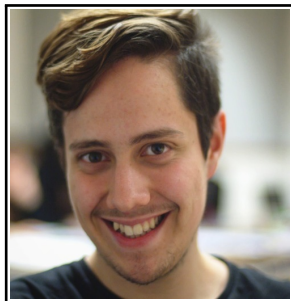


UNIVERSIDADE DO MINHO
MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA
GESTÃO E VIRTUALIZAÇÃO DE REDES
GESTÃO DE REDES

FICHA DE TRABALHO PRÁTICO Nº3

**AGENTE SNMP PARA MONITORIZAÇÃO DE
DATAS DE EVENTOS**



Filipe Guimarães **a85308**



Joana Gomes **a84912**

28 de fevereiro de 2021

Conteúdo

1	INTRODUÇÃO	2
2	ANÁLISE DE REQUISITOS	2
2.1	OBJETOS	2
3	ESPECIFICAÇÃO DA MIB	4
4	IMPLEMENTAÇÃO DA MIB	4
5	DESENVOLVIMENTO	5
5.1	Evento	5
5.2	EventosDAO	6
5.3	Admin	7
5.4	Agent	7
5.5	Controller	8
5.6	Outras	9
6	MANUAL DE UTILIZAÇÃO	10
7	CONCLUSÃO	11
8	BIBLIOGRAFIA	11
9	ANEXOS	12
9.1	EXEMPLO DE UTILIZAÇÃO DO AGENTE	12
9.2	ESPECIFICAÇÃO DA MIB	14

1 INTRODUÇÃO

Neste relatório encontra-se uma descrição detalhada não só dos resultados obtidos como também do processo em si da realização do trabalho prático nº3 da Unidade curricular de Gestão de redes.

Como proposto pela equipa docente, este projeto tem como tema central a criação de um agente SNMP que implemente uma MIB para monitorização de datas de eventos.

Como estrutura deste relatório, primeiramente apresenta-se os a interpretação de requisitos do enunciado de uma forma sucinta. Posteriormente é apresentado o método de especificação, implementação e desenvolvimento, respetivamente. Este documento contém também um manual de utilização bem como um caso de utilização para o programa.

2 ANÁLISE DE REQUISITOS

2.1 OBJETOS

Para o correto desenvolvimento do resto da criação do agente SNMP, começamos por delinear os objetos da MIB de forma a ser possível a correta elaboração dos métodos.

A nossa eventosMIB comporta dois objetos: `numeroEventos`, que representa o número de eventos na MIB, e a tabela `eventosTable`, que alberga os objetos com a informação para cada um dos eventos.

A `eventosTable` é constituída por objetos do tipo *eventoEntry* que, por sua vez, contém os seguintes objetos:

- `eventoIndex` (**Counter**) - índice do evento;
- `eventoName` (**OctectString**) - o nome do evento;
- `eventoEstado` (**Integer32**) - indica se o evento já decorreu [0], está a decorrer [1] ou ainda vai decorrer no futuro [2];
- `eventoTempoAnos`, `eventoTempoMeses`, `eventoTempoSemanas`, `eventoTempoHoras`

e eventoTempoMinutos (**Integer32**) - indicam há quanto tempo o evento ocorreu (caso o evento seja passado) ou quanto tempo falta para o evento ocorrer (caso seja no futuro);

- eventoDuracao (**Timeticks**) - duração do evento;
- eventoFrasePassado, eventoFrasePresente e eventoFraseFuturo (**OctetString**) - frases de alerta para quando o evento o já aconteceu, está a acontecer ou é futuro, respetivamente.

