



Universidade do Minho
Escola de Engenharia

UNIVERSIDADE DO MINHO

Trabalho Prático N°3

Servidor Musical com Interface SNMP

Mestrado em Engenharia Informática

Gestão de Redes

8 de janeiro de 2020

Autores:

Carlos Alves : pg41840

pg41066 : Bruno Rodrigues



1 CONTEÚDO

2	Introdução	3
3	Contextualização.....	4
3.1	Conceitos Teóricos.....	4
3.1.1	SNMP.....	4
4	Estrutura da ferramenta desenvolvida.....	5
4.1	Fase A - Definição de requisitos funcionais.....	5
4.2	Fase B - Definição da MIB para servidor musical	6
4.3	Fase C - Implementação e teste do protótipo do agente SNMP	7
4.4	Fase D - Implementação e teste do Protótipo do reproduutor musical SNMP	8
4.5	Fase E - Implementação e teste de funcionalidades avançadas	9
4.6	Fase F - Construção de Manual de Utilização´	9
5	Conclusao	9
6	Anexo a - mib	10
7	Anexo b – interface web.....	13
8	Bibliografia.....	14

2 INTRODUÇÃO

Enquadrado na disciplina Gestão de Redes, do curso Mestrado em Engenharia Informática, foi pedido aos alunos, como trabalho prático 3, o desenvolvimento de um servidor de interface para uma coleção musical armazenada no sistema de ficheiros do sistema operativo do equipamento onde se encontra o agente SNMP. Ainda, esta solução deve, através da internet disponível a aplicações gestoras que assumem o papel de *players*.

Em suma, nesta solução será necessário a estruturação e criação de objetos implementados numa MIB e que sejam acedidos por SNMP, além de uma interface gráfica que se pareça com um *player* e que aceda à listagem da coleção musical disponível no sistema. Com este *player* deverá ser possível realizar determinadas ações, como por exemplo o controlo do nível de som de cada música entre outros que serão descritos ao longo deste documento.

Este trabalho estará dividido em várias fases, sendo elas:

- Fase A – Definição de requisitos funcionais
- Fase B – Definição da MIB para servidor musical
- Fase C – Implementação e teste do protótipo do agente SNMP
- Fase D – Implementação e teste do Protótipo do reprodutor musical SNMP
- Fase E – Implementação e teste de funcionalidades avançadas
- Fase F – Construção de Manual de Utilização

No presente documento poderá ser observado a descrição detalhada do desenvolvimento de cada fase acima mencionado.

A linguagem de programação que foi usada para o desenvolvimento deste projeto foi C#, com o complemento da *framework Blazor*, que nos permite criar uma interface web com mais facilidade.

3 CONTEXTUALIZAÇÃO

3.1 CONCEITOS TEÓRICOS

3.1.1 SNMP

Essencialmente, SNMP ou *Simple Network Management Protocol*(origem **RFC 1067**) é um protocolo da camada da aplicação, utilizado principalmente na monitorização dos dispositivos de uma rede. Este protocolo permite a troca de informações da parte gestora entre dispositivos da rede.

O protocolo conforme os anos foi evoluindo e, portanto, possui três versões, sendo que a última versão é caracterizada como sendo a mais segura em níveis de autenticação e encriptação, além da proteção dos pacotes durante a fase de transporte. Além disto, SNMP possui determinadas vantagens, fazendo com que este seja um dos protocolos mais utilizados na gestão de redes IP. Pois é suportado por diversos equipamentos, como por exemplo routers, impressoras, *switches* e ainda equipamentos de IoT(*Internet of Things*). Por consequência este também apresenta grandes vantagens sendo usado em redes de maiores dimensões, permitindo ao gestor ou administrador da rede administrar e controlar todos os nós da rede a partir de uma única interface gráfica. Ainda é de salientar a sua simplicidade.

Por último, conceitos que devem ser mencionados neste trabalho:

Management Information Base – é um ficheiro de texto onde estão presentes informações organizadas hierarquicamente. (Escalar e tabular) Basicamente, é um conjunto de definições que definem as propriedades do objeto gerenciado dentro do dispositivo a ser gerenciado.

