



UNIVERSIDADE LUSÓFONA
de Humanidades e Tecnologias
Humani nihil alienum

Licenciatura em Engenharia Informática

Trabalho Final de Curso

Relatório

Drag And Drop

Orientador: José Faísca

Setembro 2012

Mário Antunes nº:20083869

Rodolfo Antunes nº: 20094017

Índice de Conteúdos

Índice de ilustrações	3
Introdução	4
Resumo.....	5
Abstract	6
Planeamento o enquadramento	7
.NET Framework	7
Desenho lógico do sistema	8
Arquitectura do sistema	8
Drag and Drop	9
O cliente Drag And Drop	9
O Servidor Drag And Drop	10
Chat	10
Arquitectura de conexão.....	11
Arquitectura mensagens.	11
Exemplo de envio de uma mensagem através da Internet.	12
Estrutura da Solução.....	12
Use Case	13
Lista de Requisitos	14
Requisitos funcionais.....	14
Requisitos não funcionais.....	15
Diagramas.....	15
Diagrama de sequência Login.....	16
Diagrama de sequência Recepção lista utilizadores Chat	17
Diagrama de Base de Dados.....	18
Tecnologias e sua Aplicação no Projecto.....	19
Windows Communication Foundation.....	19
Entity Data Model	20
LINQ to SQL	20
Universal Plug and Play (UPnP)	20
Internet Information Services (IIS)	20
SSL (Secure Socket Layer)	21
Autenticação mutua SSL.....	21
SevenZipSharp e 7-zip	22
MD5.....	22
Registo do Windows.....	22
Microsoft SQL Server 2012	22
Certificados Digitais.....	23
Considerações finais e perspectivas	24
Referências	25
Glossário.....	26
Netografia	27

Índice de ilustrações

Ilustração 1 – Arquitectura do sistema	8
Ilustração 2 – Arquitectura de Conexão Chat Server	11
Ilustração 3 – Envio Mensagem através da internet.	12
Ilustração 4 – Use Case.....	14
Ilustração 5 – Diagrama de Sequência Login	16
Ilustração 6 – Diagrama de Sequência Recepção de lista de utilizadores Chat	17
Ilustração 7 – Diagrama de Base de Dados	18
Ilustração 8 – Autenticação Mutua SSL	21

Introdução

Com o desenvolvimento das tecnologias e o aumento da competitividade entre as empresas e entre instituições de ensino vão sendo necessárias novas ferramentas para aumentar a produtividade das empresas e das instituições de ensino.

Tendo em conta estes factores decidimos criar uma ferramenta para transferir ficheiros o mais prática possível, o conceito desta aplicação situa-se no “Drag and Drop” (Pegar e largar) de ficheiro entre computadores.

Este simples conceito não nos trouxe simplicidade, tivemos que criar todo um sistema seguro para que a privacidade dos utilizadores não fosse comprometida, com a camada de encriptação Secure Socket Layer (SSL [3]), criamos ainda uma limitação ao acesso ao sistema por login para que só as pessoas credenciadas o pudessem usar.

Esta aplicação foi concretizada devido aos conhecimentos adquiridos e a alguma investigação necessária.

Resumo

Nestes últimos anos a tecnologia tem dado um grande salto, e com a tecnologia tem sido possível facilitar a vida das pessoas que se rodeiam sobre ela.

É importante que a tecnologia seja uma coisa boa e não uma coisa complexa que por vezes se poderá tornar chato.

Sentimos que por vezes torna-se difícil a partilha rápida de ficheiros por diversos utilizadores, tendo que recorrer a pens, que requer a deslocação a cada sítio onde o ficheiro tem que ser copiado, através da rede, em pastas partilhadas, que não possibilita a partilha através da internet e por vezes requer login.

Com esta situação decidimos criar uma aplicação o mais automática possível, isto é, uma aplicação capaz de receber ficheiros através da rede local e/ou através da internet. É possível configurar a aplicação para receber ficheiros automaticamente, reproduzi-los automaticamente e até a sua extracção se o utilizador assim quiser.

Esta aplicação faz a configuração dos portos automaticamente num router compatível com UPnP, novamente vindo a facilitar ao utilizador configurações que se tornam complexas.

Uma das soluções que também decidimos implementar com esta aplicação é um chat para comunicação entre todos os utilizadores, por vezes a comunicação rápida é crucial para certas questões que tem que ser resolvidas no momento, não sendo assim necessário ter que se deslocar ao local onde a pessoas se encontram, falando com ela de forma comoda e rápida.

Para a realização deste projecto foram utilizadas varias tecnologias. Para a criação geral das aplicações foi utilizado o C# em visual Studio 2010.

Para a base de dados foi utilizado o SQL Server Express 2012. Utilizamos o WCF para fazer comunicação entre a aplicação e a Base de Dados.

O login poderá ser feito através da base de dados, ou através de um servidor LDAP.

Abstract

In recent years technology has taken a big step, and the technology has been able to facilitate the lives of the people who surround with it.

It is important that the technology is a good thing and not a complex thing that sometimes can become boring.

We feel that sometimes it becomes difficult to rapid sharing of files by multiple users having to resort to usb pens, which requires moving each site where the file must be copied by the network in shared folders, which does not allow the sharing over the internet and sometimes requires authentication.

With this situation we decided to create an application as automatic as possible, ie, an application that can receive files over the local network and / or over the internet. You can configure the application to automatically receive files, and play them back automatically to its extraction if the user so desires.

This application makes the configuration of ports on a compatible router automatically with UPnP, coming again to facilitate the user settings that sometimes become complex. One of the solutions we also decided to implement in this application is a chat for communication between all users, sometimes rapid communication is crucial for certain issues that must be resolved at the right time and thus not need to go to the site where people are, talking with people fast and easy.

For this project we used various technologies. To create general applications was used C# in Visual Studio 2010.

For database was used SQL Server Express 2012. We use WCF to communicate between application and database.

The login may be done by database or via an LDAP server.

Planeamento o enquadramento

O projecto consiste num sistema de transferência de ficheiros intuitivo e de fácil utilização.

Este sistema é baseado no modelo Cliente/Servidor, onde o cliente recebe ficheiro e o servidor envia.

O servidor tem a capacidade de enviar para vários clientes em simultâneo (conceito de Grupo) ou para um cliente individual.

O cliente não se limita apenas a receber os ficheiros, ele faz toda a gestão da recepção dos ficheiros tratando-os conforme o seu tipo.

.NET Framework



Para a realização deste projecto é utilizada a plataforma (Framework). NET que é uma plataforma da Microsoft para desenvolvimento de aplicações de vários tipos tanto Web como Stand-alone, o seu ambiente de desenvolvimento é o Visual Studio, aonde foi criado todo o projecto.

Escolhe-mos este ambiente porque se ajusta na perfeição com os requisitos propostos neste projecto e facilita no design da aplicação.

