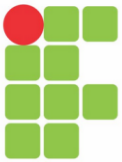


INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E T ECNOLOGIA DO TOCANTINS

CAMPUS PALMAS

CURSO SUPERIOR DE SISTEMAS PARA INTERNET

NARANMÃ SOUSA SANTOS

SISTEMA WEB UTILIZADO PARA O GERENCIAMENTO DE OBJETOS EM UM SERVIÇO DE DIRETÓRIO LDAP

PALMAS

2015

NARANMÃ SOUSA SANTOS

SISTEMA WEB UTILIZADO PARA O GERENCIAMENTO DE OBJETOS EM UM SERVIÇO DE DIRETÓRIO LDAP

Trabalho de Conclusão de Curso apresentado à Coordenação do Curso de Sistemas para Internet do Instituto Federal do Tocantins– Campus Palmas, como exigência à obtenção do grau de tecnólogo em Sistemas para Internet.

Orientador: Prof. Esp. **Fernando Jorge Ebrahim Lima e Silva**

PALMAS

2015

NARANMÃ SOUSA SANTOS

SISTEMA WEB UTILIZADO PARA O GERENCIAMENTO DE OBJETOS EM UM SERVIÇO DE DIRETÓRIO LDAP

Trabalho de Conclusão de Curso apresentado à Coordenação do Curso de Sistemas para Internet do Instituto Federal do Tocantins – Campus Palmas, como exigência à obtenção do grau de tecnólogo.

Aprovado em: _____/_____/_____

BANCA AVALIADORA

_____

FERNANDO JORGE EBRAHIM LIMA E SILVA - Especialista (Orientador)

IFTO – *Campus Palmas*

_____

MARINALDO OLIVEIRA SANTOS - Especialista

IFTO – *Campus Palmas*

_____

FRANCISCO WILLIANS MAKOTO PLÁCIDO HIRANO - Mestre

IFTO – *Campus Palmas*

RESUMO

Este trabalho apresenta a proposta de um sistema Web para o gerenciamento de objetos contidos em um servidor de diretórios LDAP (*Lightweight Directory Access Protocol*), bem como sua implementação no Tribunal Regional Eleitoral do Tocantins como uma alternativa mais prática no gerenciamento de objetos nos servidores de diretório LDAP. Estes servidores contêm objetos que correspondem a contas de usuários, grupos de usuários e computadores presentes em uma organização. O objetivo deste é fornecer uma forma mais simples e rápida no gerenciamento dos objetos para os usuários administradores, além de proporcionar mais segurança permitindo acesso de algumas áreas do diretório LDAP.

ABSTRACT

This work presents the proposal of a Web system for managing objects contained in a directory server Lightweight Directory Access Protocol (LDAP), as well as their implementation in the Regional Electoral Court of Tocantins as a more practical alternative for managing objects in LDAP directory servers. These servers contain objects that correspond to user accounts, groups of users and computers in an organization. The aim of this is to provide a more simple and quick management of objects for users, administrators as well as providing more security allowing access to some areas of the LDAP directory.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

INTRODUÇÃO

1 INTRODUÇÃO

O LDAP é um protocolo que fornece armazenamento de informações e responde a consultas por TCP/IP (*Transmission Control Protocol/Internet Protocol*) (Santos, 2007). Seus métodos de consultas são leves e rápidos devido às suas poucas operações, e por serem mapeados diretamente na camada de transporte, usando o protocolo TCP (CARTER, 2009). O LDAP permite uma busca complexa no diretório e com pouca informação de um objeto, pode se recuperar seus atributos (MOTA, 2008). Por exemplo, na busca de um usuário pelo sobrenome pode-se conseguir o endereço de e-mail, telefone, entre outros atributos.

O LDAP fornece aos administradores de rede uma forma de centralizar informações, possibilitando o gerenciamento de níveis de controle de acesso aos recursos e serviços disponíveis na rede (Isquierdo, 2001). Em uma rede que utiliza LDAP uma única conta de usuário possibilita acesso a diversos sistemas com uma única senha.

Este trabalho apresenta um modelo, bem como sua implementação de uma aplicação Web que se integre com um serviço de diretórios LDAP, permitindo que a aplicação manipule os objetos e informações em seus atributos.

A implementação foi realizada no Tribunal Regional Eleitoral do Tocantins, como alternativa para o gerenciamento de objetos nos servidores de diretórios LDAP, fornecendo ao administrador de redes operações de inserção, modificação e consulta de informações de forma rápida e fácil. Além de fornecer automatização e controle de restrições das operações no servidor aos usuários, permitindo assim que usuários com pouco conhecimento gerenciem seus diretórios.

1.1 Justificativa

Servidores de diretório LDAP fornecem formas de gerenciar seus diretórios dentre os quais podemos citar: Active Directory da Microsoft, phpLDAPadmin, JXplorer, onde permitem acesso completo aos diretórios do servidor LDAP. Logo, o usuário poderá realizar operações sem muitas restrições.

Com a implementação do sistema Web que gerencie os objetos do servidor de diretórios, é possível realizar uma simplificação das operações e restringir algumas áreas e tipos de operações sobre os objetos, de forma que o

usuário operador do sistema tenha um determinado limite em suas operações. Isso permite que usuários com pouco conhecimento sobre o diretório utilizem o sistema. A visão do sistema faz com que os usuários operadores não percebam que estão trabalhando com um serviço de diretório LDAP.

Esta solução é inédita e foi desenvolvida por não haver solução similar que atenda aos requisitos de segurança.

1.2 Objetivos

1.2.1 Objetivo geral

O objetivo do projeto é o desenvolvimento de uma aplicação Web capaz de gerenciar objetos de um servidor de diretórios LDAP, permitindo facilitar e diminuir o tempo como as operações realizadas por usuários administradores.

Após a implementação do sistema, será verificado o seu uso de acordo com suas funções.

1.2.2 Objetivos específicos

- integrar um sistema Web em um servidor de diretórios LDAP;
- reduzir o tempo de operações realizadas por administradores no servidor de diretórios LDAP;
- facilitar as operações no gerenciamento de usuários e computadores, com as diretrizes de segurança estabelecidas.

1.2.3 Organização do trabalho

O presente trabalho está distribuído em sete seções sendo a primeira seção a introdução. A segunda seção descreve as tecnologias envolvidas no sistema, como ferramentas, linguagens entre outros. Em seguida, na terceira e quarta seção é mostrada a arquitetura proposta, descrevendo as ferramentas e o ambiente, bem como a metodologia. A quinta seção explica alguns códigos e bibliotecas usadas no desenvolvimento do sistema mostrando como o sistema

funciona e suas características. Ao final, nas seções seis e sete, temos os resultados e as considerações finais.

FUNDAMENTAÇÃO TEÓRICA

2 FUNDAMENTAÇÃO TEÓRICA

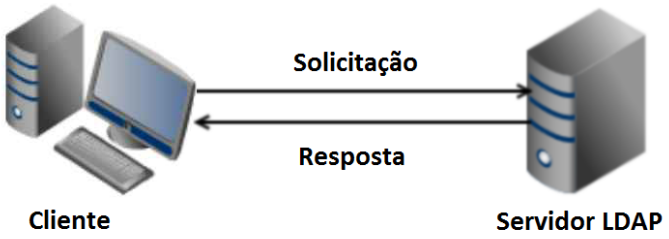
2.1 Serviço de diretório LDAP

De acordo com (Mota, 2008) e (Isquierdo, 2001) um serviço de diretório possibilita a armazenagem e organização de informações de maneira semelhante a um banco de dados, mas tendem a conter mais informações baseadas em atributos. Já seus dados são organizados de forma hierárquica semelhante a um sistema de arquivos. A base de dados pode ser fisicamente distribuída, mas logicamente centralizada. Essa base contem objetos que podem fazer parte de informações em uma rede de computadores, como, por exemplo, estes objetos podem ser pessoas ou recursos como endereços de e-mail, computadores, listas de distribuição e serviços.