Object Identifiers(OID) – Os OIDs identificam exclusivamente objetos geridos numa MIB(arvore).

SNMP Manager – Componente gestora do protocolo, tem como objetivo obter informações dos dispositivos onde está presente um agente SNMP, para isso o manager recorre às primitivas do protocolo.

SNMP Agent – componente presente no hardware onde se pretende monitorizar que tem como objetivo enviar informações recolhidas no sistema a que está integrado para o manager.

4 ESTRUTURA DA FERRAMENTA DESENVOLVIDA

4.1 FASE A - DEFINIÇÃO DE REQUISITOS FUNCIONAIS

Inicialmente no desenvolvimento do projeto, foi necessário começar por definir os requisitos funcionais da solução como um todo, isto é, desde todos os componentes que vão ser integrados no programa.

Portanto, a solução foi definida como tendo a capacidade de mostrar todas as músicas ou vídeos a partir de uma coleção armazenada numa pasta dentro do próprio sistema. A organização da coleção está definida e organizada na ordem alfabética, incluído algarismos e símbolos, exatamente como estaria organizada numa pasta do sistema. Seguindo este fluxo, também foi definido que a solução permitiria ao utilizador escolher a música ou vídeo que deseja ser assistido no sistema remoto. Ao definir os dados de cada música ou vídeo foi pensado a exibição dos seguintes atributos de cada elemento da lista:

- Nome da música / vídeo
- Artista;
- Género;
- Formato;
- Um *player*;
- Opção de download;

Em relação aos formatos suportados pela aplicação, a solução suportaria ficheiros musicais do tipo MP3, WAV e Flac(Com Problema), e ainda como complemento ficheiros de vídeo, neste caso MP4. Nos ficheiros de áudio era de importância que estes exibissem as informações anteriormente mencionadas como também um *player* que permite ao utilizador correr ou parar determinada música, podendo ainda observar a duração e o tempo decorrido de cada música e um botão que permitiria a manipulação do nível de som proveniente de cada música listada. Para proporcionar uma experiência mais amigável ao utilizador nesta componente da listagem da coleção musical, decidimos desenvolver uma interface web, acessível em qualquer browser, e com uma interface simples e intuitiva de modo a exibir todos os requisitos funcionais já mencionados.

De seguida foi realizada uma pequena discussão sobre os requisitos de segurança do sistema como um todo, pensando sempre nos possíveis ataques ou vetores que possibilitavam a falha total ou parcial do sistema SNMP – esta componente será discutida no decorrer deste documento.

Em suma, eis os requisitos funcionais que foram implementados:

- Listar remotamente uma coleção musical, de forma ordenada;
- Possibilidade do Utilizador escolher uma musica para ser tocada no sistema remoto;
- Suporte a vários tipos de ficheiros musicais (MP3, WAW) e vídeo (MP4);
- Interface web que permite o utilizador escolher e observar a coleção;
- Visualização de informações sobre cada música ou vídeo (Nome, Artista, Género, Formato, *player*);
- *Player* com informações sobre o tempo decorridos, tempo total do elemento da coleção;
- Suporte para controlo de volume auditivo, podendo desligar o som, ou colocar no nível que deseja;
- Suporte para avançar o elemento da coleção, seja música ou vídeo.

4.2 FASE B - DEFINIÇÃO DA MIB PARA SERVIDOR MUSICAL

Após definidos todos os requisitos funcionais que nos propusemos a desenvolver para esta solução, foi necessário desenvolver uma estrutura hierárquica e semântica da MIB, sendo que a MIB é por si só uma definição dos próprios objetos. Inicialmente, definimos o nome textual de um objeto para depois poder lhe atribui um significado ou uma definição. Dito isto, a baixo pode ser observada a estrutura base que desenvolvemos e que serviu de alicerce na construção da hierarquia da MIB.

