

UNIVERSIDADE FEDERAL DE PELOTAS  
CEng – Centro de Engenharias  
Curso de Engenharia de Produção



Trabalho de Conclusão de Curso

## **ANÁLISE DA CONFIABILIDADE EM SISTEMAS AUTOMÁTICOS**

Implementação da ferramenta MACSA aplicada a um estudo de caso

Monica Moura Pacheco

Orientador:  
Prof. Dr. Gilson Porciúncula

Pelotas, Novembro de 2016

Monica Moura Pacheco

**ANÁLISE DA CONFIABILIDADE EM SISTEMAS AUTOMÁTICOS**  
Implementação da ferramenta MACSA aplicada a um estudo de caso

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Produção em ..... no CEng – Centro de Engenharias da Universidade Federal de Pelotas, como requisito parcial à obtenção do título de Bacharel em Engenharia de Produção.

Orientador:  
Prof. Gilson Porciúncula

Pelotas, Novembro de 2016

Monica Moura Pacheco

**ANÁLISE DA CONFIABILIDADE EM SISTEMAS AUTOMÁTICOS**  
Implementação da ferramenta MACSA aplicada a um estudo de caso

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Produção em ..... no CEng – Centro de Engenharias da Universidade Federal de Pelotas, como requisito parcial à obtenção do título de Bacharel em Engenharia de Produção.

Data da defesa: ..29/11/2016.....

Banca examinadora:

.....  
Prof. Dr. ....(Orientador)  
Doutor em .....pela Universidade.....

.....  
Prof. Dr. ....  
Doutor em .....pela Universidade.....

.....  
Prof. Dr. ....  
Doutor em .....pela Universidade.....

À minha família, por acreditar e investir em mim. Mãe, seu cuidado e dedicação me deram a esperança para seguir. Irmão, sua presença significou segurança e certeza de que não estou sozinha nessa caminhada. Ao meu namorado que de forma especial e carinhosa me deu força e coragem, me apoiando nos momentos de dificuldades. Agradeço ao meu professor orientador que teve paciência e me ajudou a concluir este trabalho.



## RESUMO

PACHECO, Monica. Análise da confiabilidade em sistemas automáticos: Implementação da ferramenta MACSA aplicada a um estudo de caso. 2016. 102f. Trabalho de Conclusão de Curso – Curso de Graduação em Engenharia de Produção, CEng – Centro de Engenharias, Universidade Federal de Pelotas, Pelotas, 2016.

A progressiva substituição da mão de obra humana por sistemas automatizados faz com que estes sistemas estejam cada vez mais presentes na indústria. Para garantir uma posição competitiva e uma taxa elevada de produtividade é essencial que os sistemas automáticos sejam confiáveis. Com base nesse contexto, o presente trabalho avalia as principais metodologias de análise da confiabilidade para sistemas automáticos. A partir dessa contextualização, é proposto o desenvolvimento de uma ferramenta computacional baseada na Metodologia para Análise de Confiabilidade de Sistemas Automáticos (MACSA). Os conceitos da linguagem de modelagem unificada (UML) formaram a base para modelagem desta ferramenta e um estudo de caso ensaiou o desempenho da mesma. A ferramenta MACSA Web 1.0 dá subsídios para tomadas de decisão a respeito da confiabilidade e concepção do sistema. Seu desempenho foi eficiente em analisar a confiabilidade de um sistema eletropneumático com três estados operacionais distintos.

Palavras-chave: sistemas automáticos; confiabilidade; MACSA

## **ABSTRACT**

Pacheco, Monica. Reliability analysis in automated systems: Implementation of MACSA tool applied to a case study. 2016. 102f. Trabalho de Conclusão de Curso – Curso de Graduação em Engenharia de Produção, CEng – Centro de Engenharias, Universidade Federal de Pelotas, Pelotas, 2016.

The progressive replacement of human labor by automated systems makes these systems increasingly present in the industry. To ensure a competitive position and a high rate of productivity it is essential that automatic systems are reliable. Based on this context, the present work evaluates the main methodologies of reliability analysis for automatic systems. From this context, it is proposed the development of a computational tool based on the Automatic Systems Reliability Analysis Methodology (MACSA). The concepts of the unified modeling language (UML) formed the basis for modeling this tool and a case study analyzed its performance. The MACSA Web 1.0 tool provides insights for making decision regarding system reliability and design. Its performance was efficient in analyzing the reliability of an electro-pneumatic system with three different operational states.

Palavras-chave: automated systems; reliability; MACSA

## SUMÁRIO

1	INTRODUÇÃO.....	13
1.1	Objetivos Geral e Específicos.....	14
1.1.1	Objetivo Geral.....	14
1.1.2	Objetivos Específicos.....	14
1.2	Justificativa.....	15
1.3	Limitações.....	16
1.4	Estrutura do Trabalho.....	17
2	REVISÃO TEÓRICA.....	19
2.1	Sistemas Automáticos.....	19
2.1.1	Sistemas Pneumáticos.....	21
2.1.2	Sistemas Hidráulicos.....	22
2.1.3	Modelagem de sistemas automáticos.....	24
2.2	Metodologias de Desenvolvimento de Produtos.....	28
2.3	Confiabilidade.....	29
2.3.1	Confiabilidade em Sistemas Automáticos.....	32
2.4	Metodologia para Análise de Confiabilidade em Sistemas Automáticos (MACSA).....	38
2.5	Modelagem de Software.....	42
2.5.1	Análise de Requisitos.....	43
2.5.2	Análise Sistêmica.....	44
2.5.3	Projeto.....	47
2.5.4	Implementação.....	48
2.5.5	Testes e Implantação.....	48
3	METODOLOGIA.....	50
3.1	1º Etapa - Análise de Requisitos.....	51
3.1.1	Atividade 1 - Caracterizar o sistema.....	51
3.1.2	Atividade 2 - Definir configurações Operacionais.....	53
3.1.3	Atividade 3 - Definir modelos de confiabilidade.....	54
3.1.4	Atividade 4 - Estimar a confiabilidade.....	55
3.1.5	Atividade 5 - Avaliar o parâmetro confiabilidade.....	55
3.1.6	Atividade 6 - Documentar a análise da confiabilidade.....	56
3.2	2º Etapa - Análise Sistêmica.....	56
3.3	3º Etapa - Projeto.....	56



3.4	4° Etapa - Implementação .....	57
3.5	5° Etapa - Testes e Implantação.....	57
4	DESENVOLVIMENTO DO APLICATIVO MACSA .....	59
4.1	1° Etapa - Análise de Requisitos.....	59
4.2	2° Etapa - Análise Sistêmica.....	61
4.3	3° Etapa - Projeto.....	63
4.4	4° Etapa - Implementação .....	69
4.5	5° Etapa - Testes e Implantação.....	71
5	DESCRIÇÃO DO ESTUDO DE CASO .....	73
5.1	Modelagem funcional e estrutural .....	75
5.2	Modelagem comportamental.....	76
5.3	Caracterizar o sistema Eletropneumático .....	77
5.4	Caracterizar o Subsistema de Controle .....	79
5.5	Caracterizar o Subsistema Pneumático .....	82
5.6	Estimativa da Confiabilidade.....	84
6	CONCLUSÕES .....	87
7	REFERÊNCIAS .....	88

## LISTA DE FIGURAS

Figura 1	Desdobramentos da FG num sistema hidráulico.....	21
Figura 2	Sistema Pneumático simples .....	22
Figura 3	Sistema Hidráulico básico .....	24
Figura 4	Elementos básicos da rede Canal/Agência.....	26
Figura 5	Elementos básicos do Grafcet.....	27
Figura 6	A MACSA inserida no projeto de produto.....	38
Figura 7	Representação das atividades da MACSA.....	40
Figura 8	Representação do Diagrama de Casos de Uso .....	44
Figura 9	Representação do Diagrama de Classes .....	45
Figura 10	Etapas da Metodologia.....	50
Figura 11	Diagrama caso de uso do aplicativo MACSA .....	60
Figura 12	Diagrama de classes do domínio central.....	62
Figura 13	Diagrama de classes extrapolado .....	64
Figura 14	Diagrama de sequência do caso de uso 'Inserir Dados'.....	66
Figura 15	Diagrama de sequência do caso de uso 'Salvar Estado'.....	67
Figura 16	Diagrama de sequência do caso de uso 'Importar Dados' .....	67
Figura 17	Diagrama de Implantação combinado com diagrama de componentes...	68
Figura 18	Interface MACSA para dispositivos móveis com Bootstrap.....	70
Figura 19	Interface MACSA para alta resolução com Bootstrap .....	70
Figura 20	Bancada eletropneumática para ensaio de gatilhos.....	73
Figura 21	Diagrama de circuito do sistema eletropneumático .....	74
Figura 22	Modelo funcional/estrutural do sistema eletropneumático.....	75
Figura 23	Modelo comportamental do sistema eletropneumático .....	76
Figura 24	Modelo do desdobramento do SE em rede C/A .....	77
Figura 25	Modelo do desdobramento do SC em rede C/A.....	79
Figura 26	Modelo do desdobramento do SP em rede C/A .....	82
Figura 27	Redução dos modelos funcional/estrutural para Estado 1 .....	85
Figura 28	Redução dos modelos funcional/estrutural para Estado 2 .....	86

## LISTA DE TABELAS

Tabela 1	Estados Operacionais do sistema Eletropneumático .....	77
Tabela 2	Caracterização das agências do Sistema Eletropneumático.....	78
Tabela 3	Representação matricial do sistema eletropneumático .....	78
Tabela 4	Representação matricial do SC .....	80
Tabela 5	Caracterização das agências do SC .....	81
Tabela 6	Matriz de Mapeamento de Controle do SC .....	81
Tabela 7	Representação matricial do modelo desdobrado do SP.....	83
Tabela 8	Caracterização das agências do SP.....	83
Tabela 9	Matriz de Mapeamento de Controle do SP.....	84

## LISTA DE SIGLAS E ABREVIATURAS

AGV	Veículos Guiados Automaticamente
API	Interface de Programação de Aplicativos
BOP	Preventor de Explosões
BPN	Redes de Petri Comportamentais
CAS	Sistema de Assistência Cardíaca
C/A	Canal/Agência
CPU	Unidade Central de Processamento
DFT	Árvore de Falha Dinâmica
DRBD	Diagrama de Blocos de Confiabilidade Dinâmicos
DyFA	<i>Dynamic Fault-trees Analyzer</i>
E/R	Entidade/Relacionamento
FG	Função Global
FMEA	<i>Failure Mode and Effects Analysis</i>
FTA	Análise da Árvore de Falha
Grafcet	Gráfico Funcional de Controle de Etapas e Transições
GSPN	Redes de Petri Estocásticas Generalizadas
HCAS	Sistema Hipotético de Assistência Cardíaca
MACSA	Metodologia para análise da confiabilidade de sistemas automáticos
PRODIP	Processo de desenvolvimento integrado de produtos
RB	Rede Bayesina
RBD	Diagrama de Blocos de Confiabilidade
RP	Redes de Petri
SADT	<i>Structured Analysis and Design Technique</i>
SC	Subsistema de controle
SED	Sistemas a Eventos Discretos
SIS	Sistemas Instrumentados de Segurança
SP	Subsistema pneumático
STA	<i>Sucess Tree Analysis</i>
UML	Linguagem de Modelagem Unificada
VFG	Variante da função global

## 1 INTRODUÇÃO

Os sistemas automáticos estão presentes no nosso cotidiano e são utilizados principalmente para facilitar a vida e trabalho do ser humano. As transformações advindas da automação industrial podem não ser visíveis para quem não convive no meio fabril, entretanto a automação vem se desenvolvendo e está cada vez mais presente. Essa tendência tem determinado nos últimos anos a crescente automatização dos bancos, comércio e empresas em geral. Vilela e Vidal (2003) afirmam que a história da humanidade é composta por um processo de redução do esforço humano requerido e devido a isso ocorreu o progresso da automação.

A expansão do campo da automação tem reduzido a intervenção do operador. Isso ocorre devido às capacidades de auto-regulação dos sistemas e execução autônoma de diversas funções pré-programadas. A implementação da execução automática demanda a integração e uso de componentes de diferentes tecnologias (PORCIÚNCULA *et al.*, 2015).

Os sistemas automáticos têm papel importante no aumento do desempenho e produtividade no meio industrial, entretanto o crescente aumento das suas funcionalidades tem aumentado a complexidade destes sistemas e portanto as tarefas de modelagem e análise. Riascos (2002) considera que a confiabilidade nestes sistemas é essencial para o desenvolvimento de sistemas mais flexíveis e autônomos.

A confiabilidade é um requisito não apenas para reduzir custos e o tempo de produção. Ela também melhora a qualidade do produto, aumenta a competitividade e garante o desenvolvimento sustentável das empresas (CHU, YUAN e CHEN, 2015). A avaliação da confiabilidade é um passo importante, muitas vezes indispensável, para projetar e analisar sistemas. Sua importância está em constante crescimento (DISTEFANO e PULIAFITO, 2009).

A garantia da confiabilidade e segurança dos sistemas automáticos são fundamentais. No entanto a análise e avaliação desses parâmetros podem ser dificultadas, em virtude da confiabilidade e da segurança estarem sujeitas ao estado de falha de cada componente e a sequência de ocorrência destas falhas (PORCIÚNCULA, 2009).

Desta forma, Porciúncula (2009) apresentou a Metodologia para Análise de Confiabilidade de Sistemas Automáticos (MACSA) que é aplicável nas fases iniciais do processo de projeto e permite a identificação automática das configurações do

sistema nos seus potenciais estados operacionais. A estimativa da confiabilidade do sistema é baseada em modelos de confiabilidade que são definidos a partir das configurações operacionais citadas acima.

A MACSA emprega modelos que podem ser representados na forma matricial, então pode ser implementada computacionalmente. A proposta desse trabalho consiste no desenvolvimento de uma ferramenta computacional aplicando essa metodologia para o cálculo da confiabilidade. Um estudo de caso será contemplado para validação da ferramenta.

## 1.1 **Objetivos Geral e Específicos**

Os objetivos deste trabalho encontram-se divididos em geral e específicos, sendo estes:

### 1.1.1 **Objetivo Geral**

O objetivo deste trabalho é propor a implementação de uma ferramenta computacional baseada em uma metodologia para a análise da confiabilidade de sistemas automáticos e a aplicação desta a um estudo de caso para sua validação.

### 1.1.2 **Objetivos Específicos**

O presente trabalho possui como objetivos específicos:

- (a) Levantamento dos principais métodos para análise de confiabilidade;
- (b) Análise da aplicação dos métodos de análise da confiabilidade em sistemas automáticos;
- (c) Posicionar a MACSA mediante demais métodos de análise da confiabilidade;
- (d) Implementação computacional da MACSA;
- (e) Aplicar a ferramenta computacional desenvolvida a um estudo de caso;
- (f) Comparar os resultados obtidos pela ferramenta com os cálculos realizados previamente;
- (g) Análise dos resultados obtidos e deliberação das conclusões.

## 1.2 Justificativa

A globalização da economia e a crescente competitividade entre as empresas exigem que sejam colocados no mercado produtos cada vez mais complexos e confiáveis, com baixo custo e alta qualidade.

A garantia da confiabilidade de um sistema potencializa o mesmo para uma vida útil com poucas paradas devido a falhas, ao realizar uma função específica num tempo pré determinado.

A busca por incrementar a produtividade e o crescimento tecnológico tem implicado na utilização cada vez mais comum de sistemas automáticos. De Negri e Santos (2007) afirmam que estes são muito aplicados em sistemas de manufatura e equipamentos industriais automatizados.

A substituição do trabalho humano por máquinas motivou a utilização de graus cada vez mais elevados de automação e o uso de componentes de tecnologias diversas. Estes fatos aliados a crescente demanda no mercado por produtos tecnológicos, trouxe a tona nas equipes de projeto a preocupação em relação à confiabilidade e a manutenibilidade dos sistemas automáticos. Por consequência, obter soluções que aprimorem as etapas de projeto e que afetem de forma positiva os atributos de confiabilidade e manutenibilidade do sistema é imprescindível para o desenvolvimento dos sistemas automáticos (PORCIÚNCULA, 2009).

A confiabilidade está relacionada ao sucesso na operação de um sistema ou produto, ou seja, a inexistência de quebras ou falhas. Fogliatto e Ribeiro (2009) afirmam que os conceitos de confiabilidade e qualidade estão atrelados. Muitas vezes a confiabilidade pode ser considerada como uma característica importante de qualidade que deve ser levada em conta no projeto.

Porciúncula (2009) afirma que o avanço da tecnologia favoreceu a interação entre componentes mecânicos, eletrônicos e de *softwares*. Assim introduziu novas possibilidades de falhas nos sistemas. Bernard, Cauffriez e Renaux (2008) declaram ainda que o uso sistemático de novas tecnologias, como: tecnologias da informação, inteligência artificial, sistemas especialistas ou redes de comunicação, podem levar a falhas. O desconhecimento de como essas tecnologias irão interagir no sistema em determinado ambiente é a principal causa das falhas.

Apesar dos avanços na tecnologia de automação, falhas são eventos que não podem ser ignorados numa máquina automática real (RIASCOS, COZMAN e

MIYAGI, 2003). A análise e avaliação dos parâmetros confiabilidade e segurança dos sistemas automáticos são tarefas difíceis e podem apresentar vários contratempos. Isto ocorre pois estes atributos dependem do estado de falha de cada componente e da sequência de ocorrência destas falhas (PORCIÚNCULA, 2009).

Os sistemas automáticos devem ser altamente confiáveis para que seu comportamento seja controlado. Bernard, Cauffriez e Renaux (2008) definem sistema como um conjunto de elementos discretos, interconectados e interativos que trabalham juntos para atender uma função pré determinada. À medida que o número de componentes cresce, da mesma forma aumenta a probabilidade do sistema falhar.

Ao longo do ciclo de vida dos sistemas automáticos é essencial realizar um planejamento estratégico da manutenção, levando em conta as características do funcionamento automático. Todas as funções devem ser atendidas, pois vários efeitos indesejados podem derivar de uma falha no atendimento de uma função num sistema automático. Estas falhas podem ser pequenas perdas na qualidade ou até o não atendimento total da função do sistema. A confiabilidade tem sido um dos parâmetros mais importantes a serem considerados no processo de projeto de sistemas automáticos (PORCIÚNCULA, 2009).

Uma questão fundamental para estimar a confiabilidade é conhecer a inter-relação entre a estrutura do sistema e as funções que devem ser executadas pelo mesmo em cada estado operacional. Máquinas e equipamentos automáticos são projetados usando representações padrão, como: diagramas elétricos e de circuitos hidráulicos e/ou pneumáticos, desenhos mecânicos e linguagens de programação. Portanto, é útil que estes diagramas possam ser formalmente e automaticamente integrados e convertidos em modelos adequados para a análise da confiabilidade (PORCIÚNCULA *et al.*, 2015).

### 1.3 Limitações

No que se refere ao universo do presente estudo existem restrições de natureza estrutural inerentes à metodologia selecionada para implementação, uma vez que somente as etapas de manipulação de modelos matemáticos serão aplicados na ferramenta computacional a ser desenvolvida.



De Negri e Santos (2007) definem como sistemas complexos aqueles em que o conjunto constitui mais que a soma das partes. Isto é, conhecidas as propriedades das partes e as leis que regem suas interações não é trivial inferir sobre as propriedades do todo.

A complexidade dos sistemas automáticos pode ser caracterizada em função de diferentes atributos, como: tecnologia, estados, tamanho, interação de componentes e sistemas. Quanto maior o valor destes atributos, maior a complexidade do sistema. Então, para a parte referente ao estudo de caso foi considerada a análise de um sistema pouco complexo, ou seja, um sistema não muito extenso e com poucos estados operacionais. Isso se deve ao fato da Metodologia para Análise da Confiabilidade de Sistemas Automáticos (MACSA) manipular modelos representativos desenvolvidos na fase de projeto conceitual para o cálculo da confiabilidade. Esses modelos necessitam ser convertidos em modelos matemáticos para possibilitar seu uso pela ferramenta computacional.

Vale observar que a programação de uma interface que interprete os modelos representativos e os converta em modelos matemáticos de forma automática, demandaria um conhecimento que está além do que foi obtido no curso de engenharia de produção.

#### 1.4 Estrutura do Trabalho

Este trabalho está dividido em 4 capítulos com as seguintes disposições de conteúdo:

No capítulo 2 é apresentada uma revisão bibliográfica dos principais assuntos relacionados aos objetivos deste trabalho. São apresentadas a caracterização e conceituação de sistemas automáticos. Inserem-se nestes os sistemas hidráulicos e pneumáticos e é discutida também as principais formas de modelagem dos sistemas automáticos. São apresentadas algumas das principais metodologias de desenvolvimento de produtos voltadas para esses sistemas. Em seguida é realizado um levantamento dos principais métodos de análise da confiabilidade orientados para sistemas automáticos. A Metodologia de Análise da Confiabilidade para Sistemas Automáticos (MACSA) é apresentada, seus principais conceitos e princípios de funcionamento, além de expostas suas vantagens frente aos demais métodos. Por fim, são apresentados os conceitos de modelagem de *software*.

O capítulo 3 esclarece a metodologia que será utilizada para atingir os objetivos gerais e específicos. O capítulo 4 apresenta as etapas fundamentadas na UML utilizadas para modelagem e desenvolvimento do aplicativo MACSA Web 1.0. Um estudo de caso é apresentado no capítulo 5, com o objetivo de elucidar a eficiência no desempenho do aplicativo desenvolvido.

Por fim, o capítulo 6 traz as considerações acerca do trabalho desenvolvido. O capítulo 7 registra as referências que serviram de embasamento e tornaram possível o desenvolvimento deste trabalho de conclusão de curso.

## 2 REVISÃO TEÓRICA

Essa sessão aborda o domínio da pesquisa deste trabalho. Primeiramente foram definidos os sistemas automáticos. Baseada nessa conceituação foram ponderados os sistemas hidráulicos e pneumáticos. A evolução tecnológica destes incorporou a tecnologia eletrônica e de *software*. Isto definiu novos tipos de relações entre componentes de diferentes tecnologias, e delineou novas variáveis a serem levadas em conta na análise da confiabilidade. Além disto, apresentam-se os principais métodos e ferramentas utilizadas para a modelagem de sistemas automáticos na fase de processo de projeto.

A atividade de projeto deve ser norteada por uma metodologia que abrange o uso de ferramentas e métodos, de forma sistematizada, e que auxilie nas tomadas de decisões no desenvolvimento de produtos. Fundamentado nisso, são abordadas as metodologias de projeto voltadas para o contexto dos sistemas automáticos.

Em seguida são apresentadas as definições do atributo confiabilidade de sistemas. Uma delimitação das pesquisas na linha da confiabilidade no contexto dos sistemas automáticos foi realizada, considerando os principais conceitos, métodos e ferramentas utilizados na análise de confiabilidade. Em seguida é apresentada a Metodologia para Análise de Confiabilidade em Sistemas Automáticos (MACSA) e sua contextualização. Por fim são mostrados os conceitos de modelagem de *software*.

### 2.1 Sistemas Automáticos

Porciúncula (2009) define sistemas automáticos como dispositivos ou equipamentos que fazem uso de recursos variados, como softwares de controle, componentes mecânicos e eletroeletrônicos, para realizar uma tarefa estabelecida e que são programados para desenvolver suas tarefas de forma autônoma.

Conforme De Negri e Santos (2007) sistema automático se refere a uma aplicação que compreenda automação e/ou controle. Nele é possível perceber o problema por uma visão discreta ou de forma mais minuciosa. Além de avaliar as respostas da posição, força, velocidade, temperatura ou outra variável ao longo do tempo. Os sistemas de automação e de controle são designados por um conjunto de componentes interconectados. O sistema de automação tem como função principal executar uma ou mais ações de acordo com uma lógica predeterminada, em resposta ao estado atual do sistema e à ocorrência de eventos. O sistema de

controle tem como principal atribuição realizar uma ou mais ações ao longo do tempo, cujas modificações decorrem da aplicação de sinais de entrada.

Porciúncula, *et al.* (2005) discorrem sobre os sistemas automáticos como uma combinação adequada de três áreas: a mecânica, de *software* e eletrônica. Essa combinação confere ao sistema as seguintes características: simplificação do sistema mecânico, redução do custo e tempo de desenvolvimento, facilidade na introdução de modificações ou novas capacidades e flexibilidade para receber modificações futuras ou novas funcionalidades.

Os sistemas automáticos possuem algum grau de autonomia e flexibilidade para realizar suas funções e possuem diferentes maneiras de atender a sua função global. A Função Global (FG) representa a relação entre as entradas e as saídas, fundamentada no fluxo de energia, matéria e sinal do sistema. A FG pode ser atendida por diferentes Variantes da Função Global (VFG) que expressam as relações entre as entradas e saídas do sistema que resolvem um problema específico de formas distintas. Num sistema automático o atendimento de uma VFG sugere que o sistema realiza uma demanda do usuário para uma necessidade específica, a qual está definida no sistema de controle que indica a configuração funcional/estrutural específica do sistema (PORCIUNCULA, 2009).

Dessa forma a função do sistema é constituída por várias VFGs, que se revezam conforme o estado operacional em que o sistema se encontra. Nos sistemas automáticos o *software* de controle determina quais VFGs precisam ser atendidas e define qual configuração dos arranjos funcionais e estruturais do sistema que estará operante durante um intervalo de tempo. Os diferentes arranjos configurados incorporam componentes variados do sistema, sendo que alguns componentes ficam desativados e outros são compartilhados por diferentes configurações. A figura 1 mostra um exemplo dos desdobramentos da FG em diferentes VFGs para um sistema hidráulico. Nesse diagrama a Função Global se desenvolve para três variantes: VFG1, VFG2 e VFG3.

A solicitação feita pelo usuário do sistema irá determinar qual das três variantes da função global (VFG1, VFG2 ou VFG3) será atendida. Contudo cabe ao sistema de controle a determinação da estrutura funcional que vai satisfazer cada uma das VFGS.

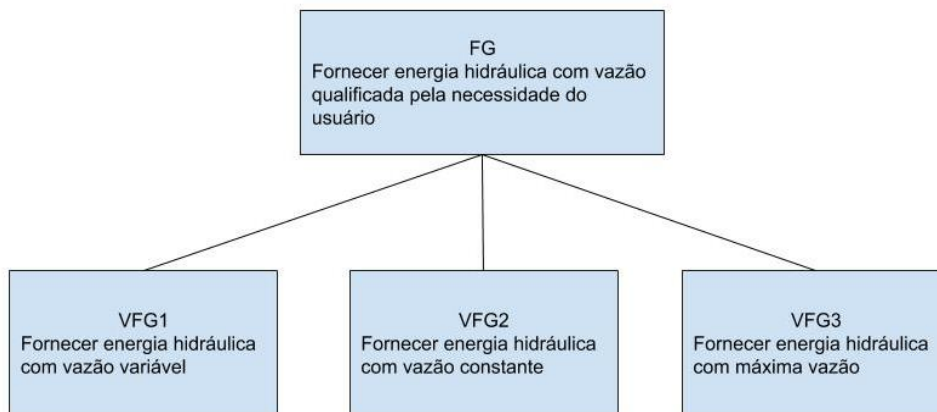


Figura 1 Desdobramentos da FG num sistema hidráulico  
Fonte: Adaptado de PORCIÚNCULA (2009)

### 2.1.1 **Sistemas Pneumáticos**

Diversas máquinas e processos empregam um fluido para desenvolver uma força que move ou sujeita um objeto e/ou controla uma ação. A indústria moderna está utilizando muitas unidades a fluido e suas máquinas e processos estão se tornando mais automatizados para manter uma posição competitiva no mercado. Muitos fluidos podem ser utilizados em dispositivos e sistemas, um sistema pneumático utiliza o ar comprimido como meio de energia para máquinas e equipamentos (STEWART, 2002).

Pequeno (2008) afirma que as limitações e vantagens dos sistemas pneumáticos derivam da compressibilidade e da baixa velocidade do ar comprimido. Dentre as vantagens dos acionamentos pneumáticos destacam-se: matéria-prima abundante e de custo baixo; simplicidade no transporte e armazenamento de energia; não poluente; segurança, boa velocidade dos atuadores e auto proteção contra sobrecargas. Dentre as limitações desses sistemas encontram-se: inviável economicamente para pressões altas; escape ruidoso, pequena força exercida; o ar necessita de tratamento inicial, imprecisão no controle de velocidade.

Parker (2010) destaca como vantagens dos sistemas pneumáticos a rapidez nos movimentos e a liberação do operário de operações repetitivas, o que torna possível o aumento do ritmo de trabalho e da produtividade. Os controles pneumáticos não exigem grande especialização dos funcionários para operá-los, sendo de fácil manipulação. A robustez dos componentes pneumáticos os tornam resistentes a vibrações e golpes, permitindo que ações mecânicas, próprias do

processo, sejam utilizadas como sinal para as várias sequências da operação. Uma limitação relevante consiste na impossibilidade obter paradas intermediárias e velocidades uniformes, pois o ar é um fluido altamente compressível.

