



**GOVERNO DO
ESTADO DO
TOCANTINS**

CURSO DE SISTEMAS DE INFORMAÇÃO

APLICAÇÃO DA TEORIA DE CONTROLE SUPERVISÓRIO NA AUTOMAÇÃO DE UM SISTEMA DE ARMAZENAGEM

MARCUS VINÍCIUS FERREIRA LIMA

Palmas

2018



**GOVERNO DO
ESTADO DO
TOCANTINS**

CURSO DE SISTEMAS DE INFORMAÇÃO

APLICAÇÃO DA TEORIA DE CONTROLE SUPERVISÓRIO NA AUTOMAÇÃO DE UM SISTEMA DE ARMAZENAGEM

MARCUS VINÍCIUS FERREIRA LIMA

Projeto apresentado ao Curso de Sistemas de Informação da Fundação Universidade do Tocantins - UNITINS como parte dos requisitos para a obtenção do grau de Bacharel em Sistemas de Informação, sob a orientação do professor Me. Douglas Chagas.

Palmas

2018



**GOVERNO DO
ESTADO DO
TOCANTINS**

CURSO DE SISTEMAS DE INFORMAÇÃO

APLICAÇÃO DA TEORIA DE CONTROLE SUPERVISÓRIO NA AUTOMAÇÃO DE UM SISTEMA DE ARMAZENAGEM

MARCUS VINÍCIUS FERREIRA LIMA

Projeto apresentado ao Curso de Sistemas de Informação da Fundação Universidade do Tocantins - UNITINS como parte dos requisitos para a obtenção do grau de Bacharel em Sistemas de Informação, sob a orientação do professor Me. Douglas Chagas.

Prof. Me. Douglas Chagas da Silva
Orientador

**Prof. Me. Marco Antonio Firmino
Sousa**
Convidado 1

**Prof. Esp. Marcelo Ribeiro de Oliveira
Mello**
Convidado 2

Palmas
2018

Aos meus pais, minha filha e minha esposa, pelo apoio e incentivo.

Agradecimentos

A Deus por ter me dado saúde e força para superar as dificuldades.

A minha mãe, meu pai, minha filha e minha esposa, pelo amor, incentivo e apoio.

A esta Universidade, seu corpo docente, direção e administração.

Ao Prof. Me. Douglas pela oportunidade e apoio na elaboração deste trabalho.

A todos que direta ou indiretamente fizeram parte da minha formação.

“Tente uma, duas, três vezes e se possível tenta a quarta, a quinta e quantas vezes forem necessárias. A persistência é amiga da conquista. Se você quer chegar aonde a maioria não chega, faça o que a maioria não faz”
(Bill Gates).

Resumo

Este projeto apresenta um estudo da aplicação da teoria de controle supervisório na automação de um sistema de armazenagem. Uma análise do sistema é apresentada, descrevendo o contexto sobre a relevância e utilização dos sistemas de armazenagem. São apresentados os conceitos envolvendo sistemas de eventos discretos e aplicação desta teoria na resolução de problemas de automação. Além disso, obtém-se a modelagem do sistema via controle supervisório modular local demonstrando de forma detalhada e validada a partir do uso de técnicas de simulação.

Palavras-chaves: Autômatos, controle supervisório modular local, sistemas de armazenagem, sistemas a eventos discretos.

Abstract

This project presents a study of the application of the theory of supervisory control in the automation of a storage system. A system analysis is presented, describing the context about the relevance and use of storage systems. The concepts involving systems of discrete events and application of this theory is presented in the resolution of automation problems. In addition, the modeling of the system is obtained through local modular supervisory control demonstrating in detail and validated form the use of simulation techniques.

Key-words: Automata, local modular supervisory control, storage systems, Discrete Event Systems.

Lista de ilustrações

Figura 1 – Variável de estado do sistema discreto. (BANKS, 2010)	5
Figura 2 – Crescimento populacional de acordo com o tempo Reiser. (REISER, 2005)	5
Figura 3 – Diagramas de transições de estados para as linguagens L1, L2 e L3. (MENEZES, 2011)	7
Figura 4 – Autômato Finito. (HARA, 2017)	8
Figura 5 – Autômato finito não determinístico (AFN). (SIPSER, 2007)	9
Figura 6 – Esquema de controle monolítico. (SIPSER, 2007)	10
Figura 7 – Célula de Manufatura (CUNHA, 2003)	11
Figura 8 – Esquema de funcionamento do controle modular Adaptado (CURY, 2001)	12
Figura 9 – Esquema de Funcionamento do Controle Modular Local Adaptado (KAWAMURA, 2012)	13
Figura 10 – RSP: (a) modelo dos subsistemas (b) RSP mais refinada (MELLO, 2006)	13
Figura 11 – Interface Software TCT	14
Figura 12 – Menu do TCT	15
Figura 13 – Gestão de estoques (MAIA, 2013)	16
Figura 14 – Sistema de Armazenagem (MAIA, 2013).	17
Figura 15 – Interface Gráfica do Software Simul8 (BOTTER, 2016)	19
Figura 16 – Interface do Software Arena (TREBESQUIM, 2013)	20
Figura 17 – Interface do Software FlexSim	21
Figura 18 – Interface do Software FlexSim	22
Figura 19 – Estrutura da Estação de Armazenamento	24
Figura 20 – Diagrama de Fluxos	25
Figura 21 – Autômatos das Plantas dos Equipamentos	27
Figura 22 – Conjunto de Especificações	27
Figura 23 – Estado Máquinas Plantas no <i>FlexSim</i>	30
Figura 24 – Supervisores Ótimos. (a) S_{A1} . (b) S_{A2} e S_B . (c) S_{C1} , S_{C2} e S_D	32
Figura 25 – Simulação Sistema de Armazenagem com os supervisores.	34
Figura 26 – Fluxograma de Configuração do tipo da Caixa	35
Figura 27 – Fluxograma Configuração do Ponto de Decisão	35
Figura 28 – Configuração dos <i>Buffers</i> para transporte das Caixas pelo "Coletor"	36
Figura 29 – Nível de Processamento das máquinas vs Tempo Real	37
Figura 30 – Teste 1	38
Figura 31 – Teste 2	39
Figura 32 – Teste 3	39
Figura 33 – Teste 4	40
Figura 34 – Teste 5	41

Figura 35 – Teste 6	41
Figura 36 – Teste 7	42
Figura 37 – Teste 8	42
Figura 38 – Teste 9	43
Figura 39 – GA1	51
Figura 40 – GA2	51
Figura 41 – GAB	52
Figura 42 – GC1	52
Figura 43 – GC2	52
Figura 44 – GD	52
Figura 45 – KA1	53
Figura 46 – KA2	53
Figura 47 – KB	54
Figura 48 – KC1	54
Figura 49 – KC2	54
Figura 50 – KD	55

Lista de tabelas

Tabela 1 – Diferença entre SED e SEC. (VILLANUEVA, 2016)	6
Tabela 2 – Autômato Finito. Função de Transição. (SIPSER, 2007)	8
Tabela 3 – Materiais Utilizados	23
Tabela 4 – Transições da Planta	26
Tabela 5 – Estrutura e Transições	28
Tabela 6 – Composição das Plantas	28
Tabela 7 – Composição das Especificações + Plantas	29
Tabela 8 – Supervisores ótimos	31
Tabela 9 – Testes de entradas	38
Tabela 10 – (a)Caixas por simulação	44
Tabela 11 – (b)Caixas por simulação	44

Lista de abreviaturas e siglas

AFD - Autômato Finito Determinístico.

AFN - Autômato Finito Não Determinístico

SED - Sistema a Eventos Discretos

TCS - Teoria de Controle Supervisório

SEC - Sistema a Eventos Contínuos

RSP - Representação por Sistema Produto

Sumário

1	INTRODUÇÃO	1
1.1	Justificativa	3
1.2	Objetivos	4
1.2.1	Objetivo Geral	4
1.2.2	Objetivos Específicos	4
2	REFERENCIAL TEÓRICO	5
2.1	Sistemas a Eventos Discretos	5
2.2	Autômatos	6
2.2.1	Autômato Finito Determinístico	7
2.2.2	Autômato Finito Não Determinístico	9
2.3	Teoria de Controle Supervisório	10
2.3.1	Controle Supervisório Modular Clássico	12
2.3.2	Controle Supervisório Modular Local	12
2.4	TCT	14
2.5	Administração de Materiais	15
2.5.1	Estoque	16
2.5.2	Sistemas de Armazenagem	17
2.6	Simuladores	18
2.6.1	Software Simul8	19
2.6.2	Software Arena	20
2.6.3	Software FlexSim	21
3	METODOLOGIA	23
3.1	Materiais	23
3.2	Modelagem do Sistema	23
3.2.1	Planta do Sistema de Armazenagem	23
3.2.2	Modelagem Modular Local	25
3.2.3	Definição do Software de Simulação	29
4	RESULTADOS	31
4.1	Obtenção dos Supervisores	31
4.2	Validação da Modelagem	33
4.3	Resultados da Simulação	36
5	CONCLUSÃO	46

	REFERÊNCIAS	48
	APÊNDICES	50
	APÊNDICE A – IMAGENS DOS AUTÔMATOS DAS PLANTAS .	51
A.1	Composição das Plantas	51

1 Introdução

A automação industrial é um campo de exploração que vem expandindo sucessivamente. A utilização de instrumentos tecnológicos e a sua aplicação para solucionar problemas envolvendo automação industrial possui grande aceitação do ponto de vista de mercado. As aplicações não se limitam apenas em sobrepor o trabalho do homem em ocupações cansativas, rotineiras e que coloca em risco a vida; elas apresentam benefícios na qualidade da técnica, otimização dos espaços, diminuição do período da manufatura e custos ([PAREDE J; GOMES, 2011](#)).

A tecnologia vem gerando, em grande proporção, sistemas com a intenção de realizar funções que justificam o empenho gasto na sua melhoria e automação. Esses sistemas encontram-se em uma porção de aplicações, por exemplo, a automação da indústria, a robótica, o controle de tráfego, a gestão, softwares, redes de computadores, administração de banco de dados e otimização de processos distribuídos ([BRYNJOLFSSON2015, 2015](#)).

As operações que ocorrem, são geradas diretamente pela entrada de impulsos, e essas ações vêm a ser chamadas de eventos. Tendo como exemplo de eventos, o começo e o fim de um determinado processo e a captura de uma troca de estado em um sensor. Estes exemplos de eventos instantâneos, que é quase impercebível quando se fala de tempo são chamados de Sistemas a Eventos Discretos (SED) ([BANKS, 2010](#)).

Esses SED's podem ser reconhecidos como uma ação intencional, como por exemplo, ligar um interruptor, ou um evento involuntário, uma perda de conexão com a internet, e até mesmo uma averiguação de uma circunstância, por exemplo, quando a temperatura de um reator excede o limite de segurança. Os eventos geram transições de estado, e a maior parte acontece em momentos de tempo irregulares ([BROOKSHEAR, 2013](#)).

Para gerenciar esses eventos é aplicada a Teoria de Controle Supervisório (TCS) que nada mais é que, um supervisor que garante uma execução simultânea e a garantia que não irá ocorrer o bloqueio das atividades em uma célula de manufatura. Porém, não é possível determinar que somente com um controle supervisório implique em um melhor desempenho, pois pode gerar um grande número de estados para um mesmo sistema ou subsistema ([KAWAMURA, 2012](#)).

O objetivo do controle supervisório de acordo com [Mott \(2017\)](#) é resolver esse problema projetando somente um supervisor (agente controlador que observa a sequência de eventos gerada pela planta e aplica entradas de controle) que possui o posto de ativar ou desativar eventos controláveis, de acordo com a sequência de eventos que são vistos na planta, de uma maneira que o sistema detecte atividade dos sensores e obedeçam as regras que vão ser especificadas ao decorrer do processo.

A fim de diminuir o número de estados gerados pelo controle supervísório foi estudado algumas extensões da TCS, assim como, Controle Modular e Controle Modular Local. As duas extensões proporcionam uma concentração de diversos supervisores, consistindo um para cada tipo específico, onde operam juntos, como se fossem um único supervisor que respondesse a todas as especificações concomitantemente. A diferença é que o Controle Modular calcula os supervisores considerando o sistema como um todo, e o Controle Modular Local considera somente os subsistemas que são afetados de acordo com a especificação ([MELLO, 2006](#)).

Afim, este trabalho visa apresentar uma modelagem e a implementação da automação de um sistema de armazenagem aplicando a metodologia da Teoria de Controle Supervísório.

A organização deste trabalho segue a seguinte forma: no capítulo 2 é apresentada uma breve explicação sobre o funcionamento da teoria dos autômatos e controle supervísório. E, no capítulo 3 têm-se a modelagem e apresentação dos sistemas de armazenamento propostos. Já no capítulo 4, são discutidos os resultados das simulações para o modelo proposto, e por fim têm-se as considerações finais no capítulo 5.

1.1 Justificativa

Neste trabalho optou-se por utilizar SED's para simular um sistema de armazenagem, pois ela molda a operação de acordo com a sequência de eventos discretos no tempo, ou seja, cada evento acontece em certo tempo e realiza uma alteração do estado no sistema, muda somente se ocorrer algum evento. Diferente da simulação de eventos contínuos na qual, de acordo com [Cunha \(2003\)](#), é acompanhado continuamente ao longo do tempo, sem realizar saltos discretos de um evento para outro.

Ao utilizar SED, é preciso definir qual teoria será mais bem aplicada dentro do modelo de implementação: Teoria de Controle Supervisória (TCS) ou Teoria de Controle Clássica (TCC). A vantagem da TCS é que ela constitui-se na possibilidade de tratar vários eventos em um mesmo ciclo de execução, diferente da TCC, que segundo [Banks \(2010\)](#) são imprecisos, e caso ocorra atrasos de tempo, podem vir a ocorrer oscilações na saída e inconsistência, em alguns casos podem ter um maior custo e uma complexidade de ser implementado. A TCS em [Reiser \(2005\)](#) é bastante analisada e como decorrência expandida, e a partir disso efetuou-se como uma teoria propícia para que pudesse resolver problemas de Sistemas a Eventos Discretos. O Sistema de Armazenagem proposto aqui corresponde a validação do estudo realizado, e tem como objetivo colaborar para a análise prática e teórica da TCS e as suas extensões.

No sistema logístico, segundo [Guerra \(2006\)](#), a armazenagem dos produtos tem como ideia solucionar problemas de armazenamento de materiais proporcionando a associação entre suprimento, produção, distribuição e os clientes.

1.2 Objetivos

1.2.1 Objetivo Geral

Apresentar a modelagem e a simulação de um sistema de armazenagem, utilizando controle supervisão modular local.

1.2.2 Objetivos Específicos

- Compreender e fundamentar os conceitos envolvendo teoria do controle supervisão;
- Utilizar autômatos no processo de modelagem;
- Descrever o cenário e a metodologia utilizada para resolver o problema do sistema de armazenagem;
- Analisar e discutir os resultados dos testes e simulações.

2 Referencial Teórico

2.1 Sistemas a Eventos Discretos

De acordo com [Law \(2007\)](#) poucos sistemas na prática são totalmente discretos ou contínuos, mas uma vez que um tipo de mudança predomina para a maioria dos sistemas, geralmente será possível classificar um sistema como sendo discreto ou contínuo.

A prototipagem dos modelos da simulação não é trabalhada de forma analítica, e sim formas numéricas. No método analítico é utilizado somente o raciocínio de dedução da matemática, ou seja, de certa forma, solucionar os modelos. Diferente das formas numéricas, onde vão ser aplicados métodos de computação para poder definir uma resolução dos modelos matemáticos ([BOUZADA, 2013](#)).

Um exemplo ilustrativo da aplicação de SED pode ser um observado no sistema bancário, onde a variável de estado, o número de clientes no banco, muda apenas quando um cliente chega ou quando o cliente está sendo atendido. A Figura 1 mostra como os clientes mudam apenas em pontos discretos no tempo ([BANKS, 2010](#)).

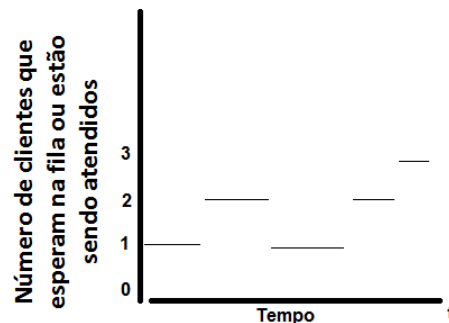


Figura 1 – Variável de estado do sistema discreto. ([BANKS, 2010](#))

No Sistema a Eventos Contínuos (SEC) existem variáveis que mudam ao longo do tempo de forma contínua e é possível ter como exemplo um controle populacional, onde a população de uma cidade cresce de forma contínua, sem intervalo de tempo ([VILLANUEVA, 2016](#)). Assim como podemos observar na Figura 2:

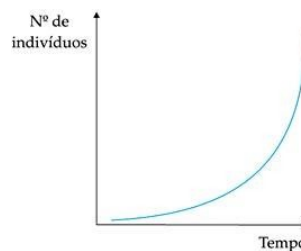


Figura 2 – Crescimento populacional de acordo com o tempo Reiser. ([REISER, 2005](#))

Em Villanueva (2016), exemplificam também algumas diferenças entre SED e SEC na Tabela 1:

Tabela 1 – Diferença entre SED e SEC. (VILLANUEVA, 2016)

SED	SEC
O espaço de estados é limitado a um conjunto discreto.	O espaço de estados é contínuo.
O sistema pode ficar em um tempo longo em um mesmo estado.	O estado muda a cada estante.
A teoria de SED trata do controle de atividades que um sistema pode executar sem se preocupar com a implementação física da atividade.	A teoria de SEC se preocupa com a realização física das atividades.

