

**UNIVERSIDADE SANTA CECÍLIA**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA**  
**MESTRADO EM ENGENHARIA MECÂNICA**

**CLAUDIO SOUZA NUNES**

**SISTEMA *ONLINE* FOCADO EM ANÁLISE PARA CONSISTENTE APLICADA A  
PROJETOS DE AUTOMAÇÃO**

**SANTOS/SP**

**2015**

**CLAUDIO SOUZA NUNES**

**SISTEMA *ONLINE* FOCADO EM ANÁLISE PARACONSISTENTE APLICADA A  
PROJETOS DE AUTOMAÇÃO**

Dissertação apresentada à Universidade Santa Cecília como parte dos requisitos para obtenção de título de Mestre no Programa de Pós-Graduação em Engenharia Mecânica, sob orientação da Profa. Me. Dorotéia Vilanova Garcia.

**SANTOS/SP**

**2015**

Autorizo a reprodução parcial ou total deste trabalho, por qualquer que seja o processo, exclusivamente para fins acadêmicos e científicos.

Nunes, Claudio Souza.

Sistema *Online* Focado em Análise Paraconsistente Aplicado a Projetos de Automação / Claudio Souza Nunes

- 2015.

53p.

Orientadora: Dorotéia Vilanova Garcia.

Dissertação (Mestrado) -- Universidade Santa Cecília, Programa de Pós-Graduação em Engenharia Mecânica, Santos, SP, 2015.

1. lógica clássica. 2.lógica paraconsistente anotada 3. biblioteca javascript 4. ferramenta *online*. 5.automação 6.robotica I. Garcia, Dorotéia Vilanova II. Sistema *Online* Focado em Análise Paraconsistente Aplicado a Projetos de Automação.

Dedico este trabalho a todos os mestres que  
contribuíram com a minha formação tanto  
acadêmica como indivíduo.

## **AGRADECIMENTOS**

A todos os inúmeros e iluminados mestres com os quais tive o prazer de conviver durante minha vida. Cada um deles tem uma parcela na construção do meu conhecimento e na formação do meu caráter.

Aos professores Dr. Maurício Conceição Mário e Dr. João Inácio da Silva Filho por me apresentarem o universo da lógica Paraconsistente e por seu apoio no decorrer do curso de mestrado.

A minha orientadora, a querida mestra Dorotéa Vilanova Garcia pelo incentivo, orientação, dedicação e colaboração não restritas apenas a este trabalho, mas a toda minha vida acadêmica.

Aos meus colegas de mestrado e amigos Carlos Jair Coletto, Luiz Antonio Ferraro Mathias, Fernando José Cesílio Branquinho pela amizade, companheirismo e auxílio-mútuo.

.

## RESUMO

O presente trabalho destaca a importância da lógica Paraconsistente como alternativa à lógica clássica em situações onde há necessidade de lidar com informações contraditórias ou incompletas. Propõe a criação e disponibilização sem ônus, de uma biblioteca reutilizável, implementada na linguagem *Javascript*, capaz de permitir a criação de um reticulado de 4 vértices associado à lógica Paraconsistente Anotada e demonstrar o tratamento matemático relativo aos algoritmos extraídos dos fundamentos da lógica Paraconsistente anotada de dois valores (LPA2v). A biblioteca criada deverá servir de apoio ao desenvolvimento de projetos de automação e controle, permitindo a criação de simulações interativas a serem disponibilizadas *online* na *internet* por terceiros. Propõe também a criação de uma ferramenta *online* interativa para assimilação dos princípios da LPA2v que demonstre o uso das capacidades da biblioteca criada e que sirva de apoio instrucional. Tal ferramenta *online* também incorporou em uma de suas páginas um simulador computacional demonstrando as capacidades da biblioteca em um controlador lógico paraconsistente utilizado em um robô móvel autônomo denominado Emmy.

**Palavras-chave:** lógica clássica. lógica Paraconsistente anotada. biblioteca *javascript*. ferramenta *online*. Automação. robótica.

## **ABSTRACT**

This work highlights the importance of Paraconsistent logic as an alternative to classical logic in situations where there is a need to deal with conflicting or incomplete information. Proposes the creation and delivery without charge, a reusable library, implemented in JavaScript language, able to allow the creation of lattices of Paraconsistent Logic and demonstrate the mathematical treatment on the algorithms extracted from the logic of fundamentals Paraconsistent Annotated with two Values (PAL2v). The library created should serve to support the development of automation and control projects, enabling the creation of interactive simulations to become available online on the Internet by third parties. It also proposes the creation of an interactive online tool for assimilation of the principles of PAL2v that demonstrates the use of the library and its capabilities created to serve as instructional support. This online tool also incorporated, in one of its pages, a computer simulator demonstrating the library's capabilities in a paraconsistent logic controller used in an autonomous mobile robot named Emmy.

**Keywords:** classical logic. paraconsistent annotated logic. javascript library. online tool. Automation. Robotics.

## LISTA DE ILUSTRAÇÕES

FIGURA 1 – RETICULADO ASSOCIADO A LÓGICA PARACONSISTENTE ANOTADA - ESTADOS LÓGICOS - GERADO EM WWW.LPA2V.COM.BR .....	18
FIGURA 2 - RETICULADO ASSOCIADO À LÓGICA PARACONSISTENTE LPA2V COM GRAUS DE CRENÇA E DESCRENÇA - GERADO EM WWW.LPA2V.COM.BR .....	19
FIGURA 3 - REGISTRO DO DOMÍNIO LPA2V.COM.BR (FONTE: REGISTRO.BR). .....	23
FIGURA 4 - CLASSE LPA2V - FONTE: AUTOR (USANDO VS2013).....	25
FIGURA 5 - CLASSE RETICULADO -FONTE: AUTOR USANDO VS2013.....	26
FIGURA 6 - MÉTODO ESTADOLOGICO - FONTE: AUTOR (USANDO VS2013). .....	27
FIGURA 7 - RETICULADO <i>ONLINE</i> VALOR LÓGICO VERDADEIRO.....	28
FIGURA 8 - RETICULADO <i>ONLINE</i> VALOR LÓGICO FALSO. ....	29
FIGURA 9 -RETICULADO <i>ONLINE</i> VALOR LÓGICO INCONSISTENTE. ....	29
FIGURA 10 RETICULADO <i>ONLINE</i> VALOR LÓGICO INDETERMINADO.....	30
FIGURA 11 -RETICULADO <i>ONLINE</i> VALOR LÓGICO QUASE VERDADEIRO TENDENDO A INCONSISTENTE. ....	30
FIGURA 12 -RETICULADO <i>ONLINE</i> VALOR LÓGICO INCONSISTENTE TENDENDO A VERDADEIRO. ....	31
FIGURA 13 - RETICULADO <i>ONLINE</i> VALOR LÓGICO INCONSISTENTE TENDENDO A FALSO. ....	31
FIGURA 14 - RETICULADO <i>ONLINE</i> VALOR LÓGICO QUASE FALSO TENDENDO A INCONSISTENTE. ....	32
FIGURA 15 - RETICULADO <i>ONLINE</i> VALOR LÓGICO QUASE FALSO TENDENDO A INDETERMINADO. ....	32
FIGURA 16 -RETICULADO <i>ONLINE</i> VALOR LÓGICO INDETERMINADO TENDENDO A FALSO. ....	33
FIGURA 17 -RETICULADO <i>ONLINE</i> VALOR LÓGICO INDETERMINADO TENDENDO A VERDADEIRO. ....	33
FIGURA 18 - RETICULADO <i>ONLINE</i> VALOR LÓGICO QUASE VERDADEIRO TENDENDO A INDETERMINADO. ....	34
FIGURA 19 - ACESSO A PARTIR DE <i>SMARTPHONE</i> .....	34
FIGURA 20 - ACESSO A PARTIR DE <i>TABLET</i> .....	35
FIGURA 21 - UTILIZAÇÃO DA BIBLIOTECA LPA2V.JS - FONTE: AUTOR. ....	36
FIGURA 22 - EXPORTAÇÃO RETICULADO (WWW.LPA2V.COM.BR).....	37

FIGURA 23 - ROBÔ EMMY -FONTE (DA SILVA FILHO, ABE &TORRES, 2006).....	38
FIGURA 24 - DECISÃO EMMY PARA ESTADOS EXTREMOS - FONTE (DA SILVA FILHO, ABE &TORRES, 2006).....	38
FIGURA 25 - DECISÃO EMMY PARA ESTADOS "NÃO EXTREMOS" –FONTE (DA SILVA FILHO, ABE &TORRES, 2006). ....	39
FIGURA 26 - SIMULADOR EMMY - ESTADO VERDADEIRO - WWW.LPA2V.COM.BR .....	40
FIGURA 27 - SIMULADOR EMMY - ESTADO FALSO – WWW.LPA2V.COM.BR.....	40
FIGURA 28 - SIMULADOR EMMY - ESTADO INCONSISTENTE - WWW.LPA2V.COM.BR .....	41
FIGURA 29 - SIMULADOR EMMY - ESTADO INDETERMINADO - WWW.LPA2V.COM.BR .....	41

## LISTA DE TABELAS

TABELA 1 - ESTADOS LÓGICOS PARACONSISTENTES EXTREMOS .....	19
TABELA 2 - ESTADOS LÓGICOS PARACONSISTENTES NÃO EXTREMOS.....	19

## LISTA DE SIGLAS

HTML5	<i>Hypertext Markup Language</i> versão 5
LP	Lógica Paraconsistente
LPA	Lógica Paraconsistente Anotada
LPA2v	Lógica Paraconsistente Anotada com anotação de 2 valores
LPA4v	Lógica Paraconsistente Anotada com anotação de 4 valores
QUPC	Quadrado Unitário do Plano Cartesiano
MVC	<i>Model View Controller</i>
PNG	<i>Portable Network Graphics</i>
VS2013	<i>Visual Studio Ultimate</i> 2013

## LISTA DE SÍMBOLOS

$\tau$	Reticulado finito
$\sim$	Operador de negação
$\mu$	Grau de crença (ou de evidência favorável)
$\lambda$	Grau de descrença (ou evidência desfavorável)
$T$	Inconsistente
$V$	Verdadeiro
$\perp$	Paracompleto
$Gc$	Grau de Certeza
$Gct$	Grau de Contradição
$\perp \rightarrow f$	Paracompleto tendendo ao Falso
$\perp \rightarrow v$	Paracompleto tendendo ao Verdadeiro
$T \rightarrow f$	Inconsistente tendendo ao Falso
$T \rightarrow v$	Inconsistente tendendo ao Verdadeiro
$Qv \rightarrow T$	Quase-verdadeiro tendendo ao Inconsistente
$Qf \rightarrow T$	Quase-falso tendendo ao Inconsistente
$Qf \rightarrow \perp$	Quase-falso tendendo ao Paracompleto
$Qv \rightarrow \perp$	Quase- verdadeiro tendendo ao Paracompleto
$V_{scc}$	Valor limite superior de Controle de certeza
$V_{sci}$	Valor limite superior de Controle de incerteza
$V_{icc}$	Valor limite inferior de Controle de certeza
$V_{ici}$	Valor limite inferior de Controle de incerteza

## SUMÁRIO

1.	INTRODUÇÃO .....	12
1.1.	OBJETIVOS .....	13
1.2.	LÓGICA CLÁSSICA .....	14
1.3.	LÓGICAS NÃO CLÁSSICAS.....	15
1.4.	LÓGICA PARACONSISTENTE .....	16
1.5.	LÓGICA PARACONSISTENTE ANOTADA (LPA2V) .....	17
2.	MATERIAIS E MÉTODOS .....	21
2.1.	REPRESENTAÇÃO LPA2V USANDO CANVAS E JAVASCRIPT (HTML5) .....	22
2.2.	REGISTRO DE DOMÍNIO .....	22
2.3.	DEFINIÇÃO DAS TECNOLOGIAS .....	24
2.4.	MODELAGEM DAS CLASSES .....	24
2.5.	IMPLEMENTAÇÃO DOS ALGORITMOS .....	26
3.	RESULTADOS E DISCUSSÕES .....	28
3.1.	INTERAÇÃO COM A FERRAMENTA <i>ONLINE</i> .....	28
3.2.	UTILIZAÇÃO DA BIBLIOTECA.....	35
3.3.	EXPORTAÇÃO DE RETICULADOS.....	36
3.4.	ESTUDO DE CASO (CONTROLADOR LPA2V ROBO EMMY).....	37
4.	CONCLUSÕES .....	42
5.	TRABALHOS FUTUROS.....	42
	REFÊRENCIAS.....	44
	APÊNDICE I – CÓDIGO BIBLIOTECA LPA2V.JS.....	46

## 1. INTRODUÇÃO

O recrudescimento da complexidade dos problemas enfrentados nos processos de automação em função da multiplicidade de fatores envolvidos e a crescente expectativa de precisão dos resultados tem feito aumentar o interesse por soluções não tradicionais. Nesta linha a lógica clássica, com sua visão maniqueísta do universo, em algumas situações tem se revelado insuficiente para lidar com as situações efetivamente encontradas nos processos de automação. A lógica Paraconsistente surge como uma das novas formas de lidar com a natureza complexa, ambígua, imprecisa e incerta dos processos candidatos a automatização.

A Lógica Paraconsistente é uma Lógica Não-Clássica que revoga o princípio da Não Contradição e admite o tratamento de sinais contraditórios na sua estrutura teórica. Desde a sua recente criação as lógicas paraconsistentes têm sido utilizadas em uma ampla variedade de domínios onde se permite um tratamento lógico da crença, níveis de incerteza e de inconsistências (DA SILVA FILHO, 2010).

A lógica Paraconsistente tem sido utilizada como parte da abordagem para inúmeros trabalhos de pesquisa das mais variadas áreas do conhecimento, desde a Ecologia como utilizada por D'antonio (2013) em sua avaliação quantitativa de grânulos plásticos em praias, como na Engenharia Mecânica, vide Fernandes (2012) onde a lógica Paraconsistente é utilizada diretamente relacionada a sistemas de automação e controle. Mesmo uma pesquisa informal usando mecanismos de busca de *internet* revela uma grande diversidade de aplicações da lógica Paraconsistente.

Em comum a maioria dos trabalhos de pesquisa na área, há a necessidade de visualizar, de maneira gráfica, algumas proposições da lógica Paraconsistente especialmente a construção de reticulado de Hasse (DA SILVA FILHO & ABE, 2000). A ausência de ferramentas automatizadas gratuitas ou de baixo custo, faz com que os diagramas usados para corroborar ou demonstrar os conceitos da lógica Paraconsistente sejam construídos manualmente. O propósito deste trabalho é elaborar ferramenta de código aberto e disponibilizá-la na *internet* em sítio de fácil acesso que seja capaz de gerar os diagramas associados aos conceitos fundamentados na lógica Paraconsistente, especialmente os relativos a lógica Paraconsistente anotada de 2 valores (LPA2v).