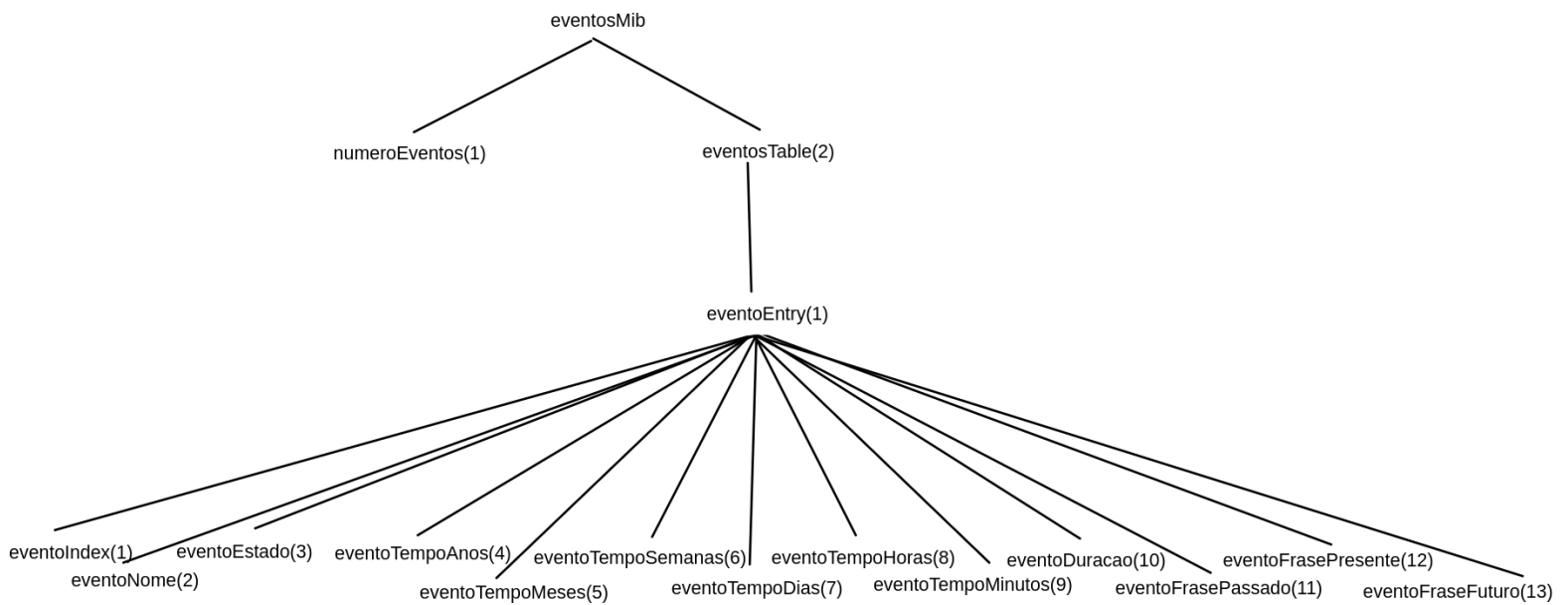


Figura 1: Esquema elaborado para a nossa MIB

3 ESPECIFICAÇÃO DA MIB

Para criarmos mais facilmente a especificação da MIB que desenvolvemos usámos, como sugerido no enunciado, o *MibDesigner*. Definimos todos os objetos e ficamos com a seguinte especificação representada em lista.

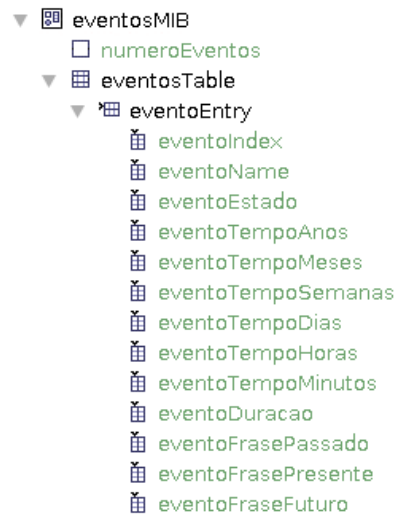


Figura 2: Mib no *MibDesigner*

Recorremos mias uma vez à ferramenta para gerar o ficheiro com a especificação da MIB para ser usado posteriormente no *AgenPro*.

4 IMPLEMENTAÇÃO DA MIB

De forma a gerar automaticamente as classes do *snmp4j* que implementam a mib que especificamos recorremos, como sugerido no enunciado, ao *AgenPro*.

Para automatizar o processo abrimos o projeto *snmp4j-agent3_x_by_selection* já configurado na pasta *projects* da aplicação e geramos o código com todos os ficheiros que serão precisos na fase de desenvolvimento.

