

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DEPARTAMENTO ACADÊMICO DE ELETROTÉCNICA  
CURSO DE ENGENHARIA DE CONTROLE E AUTOMAÇÃO

MARKOS FLAVIO BOCK GAU DE OLIVEIRA

**SISTEMA DE DIAGNÓSTICO DE MÚLTIPLAS FALHAS EM VEÍCULOS  
AUTOMOTIVOS**

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA

2016

MARKOS FLAVIO BOCK GAU DE OLIVEIRA

**SISTEMA DE DIAGNÓSTICO DE MÚLTIPLAS FALHAS EM VEÍCULOS  
AUTOMOTIVOS**

Trabalho de Conclusão de Curso de Graduação, apresentado à disciplina de Trabalho de Diplomação II, do curso de Engenharia de Controle e Automação do Departamento Acadêmico de Eletrotécnica (DAELT) da Universidade Tecnológica Federal do Paraná (UTFPR), como requisito parcial para obtenção do título de Engenheiro de Controle e Automação.

Orientador: Prof. Dr. Ricardo Lüders.

CURITIBA

2016

**Markos Flavio Bock Gau de Oliveira**

## **SISTEMA DE DIAGNÓSTICO DE MÚLTIPLAS FALHAS EM VEÍCULOS AUTOMOTIVOS**

Este Trabalho de Conclusão de Curso de Graduação foi julgado e aprovado como requisito parcial para a obtenção do Título de Engenheiro de Controle e Automação do curso de Engenharia de Controle e Automação do Departamento Acadêmico de Eletrotécnica (DAELT) da Universidade Tecnológica Federal do Paraná (UTFPR).

Curitiba, 23 de novembro de 2016.

---

Prof. Paulo Sérgio Walenia, Esp.  
Coordenador de Curso  
Engenharia de Controle e Automação

---

Prof. Marcelo Rosa, Dr.  
Responsável pelos Trabalhos de Conclusão de Curso  
de Engenharia de Controle e Automação do DAELT

### **ORIENTAÇÃO**

---

Ricardo Lüders, Dr.  
Universidade Tecnológica Federal do Paraná  
Orientador

---

Ricardo Umbria Pedroni, Me.  
Universidade Tecnológica Federal do Paraná  
Co-Orientador

### **BANCA EXAMINADORA**

---

Ricardo Lüders, Dr.  
Universidade Tecnológica Federal do Paraná

---

Ricardo Umbria Pedroni, Me.  
Universidade Tecnológica Federal do Paraná

---

Keiko Verônica Ono Fonseca, Dr.  
Universidade Tecnológica Federal do Paraná

A folha de aprovação assinada encontra-se na Coordenação do Curso de Engenharia de Controle e Automação

## RESUMO

OLIVEIRA, M. F. B. G. Sistema de Diagnóstico de Múltiplas Falhas em Veículos Automotivos. Monografia (Curso Bacharelado em Engenharia de Controle e Automação) – Universidade Tecnológica Federal do Paraná. Curitiba, 2016.

Os sistemas veiculares atuais compreendem sistemas *on-line* de monitoramento e detecção de falhas. Quando uma falha é detectada, um código de falha particular é ativado e armazenado para posterior análise. Diagramas binários de decisão codificam testes a serem realizados para cada código de falha ativo. Entretanto, o diagnóstico desses sistemas tem se tornado cada vez mais complexo devido à crescente eletrônica embarcada. Vários códigos de falha podem ser ativados por uma mesma causa, o que dificulta o procedimento de rastreamento da falha pelo especialista de manutenção. Esse trabalho tem por objetivo desenvolver uma ferramenta em *software* para combinar diagramas de decisão binários, propondo uma reordenação dos testes na presença de múltiplos códigos ativos, de forma a construir um diagrama único de testes com menor custo de diagnóstico. Nessa abordagem, o diagrama preferencia testes com baixo custo e alta probabilidade de falha. O *software* desenvolvido é um teste de conceito, no qual é possível experimentar a estratégia proposta em diversos cenários, de forma bem similar ao que ocorre nas ferramentas de diagnóstico guiado existentes. Resultados preliminares mostram uma redução significativa do custo de diagnóstico da estratégia combinada com relação à convencional.

**Palavras-Chave:** Diagnóstico de Falhas; Análise de Árvores de Falha; Diagrama de Decisão Binário; Sistemas Automotivos.

## ABSTRACT

OLIVEIRA, M. F. B. G. Automotive Diagnostic System for Multiple Faults. Monografia (Curso Bacharelado em Engenharia de Controle e Automação) – Universidade Tecnológica Federal do Paraná. Curitiba, 2016.

On-line fault detection and monitoring systems are embedded in current automotive systems. In the presence of a fault, a particular diagnostic trouble code (DTC) is triggered and logged for further analysis. Binary decision diagrams encode tests to be performed for each logged code. However, due to increased automotive embedded electronics, several DTCs can be triggered by the same root cause, which makes the *off-board* diagnostic procedure a challenging task. This work aims to develop a software tool to combine individual binary decision diagrams in presence of multiple active DTCs, in order to build a single binary decision diagram with lower diagnostic cost. In this approach, the binary decision diagram prioritizes tests in components with lower cost and high fault probability. The developed software is a proof of concept, in which it's possible to apply the proposed strategy in different scenarios, in a similar way it occurs in commercial diagnostic tools. Preliminary results show a significant reduction in the diagnostic cost of the combined strategy over the standard approach.

**Keywords:** Fault Diagnosis; Fault Tree Analysis; Binary Decision Diagram; Automotive Systems.

## LISTA DE ILUSTRAÇÕES

FIGURA 1 - Comunicação eletrônica de um Mercedes E-Class modelos W210 (1995) e W211 (2002).....	20
FIGURA 2 - Evolução dos sistemas eletrônicos automotivos.....	21
FIGURA 3 - Topologia de comunicação de um sistema automotivo.....	22
FIGURA 4 - Sistema automotivo para o controle de emissão de gases .....	27
FIGURA 5 - <i>Scanner</i> automotivo OBD II CP9575 da Actron .....	30
FIGURA 6 - Conector OBD II do veículo Nissan Sentra, modelo 2005.....	31
FIGURA 7 - <i>Layout</i> padrão do conector de diagnóstico OBD II de 16 pinos.....	31
FIGURA 8 - Árvore de falhas de um farol automotivo.....	33
FIGURA 9 - Parte da árvore de diagnóstico do DTC P1212 da F350.....	35
FIGURA 10 - Diagrama de decisão binário .....	36
FIGURA 11 - Árvore de falhas e BDD associado.....	38
FIGURA 12 - Análise de falhas e seus objetivos no diagnóstico veicular .....	39
FIGURA 13 - Relação entre a falha ocorrida e o DTC ativado.....	40
FIGURA 14 - Diagnóstico automotivo com múltiplos DTCs ativos.....	41
FIGURA 15 - Diferentes modelos de lâmpadas MIL.....	42
FIGURA 16 - Procedimento de diagnóstico a partir do sinal de falha identificado .....	46
FIGURA 17 - Cenário de diagnóstico com múltiplos DTCs ativos .....	46
FIGURA 18 - Árvore de falhas e BDD para exemplificação dos cortes mínimos.....	48
FIGURA 19 - Tabela verdade da equação booleana $S = a + bc$ .....	54
FIGURA 20 - Auxílio ao diagnóstico realizado pelo SOAD .....	57
FIGURA 21 - Figura 21. Módulos funcionais presentes no SOAD .....	59
FIGURA 22 - Diagrama UML das principais classes existentes no SOAD.....	61
FIGURA 23 - Painel de fabricantes e informações relacionadas .....	63
FIGURA 24 - Janela de especificação de informações para criação do BDD.....	65
FIGURA 25 - Janela de cadastro de testes.....	67
FIGURA 26 - Janela de cadastro de componentes .....	67

FIGURA 27 - Painel de cadastro do BDD.....	68
FIGURA 28 - BDD em formato XML gerado pelo SOAD.....	69
FIGURA 29 - Painel de cadastro do BDD via equação booleana .....	71
FIGURA 30 - BDDs para diferentes métricas de ordenação.....	72
FIGURA 31 - Janela de criação de projetos de diagnóstico .....	73
FIGURA 32 - Tabela Projetos existente no banco de dados SQL.....	74
FIGURA 33 - Tabela de histórico de procedimentos de um projeto de diagnóstico.....	74
FIGURA 34 - Painel que lista os projetos cadastrados.....	76
FIGURA 35 - Painel do histórico de falhas de um exemplo de projeto .....	77
FIGURA 36 - Painel que lista os BDD's cadastrados.....	78
FIGURA 37 - Janela de confirmação para exclusão de um BDD .....	78
FIGURA 38 - Painel de visualização do BDD .....	79
FIGURA 39 - Painel que lista os testes cadastrados.....	80
FIGURA 40 - Janela de visualização e edição de testes.....	81
FIGURA 41 - Painel que lista os componentes cadastrados.....	81
FIGURA 42 - Especificações do BDD associado ao DTC P0100.....	83
FIGURA 43 - Especificações do BDD associado ao DTC P0104.....	83
FIGURA 44 - Especificações do BDD associado ao DTC P1100.....	84
FIGURA 45 - Especificações do BDD associado ao DTC P1102.....	84
FIGURA 46 - Principais especificações dos BDDs individuais presentes no estudo de caso.....	85
FIGURA 47 - Especificações dos testes presentes no estudo de caso .....	85
FIGURA 48 - Cenário de diagnóstico do estudo de caso .....	86
FIGURA 49 - BDD resultante após a reconfiguração dos testes.....	87
FIGURA 50 - Janela com especificações do teste T4.....	88
FIGURA 51 - Painel de diagnóstico após o teste T4.....	89
FIGURA 52 - Painel de diagnóstico após o teste T3 .....	90
FIGURA 53 - Janela que solicita a informação dos códigos remanescentes.....	90
FIGURA 54 - BDD combinada com nova ordenação .....	92
FIGURA 55 - Painel com resultados do diagnóstico finalizado.....	92

## LISTA DE TABELAS

TABELA 1 - Prefixos dos DTCs especificados pela ISO 15031-6.....	23
TABELA 2 - Definição de Categoria de Falha dos DTCs.....	24
TABELA 3 - Descrição dos FTBs .....	25



## LISTA DE SIGLAS

API – *Application Programming Interface*

BDD – *Binary Decision Diagram*

CAN – *Controller Area Network*

CDIF – *Cost of Diagnostic Importance Factor*

DIF – *Diagnostic Importance Factor*

DTC – *Diagnostic Trouble Code*

ECU – *Electronic Control Unit*

EOBD – *European On-Board Diagnostics*

EVAP – *Evaporative Emission Control System*

FMEA – *Failure Mode and Effect Analysis*

FT – *Fault Tree*

FTA – *Fault Tree Analysis*

FTB – *Failure Type Byte*

ICP – *Injection Control Pressure*

JDBC – *Java Database Connectivity*

MIL – *Malfunction Indication Lamp*

PC – *Personal Computer*

PDF – *Portable Document Format*

OBD – *On-Board Diagnostic*

OSI – *Open Systems Interconnection*

SAE – *Society of Automotive Engineer*

SQL – *Structured Query Language*

UML – *Unified Modeling Language*

XML – *Extensible Markup Language*

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO.....</b>	<b>11</b>
1.1	APRESENTAÇÃO DO TEMA .....	11
1.1.1	Delimitação do Tema.....	13
1.2	PROBLEMAS E PREMISSAS.....	13
1.3	OBJETIVOS.....	14
1.3.1	Objetivo Geral .....	14
1.3.2	Objetivos específicos .....	14
1.4	JUSTIFICATIVA.....	15
1.5	PROCEDIMENTOS METODOLÓGICOS.....	15
1.6	ESTRUTURA DO TRABALHO.....	16
<b>2</b>	<b>DIAGNÓSTICO EM SISTEMAS AUTOMOTIVOS.....</b>	<b>17</b>
2.1	TAXONOMIA DO DIAGNÓSTICO AUTOMOTIVO .....	17
2.1.1	Diagnóstico <i>On-board</i> vs Diagnóstico <i>Off-board</i> .....	18
2.2	DIAGNÓSTICO VEICULAR .....	19
2.2.1	Sistemas Eletrônicos Automotivos .....	20
2.2.2	Padrões de Diagnóstico na Indústria Automotiva .....	23
2.2.3	Sistema OBD ( <i>On-Board Diagnostic System</i> ).....	26
2.2.4	<i>Scanners</i> Automotivos.....	29
2.2.5	Análise FMEA ( <i>Failure Mode and Effect Analysis</i> ) .....	32
2.2.7	Árvores de Diagnóstico .....	34
2.2.8	Procedimento de Diagnóstico na Indústria Automotiva.....	39
2.2.9	Dificuldades no Diagnóstico Atual.....	43
<b>3</b>	<b>DESENVOLVIMENTO.....</b>	<b>45</b>
3.1	ESTRATÉGIA DE SINTETIZAÇÃO PARA MÚLTIPLOS DTCs .....	45
3.1.1	Obtenção dos Cortes Mínimos.....	47
3.1.2	Concatenação dos Cortes Mínimos Individuais .....	49
3.1.3	Ordenação dos Testes .....	51
3.1.3.1	CDIF ( <i>Cost of Diagnostic Importance Factor</i> ) .....	52
3.1.3.2	IG ( <i>Information Gain</i> ) .....	53

3.1.4	Formação do BDD Resultante .....	55
3.1.5	Custo de Diagnóstico .....	55
3.2	SOAD (SISTEMA <i>OFF-LINE</i> DE AUXÍLIO AO DIAGNÓSTICO) .....	56
3.2.1	Módulos Principais do SOAD .....	58
3.2.2	Especificações de uso do SOAD .....	60
<b>4</b>	<b>RESULTADOS .....</b>	<b>62</b>
4.1	MÓDULO DADOS DE FABRICANTES .....	62
4.2	MÓDULO CONSTRUÇÃO DE BDDs .....	64
4.2.1	Construção via Diagrama .....	66
4.2.2	Construção via Equação Booleana .....	70
4.3	MÓDULO PROJETO DO DIAGNÓSTICO .....	72
4.4	MÓDULO VISUALIZAÇÃO DE DADOS .....	75
4.5	MÓDULO DE DIAGNÓSTICO GUIADO .....	82
4.5.1	Cenário de Diagnóstico.....	82
4.5.2	Relatórios .....	93
<b>5</b>	<b>CONCLUSÃO.....</b>	<b>95</b>
	<b>REFERÊNCIAS.....</b>	<b>97</b>

## 1. INTRODUÇÃO

Nos sistemas automotivos atuais, muitos parâmetros são monitorados em tempo real por diversas unidades de controle distribuídas (ECU's - *Electronic Control Units*) no veículo, que, em conjunto com sensores, chaves e atuadores formam um sistema eletrônico que realiza as diversas funções veiculares. Para satisfazer uma demanda cada vez mais exigente, essas funções vêm aumentando em número e complexidade ao longo dos anos, com aplicações na segurança, sistemas de assistência à direção e conforto.

Entretanto, quanto mais aplicações e funções são introduzidas no veículo, mais a complexidade eletrônica do sistema aumenta. Por exemplo, o custo médio associado à fração eletrônica embarcada do veículo aumentou de 2% em 1974 para 40% em 2010 sobre o preço total do veículo (SUWATTHIKUL, 2010). Além disso, um veículo comercial com apenas três ECU's em 1989, hoje possui mais de 70 ECU's com aproximadamente 10 milhões de linhas de código embarcado (FRANK; SCHMIDTS, 2006). Sabe-se também que esse número deve aumentar, tendo em vista que 90% das inovações em sistemas automotivos hoje estão no domínio da eletrônica e *software*.

Como resultado, a probabilidade de ocorrência de eventos de falha no veículo aumentou drasticamente nos últimos anos, tendo em vista o grande aumento de componentes e subsistemas independentes embarcados. Sabe-se, por exemplo, que o número de conectores e conexões elétricas aumenta exponencialmente com o número de ECU's; assim, a probabilidade de uma interrupção elétrica aumenta drasticamente. Além disso, falhas em nível de rede (comunicação) e de *software*, que antes não eram relevantes, começaram a ser modeladas e consideradas. Tudo isso obrigou as montadoras à focar nas áreas de tolerância à falhas, teste e validação, confiabilidade e diagnóstico de falhas. Esse trabalho trata do diagnóstico de falhas em sistemas automotivos.

### 1.1 APRESENTAÇÃO DO TEMA

De maneira geral, o diagnóstico de falhas pode ser genericamente definido como um conjunto de técnicas que, a partir de modelos de funcionamento normal e informações de anormalidades (efeitos observáveis da falha normalmente obtidos por sensores), identifica um ou mais componentes defeituosos (causas da falha).

Nos sistemas veiculares atuais, quando ocorre um evento de falha, a unidade de controle responsável detecta um distúrbio no sistema e ativa um código associado à esse distúrbio, chamado de DTC (*Diagnostic Trouble Code*). Esse código de falha é armazenado por unidades centrais de processamento durante a operação do veículo para posterior análise, feita por especialistas com a ajuda de ferramentas de auxílio ao diagnóstico. Essas ferramentas computacionais estão disponíveis durante a parada do veículo em manutenções preventivas ou emergenciais, caracterizando um diagnóstico *off-board*. Nessa etapa, cada DTC ativo representa uma árvore de diagnóstico que determina sequências de testes a serem executadas pelos técnicos de manutenção com o objetivo de excluir passo a passo as possíveis causas da falha até que se encontre o componente defeituoso que originou o DTC ativo. A árvore de diagnóstico é construída com base nas possíveis causas da ativação do DTC e é formada por nós que especificam testes PASS ou FAIL em um componente do veículo. Uma sequência completa de testes e de resultados permite a inferência da(s) causa(s) do DTC ativo.

Entretanto, à medida que a complexidade da interconectividade de componentes eletrônicos aumenta, abordagens *off-board* de diagnóstico vão se tornando cada vez mais custosas e algumas vezes ineficazes, resultando em altos índices de "Falha Não Encontrada", substituição incorreta de componentes e aumento de custos de garantia (SUWATTHIKUL, 2010). Assim, existem situações onde o diagnóstico não é conclusivo e o sucesso do diagnóstico (isolamento da causa da falha) depende da experiência do técnico.

Outro problema corrente devido ao aumento da complexidade eletrônica nesses sistemas é a ativação de vários códigos de falha simultaneamente. Nesse caso, quando mais de um DTC está ativo, tem-se um conjunto de árvores de diagnóstico independentes que não compartilham informações sobre seus componentes, acarretando um processo que considera todas as combinações possíveis de geração dos respectivos DTCs, abordando, na sua maioria, combinações de falha pouco prováveis de ocorrer. É mais provável, entretanto, que apenas um subconjunto mínimo de falhas à todos os DTCs tenha ocorrido, em um efeito cascata, que não é capturado pela abordagem de diagnóstico *off-board* convencional.

Nesse contexto, esse trabalho propõe o desenvolvimento de um *software* de auxílio ao diagnóstico que faz uso combinado das informações presentes nas árvores de diagnóstico individuais, num cenário onde múltiplos DTCs estão ativos. Mais especificamente, o sistema proposto tem como objetivo a síntese de uma única sequência de testes resultante a partir da combinação das árvores de diagnóstico dos códigos de falha individuais e do histórico de procedimentos de diagnóstico anteriores.

O *software* a ser desenvolvido é um teste de conceito da estratégia de diagnóstico proposta, uma vez que pacotes comerciais de diagnóstico em geral fornecem apenas o ambiente de desenvolvimento do diagnóstico *off-board*, sem suporte à decisão.

### 1.1.1 Delimitação do Tema

Desenvolvimento de uma estratégia de diagnóstico quando múltiplos códigos de falha estão ativos e implementação da estratégia em *software*.

A estratégia de diagnóstico proposta neste trabalho foi também objeto de estudo no Programa de Iniciação Científica da UTFPR, tendo gerado duas publicações científicas (OLIVEIRA; MORI; LÜDERS, 2013) e (OLIVEIRA; LÜDERS, 2016).

## 1.2 PROBLEMAS E PREMISSAS

O problema de diagnóstico envolve um produto, uma ferramenta de diagnóstico e um procedimento de reparo que pode ser entendido em cinco passos (OLIVEIRA; MORI; LÜDERS, 2013):

1. Monitoração das condições de funcionamento do veículo;
2. Identificação da falha;
3. Rastreamento da falha;
4. Reparo;

## 5. Verificação.

Esse trabalho consiste na proposta de uma estratégia que envolve os passos 2 e 3, quando múltiplos códigos de falha estão presentes, o que aumenta a complexidade do diagnóstico. Se concentra, portanto, no estudo e desenvolvimento de um sistema que recebe múltiplos códigos de falha identificados e sintetiza uma sequência de testes única, possibilitando um diagnóstico completo e mais eficiente.

## 1.3 OBJETIVOS

O objetivo principal e os específicos estão descritos abaixo.

### 1.3.1 Objetivo Geral

Desenvolver um *software* de auxílio ao diagnóstico de múltiplas falhas com aplicações na indústria automotiva.

### 1.3.2 Objetivos específicos

- Desenvolver uma estratégia de diagnóstico quando múltiplos DTCs estão ativos;
- Implementar um *software* para avaliação da estratégia proposta, contendo:
  - O cadastro das informações necessárias para diagnóstico;
  - A implementação da estratégia proposta;
  - A configuração e avaliação de casos de teste.
- Desenvolver a base de dados que armazena o histórico de procedimentos anteriormente realizados;

- Avaliar a estratégia proposta a partir de cenários simulados.

#### 1.4 JUSTIFICATIVA

O problema de diagnóstico na indústria automotiva tem por objetivo responder de forma rápida e eficiente a uma situação de anormalidade de funcionamento do veículo, identificando precisamente o(s) componente(s) defeituoso(s) e assegurando um reparo adequado. Usualmente, o diagnóstico é um processo custoso devido à complexidade crescente dos sistemas embarcados, à constante mudança no projeto dos veículos e à diversidade dos componentes empregados, assim como à falta de mão de obra especializada. Para tanto, existem ferramentas de auxílio ao diagnóstico baseadas em árvores de diagnóstico para cada código de falha.

Esse trabalho apresenta uma estratégia de fusão das árvores de diagnóstico na presença de vários códigos de falha, mantendo o procedimento padrão de desenvolvimento utilizado na indústria. Tal abordagem possibilita um diagnóstico mais eficiente, pois tende a reduzir o tempo de diagnóstico com um menor número de testes em componentes.

O diagnóstico eficiente de falhas torna-se importante para melhoria da confiabilidade (BERGER, 2002) e diminuição dos tempos de parada para manutenção e de seus custos associados.

#### 1.5 PROCEDIMENTOS METODOLÓGICOS

Definido o problema do presente trabalho, motivado pelo contexto anteriormente descrito, a pesquisa proposta é aplicada, pois objetiva primordialmente a solução de um problema real, o diagnóstico de falhas no âmbito automotivo, considerando-se múltiplos DTCs ativos.

Esta pesquisa tem cunho exploratório, tendo em vista que existem poucos estudos relacionados à reformulação de sequências de testes no contexto automotivo. Além disso, faz uso



de referencial bibliográfico publicado na literatura especializada sobre o tema em análise, pois a estratégia de síntese e posterior implementação são baseadas em materiais já elaborados.

Assim, o trabalho é composto basicamente de duas grandes fases: pesquisa e projeto; e implementação.

Na primeira fase - pesquisa e projeto - faz-se um estudo das abordagens de diagnóstico existentes bem como de algoritmos disponíveis capazes de manipular as estruturas de diagnóstico. Após isso, desenvolve-se o projeto da estratégia a ser usada na posterior implementação.

Na segunda fase - implementação - desenvolve-se a base de dados e a ferramenta de *software* que executará as funções e algoritmos desenvolvidos na fase anterior.

## 1.6 ESTRUTURA DO TRABALHO

Este trabalho é constituído por cinco capítulos complementares e integrados.

Este primeiro capítulo, apresenta uma introdução ao tema, ao problema de pesquisa, à justificativa, e aos objetivos tanto geral quanto específicos.

O segundo capítulo trata da fundamentação teórica. O capítulo se inicia com uma apresentação dos aspectos conceituais sobre o diagnóstico em sistemas automotivos. Normas e padrões da indústria também são incluídos. A seguir, conceitos preliminares relacionados à árvore de falhas, BDDs (*Binary Decision Diagrams*) e técnicas existentes de diagnóstico são discutidos.

O terceiro capítulo é dedicado ao detalhamento dos procedimentos metodológicos, em que são caracterizados a metodologia da pesquisa e o delineamento de suas etapas. Trata-se da discussão da estratégia de síntese das árvores de diagnóstico a ser realizada e a posterior implementação em *software*.

O quarto capítulo trata dos resultados obtidos. São apresentados métricas e testes necessários para avaliação da proposta de diagnóstico. Além disso, apresenta-se alguns casos de uso da ferramenta desenvolvida.

A conclusão trata das considerações finais, na qual são abordadas as contribuições da pesquisa e as sugestões para futuros trabalhos.

## 2. DIAGNÓSTICO EM SISTEMAS AUTOMOTIVOS

### 2.1 TAXONOMIA DO DIAGNÓSTICO AUTOMOTIVO

De maneira geral, a nomenclatura no contexto de diagnóstico e sistemas tolerantes a falhas é bastante diversa em diferentes comunidades. Diferentes autores possuem diferentes interpretações para os mesmos conceitos. Assim, é necessário definir uma padronização dos conceitos utilizados nesse trabalho.

Neste trabalho, os conceitos de erro (*error*), defeito (*failure*) e falha (*fault*) são obtidos de Nelson (1990). Uma falha é uma condição física anômala no sistema, ou seja, está relacionada ao universo físico (*hardware*). Uma falha pode ser causada por vários motivos, por exemplo: impacto, fadiga, deterioração, problemas de manufatura ou implementação, interferência eletromagnética, etc. Um erro é uma manifestação da falha no sistema, causando uma variação do estado lógico de algum componente do seu valor original. Nesse caso, tem-se uma anomalia no universo da informação do sistema. Logicamente, nem toda falha gera um erro no sistema. Para isso ser verdade essa deve ser “sentida” pelos elementos lógicos do sistema, em geral através de sensores. Por exemplo, uma pequena deformação na estrutura de um automóvel devido à uma colisão caracteriza uma falha. Porém não um erro, pois a mesma não é identificada pelo sistema elétrico e lógico do automóvel. Já o defeito é a impossibilidade do sistema responder corretamente a uma demanda, devido à erros internos. Novamente, nem todo erro gera um defeito.

Considere, por exemplo, um automóvel que possui uma lâmpada queimada. A falha é caracterizada pela lâmpada. O erro, por sua vez, é caracterizado pela má interpretação do sistema eletrônico quanto ao funcionamento da lâmpada, supondo que a mesma está acesa. E o defeito é caracterizado no universo do usuário, pela incorreta indicação da luz acesa no painel ao condutor.

Outra ambiguidade existente é quanto ao termo “diagnóstico” e seu objetivo. Alguns autores consideram diagnóstico apenas como a identificação da ocorrência de falha; a partir de observações dos consequentes erros gerados. Esse trabalho, entretanto, considera o diagnóstico como o processo de identificação do conjunto de falhas (componentes falhos) que eventualmente geraram erros no sistema; a partir da realização de testes no mesmo. Assaf e Dugan (2008) definem

formalmente esse conceito. Considere que um sistema  $S$ , constituído por um conjunto de componentes  $C = (C_1, C_2, \dots, C_n)$ , foi dado como falho. Um conjunto de testes  $T = (T_1, T_2, \dots, T_m)$  pode ser utilizado para determinar quais componentes em  $C$  causaram a falha em  $S$ . O processo de análise e redução de  $C$  a partir de  $T$  é conhecido como diagnóstico.

Nota-se que, na abordagem considerada, o diagnóstico é um processo mais complexo, devido à necessidade da identificação explícita das falhas em *hardware* do sistema. Nesse caso, a detecção da ocorrência de falha (realizada em geral por sensores) é um procedimento necessário para a realização subsequente do diagnóstico.

É interessante enfatizar que nesse trabalho a palavra “componente” é utilizada de forma genérica. Um componente pode representar um conector, um chicote, o pedal de freio, um pino do conector, o motor, etc. A metodologia aqui proposta busca abranger procedimentos com diferentes granularidades no que diz respeito a detecção e diagnóstico dos componentes. Por exemplo, pode-se considerar um *componente* a unidade de controle como um todo. Entretanto, pode-se dividir tal unidade em subunidades (conectores, circuitos integrados, *software*, potência), as quais podem ser tratadas individualmente como *componentes*. Essa diferenciação depende das ferramentas, da granularidade dos testes e do método utilizado no diagnóstico. Porém, assume-se neste trabalho que todo e qualquer *componente* é passível de falha e pode ser diagnosticado corretamente.