Os sistemas pneumáticos possuem três subsistemas principais: (1) Compressor de ar, (2) um atuador linear ou um atuador rotativo (motor pneumático), (3) dispositivos de controle do fluxo do fluido (válvulas pneumáticas). Todas essas características são apresentadas na figura 2, que mostra uma representação gráfica de um sistema pneumático simples. O lubrificador é uma unidade utilizada para regular e lubrificar o ar que irá fluir pelo sistema, garantindo menor desgaste por atrito dos componentes e sobrecargas.

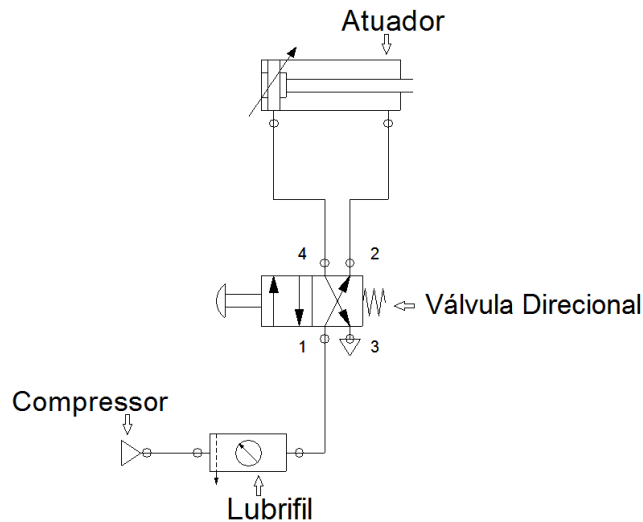


Figura 2 Sistema Pneumático simples  
Fonte: Autor

Quando o sistema for acionado pelo botão presente na válvula direcional, a passagem do ar comprimido é liberada, agindo no pistão de modo a desenvolver uma força sobre a haste daquele para mover um dispositivo ou carga. Quando o botão é liberado o sistema retorna para a posição de repouso, ou seja, o ar comprimido está fluindo pelo sistema no sentido de manter o atuador na posição recuada.

### 2.1.2 **Sistemas Hidráulicos**

Um sistema hidráulico pode fazer uso de óleo, água ou outro líquido, porém na prática as indústrias utilizam o óleo como fluido para alimentar seus sistemas

hidráulicos (STEWART, 2002). Assim sendo, esse trabalho compreende os sistemas hidráulicos industriais que utilizam o óleo como fluido.

De acordo com Pequeno (2008) as principais vantagens dos sistemas hidráulicos são: exercer grandes pressões e forças; possibilidade de variações micrométricas de velocidade; auto lubrificação devido ao tipo de fluido utilizado o que permite uma rápida e suave inversão dos movimentos em virtude da baixa inércia. Dentre as desvantagens dos acionamentos hidráulicos destacam-se: seu alto custo; o atrito, transformações de energia e vazamentos internos resultam num baixo rendimento e a sensibilidade às variações de temperatura.

A utilização do óleo como fluido hidráulico de operação apresenta algumas vantagens. Ele atua como agente lubrificante nas várias peças deslizantes do sistema, evita a oxidação e é encontrado facilmente. O óleo é um líquido que não sofre variação no seu volume quando a pressão for variada no decorrer do seu deslocamento de uma parte a outra do sistema hidráulico. Abastecendo totalmente o sistema com óleo é possível controlar corretamente o movimento do pistão pelo fluxo do mesmo (STEWART, 2002).

Os sistemas hidráulicos possuem aplicações diversas e incluem sistemas de controle de posição, por isso estão presentes em diversos setores industriais. Eles vêm ganhando destaque nas áreas de Hidráulica Industrial e Mobil, movimentação de materiais, máquinas industriais, máquinas pesadas (tratores, guindastes, escavadoras) e também são amplamente utilizados no setor de construção e agricultura. A figura 3 traz uma representação de um sistema hidráulico básico.

O circuito da unidade de potência é composto pela bomba hidráulica, filtro e reservatório e é o responsável pela conversão da energia mecânica em energia hidráulica. As válvulas redutoras de pressão e direcional são responsáveis pelo controle e direcionamento da energia hidráulica gerada pela bomba.

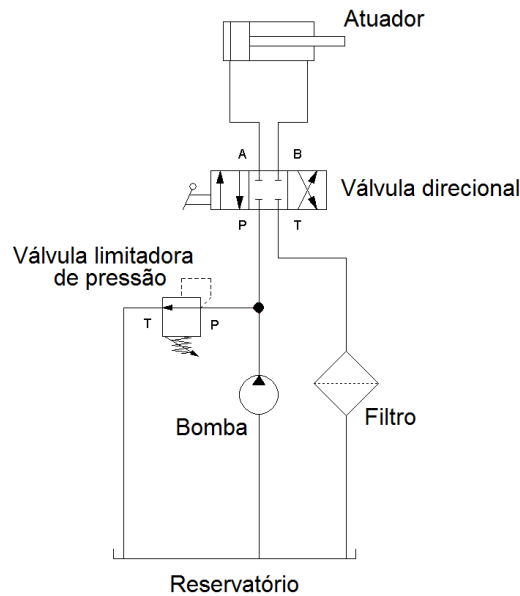


Figura 3 Sistema Hidráulico básico  
Fonte: Autor

O motor hidráulico e o atuador linear são aqueles que transformam a energia hidráulica em mecânica por meio de movimentos rotativos e lineares respectivamente.

### 2.1.3 Modelagem de sistemas automáticos

De acordo com Bernard *et al.* (2008) o projeto sistemas automáticos envolve a construção de três classes de arquitetura. A arquitetura funcional é construída de acordo com as especificações funcionais e representa as ligações e interações entre diversas funções do sistema. Ela é composta das funções elementares encontradas quando as funções principais são decompostas. A arquitetura de equipamentos reflete as escolhas feitas sobre os componentes, macro sistemas ou dispositivos de comunicação que são incorporados ao sistema. A arquitetura operacional é estabelecida projetando a arquitetura funcional para a arquitetura de equipamento e é validada em termos dos objetivos estabelecidos nas especificações funcionais.

Os métodos de modelagem aplicados a sistemas complexos são realizados segundo as perspectivas estrutural, funcional e comportamental. Os modelos funcionais definem de forma inequívoca a função de cada componente no sistema e a inter-relação entre elas. Exemplos são: diagramas de circuito, Redes de Petri Canal/Agência e diagrama de Caso de Uso (UML – Linguagem de Modelagem Unificada). Os modelos estruturais representam o conjunto de elementos de um



sistema e das relações que conectam estes elementos com outros. Em engenharia os diagramas de Entidade/ Relacionamento(E/R), mapas e maquetes são exemplos desses modelos. Modelos comportamentais definem as relações de entrada, vindas do ambiente externo, com o estado interno e com a saída ou influência que exerce sobre o ambiente externo, logo possuem semântica dinâmica. Eles são classificados conforme o tipo de sinal processado: contínuo ou discreto (DE NEGRI e SANTOS, 2007).

Existem também os modelos diagramáticos. Estes são compostos por linhas e símbolos gráficos podendo representar características funcionais, estruturais e comportamentais de um sistema real (PORCIUNCULA, 2009).

As Redes de Petri (RP) tem sido utilizadas para modelar sistemas produtivos. A RP é capaz de representar de forma natural a sincronização de processos, concorrência de atividades, presença de conflitos, causalidade, compartilhamento de recursos e a exclusão mútua (RIASCOS *et al.*, 2003). A RP é uma estrutura baseada em grafos que consiste em lugares e transições. A rede simula o comportamento dinâmico de um sistema por disparar continuamente transições que são habilitadas quando fichas são removidas e inseridas em lugares determinados. Como ferramenta gráfica pode ser usada como um meio de comunicação visual. Como ferramenta matemática possibilita implementar equações algébricas e outros modelos matemáticos para determinar o comportamento do sistema (SADOU e DEMMOU, 2009).

De Negri (1996) desenvolveu um modelo de sistemas automáticos utilizando a notação em Rede de Petri Canal/Agência (C/A). Esta faz uma representação genérica de sistemas por meio de uma estrutura simples e de fácil assimilação. Belan *et al.* (2009) reitera que a rede C/A é um modelo útil para estabelecer um *link* entre diagramas de tecnologias específicas e auxiliar na comunicação entre especialistas de diversas áreas.

A rede C/A é uma representação diagramática, composta de dois elementos básicos: unidades ativas representada por retângulos e unidades passivas representadas por círculos. Esses elementos são conectados por arcos diretos que representam a direção do fluxo de recursos. Do ponto de vista funcional, as unidades passivas correspondem aos recursos que fluem no sistema. Estes podem ser energia, matéria e informação ou suas formas de manifestação como eletricidade, peças, ar comprimido, sinais, dados e outros. As unidades ativas são

atividades correspondendo às operações aplicadas nos recursos, como bombeamento, montagem, transporte e processamento (BELAN *et al.*, 2009). A figura 4 mostra a representação gráfica dos elementos básicos utilizados na construção da rede C/A.











Elementos básicos			
Símbolo	Nome genérico	Perspectiva funcional	Perspectiva estrutural
	Unidade ativa	Atividade (função)	Agência
	Unidade passiva	Recurso	Canal
Arcos direcionados (Interconexão dos elementos)			
Símbolo	Tipo de recurso		Simbologia para canal oculto
	Fluxo de informação		 Canal de informação
	Fluxo de energia		 Canal de energia
	Fluxo de matéria		 Canal de matéria
	Fluxo de energia e matéria		 Canal de energia e matéria

Figura 4 Elementos básicos da rede Canal/Agência  
Fonte: BELAN (2007)

Segundo De Negri e Santos (2007) é possível perceber uma perspectiva estrutural na rede C/A. Neste caso as unidades passivas são designadas canais, indicando os componentes que dão suporte para o fluxo dos recursos no sistema, sem modificar o seu estado. Tubulações, eixos, fios, depósitos e memórias são exemplos de canais. As unidades ativas são as agências e correspondem aos locais onde ocorrem as atividades, tais como: bombas, componentes de máquinas, células de manufatura, reatores químicos entre outros. Sob essa perspectiva a direção apontada pelos arcos que conectam os elementos não tem significado. Eles representam as interligações e indicam qual componente passivo é necessário para estabelecer a conexão entre os componentes ativos.

A regra básica para utilizar essa notação é que a interligação pode ocorrer apenas entre canais e agências, ou seja, não é permitido dois canais ou duas agências interligadas. Um arco conectando um canal a uma agência implica que a atividade pode depender do conteúdo do canal, ou seja, os recursos podem, ou não, ser utilizados e modificados na atividade. A rede C/A pode ser matematicamente representada por matrizes binárias. Esse procedimento é equivalente ao aplicado nas redes de Petri comuns (BELAN *et al.*, 2009).

Belan (2007) apresentou uma proposta de formalização para a estrutura da rede de Petri C/A, que consiste em gerar um modelo matemático equivalente à

estrutura do modelo gráfico. Os procedimentos de análise das propriedades da rede buscam: a coerência estrutural, de fluxo de recursos e de dependência entre recursos do modelo. Também são apresentados procedimentos de condensação e análise de compatibilidade entre redes. Estes métodos de análise representam um diferencial na utilização da rede C/A frente aos outros modelos que não possuem método de verificação.

Porciúncula *et al.* (2015) discorre sobre a linguagem para modelagem de comportamento discretos de sistemas sequenciais, o Gráfico Funcional de Controle de Etapas e Transições (Grafcet). A padronização da notação rendeu credibilidade, sendo aplicada em diversas áreas da indústria como sistemas hidráulicos e pneumáticos. O Grafcet possui uma estrutura de modelagem independente do tipo de tecnologia contida no sistema. Sua representação gráfica consiste em passos representados por quadrados e transições representadas por barras horizontais, ambos interligados por arcos, como ilustrado pela figura 5.

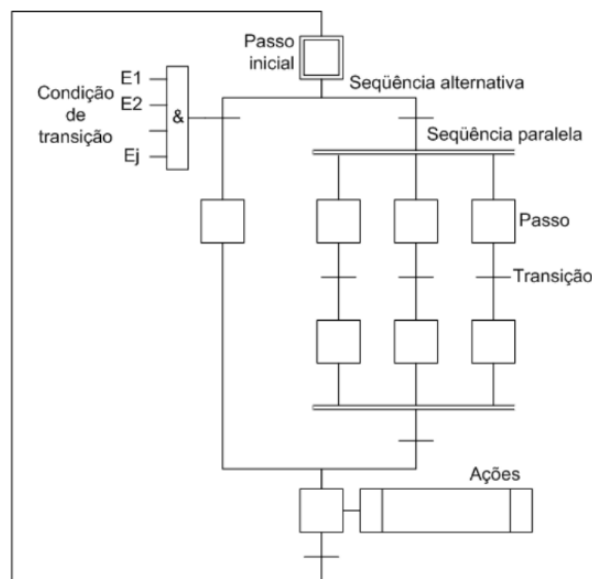


Figura 5 Elementos básicos do Grafcet  
Fonte: PORCIÚNCULA (2009)

O modelo é interpretado de cima para baixo. O passo inicial, aquele que está ativo quando o sistema é iniciado, é representado por um quadrado duplo. Várias sequências são possíveis numa única representação, como configurações de sequências alternativas, paralelas e saltos. Por Grafcet também é possível combinar diferentes formas entre os passos, isto possibilita o sincronismo, a concorrência e o paralelismo de ações (PORCIÚNCULA, 2009).

Os conceitos básicos são simples: passo, ação associada ao passo, transição e condição associada à transição. O passo representa um estado parcial do sistema, onde uma ação é realizada, e ele pode estar ativo ou inativo. A transição que conecta o passo precedente ao passo seguinte representa uma decisão para mudança de estado do sistema, ou seja, a ação do passo precedente é seguida pela ação do seguinte. Para que uma transição ocorra, são necessárias duas condições: todos os passos cuja saída está ligada à entrada da transição devem estar ativados e a condição associada à transição estar satisfeita. Ao efetuar a transição, o passo precedente fica inativo e o seguinte é ativado. Portanto, a ação associada ao passo seguinte passa a ser realizada (LOPES *et al.*, 2008).

## 2.2 Metodologias de Desenvolvimento de Produtos

De acordo com Santos (2003) a busca pelo desenvolvimento de procedimentos apropriados de projetos salienta a procura de modelos adequados para sistemas automáticos. Esses sistemas exigem o uso de procedimentos sistemáticos e habilitados para integrar e otimizar os diversos aspectos envolvidos no projeto se adaptando às tecnologias disponíveis e facilitando a comunicação e relacionamento das pessoas envolvidas, criando um processo nítido e lógico.

Segundo Belan (2007) ainda não foi concretizada uma proposta de metodologia de processo de projeto para sistemas automáticos. Por essa razão são aplicadas subdivisões conforme as fases do processo de projeto de produtos industriais baseadas principalmente na escola alemã representada por Pahl e Beitz (1998). Os modelos encontrados compreendem as fases de projeto informacional, projeto conceitual, projeto preliminar e projeto detalhado convergindo para um modelo chamado consensual.

Santos (2003) reafirma que os modelos de projeto de produto para sistemas técnicos existentes convergem para um modelo consensual. O autor destaca que as etapas de desenvolvimento são sistematizadas e caminham para alguns resultados intermediários como: especificações técnicas, estrutura funcional, *layout* preliminar, documentação detalhada e alguns outros.

Souto (2005) destaca que a dificuldade em estabelecer modelos estruturados do projeto de produtos adaptados para os sistemas automáticos está atrelada ao

aspecto multidisciplinar e multitecnológico que abrange o domínio da engenharia de controle e automação.

Na execução do projeto de sistemas automáticos não é suficiente aplicar as atividades, meios e modelos da engenharia de automação para os sistemas de controle e empregar a engenharia de produto para os sistemas físicos. É necessário integrar as diversas áreas, especialmente na fase de concepção, pois é nela que são escolhidas as tecnologias que serão utilizadas (DE NEGRI, 1996).

Um novo modelo de projeto conceitual foi elaborado por Santos (2003) procurando abordar a execução dos projetos da parte física e de controle de forma simultânea. O objetivo era conseguir a concepção do novo sistema de maneira integrada. Nessa perspectiva, ele apresenta uma proposta de descrição funcional, estrutural e comportamental utilizando redes de Petri Canal/Agência associadas a uma abordagem de modelagem e controle de sistemas a eventos discretos (SEDs), inserindo na atividade de projeto ferramentas formais de modelagem e síntese de controladores.

Em sua tese de mestrado Belan (2007) desenvolveu uma proposta para formalizar a estrutura das Rede de Petri Canal/Agência (C/A), baseada na teoria de conjuntos, funções e matrizes. Essa proposta pretende evidenciar que o formalismo inserido à estrutura da rede C/A torna possível a criação de métodos matemáticos, que auxiliam na análise da estrutura da rede e facilitam a comunicação entre as equipes de projeto. A proposta foi inserida nas diretrizes de projeto de sistemas automáticos sugeridas por De Negri (1996) e complementada por Santos (2003). As técnicas sugeridas são aplicadas a um estudo de caso para avaliar a eficiência e utilidade real dos métodos propostos. O estudo de caso consistiu na modelagem de uma pequena central hidrelétrica denominada Passo do Inferno, classificada como projeto de médio a grande porte. O estudo se desenvolveu com o intuito de tornar possível e atraente a utilização de um modelo central que facilite o entendimento do sistema e a comunicação entre equipes de projeto.

### 2.3 Confiabilidade

De acordo com Fogliatto e Ribeiro (2009) a confiabilidade está relacionada ao funcionamento bem sucedido de um produto ou sistema. Para fins de análise de engenharia deve haver uma definição quantitativa de confiabilidade em termos de probabilidade. Esta corresponde à probabilidade de um item realizar de forma

adequada a finalidade para o qual foi especificado, pelo período de tempo e condições ambientais determinadas previamente.

Huang e Chang (2007) definem confiabilidade como a probabilidade de que o sistema irá realizar a função pretendida dentro dos limites de projetos especificados. Matematicamente, a confiabilidade é a probabilidade de que um sistema será bem sucedido num intervalo de tempo determinado.

Testes acelerados de confiabilidade podem ser proibitivamente caros ou impossíveis para determinados sistemas, principalmente se muitas alternativas de projeto devem ser avaliadas. Então a modelagem da confiabilidade representa um substituto econômico. Modelos de confiabilidade podem ser usados para ajudar a definir e interpretar requisitos de um sistema e prever a confiabilidade de diferentes configurações de um sistema. Além destes, podem identificar os pontos fracos da confiabilidade de um sistema ou gargalos de confiabilidade no início do ciclo de vida do produto. Isso permite aos projetistas fazer mudanças quando elas têm um custo menor e determinar estratégias de redução de custos e manutenções de baixo custo (REIBMAN e VEERARAGHAVAN, 1991).

A definição precisa do que se percebe como desempenho adequado de um item é essencial para especificar o modelo matemático que melhor representa o desempenho do mesmo. Com base no que se entende por desempenho adequado é viável determinar quando ocorre a falha, pois quando esta acontece o item não executa de maneira adequada as suas funções. É necessário o estabelecimento de um padrão para qualificar o que se entende como desempenho adequado (FOGLIATTO e RIBEIRO, 2009).

De acordo com Reibman e Veeraraghavan (1991) o primeiro passo para avaliação da confiabilidade é determinar quais os parâmetros que serão analisados, estes devem refletir os objetivos do projeto do sistema. As informações para ajudar a identificar os parâmetros corretos podem provir de conversas com os clientes, a partir de requisitos do sistema e de uma análise cuidadosa da aplicação pretendida. Alguns parâmetros comuns incluem o tempo médio entre falhas, a disponibilidade e a quantidade esperada de falhas.

A confiabilidade aplicada a sistemas técnicos vem sendo utilizada há cerca de cinquenta anos. Nesse período as aplicações da confiabilidade se consolidaram em diversas áreas como: análise de risco e segurança, qualidade, otimização da

manutenção, proteção ambiental e projeto de produtos (FOGLIATTO e RIBEIRO, 2009).

PORCIÚNCULA (2009) afirma que a probabilidade de falha é uma medida definida pela probabilidade de falha num período, sendo que este pode assumir valores discretos como número de ciclos até falha. Entretanto, normalmente se pretende estimar a probabilidade de sobrevivência do sistema durante um certo período. Logo a medida de interesse é a função confiabilidade que é o complemento da função acumulada de probabilidade de falha. Essa função pode ser expressa por:

$$R(t) = 1 - F(t) \quad (1)$$

Derivando a função acumulada de probabilidade de falha  $F(t)$ , obtém-se a função densidade de probabilidade de falha  $f(t)$ :

$$f(t) = \frac{dF(t)}{d(t)} = - \frac{dR(t)}{dt} \quad (2)$$

Estas relações podem ser expressas na forma de integral. A Equação (3) representa a função probabilidade de falha e a Equação (4) a função confiabilidade:

$$F(t) = \int_0^t f(t) dt \quad (3)$$

$$R(t) = 1 - \int_0^t f(t) dt \quad (4)$$

Considerando que a área da função de densidade deve ser igual à unidade, então a equação de confiabilidade pode ser reescrita, conforme Equação (5):

$$R(t) = \int_{-\infty}^{\infty} f(t) dt \quad (5)$$

A taxa de falha ( $\lambda$ ) pode ser identificada na prática. Ela representa a relação entre o número de falhas ocorridas em um item num determinado período de tempo. Assim é possível determinar uma relação entre ela e a função densidade de probabilidade de falha. Assume-se que os tempos até falha dos componentes em estudo seguem uma distribuição exponencial. A taxa de falha do sistema pode ser encontrada dividindo a função densidade de probabilidade de falha pela área desta função para um tempo maior que o tempo de interesse:

$$\lambda(t) = \frac{f(t)}{\int_t^{\infty} f(t) dt} \quad (6)$$

Substituindo-se a Equação (5), obtém-se:

$$\lambda(t) = \frac{f(t)}{d(t)} \quad (7)$$

A partir da Equação (2), tem-se:

$$\lambda(t) = -\frac{1}{R(t)} \frac{dR(t)}{d(t)} \quad (8)$$

Integrando os dois lados da Equação (8), chega-se na Equação (9):

$$\int_t^{R(t)} \frac{1}{R(t)} dR(t) = \int_0^t -\lambda(t) dt \quad (9)$$

De onde se obtém a Equação (10) chamada de equação geral da confiabilidade:

$$R(t) = \exp \left[ -\int_0^t \lambda(t) dt \right] \quad (10)$$

Se considerar a taxa de falha como constante, chega-se na Equação (11):

$$R(t) = e^{-\lambda t} \quad (11)$$

As deficiências no processo de manufatura resultam em falhas precoces, concentradas no início do ciclo de vida do produto chamada fase de mortalidade infantil. As falhas incidentes na fase de vida útil ocorrem devido a condições extremas no ambiente de operação do produto e podem ocorrer de forma uniforme em qualquer momento do tempo. Na fase de envelhecimento, final da vida útil do produto, a deterioração do mesmo resulta em falhas por desgaste. A combinação dessas classificações pode ser representada de forma gráfica num gráfico conhecido como curva da banheira (FOGLIATTO e RIBEIRO, 2009).

### 2.3.1 **Confiabilidade em Sistemas Automáticos**

Os sistemas automáticos ou de automação e controle normalmente atuam em diferentes estados operacionais. Cada estado tem um conjunto definido de componentes mecânicos, hidráulicos, pneumáticos e/ou elétricos para atender as diferentes variantes da função global do sistema. No entanto a confiabilidade dos



sistemas automáticos não é obtida diretamente, como num sistema onde todos os componentes operam continuamente. A quantidade de tempo que cada estado operacional permanece operante durante o ciclo de vida do sistema determina o tempo de uso de cada componente (PORCIÚNCULA *et al.*, 2015).

Os sistemas automáticos podem possuir diversas maneiras de atender uma função. A função global pode se desmembrar em diferentes variantes da função global, que podem ser equivalentes, diferenciadas por um qualificador ou distintas. Um componente pode atender uma ou mais VFGs do sistema ou não ser solicitado, permanecendo desativado ou em *stand by*, conforme o modelo comportamental e os arranjos funcionais e estruturais (PORCIUNCULA, 2009). Essas diferentes configurações afetam o inter relacionamento dos componentes, sua utilização e também seu estado.

Segundo Marquez, Neil e Fenton (2010) a maioria dos métodos de análise da confiabilidade publicados são baseados em modelos estatísticos paramétricos e não-paramétricos dos dados de tempo até falha e métricas associadas. Esses modelos partem do pressuposto fundamental de que um modelo estatístico coerente pode ser desenvolvido, e vai se mostrar estável o suficiente para prever com precisão o comportamento do sistema ao longo do seu ciclo de vida.

Já a abordagem analítica é baseada no uso da probabilidade de falha dos componentes individuais de um sistema, com o objetivo de encontrar a probabilidade de falha total do sistema. A probabilidade de falha global do sistema não depende apenas da probabilidade de falha de suas unidades individuais, mas também da forma como estas unidades estão interligadas, a fim de formar o sistema global (DISTEFANO e PULIAFITO, 2009).

*Structured Analysis and Design Technique* (SADT) é uma ferramenta padrão usada em projeto de sistemas de manufatura integrada por computador, incluindo sistemas de manufatura flexível. Bernard *et al.* (2007) desenvolveu o chamado método Safe-SADT. Este permite a formalização funcional de interações, a identificação dos valores característicos que afetam a dependabilidade de sistemas complexos, a quantificação da confiabilidade, disponibilidade e parâmetros de segurança da arquitetura operacional do sistema. O termo dependabilidade é uma tradução do termo inglês *dependability*, que indica a qualidade do serviço fornecido por um dado sistema e a confiança depositada no serviço fornecido. A abordagem qualitativa do método aplica o formalismo do SADT para modelar os subconjuntos

funcionais, integrando as restrições do início da fase de projeto. A abordagem quantitativa valida a arquitetura operacional de acordo com as especificações, levando a dinâmica do sistema em consideração através de simulação Monte Carlo. Para testar a eficiência do método, ele foi aplicado num sistema hidráulico automatizado.

Garcia-Domínguez *et al.* (2008) introduziram uma metodologia para analisar o comportamento de sistemas tolerantes a falhas. Esses sistemas podem se adaptar e compensar por falhas de componentes de forma planejada e sistemática. Cadeias de Markov foram aplicadas para modelar o processo estocástico associado com as diferentes configurações que o sistema pode adotar. O modelo inclui não apenas o comportamento normal (ausência de falhas), mas também comportamentos degradados por componentes que falharam. Para ilustrar a metodologia é apresentado um estudo de caso de um sistema de controle de vôo direcional para um avião de caça. Essa abordagem reduz a subjetividade do julgamento do analista para aferir se uma sequência de falha dos componentes vai ou não causar a falha do sistema. Sua limitação é que para sistemas muito confiáveis, poucas simulações irão conduzir a falha do sistema.

Guo e Yang (2008) apresentaram uma técnica para criar automaticamente os modelos de Markov para avaliação da confiabilidade de sistemas instrumentados de segurança (SIS). Um SIS é um tipo especial de sistema tolerante a falhas. Políticas de reparação e causas comuns de falhas foram introduzidas no modelo de Markov. A probabilidade média de falha em demanda de uma função é determinada pelo cálculo e combinação dessa probabilidade em todos subsistemas que juntos promovem a função. A análise de Markov abrange a maioria dos aspectos que afetam a confiabilidade, apresenta maior flexibilidade que outras técnicas e descreve transições dinâmicas entre diferentes estados. Devido a modelagem automática, modelos de Markov mais complexos podem ser criados refletindo situações reais mais detalhadamente.

Distefano e Puliafito (2009) propuseram o Diagrama de Blocos de Confiabilidade Dinâmicos (DRBDs). A abordagem consiste em ampliar o formalismo do Diagrama de Blocos de Confiabilidade (RBD) para definir uma notação capaz de representar uma ampla classe comportamentos dinâmicos/dependentes. A dependência representa e quantifica a influência de uma unidade em outra unidade.

Isso permite implementar a abordagem modular, ou seja, um comportamento dinâmico complexo pode ser expresso numa composição de dependências simples. Os DRBDs fornecem as regras e mecanismos para compor e combinar o efeito de dependências simples, a fim de representar o comportamento dependente requerido. A principal contribuição do método é a capacidade de modelar dependências entre subsistemas ou unidades, conforme o comportamento da sua confiabilidade. Para ilustrar a metodologia, ela foi aplicada na análise de um sistema multiprocessador de computação distribuída e num sistema de suprimento de água.