5 DESENVOLVIMENTO

Após gerarmos o código com o *AgenPro* abrimos o projeto com o nosso *IDE* e compilamos, verificando que estava a funcionar e a aceitar pedidos como apresentado na imagem.

```
wtv@wtv-pc ~$ snmpwalk -v 2c -c public 127.0.0.1:3003 .1
SNMPv2-MIB::sysDescr.0 = STRING: SNMP4J-Agent 3.4.4 [3.4.4] - Linux - amd64 - 5.10.15-1-MANJARO
SNMPv2-MIB::sysObjectID.0 = OID: SNMPv2-SMI::enterprises.4976.10
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (4086) 0:00:40.86
SNMPv2-MIB::sysContact.0 = STRING: System Administrator
SNMPv2-MIB::sysName.0 = STRING:
SNMPv2-MIB::sysLocation.0 = STRING: <edit location>
SNMPv2-MIB::sysServices.0 = INTEGER: 72
SNMPv2-MIB::sysORLastChange.0 = Timeticks: (0) 0:00:00.00
SNMPv2-MIB::snmpInPkts.0 = Counter32: 16
SNMPv2-MIB::snmpInBadVersions.0 = Counter32: 0
```

Figura 3: *snmpwalk* ao agente.

De seguida apresentamos a classes de *java* desenvolvidas e o respetivo funcionamento.

5.1 Evento

Esta classe foi criada com o intuito de fornecer a estrutura de um evento e fornecer métodos para escrever na MIB e no ficheiro o mesmo.

Os métodos principais desta classe são:

- ***getEntry()***: Este método é chamado pelo Agent e é responsável por preencher a linha da MIB passada como argumento usando os set's que foram gerados pelo *AgenPro*.

Recorre ao método ***setEventoTime()*** que divide o tempo que falta ou já passou em anos, meses, semanas, dias, horas e minutos. Em primeiro lugar transforma a duração em milissegundos e depois faz divisões inteiras sucessivas para retirar cada um dos campos.

```
1 public void getEntry(GrEventosMib.EventoEntryRow row) {
2     row.setEventoName(new OctetString(this.name));
3     row.setEventoDuracao(new TimeTicks(this.duracao.toMillis()));
4     [...]
5
6     if (this.data.isAfter(LocalDateTime.now())) {
```

```

7         row.setEventoEstado(new Integer32(2)); //futuro
8         Duration duration = Duration.between(this.data,
9         LocalDateTime.now());
10        setEventoTime(row, duration);
11    } else if (this.data.plus(duracao).isBefore(LocalDateTime.now
12    ())) {
13        row.setEventoEstado(new Integer32(0)); //passado
14        Duration duration = Duration.between(this.data,
15        LocalDateTime.now());
16        setEventoTime(row, duration);
17    } else {
18        row.setEventoEstado(new Integer32(1)); //presente
19        setEventoTime(row, 0, 0, 0, 0, 0, 0);
20    }
21 }

```

Listing 1: Método `getEntry()`

- ***itsTime()*** : Verifica se já está na hora de apagar o evento da MIB, retornando *true* caso seja altura de remover o evento da MIB e *false* caso contrário.

```

1 public boolean itsTime() {
2     [...]
3     else if (this.data.plus(duracao).plus(timer).isBefore(
4     LocalDateTime.now())) {
5         return true;
6     }
7     [...]
8 }

```

Listing 2: Método `itsTime()`

- ***toString()*** : Utilizamos o método `toString()` para guardar a informação dos eventos numa entrada do ficheiro `eventos.txt`.

```

1 {nome="Web Summit", data="2021-02-23T11:06:19.102301213", duracao
2   ="PT16M40S", frasePassado="A Web Summit ainda nap comecou.",
   frasePresente="A Web Summit esta a decorrer!", fraseFuturo="A Web
   Summit ja terminou."}

```

Listing 3: Exemplo de uma linha do ficheiro `eventos.txt`

5.2 EventosDAO

Esta classe foi desenvolvida para abstrair o acesso ao ficheiro de eventos. Fornece métodos para obter os dados numa lista e também para adicionar e remover eventos a pedido.

