

**MINISTÉRIO DA DEFESA
EXÉRCITO BRASILEIRO
DEPARTAMENTO DE CIÊNCIA E TECNOLOGIA
INSTITUTO MILITAR DE ENGENHARIA
CURSO DE GRADUAÇÃO EM ENGENHARIA ELETRÔNICA**

SÉRGIO GABRIEL DOS SANTOS DIAS

**ENGENHARIA REVERSA: MODELAGEM AUTOMÁTICA DE UMA
PLATAFORMA DE MANUFATURA INTEGRADA POR COMPUTADOR**

**RIO DE JANEIRO
2021**

SÉRGIO GABRIEL DOS SANTOS DIAS

ENGENHARIA REVERSA: MODELAGEM AUTOMÁTICA DE UMA
PLATAFORMA DE MANUFATURA INTEGRADA POR COMPUTADOR

Projeto de Final de Curso apresentado ao Curso de Graduação em Engenharia Eletrônica do Instituto Militar de Engenharia, como requisito parcial para a obtenção do título de Bacharel em Engenharia Eletrônica.

Orientador(es): Antonio Eduardo Carrilho da Cunha, Dr.
Eng

Rio de Janeiro

2021

©2021

INSTITUTO MILITAR DE ENGENHARIA

Praça General Tibúrcio, 80 – Praia Vermelha

Rio de Janeiro – RJ CEP: 22290-270

Este exemplar é de propriedade do Instituto Militar de Engenharia, que poderá incluí-lo em base de dados, armazenar em computador, microfilmar ou adotar qualquer forma de arquivamento.

É permitida a menção, reprodução parcial ou integral e a transmissão entre bibliotecas deste trabalho, sem modificação de seu texto, em qualquer meio que esteja ou venha a ser fixado, para pesquisa acadêmica, comentários e citações, desde que sem finalidade comercial e que seja feita a referência bibliográfica completa.

Os conceitos expressos neste trabalho são de responsabilidade do autor e do orientador.

Santos Dias, Sérgio Gabriel dos.

Engenharia Reversa: Modelagem Automática de uma Plataforma de Manufatura Integrada por Computador / Sérgio Gabriel dos Santos Dias. – Rio de Janeiro, 2021.

35 f.

Orientador: Antonio Eduardo Carrilho da Cunha.

Projeto de Final de Curso (graduação) – Instituto Militar de Engenharia, Engenharia Eletrônica, 2021.

1. Mecatrônica. 2. Engenharia de Sistemas. 3. Engenharia Reversa. 4. FS100. 5. ScorBase. i. da Cunha, Antonio Eduardo Carrilho (orient.) ii. Título

SÉRGIO GABRIEL DOS SANTOS DIAS

Engenharia Reversa: Modelagem Automática de uma Plataforma de Manufatura Integrada por Computador

Projeto de Final de Curso apresentado ao Curso de Graduação em Engenharia Eletrônica do Instituto Militar de Engenharia, como requisito parcial para a obtenção do título de Bacharel em Engenharia Eletrônica.

Orientador: Antonio Eduardo Carrilho da Cunha.

Aprovado em Rio de Janeiro, 21 de Outubro de 2021, pela seguinte banca examinadora:

Cel. Antonio Eduardo Carrilho da Cunha - Dr. Eng

Maj. Raquel Stella da Silva de Aguiar - Dr. ISAE

Maj. Erick Menezes Moreira - Dr. IME

Sr. Igor Cohen Calixto - MsC. IME

Rio de Janeiro
2021

Ao Instituto Militar de Engenharia, alicerce da nossa formação e conhecimento.

AGRADECIMENTOS

Os agradecimentos principais são direcionados à minha família e amigos, que juntamente com todos os meus professores me deram o suporte necessário durante esse projeto e todos os meus anos de estudos no IME.

Um agradecimento especial ao orientador Cel Antonio Eduardo Carrilho da Cunha, Maj Raquel Stella da Silva de Aguiar e Maj Erick Menezes Moreira e ao aluno de doutorado Igor Calixto Cohen, por todo o auxílio e disponibilidade.

“Todos podem ver as táticas de minhas conquistas, mas ninguém consegue discernir a estratégia que gerou as vitórias.”

Sun Tzu

RESUMO

Com o intuito de modernizar o ensino do Instituto Militar de Engenharia, foi adquirida uma plataforma de Manufatura Integrada por Computador, do inglês CIM - Computer Integrated Manufacture -, que possui a capacidade produtiva de uma planta de pequeno porte, a MecatrIME. Essa plataforma foi projetada para operar com grande produtividade de forma automatizada e integrada, trabalhando dessa forma dentro dos preceitos da Indústria 4.0.

O presente projeto visa criar uma ferramenta, que juntamente com as pesquisas de mestrado e doutorado relativas à plataforma, poderá aumentar ainda mais a sua produtividade por meio da utilização de autômatos. A principal função da ferramenta é automatizar a extração dos autômatos a partir de arquivos-texto da linguagem ScorBase, podendo ser utilizada como base para a criação de extratores automáticos semelhantes em outras linguagens e também de arquivos de código a partir de autômatos.

Para que o desenvolvimento da ferramenta possa acontecer de forma contínua, incluindo, mas não se restringindo somente a novas funcionalidades e melhorias no código, em conjunto é entregue um manual com os conhecimentos necessários e tutoriais para o desenvolvimento, por meio da plataforma GitHub.

ABSTRACT

In order to modernize the education of the Military Institute of Engineering (IME), a Computer Integrated Manufacture (CIM) platform was acquired, which has the production capacity of a small plant, the MecatrIME. This platform was designed to operate with high productivity in an automated and integrated way, working within the precepts of Industry 4.0.

The present project aims to create a tool, which together with the Master's and Doctoral research on the platform, will be able to further increase its productivity through the use of automata. The main function of the tool is to automate the extraction of automata from text files of the ScorBase language, and it can be used as a basis for the creation of similar automatic extractors in other languages and also of code files from automata.

Therefore, the development of the instrument can happen continuously, including but not restricted to new features and improvements in the code, a manual with the necessary knowledge and tutorials for the development is provided via the GitHub platform.

LISTA DE ILUSTRAÇÕES

Figura 1 – Plataforma MecatrIME	13
Figura 2 – Organização da Planta MecatrIME Identificada por Estações	13
Figura 3 – Diagrama do Funcionamento da Ferramenta.	17
Figura 4 – Estrutura Analítica do Projeto.	17
Figura 5 – Braços Robóticos Atuadores da Plataforma MecatrIME	20
Figura 6 – Exemplo de um Autômato Finito Determinístico	21
Figura 7 – Diagrama de Estados do Autômato da Subrotina GET023 do ScorBase.	25
Figura 8 – Diagrama de Estados do Autômato da Variável de Memória LATHE_LOAD.	26
Figura 9 – Autômato da subrotina GET023 do ScorBase gerado pela ferramenta.	27
Figura 10 – Diagrama de Blocos da Planta MecatrIME	33
Figura 11 – Diagrama de Blocos Cibernético da Plataforma MecatrIME.	34
Figura 12 – Diagrama Ciberfísico da Plataforma	35

LISTA DE ABREVIATURAS E SIGLAS

CIM	<i>Computer Integrated Manufacture</i> , ou Manufatura Integrada por Computador
EAP	Estrutura Analítica do Projeto
CPS	<i>Cyber-physical System</i> , ou Sistema Ciberfísico
SED	Sistemas a Evento Discreto

