

# APRENDENDO REGRAS POR OBSERVAÇÃO EM UMA CASA INTELIGENTE

JULIO ANDRÉ SGARBI, FLAVIO TONIDANDEL

*Centro Universitário da FEI – UniFEI*  
*Av. Humberto de A. C. Branco, 3972*  
*09850-901 - São Bernardo do Campo – SP - Brasil*  
*E-mails: unimjsgarbi@fei.edu.br, flaviot@fei.edu.br*

**Abstract** — Many advances have happened in residential automation, however little importance has been given to residential automation based in the behavior of the inhabitants. This article describes a system whose objective is to observe and to learn rules in a house according to the behavior of its inhabitants. The system uses the concept of learning with induction rules, which it was already used in another system, known as ABC System (Automation based on behavior). In the considered system, called ABC+, the events observation window and the embryonic rules are presented among others advances.

**Keywords** — Residential Automation, Intelligent domotics, Learning, Artificial Intelligence.

**Resumo** — Muitos avanços têm acontecido em automação residencial, entretanto pouco destaque é dado à automação residencial baseada no comportamento dos habitantes. Este artigo descreve um sistema cujo objetivo é observar e aprender regras em uma casa de acordo com o comportamento de seus habitantes. O sistema utiliza o conceito de aprendizado com regras de indução, o qual já foi utilizado em outro sistema, conhecido como Sistema ABC (Automação baseada em comportamento). No sistema proposto, o Sistema ABC+, são apresentados, entre outros avanços, a janela de observação de eventos e as regras embrionárias.

**Palavras-chave** — Automação residencial, domótica inteligente, aprendizado, inteligência artificial.

## 1 Introdução

Existe uma série de trabalhos que abordam o tema automação residencial. Os avanços tecnológicos e a busca por conforto e segurança vêm fazendo com que a cada dia as residências tenham mais e melhores sistemas automatizadores.

Algumas vezes estes sistemas são implementados com técnicas de inteligência artificial, entretanto a maioria deles possui apenas mecanismos automáticos (Bolzani, 2004).

A principal ênfase destes sistemas está no gerenciamento de recursos, segurança, conforto e atualmente em entretenimento (Muratori, 2005).

Muitos dos trabalhos publicados alegam que seus sistemas são inteligentes. Alguns deles detectam eventos como a presença dos habitantes e suas ações, ou possuem sensores que conseguem capturar mudanças nas condições do ambiente; a estes eventos são aplicadas reações que estão previamente configuradas (Bolzani, 2004). O conceito de inteligência deve ir além de automatizar ou simplesmente aplicar regras pré-estabelecidas; os sistemas inteligentes devem interagir com os habitantes da casa e aprender com seus comportamentos.

Automação residencial tem de evoluir para o conceito de Domótica Inteligente, no qual se entende que os dados obtidos pelos sensores da casa devem ser analisados de modo a adaptar as regras de automação do ambiente ao comportamento dos habitantes (Tonidandel et al, 2004).

Os seres humanos estão em constante mudança; o que é uma regra ou rotina hoje, amanhã pode não

ser. Os hábitos, horários e atividades mudam com o passar do tempo. Os sistemas têm de aprender e se adaptar a isto.

Neste artigo será apresentado um sistema de automação residencial inteligente que cria regras para ação de atuadores em função do aprendizado com o comportamento dos habitantes de uma casa, dando ênfase à obtenção, tratamento e manutenção das regras.

O artigo é composto pela seção 2, que descreve resumidamente o sistema ABC (Automação baseada em comportamento); a seção 3 contém o algoritmo C4.5; seção 4, que detalha o novo sistema proposto, ABC+; a seção 5 aborda a possibilidade de uso da tecnologia RFID em casas inteligentes; a seção 6 apresenta a implementação do teste realizado e a seção 7 conclui o artigo.

## 2 Sistema ABC

O processo normal de criação de regras é aquele onde o habitante é quem cria as regras, inserindo-as em um sistema. O sistema ABC (Automação Baseada em Comportamento) (Tonidandel et al, 2004), o qual aprende regras em função do comportamento do habitante, foi testado e demonstrou através de simulações que é possível reverter o processo normal de criação de regras.

A arquitetura do sistema ABC (fig. 1), que será resumidamente descrita, define a existência de sensores (detectores de presença, medidores de temperatura, medidores de luminosidade, etc.), atuadores (interruptores de luz, ar-condicionado,

etc.), bancos de dados e demais elementos necessários para criação e controle das regras.