A Lógica Paraconsistente Anotada LPA pode ser representada de modo particular, através de um Reticulado de quatro Vértices em que, intuitivamente, as constantes de anotação representadas nos seus Vértices vão dar conotações de estados lógicos extremos às proposições. A anotação pode ser composta por 1, 2 ou n valores, dependendo da classe de Lógica Paraconsistente utilizada (DA SILVA FILHO, 2010).

## 1.1. OBJETIVOS

É inegável a crescente utilização da *Internet* como parte integrante de projetos de pesquisa acadêmica, seja na busca de artigos científicos em publicações *online*, seja no consumo de ferramentas de apoio. Por ser uma área de conhecimento relativamente nova, a lógica Paraconsistente não dispõe ainda de ferramentas de apoio *online*. Grande parte dos trabalhos existentes reimplementa os algoritmos e refaz as representações gráficas usando ferramentas manuais de desenho por computador. O uso de tal expediente pode ser facilmente verificado em artigos como em Da Silva Filho (2010) ou Ferrara (2011) ou ainda em dissertações de mestrado como em Fernandes (2012). O presente trabalho tem como objetivo a criação de uma ferramenta *online* que permita explorar de maneira interativa os conceitos associados a lógica Paraconsistente Anotada, servindo de apoio ao aprendizado e domínio dos conceitos envolvidos, especialmente os relacionados a lógica Paraconsistente anotada de dois valores (LPA2v). Esta ferramenta deve também possibilitar a extração de figuras para inserção em trabalhos científicos. Ainda como objetivo paralelo está a criação de uma biblioteca reutilizável de código *Javascript* que possa ser incorporada em páginas construídas com a linguagem de marcação HTML5, para criação de reticulados estáticos ou interativos.

De maneira geral podemos listar os seguintes objetivos para o presente trabalho:

- Criar e disponibilizar biblioteca reutilizável para uso em projetos envolvendo simuladores de controladores lógicos paraconsistentes;
- Incluir no seu processo de construção a reserva de um domínio de fácil localização e passível de indexação por mecanismos de busca da *internet*;
- Apresentar o processo de transcrição dos algoritmos da lógica Paraconsistente para código fonte reutilizável que respeite os conceitos

modernos de programação tais como encapsulamento, herança e separação de responsabilidades;

- Listar em sua revisão bibliográfica outras fontes de aprendizado para lógica Paraconsistente;
- Criar e disponibilizar sitio na *internet*, relacionado a produção do trabalho, capaz de ser acessível por computadores tradicionais ou dispositivos móveis de maneira otimizada e interativa, que possibilite a criação de um de reticulado utilizado na lógica Paraconsistente Anotada de dois valores (LPA2v);
- Possibilitar que a ferramenta a ser criada exporte os reticulados gerados para incorporação direta em trabalhos científicos, especialmente os relacionados a área de automação;
- Possibilitar ainda, que a ferramenta gerada seja suficiente genérica para utilização em outras áreas;
- Possibilitar que a ferramenta gerada seja interativa e prontamente atualize a representação gráfica do reticulado sem acesso a servidores externos.

## 1.2. LÓGICA CLÁSSICA

Lógica trata do raciocínio, de como a partir de premissas se chega a uma conclusão (GENSLER, 2010).

Segundo Da Silva Filho, Abe, & Lambert-Torres (2008) os conceitos explicitados pela lógica têm origem nos trabalhos do filósofo grego Aristóteles (384 a 322 a.C.) e de seus discípulos, reunidos na obra denominada *Organon*.

Como citado por Graham (2008) entre o final do século dezenove e o início do século vinte, alguns matemáticos e filósofos – dentre eles George Boole (1815–1864), Augustus De Morgan (1806–1871), Gottlob Frege (1848–1925), Bertrand Russell (1872–1970) e Alfred North Whitehead (1861–1947) – começaram a dar um tratamento matemático a lógica formal, uma vez que a linguagem natural não possui o rigor necessário para validar as conclusões obtidas. Desta forma foi adotada uma linguagem estrita e universal, constituída por símbolos específicos.

Usando esta notação matemática (lógica proposicional) pode-se representar os quatro princípios que formam a base da lógica clássica, a saber:

**Princípio da Identidade  $p = p$** 

Toda proposição ou objeto é idêntico a si mesmo

**Princípio da Identidade proposicional  $p \rightarrow p$** 

Toda proposição implica nela mesma

**Princípio do Terceiro Excluído  $p \vee \neg p$** 

De duas proposições contraditórias, isto é, uma nega a outra, uma delas é verdadeira.

**Princípio da Não-contradição  $\neg (p \vee \neg p)$** 

Entre duas proposições contraditórias, uma delas é Falsa.

De acordo com Da Silva Filho, Abe, & Lambert-Torres (2008), a lógica clássica é binária, portanto uma declaração é falsa ou verdadeira, não admitindo ser ao mesmo tempo parcialmente verdadeira e parcialmente falsa. Esta natureza binária faz com que a lógica clássica seja de fácil transcrição para sistemas computacionais que também lidam com natureza binária (o valor de um “bit” ou é “0” ou “1”).

A natureza complexa do mundo atual com valores ausentes, imperfeitos, contraditórios e por vezes aparentemente incoerentes, faz que a lógica clássica tenha se tornado ineficiente para tratar certas classes de problemas. Por volta dos últimos cinquenta anos tem surgido várias alternativas para a lógica clássica convencional, coletivamente definidas como “lógicas não clássicas” (GRAHAM, 2008).

**1.3. LÓGICAS NÃO CLÁSSICAS**

As lógicas não clássicas rejeitam as premissas básicas da lógica clássica que assumem que uma afirmativa é falsa ou verdadeira, mas não ambas simultaneamente, e que “verdadeiro” e “falso” são as únicas conclusões possíveis. Segundo Gensler (2010), a partir do momento que estes valores são questionados se abre espaço para diferentes concepções

As Lógicas Não-Clássicas violam justamente estas suposições binárias que não admitem ambiguidades e contradições. Por outro lado, o conceito de dualidade, estabelecendo que algo pode e deve coexistir com o

seu oposto, faz as aplicações das Lógicas Não-Clássicas parecer natural e até mesmo inevitável. Em muitas das experiências humanas não se pode afirmar categoricamente como sendo verdadeiras ou falsas, sim ou não como exige as leis da Lógica Clássica. Com uma verificação mais precisa das coisas que nos rodeia chega-se à conclusão que, entre a certeza de ser e a certeza de não ser, existem infinitos graus de incerteza (DA SILVA FILHO, 2010).

Para o propósito deste trabalho o especial interesse está na lógica não clássica denominada como “paraconsistente”.

#### **1.4. LÓGICA PARACONSISTENTE**

Segundo Gensler (2010), o princípio da “não contradição” definido por Aristóteles estabelece que uma propriedade não pode “pertencer” e “não pertencer” simultaneamente ao mesmo objeto, ou seja “S é P” e “S não é P” não podem ser ambas verdadeiras ao mesmo tempo. Mesmo entre os gregos precursores dos princípios da lógica clássica, este princípio já era questionado. Aristóteles menciona Heráclito como um dos que negavam o princípio da não contradição.

Modernamente podemos considerar como precursores da lógica Paraconsistente, entre outros, lógico polonês Jean Lukasiewicz (1876-1956), o lógico russo Nicolai Vasiliev (1880 – 1940) e o lógico polonês Stanisław Jaśkowski (1906 – 1965) discípulo de Lukasiewicz. Entre 1910 e 1913, trabalhando de maneira independente, Lukasiewicz e Vasiliev chamaram a atenção de que, similarmente ao que já havia sido constatado com os axiomas da geometria euclidiana, alguns princípios da lógica aristotélica admitiam revisão, inclusive o da contradição (KRAUSE, 2004).

Foi Jaśkowski que em 1948 apresentou uma lógica para sistemas envolvendo contradições, porém sem ser trivial. O sistema criado por Jaśkowski, denominado como lógica discursiva ou discussiva (KRAUSE, 2004).

No Brasil os trabalhos realizados por Newton C. A. da Costa (1929 -), de maneira independente dos autores citados, a partir da década de 1950, deram um tratamento tanto filosófico como matemático, a sistemas lógicos que pudessem envolver contradições.

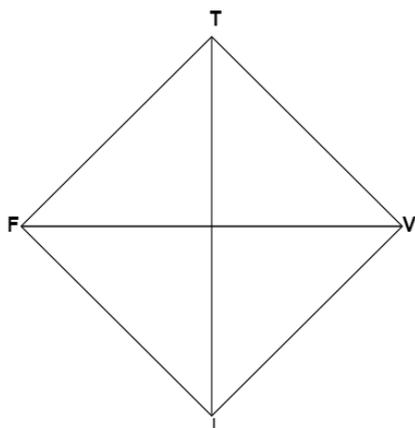
Uma teoria (dedutiva)  $T$ , assentada na lógica  $L$ , diz-se consistente se entre seus teoremas não existem tais, que um seja a negação do outro; em hipótese contrária,  $T$  denomina-se inconsistente. A teoria  $T$  chama-se trivial se todas as sentenças (fórmulas fechadas) de sua linguagem forem teoremas; se isto não ocorrer,  $T$  é não-trivial. Se  $L$  for uma das lógicas comuns, como a clássica e a intuicionista de Brouwer-Heyting,  $T$  é trivial se e somente se for inconsistente. Em outras palavras, lógicas como essas não separam os conceitos de inconsistência e de trivialidade.  $L$  denomina-se paraconsistente se puder funcionar como fundamento de teorias inconsistentes e não triviais. Dito de outro modo, uma lógica paraconsistente mostra-se capaz de manipular sistemas inconsistentes de informações sem o perigo de trivialização. (Somente em certas circunstâncias específicas a presença de contradição implica em trivialização) (DA SILVA FILHO, 2000).

### **1.5. LÓGICA PARACONSISTENTE ANOTADA (LPA2v)**

Para que a lógica Paraconsistente não seja limitada apenas a discussão filosófica e possa ser empregada em aplicações práticas, é necessário dotar a mesma de um tratamento matemático.

Conforme Subrahmanian (1987), os estudos da Lógica Paraconsistente deram origem a outros tipos de estudos relacionados às Lógicas Não-Clássicas, dentre estas as Lógicas Paraconsistentes Anotadas (LPA). A LPA propõe tratar os conceitos associados a lógica Paraconsistente através do uso de graus de crença relativos a uma proposição. Os graus de crença são expressados através de valores reais contidos no intervalo entre 0 e 1 (inclusive). Estes graus de crença podem ser obtidos através de medições, estatísticas, probabilidades ou serem provenientes da atribuição de um especialista (LEMES NETO & VENSON, 2015).

Para possibilitar uma representação gráfica, a LPA pode ser associada a um reticulado onde cada vértice é associado a um estado lógico extremo (DA SILVA FILHO & ABE, 2000). Vide Figura 1 – Reticulado associado a lógica Paraconsistente anotada - Estados lógicos - gerado em [www.lpa2v.com.br](http://www.lpa2v.com.br) e Tabela 1 - Estados Lógicos Paraconsistentes Extremos



**Figura 1 – Reticulado associado a lógica Paraconsistente anotada - Estados lógicos - gerado em [www.lpa2v.com.br](http://www.lpa2v.com.br)**

O estado lógico é obtido através de dois valores que compõe a anotação  $(\mu, \lambda)$  onde  $\mu$  representa o Grau de crença ou de evidência favorável e  $\lambda$  representa o Grau de descrença ou evidência desfavorável atribuído a uma proposição.

Os estados lógicos extremos, com os valores dos graus de crença e de descrença, podem ser relacionados da seguinte forma (vide Figura 2 - Reticulado associado à Lógica Paraconsistente LPA2v com graus de crença e descrença - gerado em [www.lpa2v.com.br](http://www.lpa2v.com.br)):

T = (1, 1) Inconsistente

V = (1, 0) Verdadeiro

F = (0, 1) Falso

⊥ = (0, 0) Indeterminado

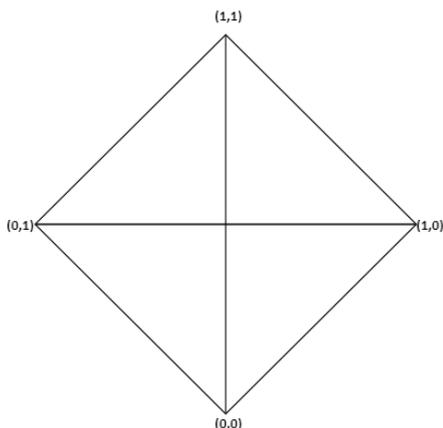


Figura 2 - Reticulado associado à Lógica Paraconsistente LPA2v com graus de crença e descrença - gerado em [www.lpa2v.com.br](http://www.lpa2v.com.br)

Tabela 1 - Estados Lógicos Paraconsistentes Extremos

Estado	Definição
V	Verdadeiro
F	Falso
$\perp$	Indeterminado
T	Inconsistente

Além dos estados lógicos “extremos” a LPA2v permite a determinação de outros estados lógicos paraconsistentes listados abaixo:

Tabela 2 - Estados Lógicos Paraconsistentes Não Extremos

Estado	Definição
$T \rightarrow f$	Inconsistente tendendo ao falso
$T \rightarrow v$	Inconsistente tendendo ao verdadeiro
$\perp \rightarrow f$	Indeterminado tendendo ao falso
$\perp \rightarrow v$	Indeterminado tendendo ao verdadeiro
$Qf \rightarrow T$	Quase falso tendendo ao inconsistente
$Qf \rightarrow \perp$	Quase falso tendendo ao indeterminado
$Qv \rightarrow T$	Quase verdadeiro tendendo ao inconsistente
$Qv \rightarrow \perp$	Quase verdadeiro tendendo ao indeterminado

Segundo Da Silva Filho, Abe, & Lambert-Torres (2008) é possível obter os valores discretos dos possíveis estados lógicos através do uso de um algoritmo “para-

analisador”. Além dos valores discretos relativos aos estados lógicos o algoritmo permite também obter duas saídas reais representando o “grau de certeza” ( $Gc$ ) e o “grau de contradição” ( $Gct$ ) calculados a partir dos valores de evidência favorável e desfavorável fornecidos. Este algoritmo pode ser definido usando uma representação em “português estruturado”, de maneira a ser implementado em qualquer linguagem de programação. O algoritmo pode ser descrito da seguinte forma (DA SILVA FILHO, ABE, & LAMBERT TORRES, 2008):

Valores de Entrada:

$\mu$  - Grau de Evidência Favorável

$\lambda$  – Grau de Evidência Desfavorável

$V_{sc}$  – Valor Superior de Controle de Certeza

$V_{scct}$  – Valor Superior de Controle de Contradição

$V_{ic}$  – Valor Inferior de Controle de Certeza

$V_{icct}$  – Valor Inferior de Controle de Contradição

Valores de Saída:

$S_1$  = saída discreta

$S_{2a}$  = saída analógica

$S_{2b}$  = saída analógica

Consistências:

$$0 \leq \mu \leq 1$$

$$0 \leq \lambda \leq 1$$

Algoritmo:

/\* Calcular graus de Certeza e Contradição: \*/

$$Gc = \mu - \lambda$$

$$Gct = \mu + \lambda - 1$$

/\* Estados Lógicos Extremos \*/

Se  $Gc \geq V_{sc}$  então  $S_1 = V$

Se  $Gc \leq V_{ic}$  então  $S_1 = F$

Se  $Gct \geq V_{scct}$  então  $S_1 = T$

Se  $Gct \leq V_{icct}$  então  $S_1 = \perp$

```

/*Estados Lógicos Não Extremos */
Para  $0 \leq Gc < Vsc$  e  $0 \leq Gct < Vscct$ 
  Se  $Gc \geq Gct$  então  $S1 = Qv \rightarrow T$ 
  Senão  $S1 = T \rightarrow v$ 
Para  $0 \leq Gc < Vsc$  e  $Vicct < Gct \leq Vscct$ 
  Se  $Gc \geq |Gct|$  então  $S1 = Qv \rightarrow \perp$ 
  Senão  $S1 = \perp \rightarrow v$ 
Para  $Vicc < Gc \leq 0$  e  $Vicct < Gct \leq Vscct$ 
  Se  $|Gc| \geq |Gct|$  então  $S1 = Qf \rightarrow \perp$ 
  Senão  $S1 = \perp \rightarrow f$ 
Para  $Vicc < Gc \leq 0$  e  $0 \leq Gct < Vscct$ 
  Se  $|Gc| \geq Gct$  então  $S1 = Qf \rightarrow T$ 
  Senão  $S1 = T \rightarrow f$ 

S2a = Gct
S2b = Gc
/* fim */

```

Os resultados obtidos através do uso do algoritmo podem então ser utilizados em processos de tomada de decisão nas mais variadas áreas incluindo robótica e engenharia de controle (DA SILVA FILHO & ABE, 2000).