SUMÁRIO

1	INTRODUÇÃO	12
1.1	CONTEXTUALIZAÇÃO	12
1.2	OBJETIVO	14
1.3	METODOLOGIA	14
1.4	GUIA A LEITURA	15
2	VISÃO GERAL DO PROBLEMA E ESTRUTURAÇÃO DO PROJETO	16
2.1	VISÃO GERAL DO PROJETO	16
2.2	ESTRUTURA DO PROJETO	16
3	BASE CONCEITUAL	19
3.1	ORGANIZAÇÃO DA PLATAFORMA MECATRIME	19
3.2	SCORBASE E FS100	19
3.3	AUTÔMATOS	20
3.4	GRAPHVIZ	21
3.5	PARSER	21
4	ENSAIOS	23
4.1	EXTRAÇÃO MANUAL DE AUTÔMATOS	23
4.2	VARIÁVEL DE MEMÓRIA	26
4.3	EXTRAÇÃO AUTOMÁTICA DOS AUTÔMATOS UTILIZANDO A FERRAMENTA	27
5	CONCLUSÃO	30
5.1	CONTRIBUIÇÕES	30
5.2	PONTOS NÃO ABORDADOS	30
	REFERÊNCIAS	32
	ANEXO A – ANEXOS	33

1 INTRODUÇÃO

Nesse capítulo será apresentada a idealização do projeto e o contexto no qual está inserido, além disso conta também com a metodologia do trabalho, os objetivos esperados finais e um breve guia a leitura.

1.1 Contextualização

A Quarta Revolução Industrial, também chamada de Indústria 4.0, engloba os conceitos de inteligência artificial, robótica, internet das coisas e computação em nuvem, com o objetivo de melhorar processos e aumentar de forma inteligente e automatizada com otimizações em tempo real. A capacidade de operação de forma inteligente das fábricas é obtida a partir da utilização dos Sistemas Ciberfísicos, também identificado como CPS, do inglês *Cyber-physical Systems*, sistemas que combinam componentes físicos inteligentes com programação embarcada conectados por meio de uma rede. De forma resumida, pode-se dizer que um Sistema Ciberfísico é aquele no qual existe uma relação de conexão entre a parte visual e a parte física, num contexto industrial.(1)

O laboratório de mecatrônica do IME por meio da plataforma MecatrIME possui a capacidade de operação de uma planta industrial de pequeno porte. O funcionamento da plataforma se dá a partir da integração de atuadores e sensores com um controle central e dividido em dois níveis de software, as rotinas do ScorBase e os jobs do controlador FS100, por meio de uma rede de comunicações.

A plataforma MecatrIME, com toda a sua capacidade de atuação, caracteriza-se como um CPS e dessa forma é destacada sua importância como ferramenta de estudo e pesquisa de funcionamento e técnicas de produção de plantas industriais de pequeno porte. Na Figura 1 é possível observar a organização da plataforma a partir da estação de gerenciamento central e supervisorio, da qual são enviados os comandos para execução em cada uma das estações de trabalho especializada, que operam de forma automática e autônoma para a criação de um produto no final da linha de produção.

A organização da plataforma MecatrIME encontra-se detalhada na Figura 2, identificando cada uma das 6 estações de trabalho. Com base na figura, P1 é a estação de armazenamento, P2 a estação de torneamento, P3 a estação de usinagem, P4 a estação de soldagem, P5 a estação de metrologia, P6 a estação de montagem e inspeção visual, P7 a estação de gravação a laser. A estação P8 corresponde ao controle da esteira de transporte e o P9 corresponde ao *Manager* que faz o gerenciamento de todo o sistema. Dessas estações, P2 até P7 representam as 6 estações de trabalho, enquanto P1 é uma



Figura 1 – Plataforma MecatrIME

estação adjacente para armazenamento dos produtos e P8 e P9 estações de controle do sistema.

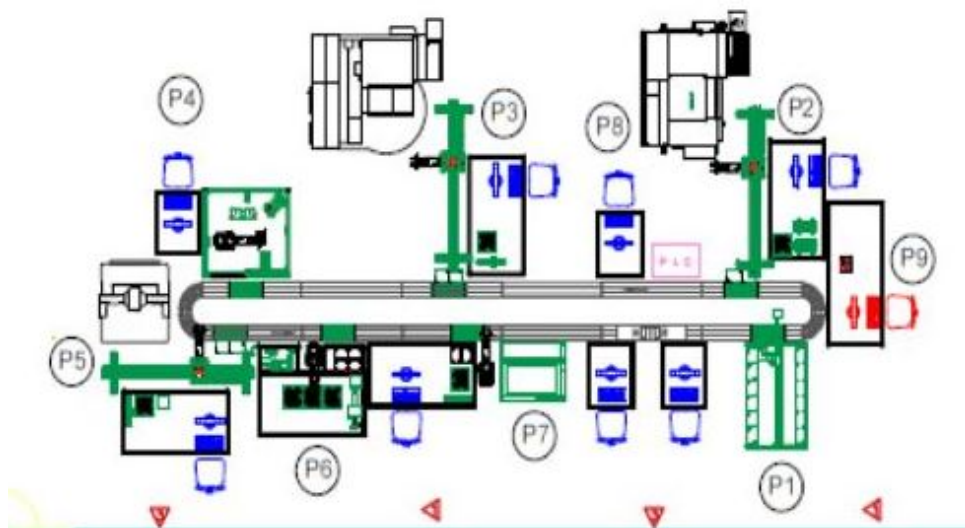


Figura 2 – Organização da Planta MecatrIME Identificada por Estações

A complexidade do sistema exige então o conhecimento de conceitos de Engenharia de Sistemas, que representam a organização e a relação entre os componentes físicos e programas computacionais, bem como a relação entre as estações e o gerente.

A relação entre os atuadores (como a esteira transportadora, os braços robóticos, o robô cartesiano e o torno por exemplo) e as estações virtuais com seus códigos de operação é descrita por meio de um diagrama de blocos conhecido como diagrama ciberfísico. A Figura 11, presente no Anexo A, apresenta como exemplo a representação de um diagrama cibernético, um dos componentes do diagrama ciberfísico para a plataforma MecatrIME.

Cada uma das estações de trabalho está interligada para uma produção completa e integrada. O *Manager* envia as instruções que devem ser executadas em cada uma das

estações de trabalho por meio de rotinas na linguagem ScorBase, que por sua vez faz uso dos jobs do controlador FS100 para a utilização dos atuadores, sensores e equipamentos envolvidos.

O conhecimento total do sistema requer a compreensão das linguagens de programação com o intuito de entender e modificar o código para a exploração completa das funcionalidades da MecatrIME, conhecimento até então não dominado devido à falta de informação fornecida por parte da empresa fornecedora.

Dessa forma, este trabalho apresenta uma proposta de extração automática de modelos na forma de autômatos a partir das rotinas de controle numa das camadas de software da plataforma MecatrIME. A importância dessa ferramenta está na facilidade do fornecimento de informações para pesquisas de mestrado e doutorado, que utilizando-se de conceitos da Teoria de Controle de Sistema Supervisório para Sistemas a Eventos Discretos(2) podem aumentar a produtividade da plataforma como um todo.