### 2.1.1 Diagnóstico *On-board* vs Diagnóstico *Off-board*

Uma categorização comumente encontrada do procedimento de diagnóstico é quanto à ocorrência (ou não) desse procedimento durante a operação do sistema sob análise. Particularmente, tem-se na literatura o diagnóstico *on-board* (*online*) e o diagnóstico *off-board* (*offline*).

No diagnóstico *on-board* algoritmos embarcados no próprio sistema detectam a falha (e eventualmente a corrigem) com o sistema em operação. Em sistemas tolerantes à falhas por exemplo, quando um componente ou subsistema é suspeito de falha, um dispositivo de reserva (redundante àquele passível de falha) é ligado ao sistema de maneira tal que a correção é feita mantendo o sistema operacional durante todo o processo (embora algumas funções podem estar

não disponíveis no decorrer do procedimento). Nesse caso, tanto o processo da detecção da presença da falha quanto a identificação do seu local (componente ou subsistema) são realizados com o sistema em funcionamento.

O diagnóstico *off-board* entretanto, é caracterizado pela identificação da falha em *hardware* por meio de testes realizados com o sistema parado, em manutenção. Em geral, um procedimento prévio de detecção da existência da falha com o sistema em operação é necessário. É esse procedimento que acusa a falha no sistema e obriga o mesmo a passar pela manutenção.

Apesar dos sistemas automotivos possuírem mecanismos de “auto-diagnóstico” realizados por sistemas conhecidos como OBD (*On-Board Diagnostic*), os mesmos não identificam a falha diretamente. O objetivo do OBD é gerar alarmes específicos de forma que a falha possa ser posteriormente encontrada. Assim, o OBD atua apenas na detecção de falha (ou do erro gerado pela falha). Após a detecção de falha, testes são realizados em concessionárias ou oficinas terceirizadas de forma a identificar os componentes em estado de falha. Logo, o diagnóstico em sistemas automotivos abordado neste trabalho é classificado como *off-board*.

## 2.2 DIAGNÓSTICO VEICULAR

O diagnóstico veicular é um processo complexo que envolve diversas etapas. O processo inicia-se pela modelagem das falhas e seus efeitos na fase de desenvolvimento do projeto do automóvel. A partir dessa análise, mecanismos de identificação das falhas são projetados de forma que contramedidas possam ser tomadas no caso da presença de uma eventual falha. Caso uma falha ocorra e seja corretamente identificada, um código de falha (DTC) é ativado e o condutor é alertado da existência de um problema no veículo. Na manutenção, com o veículo parado, técnicos especialistas realizam o procedimento de diagnóstico utilizando ferramentas auxiliares. Cada etapa desse processo será detalhada nas subseções seguintes. Além disso, como esse procedimento é específico para sistemas automotivos, serão apresentadas também algumas características particulares desses sistemas bem como as normas internacionais que regem o diagnóstico na indústria automotiva.

### 2.2.1 Sistemas Eletrônicos Automotivos

Sistemas automotivos são sistemas distribuídos formados por diversos componentes de *software* e *hardware*. Esses componentes são monitorados em tempo real por unidades microcontroladas conhecidas como ECU's. Essas unidades compreendem algoritmos e *software* cada vez mais complexos que provém conforto e segurança ao condutor e aos passageiros com maiores níveis de automação e controle (LANIGAN; KAVULYA; NARASIMHAN, 2011).

Em geral, as diversas ECU's são interconectadas pelo protocolo CAN (*Controller Area Network*), que utiliza barramentos de comunicação serial juntamente com sensores e atuadores para formar o sistema eletrônico distribuído do automóvel. Além disso, a distribuição eletrônica diminui a complexidade de cabeamento pois permite uma melhor disposição dos sensores e atuadores.

Com o aumento das funcionalidades de veículos automotores proposto pelas montadoras e com a intensa necessidade de índices cada vez mais altos de confiabilidade, a parcela eletrônica e computacional (de *software*) vem aumentando sua proporção no custo de produção do veículo. Um veículo comercial de 1989 com apenas três barramentos de comunicação, em 2002 já possuía mais de 70 barramentos. Neste mesmo ano, a quantidade de *software* embarcado ultrapassava as 10 milhões de linhas de código (VECTOR, 2007). A Figura 1 e a Figura 2 ilustram essa evolução. A Figura 1 ilustra o aumento da quantidade de eletrônica embarcada num automóvel (de mesma classe) no período de 1995 à 2002.

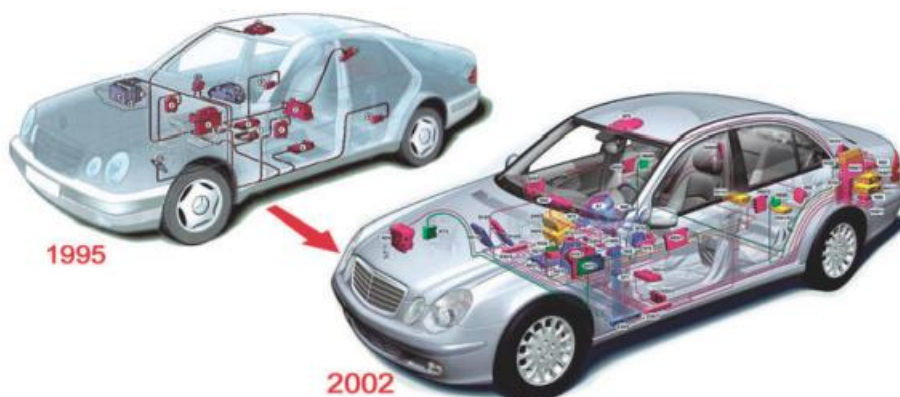


Figura 1 - Comunicação eletrônica de um Mercedes E-Class modelos W210 (1995) e W211 (2002)

Fonte: VECTOR (2007).

A Figura 2 ilustra o aumento da complexidade eletrônica explicitando as diferentes funcionalidades adicionadas no período entre 1980 à 2010.

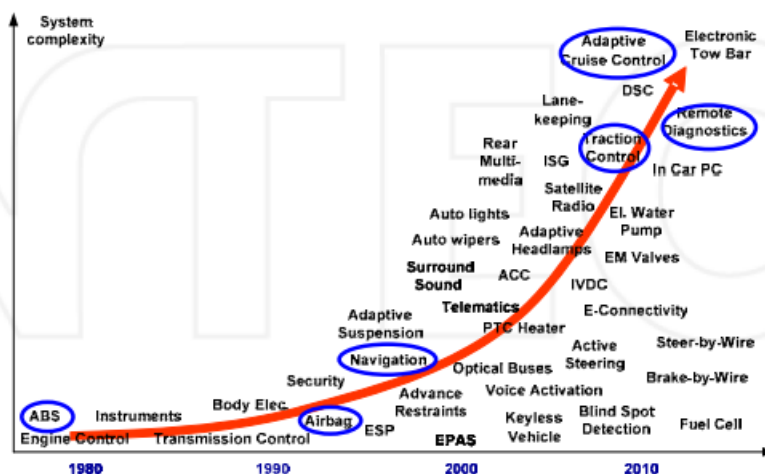


Figura 2 - Evolução dos sistemas eletrônicos automotivos

Fonte: SUWATTHIKUL (2010).

Com o aumento da complexidade eletrônica nos sistemas automotivos fez-se necessário a divisão dos circuitos eletrônicos em diferentes domínios: transmissão (*powertrain*), chassi (*chassis*), carroceria (*body*) e comunicação (*network*). O sistema de transmissão se refere ao motor e toda a parcela de transmissão da potência do motor às rodas do automóvel. A transmissão inclui os eixos de transmissão, o diferencial, a embreagem e a caixa de câmbio. O chassi se refere ao sistema de interação entre o condutor e o veículo, relacionado, portanto, aos subsistemas de aceleração, freio e estabilidade. O sistema de carroceria inclui os sistemas não envolvidos na dinâmica do veículo, incluindo o controle elétrico das portas, janelas e assentos. Finalmente, o sistema de comunicação inclui o sistema de computador de bordo que permite ao condutor acesso às informações do veículo através de uma interface personalizada (SUWATTHIKUL, 2010).

A divisão eletrônica em diferentes domínios é importante no procedimento de diagnóstico, pois cada subsistema é projetado para ser robusto à falhas e erros externos devido a criação de regiões de contenção de falhas entre os subsistemas pelos projetistas. Dessa forma, espera-se que uma falha no subsistema de transmissão não afete o subsistema da carroceria, por exemplo. Isso é possível em virtude do controle eletrônico distribuído dos automóveis, que auxilia o

procedimento de diagnóstico até certo ponto. Inclusive os códigos de falha de domínios distintos são prefixados diferentemente, como mostrado na subsecção 2.2.2.

Além disso, os sistemas automotivos compreendem múltiplas ECU's; atualmente 20 a 60 unidades são embarcadas em veículos médios e *premium*, cuja comunicação se dá por protocolos de comunicação serial, como LIN, CAN e FlexRay (LANIGAN; KAVULYA; NARASIMHAN, 2011). Múltiplos protocolos podem estar presentes numa mesma arquitetura eletrônica; nesse caso, *gateways* realizam a comunicação entre essas diferentes tecnologias. Na Figura 3, tem-se a topologia de comunicação de um sistema automotivo genérico. Nota-se na mesma figura a arquitetura distribuída representada pelas múltiplas ECU's embarcadas.

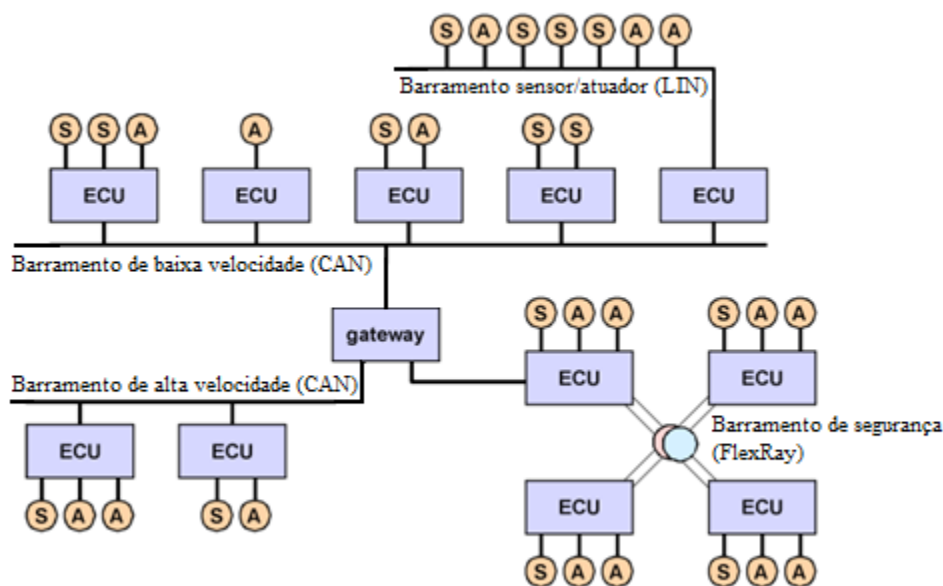


Figura 3 - Topologia de comunicação de um sistema automotivo  
**Fonte:** Adaptado de LANIGAN; KAVULYA; NARASIMHAN (2011).

As ECU's têm o objetivo de controlar o veículo como um todo, recebendo sinais de diferentes sensores e gerando sinais aos atuadores. Por exemplo, em carros com direção elétrica, um sensor óptico envia a velocidade angular de giro e o sentido de giro do volante à ECU, que, com essas informações, envia um sinal elétrico proporcional à um motor elétrico, que vai auxiliar o movimento do condutor. Além disso, com sistemas OBD integrados, as ECU's embarcam sofisticados algoritmos de detecção de falha, além do armazenamento dos código de falha na memória, para posterior rastreamento num procedimento *off-board* (SUWATTHIKUL, 2010).

Nas subseções seguintes apresentam-se os padrões, as normas e os modelos utilizados na indústria automotiva no que diz respeito ao diagnóstico de falhas.

## 2.2.2 Padrões de Diagnóstico na Indústria Automotiva

De forma a padronizar o procedimento de diagnóstico, diversos padrões têm sido criados ao longo dos anos. Tal padronização facilita a documentação, fiscalização e interoperabilidade entre ferramentas e serviços de diagnóstico comumente empregados na indústria automotiva.

O padrão ISO 14229 de 1999 define requisitos comuns de serviços de diagnóstico que permitem ferramentas externas controlar serviços de diagnóstico em uma ECU no veículo através de um *link* de dados. O padrão é baseado no modelo OSI (*Open Systems Interconnection*) ISO/IEC 7498-1 e ISO/IEC 10731 (ambos de 1994) que estruturam os sistemas de comunicação em sete camadas, cujos serviços são divididos em serviços de diagnóstico (camada 7) e serviços de comunicação (camadas 1 a 6) - ver detalhes em Mori (2014).

O padrão ISO 15031-6 de 2005 especifica o formato, a estrutura e as mensagens associadas ao diagnóstico veicular. Uma importante definição diz respeito à padronização dos códigos de falha. Nesse padrão, cada DTC possui um código alfanumérico num formato específico, formado por um prefixo que representa o subsistema eletrônico do veículo (ver subseção 2.2.1) em que a falha se encontra, seguido por um código de três dígitos que define o objeto diagnosticado. Na Tabela 1 tem-se os prefixos especificados pela ISO 15031-6 de acordo com os diferentes subsistemas eletrônicos.

Tabela 1 - Prefixos dos DTCs especificados pela ISO 15031-6

<b>Subsistema</b>	<b>Prefixo</b>
Carroceria ( <i>body</i> )	<b>B0XXX – B3XXX</b>
Chassi ( <i>chassis</i> )	<b>C0XXX – C3XXX</b>
Transmissão ( <i>powertrain</i> )	<b>P0XXX – P3XXX</b>
Comunicação ( <i>network</i> )	<b>U0XXX – U3XXX</b>

Fonte: Adaptado de ISO 15031-6.



O mesmo padrão define o *Byte* de Tipo de Falha (*FTB - Failure Type Byte*) que é composto pela categoria de falha do DTC. A Tabela 2 relaciona os códigos às diferentes categorias de falha adotadas pela norma ISO 15031-6.

Tabela 2 - Definição de Categoria de Falha dos DTCs

Nibble Alto (0000b – 1111b)	Nr. da Categoria (hexa)	Descrição da Categoria
0000	0	<b>Informação de falha geral</b> – Esta categoria inclui todas as categorias e é utilizada quando a falha nesta categoria é única (não padronizável através de subtipos) ou quando a falha detectada é melhor descrita por dois ou mais subtipos nesta categoria de falha.
0001	1	<b>Falha elétrica geral</b> – Esta categoria inclui modos de falha padrão de chicote (por exemplo, curtos-circuitos e circuitos aberto), e corrente contínua (CC) de acordo com a lei de Ohm.
0010	2	<b>Falha de sinal geral</b> – Esta categoria inclui valores relacionados a amplitude, frequência ou taxa de mudança e forma do sinal.
0011	3	<b>Falha FM (Modulação por frequência)/PWM (Modulação por Largura de Pulso)</b> – Esta categoria inclui falhas relacionadas às entradas e saídas FM ou PWM dos módulos de controle. Esta categoria também inclui falhas em que a posição é determinada por contadores.
0100	4	<b>Falha interna de sistema</b> – Esta categoria inclui falhas relacionadas à memória, <i>software</i> e de circuito interno, que requer a substituição do componente (módulo de controle, sensor, etc.).
0101	5	<b>Falha de programação de sistema</b> – Esta categoria inclui falhas relacionadas ao <i>software</i> operacional, calibrações e configurações que são resolvidas através da configuração/programação de uma parte do sistema (módulo de controle, sensor, etc.).
0110	6	<b>Falhas baseada em algoritmos</b> – Esta categoria inclui falhas baseadas na comparação de dois ou mais entradas de comparação para plausibilidade ou comparando um único parâmetro com relação ao tempo.
0111	7	<b>Falhas mecânicas</b> – Esta categoria inclui falhas detectadas pelo movimento inapropriado em resposta à entrada ou saída controlada do módulo de controle.
1000	8	<b>Falhas de sinal/mensagens de barramento</b> – Esta categoria inclui falhas relacionadas ao hardware em um barramento e à integridade do sinal do barramento. Esta categoria é também utilizada quando o sinal de falha se localiza em um módulo de controle no barramento, mas é outro módulo que diagnostica o circuito e inibe a operação do primeiro.
1001	9	<b>Falha de componente</b> – Esta categoria inclui falhas relacionadas a erros nos componentes (incluindo falhas de parametrizações, de montagem e de condições ambientais).
1010 - 1110	A – E	<b>Reservadas a ISO/SAE</b> – Estes valores estão reservados ao ISO 15031 para expansões futuras.

1111	F	<b>Fabricante do veículo/Fornecedor de sistema</b> – Esta categoria é reservada para a utilização do fabricante do veículo/fornecedor de sistema.
------	---	---

Fonte: MORI, 2014.

Para cada categoria tem-se diferentes subtipos que compõe o FTB e o respectivo DTC. A Tabela 3 relaciona os subtipos existentes da Categoria 1 presentes na Tabela 2.

Tabela 3 - Descrição dos FTBs

<b>FTB</b>	<b>Informação Geral da Falha</b>
10	Reservado para ISO/SAE
11	Curto-circuito para referência
12	Curto-circuito para bateria
13	Circuito aberto
14	Curto-circuito para referência ou aberto
15	Curto-circuito para bateria ou aberto
16	Tensão abaixo do limite
17	Tensão acima do limite
18	Corrente abaixo do limite
19	Tensão acima do limite
1A	Resistência abaixo do limite
1B	Tensão acima do limite
1C	Tensão fora do limite operacional
1D	Corrente fora do limite operacional
1E	Resistência fora do limite operacional
1F	Circuito Intermitente

Fonte: MORI, 2014.

Um exemplo de DTC na norma ISO 15031-6 é o código C100413 que se traduz em uma falha de circuito aberto (FTB 13) no sensor de velocidade da roda dianteira esquerda do ABS que se encontra no chassi (C1004).

Como cada montadora possui o seu conjunto específico de fornecedores, componentes, método de montagem e estratégia de diagnóstico, além dos DTCs padronizados pela ISO/SAE existem códigos proprietários para determinadas falhas. Neste caso, não existe um DTC padronizado para uma falha específica. Portanto, fica a critério da montadora padronizá-lo internamente até que um DTC padrão seja normatizado pelo órgão regulador.

Além dos DTCs, o mesmo padrão ISO 15031-6 também define os parâmetros e as rotinas embarcadas nas unidades de controle do veículo. Os parâmetros de *software* da ECU fazem a

interface com as ferramentas de diagnóstico externas conectadas através de conectores OBD II, já as rotinas são os serviços prestados pelas ECU's à essas ferramentas externas.

Além dos padrões descritos anteriormente, outras normas regem o procedimento de diagnóstico em outros contextos. Por exemplo, o padrão SAE J1939-73 define o formato do DTC para veículos pesados e ônibus da SAE (MORI, 2014).

### 2.2.3 Sistema OBD (*On-Board Diagnostic System*)

Outra padronização bastante significativa é a integração dos veículos ao sistema OBD que acrescenta aos automóveis um sistema *on-board* de “autodiagnóstico” que tem como objetivo auxiliar os veículos no monitoramento e na detecção de falhas. Esse sistema foi criado nos Estados Unidos devido a forte pressão pública para o controle de emissões de poluentes à atmosfera, e tinha o objetivo de garantir que o controle de emissões veiculares respeitasse determinadas exigências internacionais. Nesse sentido, foi estabelecido que toda e qualquer falha que impedisse a capacidade do veículo em atender as exigências fosse passível de detecção em tempo real e codificada com DTCs especificados de acordo com a legislação OBD (RIZZONI; ONORI; RUBAGOTTI, 2009).

Em 1994, o OBD se tornou mais sofisticado dando origem ao OBD II, embarcado em quase todos os veículos em circulação atualmente. Estabelecido pela SAE (*Society of Automotive Engineer*), a regulação OBD II especificou que todos os veículos dos Estados Unidos à gasolina, Diesel e à combustíveis alternativos atendessem às suas especificações. A União Europeia desenvolveu um conjunto de regulações conhecido como EOBD (*European On-Board Diagnostics*), similar às definidas pelo Estados Unidos. Porém, com maiores limites de tolerância. Em prática, a maioria das montadoras europeias utiliza a especificação OBD II Norte-Americana para atender os requisitos europeus (RIZZONI; ONORI; RUBAGOTTI, 2009).

É importante enfatizar que apesar do sistema OBD se referir à um procedimento de “diagnóstico” *on-board*, o mesmo auxilia na identificação e categorização (através da associação da falha à um DTC específico) da falha apenas. Um procedimento de diagnóstico *off-board* posterior à esse processo é necessário para a localização efetiva da falha. Assim, os sistemas OBD

são formados por algoritmos integrados às ECU's com o objetivo de identificar falhas e traduzi-las em DTCs (SUWATTHIKUL, 2010).

Um exemplo da efetividade dos sistemas com integração OBD II é no controle de emissões de gases por evaporação. Esse sistema (conhecido como EVAP – *Evaporative Emission Control System*) é utilizado para reduzir, armazenar e conduzir os gases provenientes da combustão à atmosfera. O sistema, mostrado na Figura 4, tem como principais componentes o tanque de combustível (*fuel tank*), o sensor de pressão e nível de combustível (*fuel pump and fuel level sensor*), a caixa de vapores (*vapor canister*), a válvula de purga (*purge solenoid*) e a válvula de entrada de ar (*vent solenoid*).

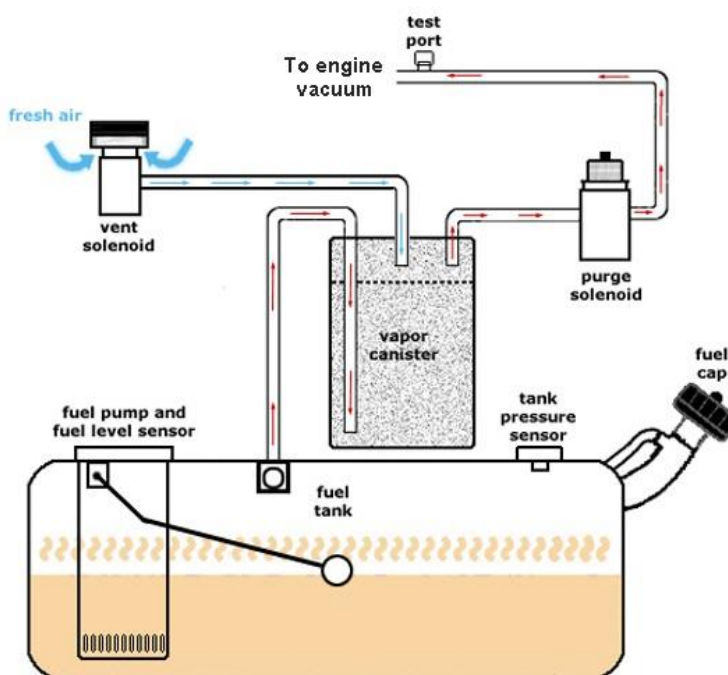


Figura 4 - Sistema automotivo para o controle de emissão de gases  
Fonte: AGCOAUTO (2016).

Apesar desse sistema variar de acordo com o veículo e o fabricante, seu propósito e princípios de funcionamento são praticamente os mesmos. Na operação normal, os gases provenientes da evaporação do combustível são armazenados na caixa de vapores que, em geral, está localizada próxima ao tanque de combustível ou ao compartimento do motor. Quando o motor está em funcionamento esses gases são eliminados pelo processo de combustão. Esse processo inicia-se pela unidade de controle responsável, que verifica o estado da placa do acelerador (*throttle plate*) e abre ambas as válvulas de purga e de entrada de ar. A válvula de entrada de ar

permite a mistura dos gases do combustível com o oxigênio proveniente do ar, enquanto a válvula de purga envia a mistura ao motor, onde ocorre o processo de combustão.

Entretanto, caso as válvulas não operem corretamente ou se houver entradas ou saídas de ar indesejadas, o processo torna-se ineficiente ou até mesmo ineficaz. Nesse contexto, desenvolveu-se um sistema OBD de autodiagnóstico para identificação de falhas no sistema de emissões. Todo processo é realizado por algoritmos embarcados na ECU responsável pelo sistema. Primeiramente, fecha-se a válvula de entrada de ar, de forma a isolar o sistema de qualquer fluxo externo de ar. Em seguida, é estabelecido um vácuo no sistema até que o sensor de pressão observe um valor predefinido. Observado esse valor, a ECU fecha a válvula de purga, mantendo uma pressão constante no interior do sistema de emissões. Durante um certo período de tempo, a ECU verifica a existência de quedas de pressão no sistema; caso não haja, a ECU finaliza o procedimento de detecção e o sistema volta à operação regular. Entretanto, se a ECU detectar uma queda de pressão, a unidade identifica um vazamento, aciona um DTC que varia de acordo com a magnitude do mesmo e ativa a lâmpada MIL (*Malfunction Indication Lamp*) no painel do veículo, indicando ao condutor a ocorrência da falha.

No caso da presença da falha, a ECU pode ativar um (ou mais) códigos de até 14 DTCs disponíveis, baseando-se nas condições observadas. Os códigos relacionados e uma breve descrição dos mesmos estão listados abaixo:

- P0400: o sistema não está apto a atingir o vácuo necessário;
- P0441: não existe fluxo durante o ciclo de purga;
- P0442, P0456, P0457: pequeno vazamento no sistema;
- P0455: grande vazamento no sistema;
- P0443, P0446, P0449: válvula de purga danificada;
- P0450, P0451, P0452, P0453: sensor de pressão danificado;
- P0496: fluxo de emissões incorreto.

Apesar da existência de diversos DTCs em um mesmo subsistema, ainda assim, não é possível realizar a inferência da falha apenas baseando-se no DTC ativo devido ao grande número de componentes. A existência dos diferentes DTCs apenas objetiva direcionar o posterior procedimento de testes às causas mais prováveis de falha. Considere por exemplo, o código P0443, que a princípio especifica uma falha na válvula de purga. Nesse caso, pode-se ter as seguintes falhas (OBD-CODES, 2016):

- Falha mecânica na válvula de purga;
- Falha de *software* na ECU responsável;
- Falha nos conectores;
- Falha nos chicotes elétricos;

Nota-se, portanto, a necessidade de um procedimento de testes *off-board* para identificação correta do mesmo; e para posterior substituição ou reparo do componente falho.

A universalização do padrão OBD trouxe consigo a abertura de protocolos das ferramentas de auxílio ao diagnóstico, como os *scanners* automotivos. A subseção 2.2.4 seguinte aborda essa ferramenta e suas características.

#### 2.2.4 *Scanners* Automotivos

O aumento do desenvolvimento de métodos e ferramentas de diagnóstico é similar ao crescimento do domínio eletrônico nos veículos comerciais. Em meados de 1990 as montadoras desenvolviam as próprias ferramentas de auxílio. Especialistas de diferentes departamentos personalizavam ferramentas proprietárias de acordo com suas exigências e aplicações específicas, resultando em um grande conjunto de soluções *in-house* existentes para diferentes etapas de produção e análise (VECTOR, 2007).

Aproximadamente em 1995, o desenvolvimento de ferramentas de auxílio ao diagnóstico foi terceirizado, o que resultou numa diminuição das diferenças entre as ferramentas utilizadas pelas montadoras. A abertura dos protocolos foi se expandindo até que nos anos 2000 ocorreu a padronização e a abertura completa dos códigos e protocolos de comunicação e diagnóstico para o público geral, de forma a unificar o procedimento de diagnóstico entre diferentes montadoras e possibilitar ao usuário comum o acesso à informações de diagnóstico de seu veículo, possibilitando o mesmo realizar procedimentos de reparo sem a necessidade de ir a uma concessionária ou oficina para manutenção. Dessa forma, dispositivos de leitura e verificação de DTCs universais com compatibilidade OBD puderam ser comercializados.

A principal ferramenta de diagnóstico comercializada atualmente é o *scanner* automotivo que permite que o usuário acesse diversas informações a respeito do veículo, incluindo os códigos

de falha ativos (tanto os genéricos quanto os específicos de cada marca). Com conexão *bluetooth* e *wi-fi* integrados, os *scanners* universais permitem a verificação do estado do veículo no próprio celular do usuário. Em resumo, o *scanner* apenas viabiliza a interface com o usuário, fornecendo informações tais como:

- Códigos de falha ativos;
- Gasto médio de combustível;
- *Status* do sistema de combustível
- Taxa de fluxo de ar;
- Posição absoluta do regulador de pressão;
- Temperatura do líquido refrigerante;
- Leitura das emissões de CO<sub>2</sub>.