Baseado no modelo cliente-servidor o servidor de diretório LDAP é responsável pela disponibilização do serviço de diretório, onde é realizado o armazenamento das informações de forma hierárquica. Uma característica do serviço é que este pode ser distribuído por mais de um servidor, assim um cliente pode se conectar a um dos servidores e ter acesso às mesmas informações. Conforme (Mota, 2008), o cliente se conecta iniciando uma sessão com o servidor, este verifica as credenciais e permite, ou não, a realização de consultas ou alterações dessas informações.

A figura 1 demonstra a comunicação do cliente com o servidor LDAP, onde o cliente inicia uma solicitação, em seguida o servidor responde a solicitação do cliente.

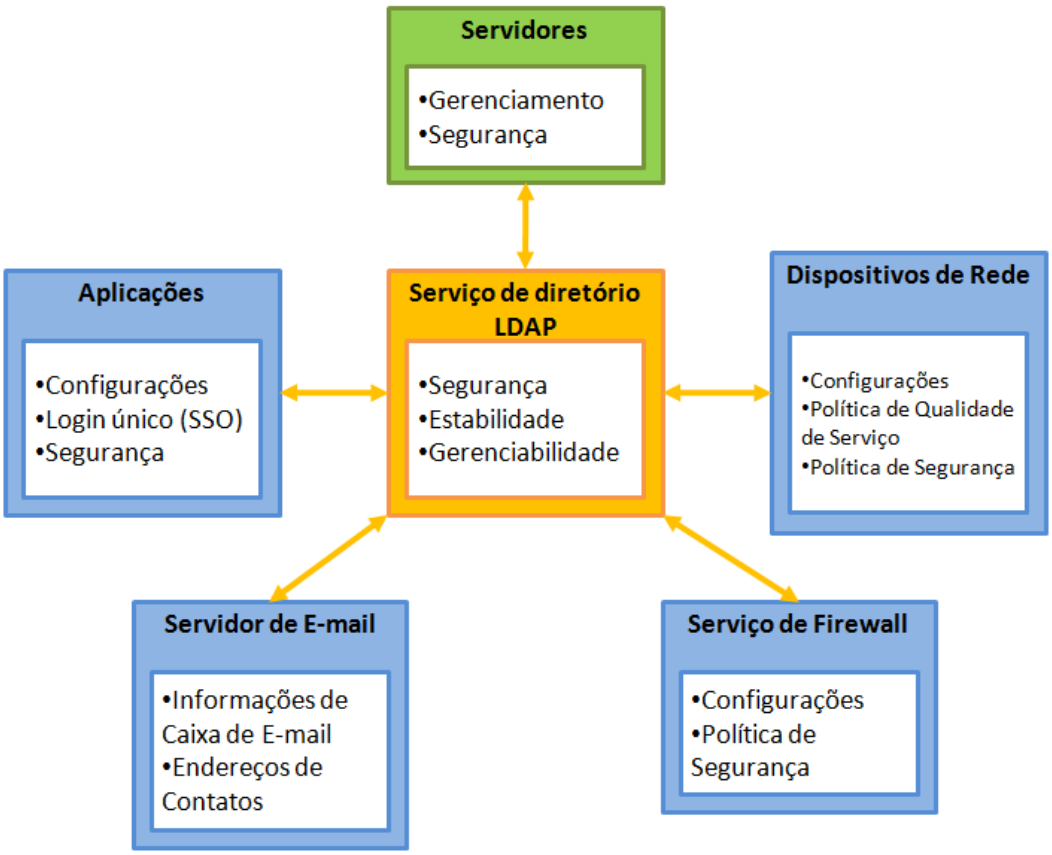
Figura 1–Comunicação entre o cliente e servidor LDAP



Fonte: Elaborado pelo autor

Um servidor LDAP permite que vários sistemas, aplicações e serviços contidos em um ambiente de rede possam utilizar um repositório de informações centralizado, usado principalmente para realizar autenticação de usuários e estabelecer regras e permissões sobre um determinado serviço ou recurso (Gil, 2012). A figura 2 mostra como o serviço de diretório LDAP pode ser usado por vários serviços e sistemas.

Figura 2– Integração do serviço de diretório LDAP com outros sistemas



Fonte: Adaptado de msdn.microsoft.com

2.2 LDAP

De acordo com (Gil, 2012) o LDAP (*Lightweight Directory Access Protocol*) é um protocolo que define a forma de funcionamento de um serviço de diretórios, fornecendo as especificações de como as informações serão armazenadas e fornecidas.

2.2.1 Modelos LDAP

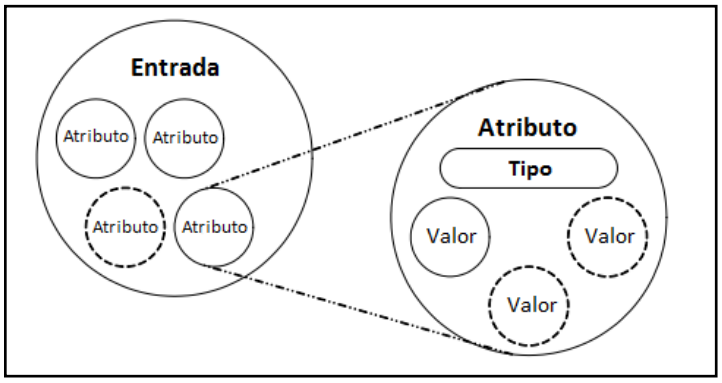
O LDAP é composto por quatro modelos básicos que descrevem seu funcionamento por completo. Estes definem suas operações, e como as informações devem ser armazenadas.

Modelo de Informação: Define a estrutura da informação e como pode ser armazenada em um diretório LDAP.
Modelo de Nomes: Define como a informação em um diretório LDAP é organizada e identificada.
Modelo Funcional: Descreve as operações que podem ser feitas com as informações no diretório LDAP, é como ela pode ser acessada e alterada.
Modelo de Segurança: Descreve como a informação no diretório LDAP pode ser protegida contra acesso não autorizados.

2.2.2 Modelo de informação

As informações que são armazenadas no diretório LDAP são chamadas de entradas (ou objetos). Cada entrada representa um objeto como, pessoas, servidores, organizações, dentre outros. Um objeto possui um conjunto de informações denominadas de atributos e podem armazenam um tipo e um ou mais valores que são associados a uma sintaxe. A sintaxe define qual tipo de valor deve ser armazenado no atributo como texto, números, se este é opcional, o tamanho, entre outros. A figura 3 demonstra as informações de uma entrada.

Figura 3 – Representação de uma entrada, atributos e valores.



Fonte: (TUTTLE et al., 2006)

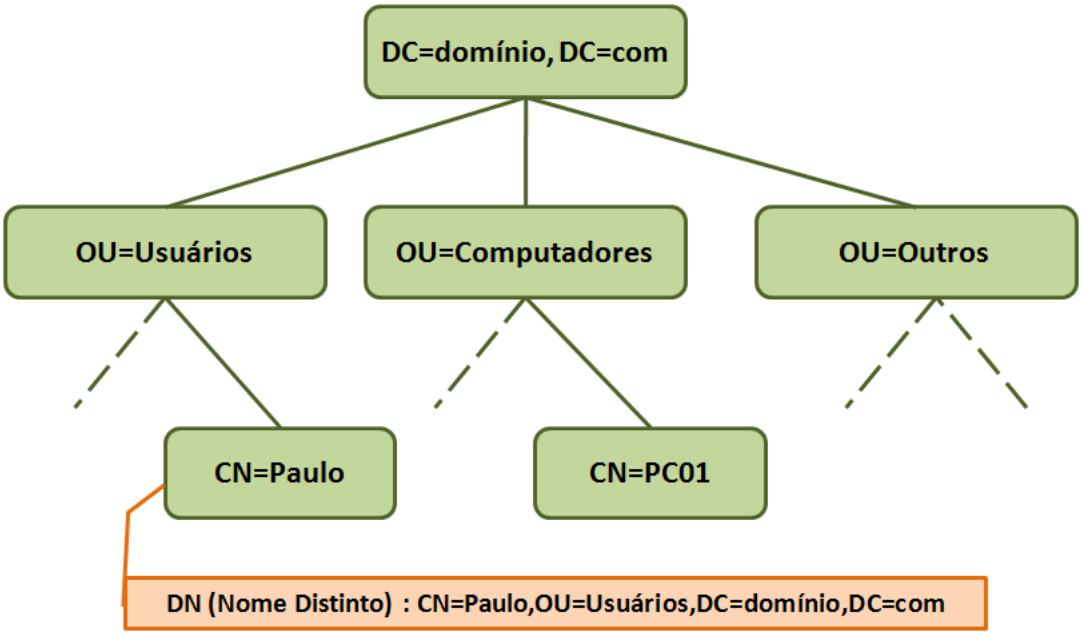
2.2.2.1 LDIF

LDIF (*Data Interchange Format*) é um formato de troca de dados suportado pelo LDAP, este permite a manipulação de grande quantidade de dados em um arquivo de texto plano que representa os atributos de uma ou várias entradas (objetos) do diretório. Segundo (CARTER, 2009) os arquivos no formato LDIF são utilizados para importar novas entradas ou realizar alterações nas entradas existentes.