Huang e Chang (2007) também utilizaram a abordagem modular num esquema melhorado de decomposição. O comportamento de modularização dos módulos dinâmicos foi chamado de “decomposição”. Nesse esquema as sub-árvores dinâmicas, de uma árvore de falha dinâmica (DFT), podem ser ainda mais modularizadas se existirem algumas sub-árvores independentes na mesma. Isso reduz o espaço de estados resultantes no modelo de Markov, e portanto melhora o tempo de processamento. Além disso, essa abordagem mantém a capacidade de avaliar as sensibilidades que compõem os eventos básicos da DFT que são eliminadas pelo método modularização. Um exemplo da implementação foi realizado num sistema de assistência cardíaca (CAS). Essa abordagem deu origem a um *kit* ferramenta para software, o DyFA (*Dynamic Fault-trees Analyzer*).

Fazlolshtabar *et al.* (2015) propuseram um estudo de caso onde implementaram uma rede neural *backpropagation* integrada a um modelo Markoviano para cálculo da confiabilidade. Embora, um modelo Markoviano seja proposto para avaliação da confiabilidade, um modelo de rede neural foi desenvolvido para evidenciar a diferença na precisão do modelo de Markov em comparação com a rede neural. O estudo de caso foi extraído de um ambiente industrial. Foi considerado um sistema de manufatura *Jobshop* tendo múltiplos AGVs (Veículos Guiados Automaticamente), para o manuseio de materiais. O objetivo da otimização proposta foi maximizar a confiabilidade total dos veículos guiados automaticamente (AGV) e minimizar o custo total de reparo no sistema.

Alguns estudos recentes focaram no uso de Redes Bayesianas para fornecer uma estrutura unificada para modelagem e análise da confiabilidade de sistemas. Cai *et al.* (2012) realizaram o estudo da confiabilidade de um sistema BOP *Stack* (Preventor de explosão) numa plataforma de petróleo. A BOP *Stack* realiza um papel importante proporcionando segurança para as atividades de *drilling*. Falhas nesse

sistema podem causar acidentes catastróficos. O estudo focou nos modelos das redes Bayesianas para avaliar a confiabilidade do sistema de controle de um BOP submarino. O sistema de controle do BOP submarino é muito complexo, então um software para redes Bayesianas, MSBNx [32], foi utilizado como ferramenta para construir o modelo e avaliar a confiabilidade.

Marquez *et al.* (2010) apresentaram um método de modelagem de rede Bayesina (RB) híbrida baseada em eventos simples, que é adaptável para sistemas grandes e complexos. A ideia da abordagem é definir o tempo até falha da árvore de falha, como função determinística dos tempos até falha dos componentes de entrada correspondentes. Um poderoso algoritmo de inferência aproximada para RB híbridas foi incorporado, baseado no processo de discretização dinâmica do domínio de todas as variáveis contínuas na RB. Variáveis discretas randômicas foram incluídas no modelo para representar o estado do sistema (ou qualquer subsistema) num período de tempo específico. Esse método foi aplicado no módulo da CPU de um sistema hipotético de assistência cardíaca (HCAS), concebido para tratar falhas mecânicas e elétricas no coração. Os resultados foram aproximados dos obtidos por representações analíticas e obtidos com menos esforço.

Riascos *et al.* (2003) propuseram uma metodologia, em dois passos, para detecção e tratamento de falhas em máquinas automáticas. O primeiro passo consiste na construção do módulo que representa a detecção e tratamento de falhas, definido como "O módulo de diagnóstico". O segundo é a construção de um módulo representando o processo produtivo da máquina, definido como "O módulo operacional". Na construção do módulo de diagnóstico, uma atenção especial é dedicada às Redes de Petri comportamentais (BPN) e redes Bayesianas. As redes de Petri lidam com sequências de estados discretos e as redes Bayesianas lidam com incertezas probabilísticas. Um estudo de caso em AGVs foi considerado. Essa metodologia também foi aplicada em Riascos (2002).

Redes de Petri são extremamente úteis para avaliação da confiabilidade e descrição dos sistemas. Chu *et al.* (2015) propuseram um método de modelagem e simulação para a análise do comportamento dinâmico de sistemas. A teoria das Redes de Petri Estocásticas Generalizadas (GSPN) foi aplicada na modelagem dinâmica, simulação e cálculo de parâmetros da confiabilidade de uma servo-válvula. Servo-válvulas eletro hidráulicas são um tipo de componente hidráulico de

precisão, capaz de converter proporcionalmente um sinal elétrico de controle numa resposta correspondente de fluxo e pressão de fluido hidráulico. Com o objetivo de verificar a efetividade do método GSPN na análise da confiabilidade dinâmica de uma servo-válvula, o método de Markov foi introduzido. Os resultados dos experimentos de simulação foram ajustados usando o MATLAB.

Sadou e Demmou (2009) desenvolveram um método de análise da confiabilidade em sistemas dinâmicos, que utiliza lógica linear aplicada a redes de Petri. Uma forma clássica de obter cenários em redes de Petri é gerar o gráfico de acessibilidade, entretanto, para sistemas complexos ele leva à explosão de estados. Para evitar este problema bases lógicas lineares são introduzidas e utilizadas para definir formalmente cenários e minimalidade de cenários. Estas definições permitem que o método produza apenas cenários pertinentes. A abordagem baseada em lógica linear transpõe o problema de acessibilidade num simples problema de provas sucessivas. Além disso, é possível extrair da árvore de provas, informações sobre a ordem parcial entre os disparos das transições de interesse. O método foi descrito e ilustrado através do estudo de caso de um sistema de trem de pouso.

Porciúncula *et al.* (2005) estimaram a confiabilidade de uma Plataforma Hidráulica Proporcional, modelando o sistema em rede de Petri Canal/Agência. Isso permitiu trabalhar com os módulos separadamente, identificando e analisando os parâmetros de confiabilidade de cada um. Para obter a confiabilidade total do sistema foi construída uma árvore de falhas a partir da rede de Petri, o que facilitou a consideração das interações entre os módulos. A identificação dos itens do sistema e a definição das relações possibilitaram sistematizar o FMEA dos componentes. A aplicação do FTA junto com o FMEA ajudou, de forma fundamental, nesse tipo de análise. A distribuição das falhas também pode ser considerada, o que ajuda na tomada de decisão a respeito do modo de falha que tem a maior probabilidade de causar efeitos indesejados no sistema.

Porciúncula *et al.* (2015) apresentaram os conceitos e métodos, baseados na rede C/A, que permitem a determinação sistemática das configurações operacionais de cada estado operacional do sistema. A partir dessas configurações, foi determinada a confiabilidade de um sistema eletro hidráulico de um equipamento automático. As atividades ocorreram na fase conceitual do processo de projeto, quando os diagramas de circuito e gráficos seqüenciais estão sendo criados.

## 2.4 Metodologia para Análise de Confiabilidade em Sistemas Automáticos (MACSA)

A Metodologia para Análise de Sistemas automáticos (MACSA) foi desenvolvida por Porciúncula (2009) com o intuito de sistematizar a análise da confiabilidade considerando os diferentes estados operacionais e tecnologias envolvidas nos sistemas automáticos. A figura 6 mostra como a MACSA está inserida na fase conceitual de projeto de produto.

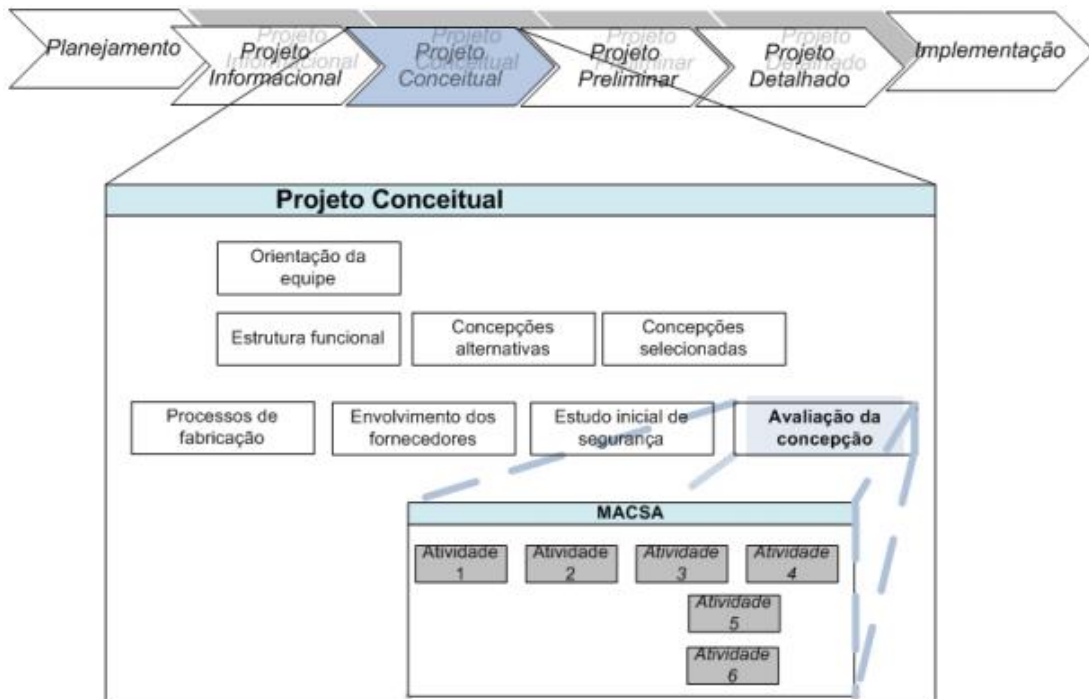


Figura 6 A MACSA inserida no projeto de produto  
Fonte: PORCIÚNCULA (2009)

Na fase de projeto conceitual as especificações de projeto são consumidas e transformadas em conceitos funcionais do produto. As decisões tomadas nessa fase vão afetar intensamente as fases seguintes, por essa razão essa etapa de projeto é considerada a mais importante. As principais atividades desta fase são: estabelecer a estrutura funcional, desenvolver as concepções alternativas do produto e selecionar e avaliar a concepção do produto.

A tomada de decisão sobre os atributos confiabilidade e manutenção é mais eficaz e tem menor custo quando executada nas fases iniciais do processo de projeto (PORCIÚNCULA *et al.*, 2015).

Essa metodologia é aplicada nas fases iniciais de projeto onde o projetista trabalha com as concepções do projeto e as informações estão refletidas por

modelos gráficos. Em sistemas pneumáticos e hidráulicos os fundamentos adotados para desenvolver as concepções de projeto geralmente consistem em fazer uso de componentes já disponíveis no mercado e a principal função do projetista é escolher e adaptar o arranjo dos componentes para que estes desempenhem as várias funções do sistema.

Essa metodologia captura as informações do modelo funcional do sistema de forma estruturada, definindo a relação entre os componentes por meio da árvore de sucesso. Além disso, as árvores de sucesso deverão ser geradas de acordo com a configuração operacional definida pelo modelo comportamental do sistema. Assim, o objetivo é realizar uma análise do atributo de confiabilidade com menor tempo, maior facilidade e com estimativa de confiabilidade teórica coerente com a confiabilidade apresentada na prática.

A partir dos modelos resultantes que representam as configurações operacionais do sistema, é possível executar a estimativa da confiabilidade. Um problema comum na análise de confiabilidade é considerar a interligação existente entre os componentes do sistema. Os tipos de interconexão, como em série ou em paralelo, são obtidos observando manualmente os diagramas funcionais, tais como circuitos elétricos ou hidráulicos ou modelos estruturais, tais como desenhos mecânicos (PORCIÚNCULA *et al.*, 2015).

A MACSA abrange seis atividades que estão representadas na figura 7. Para prosseguir estas atividades é essencial que usuário relacione as informações do projeto sob dois aspectos: (1) Informações sobre a visão funcional/ estrutural do sistema, obtida a partir da modelagem do sistema em Rede C/A. (2) Informações sobre a visão comportamental do sistema, discernindo os possíveis estados que o sistema pode assumir no decorrer de sua operação. Estes estados devem estar relacionados com as VFGs.

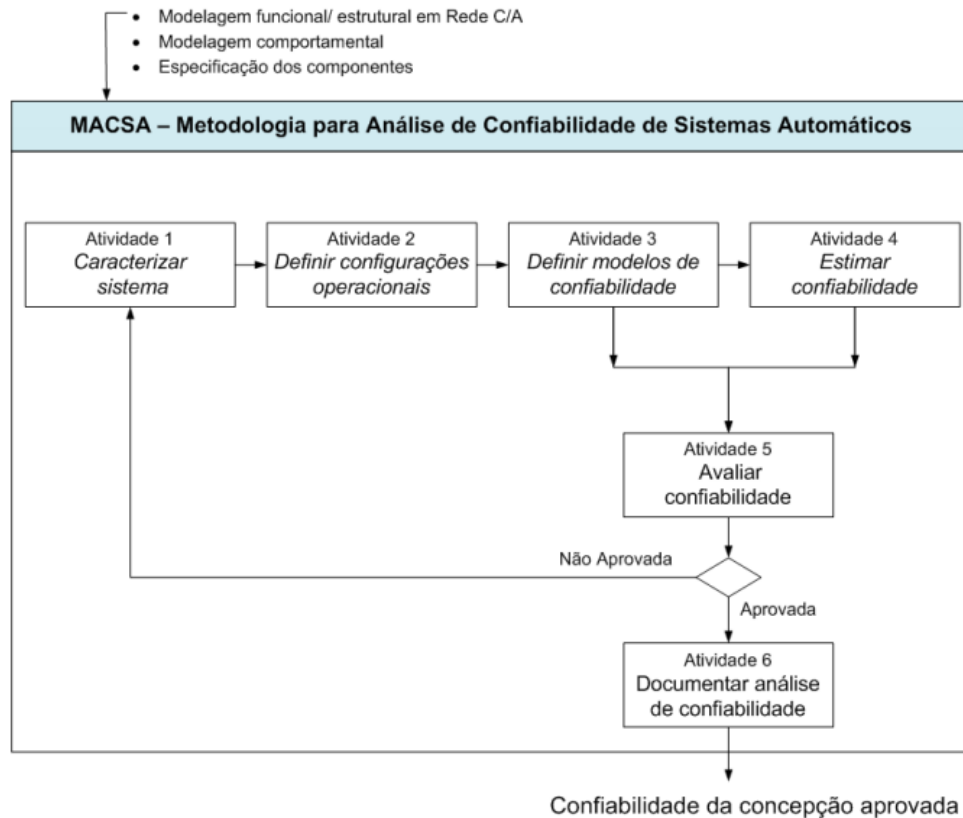


Figura 7 Representação das atividades da MACSA  
 Fonte: PORCIÚNCULA (2009)

Para o desenvolvimento dessa metodologia, essas atividades foram subdivididas em tarefas que são decompostas em etapas específicas. Ao final, ela tem como saída a aprovação do atributo confiabilidade.

A aplicação da rede de Petri C/A permite que o modelo estrutural/ funcional definido no projeto, seja convertido em modelos matemáticos que são manipulados de acordo com os diferentes estados operacionais. Dessa forma é possível calcular a confiabilidade considerando os diferentes desdobramentos das VFGs do sistema.

A Rede de Petri proporciona a representação matricial, a qual facilita a implementação computacional do sistema e sua verificação automática. Nessas matrizes, as colunas são mapeadas pelas transições e as linhas pelos lugares da rede. Quando não existe relação de precedência ou procedência o valor do elemento na matriz é nulo.

A definição das diferentes configurações operacionais do sistema automático terá por base a manipulação matricial do modelo funcional e estrutural. Os modelos em Rede C/A passam pelos métodos de análise de coerência estrutural e o método de análise de coerência de fluxo de recurso, definidos por Belan (2007):



- Coerência estrutural: tem a finalidade de verificar se o modelo está correto estruturalmente, ou em outras palavras, se não existe nenhuma ligação canal/canal ou agência/agência e se os arcos estão coerentes com a classe dos recursos que o percorrem.
- Coerência de fluxo de recursos: tem como objetivo verificar se os recursos que percorrem a rede apresentam um fluxo coerente, ou seja, todo recurso que entra no sistema modelado, deve poder sair e vice e versa.

O método de Análise da Árvore de Sucesso (STA –*Sucess Tree Analysis*) é um complemento da Análise da Árvore de Falha. Uma operação de um sistema pode ser considerada sob dois pontos de vista: 1) as diferentes possibilidades de sucesso e 2) as diferentes possibilidades de falha do sistema. Esta metodologia optou pelas árvores de sucesso, pois o sucesso do sistema está focado em atender o evento de topo, ou seja, atender uma das VFG do sistema. As principais razões de se utilizar as árvores de sucesso são:

- A análise da confiabilidade do sistema deverá ser desenvolvida na fase conceitual do processo de projeto, quando o agente de projeto está interessado em encontrar soluções.
- A estrutura proposta para gerar as árvores de sucesso segue a mesma lógica de busca de soluções do problema.
- As árvores de sucesso poderão absorver de forma direta as informações geradas nos modelo funcional, estrutural e comportamental que são utilizados na fase do processo de projeto.
- Organizar, armazenar e disponibilizar de forma adequada informações geradas na fase inicial do processo de projeto.
- Possibilidade de estimar a confiabilidade a partir da concepção do sistema.

Uma das vantagens da MACSA é que as informações que abastecem a metodologia são provenientes de modelos técnicos desenvolvidos nas fases iniciais do processo de projeto. Isso facilita a utilização da metodologia, diminui o tempo de análise e evita a interferência do agente de projeto na análise, conduzindo a tomadas de decisão mais rápidas, livre de erros e vícios. Além disso, nas fases

inicias há maior flexibilidade e alterações na concepção do produto afetarão os modelos técnicos que irão alterar diretamente os dados de entrada na MACSA.

A metodologia não é limitada pelo tamanho do sistema e considera situações mais realistas para a estimativa da confiabilidade, pois são consideradas as diferentes configurações estruturais e a probabilidade de utilização dessas durante a vida útil do sistema. A análise do uso efetivo dos componentes dá subsídios ao projetista para decidir sobre a necessidade de componentes sobressalentes. O emprego da Rede C/A permite a análise de uma vasta gama de sistemas técnicos, independente da sua área tecnológica.

Devido a MACSA empregar a Rede C/A e o Grafcet, modelos que podem ser representados na forma matricial, ela pode ser implementada computacionalmente. Por conseguinte, existe a possibilidade de utilizá-la como uma ferramenta de auxílio aos projetos de sistemas automáticos.

## 2.5 Modelagem de Software

De acordo com Booch, Rumbaugh e Jacobson (2005) a modelagem é uma parte central das atividades que conduzem a implantação de um bom *software*. A modelagem permite alcançar quatro objetivos: visualizar o sistema como ele é ou como ele deve ser, especificar a estrutura ou o comportamento do sistema, proporcionar um guia para a construção do sistema e documentar as decisões tomadas. Os modelos também são úteis para compreender melhor o sistema em elaboração e expor oportunidades de simplificação e reaproveitamento. Além disso, existem limites para a capacidade humana compreender complexidades, com a modelagem delimitamos o problema restringindo o foco a um aspecto por vez.

A Linguagem de Modelagem Unificada (UML) é uma linguagem de modelagem gráfica aplicada para descrição e documentação de projetos de *software*. Ela torna o complexo processo de análise, projeto e construção de *software* mais simples, criando visões do sistema que está em construção (TONSIG, 2008).

A UML é uma linguagem muito expressiva, abrangendo todas as visões necessárias ao desenvolvimento e implantação de sistemas. Ainda assim, não é uma linguagem difícil de compreender e utilizar. A UML tem sido implementada de forma efetiva em diversas áreas como: sistemas de informação corporativa, serviços

bancários e financeiros, telecomunicações, transportes, defesa/espço aéreo, vendas de varejo, eletrônica médica, científicos e serviços distribuídos baseados na Web (BOOCH *et al.*, 2005). A UML independe da linguagem de programação e do processo de desenvolvimento. Isso significa que ela pode ser utilizada para a o modelagem de sistemas não importando qual a linguagem de programação utilizada na implementação e nem o processo de desenvolvimento adotado (BEZERRA, 2015).

Segundo Tonsig (2008) o processo de desenvolvimento de *software* não está previsto na UML. No entanto podem-se eleger cinco etapas de desenvolvimento de software em que a UML pode ser aplicada: análise de requisitos, análise sistêmica, projeto, implementação e testes/implantação.

### 2.5.1 **Análise de Requisitos**

Essa deve ser a primeira etapa a ser desenvolvida, pois nela serão reunidos os subsídios necessários para as demais. Durante o levantamento de requisitos, a equipe de desenvolvimento tenta entender os desejos do usuário com relação ao *software* a ser desenvolvido para determinar o domínio que deve ser automatizado pelo sistema. O levantamento de requisitos compreende também um estudo exploratório das necessidades dos usuários. Para isso várias técnicas podem ser utilizadas: leitura de obras de referência, observação do ambiente do usuário, entrevistas com usuários, comparação com sistemas preexistentes do mesmo domínio (BEZERRA, 2015). A partir desse levantamento é possível fazer a modelagem das atividades encontradas utilizando o diagrama de casos de uso (TONSIG, 2008).

Um diagrama de caso de uso exibe um conjunto de casos de usos, atores e seus relacionamentos, ele abrange uma visão estática do sistema. Esses diagramas são importantes principalmente para a organização e modelagem comportamental do sistema. Um caso de uso descreve uma sequência de ações realizadas pelo sistema que gera resultados observáveis de valor para um determinado ator. Graficamente ele é representado por uma elipse com linhas contínuas, geralmente contendo seu nome. Um caso de uso pode acionar outro para poder utilizar um procedimento existente, essa associação pode empregar os esteriótipos <<*include*>> ou <<*extende*>>. O primeiro indica que será obrigatório que um caso de uso acione o comportamento expresso pelo outro caso. O segundo indica que um

caso de uso é uma variação de um outro a ele associado (BOOCH *et al.*, 2005). A figura 8 mostra uma representação do Diagrama caso de uso.

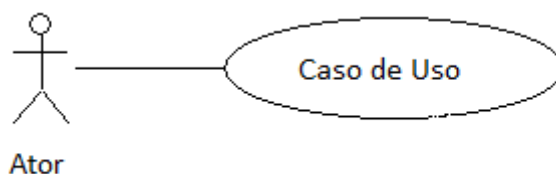


Figura 8 Representação do Diagrama de Casos de Uso  
Fonte: Autor

Na UML, a título de generalizar o conjunto de “coisas” que interagem diretamente com um *software*, se utiliza o estereótipo “ator”. Assim, as pessoas, departamentos e outros sistemas são entidades externas ao *software* e por desempenharem algum papel relevante ao sistema, são chamados atores externos. Nessa etapa também podem ser usados os *storyboards*. Eles são roteiros textuais descritos livremente, identificando os agentes de um cenários e as ações que deles participam. Os mesmos cenários apresentados pelos *storyboards* podem ser representados pelos diagramas de caso de uso. A relação entre atores e sistemas está vinculada a funcionalidade do software, assim é possível perceber previamente o que deverá existir no *software*, sem se preocupar ainda em como isto será implementado (TONSIG, 2008).

### 2.5.2 **Análise Sistêmica**

Nessa etapa serão estudados todos os dados e processos verificados na fase anterior, fazendo abstrações para identificação das classes, seus atributos e métodos. Só serão modeladas classes que pertençam ao domínio principal do problema do *software*. Essas classes deverão ser retratadas de forma que se visualize a estrutura e forma como deverão interoperar, para isso pode ser utilizado o Diagrama de Classes (TONSIG, 2008).

Um diagrama de classe apresenta um conjunto de classes, interfaces, colaboradores, e também seus relacionamentos. Graficamente ele é uma coleção de vértices e arcos. Esse diagrama abrange uma visão estática da estrutura do sistema, já que o que é descrito é válido em qualquer ponto no ciclo de vida do sistema (BOOCH *et al.*, 2005). A figura 9 mostra a representação gráfica do diagrama de classes.



Figura 9 Representação do Diagrama de Classes  
Fonte: Autor

A classe de objeto é representada por um retângulo, subdividido em três regiões. Na primeira consta o nome da classe, na segunda atributos e na terceira seus métodos (funções/procedimentos). Uma classe corresponde a um conjunto de elementos que possuem mesma estrutura e comportamento. Elas classificam objetos do mundo real, assim devem ser extraídas do domínio do problema e ser nomeadas pelo que representam no sistema (TONSIG, 2008).

As classes podem apresentar quatro tipos de relação: generalização, realização, dependência e associação. A generalização é um relacionamento onde os objetos dos elementos especializados (os filhos) são substituíveis por objetos do elemento generalizado (os pais), assim os filhos compartilham a estrutura e comportamento dos pais. Esse relacionamento é representado por uma linha sólida com uma seta vazada apontando para o pai (figura 9). A realização é um relacionamento em que um classificador especifica um contrato que outro classificador garante executar, esse relacionamento é encontrado entre interfaces e as classes ou componentes que as realizam e entre casos de uso e as colaborações que os realizam. Representado por uma linha tracejada com seta branca. A dependência é um relacionamento entre dois itens, no qual a alteração de um item (dependente) pode afetar a semântica do outro (independente). Ela é representada por linhas tracejadas, podendo ter setas e incluir rótulos. Uma associação descreve um relacionamento estrutural entre classes que decreta um conjunto de ligações, estas são conexões entre objetos que são instâncias das classes. Ela é representada por linhas sólidas, podendo ser direcionadas e incluir rótulos e outros adornos (BOOCH *et al.*, 2005).

Tonsig (2008) descreve ainda o relacionamento de agregação. Este é utilizado para mostrar que um objeto é formado pela agregação de outros objetos, ele também é chamado de relacionamento “todo-parte”. Pode ser representado por um losango cheio na agregação de valor, onde o tempo de vida das partes depende do tempo de vida do todo, ou losango vazado na agregação por referência, onde o

tempo de vida das partes não é mutuamente dependente. Esse relacionamento é um tipo especial de associação.

Uma interface é uma coleção de operações que especificam serviços de uma classe ou componente. Então, ela descreve um comportamento visível externamente desse elemento. A interface define um conjunto de especificações de operações (assinaturas), mas não um conjunto de implementações de operações. Ela é declarada de forma semelhante a uma classe, com a palavra <<interface>> acima do nome. É importante ressaltar que ela nunca aparece sozinha, pois é fornecida por uma classe ao mundo externo (BOOCH *et al.*, 2005).

Todos os objetos descobertos para modelagem de certa realidade podem ser classificados por três classes estereotipadas: *boundary*, *control* e *entity*. A classe *boundary* representa uma interface. Exemplos são: uma tela, formulários, páginas html do sistema. A classe *control* representa o mapeamento da gestão intermediária entre as classes *boundary* e o modelo de persistência de dados. A classe *entity* representa tudo que é persistente no sistema. Nela devem ficar apenas os métodos que contribuem para o armazenamento e recuperação dos dados e outros eventuais que contribuam para a consistência do modelo. Um desenho das classes estereotipadas pode ser derivado para obter de forma mais fácil o modelo final das classes, em fase de projeto (TONSIG, 2008).

Um diagrama de interação consiste num conjunto de objetos ou papéis, incluindo as mensagens que podem ser trocadas entre eles. Esse diagrama abrange a visão dinâmica do sistema e é uma junção dos diagramas de sequência e colaboração/ comunicação. Um diagrama de sequência enfatiza a ordem temporal das mensagens para que se cumpra um determinado papel ou se atenda determinado contexto (BOOCH *et al.*, 2005).

Um diagrama de sequência possui duas dimensões, uma vertical que representa o tempo e uma horizontal que representa os diferentes objetos. Ele mostra como utilizar os métodos definidos no diagrama de classes para suprir o sistema com uma funcionalidade prevista no diagrama de casos de uso (TONSIG, 2008).

O diagrama de comunicação enfatiza a organização estrutural dos objetos ou papéis que enviam e recebem as mensagens (BOOCH *et al.*, 2005). Esse diagrama não mostra a dimensão de tempo, devido a isso as sequências de mensagens e

linhas concorrentes devem ser determinadas usando uma sequência de números (TONSIG, 2008).

O diagrama de estados abrange uma visão dinâmica de um objeto. Eles são importantes para dar ênfase a comportamentos de um objeto ordenados a eventos, o que ajuda para a modelagem de sistemas reativos. São importantes principalmente para modelagem de comportamentos de uma interface, classe ou colaboração (BOOCH *et al.*, 2005).

O início de um Diagrama de Estados é indicado pelo estado inicial, representado por círculo preenchido. Ele não expressa um estado específico, apenas o início do diagrama. Na sequência ele se conecta ao primeiro estado real com uma transição, rotulada ou não. Um estado demonstra uma situação no tempo de algum aspecto do sistema, sobre o qual se deseja ter controle. Um objeto permanece num estado por um tempo finito. O estado final representa o término do ciclo de controle previsto para mudanças de estado de um dos aspectos do sistema. Para passar de um estado para outro, são necessários dois mecanismos: condição e ação. A condição deverá ser satisfeita para que ação seja executada, se ela existir. A ação é responsável pela transição dos estados (TONSIG, 2008).

