

ESTUDO E APLICAÇÃO DE VERIFICAÇÃO DE MODELOS EM SISTEMAS EMBARCADOS DESENVOLVIDOS NO IFMG - CAMPUS FORMIGA

Lucas Guimarães Bernardes¹; Gustavo Lobato Campos²; Mário Luiz Rodrigues Oliveira³;

1 Lucas Guimarães Bernardes, Bolsista IFMG, Ciência da Computação, IFMG Campus Formiga, Formiga - MG; lucasgbifmg@gmail.com

2 Gustavo Lobato Campos: Pesquisador do IFMG, Campus Formiga; gustavo.lobato@ifmg.edu.br

3 Mário Luiz Rodrigues Oliveira: Pesquisador do IFMG, Campus Formiga; mario.luiz@ifmg.edu.br

RESUMO

Os produtos de *software* e *hardware* desenvolvidos atualmente envolvem alto nível de complexidade tanto no seu processo de desenvolvimento quanto na sua lógica de operação. A abordagem tradicional para verificação e validação envolve realizar testes e simulações. Contudo tais abordagens não exploram todo o universo de possíveis estados do sistema e assim não são apropriadas para garantir que um produto de *software* ou *hardware* é isento de erros. A dependência cada vez maior da sociedade sobre esses sistemas evidencia a necessidade de estabilidade e segurança, seja para garantir que falhas não sejam catastróficas e afetem o funcionamento dos serviços que dependem desses sistemas, ou até mesmo para evitar a perda de vidas em casos mais extremos, como sistemas médicos de hospitais. Técnicas de métodos formais como verificação de modelos podem ser utilizadas em complemento às abordagens tradicionais para ajudar a garantir um sistema mais robusto e estável. Verificação de modelos é o problema de testar automaticamente e exaustivamente se um modelo que representa um sistema atende a uma dada especificação e pode ser aplicada a sistemas modelados de forma finita, tais como sistemas de *hardware* embarcados. Neste sentido o projeto de iniciação científica, aqui apresentado, realiza o estudo e aplicação da verificação de modelos em sistemas embarcados desenvolvidos no IFMG *Campus* Formiga, juntamente da capacitação de recurso humano na área de métodos formais, especificamente na técnica verificação de modelos e a validação dos sistemas analisados, através de estudos realizados nos assuntos envolvidos nas técnicas associadas aos métodos formais. Como resultado deste trabalho é obtida uma boa visão sobre as capacidades e limitações da técnica, possibilitando entender melhor os contextos nos quais ela pode ser melhor aproveitada juntamente com o entendimento dos processos de validação de produtos de *software* e *hardware* em geral.

INTRODUÇÃO:

À medida que a tecnologia evolui, sua presença no dia a dia se intensifica. Seja para auxiliar no tratamento médico ou para lidar com transações monetárias, os sistemas de *software* e *hardware* precisam de uma qualidade cada vez maior à medida que mais e mais subsistemas dependem da sua boa performance e execução. Porém, à medida que esses produtos se desenvolvem sua complexidade aumenta, tendo como consequência um maior índice de erros de desenvolvimento e execução. Devido a este problema, é necessária a utilização de metodologias de detecção e prevenção de erros que sejam completas e robustas (Christel Baier e Joost-Pieter Katoen, 2008).

Uma abordagem possível para tentar garantir a qualidade dos sistemas desenvolvidos é a utilização de testes e simulações de execução e ambientes de trabalho desses sistemas. Contudo essas abordagens não são capazes de garantir que não haja erros no sistema. Testes são feitos postulando-se uma hipótese sobre a execução do sistema e simulando uma execução do sistema em que esta hipótese ocorre para verificar se o comportamento do sistema corresponde ao esperado. O problema com esta metodologia é que as hipóteses sendo postuladas necessitam ser explicitamente especificadas. Como o número de comportamentos possível é muito grande estas hipóteses cobrem proporcionalmente um percentual muito pequeno das possibilidades de execução. Por exemplo, os projetistas do Ariane 5 proveram o sistema de tolerância a falhas de *hardware* através de módulos duplicados de cálculo, mas não previram que se o *software* causasse o erro o mesmo seria calculado por ambos módulos ao mesmo tempo (Gleick, 1996 e Lions, 1996). Uma solução alternativa e complementar ao uso de testes e simulações é a adoção de métodos formais.