Para cada atuador da casa existe um banco de dados de aquisição de comportamento, o qual é alimentado com os seus eventos e os respectivos dados dos sensores vinculados ao atuador. Como exemplo, pode-se ter o atuador “Ar-condicionado” e os sensores “Temperatura”, “Luminosidade”, “Horário” e “Presença”. Quando o estado do Ar-condicionado muda, por ação do habitante, os dados do próprio atuador e mais os dados dos sensores são armazenados em uma linha do banco de dados de aquisição de comportamento (Ar-condicionado=Ligado; Temperatura=Alta; Luminosidade=Alta; Horário=Noite; Presença=Sim).

Quando o banco de dados atinge o valor configurado de eventos armazenados, o mesmo é inserido no algoritmo de aprendizado com árvores de indução ID3 (Quinlan, 1986), o qual generaliza os dados e cria regras. As regras são armazenadas no banco de dados de regras ativas. No exemplo citado, uma regra aprendida poderia ser: SE Temperatura=Alta E Horário=Noite E Luminosidade=Alta E Presença=Sim ENTÃO Ar-condicionado=Ligado. Assim, quando os sensores indicarem Temperatura=Alta, Horário=Noite, Luminosidade=Alta e Presença=Sim o ar-condicionado será automaticamente ligado pelo Sistema ABC. Ou seja, a partir do momento em que novos eventos acontecem, de acordo com as ações do habitante, é feita uma varredura no banco de dados de regras ativas para avaliar se alguma regra deve ser aplicada e realizar uma ação no atuador.

Existe outro banco de dados onde estão as regras de segurança, nele podem existir regras do tipo: Fogo=Sim ENTÃO Energia=Desligada.

A manutenção das regras é simples, quando uma regra fica sem ser utilizada por um valor de tempo pré-configurado, ela é removida. Também é possível remover regras manualmente pelo habitante.

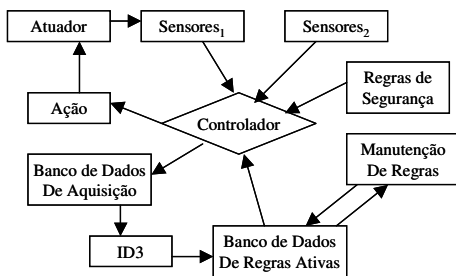


Figura 1. Arquitetura do Sistema ABC

Apesar de muito interessante, o sistema possui deficiências. Ele não detecta seqüências causais de eventos no tempo, se um evento de atuador acontece depois de pouco ou muito tempo de um evento de sensor isto não é considerado. Neste sistema, com o uso do ID3, é possível trabalhar somente com variáveis lógicas, não é possível trabalhar com valores contínuos. Outra deficiência é o fato de

regras criadas pelo ID3 se tornarem diretamente ativas, isto pode desagradar o habitante da casa.

### 3 O algoritmo C4.5

Uma diferença entre o sistema ABC e o sistema proposto é o algoritmo de aprendizado de regras utilizado. O sistema proposto utiliza o C4.5.

O algoritmo C4.5 (Quinlan, 1993) é uma extensão do ID3 que permite trabalhar com valores contínuos para os atributos, separando os possíveis resultados em duas partes: uma para aqueles  $A_i \leq N$  e outra para  $A_i > N$ , onde  $A_i$  é o  $i$ ésimo atributo e  $N$  é algum valor dentro dos limites de valores do atributo  $A_i$ . Lembrando que no exemplo do quarto na casa, os atributos são os dados dos sensores e atuador. A vantagem dos valores contínuos está no uso de medidores de temperatura, umidade e até mesmo nos valores dos horários, eliminando assim a restrição de valores discretos.

Este algoritmo foi proposto por Quinlan em 1993. O algoritmo C4.5 gera uma árvore de decisão a partir dos dados mediante partições realizadas recursivamente. A árvore é construída mediante a estratégia de busca em profundidade (*depth-first*) (Cormen et al, 2001). O algoritmo considera todos os atributos possíveis que podem dividir o conjunto de dados e seleciona o atributo que resulta no maior ganho de informação. Para cada atributo discreto, se considera um cálculo com  $n$  resultados, sendo  $n$  o número de valores possíveis que pode tomar o atributo. Para cada atributo contínuo, se realiza um cálculo binário sobre cada um dos valores que toma o atributo nos dados.

Outra diferença importante é a possibilidade de trabalhar com atributos desconhecidos, como no caso de alguma leitura de sensor indefinida, onde o valor fica desconhecido. Neste caso o C4.5 aproveita os dados e também consegue generalizar regras.

### 4 Sistema proposto: ABC+

O sistema ABC, em sua versão inicial, marcou a implementação de regras em automação residencial através da observação do comportamento de um habitante em uma casa, entretanto é possível observar que ele possui limitações que podem ser atenuadas.