A ferramenta visa auxiliar no processo decisório de validação de algoritmos por meio de autômatos representativos das rotinas de código. Essa capacidade de suporte à decisão pode ser expandida futuramente para a conversão no sentido contrário, tendo como entrada um autômato e criando o arquivo-texto da linguagem selecionada, permitindo assim o teste de novas funcionalidades a partir do funcionamento esperado dos autômatos.

1.2 Objetivo

Com base nos requisitos levantados de ferramenta de entrada e saída de autômatos, o escopo do projeto foi:

- Criação dos autômatos no formato .dot do *GraphViz* das rotinas em ScorBase do torno;
- Criação dos diagramas de estado relativo aos autômatos das rotinas em ScorBase;
- Desenvolvimento de uma ferramenta que automatize a extração de autômatos das rotinas de controle no ScorBase.

1.3 Metodologia

Primeiramente foi definido o escopo do programa que seria utilizado como alvo de pesquisa do projeto. Para isso foi delimitada uma das estações de trabalho com foco na sua operação, sendo escolhida a estação de torno. A decisão da utilização dessa estação foi feita com base na utilização de comandos abrangentes da linguagem ScorBase, tendo muitas interseções de instruções com as outras estações. Além disso, essa estação é objeto

de trabalho principal dos alunos de mestrado e doutorado do laboratório, que utilizarão a ferramenta ativamente, se adequando dessa forma às necessidades desses alunos.

Essa estação utiliza uma morça para posicionar um bloco do material a trabalhado e por meio de cortes conformá-lo como a peça que deve ser utilizada nos pontos seguintes da produção. Como se trata de um Sistema Ciberfísico, cada parte da estação de trabalho é associada a um conjunto de instruções do ScorBase, sendo necessário conhecer o funcionamento desde os braços mecânicos até o acionamento da abertura e fechamento da porta.

Inicialmente, por meio de diagramas de blocos e de estados e utilizando os conceitos de Engenharia de Sistemas, foram modelados os componentes tanto de hardware quanto de software. Com a identificação dos elementos de software foram construídos os modelos dos autômatos, nas rotinas de ScorBase da estação do torno.

Com a criação da ferramenta foi possível automatizar a criação dos autômatos no formato GraphViz convertidos a partir de arquivos-texto do ScorBase, testando-a então com outras rotinas existentes na estação do torno. Por meio de inspeção desses arquivos da saída foi possível validar o modelo e corrigi-lo para possíveis melhorias e erros encontrados.

1.4 Guia a Leitura

O presente texto está organizado seguindo a sequência: o Capítulo 1 é uma introdução com a contextualização do projeto, objetivos e metodologia; o Capítulo 2 apresenta uma visão geral do problema; O Capítulo 3 apresenta a base conceitual do projeto; o Capítulo 4 apresenta as ferramentas principais utilizadas no projeto; o Capítulo 5 apresenta a solução de tradução do ScorBase para máquina de estados; o Capítulo 6 apresenta os resultados dos ensaios com os scripts do escopo e por fim o Capítulo 7 aborda a conclusão do projeto, com as contribuições e possíveis próximos passos do projeto.

2 VISÃO GERAL DO PROBLEMA E ESTRUTURAÇÃO DO PROJETO

Esse capítulo apresentará uma visão geral do projeto, apresentando as ferramentas utilizadas e um esquemático do funcionamento. Também será apresentada a estrutura do projeto com os entregáveis principais e frentes de trabalho.

2.1 Visão geral do projeto

O projeto visa a extração automática de modelos formais na forma de autômatos a partir dos arquivo-texto das rotinas do ScorBase da plataforma MecatrIME. Essa extração tem como objetivo, primeiramente, a possibilidade de fornecer informações para basear o processo de verificação e validação do software entregue pelo fabricante da plataforma por meio da técnica de *Model Checking* ou Checagem de Modelos, da Ciência da Computação(3). A partir dos códigos fornecidos pela fabricante, utilizando a ferramenta, é possível extrair os autômatos para validação por meio dos trabalhos dos pesquisadores do laboratório.

Na fase seguinte, por meio das ferramentas da Ciência da Automação, como a Teoria de Controle de Sistema Supervisório para Sistemas a Eventos Discretos(2), com o intuito de verificar as mudanças para melhoria da capacidade operacional da plataforma por meio do aprimoramento das rotinas de controle lógico.

Para atingir esse objetivo foi idealizada uma ferramenta de extração de autômatos de forma automática, com estrutura apresentada na Figura 3. Dessa forma, utilizando-se dos conceitos de parser(4), fazendo a quebra do texto em palavras chave e identificando comandos a partir de um universo conhecido é possível realizar a tradução do código das rotinas de ScorBase para GraphViz.

2.2 Estrutura do Projeto

Com base na demanda e nos prazos, o projeto foi estruturado conforme a Estrutura Analítica do Projeto (EAP)(5) Figura 4.

Com a divisão de frentes do trabalho do projeto, foi possível organizar um cronograma conforme as atividades previstas. Na primeira fase do projeto o foco foi o entendimento do funcionamento do sistema, bem como as tecnologias envolvidas. Na segunda etapa, focou-se na análise dos arquivos-texto do ScorBase da estação de trabalho do torno e definição do escopo do código a ser trabalhado, juntamente com o desenvolvimento de autômatos de forma manual utilizando o *yEd*(6). A ferramenta *yEd* é um software

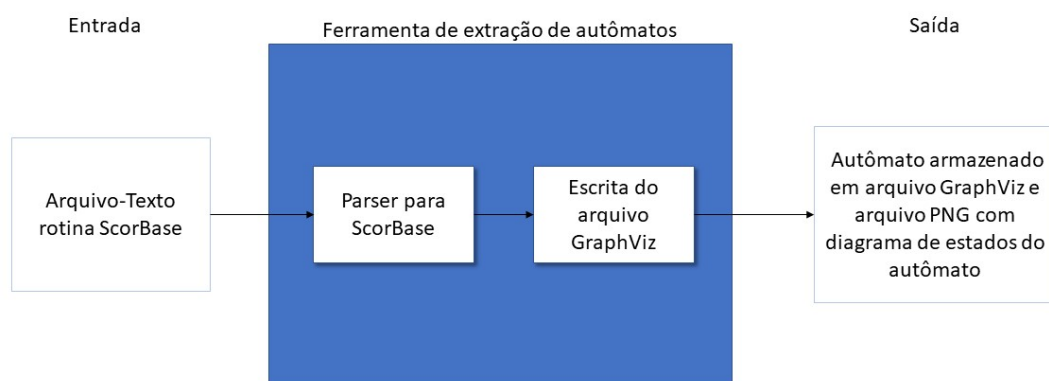


Figura 3 – Diagrama do Funcionamento da Ferramenta.