2.2 Autômatos

O autômato segundo Menezes (2011), é um tipo de robô que tem a habilidade de retratar uma linguagem conforme as regras determinadas. Por isso, as metodologias que se baseiam em autômatos vêm a ser utilizada em abundância para modelar os SED's, principalmente pela TCS.

Autômatos consistem em um raciocínio matemático que lista nitidamente todos os estados do sistema. Esses estados se conectam por meio de mudanças rotuladas que possibilitam o progresso do modelo. Por meio dos gráficos, os autômatos podem ser representados através de um diagrama de mudanças de estados (RAMOS, 2008).

A indicação de uma seta pequena é para definir o estado inicial. Há também a definição de um ou mais estados finais que podem ser indicado para um determinado comportamento, como exemplo, quando se tem o final de alguma operação de determinado equipamento (MENEZES, 2011).

Já as linguagens são, de acordo com Menezes (2011), um grupo de palavras que são aceitas por um autômato.

É possível demonstrar os autômatos que simbolizam as linguagens L1, L2 e L3 que são definidas na Figura 3:. Verifica-se ainda na Figura 3: a legenda "a,b,g" representa a alteração que acontece com um desses eventos *a, b* ou *g*.

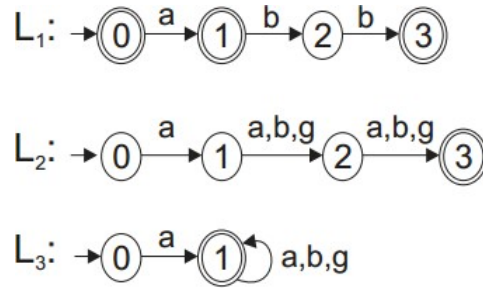


Figura 3 – Diagramas de transições de estados para as linguagens L1, L2 e L3. (MENEZES, 2011)

Os autômatos trazem muitas vantagens para os SED, assim como, uma maior flexibilidade, pois podem ser aplicadas em utilizações distintas, redução dos custos de manutenção e uma expansibilidade com o aumento das entradas e saídas de acordo com a necessidade (SIPSER, 2007).

Para um maior entendimento são estudadas as especificações de autômatos, que são os Autômatos Finitos Determinísticos (AFD) e os Autômatos Finitos Não Determinísticos (AFN). O AFD é um tipo de máquina de estados finitas, ou seja, a partir de uma transição de estado ele pode assumir um único estado, diferente da AFN que pode assumir vários estados simultâneos (HARA, 2017).

2.2.1 Autômato Finito Determinístico

Um autômato finito, de acordo com Sipser (2007), é um simples modelo computacional que também é conhecido como máquina de estados finitos que possui uma memória muito limitada.

É caracterizado em Welter R; Postal (2006), como um mecanismo para identificar os tipos de linguagens, onde recebe uma cadeia de caracteres como entrada e classifica se pode ser aceita ou não.

Esses autômatos possuem um grupo de estados e especificações a fim de ir de um estado para o seguinte, conforme o símbolo que é recebido na entrada. A partir de um alfabeto de entrada é possível saber quais são os símbolos de entrada que são autorizados. Em todo autômato é possível notar um estado de início e pode haver mais de um estado de aceitação (SIPSER, 2007).

Segundo Brookshear (2013) um autômato finito é definido com uma enumeração de cinco tipos de objetos, sendo eles: grupo de condições ou estados; quais serão os alfabetos permitidos, ou que vão ser utilizados; normas para movimentação; somente um estado inicial; e podem ocorrer vários estados finais ou de aceitação.

Na linguagem matemática Brookshear (2013) afirma que seria chamado de 5-upla e com isso é definido um autômato finito como uma 5-upla consistindo dessas cinco partes.

É definido ainda o conceito de função de transição, representado por (δ), que define as regras de movimentação. Se o autômato possui uma seta de um estado 'x' para um estado 'y' rotulada com um símbolo de entrada 1, significa que, se o autômato está no estado 'x' quando lê 1, então passa para o estado 'y'. Na representação matemática pode-se indicar da seguinte forma, $\delta(x,1) = y$.

De acordo [Hara \(2017\)](#) como tudo na forma matemática, um autômato pode ser definido como uma 5-upla $(Q, \Sigma, \delta, q_0, F)$, onde:

- Q é um conjunto finito conhecido como os estados;
- Σ é um conjunto finito chamado o alfabeto;
- $\Sigma: Q \times \Sigma \rightarrow Q$ é a função de transição;
- $q_0 \in Q$ é o estado inicial;
- $F \subseteq Q$ é o conjunto de estados de aceitação;

Um exemplo de um autômato finito é ilustrado na Figura 4:

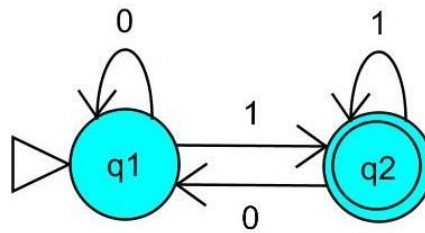


Figura 4 – Autômato Finito. ([HARA, 2017](#))

Na descrição formal a função de transição que é: $M2 = (q1, q2, 0,1, \delta, q1, q2)$, ou seja, a função recebe um símbolo de entrada junto a um estado que devolve a outro, assim, é possível definir a forma de mover-se entre estados ([HARA, 2017](#)). Na Tabela 2 a função de transição δ é definida:

Tabela 2 – Autômato Finito. Função de Transição. ([SIPSER, 2007](#))

	0	1
q1	q1	q2
q2	q1	q2

Na apresentação da Tabela 2 demonstra-se uma boa maneira para entender a máquina, testando com algumas amostras de cadeias de entrada. Quando se recebe entradas de 1101, ou seja, a cadeia, a máquina M2 começa no estado inicial q1 e procede primeiro para o estado q2 após ler o primeiro 1, e então para os estados q2, q1 e q2 após ler

1, 0 e 1. A cadeia é aceita porque q2 é um estado de aceitação. Mas a cadeia 110 deixa M2 o estado q1; portanto ela é rejeitada. Depois de tentar alguns exemplos a mais, é possível definir que M2 aceita todas as cadeias que terminam com 1 (BROOKSHEAR, 2013).

2.2.2 Autômato Finito Não Determinístico

De acordo com Menezes (2011), quando a máquina ocupa-se em um determinado estado e a partir daí faz uma leitura da seguinte representação de entrada, temos a possibilidade de saber qual será o seguinte estado. Isso é chamado pelo autor de computação determinística. Em um AFN, diversas opções podem ser encontradas para o estado subsequente em qualquer local.

Efetivamente, o significado de Não Determinismo constitui-se numa globalização de determinismo, por conseguinte, todo AFD é intuitivamente um AFN. Na Figura 5, é ilustrado que pode ser possível estar em dois estados simultaneamente ao receber o alfabeto de entrada. O estado “q1” ao receber 0, continuará no estado “q1”, e ao receber 1, vai para o estado “q2” e ao mesmo tempo para o “q1”, assim até chegar ao estado “q4”, que é o estado final (HARA, 2017).

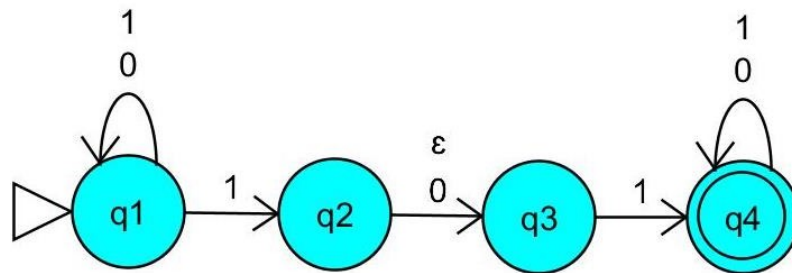


Figura 5 – Autômato finito não determinístico (AFN). (SIPSER, 2007)

Em Villanueva (2016) é dado que a diferença entre um AFD, e um AFN, é imediatamente aparente, pois na modelagem dos autômatos observamos que um tipo de entrada pode assumir diversos estados simultaneamente.

Desta forma, existe uma seta que indica uma transição em todos os estados de uma AFD para cada símbolo que foi indicado no alfabeto. O AFD também possui uma legenda por cima das setas de transição que são denominados símbolos do alfabeto, conforme ilustrado na Figura 5 (SIPSER, 2007). Em Brookshear (2013) diz-se que o não determinismo pode ser visto como uma espécie de computação paralela nas quais múltiplos e independentes processos podem estar rodando concorrentemente. Quando a AFN se divide para escolhas, isso corresponde a um processo “bifurcar” em vários filhos, cada um procedendo separadamente. Se pelo menos um desses processos for aceito, então a computação inteira aceita.

2.3 Teoria de Controle Supervisório

Algumas cadeias podem conter alguns eventos que não são desejáveis por infringirem alguma condição que há no sistema. Pode ser comparado com estados proibidos, pois provocam bloqueio ou por serem inaceitáveis, como exemplo uma colisão de um robô com outro, ou até mesmo um material em um sistema de manufatura. Por este motivo que há supervisores fazendo este controle (CURY, 2001).

O supervisor tem uma ação que habilita ou desabilita eventos, a partir disso pode se dizer que o supervisor é um controle de natureza permissiva. Dessa forma, a entrada de controle na planta é estabelecida como o conjunto de eventos habilitados pelo supervisor em um determinado momento. A entrada só será aceita se conter todos os eventos não controláveis que possam acontecer na planta. Os conjuntos totais das entradas de controle aceitas relacionado ao gerador G é chamado de estrutura de controle (MOTT, 2017).

A partir disso, pode se dizer que a principal função do supervisor é fazer a escolha das entradas de controle para cada evento observado, de modo que o sistema não saia do comportamento estabelecido e nem mesmo no meio de situações de bloqueio (MOTT, 2017).

A Figura 6 demonstra como funciona a estrutura de um controle supervisório monolítico. Apenas um supervisor para controlar a monitorar a planta e controlar os eventos.



Figura 6 – Esquema de controle monolítico. (SIPSER, 2007)

No esquema ilustrado na Figura 6, a planta gera eventos espontaneamente e é dotada de estrutura de controle, enquanto o supervisor é um agente controlador que observa a sequência de eventos gerada pela planta, e por último aplica as entradas de controles (MOTT, 2017).

Foi analisado um exemplo de célula de manufatura de Cunha (2003) para fazer uma comparação com a definição.

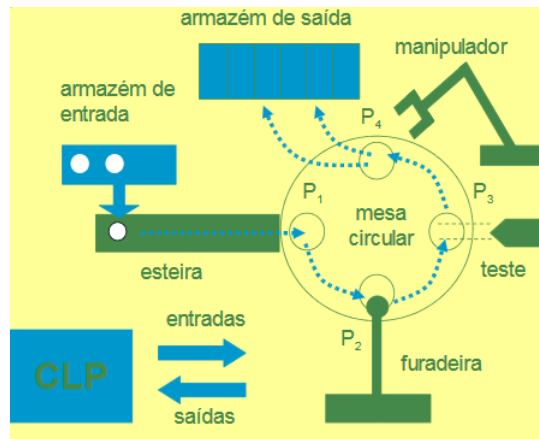


Figura 7 – Célula de Manufatura (CUNHA, 2003)

Observa-se na Figura 7 que não há informação da presença das peças nas posições P . Então se deve evitar a rotação da mesa circular sem que as peças P_2 , P_3 e P_4 não tenham sido furadas, testadas e removidas. No mesmo pensamento não devemos deixar que a esteira, furador, testador ou robô opere quando a mesa estiver girando. A partir daí os autômatos entram em vigor, onde se deve montar a quantidade de estados e transições para esse controle (CUNHA, 2003).

Em relação a uma produção industrial são observadas as seguintes vantagens quando utilizado um controle supervisor de acordo com Kawamura (2012).

1. A análise de tendências, onde são baseadas no histórico dos dados, com isso, podem ser tomadas ações proativas para potencializar a produção;
2. Os alarmes podem sinalizar em tempo real caso haja alguma falha durante o andamento do processo no qual é registrado em um banco de dados que pode ser consultado futuramente para prevenir mais falhas;
3. Procedimento remoto no processo que existe a possibilidade de interferir no andamento do mesmo, ainda que dentro de uma sala de controle;
4. Formação de informações e gráficos onde tem a possibilidade de gerá-los em conformidade com os alarmes e as tendências;
5. Ampliação da disponibilidade da planta começando do momento que as informações forem geradas em tempo real, pois é possível identificar defeitos e consequentemente aperfeiçoar as tomadas de decisão a fim de deixar que a planta continue operando, ou seja, o objetivo é que sem paradas, o processo consiga rodar por mais tempo.

2.3.1 Controle Supervisório Modular Clássico

Segundo Santos (2013) afirma que no Controle Modular Clássico, todos os detalhes de gestão serão fragmentados em diversas subtarefas, que vão ser analisadas uma por uma. Os supervisores modulares, que são identificados trabalham simultaneamente para implementar uma saída para resolver as situações difíceis de controle. Uma ação é desativada quando ao menos um dos supervisores o desativar. A Figura 8 aborda a atividade dos supervisores no sistema de acordo com os eventos que são gerados pela planta o supervisor responsável irá habilitar ou desabilitar os eventos:

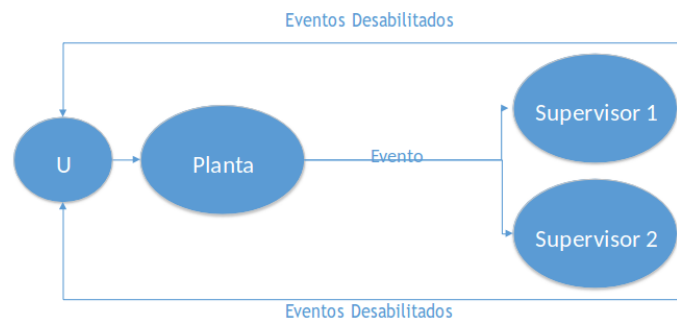


Figura 8 – Esquema de funcionamento do controle modular Adaptado (CURY, 2001)

Uma das vantagens desse modelo é que nele pode ser bem mais fácil de ser alterado, remodelado e modificado. Caso aconteça algo que seja necessário modificar só um dos detalhes, vai ser preciso alterar somente naquele supervisor correspondente, o que dá mais flexibilidade ao sistema. No caso de excluir ou incluir qualquer detalhe não interferirá nos outros supervisores. O modelo apresenta um número bem menor de estados facilitando a interpretação do mesmo pelo projetista (MOTT, 2017).

2.3.2 Controle Supervisório Modular Local

Em Mello (2006), o Controle Modular Local tem como objetivo abordar o problema de controle, explosão do espaço de estado através da síntese de um supervisor para cada tipo de especificação e comportamento. Isso reduz as dificuldades computacionais na hora de organizar a implementação do controle supervisório. A Figura 9 aborda como funciona o sistema de controle por meio de um diagrama que a partir da geração de eventos de duas subplantas, há um supervisor específico para tratar cada evento, no qual manipulará a ação e será repassada para a subplanta de origem.



Figura 9 – Esquema de Funcionamento do Controle Modular Local Adaptado (KAWAMURA, 2012)

Nesse tipo de abordagem em relação ao controle modular local, o grupo tem como função ser desenhado de acordo com a forma de uma Representação por Sistema Produto (RSP) (KAWAMURA, 2012).

O sistema produto nada mais é quando todas as subplantas podem ser desenhadas por geradores que não são síncronos. Há possibilidades de possuir um sistema produto equivalente incluindo as subplantas síncronas de qualquer sistema composto (MELLO, 2006).

Em Cury (2001), discute-se que a fim de definir a melhor RSP são analisados os subsistemas em conjunto e como resultado, um número maior possível de novos subsistemas. A Figura 10 demonstra como funciona para obter a RSP mais refinada. Primeiramente, os subsistemas vão sendo modelados, que são simbolizados pelas esferas. Essas esferas, que estão sob a outra, apresentam que os subsistemas possuem ocorrências iguais, ou seja, são conjuntas.

A Figura 10 exibe a RSP mais refinada logo depois do agrupamento dos sistemas síncronos, originando assim quatro assíncronos.