Os métodos principais desta classe são:

- ***getList()***: Este método compila todos os eventos declarados no ficheiro numa lista recorrendo ao ***lineToEvent()*** para transformar cada linha num objeto *Evento*. Faz uso do *BufferedReader* para ler o ficheiro linha a linha.

```
1 public List<Evento> getList() throws IOException {
2     BufferedReader br = new BufferedReader(new FileReader(
3         eventFile.getPath()));
4     List<Evento> eventos = new ArrayList<>();
5     String line = br.readLine();
6     while (line != null) {
7         eventos.add(lineToEvent(line.trim()));
8         line = br.readLine();
9     }
10    br.close();
11
12    eventos.removeIf(Evento::itsTime);
13
14    return eventos;
15 }
```

Listing 4: Método `getList()`

- ***add()***: Adiciona um evento ao ficheiro usando o *DataOutputStream* com a flag *Append*.
- ***remove()***: Remove uma linha do ficheiro copiando para um ficheiro temporário todas as linhas que não contêm um evento com o nome fornecido. No final é substituído o ficheiro original por este temporário.

5.3 Admin

Classe que armazena todas as instâncias necessárias para a execução do programa. Só existe uma instância desta classe de forma a que todas as outras usem a mesma.

5.4 Agent

Gerado pelo *AgenPro*. Acrescentamos o método *update*. Este método é executado por uma thread que, dentro do *while*, atualiza a MIB e espera o tempo fornecido pelo utilizador para voltar a repetir.


```

1 private void update() {
2     new Thread(() -> {
3         while (true) {
4             lock();
5             try {
6                 for (int i = 0; i<= this.oldIndex; i++){
7                     modules.getGrEventosMib()
8                         .getEventoEntry()
9                         .removeRow(new OID(String.valueOf(i)));
10                }
11
12                int index=1;
13                for (Evento e : Admin.getInstance().getEventos()) {
14                    GrEventosMib.EventoEntryRow row = modules
15                        .getGrEventosMib()
16                        .getEventoEntry()
17                        .createRow(new OID(String.valueOf(index)));
18                    row.setEventoIndex(new Counter32(index));
19                    e.getEntry(row);
20                    modules.getGrEventosMib()
21                        .getEventoEntry()
22                        .addRow(row);
23                    index++;
24                }
25                index--;
26                modules.getGrEventosMib()
27                    .getNumeroEventos()
28                    .setValue(new Counter32(index));
29                this.oldIndex = index;
30            } finally {
31                unlock();
32            }
33            try {
34                Thread.sleep(Admin.getInstance().getUpdateTime());
35            } catch (InterruptedException e) {
36                e.printStackTrace();
37            }
38        }
39    }).start();
40 }

```

Listing 5: Método update()

5.5 Controller

Controlador para o *javaFx*.

5.6 Outras

As outras classes foram geradas pelo *AgenPro* e não efetuamos qualquer modificação.

6 MANUAL DE UTILIZAÇÃO

Agente

IP: 127.0.0.1 PORT: 3003 **Start** (1)

Update Time

Segundos: 10 **Change** (2)

Adicionar Evento

Nome: _____

Data: _____ Hora: 0 H^0 M

Duração: 0 H^0 M

Apagar em: ☐ 0 D^0 H^0 M

Frase Passado: _____

Frase Presente: _____

Frase Futuro: _____

Add (3)

Remover Evento

Nome: _____

Remove (4)

Figura 4: Apresentação da interface gráfica

1. Introduz-se o IP e porta onde o agente irá correr e clica-se para executar o agente.
2. Introduz-se o tempo de atualização da MIB e carrega-se para aplicar.
3. Introduz-se todas as informações sobre o evento e clica-se para adicionar.
4. Introduz-se o nome do evento e clica-se para remover.

Um cenário de utilização encontra-se na secção 9 - Anexos.

7 CONCLUSÃO

Com este terceiro e último projeto de Gestão de Redes podemos obter um ainda maior conhecimento do protocolo SNMP mais propriamente do funcionamento de um agente SNMP.

Simultaneamente, foi também conseguida a familiarização com as ferramentas *MIB Designer* e *AgenPro* e mais experiência com a API SNMP4J.