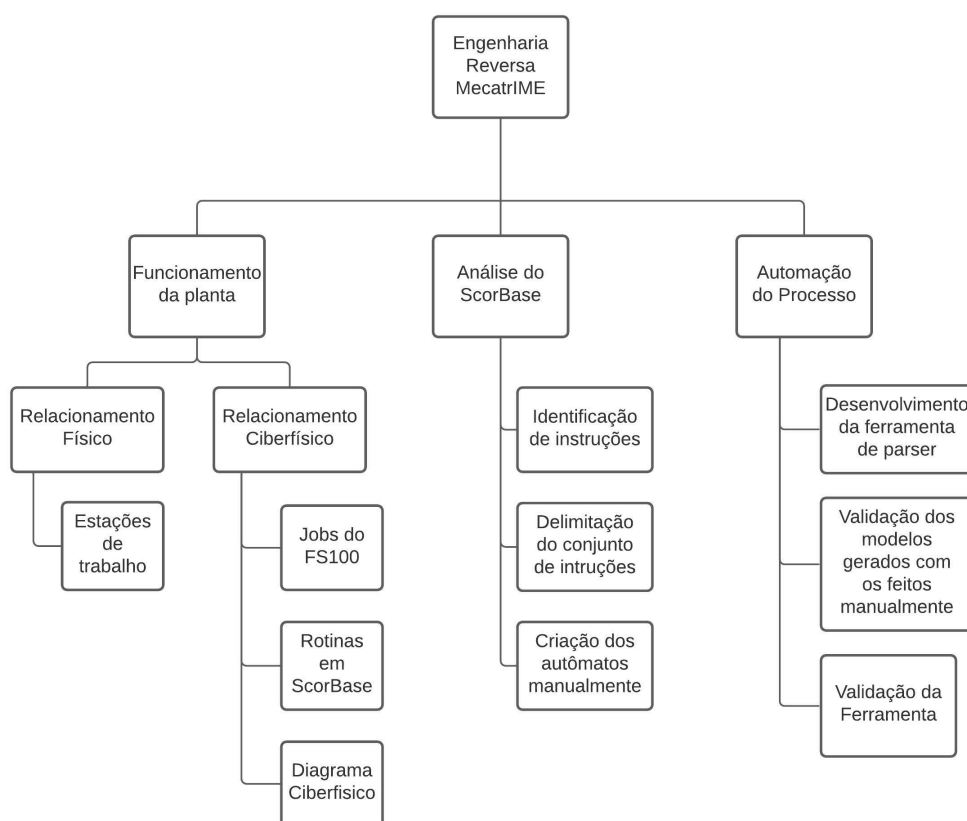


Figura 4 – Estrutura Analítica do Projeto.

aberto amplamente utilizado para o design de diagramas de forma rápida e organizada, de forma manual ou a partir da importação de arquivos no formato *.xls* do *Excel*.

Por fim, foi desenvolvida a ferramenta de extração de autômatos a partir de código de ScorBase armazenando em um arquivo GraphViz e também uma representação gráfica por meio de um diagrama de estados desse autômato para inspeção.

3 BASE CONCEITUAL

Nesse capítulo serão expostas as principais ferramentas utilizadas e envolvidas no projeto, juntamente com conceitos da Ciência da Computação e da Automação importantes para o entendimento e execução do projeto.

3.1 Organização da plataforma MecatrIME

A plataforma da MecatrIME, por ter a capacidade produtiva de uma planta industrial de pequeno porte apresenta uma grande complexidade e integração de partes físicas e mecânicas com níveis de software de controle, sendo assim um Sistema Ciberfísico. Os Sistemas Ciberfísicos representam um pilar importante da Indústria 4.0, sendo a ligação entre o mundo físico dos hardwares com os computacionais de software por meio de uma rede de comunicação(7). Isso ocorre na plataforma MecatrIME com as rotinas de ScorBase fazendo utilização dos jobs do controlador FS100 para leitura de sensores e movimentação de atuadores, como os braços robóticos da Figura 5.

Para a plataforma MecatrIME a organização em subsistemas de forma visual é organizada conforme a Figura 10, no Anexo A, apresentando os principais subsistemas juntamente com a melhor forma de realizar a representação visual de cada.

As conexões do sistema ciberfísico é a presente na Figura 11, no Anexo A, no qual são identificadas quais estações de trabalho da plataforma são afetadas por rotinas da linguagem ScorBase e como essas estações interagem com o *Manager* e entre si. O esquema de cores identifica as rotinas do ScorBase, em rosa, os jobs da FS100, em azul, e as partes mecânicas do sistema, em amarelo. Por meio do diagrama cyberfísico também fica evidente a relação entre o ScorBase e os jobs da FS100.

3.2 ScorBase e FS100

A linguagem utilizada em cada uma das estações, em nível mais alto, é o ScorBase da Intelitek(8). Essa linguagem é amplamente utilizada em aplicações de robótica. Juntamente com o FS100 compõem a parte computacional do sistema.

A nível de controlador, a plataforma utiliza o FS100(9) para controle da atuação no robô Motoman MH5 F da fabricante Yaskawa, que possui dentro de suas vantagens um software aberto, permitindo novas implementações e alterações. O controlador representa os jobs responsáveis pelas leituras dos sensores e funcionamento dos atuadores.



Figura 5 – Braços Robóticos Atuadores da Plataforma MecatrIME

3.3 Autômatos

A Teoria dos Autômatos tem como objeto de estudo aparelhos abstratos de computação, chamados de máquinas. A principal aplicação no escopo do projeto é relativa aos autômatos finitos, sendo aqueles representativos de sistemas que operam em um número finito de estados definidos. Além disso é importante o conhecimento dos conceitos de gramática, modelos de estruturas recursivas que podem ser usadas no processamento de informações, e expressões regulares, que são estruturas de informações, principalmente *strings* que identificam operações da máquina estudada.(10). A Figura 6 apresenta uma representação de um autômato com 2 estados e seus respectivos sinais de transição e sinal de entrada para início da máquina.

Uma aplicação importante de autômatos e na qual a ferramenta produzida nesse projeto auxiliará é na Teoria de Sistemas a Eventos Discretos(SED). Nos sistemas tratados

por esse modelo, o equacionamento por meio de equações diferenciais não é adequado, sendo a forma mais aceita de lidar com o sistema o Modelo RW, baseado em linguagem e autômatos(11). Nesse método a plataforma é dividida entre planta e supervisor, com a primeira sendo um modelo físico correspondente ao sistema com todas as ações que podem ser executadas sem um controle. Já o supervisor tem como função enviar os comandos de controle para que não ocorram erros de operação.

Com a separação da planta e do supervisor é possível a síntese de novos supervisores que se adequem à planta e melhorem seu desempenho. A ferramenta criada nesse projeto oferece a capacidade de visualização e extração dos autômatos dos novos códigos de supervisores criados, servindo de insumos para decisão da sua utilização.

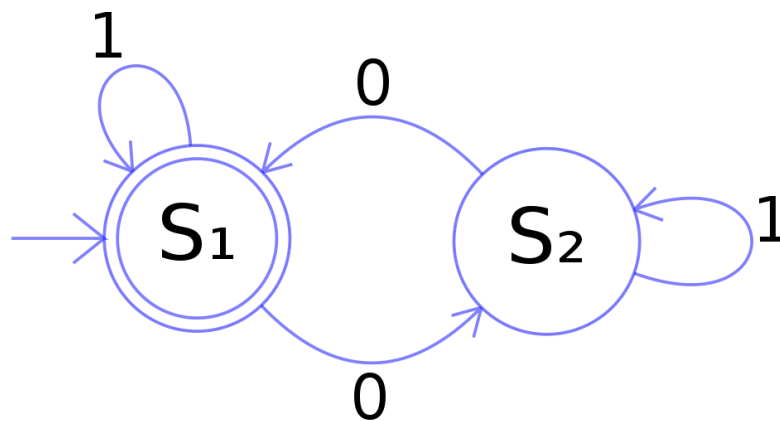


Figura 6 – Exemplo de um Autômato Finito Determinístico

3.4 GraphViz

Para a automatização da criação dos autômatos foi escolhido utilizar o GraphViz, uma ferramenta de construção de gráficos de código aberto, e utilizando a linguagem DOT(12). Essa linguagem utiliza-se de uma gramática abstrata para criação de diagramas de acordo com sua sintaxe.