## 2. MATERIAIS E MÉTODOS

Para a elaboração do presente trabalho foram adotadas metodologias e técnicas que serão apresentadas nos próximos tópicos.

### 2.1. REPRESENTAÇÃO LPA2v USANDO CANVAS e JAVASCRIPT (HTML5)

Embora o algoritmo “para analisador” descrito no item 1.5 possa ser implementado em qualquer linguagem de programação, neste trabalho foi feita a opção por se criar uma biblioteca reutilizável que contemplasse tanto o tratamento matemático necessário como também a sua representação gráfica.

Visando a utilização na *internet* foi adotada a linguagem *Javascript* suportada por todos os navegadores e um objeto específico do HTML5 denominado CANVAS. Trata-se de uma área retangular onde podem ser desenhados elementos gráficos de diversas natureza e suportado tanto por navegadores de *internet* tradicionais, quanto os usados em dispositivos móveis (GEARY, 2012).

Para atingir os objetivos definidos como alvo do presente trabalho foi necessária uma combinação de atividades administrativas, tais como reserva de domínio e alocação de hospedagem, e atividades de cunho tecnológico como escolha de ferramentas de desenvolvimento, linguagem de programação e metodologias de desenvolvimento. Após a realização destas tarefas foi necessário definir e implementar o código fonte na linguagem de programação *Javascript*, de maneira a permitir a reutilização do mesmo de maneira simples.

### 2.2. REGISTRO DE DOMÍNIO

Para acessar a ferramenta proposta no presente trabalho o usuário deverá apontar seu navegador para o endereço eletrônico “[www.lpa2v.com.br](http://www.lpa2v.com.br)”. No Brasil o registro de domínios é feito pelo “Núcleo de Informação e Coordenação do Ponto BR”, uma entidade civil, sem fins lucrativos, que desde dezembro de 2005 implementa as decisões e projetos do Comitê Gestor da Internet no Brasil. Para registro de um domínio é necessário realizar os seguintes passos:

- Verificar a disponibilidade do nome do domínio através da pesquisa de disponibilidade. Disponível em [https://registro.br/cgi-bin/nicbr/busca\\_dominio](https://registro.br/cgi-bin/nicbr/busca_dominio)

- Se o domínio estiver disponível proceder com o cadastro do administrador da conta. Disponível em <https://registro.br/cgi-bin/nicbr/idmng>
- Proceder com registro do domínio indicando as responsabilidades técnica e de cobrança.

Após a realização dos passos descritos acima, o domínio lpa2v.com.br ficou reservado para ser utilizado como ponto de acesso a ferramenta proposta no presente trabalho. Conforme indicado em Figura 3 - Registro do domínio lpa2v.com.br

The screenshot shows the Registro.br website interface. At the top, there is a navigation bar with the Registro.br logo and a '25 anos do Brasil' badge. The main navigation includes 'PESQUISAR E REGISTRAR DOMÍNIOS', 'CRIAR CONTA', and 'ACESSAR CONTA'. Below this is a dark navigation bar with links for 'Sobre Domínios', 'Tecnologia', 'Ajuda', 'Quem Somos', and 'Contato'. The main content area is titled 'Whois' and features a search input field with the placeholder 'Faça sua consulta' and a green 'CONSULTAR' button. Below the search bar, there is a link for 'Versão com informações de contato'. The search results are displayed in a pre-formatted text block, showing the following information:

```

% Copyright (c) Nic.br
% A utilização dos dados abaixo é permitida somente conforme
% descrito no Termo de Uso (http://registro.br/termo), sendo
% proibida a sua distribuição, comercialização ou reprodução,
% em particular para fins publicitários ou propósitos
% similares.
% 2015-03-03 16:55:48 (BRT -03:00)

dominio: lpa2v.com.br
titular: CLAUDIO SOUZA NUNES
documento: 074.195.988-78
pais: BR
c-titular: CLSNU
c-admin: CLSNU
c-técnico: CLSNU
c-cobrança: CLSNU
servidor DNS: d.sec.dns.br
status DNS: 03/03/2015 AA
ultimo AA: 03/03/2015
servidor DNS: e.sec.dns.br
status DNS: 03/03/2015 AA
ultimo AA: 03/03/2015
registro DS: 2312 RSASHA1 98D025E9A1DAF0B3027F647C05F40605858809C7
status DS: 27/02/2015 DSOK
ultimo OK: 27/02/2015
saci: sim
criado: 17/10/2014 #13460242
expiração: 17/10/2015
alterado: 26/02/2015
status: publicado

Contato (ID): CLSNU
nome: Claudio Souza Nunes
e-mail: claudio.souza.nunes@globo.com
criado: 20/09/2006
alterado: 20/09/2006

% Problemas de segurança e spam também devem ser reportados ao
% cert.br, http://cert.br/, respectivamente para cert@cert.br
% e mail-abuse@cert.br
%
  
```

Figura 3 - Registro do domínio lpa2v.com.br (Fonte: Registro.br).

### 2.3. DEFINIÇÃO DAS TECNOLOGIAS

Para execução do projeto proposto neste trabalho foi necessário a definição de quais ferramentas tecnológicas poderiam ser utilizadas, levando em conta a possibilidade da continuidade do projeto por terceiros ou a migração para um modelo de desenvolvimento comunitário.

Foram utilizados os seguintes critérios:

1. A ferramenta de desenvolvimento deveria incorporar as melhores práticas de desenvolvimento e possuir uma versão gratuita. Após análise das opções disponíveis foi escolhido o “*Microsoft Visual Studio Ultimate*” (MARTIN & BANKS, 2014).
2. O código deveria ser disponibilizado em um repositório que permitisse o compartilhamento, mas que também integrasse ferramentas de gerenciamento que possibilitassem o controle de versão. Para tanto foi escolhida a plataforma “*Visual Studio Online*” (MARTIN & BANKS, 2014).
3. A arquitetura utilizada no projeto deveria ser “não proprietária” e se basear em padrões de desenvolvimento estabelecidos. Para tanto foi escolhido o modelo “*Model View Controller*” (MVC) (FREEMAN, 2014).
4. A ferramenta deveria ser acessível tanto de computadores de mesa, como de dispositivos móveis. Em função deste requisito foi necessário a adoção de técnicas “responsivas”, com incorporação do *framework* “*Bootstrap*” (SHENOY & SOSSOU, 2014).

### 2.4. MODELAGEM DAS CLASSES

Para realizar um dos objetivos do presente trabalho, mas especificamente “criar e disponibilizar biblioteca reutilizável” foi adotada a metodologia de análise orientada para objetos.

Foram definidas duas classes: lpa2v (Figura 4 - Classe lpa2v - Fonte: autor (usando VS2013) e reticulado (Figura 5 - Classe reticulado -Fonte: autor usando VS2013). Na classe lpa2v foi incorporado a algoritmo “para analisador” descrito em 1.5, incluindo os valores de entrada (grau de evidencia favorável, grau de evidência desfavorável, valores limites para certeza e contradição, a saber: valor limite superior de controle de certeza, valor limite superior de controle de incerteza, valor limite

inferior de controle de certeza e valor limite inferior de controle de incerteza). A classe também encapsula o processo de obtenção dos valores de saída “analógicos” (grau de certeza e grau de contradição) e também o estado lógico “discreto” correspondente a um dos doze valores definidos pela LPA2v.

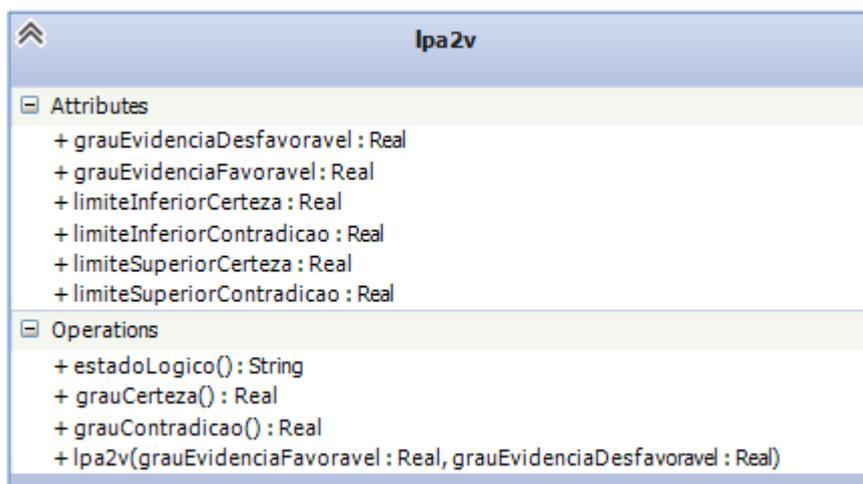


Figura 4 - Classe Ipa2v - Fonte: autor (usando VS2013).

Visando o conceito de separação de responsabilidades, a classe para representar graficamente o reticulado, apenas “plota” os valores em uma superfície gráfica bidimensional, mas especificamente um objeto “*canvas*” do HTML5, passada em seu construtor. A classe foi equipada de propriedades do tipo “verdadeiro” ou “falso” para controlar os diversos aspectos do reticulado gerado, a saber:

- 1) Representação dos valores lógicos extremos nos vértices do reticulado.
- 2) Representação numérica dos valores extremos de grau de evidência favorável e desfavorável nos vértices do reticulado.
- 3) Representação “colorida” do reticulado. Para escolha das cores foi adotado a representação de cores usada por DA SILVA FILHO & ABE (2000).
- 4) Representação dos eixos relativo ao grau de certeza e grau de contradição.
- 5) Representação dos limites usados pelo algoritmo “para analisador”.

A classe também foi equipada com métodos para desenho efetivo do reticulado e para exportação no formato “PNG”.

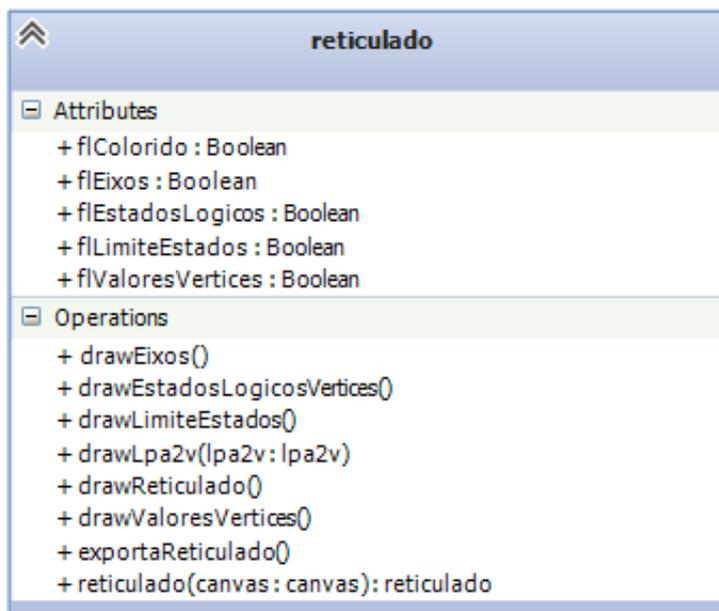


Figura 5 - Classe reticulado -Fonte: autor usando VS2013.

## 2.5. IMPLEMENTAÇÃO DOS ALGORITMOS

O principal objetivo da classe lpa2v, descrita no item 2.4, era o de encapsular o algoritmo “para analisador”. Para isso foi criado um método “estadoLogico” onde o algoritmo foi portado para linguagem *Javascript*.