O grupo perfaz uma apreciação geral positiva desta Ficha de Trabalho, sendo implementadas com sucesso as funcionalidades sugeridas e ainda um dos requisitos funcionais opcionais.

8 BIBLIOGRAFIA

9 ANEXOS

9.1 EXEMPLO DE UTILIZAÇÃO DO AGENTE

1. Abre-se o agente executando o *"Eventos-1.0.jar"*.

```
wtv@wtv-pc ~/universidade/GR/GR TP3 main ± java -jar Eventos-1.0.jar
```

2. A pagina inicial abre.

Agente

IP
127.0.0.1

PORT
3003

Start

Update Time

Segundos
10

Change

Adicionar Evento

Nome

Data

Hora

0

H

0

M

Duração

0

H

0

M

Apagar em

☐

0

D

0

H

0

M

Frase Passado

Frase Presente

Frase Futuro

Add

Remover Evento

Nome

Remove

3. O Agente é iniciado.

Agente		Update Time
IP 127.0.0.1	PORT 3003	Segundos 10
<input type="button" value="Start"/>		<input type="button" value="Change"/>

4. É adicionado aos eventos a *Aula de VR* com os respetivos campos preenchidos.

Adicionar Evento	
Nome Aula de VR	Frase Passado A aula de VR começa em:
Data 2/26/2021	Frase Presente A aula de VR esta a decorrer.
Hora 16 H 0 M	Frase Futuro A aula de VR acabou ha:
Duração 3 H 0 M	
Apagar em <input checked="" type="checkbox"/> 7 D 0 H 0 M	<input type="button" value="Add"/>

5. Verifica-se que o ficheiro *events.txt* tem agora uma linha com o evento criado.

```
{nome="Aula de VR", data="2021-02-26T16:00", duracao="PT3H", frasePassado="A aula de VR começa em: ",
```

```
frasePresente="A aula de VR esta a decorrer.", fraseFuturo="A aula de VR acabou ha: ", timer="PT168H"}
```

6. Pode-se fazer GET's no MibBrowser verificando que o evento está presente.

Name/OID	Value	Type	IP:Port
numeroEventos.0	1	Counter32	127.0.0.1:3003
eventoIndex.1	1	Counter32	127.0.0.1:3003
eventoName.1	Aula de VR	OctetString	127.0.0.1:3003
eventoEstado.1	0	Integer	127.0.0.1:3003
eventoTempoAnos.1	0	Integer	127.0.0.1:3003
eventoTempoMeses.1	0	Integer	127.0.0.1:3003
eventoTempoSemanas.1	0	Integer	127.0.0.1:3003
eventoTempoDias.1	1	Integer	127.0.0.1:3003
eventoTempoHoras.1	5	Integer	127.0.0.1:3003
eventoTempoMinutos.1	14	Integer	127.0.0.1:3003
eventoDuracao.1	30 hours (10800000)	TimeTicks	127.0.0.1:3003
eventoFrasePassado.1	A aula de VR esta a decorrer.	OctetString	127.0.0.1:3003
eventoFrasePresente.1	A aula de VR acabou ha:	OctetString	127.0.0.1:3003
eventoFraseFuturo.1	A aula de VR comeca em:	OctetString	127.0.0.1:3003

7. Remove-se o evento introduzindo o nome.

Remover Evento

Nome

Remove

9.2 ESPECIFICAÇÃO DA MIB

```

1 GR-EVENTOS-MIB DEFINITIONS ::= BEGIN
2
3 IMPORTS
4     enterprises,
5     MODULE-IDENTITY,
6     OBJECT-TYPE,
7     Counter32,
8     TimeTicks
9     FROM SNMPv2-SMI
10    OBJECT-GROUP
11    FROM SNMPv2-CONF;
12
13 eventosMIB MODULE-IDENTITY
14     LAST-UPDATED "202102201414Z" -- Feb 20, 2021, 2:14:00 PM
15     ORGANIZATION ""
16     CONTACT-INFO
17         ""
18     DESCRIPTION
19         "MIB para monitorizacao de datas para
20         eventos."
21     REVISION "202102201414Z" -- Feb 20, 2021, 2:14:00 PM
22     DESCRIPTION
23         "Initial version."