2.2.3 Modelo de nomes

Um diretório LDAP possui uma estrutura de árvores hierárquica conhecida como DIT (*Directory Information Tree*). Essa árvore possui entradas (objetos) que são compostos de atributos e estes variam dependendo do tipo de entrada. Cada entrada possui um caminho único chamado de nome distinto ou DN (*Distinguished Name*). A figura 4 exemplifica as relações em uma árvore de diretórios LDAP.

Figura 4– Estrutura de uma arvore de diretório LDAP



Fonte: Elaborado pelo autor

Na figura 4 cada retângulo representa um objeto na árvore de diretório, em que, o primeiro objeto da árvore é chamado de componente do domínio ou DC

(*Domain Component*) e a partir dele temos os outros objetos. Em seguida, as unidades organizacionais ou OU (*Organizational Unit*) podem ser comparadas como uma pasta de um sistema de arquivos possuem a função de organizar. Na seguência temos os objetos que estão na ponta e são chamados de nome comum ou CN (*Common Name*). Cada entrada no diretório possui um atributo único chamado de nome distinto ou DN (*Distinguished Name*) que possui o caminho completo até chegar ao objeto, um exemplo pode ser visto na figura 4, onde é mostrado o DN do objeto Paulo.

2.2.4 Modelo Funcional

O modelo funcional do protocolo LDAP descreve as operações básicas que podem ser realizadas nos objetos contidos em uma árvore de diretórios. Estas operações são organizadas em três grupos como mostrado abaixo:

Autenticação

- Associar, autenticar;
- Dissociar, encerrar a conexão;
- Abandonar uma operação.

Busca

- Buscar entradas no diretório;
- Comparar se uma entrada possui um valor de atributo dado.

Modificação

- Adicionar uma entrada;
- Remover uma entrada;
- Modificar valores de uma entrada.

2.2.4.1 Operação de busca

A operação de busca permite que o cliente possa realizar consultas das informações no servidor LDAP. Estas operações podem ser simples ou complexas em visto que a operação de busca permite usar filtros que podem especificar em

qual ponto da árvore de diretórios deve ser iniciada, além da profundidade da árvore e quais atributos de uma entrada dever ser consideradas.

2.2.4.2 Comparação

Essa operação permite comparar os valores de atributos de uma entrada. Se um valor corresponder com o valor de comparação, este retorna verdadeiro, caso contrário, retorna falso. Compare é semelhante à operação busca utilizando filtros de pesquisa.

2.2.4.3 Operação de atualização

As operações de atualização modificam o conteúdo de um diretório. A tabela 1 detalha as operações.

Tabela 1- Operações de atualização

Operação	**Descrição**
Adicionar (add)	Insere uma nova entrada no diretório.
Apagar (delete)	Apaga entradas existentes de um diretório.
Modificação (modify)	Faz alteração dos valores dos atributos de uma entrada existente.
Modificação do DN (modify DN)	Faz alteração do DN ou move de um subdiretório para uma nova localização.

Fonte: (TUTTLE et al., 2006)

2.2.4.4 Operação de autenticação

Operação usada para estabelecer um início ou término de uma sessão entre o cliente e um servidor LDAP. Essas sessões podem ser anônimas ou autenticadas. Nas sessões autenticadas o cliente se identifica por uma senha que pode ser, ou não, segura pelo método de criptografia. A tabela 2 resume as operações.

Tabela 2 - Operações de autenticação

Operação	Descrição
Associa (bind)	Inicia uma sessão LDAP entre um cliente e um servidor.
Desassocia (ubind)	Termina uma sessão entre um cliente e um servidor.
Abandonar (abondon)	Abandona uma solicitação iniciada pelo cliente.

Fonte: (TUTTLE et al., 2006)

2.2.5 Modelo de segurança

O modelo de segurança é baseado na operação de associação (*bind*). Existem diferentes tipos de operações de associação, assim diferentes mecanismos de segurança são aplicadas. Um exemplo possível é um cliente solicitar acesso fornecendo apenas uma identificação DN e uma senha simples sem criptografia. O uso de senhas não criptografadas é fortemente desencorajada quando o serviço de transporte subjacente não pode ter garantia de confidencialidade e, portanto, pode resultar na divulgação da senha para partes não autorizadas.

A versão três do protocolo LDAP possui um comando de associação com suporte ao mecanismo SASL (*Simple Authentication and Security Layer*). Este é um *framework* de autenticação geral, onde vários métodos de autenticação diferentes são disponíveis para autenticar o cliente com o servidor.

2.3 Plataforma .NET

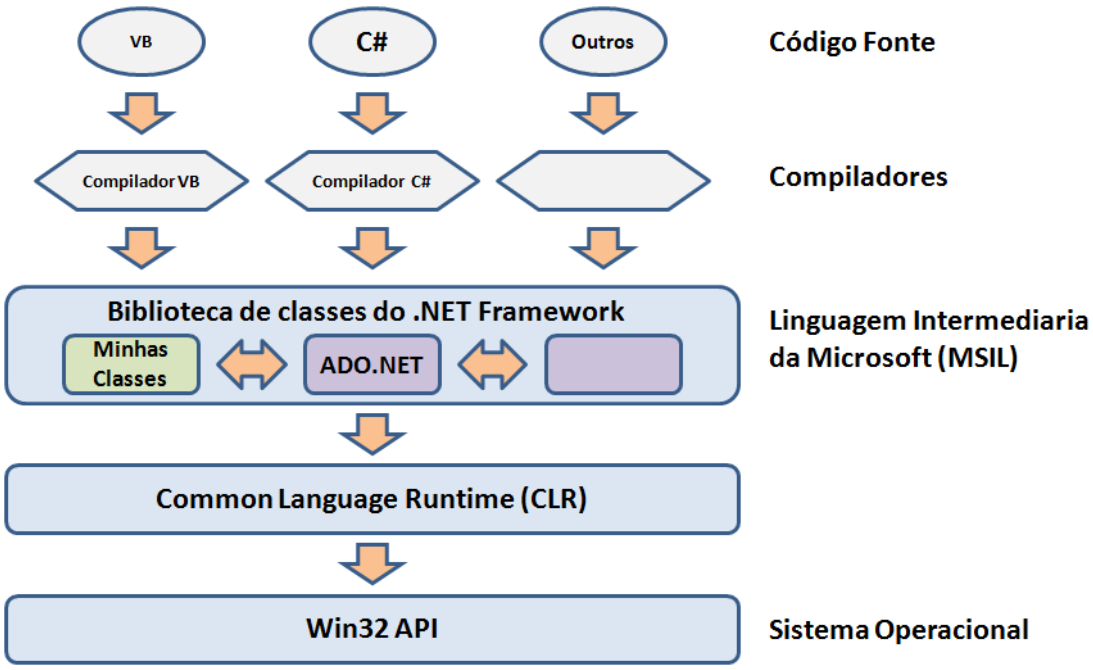
A plataforma .NET *Framework* foi desenvolvida pela Microsoft e permite que aplicações executem independente do sistema operacional, idéia semelhante da máquina virtual Java. Os desenvolvedores criam programas para executarem na plataforma, em vez do sistema operacional, assim qualquer aplicação que for baseada no .NET será possível executar em qualquer sistema que possui a máquina virtual .NET *Framework*(DURAES, 2008).