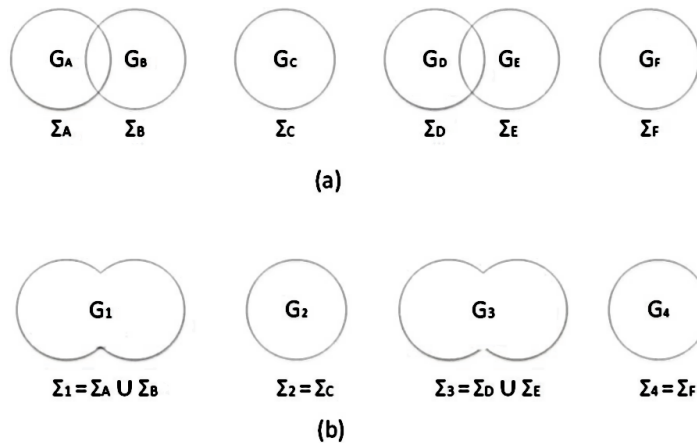


Figura 10 – RSP: (a) modelo dos subsistemas (b) RSP mais refinada (MELLO, 2006)

Quando se fala de controle modular local, podemos definir de acordo com o que já

foi escrito neste tópico que, os resultados serão melhores quanto mais refinadas for a RSP. Os resultados que serão obtidos podem ser o mesmo da representação modular clássica, quando o resultado de todas as subplantas é síncrono (KAWAMURA, 2012).

Comparando com o controle modular clássico, o modular local leva vantagem, pois é analisada a síntese de um supervisor considerando apenas os subsistemas que contém eventos iguais com a especificação. Desta forma é possível obter maior flexibilidade, além de segurança e uma maior economia computacional quando é aplicado no controle supervisorio.

2.4 TCT

O TCT¹ (*Tool for Supervisory Control Synthesis*) consiste em uma ferramenta de computação para síntese de controle de supervisão, ou seja, faz a composição dos autômatos das subplantas (WONHAM, 2017).

A Figura 11 apresenta a interface inicial do software em *Shell Script*, executado no Windows 10.



Figura 11 – Interface Software TCT

O software é simples em questões gráficas, mas a sua funcionalidade foi desenvolvida muito bem, sendo possível fazer a síntese do supervisor com sucesso. A Figura 12 apresenta o menu para criação de plantas, realização da síntese das subplantas, e cálculo do supervisor.

¹ Disponível em: <<https://www.control.utoronto.ca/~wonham/>>



Figura 12 – Menu do TCT

Os itens do menu que são utilizados neste projeto são, *Create*, para criar os autômatos das plantas dos equipamentos e também do conjunto de especificações, onde o usuário informa o número de estados, transições e é gerado o autômato (WONHAM, 2017).

É utilizado em seguida o *Sync* para fazer a composição das plantas dos equipamentos e das especificações da planta. O usuário informa o nome definido dos autômatos criados para fazer a composição e é gerada a planta composta com seus estados e transições (WONHAM, 2017).

E por último é utilizado o *Supcon* que tem a função de obter os supervisores, implementar as máximas sublinguagens controláveis, gerar os supervisores com seus estados e suas transições (WONHAM, 2017).

Uma das suas maiores vantagens, é que o programa faz todo o cálculo, sem o usuário precisar se preocupar, além de obter as linguagens desejadas, e a obtenção dos supervisores que implementam as máximas sublinguagens controláveis. Além disso, permite ainda a geração de imagens dos autômatos a partir das plantas que são geradas. Seu principal requisito, é que o usuário precisa estudar como funciona o sistema para gerar as linguagens, disponível ao baixar o programa.

Neste projeto o TCT foi de extrema importância na síntese dos supervisores, para identificar a quantidade de estados e transições das subplantas, além de ser possível obter os supervisores ótimos da planta, identificá-los na simulação realizada e gerar os resultados das composições, facilitando o entendimento e ajudando a comprovar os resultados.

2.5 Administração de Materiais

A administração segundo Maia (2013), é entendida como o planejamento, a atividade, domínio e gestão. Já os materiais são produtos que podem ser contabilizados e entram em um processo de fabricação de uma organização e outros materiais exclusivos.

O real objetivo da administração de materiais é afirmar que o fornecimento sucessivo dos materiais estão na criação dos itens em virtude da programação ligada nas áreas de venda e fabricação (GUERRA, 2006).

A baixa dos custos para investir mais em estoques, ou seja, aumentar o uso competente dos tipos de planejamento e controle, também é um dos objetivos dessa administração dos objetos. A partir disso é provável que haja uma contribuição maior para que haja maximização do lucro acima do dinheiro aplicado (GONCALVES, 2007).

2.5.1 Estoques

Estoque é toda acumulação de produtos que são armazenadas em um sistema e muitas vezes isso é causado porque a demanda é menor que o fornecimento. Porém, esses armazenamentos são custosos e podem representar riscos, assim como danos ou até mesmo o fim da validade do produto (BOWERSOX, 2014).

A Figura 13 explica bem a teoria de (BOWERSOX, 2014):

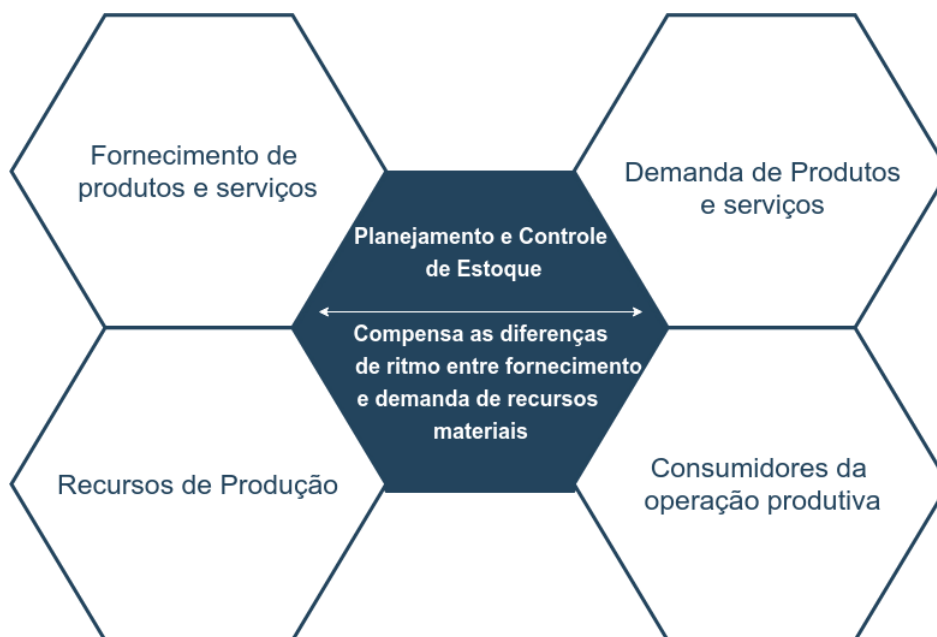


Figura 13 – Gestão de estoques (MAIA, 2013)

Os estoques são divididos também de acordo com Guerra (2006), em:

- Estoque de segurança: tem a função de substituir um produto caso haja algum defeito;
- Estoque de ciclo: é quando não há possibilidades de entregar os produtos todos de uma vez, pois estes são fabricados por etapas, por exemplo, uma padaria;

- Estoque de antecipação: são aqueles produtos de grande demanda e que pode variar, como exemplo, refrigerante;
- Estoque no canal de distribuição: são aqueles estoques que não podem ser levados no mesmo momento para o ponto de venda.

2.5.2 Sistemas de Armazenagem

Esse tipo de armazenagem ou depósito é uma prática realizada característica da logística de ferramentas, no qual é encarregada pela conservação das ferramentas armazenadas de uma instituição. Essas ferramentas são exemplos de aplicação, pois os sistemas de armazenagem podem ser aplicados a qualquer gestão de estoques. Seu maior intuito é conservar sua integridade até que alguém obtenha o produto ([BOWERSOX, 2014](#)).

A aplicação de procedimentos da armazenagem facilita para que haja uma baixa nos gastos com procedimentos e danos por estragos nos produtos, além de obter uma melhoria na qualidade dos produtos e nas condições de segurança de operação dos depósitos de armazenagem, com isso, é possível detectar também uma grande diminuição em relação aos danos dessas ferramentas ([GUERRA, 2006](#)).

Na Figura 14, apresenta um exemplo de um sistema de armazenagem de uma empresa, no qual é possível visualizar um grande galpão com estruturas para armazenar caixas com produtos:



Figura 14 – Sistema de Armazenagem ([MAIA, 2013](#)).

Desse modo, quando há um sistema de armazenagem, há também rapidez nos fluxos internos, pois potencializa a assistência ao cliente, ou seja, o cliente recebe o produto de forma mais rápida, e isso é considerado um sistema de armazenagem eficaz (GONCALVES, 2007).

Neste contexto, a implantação de soluções de controle que viabilizem uma maior eficiência entre as operações dos equipamentos, permitindo favorecer a melhoria da dinâmica de coordenação das máquinas são de extrema importância. A teoria de controle supervisorio aplicada a SED's apresenta-se como uma boa alternativa para resolução dos problemas de coordenação de equipamentos. Esta teoria oferece uma metodologia para modelar o conjunto de operações, inclusive as que acontecem de forma concorrente, e fornecer um supervisor ótimo que evita possíveis situações de bloqueio ou conflitos entre operações (GONCALVES, 2007).

2.6 Simuladores

A simulação é nada mais que, uma maneira de obter uma resposta de um determinado problema a partir de um estudo de um modelo que retrata o funcionamento de um sistema utilizando alguma máquina digital (PEREIRA, 2006).

O uso da simulação está cada vez mais frequente em diversas áreas do mercado de trabalho, assim como, na administração, engenharia, biologia, informática, medicina e entre várias outras, que possuem como intuito buscar soluções para os problemas que passam no dia a dia (TREBESQUIM, 2013).

O crescimento dos estudos na área de simulação implica diretamente no crescimento das ferramentas disponíveis para simular, que são os simuladores. A facilidade do uso dessas ferramentas e do aprimoramento dos ambientes, além das interfaces gráficas que são intuitivas, fornece uma melhor compreensão no desenvolvimento de modelos computacionais (SANTOS, 2013).

Desta forma, é possível descobrir algumas vantagens e desvantagens no uso da simulação que alguns autores afirmam. Os autores Banks (2010) e Bouzada (2013) reconhecem algumas das vantagens que são: a possibilidade do controle do tempo onde se pode alterar a velocidade em que cada processo pode ocorrer e a reparação com uma maior facilidade dos gargalos por meio visual.

Já a desvantagem principal é que, para conseguir construir o modelo correto é exigida certa experiência, pois são muitos detalhes a ser tratado, e os resultados que o sistema emite não são tão fáceis de compreender.

2.6.1 Software Simul8

O Simul8² é software de simulação para eventos discretos, ele permite que você crie um modelo visual do sistema desenhando objetos diretamente na tela. Esses objetos podem ser filas ou processos.

As características dos objetos podem ser definidas em termos, por exemplo, de capacidade ou mesmo, velocidade. Uma vez que o sistema foi modelado, uma simulação foi realizada e o fluxo de itens de trabalho ao redor do sistema é exibido pela animação na tela para que a adequação do modelo possa ser avaliada (CHWIF, 2006).

Na Figura 15 ilustra a interface do sistema junto a um modelo já realizado como exemplo:

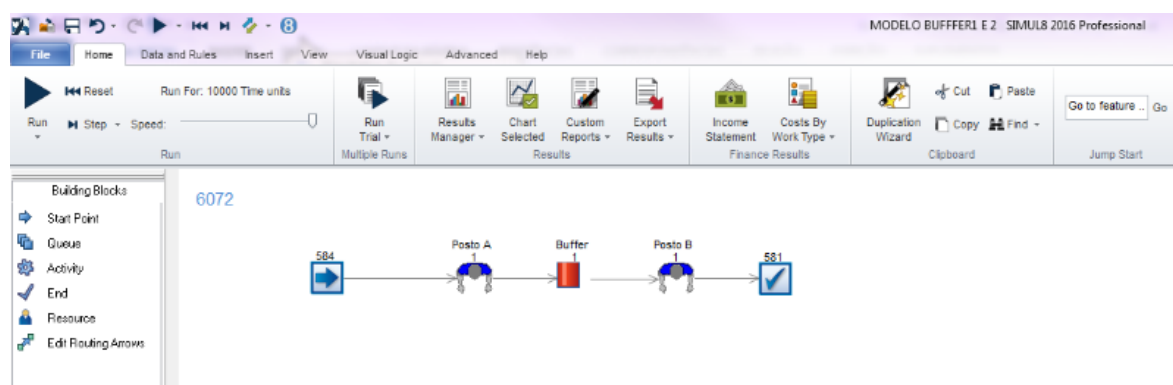


Figura 15 – Interface Gráfica do Software Simul8 (BOTTER, 2016)

Os *Building Blocks* como ilustrado no canto esquerdo da Figura 15 são os blocos de construção do modelo, ou seja, são os comandos que permitirá que seja construído os modelos para simular. O *Start Point* gera os objetos que passam por um processo denominado *Activity*, que é o responsável pelas atividades que são processadas. Podem também passar por uma fila que no bloco é encontrada por *Queue*, e por fim, o bloco *End*, onde finaliza todo o modelo (BOTTER, 2016).

A Figura 15 ilustra exatamente a interface do sistema e um modelo de uma simulação cuja seta representa o início da simulação, onde passa para um processo identificado pelo posto A, logo após o término do processo é enviado para uma fila chamada de *Buffer*, onde espera o processo do posto B terminar, e só então o objeto que estava esperando no *Buffer*, vai para o posto B, e então após o término do processo vai para o fim do modelo, identificado pelo marcador correto (BOTTER, 2016).

Quando toda a estrutura do modelo for confirmada, vários testes podem ser executados e o desempenho do sistema é descrito estatisticamente. As estatísticas de interesse podem ser: tempo médio de espera e utilização dos processos ou recursos (MOREIRA, 2003).

² Disponível em: <<https://www.simul8.com/>>

2.6.2 Software Arena

O software de simulação ARENA³ utiliza uma interface gráfica intuitiva, onde o usuário possui menos dificuldades para poder montar a representação real do problema, a partir disso é otimizado a complexidade, e a pessoa, não precisa ter conhecimento avançado de programação, pois utiliza praticamente somente o mouse arrastando blocos (TREBESQUIM, 2013).

Na Figura 16 tem-se a interface do sistema:

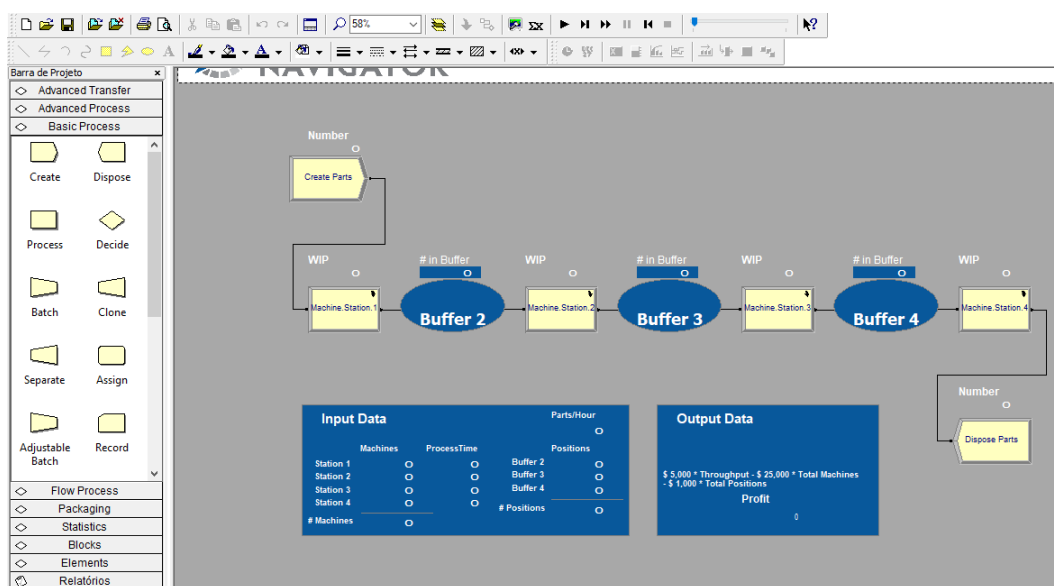


Figura 16 – Interface do Software Arena (TREBESQUIM, 2013)

O software possui também alguns instrumentos bem importantes, como é o caso do *Analyzer* e *Output Analyser*. O *Input Analyzer* escolhe a distribuição que possui a melhor estatística aplicada a eles, a partir da análise real dos dados do processo. Já o *Output Analyser* no decorrer da simulação faz um estudo dos dados.

O software Arena possui muitas vantagens, além de possuir uma versão paga que possuem elementos 3D. A versão básica também é muito funcional e pode resolver os problemas encontrados no sistema podendo reduzir os custos operacionais melhorando a previsão financeira (MAROLLA, 2017).

O Arena foi desenvolvido sobre linguagem SIMAN (Simulation Modeling and Analysis) e possui a habilidade de manipular modelos que são baseados em interação de processos, simulação contínua, eventos discretos ou até mesmo a fusão destas.