A **Microsoft .NET** é uma iniciativa da Microsoft para criação de uma plataforma única para desenvolvimento, execução de sistemas e aplicações.

Todo e qualquer código gerado para o .NET pode ser executado em qualquer dispositivo que possua a framework instalada.

Esta plataforma é executada sobre uma Common Language Runtime - CLR (Ambiente de Execução Independente da Linguagem) interagindo com um conjunto de bibliotecas unificadas (framework). Esta CLR é capaz de executar, actualmente, mais de 33 diferentes linguagens de programação, interagindo entre si como se fossem uma única linguagem.

De tantas linguagens nos decidimos utilizar o C#.

Desenho lógico do sistema

Arquitectura do sistema

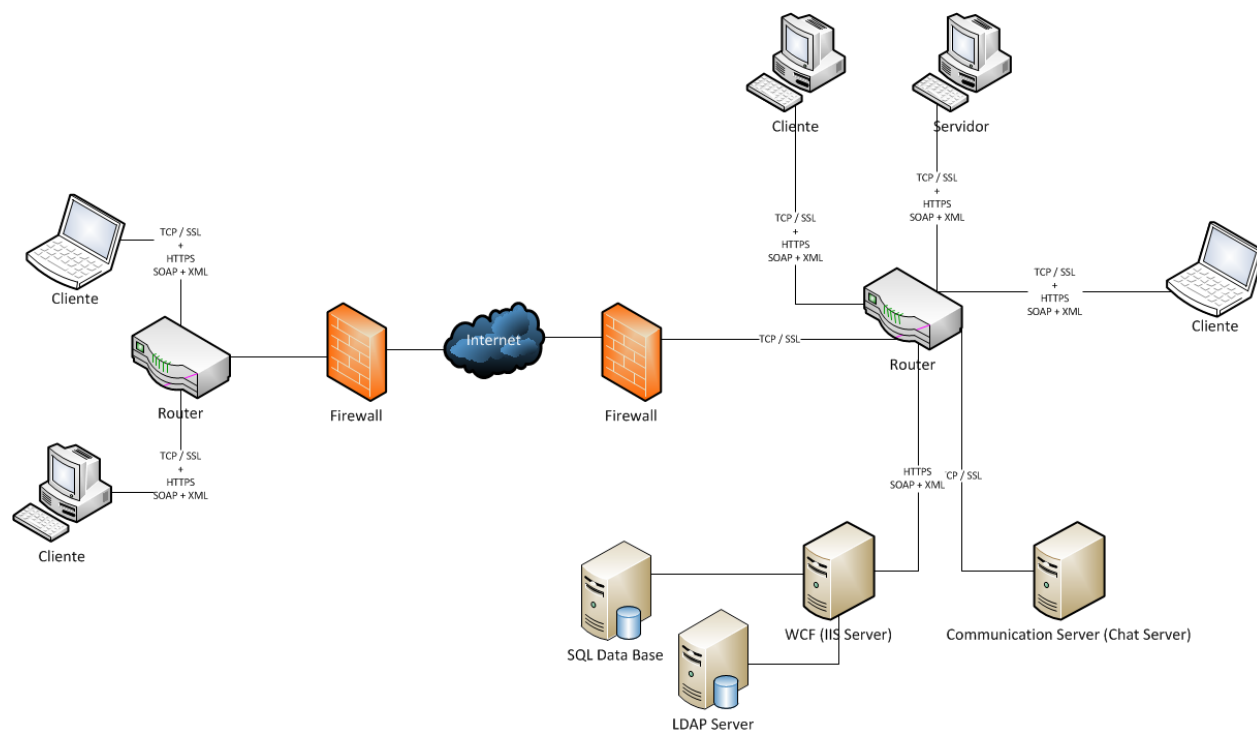


Ilustração 1 – Arquitectura do sistema

Com o diagrama acima representado reparamos que o projecto dispõe de vários serviços, tais quais o WCF, SQL Data Base, LDAP Server e o Communication Server (Chat Server), o servidor IIS onde está alocado o WCF, temos todos os serviços de consulta de dados ao SQL Server e consulta de Utilizadores ao LDAP Server.

O Communication Server (Chat Server) serve para receber e reencaminhar todas as mensagens trocadas entre os utilizadores.

Todos os “Cliente” denominados no diagrama terão acesso ao lado do cliente da aplicação, onde dispõem do serviço de recepção de ficheiros e serviço de chat.

Todos os “Servidor” denominados no diagrama terão acesso ao lado do servidor que dispõe de serviços de envio de ficheiros, chat e de toda a gestão dos utilizadores.

Cada **Cliente** dispõe das seguintes ligações:

- **TCP / SSL** - Para cada ficheiro recebido e uma permanente com o Chat Server para envio e recepção de mensagens.
- **HTTPS (SOAP / XML)** - Para pedidos ao WCF para informações de autenticação, recuperação palavra passe, definição de localização.

Cada **Servidor** dispõe das seguintes ligações:

- **TCP / SSL** - Para cada ficheiro enviado e uma permanente com o Chat Server para envio e recepção de mensagens.
- **HTTPS (SOAP / XML)** - Para pedidos ao WFC para informações de autenticação, recuperação palavra passe, definição de localização, gestão de utilizadores e gestão de grupos.

Drag and Drop

O cliente Drag And Drop

O cliente tem o papel de receber os ficheiros enviados pelo servidor. Quando este é iniciado começa por abrir as portas necessárias para poder receber ligações do servidor. Após isso é iniciada uma socket que aceita ligações, em cada ligação é feita uma autenticação mutua entre o cliente e o servidor.

Entretanto é necessário também fazer a autenticação através de um login de credenciais para prosseguir, caso o utilizador tenha perdido ou esquecido a palavra-passe, é possível recupera-la através do assistente.

Após feita a autenticação é enviado do servidor para o cliente a informação relativa ao ficheiro, que inclui, o tamanho, nome, MD5 e o emissor. Com a informação do ficheiro já é possível criar graficamente um User control com a informação do ficheiro para o cliente decidir se aceita o ficheiro.

Caso o cliente não aceite o ficheiro, a transferência não é realizada e a ligação termina.

Caso o cliente aceite o ficheiro, esse ficheiro começa a ser transferido e ao menos tempo a ser gravado no disco rígido, no caso de existir um ficheiro com o mesmo nome, a aplicação vai adicionar ao nome do ficheiro um número que começa do 1 e vai incrementando até que não exista um nome igual.

Ao finalizar a transferência é feita uma verificação do MD5 do ficheiro recebido com o que esta na informação do ficheiro recebida anteriormente, se forem iguais, a transferência foi feita com sucesso.

De seguida é verificado se o ficheiro é uma das extensões suportadas pelo sistema de extracção, se for, é perguntado ao utilizador deseja extrair o ficheiro, se sim, é pedido a localização onde deseja extrair, após a extracção é aberta uma janela com os ficheiros.