Na Figura 5 tem-se a imagem de um *scanner* automotivo universal com compatibilidade OBD II que permite o acesso às informações descritas acima.



Figura 5 - *Scanner* automotivo OBD II CP9575 da Actron  
Fonte: AMAZON (2016).

Apesar da existência dos *scanners* universais e da abertura dos protocolos, as montadoras continuam utilizando ferramentas proprietárias. Nesse caso, os dispositivos de leitura são privados e não comercializáveis, e os programas de leitura e correção de erros são alterados anualmente de modo a evitar um mercado paralelo de reparação.

A leitura dos códigos de falha pelo *scanner* automotivo é realizada através do conector OBD II. Esse conector permite o acesso à unidade principal de comando do automóvel através de

comunicação serial ou pelo padrão CAN para a obtenção das informações do veículo. Na Figura 6, tem-se o conector OBD II do Nissan Sentra, modelo 2005.



Figura 6 - Conector OBD II do veículo Nissan Sentra, modelo 2005  
Fonte: NISSANHELP (2016).

Além da possibilidade de acesso aos códigos ativos, o conector OBD II é utilizado para atualizar o *software* embarcado e calibrar parâmetros (LANIGAN; KAVULYA; NARASIMHAN, 2011). A localização do conector de diagnóstico e o *layout* dos pinos de comunicação devem atender à norma ISO 15031-3. O conector OBD II possui 16 pinos de encaixe elétrico, cujo posicionamento padrão é mostrado na Figura 7.

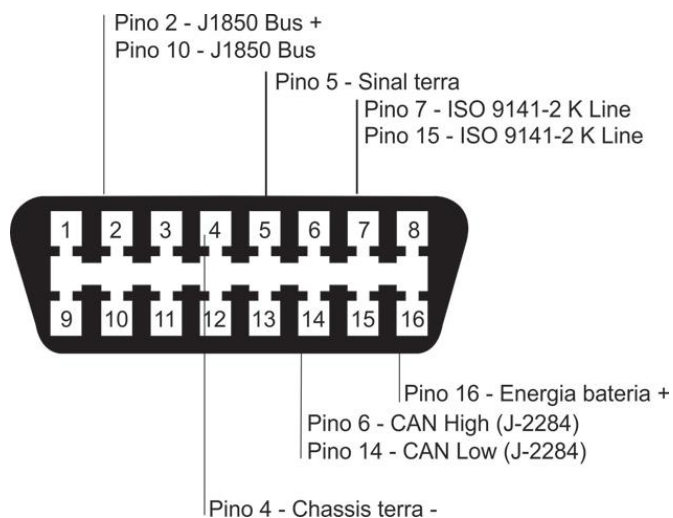


Figura 7 - *Layout* padrão do conector de diagnóstico OBD II de 16 pinos  
Fonte: NOTÍCIAS DA OFICINA VOLKSWAGEN (2016).



Além das padronizações definidas por regulamentos internacionais, os fabricantes de veículos geralmente utilizam análises, modelos e métodos padronizados no que diz respeito ao estudo e implementação do diagnóstico veicular, de forma a sistematizar todo o processo. Na subseção 2.2.5 apresenta-se uma ferramenta de análise de falhas realizada na fase de projeto. Em seguida, inicia-se a apresentação das estruturas de auxílio direto ao procedimento diagnóstico: a árvore de falhas e a árvore de diagnóstico.

### 2.2.5 Análise FMEA (*Failure Mode and Effect Analysis*)

A análise sistemática de falhas em que são avaliadas tanto a severidade quanto a probabilidade de falhas no veículo é realizada na fase de projeto do veículo, com o veículo *off-line* (LANIGAN; KAVULYA; NARASIMHAN, 2011). Nessa etapa, são testados componentes e analisadas as condições necessárias para ocorrência de certas falhas. A partir dessas análises, mecanismos de detecção de falhas são desenvolvidos de forma que o condutor seja alertado caso alguma falha esteja presente no veículo.

Um método de estudo e desenvolvimento de contramedidas à ocorrência de falhas é a análise de modo de falha e seus efeitos (FMEA – *Failure Mode and Effect Analysis*). Nessa metodologia, estuda-se o cenário de ocorrência de potenciais falhas e analisa-se os seus efeitos. Medidas de identificação e de adaptação à essas falhas são projetadas de forma a minimizar seus impactos no sistema; por exemplo, através da instalação de novos sensores.

O FMEA é uma técnica que objetiva detalhar todos os modos possíveis de falha e seus efeitos resultantes no sistema. É uma metodologia de confiabilidade a nível de arquitetura do sistema, pois o analista foca nos modos de falha de cada componente individual, bem como nos seus efeitos e severidade (YACOUB; AMMAR, 2002). Mais especificamente, para cada falha possível (modo de falha), analisa-se as possíveis causas, os efeitos (ou sintomas), as medidas de detecção e prevenção, os índices de risco e severidade e as contramedidas necessárias (MORI, 2014).

## 2.2.6 Análise por Árvore de Falhas

Um outro método de estudo das falhas e seus impactos, presente na fase de projeto do veículo, é a análise por Árvores de Falhas (FTA – *Fault Tree Analysis*). A FTA inicia-se com um estado indesejado (crítico ao sistema em termos de confiabilidade ou segurança) conhecido como evento raiz e consiste numa análise dedutiva de procura de todos os eventos que podem gerar tal estado (VESELY, 2002). No contexto automotivo, o evento raiz se traduz em um DTC e os eventos básicos compreendem as falhas capazes de acionar o respectivo código de falha.

O produto da FTA é uma Árvore de Falhas (FT – *Fault Tree*) que é uma explicação lógica do evento raiz em termos dos eventos básicos. Apesar da árvore de falhas consistir em um modelo qualitativo, é possível realizar análises quantitativas utilizando a sua estrutura, como por exemplo obter valores de importância dos eventos básicos e obter a probabilidade do evento raiz. A Figura 8, ilustra uma árvore de falhas cujo evento raiz é o mau funcionamento de um farol automotivo.

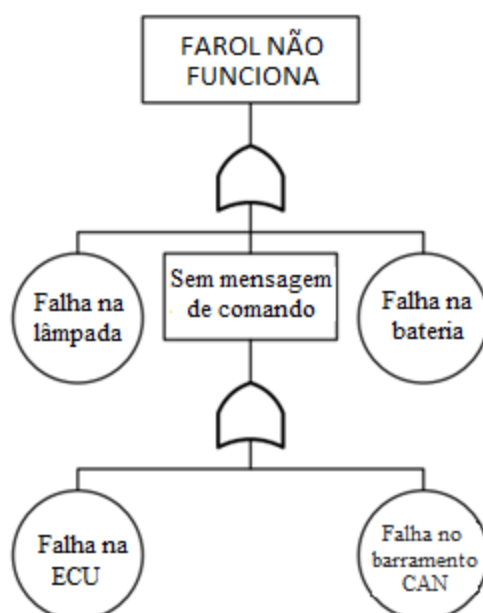


Figura 8 - Árvore de falhas de um farol automotivo  
Fonte: Adaptado de SUWATTHIKUL (2010).

Na Figura 8 quatro eventos básicos individuais podem explicar o mau funcionamento do farol: uma falha no próprio farol, uma falha na alimentação, uma falha na unidade de controle e

uma falha no barramento de comunicação. Os dois últimos estão relacionados com o evento raiz através do evento intermediário “Sem mensagem de comando”.

É importante enfatizar que nem todas as causas são abordadas pela árvore de falhas, ou seja, o modelo não é exaustivo no sentido de cobrir todas as possibilidades de eventos básicos. Apenas as falhas consideradas “realistas” para o analista são consideradas (VESELY, 2002).

A relação lógica entre os eventos básicos e o evento raiz proveniente da árvore de falhas permite uma melhor tomada de decisão no procedimento do diagnóstico. Resumidamente, pode-se dizer que a FTA auxilia o projetista nos seguintes aspectos (VESELY, 2002):

- Proporciona um melhor entendimento da relação lógica de falhas que explica o evento raiz;
- Auxilia no esclarecimento dos maiores contribuidores ao surgimento do evento raiz;
- Desempenha o papel de uma ferramenta preventiva ao aparecimento do evento raiz;
- Auxilia na monitoração do desempenho do sistema;
- Permite a otimização de recursos;
- Auxilia no projeto do sistema;
- Desempenha o papel de uma ferramenta de diagnóstico para identificação e correção das causas do evento raiz.

Percebe-se, portanto, a importância da FTA na indústria e no diagnóstico automotivo. Tal análise é realizada durante todo o ciclo de vida do automóvel, do desenvolvimento até a distribuição, período o qual utilizam-se informações obtidas de assistências técnicas distribuídas globalmente para futuros aperfeiçoamentos nos modelos adotados.

### 2.2.7 Árvores de Diagnóstico

A árvore de diagnóstico é uma ferramenta utilizada pelos técnicos de manutenção para conduzir o procedimento *off-board* de testes. Além de explanar a rede sequencial de testes para cada DTC, a ferramenta define os valores esperados, os instrumentos a serem utilizados e as instruções específicas para a realização de cada teste. Na Figura 9, tem-se parte da árvore de diagnóstico do DTC P1212 de um veículo F350 da marca Ford. O código especifica um problema

de leitura de valores do sensor de injeção de pressão (ICP – *Injection Control Pressure*) fora da especificação.

	Test Steps	Results	Action to Take
<b>DC15</b>	<b>CONTINUOUS DIAGNOSTIC TROUBLE CODE (DTC) P1212</b>		
	<p>Note: This code may set after a high-pressure oil system repair (i.e., injector change). Drive vehicle to clear air from system after repair, and clear code.</p> <ul style="list-style-type: none"> <li>• Continuous DTC P1212 indicates that there was no ICP detected during crank (long crank time).</li> </ul> <p>Possible causes:</p> <ul style="list-style-type: none"> <li>— high-pressure oil system repair</li> <li>— empty/low oil reservoir at crank</li> <li>— low engine oil level or incorrect viscosity</li> <li>— IPR circuit fault</li> <li>— damaged low-pressure oil pump</li> <li>— damaged IPR valve</li> <li>— damaged ICP sensor</li> <li>— damaged high-pressure oil pump</li> </ul> <ul style="list-style-type: none"> <li>• Check engine oil level.</li> <li>• Perform KOEO On-Demand Self Test to verify ICP/IPR circuit faults are not present at this time.</li> <li>• <b>Are KOEO DTCs present?</b></li> </ul>	<p>Yes →</p> <p>No →</p>	<p>REPAIR KOEO DTCs before continuing. If DTC P1212 returns, GO to <b>DC16</b>.</p> <p>GO to <b>DC16</b>.</p>
<b>DC16</b>	<b>CHECK OIL RESERVOIR LEVEL</b>		
	<p>Note: If vehicle stalls after start and reservoir empties, oil is not being supplied from low pressure oil system.</p> <ul style="list-style-type: none"> <li>• Verify correct oil level, quality and viscosity.</li> <li>• Check oil level in oil reservoir.</li> <li>• Verify Steps 5 and 9 on the Hard Start/No Start Diagnostic Guide have been performed</li> <li>• Clear DTCs.</li> <li>• Monitor ICP_V PID while cranking.</li> <li>• <b>Is ICP_V reading at least 0.85 V within 5 seconds at crank?</b></li> </ul>	<p>Yes →</p> <p>No →</p>	<p>DTC is intermittent. GO to <b>DC20</b>. If other DTCs are set, SERVICE them first.</p> <p>GO to <b>DC17</b>.</p>
<b>DC17</b>	<b>CHECK ICP SENSOR</b>		
	<ul style="list-style-type: none"> <li>• Disconnect ICP sensor harness connector.</li> <li>• <b>Does vehicle start?</b></li> </ul>	<p>Yes →</p> <p>No →</p>	<p>INSTALL a new ICP sensor. RESTORE vehicle. CLEAR DTCs and RETEST.</p> <p>GO to <b>DC18</b>.</p>
<b>DC18</b>	<b>HIGH-PRESSURE LEAKAGE TEST</b>		
	<ul style="list-style-type: none"> <li>• Connect ICP sensor harness connector.</li> <li>• Clear ICP DTCs.</li> <li>• Perform the Injection Control Pressure Tests per the procedures in Diagnostic Subroutines Hard Start/No Start.</li> <li>• <b>Is a leak fault indicated?</b></li> </ul>	<p>Yes →</p> <p>No →</p>	<p>REPAIR condition causing the leak as necessary. RESTORE vehicle. CLEAR DTCs and RETEST.</p> <p>GO to <b>DC19</b>.</p>
<b>DC19</b>	<b>IPR VALVE CHECK</b>		
	<ul style="list-style-type: none"> <li>• Connect all hoses and ICP sensor harness connector.</li> <li>• Install a new IPR valve.</li> <li>• Refill oil reservoir.</li> <li>• <b>Does vehicle start?</b></li> </ul>	<p>Yes →</p> <p>No →</p>	<p>IPR valve was faulty. RESTORE vehicle. CLEAR DTCs and RETEST.</p> <p>INSTALL a new high pressure pump. RESTORE vehicle. CLEAR DTCs and RETEST.</p>
<b>DC20</b>	<b>CHECK FOR INTERMITTENT CIRCUIT FAULTS</b>		
	<ul style="list-style-type: none"> <li>• Key on, engine off.</li> <li>• Access ICP PID.</li> <li>• Grasp the vehicle harness close to the sensor connector. Wiggle, shake vehicle harness while working towards the PCM.</li> <li>• <b>Does ICP reading fluctuate?</b></li> </ul>	<p>Yes →</p> <p>No →</p>	<p>REPAIR circuit as required. CLEAR DTCs and RETEST.</p> <p>GO to <b>DC21</b>.</p>
<b>DC21</b>	<b>INSPECT IPR CIRCUIT</b>		
	<ul style="list-style-type: none"> <li>• Inspect IPR circuit from IPR valve through 42-way connector to the PCM.</li> <li>• <b>Are any faults indicated?</b></li> </ul>	<p>Yes →</p> <p>No →</p>	<p>REPAIR circuit as required. RESTORE vehicle. CLEAR DTCs and RETEST.</p> <p>Unable to verify concern at this time. RESTORE vehicle. CLEAR DTCs and RETEST.</p>

Figura 9 - Parte da árvore de diagnóstico do DTC P1212 da F350

Fonte: JUSTANSWER (2011).



No BDD, cada nó representa um componente ou um grupo de componentes a serem testados e, em geral, é criado a partir das árvores de falha (ASSAF; DUGAN, 2005). Dessa forma, a raiz do BDD representa um determinado DTC, assim como na árvore de falhas. Além disso, a cada teste (nó) do BDD associa-se um estado operacional (*pass*) ou falho (*fail*) (VESELY, 2002).

Nesse trabalho, os eventos básicos da árvore de falhas serão tratados como “eventos de falha” enquanto que os nós do BDD serão conhecidas como “testes”, pois é mais usual encontrar essas associações na literatura.

Diferentes métodos de conversão da árvore de falhas para o diagrama de decisão binário existem na literatura. Por exemplo, Assaf e Dugan (2005), Andrews e Remenyte (2005) e Rauzy (1993) apresentam alguns desses métodos. Como a árvore de falhas especifica uma relação lógica, a mesma pode ser expressa por uma expressão booleana. Assim, para a codificação da árvore de falhas e posterior conversão desta para o BDD, utiliza-se a sua expressão booleana correspondente.

Uma importante etapa no procedimento de conversão é a definição da ordem em que os eventos básicos de falha da árvore de falhas serão dispostos no BDD pois, dependendo da ordenação, o tamanho do BDD pode aumentar consideravelmente (ANDREWS; REMENYTE, 2005). Dessa forma, tem-se diversos BDDs para uma mesma árvore de falhas, dependendo da ordenação utilizada para os testes. Na Figura 11 tem-se uma árvore de falhas, cuja expressão booleana é  $DTC = (c \text{ OR } a \text{ OR } d) \text{ AND } (a \text{ OR } b)$ , com o respectivo BDD para a ordem  $c, a, d, b$ . Note que os componentes com maior prioridade estão dispostos acima no BDD. No caso de uma eventual ativação do DTC raiz, estes componentes são testados primeiramente.

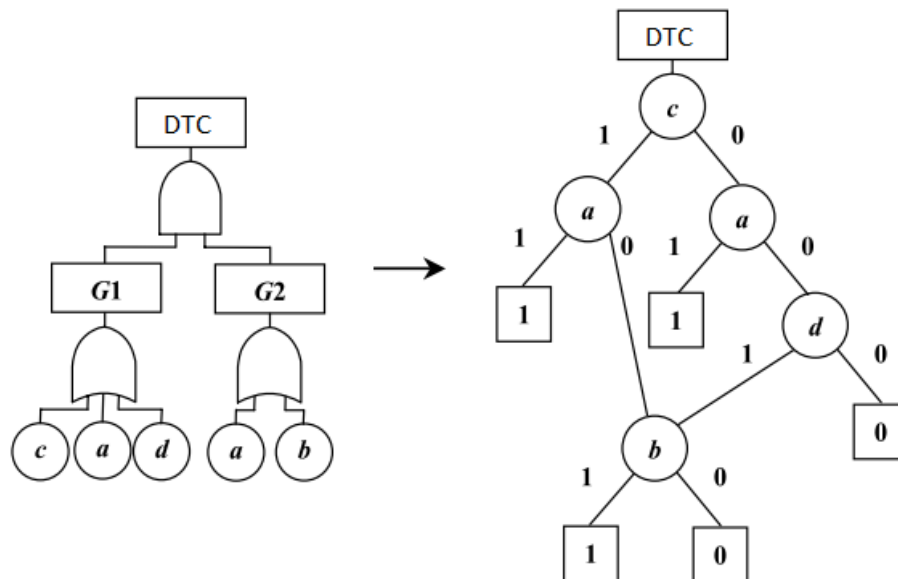


Figura 11 - Árvore de falhas e BDD associado

Fonte: Adaptado de ANDREWS e REMENYTE (2005).

No diagrama de decisão binário da Figura 11 tem-se ramos com os números “1” e “0”. O ramo binário “1” é convencionado como FAIL; ou seja, o teste associado ao nó acima do ramo acusou uma falha no componente em questão. De forma complementar, o ramo binário “0” determina que o componente associado ao nó diretamente acima está operacional e não apresenta problemas.

Além disso, as folhas do BDD também podem assumir valores “0” ou “1”. Alguns autores classificam esses valores como a seguir: 1 – “Falha Encontrada” (FI – *Fault Identified*); 0 – “Falha Não Encontrada” (NFF – *No Fault Found*). Entretanto, tal associação tem um significado simplista e pode resultar numa interpretação incorreta. Considere por exemplo a sequência de testes  $c = 1$ ,  $a = 0$  e  $b = 0$  no BDD da Figura 11. Nessa sequência o “resultado” do diagnóstico foi dado como “0” (especificado na folha do final da sequência de testes). Note que uma conclusão de “Falha Não Encontrada” não é muito correta pois o componente  $c$  foi acusado de falha e deve ser reparado ou substituído. Porém, note também que apenas o componente  $c$  não explica a ativação do DTC (observe a árvore de falhas). Ou seja, quando o resultado do diagnóstico via BDD for “0” não se tem uma explicação lógica para a ativação do DTC utilizando o modelo de falhas disponível. Observe que todas as sequências de testes terminadas em folhas “1” possuem uma relação lógica verdadeira na árvore de falhas quando o DTC está ativo.

Resumindo, um dado DTC possui duas estruturas relacionadas. A primeira é a árvore de falhas (FT) que explica logicamente a existência do DTC em função de seus eventos de falha. A segunda é o diagrama de decisão binário (BDD) (ou árvore de diagnóstico) que associa à uma eventual ocorrência do DTC sequências de testes sobre os componentes passíveis de falha.

A formalização da árvore de falhas e do BDD e a utilização dos mesmos pela ferramenta de auxílio ao diagnóstico proposta nesse trabalho são abordadas no Capítulo 3.

## 2.2.8 Procedimento de Diagnóstico na Indústria Automotiva

Definidos os padrões, estruturas e modelos utilizados na indústria, pode-se detalhar o procedimento de diagnóstico e suas particularidades.

Como abordado nas subseções 2.2.6 e 2.2.7, ainda no estágio de projeto do veículo, um procedimento de análise de falhas através das técnicas padrão FMEA e FTA permite o desenvolvimento de algoritmos de detecção de falha para aquele veículo em particular bem como da árvore de falhas e do BDD associados a cada DTC. Esse cenário é ilustrado na Figura 12.

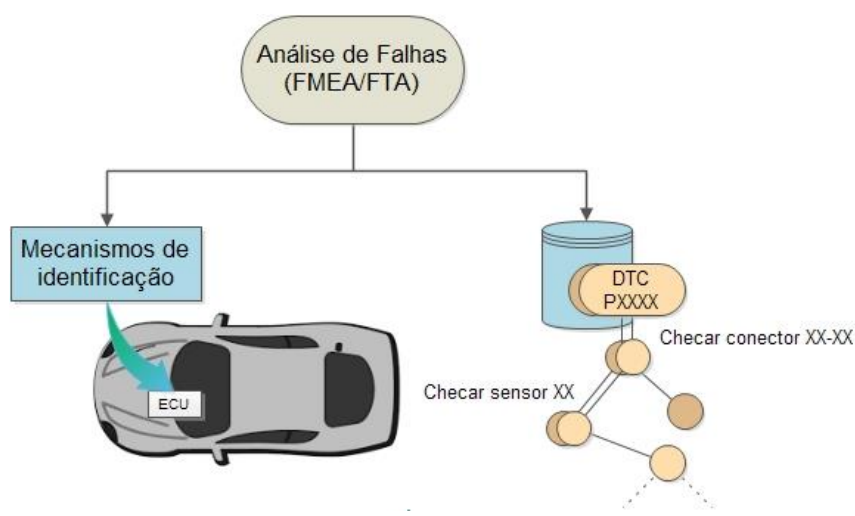


Figura 12 - Análise de falhas e seus objetivos no diagnóstico veicular  
**Fonte: Autoria própria.**

Antes mesmo do veículo ser liberado para distribuição, as informações a respeito dos códigos de falha, sequências de testes e dos procedimentos de diagnóstico particulares àquele



automóvel são distribuídas para as concessionárias de forma que seja possível a realização do procedimento diagnóstico.

Pode-se dividir o processo de investigação da falha em duas etapas: a etapa de identificação da ocorrência da falha *on-board*; e a etapa do diagnóstico, constituído por ensaios e testes *off-board*. No procedimento *on-board* sensores distribuídos e algoritmos embarcados monitoram o conjunto de estados do veículo. Caso alguma anomalia seja detectada, é acionado um código de falha associado. Pode-se dizer que as falhas, em geral, são eventos “não observáveis”, pois não é possível a identificação direta das mesmas. Entretanto, o DTC gerado pelas unidades de controle, quando a falha é detectada, é diretamente observável.

Nesse sentido, o diagnóstico pode ser entendido como o processo pelo qual se obtém a falha ocorrida a partir do sinal de falha gerado (DTC). A Figura 13 ilustra esse conceito.

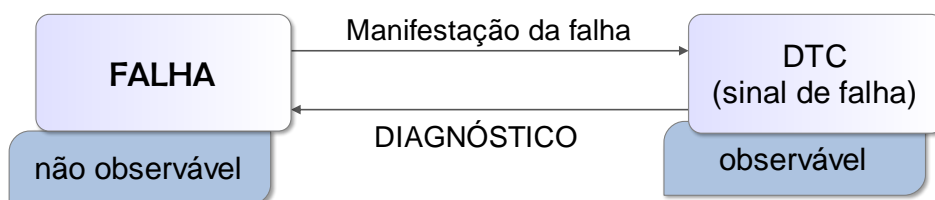


Figura 13 - Relação entre a falha ocorrida e o DTC ativado  
**Fonte: Autoria própria.**

Como mencionado na subseção 2.2.6 cada DTC mapeia um conjunto de falhas particular, cuja relação lógica é representada pela árvore de falhas. Além disso, seu formato é padronizado pela ISO 15031-6 no caso dos veículos comerciais leves.

Um modelo de falhas ideal mapeia cada falha de um componente individual em um DTC específico. Nesse caso a inferência da falha seria direta pelo sistema eletrônico do automóvel e o procedimento *off-board* não seria necessário. Porém, tal cenário é atualmente infactível tendo em vista o alto custo resultante da instalação dos inúmeros sensores, além do processamento adicional necessário.

Além disso, apesar da tentativa dos projetistas de isolar determinadas falhas e componentes por subsistemas associando códigos de falha individuais à estes, devido à complexidade eletrônica automotiva atual, o isolamento completo do sistema eletrônico é infactível. Por consequência, tem-se testes e componentes em comum associados à diferentes DTCs pelas respectivas árvores de falha e BDDs. Esse cenário pode causar a ativação de múltiplos DTCs caso uma única falha ocorra.

Pode-se notar na Figura 14 o cenário apresentado acima.

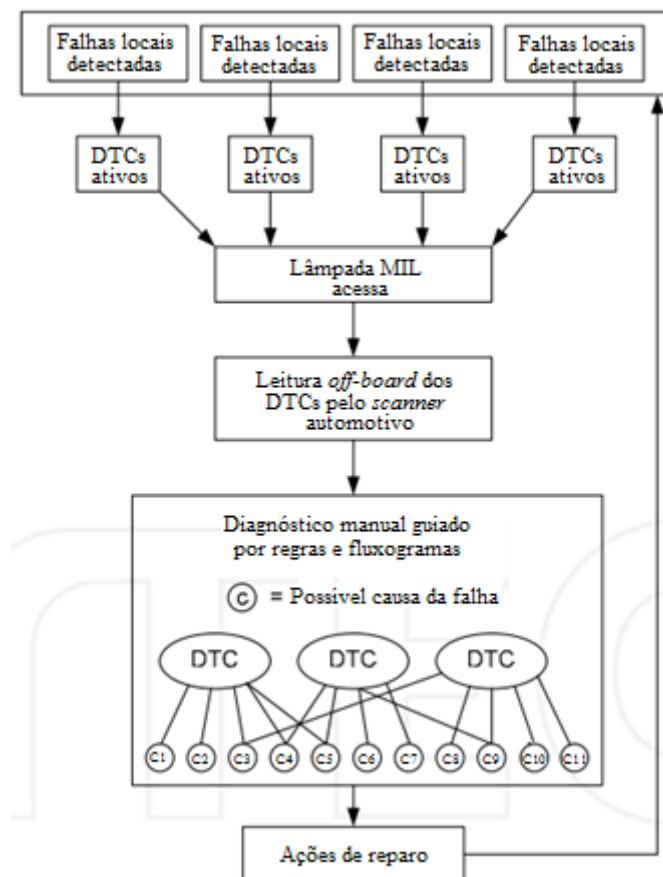


Figura 14 - Diagnóstico automotivo com múltiplos DTCs ativos  
**Fonte: Adaptado de LANIGAN (2011).**

Nota-se na Figura 14 o caso dos múltiplos códigos ativos; observa-se ainda que cada DTC compreende um conjunto particular de componentes passíveis de falha (*root-causes*). Entretanto, tem-se componentes individuais que são compartilhados por diferentes DTCs.

Esse cenário de sobreposição de eventos de falha associados à diferentes DTCs representa uma dificuldade adicional no procedimento de diagnóstico, pois no caso da ativação de múltiplos DTCs o técnico em manutenção tem a disposição um conjunto de sequências de testes (BDDs) de igual relevância, tornando o procedimento de diagnóstico pouco sistemático, pouco intuitivo e não sequencial.

Assim sendo, um dos objetivos específicos desse trabalho é a criação de um algoritmo para a construção de um BDD que privilegie testes em componentes comuns, pois, a princípio, tais componentes possuem uma maior probabilidade de falha, tendo em vista a presença dos mesmos

em diferentes DTCs. Uma análise probabilística pode determinar a precedência dos testes em determinados componentes do BDD. A análise considera tanto a probabilidade de uma falha individual do componente quanto o custo de realização do teste sob o respectivo componente. O funcionamento detalhado da abordagem proposta é apresentado na seção 3.1.

