



UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO  
CAMPUS PAU DOS FERROS  
DEPARTAMENTO DE ENGENHARIAS E TECNOLOGIA  
BACHARELADO EM INTERDISCIPLINAR EM TECNOLOGIA DA INFORMAÇÃO

TÁSSIO FERNANDES COSTA

**UM MODELO DE REFERÊNCIA PARA SISTEMAS DE BOMBA DE INFUSÃO DE  
INSULINA**

PAU DOS FERROS – RN

2019

TÁSSIO FERNANDES COSTA

**UM MODELO DE REFERÊNCIA PARA SISTEMAS DE BOMBA DE INFUSÃO DE  
INSULINA**

Monografia apresentada à Universidade Federal Rural do Semi-Árido - UFERSA, Campus Pau dos Ferros para a obtenção do título de Bacharel em Interdisciplinar em Tecnologia da Informação.

Orientador: Prof. Dr. Álvaro Alvares de Carvalho César Sobrinho.

Co-Orientador: Prof. Dr. Lenardo Chaves e Silva.

PAU DOS FERROS – RN

2019

© Todos os direitos estão reservados a Universidade Federal Rural do Semi-Árido. O conteúdo desta obra é de inteira responsabilidade do (a) autor (a), sendo o mesmo, passível de sanções administrativas ou penais, caso sejam infringidas as leis que regulamentam a Propriedade Intelectual, respectivamente, Patentes: Lei nº 9.279/1996 e Direitos Autorais: Lei nº 9.610/1998. O conteúdo desta obra tomar-se-á de domínio público após a data de defesa e homologação da sua respectiva ata. A mesma poderá servir de base literária para novas pesquisas, desde que a obra e seu (a) respectivo (a) autor (a) sejam devidamente citados e mencionados os seus créditos bibliográficos.

C837m Costa, Tássio Fernandes.  
Um modelo de referência para sistemas de bomba  
de infusão de insulina / Tássio Fernandes Costa. -  
2019.  
68 f. : il.

Orientador: Álvaro Alvares de Carvalho César  
Sobrinho.  
Coorientador: Lenardo Chaves e Silva.  
Monografia (graduação) - Universidade Federal  
Rural do Semi-árido, Curso de Tecnologia da  
Informação, 2019.

1. Modelagem formal. 2. Validação. 3.  
Verificação. 4. Eficácia. 5. Segurança. I.  
Sobrinho, Álvaro Alvares de Carvalho César,  
orient. II. Silva, Lenardo Chaves e, co-orient.  
III. Título.

O serviço de Geração Automática de Ficha Catalográfica para Trabalhos de Conclusão de Curso (TCC's) foi desenvolvido pelo Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (USP) e gentilmente cedido para o Sistema de Bibliotecas da Universidade Federal Rural do Semi-Árido (SISBI-UFERSA), sendo customizado pela Superintendência de Tecnologia da Informação e Comunicação (SUTIC) sob orientação dos bibliotecários da instituição para ser adaptado às necessidades dos alunos dos Cursos de Graduação e Programas de Pós-Graduação da Universidade.

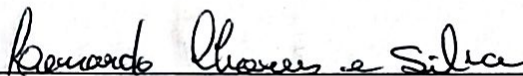
TÁSSIO FERNANDES COSTA

**UM MODELO DE REFERÊNCIA PARA SISTEMAS DE BOMBA DE INFUSÃO DE  
INSULINA**

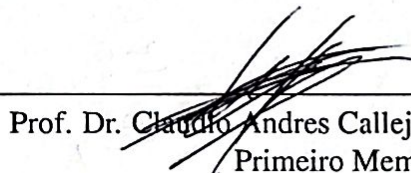
Monografia apresentada à Universidade Federal Rural do Semi-Árido - UFERSA, Campus Pau dos Ferros para a obtenção do título de Bacharel em Interdisciplinar em Tecnologia da Informação.

Aprovada em 20/03/2019

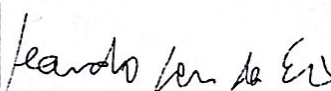
**BANCA EXAMINADORA**



Prof. Dr. Lenardo Chaves e Silva (UFERSA)  
Presidente



Prof. Dr. Claudio Andres Callejas Olguin (UFERSA)  
Primeiro Membro



Prof. Dr. Leandro Dias da Silva (UFAL)  
Segundo Membro

Aos Meus Pais, Irmãos, Tios(as), Namorada,  
Amigos e Professores(as)!

## **AGRADECIMENTOS**

Agradeço a Javé por ter me concedido saúde e paz durante toda a caminhada acadêmica, me possibilitando alcançar mais um objetivo de vida: o título de Bacharel em Tecnologia da Informação. Que a sua poderosa e portentosa mão continue sempre estendida ao nosso favor, nos protegendo de todo mal e nos permitindo alçar voos maiores.

Agradeço ao meu orientador, Professor Dr. Álvaro Alvares, pelas brilhantes contribuições durante toda a construção desse trabalho, o qual não seria possível sem a sua incomensurável ajuda. Obrigado pela compreensão, pela amizade, pelas críticas construtivas, pelo incentivo e motivação para fazer um trabalho de qualidade, pela disponibilidade e pela enorme paciência no esclarecimento de todas as dúvidas. Enfim, muito obrigado por me ensinar a fazer ciência de qualidade. Tens a minha admiração e o meu respeito!

Agradeço ao meu co-orientador, Professor Dr. Lenardo, pelo apoio e por todas as contribuições dadas, as quais indiscutivelmente agregaram muita qualidade a esse trabalho. Contar com a sua amizade e os seus preciosos ensinamentos é motivo de muita honra. Parabéns pelo zelo que tens ao fazer ciência, isso explica o porquê és um grande mestre. Conte também com a minha admiração e o meu respeito!

Agradeço ao amigo Josaias Moura pelo apoio incondicional durante a etapa de correções do trabalho ocorrida após a defesa. Muito obrigado pela disponibilidade e pela atenção!

Gostaria de agradecer também a todos os funcionários e servidores da UFERSA, desde os terceirizados até o diretor do Campus Pau dos Ferros - RN, os quais, de alguma forma, no exercício de suas atribuições contribuem para a nossa formação como cidadão e como profissional.

Quero agradecer aos meus pais, Geraldo Fernandes Costa e Auxiliadora Maria da Costa, que diante de suas limitações sempre me apoiaram emocionalmente e financeiramente, me incentivando sempre a buscar o conhecimento como meio para conquistar um futuro mais promissor. Aos meus irmãos, Tiago, Gracinha, Talison, Talismar, Mateus e Débora, muito obrigado por todo o apoio. Amo todos vocês!

A minha namorada, Lilliane Domingos, quero agradecer pela paciência e compreensão diante

das minhas muitas ausências após ter ingressado na graduação. A escolha por fazer o ensino superior implicou em um distanciamento entre nós, porém, acredito que essa é uma etapa da minha vida na qual estou construindo algo melhor para o nosso futuro. A vida é repleta de momentos e o nosso momento de estarmos mais juntos um dia chegará, precisamos ter paciência e saber aguardar o momento certo, vivendo uma etapa de cada vez. Te amo!

Por último e não menos importante, agradeço a tia Rosa, a tia Dorotéia e a tio Francisco Fernandes, os quais têm me ajudado muito durante essa caminhada em busca da formação superior. A ajuda de vocês tem agregado mais leveza e conforto a esse caminho. Em nome de vocês, estendo o meu agradecimento a todos os meus demais familiares.

## RESUMO

Sistemas médicos críticos de segurança são sistemas cuja falha pode resultar em danos ao paciente, podendo até mesmo levá-lo ao óbito. Neste contexto, o principal requisito de qualidade neste tipo de sistema é a segurança. Um sistema de controle de bomba de infusão de insulina é um exemplo de sistema que se encaixa nessa categoria. Neste trabalho é proposto um modelo genérico de sistemas de bomba de infusão de insulina, o qual foi especificado usando Redes de Petri coloridas (*Coloured Petri Nets - CPN*) com o auxílio da ferramenta *CPN/Tools*. O modelo foi estendido em um estudo de caso sobre o equipamento *ACCU-CHECK Spirit* versão 2.XX, tendo sido validado via simulação e verificado formalmente com a técnica de verificação automática de modelos (*Model Checking*). Portanto, a principal contribuição com este trabalho é a disponibilização de um modelo genérico formal que pode ser reutilizado durante o processo de certificação de equipamentos e de um estudo de caso sobre um equipamento desenvolvido por uma empresa reconhecida na indústria.

**Palavras-chave:** Segurança. Eficácia. Verificação. Validação. Modelagem formal.



## ABSTRACT

Critical safety medical systems are systems whose failure can result in harm to the patient, and can even lead to death. In this context, the main quality requirement in this type of system is safety. An insulin infusion pump control system is an example of a system that fits into this category. In this work a generic model of insulin infusion pump systems was proposed, which was specified using Colored Petri Nets (CPN) with the help of the CPN/Tools tool. The model was extended in a case study on ACCU-CHECK Spirit version 2.XX, having been validated via simulation and formally verified with the Model Checking technique. Therefore, the main contribution with this work is the provision of a formal generic model that can be reused during the equipment certification process and a case study on an equipment developed by a recognized company in the industry.

**Keywords:** Safety. Efficiency. Verification. Validation. Formal modeling.

## LISTA DE ILUSTRAÇÕES

|                                                                                                                                             |    |
|---------------------------------------------------------------------------------------------------------------------------------------------|----|
| Figura 1 – Modelo de atividade da bomba de infusão de insulina. . . . .                                                                     | 22 |
| Figura 2 – Elementos gráficos da rede de petri . . . . .                                                                                    | 25 |
| Figura 3 – Problema do leitor-escritor . . . . .                                                                                            | 25 |
| Figura 4 – Sequência . . . . .                                                                                                              | 26 |
| Figura 5 – Simultaneidade . . . . .                                                                                                         | 27 |
| Figura 6 – Paralelismo e sincronização . . . . .                                                                                            | 27 |
| Figura 7 – Decisão . . . . .                                                                                                                | 27 |
| Figura 8 – Compartilhamento de recurso . . . . .                                                                                            | 28 |
| Figura 9 – Estrutura do verificador de modelo . . . . .                                                                                     | 31 |
| Figura 10 – Modelo <i>CPN</i> de visão geral do sistema de controle da bomba de infusão de<br>insulina da primeira versão . . . . .         | 38 |
| Figura 11 – Módulo de configuração da primeira versão . . . . .                                                                             | 39 |
| Figura 12 – Módulo de configuração do modo padrão da primeira versão . . . . .                                                              | 39 |
| Figura 13 – Módulo de configuração do modo personalizado da primeira versão . . . . .                                                       | 41 |
| Figura 14 – Módulo de aplicação de insulina do modo padrão da primeira versão . . . . .                                                     | 42 |
| Figura 15 – Módulo de aplicação de insulina do modo personalizado da primeira versão . . . . .                                              | 43 |
| Figura 16 – Módulo de aplicação de insulina do modo padrão da segunda versão . . . . .                                                      | 47 |
| Figura 17 – Propriedade 1 . . . . .                                                                                                         | 50 |
| Figura 18 – Propriedade 2 . . . . .                                                                                                         | 52 |
| Figura 19 – Captura de tela 1 do módulo de configuração do modo personalizado do<br><i>ACCU-CHECK Spirit</i> versão 2.XX . . . . .          | 54 |
| Figura 20 – Captura de tela 2 do módulo de configuração do modo personalizado do<br><i>ACCU-CHECK Spirit</i> versão 2.XX . . . . .          | 55 |
| Figura 21 – Captura de tela 1 do módulo de aplicação de insulina do modo personalizado<br>do <i>ACCU-CHECK Spirit</i> versão 2.XX . . . . . | 56 |
| Figura 22 – Captura de tela 2 do módulo de aplicação de insulina do modo personalizado<br>do <i>ACCU-CHECK Spirit</i> versão 2.XX . . . . . | 58 |
| Figura 23 – Captura de tela 3 do módulo de aplicação de insulina do modo personalizado<br>do <i>ACCU-CHECK Spirit</i> versão 2.XX . . . . . | 59 |
| Figura 24 – Captura de tela 4 do módulo de aplicação de insulina do modo personalizado<br>do <i>ACCU-CHECK Spirit</i> versão 2.XX . . . . . | 61 |

Figura 25 – Captura de tela 5 do módulo de aplicação de insulina do modo personalizado  
do *ACCU-CHECK Spirit* versão 2.XX . . . . . 62

## LISTA DE TABELAS

|                                                                                                                     |    |
|---------------------------------------------------------------------------------------------------------------------|----|
| Tabela 2 – Requisitos técnicos chave da bomba de infusão de insulina . . . . .                                      | 23 |
| Tabela 3 – Resumo dos trabalhos relacionados . . . . .                                                              | 33 |
| Tabela 4 – Parâmetros de entrada da primeira versão do modelo genérico de bomba de<br>infusão de insulina . . . . . | 45 |
| Tabela 5 – Parâmetros de entrada da segunda versão do modelo genérico de bomba de<br>infusão de insulina . . . . .  | 46 |
| Tabela 6 – Requisitos técnicos do equipamento <i>ACCU-CHECK Spirit</i> versão 2.XX . .                              | 48 |
| Tabela 7 – Propriedades de segurança do <i>ACCU-CHECK Spirit</i> versão 2.XX . . . . .                              | 49 |
| Tabela 8 – Parâmetros de entrada da segunda versão do <i>ACCU-CHECK Spirit</i> versão 2.XX                          | 53 |

## **LISTA DE ABREVIATURAS E SIGLAS**

|        |                                          |
|--------|------------------------------------------|
| ANVISA | Agência Nacional de Vigilância Sanitária |
| CPN    | Coloured Petri Nets                      |
| CTL    | Computation Tree Logics                  |
| DES    | Discrete Event System                    |
| DNA    | deoxyribonucleic acid                    |
| ECG    | eletrocardiografia                       |
| EGG    | surface electrogastrography              |
| FDA    | Food And Drug Administration             |
| GIIP   | Generic Insulin Infusion Pump            |
| GPCA   | Generic Patient Controlled Analgesic     |
| PVS    | Prototype Verification System            |
| SMFC   | sistemas médicos físico-cibernéticos     |
| UML    | Unified Modeling Language                |

## SUMÁRIO

|              |                                                                |           |
|--------------|----------------------------------------------------------------|-----------|
| <b>1</b>     | <b>INTRODUÇÃO . . . . .</b>                                    | <b>14</b> |
| 1.1          | PROBLEMATIZAÇÃO . . . . .                                      | 16        |
| 1.2          | OBJETIVOS . . . . .                                            | 17        |
| 1.3          | JUSTIFICATIVA . . . . .                                        | 18        |
| 1.4          | METODOLOGIA . . . . .                                          | 19        |
| 1.5          | ORGANIZAÇÃO DO DOCUMENTO . . . . .                             | 20        |
| <b>2</b>     | <b>REFERENCIAL TEÓRICO . . . . .</b>                           | <b>21</b> |
| 2.1          | BOMBA DE INFUSÃO DE INSULINA . . . . .                         | 21        |
| 2.2          | REDES DE PETRI COLORIDAS . . . . .                             | 24        |
| 2.3          | VERIFICAÇÃO AUTOMÁTICA DE MODELOS . . . . .                    | 31        |
| <b>3</b>     | <b>TRABALHOS RELACIONADOS . . . . .</b>                        | <b>33</b> |
| 3.1          | BOMBA DE INSULINA . . . . .                                    | 33        |
| 3.2          | REDES DE PETRI COLORIDAS . . . . .                             | 34        |
| 3.3          | MODELOS DE REFERÊNCIA . . . . .                                | 35        |
| <b>4</b>     | <b>ESPECIFICAÇÃO DA BOMBA DE INFUSÃO DE INSULINA . . . . .</b> | <b>37</b> |
| 4.1          | PRIMEIRA VERSÃO DO MODELO . . . . .                            | 37        |
| 4.2          | SEGUNDA VERSÃO DO MODELO . . . . .                             | 44        |
| <b>5</b>     | <b>ESTUDO DE CASO . . . . .</b>                                | <b>48</b> |
| 5.1          | VERIFICAÇÃO DA PRIMEIRA VERSÃO DO MODELO . . . . .             | 49        |
| <b>5.1.1</b> | <b>Propriedade 1 . . . . .</b>                                 | <b>49</b> |
| <b>5.1.2</b> | <b>Propriedade 2 . . . . .</b>                                 | <b>51</b> |
| 5.2          | VALIDAÇÃO DA SEGUNDA VERSÃO DO MODELO . . . . .                | 52        |
| <b>6</b>     | <b>CONCLUSÃO E TRABALHOS FUTUROS . . . . .</b>                 | <b>63</b> |
|              | <b>REFERÊNCIAS . . . . .</b>                                   | <b>65</b> |

## 1 INTRODUÇÃO

Existem diversas classes de sistemas computacionais, tais como os sistemas comerciais (por exemplo, os sistemas de comércio eletrônico chamados de *e-commerce*) e os sistemas críticos, os quais são utilizados nas mais diversas áreas profissionais (por exemplo, saúde, economia e infraestrutura). O foco com este trabalho está no projeto de sistemas críticos.

Sistema crítico é “um sistema computacional cuja falha pode resultar em perdas econômicas, humanas ou ambientais significativas” (SOMMERVILLE, 2011, p. 59). Ainda de acordo com o autor, esses sistemas podem ser categorizados em três tipos: sistema crítico de segurança, sistema de missão crítica e sistema crítico de negócio.

Em particular, os sistemas críticos de segurança são aqueles cuja falha pode resultar em danos ambientais ou perda da vida humana. Tais sistemas são utilizados em diversas áreas, como, por exemplo, medicina. Na área médica, pode-se manipular dispositivos médicos embarcados. Estes dispositivos demandam segurança e eficácia, conforme Li *et al.* (2014). De acordo com a *Food And Drug Administration (FDA)*<sup>1</sup>, um sistema é seguro quando não oferece riscos que afetam a integridade física das pessoas e é eficaz quando oferece resultados clinicamente significativos. Devido a sua natureza, esses sistemas devem obedecer a rigorosos critérios estabelecidos por órgãos governamentais, como, por exemplo, a Agência Nacional de Vigilância Sanitária (ANVISA)<sup>2</sup> e a *FDA*.

Os sistemas médicos críticos de segurança podem ser classificados em três tipos: monitoramento (capta e/ou verifica biomarcadores do paciente), diagnóstico (identificação de uma doença) e tratamento (atua executando ações predefinidas no paciente). O dispositivo abordado por Sharanya, Abhishek e Reddy (2017) é um exemplo de sistema crítico de monitoramento, o qual é voltado para atuar no monitoramento de doenças cardíacas. Um exemplo de sistema de diagnóstico médico pode ser visto no trabalho apresentado por Muangprathub e Boonjing (2014), os quais construíram um sistema usando raciocínio baseado em casos capaz de diagnosticar 50 tipos de doenças. Considerando um sistema para tratamento médico, um exemplo é o sistema médico portátil proposto por Sjaheim *et al.* (2014), que, além do tratamento, é utilizado para diagnóstico de lesões cerebrais traumáticas.

O escopo neste trabalho está em sistemas médicos críticos de segurança para tratamento de pacientes. Mais especificamente, é considerado um sistema de controle de bomba de infusão

---

<sup>1</sup> <https://www.fda.gov/>

<sup>2</sup> <http://portal.anvisa.gov.br/>

de insulina. Neste sistema é realizado o tratamento de pacientes que sofrem de diabetes. O sistema é constituído por componentes de *hardware* e *software*. Os componentes de *hardware* formam a bomba de infusão de insulina, a qual simula o funcionamento do pâncreas. Por outro lado, os componentes de *software* formam o *software* embutido na bomba que é responsável por controlá-la (MERTZ, 2018).

Como se trata de um sistema crítico, os requisitos de bombas de infusão de insulina devem ser analisados com cuidado para que se tenha um certo nível de garantia de que o sistema funcionará em conformidade com o esperado. Neste sentido, linguagens de especificação formal podem ser utilizadas, pois são embasadas em princípios matemáticos que permitem exatidão na expressão das ideias vinculadas ao projeto do sistema. Redes de Petri coloridas (*Coloured Petri Nets - CPN*) é um exemplo de linguagem de modelagem formal (JENSEN; KRISTENSEN, 2009).

*CPN* é uma linguagem gráfica baseada em conceitos matemáticos que combina as capacidades das redes de Petri tradicionais com outros recursos, como, por exemplo, linguagem de programação de alto nível, modularização e temporização. A escolha de *CPN* neste trabalho é justificada porque a linguagem permite a modelagem de simultaneidade, comunicação e sincronização dos sistemas concorrentes, bem como a análise de suas propriedades. Características de simultaneidade, comunicação e sincronização estão diretamente relacionadas com sistemas de controle de bombas de infusão de insulina.

Neste trabalho é realizada a especificação formal de um modelo de referência do sistema de controle de bomba de infusão de insulina e é mostrado como utilizá-lo por meio de um estudo de caso da bomba *ACCU-CHEK Spirit* versão 2.XX. Para a construção do modelo foram consideradas as funcionalidades chave para o funcionamento adequado da bomba. O manual do equipamento *ACCU-CHEK Spirit* versão 2.XX<sup>3</sup>, o modelo deste equipamento disponibilizado por (SILVA, 2015) e o modelo de bomba de infusão de insulina genérica definido por Zhang, Jones e Jetley (2010) serviram como fonte para a identificação dos requisitos que foram incorporados ao modelo de referência. Diferente desses modelos usados como base, o novo modelo, especificado usando *CPN* com o auxílio da ferramenta *CPN/Tools*<sup>4</sup>, é parametrizável e modularizado. Além disso, considerando apenas o modelo genérico de Zhang, Jones e Jetley (2010), o novo modelo apresentado neste trabalho destaca-se por ser executável, temporizado e possível de ser verificado por meio da técnica de verificação automática de modelos (*Model Checking*) (CLARKE *et al.*,

<sup>3</sup> <http://www.webcitation.org/6ZdVHOgBx>

<sup>4</sup> <http://CPNtools.org/>



2018). Estas características são contempladas no modelo de (SILVA, 2015), porém, ele não considera alguns aspectos temporais relevantes, como, por exemplo, o controle de administração.

Portanto, a principal contribuição obtida com a condução desta pesquisa é a disponibilização de um modelo formal mais robusto que pode ser reutilizado durante o processo de certificação de equipamentos e de um estudo de caso de um sistema de bomba de infusão de insulina para mostrar como utilizar o modelo de referência. No estudo de caso, o modelo especificado é também verificado e validado por meio da técnica de verificação automática de modelos e simulações, respectivamente.

## 1.1 PROBLEMATIZAÇÃO

No contexto de modelagem de *software* existem três tipos de especificação de sistemas computacionais: informal, semi-formal e formal. De maneira geral, o que diferencia as duas primeiras da última é que a especificação formal é fortemente fundamentada em conceitos matemáticos, como, por exemplo, a teoria dos conjuntos (ALI; MIRGHANI; IBRAHIM, 2017). Além disso, a diferença básica entre as especificações informal e semi-formal está na forma de representação. Na especificação semi-formal, diferentemente da informal, são utilizadas notações (ou elementos) gráficas com sintaxe e semântica bem definida, além de descrições textuais.