Dentre as técnicas envolvidas nos métodos formais encontra-se a Verificação de Modelos. Diferente de testes simples, a Verificação de Modelos trabalha com a construção de implementações que representam todo o comportamento do sistema em questão nos mínimos detalhes. Acompanhada de todo o rigor matemático fortemente presente nos Métodos Formais, essa técnica é capaz de mostrar que o sistema é isento de erros ou apontar em que parte(s) do sistema há erros.

O presente projeto tem por objetivo principal o estudo e a aplicação da técnica de verificação de modelos em projetos de sistemas embarcados desenvolvidos no âmbito do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais (IFMG) *Campus Formiga*.

METODOLOGIA:

O projeto aqui apresentado pode ser dividido em duas grandes etapas, sendo essas a de estudos e pesquisas, bem como a de implementação e finalização das tarefas. Cada uma dessas etapas teve seu tempo de desenvolvimento estendido por um período de 5 meses, sendo subdivididas em etapas menores que podem ser visualizadas na Tabela 1.

Mês	Etapas do desenvolvimento do projeto
1	Estudar teoria a respeito de métodos formais
2	Estudar teoria de lógica temporal
3	Estudar as técnicas de verificação de modelos e verificação de modelos simbólica
4	Estudar e compreender a linguagem SMV
5	Estudar e compreender a ferramenta NuSMV. Selecionar os projetos de sistemas embarcados que serão verificados
6	Realizar a verificação de modelos nos projetos selecionados na etapa 6 usando o software NuSMV – Fase 1
7	Realizar a verificação de modelos nos projetos selecionados na etapa 6 usando o software NuSMV – Fase 2
8	Realizar a verificação de modelos nos projetos selecionados na etapa 6 usando o software NuSMV – Fase 3
9	Analisar os resultados da aplicação de verificação de modelos obtidos nas etapas 6, 7 e 8
10	Elaborar o relatório final

Tabela 1 – Etapas do desenvolvimento do projeto

Na primeira metade do período de desenvolvimento deste projeto os esforços se focaram no estudo e pesquisa das teorias, tecnologias e ferramentas por trás das técnicas de Verificação de Modelos, destacando-se os estudos de Métodos Formais e Lógica Temporal. Devido a isso, a maior parte do tempo foi utilizado na leitura de tutoriais e livros sobre estes assuntos. Dentre os materiais utilizados, K. L. McMillan (1998, c2000, c2003) e Joost-Pieter Katoen (2008) destacam-se como os principais autores utilizados.

Aproximando-se do final da primeira metade do projeto foram iniciados estudos e análises a partir de códigos reais de exemplos que foram previamente construídos na ferramenta NuSMV (FBK, 2010), essa que seria utilizada nas etapas de desenvolvimento, com o objetivo de auxiliar nos estudos teóricos e facilitando a absorção do conteúdo.

Após finalizada a etapa de estudos e pesquisas foram iniciados os trabalhos de desenvolvimento previstos para o projeto. Tais trabalhos consistiram nos processos de análise, tradução e verificação de todos os projetos de sistemas embarcados selecionados para serem verificados pelo processo de verificação de modelos.

Na segunda etapa as atividades de estudo se transformaram em atividades de aplicação dos conhecimentos adquiridos. O maior foco se deu na implementação e verificação dos projetos selecionados como alvos das técnicas.