Após a detecção de uma ou mais falhas no veículo, um ou mais DTCs são ativados e armazenados nas unidades de controle do automóvel. Além disso, a lâmpada indicadora MIL se acende no painel do veículo, indicando ao condutor que algum problema foi encontrado e um procedimento de manutenção ou reparo é necessário. Na Figura 15, tem-se alguns dos diferentes modelos de lâmpadas MIL empregadas em veículos comerciais. Quando detectadas falhas mais graves, o veículo pode ir ao estado de *fail-safe* em que o veículo é desligado à um nível de segurança; ou ao estado de *fail-operational* ou *limp-mode* em que o sistema mantém suas funcionalidades limitadas, de forma que o motorista possa levar o veículo em segurança à manutenção (LANIGAN; KAVULYA; NARASIMHAN, 2011).



Figura 15 - Diferentes modelos de lâmpadas MIL  
Fonte: NC ENVIRONMENT QUALITY (2016).

Após a identificação da falha pela unidades de controle, as mesmas registram os códigos associados para posterior investigação quando o sistema encontra-se parado. Num procedimento *off-board* e com o auxílio de ferramentas de diagnóstico, testes são realizados de forma a determinar a localização da falha; para posterior substituição ou reparo do componente avariado.

Um exemplo da necessidade de diagnóstico *off-board* é a presença de uma falha no sensor de posição do pedal do acelerador. O sensor de posição do pedal do acelerador é constituído por dois potenciômetros, um principal e um de segurança, localizados no mesmo invólucro. O sensor reconhece a posição do pedal do acelerador e controla a quantidade de combustível a ser utilizada no processo de combustão, gerando um torque proporcional. Ambos os potenciômetros são energizados por fontes de alimentação diferentes, de forma a tornar o sistema mais robusto à falhas na alimentação. Outro interessante mecanismo que aumenta a confiabilidade do sistema é a diferença entre as tensões de saída dos dois potenciômetros. Ambas as tensões variam proporcionalmente com a posição do acelerador, entretanto um dos potenciômetros possui valor

instantâneo igual à metade do valor do outro (REPARADOR FIAT, 2016). Isso permite inferir corretamente qual dos potenciômetros possui falha mecânica, num eventual caso de detecção de erro.

No caso de detecção de falha em um dos potenciômetros o segundo potenciômetro é acionado e o veículo inicia o estado de *limp-mode*, em que o torque máximo é limitado a valores predeterminados de forma a minimizar a ocorrência de uma segunda falha no mesmo sistema. Além disso, a luz indicadora MIL e o respectivo código de falha são ativados. Em caso de avaria simultânea nos dois potenciômetros, o veículo não aceita comando do pedal do acelerador.

Apesar do exemplo acima compreender um caso em que tem-se um segundo componente de reserva, esse tipo de técnica em sistemas automotivos é pouco utilizada devido aos altos custos da adição de componentes redundantes. Geralmente, apenas componentes com baixo custo e fácil conexão e desconexão utilizam dessa técnica.

No caso descrito acima, um sensor de posição do pedal do acelerador pode estar obtendo valores que não correspondem aos valores de velocidade do motor lidos pela unidade de controle. Esse tipo de discrepância causa a ativação de um DTC associado. Entretanto, um problema detectado no sensor do pedal nem sempre é causado por um defeito no próprio sensor. Pode ser causado, por exemplo, por problemas no sensor de posição do virabrequim, no sensor de pressão, no sensor de massa de ar, na ignição, na unidade de controle, e nos chicotes elétricos e conectores. Nesse contexto, um procedimento de identificação *off-board* da falha, realizado por um técnico especialista no serviço de manutenção, é necessário.

### 2.2.9 Dificuldades no Diagnóstico Atual

A medida que os veículos são equipados com maiores funcionalidades e conseqüentemente com maior complexidade eletrônica embarcada, se torna cada vez mais difícil identificar a parte irregular do sistema; se é uma falha na ECU, no sensor, no atuador, nos chicotes, etc. (OGAWA; MOROZUMI, 2002).

Devido à esse aumento de complexidade, a probabilidade de ocorrência de falhas não previstas nos ciclos de projeto e desenvolvimento do automóvel aumenta significativamente

(SUWATTHIKUL, 2010). Dessa forma, os DTCs e as ferramentas de diagnóstico atuais estão cada vez menos aptas à identificar o componente falho responsável pela ativação do respectivo DTC (LANIGAN; KAVULYA; NARASIMHAN, 2011). A falta de informações precisas de diagnóstico, que provocam altos índices de “Falha Não Encontrada” (*No Fault Found*) e a pressão para resolver os problemas rapidamente, fazem com que os técnicos de manutenção substituam vários componentes sequencialmente num procedimento instintivo na esperança que alguma reposição resolva o problema (SÖDERHOLM, 2007); muitas vezes sem utilizar as ferramentas disponíveis, realizando um procedimento não sistemático baseado no método da tentativa e erro (THOMAS; AYERS; PECHT, 2002). No pior caso, componentes falhos permanecem no veículo sem diagnóstico e reparo. No melhor, componentes funcionais são substituídos sem necessidade.

Esse cenário é frequente na indústria automotiva pois as concessionárias são obrigadas, após o aparecimento do código, a assumir a ocorrência de falhas caso não haja evidências do correto funcionamento do componente (LANIGAN; KAVULYA; NARASIMHAN, 2011). A reposição de componentes sem o devido diagnóstico é umas maiores causas de “Falha Não Encontrada” e dos altos custos de garantia. Apenas em 2015, o mercado automotivo americano teve um custo de aproximadamente 10 bilhões de dólares com custos de garantia (WARRANTY WEEK, 2016), incluindo trocas de componentes por defeitos de fabricação, horas-homem trabalhadas, indenizações e procedimentos de *recall*.

Considerando os altos custos de garantia, é razoável que os maiores esforços dedicados à diagnóstico de falhas em sistemas automotivos se estabeleçam para a criação de novas soluções *off-board*, no intuito de a reduzir o número/custo de reivindicações de garantia e substituição de componentes (LANIGAN; KAVULYA; NARASIMHAN, 2011).

### 3. DESENVOLVIMENTO

Um dos objetivos específicos deste trabalho é o desenvolvimento de uma estratégia de composição das árvores de diagnóstico individuais, para o caso de múltiplos DTCs ativos. Tal estratégia reduziria o tempo de diagnóstico fornecendo ao técnico uma única sequência de testes baseada no conhecimento individual das árvores e em uma estratégia probabilística. A seção 3.1 detalha essa estratégia.

Um segundo objetivo específico é o desenvolvimento de uma base de dados externa, que armazena os dados de diagnóstico obtidos após cada procedimento, criando um histórico de falhas e de diagnóstico inter-concessionárias. Quando inseridos novos dados, os parâmetros que definem a prioridade dos testes nos procedimentos futuros são alterados em tempo real. Isso altera diretamente a estratégia de síntese apresentada na seção 3.1 e permite ao técnico um conhecimento de ocorrência de falhas mais abrangente, pois é disposto ao técnico informações atualizadas de diagnóstico à nível regional.

Esses objetivos serão atingidos por meio do SOAD – Sistema *Offline* de Auxílio ao Diagnóstico, que é o *software* para computador (PC) proposto nesse trabalho. O SOAD tem o objetivo de ser uma ferramenta auxiliar completa para PC que não só dispõe da estratégia de síntese e do acesso à base de dados, mas também das funcionalidades usuais de uma ferramenta de diagnóstico utilizada na indústria, incluindo cadastro de informações, acesso à informações referente aos ensaios e valores de referência, etc. A caracterização completa do SOAD e suas funcionalidades serão descritas na seção 3.2.

#### 3.1 ESTRATÉGIA DE SINTETIZAÇÃO PARA MÚLTIPLOS DTCS

O processo de diagnóstico envolve múltiplas etapas, que vão desde a detecção da falha *on-board* à identificação *off-board*, utilizando de estruturas desenvolvidas no estágio de projeto do veículo. Esse processo é ilustrado pela Figura 16.

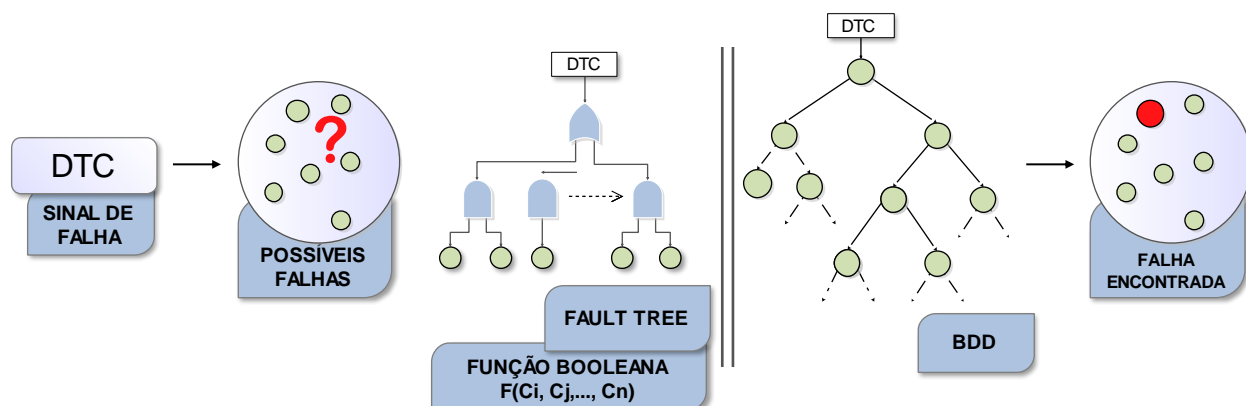


Figura 16 - Procedimento de diagnóstico a partir do sinal de falha identificado  
**Fonte: Autoria própria.**

Quando uma falha é detectada pelo veículo é gerado um DTC correspondente. Esse DTC mapeia um conjunto formado pelos componentes passíveis de falhas. A relação lógica entre o DTC e as possíveis causas é mapeada pela árvore de falhas. A árvore de falhas codifica uma função booleana particular que pode ser traduzida em um BDD. O BDD é utilizado para conduzir o procedimento de diagnóstico, provendo ao técnico a sequência dos testes e ensaios a serem realizados. A partir desse procedimento é possível encontrar o componente falho que originou o DTC, para posteriormente substituí-lo ou repará-lo.

Entretanto, um problema frequente é a ativação de múltiplos códigos ativos devido à uma mesma falha (ou à um conjunto mínimo de falhas). Esse fenômeno é ilustrado na Figura 17.

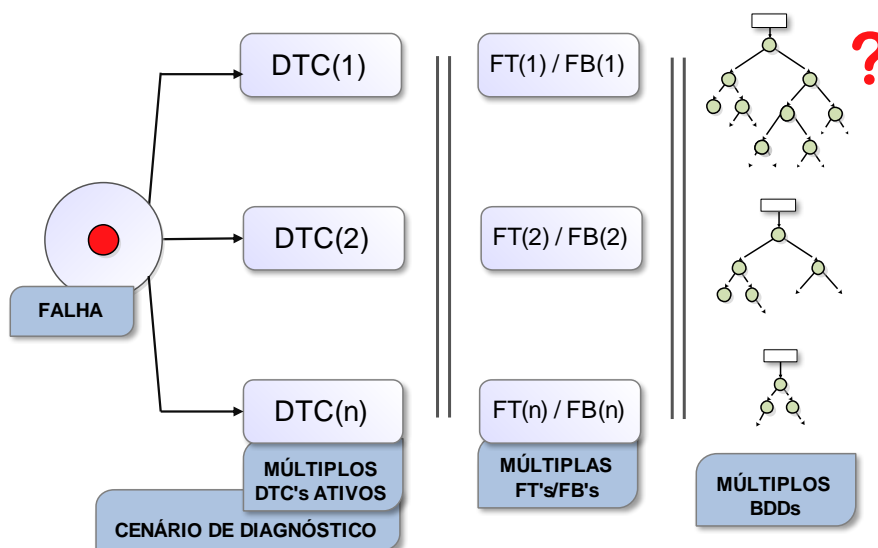


Figura 17 - Cenário de diagnóstico com múltiplos DTCs ativos  
**Fonte: Autoria própria.**

Nesse caso, é apresentado ao técnico um *cenário de diagnóstico*, que especifica o conjunto de DTCs ativos e o conjunto de DTCs inativos. Tem-se aqui o problema descrito no Capítulo 2, em que múltiplos BDDs presentes e sem relação entre si estabelecem um procedimento de diagnóstico confuso e pouco sistemático que necessita da experiência do técnico de manutenção para sua realização. Assim, a metodologia busca a síntese de um único BDD resultante a partir dos BDDs individuais, cujo objetivo é facilitar o diagnóstico no contexto com múltiplos DTCs ativos, provavelmente economizando o tempo de parada para manutenção.

O BDD resultante privilegia testes sobre os componentes mais prováveis de falha. A probabilidade de falha de cada componente é calculada analisando-se o cenário de diagnóstico corrente. Considere, por exemplo, um simples cenário em que é disposto ao técnico três BDDs que apresentem um teste em um componente em comum. Provavelmente, um procedimento de teste sobre esse componente deva ser priorizado, pois é mais provável a ocorrência de uma única falha (no componente comum) do que a ocorrência de três falhas em componentes individuais. Naturalmente, a metodologia de síntese é sistemática e apresenta argumentos probabilísticos retirados da literatura para justificar a preferência entre os testes.

A estratégia de síntese e suas etapas são apresentadas nas subseções seguintes.

### 3.1.1 Obtenção dos Cortes Mínimos

A estratégia de síntese inicia-se pela obtenção dos cortes mínimos da função booleana de cada BDD. Os cortes mínimos ou as soluções mínimas associadas a um código de falha é todo conjunto mínimo de componentes falhos capaz de gerar aquele código. Esse conceito pode ser melhor entendido a partir da Figura 18.



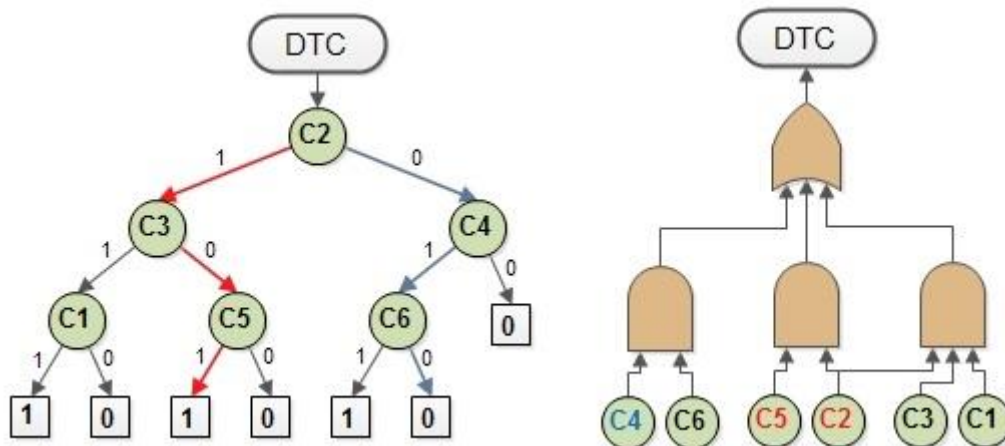


Figura 18 - Árvore de falhas e BDD para exemplificação dos cortes mínimos

Fonte: Autoria própria.

A expressão booleana relacionada ao diagrama lógico da árvore de falhas da Figura 18 é  $DTC = (C4 \text{ AND } C6) \text{ OR } (C2 \text{ AND } C5) \text{ OR } (C2 \text{ AND } C3 \text{ AND } C1)$ . Observando-se a árvore de falhas e a expressão booleana pode-se concluir que existem três combinações mínimas de falha que podem gerar o respectivo DTC: os componentes C4 e C6; os componentes C2 e C5; e os componentes C2, C3 e C1. Cada conjunto de componentes é um corte mínimo da árvore de falhas e é um termo da expressão booleana na sua forma disjuntiva.

É possível identificar os cortes mínimos no BDD observando-se os caminhos do nó raiz até as folhas individuais. O produto dos componentes cujo resultado é FAIL (valor binário 1) de um caminho do nó raiz até uma folha terminada em '1' especifica um corte mínimo da árvore de falhas ou uma solução mínima do BDD. Um exemplo de uma solução mínima válida corresponde aos componentes C2 e C5, presentes no caminho em vermelho do BDD da Figura 18. A mesma lógica pode ser estendida para os outros caminhos da árvore, gerando um conjunto de soluções total composto das soluções mínimas individuais. Note que o componente C3 não está presente na solução pois, nesse caminho, a saída do nó é PASS (valor binário 0). Além disso, note que um caminho terminado em '0' também não especifica um conjunto mínimo de falhas capaz de ativar o DTC correspondente. No caso do caminho em azul apenas o componente C4 é acusado como falho. Observando-se a árvore de falhas confirma-se que uma falha em apenas C4 não é o suficiente para a ativação do DTC.

Uma distinção importante é entre corte e corte mínimo (solução e solução mínima). Naturalmente, os componentes C1, C2, C3, C4, C5 e C6 formam um conjunto de falhas, que, quando falhos, ativam o DTC. Dessa forma, esse conjunto representa um corte da expressão

booleana. Entretanto, não representa um corte mínimo. Note que se a expressão booleana possuísse o termo  $(C1 \text{ AND } C2 \text{ AND } C3 \text{ AND } C4 \text{ AND } C5 \text{ AND } C6)$  o mesmo seria eliminado num procedimento de minimização. Na estratégia proposta, trabalha-se apenas com as soluções mínimas, pois retira-se da análise casos muito improváveis de falha, como o acima, que envolve falha em todos os componentes; e concentra-se apenas nos casos mais prováveis, que assumem um conjunto mínimo de falhas para a geração dos múltiplos DTCs.

A estratégia utilizada para obter as soluções das BDDs individuais é baseada em um algoritmo apresentado em Rauzy (1993). O algoritmo utiliza da recursividade para percorrer os caminhos do BDD e armazenar as soluções existentes. O resultado é o conjunto completo de soluções do BDD. Cada solução especifica um conjunto de componentes, que, quando falhos simultaneamente, são capazes de ativar o DTC correspondente.

### 3.1.2 Concatenação dos Cortes Mínimos Individuais

Considere o conjunto de DTCs total ( $S_T$ ) de um sistema (ou subsistema) a ser analisado. Considere também o conjunto de DTCs ativos dado por  $S_A = \{f_1, f_2, \dots, f_n\}$ , e o conjunto de DTCs inativos dado por  $S_I = \{g_1, g_2, \dots, g_m\}$ .  $f_i$  ( $i = 1, 2, \dots, n$ ) é uma expressão booleana associada a um DTC ativo; enquanto que  $g_j$  ( $j = 1, 2, \dots, m$ ) é a expressão booleana de um DTC não ativo.

Também, as seguintes expressões são válidas:

$$S_A \subset S_T$$

$$S_I \subset S_T$$

$$S_A \cup S_I = S_T$$

$$S_A \cap S_I = \emptyset$$

O cenário de diagnóstico é portanto definido pela composição dos conjuntos  $S_A$  e  $S_I$ . Como a estratégia objetiva sintetizar um único BDD, tal estrutura deve ser capaz de reproduzir as relações lógicas individuais e satisfazer o cenário de diagnóstico corrente. Nesse sentido, para capturar o contexto dos múltiplos DTCs ativos realiza-se a operação lógica AND entre os conjuntos de cortes mínimos desses DTCs. Após a operação conjuntiva minimiza-se a expressão booleana resultante.

O resultado dessa operação é um conjunto de soluções mínimas (ou cortes mínimos) capaz de ativar todos os DTCs simultaneamente.

Considere por exemplo, um cenário cujos DTCs ativos possuem as relações lógicas representadas pelas expressões booleanas abaixo.

$$f_1 = C_1C_2 + C_3$$

$$f_2 = C_1 + C_2C_4$$

Como ambos os DTCs são considerados ativos é necessária uma combinação de falhas nos componentes do sistema de tal forma que ambas as funções  $f_1$  e  $f_2$  assumam valor binário '1'. Note por exemplo, que apenas a falha no componente  $C_3$  não é suficiente para estabelecer o cenário de diagnóstico proposto, pois a expressão  $f_2$ , associada à um DTC ativo, não assumiria o valor '1'. A operação de conjunção das funções booleanas  $f_1$  e  $f_2$  e a de posterior minimização, para o exemplo considerado, são mostradas abaixo.

$$f_{AND} = f_1 \text{ AND } f_2$$

$$f_{AND} = (C_1C_2 + C_3) \cdot (C_1 + C_2C_4)$$

$$f_{AND} = C_1C_2 + C_1C_2C_3 + C_1C_3 + C_2C_3C_4$$

$$f_m = \min(f_{AND})$$

$$f_m = C_1C_2 + C_1C_3 + C_2C_3C_4$$

A função  $f_m$  codifica os conjuntos mínimos de falha capazes de acionar todos os DTCs ativos. Ou seja, cada corte mínimo de  $f_m$  é uma provável condição de falha para o estabelecimento de todos os códigos simultaneamente.  $f_m$  é portanto uma representação booleana do cenário de diagnóstico. RAUZY (1993) apresenta um algoritmo recursivo para a realização da operação lógica de conjunção sobre as diferentes expressões booleanas, cujo resultado é a função  $f_m$ .

A fim de diminuir as possibilidades de falha e aprimorar ainda mais o procedimento, é acrescentada a informação dos códigos inativos (presentes no conjunto  $S_I$ ) à concatenação. Esse conceito é similar à incorporação de *evidências* ao diagnóstico presente em Assaf (2008). Neste último, o autor acrescenta ao diagnóstico o que chama de *evidências*, que são sinais adicionais vindo da monitoração de sensores que não fazem parte das BDDs originais. Essa integração causa a redução do número de testes necessários para a localização da causa da falha, pois diminui o conjunto de possibilidades de falha. No caso desse trabalho, o conjunto de possibilidades pode ser reduzido significativamente, pois, em geral,  $|S_I|$  é consideravelmente maior do que  $|S_A|$ . Ou seja, a quantidade de informação adicionada ao problema é bastante expressiva.

A incorporação dos códigos inativos ao processo de concatenação pode ser entendida a partir da análise de um exemplo. Considere a adição de um código que não foi ativado ao cenário descrito anteriormente. A expressão booleana associada à esse código é dada por:

$$g_1 = C_2 + C_3C_4$$

Como o código associado à função  $g_1$  está inativo é possível inferir que os cortes mínimos de  $g_1$  especificam conjuntos de componentes completamente operacionais, pois caso estivessem falhos, o código seria ativado. Pode-se inferir por exemplo que o componente  $C_2$  não possui falha, pois caso estivesse falho o DTC da função  $g_2$  estaria ativado. Entretanto, não é possível inferir a presença de falhas em  $C_3$  ou  $C_4$  diretamente. Apenas é possível afirmar que não existem falhas em ambos os componentes  $C_3$  e  $C_4$ . Note que se somente o componente  $C_3$  (ou somente  $C_4$ ) apresentar falha, o DTC em questão continua não sendo acionado.

A análise que utiliza somente os DTCs ativos resulta em uma função ( $f_m$ ) que possui três cortes mínimos:  $C_1C_2$ ,  $C_1C_3$  e  $C_2C_3C_4$ . Cada corte representa um conjunto mínimo de falhas capaz de gerar os DTCs. Entretanto, pode-se reduzir as possibilidades de falha adicionando as informações presentes em  $g_1$ . Incluindo esse conhecimento, o estabelecimento do cenário de diagnóstico  $f_1$ ,  $f_2$  e  $g_1$  devido à falhas em  $C_1C_2$  e  $C_2C_3C_4$  se torna impossível, pois sabe-se de  $g_1$  que o componente  $C_2$  não apresenta falhas.

Assaf (2008) apresenta um algoritmo para a incorporação de informações adicionais (nesse caso as expressões presentes em  $S_I$ ) à função de representação do sistema ( $f_m$ ). O produto desse algoritmo é a criação de uma expressão booleana resultante ( $f_R$ ) que satisfaz o cenário de diagnóstico considerado. Nesse sentido, os cortes mínimos de  $f_R$  são capazes de ativar todos os DTCs presentes em  $S_A$ ; e, ao mesmo tempo, não ativar qualquer DTC presente  $S_I$ . No caso do exemplo anterior, a função booleana resultante é  $f_R = C_1C_3$ .

### 3.1.3 Ordenação dos Testes

O produto do processo de concatenação é a formação de uma função booleana  $f_R$  que satisfaz o cenário de diagnóstico por completo e especifica os conjuntos mínimos de componentes passíveis de falha. Como mencionado no Capítulo 2, diversos algoritmos realizam a conversão de

uma árvore de falhas (ou da expressão lógica que a define) para o BDD. Entretanto, é necessário definir uma ordem a qual os testes serão realizados. Em geral, a ordenação das variáveis e tem um impacto significativo no tamanho do BDD resultante.

### 3.1.3.1 CDIF (*Cost of Diagnostic Importance Factor*)

Como nesse trabalho deseja-se sintetizar um BDD que reduza o número médio de testes, escolheu-se como parâmetro de ordenação o CDIF (*Cost of Diagnostic Importance Factor*), introduzido em Assaf (2008). O CDIF considera tanto a probabilidade de falha em um componente dado o cenário de diagnóstico atual, quanto o custo de realização de cada teste. O custo de execução do teste é um parâmetro arbitrário, que, em geral, é proporcional ao tempo médio de execução ou do seu custo monetário.

O CDIF é uma função do DIF (*Diagnostic Importance Factor*), que é uma medida da probabilidade de falha de um determinado componente, calculado a partir do cenário de diagnóstico (conjunto de DTCs ativos e inativos) e da probabilidade de falha individual – ver detalhes em Rauzy (1993) e em Dutuit e Rauzy (2014). A probabilidade de falha individual de cada componente é obtida na fase de projeto do veículo e é resultado das análises FMEA e por árvore de falhas. Dessa forma, considerando-se a ocorrência de um conjunto específico de DTCs ativos, o DIF discrimina probabilisticamente quais variáveis, ou possíveis causas, são mais prováveis de terem ocorrido para a ativação daquele conjunto de DTCs particular.

Além de considerar as probabilidades de falha, o CDIF introduz à ordenação um parâmetro relacionado ao custo de realização do teste. Esse parâmetro é uma razão entre o DIF e o custo de diagnóstico sob o componente e resulta num parâmetro que privilegia testes sob componentes com alta probabilidade e com baixo custo de execução do teste. É importante considerar o custo de execução do teste na ordenação, pois é vantajoso realizar um teste de fácil medição sob um componente com baixa probabilidade de falha do que realizar um teste despendioso em um componente com uma probabilidade de falha um pouco maior.

### 3.1.3.2 IG (*Information Gain*)

Além do CDIF, uma outra métrica de ordenação de BDDs existente é o IG (*Information Gain* ou Ganho de Informação).

O ganho de informação é uma métrica utilizada em inteligência artificial para selecionar os atributos que melhor classificam exemplos, segundo um modelo de árvore de decisão binária (BDD) num problema de classificação (MITCHELL, 1997).

Particularmente, o IG é definido como o ganho de informação obtido a partir da partição dos exemplos por um atributo específico. No contexto de diagnóstico, os atributos traduzem-se em componentes e os exemplos se tornam todas as combinações possíveis de estados (falho/operacional) dos componentes presentes na equação booleana. Na equação (3) tem-se a fórmula do IG obtida de Mitchell (1997).

$$IG(S, a) \equiv Entropia(S) - \sum_{v \in vals(a)} \frac{|S_v|}{|S|} \cdot Entropia(S_v) \quad (3)$$

Em que  $v$  é o conjunto de todos os valores possíveis do atributo  $a$  e  $S_v$  é o subconjunto de  $S$  em que  $a$  assume o valor  $v$ . No contexto de diagnóstico,  $a$  é uma variável booleana que assume valor '1' quando o teste sobre o componente acusa falha e '0' quando o resultado especifica o componente como operacional; dessa forma  $v \in \{0, 1\}$ . A  $Entropia(S)$ , cuja equação é apresentada em (4), caracteriza a impureza da equação booleana quanto às combinações individuais dos estados lógicos de suas variáveis. Ou seja, quanto maior a diferença entre a fração de combinações das variáveis booleanas em que se obtém  $S = 1$  e a fração de combinações em que se obtém  $S = 0$ , maior é a entropia.

$$Entropia(S) \equiv -p_+ \log_2 p_+ - p_- \log_2 p_- \quad (4)$$

Em (4)  $p_+$  é a fração das combinações das variáveis que  $S$  assume valor '1' e  $p_-$  é a fração das combinações em que  $S$  assume o valor '0'.