Os tipos mais comuns de especificação informal incluem linguagem natural e linguagem estruturada. A especificação semi-formal inclui as notações gráficas, dentre as quais a mais conhecida é a linguagem de modelagem unificada (*Unified Modeling Language - UML*), que é bastante utilizada no contexto de sistemas comerciais.

A linguagem natural tem como vantagem principal a facilidade de ser compreendida por todos. Isso ocorre porque é utilizada a língua nativa falada por todas as pessoas de um grupo cultural específico como meio para expressar os requisitos de uma especificação. Apesar dessa vantagem, algumas características intrínsecas à linguagem natural inviabilizam o seu uso isolado para elaboração de especificação de sistemas críticos. Dentre as limitações encontradas, pode-se destacar ambiguidade e significado dependente do conhecimento do leitor (MIRANDA *et al.*, 2018). Essas características implicam que um determinado requisito expresso em linguagem natural pode ser compreendido de formas diferentes por leitores diferentes, o que dificulta o entendimento e análise de propriedades de segurança relacionadas com os principais requisitos de sistemas críticos (LEE; CHOI, 2018).

Uma forma de diminuir ambiguidade da especificação é limitar a liberdade do escritor dos requisitos. Isso caracteriza o que é denominado de linguagem estruturada. Essa continua utilizando linguagem natural, mas define um formulário padrão para a escrita dos requisitos, o que garante uma certa uniformidade sobre a especificação. Entretanto, por usar linguagem natural, a ambiguidade continua existindo (isso é um problema a ser evitado ou minimizado o máximo possível).

A notação gráfica é uma outra alternativa para tentar minimizar ainda mais esse problema, porém, não é possível fornecer garantia de que um sistema especificado por meio de notação gráfica funcionará adequadamente. Os recursos desse tipo de especificação ainda proporcionam um nível de ambiguidade muito elevado. Até mesmo linguagens como *UML*, a qual é formada por componentes com sintaxe e semântica, provê bastante liberdade durante a construção de especificações por não possuir uma definição matemática (KUMAR *et al.*, 2018).

Como descrito anteriormente, tratando-se de sistemas médicos críticos de segurança, é preciso demonstrar que o sistema funcionará como o esperado para obter sua certificação por alguma agência reguladora, a qual o autorizará a ser comercializado. O uso isolado das formas de especificação apresentadas anteriormente não é viável porque elas não permitem prover esse tipo de garantia.

Vale destacar que um problema que pode ser amenizado com a construção de um modelo formal de bomba de infusão de insulina é a quantidade de notificações (*recalls*) ou defeitos existentes em sistemas dessa natureza (defeitos que colocam a vida de usuários em risco) (RATHORE *et al.*, 2018). A regulamentação desse tipo de sistema, que ocorre por meio da sua certificação, e, conseqüentemente, a sua monitoração, resulta na identificação de defeitos em equipamentos comercializados. Por não permitir uma análise mais profunda dos requisitos, a utilização somente de especificações informais e semiformais durante o processo de certificação pode diminuir a confiança no funcionamento deste tipo de sistema.

## 1.2 OBJETIVOS

O objetivo geral com este trabalho é disponibilizar um artefato para o projeto do sistema de controle de bomba de infusão de insulina, que possa ser reutilizado durante um processo de certificação, e que atenda os requisitos preestabelecidos na especificação de equipamentos já existentes. Além disso, realizar um estudo de caso do equipamento *ACCU-CHEK Spirit* versão 2.XX.

Para alcançar o objetivo geral destacado, os seguintes objetivos específicos foram realizados:

- Elencar os requisitos essenciais de bomba de infusão de insulina, verificando se estes estão, de modo geral, de acordo com o estado da arte sobre sistemas deste tipo.
- Estender os resultados sobre o sistema de controle de bomba de infusão de insulina apresentado na tese de Silva (2015) utilizando Redes de Petri coloridas.

### 1.3 JUSTIFICATIVA

Neste trabalho é utilizado um tipo de linguagem de especificação formal. Porém, é importante destacar que existem várias outras linguagens e representações gráficas com esse mesmo propósito, como, por exemplo, Autômatos. Um exemplo de utilização de Autômatos é o trabalho de Niezen e Eslambolchilar (2016), no qual autômatos híbridos baseados em comportamentos foi usado para a construção de um modelo de operador humano para interação de dispositivos médicos. Além disso, conforme destacado na Seção 1.1, no contexto de modelagem de *software* existem também diversas técnicas voltadas para elaboração de especificação informal e semi-formal, como a linguagem estruturada e a linguagem de modelagem unificada.

Conforme apontado na contextualização, o novo modelo de bomba de infusão de insulina é relevante porque possui características que não são totalmente contempladas nos modelos já existentes. Entre essas características estão as seguintes: o modelo é parametrizável, podendo ser instanciado para várias especificações de bombas de infusão de insulina; o modelo é temporizado, permitindo a inclusão de questões relacionadas com tempo, que está presente em propriedades contidas no sistema em estudo; o modelo é modularizado (dividido em partes), o que o torna mais fácil de ser compreendido e gerenciado; o modelo pode ser verificado por meio da técnica de verificação automática de modelos e; o modelo é executável, possibilitando a sua análise via simulação. O estudo de caso do equipamento *ACCU-CHEK Spirit* versão 2.XX é útil para demonstrar como utilizar o modelo genérico a partir de sua especificação, além de ser útil para a verificação e validação do modelo.

A especificação formal pode ser considerada como adequada para o *software* objeto de estudo neste trabalho porque, ao contrário das técnicas de especificação informal, é uma maneira precisa de especificação de sistema, que permite a redução da possibilidade de mal-entendidos (que são geralmente resultados de modelos ambíguos). Essa precisão existe porque ela é embasada em conceitos matemáticos, os quais permitem representar as ideias de forma exata

e assim minimizar a ambiguidade. Portanto, com a especificação formal é possível expressar os requisitos com exatidão e com o mínimo de ambiguidade, provendo alguma garantia de que o sistema funcionará corretamente. Neste sentido, torna-se a melhor opção para especificação de sistemas críticos. Ainda, com a especificação formal, é possível realizar análises mais detalhadas dos requisitos, sendo assim uma maneira eficaz para descoberta de erros humanos em especificações (ALI; MIRGHANI; IBRAHIM, 2017).

A escolha de *CPN* para a modelagem do sistema de controle de bomba de infusão de insulina justifica-se por se tratar de uma linguagem de modelagem formal “bem fundamentada e ao mesmo tempo versátil o suficiente para ser usada na prática em sistemas de tamanho e complexidade encontrados em projetos industriais típicos” (JENSEN; KRISTENSEN, 2009, p. 4, tradução nossa). Além disso, foi o uso de *CPN* que possibilitou construir um modelo executável, parametrizável, temporizado e modularizado.

Os modelos *CPN* gerados por meio da ferramenta *CPN/Tools* consistem em artefatos de *software* que, quando analisados, podem aumentar a confiança de que o sistema funciona de forma adequada e com isso aumentar também a possibilidade de obter a certificação do sistema para o uso no mercado. Isto é um ponto importante e que agrega relevância a este trabalho.

Esse tipo de certificação é dado por agências reguladoras. Como destacado anteriormente, um exemplo de agência com esta competência é a *FDA* dos EUA que define requisitos rigorosos para aprovação de sistemas médicos embarcados antes que eles cheguem ao mercado (HAWKINS *et al.*, 2013). Conforme Sobrinho *et al.* (2017) a certificação é útil para garantir o funcionamento adequado dos sistemas críticos.

## 1.4 METODOLOGIA

De modo geral, para a elaboração deste trabalho, foi realizada uma revisão bibliográfica, seguida de um estudo de caso do equipamento *ACCU-CHEK Spirit* versão 2.XX baseado na tese de Silva (2015).

Para alcançar o primeiro objetivo específico definido na Seção 1.2, uma revisão bibliográfica foi realizada com base em alguns dos direcionamentos contidos na diretriz para realização de revisão sistemática de literatura disponibilizadas por Kitchenham (2007). Mais especificamente, direcionamentos relacionados com definição de palavras chaves, base de dados, critério de inclusão, critério de exclusão e *strings* de busca.

Por outro lado, para alcançar o segundo objetivo específico definido na Seção 1.2, foi

realizado o estudo de caso destacado anteriormente. Um estudo de caso, segundo Gil (2002, p. 54), “consiste no estudo profundo e exaustivo de um ou poucos objetos, de maneira que permita seu amplo e detalhado conhecimento”. Como já mencionado, essa foi uma das metodologias usadas neste trabalho, e, para tanto, o objeto de estudo escolhido é um sistema de controle de bomba de infusão de insulina.

## 1.5 ORGANIZAÇÃO DO DOCUMENTO

Este documento de monografia está estruturado em 6 capítulos. Neste primeiro capítulo foi apresentada uma introdução sobre o trabalho, composta por contextualização do tema pesquisado, problematização, objetivos geral e específicos, justificativa para a execução dessa pesquisa, e metodologia aplicada. O restante dos capítulos está organizado da seguinte maneira:

- no Capítulo 2 são introduzidos conceitos sobre bomba de infusão de insulina (inclusive, os requisitos do modelo que foi especificado com este trabalho são apresentados), *CPN* e verificação automática de modelos. Estes conceitos foram utilizados durante o desenvolvimento do trabalho;
- no Capítulo 3 são apresentados trabalhos relacionados com o tema de pesquisa;
- no Capítulo 4 é apresentada a especificação da bomba de infusão de insulina em *CPN*. Essa especificação inclui um modelo inicial com a utilização apenas de dados do tipo inteiro e um modelo mais detalhado que faz uso de dados do tipo real e de temporização;
- no Capítulo 5 é apresentado o estudo de caso do equipamento *ACCU-CHEK Spirit* versão 2.XX. Nesse estudo de caso é realizada a validação e a verificação dos modelos apresentados no Capítulo 4;
- no Capítulo 6 são apresentadas conclusões obtidas com a realização deste trabalho, bem como sugestões de trabalhos futuros.

## 2 REFERENCIAL TEÓRICO

Neste capítulo são apresentados os conceitos utilizados ao longo deste trabalho de pesquisa. Os conceitos envolvem bomba de infusão de insulina, redes de Petri coloridas (*Coloured Petri Nets - CPN*) e verificação automática de modelos, os quais são apresentados, respectivamente, nas Seções 2.1, 2.2 e 2.3.

### 2.1 BOMBA DE INFUSÃO DE INSULINA

Uma bomba de infusão de insulina é um sistema médico crítico de segurança que simula o funcionamento do pâncreas (um órgão interno que produz insulina) controlando os níveis de glicose do sangue do corpo humano (ISLAM *et al.*, 2016). Conforme foi mencionado no Capítulo 1, esse sistema é constituído por componentes de *hardware* e *software*, os quais em conjunto atuam no monitoramento e tratamento de pessoas que sofrem de diabetes.

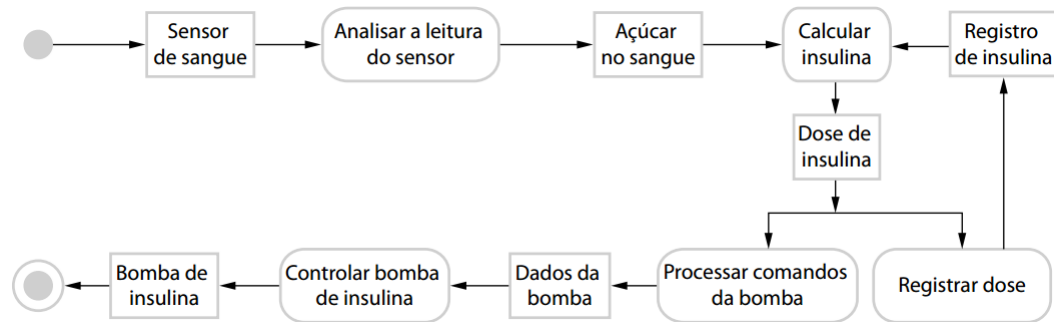
Os componentes gerais de *hardware* de uma bomba de infusão de insulina são alarme (usado para fornecer alguma informação sonora ao usuário), bomba (atua em conjunto com o módulo de agulha fornecendo a insulina ao corpo de um paciente), controlador (realiza ações de controle de todos os demais componentes), *display* (informa estado da bomba e propriedades de segurança), fonte de energia (fornece a energia necessária para o sistema funcionar), módulo de agulha (auxilia na injeção de insulina), relógio (permite ao controlador definir questões relativas a tempo, como, por exemplo, o tempo exato para injeção de insulina), reservatório de insulina (responsável por armazenar a insulina) e sensor (coleta informações do sangue que permitem definir os níveis de glicose do paciente). A estrutura de *hardware* mais detalhada de uma bomba de infusão insulina pode ser visto no trabalho de Hu e Li (2015) sobre o projeto de uma bomba de insulina inteligente.

Quanto aos componentes de *software* que fazem parte do sistema cabe a cada projetista defini-los, entretanto, o funcionamento do sistema deve contemplar as atividades apresentadas no modelo de atividade da Figura 1. O modelo foi construído por meio da linguagem de modelagem unificada (*Unified Modeling Language - UML*) e ilustra as atividades realizada pelo *software* no controle da bomba de infusão de insulina.

Segundo Sommerville (2011), o controle realizado pelo *software* embutido na bomba de infusão de insulina ilustrado na Figura 1 segue basicamente as seguintes ações:

1. Um microsensor embutido no paciente coleta informações do sangue (em geral, a condu-

Figura 1 – Modelo de atividade da bomba de infusão de insulina.



Fonte: Sommerville (2011, p. 13).

tividade elétrica do sangue);

2. A informação coletada é enviada para o controlador da bomba que por meio de uma análise definirá o nível de glicose no sangue;
3. O controlador calcula a quantidade de insulina necessária com base no nível de glicose atual e no tempo desde a última dosagem registrada;
4. O controlador registra a dosagem atual em um buffer e processa os comandos da bomba necessários para dar início a aplicação da dosagem de insulina. Neste passo um sinal ou pulso é enviado para a bomba miniaturizada para fornecer a insulina através de uma agulha permanente;
5. O controlador processa os demais comandos para que a bomba forneça por meio da agulha a dosagem de insulina correta – a bomba fornece uma unidade de insulina como resposta a um único pulso do controlador.

Uma falha nesse sistema pode elevar ou diminuir o nível de açúcar no sangue do usuário, podendo provocar sérios riscos a sua saúde (AHMAD *et al.*, 2018). Para evitar a ocorrência desse cenário crítico, no projeto e na implementação de uma bomba de infusão de insulina, dois requisitos essenciais devem ser satisfeitos: disponibilidade (sistema sempre estará disponível para fornecer a insulina) e confiabilidade (o sistema sempre funcionará corretamente) (MARTINS *et al.*, 2015). Isso aumenta a confiança do sistema e, conseqüentemente, as chances de ser liberado para a comercialização (ATA; KHAN, 2017).

Esse modelo geral discutido é um cenário de uso de bombas de infusão de insulina. Existem no mercado diversos modelos de bomba de infusão de insulina. O modelo *ACCU-CHEK Spirit* versão 2.XX, objeto do estudo de caso apresentado no Capítulo 5, é um exemplo. Além dos modelos em utilização no mercado, é possível encontrar na literatura modelos genérico de bomba de infusão de insulina que podem ser utilizados por fabricantes durante o processo de certificação. Um exemplo é o modelo proposto por Zhang, Jones e Jetley (2010). Na Tabela

2 são apresentadas as principais funcionalidades que uma bomba de infusão de insulina deve contemplar, as quais foram extraídas a partir da análise desses dois modelos citados como exemplo. Como o objetivo é simular o comportamento funcional da bomba, as características físicas e eletromagnéticas foram ignoradas.

Tabela 2 – Requisitos técnicos chave da bomba de infusão de insulina

| Parâmetros de Configuração                                                                                                                                                                                                                                                                                                                                    | Funcionalidades                                                                                                                                                  | Propriedades de Segurança                                                                                                           |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| 1. Tipo de Insulina<br>2. Capacidade do Cartucho<br>3. Dose Basal (U/h):<br>Min<br>Max<br>4. Perfil Basal:<br>a. Padrão (doses fixas/min):QDD <sup>(1)</sup><br>b. Personalizado(doses ajustadas/h):QDD<br>5. Bolus Padrão (U):<br>Min<br>Max<br>6. Taxa de Administração (dose/min):<br>a. Dose Basal<br>b. Dose Bolus<br>7. Limite de Aplicação (U):<br>Max | 1. Modos Executar/Parar<br>2. Selecionar perfil basal<br>3. Verificar nível de cartucho<br>4. Programar dose de bolus padrão<br>5. Prover informações ao usuário | 1. Erros(E)/Alertas(A)<br>a. E: "Cartucho Vazio!"<br>b. A: "Nível Baixo do Cartucho!"<br>2. Estado:<br>a."PARADO"<br>b."EXECUTANDO" |

<sup>(1)</sup> Quantidade de Doses Diária

Fonte: Adaptado de Silva (2015).

Existem pelo menos dois tipos de insulina que o endocrinologista prescreve ao paciente diabético sob tratamento por insulino terapia (terapia de insulina): a insulina basal (caracterizada por ser de ação longa e usada para manter os níveis de glicose sanguínea dentro do intervalo normal de um indivíduo durante os períodos em jejum) e a insulina bolus (insulina de ação rápida que é tomada em períodos específicos para manter os níveis de glicose sob controle)<sup>1</sup>. A insulina bolus possui dois subtipos de dosagens: o bolus padrão (usado em situações excepcionais previstas, por exemplo, após uma refeição regular ou prática de exercícios físicos rotineira) e o bolus corretivo (usado em situações excepcionais imprevistas, por exemplo, após uma refeição fora do planejamento alimentar normal do paciente ou prática de exercícios físicos extraordinária) (SILVA, 2015).

Os valores de dosagens desses tipos de insulinas, o valor da capacidade do cartucho e o perfil de administração da insulina, são configurados na bomba obedecendo aos limiares de

<sup>1</sup> <http://www.diabetes.co.uk/insulin/basal-bolus.html>



cada modelo específico. A configuração de todos os parâmetros pode ser feita pelo usuário de acordo com as suas necessidades e com a prescrição médica. As saídas fornecidas pela bomba são informações sobre o estado da mesma, o nível de insulina restante no cartucho e a quantidade de insulina administrada durante sua execução.

## 2.2 REDES DE PETRI COLORIDAS

A rede de Petri é um modelo matemático com representação gráfica, proposto por Carl Petri em 1962, em sua tese de doutorado, intitulada “Comunicação com autômatos”. Este formalismo vem sendo amplamente utilizado, há mais de 30 anos, em diversos domínios de aplicações, dentre os quais destacam-se os sistemas de manufatura, de comunicação, de transporte, de informação e de logísticas. As principais motivações para o seu uso nesses sistemas são especificação, análise do comportamento lógico, avaliação do desempenho e implementação (CARDOSO; VALETTE, 1997). A sua aplicabilidade está voltada para a modelagem de sistemas de eventos discretos (DING; CHEN; WANG, 2017). Sistemas de eventos discretos (*Discrete Event System - DES*) são sistemas nos quais a mudança de um estado para outro são provocadas pela ocorrência de eventos específicos (SAIVES; FARAUT; LESAGE, 2018).

A modelagem de *DESs* é baseada no conceito de evento (mudança no estado do modelo), que se relacionam por meio de *links* (representam as relações entre esses eventos) (WAINER, 2009). A mudança de estado em função do acontecimento de algum evento leva o sistema a executar nova(s) atividade(s). Uma sequência de eventos e de atividades formam processos como simultaneidade, paralelismo, dependência causal, compartilhamento de recursos e sincronização, que são característicos dos *DESs* (TANG; ZHU; FAN, 2018).

Os elementos de rede de Petri que possibilitam modelar os *DESs* são: arco direcionado, lugar, peso de arco, transição e ficha (DOTOLI *et al.*, 2016). O arco direcionado é o elemento que permite conectar lugar a transição e vice-versa. O lugar possui várias interpretações, pode representar, por exemplo, uma atividade, um predicado ou um estado parcial do sistema. O peso de arco refere-se a um número natural positivo, que é atribuído a cada arco. A transição simboliza um evento que ocorre no sistema. Por último, a ficha é o elemento indicador de que um predicado associado a um lugar é verificado. Na Figura 2 está a representação gráfica desses elementos, exceto do peso, que é um número.

Para uma melhor explanação a cerca desses conceitos e de como utilizá-los na modelagem de *DESs* considere a rede de Petri da Figura 3. Trata-se do problema de concorrência do leitor-

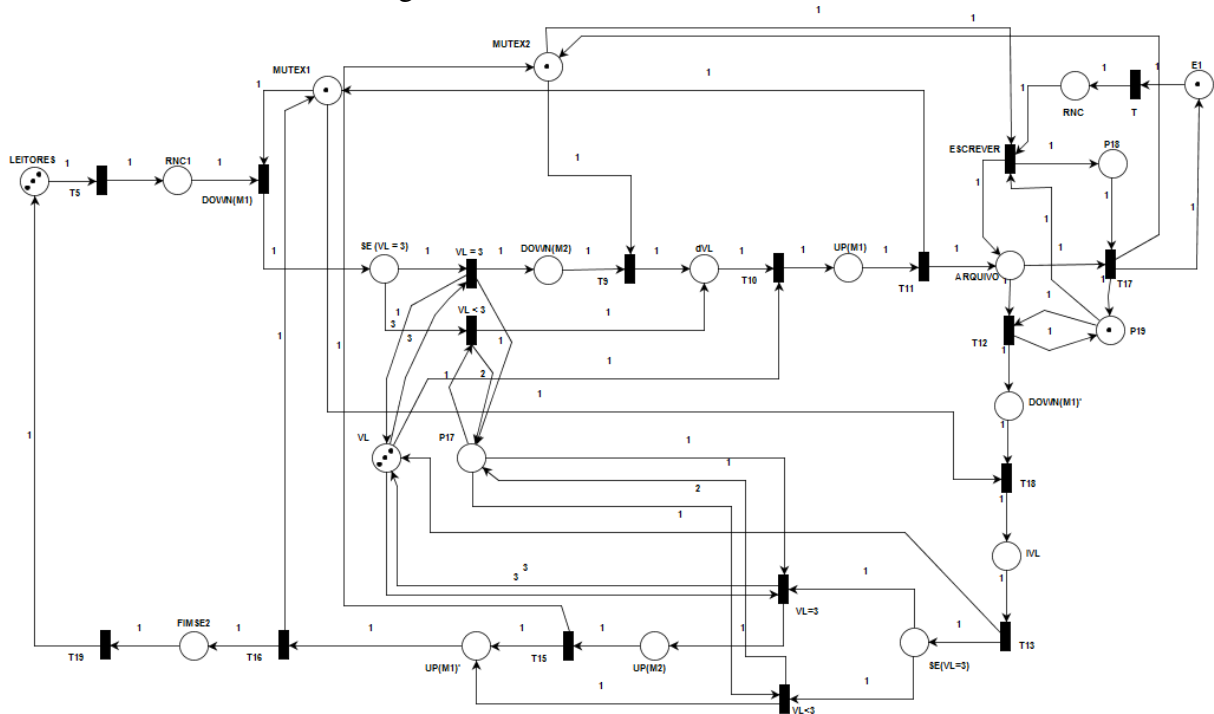
Figura 2 – Elementos gráficos da rede de petri

|                                                                                   |                  |
|-----------------------------------------------------------------------------------|------------------|
|  | Lugar            |
|  | Transição        |
|  | Arco direcionado |
|  | Ficha            |

Fonte: Próprio autor, 2019.

escritor, no qual existem duas entidades (o leitor e o escritor) que compartilham um recurso de tal forma que, se uma delas estiver usando o recurso em um instante de tempo, a outra deverá aguardar a liberação para poder usá-lo.