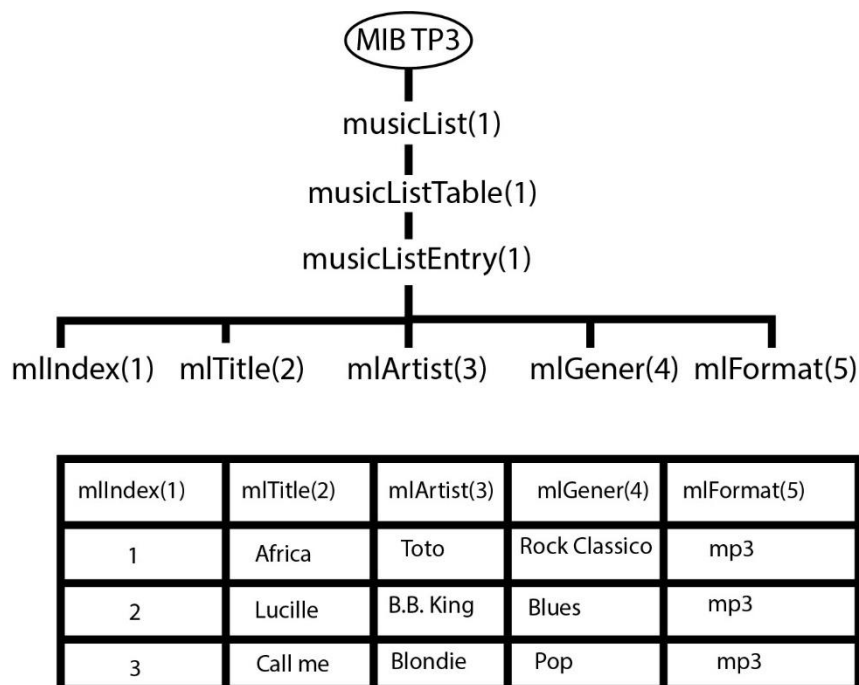


Figura 1 – Estrutura em arvore da MIB e a tabela correspondente

Esta estrutura posteriormente sofreu algumas mudanças, mas nada muito significativo como se pode verificar no Anexo A. No entanto, o componente mais importante a referir, é o grupo *musicList* que contém a tabela dos dados sobre cada elemento da coleção que definimos na fase A – Definição de Requisitos Funcionais. (índice, título, artista, género, formato e caminho)

Para finalizar esta componente, é de mencionar a importância do auxílio de ferramentas como o MIB Designer e ferramentas de verificação (*Web Interface to libsmi Tools*).

4.3 FASE C - IMPLEMENTAÇÃO E TESTE DO PROTÓTIPO DO AGENTE SNMP

Para desenvolver esta solução, construímos um agente SNMP em linguagem C#, com apoio de algumas bibliotecas que nos ajudaram a realizar determinados processos, entre elas usamos a biblioteca *SharpSNMP* ou *#SNMP*. Juntamente com isso, desenvolvemos várias classes que ajudaram na criação das tabelas e dos próprios elementos pertencentes a essas mesmas tabelas.

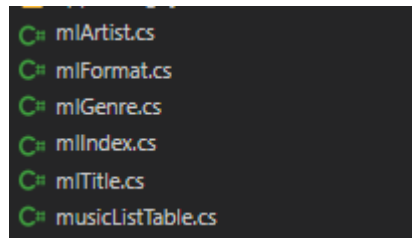


Figura 2 – Elementos da Tabela e a classe da tabela

Na construção do agente e tendo em mente a discussão sobre requisitos de segurança na fase inicial do projeto, fizemos uso da componente *Security* da biblioteca *#SnmpLib* para incorporar algumas características de segurança ao agente. Salvo, que a versão do snmp utilizada neste projeto é a **Snmp v2**, sendo que esta não é a versão mais comumente associada a segurança. Caso tivesse sido usado a versão superior que possui um novo modelo de segurança, as diversas falhas presentes nas versões anteriores seriam mitigadas. Visto que o novo modelo garante confidencialidade, integridade, autenticação e controlo de acesso, o que não é verificado na versão que utilizamos, pois, a autenticação SNMP nesta versão e na anterior equivale a pouco mais que uma senha (*string de comunidade*), enviada em texto limpo, isto é, não é encriptada entre o gestor e o agente. Portanto no agente, foram utilizadas classes de base de provedores de privacidade para **AES**, e um algoritmo de *message-digest* – o **MD5**. Ainda no agente, este foi desenvolvido para fazer procura no repositório de 5 em 5 segundos, se for adicionado novos elementos, estes em 5 segundos serão exibidos e o agente estará a ouvir na porta 2002.