Figura 6 - Método estadoLogico - Fonte: autor (usando VS2013)). Para garantir a separação de responsabilidades, este método é agnóstico em relação a sua representação visual.

```

this.estadoLogico = function () {
    var retorno = "";
    // determinacao estados "extremos"
    if (this.grauCerteza() >= this.limiteSuperiorCerteza) {
        retorno = "verdadeiro";
    }
    else if (this.grauCerteza() <= this.limiteInferiorCerteza) {
        retorno = "falso";
    }
    else if (this.grauContradicao() >= this.limiteSuperiorContradicao) {
        retorno = "inconsistente";
    }
    else if (this.grauContradicao() <= this.limiteInferiorContradicao) {
        retorno = "indeterminado";
    }
    // determinacao dos estados "nã0 extremos"
    else if ((this.grauCerteza() >= 0 && this.grauCerteza() < this.limiteSuperiorCerteza) &&
        (this.grauContradicao() >= 0 && this.grauContradicao() < this.limiteSuperiorContradicao)) {
        if (this.grauCerteza() >= this.grauContradicao()) {
            retorno = "quase verdadeiro tendendo inconsistente";
        }
        else {
            retorno = "inconsistente tendendo a verdadeiro";
        }
    }
    else if ((this.grauCerteza() >= 0 && this.grauCerteza() < this.limiteSuperiorCerteza) &&
        (this.grauContradicao() <= 0 && this.grauContradicao() >= this.limiteInferiorContradicao)) {
        if (this.grauCerteza() >= Math.abs(this.grauContradicao())) {
            retorno = "quase verdadeiro tendendo indeterminado";
        }
        else {
            retorno = "indeterminado tendendo a verdadeiro";
        }
    }
    else if ((this.grauCerteza() <= 0 && this.grauCerteza() > this.limiteInferiorCerteza) &&
        (this.grauContradicao() <= 0 && this.grauContradicao() >= this.limiteInferiorContradicao)) {
        if (Math.abs(this.grauCerteza()) >= Math.abs(this.grauContradicao())) {
            retorno = "quase falso tendendo indeterminado";
        }
        else {
            retorno = "indeterminado tendendo a falso";
        }
    }
    else if ((this.grauCerteza() <= 0 && this.grauCerteza() > this.limiteInferiorCerteza) &&
        (this.grauContradicao() >= 0 && this.grauContradicao() < this.limiteSuperiorContradicao)) {
        if (Math.abs(this.grauCerteza()) >= Math.abs(this.grauContradicao())) {
            retorno = "quase falso tendendo a inconsistente";
        }
        else {
            retorno = "inconsistente tendendo a falso";
        }
    }
    return retorno;
}

```

Figura 6 - Método estadoLogico - Fonte: autor (usando VS2013).

### 3. RESULTADOS E DISCUSSÕES

Uma vez implementados os algoritmos para gerar a representação gráfica do reticulado e encapsular o tratamento matemático da lógica Paraconsistente Anotada, foi necessário verificar se os objetivos propostos inicialmente foram atingidos. Especial interesse para as seguintes premissas:

- Criar e disponibilizar sitio na internet, relacionado a produção do trabalho, capaz de ser acessível por computadores tradicionais ou dispositivos móveis de maneira otimizada e interativa, que possibilite a criação dos reticulados utilizados na lógica Paraconsistente de dois valores (LPA2v).
- Criar e disponibilizar biblioteca reutilizável para uso em projetos futuros.

#### 3.1. INTERAÇÃO COM A FERRAMENTA *ONLINE*

Com intuito de analisar os resultados foi feito ao acesso ao sitio de *internet* [www.lpa2v.com.br](http://www.lpa2v.com.br) e foram realizadas doze interações afim de verificar visualmente se os estados lógicos estavam corretamente representados como demonstrado nas figuras a seguir:

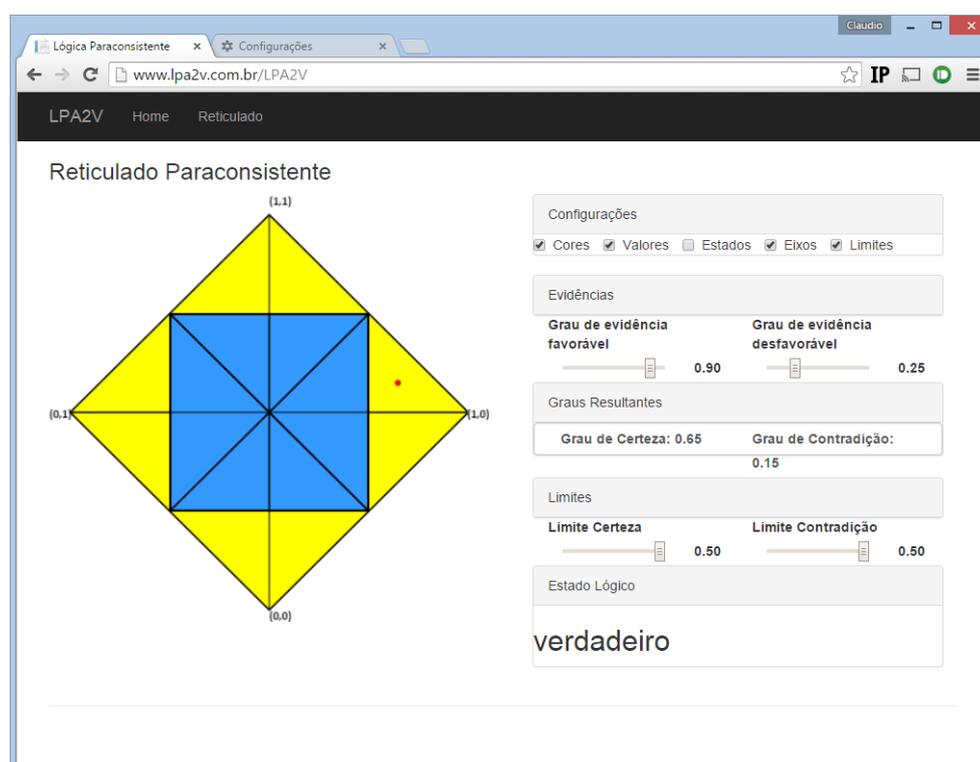


Figura 7 - Reticulado *online* valor lógico verdadeiro.

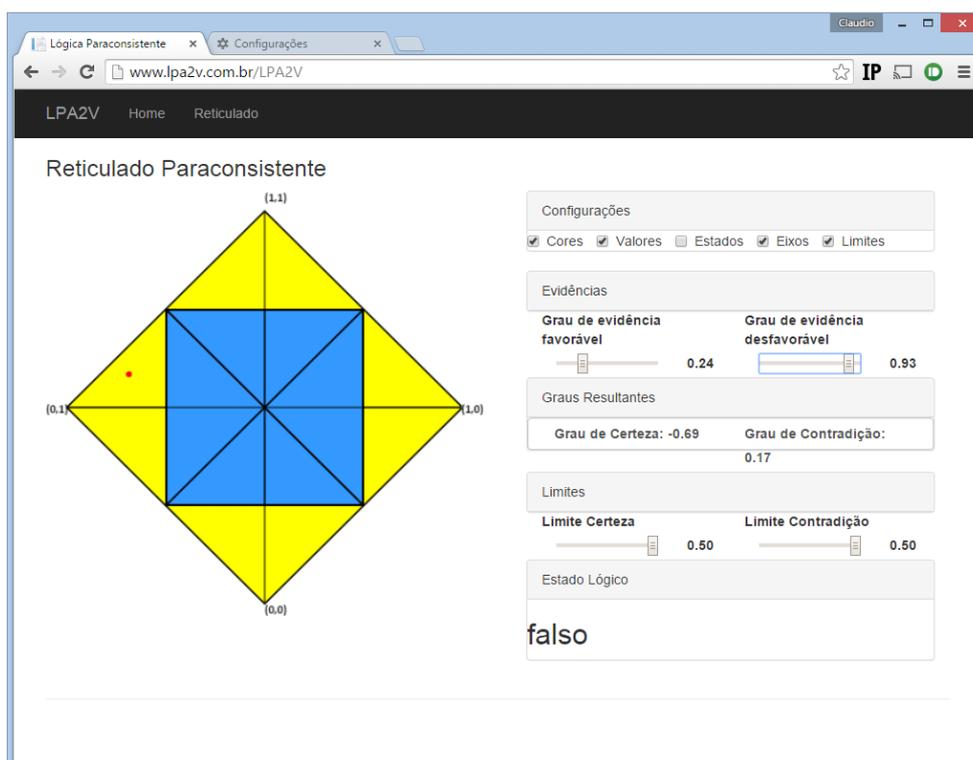


Figura 8 - Reticulado *online* valor lógico falso.

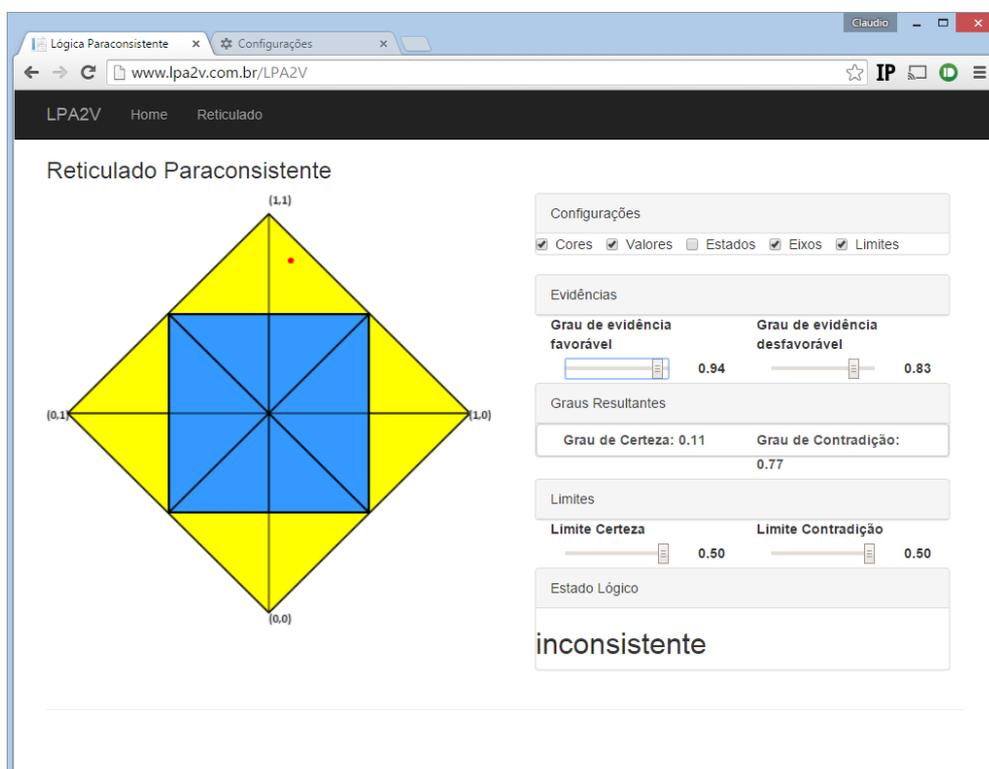


Figura 9 - Reticulado *online* valor lógico inconsistente.

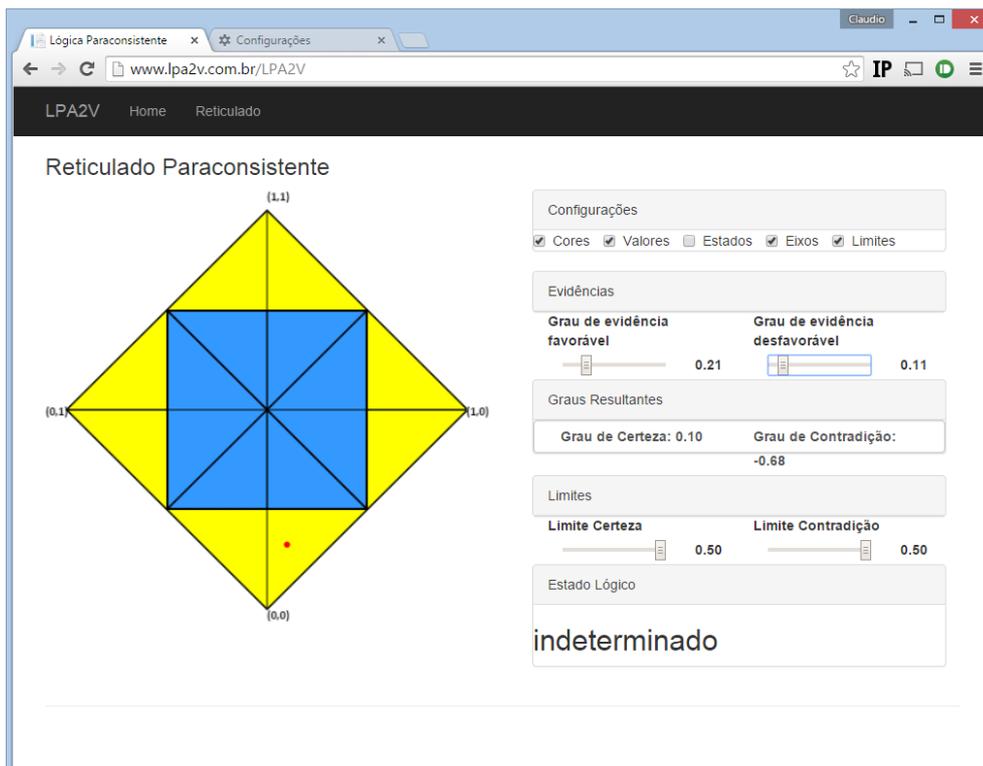


Figura 10 Reticulado *online* valor lógico indeterminado.

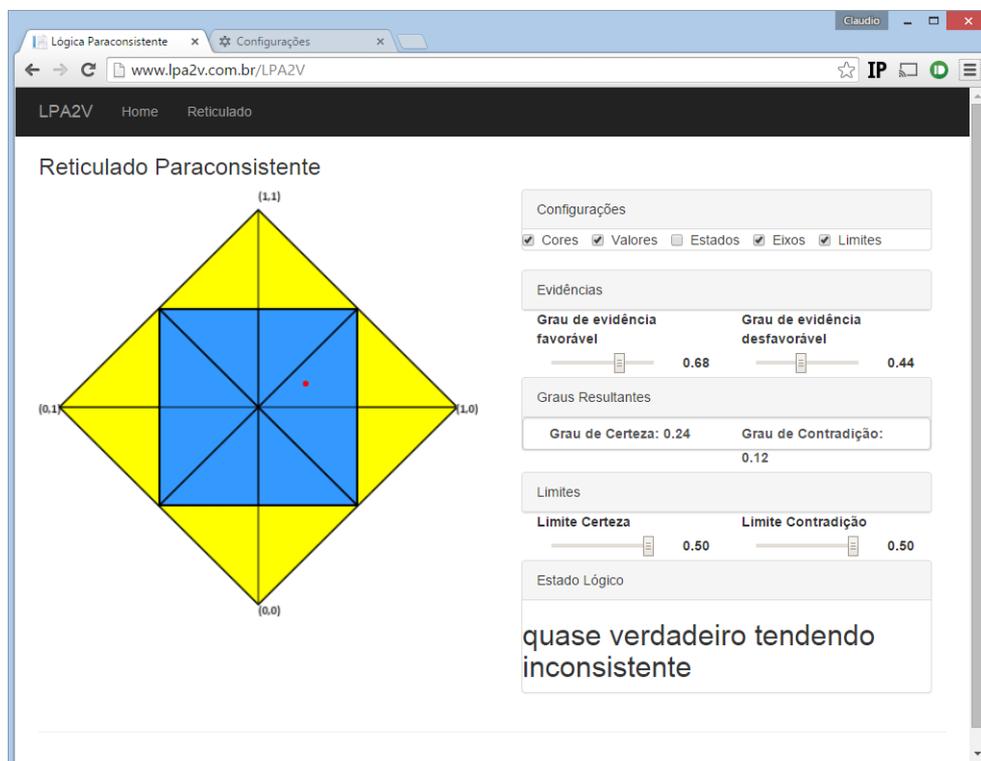


Figura 11 -Reticulado *online* valor lógico quase verdadeiro tendendo a inconsistente.