Figura 3 – Problema do leitor-escritor



Fonte: Próprio autor, 2019.

Nesse exemplo, o problema do leitor-escritor é modelado com três leitores e um escritor. Nesse caso, um leitor é representado por uma ficha no lugar LEITORES, que representa o estado parcial “leitor parado”. Portanto, a presença de três fichas nesse lugar indica que nenhum dos três leitores estão lendo e, por consequência, o arquivo está livre para ser acessado pelo escritor. O escritor, por sua vez, é representado pela ficha no lugar E1, que representa o estado parcial “escritor parado”.

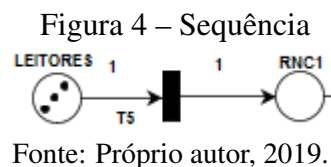
O estado de um DES é dado pelos valores das suas variáveis de estados (TANG; ZHU, 2018). Em rede de Petri, a distribuição das fichas nos lugares determina o estado do sistema (também chamado de marcação), que é alterado após o disparo de uma transição. Nesse sentido,

o disparo de uma transição faz o sistema transitar de um estado atual  $E_i$  para o próximo estado  $E_{i+1}$  implicando também na mudança de uma marcação  $M_i$  para uma nova marcação  $M_{i+1}$ .

Uma transição apenas pode ser disparada se ela estiver habilitada (ou sensibilizada) ao disparo. Isso ocorre quando o número de fichas em cada um dos lugares de entrada é maior ou igual ao peso do arco que liga o lugar à transição. O peso do arco define quantas fichas devem ser retiradas dos lugares de entrada (lugar cujo arco direcionado chega a uma transição) e quantas fichas devem ser inseridas nos lugares de saída (lugar cujo arco direcionado sai de uma transição) quando uma transição é disparada. Por padrão o peso de um arco é 1, mas pode ser alterado pelo projetista caso necessário. Sendo assim, após o disparo de uma transição, as fichas dos seus lugares de entradas são retiradas e outras fichas são depositadas nos seus lugares de saída, de acordo com o peso de cada arco. É importante ressaltar que a retirada de uma ficha do lugar indica que o predicado por ele representado não é mais verdadeiro. Em contrapartida, a inserção de ficha no lugar indica que a atividade associada aquele lugar está sendo executada.

Observe que apenas as transições T e T5 do modelo estão sensibilizadas, enquanto que todas as outras não podem ser disparadas. No caso da transição T5, mesmo após o seu disparo, ela continuará sensibilizada por mais duas vezes, pois possui 3 fichas em seu único lugar de entrada e o peso do arco é 1. No primeiro disparo da transição T5 uma ficha é depositada no lugar RNC1 configurando uma nova marcação e, por conseguinte, levando a habilitação da transição DOWN(M1). Note que essa transição possui dois lugares de entradas: RNC1 e MUTEX1. No estado inicial ela estava desabilitada porque faltava ficha suficiente no lugar RNC1.

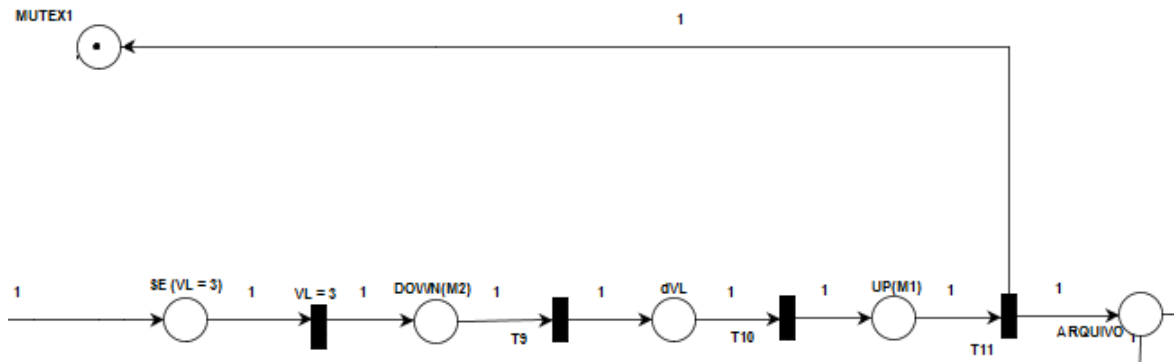
Alguns processos podem ser visualizados no modelo. As Figura 4, 5, 6, 7 e 8 são recortes da rede de Petri da Figura 3 que enfatiza esses processos. Na Figura 4 é apresentada a modelagem de um processo sequencial, no qual as atividades são realizadas uma após a outra sem nenhum caminho alternativo.



Na Figura 5 é apresentado um processo que envolve simultaneidade. A transição T11, quando disparada, gera uma ficha para o lugar MUTEX1 e uma ficha para o lugar ARQUIVO. Essa situação pode ser útil, por exemplo, quando deseja-se modelar uma situação de sequência de atividades paralelas, sendo que o início dessas atividades deva ocorrer no mesmo tempo.

Um cenário envolvendo paralelismo e sincronização é ilustrado na Figura 6. O lugar

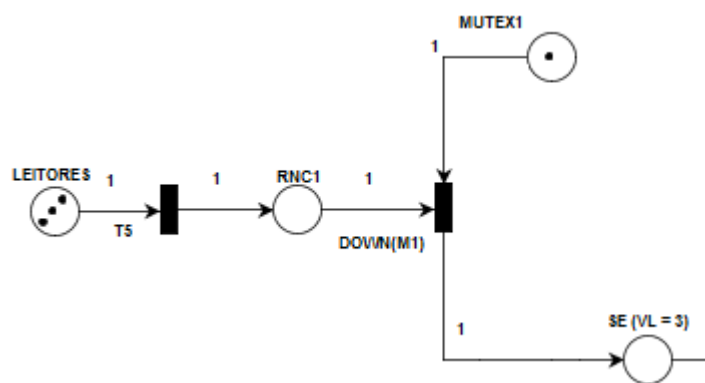
Figura 5 – Simultaneidade



Fonte: Próprio autor, 2019.

MUTEX1 e os conjunto de lugares e transição LEITORES-T5-RNC1 podem ser vistos como processos distintos que evoluem em paralelo e de forma assíncrona até a transição DOWN(M1), a qual pode ser vista como um ponto de sincronização que une os dois processos. Esse ponto exige que exista pelo menos uma ficha nos lugares MUTEX1 e RNC1.

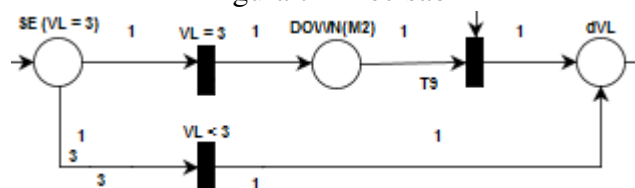
Figura 6 – Paralelismo e sincronização



Fonte: Próprio autor, 2019.

Uma situação comum no contexto de sistemas computacionais é a decisão, isto é, decidir entre alternativas para realizar alguma atividade. Situação similar a essa é colocada na Figura 7. Quando o lugar SE(VL = 3) tiver uma ficha, as transições VL = 3 e VL < 3 são habilitadas. Se VL = 3 for disparada, então VL < 3 é desabilitada e o processo continua pelo caminho DOWN(M2)-T9-dVL. Caso contrário, o processo continua por dVL diretamente.

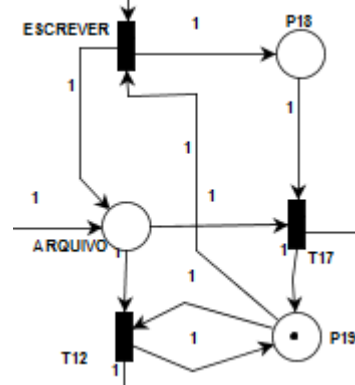
Figura 7 – Decisão



Fonte: Próprio autor, 2019.

Por último, na Figura 8 é mostrado o processo de compartilhamento de recurso. Nesse processo há dois processos que concorrem por um único recurso para realizar determinada atividade. O recurso é representado pelo lugar P19 e o seu estado parcial encontra-se disponível. A concorrência pelo recurso está sendo implementada nas transições ESCREVER e T12. Se alguma dessas transições for disparada a outra automaticamente fica inabilitada ao disparo. A liberação do recurso para que ele possa ser novamente utilizado ocorre quando a transição T12 ou a transição T17 é finalizada. Em geral, essa abordagem é usada para modelar uma situação na qual processos devem ter acesso a um dado recurso para realizar sua tarefa ou parte de suas tarefas, de modo que apenas uma tarefa por vez seja executada e a outra deva aguardar o fim da execução para prosseguir.

Figura 8 – Compartilhamento de recurso



Fonte: Próprio autor, 2019.

Todos os conceitos de rede de Petri apresentados até o momento fazem parte de elementos de sua definição, entretanto, associado a cada um existe uma definição matemática que os fundamenta. Do ponto de vista formal, Murata (1989) define que uma rede de Petri é uma 5-tupla,  $PN = (P, T, F, W, M_0)$  onde:

- $P = \{p_1, p_2, \dots, p_m\}$  é um conjunto finito de lugares,
- $T = \{t_1, t_2, \dots, t_n\}$  é um conjunto finito de transições,
- $F \subseteq (P \times T) \cup (T \times P)$  é um conjunto de arcos (relação de fluxo),
- $W : F \rightarrow \{1, 2, 3, \dots\}$  é uma função de peso,
- $M_0 : P \rightarrow \{0, 1, 2, 3, \dots\}$  é a marcação inicial,
- $P \cap T = \emptyset$  e  $P \cup T \neq \emptyset$ .

Apesar da grande importância desse formalismo matemático para a modelagem de sistemas concorrentes, a rede de Petri apresenta algumas limitações, dentre as quais duas mais notáveis são: modelos grandes ficam inviáveis de serem construídos; os recursos do sistema

são todos abstraídos para o conceito de ficha, não permitindo definir tipos de dados para os diversos recursos que um sistema pode manipular. Diante disso, surgiu uma extensão da rede de Petri denominada *CPN*, a qual dispõe de recursos para a modelagem permitindo incluir detalhes que se aproximam mais do funcionamento real dos sistemas. *CPN* é apenas uma das diversas extensões de rede de Petri (PETERSON, 1981). Os principais recursos de *CPN*, além daqueles herdados de rede de Petri tradicionais, são modularização e linguagem de programação *CPN ML* (TARI *et al.*, 2013).

A modularização permite que o modelo *CPN* seja dividido em partes gerenciáveis com interfaces bem definidas. A linguagem de programação *CPN ML*, baseada na linguagem de programação funcional *Standard ML*, fornece os primitivos para a definição de tipos de dados, para descrever a manipulação de dados e para criar modelos compactos e parametrizáveis.

Além disso, assim como os modelos de rede de Petri, os modelos *CPN* são executáveis possibilitando a sua simulação, a qual permite investigar diferentes cenários e explorar o comportamento do sistema. Essa simulação pode ser feita de duas formas: interativa e automática. Na simulação interativa o projetista determina a próxima etapa selecionando entre os eventos ativados no estado atual. De outro modo, na simulação automática, o modelo *CPN* é posto em execução similar a um programa (JENSEN; KRISTENSEN, 2009).

Formalmente uma *CPN* pode ser definida por partes: uma para definir *CPN* não hierárquica (*CPN* sem módulos); outra para definir um módulo de *CPN*; e outra para definir *CPN* hierárquica. A definição da primeira parte, dada por Jensen e Kristensen (2009), e das outras duas partes, conforme Jensen e Kristensen (2015), são, respectivamente, as seguintes:

- *CPN* não hierárquica é uma tupla de nove elementos  $CPN = (P, T, A, \Sigma, V, C, G, E, I)$  na qual:
  1.  $P$  é um conjunto finito de lugares. Em outras palavras, a quantidade de lugares de uma *CPN* é finito.
  2.  $T$  é um conjunto finito de transições tal que  $P \cap T = \emptyset$ . Ou seja, não existe algum elemento em  $T$  igual a algum elemento em  $P$ .
  3.  $A \subseteq P \times T \cup T \times P$  é um conjunto de arcos direcionados. Isso significa que um arco pode ligar apenas um lugar a uma transição ou vice-versa. Não é possível usar um arco para ligar um lugar a outro lugar ou uma transição a outra transição.
  4.  $\Sigma$  é um conjunto não vazio de cores. Isto é, os lugares de *CPN* possui algum tipo de dado (ou cor), o qual pertence a  $\Sigma$ .
  5.  $V$  é um conjunto finito de variáveis tipadas tal que  $Tipo[v] \in \Sigma$  para todas as variáveis

$v \in V$ . Ou melhor, existe uma cor associada a cada variável tal que essa cor é um elemento de  $\Sigma$ .

6.  $C : P \rightarrow \Sigma$  é uma função de conjunto de cor que associa uma cor a cada lugar. Dito de outro modo, cada lugar da  $CPN$  possui uma cor.
  7.  $G : T \rightarrow EXPR_v$  é uma função de guarda que associa uma guarda a cada transição  $t$  tal que  $Tipo[G(t)] = Bool$ . Isto é, a cor de guarda de cada transição é um *booleano*.
  8.  $E : A \rightarrow EXPR_v$  é uma função de expressão de arco que associa uma expressão de arco a cada arco  $a$  tal que  $Tipo[E(a)] = C(p)_{MS}^2$ , no qual  $p$  é o lugar conectado ao arco  $a$ . Em outros termos, a cor associada a  $a$  deve ser a mesma cor associada a  $p$ , que é o lugar com o qual  $a$  está conectado.
  9.  $I : P \rightarrow EXPR_\theta$  é uma função de inicialização que associa uma expressão de inicialização a cada lugar  $p$  tal que  $Tipo[I(p)] = C(p)_{MS}$ . Ou seja, a cor dos valores de inicialização de um lugar deve ser igual a cor daquele lugar.
- Módulo de  $CPN$  é uma tupla de quatro elementos  $CPN_M = (CPN, T_{sub}, P_{porta}, PT)$ , na qual:
    1.  $CPN = (P, T, A, \Sigma, V, C, G, E, I)$  é uma  $CPN$  não hierárquica.
    2.  $T_{sub} \subseteq T$  é um conjunto de transições de substituição. Isso significa que uma transição de substituição é também uma transição.
    3.  $P_{porta} \subseteq P$  é um conjunto de lugares porta. Dito de outro modo, um porta é também um lugar.
    4.  $PT : P_{porta} \rightarrow IN, OUT, I/O$  é um conjunto de tipo porta que associa tipos de porta a lugares. Em outras palavras, uma porta deve possuir um dos seguintes tipos:  $IN, OUT, I/O$ .
  - $CPN$  hierárquica é definida como uma tupla de quatro elementos  $CPN_H = (S, SM, PS, FS)$ , na qual:
    1.  $S$  é um conjunto finito de módulos. Cada módulo é um módulo de Rede de Petri Colorida  $s = ((P^s, T^s, A^s, \Sigma^s, V^s, C^s, G^s, E^s, I^s), T_{sub}^s, P_{porta}^s, PT_s)$ . É necessário que  $(P^{s_i} \cup T^{s_i}) \cap (P^{s_j} \cup T^{s_j}) = \emptyset$  para todo  $s_i, s_j \in S$  tal que  $i \neq j$ .
    2.  $SM : T_{sub} \rightarrow S$  é uma função de sub-módulo que associa um sub-módulo a cada transição de substituição. É necessário que a hierarquia de módulo seja acíclica, ou seja, um sub-módulo não pode possuir transição de substituição associada a um sub-módulo superior na hierarquia.

<sup>2</sup> MS está relacionado a “multiconjunto”.

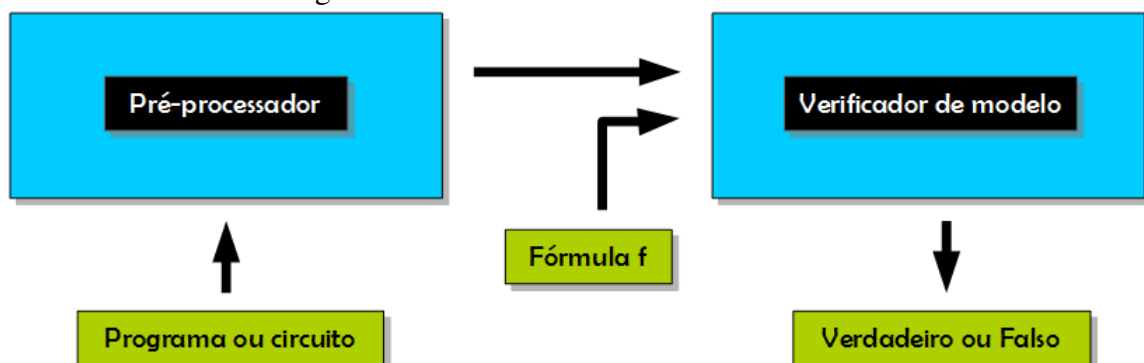
3.  $PS$  é uma função de relação *porta-socket* que associa uma relação *porta-socket*  $PS(t) \subseteq P_{\text{sock}}(t) \times P_{\text{porta}}^{\text{SM}(t)}$  a cada transição de substituição  $t$ . É necessário que  $PT(p) = PT(p')$ ,  $C(p) = C(p')$  e  $I(p)\langle \rangle = I(p')\langle \rangle$  para todo  $(p, p') \in PS(t)$  e  $t \in T_{\text{sub}}$ . Em síntese, a cada transição de substituição existe uma associação de uma relação *porta-socket*, de modo que o tipo do *socket* deve ser igual ao tipo da *porta* e a cor do *socket* deve ser igual a cor da *porta*.
4.  $FS \subseteq 2^P$  é uma família de conjuntos de fusão (*fusion sets*) não vazios tal que  $C(p) = C(p')$  e  $I(p)\langle \rangle = I(p')\langle \rangle$  para todo  $p, p' \in fs$  e todo  $fs \in FS$ . Em outras palavras, é possível fundir dois lugares de modo que a marcação de ambos sejam sempre iguais. São semelhantes às variáveis globais conhecidas em muitas linguagens de programação.

## 2.3 VERIFICAÇÃO AUTOMÁTICA DE MODELOS

A verificação automática de modelos é uma técnica de verificação automática para sistemas concorrentes de estado finito, criada, em 1980, por Edmund Clarke, Allen Emerson e Joseph Sifakis. Ela permite verificar a adequação de um sistema aos seus requisitos, facilitando o trabalho do projetista com relação a correção de eventuais erros de projeto (CLARKE; EMERSON; SIFAKIS, 2009).

O funcionamento dessa técnica é ilustrado na Figura 9. O verificador de modelo consiste em um algoritmo que, dado o gráfico de transição de estado do sistema e a propriedade que se deseja verificar, informa se essa propriedade é satisfeita ou não, isto é, se ela é verdadeira ou falsa. Nesse último caso, é apresentado um contraexemplo que mostra a origem do problema para que ele seja corrigido (VIZEL; WEISSENBACHER; MALIK, 2015).

Figura 9 – Estrutura do verificador de modelo



Fonte: Clarke (2008), tradução nossa.



O gráfico de transição de estado é resultado do processamento realizado pelo pré-processador, que recebe como entrada o sistema (programa ou circuito) cujas propriedades serão verificadas. A propriedade do sistema que será submetida a verificação automática deve ser especificada formalmente via fórmula de lógica temporal (CLARKE, 2008). O *ASK-CTL* é um exemplo de biblioteca que permite a especificação desse tipo de fórmula e a sua verificação. O *ASK-CTL* é uma extensão da lógica de árvore de computação (*Computation Tree Logics - CTL*) e é interpretada sobre o espaço de estado de *CPN* por meio da ferramenta *CPN/Tools* (CHRISTENSEN; MORTENSEN, 1996). Alguns operadores lógicos *ASK-CTL* são descritos a seguir:

- $EV(\varphi)$ : significa que é eventualmente possível alcançar um estado onde a fórmula  $\varphi$  é satisfeita. Com a aplicação do operador  $EV$ , a fórmula  $\varphi$  é satisfeita para todos os caminhos, em algum momento no futuro;
- $NF(< \text{expressão de nó} >)$ : significa uma função de nó para subfórmulas de estado;
- $NOT(\varphi)$ : possui a mesma interpretação padrão do operador lógico  $\neg$ , com notação pré-fixa;
- $AND(\varphi, \varphi')$ : possui a mesma interpretação padrão do operador lógico  $\wedge$ , com notação pré-fixa;
- $OR(\varphi, \varphi')$ : possui a mesma interpretação padrão do operador lógico  $\vee$ , com notação pré-fixa.

De fato, essa técnica facilita o processo de verificação de sistemas, especialmente, por fornecer contraexemplos nos casos de uma determinada propriedade ser verificada como falsa. Entretanto, ela é suscetível ao problema da explosão de espaço de estados, o qual é caracterizado quando um sistema contém um número demasiadamente grande de estados de forma a inviabilizar a sua verificação (FAKHIR; KAZMI, 2018). Para contornar esse tipo de problema existem algumas técnicas baseadas na redução parcial de estados, dentre as quais está o grafo de componentes fortemente conectados, que é um subgráfico máximo do espaço de estado original do sistema em que todos os nós são alcançáveis uns dos outros (CHENG; CHRISTENSEN; MORTENSEN, 1997). O problema de explosão de espaço de estados é tratado neste documento apenas para a primeira versão do modelo da bomba de infusão de insulina apresentada no Capítulo 4 deste trabalho.

### 3 TRABALHOS RELACIONADOS

Neste capítulo são apresentados alguns trabalhos relacionados ao tema da pesquisa abordado neste documento. A Seção 3.1 apresenta trabalhos sobre bombas de insulina. A Seção 3.2 apresenta trabalhos nos quais redes de Petri coloridas (*Coloured Petri Nets - CPN*) são utilizadas. A seção 3.3 apresenta alguns modelos de referência. Na Tabela 3 é apresentado um resumo sobre esses trabalhos.