Resumidamente, nesta componente implementamos grande parte do que nos propusemos exceto a componente gráfica.

4.4 FASE D - IMPLEMENTAÇÃO E TESTE DO PROTÓTIPO DO REPRODUTOR MUSICAL SNMP

Nesta fase foram melhorados determinados aspetos que estavam menos bem na fase anterior e ainda foi desenvolvida uma interface gráfica, num acessível através de qualquer browser. Para isto, foi utilizado a *framework Blazor* que por si só já nos oferece um “ambiente” com vários elementos gráficos já implementados, e ainda é simples de ser manipulado. Na construção da interface tentamos que este fosse o mais amigável possível para o utilizador, essencialmente, foi construída uma tabela onde é listada toda a coleção musical ou vídeos, ordenadas por ordem alfabética e que possuem as informações que inicialmente planeamos implementar.

Nome da música, nome do artista, o género que pertence a música podendo este ter mais que um género, o formato do item que aparece na lista, seja ele *mp3*, *wav*, *mp4*; E por fim o *player*. Este *player* possibilita dar play e stop de qualquer elemento da lista (seja música ou vídeo), aumentar ou diminuir o volume do som, acompanhar o tempo da música, avançar e diminuir o tempo da música ou vídeo e por fim uma opção que permite realizar o download do elemento para a própria máquina. Em relação ao vídeo é ainda possível colocá-lo em tela cheia. Abaixo pode ser observado a interface.