³ Disponível em: < <http://www.paragon.com.br/arena-academico-student/> >

2.6.3 Software FlexSim

O Software de simulação FlexSim⁴ permite facilitar a modelagem da realidade virtual, o projeto de experimentos, a análise de resultados, a coleta de dados, a otimização do sistema e a criação de aplicativos pelo usuário de forma contínua ou discreta no tempo, com supervisores que habilitam ou desabilitam eventos que são gerados. Todos os modelos criados de simulação são visualizados em 3D, onde fica muito mais fácil de reconhecer alguns gargalos que são formados em uma linha de produção e até mesmo outros tipos de problemas. Além disso é fornecido ferramentas para fazer um relatório estatístico a partir de qualquer objeto no software.

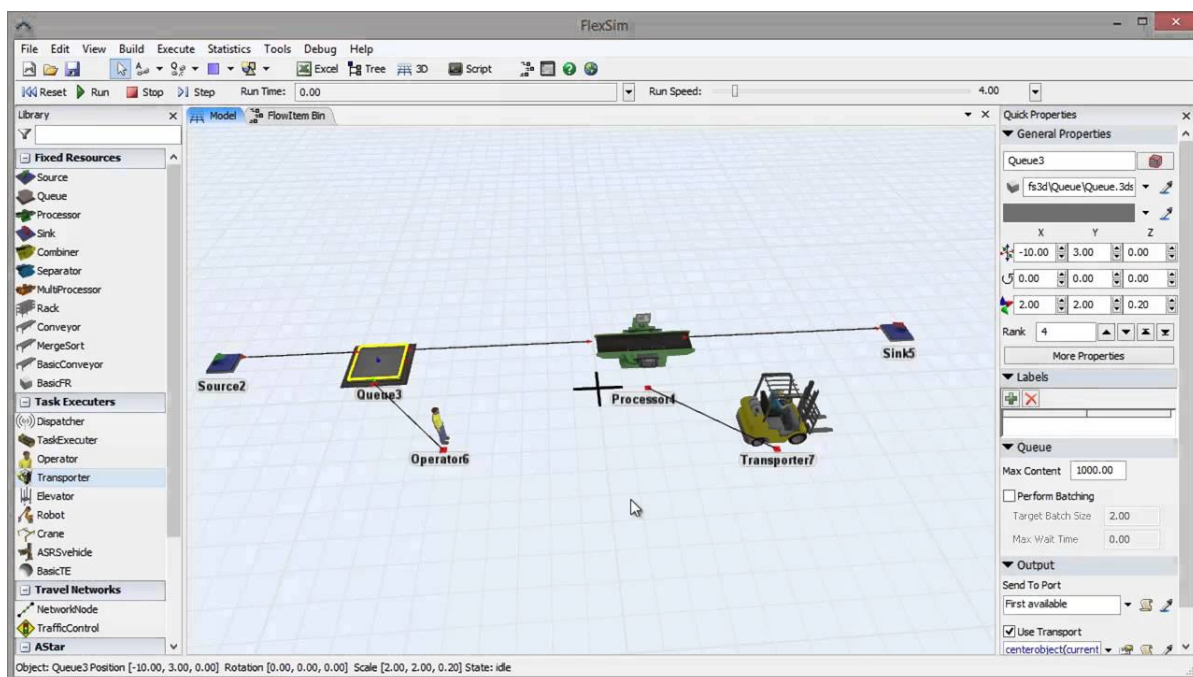


Figura 17 – Interface do Software FlexSim

A partir da análise da Figura 17 o FlexSim permite visualizar com mais facilidade os movimentos dos objetos de uma forma muito melhor que os outros softwares já apresentados, como o Arena e Simul8. O FlexSim ainda permite atribuir variáveis, e funções via código-fonte do próprio simulador. É possível cada objeto ter seu código como na Figura 18 a seguir:

⁴ Disponível em: < <https://www.flexsim.com/pt/> >

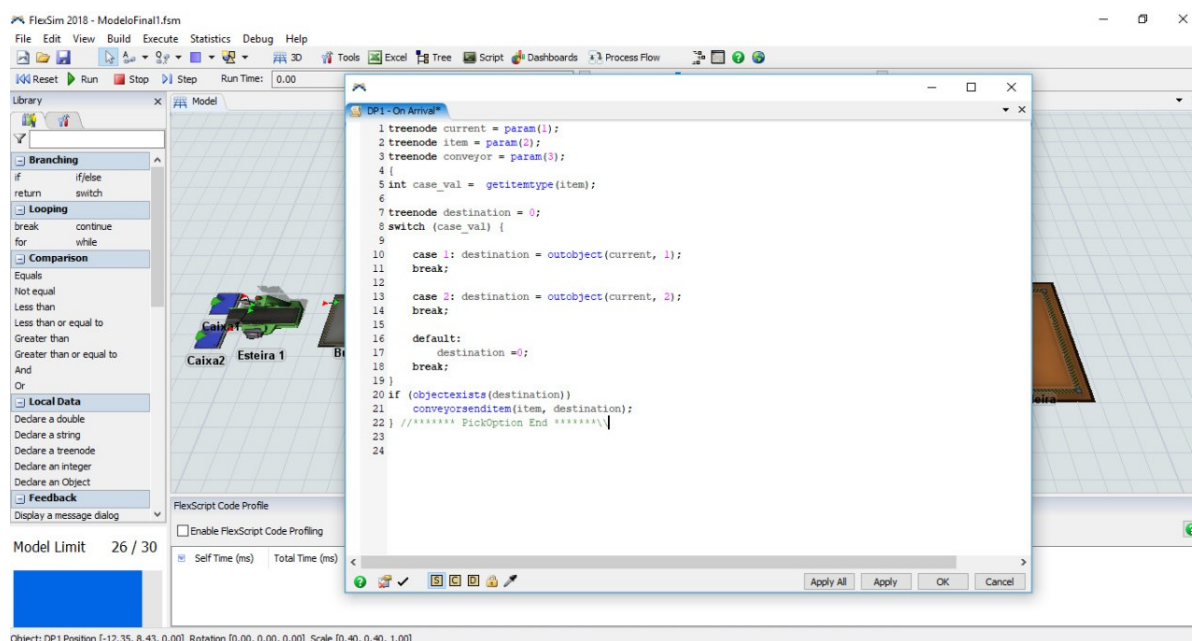


Figura 18 – Interface do Software FlexSim

O FlexSim suporta C++, OpenGL e fornece seu próprio script de linguagem. Foi desenvolvido usando métodos orientados a objeto e C++, com uma camada de abstração de banco de dados proprietária, além de facilitar a conexão com banco de dados SQL externos.

As desvantagens encontradas no software é que o programa necessita de no mínimo 4 GB de RAM, sistema operacional Windows e uma GPU que suporta OpenGL 3.1 ou superior. Sem esses requisitos o computador pode se tornar muito lento e não ter o mesmo aproveitamento. Outra desvantagem é que a versão gratuita oferecida ao usuário tem um limite de 30 objetos para inserir no modelo, caso necessite de um limite maior, é necessário comprar a licença que tem um alto custo.

3 Metodologia

3.1 Materiais

Para criação dos diagramas de fluxo utilizou-se o software Microsoft Office Visio¹, na versão Visio 2016. Esta ferramenta é muito utilizada para facilitar o entendimento do fluxo de processos. É um programa intuitivo, muito fácil de ser usado, pois é preciso apenas arrastar blocos e possui imagens para customizar o diagrama da melhor forma possível (PARKER, 2016).

Para gerar as imagens dos autômatos com seus estados e transições foi utilizado o programa Auger², na versão 3.1. O software é um ambiente para construção e também simulação de autômatos finitos que foi utilizado para demonstrar o funcionamento dos autômatos.

Para viabilidade do trabalho, utilizou-se as seguintes ferramentas de hardware e software na Tabela 3:

Tabela 3 – Materiais Utilizados

DESCRIÇÃO	FABRICANTE	VERSÃO/MODELO
SO WINDOWS	Microsoft	10 - 64 bits
Processador	Intel	I3-3110M 2,4Ghz
Memoria RAM	Kingston	8gb DDR3
Disco Rígido	Seagate	500GB, 5400 rpm
Visio	Microsoft	2016
Auger	Evoluma	3.1
FlexSim	FlexSim Software Products	2018 Update 2

3.2 Modelagem do Sistema

Nesta seção é apresentada a modelagem do sistema de armazenagem, inicialmente pela a diagramação da planta, e o detalhamento das entidades que o compõem o cenário. Tem-se ainda a dinâmica das operações e funcionalidades dessas entidades.

3.2.1 Planta do Sistema de Armazenagem

O sistema de armazenagem é composto por três esteiras. Sendo um coletor de cargas, um atuador hidráulico e um robô, ilustrado na Figura 19.

¹ Disponível em: <<https://products.office.com/pt-br/visio/flowchart-software?tab=tabs-1/>>

² Disponível em: <<http://www.evoluma.com/auger/index.html/>>

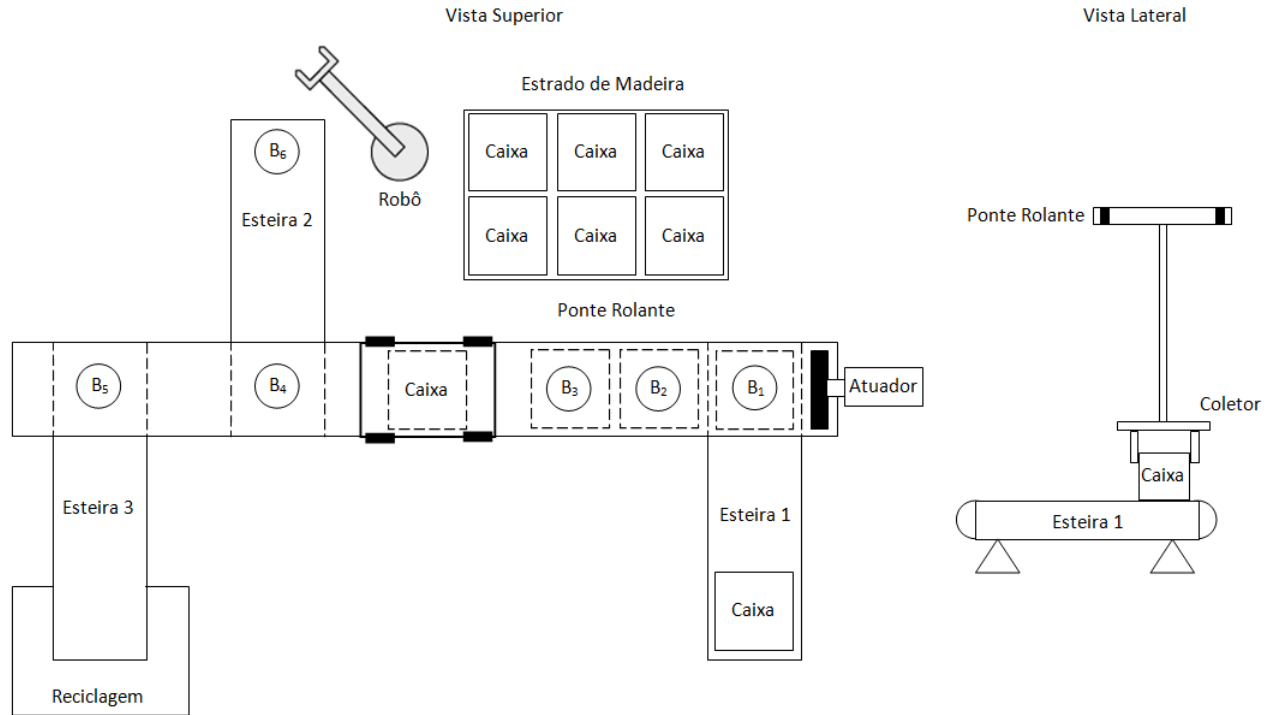


Figura 19 – Estrutura da Estação de Armazenamento

Em cada área de máquinas há um sistema de armazenamento provisório unitário que pode ser chamado de *buffer* e sendo identificado por B_i , $i = 1, 2, 3, 4, 5, 6$.

Conforme ilustrado na Figura 19 as caixas que serão armazenadas passam por um processo de vistoria dimensional que fica situado no atuador hidráulico. Quando apresentam dimensões uniformes são conduzidas para a esteira 2, caso contrário, serão enviadas para a esteira 3. O atuador hidráulico tem a finalidade de mover as caixas uniformes para o *buffer* B_2 e as fora do padrão para o *buffer* B_3 . O coletor de cargas coleta as caixas nos *buffers* B_2 e B_3 e as desloca até os *buffers* B_4 e B_5 , localizados no início das esteiras 2 e 3, respectivamente.

0As caixas colocadas na esteira 3 são conduzidas até um reservatório para reciclagem. Já as caixas conduzidas para a esteira 2 são encaminhadas até o seu final, onde o robô as coleta e faz o depósito sobre um tablado de madeira de modo a formar um bloco de caixas.

O comportamento desejado para a planta poderá ser descrito pela seguinte sequência operacional:

1. Esteira 1, avança e posiciona a caixa no *buffer* B_1 ;
2. O atuador hidráulico empurra as caixas com dimensões padronizadas para o *buffer* B_2 , ou para B_3 caso tenham dimensões despadronizadas;
3. O coletor coleta a caixa nos *buffers* B_2 ou B_3 ;

4. O coletor movimenta-se para depositar a caixa em B_4 (tamanho padrão) ou B_5 (tamanho fora do padrão);
5. Se a caixa está no buffer B_5 a esteira 3 avança até que a caixa seja depositada em um recipiente para reciclagem localizado ao final da esteira 3;
6. Se a caixa está no buffer B_4 a esteira 2 avança e posiciona a caixa no buffer B_6 ;
7. O robô coleta as caixas em B_6 as coloca sobre um estrado de madeira até que um bloco de caixas seja formado para ser levado até o depósito.

3.2.2 Modelagem Modular Local

O fluxo de tarefas para permitir o correto funcionamento do sistema de armazenagem é ilustrado na Figura 20. Seguindo a metodologia de representação dos eventos sugeridos por Wonham (2017), os eventos sinalizados com números ímpares correspondem aos eventos controláveis, ou seja, que podem ser habitados e desabilitados pelo sistema de controle, e os eventos sinalizados com números pares representam os eventos sobre os quais não existe atuação de controle.

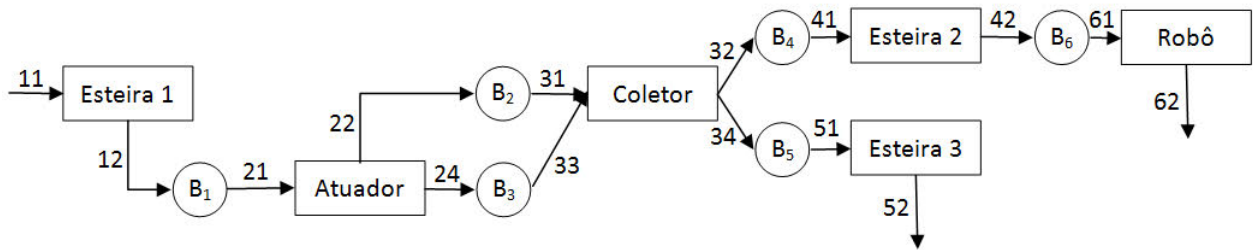


Figura 20 – Diagrama de Fluxos

Conforme ilustrado na Figura 20, as seguintes transições podem ser observadas na Tabela 4.

Tabela 4 – Transições da Planta

Transição	Operação na Planta
11	Início de operação da esteira 1
12	Caixa posicionada no buffer B_1 (final da esteira 1)
21	Início de operação do atuador hidráulico
22	Caixa padronizada posicionada no buffer B_2
24	Caixa despadronizada posicionada no buffer B_3
31	Coletor de cargas pega caixa no buffer B_2
32	Coletor de cargas deposita caixa no buffer B_4 (início da esteira 2)
33	Coletor de cargas pega caixa no buffer B_3
34	Coletor deposita caixa no buffer B_5 (início da esteira 3)
41	Início de operação da esteira 2
42	Caixa padronizada posicionada no buffer B_6 (final da esteira 2)
51	Início de operação da esteira 3
52	Caixa despadronizada depositada no recipiente de reciclagem
61	Robô coletou caixa no buffer B_6
62	Robô depositou caixa no estrado de madeira

Os procedimentos para síntese do Controle Modular Local, foram obtidos usando a seguinte sequência de passos (PENA, 2013):

- Modelagem das plantas locais G_i , para $i \in I$;
- Cálculo da mais refinada representação por sistema produto (RSP);
- Obtenção das especificações genéricas E_j , para $j \in J$;
- Obtenção da planta local G_j , para cada especificação, $j \in J$;
- Obtenção das linguagens desejadas K_j , $K_j = E_j \parallel L_M(G_j)$ e obtenção das componentes aparadas para todo $j \in J$;
- Obtenção dos supervisores que implementam as máximas sublinguagens controláveis e.r.a G contidas em K_j , $S_j = \text{SupC}(K_j, L(G))$, para $j \in J$;
- Verificar a modularidade local dos supervisores obtidos.

Dessa forma, após a identificação e definição da dinâmica do sistema de armazenagem, e seguindo os procedimentos de síntese supracitados, partiu-se propriamente para a obtenção dos autômatos que representassem o funcionamento isolado das máquinas, e por conseguinte a modelagem das especificações, consistindo nas inter operações entre os equipamentos envolvidos (CURY, 2001).