Tabela 3 – Resumo dos trabalhos relacionados

| <b>Autores</b>                                     | <b>Ano de publicação</b> | <b>Título da pesquisa</b>                                                                                                |
|----------------------------------------------------|--------------------------|--------------------------------------------------------------------------------------------------------------------------|
| Zhang, Jones e Jetley                              | 2010                     | Uma análise de risco para uma bomba de infusão de insulina genérica                                                      |
| Masci, Ayoub, Curzon, Lee, Sokolsky e Thimbleby    | 2013                     | Desenvolvimento Baseado em Modelo do Protótipo da Interface com o Usuário da Bomba de Infusão Genérica PCA em PVS        |
| Sobrinho, Silva, Perkusich e Cunha                 | 2014                     | Utilização de Redes Coloridas de Petri para a Engenharia de Requisitos de um Sistema de Eletrogastrografia de Superfície |
| Hei, Du, Lin, Lee e Sokolsky                       | 2015                     | Esquemas de Controle de Acesso Baseados em Padrão de Infusão do Paciente para o Sistema de Bomba de Insulina sem Fio     |
| Silva                                              | 2015                     | Uma Abordagem Baseada em Modelos para Suporte à Validação de Sistemas Médicos Físico-Cibernéticos                        |
| Sobrinho, Silva, Perkusich, Cunha, Cordeiro e Lima | 2017                     | Modelagem Formal de Sistemas de Aquisição de Sinais Biomédicos: Fonte de Evidências para Certificação                    |
| Yang, Pu, Lian, Gu e Fan                           | 2018                     | Modelagem e Análise da Síntese de Proteínas e Mutação de DNA usando Redes de Petri Coloridas                             |

Fonte: Próprio autor, 2019.

#### 3.1 BOMBA DE INSULINA

Os trabalhos abordando bomba de insulina que serão destacados nesta seção incluem os autores Hei *et al.* (2015) e Silva (2015). O primeiro propõe esquemas de controle de acesso baseados em padrão de infusão do paciente para sistemas de bomba de insulina sem fio. O segundo discorre sobre uma abordagem baseada em modelos para suporte à validação de sistemas médicos físico-cibernéticos (SMFC), na qual inclui um estudo de caso de uma bomba de infusão de insulina.

Hei *et al.* (2015) realizou um estudo sobre mecanismos de segurança de sistemas de bomba de insulina sem fio e identificou que a maioria desses sistemas possuem limitações que os tornam propensos a ataques maliciosos que podem afetar a vida de um paciente. Em particular, dois ataques possíveis foram investigados: uma única overdose aguda com uma quantidade

significativa de medicação e uma overdose crônica com uma pequena quantidade de medicação extra durante um longo período de tempo. A partir da compreensão de que esses ataques ocorrem devido a vulnerabilidades de controle de acesso, os autores propuseram um algoritmo para o controle de acesso baseado no padrão de infusão do paciente. Esse algoritmo permite identificar ataques de overdose mal-intencionados em bombas de insulina. A aplicação do algoritmo em bombas de insulina real mostraram que ele pode efetivamente identificar o único ataque de overdose com uma probabilidade de sucesso de até 98% e detectar o ataque de overdose crônica com uma taxa de sucesso de cerca de 100%.

Silva (2015), em sua tese de doutorado, propôs uma abordagem baseada em modelos para validação de sistemas médicos físico-cibernéticos, focando em promover o reúso e a produtividade. Para avaliar o potencial de reúso do modelo para diferentes contextos, a abordagem foi aplicada a três diferentes cenários clínicos, sendo a bomba de infusão de insulina denominada *ACCU-CHEK Spirit* versão 2.XX utilizada em um desses cenários. Em particular, para a avaliação desse cenário envolvendo a bomba de insulina, foi necessário construir o modelo da bomba com seus principais requisitos e propriedades de segurança, os quais foram extraídos do seu manual. Considerou-se apenas a parte discreta do sistema e o modelo foi criado usando diagrama de blocos e a ferramenta *Simulink*<sup>1</sup>.

### 3.2 REDES DE PETRI COLORIDAS

Conforme descrito no Capítulo 2, *CPN* permite a modelagem de situações que envolvam sequência, decisão, simultaneidade, compartilhamento de recurso e sincronização. Nesse sentido, *CPN* pode ser aplicado em diversos cenários, os quais envolvam essas situações. Os dois estudos apresentados a seguir ilustram a aplicação de *CPN* em outros contextos.

No primeiro estudo, publicado por Yang *et al.* (2018), *CPN* é utilizada para modelar os processos de síntese protéica (processo biológico essencial para gerar proteínas específicas em sistemas vivos) e de determinação de mutação (alterações permanentes na sequência de nucleotídeos da fita de ácido desoxirribonucleico (*deoxyribonucleic acid* - *DNA*)) do *DNA*. Para a modelagem desses processos duas *CPNs* foram construídas, sendo a *CPN* do processo de determinação de mutação resultado do refinamento da *CPN* do processo de síntese protéica. O uso de *CPN* é justificado por ser uma abordagem teórica inovadora e eficiente para a exploração dos processos biológicos. O principal resultado do trabalho é a disponibilização de um modelo

<sup>1</sup> <https://www.mathworks.com/products/simulink.html>

útil para o estudo da mutação genética e de suas implicações na estrutura e função da síntese protéica.

O segundo estudo, realizado por Sobrinho *et al.* (2014), aborda a utilização de *CPN* na engenharia de requisitos para modelar comportamentos de *software* e *hardware* de um sistema de eletrogastrografia de superfície (*surface electrogastrography* - *EKG*). Utilizando o recurso de modularização oferecido pela *CPN*, os autores construíram o modelo do sistema de *EKG* definindo, primeiro, um módulo *CPN* de nível mais alto para apresentar a estrutura geral do sistema e, depois, a partir desse módulo geral definiram seus sub-módulos para apresentar em mais detalhes as partes que o compõem. Esse modelo é apontado como um dos resultados relevante do trabalho visto que pode ser usado por fabricantes para ajudar a aprovar dispositivos médicos no processo de certificação.

### 3.3 MODELOS DE REFERÊNCIA

Existem vários trabalhos na literatura que discorrem sobre modelos de referência de sistemas médico. Entre eles estão os trabalhos publicados por Sobrinho *et al.* (2017), por Masci *et al.* (2013) e por Zhang, Jones e Jetley (2010). Cada um desses trabalhos serão brevemente discutidos, respectivamente, nos próximos parágrafos desta seção.

O primeiro trabalho propôs um modelo de referência de sistemas de aquisição de sinais biomédicos desenvolvido usando *CPN*. Além do modelo, foi apresentado um estudo de caso sobre um sistema de eletrocardiografia (ECG). O modelo genérico é apontado como um artefato de *software* útil para ajudar os fabricantes a reduzir o número de defeitos nos sistemas e gerar evidências de segurança e eficácia no processo de certificação. O estudo de caso de ECG foi realizado para demonstrar como os fabricantes podem gerar essas evidências. Técnicas de verificação e validação foram aplicadas ao modelo para identificar a conformidade do sistema aos seus requisitos. O modelo genérico e o estudo de caso de ECG são os principais resultados do trabalho.

A diferença do anterior, o segundo trabalho discute sobre um modelo de referência que assemelha-se mais ao modelo que será apresentado neste documento, porém, com o foco na infusão de um outro tipo de fluido. Trata-se de um modelo de bomba de infusão de analgésico genérica controlada pelo paciente (*Generic Patient Controlled Analgesic* - *GPCA*). Ele é definido como parte do trabalho desenvolvido em Masci *et al.* (2013), que consistiu em desenvolver um protótipo da interface de usuário para esse dispositivo médico. Primeiro, foi apresentado

o modelo arquitetural em camadas da bomba *GPCA*. Depois, o desenvolvimento do protótipo da interface foi detalhado. Na etapa de desenvolvimento, o protótipo foi verificado usando o Sistema de Verificação de Protótipo (*Prototype Verification System - PVS*). O desenvolvimento do protótipo usando o *PVS*, bem como a abordagem baseada em modelo aplicada na construção desse protótipo, são colocadas como contribuições do trabalho.

Por último, o modelo de referência apresentado no terceiro artigo guarda uma relação ainda mais próxima com este trabalho, servindo inclusive como fundamento para a construção do modelo apresentado no Capítulo 4. O artigo trata sobre análise de risco para uma bomba de infusão de insulina genérica (*Generic Insulin Infusion Pump - GIIP*). Para a realização dessa análise foi definida uma arquitetura genérica de modelo de bomba de infusão de insulina. Foi ressaltado que o modelo é uma abstração de bombas de insulina do mundo real, encapsulando características de projetos comuns. A partir desse modelo um levantamento de riscos foi realizado. Como resultado, são apontados a disponibilização da arquitetura genérica de modelo de bomba de infusão de insulina e de um documento de análise de risco abrangente para esse sistema.

## 4 ESPECIFICAÇÃO DA BOMBA DE INFUSÃO DE INSULINA

Neste capítulo é apresentado a primeira parte dos resultados obtidos com a execução do trabalho, a qual consiste na especificação da bomba de infusão de insulina. Para isso foram criados dois modelos, os quais são brevemente descritos a seguir:

1. Modelo *CPN* abstrato da bomba de infusão de insulina, apresentado na Seção 4.1: modela o funcionamento da bomba considerando apenas valores inteiros para as dosagens de insulina e para a capacidade de armazenamento de insulina suportada pelo cartucho da bomba.
2. Modelo *CPN* mais detalhado da bomba de infusão de insulina, apresentado na Seção 4.2: modela o funcionamento da bomba considerando valores reais e questões relacionadas com tempo.

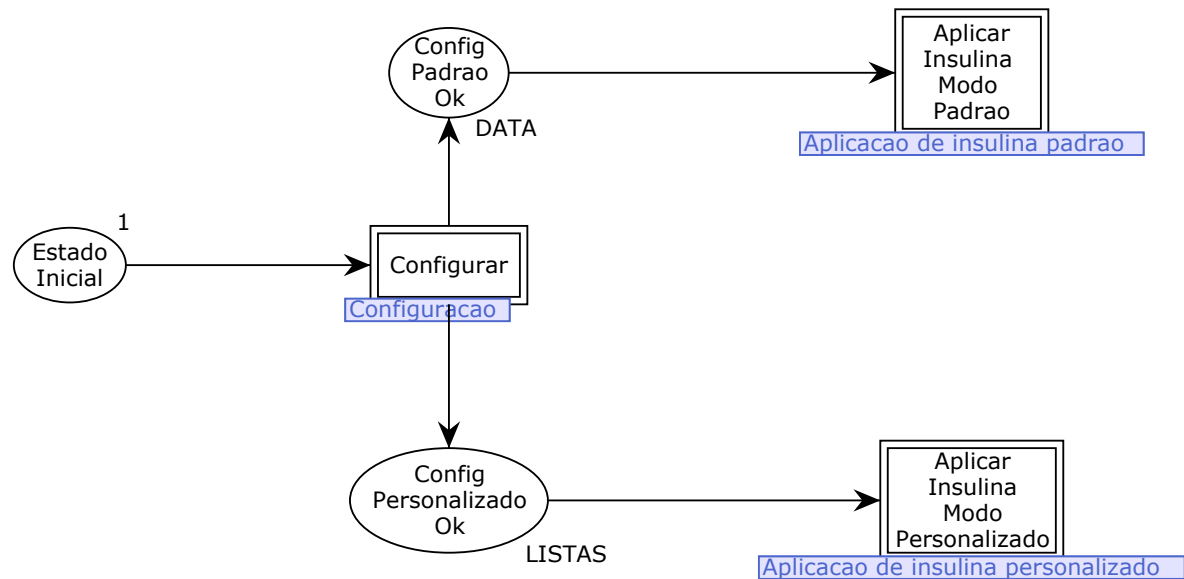
Essas duas versões de modelos foram criadas para possibilitar a análise de questões diferentes relacionadas com o funcionamento da bomba. Nelas são implementados todos os requisitos apresentados na Tabela 2. O primeiro modelo foi criado para possibilitar a aplicação da técnica de verificação de modelos, buscando evitar o problema da explosão de espaço de estados. A aplicação dessa técnica em um modelo que considera apenas dados do tipo inteiro é mais facilitada porque o espaço de estados do sistema é menor quando comparado ao modelo que considera valores numéricos reais e a variável tempo. O segundo modelo foi criado para possibilitar a análise de questões da bomba que envolvam tempo, como, por exemplo, a taxa de administração de insulina.

No Capítulo 5, essas duas versões do modelo aqui apresentado foram instanciadas para uma bomba de infusão de insulina específica. Com isso, foi possível verificar e validar, respectivamente, a primeira e a segunda versão.

### 4.1 PRIMEIRA VERSÃO DO MODELO

A primeira versão do modelo do sistema da bomba de infusão de insulina possui cinco módulos. Em cada módulo são implementadas algumas funcionalidades, sendo que a junção de todos os módulos representa o funcionamento completo do sistema. Esses módulos foram criados a partir do modelo *CPN* que representa a visão geral do sistema, o qual é ilustrado na Figura 10. Observe que a representação gráfica dos módulos estão em consonância com o seu conceito formal dado no Capítulo 2.

Figura 10 – Modelo CPN de visão geral do sistema de controle da bomba de infusão de insulina da primeira versão



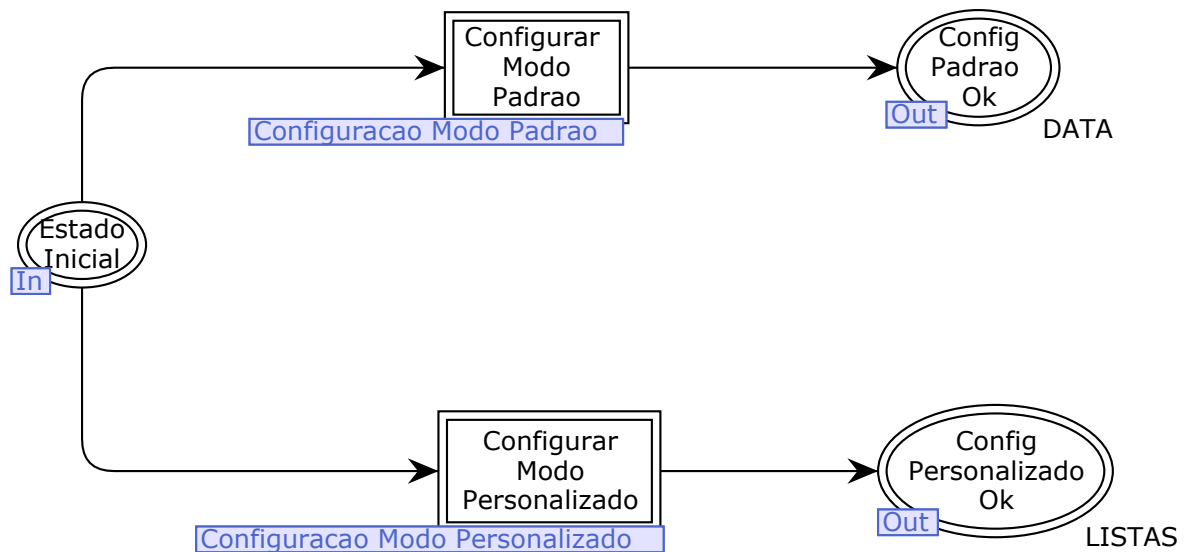
Fonte: Próprio autor, 2019.

Como pode ser observado, a visão geral do sistema é relativamente simples, envolve duas partes: uma parte é a configuração da bomba e a outra parte é a aplicação da insulina. Estas partes serão detalhadas mais adiante, por enquanto, vale destacar apenas que, para a aplicação de insulina ser liberada, a configuração da bomba deve ter sido realizada.

De forma mais detalhada, a visão geral é composta por três lugares e três transições de substituição, isto é, transições que encapsulam um módulo do sistema. O termo sob cada transição, destacado na cor azul, indica o módulo ao qual a transição está associada. Portanto, a transição Configurar está associada ao módulo Configuracao, a transição Aplicar Insulina Modo Padrao está associada ao módulo Aplicacao de insulina padrao, e a transição Aplicar Insulina Modo Personalizado está associada ao módulo Aplicacao de insulina personalizado. Cada módulo será apresentado a seguir.

O módulo Configuracao pode ser visto na Figura 11. Nesse módulo é feita a primeira etapa de configuração do sistema, que é definir o perfil basal no qual o sistema irá operar: padrão ou personalizado. Dependendo do tipo selecionado, a forma como os valores de dosagens de insulina do tipo basal são configurados diferencia. Por isso, a partir deste módulo foram criados dois outros módulos: o módulo Configuracao Modo Padrao que está associado a transição Configurar Modo Padrao e o módulo Configuracao Modo Personalizado que está associado a transição Configurar Modo Personalizado. Nestes dois módulos, além da configuração da insulina basal, é modelado também a configuração detalhada de todos os parâmetros da bomba de infusão de insulina.

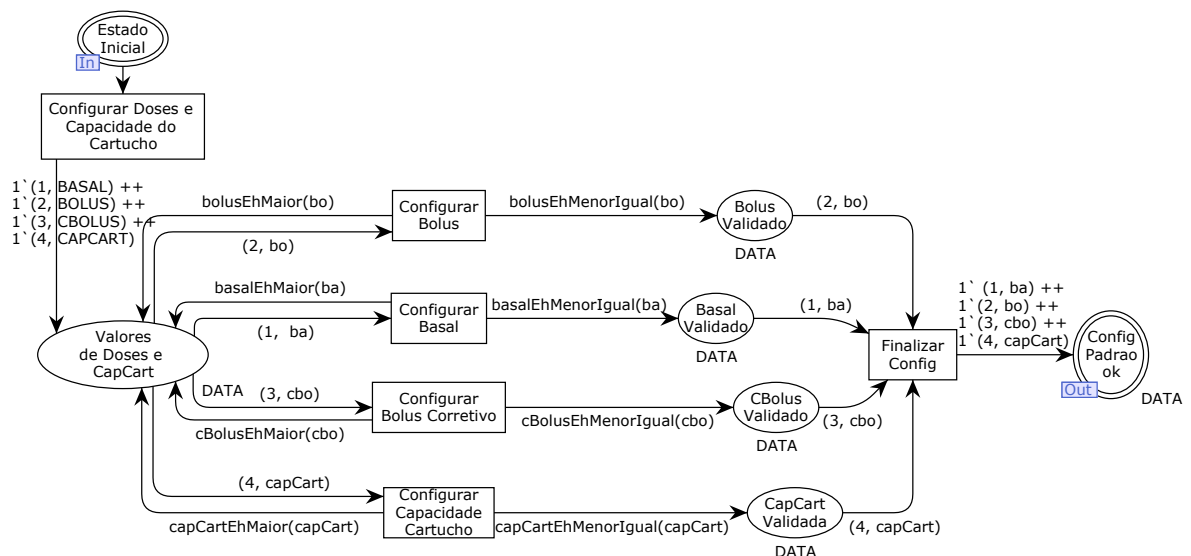
Figura 11 – Módulo de configuração da primeira versão



Fonte: Próprio autor, 2019.

No módulo Configuracao Modo Padrao, mostrado na Figura 12, por a insulina basal possuir um valor padrão, a sua configuração é feita a partir de um valor constante do tipo inteira pré-definida. Essa mesma abordagem é usada para configurar os valores de dosagens das insulinas do tipo bolus padrão e bolus corretivo, e o valor da capacidade de insulina do cartucho da bomba. Conforme a Tabela 2, alguns desses valores devem obedecer certos limiares. No modelo, para cada um desses parâmetros foram definidos limiares inferior e superior. Foram definidos também funções para verificar esses valores e impedir que eles sejam configurados fora dos limiares estabelecidos. Observe que é somente após a configuração correta de todos os parâmetros que a aplicação da insulina é liberada.

Figura 12 – Módulo de configuração do modo padrão da primeira versão



Fonte: Próprio autor, 2019.



No módulo `Configuracao Modo Personalizado`, mostrado na Figura 13, por a insulina basal ser personalizada, isto é, possuir valores diferentes para as diversas dosagens diárias do paciente, a configuração dessas dosagens é feita a partir de um arquivo, no qual os valores de dosagens são definidos. Esses valores lidos do arquivo são armazenados em uma lista para serem, posteriormente, utilizados na aplicação. A quantidade de valores e os valores que serão armazenados na lista depende da prescrição médica e é definida pelo usuário da bomba.

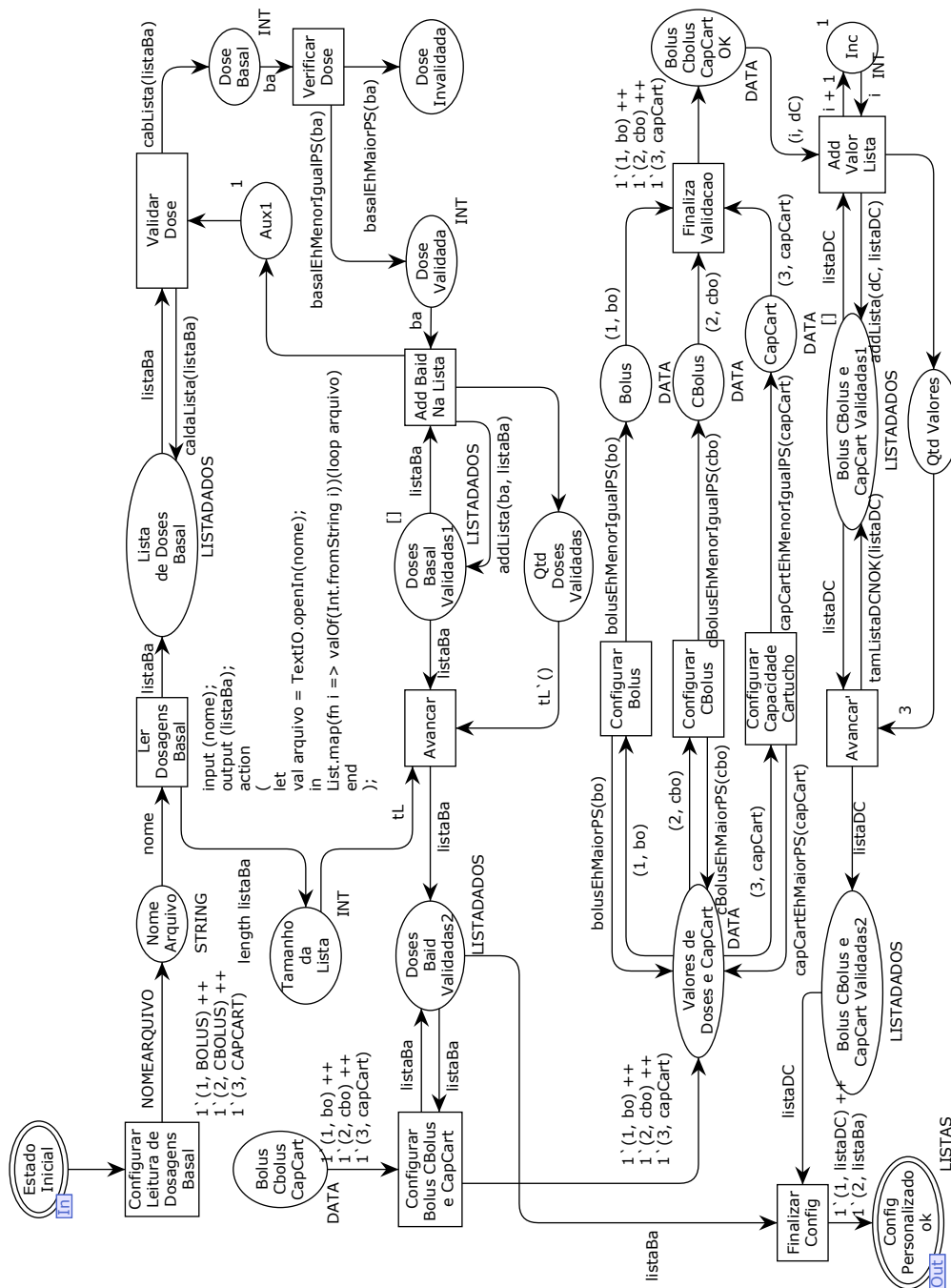
Para a configuração dos valores de dosagens das insulinas do tipos bolus e bolus corretivo, e o valor da capacidade de insulina do cartucho da bomba, é utilizada a mesma abordagem do módulo `Configuracao Modo Padrao`, a diferença passa a ser na forma como esses valores são armazenados após a configuração, que é em uma lista. Portanto, ao final de todo o processo de configuração, duas listas (`listaBa` que armazenada os valores de doses basal e `listaDC` que armazena os valores das doses bolus e bolus corretivo e da capacidade do cartucho) são geradas, as quais serão utilizadas na aplicação da insulina. Perceba que nesse módulo também é feita a verificação, via funções, dos valores que estão sendo configurados a fim de que não estejam fora de limiares pré-estabelecidos.