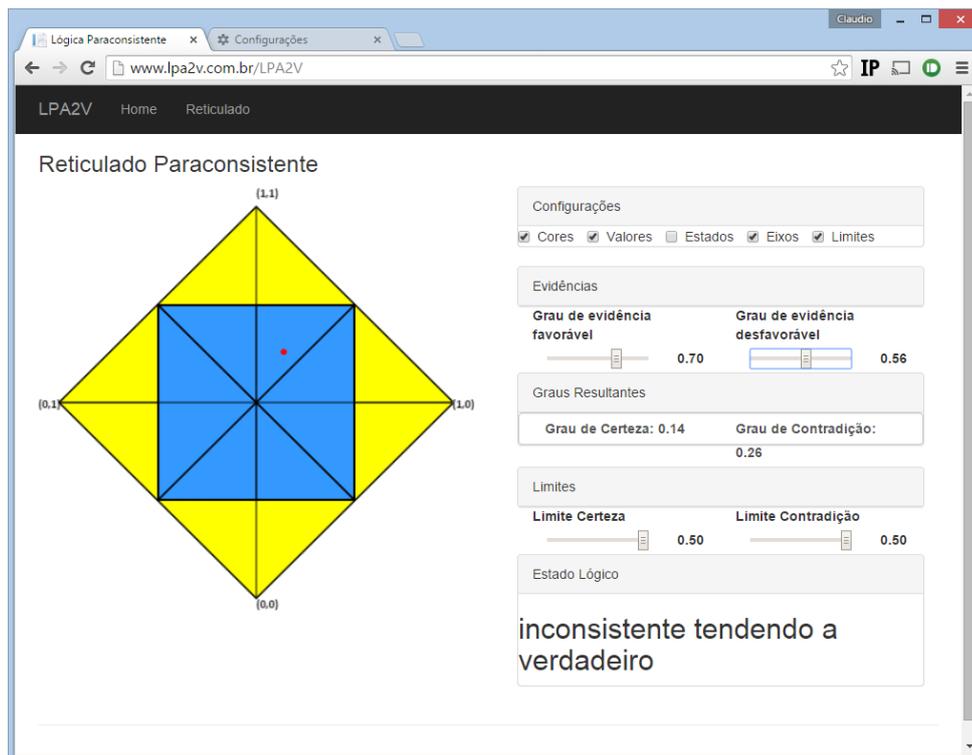


Figura 12 -Reticulado *online* valor lógico inconsistente tendendo a verdadeiro.

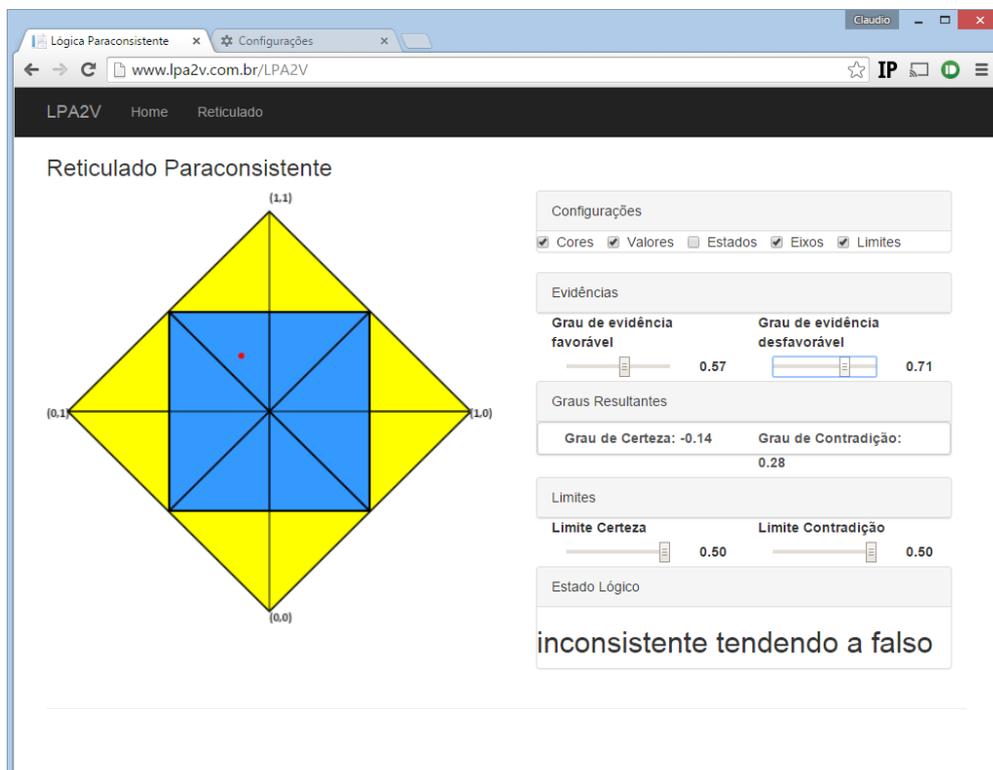


Figura 13 - Reticulado *online* valor lógico inconsistente tendendo a falso.

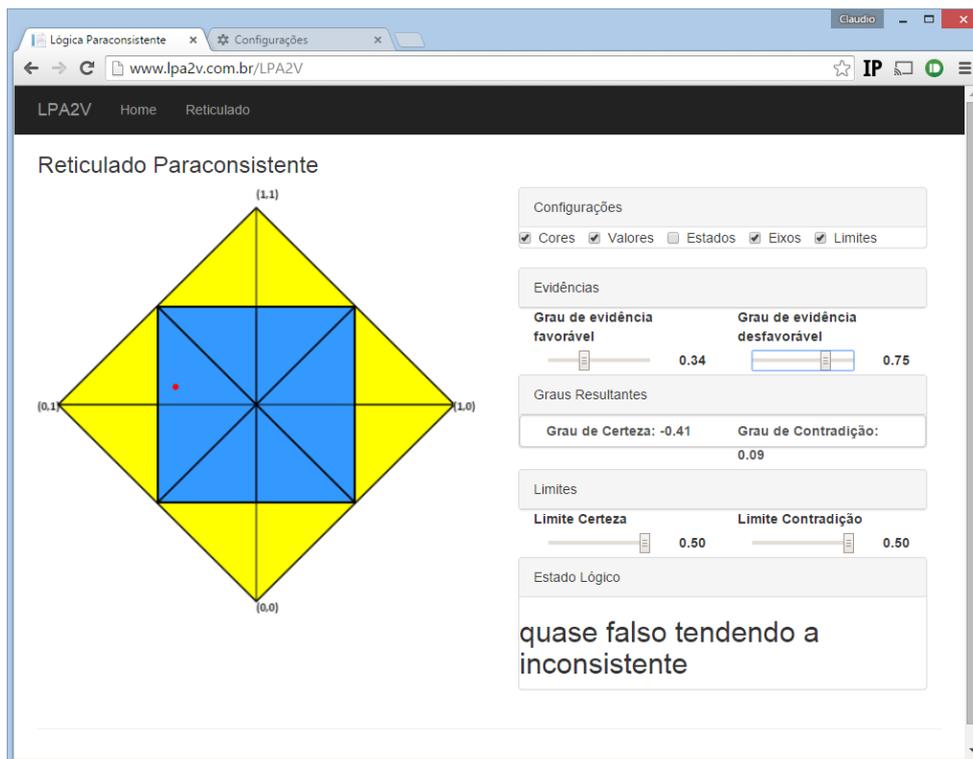


Figura 14 - Reticulado *online* valor lógico quase falso tendendo a inconsistente.

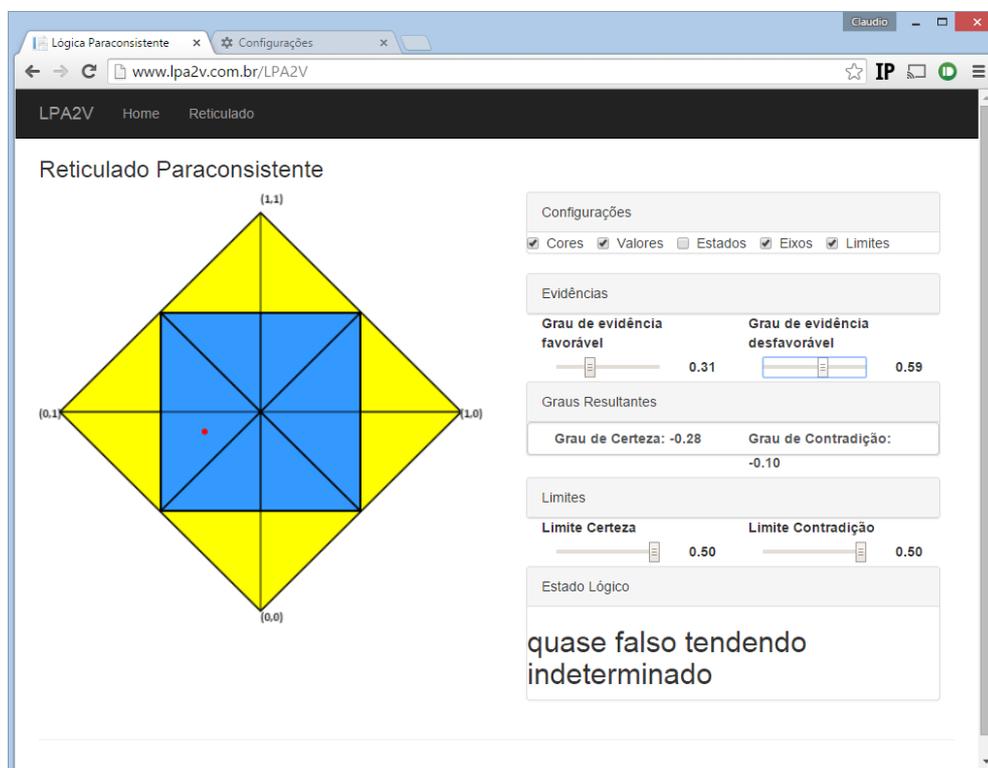


Figura 15 - Reticulado *online* valor lógico quase falso tendendo a indeterminado.

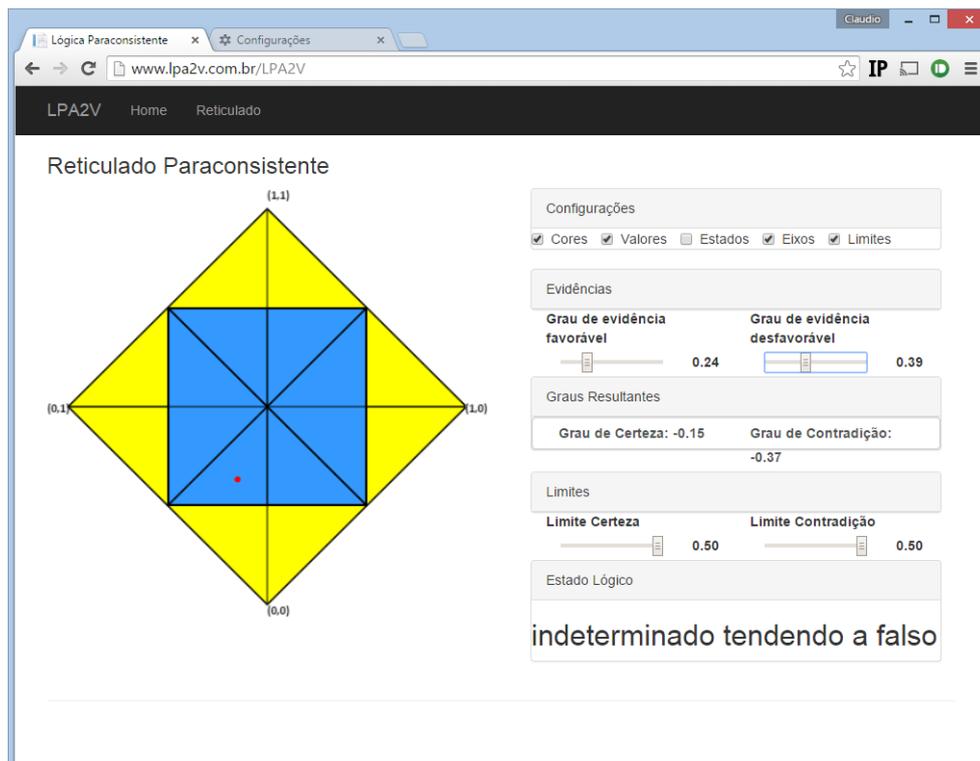


Figura 16 -Reticulado *online* valor lógico indeterminado tendendo a falso.

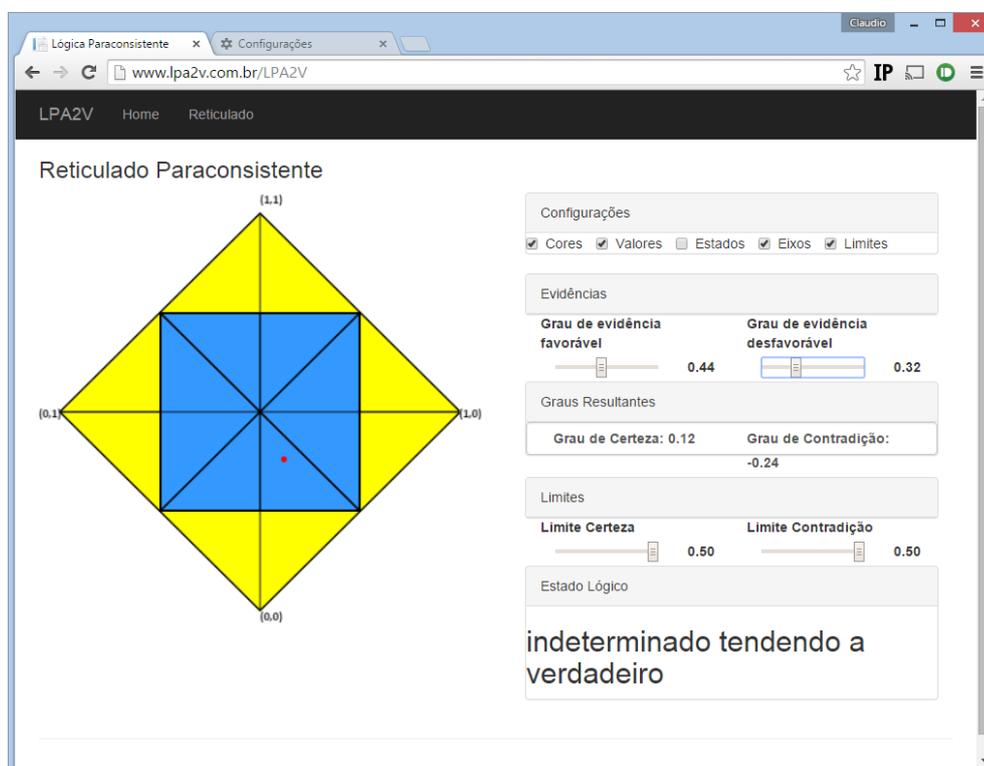


Figura 17 -Reticulado *online* valor lógico indeterminado tendendo a verdadeiro.