Considere por exemplo a equação booleana  $DTC = c_1 + c_2c_3$ , cuja tabela verdade aparece na Figura 19.

c1	c2	c3	DTC
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Figura 19 - Tabela verdade da equação booleana  $S = a + bc$   
**Fonte: Autoria própria.**

Note que cada combinação de estados (falho/operacional) dos componentes  $c_1$ ,  $c_2$  e  $c_3$  especifica um resultado booleano de ativação ou não do DTC. Nesse caso, tem-se um resultado positivo (DTC ativo) em 5 das 8 combinações. O cálculo da entropia para essa equação booleana é mostrado abaixo.

$$Entropia(S) = -\left(\frac{5}{8}\right) \log_2 \left(\frac{5}{8}\right) - \left(\frac{3}{8}\right) \log_2 \left(\frac{3}{8}\right) = 0.9544$$

Para determinar a prioridade entre os testes sobre os componentes é necessário obter o ganho de informação ao realizar cada teste individualmente. Dessa forma, deve-se calcular a  $Entropia(S_{c_i=0})$  e  $Entropia(S_{c_i=1})$  para todos os componentes  $c_i$  presentes na equação booleana. Por exemplo, para calcular o ganho de informação após um teste em  $c_1$  é necessário calcular as seguintes entropias:  $Entropia(S_{c_1=0})$  e  $Entropia(S_{c_1=1})$ , utilizando a equação (4). Os valores obtidos são:  $Entropia(S_{c_1=0}) = 0.5436$  e  $Entropia(S_{c_1=1}) = 0$ . Com esses valores pode-se calcular  $IG(S, c_1)$  segundo a equação (3); como é mostrado abaixo:

$$IG(S, c_1) = 0.9544 - \left(\frac{1}{2}\right) \cdot 0 - \left(\frac{1}{2}\right) \cdot 0.5436 = 0.6826$$

Seguindo esse mesmo raciocínio pode-se obter os valores de IG para  $c_2$  e  $c_3$ , obtendo-se  $IG(S, c_2) = IG(S, c_3) = 0.0488$ .

A partir dos valores obtidos, prioriza-se o teste em  $c_1$  devido ao maior ganho de informação associado. Observe que esse resultado é esperado, pois a princípio, sem o conhecimento das probabilidades individuais de falha dos componentes, é mais provável um cenário em que o componente  $c_1$ , em estado de falha, esteja causando a ativação do DTC individualmente, do que um cenário onde duas falhas simultâneas (em  $c_2$  e  $c_3$ ) causem essa ativação.

Pode-se observar no exemplo descrito acima que o IG, como definido em (3), não utiliza a probabilidade de falha individual dos componentes para a classificação. Dessa forma, pode-se utilizar a métrica  $IG \cdot Pr(c_i)$  (em que  $Pr(c_i)$  é a probabilidade de falha do componente  $c_i$ ), que prioriza testes em componentes com algo ganho de informação e alta probabilidade de falha.

#### 3.1.4 Formação do BDD Resultante

A partir da função booleana  $f_R$  que modela e satisfaz o cenário de diagnóstico corrente e da prioridade entre testes definida pelo CDIF é possível sintetizar o BDD final. Esse BDD é a ferramenta de auxílio que conduz o diagnóstico expondo ao técnico a sequência de testes mais eficiente. Assaf e Dugan (2005), Andrews e Remeneye (2005) e Rauzy (1993) apresentam métodos para obtenção da BDD dada a expressão booleana e a ordenação.

É importante salientar que a estratégia resulta em um BDD resultante específico para cada cenário de diagnóstico, personalizando o procedimento para conjunto de DTCs ativos e inativos. Sendo assim, ela analisa o cenário, calcula o CDIF e reconfigura a sequência de testes a cada procedimento a fim de evitar testes desnecessários. A estratégia de síntese descrita nessa seção é realizada pela ferramenta proposta, cuja discussão é apresentada na seção seguinte.

#### 3.1.5 Custo de Diagnóstico

Quantitativamente, é possível diferenciar o custo de diagnóstico de um procedimento que considera as árvores individuais de um que utiliza a estratégia combinada. Tal custo é apresentado explicado em detalhes em Oliveira e Lüders (2016) e apresentado em (1). É uma média ponderada dos custos das soluções do BDD, sendo os pesos as probabilidades de falha dos componentes presentes na solução.



$$DC(b) = \frac{1}{Q_s} \sum_{i=1}^n \left[ \prod_{j=1}^{m(i)} [w_j \cdot \Pr\{c_j\} + (1 - w_j) \cdot (1 - \Pr\{c_j\})] \right] \cdot cp_i \quad (1)$$

Em (1),  $Q_s$  é a falta de fiabilidade do BDD cuja equação é apresentada em (2),  $n$  é o número de soluções (ou cortes) presentes no BDD,  $m(i)$  é o número de componentes presentes na solução  $i$  e  $cp_i$  é a somatória dos custos individuais dos componentes presentes na solução  $i$ .

$$Q_s = 1 - \prod_{i=1}^n (1 - q_{sol_i}) \quad (2)$$

Em (2),  $q_{sol_i}$  é a falta de fiabilidade da solução  $i$  calculada pelo produto das probabilidades individuais de falha dos componentes presentes na solução.

De modo geral, a equação (1) pode ser utilizada para comparar quantitativamente duas BDDs individuais. Para comparação entre a abordagem convencional de diagnóstico (que utiliza dos BDDs individuais) e a abordagem proposta (que utiliza do BDD combinado), compara-se o custo do BDD combinado com a soma dos custos dos BDDs individuais relativos aos DTCs ativos presentes no cenário de diagnóstico sob análise.

### 3.2 SOAD (SISTEMA *OFF-LINE* DE AUXÍLIO AO DIAGNÓSTICO)

Um dos objetivos desse trabalho é a implementação da estratégia de síntese do BDD em um *software* para computador. Além de compreender a estratégia, a ferramenta deve possuir as funcionalidades usuais dessa interface presente na indústria. Desta forma, a proposta deste trabalho é a criação do SOAD, cujo objetivo é assistir o diagnóstico na fase de identificação da causa dos DTCs ativos. Utilizando casos de diagnóstico anteriores e algoritmos específicos de reordenação, uma reconfiguração de testes é feita dinamicamente, de forma a melhorar esse processo de identificação do componente com defeito.

O sistema consiste de um *software* desenvolvido em JAVA específico para ser executado em um computador pessoal (PC – *Personal Computer*). Além disso, o *software* possui conectividade com um servidor SQL (*Structured Query Language*). Numa aplicação real, o *software* funcionaria em um terminal de computador externo ao automóvel e conduziria o

procedimento de diagnóstico recebendo como entrada os DTCs ativos do diagnóstico atual e retornando ao técnico sequências de testes mais promissoras para o problema de diagnóstico específico. O processo completo de utilização do SOAD pelo técnico de manutenção é mostrado no diagrama da Figura 20. Nessa figura, os blocos descrevem as ações realizadas pelo técnico; com exceção dos blocos com o ícone “SOAD” à direita, que caracterizam ações realizadas pelo programa.

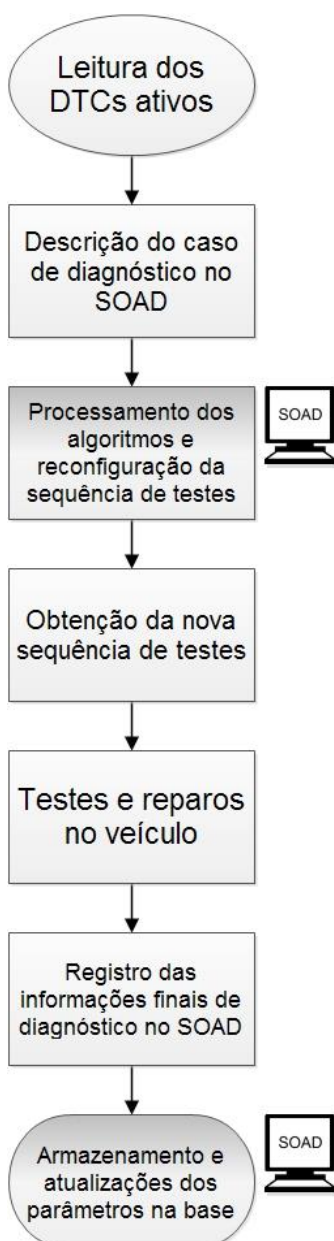


Figura 20 - Auxílio ao diagnóstico realizado pelo SOAD

Fonte: Autoria própria.

Assumindo a ocorrência de uma falha em um componente do veículo, que gerou um ou mais DTCs associados, o processo do diagnóstico assistido inicia-se pela leitura desses códigos pelo *scanner* automotivo. Esses DTCs são passados ao SOAD juntamente com a montadora e o modelo do motor do veículo. A partir da análise em seu banco de dados das estruturas individuais previamente cadastradas associadas a esse veículo e do cenário de diagnóstico atual, o SOAD reconfigura a sequência de testes aplicando a estratégia descrita na subseção 3.1. Além disso, a ferramenta pretende utilizar as padronizações ISO e SAE de diagnóstico apresentadas na subseção 2.2.2.

Com a reconfiguração, o procedimento *off-board* de testes pode ser realizado. Após o procedimento de identificação e reparo é possível agregar as informações desse procedimento particular à base de dados existente. Particularmente, é possível cadastrar para cada procedimento o cenário de diagnóstico a sua solução; ou seja, os conjuntos de DTCs ativos e inativos e o conjunto efetivo de falhas que causou a ativação dos DTCs ativos. Isso permite a alteração das probabilidades individuais de falha segundo o histórico de diagnósticos realizados.

### 3.2.1 Módulos Principais do SOAD

O SOAD pode ser dividido em módulos funcionais distintos como é mostrado na Figura 21.

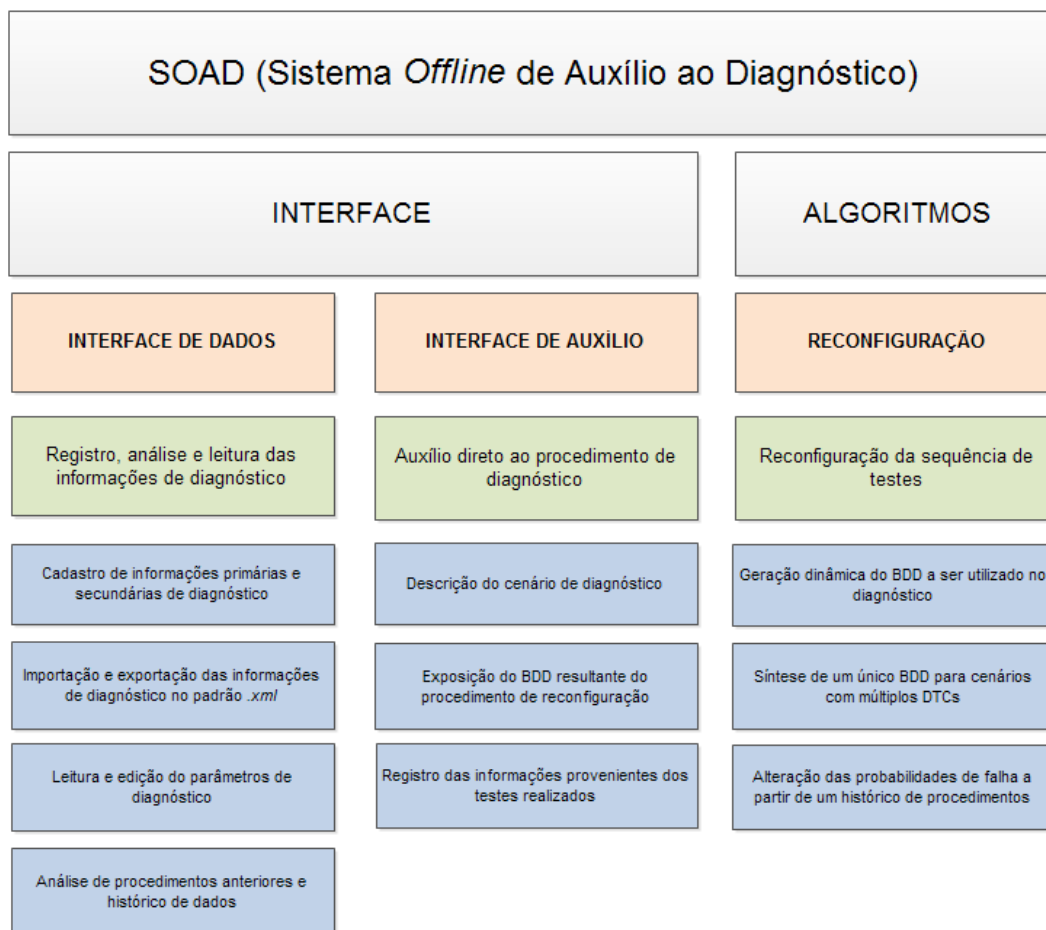


Figura 21 - Figura 21. Módulos funcionais presentes no SOAD

Fonte: **Autoria própria.**

A interface ao usuário é dividida em dois módulos principais: o módulo de cadastro e leitura das informações de diagnóstico (interface de dados) e o módulo de auxílio direto ao diagnóstico (interface de auxílio). Agregada à interface de auxílio tem-se a estratégia de síntese (descrita na seção 3.1) que utiliza dos algoritmos para realizar a reconfiguração dos testes.

A interface de dados permite ao usuário o acesso ao conhecimento armazenado na base, incluindo as informações primárias e secundárias de diagnóstico. Por informações primárias entendem-se as informações que se relacionam diretamente ao procedimento de síntese anteriormente descrito. Isso inclui as expressões booleanas associadas à cada código de falha, o identificador de cada teste e componente, as probabilidades de falha e os BDDs existentes. As informações secundárias incluem informações adicionais específicas a cada teste. Ou seja, descrevem e ilustram os procedimentos de ensaio, a instrumentação necessária e os valores de referência. Para acesso à essas informações é necessária uma interface de cadastro das mesmas.

Oliveira e Lüders (2013) apresentam uma interface de cadastro, que permite também a importação e a exportação dessas informações no padrão XML (*Extensible Markup Language*), comumente utilizado na indústria para codificar as BDDs. A interface de dados também permite o acesso às informações provenientes do histórico de procedimentos armazenados na base de dados existente.

A interface de auxílio permite a assistência ao procedimento de diagnóstico *off-board*, cujo processo é ilustrado na Figura 20. Nesse módulo descreve-se o cenário de diagnóstico e obtém-se a BDD resultante do processo de síntese. Além disso, registra-se a solução de diagnóstico obtida após o procedimento, que consiste nas falhas encontradas.

### 3.2.2 Especificações de uso do SOAD

O sistema SOAD é implementado em JAVA com conexão a um banco de dados SQL. A ferramenta consiste de uma interface com o usuário mais os algoritmos de reconfiguração, cujos módulos principais aparecem na Figura 21. O banco de dados SQL armazena os objetos criados pelo usuário.

O sistema possui cinco classes principais, a saber: Árvores, Componentes, Nos, Projetos e Testes, cujo diagrama de classes em formato UML (*Unified Modeling Language*) é apresentado na Figura 22. Por simplificação, fez-se uma abstração na representação da Figura 22, em que aparecem apenas os principais atributos e métodos de cada classe.

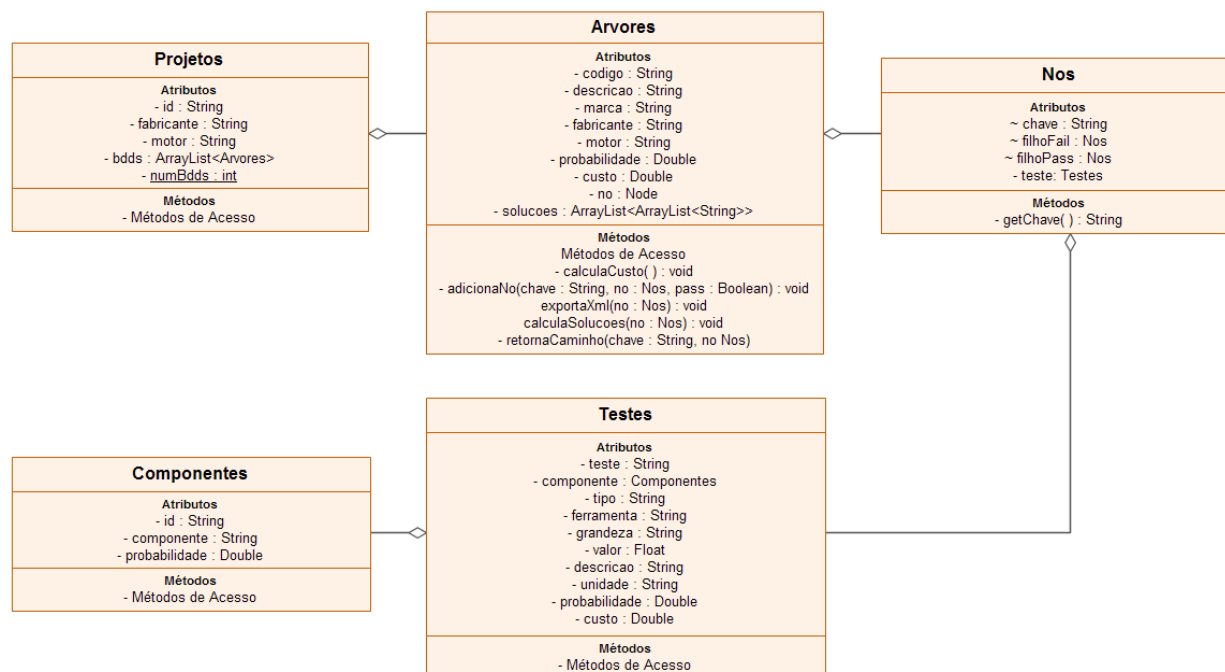


Figura 22 - Diagrama UML das principais classes existentes no SOAD

Fonte: Autoria própria.

Cada classe mencionada acima possui uma tabela no banco de dados, cujas entradas são os objetos respectivos à classe particular salvos pelo usuário. Para cadastro e acesso às informações presentes no banco de dados, duas classes auxiliares foram criadas, a classe ‘Conexao’, que efetivamente abre a conexão com o banco de dados através do *driver* disponível; e a classe ‘DadosConexao’, que possui os métodos PesquisarDado, EditarDado, DeletarDado, InserirDado, CriarTabela e ExcluirTabela para cada tabela existente. A conexão é realizada utilizando o JDBC (*Java Database Connectivity*), que é um conjunto de classes e interfaces que fazem o envio de instruções SQL para o banco de dados. Dessa forma, é possível fazer modificações no banco diretamente através da ferramenta. As tabelas existentes, seus atributos o acesso às informações pelo usuário e a relação deste com as classes principais, são explicados na seção de Resultados.

## 4. RESULTADOS

Nessa seção serão apresentadas detalhadamente as funcionalidades da ferramenta desenvolvida e a interação do usuário com a mesma. Informações relativas ao banco de dados SQL também são apresentadas. Especificamente, apresentam-se nessa seção as tabelas, janelas e painéis existentes de uma maneira funcional e não descritiva, ou seja, foca-se nas funcionalidades da ferramenta, abstraindo descrições e informações específicas da relação entre classes, bem como de seus métodos e atributos.

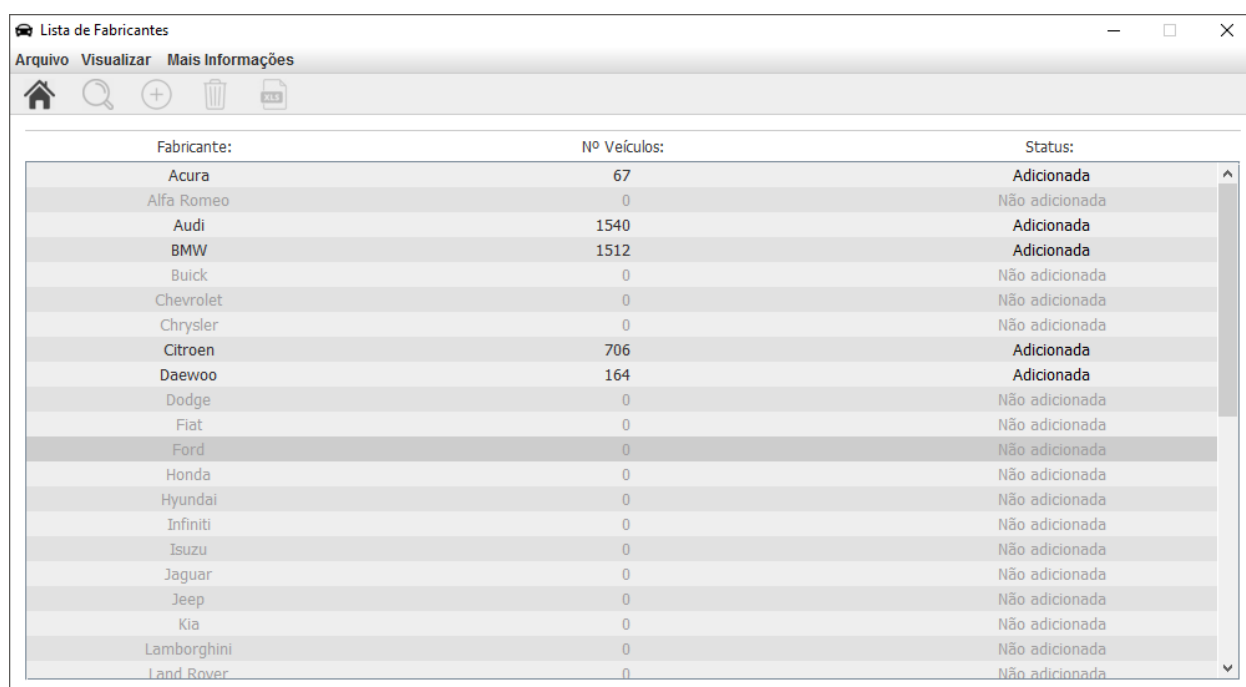
### 4.1 MÓDULO DADOS DE FABRICANTES

O SOAD é um *software* de auxílio ao diagnóstico que, em real funcionamento, necessitaria de informações da indústria automotiva para seu funcionamento. Especificamente, seria necessário, no mínimo, o conhecimento de componentes, testes, valores de referência, códigos de falha e árvores de diagnóstico. Entretanto, tais informações são de difícil acesso, o que dificulta a criação de uma ferramenta que trate de casos reais de diagnóstico. Dessa forma, a ferramenta desenvolvida nesse trabalho é um protótipo que implementa a estratégia de reconfiguração descrita na seção 3.1.

Entretanto, para se aproximar mais de uma aplicação real, foram utilizados dados de veículos obtidos nos seguintes sítios: OBDII CODES (<http://obdii.pro/>) e AUTO DATA (<http://www.auto-data.net/>). Particularmente, obtêm-se informações específicas de veículos de diversos fabricantes na página AUTO DATA; e de códigos de falha genéricos e particulares, além de seus significados, na página OBDII CODES.

Naturalmente, nem todas as informações relevantes ao diagnóstico estão presentes nessas fontes. Por exemplo, em geral, em OBDII CODES tem-se apenas uma parcela dos códigos associados ao sistema de transmissão (prefixo ‘P’) para cada fabricante. Além disso, algumas das informações obtidas nas páginas citadas podem estar incorretas ou desatualizadas.

Na primeira utilização, o SOAD busca automaticamente as informações de códigos de falha genéricos (padronizados entre os fabricantes) em OBDII CODES, permitindo a criação de BDDs relacionados a esses DTCs. Entretanto, para a criação de um BDD para um DTC específico, há a necessidade da inclusão do fabricante no banco de dados SQL. Apenas os fabricantes de veículos considerados relevantes no cenário automobilístico nacional e mundial podem ser adicionados ao banco. Na Figura 23 tem-se o painel que permite o usuário adicionar ou remover os fabricantes do banco de dados. Além disso, pode-se acessar diretamente em AUTO DATA e OBDII CODES as informações do fabricante via *browser*.



Fabricante:	Nº Veículos:	Status:
Acura	67	Adicionada
Alfa Romeo	0	Não adicionada
Audi	1540	Adicionada
BMW	1512	Adicionada
Buick	0	Não adicionada
Chevrolet	0	Não adicionada
Chrysler	0	Não adicionada
Citroen	706	Adicionada
Daewoo	164	Adicionada
Dodge	0	Não adicionada
Fiat	0	Não adicionada
Ford	0	Não adicionada
Honda	0	Não adicionada
Hyundai	0	Não adicionada
Infiniti	0	Não adicionada
Isuzu	0	Não adicionada
Jaguar	0	Não adicionada
Jeep	0	Não adicionada
Kia	0	Não adicionada
Lamborghini	0	Não adicionada
Land Rover	0	Não adicionada

Figura 23 - Painel de fabricantes e informações relacionadas

Fonte: Autoria própria.

Ao adicionar as informações de um fabricante específico no banco de dados, criam-se duas tabelas. A primeira armazena as informações de diferentes veículos daquele fabricante, como por exemplo, modelo, motor, potência, ano de fabricação, etc. No total são 79 atributos referentes a cada veículo pertencente à marca cadastrada, que vão desde o número de assentos ao tipo do sistema de combustível. Algumas informações podem não estar disponíveis para um veículo em particular. A segunda tabela armazena os códigos de falha particulares daquele fabricante, bem como os seus significados.



Após a inclusão do fabricante à base, pode-se exportar as informações armazenadas em formato *.xls* (arquivo Excel).

## 4.2 MÓDULO CONSTRUÇÃO DE BDDS

A principal contribuição do SOAD é realizar a estratégia de reconfiguração apresentada na seção 3.1. Para tal, é necessário cadastrar previamente um conjunto de BDDs na ferramenta. Dessa forma, é disponibilizado ao usuário uma interface para construção de BDDs. Esta seção trata desta etapa.

Como mencionado no Capítulo 2, existem dois tipos de códigos de falha: os códigos genéricos e os específicos de cada fabricante. Os códigos genéricos possuem um mesmo significado para todos os fabricantes, apesar da sequência de testes e dos componentes envolvidos, em geral, serem diferentes. Por exemplo, o código genérico P0198 especifica para todos os fabricantes um nível elevado de entrada no sensor de temperatura do óleo do motor. Já o código P1246 especifica uma falha na entrada de carga no alternador para veículos Peugeot, enquanto que para a marca Kia especifica uma entrada alta do sinal de manivela (OBDII CODES).

Em um caso real de diagnóstico, a sequência de testes e os procedimentos de instrumentação realizados são dependentes de uma série de fatores. Por exemplo, o conjunto de testes necessários para diagnosticar uma falha no sistema de transmissão é específico para veículos com um determinado tipo de motor, sistema de injeção, componentes auxiliares, etc. Ou seja, para um DTC particular, pode-se ter diferentes sequências de testes para um mesmo modelo de veículo, porém com diferentes especificações. Dessa forma, além do BDD estar relacionado a um código de falha, as sequências de testes que este BDD codifica estão relacionadas à um conjunto de especificações do veículo em questão. Seria necessário, portanto, especificar todas essas informações no momento de construção do BDD.

Devido à limitação de informações da indústria, para a criação de um BDD na ferramenta, é necessário especificar apenas o código de falha e, caso o código de falha seja específico de um fabricante, o modelo do motor. As informações dos códigos de falha e de modelos de motor são obtidas via *web* (ver detalhes na seção 4.1). A especificação do motor foi escolhida, pois a maioria

dos códigos de falha obtidos em AUTO DATA são do sistema de potência (prefixo ‘P’), cujas falhas estão, em geral, relacionadas ao motor do veículo e seus componentes auxiliares. Dessa forma, faz sentido especificar o tipo de motor na construção do BDD, pois para diferentes motores, o procedimento de diagnóstico é diferente.

Na Figura 24 tem-se a janela anterior à criação do BDD, onde se definem as especificações necessárias.

Figura 24 - Janela de especificação de informações para criação do BDD

**Fonte: Autoria própria.**

Definidas as especificações, é possível criar o BDD via diagrama, onde cria-se efetivamente a sequência de testes formando a estrutura do BDD passo-a-passo; ou via equação booleana. Nas subseções seguintes tem-se a descrição detalhada de ambos os processos de construção.

Vale lembrar que por “criação” ou “construção” entende-se apenas o cadastro dos BDDs na ferramenta, pois esses foram efetivamente criados na fase de projeto do veículo. Em geral, essas estruturas estão em documentos eletrônicos ou manuais de instrumentação disponíveis ao técnico de manutenção no momento do diagnóstico. Deseja-se portanto, cadastrar a estrutura do BDD (utilizada no processo de reconfiguração) mais algumas informações básicas, que permitam ao técnico acessar corretamente o conhecimento de instrumentação em outras fontes. Dessa forma, a ferramenta não tem o objetivo de substituir tais manuais e fontes de informação; assume-se que o técnico de manutenção disponha e utilize tais informações enquanto é guiado pela nova sequência de testes proposta pela ferramenta.