### 2.5.3 **Projeto**

No projeto se extrapola o domínio central do problema do *software*. Podem ser adicionadas outras classes ao modelo para propiciar uma infra-estrutura tecnológica, como interfaces, gerenciamento de banco de dados, comunicação com outros sistemas, entre outros (TONSIG, 2008). Na fase de projeto é determinado como o sistema vai funcionar para atender aos requisitos, conforme os recursos tecnológicos existentes. Essa fase produz uma descrição computacional do que o *software* deve fazer de uma forma coerente com a descrição feita nas análises anteriores (BEZERRA, 2015).

Tonsig (2008) também afirma que a UML prevê que se faça uma documentação da organização física do *software* que será gerado. Para isso podem ser utilizados os diagramas de artefatos ou componentes e de implantação ou distribuição. Essa documentação refletirá o mundo físico e vai considerar a existência de componentes, como código-fonte, executáveis ou bibliotecas.

O diagrama de componentes mostra as dependências entre componentes de software. Ele apresenta uma visão estática de como o sistema será implementado e

quais seus módulos de *software*, ou seja, seus componentes. Eles são utilizados junto com o diagrama de implantação e mostram como as classes e componentes se complementam. O diagrama de implantação mostra as configurações dos nós de processamento em tempo de execução e os componentes neles existentes. Esse diagrama abrange uma visão estática da implantação de uma arquitetura. Um nó inclui um ou mais artefatos, ele é um elemento físico existente em tempo de execução que representa um recurso computacional, geralmente com pelo menos alguma memória e, geralmente capacidade de processamento (BOOCH *et al.*, 2005).

Essa etapa consiste num aprimoramento da etapa anterior e seu resultado será um detalhamento das especificações para tornar possível a programação do *software* (TONSIG, 2008).

#### 2.5.4 **Implementação**

Na implementação o sistema é codificado, ou seja, ocorre a tradução da descrição computacional obtida nas etapas anteriores em código executável por intermédio do uso de uma ou mais linguagens de programação. A implementação envolve a criação de um código-fonte correspondente às classes de objetos do sistema, utilizando linguagens como C#, C++, Java, PHP, Python, Ruby entre outras (BEZERRA, 2015).

A codificação dos programas de computador pode acontecer de forma automática. Existem softwares do tipo CASE (Engenharia de Software auxiliada por Computador) que realizam a conversão automática do modelo de classe para o código da linguagem de programação escolhida. Porém eles geram apenas a espinha dorsal do código, existindo a necessidade de intervenção manual para a criação do software. As etapas anteriores criam modelos que traduzem tecnicamente o significado do entendimento da estrutura do sistema. Assim a programação é o desfecho em que os modelos criados ganham vida (TONSIG, 2008).

#### 2.5.5 **Testes e Implantação**

O *software* deve passar por testes exaustivos e rigorosos na busca contínua por erros e sua eliminação. Nessa etapa quatro aspectos devem ser abordados. O primeiro são os testes de unidade, nele cada programa é testado individualmente. O



segundo aspecto é um teste do conjunto, pois não existem garantias que unidades funcionando individualmente irão funcionar em conjunto. O terceiro é um teste de integração quando o *software* criado tiver algum mecanismo de interface com outros sistemas. O quarto e último aspecto é o teste de adequação aos requisitos, com o envolvimento do usuário que dará a aprovação final (TONSIG, 2008).

Os testes de unidades, de integração e de sistemas são realizados pelos próprios desenvolvedores ou uma equipe independente de testadores. Os testes de aceitação, no entanto, são realizados pelos usuários da aplicação. Assim como os testes de sistemas, os testes de aceitação objetivam apurar a aderência do sistema desenvolvido aos requisitos definidos previamente (BEZERRA, 2015).

Após o teste de adequação e aprovação pelo usuário do *software*, este estará apto para a implantação (TONSIG, 2008). Na implantação o *software* é empacotado, distribuído e instalado no ambiente do usuário. São escritos os manuais, carregados arquivos e importados dados para o sistema, e os usuários são treinados para utilizá-lo corretamente (BEZERRA, 2015).

### 3 METODOLOGIA

O presente trabalho pode ser caracterizado como um processo de desenvolvimento de produto, neste caso um aplicativo para análise de confiabilidade de sistemas automáticos segundo a MACSA. O *software* irá se caracterizar como uma variante de um produto existente.

Segundo Back *et al.* (2008), essa classificação inclui extensões de linha, reposicionamento de produtos em termos de seu uso e mercado, formas novas, versões modificadas e novas embalagens de produtos existentes. Já existem diversos softwares cuja função é executar o cálculo da confiabilidade, como o ProConf por exemplo. A ferramenta desenvolvida neste trabalho representa uma forma nova de estimar a confiabilidade, caracterizando também um reposicionamento de mercado pois será desenvolvida para atingir o nicho específico dos projetistas de sistemas automáticos.

O trabalho também vai abranger um estudo de caso. Segundo Gil (2002) um estudo de caso é uma modalidade de pesquisa amplamente utilizada e consiste no estudo profundo e exaustivo de um ou poucos objetos, de maneira que permita seu amplo e detalhado conhecimento.

A metodologia do presente trabalho irá implementar as atividades definidas pela Metodologia para Análise da Confiabilidade em Sistemas Automáticos (MACSA) na forma de uma ferramenta computacional, como mostra a figura 10.

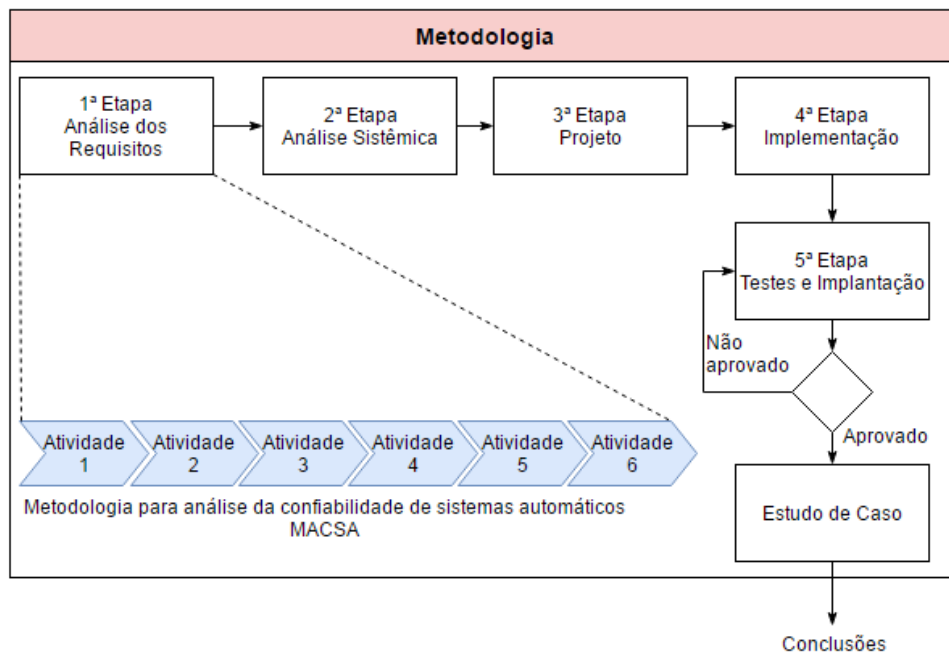


Figura 10 Etapas da Metodologia  
Fonte: Autor

A atividade 1 será realizada parcialmente de forma manual. A conversão automática dos modelos representativos em modelos matemáticos não será implementada na ferramenta devido a motivos já delineados na sessão Limitações. Esta conversão será um passo chave para o cálculo da confiabilidade. Esses modelos serão um dos *inputs* que vão alimentar a ferramenta que realizará o cálculo da confiabilidade conforme os passos estabelecidos pela metodologia. A atividade 5 depende do julgamento do projetista e requisitos do projeto, porém os gráficos e relatórios fornecidos pela ferramenta irão servir de sustentação para a tomada de decisão nesta atividade.

Para modelagem e desenvolvimento da ferramenta serão seguidas as etapas de modelagem baseadas em UML relatadas por Tonsig (2008), e aplicadas as ferramentas indicadas pelo autor.

### 3.1 **1º Etapa - Análise de Requisitos**

Nesta etapa será realizado um levantamento dos requisitos necessários ao software a ser desenvolvido. A partir disto será determinado o domínio que deve ser automatizado pelo sistema. Após a identificação dos requisitos, as atividades encontradas serão modeladas utilizando o diagrama de casos de uso. Desta forma será possível visualizar as funcionalidades que deverão existir no *software*.

Esta etapa será documentada por meio da modelagem dos diagramas de casos de uso.

O levantamento de requisitos compreende um estudo exploratório das necessidades dos usuários. Assim sendo, as atividades da MACSA serão utilizadas para alimentar este estudo. A seguir consta uma descrição detalhada de todas as atividades necessárias para cálculo da confiabilidade utilizando a MACSA.

#### 3.1.1 **Atividade 1 - Caracterizar o sistema**

O objetivo desta atividade é extrair dos modelos estrutural/funcional e comportamental as informações importantes para definição das diferentes configurações operacionais do sistema. Para esta finalidade, é adotada como estratégia de análise a manipulação do modelo funcional/estrutural em Rede C/A. Esta análise é realizada a partir de técnicas de análise estrutural somadas a informações capturadas do modelo comportamental. Esta atividade é detalhada em quatro etapas que são desdobradas em tarefas (PORCIÚNCULA, 2009).

Os diagramas de circuito são modelos funcionais, então pode ser feita uma correspondência direta entre estes e a rede C/A. Essa equivalência existe onde os componentes elétricos, eletrônicos, pneumáticos e hidráulicos são considerados agências. Os fios, tubulações, cabos e hastes são considerados canais. Assim como as Redes de Petri, as redes C/A podem ser representadas matematicamente em matrizes. Operadores lógicos podem ser aplicados e implementados métodos de análise. Da Rede C/A são extraídas duas matrizes que servem de base para essa metodologia. A matriz de precedência relaciona a precedência de canais para cada agência, e a de posterioridade relaciona os canais posteriores a cada agência. Os canais são organizados nas linhas e as agência nas colunas de cada matriz. Os valores de '1' e '0' simbolizam a existência ou não de relação entre cada canal e agência (PORCIÚNCULA *et al.*, 2015).

A identificação dos canais consumidores e fornecedores é realizada aplicando a análise de coerência de recursos, proposta por Belan (2007), nas matrizes obtidas a partir dos modelos em rede C/A. O objetivo principal desta análise é obter um modelo coerente quanto à estrutura e também gerar uma lista com os canais e agências fornecedoras e consumidoras de recursos. Estas são pré-requisitos para os outros processos de análise das propriedades da rede.

A primeira etapa da análise estrutural consiste em mapear quais são os elementos de suprimento e/ou consumo de recursos de uma rede C/A, ou seja, descobrir quais são os canais e agências que fornecem e/ou consomem os recursos que fluem pela Rede C/A. Estes elementos de suprimento e/ou consumo de recursos são definidos como elementos limites, pois representam de certa forma os limites iniciais e/ou finais para o fluxo dos recursos (BELAN, 2007). Essa análise terá como resultado um vetor coluna resultante e um vetor linha resultante. No vetor coluna resultante os canais que estiverem representados por '1' são considerados canais fornecedores e por '-1' são os canais consumidores. No caso do vetor linha resultante, as agências que estiverem representadas por '1' são consideradas as agências consumidoras e por '-1' são as agências fornecedoras (PORCIÚNCULA, 2009).

Em seguida são reunidas informações relativas a cada estado operacional do sistema, as quais estão representadas no modelo comportamental. Entre essas informações estão: código do estado, nome do estado, função relacionada ao estado

e a probabilidade de utilização do estado. Esta última informação é decorrente da expectativa do projetista ou do futuro usuário do equipamento quanto à intensidade de utilização do mesmo nos diferentes estados (PORCIÚNCULA, 2009).

Nesta atividade também são definidos os modelos lógicos das agências controláveis. Este modelo define a lógica de fluxo de recursos na agência, para isso é analisada a estrutura funcional do componente de cada agência. Para obter esses modelos são retiradas informações dos modelos estrutural/funcional, comportamental e das especificações dos componentes do sistema. Estes modelos lógicos são utilizados para determinar os canais sem recursos em cada estado operacional do sistema (PORCIÚNCULA *et al.*, 2015).

Como saída são determinados os canais fornecedores de recursos, canais consumidores de recursos e os conjuntos de canais sem recursos (PORCIÚNCULA, 2009).

### 3.1.2 **Atividade 2 - Definir configurações Operacionais**

Nesta atividade são definidas as diferentes configurações operacionais do sistema, isto é feito manipulando os modelos estrutural/funcional e de informações que caracterizam os estados operacionais. Esta atividade é detalhada em quatro etapas que são desdobradas em tarefas (PORCIÚNCULA, 2009).

Os sistemas automáticos geralmente operam sua função global em diferentes estados operacionais conforme os estados discretos programados. Para cada estado operacional, um conjunto específico de componentes estarão operantes e outros permanecerão desativados. A identificação dos estados operacionais de um sistema automático é baseada nas informações fornecidas pelos modelos funcional/estrutural e comportamental. Esses modelos são necessários para a concepção técnica do sistema (PORCIÚNCULA *et al.*, 2015).

Inicialmente são eliminados os canais e agências que não pertencem a estrutura funcional que atende cada estado operacional, assim como as colunas simultaneamente nulas nas matrizes de precedência e posteridade. Esse procedimento resulta nas matrizes reduzidas. A análise de coerência de fluxo é aplicada a cada estado operacional, a partir de qualquer canal fornecedor na matriz de precedência reduzida do estado operacional analisado (PORCIÚNCULA, 2009).

A análise de coerência de fluxo de recursos tem como objetivo verificar se os recursos que percorrem a rede C/A apresentam um fluxo coerente. Isto significa que

um determinado recurso que entra no sistema precisa ter condições de ser retirado ou, um canal de consumo exige a existência de algum canal de suprimento de onde este recurso tenha se originado. Como ponto de partida para a análise da coerência de fluxo de recursos, considera-se que o mapeamento dos elementos limites já tenha sido realizado e que estes correspondam aos que realmente se deseja projetar (BELAN, 2007). Essa análise é realizada até que um canal consumidor ou um canal bloqueado por um agência controlável seja encontrado. Quando essa análise for concluída para todos os canais fornecedores, as agências e canais não encontrados são excluídos (PORCIÚNCULA *et al.*, 2015).

Como saída dessa atividade são gerados os vetores agências, onde são representadas as agências que restaram na estrutura do sistema para um estado operacional determinado (PORCIÚNCULA, 2009).

### 3.1.3 **Atividade 3 - Definir modelos de confiabilidade**

Na atividade 3 são determinados os modelos de confiabilidade para cada configuração operacional do sistema. São utilizados os vetores agências para definição das matrizes de incidência das árvores de sucesso que representam os modelos de confiabilidade dos estados operacionais do sistema. Esta atividade é detalhada em três etapas que são desdobradas em tarefas (PORCIÚNCULA, 2009).

Utilizando as matrizes de precedência e posteridade, a identificação dos macro estados do sistema e os modelos lógicos dos componentes controláveis, é possível obter, de forma sistemática, as configurações operacionais para cada macro estado do sistema (PORCIÚNCULA *et al.*, 2015).

A representação dos modelos de confiabilidade em árvore de sucesso permite a aplicação de análises qualitativas ao sistema. A representação das árvores de sucesso na forma gráfica facilita a interpretação qualitativa sobre as relações de dependência entre os componentes e subsistemas para atender determinadas funções do sistema. Para representar as árvores de sucesso de forma gráfica são utilizadas as matrizes de incidência. Estas definem os lugares e transições das árvores de sucesso do sistema. Como saída são definidos os modelos de confiabilidade representados pelas árvores de sucesso na forma gráfica, construídas em Grafcet (PORCIÚNCULA, 2009).

### 3.1.4 **Atividade 4 - Estimar a confiabilidade**

Na atividade 4 é realizada a análise da confiabilidade do sistema conforme as configurações dos diferentes estados operacionais do sistema. Para esta análise são utilizadas as matrizes de incidência das árvores de sucesso, a probabilidade de tempo de atuação para cada estado operacional durante a vida útil do sistema e as taxas de falha dos componentes. Esta atividade é composta por três etapas que são desdobradas em tarefas (PORCIÚNCULA, 2009).

A confiabilidade do sistema é dependente do uso efetivo de cada componente, o que varia de acordo com a probabilidade de ocorrência de cada estado operacional durante a vida útil do sistema. Então considerando um sistema com componentes descritos por uma função de densidade de falha exponencial e baseada na equação geral da confiabilidade, deriva-se a equação (12). Esta pode ser usada para estimar a confiabilidade do sistema automático (PORCIÚNCULA *et al.*, 2015).

$$R(t) = \exp \left\{ - \sum_i \lambda_i \left( \sum_j P(M_j)_i \right) t \right\} \quad (12)$$

Onde  $t$  é o tempo de uso do sistema automático (unidade de tempo),  $P(M_j)_i$  é a probabilidade de ocorrência do  $j$ -ésimo estado ( $M$ ) relacionado ao  $i$ -ésimo componente,  $\lambda_i$  é a taxa de falha do  $i$ -ésimo componente (número de falhas/unidade de tempo). O termo  $P(M_j)_i$  leva em conta se o componente (agência  $a_i$ ) pertence ou não ao conjunto de agências com fluxo coerente ( $A_{F,M_i}$ ) relacionado ao estado  $M_j$ , que é:

$$P(M_j)_i = \begin{cases} P(M_j)_i & \text{se } a_i \in A_{F,M_i} \\ 0 & \text{se } a_i \notin A_{F,M_i} \end{cases}$$

Essa atividade tem como saída a estimativa matemática da confiabilidade do sistema automático (PORCIÚNCULA, 2009).

### 3.1.5 **Atividade 5 - Avaliar o parâmetro confiabilidade**

Na atividade 5 é avaliado o resultado da estimativa da confiabilidade obtida na atividade anterior. Esta avaliação será dependente dos requisitos e restrições de projeto que foram definidos na fase de projeto informacional. Devem ser considerados os custos agregados para atender a confiabilidade estimada, os custos

de uma falha no sistema e os custos de manutenção dos componentes durante a vida útil do sistema. Esta atividade tem como saída a decisão sobre o atributo confiabilidade e é possível optar em aprovar ou não a estimativa obtida. Em caso de não aprovação, providências na concepção do sistema devem ser tomadas e a confiabilidade avaliada novamente pela MACSA a partir da atividade 1. No caso da aprovação da confiabilidade, a concepção do sistema deve seguir para próxima atividade (PORCIÚNCULA, 2009).

### 3.1.6 **Atividade 6 - Documentar a análise da confiabilidade**

Nesta atividade são registrados os resultados obtidos na estimativa da confiabilidade do sistema. São documentados os resultados obtidos na análise da confiabilidade, como o gráfico da função confiabilidade do sistema e os gráficos que representam o uso efetivo dos componentes do sistema. A saída desta atividade é um relatório que deve ser anexado aos documentos do projeto do sistema e encaminhado junto a concepção com a aprovação do atributo confiabilidade, para que se dê seguimento nas atividades finais da fase de projeto conceitual (PORCIÚNCULA, 2009).

## 3.2 **2º Etapa - Análise Sistêmica**

Nesta etapa todos os dados e processos verificados na 1º etapa serão estudados. Em seguida serão modelados os Diagramas que servirão para identificar as classes, seus atributos e métodos. Importante ressaltar que aqui só serão modeladas classes que pertençam ao domínio principal do problema do *software*.

Esta etapa irá gerar uma série de documentos como: diagrama de classes, diagrama de sequência, diagrama de colaboração e diagrama de estados. Todos esses diagramas terão como base os diagramas de casos de uso estabelecidos na primeira etapa.

Como resultado, será possível visualizar a estrutura e a forma de interação entre as classes.

## 3.3 **3º Etapa - Projeto**

Essa etapa consiste em extrapolar o domínio central do problema do *software*. Para isso, os diagramas gerados na etapa anterior podem ser acrescentados



de novas classes, de forma a proporcionar uma infra-estrutura tecnológica. Aqui será determinado como o funcionamento do sistema vai atender aos requisitos determinados previamente.

Nesta etapa além da expansão dos diagramas da etapa anterior, também serão gerados dois novos documentos: o diagrama de componentes e diagrama de implantação. Estes tem o objetivo de refletir o mundo físico e irão considerar a existência de componentes e especificar uma visão que reúne todos os elementos de processamento, incluindo *hardware* e *software*.

Uma descrição computacional do que o *software* deverá fazer de forma coerente com a descrição realizada na análise além da documentação da organização física dos componentes, será o resultado dessa etapa.

#### 3.4 **4° Etapa - Implementação**

Na implementação será realizada a tradução da descrição computacional, obtida nas etapas anteriores, em código executável por intermédio do uso de uma ou mais linguagens de programação. O primeiro passo será a escolha da linguagem de programação para criação do código-fonte. Será utilizado um *software* do tipo CASE para executar a conversão automática do modelo de classe para o código da linguagem de programação escolhida. Em seguida será realizada a intervenção manual para a criação do *software*.

Nesta etapa os modelos gerados nas etapas anteriores, que traduzem tecnicamente o entendimento da estrutura do sistema, ganham vida.

#### 3.5 **5° Etapa - Testes e Implantação**

Essa etapa é constituída de testes exaustivos e rigorosos, buscando por erros e sua eliminação. Quatro aspectos devem ser considerados: testes individuais, testes de conjunto, testes de integração e testes de aceitação. Os três primeiros são executados pelo desenvolvedor do software. Os testes de aceitação considerados nesse trabalho, irão considerar como usuário o professor orientador do trabalho. Portanto, a aprovação será deliberada pelo mesmo.

Após a aprovação do *software*, este estará apto para a implantação. No caso deste trabalho, a implantação será considerada na forma de um estudo de caso.

O estudo de caso vai abranger todas as etapas da metodologia. Iniciando pela caracterização do sistema que fornecerá os dados necessários para alimentar a

ferramenta e gerar os modelos matemáticos. Inseridos estes modelos na ferramenta, ela executará o cálculo da confiabilidade de forma automática seguindo as atividades descritas pela MACSA. Os gráficos e relatórios gerados servirão de suporte para análise do parâmetro confiabilidade. O sistema automático considerado consiste num sistema eletropneumático que possui três configurações diferentes para atendimento da função global.

Por fim, os resultados obtidos serão analisados e deliberadas as conclusões a cerca do trabalho realizado.

## 4 DESENVOLVIMENTO DO APLICATIVO MACSA

Este capítulo aborda as cinco etapas da construção do aplicativo MACSA Web 1.0. Para isto seguiu as etapas da metodologia UML, de forma a conceber uma aplicação estruturada. Este processo contou com a consultoria de um cientista da computação, atuando na revisão da linguagem de programação.

### 4.1 1º Etapa - Análise de Requisitos

A primeira etapa da modelagem de software consiste no levantamento de requisitos. Aqui procura-se compreender as necessidades do usuário e o que ele deseja que o sistema a ser desenvolvido realize. Esse levantamento envolveu um estudo exploratório da metodologia MACSA. O estudo consistiu em analisar cada atividade, etapa e tarefa da metodologia de forma minuciosa, para compreender as funcionalidades que deveriam ser implementadas no *software* e os de dados de entrada necessários. Neste estudo foram identificadas as necessidades dos usuários, de forma a obter os requisitos do sistema como um todo, estabelecendo um conjunto de objetivos gerais que o sistema deve cumprir. Os requisitos identificados foram:

- (1) O sistema deve permitir a inclusão e alteração dos dados necessários para determinar a confiabilidade de acordo com a MACSA;
- (2) O usuário deve ser capaz de salvar esses dados para avaliação posterior;
- (3) O usuário deve ser capaz de importar os dados salvos previamente para executar nova avaliação;
- (4) O sistema deve fornecer telas apropriadas para o usuário incluir os dados com a menor chance de erro possível;
- (5) O usuário deve ser capaz de submeter os dados para cálculo da confiabilidade pela MACSA e pelo método de estado único;
- (6) O usuário deve ser capaz de salvar o relatório fornecido pelo sistema após a avaliação da confiabilidade pela metodologia MACSA;
- (7) Cada avaliação deve gerar um novo relatório, de forma que seja possível comparar os resultados com pequenas alterações nos dados;

Após o levantamento dos requisitos do sistema, as necessidades identificadas foram analisadas para fazer a modelagem das atividades encontradas. Para esse propósito foi empregado o diagrama caso de uso, como mostra a figura 11.

Para modelar os diagramas foi utilizada a ferramenta on line: <https://www.draw.io/>. Essa ferramenta é gratuita e permite a construção de fluxogramas, diagramas de processo, UML, ER, mockups entre outros.

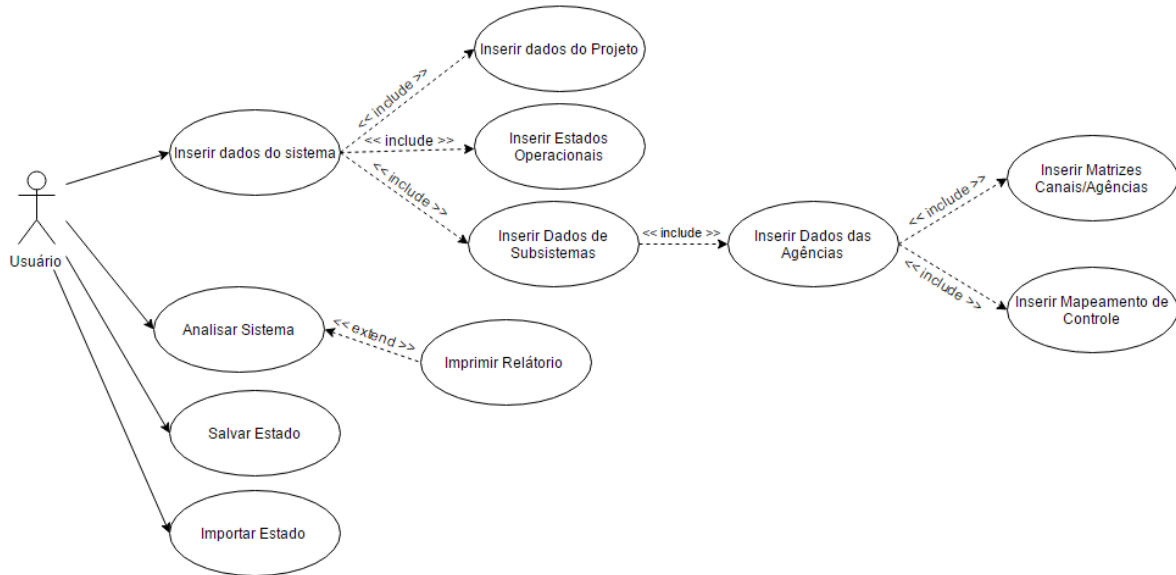


Figura 11 Diagrama caso de uso do aplicativo MACSA  
Fonte: Autor

Um relacionamento *include* entre um caso de uso A e B indica que B é essencial para o comportamento de A. Um relacionamento *extend* de um caso de uso B para A é uma indicação de que o caso de uso B pode ser adicionado ao A.

Esse diagrama descreve um cenário que mostra as funcionalidades que deverão existir no sistema do ponto de vista do usuário. Cada caso de uso atende a um ou mais dos requisitos levantados. O caso Inserir dados do sistema atende ao requisito (1), onde o usuário pode inserir e alterar os dados fornecidos para calcular a confiabilidade do sistema. Ele faz uso do *include* para atender o requisito (4). Então, cada vez que o usuário inserir dados no sistema ele obrigatoriamente deve passar pelos demais casos, que serão programados para reduzir a possibilidade de erro por parte do usuário. O caso analisar sistema deverá satisfazer o requisito (5) e pode adicionar o caso Imprimir Relatório, que deve cumprir as necessidades expressas no (6) e (7). Salvar estado e Importar Estado foram modelados para satisfazer os requisitos (2) e (3) respectivamente.

De acordo com Tonsig (2008), no diagrama caso de uso a relação entre ator e o sistema tem vínculo com uma funcionalidade do *software* que será desenvolvido. Assim é possível conhecer antecipadamente o que deverá existir no software.

Esta etapa pode ser comparada a fase de projeto informacional da metodologia PRODIP. Segundo Back *et al.* (2007) a fase de projeto informacional está destinada a definição das especificações do produto. Para isso são identificadas as necessidades dos clientes e usuários e estas são desdobradas em requisitos. A partir dos requisitos do projeto são definidas as especificações do projeto, ou seja, os objetivos que o produto a ser projetado deve atender.