Figura 18 - Reticulado *online* valor lógico quase verdadeiro tendendo a indeterminado.

Entre os objetivos também foi estabelecido que a ferramenta deveria ser acessível a partir de dispositivos móveis através de técnicas responsivas, onde a aparência se adequa as características do dispositivo de acesso. Abaixo são mostrados os acessos feitos a partir de *smartphone* e *tablet*.

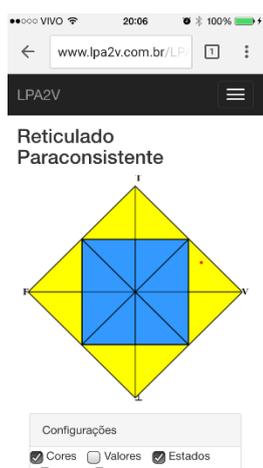


Figura 19 - Acesso a partir de *smartphone*

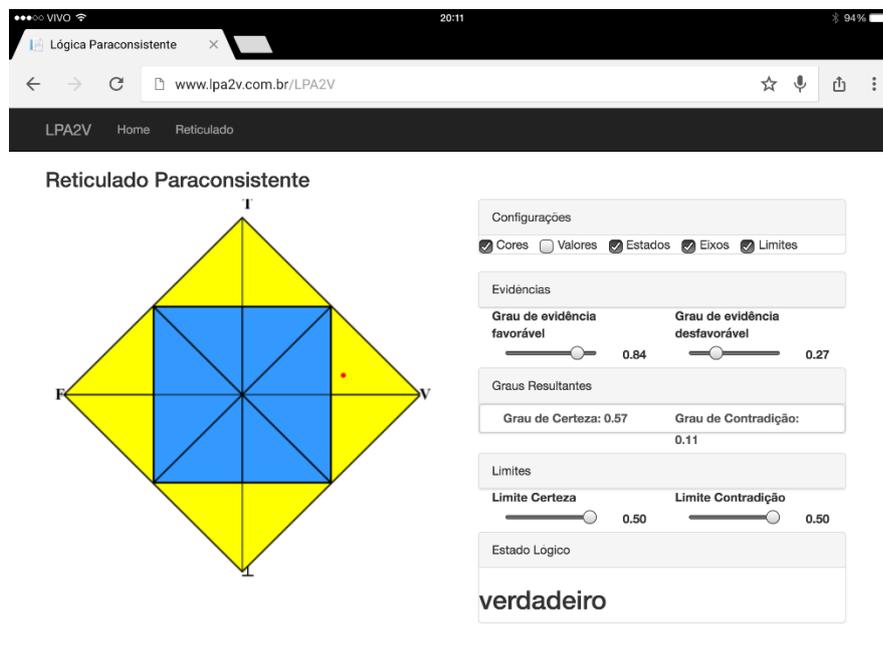


Figura 20 - Acesso a partir de *tablet*

### 3.2. UTILIZAÇÃO DA BIBLIOTECA

Uma das propostas definidas para o presente trabalho era a criação de uma biblioteca código genérica que pudesse ser reutilizada em outros projetos. No intuito de verificar esta premissa foi escrito o fragmento de código abaixo diretamente em uma página HTML. Estão destacados a inclusão da biblioteca e a criação do objeto para uso da LPA2V. O resultado da execução pode ser visto na Figura 21 - Utilização da biblioteca lpa2v.js - Fonte: autor.

```
<!DOCTYPE html>
<html>
<head>
  <script src="lpa2v.js"></script>
  <meta name="viewport" content="width=device-width" />
  <title>Demo</title>
</head>
<body>
```

```

<div>
  <button onclick="lpa2vteste()">Teste da BIBLIOTECA</button>
</div>
</body>
</html>
<script>
function lpa2vteste() {
  var l = new lpa2V(0.7, 0.2);
  alert("Grau de Certeza: " + l.grauCerteza().toFixed(4) +
        "\n Grau de Contradição: " + l.grauContradicao().toFixed(4) +
        "\nEstado Lógico: " + l.estadoLogico());
}
</script>

```

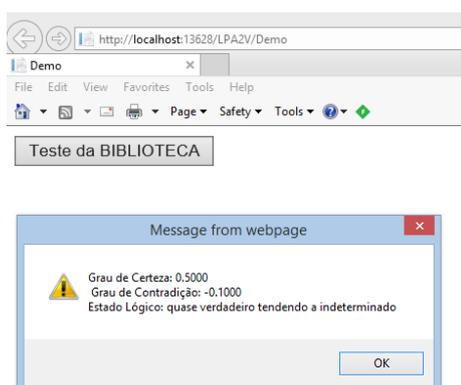


Figura 21 - Utilização da biblioteca lpa2v.js - Fonte: autor.

### 3.3. EXPORTAÇÃO DE RETICULADOS

Para cumprir um dos objetivos do presente trabalho, a ferramenta deveria permitir que o reticulado gerado interativamente fosse exportado como imagem possibilitando seu uso em outros trabalhos (artigos, dissertações, etc.). Os reticulados usados no presente trabalho foram obtidos usando esta funcionalidade.

Esta funcionalidade foi associada ao duplo clique sobre a área do reticulado e abre uma nova janela com a imagem do reticulado conforme Figura 22 - Exportação reticulado. Para salvar a imagem basta usar a opção disponível no próprio navegador.

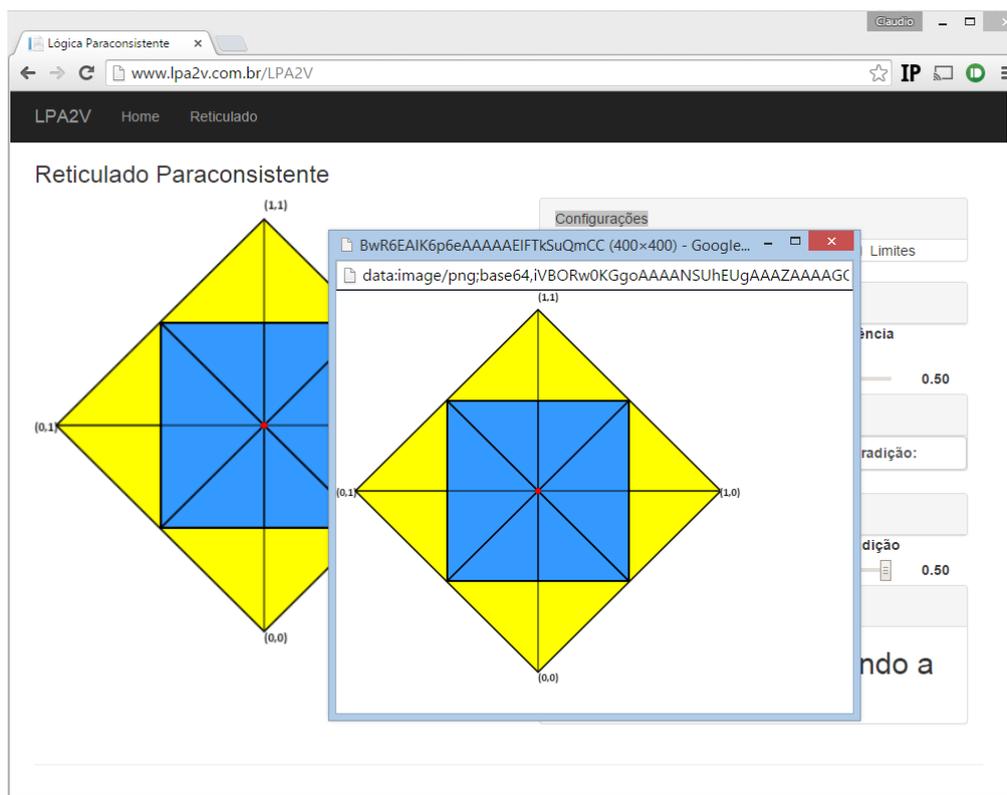


Figura 22 - Exportação reticulado (www.lpa2v.com.br).

### 3.4. ESTUDO DE CASO (CONTROLADOR LPA2V ROBO EMMY)

Para avaliar uma das premissas do presente trabalho, o de servir de base para criação de simulações em projetos de automação, foi implementado um simulador para o controlador paraconsistente usado no projeto do robô Emmy.

Segundo Da Silva Filho, Abe, & Torres (2006) o robô Emmy é um robô móvel autônomo com sistema de controle construído com os princípios da Lógica Paraconsistente Anotada constituído de uma plataforma circular de 30 cm. Os graus de evidência usados pelo robô são obtidos a partir do tratamento da informação oriunda de dois sensores de ultrassom para avaliar a distância do obstáculo. A partir das informações coletadas e tratadas através de um controlador lógico paraconsistente, o robô Emmy é capaz de trafegar em ambientes não estruturados evitando obstáculos.

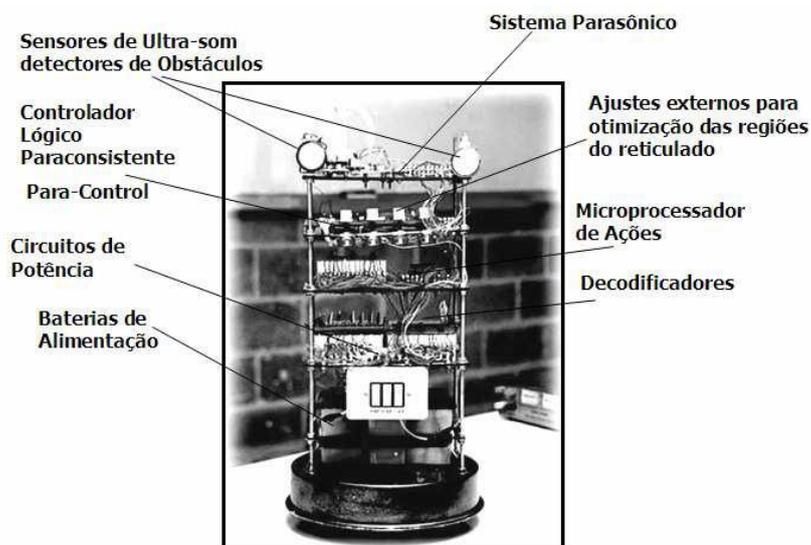


Figura 23 - Robô Emmy -Fonte (DA SILVA FILHO, ABE &TORRES, 2006)

O processo de decisão usado pelo robô Emmy para os estados extremos pode ser visto na figura abaixo:

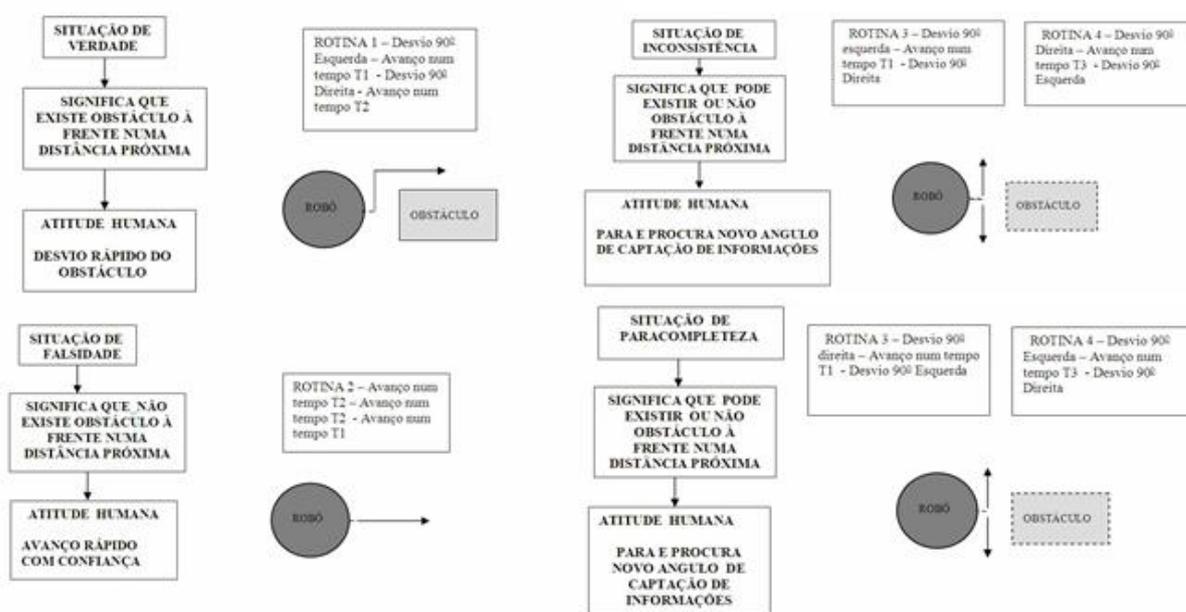
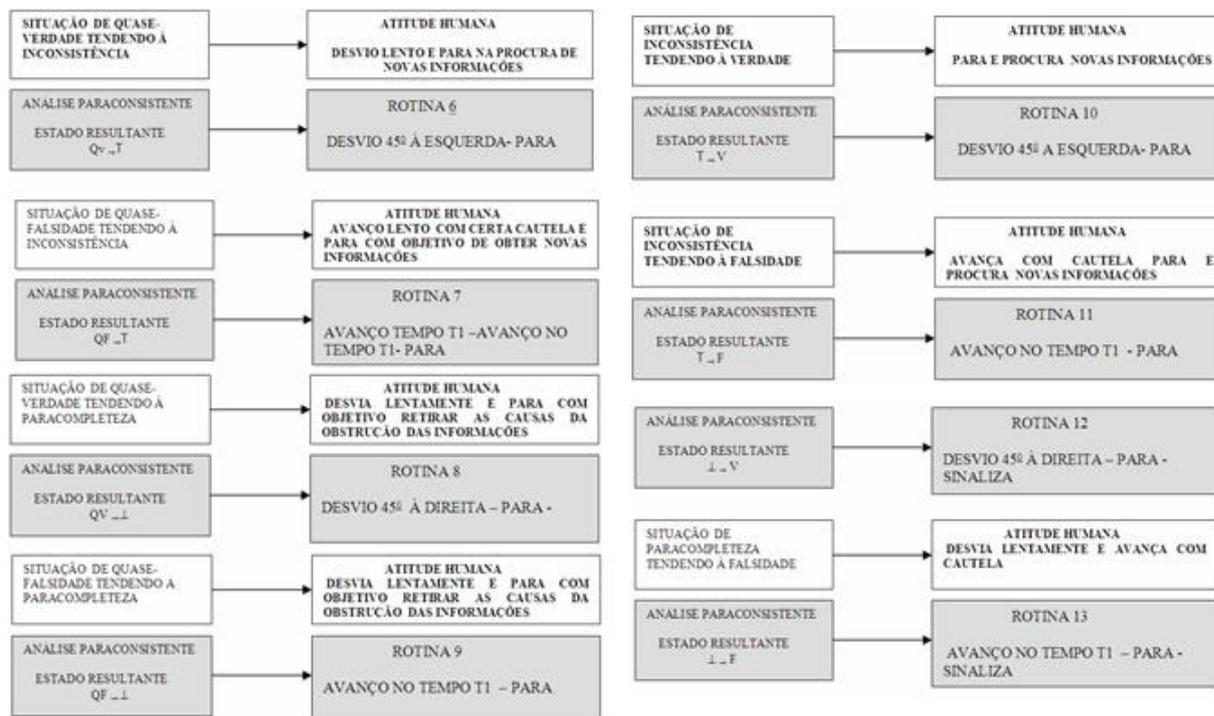


Figura 24 - Decisão Emmy para estados extremos - Fonte (DA SILVA FILHO, ABE &TORRES, 2006).