3.5 Parser

Parser é uma parte principal dos compiladores, sendo responsável por fazer a quebra do texto em elementos que possam ser interpretados. Com base em tabelas da linguagem para qual o parser foi construído é possível identificar se a sintaxe está correta e apontar erros caso encontrado. Por meio da identificação da ordem da escrita com as instruções, sendo palavras reservadas e escritas nas tabelas da linguagem que são consultadas pelo

parser, e os parâmetros associados à instrução é possível identificar os casos de erro de utilização de funções ou instruções.(4)

4 ENSAIOS

Nesse capítulo serão descritos os ensaios realizados e os resultados obtidos na extração manual e automática utilizando a ferramenta dos autômatos, realizando em seguida uma comparação desses resultados.

4.1 Extração Manual de Autômatos

Na primeira fase de execução do projeto foram criadas as premissas do trabalho, optando pelo foco nas rotinas do ScorBase. Cada linha de uma rotina ScorBase foi identificada com uma possível operação, sendo dividida em casos especiais, como se segue:

- Chamada: Instruções de *Call* e *Run* não necessariamente resolvem sua instrução no estágio seguinte, necessitando ter um tratamento diferente;
- Comentários: Linhas do código que devem ser ignoradas na criação do autômato;
- Casos: Instrução de *If* fará com que o código tenha 2 estados possíveis dependendo da validação da condição;
- Espera: Instrução de *Wait* dos Jobs da FS100 podem resultar em *timeouts* encerrando a execução da subrotina.

As instruções foram, então, divididas em comandos de *START* e *FINISH*, e utilizadas como os labels das transições de estados. Dessa forma, analisando o código da subrotina do ScorBase *GET023* abaixo foi possível extrair de forma manual para a linguagem DOT do GraphViz, com as transições de estado e considerando os casos especiais:

```
Set Subroutine GET023
Print to Screen: GET FROM Gravity Feeder
Call Subroutine SCRIPT.GET_FROM_GFDR1
Print to Screen: P1,P2, PB1:  'SCRIPT.P1', 'SCRIPT.P2' , 'SCRIPT.PB1'
Print to Screen: P3,P4, P5:  'SCRIPT.P3', 'SCRIPT.P4' , 'SCRIPT.P5'
Copy FS100 Position SCRIPT.P1 to Position 1001
Copy FS100 Position SCRIPT.P2 to Position 1002
Copy FS100 Position SCRIPT.P3 to Position 1003
Copy FS100 Position SCRIPT.P4 to Position 1004
Go to Position SCRIPT.PB1 Speed 50 (%)
```



```

FS100 Start Job GT023
Wait FS100_DELAY_TIME (10ths of seconds)
FS100 Job Wait 60 (seconds)
Send Message $Start to MANAGER ID=TASK_ID
Return from Subroutine

```

```

digraph GET023 {

```

```

A -> B [label = "START Subroutine GET023"];
B -> C [label = "START Call Subroutine SCRIPT.GET_FROM_GFDR1"];
C -> D [label = "FINISH Call Subroutine SCRIPT.GET_FROM_GFDR1"];
D -> E [label = "START Copy FS100 Position SCRIPT.P1 to Position 1001"];
E -> F [label = "FINISH Copy FS100 Position SCRIPT.P1 to Position 1001"];
F -> G [label = "START Copy FS100 Position SCRIPT.P1 to Position 1002"];
G -> H [label = "FINISH Copy FS100 Position SCRIPT.P1 to Position 1002"];
H -> I [label = "START Copy FS100 Position SCRIPT.P1 to Position 1003"];
I -> J [label = "FINISH Copy FS100 Position SCRIPT.P1 to Position 1003"];
J -> K [label = "START Copy FS100 Position SCRIPT.P1 to Position 1004"];
K -> L [label = "FINISH Copy FS100 Position SCRIPT.P1 to Position 1004"];
L -> M [label = "START Go to Position SCRIPT.PB1 Speed 50(%)"];
M -> N [label = "FINISH Go to Position SCRIPT.PB1 Speed 50(%)"];
N -> O [label = "START FS100 Job GET023"];
O -> P [label = "START Wait FS100_DELAY_TIME (10ths of seconds)"];
P -> Q [label = "FINISH Wait FS100_DELAY_TIME (10ths of seconds)"];
Q -> R [label = "TIMEOUT Job GET023"];
Q -> S [label = "FINISH Job Get023"];
S -> T [label = "Send Message $Start to MANAGER ID = TASK_ID"];
T -> U [label = "FINISH Subroutine GET023"];
}

```

A partir de cada arquivo-texto relativo às rotinas da estação do torno foi gerado um autômato como o da Figura 7, presentes no endereço do *GitHub* <<https://github.com/S-Santos17069/PFC-2021.git>>.

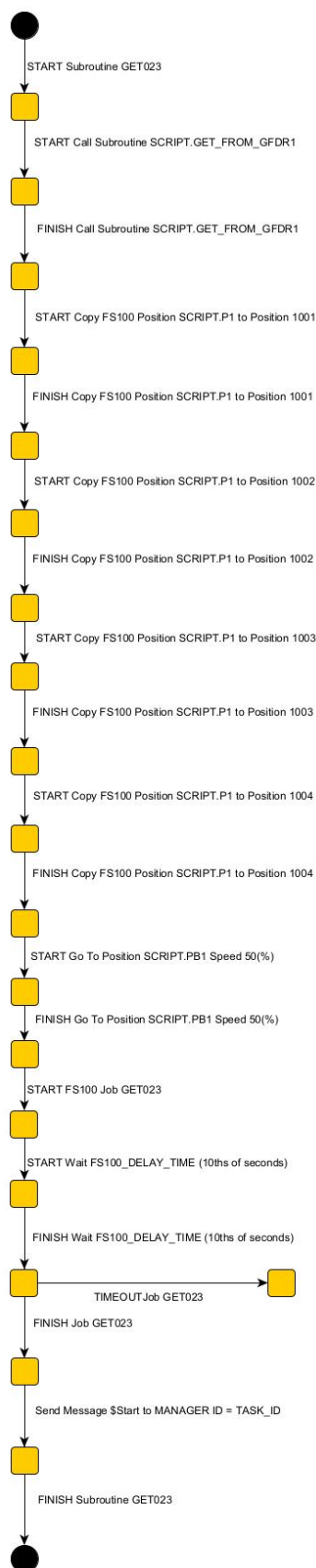


Figura 7 – Diagrama de Estados do Autômato da Subrotina GET023 do ScorBase.

4.2 Variável de Memória

Além dos autômatos das rotinas, também é importante o entendimento do funcionamento das variáveis de memória do ScorBase, que apresentam uma lógica de autômatos diferente da lógica de uma subrotina. As variáveis de memória tem função de receber leituras de pinos escolhidos e podem ser utilizados como parâmetros de transição de estado, como no caso do *If*, que dependendo do valor da variável de memória executará uma transição diferente.