Fazendo um paralelo a fase de projeto informacional, as necessidades dos usuários foram identificadas a partir do estudo exploratório da metodologia MACSA. Com base nesses requisitos, os objetivos que o *software* deve atender foram definidos e modelados por meio do diagrama de caso de uso.

Uma das principais necessidades dos usuários de aplicativos de engenharia atualmente é a facilidade de acesso e a possibilidade de utilizá-lo em dispositivos móveis. Além disso, melhorar os atributos de manutenibilidade, de atualização e distribuição do aplicativo é uma meta a ser atendida neste trabalho.

Desta forma, optou-se pela utilização da plataforma web no desenvolvimento de aplicação, o que possibilita o usuário avaliar a confiabilidade de seu sistema automático usando apenas o navegador (*browser*) do seu dispositivo.

#### 4.2 2º Etapa - Análise Sistêmica

Nesta etapa todos os dados e processos verificados foram estudados. Com base neste estudo foi gerado o documento diagrama de classes, conforme mostra a Figura 12. Este diagrama tomou como base o diagrama de caso de uso modelado na primeira etapa. Importante resaltar que nesta etapa foi modelado apenas o domínio principal do *software*.

No processo de desenvolvimento de produtos, a fase de projeto conceitual se destina ao desenvolvimento da concepção do produto. Para tanto se busca estabelecer a estrutura funcional do produto. Essa atividade engloba o processo de definição da função global a ser executada e suas subfunções (Back *et al.*, 2007).

Fazendo um paralelo com a fase de projeto conceitual, o diagrama de classe define a estrutura estática do sistema e a possibilidade de interação entre as classes. Dessa forma é possível visualizar as classes que irão compor o sistema e como elas se complementam e transmitem informações entre si.

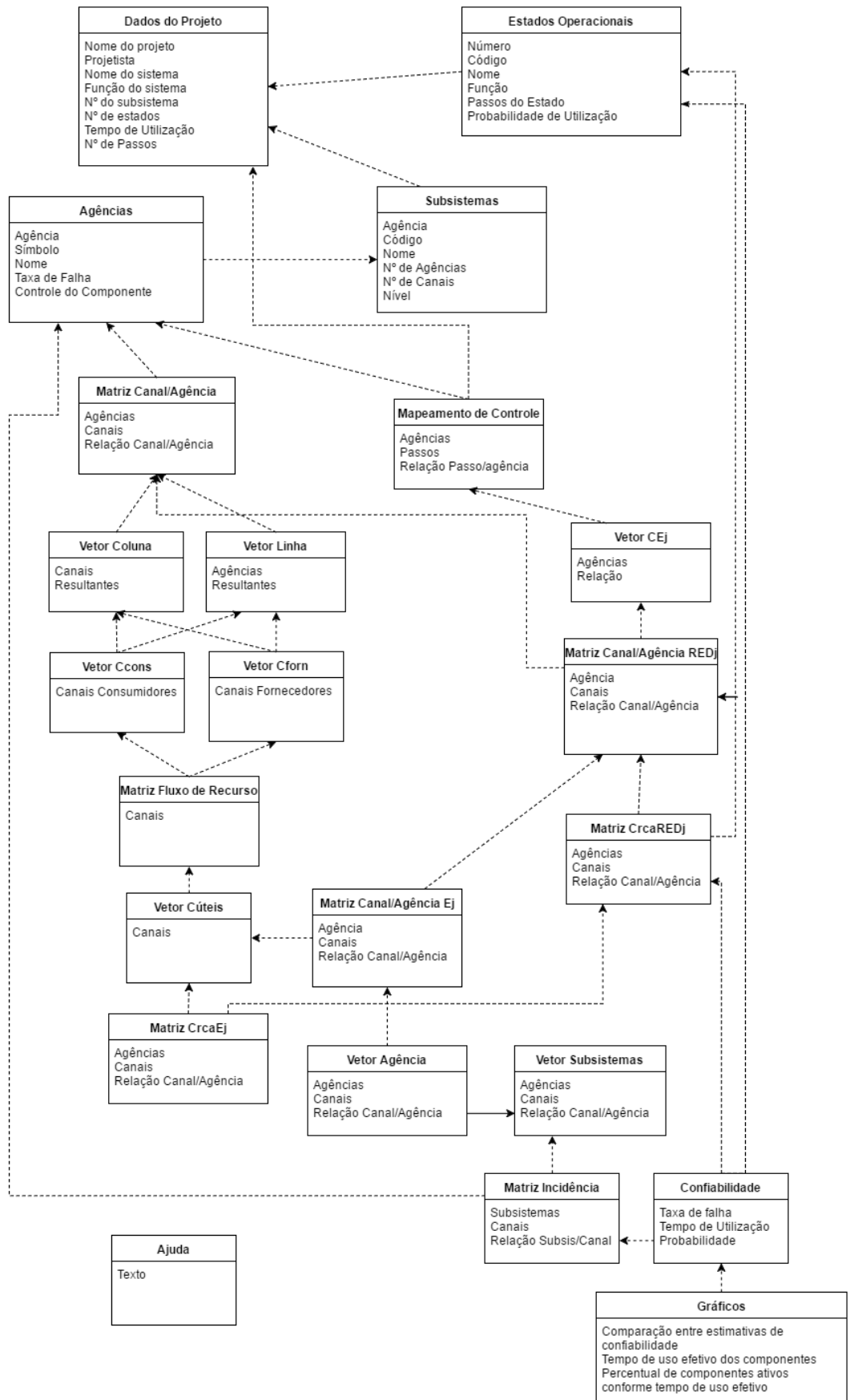


Figura 12 Diagrama de classes do domínio central  
Fonte: Autor

Um diagrama de classes pode oferecer três perspectivas: conceitual, especificação e implementação. Para este projeto foi escolhida uma perspectiva conceitual, pois ela possui menor complexidade e pode ser considerada independente da linguagem implementada. Nesta perspectiva o diagrama representa os conceitos do domínio que está sendo estudado e estes conceitos são naturalmente relacionados às classes que irão executá-los.

As classes não são elementos isolados com absoluta autonomia, muitas tarefas só são possíveis pela colaboração existente entre as classes (TONSIG, 2008). O objetivo principal do diagrama é permitir a visualização das classes que irão compor o sistema e como elas se complementam e transmitem informações entre si.

As classes do diagrama estão relacionadas por dependência. De acordo com Tonsig (2008) o relacionamento de dependência entre duas classes mostra que uma instância de uma classe depende da instância de outra classe. Então, uma mudança na especificação de um elemento pode alterar a especificação do elemento dependente. Esse relacionamento foi implementado, pois a alteração nos dados de uma classe alterará as instâncias das demais classes relacionadas. Conseqüentemente, o resultado da confiabilidade do sistema será alterado.

Os diagramas de colaboração e de estados não foram modelados. Estes diagramas são formas alternativas de representar a troca de mensagens entre objetos. Entretanto, esta visão do sistema é modelada pelos diagramas de sequência, de forma que não são essenciais para o andamento do projeto. Ademais, segundo Tonsig (2008) os diagramas de caso de uso, classes e sequência formam um tripé mínimo necessário para a documentação da análise orientada a objetos utilizando a UML, desde que tais diagramas estejam consistentes entre si.

Os diagramas de sequência serão apresentadas na 3ª etapa, após extrapolar o domínio central do problema do *software* e acrescentar novas classes ao diagrama de classes, de forma a proporcionar uma infra-estrutura tecnológica.

#### 4.3 3º Etapa - Projeto

Esta etapa consiste em extrapolar o domínio central do problema do *software*. Para isso, o diagrama de classe gerado na etapa anterior foi acrescido de novas classes, de forma a proporcionar uma infra-estrutura tecnológica. O novo diagrama de classes está representado na Figura 13.

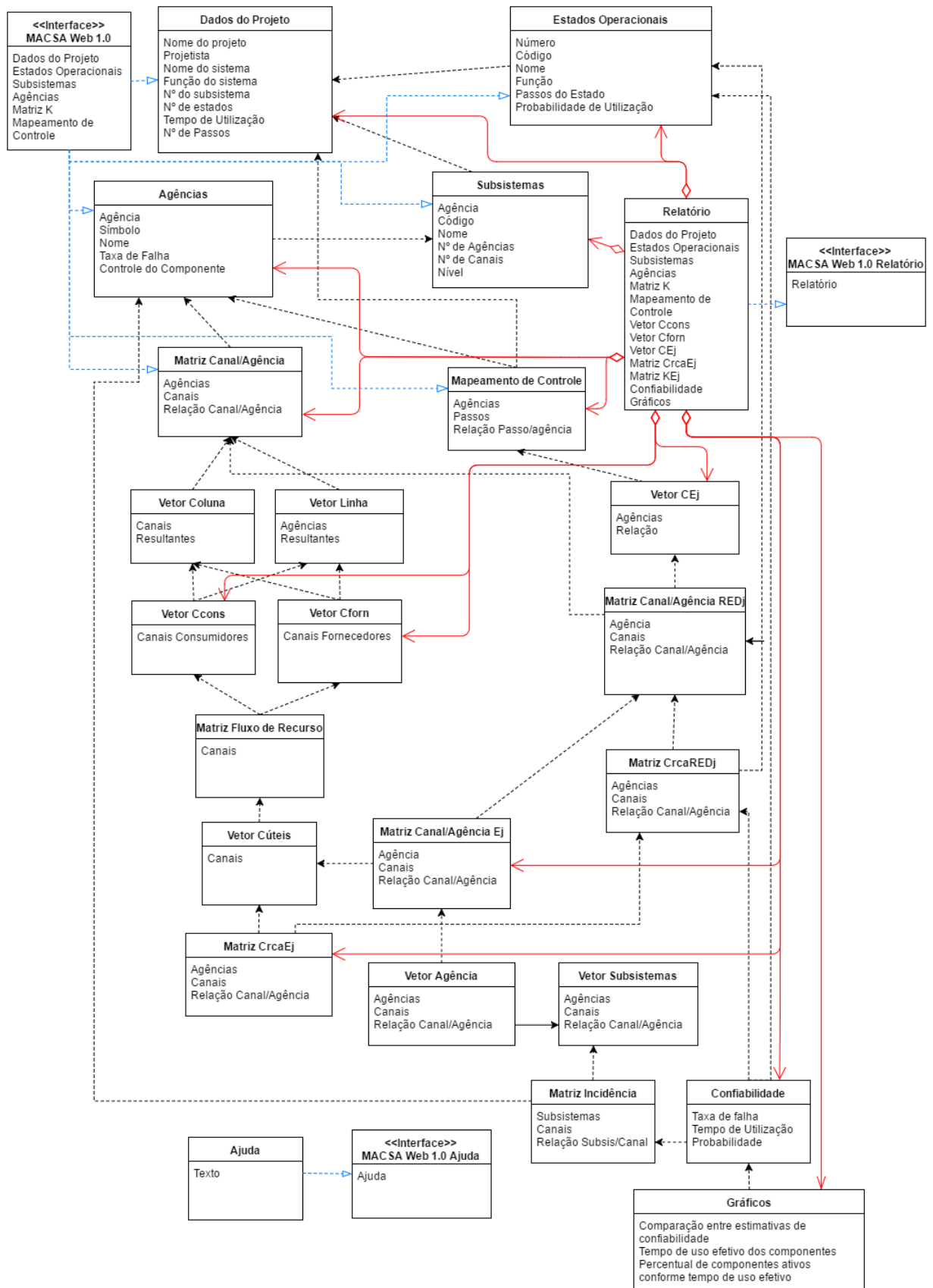


Figura 13 Diagrama de classes extrapolado  
Fonte: Autor



O diagrama de classes extrapolado foi acrescido das interfaces. As classes se relacionam com as interfaces pelo relacionamento de realização, representado por uma linha pontilhada azul claro com seta vazada.

A interface é um tipo especial de classe a qual não pode ser instanciada, o que a torna uma classe virtual. Ela serve para especificar operações externamente visíveis para uma outra classe implementar e descreve os padrões legais de interação entre dois objetos (TONSIG, 2008).

Uma realização é um relacionamento em que uma das classes especifica um contrato cujo cumprimento é assegurado por outra classe. Na maioria dos casos, ela será empregada para especificar a relação entre uma interface e a classe ou componente que proporciona uma operação ou serviço para a interface (BOOCH *et al.*, 2005).

A classe relatório também foi acrescida ao diagrama. Ela se relaciona com algumas classes por agregação, representada por uma linha vermelha com losango sem preenchimento do lado da classe dona do relacionamento. Segundo Tonsig (2008) um relacionamento de agregação, também chamado todo-parte, é uma forma especial de associação que é usada para mostrar que um tipo de objeto é formado pela agregação de outros. O objeto parte é um atributo do todo e sua existência faz sentido, mesmo não existindo o objeto-todo.

Após finalizar o diagrama de classe, foram modelados os diagramas de sequências. Estes diagramas mostram a interação entre objetos com a preocupação de documentar os métodos executados ao longo do tempo. Portanto, tem o objetivo de mostrar como as mensagens são trocadas entre os objetos no decorrer do tempo para a realização de uma operação (TONSIG, 2008). O diagrama de sequência é utilizado para especificar o fluxo principal de um caso de uso (BOOCH *et al.*, 2005).

Na figura 14 está representado o diagrama de sequência modelado a partir do caso de uso Inserir Dados.

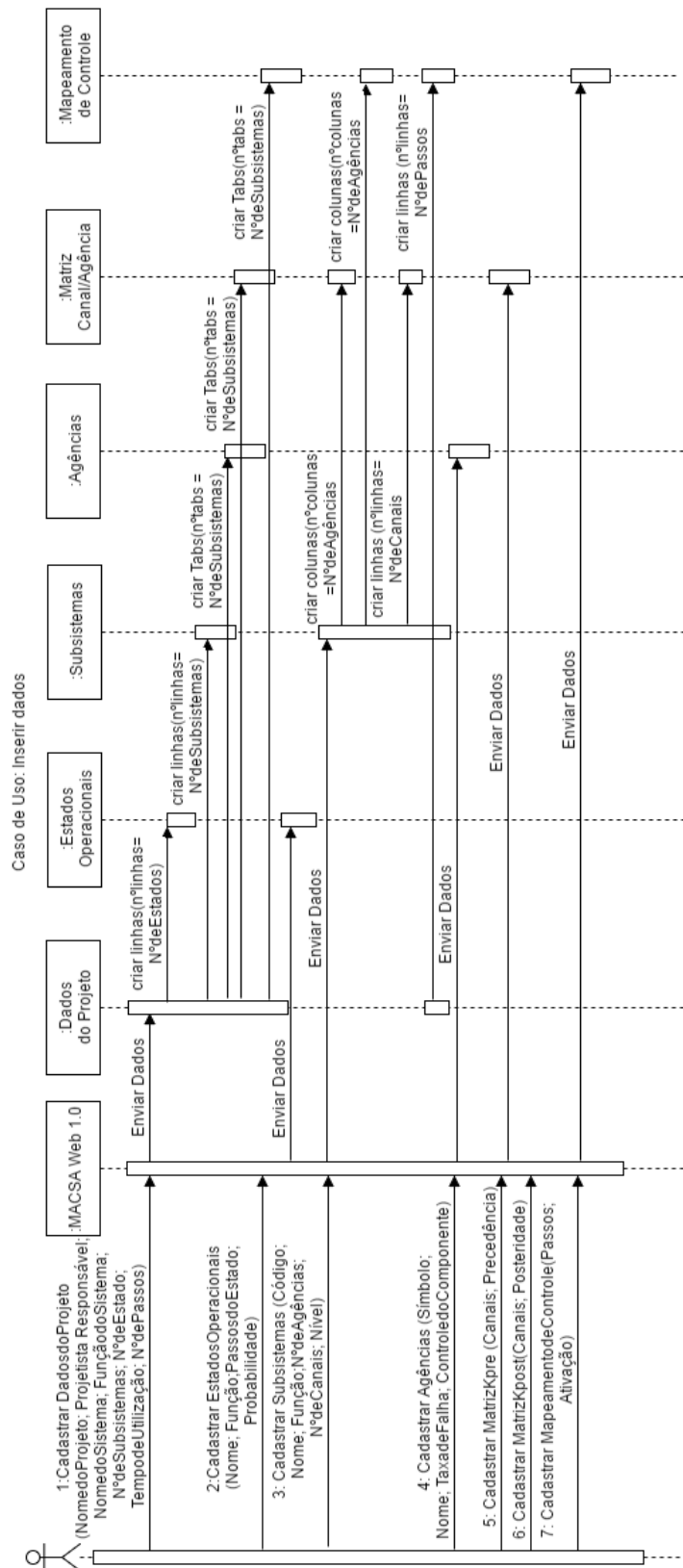


Figura 14 Diagrama de sequência do caso de uso 'Inserir Dados'  
Fonte: Autor

O diagrama de sequência do caso de uso Analisar Sistema ficou muito extenso. Então a imagem foi dividida e inserida no Apêndice A deste trabalho. A figura 15 mostra o diagrama de sequência modelado para o caso de uso Salvar Estado.

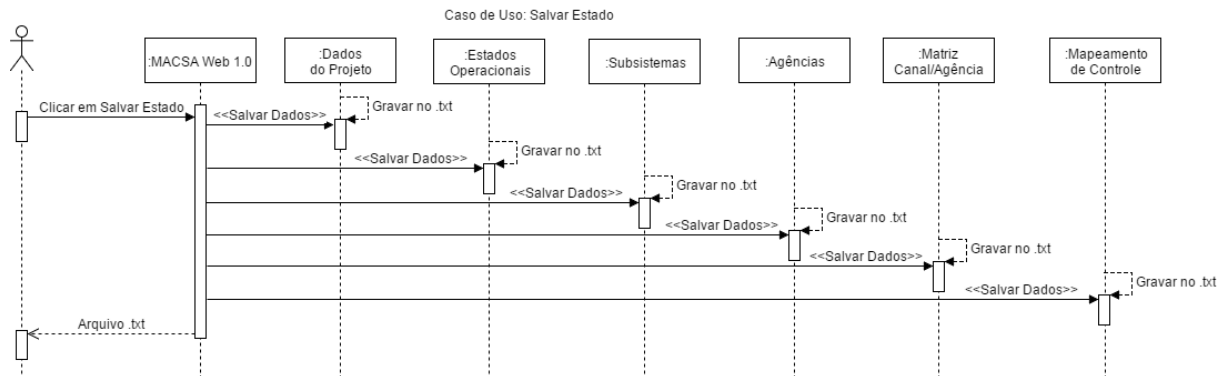


Figura 15 Diagrama de sequência do caso de uso 'Salvar Estado'  
Fonte: Autor

A linha pontilhada vertical representa o tempo de vida de um objeto e é preenchida por barras verticais que representam o tempo em que o mesmo fica ativo. As linhas horizontais representam as mensagens trocadas entre os objetos e são acompanhadas por um rótulo com o nome da mensagem. A figura 16 mostra o diagrama de sequência modelado para o caso de uso Importar Estado.

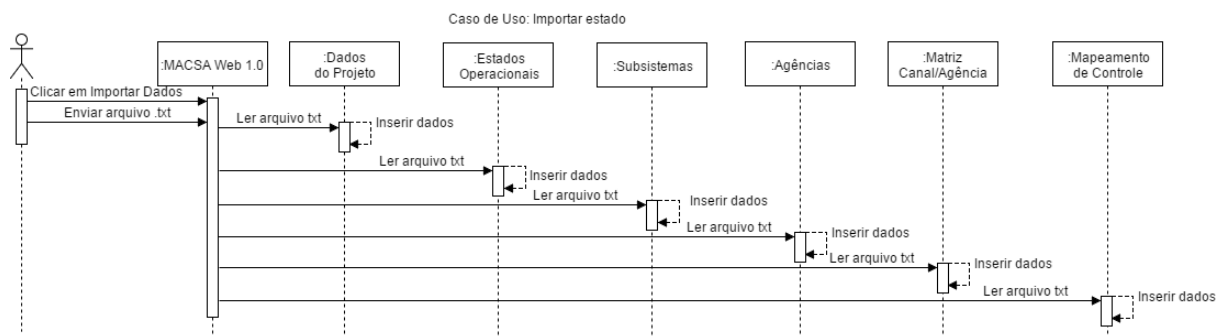


Figura 16 Diagrama de sequência do caso de uso 'Importar Dados'  
Fonte: Autor

Esses diagramas representam a sequência de mensagens passadas entre objetos, no programa de computador. Entende-se por mensagens os serviços solicitados de um objeto a outro, e as respostas desenvolvidas para as solicitações. O diagrama representa essa informação de uma forma simples e lógica e descreve a maneira como os grupos de objetos colaboram em algum comportamento ao longo do tempo. Cada diagrama registra o comportamento de um único caso de uso e exibe os objetos e as mensagens passadas entre esses objetos.

Os últimos diagramas a serem modelados foram os diagramas de componentes e de implantação. Um diagrama de componentes captura a estrutura física da implementação. Ele apresenta uma visão estática de como o sistema está implementado e quais seus módulos de *software*, ou seja, seus componentes. Na UML os componentes podem ser criados separadamente ou combinados com o diagrama de implantação. Um diagrama de implantação mostra o *layout* físico de um sistema, revelando quais partes do *software* são executadas em quais partes do *hardware* (BOOCH *et al.*, 2005).

A figura 17 mostra o diagrama de componentes combinado ao diagrama de implantação modelado para o aplicativo MACSA.

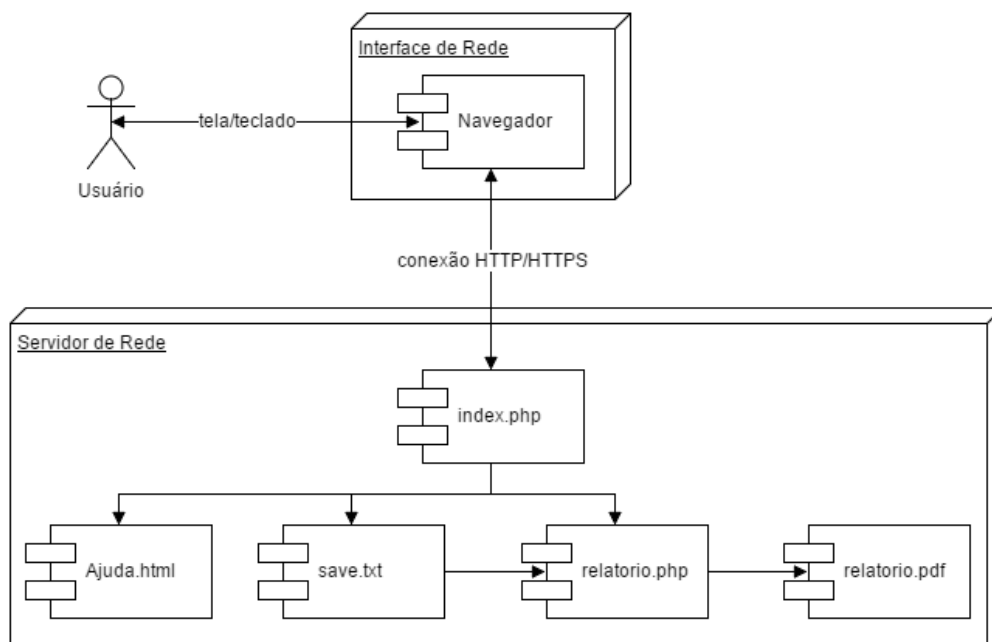


Figura 17 Diagrama de Implantação combinado com diagrama de componentes  
Fonte: Autor

Esses diagramas mostram quais componentes do *software* funcionam em quais nós de *hardware*. O servidor de rede representa o domínio onde está hospedada a aplicação MACSA Web 1.0. A interface de rede pode ser um dispositivo móvel (celular/ tablet), notebook ou um desktop. Essa interface precisa apenas de um navegador e uma conexão com a internet para rodar a aplicação.

Na metodologia PRODIP, a fase de projeto preliminar se destina ao estabelecimento do leiaute final do produto e à determinação da viabilidade técnica e econômica. O projeto preliminar faz uso de diferentes tipos de modelos: icônicos, analógicos, numéricos e computacionais, também conhecidos como protótipos

virtuais. Estabelecido o leiaute final, é iniciado o desenvolvimento do plano de fabricação e de teste do protótipo e a elaboração da estrutura preliminar do protótipo, que serve de parâmetro para o cálculo inicial de custo (BACK *et al.*, 2007).

Fazendo um paralelo com a fase de projeto preliminar, nesta etapa foi determinado como o funcionamento do sistema vai atender aos requisitos determinados previamente. Foi contruída uma descrição computacional do que o software deverá fazer e uma documentação da organização física dos componentes, ou seja, foi determinado o plano de “fabricação” do aplicativo em desenvolvimento.

#### 4.4 4º Etapa - Implementação

Foi determinado na 1ª Etapa que seria desenvolvida uma aplicação web. Para construção dessa aplicação foi utilizado o *framework* Bootstrap como uma base de estilo para a infraestrutura. Como o diagrama de classe foi modelado sob uma perspectiva conceitual, não foi utilizado um software do tipo CASE para executar a conversão automática do modelo de classe para o código da linguagem de programação escolhida.

O Bootstrap é uma coleção de vários elementos e funções personalizáveis para projetos da web, empacotados previamente em uma única ferramenta. O código fonte é aberto e está disponível no GitHub. Ao projetar um site com o Bootstrap, os desenvolvedores podem escolher quais elementos querem usar. Esses elementos personalizáveis contidos no Bootstrap são uma combinação de HTML, CSS e JavaScript. Ele tem diversas funções, como capacidade de resposta móvel e várias opções de *plug-in* com muitos recursos (UTTERBACK, 2014).

Entre as diversas características do Bootstrap, o projeto MACSA se aproveitou principalmente dos modelos pré-definidos de interface CSS e Javascript. Dentre os quais foram utilizados layouts de tabela, espaçamento do corpo da página, menu lateral e menu superior. A principal vantagem desta ferramenta é a escalabilidade, tornando as páginas Web robustas e responsivas ao crescimento de dados inseridos, e a dinamicidade, tornando-as adaptáveis aos diversos dispositivos em que pode ser acessada. Como ilustrado pela figuras 18 e 19.

A linguagem de programação PHP (*Hiper Text Preprocessor*), mesclada ao código HTML, foi utilizada na manipulação dos modelos matemáticos na etapa de tradução da metodologia MACSA para a interface digital.

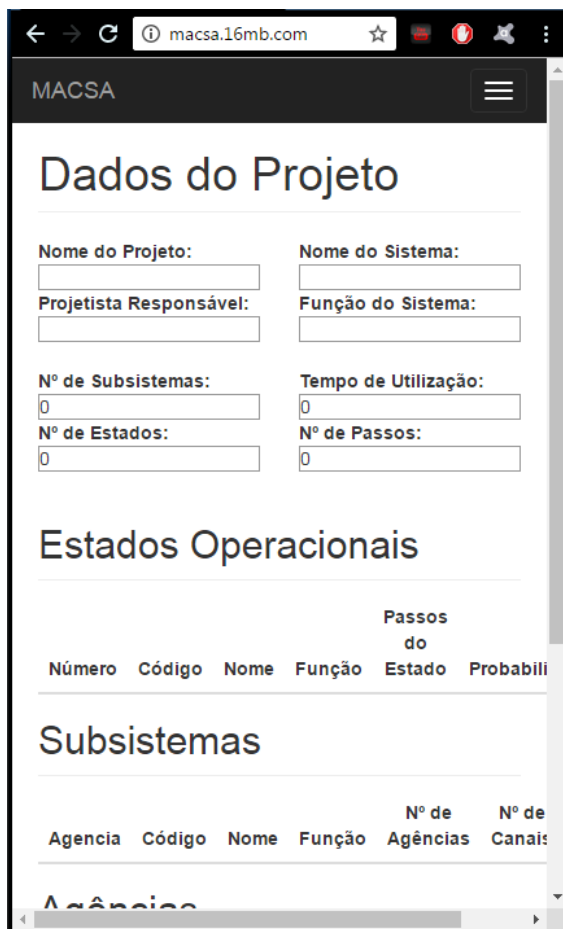


Figura 18 Interface MACSA para dispositivos móveis com Bootstrap  
Fonte: Autor

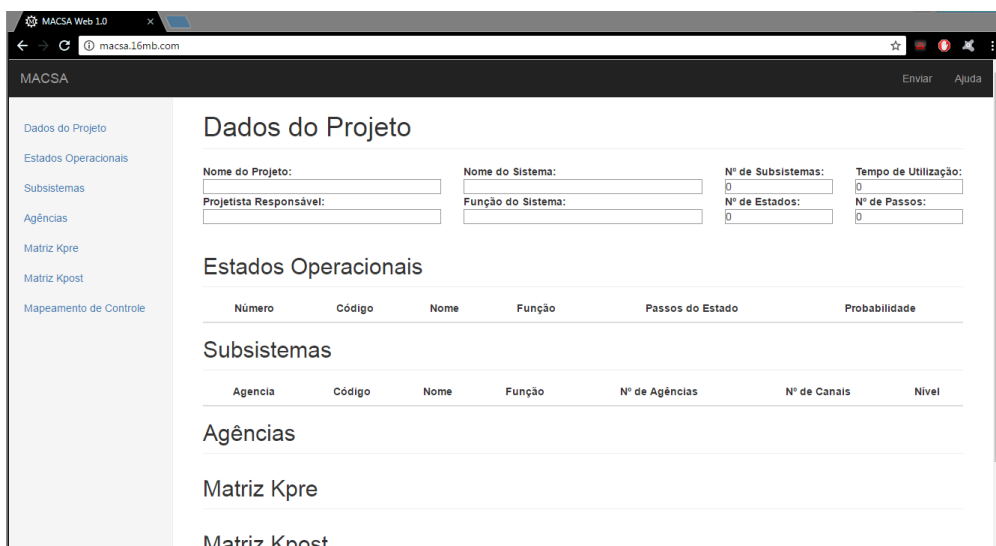


Figura 19 Interface MACSA para alta resolução com Bootstrap  
Fonte: Autor

As vantagens na utilização da linguagem PHP são: ampla utilização para construção de aplicações Web complexas, adaptação à qualquer plataforma onde

for possível instalar um servidor Web, a curva de aprendizagem inicial é baixa e a linguagem possui código-fonte aberto.