#### 4.2.1 Construção via Diagrama

No painel de construção do BDD via diagrama pode-se cadastrar um BDD adicionando sequencialmente nós do tipo PASS ou FAIL. Cada nó está associado a um teste específico. Dessa forma, existe a necessidade do cadastro de testes previamente à construção do BDD.

Como mostrado na Figura 9, cada teste é um procedimento completo de instrumentação e medição, que envolve uma grande quantidade de passos. Como nem todas essas informações são úteis à estratégia de reconfiguração, escolheu-se apenas alguns atributos básicos para o cadastro na ferramenta, são esses:

- ID\*;
- Componente\*;
- Tempo médio de execução (em minutos)\*.
- Tipo de teste;
- Ferramenta;
- Grandeza;
- Valor de referência;
- Unidade;
- Descrição.

As especificações marcadas com \* são obrigatórias, pois são necessárias para a estratégia de combinação. O restante das especificações foram adicionadas para à mínima caracterização do teste à ferramenta. A janela de cadastro de testes é mostrada na Figura 25.

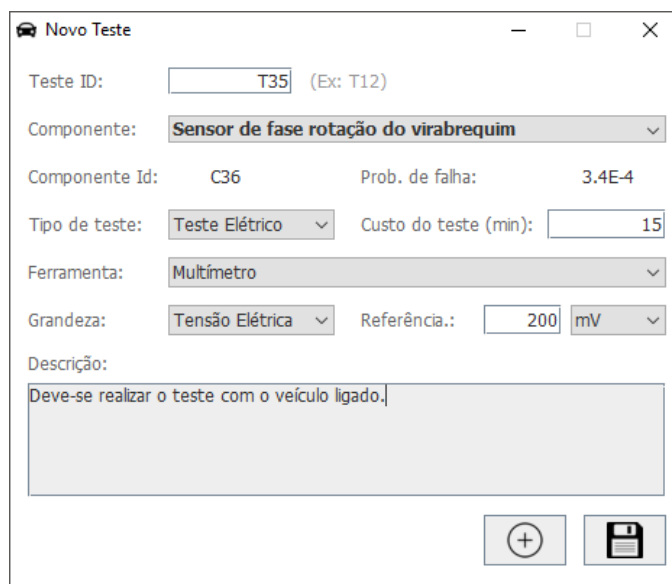


Figura 25 - Janela de cadastro de testes

Fonte: Autoria própria.

Como cada teste é realizado sobre um componente específico, anteriormente ao cadastro dos testes é necessário cadastrar novos componentes, especificando-se suas probabilidades de falha. A janela de cadastro de componentes é mostrada na Figura 26.

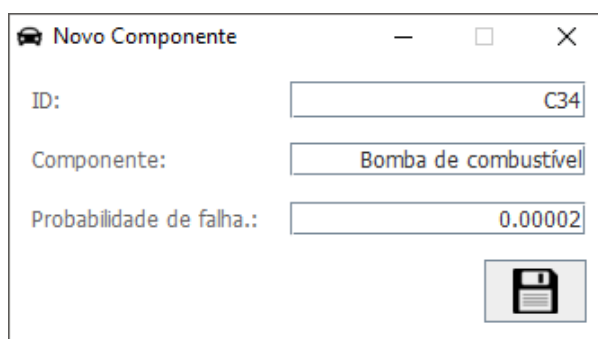


Figura 26 - Janela de cadastro de componentes

Fonte: Autoria própria.

Enquanto que as informações de código de falha, fabricante e motor são reais e obtidas via *web*, as especificações dos componentes e testes (Figuras 25 e 26) são apenas exemplos ilustrativos. Em aplicação real haveria a necessidade de cadastro dessas especificações; ou de integração com um banco de dados existente que contenham tais informações. No protótipo desenvolvido foram cadastradas apenas algumas especificações *default* (não existe a possibilidade

de cadastro de novas especificações) que, em geral, fazem parte do contexto automotivo. Estas especificações não são exaustivas e nem válidas para todos os fabricantes.

O cadastro de um novo teste cria uma nova entrada na tabela Testes no banco de dados SQL. Cada teste possui um ID único, formado por um prefixo 'T' seguido de um identificador numérico de até três dígitos. Dessa forma, a ferramenta verifica na base se o ID especificado já foi associado a um teste anterior, caso positivo, não permite o cadastro. No caso do cadastro de componentes, cria-se uma entrada na tabela Componentes no banco de dados. Esses também devem ter um ID único, iniciado por um prefixo 'C'.

Cadastrados alguns testes e componentes à base de dados, é possível cadastrar o BDD desejado. A Figura 27 mostra o painel de cadastro de um BDD em construção via diagrama.

The screenshot shows the 'Adicionar BDD via Diagrama' window. The left panel displays the following details for the current node 'T76\_1':

- Nó atual:** T76\_1 (with checkboxes for FAIL and PASS, and a dropdown for 'Teste')
- Código de falha:** P1102
- Fabricante:** DAEWOO
- Motor:** 3.2 i V6 24V LX (220 Hp) Automatic
- Descrição do código:** MAP SNSR - LOW INPUT
- Falta de fiabilidade:** 0.055881202
- Custo total de diagnóstico (min):** 1595.1177
- Teste ID:** T76
- Tempo médio (min):** 5
- Componente:** Cabo de vela (Prob. falha: 0.002)
- Tipo:** Teste Elétrico
- Grandeza:** Tensão Elétrica (Referência: 5 V)
- Ferramenta:** Multímetro
- Descrição:** Teste do cabo realizado com carro ligado.

The right panel shows a decision tree diagram starting with root node P1102. The tree structure is as follows:

```

graph TD
    P1102 --> T2_1
    T2_1 -- FAIL --> T6_1
    T2_1 -- PASS --> T6_2
    T6_1 -- FAIL --> T1_1
    T6_1 -- PASS --> T9_1
    T1_1 -- FAIL --> T45_1
    T1_1 -- PASS --> T5_1
    T6_2 -- PASS --> T76_1
  
```

Figura 27 - Painel de cadastro do BDD

Fonte: Autoria própria.

Para adicionar um novo nó (que codifica um teste já cadastrado), clica-se em um nó existente, especifica-se o teste desejado e escolhe-se a saída (caso disponível) PASS ou FAIL. A ferramenta não permite a criação de um novo nó cujo teste já está associado a algum nó presente no caminho do novo nó até o nó raiz do BDD, o que significaria realizar um mesmo teste mais de uma vez num mesmo procedimento. Note, entretanto, que é possível ter diferentes nós associados

à um mesmo teste no mesmo BDD (adicionou-se um sufixo numérico para diferenciação). Na Figura 27 por exemplo, têm-se os nós T6\_1 e T6\_2, que estão relacionados ao mesmo teste T6.

Ao selecionar um novo teste para ser adicionado ao BDD, suas especificações são exibidas no painel, juntamente com as informações do código de falha, descrição, fabricante e motor (no caso de um DTC específico). Além disso, o custo de diagnóstico e falta de fiabilidade (ver (1) e (2)) também são exibidos e atualizados automaticamente à medida que a construção do BDD vai sendo realizada.

Após a construção do BDD é possível salvá-lo na tabela Arvores do banco de dados. Uma vez salvo, este pode ser utilizado num posterior procedimento de diagnóstico auxiliado pela ferramenta. Além disso, é possível exportar o BDD para XML, formato padrão utilizado na indústria para codificar os BDDs. Um exemplo de BDD gerado pela ferramenta em formato XML é mostrado na Figura 28.

```
<?xml version="1.0" encoding="UTF-8"?>
<Code id="B1232" description="Refrigerant pressure sensor (B12)">
  <Field type="Manufacturer specific" brand="Mercedes-Benz" engine="2.3 i (136 Hp)" />
  <Values cost="45.38289" unreliability="1.66893E-5" />
  <!-- ----->
  <Node Id="T76_1">
    <NodeStructure>
      <Parent parentId="ROOT" />
      <Children Fail="T12_1" Pass="T67_1" />
    </NodeStructure>
    <TestStructure>
      <Component componentId="Cabo de vela" />
      <Measurement type="Teste Elétrico" tool="Multimetro" reference="5" great="V" />
      <Description descr.="Teste do cabo realizado com carro ligado." />
      <Values cost="5" probability="0.002" />
    </TestStructure>
  </Node>
  <!-- ----->
  <Node Id="T12_1">
    <NodeStructure>
      <Parent parentId="T76_1" />
      <Children Fail="" Pass="" />
    </NodeStructure>
    <TestStructure>
      <Component componentId="Sensor de pastilha" />
      <Measurement type="Teste Elétrico" tool="Injetor de Sinal 12V" reference="500" great="µA" />
      <Description descr.="--" />
      <Values cost="40" probability="0.0068" />
    </TestStructure>
  </Node>
  <!-- ----->
  <Node Id="T67_1">
    <NodeStructure>
      <Parent parentId="T76_1" />
      <Children Fail="T5_1" Pass="" />
    </NodeStructure>
    <TestStructure>
      <Component componentId="Bomba de óleo" />
      <Measurement type="Teste Elétrico" tool="Testador para cabo de vela" reference="58" great="" />
      <Description descr.="Teste realizado com ferramenta especial Bosch." />
      <Values cost="12" probability="3.1E-4" />
    </TestStructure>
  </Node>
  <!-- ----->
```

Figura 28 - BDD em formato XML gerado pelo SOAD

Fonte: Autoria própria.

Nesse formato, o BDD é composto por estruturas independentes delimitadas por *tags* (em azul) de início e fim que representam elementos que especificam o BDD de acordo com a posição hierárquica da *tag* na estrutura XML. Cada *tag* individual descreve um conjunto de especificações do BDD através de seus atributos (em vermelho). Por exemplo, as *tags Parent* e *Children* (com os atributos *Pass* e *Fail*) caracterizam a posição relativa do nó no BDD. O conjunto desses nós especifica o BDD por completo.

Para a reprodução do BDD em XML pela ferramenta, utilizou-se a API (*Application Programming Interface*) JDOM (<http://www.jdom.org/>), que é uma solução desenvolvida em JAVA para manipulação e processamento de arquivos XML.

#### 4.2.2 Construção via Equação Booleana

Além da construção do BDD descrita na seção anterior, a ferramenta possibilita também a construção do BDD a partir de uma equação booleana de entrada. Tal funcionalidade é interessante, pois permite a criação do BDD pelo usuário que dispõe apenas da árvore de falhas (que codifica uma equação booleana correspondente) e não do BDD propriamente dito. O painel de cadastro do BDD via equação booleana é mostrado na Figura 29.

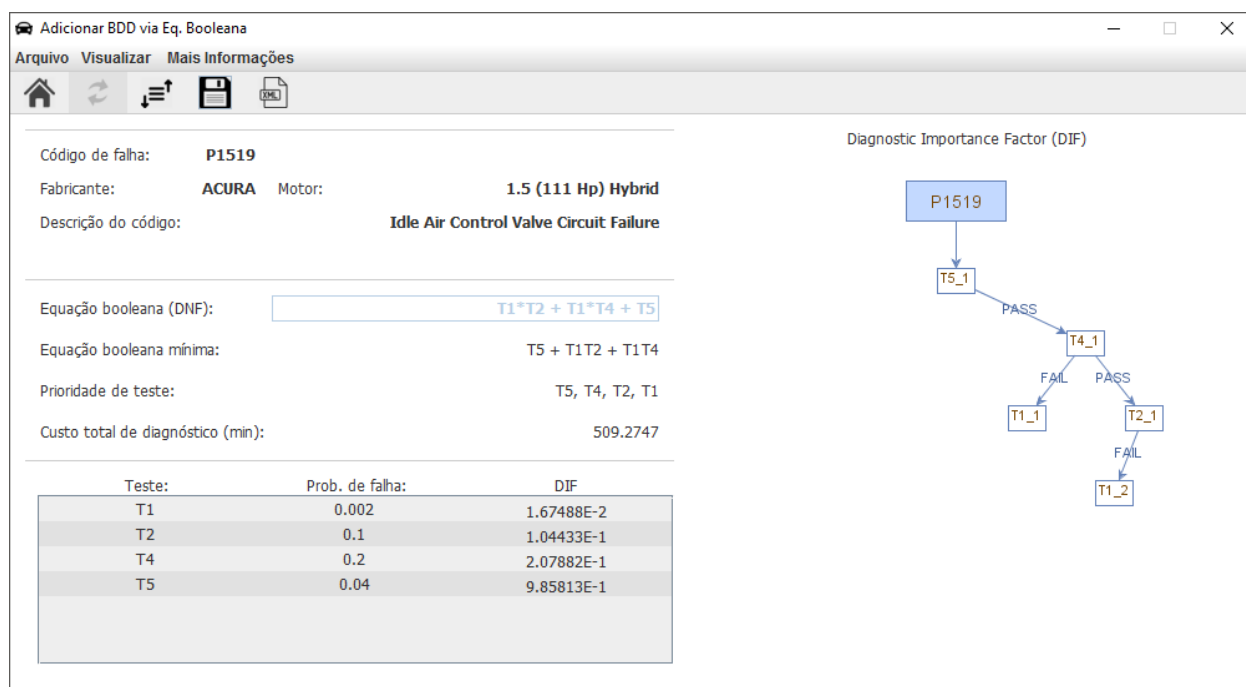


Figura 29 - Painel de cadastro do BDD via equação booleana

Fonte: Autoria própria.

Para a criação do BDD é necessário especificar uma ordenação. Quando carregada a equação booleana, a ferramenta dispõe ao usuário cinco árvores de diagnóstico, relativas às diferentes ordenações obtidas a partir das seguintes métricas:

- DIF (*Diagnostic Importance Factor*);
- CDIF (*Cost of Diagnostic Importance Factor*);
- IG (*Information Gain*);
- $\Pr(T_i)$ ;
- $IG \cdot \Pr(T_i)$ .

A possibilidade de escolha da ordenação traz ao usuário flexibilidade na construção e permite, por exemplo, que o mesmo escolha a BDD com o menor custo de diagnóstico. Na Figura 30 têm-se os BDDs para diferentes ordenações da equação booleana e probabilidades individuais da Figura 29.



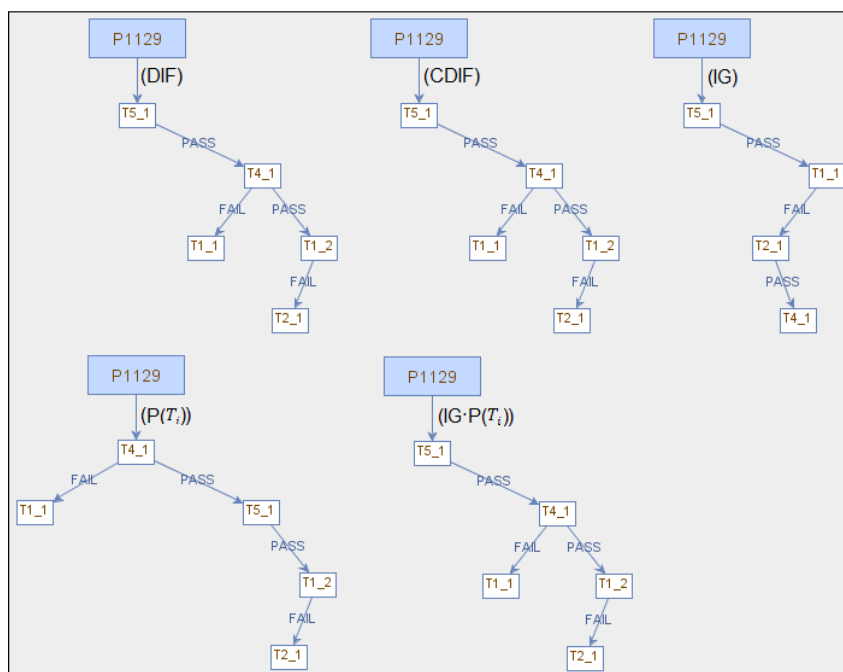


Figura 30 - BDDs para diferentes métricas de ordenação  
**Fonte: Autoria própria.**

### 4.3 MÓDULO PROJETO DO DIAGNÓSTICO

Um conjunto de BDDs existentes por si só não é suficiente para a realização de um procedimento de diagnóstico. É necessário especificar um projeto de diagnóstico, que restringe à realização do procedimento de reconfiguração aos BDDs de um mesmo fabricante, motor e de acordo com as necessidades do usuário. A Figura 31 mostra a janela de criação de projetos de diagnóstico.

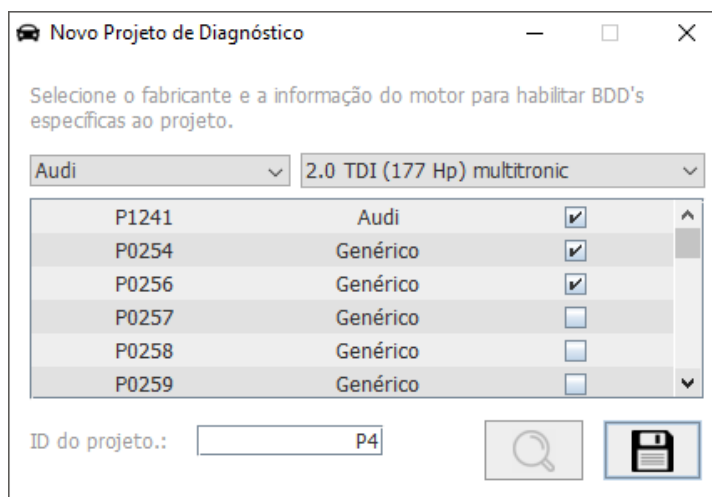


Figura 31 - Janela de criação de projetos de diagnóstico  
**Fonte: Autoria própria.**

No caso da Figura 31, tem-se selecionado o fabricante Audi com o motor 2.0 TDI (177 Hp) multitronic. Dessa forma, para a criação do projeto, só é possível selecionar BDDs associados aos DTCs que satisfaçam essas especificações de fabricante e motor, além dos DTCs genéricos.

Apesar de um BDD poder satisfazer as especificações selecionadas, tem-se a opção de não adicioná-lo ao projeto. Em geral, são selecionados BDDs associados a códigos de falha de um mesmo subsistema do veículo. Como diferentes subsistemas são, de certa forma, isolados entre si, espera-se não encontrar componentes comuns em tais BDDs. Por exemplo, uma falha no sistema de potência dificilmente cause a ativação de uma falha no sistema de carroceria do automóvel. Pode-se restringir, portanto, a abrangência do diagnóstico ao selecionar apenas BDDs de um certo subsistema. Apesar da ferramenta restringir a criação de projetos com BDDs de fabricantes e modelos de motor diferentes, fica à critério do usuário a seleção das BDDs que fazem parte de um mesmo subsistema.

Foi definido um mínimo de três BDDs para a criação de um projeto de diagnóstico. Igualmente aos testes e componentes, cada projeto tem um identificador único, cujo prefixo é letra 'P'. Não é possível criar um projeto de diagnóstico com um identificador já utilizado.

Após a criação bem sucedida de um projeto, uma entrada é adicionada na tabela *Projetos* do banco de dados, cujos atributos correspondem às especificações selecionadas. Na Figura 32, tem-se parte da tabela *Projetos* presente no banco de dados.

	ID	pid	brand	engine	dtcs
4	22	P11	Mercedes-Benz	240 D (115.117) (65 Hp)	B1056, P0254, P0256, P0257, P0258
5	23	P12	Mercedes-Benz	240 D (115.117) (65 Hp)	B1056, P0256, P0011, P0001, P0005
6	24	P15	--	--	P0001, P0005, P0300, P0302, P0301
7	25	P20	Audi	2.0 TDI (177 Hp) multitr...	P1241, P0254, P0256, P0257, P0258, P0259, P0260,...
8	26	P21	Audi	2.0 TDI (177 Hp) multitr...	P0259, P0260, P0261, P0262, P0002
9	27	P22	Audi	2.0 TDI (177 Hp) multitr...	P1241, P0254, P0256, P0257, P0258, P0259, P0260,...
10	29	P1	--	--	P0256, P0257, P0258
11	30	P3	Audi	2.0 TDI (177 Hp) multitr...	P1241, P0254, P0256

Figura 32 - Tabela Projetos existente no banco de dados SQL

Fonte: Autoria própria.

Observe que um projeto é especificado pelo identificador (*id*), pelo fabricante (*brand*), pelo motor (*engine*), e pela lista de BDDs (*dtcs*) presentes. Pode-se criar projetos sem especificação de fabricante e motor; entretanto, apenas BDDs de DTCs genéricos podem ser adicionados.

Além da inserção de uma nova entrada na tabela Projetos, cria-se uma nova tabela no banco de dados com nome Historico\_Px, em que *x* é o identificador numérico do projeto. Essa tabela tem o objetivo de armazenar o histórico de procedimentos daquele projeto em particular. Um exemplo é a tabela mostrada na Figura 33.

	ID	teste	P0256_g	P0257_g	P0258_g
1	1	T1	-1	0	-1
2	2	T12	1	1	-1
3	3	T2	0	0	0
4	4	T3	1	1	1
5	5	T4	1	1	1
6	6	T5	-1	2	-1
7	7	T6	2	2	2

Figura 33 - Tabela de histórico de procedimentos de um projeto de diagnóstico

Fonte: Autoria própria.

Como pode-se observar na Figura 33, as colunas da tabela são códigos de falha (o sufixo 'g' indica DTCs genéricos enquanto que o sufixo 'f' indica DTCs específicos de um fabricante) e as linhas são os testes cadastrados presentes em todos os BDDs associados aos DTCs do projeto.

As células com valores '-1' especificam testes que não estão presentes no BDD associado ao código. Por exemplo, sabe-se que o código P0256 possui um BDD composto apenas pelos testes T12, T2, T3, T4 e T6. Os valores inteiros maiores e iguais a zero indicam o número de vezes que um teste acusou falha dado a ativação de um código de falha particular. Por exemplo, a célula

correspondente ao código P0257 e ao teste T5 indica que, por duas vezes, quando o DTC P0257 esteve ativo, o teste T5 acusou falha.

À medida que os procedimentos são realizados e falhas são encontradas, as células cujos códigos de falha estiveram ativos e os testes acusaram falha, têm seus valores incrementados. Por exemplo, considere um cenário de diagnóstico em que ambos os códigos de falha P0256 e P0257 estiveram ativos devido à uma falha no componente associado ao teste T12, comum entre eles. Caso o diagnóstico tenha sido concluído com sucesso e o componente de T12 tenha sido identificado como causa da ativação dos DTCs, incrementa-se os valores de ambas as células que relacionam este teste aos códigos ativos.

Além disso, à medida que os procedimentos são realizados e o histórico vai sendo alimentado é possível utilizar tais informações para inferir o melhor teste à ser executado, dado o cenário de diagnóstico corrente. É possível ainda adicionar a contribuição do histórico nas métricas de ordenações já apresentadas neste trabalho, definindo prioridades que satisfaçam tanto as métricas quanto o histórico existente. Neste trabalho, optou-se apenas por visualizar o histórico de diagnóstico, deixando o desenvolvimento de estratégias que se utilizam do histórico de procedimentos para trabalhos futuros.

#### 4.4 MÓDULO VISUALIZAÇÃO DE DADOS

Naturalmente, tem-se a necessidade de informar o usuário dos objetos cadastrados pela ferramenta, pois o mesmo não tem acesso às informações diretamente do banco de dados. Dessa forma, a ferramenta dispõe de painéis que apresentam para o usuário os BDDs, os testes, os componentes, os projetos e os históricos de procedimentos cadastrados.

A Figura 34 mostra o painel que lista os projetos existentes. Este painel é análogo à tabela da Figura 32 presente no banco de dados.

Projeto ID:	DDB's cadastradas:
P3	P0008(g) P0011(g) P0001(g) P0301(g) P0238(g) P0022(g) P0003(g)
P7	P0262(g) P0002(g) P0001(g) P0005(g) P0300(g) P0302(g)
P8	P0004(g) P0008(g) P0001(g) P0301(g) P0238(g) P0022(g)
P9	P0011(g) P0001(g) P0005(g) P0300(g) P0302(g) P0238(g) P0022(g)
P12	P0254(g) P0256(g) P0257(g) P0258(g) P0259(g) P0260(g) P0261(g) P0262(g) P0002(g) P0004(g) P0008(g) P0011(g) P00
P15	P0001(g) P0005(g) P0300(g) P0302(g) P0301(g)

Figura 34 - Painel que lista os projetos cadastrados

**Fonte: Autoria própria.**

É possível filtrar a busca selecionando-se o fabricante e o motor. O sufixo 'g' e 'f' explicita o tipo do BDD, se é associado à um DTC genérico ou específico para um fabricante. Ao selecionar um projeto específico, pode-se removê-lo ou visualizar o histórico de procedimentos do projeto. A Figura 35 apresenta o painel do histórico para um exemplo de projeto.

Histórico de falhas do Projeto P12

Arquivo View

Projeto **P12** Fabricante não especificado

	P0254(g)	P0256(g)	P0257(g)	P0258(g)	P0259(g)	P0260(g)	P0261(g)	P0262(g)	P0002(g)	P0004(g)	P0008(g)
T1	0	--	0	--	--	0	0	3	--	--	0
T12	--	0	3	--	--	--	--	--	--	--	0
T2	0	1	2	1	1	0	0	0	1	0	1
T3	0	1	0	1	2	0	2	0	0	1	0
T4	3	0	0	1	0	1	0	0	0	4	0
T5	--	--	2	--	0	--	0	--	--	--	0
T6	--	0	0	0	0	0	2	0	--	0	--
T9	--	--	--	--	--	0	--	--	--	--	0

Valor atual.: 2 Novo valor.:

Figura 35 - Painel do histórico de falhas de um exemplo de projeto

**Fonte: Autoria própria.**

Na Figura 35, tem-se o número de vezes que cada teste registrou uma falha para cada código ativo do projeto. É possível alterar manualmente o conhecimento registrado, selecionando-se o valor atual e alterando-o para um valor específico. A alteração manual é útil, por exemplo, para o cadastro de um histórico de falhas já existente.

Na Figura 36, tem-se o painel que lista os BDDs cadastrados.

DTC:	Descrição:
P0012	A Camshaft Position - Timing Over-Retarded (Bank 1)
P0022	A Camshaft Position - Timing Over-Retarded (Bank 2)
P0026	Intake Valve Control Solenoid Circuit Range/Performance (Bank 1)
P0027	Exhaust Valve Control Solenoid Circuit Range/Performance (Bank 1)
P0049	Turbo/Super Charger Turbine Overspeed
P0050	H02S Heater Control Circuit (Bank 2 Sensor 1)
P0053	H02S Heater Resistance (Bank 1 Sensor 1)
P0059	H02S Heater Resistance (Bank 2 Sensor 1)
P0100	Mass or Volume Air flow Circuit Malfunction
P0104	Mass or Volume Air flow Circuit Intermittent
P0238	Turbocharger Boost Sensor A Circuit High
P0254	Injection Pump Fuel Metering Control "A" High (Cam/Rotor/Injector)
P0256	Injection Pump Fuel Metering Control "B" Malfunction (Cam/Rotor/Injector)
P0257	Injection Pump Fuel Metering Control "B" Range/Performance (Cam/Rotor/Injector)
P0258	Injection Pump Fuel Metering Control "B" Low (Cam/Rotor/Injector)
P0259	Injection lump Fuel Metering Control "B" High (Cam/Rotor/Injector)
P0261	Cylinder 1 Injector Circuit Low
P0262	Cylinder 1 Injector Circuit High

Figura 36 - Painel que lista os BDD's cadastrados

Fonte: Autoria própria.

De forma semelhante à lista de projetos, é possível filtrar os BDDs por tipo de DTC associado (genérico ou específico) e motor. Ao selecionar um BDD cadastrado, é possível removê-lo ou visualizá-lo. Ao remover um BDD, todos os projetos relacionados à este BDD são excluídos. Isso ocorre, pois, como pode-se observar na Figura 32, cada projeto não possui explicitamente as informações de cada BDD adicionada à ele. É necessário acessar a tabela Arvores localizada no banco para obter as informações. Como o registro do BDD foi excluído, tais informações foram perdidas e há a necessidade portanto, da exclusão de cada projeto que o referencia.

A ferramenta solicita a confirmação do usuário para a exclusão de um BDD, como mostra a Figura 37. Janelas de confirmação como a da Figura 37 estão presentes após diversas ações do usuário na ferramenta.

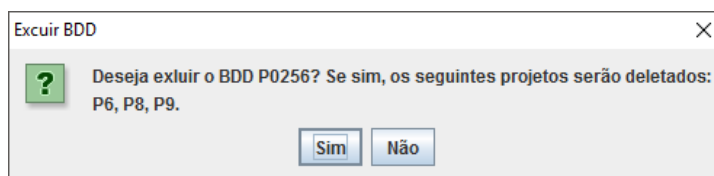


Figura 37 - Janela de confirmação para exclusão de um BDD

Fonte: Autoria própria.