A CLR (*Common Language Runtime*) é responsável pela execução de todas as aplicações feitas em .NET. Ela gerencia o código fonte da aplicação em tempo de execução fornecendo serviços como requisições de acesso a memória e *hardware*.

Várias linguagens são suportadas pela plataforma, isso por que todos os códigos serão convertidos para uma linguagem intermediária MSIL (*Microsoft Intermediate Language*) ou simplesmente IL, o qual é independe de CPU (*Central Processing Unit*). Para o código MSIL ser executado é necessário converter para o código nativo da máquina, esse papel e realizado pelo compilador JIT (*Just-In-Time*) presente na CLR, no qual interpreta e executa o código (LOTAR, 2010).

Na figura 5 e possível observar como os componentes da arquitetura da plataforma .NET são organizados.

Figura 5– Arquitetura da plataforma .NET Framework



Fonte: Adaptado de msdn.microsoft.com

2.4 ASP.NET

ASP.NET é uma plataforma baseada na .NET *Framework* para o desenvolvimento de aplicações Web. Possui grande número de controles pré-construídos, evitando que o desenvolvedor construa páginas complexas. Estes

controles são manipulados pelo código fonte da página e segue o modelo de programação baseada em eventos tornando mais rápido a construção de formulários (SANTANA FILHO, 2002).

As aplicações em ASP.NET suportam todos os recursos da plataforma .NET *Framework* incluindo as linguagens VB.NET (*Visual Basic .NET*) e o C# (*C Sharp*), compilação e depuração. O código do aplicativo é separado do código HTML (*Hyper Text Markup Language*) devido ao seu novo modelo de codificação chamado de *Code Behind.* Isso permite que o HTML seja escrito em um arquivo separado da parte lógica da aplicação tornado-a mais limpa. (DURAES, 2008).

2.5 Linguagem C#

O C# (*C Sharp*) é uma linguagem de programação orientada a objeto projetada para o desenvolvimento de uma variedade de aplicações compatíveis com a plataforma .NET *Framework*. Foi derivada das linguagens de programação C e C++. A sua sintaxe é simples e de fácil aprendizado semelhante às linguagens C, C++ e Java. Devido essa semelhança os desenvolvedores destas linguagens terão mais facilidade de desenvolver aplicações (Microsoft Developer Network, 2014).

2.6 LDAP na indústria

2.6.1 Microsoft Active Directory

O AD (*Active Directory*) da Microsoft fornece um serviço de diretórios que trabalham em redes distribuídas. O AD já vem integrado em versões do sistema operacional Windows Server, permitindo que as organizações gerenciem seus recursos e usuários através de um servidor centralizado (Amol, 2004).

Podemos dizer ainda que:

> "Ele permite que os administradores gerenciem as informações de toda a empresa de forma eficiente a partir de um repositório central que pode ser distribuído globalmente. Uma vez que informações sobre os usuários e grupos, computadores e impressoras, aplicativos e serviços foram adicionadas ao Active Directory, podem ser disponibilizadas para utilização em toda a rede para muitas ou poucas pessoas, como você preferir. A estrutura da informação pode corresponder à estrutura da sua organização e seus usuários podem consultar o Active Directory para encontrar a localização de uma impressora ou o endereço de e-mail de um colega. Com

> unidades organizacionais, você pode delegar controle e gestão dos dados, contudo, você vê o ajuste."(DESMOND et al., 2013).

2.6.2 OpenLDAP

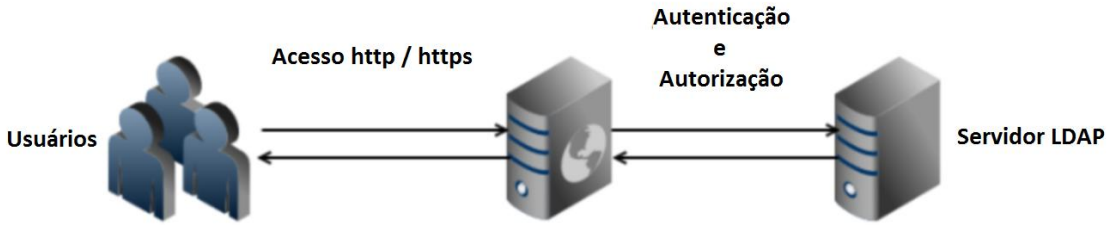
O OpenLDAP é um projeto de código fonte aberto que fornece um serviço de diretório para diferentes sistemas operacionais. O OpenLDAP é compatível com as especificações do LDAPv3, que pode ser usado sobre as versões do IPv4 (*Internet Protocol version 4*) e IPv6 (*Internet Protocol version 6*) (Amol, 2004).

ARQUITETURA PROPOSTA

3 ARQUITETURA PROPOSTA

A arquitetura proposta é formada por dois servidores, sendo um servidor Web e um servidor de diretório LDAP. O cliente usará o navegador (*browser*) para acessar o sistema no servidor Web o qual se comunica com o servidor de diretórios LDAP. Todas as comunicações foram realizadas através de uma rede interna LAN (*Local Area Network*), mas nada impede que esse acesso seja feito por WAN (*Wide Area Network*).

Figura 6– Arquitetura da aplicação Web



Fonte: Elaborado pelo autor

Como podemos observar na figura 6, o usuário inicia o acesso trocando mensagens entre o navegador e o servidor Web utilizando o protocolo HTTPS, esse inicia a comunicação com o servidor LDAP, inicialmente são trocadas mensagens de verificação de autenticação. Caso o usuário se autentique, as mensagens trocadas entre os dois servidores podem ser de consultas ou modificações de objetos no servidor LDAP, dependendo do tipo de acesso que o usuário tiver estarão disponíveis todas ou algumas das funcionalidades.

METODOLOGIA

4 METODOLOGIA

Foram realizadas pesquisas bibliográficas sobre o tema relacionado em artigos científicos e livros atualizados.

Foi realizado o levantamento sobre as funcionalidades do sistema (requisitos funcionais e não funcionais) bem como, sua integração com os outros sistemas envolvidos. Foi necessário o uso de ferramentas que auxiliaram na modelagem dos requisitos do sistema, como a linguagem UML (*Unified Modeling Language*) capaz de gerar modelos visuais do comportamento e estrutura do sistema. Para o desenvolvimento foi utilizada a linguagem de programação orientada a objeto C# na plataforma ASP.NET que fornece a construção de aplicativos dinâmicos na Web.

A comunicação realizada entre a aplicação Web ASP.NET e o servidor que possui o serviço de domínio do Active Directory, são feitos através das classes *DirectoryEntry* e *DirectorySearcher* presentes na biblioteca do .NET *Framework*. Estas usam a tecnologia ADSI (*Active Directory Services Interfaces*) que permite as aplicações interagirem com diversos diretórios em uma rede usando uma única interface.

A classe *DirectoryEntry* encapsula um nó ou um objeto da hierarquia de serviços de domínio do Active Directory. Permite acesso às propriedades do objeto como leitura e modificação de atributos, além de permitir navegação e gerenciamento como por exemplo, criar, excluir e renomear um nó ou objeto. Já a classe *DirectorySearcher* permite realizar consultas na arvore de diretório através dos atributos de um objeto.

Foram utilizadas outras linguagens no desenvolvimento do sistema como:

- HTML (*Hyper Text Markup Language*) linguagem usada para criar página Web;
- CSS (*Cascading Style Sheets*) linguagem usada para manipular a visualização de elementos em HTML;
- JavaScript usado para tornar páginas HTML mais dinâmicas e interativas.