Todo este acontecer é acompanhado pela UI da Aplicação.

O Servidor Drag And Drop

O servidor tem o papel de enviar ficheiros e fazer a gestão dos utilizadores e grupos. Para efectuarmos uma transferência temos que seleccionar nas configurações os utilizadores ou grupos que vamos associar a cada extremidade do ecrã.

Após fazermos as configurações, basta arrastar um ficheiro para um dos cantos do ecrã e largar, logo de seguida é feita uma tentativa de se ligar ao utilizador, no caso do grupo é efectuada uma ligação a cada um dos utilizadores do grupo, se a ligação for efectuada com sucesso, é feita uma autenticação mutua entre o cliente e o servidor.

Entretanto é gerada a informação relativamente ao ficheiro e é enviada ao cliente.

Quando o utilizador no lado do cliente aceitar o ficheiro, a transferência é efectuada, após estar concluída, o servidor aguarda resposta do lado do cliente para ter informação se o ficheiro foi recebido com sucesso.

Este processo é totalmente acompanhado no UI da Aplicação para o cliente ter conhecimento do que está a acontecer.

No servidor é também possível adicionar, remover e alterar utilizadores e grupos e gerir a relação entre os utilizadores e os grupos. A relação entre utilizadores e grupos é bastante simples e fácil de usar.

Chat

Esta componente do projecto disponibiliza ao utilizador um formulário com a lista de utilizadores disponíveis, e poderá iniciar uma conversa privada com cada um dos utilizadores que estejam com a aplicação iniciada e com a respectiva autenticação efectuada.

Essa informação é transmitida ao utilizador através de um ícone (redondo) que cada utilizador tem, verde para disponível para conversação e vermelho caso não esteja disponível.

As mensagens entre os utilizadores serão apenas do formato de texto.

Cada vez que um utilizador entra em contacto com outro, o utilizador receptor é alertado com um som e é iniciada uma nova janela em Modo Flash.

Arquitectura de conexão

Todos os clientes estão conectados com o Communication server (CS) através de uma ligação segura (SSL), estas ligações são guardadas no CS para que cada mensagem que chegue ao CS seja devidamente redireccionada para o seu destino, a vantagem desta arquitectura está na não utilização de uma ligação P2P entre todos os utilizadores, ou seja, desta forma temos apenas uma conexão por cliente.

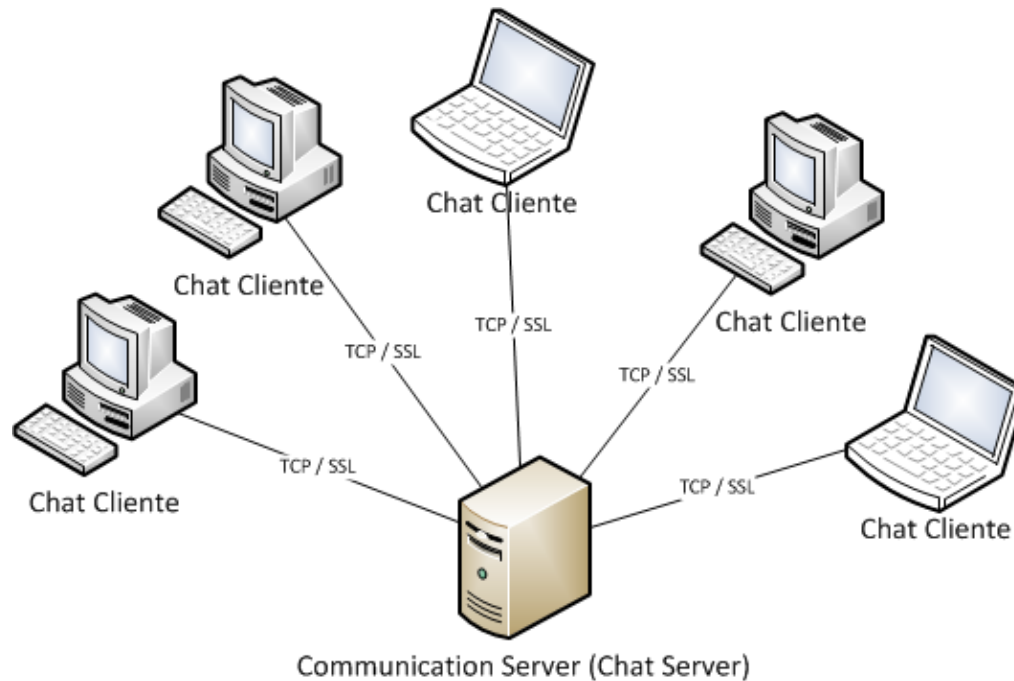


Ilustração 2 – Arquitectura de Conexão Chat Server

Arquitectura mensagens.

String Message - Texto com o conteúdo da mensagem

String Source - Nome do utilizador da origem da mensagem

String Destination - Nome do utilizador do destino da mensagem

int DestinationID - ID do utilizador na Base de dados do destino da mensagem

int SourceID - ID do utilizador na Base de dados da origem da mensagem

bool isChatList - Caso seja necessário enviar a lista de utilizadores para o chat.

List<ChatPersonInfo> ChatList Lista de utilizadores, esta lista só é preenchida quando a lista é pedida, ou seja, nas mensagens trocadas entre os utilizadores esta lista está vazia.

Exemplo de envio de uma mensagem através da Internet.

