Mapeamento de atributos

Competências técnicas desenvolvidas

- Criação de arquivo de configuração do Shibboleth-IdP que contém o mapeamento necessário para utilizar atributos já existentes em uma base LDAP para compartilhar dados com a federação CAFé.

Tempo previsto para as atividades

- 40 – 60 minutos (trabalho em duplas e pesquisa de documentação sobre a linguagem Java).

Atividade 1 – Renomeando um atributo

Defina um mapeamento para o atributo “rfc822MailBox” disponível no esquema abaixo, de modo que o seu conteúdo seja enviado para a federação com o nome de “mail”:

```
dn: uid=00123456,ou=People,dc=ufrgs,dc=br  
uid: 00123456  
cn: JOAO  
sn: DA SILVA  
rfc822MailBox: 00123456@ufrgs.br
```

Atividade 2 – Alterando o valor de um atributo

Defina um mapeamento para o atributo “uid” disponível no esquema abaixo de modo a remover os zeros à esquerda do número:

```
dn: uid=00123456,ou=People,dc=ufrgs,dc=br  
uid: 00123456  
cn: JOAO  
sn: DA SILVA  
rfc822MailBox: 00123456@ufrgs.br
```

Atividade 3 – Múltiplos atributos

Defina um mapeamento para o novo atributo “displayName” concatenando os valores de *cn* e *sn*. Retorne dois resultados diferentes, nas formas “*cn sn*” e “*sn, cn*”.

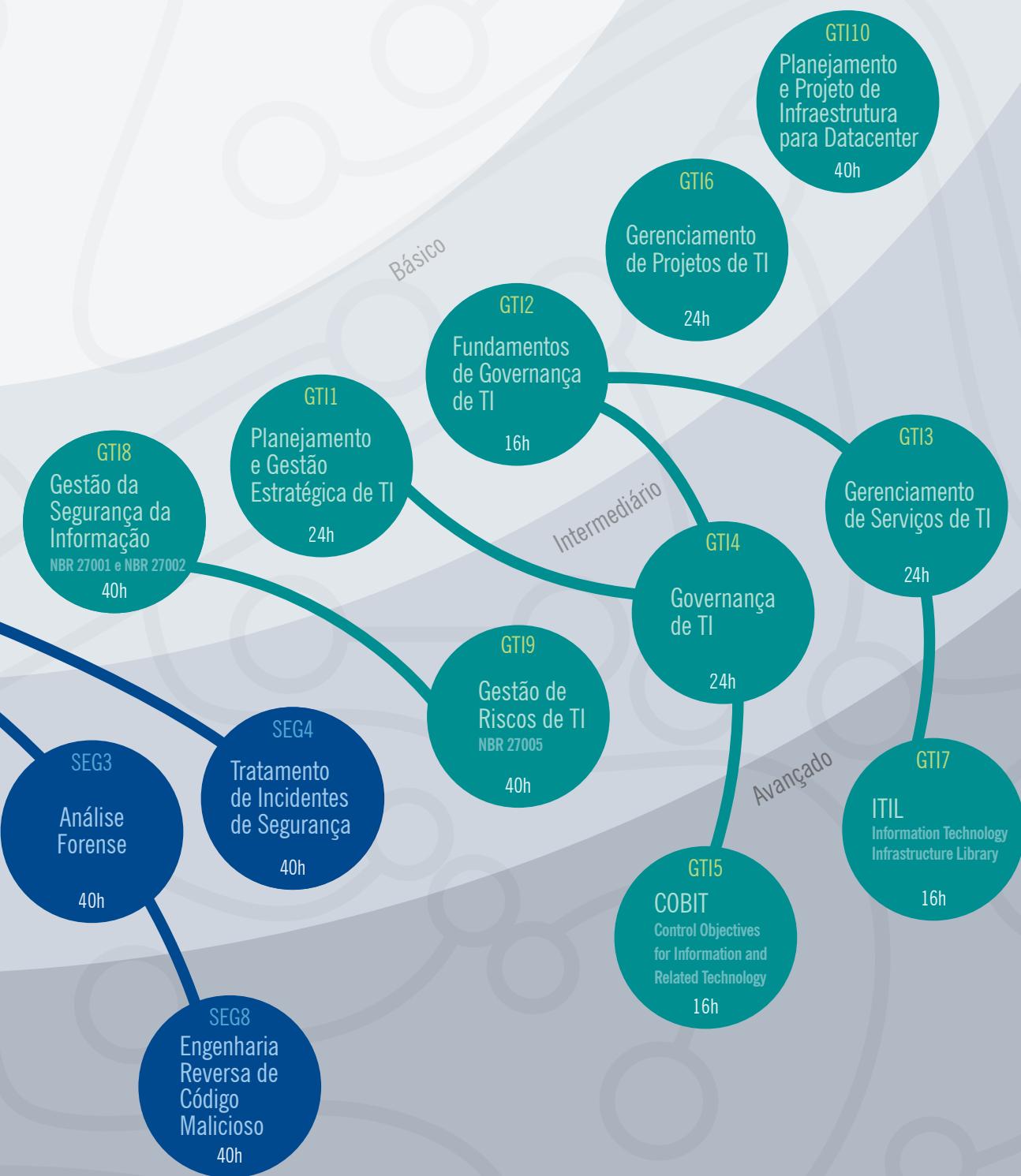
```
dn: uid=00123456,ou=People,dc=ufrgs,dc=br  
uid: 00123456  
cn: JOAO  
sn: DA SILVA  
rfc822MailBox: 00123456@ufrgs.br
```