Na Figura 14 é apresentado o módulo `Aplicacao de insulina padrao`. Ao iniciar a aplicação, as primeiras ações do sistema são salvar os dados de configuração e informar ao usuário que a bomba está em execução para aplicação e com o cartucho preenchido. Posteriormente, a seleção das insulinas dos tipos basal, bolus e bolus corretivo é liberada. O usuário pode escolher um ou mais de um tipo de insulina por vez para ser aplicada. Os tipos escolhidos são somados gerando uma única dose que será aplicada no organismo do paciente. Vale destacar que essa dose única será aplicada se não exceder o limite de segurança de aplicação. Se a dose única for superior a esse limite, então será aplicado apenas o valor limite.

Para que a aplicação ocorra, o usuário deve selecionar a opção `Aplicar Insulina`, momento no qual as opções de seleção de insulina para aplicação são desativadas e é verificado se há insulina suficiente no cartucho da bomba para que a dosagem seja aplicada. Existindo insulina suficiente, a aplicação é liberada e a capacidade do cartucho é atualizada. Após a aplicação, as opções de seleção de insulina são novamente liberadas. Não existindo insulina suficiente, um processo de recarregamento da capacidade do cartucho é iniciado. Existem duas situações que levam a inicialização desse processo: a primeira é quando a capacidade de insulina no cartucho da bomba chegou a zero; a segunda é quando a capacidade de insulina no cartucho da bomba é inferior a dose que é escolhida para a aplicação.

Na primeira situação o sistema transita do estado de `EXECUTANDO` para o estado de

Figura 13 – Módulo de configuração do modo personalizado da primeira versão

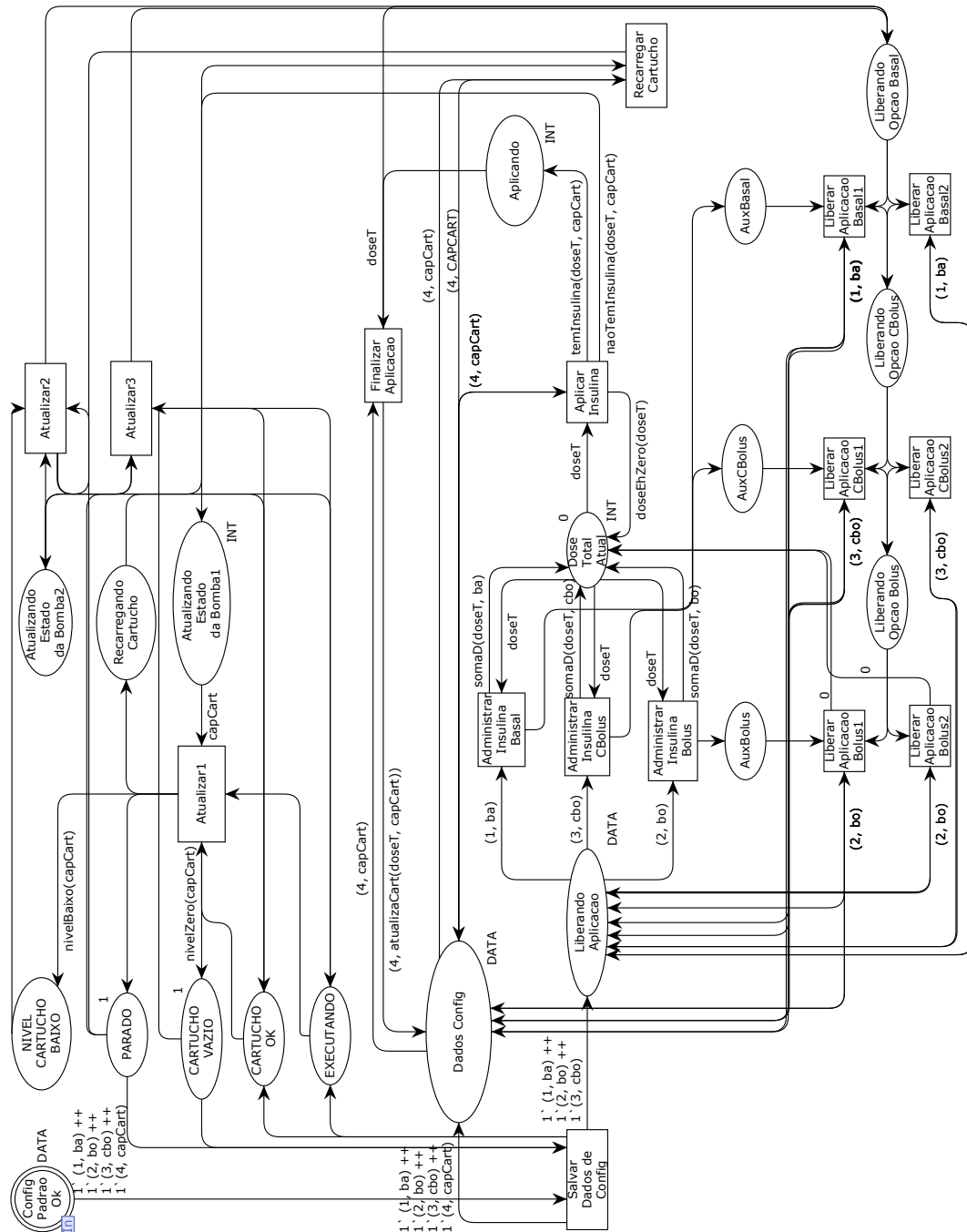


Fonte: Próprio autor, 2019.

PARADO e exibe uma mensagem informando que o cartucho da bomba está vazio. Posteriormente, o cartucho é recarregado, o sistema transita do estado de PARADO para o estado de EXECUTANDO e exibe uma mensagem informando que o cartucho da bomba está completo, e a aplicação de insulina é novamente liberada. Na segunda situação o processo é similar a primeira, o que diferencia é que a mensagem exibida após o sistema transitar do estado de EXECUTANDO para o estado de PARADO informa que o cartucho da bomba está com o nível baixo.

O módulo Aplicacao de insulina personalizado, apresentado na Figura 15, tem

Figura 14 – Módulo de aplicação de insulina do modo padrão da primeira versão

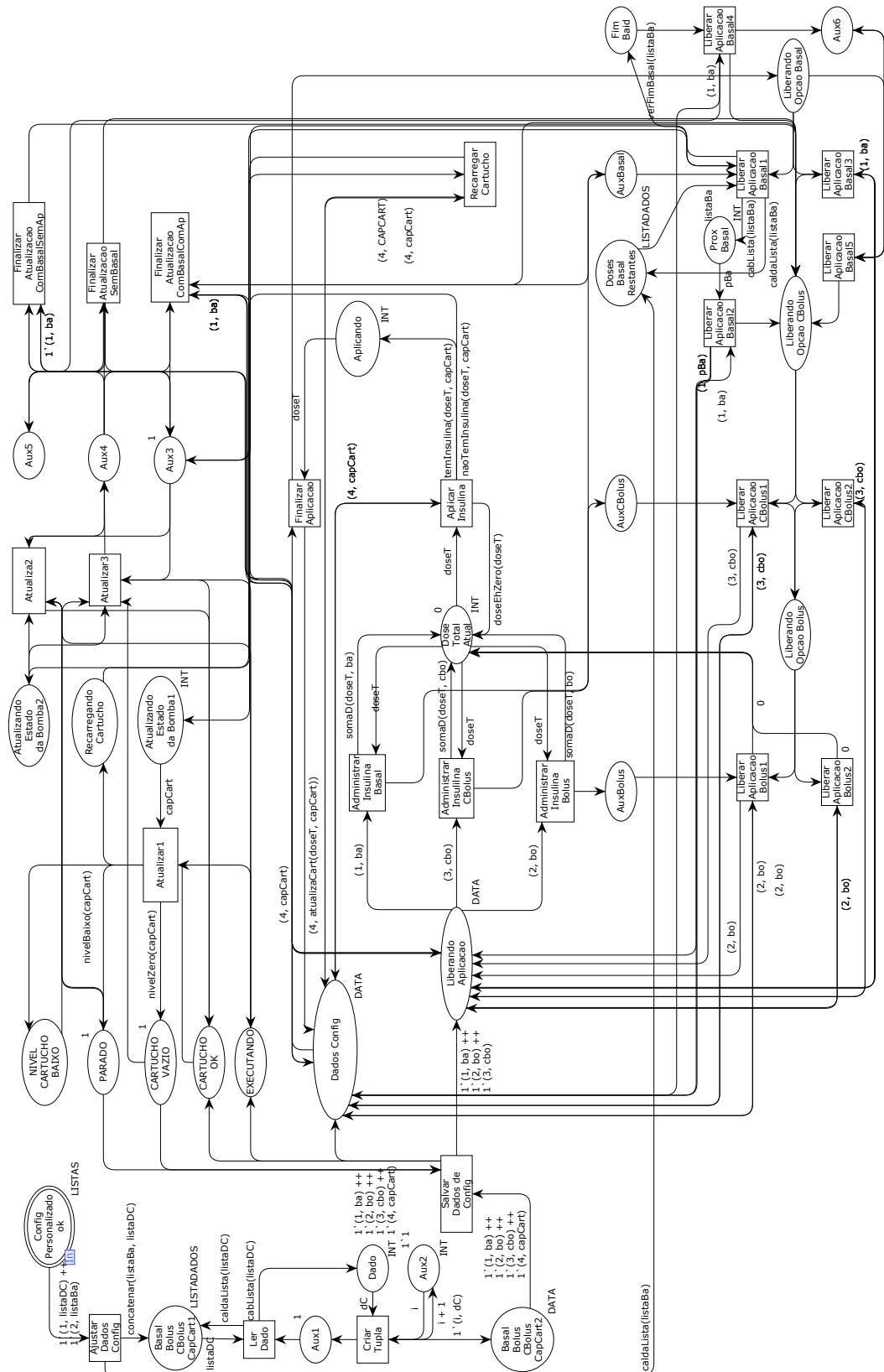


Fonte: Próprio autor, 2019.

um funcionamento similar ao módulo Aplicacao de insulina padrao, porém, por receber os dados de configuração via duas listas algumas mudanças tiveram que ser realizadas. Para construção desse módulo foi feito o reuso do módulo de aplicação padrão, a partir do qual as mudanças foram feitas.

A primeira mudança foi transformar a representação dos dados no formato de lista para o formato de duplas, o qual facilita a implementação das funcionalidades e permitiu reusar ao máximo o módulo `Aplicacao de insulina padrao`. Nesta etapa, a primeira dose de insulina

Figura 15 – Módulo de aplicação de insulina do modo personalizado da primeira versão



Fonte: Próprio autor, 2019.

basal é retirada da listaBa, a qual armazena os valores diários de insulina basal, e é adicionada no início da listaDC, que armazena os valores de doses bolus e bolus corretivo, e o valor da

capacidade de insulina do cartucho da bomba. Além disso, o restante dos valores da `listaBa` são guardados no lugar `Doses Basal Restantes` para quando for necessário aplicar uma nova dose de insulina basal o valor dessa dose ser obtido a partir desse lugar. As duplas geradas são iguais aquelas usadas para guardar os valores de configuração no módulo `Aplicacao de insulina padrao`.

Após as duplas de configuração serem salvas no lugar `Dados Config`, a aplicação é liberada e o processo de aplicação de insulina e recarregamento do cartucho ocorre de maneira muito similar ao módulo `Aplicacao de insulina padrao`, porém, com algumas modificações.

O processo de aplicação de insulina manteve-se igual até o ponto que envolve a liberação da insulina basal. Neste ponto algumas modificações foram realizadas: caso a insulina basal tenha sido aplicada e exista algum valor desse tipo de insulina no lugar `Doses Basal Restantes`, um novo valor de dosagem para a próxima aplicação é obtido e enviado ao lugar `Liberando Aplicacao`; caso não exista valores neste lugar, a aplicação de insulina desse tipo não é mais permitida, podendo o usuário aplicar apenas insulina do tipo bolus e bolus corretivo.

O processo de recarregamento do cartucho manteve-se igual até o ponto em que a atualização é finalizada e o sistema volta ao estado de `EXECUTANDO` com o `CARTUCHO OK`. Deste ponto até chegar na parte de liberação de aplicação de insulina algumas alterações foram feitas, as quais implicaram na inserção de novos lugares e transições ao modelo para possibilitar o funcionamento adequado do sistema diante das seguintes situações possíveis que ele poderia se encontrar quando foi verificado a necessidade de atualização do cartucho: a insulina do tipo basal ter sido administrada para a aplicação e ainda existir doses para serem aplicadas no futuro; a insulina do tipo basal não ter sido administrada para a aplicação e ainda existir doses para serem aplicadas no futuro; a insulina do tipo basal não ter sido selecionada para a aplicação e não existir doses para serem aplicadas no futuro.

Na Tabela 4 são apresentados os parâmetros de entrada do modelo com suas descrições e seus valores. Os valores dos parâmetros para o modelo de referência aqui apresentado são ilustrativos, podendo ser alterado para cada modelo de bomba em particular.

## 4.2 SEGUNDA VERSÃO DO MODELO

A segunda versão do modelo do sistema da bomba de infusão de insulina é uma evolução da primeira versão para incluir o controle de administração e a possibilidade de utilizar valores do tipo real (tipo de dado usado na codificação com a linguagem *ML*, o qual implementa o

Tabela 4 – Parâmetros de entrada da primeira versão do modelo genérico de bomba de infusão de insulina

| Parâmetro       | Descrição                                                   | Valor           |
|-----------------|-------------------------------------------------------------|-----------------|
| BASAL           | A dose de insulina basal                                    | 3               |
| BOLUS           | A dose de insulina bolus                                    | 3               |
| CBOLUS          | A dose de insulina bolus corretivo                          | 3               |
| CAPCART         | A capacidade do cartucho                                    | 9               |
| LIMITEDOSESUP   | O limite superior da dose de qualquer tipo de insulina      | 3               |
| LIMITEDOSEINF   | O limite inferior da dose de qualquer tipo de insulina      | 1               |
| LIMITECAPCART   | O limite da capacidade do cartucho                          | 9               |
| LIMITEAPLICACAO | O limite de aplicação de insulina                           | 6               |
| NOMEARQUIVO     | O nome do arquivo com valores de doses basal personalizadas | "BasalV1MR.txt" |

Fonte: Próprio autor, 2019.

racional). Sendo assim, essa nova versão é composta pelos mesmos módulos da primeira versão, porém, com alguns deles tendo sido expandidos para contemplar os novos requisitos.

A visão geral do sistema e os módulos Configuracao, Configuracao Modo Padrao e Configuracao Modo Personalizado permaneceram graficamente e funcionalmente iguais aos da primeira versão. Em particular, para esses dois últimos módulos, a única ressalva é que o tipo DATA, que na primeira versão é uma dupla de inteiros, agora, nessa nova versão, é uma dupla de inteiro e real. Além disso, os tipos LISTADADOS e LISTAS usados no módulo Configuracao Modo Personalizado também passaram a ser do tipo real.

As principais mudanças com relação a primeira versão estão nos módulos Aplicacao de insulina padrao e Aplicacao de insulina personalizado. Nos dois foram inseridos o controle de administração de aplicação de insulina e o controle de administração de insulina basal, os quais são apontados na Tabela 2. A inserção desses novos requisitos exigiu a definição de dois novos tipos de dados temporizados: REALTEMP e DATATEMP. Ambos foram inseridos para permitir a manipulação de valores reais temporizados e de duplas de inteiro e real temporizadas. Como a implementação desses controles de administração é semelhante entre os modos de aplicação, é apresentado nesta seção apenas o módulo Aplicacao de insulina padrao. Para ter acesso a todas as versões do modelo acesse o *site*<sup>1</sup> no qual os modelos são disponibilizados.

O novo módulo Aplicacao de insulina padrao é apresentado na Figura 16. Conforme pode ser observado, o controle de administração de aplicação de insulina é implementado na parte mais a direita do modelo, iniciando na transição Preparar Aplicacao Parcial e finalizando na transição Finalizar Aplicacao Total. Ele obedece a uma taxa de administração predefinida pelo usuário da bomba. Portanto, dado uma dose de insulina que deve ser aplicada,

<sup>1</sup> <https://sites.google.com/view/acbm/case-studies/insulin-infusion-pump>

não necessariamente ela será aplicada por completo de uma única vez, podendo ser dividida em várias partes a serem aplicadas de tempo em tempo. A quantidade de partes depende da taxa de administração, ou seja, de quantas unidades de insulina podem ser aplicados em tantas unidades de tempo. Por outro lado, o controle de administração de insulina basal é implementado por meio do lugar `Tempo Adm Basal Aux` (armazena o tempo de administração de insulina basal) e seus arcos de entradas e saídas. Após ser feita a aplicação da insulina basal, no momento em que ocorre a liberação para que ela seja novamente aplicada (isso ocorre quando a transição `Liberar Aplicacao Basal1` é disparada), o próximo tempo de aplicação desse tipo de insulina é obtido do lugar `Tempo Adm Basal Aux` e a próxima dose basal é configurada. Nesse mesmo momento, o próximo tempo em que a insulina basal pode ser aplicada é enviado para o lugar `Tempo Adm Basal Aux` para possibilitar que na próxima vez que a insulina basal for aplicada seja feita a sua correta configuração.

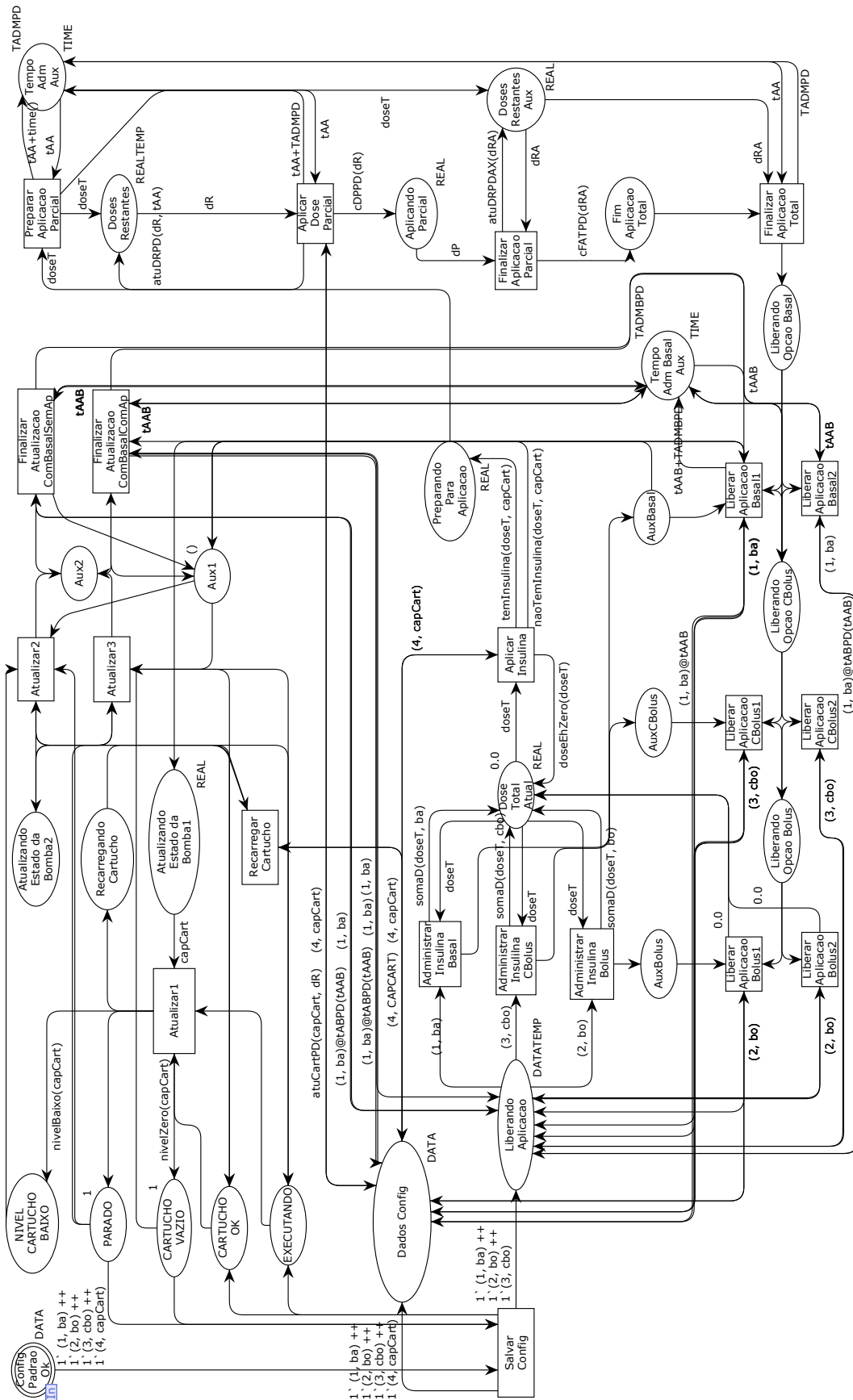
Na Tabela 5 são apresentados os parâmetros de entrada da segunda versão do modelo com suas descrições e seus valores. Como pode ser visto, os parâmetros da Tabela 4 foram mantidos, sendo acrescentados mais seis parâmetros, os quais são utilizados pelo novos requisitos implementados. Novamente, os valores dos parâmetros são ilustrativos, podendo ser alterado para cada modelo de bomba em particular.