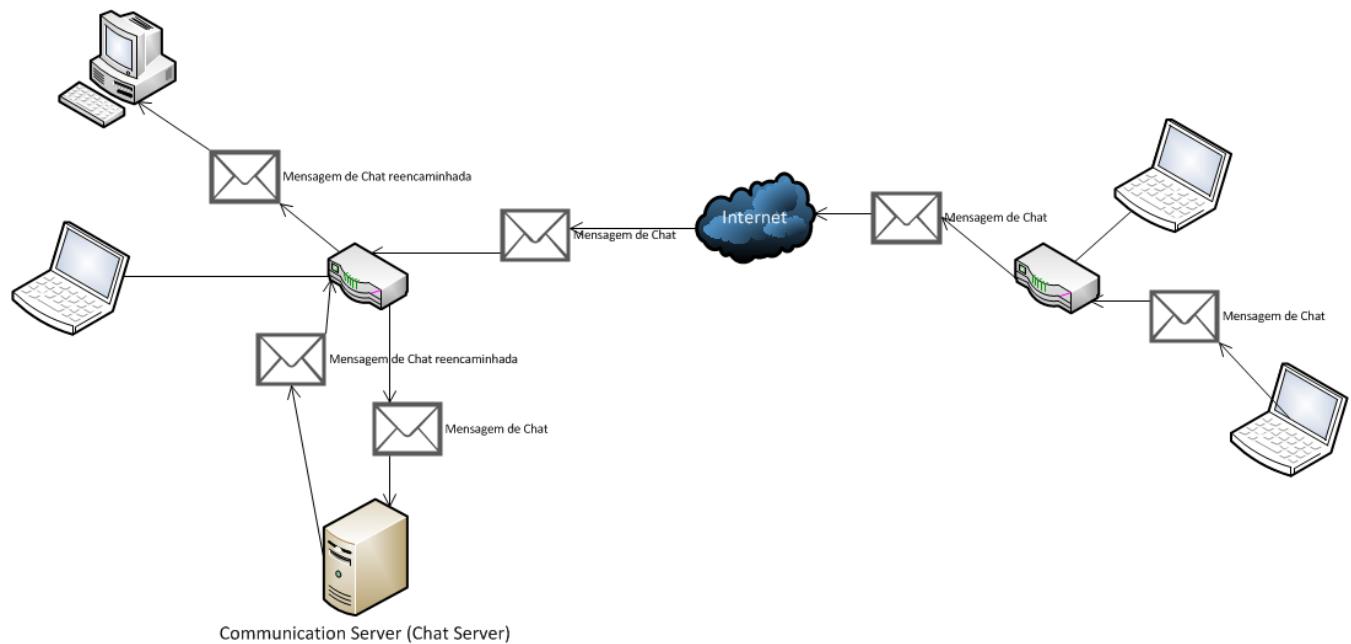


Ilustração 3 – Envio Mensagem através da internet.

Estrutura da Solução

A Solução desenvolvida é constituída por 6 projectos:

O **ChatClient** – que é onde o utilizador interage no chat, este projecto apenas comunica com o ChatServer

O **ChatServer** – que é o Servidor do chat que faz a gestão de todas a mensagens enviadas entre os utilizadores.

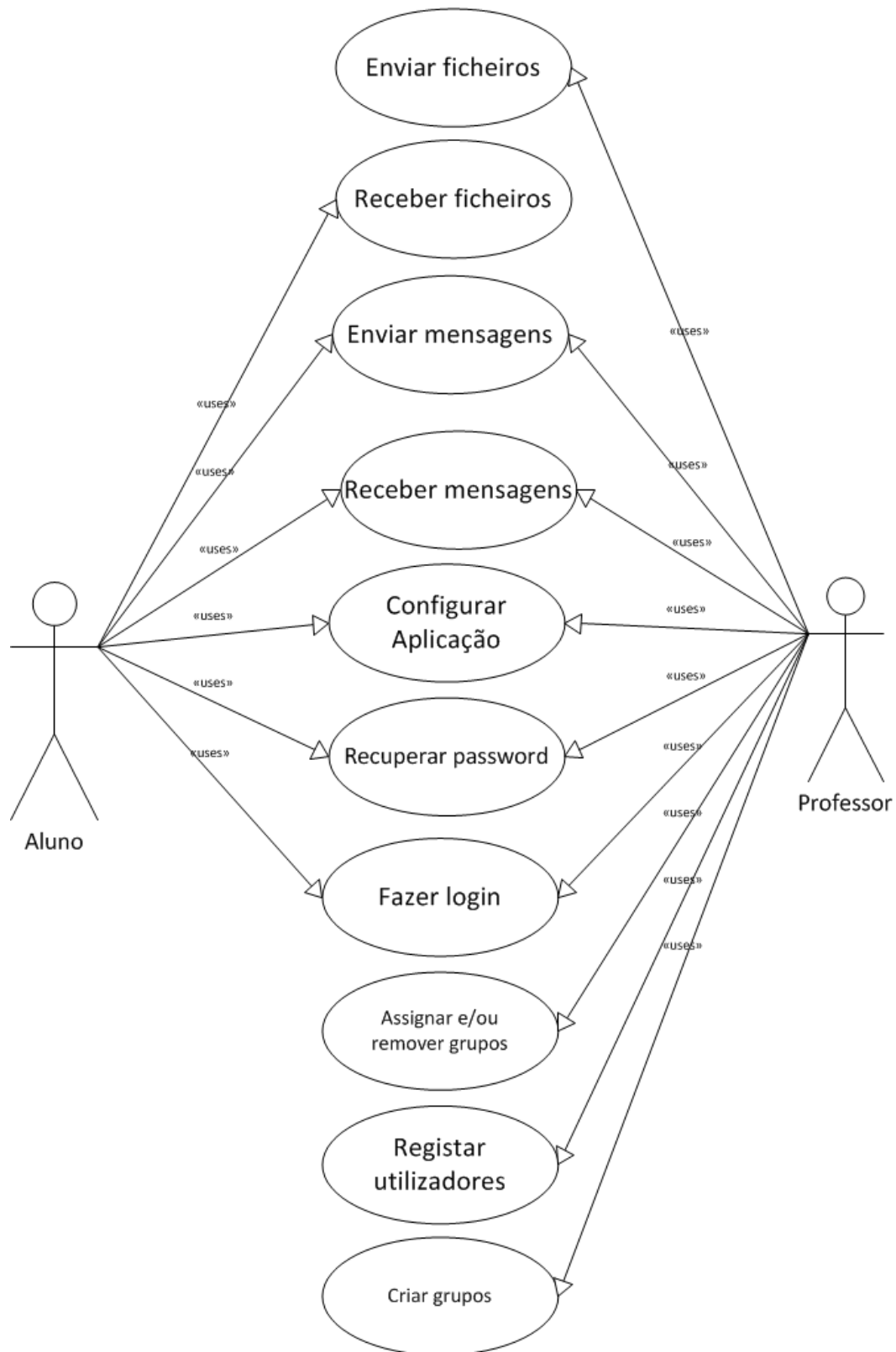
O **Cliente** – o cliente da transferência de ficheiros, ou seja, o que recebe os ficheiros.

O **Geral** – uma biblioteca com vários recursos que é utilizado nos outros projectos.

O **Servidor** – o Servidor que é responsável pelo envio dos ficheiros para a aplicação Cliente e gestão dos clientes e grupos.

O **WCFSservice** – o serviço que alimenta todos os projectos excepto o ChatClient. Este que faz a interligação entre a BD e as aplicações.

Use Case



Lista de Requisitos

A proposta inicial incidia apenas na simples transferência de ficheiros de um modo simplificado, mas como reparamos que iria ser uma aplicação pequena e não suficiente para o âmbito do projecto, decidimos adicionar funcionalidades, para que os requisitos exigidos fossem cumpridos.