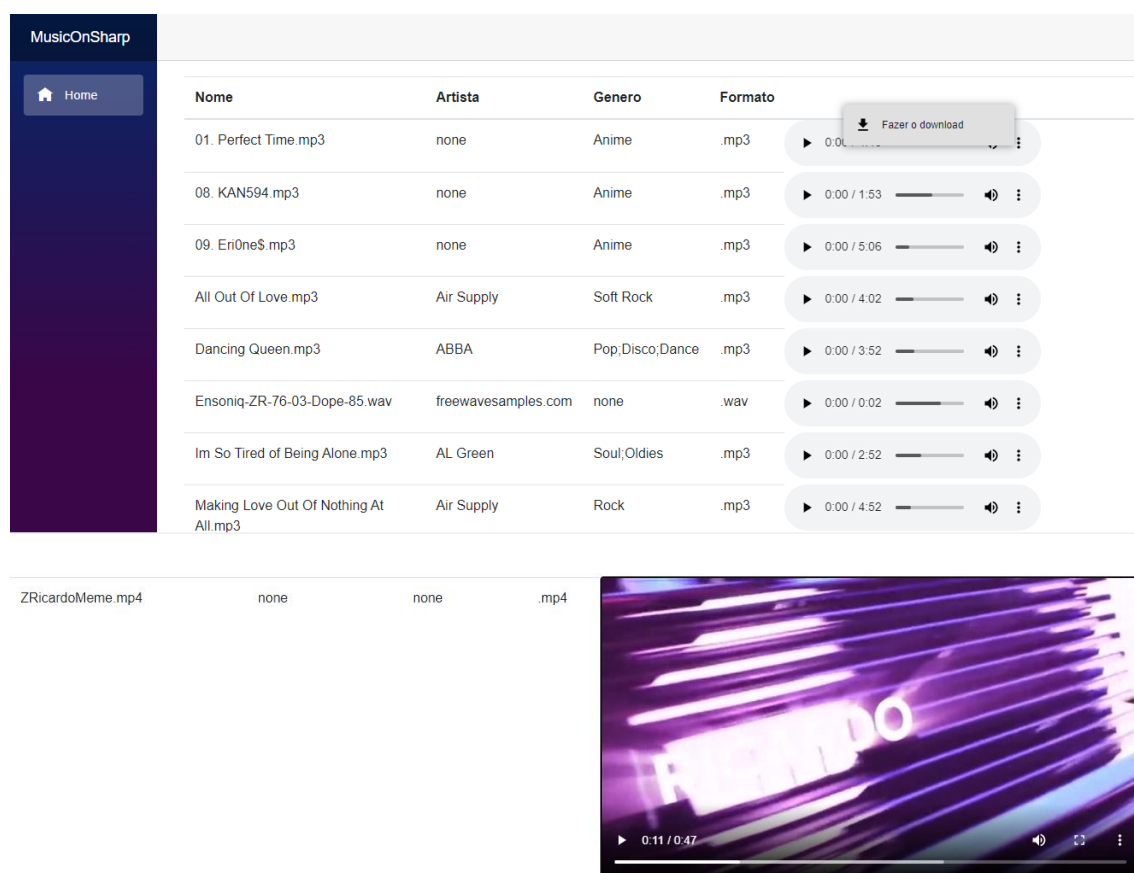


Figura 3 – Interface Gráfica com vários elementos da coleção (mp3, wav, mp4)

4.5 FASE E - IMPLEMENTAÇÃO E TESTE DE FUNCIONALIDADES AVANÇADAS

Nesta fase foram melhoradas algumas componentes, como por exemplo a própria MIB. Pois verificou-se a falta de alguns objetos, mas nada de muito relevante. Mas como já referimos a MIB pode ser observada na íntegra no anexo A deste documento.

4.6 FASE F - CONSTRUÇÃO DE MANUAL DE UTILIZAÇÃO'

Iniciar o programa

- Abrir a pasta do programa;
- Procurar pelo executável MusicOnSharp
- Executar MusicOnSharp
- De seguida, aceder ao browser e entrar no localhost:5000 ou 5001

Caso se verifique algum problema na execução do programa, é necessário abrir o repositório do projeto e executar o seguinte comando: *dotnet run*. Sendo necessário que o gestor tenha instalado o .NET 5SDK.

5 CONCLUSÃO

Ao longo do desenvolvimento deste projeto, é de salientar que a maior dificuldade foi no início do desenho da solução, principalmente em questões relacionadas à MIB. Pois após desenhar a estrutura da MIB não estávamos a perceber como proceder à implementação da mesma. No entanto, consideramos que esta experiência de trabalho, um pouco exaustiva, foi de mais valia para afinar os nossos conhecimentos adquiridos no âmbito da unidade curricular.

Para terminar, resta-nos agradecer aos docentes responsáveis pela disciplina pela disponibilidade no esclarecimento das dúvidas que foram sendo encontradas. Além disso, e tendo em conta o trabalho desenvolvido, achamos que grande parte dos objetivos propostos inicialmente foram alcançados.

6 ANEXO A - MIB

```

GR-MUSICTP3-MIB DEFINITIONS ::= BEGIN

IMPORTS
    experimental,
    MODULE-IDENTITY,
    OBJECT-TYPE,
    Integer32
        FROM SNMPv2-SMI
    DisplayString
        FROM SNMPv2-TC
    OBJECT-GROUP
        FROM SNMPv2-CONF;

grtp3MIB MODULE-IDENTITY
    LAST-UPDATED "20201222156Z"      -- Dec 22, 2020, 9:56:00 PM
    ORGANIZATION ""
    CONTACT-INFO
        ""
    DESCRIPTION
        ""
    REVISION "20201222156Z"          -- Dec 22, 2020, 9:56:00 PM
    DESCRIPTION
        "Initial version."
    -- 1.3.6.1.3.3
    ::= { experimental 3 }

musicListTable OBJECT-TYPE
    SYNTAX SEQUENCE OF MusicListEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Lista das musicas"
    -- 1.3.6.1.3.4
    ::= { experimental 4 }

musicListEntry OBJECT-TYPE
    SYNTAX MusicListEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "contem informacao de uma determinada musica"
    INDEX {
        mlIndex }
    -- 1.3.6.1.3.4.1
    ::= { musicListTable 1 }

MusicListEntry ::= SEQUENCE {

    mlIndex Integer32,
    mlTitle DisplayString,
    mlArtist DisplayString,
    mlGenre DisplayString,
    mlFormat DisplayString }