Para os estados lógicos “não extremos”, o comportamento do robô Emmy define as ações descritas na figura abaixo:



**Figura 25 - Decisão Emmy para estados "não extremos" –Fonte (DA SILVA FILHO, ABE & TORRES, 2006).**

No intuito de demonstrar a uso das classes reutilizáveis descritas no item 2.4 foi implementado um simulador do comportamento decisório do robô Emmy. As classes foram utilizadas de maneira a poder demonstrar as variações provocadas pelos sensores no processo decisório e ilustrar o funcionamento do controlador lógico Paraconsistente.

Nas figuras 26, 27, 28, 29 podem ser visualizadas a representação para os estados lógicos extremos, incluindo saídas analógicas e discreta com o respectivo tratamento para cada situação.

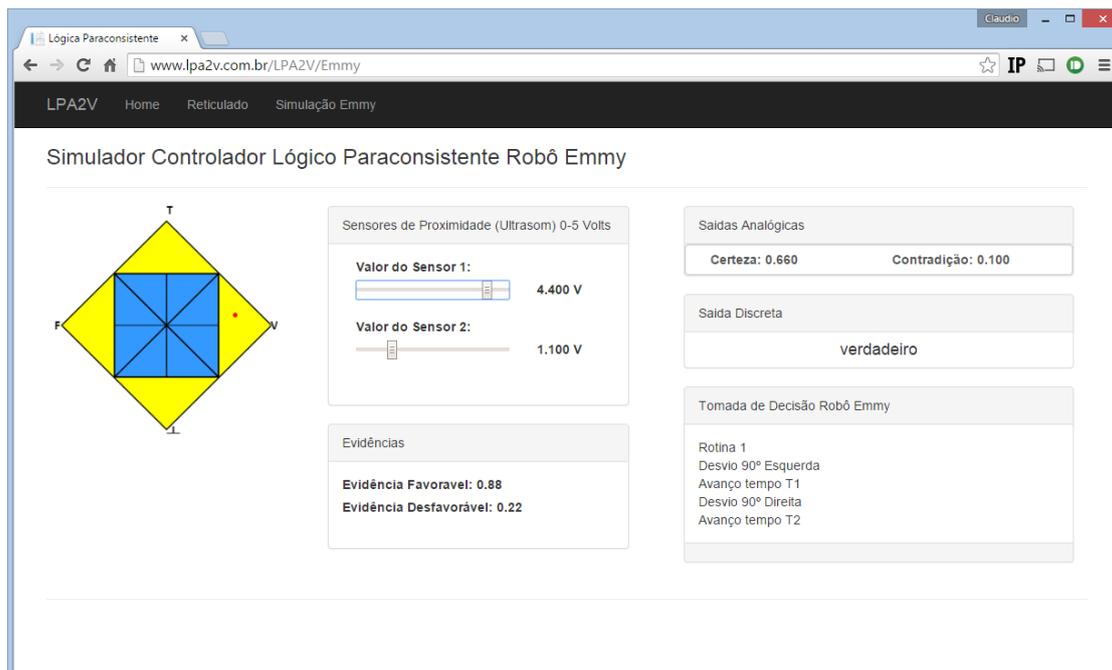


Figura 26 - Simulador Emmy - estado verdadeiro - [www.lpa2v.com.br](http://www.lpa2v.com.br)

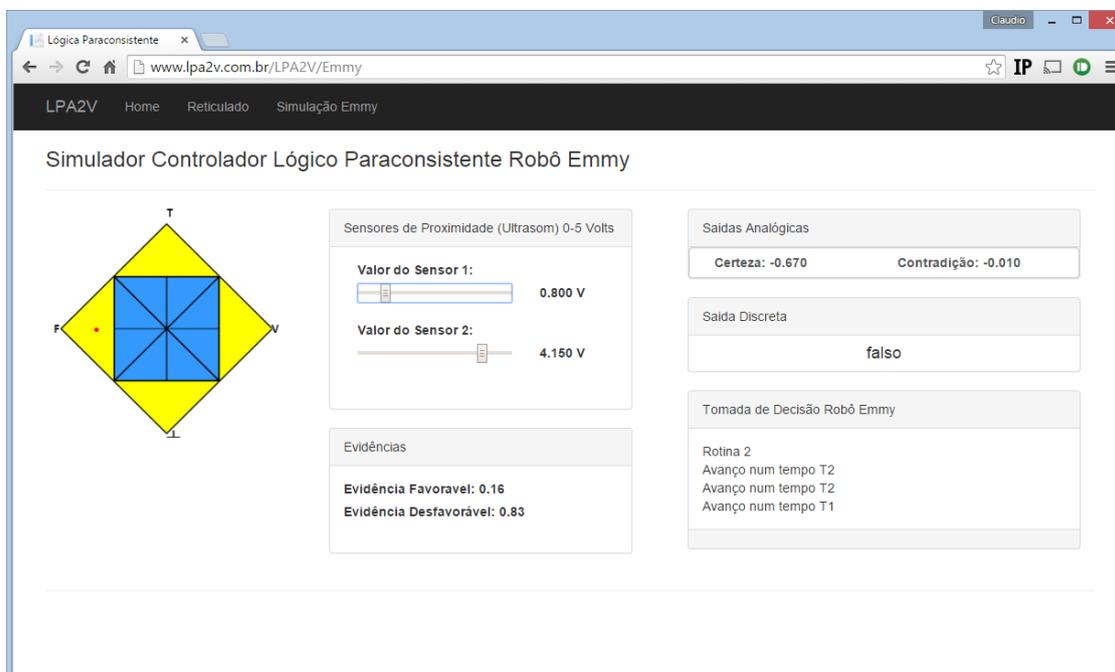


Figura 27 - Simulador Emmy - estado falso – [www.lpa2v.com.br](http://www.lpa2v.com.br)

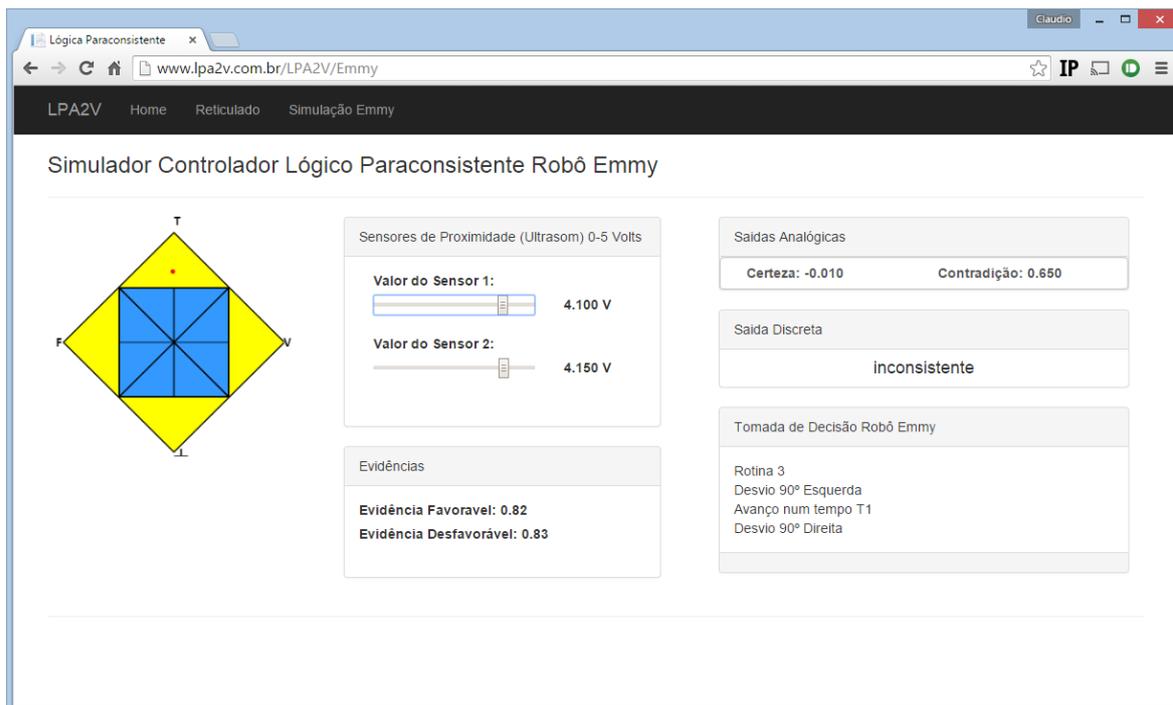


Figura 28 - Simulador Emmy - estado inconsistente - [www.lpa2v.com.br](http://www.lpa2v.com.br)

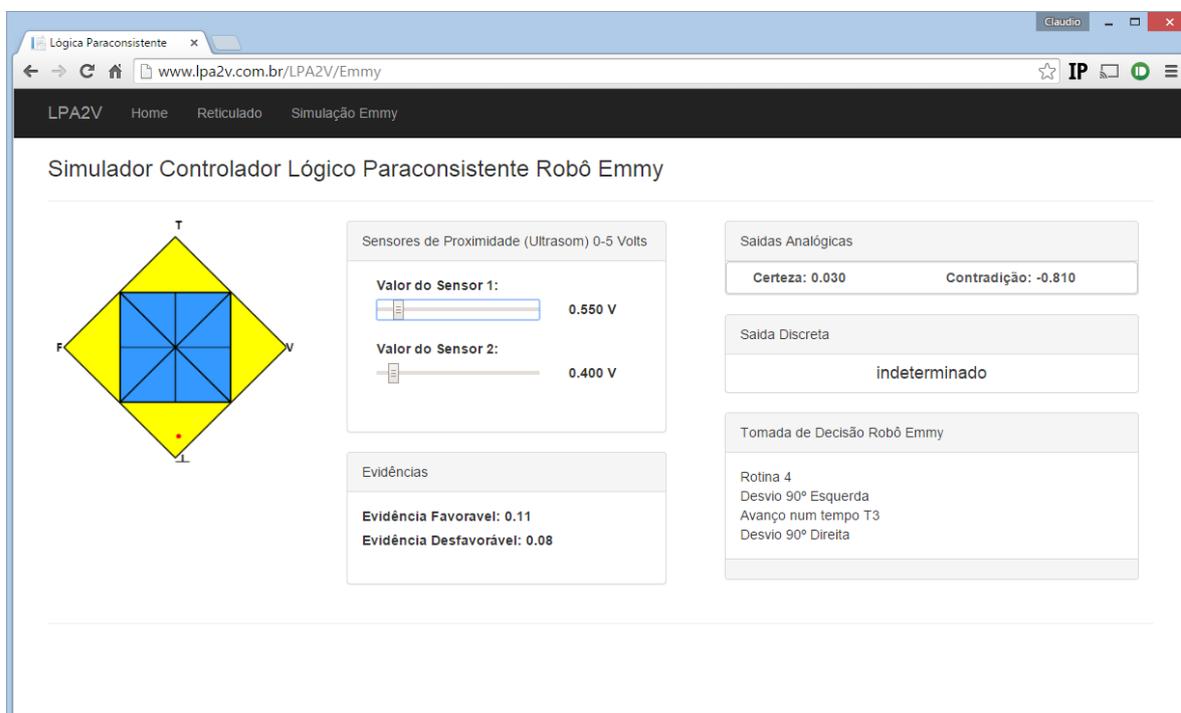


Figura 29 - Simulador Emmy - estado indeterminado - [www.lpa2v.com.br](http://www.lpa2v.com.br)

#### 4. CONCLUSÕES

Nos objetivos identificados para este trabalho constavam como metas primárias a criação de uma ferramenta *online* para criação de reticulado interativo e a criação de uma biblioteca reutilizável em *Javascript*. Esta biblioteca foi concebida para facilitar a criação de simulações utilizando a lógica Paraconsistente, especialmente a lógica Paraconsistente anotada de dois valores (LPA2v). O uso efetivo da biblioteca como auxiliar em criação de simuladores que permitam interagir com o comportamento do controlador lógico Paraconsistente, foi demonstrado através da construção de um simulador para o processo decisório do robô Emmy. Como o objeto deste trabalho se encontra disponível na *internet*, em sitio próprio, foi possível verificar que as premissas foram atendidas. A proposta deste trabalho, porém não se findou neste projeto. Acredita-se que a partir da divulgação do sitio, poderão ser obtidas novas contribuições, expandindo e estendendo a proposta original. Algumas das possibilidades estão descritas no item 5 do presente trabalho.

#### 5. TRABALHOS FUTUROS

A presente dissertação tem como parte da sua proposta a criação e disponibilização de sitio na internet ([www.lpa2v.com.br](http://www.lpa2v.com.br)) implementando um reticulado interativo. A partir da disponibilização do repositório contendo o código fonte, abre-se a possibilidade de obter colaboração de terceiros e estender as capacidades da biblioteca gerada para a criação do site.

Entre as muitas possibilidades a serem exploradas para extensão da biblioteca de código é possível elencar:

- Implementação de algoritmos alternativos para obtenção dos estados lógicos paraconsistentes.
- Modelagem e implementação de algoritmos da Lógica Paraconsistente Anotada de 4 valores (LPA4v).
- Métodos para representação gráfica no Quadrado Unitário no Plano Cartesiano (QUPC).
- Criação de classes modelando o funcionamento da Célula Neural Artificial Paraconsistente (CNAP).

Outra possível linha de continuidade para o presente trabalho incluem a criação e bibliotecas de código para serem utilizadas em projetos do tipo “desktop” utilizando linguagens de programação orientadas a objeto tais como *Java* ou *C#*.

## REFERÊNCIAS

DA SILVA FILHO, J. I. **Fundamentos das Redes Neurais Artificiais Paraconsistentes**. São Paulo: Arte & Ciência, 2000. 188p p.