4.1 Ferramentas

As seguintes ferramentas foram utilizadas na implementação da proposta:

- #develop (abreviação de SharpDevelop): é uma IDE gratuito para o desenvolvimento de aplicações para a plataforma .NET *Framework*. Suporta várias linguagens como C#, VB.NET, F#, IronPython e IronRuby, além de ambientes gráficos como Windows Forms, MVC ASP.NET. O projeto foi iniciado em setembro de 2000 por Mike Krüger.
- Astah community: ferramenta capaz de criar diagramas dinâmicos, usado para modelagem de sistema utilizando a UML. Desenvolvido na plataforma Java.

4.2 O Ambiente

Composto por dois servidores, ambos com sistema operacional Windows 2008 Sever R2, onde um é responsável pelo serviço de diretórios Active Directory e o outro pelo serviço Web IIS 7.5 (*Internet Information Services*) além de possuírem o *Framework* .NET instalado para o correto funcionamento da plataforma ASP.NET.

Tabela 3- Principais características dos servidores utilizados

	Servidor de diretórios LDAP	**Servidor Web**
Sistema Operacional	Windows 2008 Sever R2 Enterprise Server Pack 1 64-bits	Windows 2008 Sever R2 Enterprise Server Pack 1 64-bits
Processador	Westmere E56xx/L56xx/X56xx (Nehalem-C) 3.46 GHz 2 Processadores	Westmere E56xx/L56xx/X56xx (Nehalem-C) 3.46 GHz 2 Processadores
Memória	4,00GB	4,00GB
Serviço / Software	Active Directory Domain Services Microsoft .NET *Framework* 4.5.1	Web Service IIS 7.5 Microsoft .NET *Framework* 4.5.1

Fonte: Elaborado pelo autor

O sistema foi desenvolvido de forma iterativa e incremental, disponibilizando aos usuários partes do sistema que ficarem prontas. Na medida em que os módulos forem desenvolvidos serão testados e liberados para os usuários. Estes poderão ser melhorados conforme forem testados e usados pelos operadores.

DESENVOLVIMENTO

5 DESENVOLVIMENTO

No desenvolvimento do sistema foram utilizadas várias bibliotecas da plataforma .NET,mas somente as consideradas principais serão abordadas usando exemplos de implementação.

5.1 Exemplos de programação em C# usando o Serviço de Diretórios

O .NET *framework* usa os *namespaces* para organizar suas muitas classes, assim as classes *DirectoryEntry* e *DirectorySearcher* estão presentes no *namespace System.DirectoryServices.* Dessa forma, para utilizá-las na programação, é necessário fazer referência ou chamada através do bloco de código informado no quadro 1.

Quadro 1 – Referenciando um namespace

```
using System.DirectoryServices;
```

Fonte: Adaptado de msdn.microsoft.com

5.1.1 Conectando-se ao Diretório

O bloco de código representado pelo quadro 2, demonstra uma simples conexão ao domínio “servidor.com” utilizando a classe *DirectoryEntry*. O objeto *entrada* foi iniciado colocando-se o diretório LDAP no método construtor da classe.

Quadro 2 – Construindo um objeto de diretório LDAP

```
// Conectando ao diretório raiz servidor.com
DirectoryEntry entrada = new DirectoryEntry(
                              "LDAP://DC=servidor,DC=com"
                              );
```

Fonte: Adaptado de msdn.microsoft.com

O exemplo demonstrado no quadro 3, foram acrescentados mais três parâmetros no método construtor que são responsáveis pela autenticação de um

usuário para conexão no servidor LDAP. O primeiro é o diretório raiz do domínio, em seguida os dois parâmetros usuário e senha servem para autenticação de um usuário. O último marca o tipo de conexão como segura.

Quadro 3 – Construindo um objeto de diretório LDAP com autenticação

```
DirectoryEntry entrada = new DirectoryEntry(
                              "LDAP://DC=servidor,DC=com",
                              "usuario",
                              "senha",
                              AuthenticationTypes.Secure
                              );
```

Fonte: Adaptado de msdn.microsoft.com

5.1.2 Buscando em um diretório

A operação de busca no diretório é feita através da classe *DirectorySearcher*. Esta classe possui várias propriedades e métodos para realizar desde uma simples busca no diretório, como também buscas complexas utilizando a opção de filtros. Um objeto *DirectorySearcher* sempre precisa de um objeto *DirectoryEntry* como forma de conexão com o diretório e como objeto base para a busca. No bloco de código do quadro 4, é demonstrada uma simples busca no diretório raiz *servidor.com* a procura de todas as entradas do tipo usuário.

Quadro 4 – Consulta em um diretório com filtro

```
// Conectando ao diretório raiz servidor.com.
DirectoryEntry entrada = new DirectoryEntry(
                              "LDAP://DC=servidor,DC=com"
                              );
// Cria um objeto DirectorySearcher.
DirectorySearcherbusca = new DirectorySearcher(entrada);

// A propriedade Filter e usada para filtrar apenas os objetos do
// tipo usuário
busca.Filter = "(&(objectClass=user)(objectCategory=person))";

// Cria um objeto SearchResultCollection para
// receber a coleção de objetosSearchResults
// retornado pelo métodoFindAll.
SearchResultCollection resultado = busca.FindAll();
// As informações da pesquisa pode ser obtidas pelo objeto resultado
```

Fonte:Adaptado de msdn.microsoft.com

5.2 O SISTEMA

O Sistema foi desenvolvido em ASP.NET e C# devido à fácil utilização e integração de recursos e serviços. Outro fator foi o ambiente que já estava disponível com servidores instalados e com as configurações adequadas. O acesso é feito pela Web através de um navegador utilizando o protocolo HTTPS (*Hyper Text Transfer Protocol Secure*) que permite a transferência de informações de maneira segura. O sistema permite realizar tarefas de gerenciamento de contas de usuários, computadores e grupos no servidor de domínio com Active Directory.

O sistema é formado por módulos, cada um realiza uma tarefa específica no servidor de diretórios. Os usuários operadores possuem acesso apenas aos módulos que tem permissão. Abaixo os módulos disponíveis no sistema e a tela inicial apresentado ao usuário operador na figura 7:

- Criação de usuário;
- Edição de usuário;
- Desativação de usuário;
- Desbloqueio e alteração de senha;
- Visualização de listas e grupos;
- Edição de computadores.

Figura 7– Página principal do sistema

Fonte: Elaborado pelo autor

A cada operação de modificação realizada pelo usuário no sistema, é gerado um registro em um arquivo de *log*. Esse arquivo armazena as informações das operações que foram realizadas nos módulos, o operador e os objetos que foram modificados, sendo possível assim, o monitoramento das operações que estão sendo realizadas pelo sistema. Além da possibilidade de desfazer algumas modificações realizadas sobre um objeto do serviço de diretório.

Para fácil implantação e utilização do sistema, foi criado um arquivo de configuração no formato XML (*Extensible Markup Language*), esse arquivo permite guardar informações que podem ser facilmente modificadas sem a necessidade de mexer diretamente no código fonte da aplicação. As configurações são definidas através de atributos pré-definidos, chave e valor. Assim, esse possui informações sobre o servidor de diretório LDAP e configurações referentes às permissões dos módulos entre outras informações importantes para o correto funcionamento do sistema.

O acesso de usuários operadores no sistema é feito através de um cadastro realizado no Active Directory. A conta de usuário criada deve pertencer aos grupos de segurança que foram criados e vinculados aos módulos do sistema. Estes grupos, estão presentes no arquivo de configuração da aplicação, onde consta o módulo do sistema e o grupo no AD que o pertence. Dessa forma, quando um usuário operador acessa o sistema e tenta se autenticar, será verificado se o mesmo é membro de algum grupo, se for, o módulo pertencente ao grupo será liberado, caso contrário não será.

5.3 Funcionalidades do sistema

As funcionalidades são basicamente as mesmas realizadas diretamente no Active Directory, a diferença é que, no sistema estas funcionalidades são organizadas de maneira mais simples, além de automatizar algumas tarefas, tornando-se mais rápido para realizá-las.

5.3.1 Criação de usuário

Fornece opção de criar novos usuários. Os campos possuem validação tanto no cliente como no servidor, Após o novo usuário ser criado à senha é gerada de forma aleatória e temporária. Possui verificação de login disponível e usuários já existentes, evitando a criação de usuário duplicado. A figura 8 mostra como o formulário é exibido no navegador.