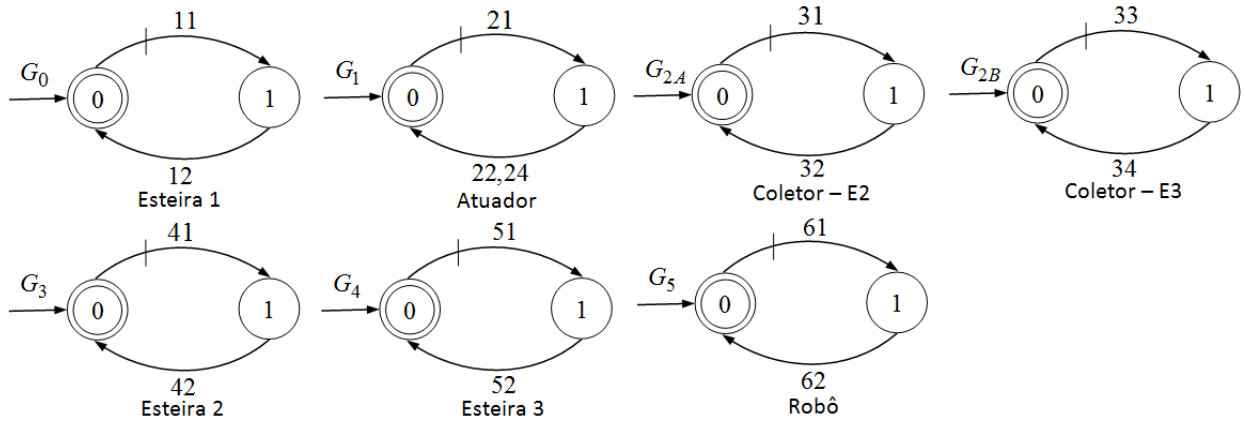


Figura 21 – Autômatos das Plantas dos Equipamentos

A modelagem dos equipamentos por autômatos, de forma geral, consiste de dois estados e dois eventos, indicando início e fim de operação do equipamento (liga e desliga, por exemplo), conforme ilustrado na Figura 21.

As esteiras 1, 2 e 3 iniciam seu funcionamento, pelos eventos controláveis 11, 41 e 51, respectivamente. A ação do atuador hidráulico será sempre precedida do acionamento do *buffer* B_1 , e a sua saída dependerá das medições realizadas pelos sensores que estão acoplados ao mesmo.

O coletor atua na saída do atuador hidráulico, onde os estados controláveis 31 e 33 implicarão, por conseguinte no acionamento das esteiras 2 e 3, respectivamente. Os *buffers* intermediários B_4 e B_5 terão suas saídas vinculadas as entradas das esteiras 2 e 3, conforme comentado anteriormente. Por fim, as saídas não controláveis das esteiras 2 e 3 implicarão na atividade final do sistema de armazenagem, onde as caixas serão armazenadas ou enviadas para reciclagem, eventos 52 e 62.

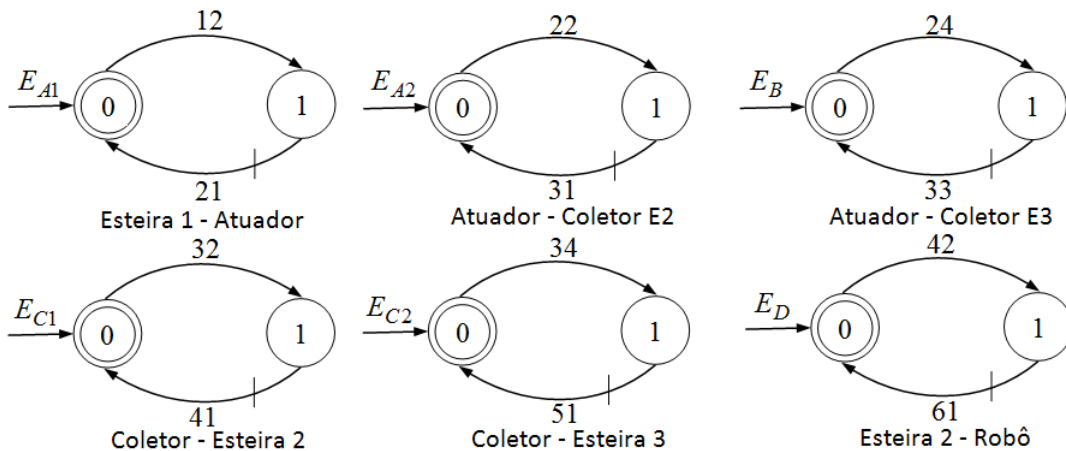


Figura 22 – Conjunto de Especificações

As especificações, que correspondem às interações entre os equipamentos, ilustradas na Figura 22, seguem a mesma estrutura das plantas dos equipamentos individuais, sendo

formadas por dois estados e dois eventos, conforme estrutura apresentada na Tabela 5.

Tabela 5 – Estrutura e Transições

Estrutura	Transições
E_{A1}	Movimentação de caixas no <i>buffer</i> B_1 (final da esteira 1)
E_{A2}	Movimentação de caixas no <i>buffer</i> B_2 (caixas padronizadas)
E_B	Movimentação de caixas no <i>buffer</i> B_3 (caixas fora do padrão)
E_{C1}	Retirada de caixas do <i>buffer</i> B_4 (início da esteira 2)
E_{C2}	Retirada de caixas do <i>buffer</i> B_5 (início da esteira 3)
E_D	Retirada de caixas do <i>buffer</i> B_6 (final da esteira 2)

A composição das plantas dos equipamentos, bem como das plantas com as especificações foram obtidas utilizando o software TCT, usando o *Sync*, que é o produto síncrono das subplantas, conforme a Tabela 6 e Tabela 7.

Tabela 6 – Composição das Plantas

Planta Composta	Estados	Transições
$G_{A1} = G_0 \parallel G_1$	4	10
$G_{A2} = G_1 \parallel G_{2A}$	4	10
$G_B = G_1 \parallel G_{2B}$	4	10
$G_{C1} = G_{2A} \parallel G_3$	4	8
$G_{C2} = G_{2B} \parallel G_4$	4	8
$G_D = G_3 \parallel G_5$	4	8

Na Tabela 6 temos o G_{A1} que é a composição das plantas G_0 , que representa a esteira 1 iniciando a operação e posicionando a caixa no *Buffer* 1 e G_1 que representa o início do atuador movimentando as caixas padronizadas para o *Buffer* 2 e despadronizadas para o *Buffer* 3, onde após a composição, a planta é composta por 4 estados e 10 transições.

Em seguida temos o G_{A2} que é composta por G_1 , atuador, e G_{2A} , coletor que pega caixa no *Buffer* 2 e deposita no *Buffer* 4, formando assim uma planta composta com 4 estados e 10 transições.

O G_B , que é composto por G_1 , atuador, e G_{2B} , coletor que pega caixa no *Buffer* 3 e deposita no *Buffer* 5, formando assim uma planta composta com 4 estados e 10 transições.

A planta composta G_{C1} , de G_{2A} representando o coletor, e G_3 representando o início da esteira 2 e a caixa padronizada posicionada no *Buffer* 6, formando 4 estados e 8 transições. A planta G_{C2} , composta por G_{2B} , representando o coletor, e G_4 representando o início da esteira 3 e caixas despadronizadas depositadas no recipiente de reciclagem, formam 4 estados e 8 transições.

Por último, temos a planta G_D , composta por G_3 , esteira 2 e G_5 que representa a coleta de caixas pelo robô no *Buffer* 6 e depósito delas no estrado de madeira, formando assim uma planta composta com 4 estados e 8 transições.

Na Tabela 7 é possível visualizar uma grande quantidade de transições que foram geradas após o procedimento do TCT, pois além da composição das plantas, possuem suas especificações, ou seja, sua relação com a outra máquina, o que gera muitas transições.

Tabela 7 – Composição das Especificações + Plantas

Linguagem Obtida	Estados	Transições
$K_{A1} = E_{A1} \parallel G_{A1}$	8	16
$K_{A2} = E_{A2} \parallel G_{2A}$	8	16
$K_B = E_B \parallel G_B$	8	16
$K_{C1} = E_{C1} \parallel G_{C1}$	8	12
$K_{C2} = E_{C2} \parallel G_{C2}$	8	12
$K_D = E_D \parallel G_D$	8	16

A obtenção dos supervisores no TCT se dá pelo "*Supcom*", que implementam as máximas sublinguagens controláveis.

3.2.3 Definição do Software de Simulação

A simulação ganhou muita importância ao passar do tempo, pois permite aos projetistas que simule um sistema de grande porte, podendo calcular gastos, seu comportamento diante situações diversas. A simulação pode ser usada para estudar, comparar projetos alternativos, e até mesmo solucionar problemas de sistema existentes.

Neste projeto, a simulação ajudou a identificar os supervisores agindo após a geração de eventos das máquinas, caracterizando-se como eventos discretos, além de poder simular de várias formas as entradas das caixas, gerando resultados esperados e validando a modelagem da planta.

Foram analisados três softwares de Simulação, *FlexSim*³, *Arena*⁴ e *Simul8*⁵. São softwares que auxiliam bastante a simular os eventos que ocorrem e os estados das máquinas, porém o *FlexSim* se destacou durante os testes, pois o mesmo apresenta uma interface em 3D que é possível visualizar com muito mais clareza os estados das máquinas e também de seus eventos que ocorrem discretamente no tempo que há a possibilidade de visualizar dados estatísticos de uma determinada máquina no momento de sua simulação, conforme ilustrado Figura 23.

³ Disponível em: <<https://www.control.utoronto.ca/~wonham/>>

⁴ Disponível em: <<http://www.paragon.com.br/software/arena/>>

⁵ Disponível em: <<https://www.simul8.com/viewer/>>

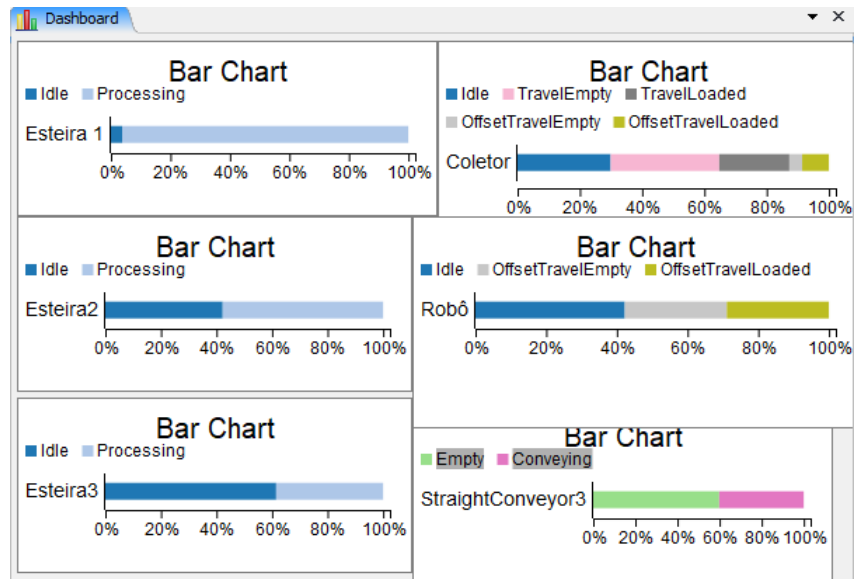


Figura 23 – Estado Máquinas Plantas no *FlexSim*

Além disso, a obtenção de uma licença de estudante durante um ano, para que pudesse usufruir de todas as ferramentas do software apenas com um limite de objetos para serem modelados que para esta planta foi de bom tamanho. O *Arena* oferece uma licença de 30 dias e recursos reduzidos. O *Simul8* oferece somente a versão paga. A partir destas vantagens, optou-se pelo software *FlexSim* para poder validar a modelagem da planta realizada.

4 Resultados

4.1 Obtenção dos Supervisores

Para que fosse possível a obtenção dos supervisores, buscou-se inicialmente obter a composição das plantas, realizada na Tabela 6 utilizando a função *Sync* do software TCT, que é o produto síncrono das subplantas. Na função inseriu-se o nome definido dos autômatos criados para fazer a composição, e gerou-se a planta composta com seus estados e transições.

Logo após a composição das plantas buscou-se obter a composição das especificações com as plantas compostas geradas a partir da Tabela 7 utilizando a função *Sync* e inserindo o nome de identificação da planta composta e da identificação do conjunto de especificações, obtendo a linguagem K_j com seu número de estados e transições.

A partir da linguagem obtida K_j , foi possível obter os supervisores utilizando a função *Supcon* do TCT, onde implementou-se as máximas sublinguagens controláveis inserindo no software o resultado da linguagem obtida com a planta composta, resultando nos supervisores, com seu número de estados e transições, conforme a Tabela 8.

Convém salientar que para realizar a composição entre as plantas, aparagem dos autômatos e cálculos dos supervisores além da verificação de bloqueios, utilizou-se a aplicação TCT, disponível em Wonham (2017). Obteve-se seis supervisores, conforme a Tabela 8 abaixo:

Tabela 8 – Supervisores ótimos

Supervisor	Estados	Transições
$S_{A1} = G_{A1} \parallel K_{A1}$	6	11
$S_{A2} = G_{A2} \parallel K_{2A}$	6	10
$S_B = G_B \parallel K_B$	6	10
$S_{C1} = G_{C1} \parallel K_{C1}$	6	8
$S_{C2} = G_{C2} \parallel K_{C2}$	6	8
$S_D = G_D \parallel K_D$	6	8

Onde:

- S_{A1} : é o supervisor da especificação E_{A1} (Movimentação de caixas no *buffer* B_1 , final da esteira 1);
- S_{A2} : é o supervisor da especificação E_{A2} (Movimentação de caixas no *buffer* B_2 , caixas padronizadas);

- S_B : é o supervisor da especificação E_B (Movimentação de caixas no *buffer* B_3 , caixas fora do padrão);
- S_{C1} : é o supervisor da especificação E_{C1} (Retirada de caixas do *buffer* B_4 , início da esteira 2);
- S_{C2} : é o supervisor da especificação E_{C2} (Retirada de caixas do *buffer* B_5 , início da esteira 3);
- S_D : é o supervisor da especificação E_D (Retirada de caixas do *buffer* B_6 , final da esteira 2);

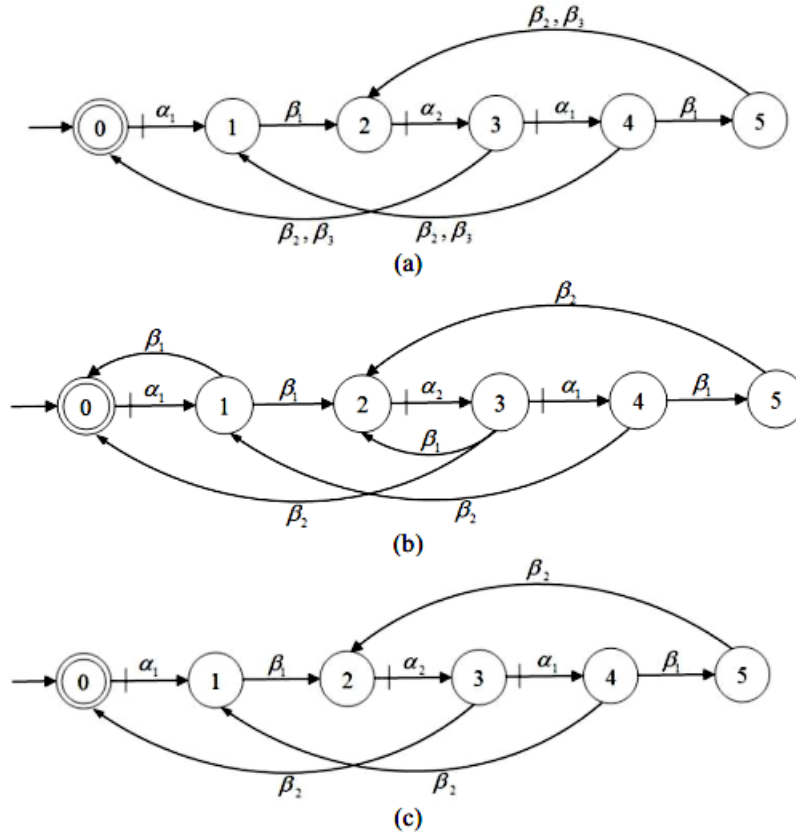


Figura 24 – Supervisores Ótimos. (a) S_{A1} . (b) S_{A2} e S_B . (c) S_{C1} , S_{C2} e S_D .

A Figura 24(a) ilustra o supervisor ótimo para a especificação E_{A1} , que controla a movimentação de caixas no *buffer* B_1 . Os eventos controláveis α_1 e α_2 correspondem aos eventos 11 e 21 da Figura 20, que iniciam a operação da esteira 1 e atuador, respectivamente. Os eventos não controláveis β_1 , β_2 e β_3 indicam a presença de caixa nos *buffer* B_1 , B_2 e B_3 , respectivamente.