Para as variáveis devem ser previstos em quais estados ela pode ser inicializada e quais as alterações podem ser executadas em seu valor. Um exemplo de código DOT do *GraphViz* está abaixo, seguido pelo seu autômato na Figura 8.

```
digraph LATHE_LOAD{
A -> B
B -> C
B -> D [label = "READ LATHE_LOAD"];
D -> B [label = "LATHE_LOAD = 0"];
B -> E [label = "SET VARIABLE LATHE_LOAD = 1"];
E -> B [label = "SET VARIABLE LATHE_LOAD = 0"];
E -> F [label = "READ LATHE_LOAD"];
F -> E [label = "LATHE_LOAD = 1"];
E -> G
H -> E
}
```

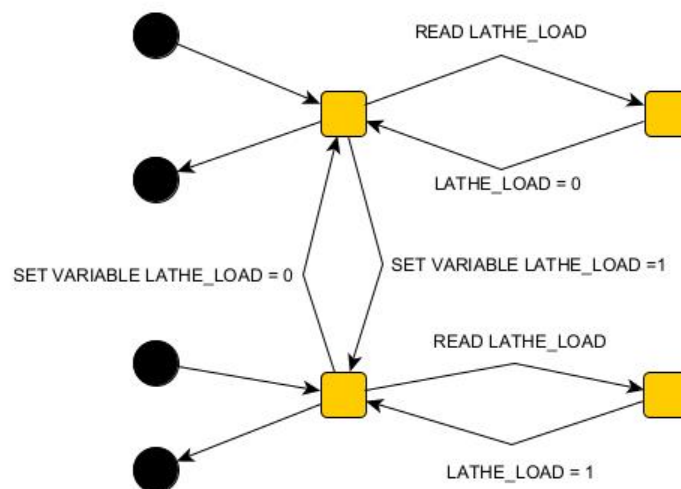


Figura 8 – Diagrama de Estados do Autômato da Variável de Memória LATHE_LOAD.

4.3 Extração Automática dos Autômatos Utilizando a Ferramenta

O próximo passo do projeto foi a construção da ferramenta de automação da criação de códigos DOT e os autômatos em png a partir do script de ScorBase. A linguagem escolhida para isso foi o Python, e o código juntamente com os resultados de todos os ensaios encontram-se no endereço do *GitHub* <<https://github.com/S-Santos17069/PFC-2021.git>>, resultando em códigos como o que segue, novamente da subrotina GET023, e o autômato gerado foi o da Figura 9.

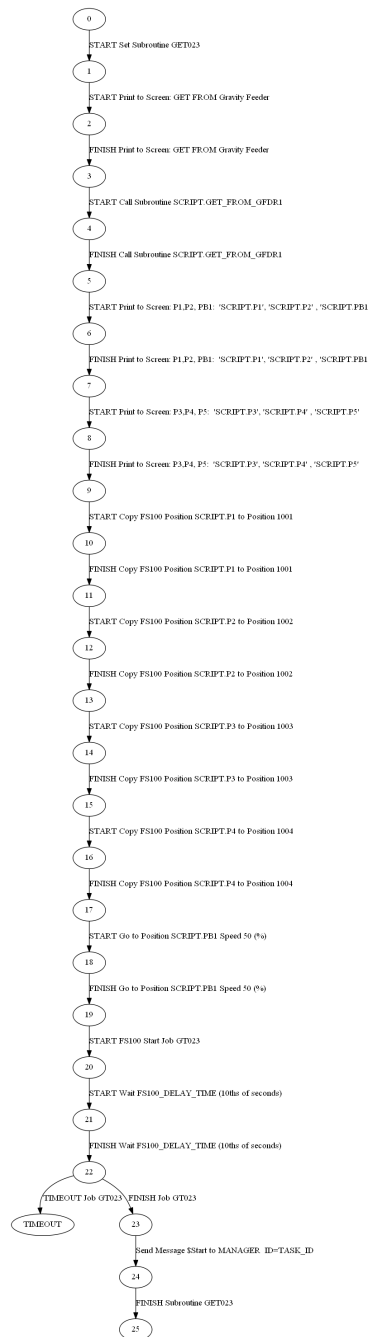


Figura 9 – Autômato da subrotina GET023 do ScorBase gerado pela ferramenta.

```

digraph GET023 {
0 -> 1 [label = "START Set Subroutine GET023"];
1 -> 2 [label = "START Print to Screen: GET FROM Gravity Feeder"];
2 -> 3 [label = "FINISH Print to Screen: GET FROM Gravity Feeder"];
3 -> 4 [label = "START Call Subroutine SCRIPT.GET_FROM_GFDR1"];
4 -> 5 [label = "FINISH Call Subroutine SCRIPT.GET_FROM_GFDR1"];
5 -> 6 [label = "START Print to Screen: P1,P2, PB1:  'SCRIPT.P1',
'SCRIPT.P2' , 'SCRIPT.PB1'"];
6 -> 7 [label = "FINISH Print to Screen: P1,P2, PB1:  'SCRIPT.P1',
'SCRIPT.P2' , 'SCRIPT.PB1'"];
7 -> 8 [label = "START Print to Screen: P3,P4, P5:  'SCRIPT.P3',
'SCRIPT.P4' , 'SCRIPT.P5'"];
8 -> 9 [label = "FINISH Print to Screen: P3,P4, P5:  'SCRIPT.P3',
'SCRIPT.P4' , 'SCRIPT.P5'"];
9 -> 10 [label = "START Copy FS100 Position SCRIPT.P1 to Position 1001"];
10 -> 11 [label = "FINISH Copy FS100 Position SCRIPT.P1 to Position 1001"];
11 -> 12 [label = "START Copy FS100 Position SCRIPT.P2 to Position 1002"];
12 -> 13 [label = "FINISH Copy FS100 Position SCRIPT.P2 to Position 1002"];
13 -> 14 [label = "START Copy FS100 Position SCRIPT.P3 to Position 1003"];
14 -> 15 [label = "FINISH Copy FS100 Position SCRIPT.P3 to Position 1003"];
15 -> 16 [label = "START Copy FS100 Position SCRIPT.P4 to Position 1004"];
16 -> 17 [label = "FINISH Copy FS100 Position SCRIPT.P4 to Position 1004"];
17 -> 18 [label = "START Go to Position SCRIPT.PB1 Speed 50 (%)"];
18 -> 19 [label = "FINISH Go to Position SCRIPT.PB1 Speed 50 (%)"];
19 -> 20 [label = "START FS100 Start Job GT023"];
20 -> 21 [label = "START Wait FS100_DELAY_TIME (10ths of seconds)"];
21 -> 22 [label = "FINISH Wait FS100_DELAY_TIME (10ths of seconds)"];
22 -> TIMEOUT [label = "TIMEOUT Job GT023"];
22 -> 23 [label = "FINISH Job GT023"];
23 -> 24 [label = "Send Message $Start to MANAGER  ID=TASK_ID"];
24 -> 25 [label = "FINISH Subroutine GET023
"];
}

```

Como é perceptível, os autômatos e o código gerado pelo programa automatizado tiveram uma grande confiabilidade quando comparados com os resultados de controle, gerados manualmente. Por isso a ferramenta pode ser considerada confiável dentro do escopo definido, fornecendo a capacidade de testes do código com alterações, devido a velocidade de testes e geração dos autômatos com a utilização da ferramenta, com uma

análise posterior para validação dos autômatos gerados como plausíveis e para encontrar possíveis estados mortos, onde o autômato estaria encerrado, terminando o código.