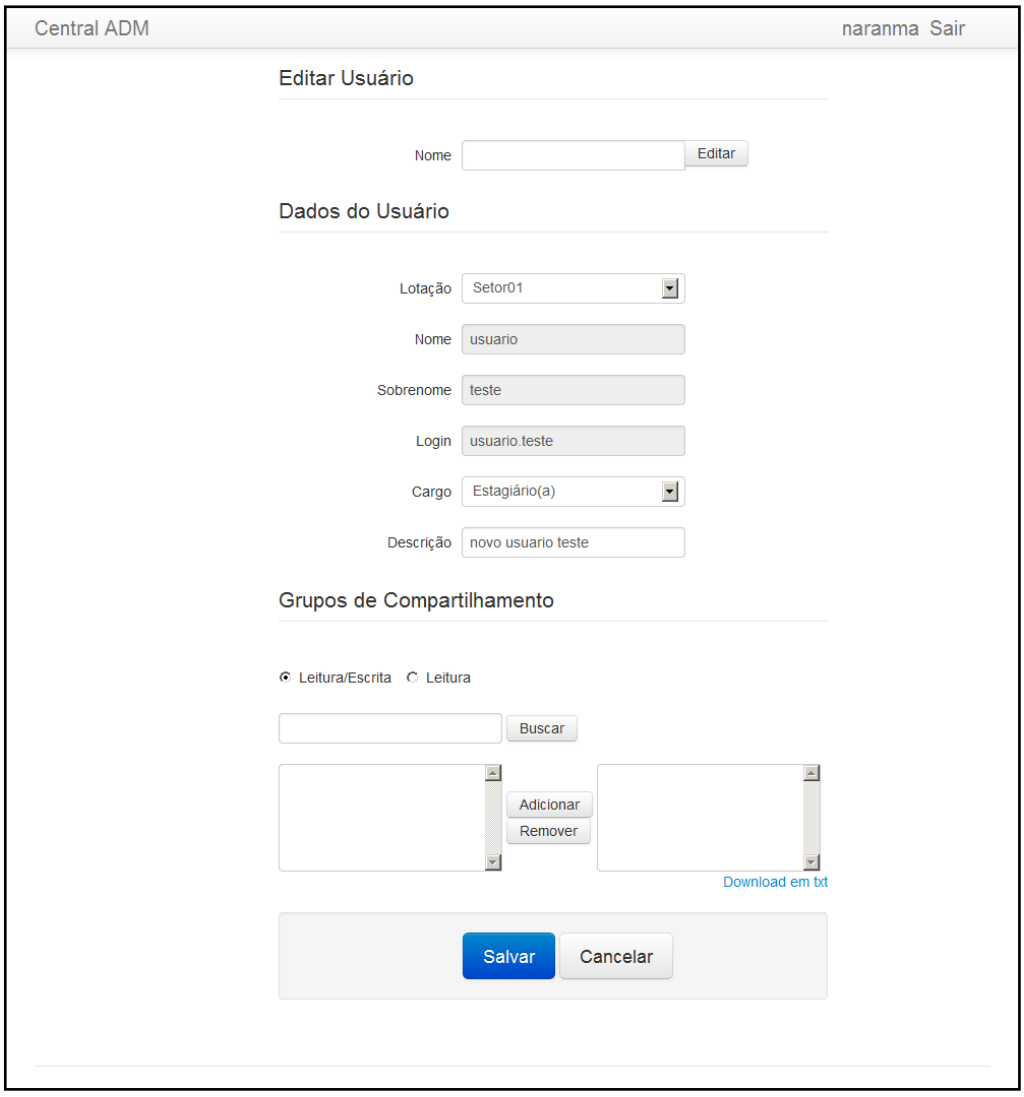
Figura 8– Página de criar usuário

Fonte: Elaborado pelo autor

5.3.2 Edição de usuário

Fornece opção de edição dos atributos do usuário, apenas alguns campos são permitidos como lotação, descrição, grupos e listas de e-mail. A seguir, na figura 9, é mostrado com a página e exibida.

Figura 9– Página de editar usuário

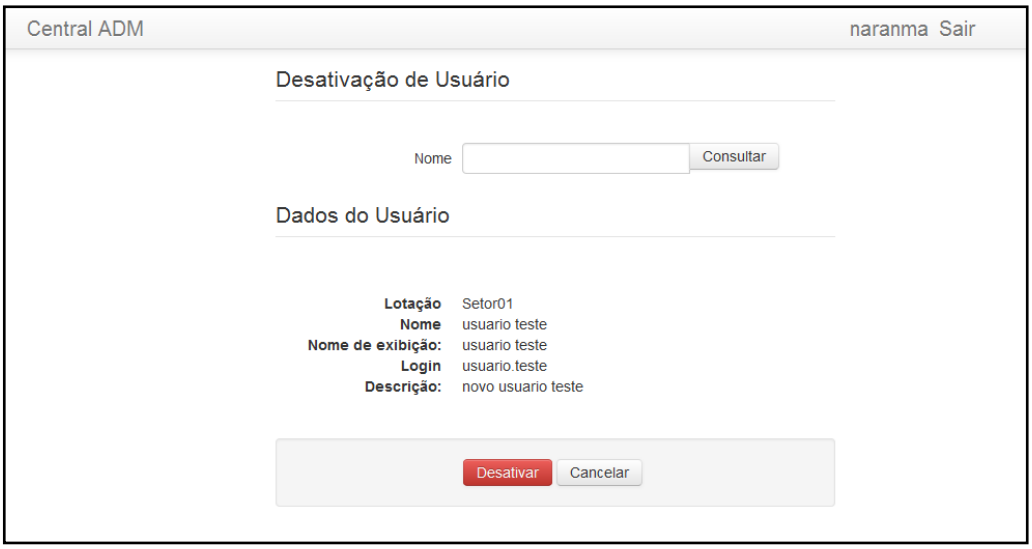


Fonte: Elaborado pelo autor

5.3.3 Desativar usuário

Fornece opção de procurar e desativar o usuário como exibido na figura 10. Nessa opção o usuário será movido para um local específico de usuários desativados. Além de remover todos os grupos e listas de e-mail ao qual participa.

Figura 10– Página de desativar usuário

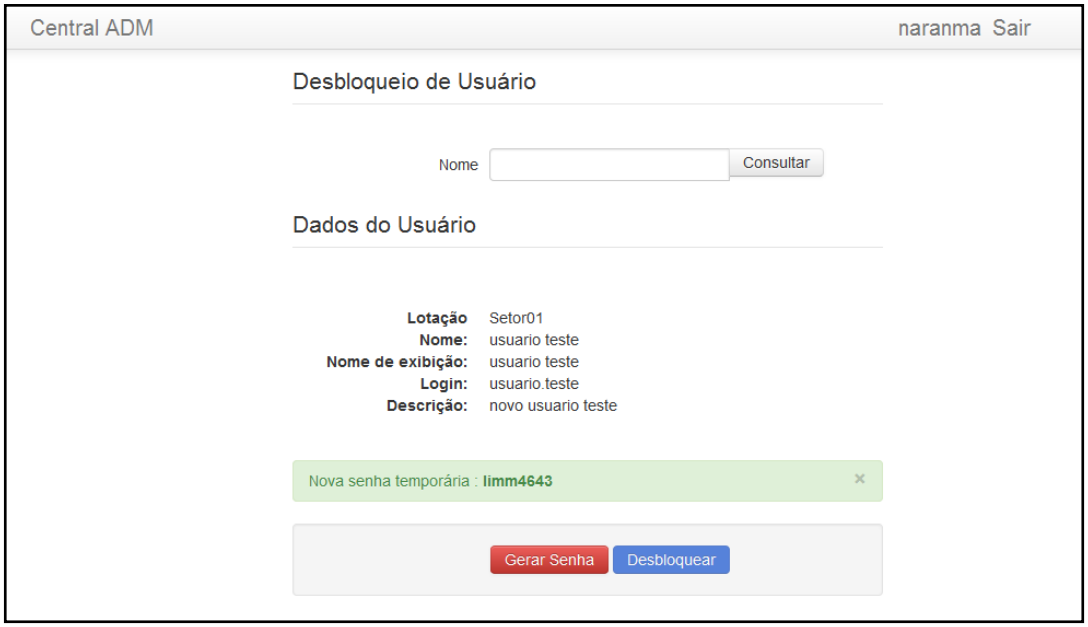


Fonte: Elaborado pelo autor

5.3.4 Desbloqueio e alteração de senha

A figura 11 exibe o módulo que fornece a opção de desbloqueio do usuário, caso esteja bloqueado e ou alteração da senha. A senha será gerada aleatória com letras e números de forma temporária, dando ao usuário após o primeiro acesso, uma forma de modificar.