A Figura 24(b) ilustra os supervisores para as especificações E_{A1} e E_{A2} que controlam a coleta de caixas nos *buffer* B_2 e B_3 . O evento α_1 representa o início de operação do

coletor, que pode assumir o valor 31, caso a caixa tenha tamanho padrão ou 33 caso as dimensões estejam fora do padrão. O evento não controlável β_1 indica a presença de caixa no *buffer* B_2 e/ou B_3 , enquanto que β_2 indica a presença de caixa nos *buffers* B_4 e/ou B_5 , ou seja, que as caixas foram depositadas na esteira 2 ou 3 pelo coletor.

A Figura 24(c) mostra os supervisores para as especificações E_{C1} , E_{C2} e E_D , que coordenam quando as esteiras 2, 3 e o robô devem ser ligados. O evento α_1 indica o início de operação do coletor e β_1 que a caixa foi depositada na esteira 2 ou na esteira 3. O evento α_2 indica o início de operação da esteira 2 ou 3, com β_2 indicando a presença de caixa para estocagem ou presença de caixa para reciclagem, respectivamente. No caso da especificações E_D o evento α_1 indica início de operação da esteira 2 com β_1 indicando a presença de caixa para estocagem. O evento α_2 representa o início de operação do robô e β_2 que a caixa foi armazenada.

Pode ser notado que os supervisores obtidos atuam sobre as plantas locais, desabilitando eventos controláveis que possam ocorrer na planta, e que possam comprometer a sequência de operações do sistema em geral, utilizando a arbodagem modular local cada supervisor tem no máximo 6 estados e 11 transições.

Os resultados comprovam que a aplicação da técnica de controle modular realmente traz muitas vantagens, pois evita o número excessivo de estados e eventos, quando comparado à outras abordagens. Esta característica possibilita o uso menor de recursos computacionais para processamento do controle supervísório, além de simplificar a manutenção do sistema por inteiro, visto que as mudanças serão pontuais, isto é, em supervisores específicos.

4.2 Validação da Modelagem

Para a simulação da planta utilizou-se o software FlexSim, pois é um simulador mais fácil de ser utilizado, além dos seus gráficos em 3D, que deixam uma visão mais realística da simulação, também por seu conteúdo de informações ser extenso na internet, com livros e vídeos que ensinam desde o básico ao avançado, dentre outros aspectos discutidos na seção 3.

A modelagem da simulação foi baseada na Figura 19 que é o modelo atual do projeto. Com a simulação é possível entender bem o processamento de todos os eventos discretos, em qual momento ocorre a mudança dos estados de forma discreta no tempo.

A Figura 25, apresenta os supervisores ótimos, controlando as máquinas, habilitando e desabilitando os eventos para que não ocorra estados proibidos, bloqueios e toda a planta funcione corretamente, sem interrupção e sem erro.

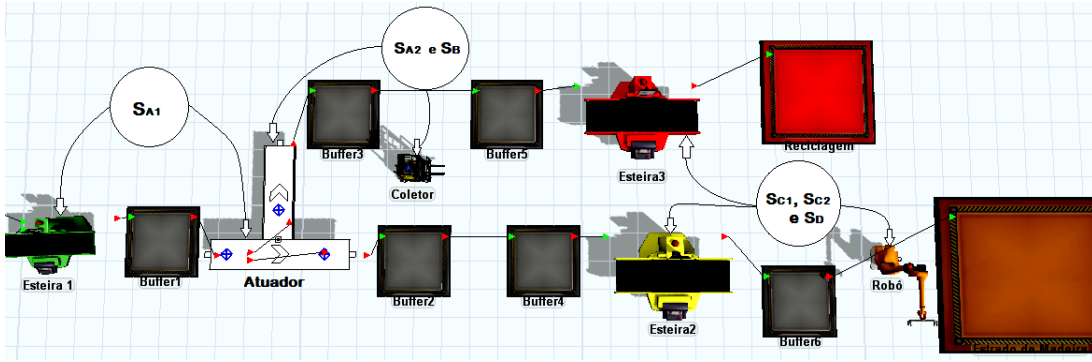


Figura 25 – Simulação Sistema de Armazenagem com os supervisores.

Observa-se na Figura 25 o supervisor ótimo " S_{A1} " para a especificação E_{A1} que controla a movimentação de caixas no *buffer* 1, onde ao enviar o evento de presença de caixas neste *buffer*, o supervisor habilita o início da operação do atuador hidráulico.

No supervisor ótimo " S_{A2} e S_B ", que são, respectivamente das especificações E_{A1} e E_{A2} , que controla a movimentação de caixas no atuador para os *buffers* B_2 e B_3 , que gera os eventos de presença das caixas e o supervisor aciona o coletor, iniciando coleta de caixas nos *buffers* B_2 e B_3 , realizando o transporte delas para os *buffers* B_4 e B_5 .

O último supervisor ótimo " S_{C1} , S_{C2} e S_D " são referente às especificações E_{C1} , E_{C2} e E_D que coordenam quando as esteiras 2, 3 e o robô devem ser ligados. A partir de um evento gerado do *buffers* 4, a esteira 2, é habilitada pelo supervisor, e caso o evento ocorra no *buffers* 5, o supervisor habilita a esteira 3 para que façam o transporte das caixas. A esteira 3 deposita as caixas para a reciclagem. Enquanto a esteira 2 deposita no *buffer* 6, e com isso é gerado um evento para o supervisor que habilita o robô para pegar as caixas e depositar no estrado de madeira.

A Simulação começa com o objeto *Source* definido pelas nomenclaturas "Caixa1" e "Caixa2" que são respectivamente, as caixas fora do padrão (caixa vermelha) e caixas padronizadas (caixa marrom), no qual foi definido a saída de 100 caixas, onde 60% da saída foram de caixas padronizadas e 40% das caixas fora do padrão.

A "Esteira 1" é representada pelo *Processor* quem tem a função de movimentar a caixa até o "Buffer 1" e capacidade de transportar apenas uma caixa por vez. Quando a caixa chega ao "Buffer 1" que é representada pela *Queue* e tem capacidade de armazenar apenas uma caixa, o objeto *Conveyor*, "Atuador" é ativado realizando a movimentação das caixas para o "Buffer 2" ou "Buffer 3"

Dentro do atuador existe ainda um objeto chamado *Decision Point*, que é um ponto de decisão, onde ele verifica o tipo da caixa, e dependendo disso, ele movimenta a caixa para um *buffer* ou para outro. Ao criar as caixas no início da simulação é atribuído um tipo para ela, onde é possível mais à frente capturar o dado do objeto para fazer escolhas.

A Figura 26 demonstra como definiu-se o tipo dos itens de entrada determinando o tipo da caixa padronizada para 1 e da caixa despadronizada para 2. A configuração dos tipos de caixas foi de extrema importância, pois é a partir deste tipo, que o atuador movimentou as caixas para os *buffers* corretos, afim de que não houvesse nenhum erro à frente de caixas em lugares errados.

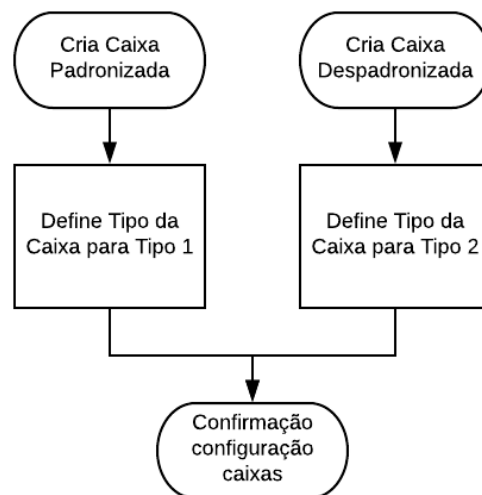


Figura 26 – Fluxograma de Configuração do tipo da Caixa

Com isso, no ponto de decisão verificou-se a alteração do destino das caixas conforme seu tipo, como ilustra na Figura 27.

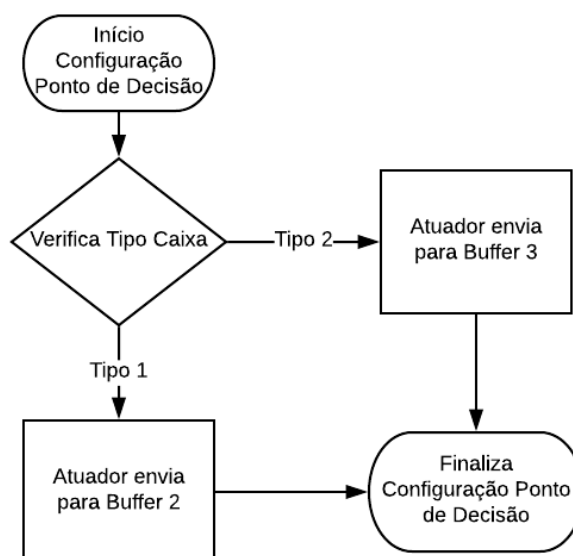


Figura 27 – Fluxograma Configuração do Ponto de Decisão

Quando as caixas são deslocadas para o "Buffer 2" ou "Buffer 3" é gerado um evento que permite a detecção e o supervisor habilita o coletor representado pela Figura 25. Ele realiza a verificação e o transporte das caixas para o "Buffer 4" caso seja padronizada, e para o "Buffer 5" caso estejam fora do padrão. A partir disso, realizou-se a configuração do coletor para fazer essa verificação para que faça o transporte correto das caixas, conforme a Figura 28.

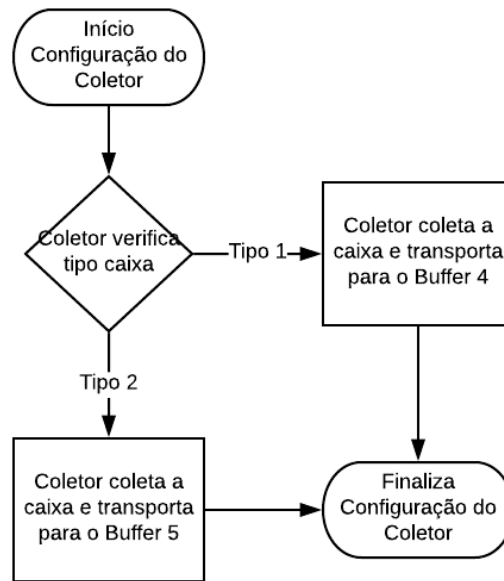


Figura 28 – Configuração dos *Buffers* para transporte das Caixas pelo "Coletor"

O evento que foi modelado permite a detecção de caixas no "Buffer 5" ou "Buffer 4" e a "Esteira 2" e "Esteira 3" iniciam a movimentação para que as caixas sejam movimentadas para seu destino. A "Esteira 3" movimenta a caixa direto para uma plataforma de reciclagem indicada na Figura 25 por "Reciclagem". Já a "Esteira 2" é acionada e a caixa é transportada para o "Buffer 6" que a partir da detecção de presença de caixa nesse *buffer* é acionado o "Robô" que faz o deslocamento da caixa do "Buffer 6" até o "Estrado de Madeira".

4.3 Resultados da Simulação

Foram realizados testes da taxa de processamento das máquinas por tempo real, onde é verificado qual máquina teve um processamento maior, e a partir disso é possível verificar a maior ocorrência de eventos na planta. A Figura 29 representa a porcentagem de processamento das máquinas em decorrer do tempo de relógio do sistema. Definiu-se inicialmente para este teste que 40% das caixas seriam fora do padrão e 60% padronizadas, com um número de entrada de caixas de 100 unidades.

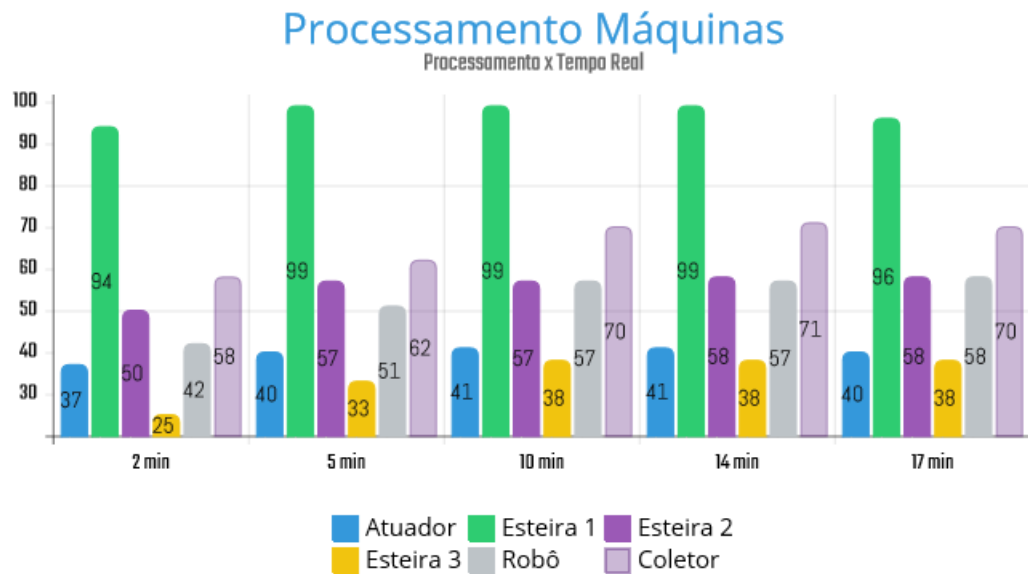


Figura 29 – Nível de Processamento das máquinas vs Tempo Real

Analisando a Figura 29, observa-se que a Esteira 1 é a mais utilizada o tempo todo, podendo definir a máquina que mais recebe eventos, e onde o supervisor mais atua, habilitando e desabilitando. A Esteira 2, tem o seu processamento baixo nos primeiros minutos, pois até a caixa padronizada chegar até a esteira, a mesma fica desabilitada, ou seja, só há processamento da esteira quando há o evento de caixas no *buffer*, por isso um menor processamento que a primeira esteira. A Esteira 3 possui uma taxa de processamento bem menor, isso interfere pois apenas 40% das caixas enviadas passam pela Esteira 3, por isso seu baixo processamento nos primeiros 2 minutos, pois as caixas demoram para chegar nela.

O processamento do Atuador não sofre tantas mudanças, pois está atrás apenas da Esteira 1, e a geração de eventos para o atuador será praticamente sempre o mesmo, independente da quantidade de caixas padronizadas e fora do padrão, pois todas passam pelo atuador.

Foram realizados 09 testes com as máquinas mostrando a quantidade de saída de caixas em determinado tempo real. Configurou-se na modelagem a saída de 100 caixas ao todo onde irão percorrer pelas máquinas em um tempo de relógio de aproximadamente 20 minutos. Pois como os eventos são discretos e a saída de caixas é porcentagem há pequenas alterações.

Os testes foram definidos conforme a Tabela 9.

A Figura 30 referente ao teste 1, apresenta um gráfico de saídas de caixas das máquinas por tempo de relógio, onde 90% das caixas que serão geradas, são padronizadas e 10% são fora do padrão.

Tabela 9 – Testes de entradas

Teste	Caixas Padronizadas (%)	Caixas Despadronizadas (%)
1	90	10
2	80	20
3	70	30
4	60	40
5	50	50
6	40	60
7	30	70
8	20	80
9	10	90

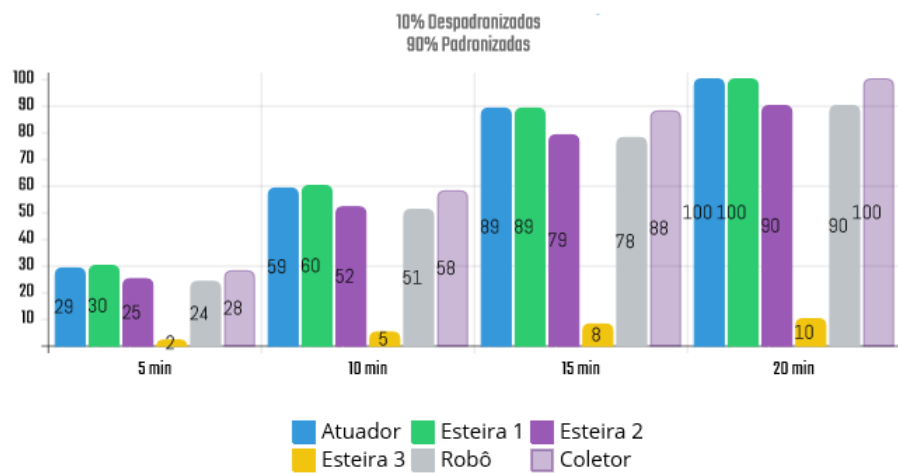


Figura 30 – Teste 1

Na Figura 30 pode ser observado que o *Atuador* e a *Esteira1* possuem quase o mesmo valor durante o início da simulação e no final eles se igualam, isso se dá pois todas as caixas padronizadas e fora do padrão passam por eles, e a pequena diferença visível é porque às vezes a caixa já passou pela esteira e está aguardando o atuador no *buffer*.