Requisitos funcionais

- C#.NET
 - ✓ Início de sessão na aplicação
 - ✓ Ligações seguras com SSL
 - ✓ Utilização de Certificados to tipo X.509 para encriptação das ligações SSL
 - ✓ Criação de grupos
 - ✓ Registo de utilizador
 - ✓ Envio de ficheiros por grupo (Internet e rede local)
 - ✓ Envio de ficheiros por utilizador (Internet e rede local)
 - ✓ Aceitar/Rejeitar ficheiros manualmente
 - ✓ Receber automaticamente ficheiros
 - ✓ Abrir ficheiros recebidos
 - ✓ Reproduzir ficheiros automaticamente
 - ✓ Guardar ficheiros e localização escolhida pelo utilizador
 - ✓ Abrir Portos automaticamente no router local
 - ✓ Iniciar aplicação com o arranque do Windows
 - ✓ Descompactar ficheiros recebidos do tipo, zip, rar, cab, gz, iso e tar.
 - ✓ Envio e recepção de mensagens através de um chat interno.
- WCF (Windows Communication Foundation)
 - ✓ Disponibilização de serviços de consulta de informação a Base de dados
- SQL Server
 - ✓ Criação e gestão da Bases de Dados
- LDAP Server
 - ✓ Gestão dos utilizadores existentes no repositório.

Requisitos não funcionais

Durante o desenvolvimento da aplicação fomos nos confrontando com desafios e tivemos que investigar e escolher as melhores e mais robustas opções para implementação das funcionalidades, dessa forma passamos a enumerar os problemas/solução para as funcionalidades da aplicação:

Segurança e integridade dos dados em todas as ligações:

- ✓ Utilização de HTTPS para os pedidos HTTP ao WCF
- ✓ Utilização de encriptação da camada de Transporte da ligação com o SSL 3.0

Gestão de dados:

- ✓ SQL Server 2012 - Neste serviço guardamos a toda a informação dos clientes.
- ✓ LDAP Server - Para que seja o mais compatível na sua instalação, suportamos que a nossa aplicação consulta no LDAP server local as contas disponíveis.

Diagramas

De seguida, serão apresentados alguns diagramas que representam alguns aspectos importantes do projecto, tal como o processo de autenticação, recepção de utilizadores disponíveis para chat,

Gestão de entidade.

Todos os utilizadores para terem acesso a aplicação, terão que fazer uma autenticação prévia, para tal existe um formulário disponível quando a aplicação inicia, após tentativa de autenticação, inicia-se o processo demonstrado pelo diagrama de sequência abaixo referido.

Diagrama de sequência | Login

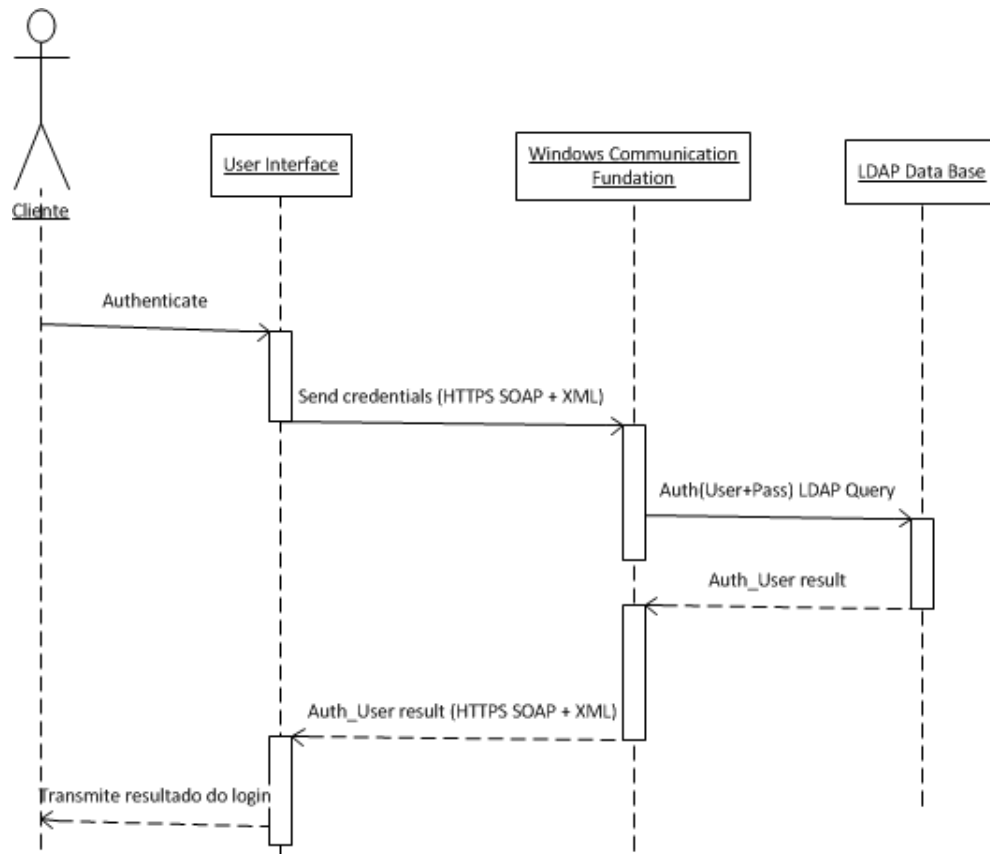


Ilustração 5 – Diagrama de Sequência | Login

Após uma autenticação com sucesso é necessário obter a lista de utilizadores disponíveis para o Chat, esse processo está demonstrado pelo diagrama de sequência abaixo referido.