Ao selecionar um BDD para visualização tem-se um painel como o da Figura 38.

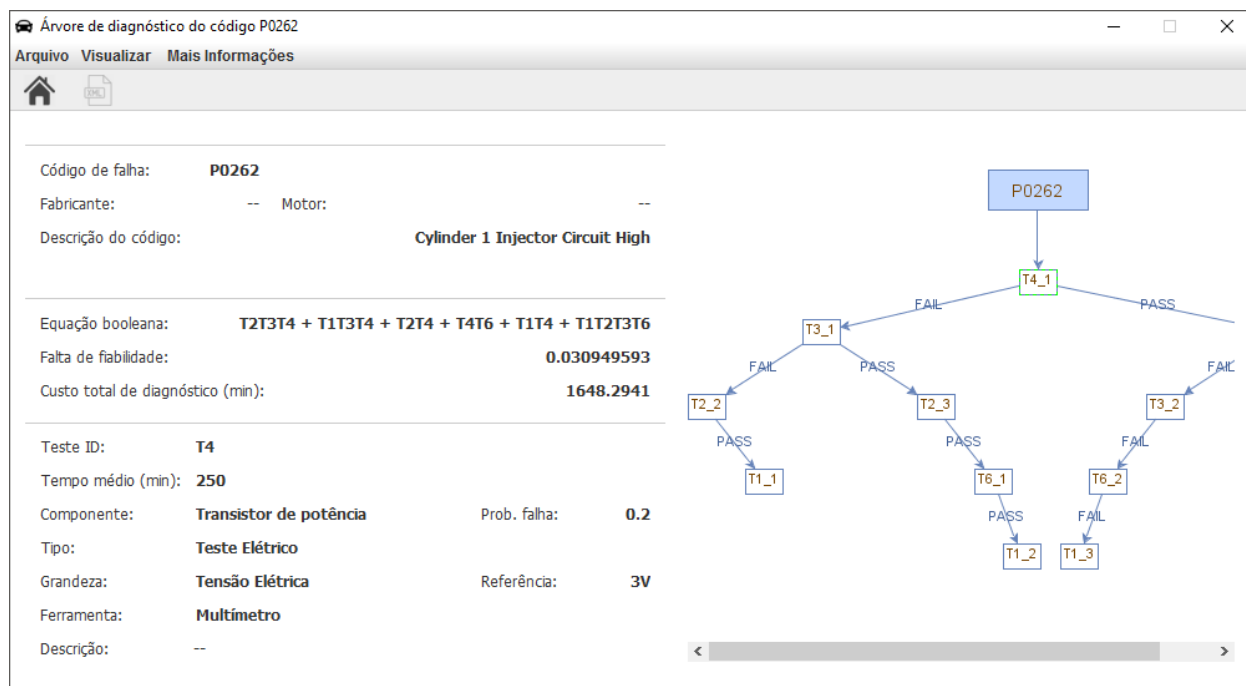


Figura 38 - Painel de visualização do BDD

Fonte: Autoria própria.

No painel da Figura 38 é possível observar algumas características relacionadas ao DTC associado, como o fabricante, motor e descrição do código de falha. Aparecem também características inerentes ao BDD propriamente dito, como a equação booleana mínima, a falta de fiabilidade e o custo do diagnóstico. Além disso, ao selecionar um teste presente no BDD é possível observar suas especificações. Ainda neste painel, a ferramenta permite que o usuário exporte o BDD em formato XML.

O usuário pode observar também a lista de testes cadastrados pela ferramenta; a Figura 39 mostra este painel.





Teste ID:	Componente:	Custo (min):
T8	Amortecedor traseiro	25
T9	Barramento elétrico	65
T11	Cilindro mestre da embreagem	42
T12	Sensor de pastilha	40
T13	Sensor de estacionamento	10
T14	Disco de freio	10
T15	Cilindro mestre da embreagem	30
T16	Kit Embreagem	60
T17	Cilindro hidráulico da correia dentada	60
T18	Braço auxiliar da direção	36
T19	Rolamento da roda dianteira	15
T20	Sensor de temperatura da água	25
T21	Cilindro de freio	70
T22	Tensor de corrente	30
T23	Cruzeta do eixo cardan	25
T24	Vela de ignição	10
T25	Filtro de óleo	10
T26	Bateria	15
T27	Lâmpada dianteira esquerda	20
T28	Cruzeta articulação da coluna de direção	80
T29	Atuador marcha lenta	40

Figura 39 - Painel que lista os testes cadastrados

**Fonte: Autoria própria.**

Nesse painel, é possível adicionar um teste através da janela de cadastro mostrada na Figura 39, visualizar e editar um teste a partir da janela da Figura 40 ou remover um teste existente. Análogo ao caso da exclusão de BDDs, ao remover um teste, é necessário excluir também os objetos que fazem referência a ele. Nesse caso, é necessário excluir os BDDs que o referenciam (diretamente) e os projetos, que o referenciam indiretamente através dos BDDs que possuem o teste excluído.

**Especificações do Teste T13**  
 Teste ID: **T13**  
 Componente: **Sensor de estacionamento**  
 Componente Id: **C7** Prob. de falha: **0.004**  
 Tipo de teste: **Teste Elétrico** Custo do teste (min): **10**  
 Ferramenta: **Multímetro**  
 Grandeza: **Tensão Elétrica** Referência.: **3** **V**  
 Descrição:  
 [ ]  
 [ ] [ + ] [ ]

Figura 40 - Janela de visualização e edição de testes  
**Fonte: Autoria própria.**

Por fim, é possível visualizar os componentes cadastrados pelo usuário, através do painel mostrado na Figura 41.

Comp. ID:	Componente:	Probabilidade de falha:
C1	Cilindro mestre da embreagem	3.0E-4
C2	Atuador de Embreagem	2.0E-4
C3	Cabo da embreagem	0.005
C4	Cilindro auxiliar da embreagem	3.5E-4
C5	Kit Embreagem	0.02
C6	Interruptor da luz de ré	0.002
C7	Sensor de estacionamento	0.004
C8	Bateria	0.0085
C9	Correia do ar condicionado	2.5E-4
C10	Cilindro hidráulico da correia dentada	0.00333
C11	Correia dentada	0.001
C13	Correia do alternador	0.002
C14	Barra de direção	1.0E-5
C15	Bomba de direção hidráulica	2.2E-4
C16	Braço auxiliar da direção	5.5E-4
C17	Caixa de direção mecânica	6.0E-4
C18	Coifa da caixa de direção	4.0E-4
C19	Cruzeta do eixo cardan	1.0E-6
C20	Cabo conector X10	0.1
C21	Cruzeta articulação da coluna de direção	3.3E-5
C22	Rolamento da roda dianteira	3.3E-5

Figura 41 - Painel que lista os componentes cadastrados  
**Fonte: Autoria própria.**

Nesse painel, é possível adicionar um novo componente, visualizar e editar um componente existente, ou remover um componente. Novamente, ao remover um componente, exclui-se automaticamente todos os testes, BDDs e projetos existentes que o referenciam.

## 4.5 MÓDULO DE DIAGNÓSTICO GUIADO

Nesta seção apresenta-se o módulo de diagnóstico guiado do SOAD, que auxilia no procedimento de diagnóstico sugerindo testes a serem realizados sequencialmente pelo técnico. Este é o módulo que justifica este trabalho, pois implementa a estratégia de reconfiguração de testes, codificadas pelas BDDs individuais do cenário de diagnóstico, para sintetizar um BDD, a princípio, mais eficiente, que prioriza testes sobre componentes mais prováveis de falha e com menor custo.

Será apresentado um cenário de diagnóstico para explicitar a utilização da interface pelo usuário no processo do diagnóstico. Em seguida, descreve-se o relatório de resultados gerado pela ferramenta.

### 4.5.1 Cenário de Diagnóstico

Para iniciar um procedimento de diagnóstico assistido pela ferramenta, é necessário especificar um projeto existente. Como mencionado, o projeto delimita os BDDs que podem fazer parte de um mesmo cenário de diagnóstico. Para o estudo de caso apresentado nessa seção, foi criado um projeto com quatro BDDs, associados aos DTCs P0100, P0104, P1100 e P1102 de um veículo Mercedes-Benz com motor Turbo-D 2.5 (122Hp). As sequências de testes codificadas pelos BDDs, bem como as especificações dos testes e componentes, são exemplos ilustrativos. Nas Figuras 42 à 45 pode-se observar os BDDs individuais construídos para este exemplo, além das suas especificações de custo, falta de fiabilidade e equação booleana.

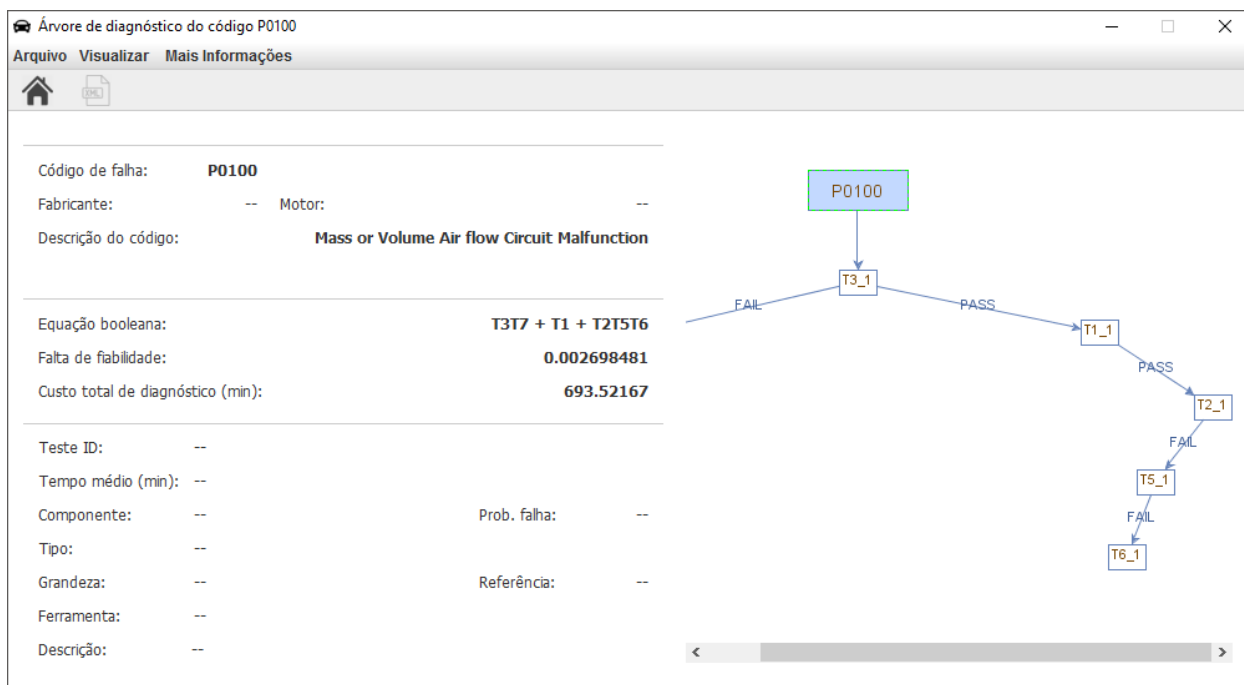


Figura 42 - Especificações do BDD associado ao DTC P0100

Fonte: Autoria própria.

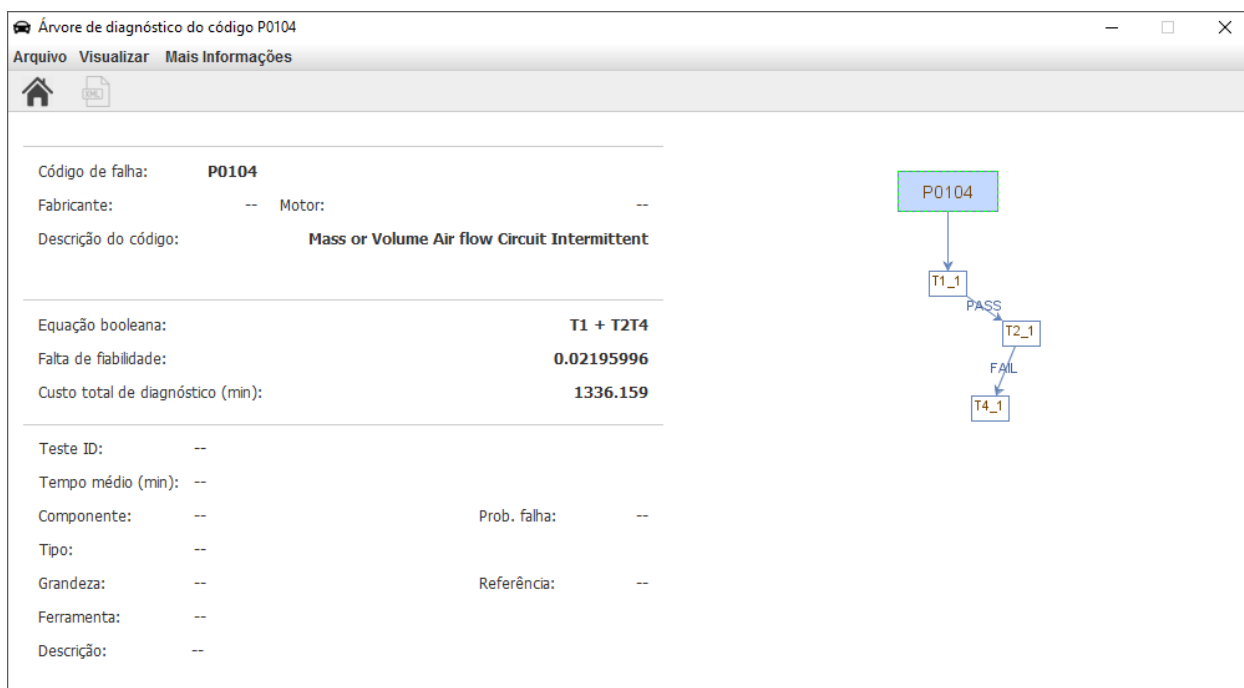


Figura 43 - Especificações do BDD associado ao DTC P0104

Fonte: Autoria própria.

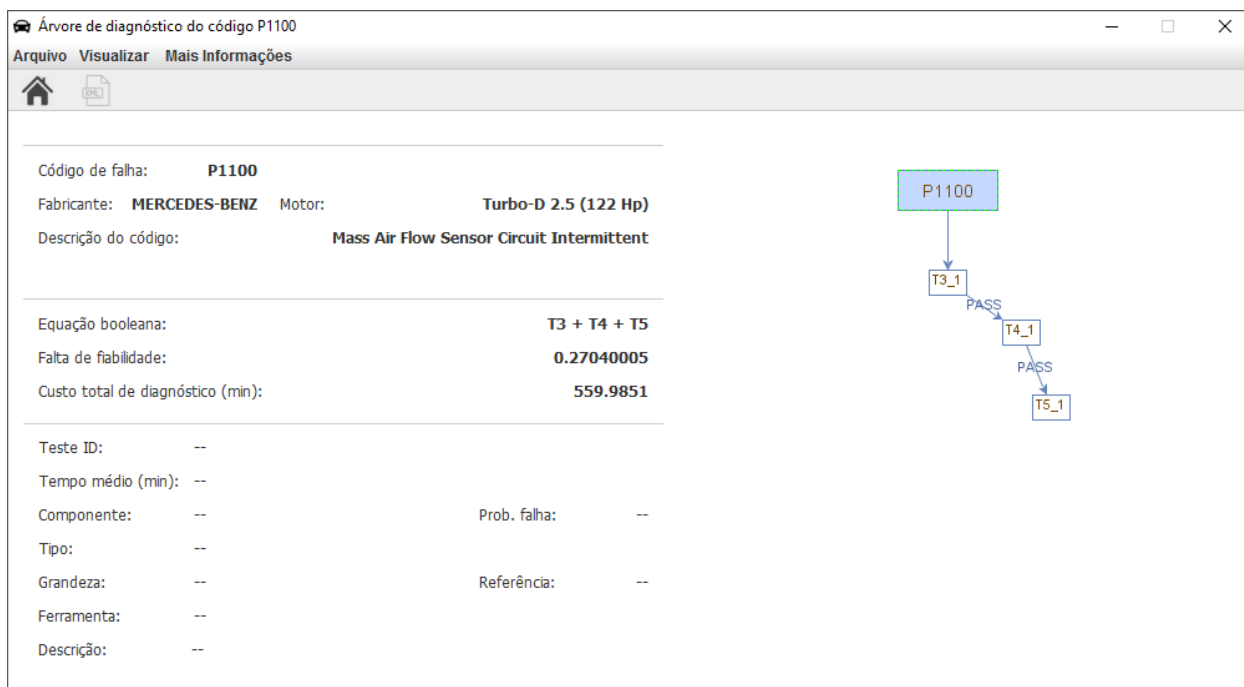


Figura 44 - Especificações do BDD associado ao DTC P1100  
**Fonte: Autoria própria.**

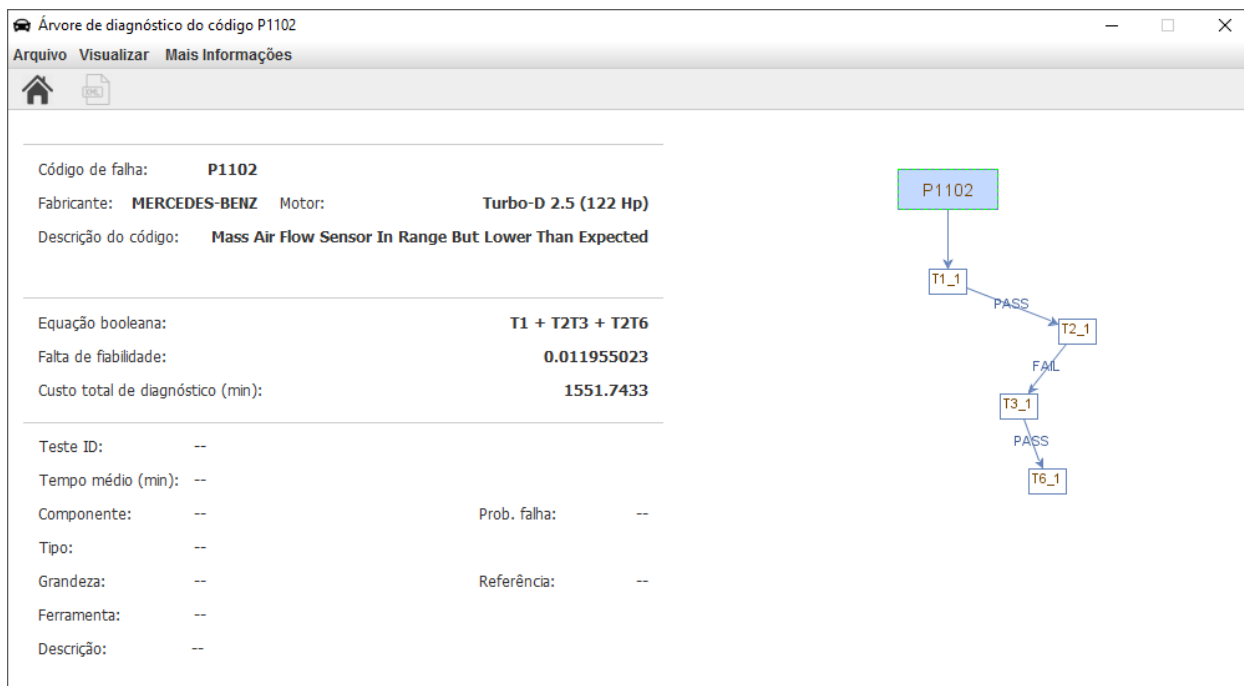


Figura 45 - Especificações do BDD associado ao DTC P1102  
**Fonte: Autoria própria.**

A Figura 46 resume as principais características dos BDDs considerados.

BDD (DTC)	Custo de Diagnóstico	Eq. Booleana
P0100	693.5	$T1 + T3 \cdot T7 + T2 \cdot T5 \cdot T6$
P0104	1336.1	$T1 + T2 \cdot T4$
P1100	560	$T3 + T4 + T5$
P1102	1551.7	$T1 + T2 \cdot T3 + T2 \cdot T6$

Figura 46 - Principais especificações dos BDDs individuais presentes no estudo de caso

**Fonte: Autoria própria.**

A partir das figuras anteriores, pode-se observar que os quatro BDDs do exemplo possuem, ao todo, sete diferentes testes. Os valores de custo e probabilidade de falha associada a cada teste são apresentados na Figura 47.

Teste	Custo	Prob. Falha
T1	200	0.002
T2	1000	0.1
T3	300	0.05
T4	250	0.2
T5	500	0.04
T6	750	0.05
T7	500	0.01

Figura 47 - Especificações dos testes presentes no estudo de caso

**Fonte: Autoria própria.**

Após construídos os BDDs de acordo com as especificações mostradas acima, criou-se um projeto de diagnóstico que os relaciona. Criado o projeto é possível iniciar o procedimento de diagnóstico pela ferramenta. A primeira etapa é selecionar o projeto desejado e especificar o cenário de diagnóstico corrente. Como explicado, o cenário caracteriza quais, dentre os possíveis DTCs adicionados ao projeto, estão ativos. A Figura 48 ilustra essa especificação para um cenário de diagnóstico em que os DTCs P0104, P1100, e P1102 estão ativos, enquanto que o DTC P0100 está inativo.

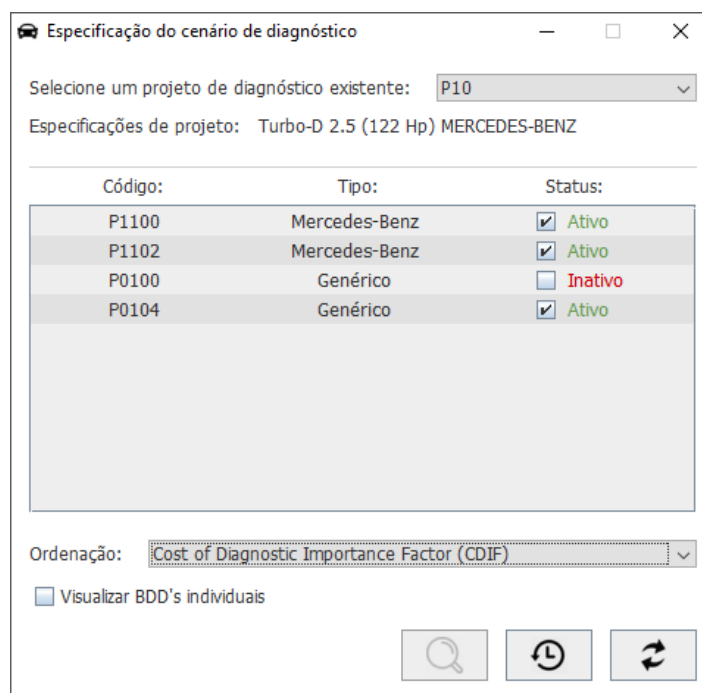


Figura 48 - Cenário de diagnóstico do estudo de caso

**Fonte: Autoria própria.**

Lembrando que os códigos ativos e inativos são dados de entrada para o diagnóstico e obtidos pelo técnico no início do procedimento com *scanners* automotivos conectados à porta OBD do veículo.

Recapitulando, ao receber um conjunto de BDDs e um cenário de diagnóstico, a ferramenta concatena essas estruturas de acordo com o procedimento explicado na seção 3.1. Para a síntese do BDD resultante é possível especificar uma métrica de ordenação, para o caso deste exemplo, utilizou-se a métrica CDIF. É possível ainda escolher a ordenação, cujo BDD possui o menor custo (calculado por (1)).

O BDD resultante da combinação lógica dos BDDs individuais e do cenário de diagnóstico apresentado é mostrado na Figura 49.

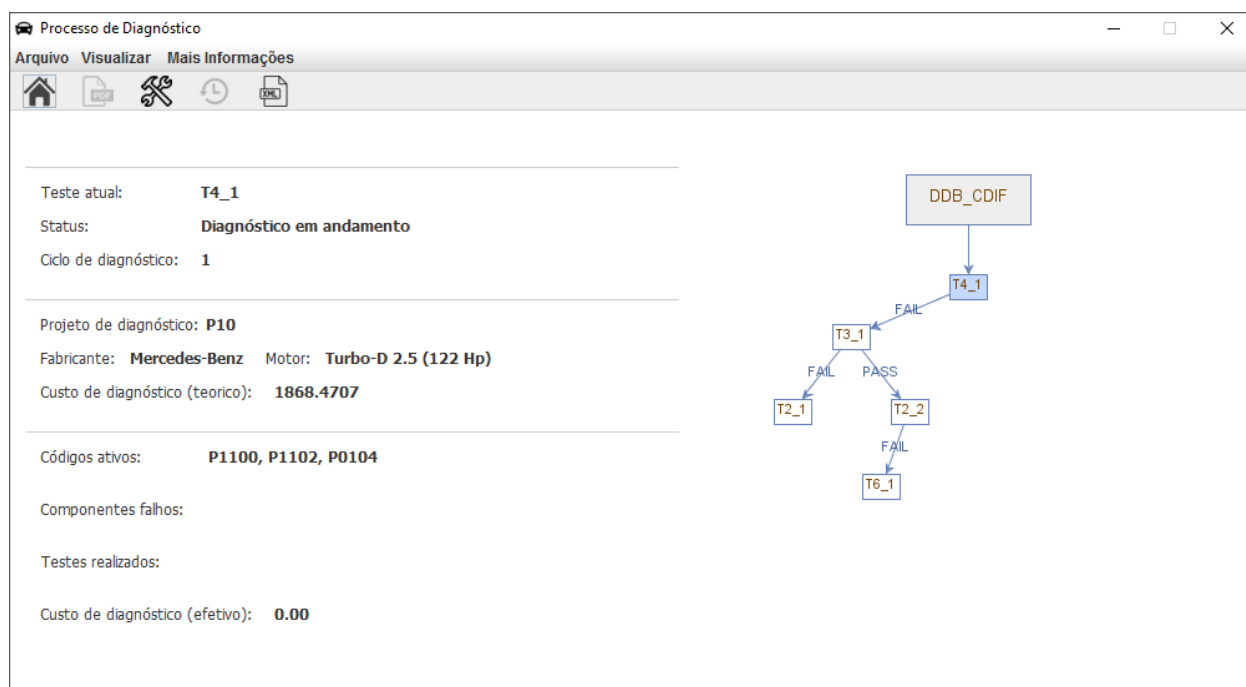


Figura 49 - BDD resultante após a reconfiguração dos testes

Fonte: Autoria própria.

O BDD resultante, substitui o procedimento de testes que considera os BDDs individuais e ordena os testes de modo a minimizar o custo do diagnóstico e maximizar a chance de identificação da falha. Observe que o custo de diagnóstico dessa árvore calculado por (1) é de 1868.5, o que corresponde a 54% da soma dos custos das BDDs dos DTCs ativos (P1100, P1102 e P0104), cujo valor é 3447.9. Na interface, esse custo foi caracterizado como “teórico”. Já o custo efetivo é caracterizado pela soma dos custos dos testes que foram realizados desnecessariamente, ou seja, soma-se os custos dos testes que não acusaram falha e poderiam ser evitados.

À medida que os testes vão sendo realizados, atualizam-se as informações de testes realizados, testes que acusaram falha e custo efetivo de diagnóstico. As figuras seguintes ilustram a atualização. No início do diagnóstico, como nenhum teste foi realizado, tem-se um custo nulo.