Outra máquina que passam todas as caixas é o Coletor, pois ele coleta tanto as caixas padronizadas quanto as fora do padrão em seus devidos *buffers*, podendo ser confirmado no final da simulação onde saem todas as 100 caixas pelo coletor. A Esteira 3 é que trafega as caixas despadrionizadas para a reciclagem, logo com essas métricas, houve o tráfego apenas 10 caixas, confirmando os 10%. Já para ser armazenada, as caixas padronizadas passam pelo robô, no qual saíram 90 caixas e confirma os 90% definidos no início da simulação.

A Figura 31 refere-se ao teste 2 que compreende que 80% das caixas são padronizadas e 20% fora do padrão. Observa-se que a quantidade de caixas que inicialmente passa pela

esteira 1, atuador e coletor são quase os mesmos ao decorrer do tempo de relógio, e ao final do teste os números de caixas se tornam iguais. Isso porque todas as caixas, independente do seu tipo, passam por essas máquinas, comprovando que o supervisor está correto sob essas máquinas. Já a esteira 2 e o robô tem uma quantidade um pouco menor, pois apenas 80% passam por essas máquinas. A esteira 3 é a que menos transporta caixas, pois apenas 20% do total movimentam por ela, fazendo com que o supervisor seja menos utilizado nessa máquina.

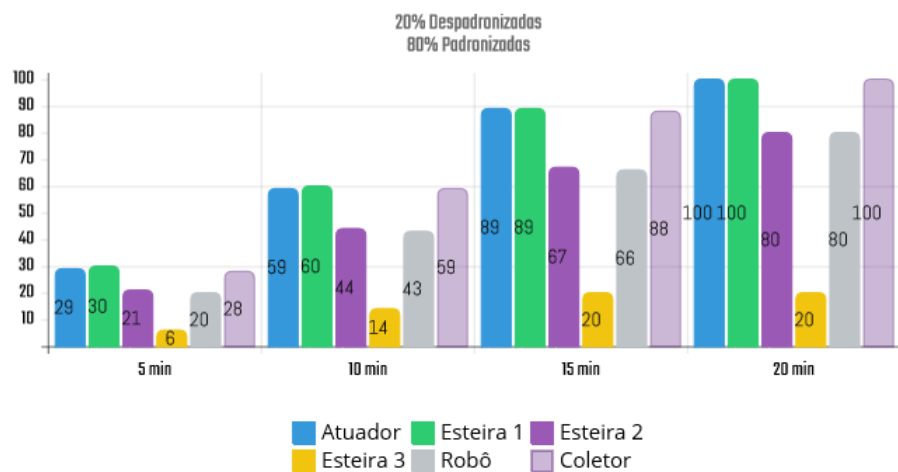


Figura 31 – Teste 2

As Figuras 32, 33, 34, 35, 36, 37 e 38 ilustram graficamente a variação dos parâmetros de simulação por porcentagem.

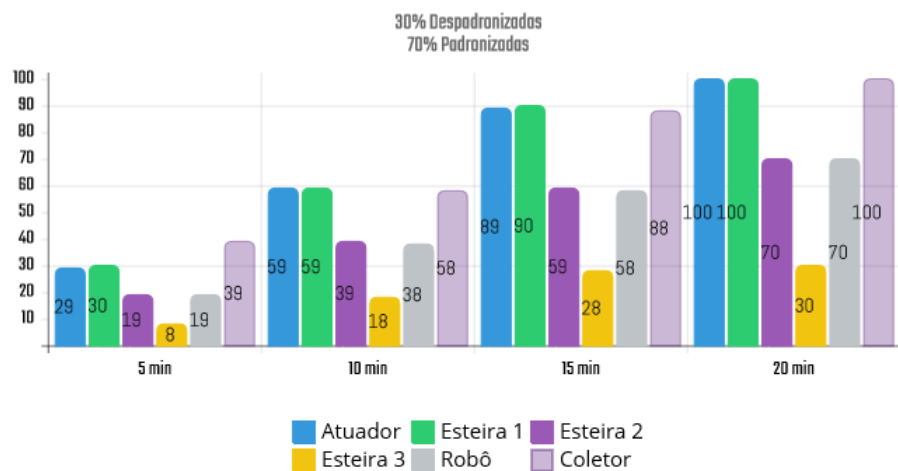


Figura 32 – Teste 3

A Figura 32 refere-se ao teste 3 que compreende que 70% das caixas são padronizadas

e 30% fora do padrão. A quantidade de caixas que passam pela esteira 1, atuador e coletor, no final da simulação, nunca vão mudar caso o supervisor esteja agindo corretamente, pois todas as caixas devem passar por essas máquinas, indiferente do seu tipo padronizada ou não padronizada.

A demais máquinas, esteira 2 e o robô tem uma quantidade um pouco menor, pois apenas 70% passam por essas máquinas, onde há o transporte de caixas padronizadas. A esteira 3 transportou menos caixas, pois apenas 30% do total movimentam por ela, que são as caixas fora do padrão.

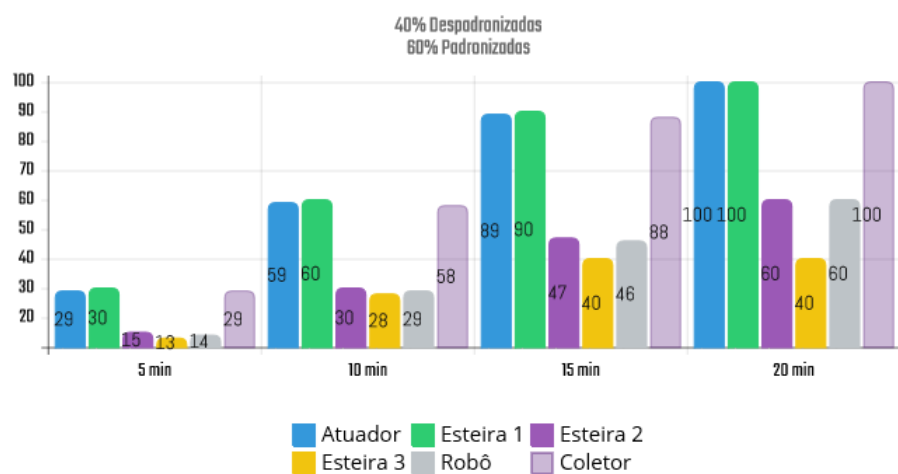


Figura 33 – Teste 4

A Figura 33 refere-se ao teste 4 que compreende que 60% das caixas são padronizadas e 40% fora do padrão. Novamente observa-se que a esteira 1, atuador e coletor que seguem bem semelhantes até o final da simulação, como foi comentado anteriormente.

Neste teste 4 observa-se uma pequena semelhança entre a esteira 2, esteira 3 e robô ao longo da simulação, isso foi possível pois 60% das caixas foram transportadas pela esteira 2 e robô, e 40% pela esteira 3.



Figura 34 – Teste 5

A Figura 34 refere-se ao teste 5 que compreende que 50% das caixas são padronizadas e 50% fora do padrão. Neste teste 4 observa-se uma semelhança maior entre a esteira 2, esteira 3 e robô ao longo da simulação, isso foi possível pois 50% das caixas passaram pela esteira 2 e robô, e 50% pela esteira 3 o que ocasionou ao final da simulação um número igual de caixas para essas máquinas.



Figura 35 – Teste 6

A Figura 35 refere-se ao teste 6 que compreende que 40% das caixas são padronizadas e 60% fora do padrão. Neste teste 4 observa-se uma pequena semelhança entre a esteira 2, esteira 3 e robô ao longo da simulação, isso foi possível pois 40% das caixas foram transportadas pela esteira 2 e robô, e 60% pela esteira 3.



Figura 36 – Teste 7

A Figura 36 refere-se ao teste 7 que compreende que 30% das caixas são padronizadas e 70% fora do padrão. A quantidade de caixas que passam pela esteira 1, atuador e coletor, no final da simulação, mesmo com mais caixas fora do padrão, nunca vão mudar caso o supervisor esteja agindo corretamente.

As máquinas, esteira 2 e o robô têm uma quantidade muito menor, pois apenas 30% passam por essas máquinas, onde há o transporte de caixas padronizadas. A esteira 3 transportou mais caixas, 70% do total movimentam por ela, que são as caixas fora do padrão, confirmando assim o uso correto dos supervisores.



Figura 37 – Teste 8

A Figura 37 refere-se ao teste 8 que compreende que 20% das caixas são padronizadas e 80% fora do padrão. Atuador, esteira 1 e coletor possuem o mesmo valor no final da

simulação.

A esteira 2 e o robô possuem uma quantidade de caixas no final da simulação bem menor, pois apenas 20% passam por essas máquinas, onde há o transporte de caixas padronizadas. A esteira 3 transportou mais caixas, 80% do total movimentam por ela, que são as caixas fora do padrão, confirmando assim o uso correto dos supervisores.

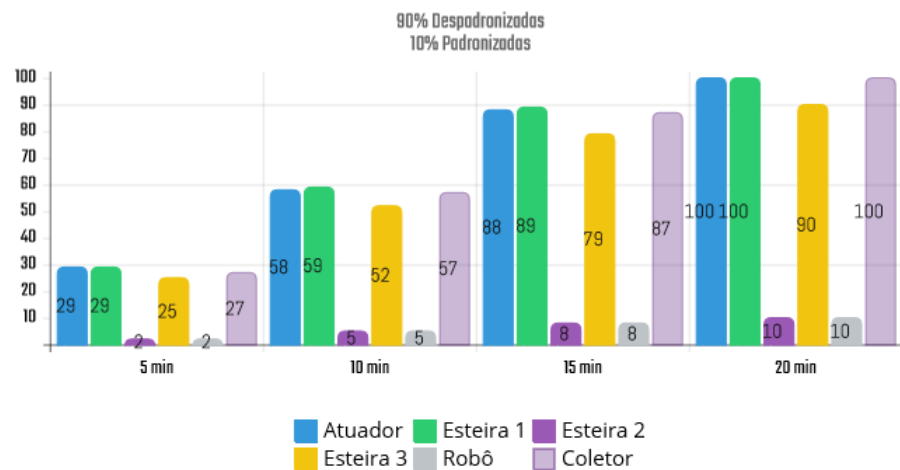


Figura 38 – Teste 9

A Figura 38 refere-se ao último teste realizado, teste 9 que compreende que 10% das caixas são padronizadas e 90% fora do padrão. Observa-se que mesmo com 90% das caixas fora do padrão, a quantidade de caixas que passam pela esteira 1, atuador e coletor no final da simulação é de 100 unidades, confirmando novamente que o supervisor está correto.

A esteira 2 e o robô apresentaram valores bem menor em relação ao do primeiro teste, pois apenas 10% passam por essas máquinas, onde há o transporte de caixas padronizadas, totalizando apenas 10 caixas em cada máquina. A esteira 3 transportou mais caixas, 90% do total movimentam por ela, que são as caixas fora do padrão, ou seja, ao final da simulação teve um resultado de 90 unidades.

A partir da análise dos resultados apresentados nas figuras, verifica-se ao final de cada simulação que a quantidade de caixas padronizadas que passam pela Esteira 2 e pela Esteira 3 são exatamente a quantidade da porcentagem definida em cada simulação.

Observa-se que o atuador, esteira 1 e no coletor, que são as máquinas que recebem os dois tipos de caixa, ao final da simulação, a quantidade de caixas em todos os testes é de 100 unidades, o mesmo número de entrada de caixas. As esteiras 2 e 3 obteve-se variações no número de caixas em relação ao tempo de relógio, porém ao final dos testes, a quantidade de caixas que saíram da esteira 3, que são caixas despadroneadas, somadas

as caixas da esteira 2, caixas padronizadas, resulta no total de 100 unidades de caixas, o mesmo número da entrada.

É possível definir que os supervisores funcionam corretamente, e que independente de tempo, a quantidade padronizada e não padronizada na saída do sistema é a mesma da entrada, sem erros, pois a modelagem está correta.

Ao final de todas as simulações realizou-se uma comparação das máquinas a cada porcentagem diferente, assim como nas Tabelas 10 e 11. A tabela foi dividida em duas para melhorar a visualização.

Tabela 10 – (a)Caixas por simulação

	Teste 1	Teste 2	Teste 3	Teste 4	Teste 5
Atuador	100	100	100	100	100
Esteira 1	100	100	100	100	100
Esteira 2	90	80	70	60	50
Esteira 3	10	20	30	40	50
Coletor	100	100	100	100	100
Robô	90	80	70	60	50

Tabela 11 – (b)Caixas por simulação

	Teste 6	Teste 7	Teste 8	Teste 9
Atuador	100	100	100	100
Esteira 1	100	100	100	100
Esteira 2	40	30	20	10
Esteira 3	60	70	80	90
Coletor	100	100	100	100
Robô	40	30	20	10

Observa-se nas Tabelas 10 e 11 o resultado final dos testes e o valor de caixas que passaram pelas máquinas. O atuador, esteira 1 e coletor foram as máquinas mais acionadas pela supervisor, pois, independente do tipo de caixa, ao detectar o evento nos *buffer's*, o supervisor habilita a máquina, justificando a quantidade de 100 caixas que passaram por essas máquinas.

Já a esteira 2, esteira 3 e o robô, variam de acordo com a porcentagem de entrada de caixas padronizadas e despadronizadas. Pois a esteira 2, só é acionada pelo supervisor ao receber o evento pelo *buffer* 4, que é por onde as caixas padronizadas trafegam. Já na esteira 3, o supervisor só habilita a máquina caso receba o evento do *buffer* 5, local onde as caixas fora do padrão trafegam. Dessa maneira, ao alterar porcentagem de caixas padronizadas e despadronizadas, irá alterar a quantidade que trafegam nessas máquinas, e o resultado final será diferente dos outros testes.

Dessa forma pode-se analisar melhor a mudança em números da quantidade de caixas dependendo do tipo da simulação e que o supervisor agiu corretamente em todas elas, não deixando que houvesse bloqueios, nem estados proibidos o que garantiu a validação da modelagem.

5 Conclusão

Conforme apresentado no capítulo 4, a teoria do controle supervísório mostrou-se um método bastante eficiente para resolução de problemas de explosão de estados, conseguindo encontrar resultados que outros métodos não conseguem ou a solução se torna dispendiosa.

O objetivo deste trabalho concentrou-se na proposta, avaliação do desempenho e validação da modelagem do sistema de armazenagem, utilizando controle supervísório modular local, e fazendo uso do simulador *FlexSim*, visando verificar os eventos ocorridos, as máquinas e o supervisor atuando na planta, habilitando ou desabilitando eventos.

Após a modelagem da planta no *FlexSim*, buscou-se ajustar detalhes, como a quantidade de caixas que seriam criadas, definir dois tipos de caixas, padronizadas e despadronizadas, a porcentagem de cada tipo de caixa que seria criada e a capturação de dados das máquinas, eventos e supervisor para que pudesse ser gerado os resultados.

Para a geração dos resultados, foram obtidos os supervisores utilizando o software *TCT* onde é realizada a síntese desses supervisores considerando apenas subsistemas que contém eventos iguais da especificação, a partir daí o software gerou autômatos com seus estados e suas transições. Com isso foram implementados esses supervisores na planta que foi modelada no *FlexSim*, habilitando ou desabilitando eventos das máquinas evitando que ocorressem erros na planta.

Depois da configuração correta da simulação com base na planta e nos seus supervisores, foram realizados diversos testes, com métricas diversas para poder comparar entre si, como a taxa de processamento das máquinas durante o decorrer do tempo de relógio, e também a taxa da porcentagem de caixas padronizadas e despadronizadas.

Verificou-se ainda com os testes, que independente do tempo a quantidade padronizada e não padronizada na saída do sistema é a mesma da entrada, sem erros, isso comprova que a modelagem do sistema está correta.

Os resultados demonstram que a utilização da técnica de controle modular de fato é bastante vantajosa, uma vez que evita o aumento demasiado do número de estados e eventos, quando comparado a outras abordagens. Tal característica permite a utilização de menos recursos computacionais para processamento do controle supervísório, além de facilitar a manutenção do sistema como um todo, visto que as mudanças são pontuais, isto é, em supervisores específicos.

O estudo de caso considerado neste trabalho permitiu além da verificação da aplicação da teoria em um problema real, a obtenção de um modelo viável de implantação

em um sistema real. Ademais, foi possível comprovar a aplicação prática da técnica através da simulação do controle da planta no *FlexSim*.