Diagrama de sequência | Recepção lista utilizadores Chat

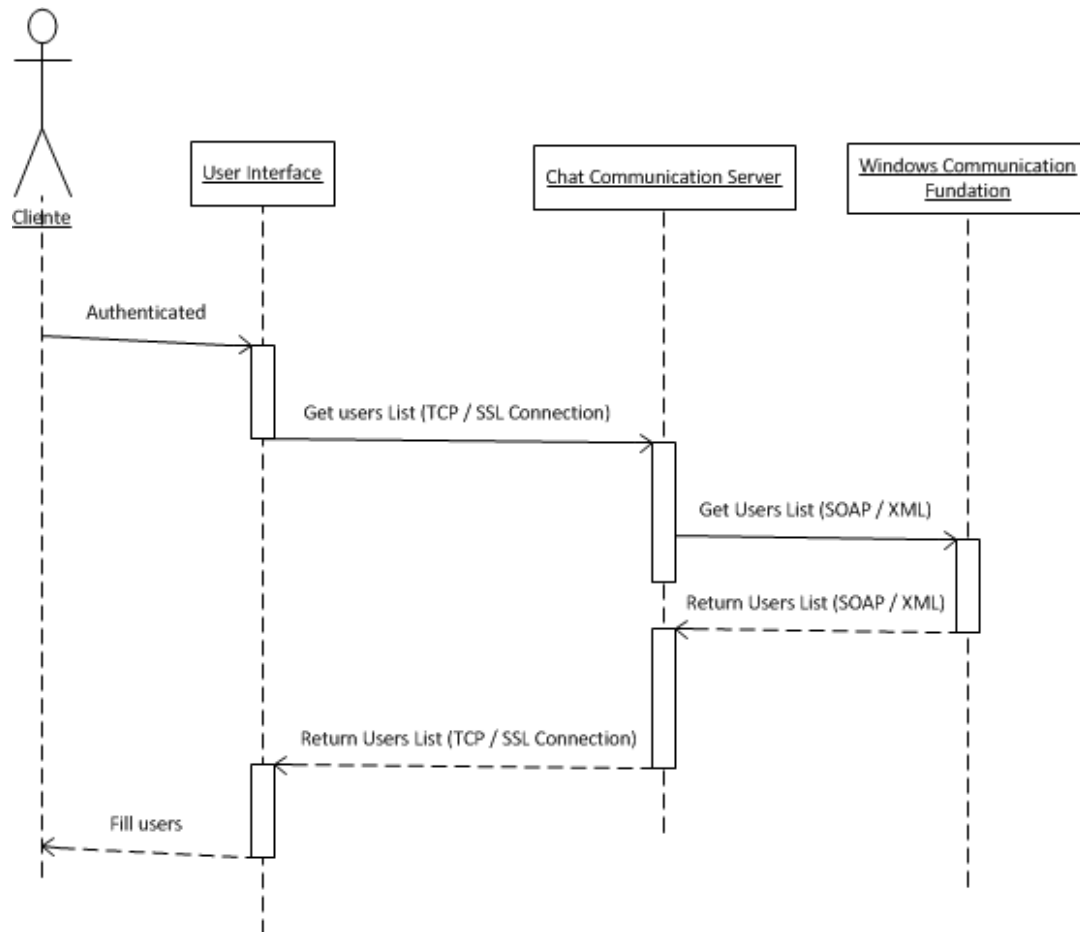


Ilustração 6 – Diagrama de Sequência | Recepção de lista de utilizadores Chat

Diagrama de Base de Dados

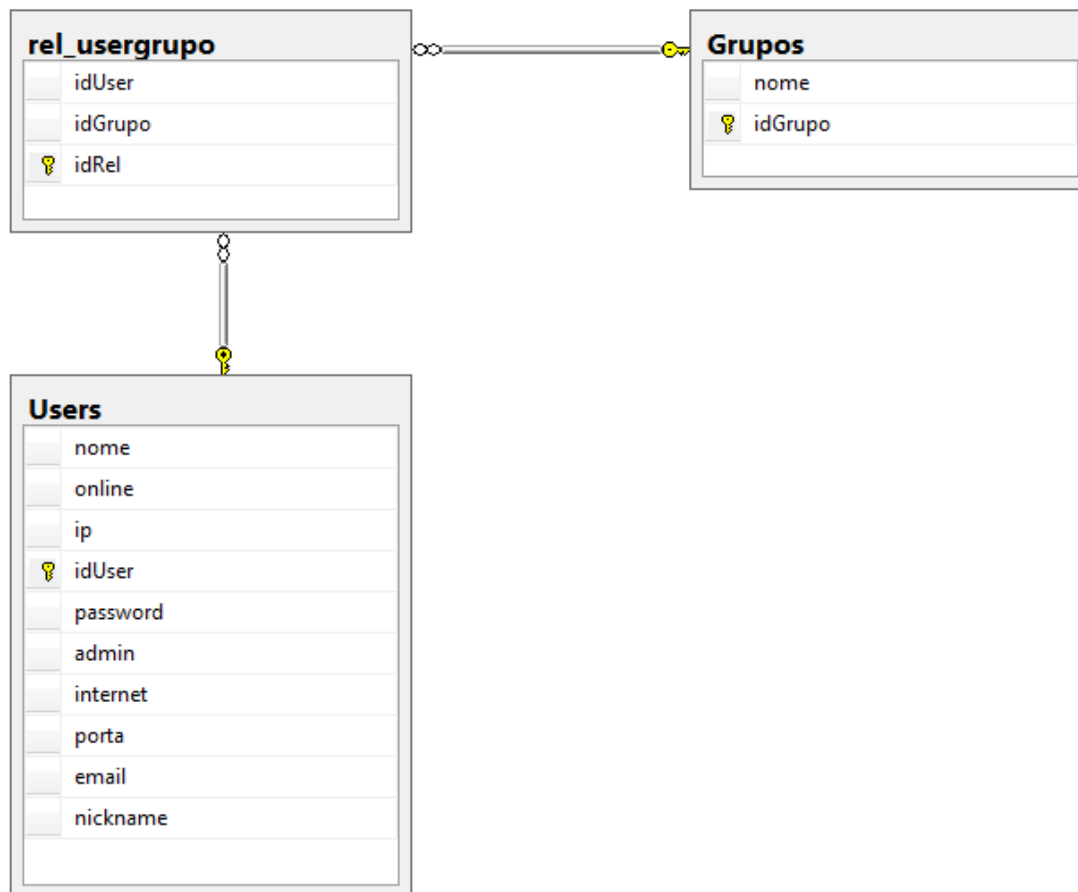


Ilustração 7 – Diagrama de Base de Dados

Tecnologias e sua Aplicação no Projecto

Windows Communication Foundation

O Windows Communication Foundation é uma framework para construção de aplicações orientadas a serviços.

Permite o envio de mensagens do serviço para o cliente, este serviço pode estar alocado no IIS ou pode estar alocado numa aplicação.

As mensagens podem ser um simples carácter, uma palavra enviada em XML ou dados complexos em formato binário.

As principais características são:

- Orientação ao serviço
- Interoperabilidade
- Múltiplos padrões de mensagens
- Serviço de metadados
- Contractos de dados
- Segurança
- Múltiplos transportes e codificações
- Fiável e filas de espera
- Transacções
- Suporte de AJAX e REST
- Extensibilidade

Dadas estas características, foi decidido apostar neste tipo de serviço para alimentar a nossa aplicação.

Instalámos o serviço num servidor IIS com AppFabric para que este serviço seja também visível na internet e permite que obtenhamos um relatório da actividade do servidor.

Assim permite que haja uma actualização em tempo real da situação e localização de cada um dos utilizadores, através da actualização da base de dados.

Utilizamos também os mecanismos de segurança disponibilizados, através da encriptação da camada de transporte com o certificado criado anteriormente.

A facilidade de implementação do WCF também foi um dos motivos que levou a implementar este método de comunicação.

O WCF tem o papel de garantir a comunicação entre a Base de dados e a aplicação.

Entity Data Model

A EDM é um conjunto de conceitos que descreve a estrutura de uma base de dados.

Através da EDM é possível interagir com uma base de dados, sendo a ela que faz a ligação directa a Base de Dados SQL, escrevendo de forma eficiente sem sacrificar a escalabilidade e acesso e gravação dos dados.

Permite trazer até ao programador as entidades, permitindo uma maior facilidade na interacção com os dados, tornando a programação mais simples e eficiente.

LINQ to SQL

Para fazer a recolha da informação foi utilizado o LINQ to SQL que permite de uma forma facilitada e eficiente pesquisas na base de dados, abstraindo o SQL complexo, e dando uma interface simples e fácil de usar.