Nesta etapa foi aplicada a prática de programação pareada (*Pair Programming*) da metodologia de desenvolvimento de *software Extreme Programming* (Programação Extrema). Essa prática consiste na programação em par/dupla num único computador. Em geral a dupla é formada por um iniciante na linguagem e um programador experiente funcionando como um instrutor. O iniciante fica à frente fazendo a codificação, e o instrutor acompanha ajudando a desenvolver, sugerindo melhorias e alternativas. Desta forma o programa sempre é revisto por duas pessoas, evitando e diminuindo assim a possibilidade de erros. Com essa prática é possível melhorar a qualidade do código fonte gerado (TELES, 2006). O cientista da computação que participou do desenvolvimento do aplicativo, exerceu o papel de instrutor na programação pareada.

A funcionalidade Importar Estado, prevista no diagrama Caso de Uso, não foi implementada devido a dinamicidade executada para criação das tabelas de preenchimento dos dados. Esse fator elevou a complexidade da implementação desta funcionalidade e não foi possível adequar um sistema para leitura de dados que atendesse a escalabilidade dos formulários dinâmicos.

Para desenvolver os gráficos presentes no relatório, utilizou-se a interface de programação de aplicativos *Google Chart Tool*. Essa API é gratuita e possui uma grande variedade de gráficos para uso. Esses gráficos e suas informações podem ser customizados e adequados a qualquer tipo de dados estatísticos. No projeto MACSA foram utilizados gráficos de barras, pizza e linhas. Apenas as suas informações de entrada foram alteradas, preservando o *layout* fornecido pela API.

#### 4.5 5° Etapa - Testes e Implantação

A última etapa foi composta de testes que buscaram por erros e sua eliminação. Os testes utilizaram os dados provenientes de estudos de caso da tese de doutorado de Porciúncula (2009) para testar sua lógica de funcionamento.

A medida que a etapa de Implementação avançou, cada tarefa da metodologia MACSA programada foi testada individualmente até obter o resultado correto. Após os testes individuais estes módulos foram unidos aos demais e testados para verificar o comportamento executado pelo conjunto. Aqui entende-se por módulo uma sequência de linhas de código unidas para executar uma tarefa da

metodologia MACSA. Caso o resultado do teste estivesse correto e os módulos trocando dados entre si de forma eficiente, seguiam-se para a programação do próximo módulo necessário.

Para o aplicativo MACSA Web 1.0 não foram necessários os testes de integração, pois o mesmo não possui nenhum mecanismo de interface com outros sistemas como, por exemplo, um banco de dados.

O teste de aceitação neste trabalho, considerou como usuário o professor orientador. Portanto, a aprovação foi deliberada pelo mesmo. Importante ressaltar que o usuário acompanhou todo o processo de desenvolvimento do produto.

Após a aprovação do software, seguiu-se para a implantação. No caso deste trabalho, a implantação foi considerada na forma de um estudo de caso que será apresentado no próximo capítulo.

Conforme Back *et al.* (2007) a fase da elaboração do projeto detalhado do produto é destinada à várias finalidades: aprovação do protótipo, finalização das especificações dos componentes, detalhamento do plano de manufatura, e preparação da solicitação de investimento. Simultaneamente com a construção, o teste e à aprovação do protótipo, é concluída a otimização das especificações dos componentes.

As etapas de Implementação e Testes e Implantação possuem similaridades com a etapa de projeto detalhado. Nestas etapas ocorreram a construção do protótipo do aplicativo, ele foi submetido a testes e aprovação. Cabe ressaltar que a aplicação MACSA Web 1.0 não está finalizada e ainda possui aspectos a serem melhorados e funcionalidades a serem implementadas.

O aplicativo MACSA Web 1.0 está disponível no endereço: <http://macsa.16mb.com/>. Ele possui uma opção de ajuda no canto superior direito da tela com orientações sobre os dados necessários para a avaliação da confiabilidade pela metodologia MACSA. Os formulários para inserção de dados são gerados de forma dinâmica a partir dos dados do projeto informados. Esta foi a solução encontrada para reduzir a chance de erro do usuário, pois o próprio aplicativo cria os formulários do tamanho necessário conforme o usuário insere os dados do seu projeto. Além disso os campos para dados numéricos não permitem a inserção de texto nos mesmos. Desta forma foi reduzida a chance de erro por excesso ou ausência de dados e preenchimento incorreto.



O botão Salvar no menu superior da tela redireciona o usuário para uma nova aba, onde é gerado um *link* que possibilita o *download* dos dados preenchidos em formato txt. O botão Enviar no menu superior da tela redireciona o usuário para uma nova aba, onde é gerado o relatório da análise da confiabilidade. Cada vez que o usuário clicar nesse botão um novo relatório é gerado. Deste modo é possível fazer pequenas alterações nos dados fornecidos, na aba de inserção de dados, e comparar os diferentes relatórios e resultados obtidos. Na aba do relatório existe o botão imprimir no menu superior. Este recurso pode ser utilizado para gerar um arquivo pdf do relatório, que pode ser salvo no computador ou dispositivo móvel.

Por fim, o menu lateral do aplicativo possui âncoras de navegação. As âncoras permitem que o usuário encontre informações úteis de forma rápida, sem ter que se deslocar para cima ou para baixo. Um clique em uma âncora move a página até o ponto solicitado pelo usuário.

## 5 DESCRIÇÃO DO ESTUDO DE CASO

Para avaliar a aplicativo MACSA Web 1.0 foi aplicado um estudo de caso sobre um sistema eletropneumático. Este sistema foi desenvolvido na tese de doutorado de Massoco (2016) e está representado na figura 20, montado na bancada eletropneumática.

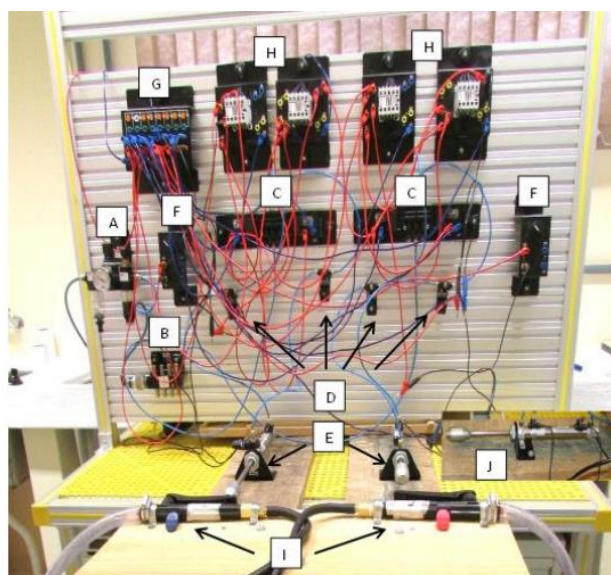


Figura 20 Bancada eletropneumática para ensaio de gatilhos  
Fonte: MASSOCO (2016).

Este sistema eletropneumático foi implementado para ensaiar a confiabilidade das válvulas de acionamento de pulverizadores costais. A função do sistema

consistia em acionar o gatilho automaticamente em tempos e movimentos constantes e reproduzíveis. Este sistema, conforme a Figura 20, possui:

- (A) Filtro regulador com manômetro e lubrificador;
- (B) Bloco distribuidor com válvula para abertura e fechamento;
- (C) Duas válvulas direcionais com acionamento por solenoide;
- (D) Quatro válvulas reguladoras de fluxo;
- (E) Dois cilindros de ação dupla com avanço e retorno pneumático, com dois sensores eletromagnéticos acoplados a cada um;
- (F) Dois botões de acionamento;
- (G) Quadro distribuidor;
- (H) Quatro relés.

Os componentes elétricos (F), (G) e (H) estavam ligados à uma fonte estabilizadora de 24 volts localizada atrás do painel. A demanda de ar comprimido era suprida por um compressor ligado ao componente (A), que fornecia aos demais componentes pneumáticos. A Figura 21 apresenta o diagrama do circuito eletropneumático desenvolvido para atender as distintas funções do sistema.

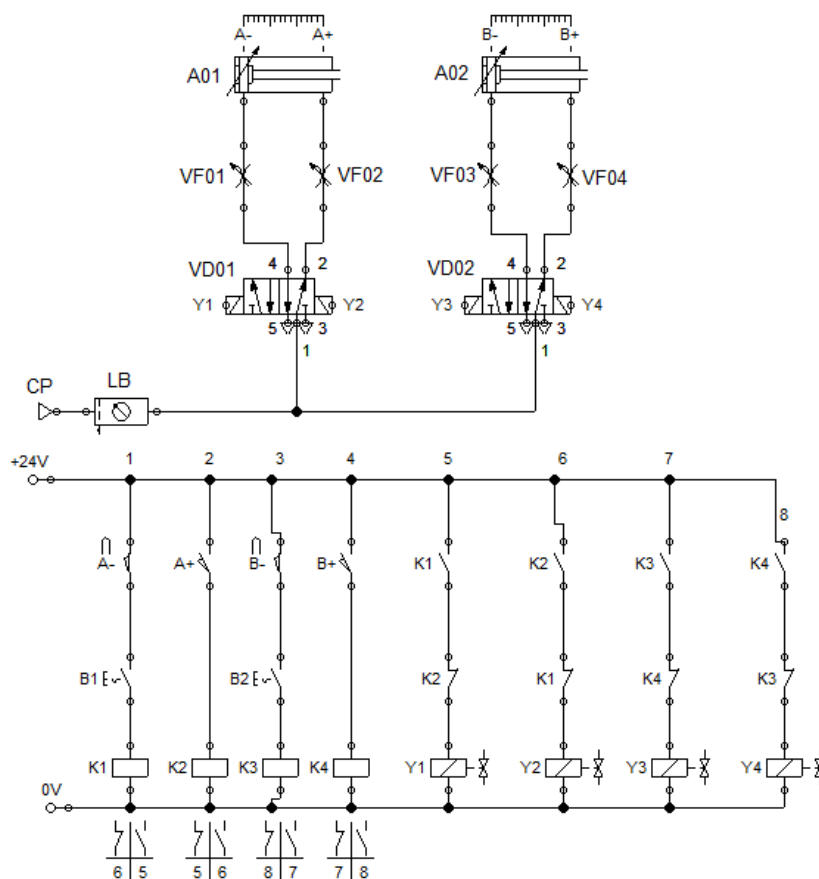


Figura 21 Diagrama de circuito do sistema eletropneumático  
Fonte: Autor

Os componentes A01 e A02 representam (E). VF01, VF02, VF03 e VF04 representam (D). VD01 e VD02 representam (C). LB corresponde a (A) e CP representa o compressor. Na parte elétrica, B1 e B2 representam (F). K1, K2, K3 e K4 representam (H). A+, A-, B+ e B- são os sensores eletromagnéticos acoplados aos cilindros de dupla ação e Y1, Y2, Y3 e Y4 são os solenóides que comandam as válvulas direcionais. Os pontos +24V e 0V representam a fonte e o quadro distribuidor (G).

Assim, o circuito assume diferentes configurações operacionais conforme a função solicitada pelo usuário. Em cada uma é possível que alguns componentes sejam solicitados com frequência, enquanto outros permanecem desativados.

Acionando os botões B1 e B2 simultaneamente, o usuário pode usufruir de uma configuração utilizando os dois atuadores de dupla ação, A01 e A02. Acionando apenas B1, o sistema irá assumir uma configuração onde apenas o atuador A01 é colocado em funcionamento. Acionando B2 o sistema irá acionar apenas A02.

### 5.1 Modelagem funcional e estrutural

A modelagem por meio de Rede Canal/Agência torna possível a identificação dos subsistemas que compõem o sistema, facilitando assim a definição das funções parciais do mesmo. Esta modelagem define o tipo de relacionamento e a troca de informações, energia e matéria entre os diferentes módulos (PORCIÚNCULA, 2009). A modelagem em rede C/A está representada na Figura 22.

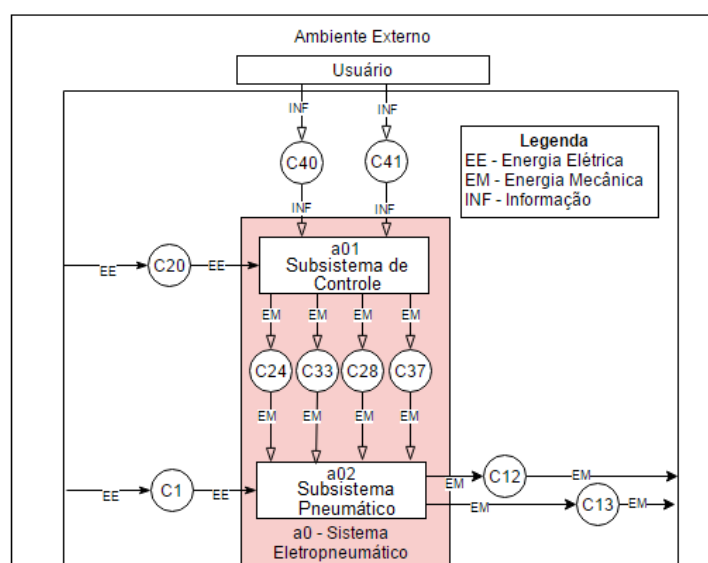


Figura 22 Modelo funcional/estrutural do sistema eletropneumático  
Fonte: Autor

Existe uma correspondência entre os diagramas de circuito e o modelo rede Canal/Agência. Essa equivalência existe onde os componentes elétricos, eletrônicos, pneumáticos e hidráulicos são considerados agências. Os fios, tubulações, cabos e hastes são considerados canais (PORCIÚNCULA *et al.*, 2015).

O detalhamento dos subsistemas permite a identificação dos componentes e suas relações para atender a função do sistema. Esta caracterização facilita a análise funcional, estrutural e comportamental.

## 5.2 Modelagem comportamental

A Figura 23 apresenta a modelagem comportamental do sistema eletropneumático. A ferramenta utilizada para essa modelagem é o Grafcet.

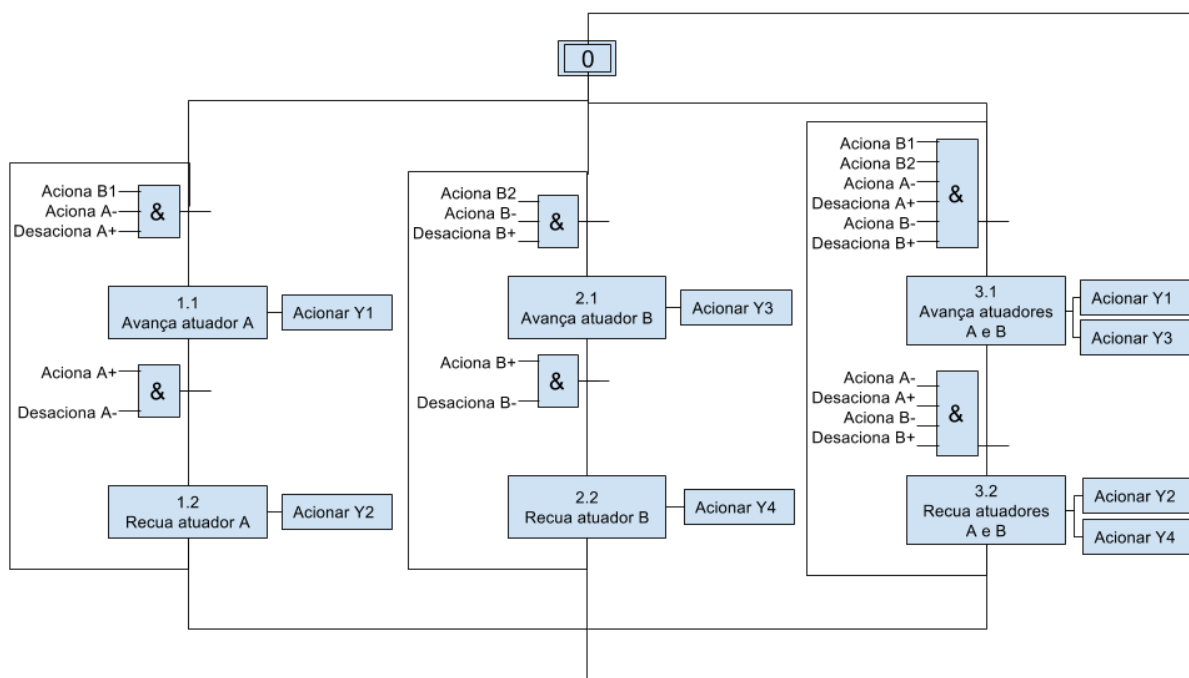


Figura 23 Modelo comportamental do sistema eletropneumático  
Fonte: Autor

O modelo comportamental está representado por passos e transições interligados por arcos. A transição representa uma decisão para mudança de estado do sistema. Para que ela ocorra, são necessários que todos os passos cuja saída está ligada à entrada da transição estejam ativados e a condição associada à transição esteja satisfeita (LOPES *et al.*, 2008). Várias sequências são possíveis conforme o estado que o sistema assumir durante sua vida útil.

A partir da modelagem comportamental é possível fazer a caracterização dos estados operacionais do sistema, representado pela Tabela 1.

Tabela 1 Estados Operacionais do sistema Eletropneumático

Número	Código	Nome	Função	Passos do Estado	Probabilidade
1	E1	Estado 1	Ensaiai gatilho 1	1.1;1.2	0,5
2	E2	Estado 2	Ensaiai gatilho 2	2.1;2.2	0,1
3	E3	Estado 3	Ensaiai gatilho 1 e 2	3.1;3.2	0,4

Fonte: Autor

De acordo com o modelo comportamental, o sistema eletropneumático pode assumir três estados:

E1 - representado no Grafcet pelos passos 1.1 e 1.2;

E2 - representado no Grafcet pelos passos 2.1 e 2.2;

E3 - representado no Grafcet pelos passos 3.1 e 3.2.

Para cada estado é atribuída uma probabilidade da utilização do mesmo ao longo do ciclo de vida do sistema. Esta informação é decorrente da expectativa do projetista quanto à intensidade da utilização do mesmo nos diferentes estados (PORCIÚNCULA, 2009).

### 5.3 Caracterizar o sistema Eletropneumático

A atividade Caracterizar Sistema parte do modelo funcional/estrutural em Rede C/A, do modelo comportamental e da especificação dos componentes do sistema (PORCIÚNCULA, 2009). Para iniciar a análise do sistema eletropneumático, o modelo em Rede C/A apresentado na Figura 22 foi detalhado para primeiro nível, conforme Figura 24.

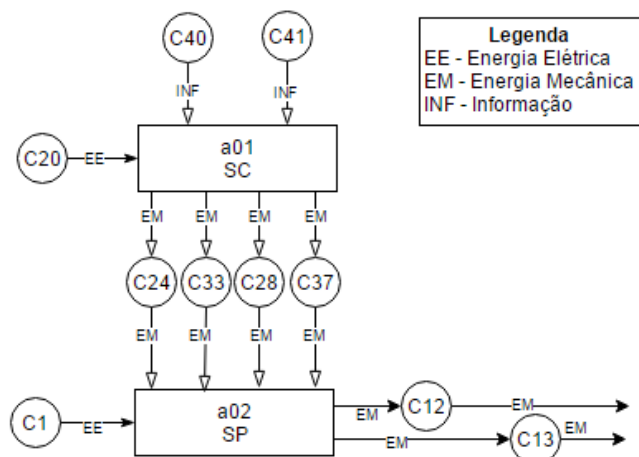


Figura 24 Modelo do desdobramento do SE em rede C/A  
Fonte: Autor

Nele foram identificados dois subsistemas, o Subsistema de Controle (SC) e o Subsistema Pneumático (SP). A partir do modelo rede C/A são identificadas as agências que compõe a concepção do sistema, conforme Tabela 2.

Tabela 2 Caracterização das agências do Sistema Eletropneumático

Agência	Símbolo	Nome	Taxa de Falha	Controle do Componente
a01	SC	Subsistema de Controle	0	C
a02	SP	Subsistema Pneumático	0	C

Fonte: Autor

Como este detalhamento representa um visão do sistema como um todo, cada agência representa um subsistema. Como estes subsistemas serão detalhados nas próximas etapas, não são consideradas taxas de falhas. A coluna controle do componente caracteriza se as agências são controláveis (C) ou não (NC). De acordo com Porciúncula (2009), agências controláveis são definidas como componentes que permitem a alteração do seu estado por meio da intervenção do ambiente externo ou de um sistema de controle. Portanto o usuário pode alterar estado de a01 e a mesma pode alterar o estado de a02, caracterizando as duas agências como controláveis.

Essa configuração do sistema não se altera com os estados assumidos pelo sistema, pois depende do sucesso de todos os componentes para atender a sua função. Por isso é necessário simplesmente identificar as matrizes Kpre e Kpost, representadas na Tabela 3.

Tabela 3 Representação matricial do sistema eletropneumático (a) Matriz Kpre (b) Matriz Kpost

Kpre			Kpost		
	a01	a02		a01	a02
C1	0	1	C1	0	0
C12	0	0	C12	0	1
C13	0	0	C13	0	1
C20	1	0	C20	0	0
C24	0	1	C24	1	0
C28	0	1	C28	1	0
C33	0	1	C33	1	0
C37	0	1	C37	1	0
C40	1	0	C40	0	0
C41	1	0	C41	0	0

Fonte: Autor

A matriz Kpre relaciona a precedência de canais para cada agência e a Kpost relaciona os canais posteriores. Os canais são organizados nas linhas e as agência nas colunas de cada matriz. Os valores de '1' e '0' simbolizam a existência ou não de relação entre cada canal e agência.

#### 5.4 Caracterizar o Subsistema de Controle

O subsistema de controle foi desdobrado de forma que seus componentes foram identificados, bem como a relação entre estes no sistema. A Figura 25 apresenta a modelagem para o SC em Rede C/A.

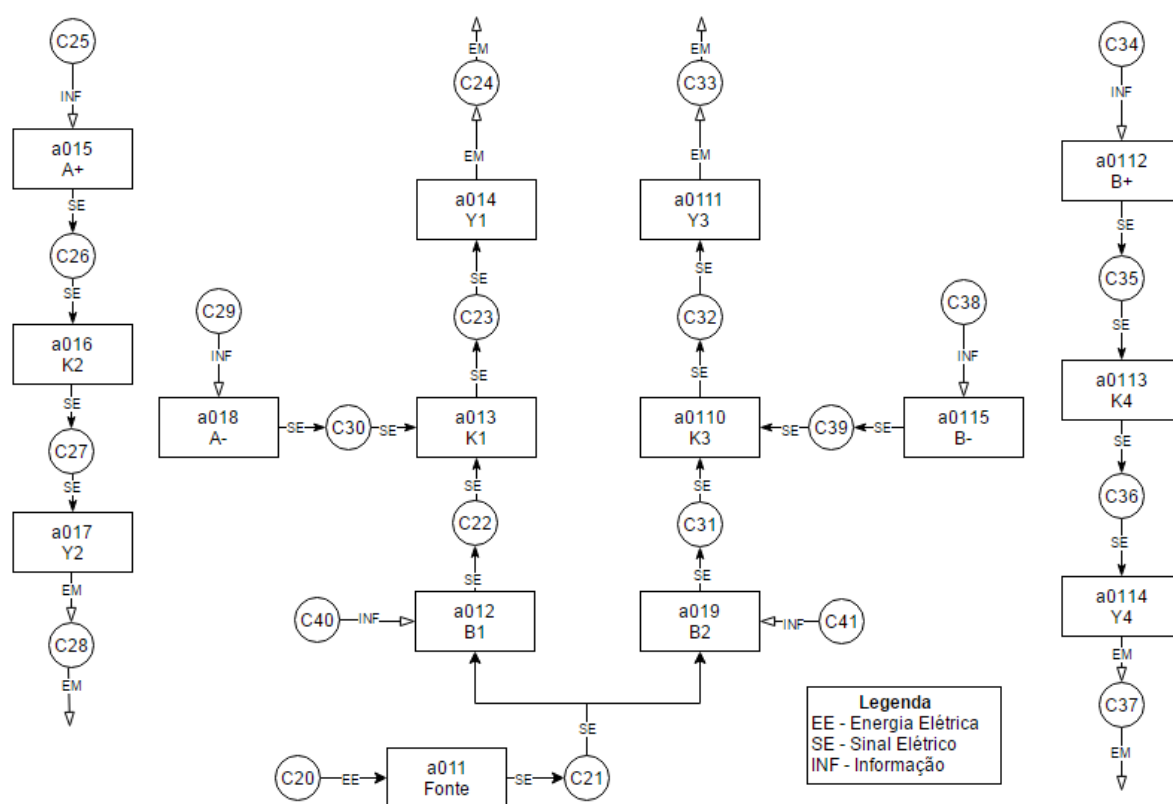


Figura 25 Modelo do desdobramento do SC em rede C/A  
Fonte: Autor

Neste modelo cada componente do subsistema está representado por uma agência. As agências estão ligadas por canais, que podem transportar energia, informação ou sinal elétrico conforme a Legenda.

Alguns canais do subsistema de controle estão presentes também no subsistema pneumático, efetuando a comunicação entre os dois subsistemas.

A Tabela 4 apresenta o modelo Rede C/A representado na forma matricial por meio das matrizes Kpre e Kpost. A matriz Kpre relaciona a precedência de canais para cada agência e a Kpost relaciona os canais posteriores.

Tabela 4 Representação matricial do SC (a) Matriz Kpre (b) Matriz Kpost

Kpre															
	a011	a012	a013	a014	a015	a016	a017	a018	a019	a0110	a0111	a0112	a0113	a0114	a0115
C20	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C21	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0
C22	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
C23	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
C24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C25	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
C26	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
C27	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
C28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C29	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
C30	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
C31	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
C32	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
C33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C34	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
C35	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
C36	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
C37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
C39	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
C40	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
C41	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

(a)

Kpost															
	a011	a012	a013	a014	a015	a016	a017	a018	a019	a0110	a0111	a0112	a0113	a0114	a0115
C20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C21	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C22	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
C23	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
C24	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
C25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C26	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
C27	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
C28	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
C29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C30	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
C31	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
C32	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
C33	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
C34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C35	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
C36	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
C37	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
C38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
C40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(b)

Fonte: Autor

Após desdobrar o subsistema de controle em representações matriciais, são identificadas as agências que compõe a concepção do sistema. Estas informações foram compiladas na Tabela 5.



Tabela 5 Caracterização das agências do SC

Agência	Símbolo	Nome	Taxa de Falha	Controle do Componente
a011	FT	Fonte estabilizadora 24 volts	9E-10	NC
a012	B1	Botão com trava 1	0,00000187	C
a013	K1	Relé	0,0000003	NC
a014	Y1	Solenóide da VD01	0,0000003	C
a015	A+	Sensor eletromagnético do A01	0,0000233	C
a015	K2	Relé	0,0000003	NC
a017	Y2	Solenóide da VD01	0,0000003	C
a018	A-	Sensor eletromagnético do A01	0,0000233	C
a019	B2	Botão 2	0,00000187	C
a0110	K3	Relé	0,0000003	NC
a0111	Y3	Solenóide da VD02	0,0000003	C
a0112	B+	Sensor eletromagnético do A02	0,0000233	C
a0113	K4	Relé	0,0000003	NC
a0114	Y4	Solenóide da VD02	0,0000003	C
a0115	B-	Sensor eletromagnético do A02	0,0000233	C

Fonte: Autor

Esta tabela apresenta a caracterização das agências do SC, incluindo informações relevantes para análise, tais como: código da agência, símbolo do componente, nome, taxa de falha e tipo de controle. O símbolo atribuído para cada agência cria um vínculo entre o modelo rede C/A e o diagrama de circuito (Figura 21). O tipo de controle caracteriza se as agências são controláveis ou não.