```

```

24 -- 1.3.6.1.4.1.8888
25 ::= { enterprises 8888 }
26
27
28 numeroEventos OBJECT-TYPE
29     SYNTAX Counter32
30     MAX-ACCESS read-only
31     STATUS current
32     DESCRIPTION
33         "Numero de eventos na MIB."
34 -- 1.3.6.1.4.1.8888.1
35 ::= { eventosMIB 1 }
36
37
38 eventosTable OBJECT-TYPE
39     SYNTAX SEQUENCE OF EventoEntry
40     MAX-ACCESS not-accessible
41     STATUS current
42     DESCRIPTION
43         "Tabela que armazena todos os eventos."
44 -- 1.3.6.1.4.1.8888.2
45 ::= { eventosMIB 2 }
46
47
48 eventoEntry OBJECT-TYPE
49     SYNTAX EventoEntry
50     MAX-ACCESS not-accessible
51     STATUS current
52     DESCRIPTION
53         "Entrada para a tabela de eventos."
54     INDEX {
55         eventoIndex }
56 -- 1.3.6.1.4.1.8888.2.1
57 ::= { eventosTable 1 }
58
59
60 EventoEntry ::= SEQUENCE {
61
62     eventoIndex          Counter32,
63     eventoName           OCTET STRING,
64     eventoEstado         INTEGER,
65     eventoTempoAnos      INTEGER,
66     eventoTempoMeses     INTEGER,
67     eventoTempoSemanas   INTEGER,
68     eventoTempoDias       INTEGER,
69     eventoTempoHoras      INTEGER,
70     eventoTempoMinutos    INTEGER,
71     eventoDuracao        TimeTicks,
72     eventoFrasePassado    OCTET STRING,
73     eventoFrasePresente   OCTET STRING,
74     eventoFraseFuturo     OCTET STRING }
75
76
77 eventoIndex OBJECT-TYPE
78     SYNTAX Counter32
79     MAX-ACCESS read-only
80     STATUS current

```



```

81  DESCRIPTION
82      "Identificador do evento."
83      -- 1.3.6.1.4.1.8888.2.1.1
84      ::= { eventoEntry 1 }
85
86
87  eventoName OBJECT-TYPE
88      SYNTAX  OCTET STRING
89      MAX-ACCESS read-only
90      STATUS  current
91      DESCRIPTION
92          "Nome do evento."
93          -- 1.3.6.1.4.1.8888.2.1.2
94          ::= { eventoEntry 2 }
95
96
97  eventoEstado OBJECT-TYPE
98      SYNTAX  INTEGER
99      MAX-ACCESS read-only
100     STATUS  current
101     DESCRIPTION
102         "Estado atual do evento.
103         (0) Futuro
104         (1) Presente
105         (2) Passado"
106         -- 1.3.6.1.4.1.8888.2.1.3
107         ::= { eventoEntry 3 }
108
109
110  eventoTempoAnos OBJECT-TYPE
111      SYNTAX  INTEGER
112      MAX-ACCESS read-only
113      STATUS  current
114      DESCRIPTION
115          "Tempo que falta para o evento ou, caso seja no passado, ha quanto tempo
116          foi o evento.
117          So a parte referente aos anos."
118          -- 1.3.6.1.4.1.8888.2.1.4
119          ::= { eventoEntry 4 }
120
121
122  eventoTempoMeses OBJECT-TYPE
123      SYNTAX  INTEGER
124      MAX-ACCESS read-only
125      STATUS  current
126      DESCRIPTION
127          "Tempo que falta para o evento ou, caso seja no passado, ha quanto tempo
128          foi o evento.
129          So a parte referente aos meses."
130          -- 1.3.6.1.4.1.8888.2.1.5
131          ::= { eventoEntry 5 }
132
133
134  eventoTempoSemanas OBJECT-TYPE
135      SYNTAX  INTEGER
136      MAX-ACCESS read-only
137      STATUS  current