Tabela 5 – Parâmetros de entrada da segunda versão do modelo genérico de bomba de infusão de insulina

| Parâmetro       | Descrição                                                          | Valor           |
|-----------------|--------------------------------------------------------------------|-----------------|
| BASAL           | A dose de insulina basal                                           | 3.0             |
| BOLUS           | A dose de insulina bolus                                           | 3.0             |
| CBOLUS          | A dose de insulina bolus corretivo                                 | 3.0             |
| CAPCART         | A capacidade do cartucho                                           | 9.0             |
| LIMITEDOSESUP   | O limite superior da dose de qualquer tipo de insulina             | 3.0             |
| LIMITEDOSEINF   | O limite inferior da dose de qualquer tipo de insulina             | 1.0             |
| LIMITECAPCART   | O limite da capacidade do cartucho                                 | 9.0             |
| LIMITEAPLICACAO | O limite de aplicação de insulina                                  | 6.0             |
| NOMEARQUIVO     | O nome do arquivo com valores de doses basal personalizadas        | "BasalV2MR.txt" |
| DADMPD          | A dose de administração de insulina no perfil basal padrão         | 1.0             |
| TADMPD          | O tempo de administração de insulina no perfil basal padrão        | 1               |
| TADMBPD         | O tempo de administração de insulina basal no perfil padrão        | 3               |
| DADMPS          | A dose de administração de insulina no perfil basal personalizado  | 1.0             |
| TADMPS          | O tempo de administração de insulina no perfil basal personalizado | 1               |
| TADMBPS         | O tempo de administração de insulina basal no perfil personalizado | 3               |

Fonte: Próprio autor, 2019.

Figura 16 – Módulo de aplicação de insulina do modo padrão da segunda versão



Fonte: Próprio autor, 2019.



## 5 ESTUDO DE CASO

Neste capítulo é apresentado o estudo de caso da bomba de infusão de insulina *ACCU-CHECK Spirit* versão 2.XX, a qual é usada para estender o modelo de referência apresentado no Capítulo 4. O estudo de caso é uma forma de demonstrar como utilizar o modelo de referência, além de uma forma de verificar e validar o modelo construído.

O estudo de caso da bomba *ACCU-CHECK Spirit* versão 2.XX envolve a verificação e a validação, respectivamente, da primeira e da segunda versão do modelo de referência. Para isso foi necessário instanciar o modelo de referência com os parâmetros específicos da bomba. Esses parâmetros são apresentados na Tabela 6. Eles foram obtidos a partir do manual da bomba e do trabalho realizado por Silva (2015), o qual é brevemente apresentado no Capítulo 3.

Tabela 6 – Requisitos técnicos do equipamento *ACCU-CHECK Spirit* versão 2.XX

| Parâmetros de Configuração                                                                                                                                                                                                                                                                                                                                                                                                      | Funcionalidades                                                                                                                                                  | Propriedades de Segurança                                                                                                           |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| 1. Tipo de Insulina: U100 <sup>(1)</sup><br>2. Capacidade do Cartucho: 3,15ml <sup>(2)</sup><br>3. Dose Basal (U/h):<br>Min = 0,1<br>Max = 25,0<br>4. Perfil Basal:<br>a. Padrão (# doses fixas/3min):480<br>b. Personalizado (#doses ajustadas/h):24<br>5. Bolus Padrão (U): Max = 25,0<br>6. Taxa de Administração (dose/min) <sup>(3)</sup> :<br>a. Dose Basal<br>b. Dose Bolus<br>7. Limite de Aplicação (U):<br>Max = 25,0 | 1. Modos Executar/Parar<br>2. Selecionar perfil basal<br>3. Verificar nível de cartucho<br>4. Programar dose de bolus padrão<br>5. Prover informações ao usuário | 1. Erros(E)/Alertas(A)<br>a. E: "Cartucho Vazio!"<br>b. A: "Nível Baixo do Cartucho!"<br>2. Estado:<br>a. "STOP"<br>b. "EXECUTANDO" |

<sup>(1)</sup> Insulina de ação curta/rápida

<sup>(2)</sup> Equivalente a 315U de insulina U100

<sup>(3)</sup> Simplificação, uma vez que na especificação adotada a taxa de insulina 0,2U/s

Fonte: Adaptado de Silva (2015).

Pode-se observar que todos esses parâmetros de configuração foram definidos no modelo genérico da bomba. Em particular, vale destacar que o caractere U refere-se a unidade de insulina. Para a versão que é verificada foram usados apenas os parâmetros que não envolvem tempo, ou seja, aqueles definidos na Tabela 4. Por outro lado, para a versão que é validada foram usados todos aqueles parâmetros definidos na Tabela 5.

## 5.1 VERIFICAÇÃO DA PRIMEIRA VERSÃO DO MODELO

A verificação do *ACCU-CHECK Spirit* versão 2.XX foi realizada usando a técnica de verificação automática de modelos (apresentada no Capítulo 2). Foram verificadas duas propriedades de segurança relevantes para a bomba de infusão de insulina, as quais são apresentadas na Tabela 7 com as suas respectivas fórmulas *ASK-CTL*.

Tabela 7 – Propriedades de segurança do *ACCU-CHECK Spirit* versão 2.XX

| ID | Propriedade                                                                                                                    | Fórmula <i>ASK-CTL</i>                                                                                                                                                                        |
|----|--------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1  | Se o nível do cartucho for igual a 0, então o estado do cartucho deve tornar-se “VAZIO” e o estado da bomba deve ser “PARADO”. | $\text{OR}(\text{NOT}(\text{EV}(\text{NF}(\text{"a\_"}, \text{cartZerado}))), \text{EV}(\text{AND}(\text{NF}(\text{"c\_"}, \text{bombaEmStop}), \text{NF}(\text{"d\_"}, \text{cartVazio}))))$ |
| 2  | Se o nível do cartucho for menor do que a dosagem de insulina administrada, então o estado da bomba deve ser “PARADO”.         | $\text{OR}(\text{NOT}(\text{EV}(\text{NF}(\text{"a\_"}, \text{nivelCartBaixo}))), \text{EV}(\text{NF}(\text{"c\_"}, \text{bombaEmStop})))$                                                    |

Fonte: Próprio autor, 2019.

A descrição de ambas as propriedades sugere o uso de uma fórmula lógica composta usando o conectivo de implicação. Entretanto, como o *ASK-CTL* não disponibiliza esse conectivo lógico, a seguinte equivalência lógica foi utilizada:  $\varphi \rightarrow \psi \equiv \neg\varphi \vee \psi$ . Ou seja, as propriedades foram definidas inicialmente com o uso da implicação, e, posteriormente, a equivalência foi aplicada para permitir a representação da propriedade com *ASK-CTL*. Isso permitiu a realização da verificação formal do modelo.

### 5.1.1 Propriedade 1

A primeira propriedade define que, quando o nível do cartucho for igual a 0, o estado do cartucho deve tornar-se “VAZIO” e o estado da bomba de ser “PARADO”. Para a formalização dessa propriedade em *ASK-CTL*, primeiro, observou-se que ela é composta pelas seguintes proposições:

- $\varphi$  = o nível do cartucho é igual a 0;
- $\psi_1$  = o estado do cartucho é “VAZIO”;
- $\psi_2$  = o estado da bomba é “PARADO”.

Posteriormente, foi definido a fórmula  $\varphi \rightarrow (\psi_1 \wedge \psi_2)$ . Aplicando a equivalência lógica, temos  $\neg\varphi \vee (\psi_1 \wedge \psi_2)$ . Feito isso, essa fórmula foi simplesmente traduzida para *ASK-CTL*, conforme ilustra a Figura 17, na qual é também apresentado o resultado da execução da verificação da

propriedade usando o *CPN/Tools*. As proposições  $\varphi$ ,  $\psi_1$  e  $\psi_2$  equivalem, respectivamente, as proposições *cartZerado*, *cartVazio* e *bombaEmStop*. Essas proposições estão associados, respectivamente, com as instâncias de lugares *Aplicacao\_de\_insulina\_padrao'Dados\_Config*, *Aplicacao\_de\_insulina\_padrao'CARTUCHO\_VAZIO* e *Aplicacao\_de\_insulina\_padrao'PARADO* para a verificação da propriedade 1 no modo de aplicação de insulina padrão. Para a verificação da propriedade 1 no modo de aplicação de insulina personalizada, a mudança ocorre apenas nas identificações das instâncias de lugares.

Figura 17 – Propriedade 1  
(a) Propriedade 1 para o modo de aplicação padrão

```
val cartZerado = fn : Node -> bool
val bombaEmStop = fn : Node -> bool
val cartVazio = fn : Node -> bool
val myASKCTLformula =
  OR
    (NOT (FORALL_UNTIL (TT,NF ("a_",fn))),
     FORALL_UNTIL (TT,NOT (OR (NOT (NF ("c_",fn)),NOT (NF ("d_",fn)))))) : A
val it = true : bool

fun cartZerado b = Mark.Aplicacao_de_insulina_padrao'Dados_Config 1 b = 1` (4,0);
fun bombaEmStop r = Mark.Aplicacao_de_insulina_padrao'PARADO 1 r = 1` ();
fun cartVazio s = Mark.Aplicacao_de_insulina_padrao'CARTUCHO_VAZIO 1 s = 1` ();

val myASKCTLformula = OR(
  NOT(EV(NF("a_",cartZerado))),
  EV( AND( NF("c_",bombaEmStop), NF("d_",cartVazio)))
);
eval_node myASKCTLformula InitNode;
```

(b) Propriedade 1 para o modo de aplicação personalizado

```
val cartZerado = fn : Node -> bool
val bombaEmStop = fn : Node -> bool
val cartVazio = fn : Node -> bool
val myASKCTLformula =
  OR
    (NOT (FORALL_UNTIL (TT,NF ("a_",fn))),
     FORALL_UNTIL (TT,NOT (OR (NOT (NF ("c_",fn)),NOT (NF ("d_",fn)))))) : A
val it = true : bool

fun cartZerado b = Mark.Aplicacao_de_insulina_personalizado'Dados_Config 1 b = 1` (4,0);
fun bombaEmStop r = Mark.Aplicacao_de_insulina_personalizado'PARADO 1 r = 1` ();
fun cartVazio s = Mark.Aplicacao_de_insulina_personalizado'CARTUCHO_VAZIO 1 s = 1` ();

val myASKCTLformula = OR(
  NOT(EV(NF("a_",cartZerado))),
  EV(AND(NF("c_",bombaEmStop),NF("d_",cartVazio)))
);
eval_node myASKCTLformula InitNode;
```

Fonte: Próprio autor, 2019.

A diferença entre a fórmula *ASK-CTL* com relação aquela escrita em lógica proposicional está na utilização dos operadores *EV* e *NF*. O primeiro é usado para especificar que a fórmula é satisfeita para todos os caminhos do espaço de estados, em algum momento no futuro. O segundo recebe uma *string* e uma proposição como argumentos, e especifica que essa proposição deve estar associada a um nó no espaço de estados. Dado que a fórmula foi definida, a função `eval_node` disponível na biblioteca *ASK-CTL* é aplicada para executar a verificação. A verificação retornou como resultado o valor `true`, o qual significa que a propriedade está contemplada pelo modelo especificado.

### 5.1.2 Propriedade 2

A segunda propriedade define que, quando o nível do cartucho for menor do que a dosagem de insulina administrada, o estado da bomba deve ser “PARADO”. A formalização dessa propriedade em *ASK-CTL* seguiu os mesmos passos usados para formalizar a propriedade 1. Primeiro, observou-se que ela é composta pelas seguintes proposições:

- $\varphi$  = o nível do cartucho é menor do que a dosagem de insulina administrada;
- $\psi$  = o estado da bomba é “PARADO”.

Posteriormente, foi definido a fórmula  $\varphi \rightarrow \psi$ . Novamente, aplicando a equivalência lógica, temos  $\neg\varphi \vee \psi$ . Com isto feito, essa fórmula foi simplesmente traduzida para *ASK-CTL*, conforme ilustra a Figura 18, na qual é também apresentado o resultado da execução da verificação da propriedade usando o *CPN/Tools*. As proposições  $\varphi$  e  $\psi$  equivalem, respectivamente, as proposições `nivelCartBaixo` e `bombaEmStop`. Essas proposições estão associados, respectivamente, com as instâncias de lugares `Aplicacao_de_insulina_padrao'Atualizando_Estado_da_Bomba1` e `Aplicacao_de_insulina_padrao'PARADO` para a verificação da propriedade 2 no modo de aplicação de insulina padrão. Para a verificação da propriedade 2 no modo de aplicação de insulina personalizada, a mudança está apenas nos nomes das instâncias de lugares.

Assim como para a primeira propriedade, foi usado também os operadores *EV* e *NF*. Com a fórmula *ASK-CTL* definida para a propriedade 2, foi executada a verificação do modelo, a qual também retornou como resultado o valor `true` que indica a satisfação da fórmula no modelo. Isso significa que a propriedade está contemplada pelo modelo especificado.

Figura 18 – Propriedade 2  
(a) Propriedade 2 para o modo de aplicação padrão

```
val nivelCartBaixo = fn : Node -> bool
val bombaEmStop = fn : Node -> bool
val myASKCTLformula =
  OR (NOT (FORALL_UNTIL (TT,NF ("a_",fn))),FORALL_UNTIL (TT,NF ("c_",fn))) : A
val it = true : bool

fun nivelCartBaixo i = Mark.Aplicacao_de_insulina_padrao'Atualizando_Estado_da_Bomba1 1 i <> [];
fun bombaEmStop r = Mark.Aplicacao_de_insulina_padrao'PARADO 1 r = 1` ();

val myASKCTLformula = OR(
  NOT(EV(NF("a_",nivelCartBaixo))),
  EV(NF("c_",bombaEmStop))
);
eval_node myASKCTLformula InitNode;
```

(b) Propriedade 2 para o modo de aplicação personalizado

```
val nivelCartBaixo = fn : Node -> bool
val bombaEmStop = fn : Node -> bool
val myASKCTLformula =
  OR (NOT (FORALL_UNTIL (TT,NF ("a_",fn))),FORALL_UNTIL (TT,NF ("c_",fn))) : A
val it = true : bool

fun nivelCartBaixo i = Mark.Aplicacao_de_insulina_personalizado'Atualizando_Estado_da_Bomba1 1 i <> [];
fun bombaEmStop r = Mark.Aplicacao_de_insulina_personalizado'PARADO 1 r = 1` ();

val myASKCTLformula = OR(
  NOT(EV(NF("a_",nivelCartBaixo))),
  EV(NF("c_",bombaEmStop))
);
eval_node myASKCTLformula InitNode;
```

Fonte: Próprio autor, 2019.

## 5.2 VALIDAÇÃO DA SEGUNDA VERSÃO DO MODELO

A validação da segunda versão do modelo foi realizada utilizando os recursos de simulação disponíveis na ferramenta *CPN/Tools*. Nesta seção é apresentada apenas a validação das funcionalidades que envolvem tempo do modo de aplicação personalizado do *ACCU-CHEK Spirit* versão 2.XX, com capturas de telas da simulação partindo da configuração do sistema até a etapa de aplicação. No entanto, o modelo foi disponibilizado para que o usuário possa realizar a sua validação por completo<sup>1</sup>. Os valores dos parâmetros do modelo foram inicializados em conformidade com a especificação do equipamento (apresentados na Tabela 8).

A convenção adotada para a representação do tempo no modelo é que 1 unidade de tempo corresponde a 1 segundo, 60 unidades de tempo correspondem a 1 minuto e 3600 unidades de tempo correspondem a 1 hora. Sendo assim, o valor 180 atribuído ao tempo de administração de insulina basal no perfil padrão corresponde a 3 minutos e o valor 3600 atribuído ao tempo de

<sup>1</sup> <https://sites.google.com/view/acbm/case-studies/insulin-infusion-pump>

administração de insulina basal no perfil personalizado corresponde a 1 hora.

O parâmetro `NOMEARQUIVO` corresponde a um variável do tipo `STRING` que armazena o nome do arquivo no qual constam as doses de insulina basal personalizadas. No arquivo utilizado, constam 24 doses de insulina desse tipo com os seguintes valores: 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0. Esses valores foram definidos arbitrariamente apenas para possibilitar a validação do modelo via simulação. Apesar dos valores definidos não serem representativos, eles são úteis pois permitem atingir ao objetivo, que é validar as funcionalidades implementadas no modelo.

Tabela 8 – Parâmetros de entrada da segunda versão do *ACCU-CHECK Spirit* versão 2.XX

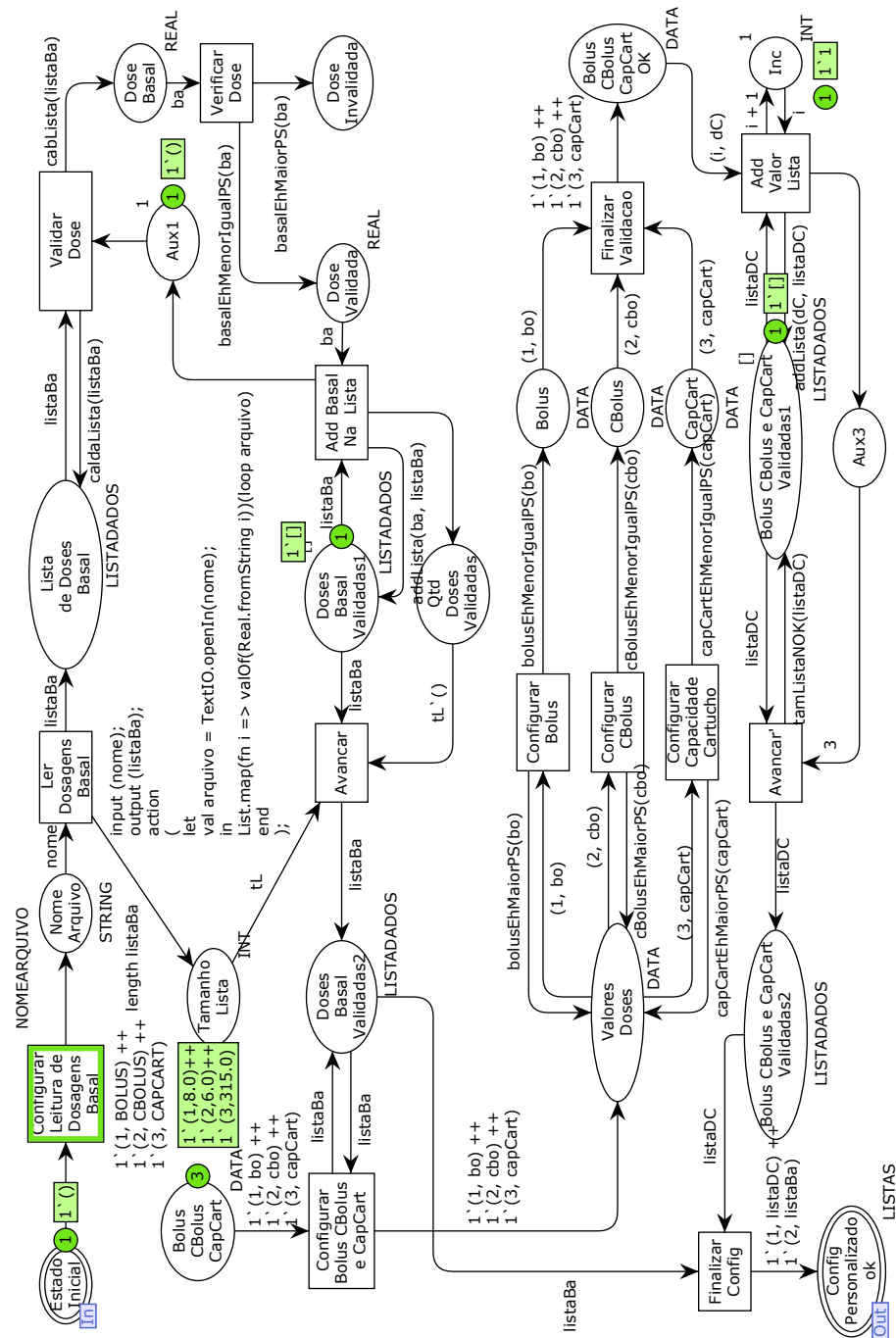
| Parâmetro       | Descrição                                                          | Valor           |
|-----------------|--------------------------------------------------------------------|-----------------|
| BASAL           | A dose de insulina basal                                           | 10.0            |
| BOLUS           | A dose de insulina bolus                                           | 8.0             |
| CBOLUS          | A dose de insulina bolus corretivo                                 | 6.0             |
| CAPCART         | A capacidade do cartucho                                           | 315.0           |
| LIMITEDOSESUP   | O limite superior da dose de qualquer tipo de insulina             | 25.0            |
| LIMITEDOSEINF   | O limite inferior da dose de qualquer tipo de insulina             | 0.1             |
| LIMITECAPCART   | O limite da capacidade do cartucho                                 | 315.0           |
| LIMITEAPLICACAO | O limite de aplicação de insulina                                  | 25.0            |
| NOMEARQUIVO     | O nome do arquivo com valores de doses basal personalizadas        | "BasalV2AC.txt" |
| DADMPD          | A dose de administração de insulina no perfil basal padrão         | 0.2             |
| TADMPD          | O tempo de administração de insulina no perfil basal padrão        | 1               |
| TADMBPD         | O tempo de administração de insulina basal no perfil padrão        | 180             |
| DADMPS          | A dose de administração de insulina no perfil basal personalizado  | 0.2             |
| TADMPS          | O tempo de administração de insulina no perfil basal personalizado | 1               |
| TADMBPS         | O tempo de administração de insulina basal no perfil personalizado | 3600            |

Fonte: Próprio autor, 2019.

Na Figura 19 é apresentada a marcação inicial do módulo de configuração do modo personalizado do *ACCU-CHECK Spirit* versão 2.XX, que é parte da marcação inicial do sistema, a qual envolve a valoração de todos os lugares do sistema. O lugar `Bolus` `CBolus` `CapCart` está marcado com três duplas, sendo elas referentes, respectivamente, a dose de insulina bolus, a dose de insulina bolus corretivo e a capacidade do cartucho da bomba. Esses valores são validados após todos os valores de doses basal personalizados também serem validados. Se algum valor de dose basal do arquivo não for válido, o restante da configuração não é realizada e, conseqüentemente, a aplicação de insulina não pode ser feita. Como descrito no Capítulo 4, os valores de doses basal são adicionados à `listaBa` e os demais valores de configuração são adicionados à `listaDC`.