Universal Plug and Play (UPnP)

Este protocolo permite que dispositivos de rede, computadores, impressoras, internet gateways e dispositivos móveis que se descubram entre eles na presença de uma rede.

Decidimos utilizar este protocolo para que fosse possível facilitar os clientes na abertura de portas no routers/gateways que por vezes tornam-se complexas para utilizadores menos experientes.

Assim este protocolo descobre automaticamente os routers/gateways compatíveis com este protocolo e abre automaticamente uma porta para a aplicação.

Sem uma porta aberta não seria possível a aplicação servidor conectar com a aplicação cliente, dado que o cliente é que vai fazer ligação entre ambas as partes da aplicação.

Com este protocolo activo na aplicação, a aplicação automaticamente vai detectar o router/gateway compatível e abrir a porta definida pelo cliente, se o cliente alterar a porta, a aplicação terá que ser reiniciada, com isto ela fecha a porta antiga e abre a porta nova.

Internet Information Services (IIS)

É um servidor Web com capacidade de alocação de serviços e páginas Web dinâmicas.

Utiliza o ASP.NET, mas também permite a adição de módulos de terceiros.

É este servidor que utilizamos para alocar o nosso serviço do WCF para que seja possível aceder ao serviço através da internet.

No nosso serviço foi configurado para aceitar apenas ligações seguras HTTPS, mas sem requerer certificado no cliente.

A autenticação do cliente é efectuada directamente na aplicação.

SSL (Secure Socket Layer)

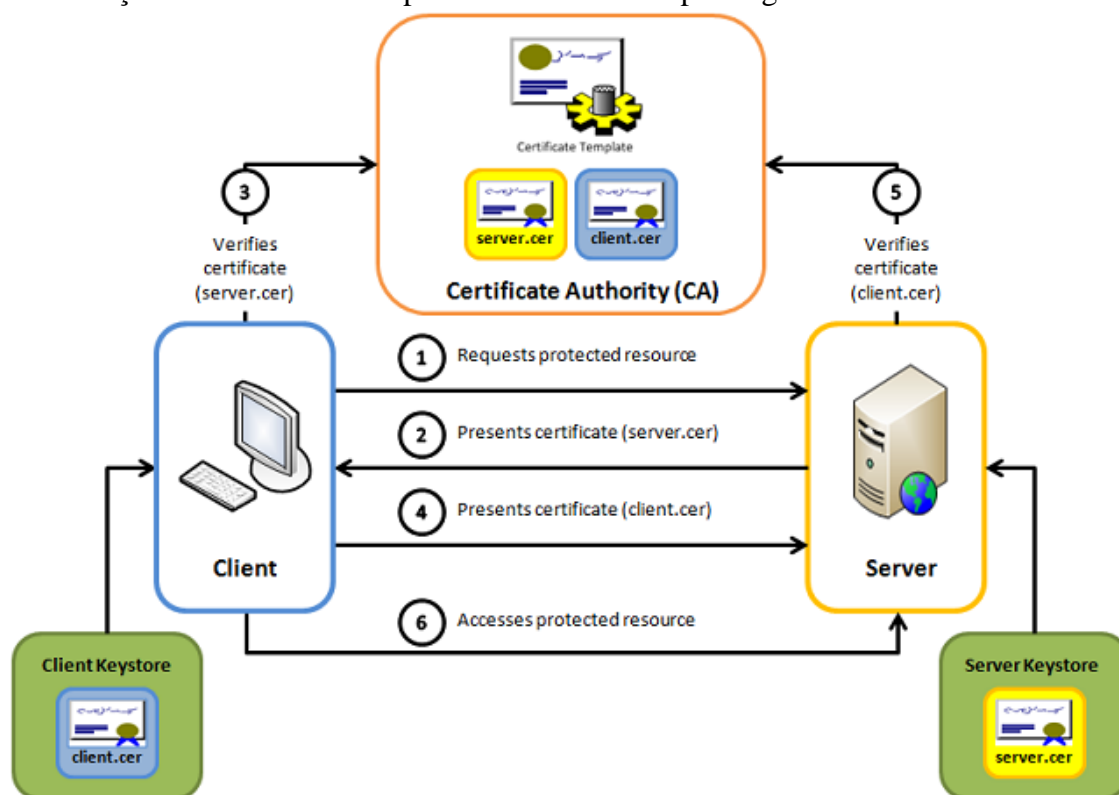
Camada de Sockets Segura - É um protocolo criptográfico que conferem segurança na comunicação na Internet, para vários serviços tais como transferência de ficheiros e troca de mensagens.

Este protocolo garante a privacidade e integridade dos dados entre duas aplicações que comuniquem pela Internet, isto ocorre através da autenticação das partes envolvidas e da encriptação dos dados transmitidos entre as partes, isto ajuda a que um 3º individuo não tenha acesso individuo ou falsifique dados transmitidos.

Autenticação mutua SSL

A autenticação mutua ou baseada em certificados refere-se a tentativa de duas partes se autenticarem através da verificação de certificados digitais, dessa forma ambas as partes estão protegidas contra entidades externas, em termos práticos refere-se a um cliente (Aplicação ou Web Browser) autenticando-se num servidor e servido autenticando-se no cliente através Chave pública (Certificado) e ambos serem acreditados pela Certificate Authority (CA)

O processo de autenticação e de criação do canal seguro usando certificados baseado na autenticação mútua envolve os passos demonstrados pela figura abaixo referida.



Mutual SSL authentication / Certificate based mutual authentication

Ilustração 8 – Autenticação Mutua SSL

SevenZipSharp e 7-zip

Uma das funcionalidades da aplicação é a de o utilizador ao receber o ficheiro ter a possibilidade de extrair automaticamente o ficheiro recebido.

Após a recepção com sucesso do ficheiro é questionado ao utilizador se deseja extrair o ficheiro e é-lhe pedido a localização onde quer que seja extraído.

Este processo apenas acontece para os ficheiros com extensão zip, rar, cab, gz, iso, tar.

Para ser efectuada a extracção foi utilizada uma biblioteca escrita em C# que envolve o 7-zip.dll, esta biblioteca chama-se SevenZipSharp.

Ambos o 7-zip.dll e o sevenzip.dll são bibliotecas Open Source.

MD5

O MD5 (Message-Digest algorithm 5) é um algoritmo de hash de 128 bits unidireccional que é geralmente utilizado para verificar a integridade de ficheiros.

Sendo assim utilizamos este algoritmo para verificar a integridade das transferências de ficheiros.

Antes do ficheiro ser enviado o transmissor obtém o MD5 do ficheiro. De seguida é enviado a informação referente ao ficheiro para o receptor onde ele guarda a informação do ficheiro que vai ser transferido. Após a transferência do ficheiro o receptor obtém o MD5 do ficheiro e compara-o com o MD5 enviado pelo transmissor. Se ambos os MD5 coincidirem o ficheiro foi transferido com sucesso.