Para os desenvolvimentos foram utilizados dois projetos de sistemas embarcados como base na construção de códigos NuSMV e verificação de comportamentos e validação das implementações. Ambos os projetos, sendo esses o “Gestão Hídrica” (MELO, Gabriel. SILVA, Heuler. 2019) e um “Sistema para Controle de Granja”, são soluções que foram construídas por alunos do curso de Engenharia Elétrica do Instituto Federal de Minas Gerais – *Campus Formiga* e pesquisadores do GSE (Grupo de Soluções em Engenharia).

Vale destacar que ambos os projetos selecionados para testes já foram finalizados e entregues, dessa forma as atividades desenvolvidas nesse trabalho não serão utilizadas para validação da correção de ambos os projetos quando aplicados na prática. A seleção de ambos se deu puramente com o objetivo da obtenção de uma mídia na qual fosse possível aplicar os conhecimentos adquiridos durante os estudos, possibilitando o aprendizado através da execução, sem nenhuma premissa de se utilizar os resultados adquiridos das validações para refinar os projetos originais.

Durante todo o período de atividades do projeto foram realizadas as construções de relatórios de acompanhamento mensal, estes que tinham por objetivo consolidar e relatar o aprendizado obtido na etapa, bem como reportar o estado em que se encontrava o progresso do desenvolvimento das atividades.

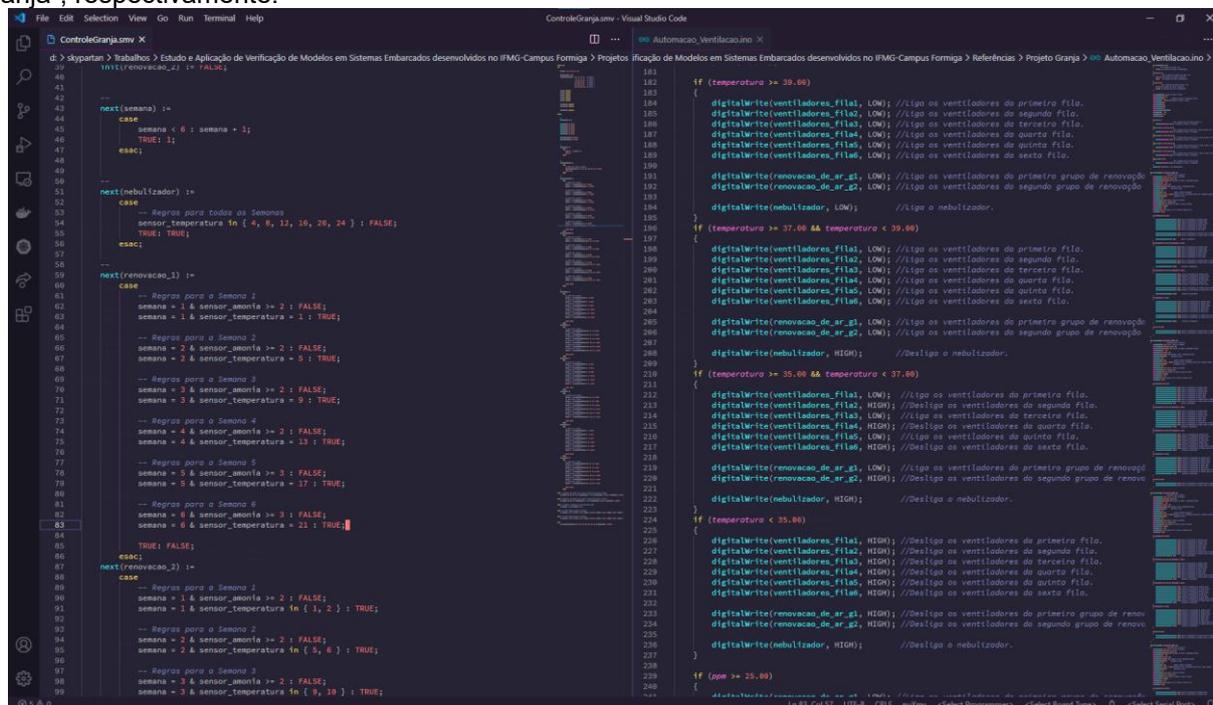
RESULTADOS E DISCUSSÕES:

Com as atividades desenvolvidas tendo grande foco na aprendizagem e introdução aos conceitos das técnicas de verificação de modelos, bem como na inicialização de uso de suas ferramentas, foi possível a construção de uma base forte de conhecimento que abre portas para a realização de trabalhos futuros que possam se basear ou inspirar nas atividades realizadas nesse.