```

```

136 DESCRIPTION
137     "Tempo que falta para o evento ou, caso seja no passado, ha quanto tempo
138     foi o evento.
139     So a parte referente as semanas."
139 -- 1.3.6.1.4.1.8888.2.1.6
140 ::= { eventoEntry 6 }
141
142
143 eventoTempoDias OBJECT-TYPE
144     SYNTAX INTEGER
145     MAX-ACCESS read-only
146     STATUS current
147     DESCRIPTION
148         "Tempo que falta para o evento ou, caso seja no passado, ha quanto tempo
149         foi o evento.
150         So a parte referente aos dias."
150 -- 1.3.6.1.4.1.8888.2.1.7
151 ::= { eventoEntry 7 }
152
153
154 eventoTempoHoras OBJECT-TYPE
155     SYNTAX INTEGER
156     MAX-ACCESS read-only
157     STATUS current
158     DESCRIPTION
159         ""
160 -- 1.3.6.1.4.1.8888.2.1.8
161 ::= { eventoEntry 8 }
162
163
164 eventoTempoMinutos OBJECT-TYPE
165     SYNTAX INTEGER
166     MAX-ACCESS read-only
167     STATUS current
168     DESCRIPTION
169         "Tempo que falta para o evento ou, caso seja no passado, ha quanto tempo
170         foi o evento.
171         So a parte referente aos minutos."
171 -- 1.3.6.1.4.1.8888.2.1.9
172 ::= { eventoEntry 9 }
173
174
175 eventoDuracao OBJECT-TYPE
176     SYNTAX TimeTicks
177     MAX-ACCESS read-only
178     STATUS current
179     DESCRIPTION
180         "Duracao do evento."
181 -- 1.3.6.1.4.1.8888.2.1.10
182 ::= { eventoEntry 10 }
183
184
185 eventoFrasePassado OBJECT-TYPE
186     SYNTAX OCTET STRING
187     MAX-ACCESS read-only
188     STATUS current
189     DESCRIPTION

```

```

190     "Frase para descrever o evento no passado."
191     -- 1.3.6.1.4.1.8888.2.1.11
192     ::= { eventoEntry 11 }
193
194
195 eventoFrasePresente OBJECT-TYPE
196     SYNTAX OCTET STRING
197     MAX-ACCESS read-only
198     STATUS current
199     DESCRIPTION
200         "Frase para descrever o evento no presente."
201         -- 1.3.6.1.4.1.8888.2.1.12
202         ::= { eventoEntry 12 }
203
204
205 eventoFraseFuturo OBJECT-TYPE
206     SYNTAX OCTET STRING
207     MAX-ACCESS read-only
208     STATUS current
209     DESCRIPTION
210         "Frase para descrever o evento no futuro."
211         -- 1.3.6.1.4.1.8888.2.1.13
212         ::= { eventoEntry 13 }
213
214
215 -- Conformance
216 --
217
218 grMIBConf OBJECT IDENTIFIER
219     -- 1.3.6.1.4.1.8888.3
220     ::= { eventosMIB 3 }
221
222 -- Groups
223 --
224
225 grMIBGroups OBJECT IDENTIFIER
226     -- 1.3.6.1.4.1.8888.3.1
227     ::= { grMIBConf 1 }
228
229 -- Compliances
230 --
231
232 grMIBCompliances OBJECT IDENTIFIER
233     -- 1.3.6.1.4.1.8888.3.2
234     ::= { grMIBConf 2 }
235
236 grMIBBasicGroup OBJECT-GROUP
237     OBJECTS {
238         numeroEventos,
239         eventoIndex,
240         eventoName,
241         eventoEstado,
242         eventoTempoAnos,
243         eventoFrasePassado,
244         eventoFrasePresente,
245         eventoFraseFuturo,
246         eventoDuracao,

```

```
247     eventoTempoDias ,
248     eventoTempoHoras ,
249     eventoTempoMinutos ,
250     eventoTempoMeses ,
251     eventoTempoSemanas }
252 STATUS    current
253 DESCRIPTION
254     "Basic objects."
255     -- 1.3.6.1.4.1.8888.3.1.1 --
256     -- 1.3.6.1.4.1.8888.3.1.1
257     ::= { grMIBGroups 1 }
258
259 END
```