A matriz de mapeamento do controle do sistema é uma matriz binária que relaciona os passos do modelo comportamental com as agências do sistema (PORCIÚNCULA, 2009). A Tabela 6 apresenta a matriz de mapeamento do controle do subsistema de controle.

Tabela 6 Matriz de Mapeamento de Controle do SC

	a011	a012	a013	a014	a015	a016	a017	a018	a019	a0110	a0111	a0112	a0113	a0114	a0115
0	1	0	1	0	0	1	0	0	0	1	0	0	1	0	0
1.1	1	1	1	1	0	1	0	1	0	1	0	0	1	0	0
2.1	1	0	1	0	0	1	0	0	1	1	1	0	1	0	1
3.1	1	1	1	1	0	1	0	1	1	1	1	0	1	0	1
1.2	1	1	1	0	1	1	1	0	0	1	0	0	1	0	0
2.2	1	0	1	0	0	1	0	0	1	1	0	1	1	1	0
3.2	1	1	1	0	1	1	1	0	1	1	0	1	1	1	0

Fonte: Autor

Esta matriz define as agências que estarão ativas nos determinados passos. Portanto define as agências que estarão ativas no estado operacional definido por determinado conjunto de passos.

### 5.5 Caracterizar o Subsistema Pneumático

O subsistema Pneumático foi desdobrado de forma que seus componentes e a relação entre eles foram identificadas. A Figura 26 apresenta a modelagem gráfica do SP em Rede C/A.

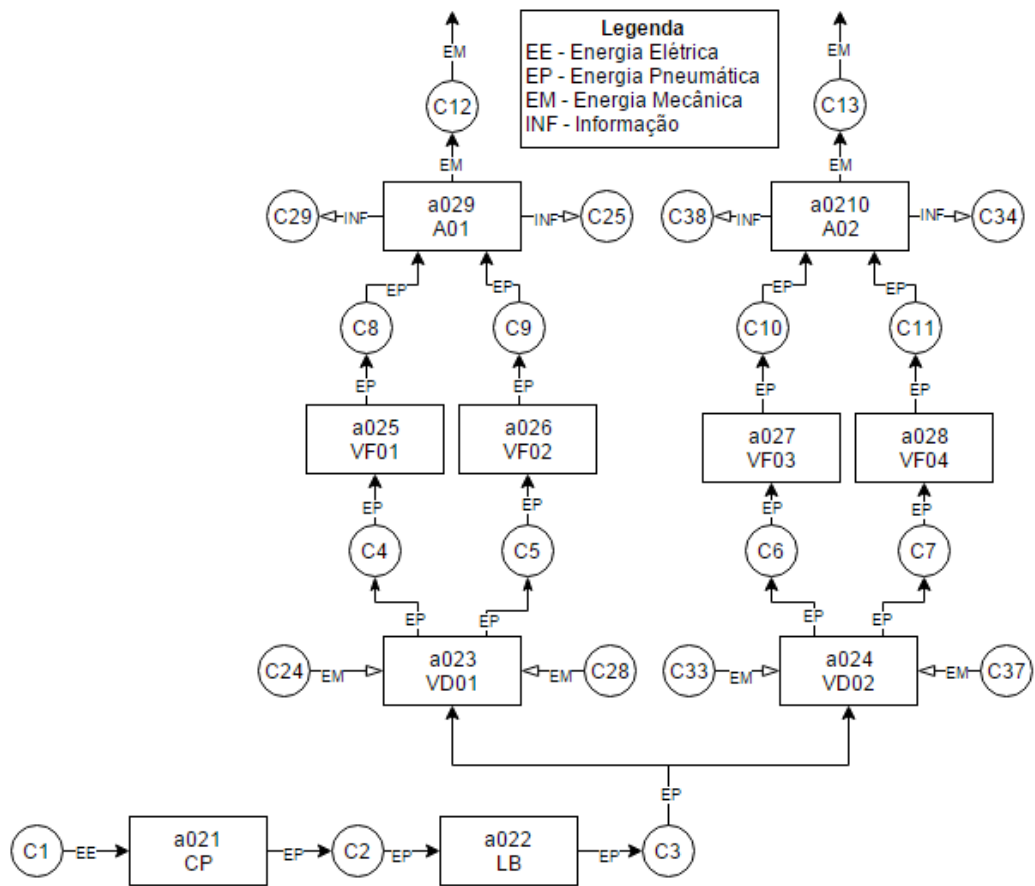


Figura 26 Modelo do desdobramento do SP em rede C/A  
Fonte: Autor

Cada componente do subsistema está representado por uma agência interligada às demais por meio dos canais, que podem transportar energia ou informação. Alguns canais presentes no subsistema pneumático são comuns ao sistema de controle, fazendo a comunicação entre os subsistemas.

O modelo Rede C/A pode ser convertido para a forma matricial, estabelecendo as relações entre os canais e agências na forma binária. Deste processo são extraídas duas matrizes, as matrizes de incidência anterior (Kpre) e incidência posterior (Kpost). Estas matrizes estão representadas na Tabela 7.

Tabela 7 Representação matricial do modelo desdobrado do SP (a) Matriz Kpre (b) Matriz Kpost

Kpre											Kpost										
	a021	a022	a023	a024	a025	a026	a027	a028	a029	a0210		a021	a022	a023	a024	a025	a026	a027	a028	a029	a0210
C1	1	0	0	0	0	0	0	0	0	0	C1	0	0	0	0	0	0	0	0	0	0
C2	0	1	0	0	0	0	0	0	0	0	C2	1	0	0	0	0	0	0	0	0	0
C3	0	0	1	1	0	0	0	0	0	0	C3	0	1	0	0	0	0	0	0	0	0
C4	0	0	0	0	1	0	0	0	0	0	C4	0	0	1	0	0	0	0	0	0	0
C5	0	0	0	0	0	1	0	0	0	0	C5	0	0	1	0	0	0	0	0	0	0
C6	0	0	0	0	0	0	1	0	0	0	C6	0	0	0	1	0	0	0	0	0	0
C7	0	0	0	0	0	0	0	1	0	0	C7	0	0	0	1	0	0	0	0	0	0
C8	0	0	0	0	0	0	0	0	1	0	C8	0	0	0	0	1	0	0	0	0	0
C9	0	0	0	0	0	0	0	0	0	1	C9	0	0	0	0	0	1	0	0	0	0
C10	0	0	0	0	0	0	0	0	0	0	C10	0	0	0	0	0	0	1	0	0	0
C11	0	0	0	0	0	0	0	0	0	1	C11	0	0	0	0	0	0	0	1	0	0
C12	0	0	0	0	0	0	0	0	0	0	C12	0	0	0	0	0	0	0	0	1	0
C13	0	0	0	0	0	0	0	0	0	0	C13	0	0	0	0	0	0	0	0	0	1
C24	0	0	1	0	0	0	0	0	0	0	C24	0	0	0	0	0	0	0	0	0	0
C25	0	0	0	0	0	0	0	0	0	0	C25	0	0	0	0	0	0	0	0	1	0
C28	0	0	1	0	0	0	0	0	0	0	C28	0	0	0	0	0	0	0	0	0	0
C29	0	0	0	0	0	0	0	0	0	0	C29	0	0	0	0	0	0	0	0	1	0
C33	0	0	0	1	0	0	0	0	0	0	C33	0	0	0	0	0	0	0	0	0	0
C34	0	0	0	0	0	0	0	0	0	0	C34	0	0	0	0	0	0	0	0	0	1
C37	0	0	0	1	0	0	0	0	0	0	C37	0	0	0	0	0	0	0	0	0	0
C38	0	0	0	0	0	0	0	0	0	0	C38	0	0	0	0	0	0	0	0	0	1

(a)

(b)

Fonte: Autor

Após desdobrar o subsistema Pneumático, as agências que compõe a concepção do sistema são caracterizadas, conforme Tabela 8.

Tabela 8 Caracterização das agências do SP

Agência	Símbolo	Nome	Taxa de Falha	Controle do Componente
a021	CP	Compressor	0,00000313	NC
a022	LB	Lubrífil	0,0000178	NC
a023	VD01	Válvula Direcional 01	0,00000385	C
a024	VD02	Válvula Direcional 02	0,00000385	C
a025	VF01	Válvula de Fluxo 01	0,00000313	NC
a026	VF02	Válvula de Fluxo 02	0,00000313	NC
a027	VF03	Válvula de Fluxo 03	0,00000313	NC
a028	VF04	Válvula de Fluxo 04	0,00000313	NC
a029	A01	Atuador linear 01	0,00000858	C
a0210	A02	Atuador linear 02	0,00000858	C

Fonte: Autor

Esta tabela apresenta a caracterização das agências do subsistema pneumático e cria um vínculo entre o modelo rede C/A e o diagrama de circuito, representado na Figura 21.

Por fim é necessário definir a matriz de mapeamento de controle do sistema. Desta forma é possível relacionar os passos do modelo comportamental com as agências do sistema que ficam ativadas durante os mesmos. Esta matriz está representada na Tabela 9.

Tabela 9 Matriz de Mapeamento de Controle do SP

	a021	a022	a023	a024	a025	a026	a027	a028	a029	a0210
0	1	1	0	0	1	1	1	1	0	0
1.1	1	1	1	0	1	1	1	1	1	0
2.1	1	1	0	1	1	1	1	1	0	1
3.1	1	1	1	1	1	1	1	1	1	1
1.2	1	1	1	0	1	1	1	1	1	0
2.2	1	1	0	1	1	1	1	1	0	1
3.2	1	1	1	1	1	1	1	1	1	1

Fonte: Autor

## 5.6 Estimativa da Confiabilidade

Para calcular a confiabilidade deste sistema será considerado um tempo de utilização de 3000 horas, dentro de um ambiente adequado para o equipamento desempenhar suas funções. Todos os dados extraídos da caracterização do sistema e seus subsistemas foram inseridos no MACSA Web 1.0. O relatório gerado está inserido no Apêndice B.

Considerando as probabilidades atribuídas para cada estado, o tempo de uso efetivo mostra que 44% dos componentes ficam ativos durante 1500 horas. Então, cerca de metade dos componentes ficará inativo durante metade do tempo de funcionamento do sistema eletropneumático. Outros 44% ficam ativos durante 2700 horas e apenas 12% ficam ativos durante todo o tempo de utilização considerado nesta análise. Essa diferença no tempo de uso efetivo gera um impacto na confiabilidade, pois os componentes que estão ativos durante um intervalo de tempo menor sofrem menos desgaste, portanto são menos propícios a falhas.

A confiabilidade de acordo com a metodologia MACSA considera os diferentes estados que o sistema pode assumir durante sua vida útil. Desta forma, utiliza o tempo de uso efetivo dos componentes para calcular a confiabilidade do sistema. Para este sistema a confiabilidade MACSA é 0,7047, considerada satisfatória. Pelo método de estado único, supondo todos os componentes ativos ao longo do ciclo de vida, esse resultado cai para 0,6231. Deste modo, se o tempo de uso efetivo não for considerado para estimativa da confiabilidade, os resultados obtidos podem conduzir para uma tomada de decisão enganosa com relação a concepção do sistema.

A partir das matrizes de incidência do estado (Apêndice B) é possível identificar os arranjos estruturais e funcionais para cada estado operacional assumido pelo sistema. As Figuras 27 e 28 mostram as reduções do (a) diagrama

pneumático, (b) modelo rede C/A do subsistema de controle e (c) rede C/A do subsistema hidráulico para os Estado 1 e 2.

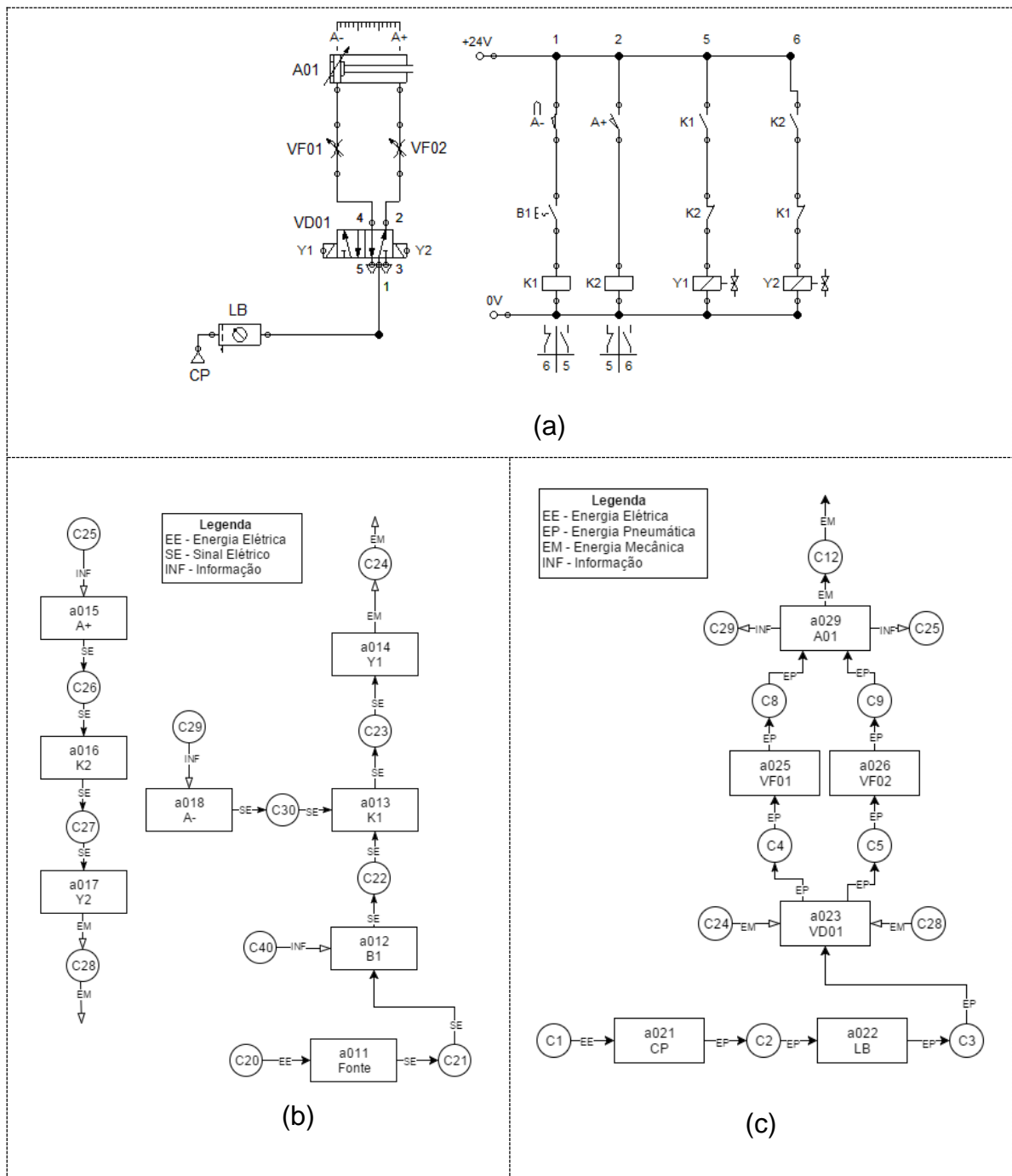


Figura 27 Redução dos modelos funcional/estrutural para Estado 1 (a) diagrama eletropneumático (b) rede C/A do SC e (c) rede C/A do SP  
Fonte: Autor

Percebe-se que para cada estado diferentes componentes foram eliminados dos diagramas e modelos rede C/A, pois estes não são solicitados para atender a função do estado em consideração. Assim, somente os componentes que restaram estão relacionados com o atendimento da função do mesmo.

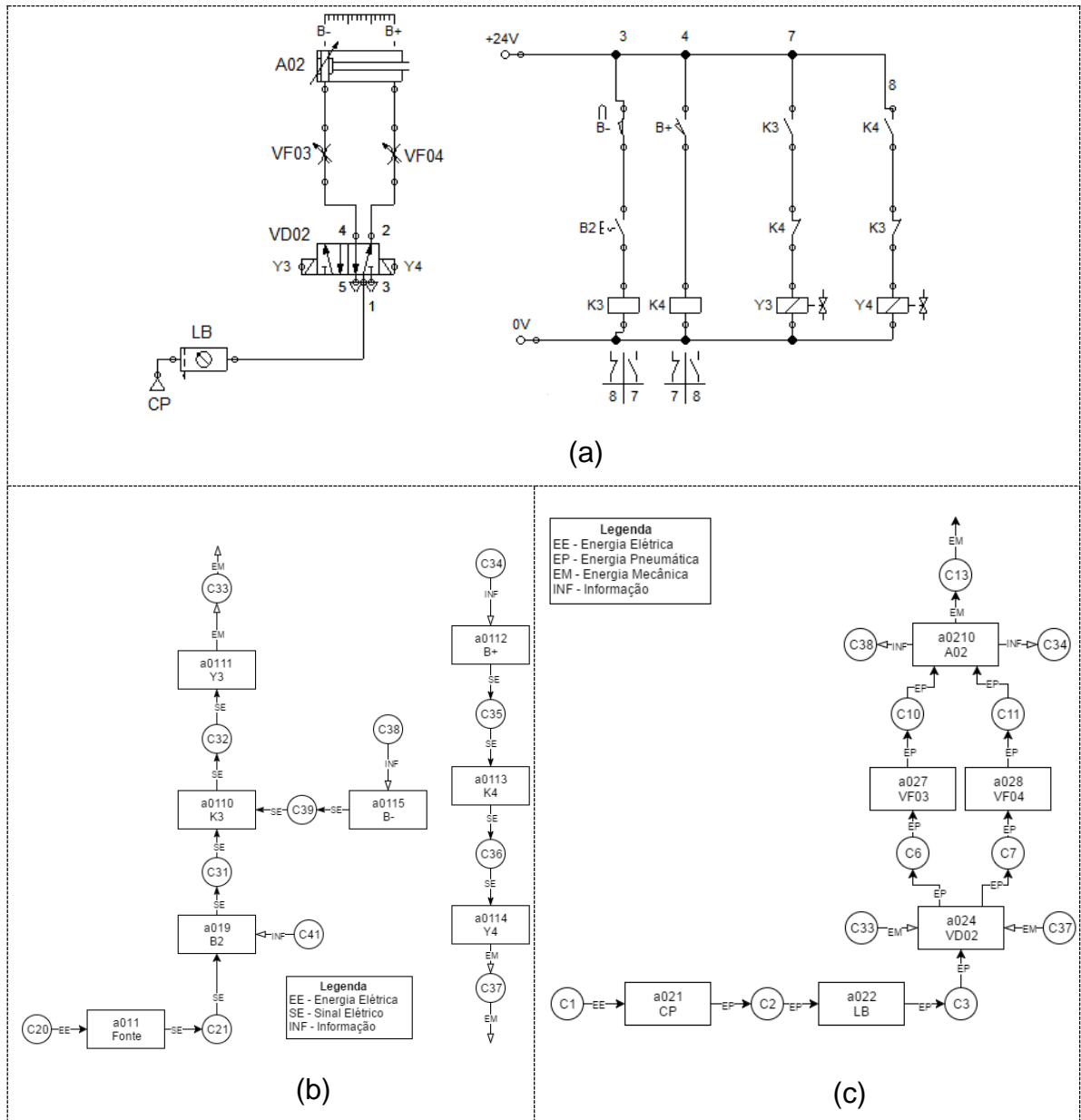


Figura 28 Redução dos modelos funcional/estrutural para Estado 2 (a) diagrama eletropneumático (b) rede C/A do SC e (c) rede C/A do SP  
 Fonte: Autor

O estado 3 não foi representado pois nele todos os componentes estão ativos, ou seja não há reduções. Portanto, para este estado, os diagramas permanecem semelhantes aos apresentados nas Figuras 21, 25 e 26.

## 6 CONCLUSÕES

Este trabalho apresentou o desenvolvimento de um aplicativo para cálculo da confiabilidade aplicando a metodologia MACSA comparada a confiabilidade de estado único. Para fins de experimentar o desempenho do aplicativo foi desenvolvido um estudo de caso sobre um sistema eletropneumático.

Os esforços de modelagem e programação resultaram num aplicativo que permite o cálculo da confiabilidade do sistema considerando as configurações do sistema nos seus potenciais estados operacionais. Este foi desenvolvido com sucesso para acesso via navegador Web.

O relatório gerado pelo aplicativo possui tabelas e gráficos que dão subsídio para tomadas de decisão a cerca da concepção do sistema. Assim, é possível fazer alterações no projeto nas fases iniciais onde o custo de mudanças é menor.

Esta ferramenta pode ser utilizada por projetistas na fase de projeto conceitual para dar suporte à análise da confiabilidade de sistemas automáticos. Ela é uma ferramenta portátil, pois pode ser acessada por qualquer pessoa em *desktops* ou *notebooks* e também dispositivos móveis, como celulares e tablets.

Algumas melhorias seriam interessantes de serem implementadas em versões posteriores do aplicativo. A geração das árvores de sucesso na forma gráfica, simplificação da inserção de dados para análise, conversão automática do modelo rede C/A para forma matricial e implementação da funcionalidade importar estados são algumas dessas melhorias.

O estudo de caso de um sistema eletropneumático mostrou que o aplicativo foi eficiente em estimar a confiabilidade por ambas metodologias, MACSA e estado único. O relatório gerado na análise evidenciou as diferentes configurações assumidas pelo sistema em cada estado operacional. Além de estimar o percentual de componentes ativos conforme o tempo de uso efetivo dos componentes. A confiabilidade MACSA foi estimada em 0,7047 e considerada satisfatória. Entretanto, se necessário as informações fornecidas pelo aplicativo poderiam ser utilizadas para a tomada de decisão acerca de mudanças na concepção do sistema.

As diferenças entre a estimativa da confiabilidade de sistemas automáticos pelas metodologias MACSA e estado único podem ser significativas para tomadas de decisão durante o processo de projeto. Tendo em vista a longa vida útil destes sistemas e sua composição por vários componentes.

## 7 REFERÊNCIAS

- BACK, Nelson; DIAS, Acires; OGLIARE, Andre; SILVA, Jonny Carlos da. Projeto Integrado de Produtos: planejamento, concepção e modelagem. 1 ed. Barueri: Manole, 2008. cap 1, p.3-28.
- BELAN, Henri Carlo; SZPAK, Rodrigo; CURY, José Eduardo Ribeiro; DE NEGRI, Victor Juliano. Channel/instance petri nets for structural and functional modeling of industrial equipment. In: 20th Internacional Congress of Mechanical Engineering, 2009, Gramado. Anais...Gramado: ABCM, 2009.
- BELAN, Henri Carlo. Formalização da rede de petri Canal/Agência para projeto de equipamentos industriais. Dissertação de mestrado em Engenharia Mecânica. Universidade Federal de Santa Catarina (UFSC), Florianópolis, SC, 2007.
- BERNARD, V.; CAUFFRIEZ, L.; RENAUX, D.; The Safe-SADT method for aiding designers to choose and improve dependable architectures for complex automated systems. Reliability Engineering and System Safety. Vol. 93, Elsevier, 2008. p. 179-196.
- BEZERRA, Eduardo. Princípios de Análise e Projeto de Sistemas com UML. 3. ed. Rio de Janeiro: Elsevier, 2015.
- BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. UML Guia do Usuário. 2. ed. Rio de Janeiro: Elsevier, 2005.
- CAI, Baoping; LIU, Yonghong; LIU, Zengkai; TIAN, Xiaojie; DONG, Xin; YU, Shilin. Using Bayesian networks in reliability evaluation for subsea blowout preventer control system. Reliability Engineering and System Safety. Vol. 108, Elsevier, 2012, p. 32-41.
- CHU, Yuanbo; YUAN, Zhaohui; CHEN, Jia. Research on dynamic reliability of a jet pipe servo valve based on generalized stochastic Petri nets. International Journal of Aerospace Engineering. Hindawi. 2015.
- DE NEGRI, Victor Juliano. Estruturação da modelagem de sistemas automáticos e sua aplicação a um banco de testes para sistemas hidráulicos. Tese de Doutorado em Engenharia Mecânica. Universidade Federal de Santa Catarina (UFSC), Florianópolis, SC, 1996.
- DE NEGRI, V.J.; SANTOS, E.A.P.. Projeto de sistemas de automação da manufatura. In: AGUIRRE, Luis Antonio *et al.*. Enciclopédia de Automática: Controle e Automação. Vol. 1, São Paulo: Bluncher, 2007. cap.15, p.382-417.



DISTEFANO, Salvatore; PULIAFITO, Antonio. Reliability and availability analysis of dependent-dynamic system with DRBDs. Reliability Engineering and System Safety. Vol. 94, Elsevier, 2009, p. 1381-1393.

DOMINGUEZ-GARCIA, Alejandro; KASSAKIAN, John G.; SCHONDALL, Joel E.; ZINCHUK, Jeffrey. An integrated methodology for dynamic performance and reliability evaluation of fault-tolerant systems. Reliability Engineering and System Safety. Vol. 93, Elsevier, 2008, p. 1628-1649.

FAZLOLLAHTABAR, Hamed; SAIDI-MEHRABAD, Mohammad; BALAKRISHNAN, Jaydeep. Integrated Markov-neural reliability computation method: A case for multiple automated guided vehicle system. Reliability Engineering and System Safety. Vol. 135, Elsevier, 2015, p. 34-44.

FOGLIATTO, F.S.; RIBEIRO, J.L.D. Confiabilidade e Manutenção Industrial. 1 ed. Rio de Janeiro: Elsevier, 2009. cap.1, p.3-22.

GIL, Antonio Carlos. Como elaborar projetos de pesquisa. 4 ed. São Paulo: Atlas, 2002. cap. 4, p.41-56.

GUO, Haitao; YANG, Xianhui. Automatic creation of Markov models for reliability assessment of safety instrumented systems. Reliability Engineering and System Safety. Vol. 93, Elsevier, 2008, p. 807-815.

HUANG, Chi-Yu; CHANG, Yung-Ruei. An improved decomposition scheme for assessing the reliability of embedded systems by using dynamic fault trees. Reliability Engineering and safety systems. Vol. 92, Elsevier, 2007, p. 1403-1412.

LOPES, Luciano de; GRANDINETTI, Francisco José; MARCELINO, Marcio Abud. Aplicação do GRAFCET na redução do retrabalho de uma linha de pintura automotiva. Produto & Produção. Vol. 9, UFRGS, 2008, p. 03-13.

MARQUEZ, David; NEIL, Martin; FENTON, Norman. Improved reliability modeling using Bayesian Networks and dynamic discretization. Reliability Engineering and System Safety. Vol. 95, Elsevier, 2010, p. 412-425.

MASSOCO, Daniel Biazus. Inspeção de Pulverizadores costais: Situação dos equipamentos utilizados pela agricultura familiar. Tese de doutorado em Agronomia. Universidade Federal de Pelotas (UFPel), Pelotas, RS, 2016.

PARKER. Tecnologia Pneumática industrial. Apostila M 1001-1. Elyria, 2010, 195p.

PEQUENO, D.A.C. Acionamentos Hidráulicos e Pneumáticos. Apostila de Hidráulica e Pneumática. CEFET Ceará, 2008, 171p.

PORCIÚNCULA, Gilson Simões; DE NEGRI, Victor J.; DIAS, Acires. Reliability of electro-hydraulic equipment: systematization and analysis. In: 18th International Congress of Mechanical Engineering, 2005, Ouro Preto. Anais...Ouro Preto: ABCM, 2005.

PORCIÚNCULA, Gilson Simões. Metodologia para análise de confiabilidade no projeto de sistemas automáticos. Tese de doutorado em Engenharia Mecânica. Universidade Federal de Santa Catarina (UFSC), Florianópolis, SC, 2009.

PORCIÚNCULA, Gilson Simões; BELAN, Henri Carlo; DE NEGRI, Victor Juliano; DIAS, Acires. Identification of operational configuration of automatic systems for the design for reliability. Journal of the Brazilian Society of Mechanical Sciences and Engineering, Springer, Março 2015. Disponível em: <[https://www.researchgate.net/publication/277579979\\_Identification\\_of\\_the\\_operational\\_configurations\\_of\\_automatic\\_systems\\_for\\_the\\_design\\_for\\_reliability](https://www.researchgate.net/publication/277579979_Identification_of_the_operational_configurations_of_automatic_systems_for_the_design_for_reliability)>. Acesso em: 26 maio 2016.