As atividades de implementação foram o maior demonstrador das capacidades da técnica, apresentando tanto suas limitações quanto suas aplicabilidades. A partir das conversões dos projetos de sistemas embarcados selecionados foram obtidos diversos resultados a respeito de seus comportamentos, bem como das operações das ferramentas de Verificação de Modelos, resultados estes que foram utilizados como mecanismos de aprendizagem.

Nas verificações dos projetos, juntamente do processo de tradução, foi possível perceber as limitações e falhas de ambos projetos “Gestão Hídrica” e “Sistema de Controle de Granja”. Enquanto o “Gestão Hídrica” realizava o trabalho especificado de maneira correta sua especificação não era completa, apresentando certas lacunas como para o caso de falhas no fornecimento de água. Já o “Sistema de Controle de Granja” apresentou um problema que, apesar de descoberto nas etapas de análise do código original e especificações, não pode ser verificado pelo NuSMV, já que este se tratava de um problema de natureza física, isto é, na comunicação eletrônica entre os módulos do projeto que causa uma variação na contagem temporal do sistema, característica que possui um nível de abstração muito baixo para ser possível representá-la no NuSMV de forma eficiente.

Os resultados desses trabalhos de implementação podem ser visualizados nas Figuras 1 e 2, que apresentam parte dos códigos originais e traduzidos dos projetos “Gestão Hídrica” e “Sistema de Controle de Granja”, respectivamente.



```

41 next(semana) :=
42   case
43     semana < 6 : semana = 1;
44     TRUE : 1;
45   esac;
46
47   next(nebulizador) :=
48     case
49       -- Regras para todos os Semanas
50       sensor_temperatura in { 4, 9, 12, 16, 20, 24 } : FALSE;
51       TRUE : TRUE;
52     esac;
53
54   next(renovacao_1) :=
55     case
56       -- Regras para a Semana 1
57       semana = 1 & sensor_semidia >= 2 : FALSE;
58       semana = 1 & sensor_temperatura = 1 : TRUE;
59     esac;
60
61     -- Regras para a Semana 2
62     semana = 2 & sensor_semidia >= 2 : FALSE;
63     semana = 2 & sensor_temperatura = 5 : TRUE;
64
65     -- Regras para a Semana 3
66     semana = 3 & sensor_semidia >= 2 : FALSE;
67     semana = 3 & sensor_temperatura = 9 : TRUE;
68
69     -- Regras para a Semana 4
70     semana = 4 & sensor_semidia >= 2 : FALSE;
71     semana = 4 & sensor_temperatura = 13 : TRUE;
72
73     -- Regras para a Semana 5
74     semana = 5 & sensor_semidia >= 3 : FALSE;
75     semana = 5 & sensor_temperatura = 17 : TRUE;
76
77     -- Regras para a Semana 6
78     semana = 6 & sensor_semidia >= 3 : FALSE;
79     semana = 6 & sensor_temperatura = 21 : TRUE;
80
81   esac;
82   next(renovacao_2) :=
83     case
84       -- Regras para a Semana 1
85       semana = 1 & sensor_semidia >= 2 : FALSE;
86       semana = 1 & sensor_temperatura in { 1, 2 } : TRUE;
87
88       -- Regras para a Semana 2
89       semana = 2 & sensor_semidia >= 2 : FALSE;
90       semana = 2 & sensor_temperatura in { 5, 6 } : TRUE;
91
92       -- Regras para a Semana 3
93       semana = 3 & sensor_semidia >= 2 : FALSE;
94       semana = 3 & sensor_temperatura in { 9, 10 } : TRUE;
95
96       -- Regras para a Semana 4
97       semana = 4 & sensor_semidia >= 2 : FALSE;
98       semana = 4 & sensor_temperatura in { 13, 14 } : TRUE;
99
100      -- Regras para a Semana 5
101      semana = 5 & sensor_semidia >= 3 : FALSE;
102      semana = 5 & sensor_temperatura in { 17, 18 } : TRUE;
103
104      -- Regras para a Semana 6
105      semana = 6 & sensor_semidia >= 3 : FALSE;
106      semana = 6 & sensor_temperatura in { 21, 22 } : TRUE;
107
108    esac;
109
110    if (temperatura >= 39.00)
111      {
112        digitalWrite(ventiladores_filas, LOW); //Ligo os ventiladores da primeira fila.
113        digitalWrite(ventiladores_filas, LOW); //Ligo os ventiladores da segunda fila.
114        digitalWrite(ventiladores_filas, LOW); //Ligo os ventiladores da terceira fila.
115        digitalWrite(ventiladores_filas, LOW); //Ligo os ventiladores da quarta fila.
116        digitalWrite(ventiladores_filas, LOW); //Ligo os ventiladores da quinta fila.
117        digitalWrite(ventiladores_filas, LOW); //Ligo os ventiladores da sexta fila.
118
119        digitalWrite(renovacao_de_ar_g1, LOW); //Ligo os ventiladores do primeiro grupo de renovação
120        digitalWrite(renovacao_de_ar_g2, LOW); //Ligo os ventiladores do segundo grupo de renovação
121
122        digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
123      }
124
125      if (temperatura >= 37.00 && temperatura < 39.00)
126      {
127        digitalWrite(ventiladores_filas, LOW); //Ligo os ventiladores da primeira fila.
128        digitalWrite(ventiladores_filas, LOW); //Ligo os ventiladores da segunda fila.
129        digitalWrite(ventiladores_filas, LOW); //Ligo os ventiladores da terceira fila.
130        digitalWrite(ventiladores_filas, LOW); //Ligo os ventiladores da quarta fila.
131        digitalWrite(ventiladores_filas, LOW); //Ligo os ventiladores da quinta fila.
132        digitalWrite(ventiladores_filas, LOW); //Ligo os ventiladores da sexta fila.
133
134        digitalWrite(renovacao_de_ar_g1, LOW); //Ligo os ventiladores do primeiro grupo de renovação
135        digitalWrite(renovacao_de_ar_g2, LOW); //Ligo os ventiladores do segundo grupo de renovação
136