Figura 11– Página de desbloqueio de usuário

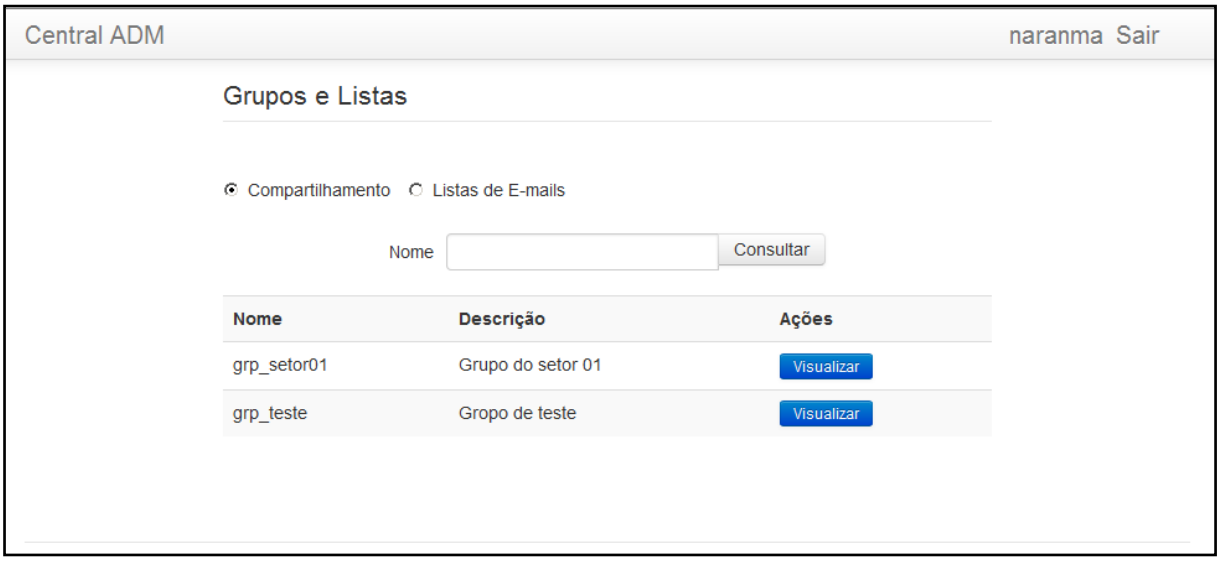


Fonte: Elaborado pelo autor

5.3.5 Visualização de listas e grupos

Permite o usuário operador pesquisar e visualizar os membros das listas de distribuição e grupos de segurança, como mostra a figura 12.

Figura 12– Página de visualizar grupos e listas

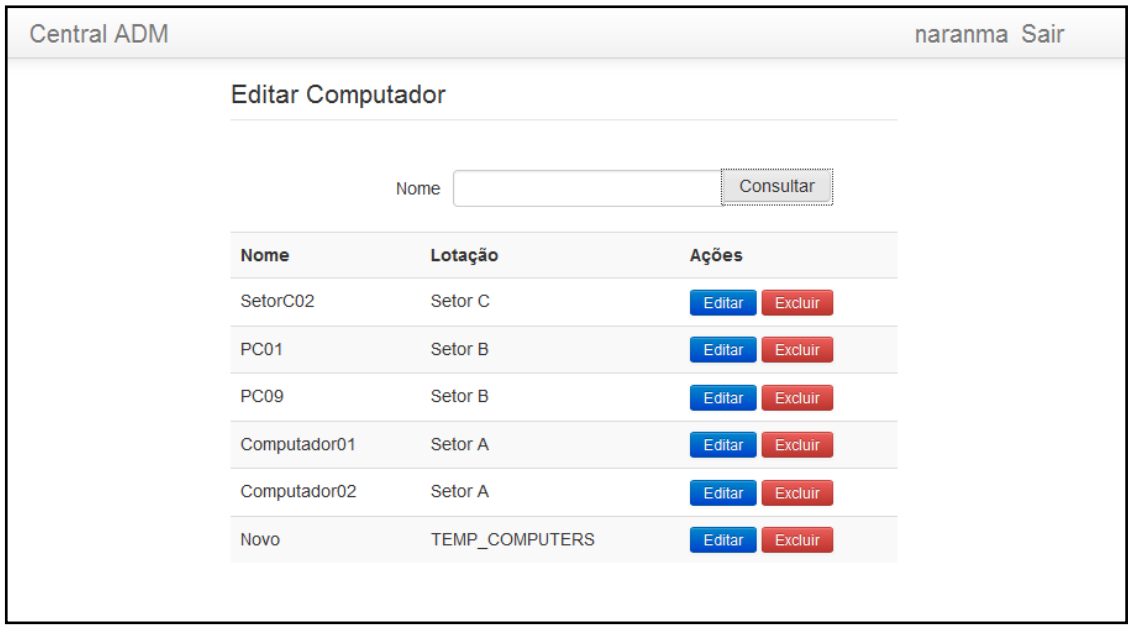


Fonte: Elaborado pelo autor

5.3.6 Edição de computadores

Fornece a opção de mover e apagar os computadores criados no servidor de diretório. A figura 13 demonstra essas opções.

Figura 13– Página de editar computador

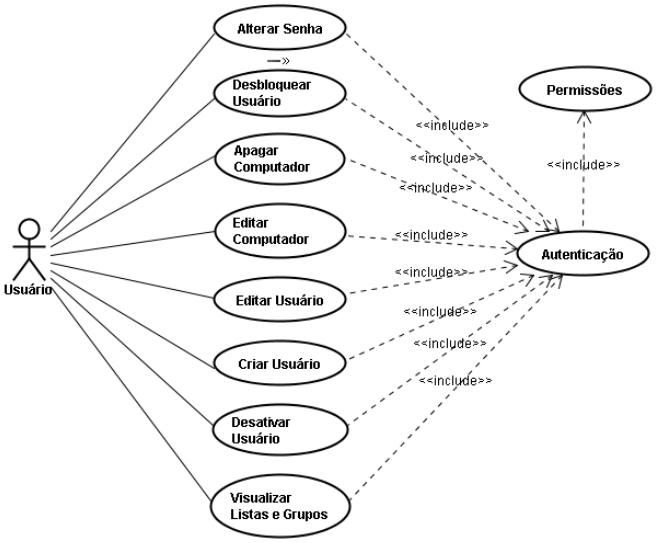


Fonte: Elaborado pelo autor

5.4 Diagrama de caso de uso

A figura 14 apresenta o diagrama de Caso de Uso do sistema. Este diagrama possui apenas um ator que é o usuário, visto que as operações que o usuário pode fazer no sistema depende das permissões que estão liberadas.