REIBMAN, A.L.; VEEREREGHAVAN, M. Reliability Modeling: An overview for system designers. IEEE Computer Society. Vol. 24, IEEE, 1991. p. 49-57.

RIASCOS Luis Alberto Martínez. Detecção e tratamento de falhas em sistemas de manufaturas através de redes de petri. Tese de doutorado em Engenharia Mecânica. Escola Politécnica da USP, São Paulo, SP. 2002, 160p.

RIASCOS, Luis Alberto Martínez; COZMAN, Fábio G.; MIYAGI, Paulo E. Detection and Treatment of faults in Automated Machines based on Petri Nets and Bayesian Networks. IEEE Industrial Electronics. Vol. 2, IEEE, 2003. p. 792-734.

SADOU, N.; DEMMOU, H. Reliability analysis of discrete event dynamic systems with Petri nets. Reliability Engineering and System Safety. Vol. 94, Elsevier, 2009. p. 1848-1861.

SANTOS, E.A.P. Contribuições ao projeto conceitual de sistemas de manipulação e montagem automatizados. Tese de doutorado em Engenharia Elétrica. Universidade Federal de Santa Catarina (UFSC), Florianópolis, SC, 2003.

SOUTO, R.B. Projeto de sistemas automáticos com modelagem e controle da comunicação com o ambiente externo. Dissertação de mestrado em Engenharia Mecânica. Universidade Federal de Santa Catarina (UFSC), Florianópolis, SC, 2005.

STEWART, H.L. Pneumática & Hidráulica. 3 ed, Curitiba: Hemus, 2002. cap.1, p.3-22.

TELES, Vinícius Manhães. Programação em Par. Outubro 2006. Disponível em <[http://www.desenvolvimentoagil.com.br/xp/praticas/programacao\\_par](http://www.desenvolvimentoagil.com.br/xp/praticas/programacao_par)>. Acesso em 05 de Novembro de 2016.

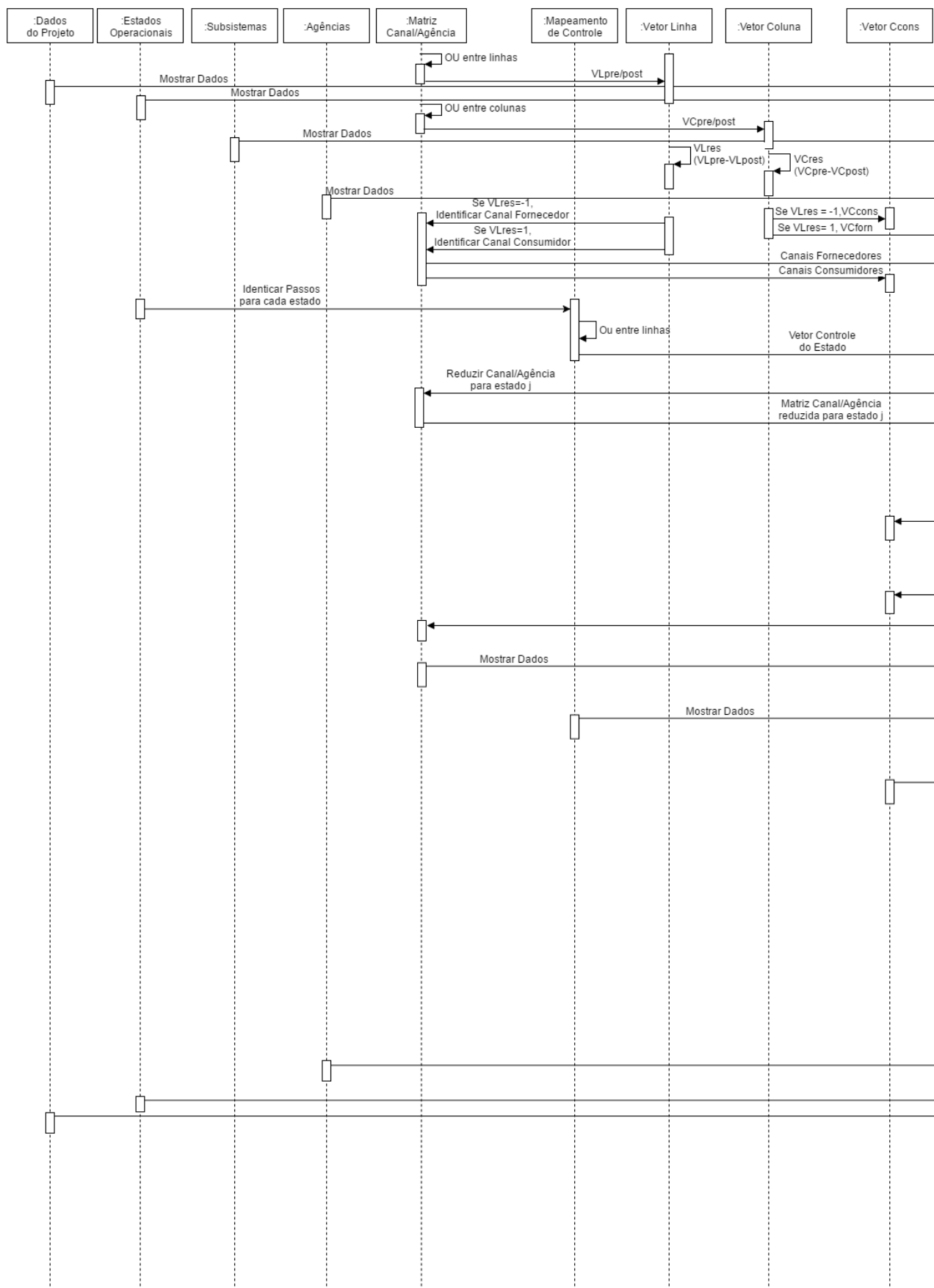
TONSIG, Sérgio Luiz. Engenharia de Software: Análise e Projeto de Sistemas. 2. ed. Rio de Janeiro: Ciência Moderna Ltda., 2008.

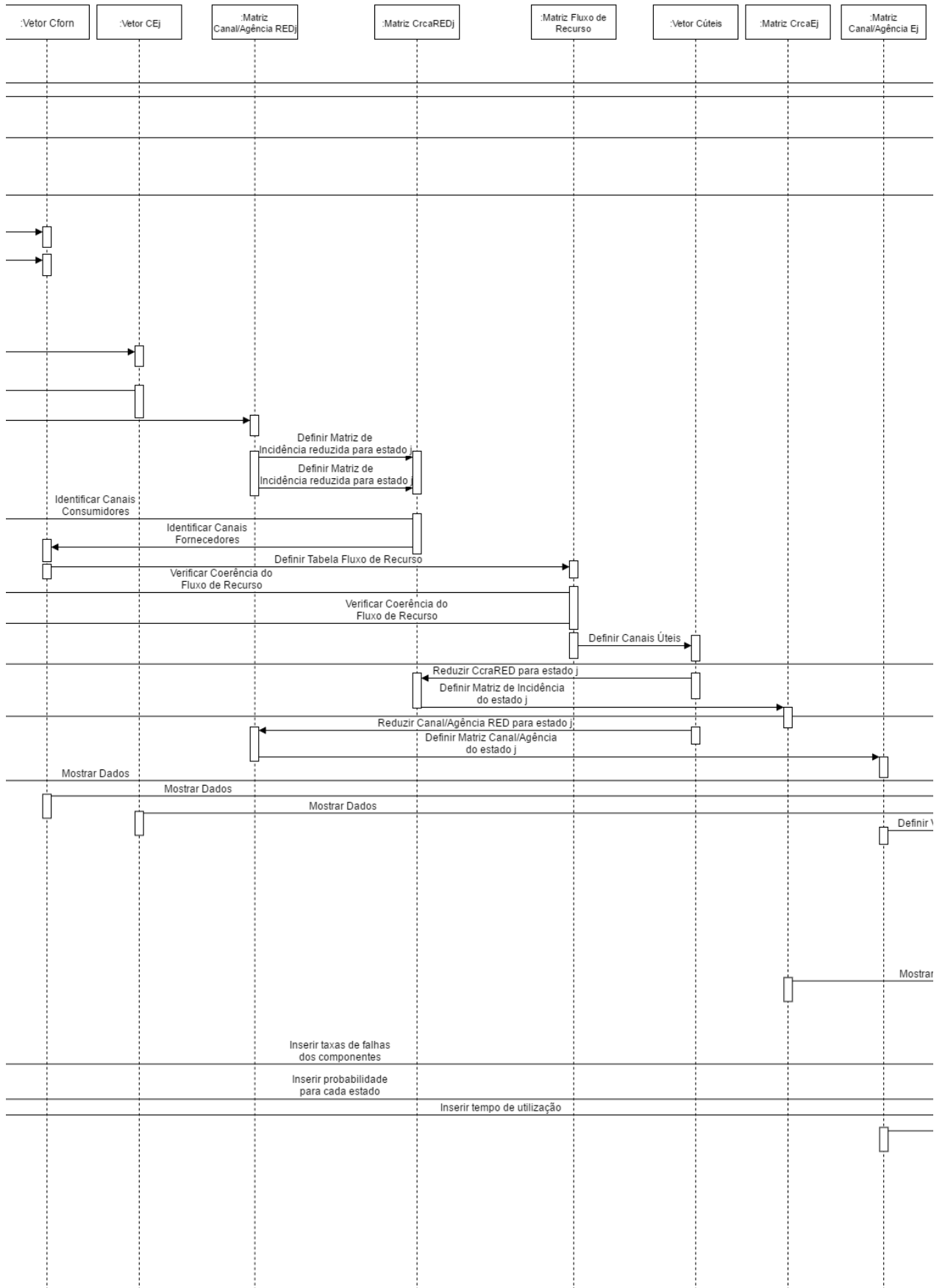
UTTERBACK, Benjamin. O que é o Bootstrap? Verdades e mitos: Parte 1 e 2. Março 2014. Disponível em <https://www.prestashop.com/blog/pt/2014/03/06/o-que-e-o-bootstrap-verdades-e-mitos-parte-1-de-2/>. Acesso em 04 de Novembro de 2016.

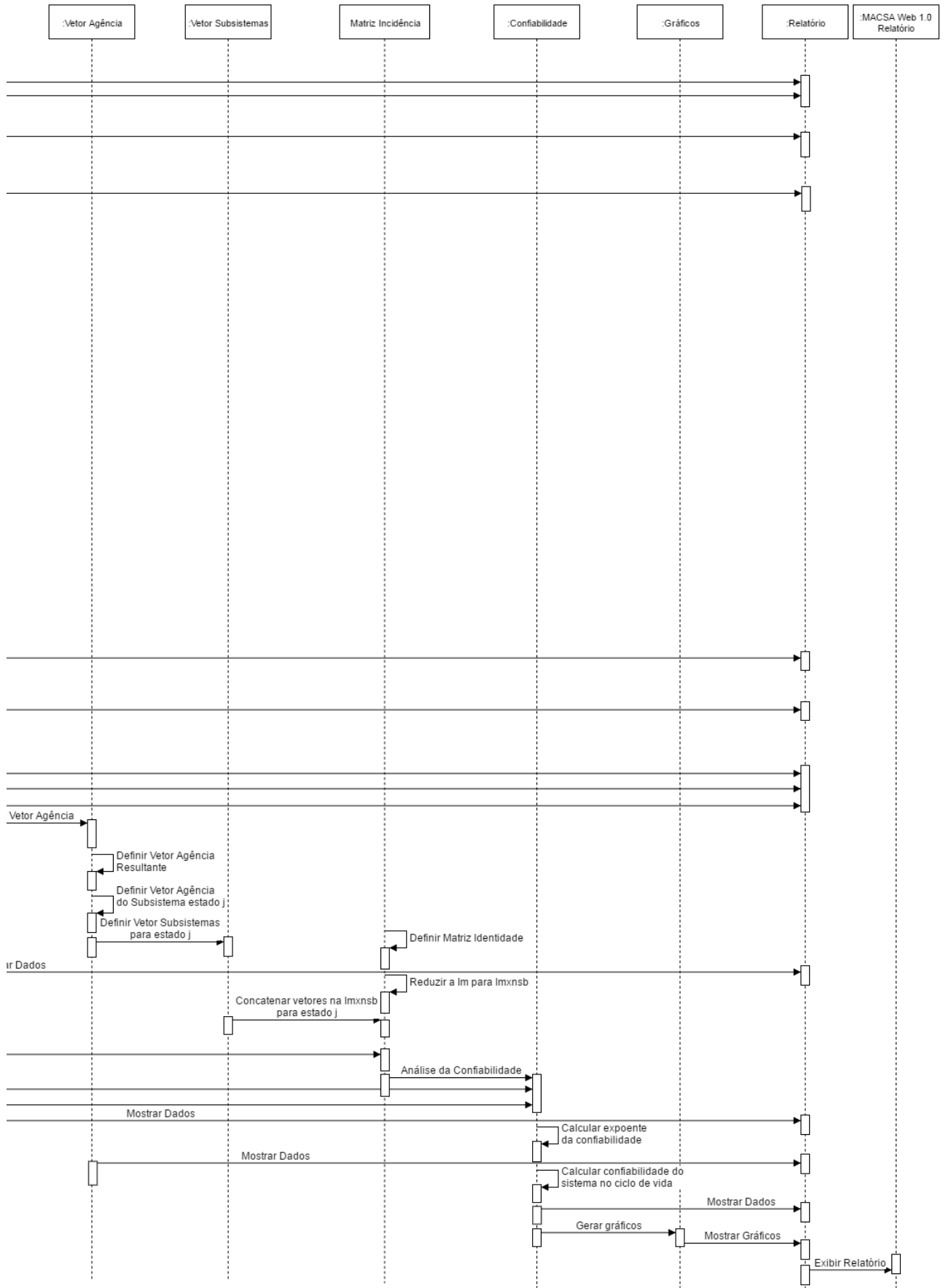
VILELA, Paulo Sérgio; VIDAL, Francisco José Targino. Automação Industrial. Maio 2003. Disponível em: <[http://www.dca.ufrn.br/~affonso/FTP/DCA447/trabalho1/trabalho1\\_19.pdf](http://www.dca.ufrn.br/~affonso/FTP/DCA447/trabalho1/trabalho1_19.pdf)>. Acesso em 21 de Abril 2016.

VINADÉ, Cesar Augusto do Canto. Sistematização do processo de projeto para confiabilidade e manutenibilidade aplicada a sistemas hidráulicos e implementação de um sistema especialista. Tese de doutorado em Engenharia Mecânica. Universidade Federal de Santa Catarina (UFSC), Florianópolis, SC, 2003.

# APÊNDICE A







## APÊNDICE B

### Dados do Projeto

<b>Nome do Projeto:</b> Estudo de Caso TCC 2	<b>Nome do Sistema:</b> Sistema Eletropneumático	<b>Nº de Subsistemas:</b> 3	<b>Tempo de Utilização:</b> 3000
<b>Projetista Responsável:</b> Monica Pacheco	<b>Função do Sistema:</b> Ensaiai válvulas de fechamento de pulverizadores costais	<b>Nº de Estados:</b> 3	<b>Nº de Passos:</b> 7

### Estados Operacionais

Número	Código	Nome	Função	Passos do Estado	Probabilidade
1	E1	Estado 1	Ensaiai gatilho 1	1.1;1.2	0.5
2	E2	Estado 2	Ensaiai gatilho 2	2.1;2.2	0.1
3	E3	Estado 3	Ensaiai gatilho 1 e 2	3.1;3.2	0.4

### Subsistemas

Agencia	Código	Nome	Função	Nº de Agências	Nº de Canais	Nível
A0	SE	Sistema Eletropneumático	Fornecer energia mecânica	2	10	1
A01	SC	Subsistema de Controle	Traduzir a configuração desejada pelo usuário para o SP	15	22	2
A02	SP	Subsistema Pneumático	Transformar energia elétrica em energia pneumática	10	21	2

### Subsistema A0

VETOR AGÊNCIA

-1

-1

### Subsistema A01

CANAIS FORNECEDORES

C20

C25

C29

C34

C38

C40

C41

CANAIS CONSUMIDORES

C24

C28

C33

C37

MATRIZ DE INCIDENCIA DO ESTADO 1

	a011	a012	a013	a014	a015	a016	a017	a018
C20	-1	0	0	0	0	0	0	0
C21	1	-1	0	0	0	0	0	0
C22	0	1	-1	0	0	0	0	0
C23	0	0	1	-1	0	0	0	0
C24	0	0	0	1	0	0	0	0
C25	0	0	0	0	-1	0	0	0
C26	0	0	0	0	1	-1	0	0
C27	0	0	0	0	0	1	-1	0
C28	0	0	0	0	0	0	1	0
C29	0	0	0	0	0	0	0	-1
C30	0	0	-1	0	0	0	0	1
C40	0	-1	0	0	0	0	0	0

MATRIZ DE INCIDENCIA DO ESTADO 2

	a011	a019	a0110	a0111	a0112	a0113	a0114	a0115
C20	-1	0	0	0	0	0	0	0
C21	1	-1	0	0	0	0	0	0
C31	0	1	-1	0	0	0	0	0
C32	0	0	1	-1	0	0	0	0
C33	0	0	0	1	0	0	0	0
C34	0	0	0	0	-1	0	0	0
C35	0	0	0	0	1	-1	0	0
C36	0	0	0	0	0	1	-1	0
C37	0	0	0	0	0	0	1	0
C38	0	0	0	0	0	0	0	-1
C39	0	0	-1	0	0	0	0	1
C41	0	-1	0	0	0	0	0	0



MATRIZ DE INCIDENCIA DO ESTADO 3

	a011	a012	a013	a014	a015	a016	a017	a018	a019	a0110	a0111	a0112	a0113	a0114	a0115
C20	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C21	1	-1	0	0	0	0	0	0	-1	0	0	0	0	0	0
C22	0	1	-1	0	0	0	0	0	0	0	0	0	0	0	0
C23	0	0	1	-1	0	0	0	0	0	0	0	0	0	0	0
C24	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
C25	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0
C26	0	0	0	0	1	-1	0	0	0	0	0	0	0	0	0
C27	0	0	0	0	0	1	-1	0	0	0	0	0	0	0	0
C28	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
C29	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0
C30	0	0	-1	0	0	0	0	1	0	0	0	0	0	0	0
C31	0	0	0	0	0	0	0	0	1	-1	0	0	0	0	0
C32	0	0	0	0	0	0	0	0	0	1	-1	0	0	0	0
C33	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
C34	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0
C35	0	0	0	0	0	0	0	0	0	0	0	1	-1	0	0
C36	0	0	0	0	0	0	0	0	0	0	0	0	1	-1	0
C37	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
C38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1
C39	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	1
C40	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0
C41	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0

## CONFIABILIDADE DO SUBSISTEMA

Agência	Símbolo	Nome	Taxa de falha	Estado 1	Estado 2	Estado 3	Tempo de uso efetivo	Confiabilidade MACSA $[\Lambda^* \sum p(E)^*t]$	Confiabilidade Estado Unico $[\Lambda^*t]$
a011	FT	Fonte estabilizadora 24 volts	0.0000000009	0.5	0.1	0.4	3000	-2.7E-6	-2.7E-6
a012	B1	Botão com trava 1	0.00000187	0.5	0	0.4	2700	-0.005049	-0.00561
a013	K1	Relé	0.00000003	0.5	0	0.4	2700	-0.00081	-0.0009
a014	Y1	Solenóide da VD01	0.00000003	0.5	0	0.4	2700	-0.00081	-0.0009
a015	A+	Sensor eletromagnético do A01	0.0000233	0.5	0	0.4	2700	-0.06291	-0.0699
a016	K2	Relé	0.00000003	0.5	0	0.4	2700	-0.00081	-0.0009
a017	Y2	Solenóide da VD01	0.00000003	0.5	0	0.4	2700	-0.00081	-0.0009
a018	A-	Sensor eletromagnético do A01	0.0000233	0.5	0	0.4	2700	-0.06291	-0.0699
a019	B2	Botão com trava 2	0.00000187	0	0.1	0.4	1500	-0.002805	-0.00561
a0110	K3	Relé	0.00000003	0	0.1	0.4	1500	-0.00045	-0.0009
a0111	Y3	Solenóide da VD02	0.00000003	0	0.1	0.4	1500	-0.00045	-0.0009
a0112	B+	Sensor eletromagnético do A02	0.0000233	0	0.1	0.4	1500	-0.03495	-0.0699
a0113	K4	Relé	0.00000003	0	0.1	0.4	1500	-0.00045	-0.0009
a0114	Y4	Solenóide da VD02	0.00000003	0	0.1	0.4	1500	-0.00045	-0.0009
a0115	B-	Sensor eletromagnético do A02	0.0000233	0	0.1	0.4	1500	-0.03495	-0.0699

## TEMPO DE USO EFETIVO DOS COMPONENTES



## Subsistema A02

### CANAIS FORNECEDORES

C1	C24	C28	C33	C37
----	-----	-----	-----	-----

### CANAIS CONSUMIDORES

C12	C13	C25	C29	C34	C38
-----	-----	-----	-----	-----	-----

### MATRIZ DE INCIDENCIA DO ESTADO 1

	a021	a022	a023	a025	a026	a029
C1	-1	0	0	0	0	0
C2	1	-1	0	0	0	0
C3	0	1	-1	0	0	0
C4	0	0	1	-1	0	0
C5	0	0	1	0	-1	0
C8	0	0	0	1	0	-1
C9	0	0	0	0	1	-1
C12	0	0	0	0	0	1
C24	0	0	-1	0	0	0
C25	0	0	0	0	0	1
C28	0	0	-1	0	0	0
C29	0	0	0	0	0	1

### MATRIZ DE INCIDENCIA DO ESTADO 2

	a021	a022	a024	a027	a028	a0210
C1	-1	0	0	0	0	0
C2	1	-1	0	0	0	0
C3	0	1	-1	0	0	0
C6	0	0	1	-1	0	0
C7	0	0	1	0	-1	0
C10	0	0	0	1	0	-1
C11	0	0	0	0	1	-1
C13	0	0	0	0	0	1
C33	0	0	-1	0	0	0
C34	0	0	0	0	0	1
C37	0	0	-1	0	0	0
C38	0	0	0	0	0	1

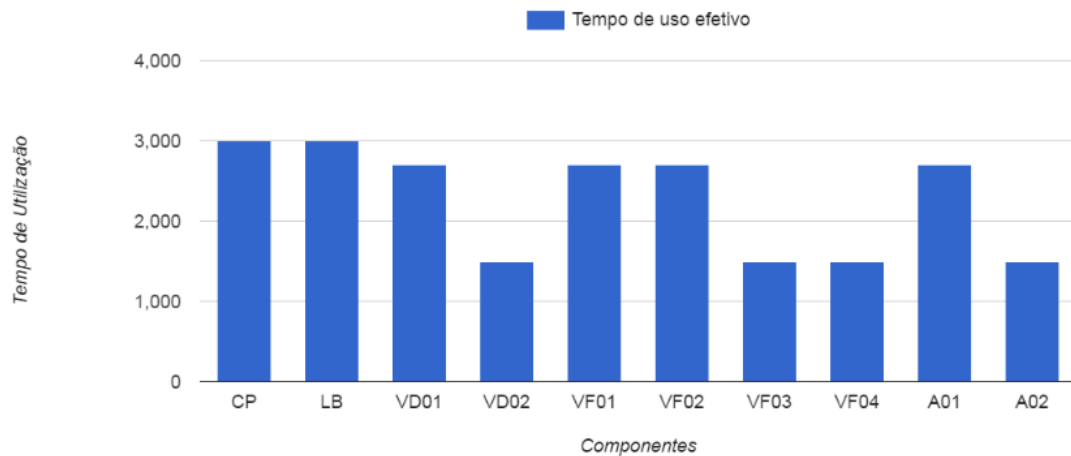
MATRIZ DE INCIDENCIA DO ESTADO 3

	a021	a022	a023	a024	a025	a026	a027	a028	a029	a0210
C1	-1	0	0	0	0	0	0	0	0	0
C2	1	-1	0	0	0	0	0	0	0	0
C3	0	1	-1	-1	0	0	0	0	0	0
C4	0	0	1	0	-1	0	0	0	0	0
C5	0	0	1	0	0	-1	0	0	0	0
C6	0	0	0	1	0	0	-1	0	0	0
C7	0	0	0	1	0	0	0	-1	0	0
C8	0	0	0	0	1	0	0	0	-1	0
C9	0	0	0	0	0	1	0	0	-1	0
C10	0	0	0	0	0	0	1	0	0	-1
C11	0	0	0	0	0	0	0	1	0	-1
C12	0	0	0	0	0	0	0	0	1	0
C13	0	0	0	0	0	0	0	0	0	1
C24	0	0	-1	0	0	0	0	0	0	0
C25	0	0	0	0	0	0	0	0	1	0
C28	0	0	-1	0	0	0	0	0	0	0
C29	0	0	0	0	0	0	0	0	1	0
C33	0	0	0	-1	0	0	0	0	0	0
C34	0	0	0	0	0	0	0	0	0	1
C37	0	0	0	-1	0	0	0	0	0	0
C38	0	0	0	0	0	0	0	0	0	1

## CONFIABILIDADE DO SUBSISTEMA

Agência	Símbolo	Nome	Taxa de falha	Estado 1	Estado 2	Estado 3	Tempo de uso efetivo	Confiabilidade MACSA $[\lambda \sum p(E) \cdot t]$	Confiabilidade Estado Único $[\lambda \cdot t]$
a021	CP	Compressor	0.00000313	0.5	0.1	0.4	3000	-0.00939	-0.00939
a022	LB	Lubrífil	0.0000178	0.5	0.1	0.4	3000	-0.0534	-0.0534
a023	VD01	Válvula Direcional 01	0.00000385	0.5	0	0.4	2700	-0.010395	-0.01155
a024	VD02	Válvula Direcional 02	0.00000385	0	0.1	0.4	1500	-0.005775	-0.01155
a025	VF01	Válvula de Fluxo 01	0.00000313	0.5	0	0.4	2700	-0.008451	-0.00939
a026	VF02	Válvula de Fluxo 02	0.00000313	0.5	0	0.4	2700	-0.008451	-0.00939
a027	VF03	Válvula de Fluxo 03	0.00000313	0	0.1	0.4	1500	-0.004695	-0.00939
a028	VF04	Válvula de Fluxo 04	0.00000313	0	0.1	0.4	1500	-0.004695	-0.00939
a029	A01	Atuador linear 01	0.00000858	0.5	0	0.4	2700	-0.023166	-0.02574
a0210	A02	Atuador linear 02	0.00000858	0	0.1	0.4	1500	-0.01287	-0.02574

## TEMPO DE USO EFETIVO DOS COMPONENTES

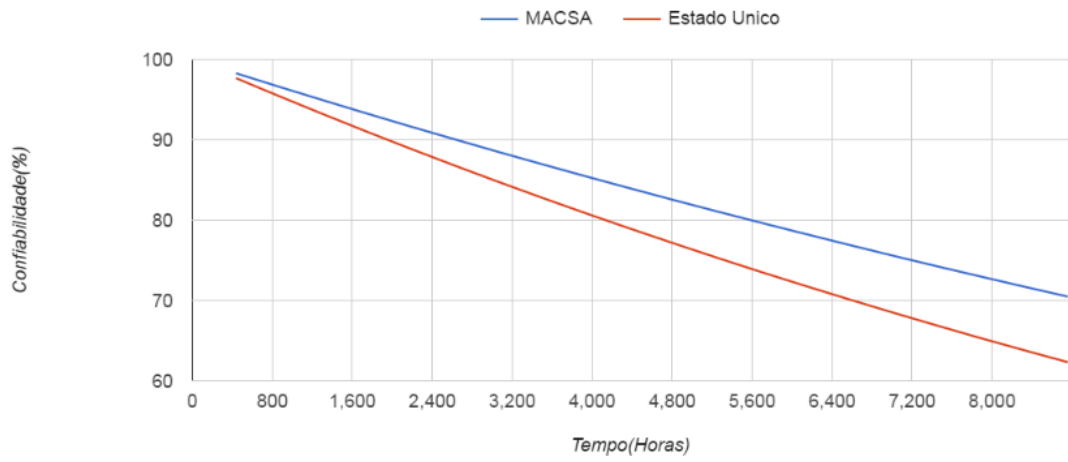


## CONFIABILIDADE DO SISTEMA

Confiabilidade MACSA =  $\exp\{-\sum[\lambda \cdot \sum p(E) \cdot t]\}$  : 0.70475524969379

Confiabilidade Estado Único =  $\exp\{-\sum[\lambda \cdot t]\}$  : 0.6231595459271

COMPARAÇÃO ENTRE AS ESTIMATIVAS DE CONFIABILIDADE



PERCENTUAL DE COMPONENTES ATIVOS CONFORME TEMPO DE USO EFETIVO