5 CONCLUSÃO

A utilização da ferramenta mostrou-se efetiva para a extração de forma automatizada dos autômatos das rotinas em ScorBase após a validação com os modelos gerados de forma manual. Esse primeiro passo para uma ferramenta abrangente em todos os níveis da plataforma, sendo uma grande contribuição para as pesquisas realizadas no laboratório.

5.1 Contribuições

A ferramenta representou o início do desenvolvimento da capacidade de operação completa do laboratório de mecatrônica do IME. Com ela é possível validar os modelos criados em pesquisa de forma mais dinâmica e rápida devido a automatização do processo, com a geração de autômatos em GraphViz e também uma visualização por meio de diagramas de estado permitindo a inspeção visual do funcionamento das rotinas alteradas.

A ferramenta também oferece um aumento na vida útil do sistema, pois a validação do código antes de utilizar na prática mitiga possíveis danos aos componetes mecânicos, diminuindo a necessidade de manutenção ou troca de equipamentos por falhas da parte de software.

5.2 Pontos não abordados

Alguns pontos observados como oportunidade de continuidade do projeto e que permitiriam uma abordagem completa da planta incluem:

- Inclusão de toda a gramática ScorBase, englobando as rotinas relativas às outras estações de trabalho;
- Inclusão das variáveis de memória do ScorBase e do controlador FS100;
- Expansão da ferramenta para operar também com os Jobs do controlador FS100;
- Inclusão dos diagramas ciberfísicos da planta.

Essas oportunidades de melhorias podem ser implementadas na ferramenta em futuros trabalhos, aumentando sua capacidade de avaliar códigos mais complexos e específicos dentro das linguagens selecionadas. Também é possível a partir da lógica de funcionamento da extração adaptar a utilização para outras linguagens, adicionando as particularidades de comandos e gramáticas.

Em um último nível, para aumento da produtividade da plataforma MecatrIME, a ferramenta pode ser adaptada para realizar a extração de forma reversa, buscando escrever os arquivos-textos a partir dos autômatos para testes e utilização direta no sistema.

REFERÊNCIAS

- 1 INDÚSTRIA 4.0. 2021. Industry 4.0. Disponível em: <<https://www.i-scoop.eu/industry-4-0/>>. Acesso em: 13 setembro 2021.
- 2 CASSANDRAS, S. L. C. G. Introduction to discrete event system. In: _____. *Introduction to Discrete Event System*. 223, Spring Street, New York NY, Estados Unidos: Springer Science+Business Media.
- 3 JR ORNA GRUMBERG, D. A. P. E. M. C. Model checking. In: _____. *Model checking*. [S.l.]: The MIT Press.
- 4 PARSE. 2021. Site GNU Bison. Disponível em: <<https://www.gnu.org/software/bison/manual/bison.html>>. Acesso em: 13 setembro 2021.
- 5 INSTITUTE, P. M. Gerenciamento de escopo. In: _____. *Guia do conhecimento em gerenciamento de projetos - Guia PMBOK*. 14 Campus Boulevard Newtown Square, Pensilvânia, Estados Unidos: Project Management Institute, 2017. p. 129–171.
- 6 YED. 2021. Site com manual do yEd. Disponível em: <<https://yed.yworks.com/support/manual/index.html>>. Acesso em: 13 setembro 2021.
- 7 WHAT are cyber-physical systems? 2021. What are cyber-physical systems? Disponível em: <<https://www.rmit.edu.au/news/c4de/what-are-cyber-physical-systems>>. Acesso em: 13 setembro 2021.
- 8 SCORBASE. 2021. Site oficial da Intelitek. Disponível em: <https://downloads.intelitek.com/Manuals/Robotics/ER-4u/Scorbase_USB_I.pdf>. Acesso em: 13 setembro 2021.
- 9 FS100 Controller. 2021. Site oficial da Motoman. Disponível em: <<https://www.motoman.com/en-us/products/controllers/fs100>>. Acesso em: 13 setembro 2021.
- 10 HOPCRAFT RAJEEV MOTWANI, J. D. U. J. E. Introduction to automata theory, languages, and computation. In: _____. *Introduction to Automata Theory, Languages, and Computation*. [S.l.]: Pearson Education.
- 11 CURY, J. E. R. Teoria de controle supervisorio de sistemas a eventos discretos. In: *Simpósio Brasileiro de Automação Inteligente*. Rio Grande do Sul: Universidade Federal de Santa Catarina, 2001. 13 set. de 2020. Disponível em: <<http://www.pb.utfpr.edu.br/mt/pdfs/Doutorado/ControleSuperv/apostila.pdf>>.
- 12 DOCUMENTATION for DOT in GraphViz. 2021. Site oficial do GraphViz. Disponível em: <<https://graphviz.org/doc/info/lang.html>>. Acesso em: 13 setembro 2021.