137        digitalWrite(nebulizador, HIGH); //Desligo o nebulizador.
138      }
139
140      if (temperatura >= 35.00 && temperatura < 37.00)
141      {
142        digitalWrite(ventiladores_filas, HIGH); //Desligo os ventiladores da primeira fila.
143        digitalWrite(ventiladores_filas, HIGH); //Desligo os ventiladores da segunda fila.
144        digitalWrite(ventiladores_filas, HIGH); //Desligo os ventiladores da terceira fila.
145        digitalWrite(ventiladores_filas, HIGH); //Desligo os ventiladores da quarta fila.
146        digitalWrite(ventiladores_filas, LOW); //Ligo os ventiladores da quinta fila.
147        digitalWrite(ventiladores_filas, HIGH); //Desligo os ventiladores da sexta fila.
148
149        digitalWrite(renovacao_de_ar_g1, LOW); //Ligo os ventiladores do primeiro grupo de renovação
150        digitalWrite(renovacao_de_ar_g2, LOW); //Ligo os ventiladores do segundo grupo de renovação
151
152        digitalWrite(nebulizador, HIGH); //Desligo o nebulizador.
153      }
154
155      if (temperatura < 35.00)
156      {
157        digitalWrite(ventiladores_filas, HIGH); //Desligo os ventiladores da primeira fila.
158        digitalWrite(ventiladores_filas, HIGH); //Desligo os ventiladores da segunda fila.
159        digitalWrite(ventiladores_filas, HIGH); //Desligo os ventiladores da terceira fila.
160        digitalWrite(ventiladores_filas, HIGH); //Desligo os ventiladores da quarta fila.
161        digitalWrite(ventiladores_filas, HIGH); //Desligo os ventiladores da quinta fila.
162        digitalWrite(ventiladores_filas, HIGH); //Desligo os ventiladores da sexta fila.
163
164        digitalWrite(renovacao_de_ar_g1, HIGH); //Desligo os ventiladores do primeiro grupo de renovação
165        digitalWrite(renovacao_de_ar_g2, HIGH); //Desligo os ventiladores do segundo grupo de renovação
166
167        digitalWrite(nebulizador, HIGH); //Desligo o nebulizador.
168      }
169
170      if (opp >= 25.00)
171      {
172        digitalWrite(ventiladores_filas, LOW); //Ligo os ventiladores da primeira fila.
173        digitalWrite(ventiladores_filas, LOW); //Ligo os ventiladores da segunda fila.
174        digitalWrite(ventiladores_filas, LOW); //Ligo os ventiladores da terceira fila.
175        digitalWrite(ventiladores_filas, LOW); //Ligo os ventiladores da quarta fila.
176        digitalWrite(ventiladores_filas, LOW); //Ligo os ventiladores da quinta fila.
177        digitalWrite(ventiladores_filas, LOW); //Ligo os ventiladores da sexta fila.
178
179        digitalWrite(renovacao_de_ar_g1, LOW); //Ligo os ventiladores do primeiro grupo de renovação
180        digitalWrite(renovacao_de_ar_g2, LOW); //Ligo os ventiladores do segundo grupo de renovação
181
182        digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
183      }
184
185      if (opp < 25.00)
186      {
187        digitalWrite(ventiladores_filas, HIGH); //Desligo os ventiladores da primeira fila.
188        digitalWrite(ventiladores_filas, HIGH); //Desligo os ventiladores da segunda fila.
189        digitalWrite(ventiladores_filas, HIGH); //Desligo os ventiladores da terceira fila.
190        digitalWrite(ventiladores_filas, HIGH); //Desligo os ventiladores da quarta fila.
191        digitalWrite(ventiladores_filas, HIGH); //Desligo os ventiladores da quinta fila.
192        digitalWrite(ventiladores_filas, HIGH); //Desligo os ventiladores da sexta fila.
193
194        digitalWrite(renovacao_de_ar_g1, HIGH); //Desligo os ventiladores do primeiro grupo de renovação
195        digitalWrite(renovacao_de_ar_g2, HIGH); //Desligo os ventiladores do segundo grupo de renovação
196
197        digitalWrite(nebulizador, HIGH); //Desligo o nebulizador.
198      }
199
200      digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
201    }
202  }
203
204  if (opp >= 25.00)
205  {
206    digitalWrite(ventiladores_filas, LOW); //Ligo os ventiladores da primeira fila.
207    digitalWrite(ventiladores_filas, LOW); //Ligo os ventiladores da segunda fila.
208    digitalWrite(ventiladores_filas, LOW); //Ligo os ventiladores da terceira fila.
209    digitalWrite(ventiladores_filas, LOW); //Ligo os ventiladores da quarta fila.
210    digitalWrite(ventiladores_filas, LOW); //Ligo os ventiladores da quinta fila.
211    digitalWrite(ventiladores_filas, LOW); //Ligo os ventiladores da sexta fila.
212
213    digitalWrite(renovacao_de_ar_g1, LOW); //Ligo os ventiladores do primeiro grupo de renovação
214    digitalWrite(renovacao_de_ar_g2, LOW); //Ligo os ventiladores do segundo grupo de renovação
215
216    digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
217  }
218
219  if (opp < 25.00)
220  {
221    digitalWrite(ventiladores_filas, HIGH); //Desligo os ventiladores da primeira fila.
222    digitalWrite(ventiladores_filas, HIGH); //Desligo os ventiladores da segunda fila.
223    digitalWrite(ventiladores_filas, HIGH); //Desligo os ventiladores da terceira fila.
224    digitalWrite(ventiladores_filas, HIGH); //Desligo os ventiladores da quarta fila.
225    digitalWrite(ventiladores_filas, HIGH); //Desligo os ventiladores da quinta fila.
226    digitalWrite(ventiladores_filas, HIGH); //Desligo os ventiladores da sexta fila.
227
228    digitalWrite(renovacao_de_ar_g1, HIGH); //Desligo os ventiladores do primeiro grupo de renovação
229    digitalWrite(renovacao_de_ar_g2, HIGH); //Desligo os ventiladores do segundo grupo de renovação