Na Figura 20 é ilustrado o resultado do disparo da transição `Ler_Dosagens_Basal`. Os

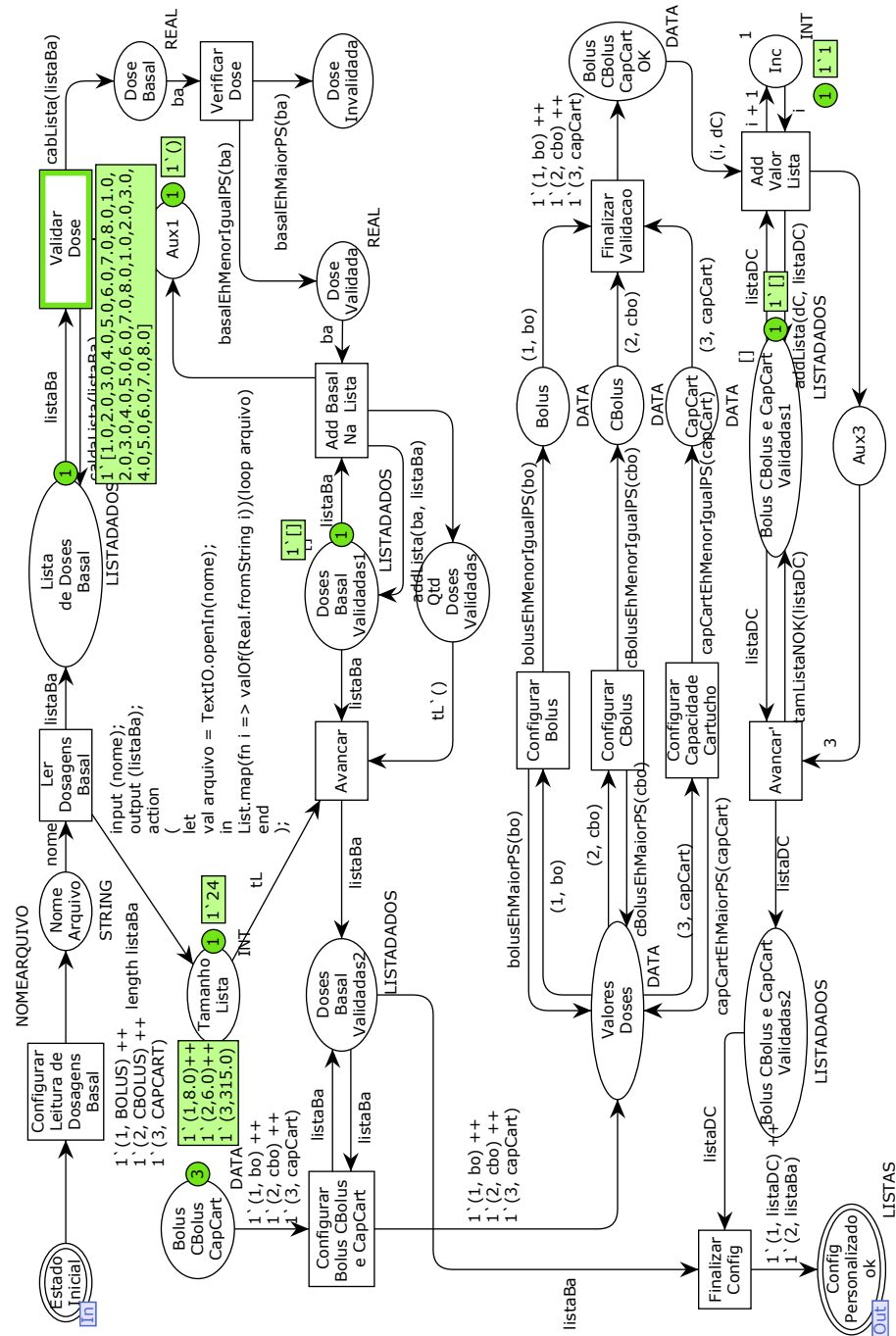
Figura 19 – Captura de tela 1 do módulo de configuração do modo personalizado do *ACCU-CHECK Spirit* versão 2.XX



Fonte: Próprio autor, 2019.

valores de doses basal são lidos do arquivo e adicionados ao lugar `Lista de Doses Basal` em formato de lista, a partir do qual eles serão um a um validados. Após a validação, a `listaBa` fica armazenada no lugar `Doses Basal Validadas2` enquanto os demais valores de configuração são também validados. Ao final do processo de configuração, as listas `listaBa` e `listaDC` são armazenadas no lugar `Config Personalizado ok`. Nesse momento, o módulo de aplicação de insulina no modo personalizado do *ACCU-CHECK Spirit* versão 2.XX é habilitado.

Figura 20 – Captura de tela 2 do módulo de configuração do modo personalizado do *ACCU-CHECK Spirit* versão 2.XX



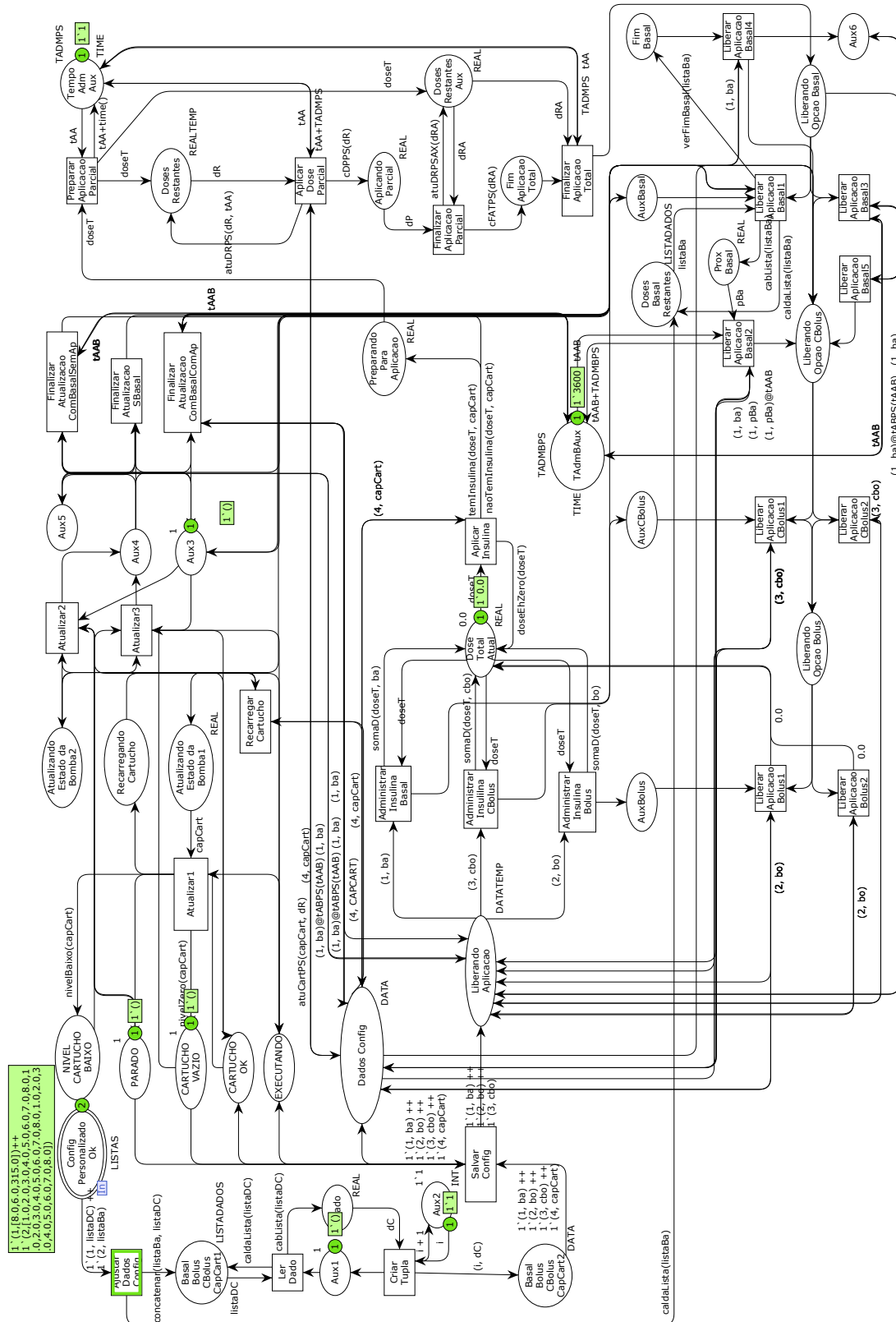
Fonte: Próprio autor, 2019.

Nesse módulo, apresentado na Figura 21, antes da liberação da aplicação, é feito um ajuste nos dados recebidos em formato de listas. Da `listaBa` é retirado o primeiro valor e o restante dos valores são armazenados em formato de lista no lugar `Doses Basal Restantes`, do qual eles serão obtidos quando for necessário definir um novo valor de dose basal para ser aplicado. O primeiro valor retirado é concatenado na cabeça da `listaDC`, que é armazenada no lugar `Basal Bolus CBolus CapCart1`, do qual serão retirados um a um e convertidos para o



formato de duplas.

Figura 21 – Captura de tela 1 do módulo de aplicação de insulina do modo personalizado do *ACCU-CHECK Spirit* versão 2.XX



Fonte: Próprio autor, 2019.

Observe que cada dupla é gerada sempre a partir do valor da cabeça da lista, de modo que apenas quatro duplas são geradas. A primeira corresponde a dupla de configuração da primeira dose basal, a segunda corresponde a dupla de configuração da dose bolus, a terceira corresponde a dupla de configuração da dose bolus corretivo e a quarta corresponde a dupla de configuração da capacidade do cartucho. Essas duplas são mantidas nos lugares `Dados Config` e `Liberando Aplicacao` e aplicação de insulina é finalmente liberada.

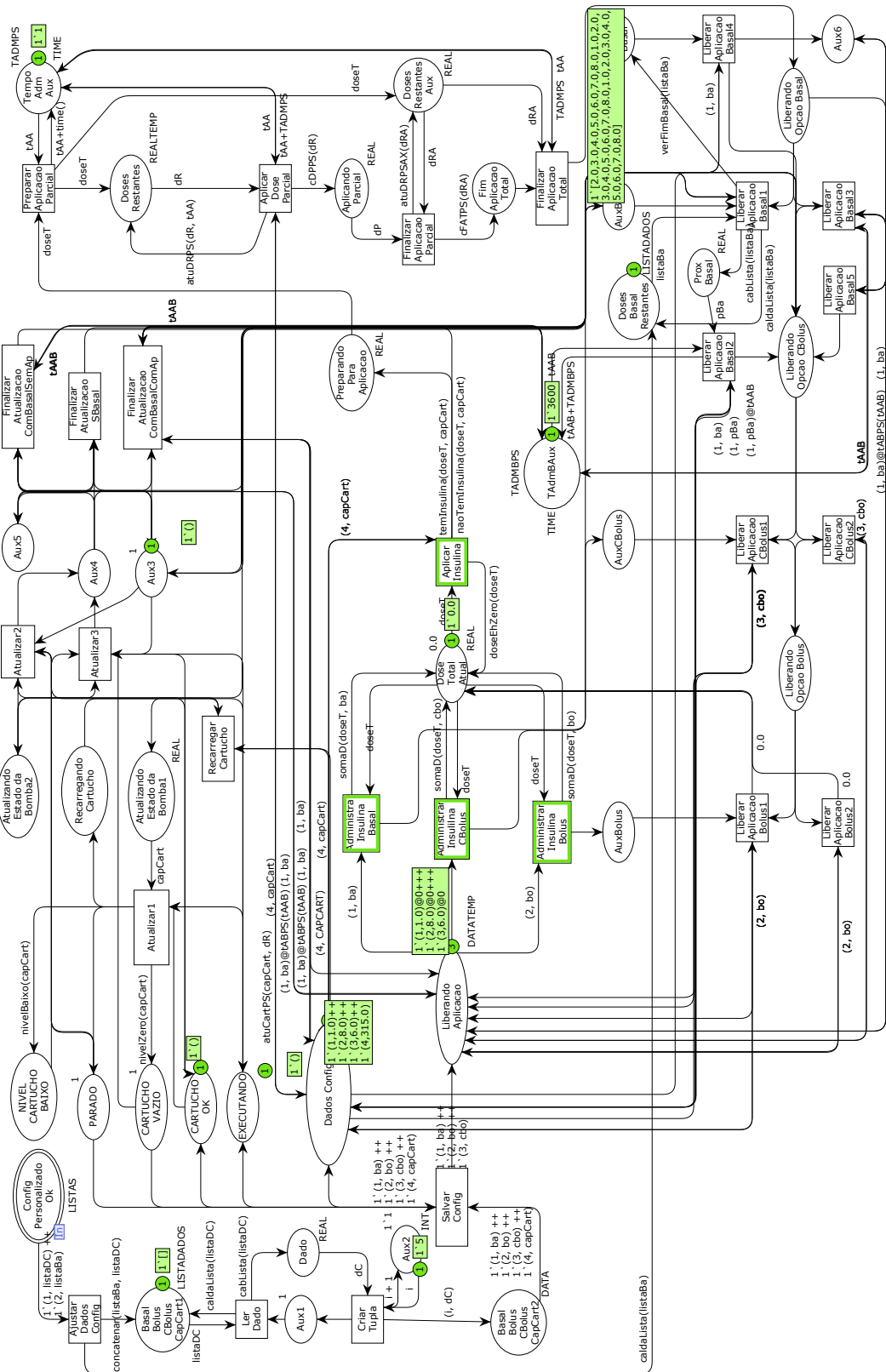
Na Figura 22 ilustrado o momento inicial da liberação. Nesse instante, qualquer um dos tipos de insulina podem ser administradas. Quando ocorre o disparo da transição `Aplicar Insulina`, se o valor armazenado no lugar `Dose Total Atual` for maior que zero, as transições de administração são desabilitadas e, havendo insulina suficiente no cartucho da bomba, o processo de aplicação é iniciado.

Na Figura 23 é ilustrado o processo de aplicação de insulina. Veja que, para essa situação apresentada, os três tipos de insulina foram selecionados, pois a soma total de insulina a ser aplicada corresponde a soma dos valores de dosagens apresentados na Figura 22. A aplicação das 15 unidades de insulina presentes no lugar `Doses Restantes` obedece a taxa de administração definida na Tabela 6, ou seja, 0,2U/s. Sendo assim, a cada disparo da transição `Aplicar Dose Parcial` é retirado do lugar `Doses Restantes` apenas 0,2 unidades de insulina. A habilitação dessa transição ocorre a cada 1 unidade de tempo (ou, 1 segundo), que é o tempo necessário para a aplicação de um dose parcial. Quando não há mais insulina para ser aplicada, o processo de aplicação encerra e as transições de administração de insulina são novamente habilitadas. Observe que a cada aplicação parcial o nível de insulina do cartucho da bomba é atualizado.

O controle do tempo para a próxima dose parcial é feito por meio do lugar `Tempo Adm Aux` e da função `atuDRPS(dR, tAA)`. Antes de iniciar o processo de aplicação de insulina em si, especificamente quando a transição `Preparar Aplicacao Parcial` é disparada, o valor atual do relógio somado com o tempo de administração atribuído a constante `TADMPS`, é guardado no lugar `Tempo Adm Aux`. A cada disparo da transição `Aplicar Dose Parcial`, esse valor é obtido e passado como argumento para a função `atuDRPS(dR, tAA)` escrita no arco entrada `Doses Restantes`. Essa função calcula o novo valor de doses restantes e marca esse valor com o tempo `tAA`, que é o próximo tempo no qual uma nova dose parcial pode ser aplicada. Ainda, com o disparo `Aplicar Dose Parcial`, o novo próximo tempo de aplicação é definido (`tAA+TADMPS`) e guardado no lugar `Tempo Adm Aux`.

Ao término da aplicação é iniciado o processo de liberação das transições de administração de insulina. Na Figura 24 é ilustrado parte desse processo. Em particular, para a liberação

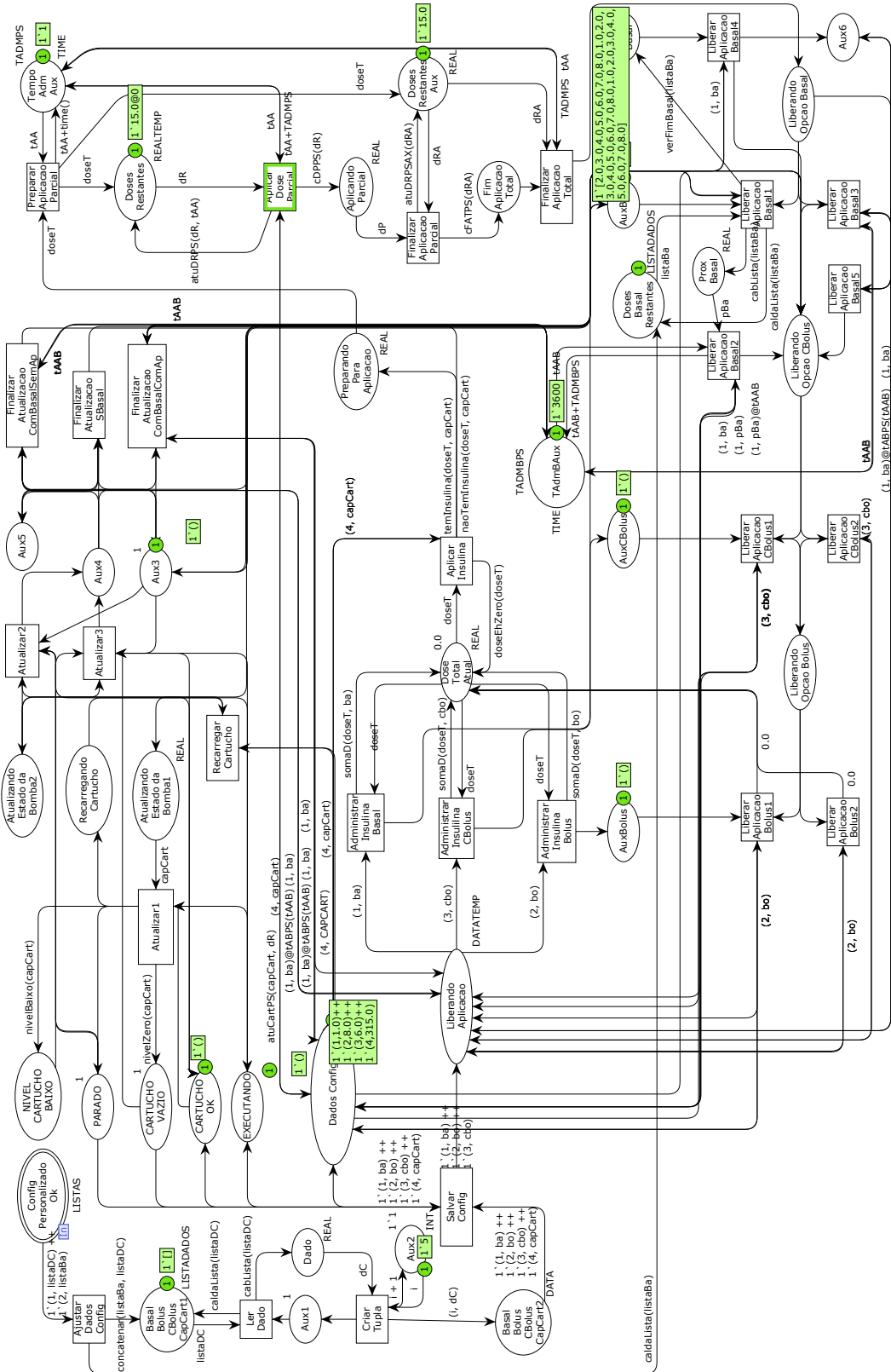
Figura 22 – Captura de tela 2 do módulo de aplicação de insulina do modo personalizado do *ACCU-CHECK Spirit* versão 2.XX



Fonte: Próprio autor, 2019.

da administração de insulina basal (a única que possui uma taxa de administração), primeiro é obtido do lugar Doses Basal Restantes a próxima dose basal que deve ser aplicada (a dose

Figura 23 – Captura de tela 3 do módulo de aplicação de insulina do modo personalizado do *ACCU-CHECK Spirit* versão 2.XX



Fonte: Próprio autor, 2019.

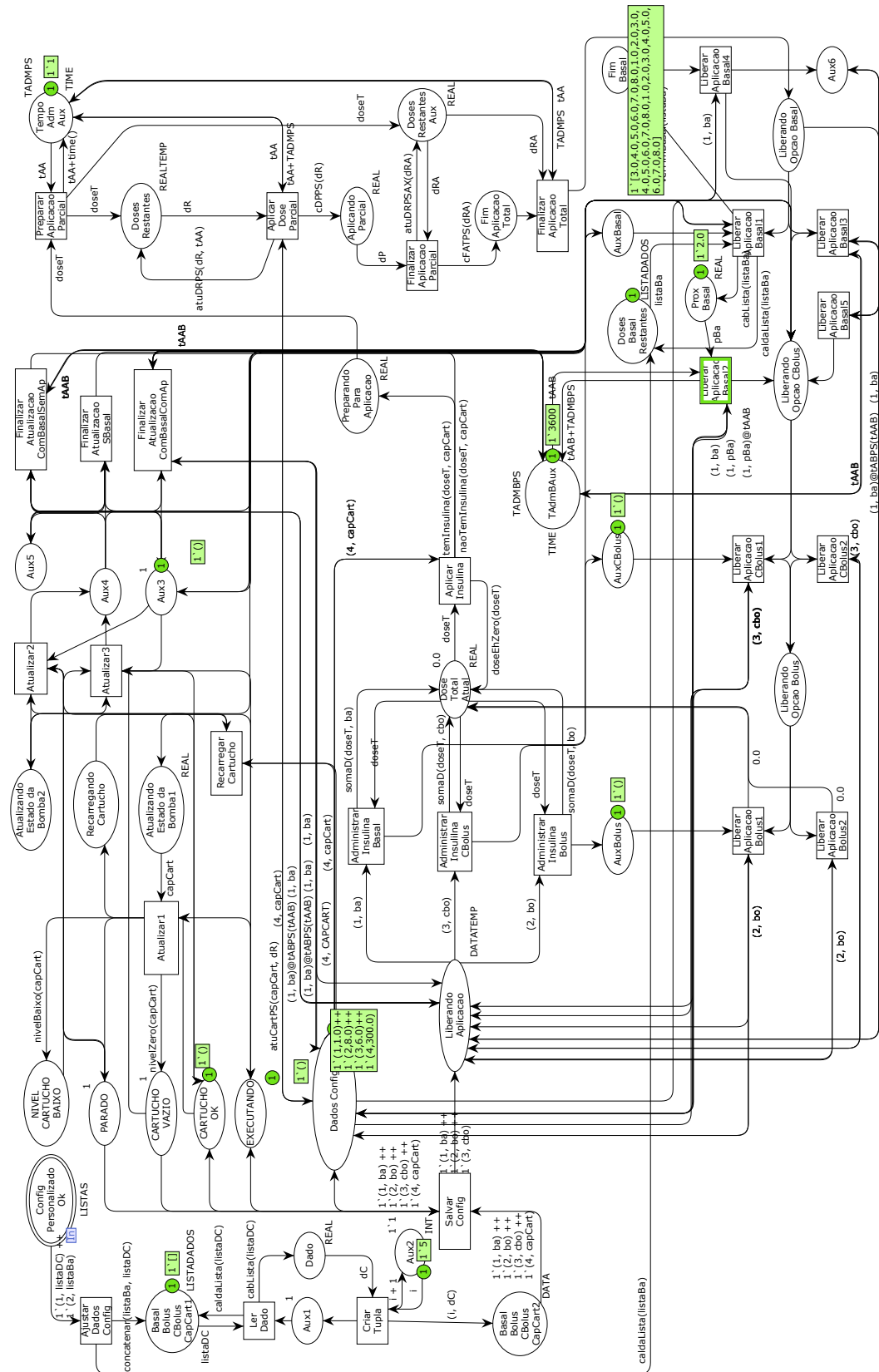
que consta na cabeça da lista). Essa dose é obtida e enviada para o lugar Prox Basal1 com o disparo da transição Liberar Aplicacao Basal1. Em seguida, com o disparo da transição

Liberar Aplicacao Basal2, a nova dupla de configuração de dose basal é gerada e enviada para os lugares Liberando Aplicacao e Dados Config. Essa nova dupla é marcada com o próximo tempo no qual a dose basal pode ser aplicada. O controle desse tempo de administração de insulina basal ocorre de maneira similar ao controle do tempo de administração de aplicação de insulina, diferenciando-se apenas com relação a nomenclatura usada para o lugar e a função.