Figura 14–Diagrama de caso de uso do sistema



Fonte: Elaborado pelo autor

RESULTADOS

6 RESULTADOS

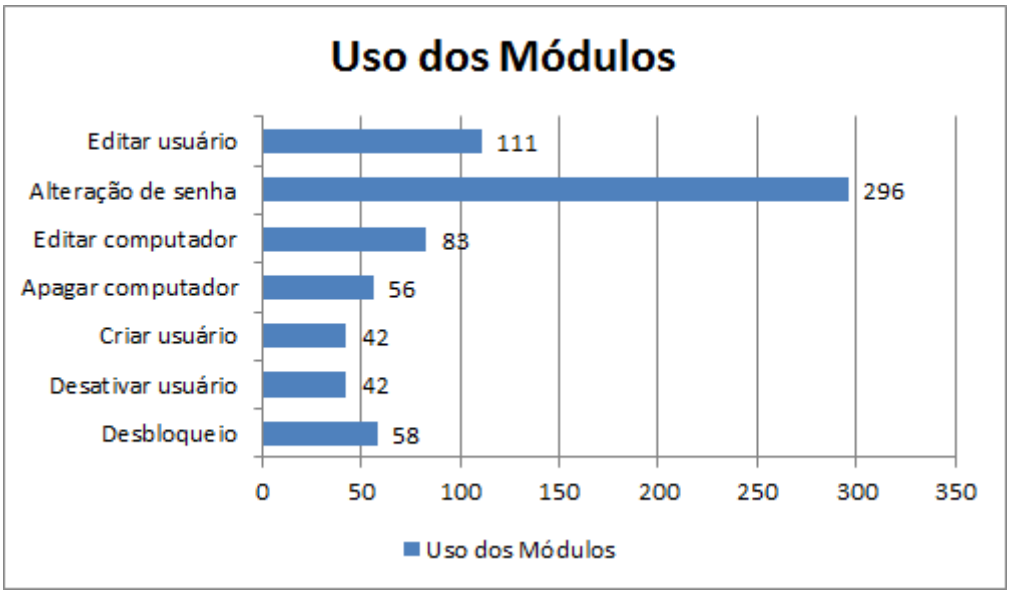
As operações no gerenciamento de usuários, computadores e outros recursos são atividades rotineiras e frequentes em uma organização. No Tribunal Regional Eleitoral do Tocantins as tarefas relacionadas ao servidor de diretório eram feitas por um setor. Com a implementação do sistema, outros setores puderam trabalhar no servidor de forma mais restrita. Um dos setores é o responsável pelo atendimento ao usuário, onde após a liberação do sistema poderão realizar tarefas sem a necessidade de repassar para o setor responsável pelo serviço de diretório LDAP.

A implantação do sistema Web reduziu o tempo de operações realizadas no gerenciamento de objetos no servidor de diretórios e aumentou o número de usuários devido seu maior controle de permissões, já que cada usuário pode apenas acessar os módulos liberados. O sistema força os usuários a ver apenas o que lhe foi permitido, assim, mesmo com pouco conhecimento sobre o serviço de diretórios LDAP, podem operá-lo sem trazer riscos à segurança.

A interface da aplicação Web é simples e de fácil uso com grande aceitação pelos usuários. Possui validação de formulário para evitar despadronização dos dados inseridos no serviço de diretório. Pelo fato do sistema ser simples, é rápido a aprendizagem dos usuários operadores, em pouco tempo já podem operar o sistema de forma produtiva.

O gráfico da figura 15 mostra o uso dos módulos do sistema durante um período de quatro meses entre o mês de junho e setembro de 2014, estes dados foram baseados nos *logs* gerado pelo sistema para cada ação de modificação realizada.

Figura 15– Uso dos módulos do sistema em quatro meses baseados nos *logs*



Fonte: Elaborado pelo autor

CONSIDERAÇÕES FINAIS

7 CONSIDERAÇÕES FINAIS

A implementação do sistema permitiu a criação de uma ferramenta que trouxe bons resultados e aceitação. Permitiu também adquirir novos conhecimentos sobre as ferramentas e tecnologias utilizadas no processo de criação do sistema. Bem como, utilizar dos conhecimentos adquiridos sobre linguagens de programação orientada a objetos, engenharia de software e aplicações Web.

Apesar de a aplicação Web ter sido projetada utilizando as linguagens C# na plataforma ASP.NET *Framework*, nada impede que seja construída em outras linguagens, por exemplo, Java, PHP e Python. Existem várias bibliotecas para acesso ao serviço de diretórios utilizando o protocolo LDAP, assim cabe analisar o ambiente verificando em qual linguagem será mais viável.

Para trabalhos futuros o sistema poderá se associar com outros serviços, como, e-mail, tanto para envio de notificações como criação de contas, visto que alguns serviços utilizam os objetos de usuários no LDAP para autenticação. Outro sistema que poderá futuramente ser integrado, é o sistema utilizado na gestão de pessoas, devido trabalharem diretamente com novos colaboradores na empresa. Com isso seu cadastro poderia ser unificado evitando redundância de cadastros.

Poderão ser implementadas novas funcionalidades como gerar relatórios e estatísticas sobre os dados contidos no servidor de diretório. Outra funcionalidade seria administrar via Web o arquivo de configuração da aplicação, permitindo a gerência das configurações do sistema bem como gerir os usuários e suas permissões.

A linguagem utilizada trouxe facilidades na implementação, por possuir uma documentação elaborada com muitos detalhes, normalmente acompanhada de exemplos de utilização. Outro fator facilitador, é a quantidade de bibliotecas que fazem parte da plataforma .NET *Framework*, que permite de forma fácil acessar recursos do sistema e serviços.

Portanto, a junção de todas as tecnologias, ferramentas e conhecimento proporcionaram a criação de um sistema funcional, tornando mais ágeis as tarefas realizadas no servidor, além de permitir que mais usuários trabalhem de forma segura no gerenciamento do servidor de diretório LDAP.

REFERÊNCIAS

CARTER, Gerald. **LDAP Administração de sistemas**. 2009.

COULOURIS, George; DOLLIMORE, Jean; KINDBERG, Tim.**Sistemas distribuídos: conceitos e projeto**. Grupo A, 2007.14

DE PAULA GIL, Anahuac. **OpenLDAP Extreme**. Brasport.

DESMOND, B.; RICHARDS, J.; ALLEN, R.; LOWE-NORRIS, A. G. **Active Directory, Designing, Deploying, and Running Active Directory**.5. ed.Sebastopol: O'Reilly, 2013.

DOS SANTOS, Alfredo Luiz. **Integração de Sistemas com Java**. Brasport, 2007.

DURAES, RAMON et al. **Desenvolvendo para web usando o Visual Studio 2008**. Brasport, 2008.

GAIKAIWARI, Amol J. **A Web-Based Corporate Directory Application Using LDAP (V3)(RFC 2251)**. 2005. Tese de Doutorado. Universityof Illinois Springfield.

ISQUIERDO, Gustavo Scalco. **Integração do Serviço de Diretorio LDAP com o Serviço de Nomes CORBA**. 2001. Tese de Doutorado. Universidade de São Paulo.

LOTAR, Alfredo. **Como Programar com ASP. NET e C#-2ª Edição: Dicas, truques, exemplos, códigos, conceitos, tutoriais**. Novatec Editora, 2010.

Micrososft Developer Network. **Introdução à linguagem C# e ao .NET Framework**. Disponível em <http://msdn.microsoft.com/pt-br/library/z1zx9t92.aspx>. Acesso em: 08/07/2014.

Micrososft Developer Network. **Visual C#**. Disponível em <http://msdn.microsoft.com/pt-br/library/kx37x362.aspx>. Acesso em: 08/07/2014.

Micrososft Developer Network. **Namespace System.DirectoryServices**. Disponível em <http://msdn.microsoft.com/pt-br/library/System.DirectoryServices(v=vs.110).aspx>. Acesso em: 08/08/2014

MOTA, ADRIANO PINHEIRO. INTEGRANDO LDAP COM SAMBA PARA UTILIZAÇÃO COMO SOLUÇÃO DE PDC NA REDE. **Monografia (Curso de Pós-Graduação "Lato Sensu" em Administração de Redes Linux), Universidade Federal de Lavras-LAVRA, Minas Gerais–Brasil**, 2008.

SANTANA FILHO, OZEAS VIEIRA; ZARA, Pedro Marcelo. **Microsoft. net: uma Visao Geral Para Programadores**. Senac, 2002.

TUTTLE, Steven et al. **Understanding LDAP-design and implementation**. IBM Redbooks, 2006.