230
231    digitalWrite(nebulizador, HIGH); //Desligo o nebulizador.
232  }
233
234  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
235
236  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
237
238  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
239
240  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
241
242  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
243
244  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
245
246  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
247
248  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
249
250  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
251
252  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
253
254  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
255
256  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
257
258  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
259
260  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
261
262  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
263
264  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
265
266  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
267
268  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
269
270  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
271
272  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
273
274  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
275
276  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
277
278  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
279
280  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
281
282  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
283
284  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
285
286  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
287
288  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
289
290  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
291
292  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
293
294  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
295
296  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
297
298  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
299
300  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
301
302  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
303
304  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
305
306  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
307
308  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
309
310  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
311
312  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
313
314  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
315
316  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
317
318  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
319
320  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
321
322  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
323
324  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
325
326  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
327
328  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
329
330  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
331
332  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
333
334  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
335
336  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
337
338  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
339
340  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
341
342  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
343
344  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
345
346  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
347
348  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
349
350  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
351
352  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
353
354  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
355
356  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
357
358  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
359
360  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
361
362  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
363
364  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
365
366  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
367
368  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
369
370  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
371
372  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
373
374  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
375
376  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
377
378  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
379
380  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
381
382  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
383
384  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
385
386  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
387
388  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
389
390  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
391
392  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
393
394  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
395
396  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
397
398  digitalWrite(nebulizador, LOW); //Ligo o nebulizador.
399
399