Como os valores das dosagens de insulina bolus e bolus corretivo são constantes e elas não possui uma taxa de administração, o processo de liberação de cada uma delas consiste simplesmente em disparar, respectivamente, as transições Liberar Aplicacao CBolus1 e Liberar Aplicacao Bolus1 (elas foram habilitadas porque as insulinas bolus e bolus corretivo foram administradas, do contrário, as transições Liberar Aplicacao CBolus2 e Liberar Aplicacao Bolus2 seriam as habilitadas). Com isso, as duplas de configuração desses dois tipos de insulina são enviadas para o lugar Liberando Aplicacao, as quais são obtidas do lugar Dados Config. Como o lugar Liberando Aplicacao possui o tipo real temporizado, essas duas duplas são armazenadas com o tempo atual do relógio.

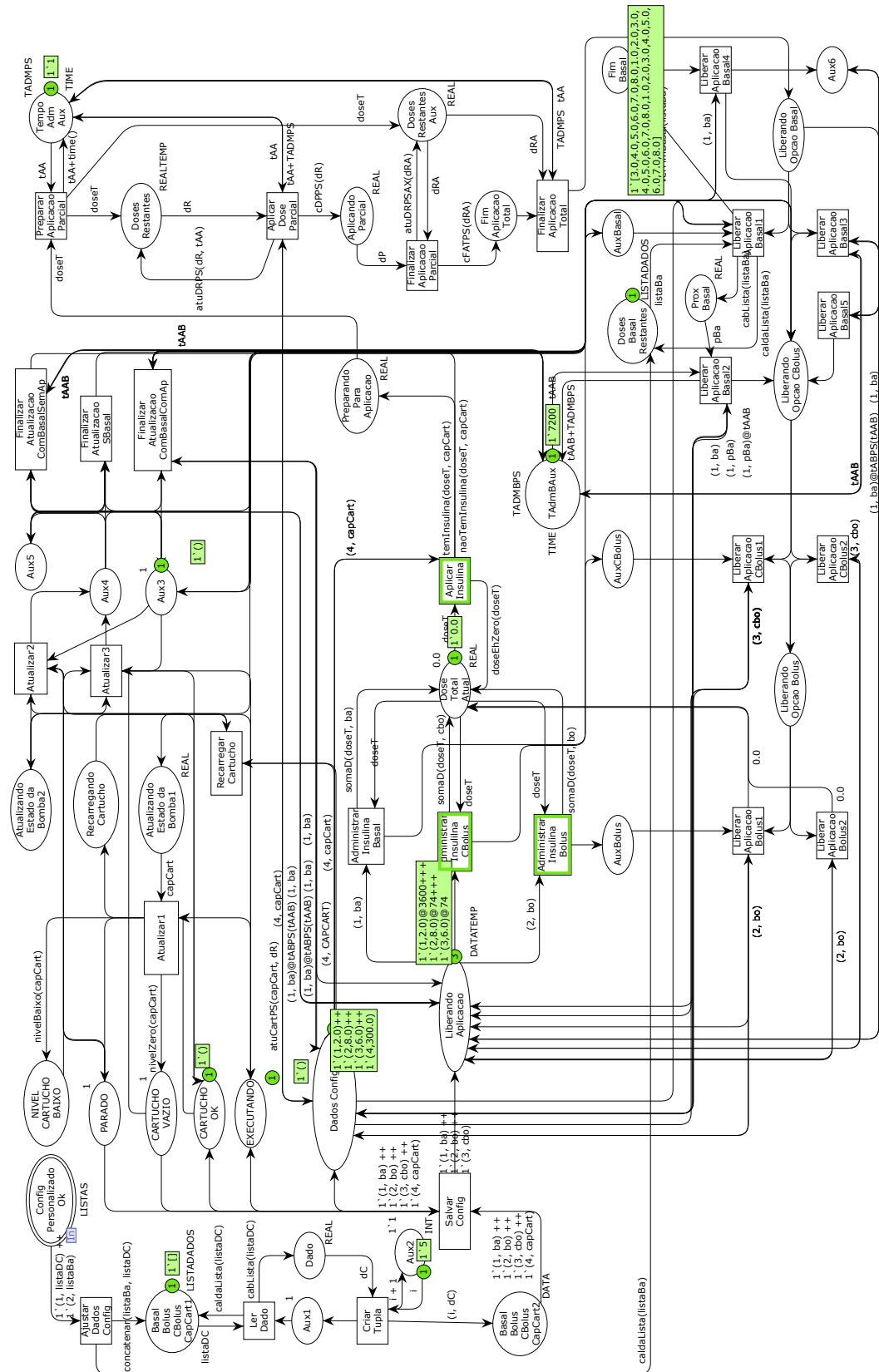
Na Figura 25 é apresentado o resultado da simulação do modelo após a liberação dos três tipos de administração de insulina. Perceba que apenas as transições de administração de bolus e de bolus corretivo foram habilitadas, pois o tempo para a administração da próxima dose de insulina basal ainda não foi alcançado. Apenas quando o relógio marcar no mínimo 3600 unidades de tempo a transição Administrar Insulina Basal será novamente habilitada.

Figura 24 – Captura de tela 4 do módulo de aplicação de insulina do modo personalizado do *ACCU-CHECK Spirit* versão 2.XX



Fonte: Próprio autor, 2019.

Figura 25 – Captura de tela 5 do módulo de aplicação de insulina do modo personalizado do *ACCU-CHECK Spirit* versão 2.XX



Fonte: Próprio autor, 2019.

## 6 CONCLUSÃO E TRABALHOS FUTUROS

Neste trabalho foi realizada a especificação formal de um modelo genérico de bomba de infusão de insulina. Além disso, o modelo foi instanciado para o equipamento *ACCU-CHECK Spirit* versão 2.XX usado como objeto do estudo de caso apresentado neste documento. Conforme foi apresentado, uma bomba de infusão de insulina é um sistema médico crítico de segurança para monitoramento e/ou tratamento de pacientes acometidos por diabetes, composta por componentes de *hardware* e *software*, os quais em conjunto simulam o funcionamento do pâncreas humano.

No modelo genérico foram implementadas as funcionalidades essenciais para o adequado funcionamento da bomba. Ele foi especificado usando redes de Petri coloridas (*Coloured Petri Nets - CPN*) com o auxílio da ferramenta *CPN/Tools*. O uso de *CPN* permitiu a construção de um modelo parametrizável, modularizado, executável, temporizado e que pode ser verificado por meio da técnica de verificação automática de modelos.

O estudo de caso foi realizado para estender o modelo genérico por meio da verificação e validação, respectivamente, da primeira e da segunda versão. Isso foi útil para demonstrar como o modelo pode ser utilizado durante o processo de certificação de equipamentos, além de aumentar a confiança na sua especificação. Na primeira etapa foram verificadas duas propriedades de segurança relevantes para o funcionamento da bomba. Na segunda etapa foram validadas as funcionalidades de controle de aplicação de insulina e de controle de administração de insulina basal.

Portanto, a principal contribuição obtida com a realização desta pesquisa foi a disponibilização de um modelo formal que pode ser reutilizado durante o processo de certificação de equipamentos e de um estudo de caso de um sistema de bomba de infusão de insulina desenvolvida por uma empresa reconhecida na indústria de equipamentos médicos. No estudo de caso, o modelo especificado foi verificado e validado por meio da técnica de verificação automática de modelos e simulações, respectivamente.

Para trabalhos futuros, recomenda-se a análise dos requisitos impostos pela *Food And Drog Administration (FDA)*, os quais podem ser incorporados ao modelo para torná-lo ainda mais robusto. Entre esses requisitos está, por exemplo, o controle de acesso. Um requisito que também pode ser incorporado ao modelo é a verificação do cartucho da bomba no momento de liberação da administração de novas doses de insulina a fim de identificar se existe o mínimo de insulina que permita a aplicação de ao menos um dos tipos de insulina, caso contrário, nesse



instante o processo de recarregamento poderia ser iniciado a fim de reduzir o atraso com a recarga. Outro trabalho que também pode ser realizado é a modelagem de uma bomba de infusão de insulina inteligente, que além do tratamento também realiza o monitoramento. Por último, pode ser feito também a ampliação do modelo a fim de que ele se torne genérico para a infusão de qualquer tipo de fluido.

## REFERÊNCIAS

- AHMAD, U.; SONG, H.; BILAL, A.; SALEEM, S.; ULLAH, A. Securing insulin pump system using deep learning and gesture recognition. In: IEEE. **2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)**. [S.l.], 2018. p. 1716–1719.
- ALI, N. A.; MIRGHANI, A. A.; IBRAHIM, A. Y. Alneelain: A formal specification language. In: IEEE. **2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE)**. [S.l.], 2017. p. 1–9.
- ATA, S.; KHAN, Z. H. Model based control of artificial pancreas under meal disturbances. In: **2017 International Symposium on Recent Advances in Electrical Engineering (RAEE)**. IEEE, 2017. Disponível em: <<https://doi.org/10.1109/raee.2017.8246033>>.
- CARDOSO, J.; VALETTE, R. **Redes de petri**. [S.l.]: Universidade Federal de Santa Catarina, 1997.
- CHENG, A.; CHRISTENSEN, S.; MORTENSEN, K. H. Model checking coloured petri nets-exploiting strongly connected components. **DAIMI report series**, n. 519, 1997.
- CHRISTENSEN, S.; MORTENSEN, K. H. **Design/CPN ASK-CTL Manual. 0.9**. [S.l.]: University Of Aarhus, 1996.
- CLARKE, E. M. The birth of model checking. In: **25 Years of Model Checking**. [S.l.]: Springer, 2008. p. 1–26.
- CLARKE, E. M.; EMERSON, E. A.; SIFAKIS, J. Model checking: algorithmic verification and debugging. **Communications of the ACM**, ACM, v. 52, n. 11, p. 74–84, 2009.
- CLARKE, E. M.; GRUMBERG, O.; KROENING, D.; PELED, D.; VEITH, H. **Model Checking**. Massachusetts: The Mit Press, 2018.
- DING, J.; CHEN, X.; WANG, R. A compositional analysis method for petri-net models. **IEEE Access**, Institute of Electrical and Electronics Engineers (IEEE), v. 5, p. 27599–27610, 2017. Disponível em: <<https://doi.org/10.1109/access.2017.2772829>>.
- DOTOLI, M.; EPICOCO, N.; FALAGARIO, M.; CAVONE, G. A timed petri nets model for performance evaluation of intermodal freight transport terminals. **IEEE Transactions on Automation Science and Engineering**, Institute of Electrical and Electronics Engineers (IEEE), v. 13, n. 2, p. 842–857, apr 2016. Disponível em: <<https://doi.org/10.1109/tase.2015.2404438>>.
- FAKHIR, M. I.; KAZMI, S. A. R. Formal specification and verification of self-adaptive concurrent systems. **IEEE Access**, Institute of Electrical and Electronics Engineers (IEEE), v. 6, p. 34790–34803, 2018. Disponível em: <<https://doi.org/10.1109/access.2018.2849821>>.
- GIL, A. C. **Como Elaborar Projetos de Pesquisa**. São Paulo: Atlas, 2002.
- HAWKINS, R.; HABLI, I.; KELLY, T.; MCDERMID, J. Assurance cases and prescriptive software safety certification: A comparative study. **Safety Science**, Elsevier BV, v. 59, p. 55–71, nov 2013. Disponível em: <<https://doi.org/10.1016/j.ssci.2013.04.007>>.

HEI, X.; DU, X.; LIN, S.; LEE, I.; SOKOLSKY, O. Patient infusion pattern based access control schemes for wireless insulin pump system. **IEEE Transactions on Parallel and Distributed Systems**, Institute of Electrical and Electronics Engineers (IEEE), v. 26, n. 11, p. 3108–3121, nov 2015. Disponível em: <<https://doi.org/10.1109/tpds.2014.2370045>>.

HU, R.; LI, C. The design of an intelligent insulin pump. In: **2015 4th International Conference on Computer Science and Network Technology (ICCSNT)**. IEEE, 2015. Disponível em: <<https://doi.org/10.1109/iccsnt.2015.7490848>>.

ISLAM, F.; KHAN, A. H. A.; SINGH, N.; ALI, A. A. Advanced simulation of an insulin & glucagon pump. In: **2016 International Conference on Information Technology (InCITe) - The Next Generation IT Summit on the Theme - Internet of Things: Connect your Worlds**. IEEE, 2016. Disponível em: <<https://doi.org/10.1109/incite.2016.7857616>>.

JENSEN, K.; KRISTENSEN, L. M. **Coloured Petri nets: modelling and validation of concurrent systems**. [S.l.]: Springer Science & Business Media, 2009.

JENSEN, K.; KRISTENSEN, L. M. Colored petri nets. **Communications of the ACM**, Association for Computing Machinery (ACM), v. 58, n. 6, p. 61–70, may 2015. Disponível em: <<https://doi.org/10.1145/2663340>>.

KITCHENHAM, B. **Guidelines for Conducting Reviews of Systematic Literature in Software Engineering**. [S.l.]: EBSE Technical Report, 2007.

KUMAR, V.; SINGH, L. K.; SINGH, P.; SINGH, K. V.; MAURYA, A. K.; TRIPATHI, A. K. Parameter estimation for quantitative dependability analysis of safety-critical and control systems of NPP. **IEEE Transactions on Nuclear Science**, Institute of Electrical and Electronics Engineers (IEEE), v. 65, n. 5, p. 1080–1090, may 2018. Disponível em: <<https://doi.org/10.1109/tns.2018.2827106>>.

LEE, H.; CHOI, J.-Y. Constraint solving approach to schedulability analysis in real-time systems. **IEEE Access**, Institute of Electrical and Electronics Engineers (IEEE), v. 6, p. 58418–58426, 2018. Disponível em: <<https://doi.org/10.1109/access.2018.2874637>>.

LI, T.; TAN, F.; WANG, Q.; BU, L.; CAO, J.-n.; LIU, X. From offline toward real time: A hybrid systems model checking and cps codesign approach for medical device plug-and-play collaborations. **IEEE Transactions on Parallel and Distributed Systems**, IEEE, v. 25, n. 3, p. 642–652, 2014.

MARTINS, L. E. G.; FARIA, H. de; VECCHETE, L.; CUNHA, T.; OLIVEIRA, T. de; CASARINI, D. E.; COLUCCI, J. A. Development of a low-cost insulin infusion pump: Lessons learned from an industry case. In: **2015 IEEE 28th International Symposium on Computer-Based Medical Systems**. IEEE, 2015. Disponível em: <<https://doi.org/10.1109/cbms.2015.14>>.

MASCI, P.; AYOUB, A.; CURZON, P.; LEE, I.; SOKOLSKY, O.; THIMBLEBY, H. Model-based development of the generic PCA infusion pump user interface prototype in PVS. In: **Lecture Notes in Computer Science**. Springer Berlin Heidelberg, 2013. p. 228–240. Disponível em: <[https://doi.org/10.1007/978-3-642-40793-2\\_21](https://doi.org/10.1007/978-3-642-40793-2_21)>.

MERTZ, L. Automated insulin delivery: Taking the guesswork out of diabetes management. **IEEE Pulse**, Institute of Electrical and Electronics Engineers (IEEE), v. 9, n. 1, p. 8–9, jan 2018. Disponível em: <<https://doi.org/10.1109/mpul.2017.2772685>>.

MIRANDA, M. A.; RIBEIRO, M. G.; MARQUES-NETO, H. T.; SONG, M. A. J. Domain-specific language for automatic generation of UML models. **IET Software**, Institution of Engineering and Technology (IET), v. 12, n. 2, p. 129–135, apr 2018. Disponível em: <<https://doi.org/10.1049/iet-sen.2016.0279>>.

MUANGPRATHUB, J.; BOONJING, V. Online thai medical diagnostic system using case-based reasoning. In: **2014 International Computer Science and Engineering Conference (ICSEC)**. IEEE, 2014. Disponível em: <<https://doi.org/10.1109/icsec.2014.6978179>>.

MURATA, T. Petri nets: Properties, analysis and applications. **Proceedings of the IEEE**, Institute of Electrical and Electronics Engineers (IEEE), v. 77, n. 4, p. 541–580, apr 1989. Disponível em: <<https://doi.org/10.1109/5.24143>>.

NIEZEN, G.; ESLAMBOLCHILAR, P. A human operator model for medical device interaction using behavior-based hybrid automata. **IEEE Transactions on Human-Machine Systems**, Institute of Electrical and Electronics Engineers (IEEE), v. 46, n. 2, p. 291–302, apr 2016. Disponível em: <<https://doi.org/10.1109/thms.2015.2487509>>.

PETERSON, J. L. **Petri nets theory and the modeling of systems**. Englewood Cliffs: Prentice Hall, 1981.

RATHORE, H.; WENZEL, L.; AL-ALI, A. K.; MOHAMED, A.; DU, X.; GUIZANI, M. Multi-layer perceptron model on chip for secure diabetic treatment. **IEEE Access**, Institute of Electrical and Electronics Engineers (IEEE), v. 6, p. 44718–44730, 2018. Disponível em: <<https://doi.org/10.1109/access.2018.2854822>>.

SAIVES, J.; FARAUT, G.; LESAGE, J.-J. Automated partitioning of concurrent discrete-event systems for distributed behavioral identification. **IEEE Transactions on Automation Science and Engineering**, Institute of Electrical and Electronics Engineers (IEEE), v. 15, n. 2, p. 832–841, apr 2018. Disponível em: <<https://doi.org/10.1109/tase.2017.2718244>>.

SHARANYA, G.; ABHISHEK, C. S.; REDDY, K. M. Health monitoring device. In: **2017 2nd International Conference on Communication and Electronics Systems (ICCES)**. IEEE, 2017. Disponível em: <<https://doi.org/10.1109/cesys.2017.8321163>>.

SILVA, L. C. e. **Uma Abordagem Baseada em Modelos para Suporte à Validação de Sistemas Médicos Físico Cibernéticos**. Tese (Doutorado) — Centro de Engenharia Elétrica e Informática, Universidade Federal de Campina Grande, Campina Grande, 2015.

SJAAHEIM, H.; ALBERT, B.; SETCHI, R.; STRISLAND, F. A portable medical system for the early diagnosis and treatment of traumatic brain injury. In: **2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)**. IEEE, 2014. Disponível em: <<https://doi.org/10.1109/smc.2014.6974307>>.

SOBRINHO, A.; PERKUSICH, A.; SILVA, L. D. da; CUNHA, P. Using colored petri nets for the requirements engineering of a surface electrogastrography system. In: **2014 12th IEEE International Conference on Industrial Informatics (INDIN)**. IEEE, 2014. Disponível em: <<https://doi.org/10.1109/indin.2014.6945511>>.

SOBRINHO, A.; SILVA, L. D. da; PERKUSICH, A.; CUNHA, P.; CORDEIRO, T.; LIMA, A. M. N. Formal modeling of biomedical signal acquisition systems: source of evidence for certification. **Software & Systems Modeling**, Springer Nature, aug 2017. Disponível em: <<https://doi.org/10.1007/s10270-017-0616-7>>.

SOMMERVILLE, I. Engenharia de software, 9a. **São Palo, SP, Brasil**, 2011.

TANG, J.; ZHU, F. Graphical modeling and analysis software for state space-based optimization of discrete event systems. **IEEE Access**, Institute of Electrical and Electronics Engineers (IEEE), v. 6, p. 38385–38398, 2018. Disponível em: <<https://doi.org/10.1109/access.2018.2852324>>.

TANG, J.; ZHU, F.; FAN, L. Simulation modelling of traffic collision avoidance system with wind disturbance. **IEEE Aerospace and Electronic Systems Magazine**, Institute of Electrical and Electronics Engineers (IEEE), v. 33, n. 4, p. 36–45, apr 2018. Disponível em: <<https://doi.org/10.1109/maes.2018.170044>>.

TARI, Z.; SAKIB, K.; BERTOK, P.; MUKHERJEE, A. **Verification of Communication Protocols in Web Services: Model-Checking Service Compositions**. [S.l.]: John Wiley & Sons, Inc, 2013.

VIZEL, Y.; WEISSENBACHER, G.; MALIK, S. Boolean satisfiability solvers and their applications in model checking. **Proceedings of the IEEE**, Institute of Electrical and Electronics Engineers (IEEE), v. 103, n. 11, p. 2021–2035, nov 2015. Disponível em: <<https://doi.org/10.1109/jproc.2015.2455034>>.

WAINER, G. A. **Discrete-Event Modeling and Simulation: A Practitioner's Approach**. Boca Raton: Crc Press, 2009.

YANG, J.; PU, H.; LIAN, J.; GU, J.; FAN, M. Modeling and analysis of protein synthesis and DNA mutation using colored petri nets. **IEEE Access**, Institute of Electrical and Electronics Engineers (IEEE), v. 6, p. 22386–22400, 2018. Disponível em: <<https://doi.org/10.1109/access.2018.2825782>>.

ZHANG, Y.; JONES, P. L.; JETLEY, R. A hazard analysis for a generic insulin infusion pump. **Journal of Diabetes Science and Technology**, SAGE Publications, v. 4, n. 2, p. 263–283, mar 2010. Disponível em: <<https://doi.org/10.1177/193229681000400207>>.