Registo do Windows

Para facilitar a utilização da aplicação é possível o utilizador definir se prefere que a aplicação seja iniciada com o Windows.

Para resolvermos esta questão, utilizamos o registo do Windows onde registamos a aplicação para que ela inicie com o Windows.

Para desactivar apenas é removido esse registo do registo do Windows.

Microsoft SQL Server 2012

O Sql Server é um servidor de base de dados abrangentes, pronto para as cargas de trabalho mais exigentes com altos níveis de desempenho, disponibilidade e segurança.

Este é o servidor SQL que decidimos utilizar para armazenar a informação vital da nossa aplicação. Nesta base de dados é guardado a informação referente aos utilizadores, aos grupos e a relação entre eles.

Certificados Digitais

Um certificado digital é um ficheiro onde está contida um conjunto de informação referente a entidade para qual o certificado foi emitido, sendo este uma empresa, pessoa ou computador. Este certificado contém uma chave pública e uma chave privada.

Uma Autoridade de Certificação é uma autoridade que assina certificados digitais.

Para fazermos as ligações SSL e HTTPS tivemos que criar um certificado digital que ficou com o nome de Servidor, e uma Autoridade de Certificação com nome TFC Certification Authority.

Apos a criação de ambos foi assinado o certificado digital Servidor pela TFC Certification Authority. Estes certificados foram criados no OpenSSL do Linux.

Considerações finais e perspectivas

Com o projecto concluído e em relação aos requisitos propostos denotamos que ainda poderíamos adicionar muitas funcionalidades tais quais como streaming de vídeo e áudio ao enviar um ficheiro de vídeo ou de áudio, permitir bloquear o computador para que apenas o conteúdo transferido possa estar em execução, enviar comandos para o cliente como por exemplo encerrar o computador e enviar printscreens em tempo real para os clientes.

Com os conhecimentos adquiridos no Curso de Engenharia informática sentimos que nos proporcionou um nível de auto desempenho suficiente para desenvolver projectos mais complexos.

Olhando para o produto final denotamos que nossa solução é bastante prática, fácil de instalar e user friendly.

Não vimos a solução como um produto para venda, mas sim uma ferramenta freeware para complementar os desafios que o processo de transferência de ficheiros requer no dia-a-dia.

Apesar de este projecto não ser de dificuldade muito elevada, foi um projecto que nos permitiu alargar os conhecimentos na área, ajudando a preparar para o mundo profissional.

Foi um projecto bastante extenso, para além de não parecer, envolveu muitas tecnologias das quais damos especial ênfase a tecnologia WCF da Microsoft que facilita o desenvolvimento de aplicações orientadas a serviços, damos também ênfase ao protocolo de segurança SSL que nos permitiu toda a segurança da confidencialidade e integridade dos dados.

O Visual Studio é uma ferramenta muito intuitiva e fácil de trabalhar que favoreceu muito no desenvolvimento da solução, onde tem uma forma muito fácil de desenhar graficamente as aplicações.

Foi uma experiência boa e enriquecedora que nos permitiu adquirir muitos conhecimentos tanto a nível da Linguagem de programação C# como a nível de desenvolvimento de soluções complexas.

A perspectiva futura será melhorar a aplicação para que se torne ainda mais robusta e com um elevado nível de desempenho e a criação de clientes e servidores para Sistemas operativos diferentes incluído plataformas móveis.

Referências

Visual Studio

- <http://www.microsoft.com/visualstudio/ptb/products/visual-studio-overview>

OpenSSL

- <http://www.openssl.org/docs/>

OpenLDAP

- <http://www.openldap.org/>

SQL Server

- <http://www.microsoft.com/sqlserver/pt/br/default.aspx>

Windows Communication Foundation

- http://en.wikipedia.org/wiki/Windows_Communication_Foundation

Biblioteca MSDN

- <http://msdn.microsoft.com/pt-br/library/default.aspx>

SevenZipSharp

- <http://sevenzipsharp.codeplex.com/>

7-Zip

- <http://www.7-zip.org/>

Mono.Nat

- <http://projects.qnetp.net/projects/show/mono-nat>

How to and Questions

- <http://stackoverflow.com>

AppFabric

- <http://msdn.microsoft.com/en-us/windowsserver/default>

Glossário

WCF (Windows Communication Foundation) - É um modelo de programação unificado e ambiente de execução (Framework) criado pela Microsoft que visam a construção de aplicações orientadas a serviços

SOAP/XML – SOAP é um protocolo para troca de informações estruturadas em plataformas distribuídas. Baseia-se em linguagem XML para o formato das mensagens

Certificados X.509 - Padrão da criptografia de infra-estruturas de chave pública.

SSL - Ligação entre dois pontos de redes com encriptação na Camada de Transporte.

Camada de Transporte - A camada de transporte é responsável pelo transporte e regular do fluxo de informação da origem até ao destino de uma forma fiável e precisa.

SQL Server - É um sistema gerenciador de Banco de dados relacional (SGBD)

LDAP (Lightweight Directory Access Protocol) - É um protocolo para actualizar e pesquisar directórios através do protocolo TCP/IP.

Visual Studio 2010 - O **Microsoft Visual Studio** é um pacote de programas da Microsoft para desenvolvimento de software especialmente dedicado ao .NET Framework e às linguagens Visual Basic (VB), C, C++, C# (C Sharp) e J# (J Sharp).

P2P - Uma rede peer-to-peer (abreviado **P2P**) é arquitectura de conexão entre computadores em que todos eles actuam como servidor e clientes ao mesmo tempo.

Certificados digitais - Um certificado digital é um ficheiro que contém um conjunto de informações referentes a entidade para o qual o certificado foi emitido (seja uma empresa, pessoa física ou computador) mais a chave pública referente a chave privada que acredita-se ser de posse unicamente da entidade especificada no certificado.

Netografia

C#.NET

- <http://msdn.microsoft.com/en-us/library/system.net.security.sslstream.aspx>
- <http://msdn.microsoft.com/en-us/library/system.threading.thread.aspx>
- <http://msdn.microsoft.com/en-us/library/system.net.sockets.socket.aspx>
- <http://www.codeproject.com/Articles/326574/An-Introduction-to-Mutual-SSL-Authentication>
- <http://stackoverflow.com/questions/1669318/override-standard-close-x-button-in-a-windows-form>
- <http://msdn.microsoft.com/en-us/library/system.collections.hashtable.aspx>
- http://en.wikipedia.org/wiki/Secure_Sockets_Layer
- <http://msdn.microsoft.com/en-us/library/aa645739%28v=vs.71%29.aspx>
- <http://msdn.microsoft.com/en-us/library/ms173179.aspx>
- <http://www.7-zip.org/>
- <http://sevenzipsharp.codeplex.com/>
- <http://projects.gnetp.net/projects/show/mono-nat>