ANEXO A – ANEXOS

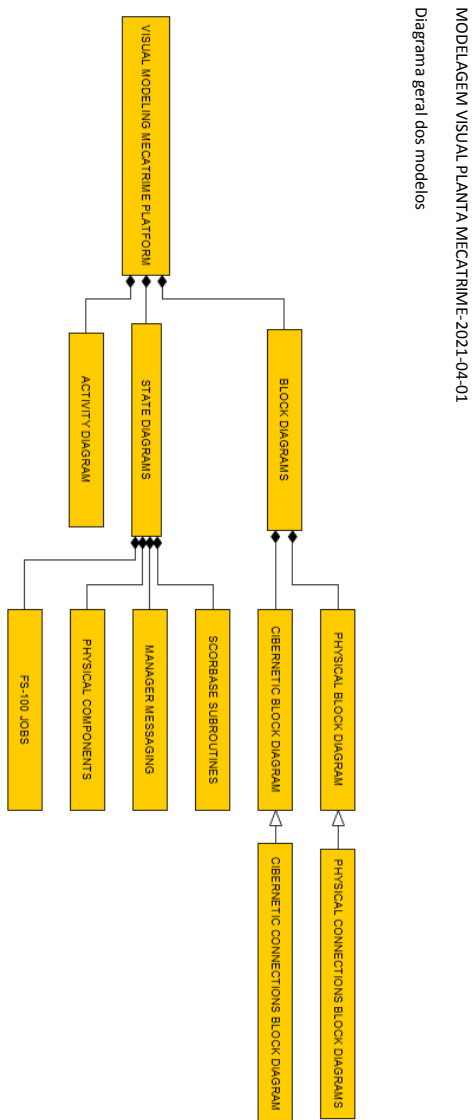


Figura 10 – Diagrama de Blocos da Planta MecatrIME

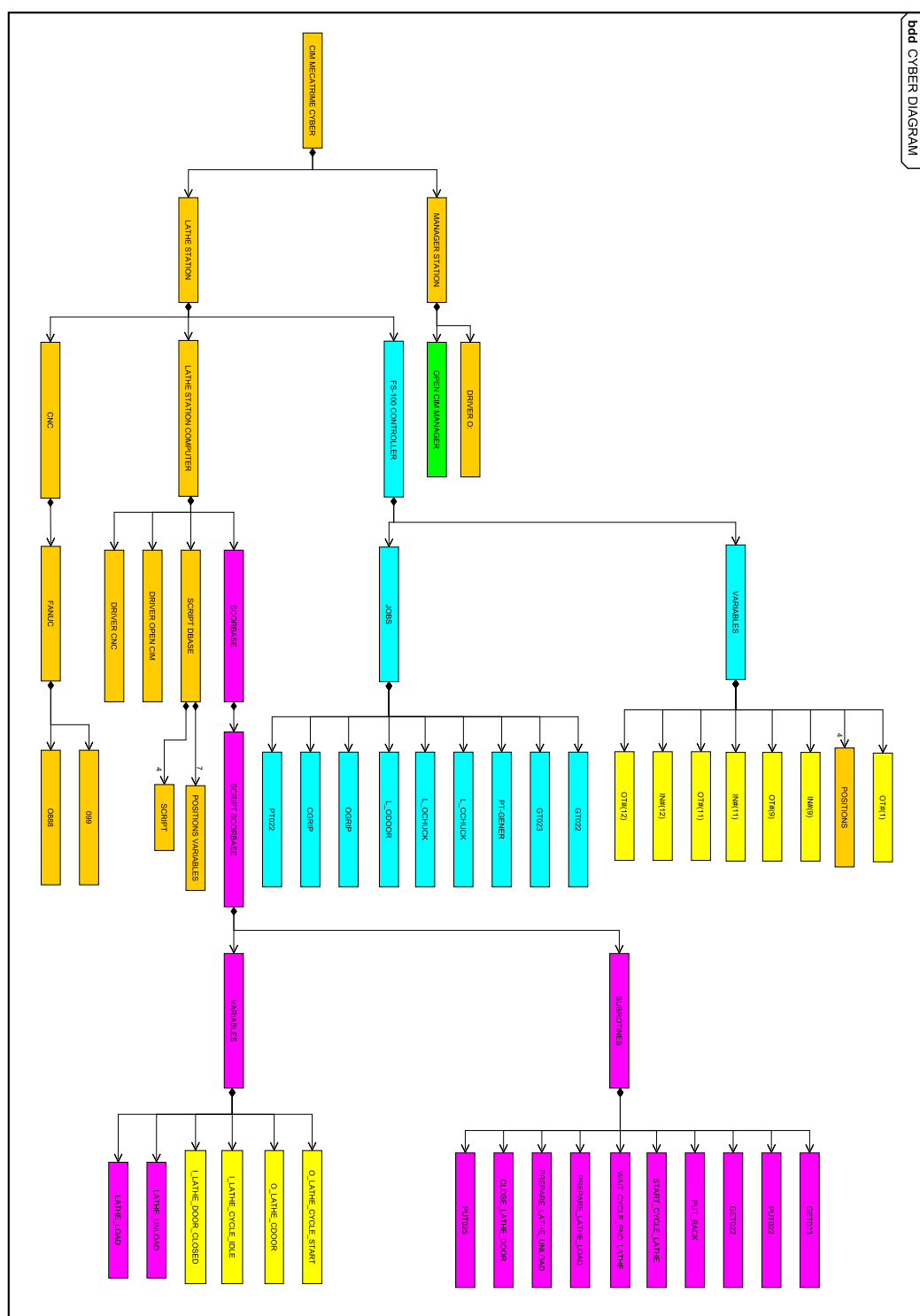


Figura 11 – Diagrama de Blocos Cibernético da Plataforma MecatrIME.

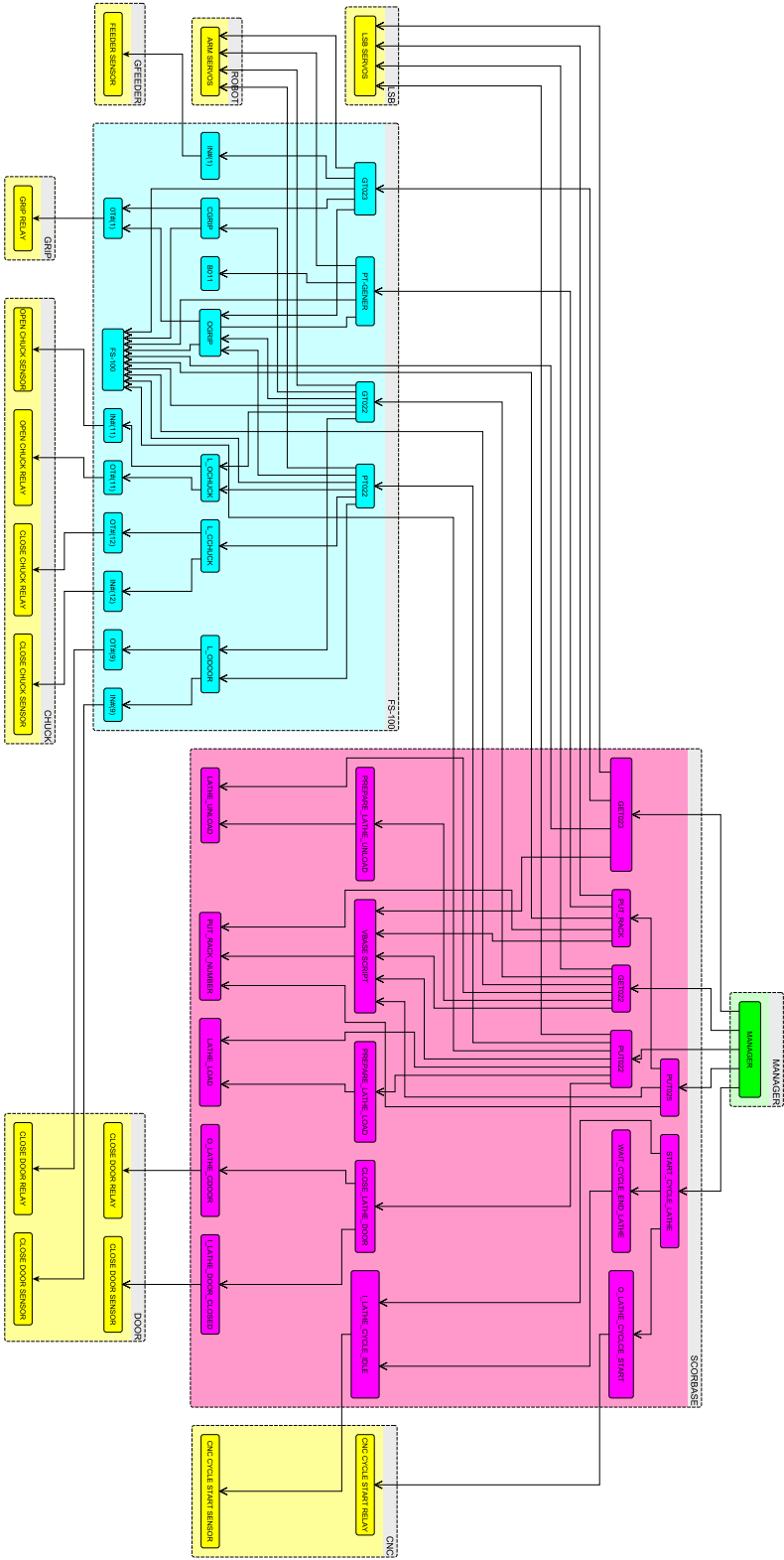


Figura 12 – Diagrama Ciberfísico da Plataforma