Bibliografia

- RFC 1558 – A String Representation of LDAP Search Filters
- RFC 2251 – Lightweight Directory Access Protocol (v3)
- RFC 2252 – Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions
- RFC 2254 – The String Representation of LDAP Search Filters
- RFC 2255 – The LDAP URL Format
- RFC 3296 – Named Subordinate References in Lightweight Directory Access Protocol (LDAP) Directories
- Apache Directory Studio: <http://directory.apache.org/studio/>
- OpenLDAP Fundation: <http://www.openldap.org>
- Shibboleth About: <http://shibboleth.internet2.edu/about.html>
- SWITCHaaI Expert Demo: <http://www.switch.ch/aaI/demo/expert.html>
- Shibboleth 1 Documentation: <http://spaces.internet2.edu/display/SHIB/>
- Metadata for the OASIS Security Assertion Markup Language (SAML) v1.1: <http://docs.oasis-open.org/security/saml/v1.1/saml-metadata-1.1-os.pdf>
- SAML V1.1 Technical Overview:
<http://www.oasis-open.org/committees/download.php/6837/sstc-saml-tech-overview-1.1-cd.pdf>
- Shibboleth Update: <http://www.educause.edu/ir/library/powerpoint/EAF0455.pps>
- Roteiro de atividades para entrada de um IdP na CAFé: <http://wiki.rnp.br/display/cafewebsite/Roteiro+de+Atividades+para+Entrada+de+um+IDP>
- Internet2 – Documentação do Shibboleth-IdP:
<https://spaces.internet2.edu/display/SHIB/StaticDataConnector>
<https://spaces.internet2.edu/display/SHIB/ScriptletAttributeDefinition>





Todos os cursos da ESR requerem inglês para leitura e noções de informática e Internet.

Integre sua instituição à Federação CAFé, e faça parte dessa rede de confiança de autenticação de usuários

Este curso é voltado para técnicos ou gerentes de TI que administram e operam as bases de dados das suas instituições. A finalidade é apresentar o conceito de federação acadêmica e capacitar os alunos para a execução das atividades envolvidas na integração da sua instituição à Federação CAFé. O conceito de federação está associado a uma infraestrutura de autenticação e autorização interdomínios que permite ao usuário manter todos os seus atributos e informações na sua instituição de origem, e acessar recursos oferecidos via web por outras instituições da federação. No curso serão estudados os passos para implantar um provedor de identidade institucional — usando a plataforma Shibboleth e o serviço de diretório LDAP — e acoplá-lo à Federação CAFé.