```

Figura 1 – Parte da implementação do projeto Gestão Hídrica

Christel Baier, Joost-Pieter Katoen. Principles of Model Checking. [S. l.], 2008.

FBK. NuSMV. NuSMV: a new symbolic model checker. Disponível em: <http://nusmv.fbk.eu/>. Acesso em: 1/07/2020

Gleick, James. A Bug and a Crash. <http://www.around.com/ariane.html>, 1996. Acesso em 14 fevereiro de 2020.

K. L. McMillan, The SMV system. Carnegie Mellon University, 2020. <<https://www.cs.cmu.edu/~emc/15817-f09/cmu-smv-manual.pdf>> Acesso em 20/06/2020.

K. L. McMillan. Kansas State University, 2020.<<http://santos.cs.ksu.edu/smv-doc/language/language.html>> Acesso em 15/06/2020.

K. L. McMillan. Universidade Federal de Minas Gerais, 2020. <<https://homepages.dcc.ufmg.br/~jefchaves/smv/tutorial.html>> Acesso em 15/06/2020.

Lions, J. L. ARIANE 5 Flight 501 Failure. <http://sunnyday.mit.edu/accidents/Ariane5accidentreport.html>, 1996. Acesso em 14 fevereiro de 2020.

MELO, Gabriel. SILVA, Heuler. 2019. Desenvolvimento de protótipo de sistema de abastecimento de água automatizado.