Referências

- BANKS, J. *Discrete-Event System Simulation*. 50. ed. [S.l.]: Pearson, 2010.
- BOTTER, R. *Processos Estocásticos – Introdução ao Simul8*. São Paulo, 2016.
- BOUZADA, M. Simulação versus métodos analíticos: uma ferramenta didática na forma de discussão teórica. *Revista Eletrônica de Administração*, v. 22, n. 12, p. 84–95, 5 2013.
- BOWERSOX, C. *Gestão Logística da Cadeia de Suprimentos*. 4. ed. [S.l.]: Bookman, 2014.
- BROOKSHEAR, G. *Ciência da Computação: Uma visão abrangente*. 11. ed. [S.l.]: Bookman, 2013.
- BRYNJOLFSSON, E. *A segunda era das máquinas*. 1. ed. [S.l.]: Alta Books, 2015.
- CHWIF, L. *Introdução ao software de simulação Simul8*. São Caetano do Sul, 2006.
- CUNHA, A. *Controle Supervisório de SEDs*. Rio de Janeiro, 2003.
- CURY, J. Teoria de controle supervisório de sistemas a eventos discretos. 2001.
- GONCALVES, P. *Administração de Materiais*. 3. ed. [S.l.]: Elsevier, 2007.
- GUERRA, C. Sistemas de armazenagem. 2006.
- HARA, H. Introdução a teoria da computação. 2017.
- KAWAMURA, C. *Contribuições à análise de desempenho de células de manufatura baseada na teoria de controle supervisório*. Dissertação (Mestrado) — Universidade Tecnológica Federal do Paraná, Cornélio Procópio, Brasil, 6 2012.
- LAW, A. *Simulation Modeling and Analysis*. 4. ed. [S.l.]: McGraw- Hill, 2007.
- MAIA, F. *Administração de Materiais*. Rio Grande do Sul, 2013.
- MAROLLA, A. Utilização do software arena para análise de um sistema de atendimento comercial automotivo 2017. *Tekhne e Logos*, v. 1, n. 2, 9 2017.
- MELLO, F. *Implementação Distribuída de Controle Supervisório para sistemas a eventos discretos*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, Florianópolis, Brasil, 2 2006.
- MENEZES, P. *Linguagens formais e autômatos*. 6. ed. [S.l.]: Bookman, 2011.
- MOREIRA, N. *Introdução à simulação com o Simul8*. Lisboa, Portugal, 2003.
- MOTT, A. O que são sistemas supervisórios? 2017.
- PAREDE J; GOMES, L. *Eletrônica - Automação industrial*. São Paulo, 2011.
- PARKER, D. *Mastering Data Visualization with Microsoft Visio Professional 2016*. 1. ed. [S.l.]: Packt Publishing, 2016.

- PENA, P. *Notas de Aula – Controle Supervisório. Modelagem e Controle de Sistemas Dinâmicos a Eventos Discretos*. Minas Gerais, 2013.
- PEREIRA, D. *Modelagem e Simulação Computacional Numérica do Sistema de Trânsito do Anel Viário do Campus da UFJF*. Juiz de Fora, 2006.
- RAMOS, M. *Linguagens Formais e Autômatos*. Petrolina, 2008.
- REISER, C. *O Ambiente GRAIL para Controle Supervisório de Sistemas a Eventos Discretos: Reestruturação e Implementação de Novos Algoritmos..* Florianópolis, 2005.
- SANTOS, J. Simulação de sistemas a eventos discretos sob o controle supervisório modular local. *Simpósio Brasileiro de Automação Inteligente*, n. 12, p. 1376–1381, 3 2013.
- SIPSER, M. *Introdução à teoria da computação*. 2. ed. [S.l.]: Cengage Learning, 2007.
- TREBESQUIM, A. *Aplicação de um modelo computacional na simulação do processo de pintura do produto de maior giro de uma empresa moveleira*. Dissertação (Mestrado) — Universidade Tecnológica Federal do Paraná, Medianeira, Brasil, 2013.
- VILLANUEVA, J. *Teoria de Linguagens Formais e Autômatos*. João Pessoa, 2016.
- WELTER R; POSTAL, A. C. J. *Módulo Resolvedor de Autômatos Finitos Para um Ambiente de Apoio e Ensino de Teoria da Computação Utilizando Algoritmos Genéticos*. Dissertação (Mestrado) — Universidade Estadual do Oeste do Paraná, Cascavel, 9 2006.
- WONHAM. *Notes on Supervisory Control of Discrete Event Systems*. 2017. Disponível em: <<https://www.control.utoronto.ca/~wonham/Research.html>>.

Apêndices

APÊNDICE A – IMAGENS DOS AUTÔMATOS DAS PLANTAS

A.1 Composição das Plantas

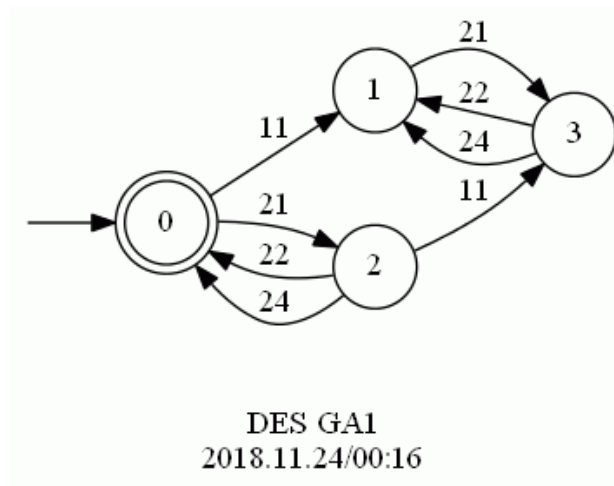


Figura 39 – GA1

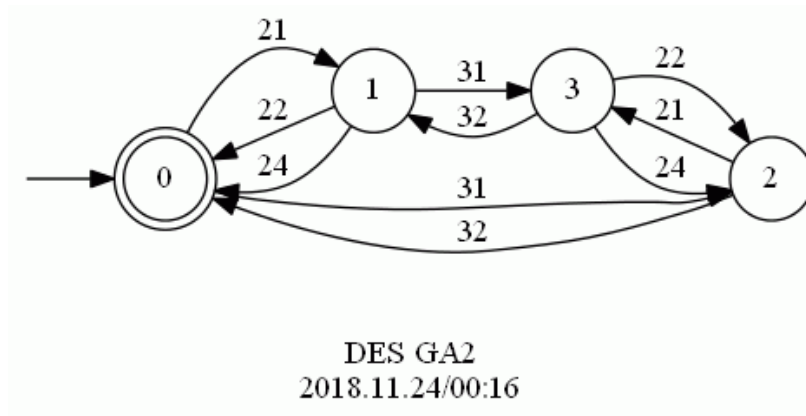
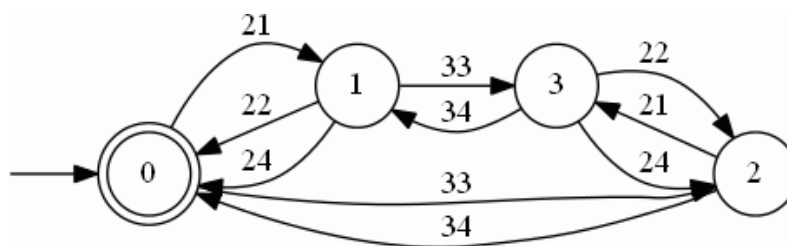
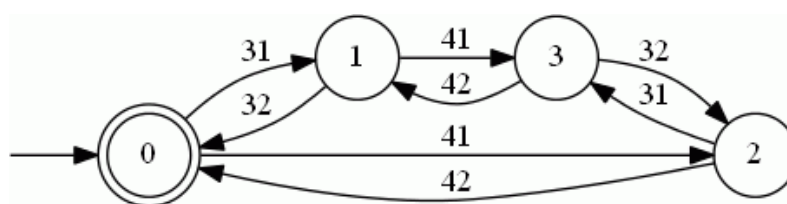


Figura 40 – GA2



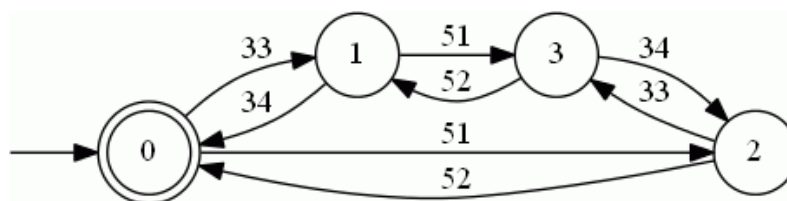
DES GB
2018.11.24/00:17

Figura 41 – GAB



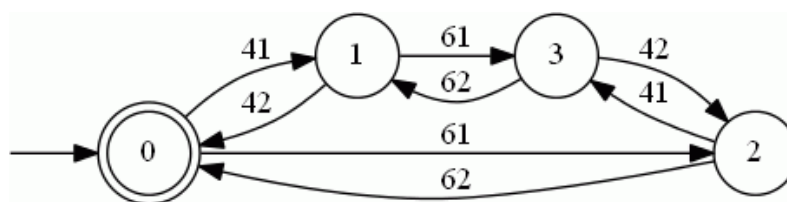
DES GC1
2018.11.24/00:17

Figura 42 – GC1



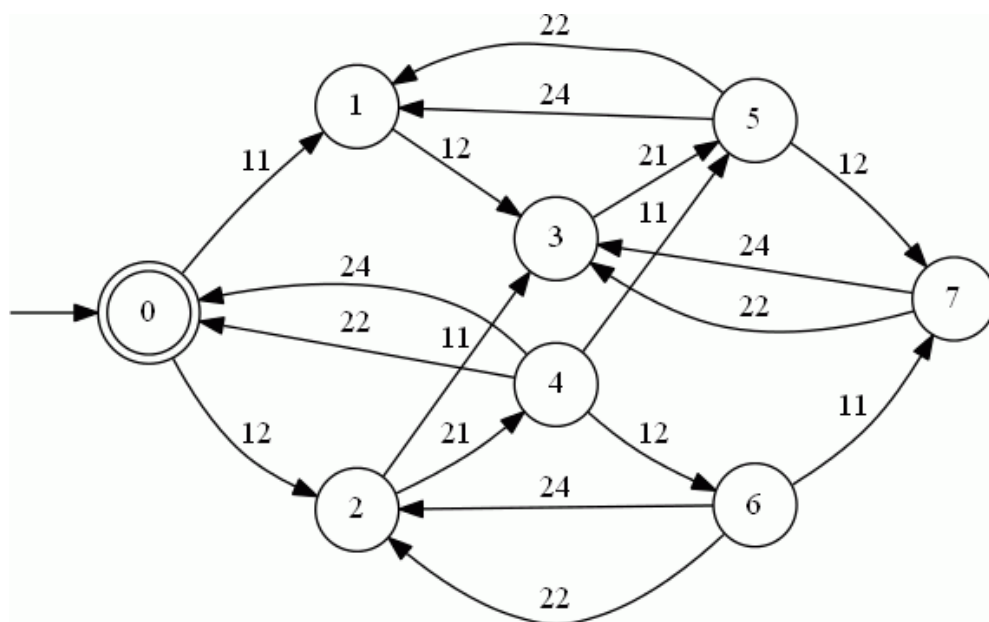
DES GC2
2018.11.24/00:17

Figura 43 – GC2



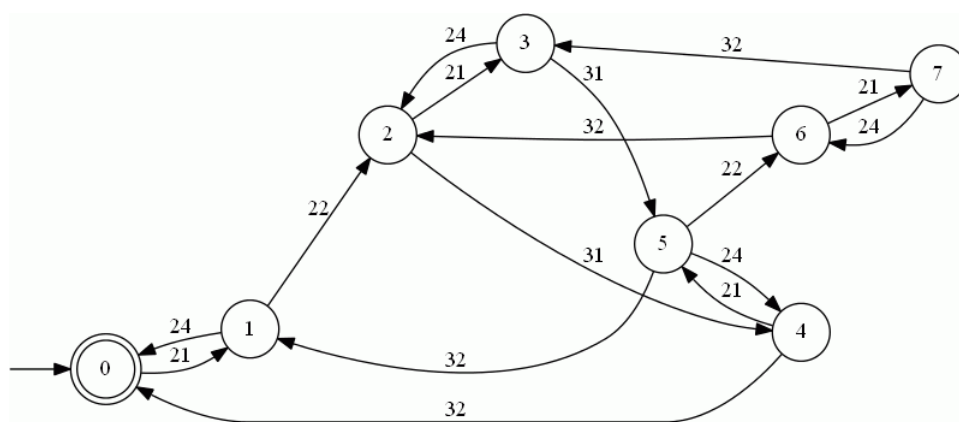
DES GD
2018.11.24/00:17

Figura 44 – GD



DES KA1
2018.11.24/00:17

Figura 45 – KA1



DES KA2
2018.11.24/00:17

Figura 46 – KA2

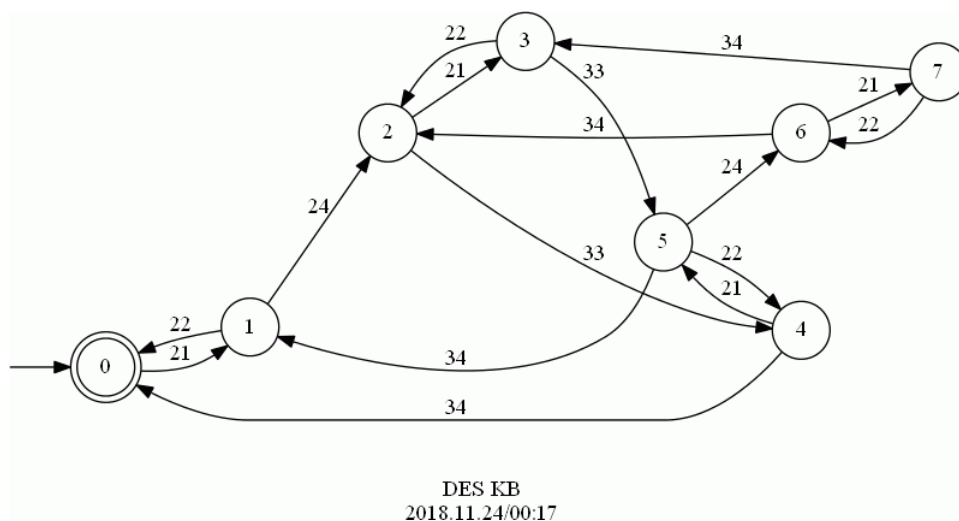


Figura 47 – KB

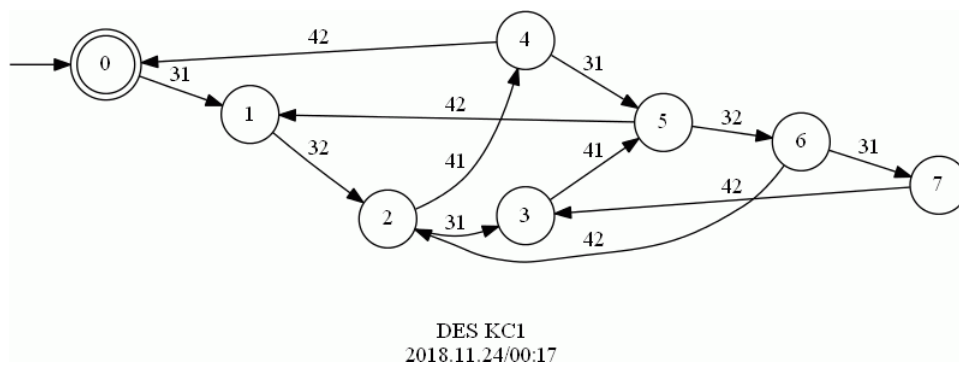


Figura 48 – KC1

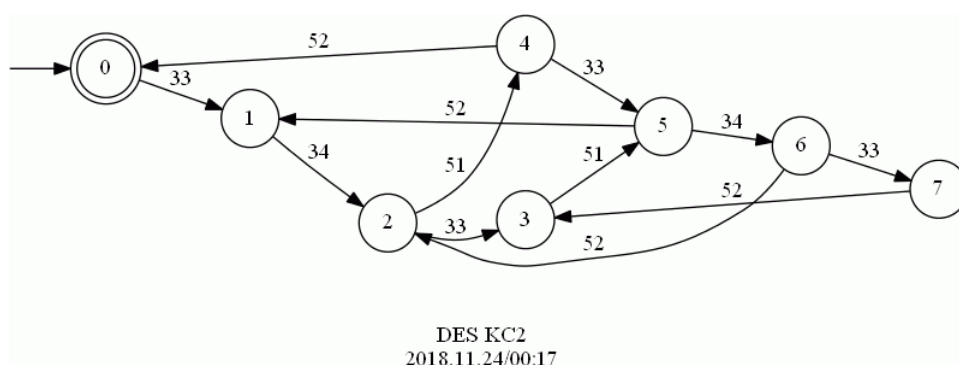


Figura 49 – KC2

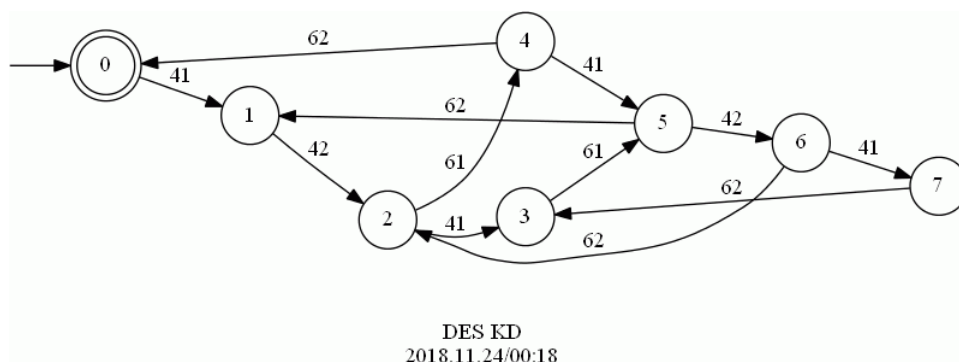


Figura 50 – KD