Apesar da equação booleana não estar explicitada no painel, tem-se que  $BDD_{CDIF} = T2 \cdot T3 \cdot T4 + T2 \cdot T4 \cdot T6$ . Note que o teste T1 não aparece na Figura 49. Isso ocorre pois a ferramenta inferiu o estado lógico de T1 como PASS a partir do conhecimento do cenário de diagnóstico. Esta inferência foi obtida da observação de que o código P0100 não está ativo. Em outras palavras, uma falha no componente associado à T1 é uma condição suficiente para a ativação do código P0100; como P0100 não foi ativado, o resultado de T1 é necessariamente PASS. Observe que não é possível realizar tal inferência observando-se individualmente os BDDs. Dessa forma, utilizando

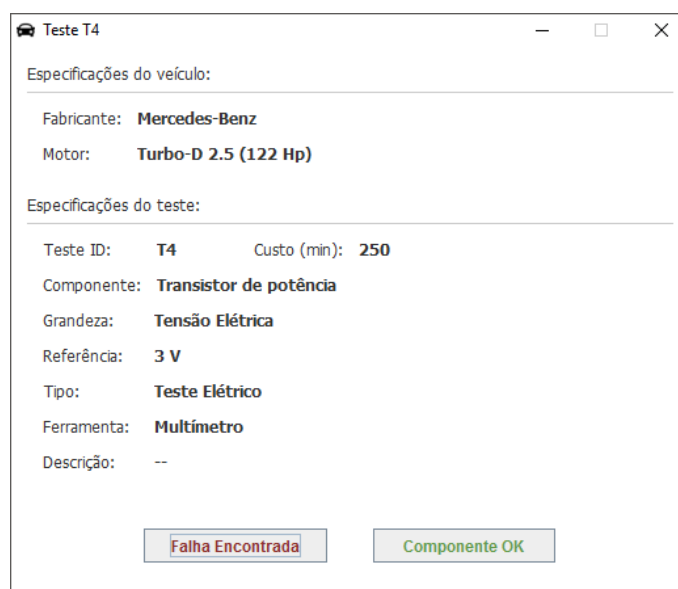


a abordagem convencional, dependendo da ordem em que se escolhe os BDDs para o diagnóstico, o teste T1 poderia ter sido realizado desnecessariamente; enquanto que na abordagem proposta, este teste é excluído do procedimento. A combinação das estruturas individuais ainda permitiu a exclusão dos testes T5 e T7 no BDD resultante.

Ainda no painel da Figura 49, tem-se o número de ciclos de diagnóstico. Ao chegar ao final do BDD (após a realização dos testes), novos BDDs (que satisfazem o cenário de diagnóstico corrente e os resultados dos testes anteriores) podem ser disponibilizados para o usuário, caso algum código de falha permaneça ativo. O número de ciclos mostra a quantidade de BDDs que foram utilizados até o momento. Os detalhes sobre o procedimento de diagnóstico no caso de DTCs que permanecem ativos serão explanados mais à frente.

O BDD da Figura 49 indica que o primeiro teste a ser realizado é o teste T4 (o sufixo apenas diferencia testes comuns no BDD); observe que o teste que deve ser realizado é destacado em azul no painel.

Ao selecionar o teste T4, uma janela como a da Figura 50 é apresentada ao usuário.



The image shows a software window titled "Teste T4" with a standard Windows-style title bar (minimize, maximize, close buttons). The window is divided into two main sections: "Especificações do veículo:" and "Especificações do teste:". The vehicle specifications include "Fabricante: Mercedes-Benz" and "Motor: Turbo-D 2.5 (122 Hp)". The test specifications include "Teste ID: T4", "Custo (min): 250", "Componente: Transistor de potência", "Grandeza: Tensão Elétrica", "Referência: 3 V", "Tipo: Teste Elétrico", "Ferramenta: Multímetro", and "Descrição: --". At the bottom of the window, there are two buttons: "Falha Encontrada" (highlighted in red) and "Componente OK" (highlighted in green).

Especificações do veículo:	
Fabricante:	Mercedes-Benz
Motor:	Turbo-D 2.5 (122 Hp)

Especificações do teste:			
Teste ID:	T4	Custo (min):	250
Componente:	Transistor de potência		
Grandeza:	Tensão Elétrica		
Referência:	3 V		
Tipo:	Teste Elétrico		
Ferramenta:	Multímetro		
Descrição:	--		

Figura 50 - Janela com especificações do teste T4

Fonte: Autoria própria.

Nessa janela tem-se as especificações do teste que deverá ser realizado. Entretanto, como já mencionado, tais especificações não caracterizam o teste por completo. Em geral, é necessário informações adicionais de outras fontes para os ensaios e instrumentação. Após a realização do

teste, o usuário entra com o resultado obtido: “Falha Encontrada” ou “Componente OK”. Supondo que o técnico identifique uma falha no componente associado ao teste T4 e entre com o resultado “Falha Encontrada”, a ferramenta atualiza o painel de acordo, como mostra a Figura 51.

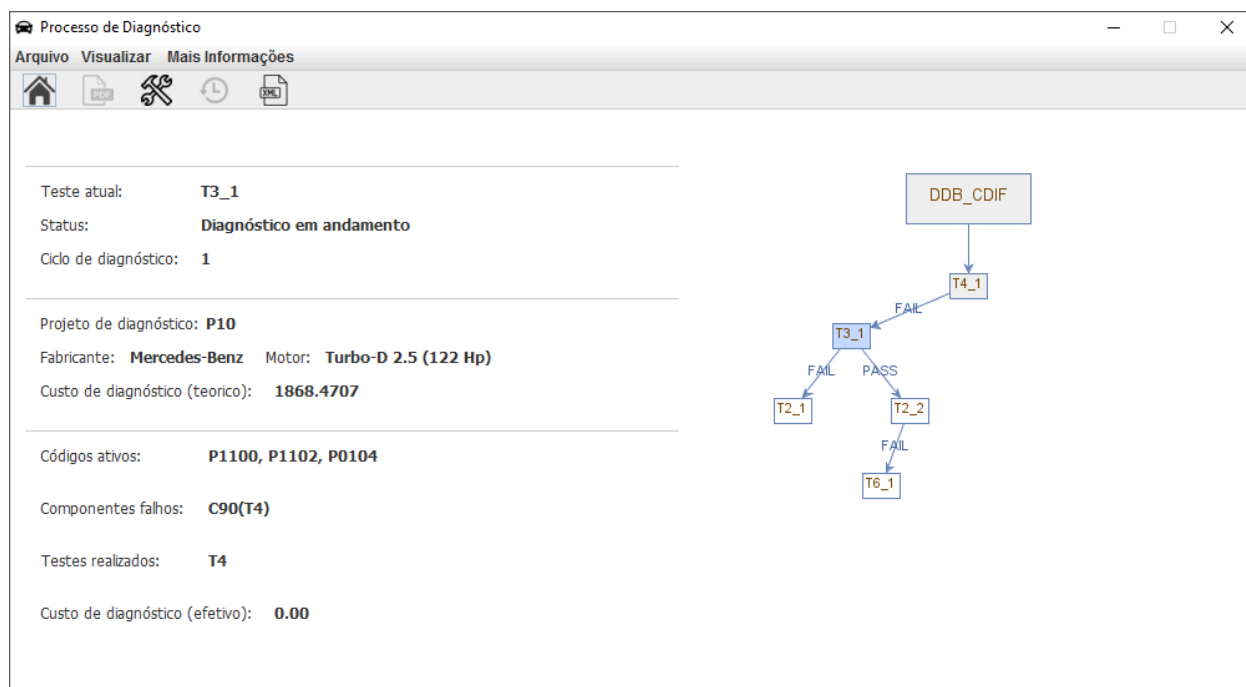


Figura 51 - Painel de diagnóstico após o teste T4

Fonte: **Autoria própria.**

Observe que as informações foram atualizadas corretamente. Particularmente, note que o custo efetivo de diagnóstico não foi alterado, pois o teste T4 acusou a presença de falha.

Continuando o procedimento do diagnóstico, a ferramenta sugere que o técnico realize o teste T3. Uma janela semelhante a da Figura 50, agora com as especificações do teste T3, é mostrada ao usuário. Supondo que o usuário, após a realização deste teste, identifique o componente relacionado como operacional (nenhuma falha foi encontrada), o painel é atualizado com os novos resultados parciais; a Figura 52 mostra o painel atualizado após T3.

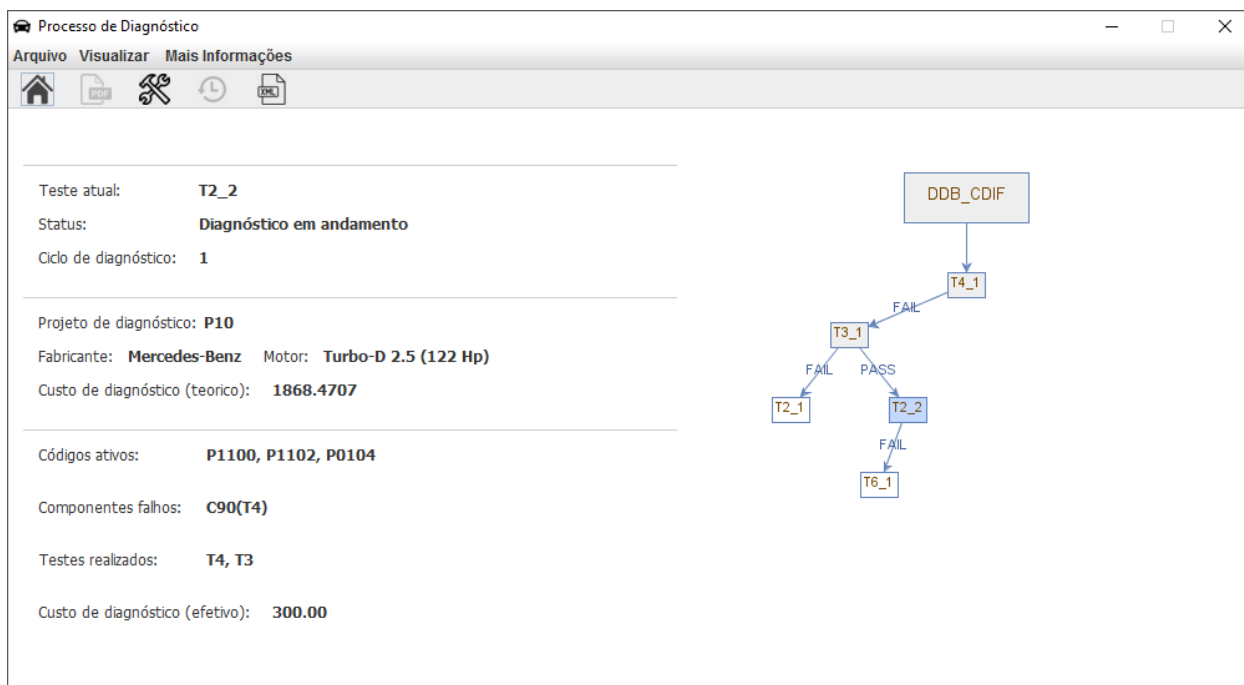


Figura 52 - Painel de diagnóstico após o teste T3

Fonte: Autoria própria.

Continuando com esse mesmo procedimento e supondo que o usuário identifique falha nos componentes dos testes T4 e T6, chega-se ao final do BDD. Como o resultado do último teste T6 foi dado como FAIL, a princípio, o diagnóstico foi concluído. Entretanto, é possível que algum DTC ainda esteja ativo. A janela mostrada na Figura 53 solicita que o usuário informe a permanência de algum DTC ativo.

Código:	Tipo:	Status:
P1100	Mercedes-Benz	<input type="checkbox"/> Inativo
P1102	Mercedes-Benz	<input type="checkbox"/> Inativo
P0104	Genérico	<input type="checkbox"/> Inativo

Figura 53 - Janela que solicita a informação dos códigos remanescentes

Fonte: Autoria própria.

Após a seleção dos códigos remanescentes pelo usuário, tem-se três resultados possíveis. No primeiro caso, o usuário informa que nenhum DTC permaneceu ativo. Isso significa que as falhas encontradas explicaram completamente a ativação dos códigos. Nesse caso, o diagnóstico é finalizado com sucesso. No segundo caso, o usuário informa a permanência de um ou mais códigos ativos, e a ferramenta encontra uma inconsistência entre a ativação dos códigos e os resultados obtidos pelos testes anteriores.

Para explicar o segundo caso, será abordado um exemplo. No estudo de caso abordado até aqui, quatro testes foram realizados: T4, T3, T2 e T6. Foram encontradas falhas nos componentes de T4, T2 e T6. Observe na Figura 46, que a equação booleana de P0104 é  $P0104 = T1 + T2T4$ . Sabe-se, a partir do conhecimento do código inativo P0100, que o componente de T1 é operacional. Logo, apenas a combinação de códigos dos componentes de T2 e T4 pode explicar o aparecimento do código. Como já foi identificada a falha em ambos os componentes associados à T2 e T4, o DTC P0104 deveria estar inativo. Dessa forma, caso o usuário indique a permanência do código P0104 como ativo, não é possível explicar esse estado com base na relação lógica existente entre o código e os testes (baseada nas informações cadastradas anteriormente). Neste caso, o diagnóstico é definido como inconsistente e concluído com *status* “No Fault Found (NFF)”.

O terceiro e último caso está relacionado com a persistência de algum código ativo que respeita os resultados dos testes obtidos anteriormente e o novo cenário de diagnóstico (tem-se um novo conjunto de DTCs ativos e inativos). Um novo BDD mínimo é apresentado ao usuário, caracterizando um novo ciclo. Naturalmente, as informações de testes realizados, falhas encontradas e custo são mantidas de um ciclo para outro.

É possível obter um novo ciclo de diagnóstico ao chegar ao final do BDD através de uma saída do tipo PASS. Considere, por exemplo, o BDD da Figura 54. Nesse caso, tem-se um BDD cuja equação booleana é  $P0025 = T1 \cdot T2 + T3$ . Caso os seguintes resultados dos testes sejam obtidos: T1 – FAIL, T2 – PASS, chega-se ao final do BDD através de uma conclusão do tipo PASS. Entretanto, o componente associado à T3 ainda pode explicar a ativação do código. Num caso como este, a ferramenta disponibiliza automaticamente o teste T3 para ser realizado em um novo ciclo de diagnóstico.

Observe que existem casos em que o final do BDD pela conclusão PASS podem resultar em inconsistência. Por exemplo, caso os seguintes resultados sejam obtidos: T1 – PASS, T3 –

PASS, não existe uma explicação lógica para o aparecimento de P0025 independente do estado de T2. A conclusão do diagnóstico é, nesse caso, NFF.

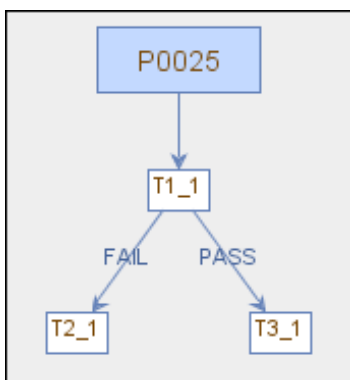


Figura 54 - BDD combinada com nova ordenação  
**Fonte: Autoria própria.**

Continuando com o estudo de caso e supondo que o usuário indique que nenhum código permaneceu ativo, obtém-se como resultado o painel da Figura 55.

Processo de Diagnóstico

Arquivo Visualizar Mais Informações

Teste atual: --

Status: **Fault Identified (FI)**

Ciclo de diagnóstico: 1

Projeto de diagnóstico: P10

Fabricante: Mercedes-Benz Motor: Turbo-D 2.5 (122 Hp)

Custo de diagnóstico (teórico): 1868.4707

Códigos ativos: P1100, P1102, P0104

Componentes falhos: C90(T4), C88(T2), C92(T6)

Testes realizados: T4, T3, T2, T6

Custo de diagnóstico (efetivo): 300.00

Figura 55 - Painel com resultados do diagnóstico finalizado  
**Fonte: Autoria própria.**

Observe que no painel da Figura 55 tem-se o *status* “Fault Identified (FI)” (“Falha Encontrada”). A lista de códigos ativos, componentes falhos, testes realizados e o valor do custo

efetivo de diagnóstico são apresentados corretamente. Após a conclusão do diagnóstico, independentemente do *status*, é possível gerar um relatório de diagnóstico em formato PDF (*Portable Document Format*). A subseção seguinte apresenta o relatório para o estudo de caso considerado.

#### 4.5.2 Relatórios

Como mencionado, ao final do procedimento de diagnóstico guiado pela ferramenta, pode-se gerar um relatório com os principais resultados obtidos no formato PDF. Primeiramente, apresentam-se as especificações do projeto de diagnóstico, o número de ciclos realizados e a conclusão (FI ou NFF) obtida. Em seguida, tem-se a lista dos códigos ativos e inativos, bem como os seus significados, caracterizando o cenário de diagnóstico.

Após a descrição do cenário, tem-se a comparação teórica entre a estratégia que considera a árvore combinada e a estratégia que utiliza as árvores individuais. Como no caso das árvores individuais, é necessário especificar uma ordem de prioridade entre os BDDs, adotou-se a ordenação em ordem crescente de custo. Ou seja, prioriza-se a análise dos BDDs com menor custo. Para o caso apresentado, tem-se a ordem: P1100, P0100, P0104, P1102. Especificamente, apresenta-se o tipo de ordenação adotado na estratégia de combinação, o custo de ambas as metodologias (calculado por (1)) e o número total de testes presentes nos BDDs. Caso mais de um ciclo de diagnóstico tenha sido realizado com a árvore combinada, somam-se os custos e testes presentes nos BDDs de cada ciclo.

Por fim, apresenta-se a comparação efetiva entre as duas abordagens. Exibe-se aqui, para ambas as abordagens: os testes realizados até a conclusão final do diagnóstico, os testes que acusaram falha, os componentes identificados como falhos e o custo efetivo (medido) de diagnóstico. Além disso, apresenta-se a redução do número de testes e a redução do custo de diagnóstico efetivo ao utilizar a estratégia combinada ao invés da abordagem convencional.

Os seguintes resultados foram obtidos para o exemplo considerado na seção 4.5.1:

**Projeto de diagnóstico: P10**

**Fabricante: Mercedes-Benz**

**Motor: Turbo-D 2.5 (122 Hp)**

**Ciclos de diagnóstico: 1**

**Conclusão: Fault Identified (FI)**

**Cenário de diagnóstico**

**Códigos ativos:**

P1100 (Específico): Mass Air Flow Sensor Circuit Intermittent

P1102 (Específico): Mass Air Flow Sensor In Range But Lower Than Expected

P0104 (Genérico): Mass or Volume Air flow Circuit Intermittent

**Códigos inativos:**

P0100 (Genérico): Mass or Volume Air flow Circuit Malfunction

**BDDs Combinadas vs BDDs Individuais (comparação teórica)**

Ordenação: Cost of Diagnostic Importance Factor (CDIF)

Custo de diagnóstico das árvores combinadas: 1868.47

Custo de diagnóstico das árvores individuais: 3447.89

Número de testes presentes nas árvores combinadas: 5

Número de testes presentes nas árvores individuais: 10

**BDDs Combinadas vs BDDs Individuais (comparação efetiva)**

Testes realizados usando as árvores combinadas: T2, T3, T4, T6

Testes realizados usando as árvores individuais: T1, T2, T3, T4, T6

Testes que acusaram falha usando as árvores combinadas: T4, T2, T6

Testes que acusaram falha usando as árvores individuais: T2, T4, T6

Componentes falhos encontrados usando as árvores combinadas: C90, C88, C92

Componentes falhos encontrados usando as árvores individuais: C88, C90, C92

Custo de diagnóstico efetivo usando as árvores individuais: 500

Custo de diagnóstico efetivo usando as árvores combinadas: 300

Redução do número de testes usando as árvores combinadas: 1

Redução do custo de diagnóstico efetiva usando as árvores combinadas: 200.00

Pode-se observar pelos resultados obtidos que houve uma redução de 200 minutos (40%) do custo efetivo do diagnóstico (os valores em minutos são apenas ilustrativos e não representam o tempo real de um procedimento). Note que na estratégia convencional realizou-se os testes T1 e T3 desnecessariamente (ambos estavam operacionais), enquanto que na abordagem combinada, apenas o T3 foi dispensável (os testes restantes foram necessários para identificar as falhas).

## 5. CONCLUSÃO

Esse trabalho de conclusão do curso de Engenharia de Controle e Automação apresenta o desenvolvimento de um *software* de auxílio ao diagnóstico de múltiplas falhas com aplicações na indústria automotiva. Este *software* apresenta funcionalidades para cadastrar dados de fabricantes, inserir árvores de diagnóstico e guiar o diagnóstico através de testes em componentes. Possibilita ainda o armazenamento de dados históricos. A estratégia proposta de diagnóstico guiado é representada por uma combinação das árvores individuais de diagnóstico dos códigos de falha ativos e inativos. Além disso, o *software* fornece um relatório detalhado do processo de diagnóstico, no qual se destaca a redução no número de testes necessários e do custo efetivo para o diagnóstico do cenário considerado, devido à estratégia proposta.

A estratégia proposta está implementada no módulo de diagnóstico guiado, tendo gerado duas publicações científicas (OLIVEIRA; MORI; LÜDERS, 2013) e (OLIVEIRA; LÜDERS, 2016).

A estratégia propõe a reconfiguração das sequências individuais utilizando algoritmos e métricas encontradas na literatura. Resultados preliminares indicam uma redução do custo efetivo e do número de testes necessários para o diagnóstico em comparação com a abordagem convencional, que utiliza os BDDs de maneira independente. Por exemplo, no exemplo apresentado na seção de Resultados, evitou-se um teste desnecessário utilizando a estratégia combinada, o que significou uma redução de 40% do custo efetivo de diagnóstico.

O sistema foi implementado em linguagem Java e possui conexão com uma base de dados SQL e acesso à informações via *web*. Utilizou-se de conhecimentos de programação adquiridos no curso. Entretanto, devido à falta de profundidade na área de computação no curso de Engenharia de Controle e Automação, foi necessária a pesquisa por conta própria para a obtenção de conhecimentos mais profundos de programação orientada à objetos e desenvolvimento de algoritmos. Além disso, este trabalho utilizou de conhecimentos adquiridos na iniciação científica para seu desenvolvimento.

O *software* desenvolvido é um teste de conceito, no qual é possível experimentar a estratégia proposta em diversos cenários, de forma bem similar ao que ocorre nas ferramentas de diagnóstico guiado. Entretanto, a interface é simplificada. Não existem instruções para a realização



dos testes, nem informações de manual dos componentes a serem testados. Neste aspecto, o *software* se destina mais aos engenheiros de diagnóstico, responsáveis pela autorização de ferramentas comerciais de diagnóstico, ao invés dos técnicos especializados que realizam o diagnóstico nos veículos.

Futuramente, esta ferramenta poderá ser incorporada a pacotes comerciais de diagnóstico guiado ou mesmo portada para um dispositivo móvel com conexão a um *scanner* comercial da indústria automotiva. Isso poderá incentivar um maior uso de ferramentas de diagnóstico em conjunto com a criação de uma base de dados com o histórico dos diagnósticos realizados, contribuindo para a ajuste das probabilidades de falha dos componentes.

## REFERÊNCIAS

AGCOAUTO. Gas Caps and Check Engine Lights, The Evap System, 2016. Disponível em: <[http://www.agcoauto.com/content/news/p2\\_articleid/151](http://www.agcoauto.com/content/news/p2_articleid/151)>. Acesso em: 29 de maio de 2016.

AUTO DATA. Wiki Automotive Catalog, 2016. Disponível em: <<http://www.auto-data.net>>. Acesso em: 05 de novembro de 2016.

AMAZON. Actron CP9575 AutoScanner, 2016. Disponível em: <<http://www.amazon.com/Actron-CP9575-AutoScanner/dp/B003IOKPLO>>. Acesso em: 28 de maio de 2016.

AMOR-SEGAN, M. L.; MCMURRAN, R.; DHADYALLA, G.; Jones, R. P. **Towards the Self-Healing Vehicle**. In: 3rd IET Automotive Electronics Conference, 2007. Warwick, UK.

ANDREWS, J. D.; REMENYTE, Rasa. Fault tree conversion to binary decision diagrams. In: **Proceedings of the 23rd International System Safety Conference**, 2005.

ASSAF, T.; DUGAN, J. B. Diagnosis based on reliability analysis using monitors and sensors. **Reliability Engineering & System Safety**, v. 93, n. 4, p. 509-521, 2008.

ASSAF, Tariq; DUGAN, Joanne Bechta. Build better diagnostic decision trees. **Instrumentation & Measurement Magazine, IEEE**, v. 8, n. 3, p. 48-53, 2005.

BERGER, I. Can you trust your car? *IEEE Spectrum*, April, pp. 40-45, 2002.

DUTUIT, Yves; RAUZY, Antoine. Importance factors of coherent systems: A review. **Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability**, v. 228, n. 3, p. 313-323, 2014.

FRANK, H.; SCHMIDTS, U. **Vehicle Diagnostics - The whole Story**. In: VECTOR CONGRESS, 2006. Stuttgart, GER.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **ISO 15031-6**: Road Vehicles – Communication between vehicle and external equipment for emissions-related diagnostics – Diagnostic trouble code definitions, 2005.

JUST ANSWER. **Fórum**, 2011. Disponível em: <<http://www.justanswer.com/ford/3sfwp-2002-f350-superduty-7-31-powerstroke-diesel-does.html>> Acesso em: 29 de maio de 2016.

LANIGAN, Patrick E. et al. Diagnosis in automotive systems: A survey. **Parallel Data Laboratory Carnegie Mellon University, Pittsburgh**, p. 11-110, 2011.

MITCHELL, Thomas M. Machine learning. New York, 1997.

MORI, Fernando Maruyama. **Uma metodologia de desenvolvimento de diagnóstico guiado para veículos automotivos**. Dissertação de Mestrado, UTFPR, 2014.

NC ENVIRONMENTAL QUALITY. **Emissions Inspection Results Information**, 2007. Disponível em <<https://deq.nc.gov/about/divisions/air-quality/motor-vehicles-air-quality/inspection-maintenance-program/emissions-inspection-results-information>>. Acessado em 08 de junho de 2016.

NELSON, Victor P. Fault-tolerant computing: Fundamental concepts. **Computer**, v. 23, n. 7, p. 19-25, 1990.

NISSAN HELP. **OBDII Data Link Connector (DLC) Location**, 2016. Disponível em: <[http://nissanhelpp.com/diy/maxima/projects/images/2005\\_nissan\\_maxima\\_obd\\_connector.jpg](http://nissanhelpp.com/diy/maxima/projects/images/2005_nissan_maxima_obd_connector.jpg)>. Acesso em: 18 de março de 2016.

OBDII CODES. OBDII/EOBD Diagnostics Trouble Codes Definition, 2016. Disponível em: <<http://obdii.pro/>>. Acesso em: 05 de novembro de 2016.

OBD-CODES. **P0443 - Evaporative Emission Control System Purge Control Valve Circuit**, 2016. Disponível em: <<http://www.obd-codes.com/p0443>>. Acesso em: 04 de junho de 2016.

OGAWA, Takashi; MOROZUMI, Hiroki. **Diagnostics trends for automotive electronic systems**. SAE Technical Paper, 2002.

OLIVEIRA, Markos F. B. G.; MORI, Fernando. M.; LÜDERS, Ricardo. Uma estratégia de fusão de árvores de diagnóstico para identificação de falhas em veículos automotores. In: **Simpósio Brasileiro de Automação Inteligente**, 2013, Fortaleza. Anais do XI Simpósio Brasileiro de Automação Inteligente, 2013. v.1. p.1-6.

OLIVEIRA, M. F. B. G. ; LUDERS, R . Combining multiple diagnostic trouble codes into a single decision tree. In: **8th IFAC Symposium on Advances in Automotive Control**, 2016, Norrköping. IFAC-PapersOnLine. Holanda: Elsevier, 2016. v. 49. p. 555-561.

RAUZY, Antoine. New algorithms for fault trees analysis. **Reliability Engineering & System Safety**, v. 40, n. 3, p. 203-211, 1993.

REPARADOR FIAT. **Sensor de posição do pedal do acelerador Fiat TT**, 2016. Disponível em: <<https://goo.gl/JcXy3f>>. Acesso em: 13 de maio de 2016.

RIZZONI, Giorgio; ONORI, Simona; RUBAGOTTI, Matteo. Diagnosis and prognosis of automotive systems: motivations, history and some results. In: **Proceedings of the 7th IFAC Symposium on fault detection, supervision and safety of technical processes (SAFEPROCESS'09)**, 2009.

SÖDERHOLM, Peter. A system view of the No Fault Found (NFF) phenomenon. **Reliability Engineering & System Safety**, v. 92, n. 1, p. 1-14, 2007.

SUWATTHIKUL, J. **Fault detection and diagnosis for in-vehicle networks**. In W. Zhang (ed.), *Fault Detection*, chapter 13, 283-306. InTech, 2010.

THOMAS, Dawn A.; AYERS, Ken; PECHT, Michael. The “trouble not identified” phenomenon in automotive electronics. **Microelectronics reliability**, v. 42, n. 4, p. 641-651, 2002.

VESELY, W. E. et al. *Fault Tree Handbook with Aerospace Applications*, 2002. **NASA Office of Safety and Mission Assurance**.

WARRANTY WEEK. **Automotive OEM Warranty Report**, 2016. Disponível em: <<http://www.warrantyweek.com/archive/ww20160331.html>>. Acesso em: 29 de maio de 2016.

YACOUB, Sherif M.; AMMAR, Hany H. A methodology for architecture-level reliability risk analysis. **Software Engineering, IEEE Transactions on**, v. 28, n. 6, p. 529-547, 2002.