```

```

mlIndex OBJECT-TYPE
    SYNTAX  Integer32 (0..999)
    MAX-ACCESS read-only
    STATUS  current
    DESCRIPTION
        "Valor atribuido a cada musica"
    -- 1.3.6.1.3.4.1.1
    ::= { musicListEntry 1 }

mlTitle OBJECT-TYPE
    SYNTAX  DisplayString
    MAX-ACCESS read-only
    STATUS  current
    DESCRIPTION
        "String que contem titulo da musica ou peca"
    -- 1.3.6.1.3.4.1.2
    ::= { musicListEntry 2 }

mlArtist OBJECT-TYPE
    SYNTAX  DisplayString
    MAX-ACCESS read-only
    STATUS  current
    DESCRIPTION
        "String que contem o nome do artista da musica"
    -- 1.3.6.1.3.4.1.3
    ::= { musicListEntry 3 }

mlGenre OBJECT-TYPE
    SYNTAX  DisplayString
    MAX-ACCESS read-only
    STATUS  current
    DESCRIPTION
        "String que contem o genero da musica ou peca"
    -- 1.3.6.1.3.4.1.4
    ::= { musicListEntry 4 }

mlFormat OBJECT-TYPE
    SYNTAX  DisplayString
    MAX-ACCESS read-only
    STATUS  current
    DESCRIPTION
        "String que contem formato da musica ou peca"
    -- 1.3.6.1.3.4.1.5
    ::= { musicListEntry 5 }

mlPath OBJECT-TYPE
    SYNTAX  DisplayString
    MAX-ACCESS read-only
    STATUS  current
    DESCRIPTION
        "String que contem o path da musica"
    -- 1.3.6.1.3.4.1.6
    ::= { musicListEntry 6 }
    
```

```
musicList OBJECT-GROUP
  OBJECTS {
    mlIndex,
    mlTitle,
    mlArtist,
    mlGenre,
    mlFormat,
    mlPath }
  STATUS current
  DESCRIPTION
    "Grupo Music List"
  -- 1.3.6.1.3.7
  ::= { experimental 7 }


END
```

7 ANEXO B – INTERFACE WEB

MusicOnSharp

Home

Nome	Artista	Genero	Formato	
01. Perfect Time.mp3	none	Anime	.mp3	▶ 0:00 / 4:43 🔊 ⋮
08. KAN594.mp3	none	Anime	.mp3	▶ 0:00 / 1:53 🔊 ⋮
09. En0ne\$.mp3	none	Anime	.mp3	▶ 0:00 / 5:06 🔊 ⋮
All Out Of Love.mp3	Air Supply	Soft Rock	.mp3	▶ 0:00 / 4:02 🔊 ⋮
Dancing Queen.mp3	ABBA	Pop,Disco,Dance	.mp3	▶ 0:00 / 3:52 🔊 ⋮
Ensoniq ZR-76-03:Dope-85.wav	freewavesamples.com	none	.wav	▶ 0:00 / 0:02 🔊 ⋮
Im So Tired of Being Alone.mp3	Al Green	Soul,Oldies	.mp3	▶ 0:00 / 2:52 🔊 ⋮
Making Love Out Of Nothing At All.mp3	Air Supply	Rock	.mp3	▶ 0:00 / 4:52 🔊 ⋮
Sample_BeeMoved_96kHz24bit.flac	Blue Monday FM	none	.flac	▶ 0:00 / 0:00 🔊 ⋮
ZRicardoMeme.mp4	none	none	.mp4	▶ 0:11 / 0:47 🔊 ⋮



8 BIBLIOGRAFIA

agentpp. (s.d.). Obtido de MIB Designer: <https://agentpp.com/tools/mibdesigner.html>

Mars. (s.d.). *Construção de Agentes SNMP*. Obtido de penta2.ufrgs:
http://penta2.ufrgs.br/ppt/constr_agente/

net-snmp. (24 de 08 de 2019). Obtido de <http://net-snmp.sourceforge.net/docs/mibs/host.html#hrSWRunTable>

sharpsnmp. (2015). Obtido de <https://docs.sharpsnmp.com/>

stsrki. (s.d.). *Blazorise*. Obtido de <https://github.com/stsrki/Blazorise>