DA SILVA FILHO, J. I. “**Introdução ao conceito de estado Lógico Paraconsistente  $\epsilon T$ .**”, Artigo Técnico, Revista Seleção Documental, n.17, ISSN 1809-0648 – Ed. Paralogike – Santos – SP – Brasil, 2010 pp 20-24.

DA SILVA FILHO, J. I.; ABE, J. M. **Introdução a Lógica Paraconsistente Anotada com Ilustrações**. 1ª. ed. Santos: Emmy, 2000.

DA SILVA FILHO, J. I.; ABE, J. M.; LAMBERT-TORRES, G. **Inteligência Artificial com as Redes de Análises Paraconsistentes**. 1ª. ed. Rio de Janeiro: LTC, 2008.

DA SILVA FILHO, J.I.; ABE, J.M.; Torres, C. R., “**Robô Móvel Autônomo Emmy: Uma Aplicação eficiente da Lógica Paraconsistente Anotada.** ”, Artigo Técnico, Revista Seleção Documental, n. 3 – Ano 1 ISSN 1809-0648 – Ed. Paralogike – Santos – SP – Brasil, 2006 pp 19-26.

D'ANTONIO, V. J. A. “**Método de Avaliação Quantitativa de Grânulos Plásticos (Pellets) em Praias Utilizando Análise Baseada em Lógica Paraconsistente Anotada**”, Dissertação de Mestrado em Sustentabilidade de Ecossistemas Costeiros Marinhos, Unisanta. Santos. 2013. 69p

FERNANDES, C. L. M. “**Lógica Paraconsistente Aplicada em Sistemas de Automação e Controle**”, Dissertação de Mestrado em Engenharia Mecânica, Unisanta, 2012, 109p.

FERRARA L.F., CAMARGO, J.M., ROCCO A., DA SILVA FILHO, J.I., ONUKI, A.S., SANTOS, M.R., LAMBERT-TORRES, G., “**Para-Previsor LPA2v: Um Sistema Computacional Previsor de Perfil de Cargas em Sistemas Elétricos de Potência construído com os fundamentos da Lógica Paraconsistente Anotada**”, Artigo Técnico, Revista Seleção Documental, n. 21 – Ano 6 ISSN 1809-0648 – Ed. Paralogike – Santos – SP – Brasil, 2011 pp 11-17.

FREEMAN, A. **Pro ASP.NET MVC 5**. New York: Springer, 2014

GEARY, D. M. **Core HTML5 Canvas: Graphics, Animation and Game Development**. New Jersey: Prentice Hall, 2012.

GENSLER, H. J. **Introduction to Logic**. New York: Routledge's, 2010.

GRAHAM, P. **An Introduction to Non-Classical Logic**. 2ª. ed. Cambridge: Cambridge University Press, 2008.

KRAUSE, D. Lógica Paraconsistente. **Scientific American Brasil**, p. 69-77, Novembro 2004.

LEMES NETO, M. C.; VENSON, N. Lógica Paraconsistente.

**ESCOLA DE CURSOS EDUCAONLINE**, 2015. Disponível em: <[http://educaonline.eng.br/UNISANTA/HTML/DOWNLOAD/LIVRO/LPA/LP\\_Nerio\\_Mauricio.pdf](http://educaonline.eng.br/UNISANTA/HTML/DOWNLOAD/LIVRO/LPA/LP_Nerio_Mauricio.pdf)>. Acesso em: 20 jan. 2015.

MARTIN, J.; BANKS, R. **Visual Studio 2013 Cookbook**. Birmingham: Packt Publishing Ltd., 2014.

SHENOY A.; SOSSOU, U. **Learning Bootstrap**. Birmingham: Packt Publishing Ltd., 2014

SUBRAHMANIAN, V.S., “**On the semantics of quantitative Logic Programs**”, Proc. 4th. IEEE Symposium on Logic Programming, Computer Society, press, Washington D.C., 1987.

## APÊNDICE I – CÓDIGO BIBLIOTECA LPA2v.JS

```
// define constantes para facilitar o uso do valor do estado logico resultante do algoritmo para
// analisador
```

```
lpa2vEstado_ naoatribuido = -1;
lpa2vEstado_falso = 0;
lpa2vEstado_verdadeiro = 1;
lpa2vEstado_indeterminado = 2;
lpa2vEstado_inconsistente = 3;
lpa2vEstado_inconsistenteFalso = 4;
lpa2vEstado_inconsistenteVerdadeiro = 5;
lpa2vEstado_indeterminadoFalso = 6;
lpa2vEstado_indeterminadoVerdadeiro = 7;
lpa2vEstado_quaseFalsoInconsistente = 8;
lpa2vEstado_quaseFalsoIndeterminado = 9;
lpa2vEstado_quaseVerdadeiroInconsistente = 10;
lpa2vEstado_quaseVerdadeiroIndeterminado = 11;
```

```
// reticulado paraconsistente
```

```
function reticulado(canvas) {
  this.canvas = canvas;
  this.ctx = canvas.getContext('2d');
  this.xoffset = (canvas.width / 2);
  this.yoffset = (canvas.height / 2);
  this.ctx.translate(this.xoffset, this.yoffset);
  this.borda = 20;
  this.flValoresVertices = false;
  this.flEstadosLogicos = true;
  this.flEixos = false;
  this.flColorido = true;
  this.flLimiteEstados = false;
  this.flVisivel = true;

  this.drawReticulado = function () {
    var ctx = this.ctx;
    var borda = this.borda;

    ctx.translate(-this.xoffset, -this.yoffset);
    ctx.clearRect(0, 0, this.canvas.width, this.canvas.height);
    ctx.translate(this.xoffset, this.yoffset);
    var x = this.xoffset - borda;
    var y = this.yoffset - borda;
    ctx.lineWidth = 3;
    ctx.strokeStyle = 'Black';

    ctx.beginPath();
```

```

    ctx.moveTo(x, 0);
    ctx.lineTo(0, -y);
    ctx.lineTo(-x, 0);
    ctx.lineTo(0, y);
    ctx.lineTo(x, 0);
    ctx.stroke();
    if (this.flColorido) {
        ctx.fillStyle = 'Yellow';
        ctx.fill();
    }
    ctx.closePath();
    if (this.flEstadosLogicos) {
        this.drawEstadosLogicosVertices();
    };
    if (this.flValoresVertices) {
        this.drawValoresVertices();
    }
    if (this.flEixos) {
        this.drawEixos();
    }
};

this.drawLpa2v = function (lpa2v) {
    if (this.flVisivel) {
        var ctx = this.ctx;
        var borda = this.borda;
        var x = lpa2v.grauCerteza() * (this.xoffset - borda);
        var y = (lpa2v.grauContradicao() * (this.yoffset - borda)) * (-1);
        if (this.lastx != -1) {
            ctx.restore();
        }
        ctx.save();
        if (this.flLimiteEstados) {
            this.drawLimiteEstados(lpa2v);
        }
        ctx.beginPath();
        ctx.fillStyle = '#FF0000';
        ctx.strokeStyle = '#FF0000';
        ctx.arc(x, y, 2, 0, 2 * Math.PI);
        ctx.stroke();
        ctx.moveTo(x, y);
        ctx.fill();
        ctx.closePath();
    }
}

this.drawEstadosLogicosVertices = function () {
    var ctx = this.ctx;

```

```

var x = this.xoffset;
var y = this.yoffset;
var borda = this.borda;
ctx.fillStyle = 'Black';
ctx.font = 'bold 12pt Calibri';
ctx.fillText("V", x - borda, 5);
ctx.fillText("T", 0, -y + (borda / 2));
ctx.fillText("F", -x + (borda / 2), 5);
ctx.fillText("⊥", 0, y - (borda / 2));
}

this.drawValoresVertices = function () {
  var ctx = this.ctx;
  var x = this.xoffset;
  var y = this.yoffset;
  var borda = this.borda;
  ctx.fillStyle = 'Black';
  ctx.font = 'bold 8pt Calibri';
  ctx.fillText("(1,0)", x - borda, 5);
  ctx.fillText("(1,1)", 0, -y + (borda / 2));
  ctx.fillText("(0,1)", -x, 5);
  ctx.fillText("(0,0)", 0, y - (borda / 2));
}

this.drawEixos = function () {

  var ctx = this.ctx;
  var borda = this.borda;
  var x = this.xoffset - borda;
  var y = this.yoffset - borda;

  ctx.strokeStyle = 'Black';
  ctx.lineWidth = 1;
  ctx.beginPath();
  ctx.moveTo(x, 0);
  ctx.lineTo(-x, 0);
  ctx.stroke();
  ctx.fillStyle = 'Dotted';
  ctx.moveTo(0, y);
  ctx.lineTo(0, -y);
  ctx.stroke();
  ctx.closePath();
}

this.drawLimiteEstados = function (lpa2v) {

  var ctx = this.ctx;
  var borda = this.borda;
  var x = this.xoffset - borda;
  var y = this.yoffset - borda;

```

```

var x = lpa2v.limiteInferiorCerteza * (this.xoffset - borda);
var y = (lpa2v.limiteSuperiorContradicao * (this.yoffset - borda)) * (-1);
var larg = (lpa2v.limiteSuperiorCerteza - lpa2v.limiteInferiorCerteza) * (this.xoffset -
borda);
var alt = (lpa2v.limiteSuperiorContradicao - lpa2v.limiteInferiorContradicao) *
(this.yoffset - borda);

ctx.beginPath();

ctx.fillStyle = '#3399FF';
ctx.strokeStyle = 'Black';
ctx.lineWidth = 1;

ctx.rect(x, y, larg, alt);

if (this.flColorido) {
  ctx.fill();
}
ctx.stroke();
ctx.moveTo(x, y);
ctx.lineTo(x + larg, y + alt);
ctx.stroke();

ctx.moveTo(x + larg, y);
ctx.lineTo(x, y + alt);
ctx.stroke();

ctx.moveTo(x + larg / 2, y);
ctx.lineTo(x + larg / 2, y + alt);
ctx.stroke();

ctx.moveTo(x, y + alt / 2);
ctx.lineTo(x + larg, y + alt / 2);
ctx.stroke();
ctx.closePath();

}

this.exportaReticulado = function () {
  var dataUrl = this.canvas.toDataURL();
  window.open(dataUrl, "toDataURL() image", "width=" + this.canvas.width.toString() +
", height=" + this.canvas.height.toString());
}

```

```
}

```

```
// algoritmo para analisador simulando controlador lógico para consistente
function lpa2V(grauEvidenciaFavoravel, grauEvidenciaDesfavoravel) {

  this.limiteInferiorCerteza = -0.5;
  this.limiteSuperiorCerteza = 0.5;
  this.limiteInferiorContradicao = -0.5;
  this.limiteSuperiorContradicao = 0.5;
  this.grauEvidenciaFavoravel = grauEvidenciaFavoravel;
  this.grauEvidenciaDesfavoravel = grauEvidenciaDesfavoravel;
  this.grauCerteza = function () { return this.grauEvidenciaFavoravel -
this.grauEvidenciaDesfavoravel; }
  this.grauContradicao = function () { return (this.grauEvidenciaFavoravel +
this.grauEvidenciaDesfavoravel) - 1; }

  this.estadoLogicoDiscreto = lpa2vEstado_ naoatribuido;

  this.estadoLogico = function () {
    var retorno = "";
    // determinacao estados "extremos"
    if (this.grauCerteza() >= this.limiteSuperiorCerteza) {
      retorno = "verdadeiro";
      this.estadoLogicoDiscreto = lpa2vEstado_verdadeiro;
    }
    else if (this.grauCerteza() <= this.limiteInferiorCerteza) {
      retorno = "falso";
      this.estadoLogicoDiscreto = lpa2vEstado_falso;
    }
    }
    else if (this.grauContradicao() >= this.limiteSuperiorContradicao) {
      retorno = "inconsistente";
      this.estadoLogicoDiscreto = lpa2vEstado_inconsistente;
    }
    }
    else if (this.grauContradicao() <= this.limiteInferiorContradicao) {
      retorno = "indeterminado";
      this.estadoLogicoDiscreto = lpa2vEstado_indeterminado;
    }
    }
    // determinacao dos estados "não extremos"
    else if ((this.grauCerteza() >= 0 && this.grauCerteza() < this.limiteSuperiorCerteza) &&
      (this.grauContradicao() >= 0 && this.grauContradicao() <
this.limiteSuperiorContradicao)) {
      if (this.grauCerteza() >= this.grauContradicao()) {
        retorno = "quase verdadeiro tendendo a inconsistente";
        this.estadoLogicoDiscreto = lpa2vEstado_quaseVerdadeiroInconsistente;
      }
    }
  }
}
```

```

    }
    else {
        retorno = "inconsistente tendendo a verdadeiro";
        this.estadoLogicoDiscreto = lpa2vEstado_inconsistenteVerdadeiro;
    }
}
else if ((this.grauCerteza() >= 0 && this.grauCerteza() < this.limiteSuperiorCerteza) &&
        (this.grauContradicao() <= 0 && this.grauContradicao() >=
this.limiteInferiorContradicao)) {
    if (this.grauCerteza() >= Math.abs(this.grauContradicao())) {
        retorno = "quase verdadeiro tendendo a indeterminado";
        this.estadoLogicoDiscreto = lpa2vEstado_quaseVerdadeiroIndeterminado;
    }
    else {
        retorno = "indeterminado tendendo a verdadeiro";
        this.estadoLogicoDiscreto = lpa2vEstado_indeterminadoVerdadeiro ;
    }
}
else if ((this.grauCerteza() <= 0 && this.grauCerteza() > this.limiteInferiorCerteza) &&
        (this.grauContradicao() <= 0 && this.grauContradicao() >=
this.limiteInferiorContradicao)) {
    if (Math.abs(this.grauCerteza()) >= Math.abs(this.grauContradicao())) {
        retorno = "quase falso tendendo indeterminado";
        this.estadoLogicoDiscreto = lpa2vEstado_quaseFalsoIndeterminado;
    }
    else {
        retorno = "indeterminado tendendo a falso";
        this.estadoLogicoDiscreto = lpa2vEstado_indeterminadoFalso;
    }
}
else if ((this.grauCerteza() <= 0 && this.grauCerteza() > this.limiteInferiorCerteza) &&
        (this.grauContradicao() >= 0 && this.grauContradicao() <
this.limiteSuperiorContradicao)) {
    if (Math.abs(this.grauCerteza()) >= Math.abs(this.grauContradicao())) {
        retorno = "quase falso tendendo a inconsistente";
        this.estadoLogicoDiscreto = lpa2vEstado_quaseFalsoInconsistente;
    }
    else {
        retorno = "inconsistente tendendo a falso";
        this.estadoLogicoDiscreto = lpa2vEstado_inconsistenteFalso;
    }
}
}
return retorno;
}

```