Com o intuito de corrigir o sistema ABC e diminuir suas limitações, é proposto o sistema ABC+, o qual possui arquitetura (fig.2) parecida com o sistema inicial, porém a Rede de Petri é bastante diferente.

As principais diferenças são a janela de observação de eventos, as regras embrionárias, C4.5 (Quinlan, 1993) e o novo processo de manutenção das regras. Estas diferenças serão detalhadas na seqüência.

Um quarto de uma casa onde existem um sensor de entrada no ambiente, um sensor de saída do ambiente e um atuador para ligar e desligar uma lâmpada servirá de exemplo para entender as implementações feitas no sistema ABC+.

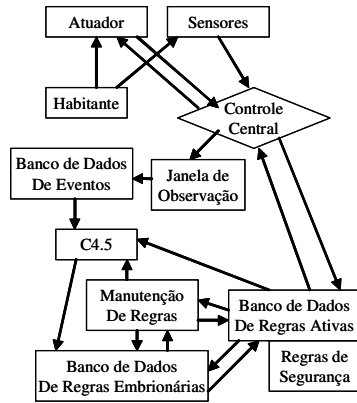


Figura 2. Arquitetura do Sistema ABC+

#### 4.1 A Janela de Observação

Nas condições em que o habitante entra no quarto e imediatamente acende a lâmpada ou quando sai do quarto e imediatamente desliga a lâmpada, pode-se facilmente assimilar regras, que seriam: SE habitante entra ENTÃO acenda lâmpada; SE habitante sai ENTÃO apague lâmpada.

Note que se o habitante entra no quarto e após uma hora ele acende a lâmpada, os sensores detectarão os mesmos dados da situação em que ele imediatamente acende a lâmpada após entrar no quarto. Isto leva a criação de uma regra errada.

Para eliminar esta limitação a janela de observação é utilizada. A janela consiste em se armazenar e comparar cada evento anterior e posterior ao evento em análise, inclusive com horários. Os eventos em análise são eventos de atuador, ou seja, eventos em que o atuador muda de estado, no exemplo a lâmpada. Eventos de sensores não são utilizados para gerar regras, mas sim para ativar uma regra, se a mesma existir. Na janela os eventos anterior e posterior podem ser eventos de sensores, pois servirão de comparação e não para criação de regra.

No exemplo, usando uma janela de observação configurada com valor de 5 segundos, quando o habitante entra no ambiente ele dispara um evento de sensor, que é armazenado temporariamente com seu devido horário. Após uma hora ele acende a lâmpada. Este evento de atuador é armazenado e comparado com o evento anterior, se o anterior aconteceu dentro de 5 segundos, então significa que o evento do atuador está vinculado ao evento anterior, neste caso o evento do atuador é armazenado para gerar uma futura regra. Quando existe vínculo com o evento anterior não se realiza a comparação com o posterior e o sistema volta a situação de início. Porém, se foi

constatado que não existe vínculo entre evento do atuador e o evento anterior, é necessário verificar o evento posterior. Novamente se compara a diferença de horários entre eles e se o posterior aconteceu dentro de 5 segundos, então significa que o evento do atuador está vinculado ao evento posterior, caso contrário não existe vínculo. No caso de estarem vinculados armazenam-se os dados do evento de sensor posterior para gerar futura regra, pois é este evento que reflete as reais condições para o atuador ter sido mudado.

#### 4.2 As Regras Embrionárias

No sistema ABC, sempre que o ID3 cria uma regra, ela é imediatamente colocada como regra ativa. Sendo assim, não existe uma etapa de validação da regra. A primeira vez que o habitante gerar as condições dos sensores iguais às condições da regra, a mesma será executada e alguma ação no atuador será feita. Se o habitante contrariar a regra nada acontece, e se ele quiser acabar com a regra terá de removê-la manualmente.

Ao criar uma regra, deve existir uma etapa de validação da mesma, ou seja, ter certeza que a regra é aceita pelo habitante. No sistema ABC+, toda vez que novas regras são criadas, elas são armazenadas inicialmente no banco de regras embrionárias.

As regras embrionárias são regras que estão em validação. Quando o habitante gerar condições dos sensores iguais às condições de uma regra embrionária, o que acontece é que a regra ganha pontos positivos. Somente quando a regra atinge uma certa pontuação superior é que ela passa para o banco de regras ativas.

Por outro lado, se o habitante contrariar a regra, ela ganha pontos negativos e ao atingir uma pontuação inferior a mesma é eliminada do banco de regras embrionárias.

#### 4.3 A Manutenção das Regras

A maneira como se mantém as regras em uma casa influencia a interação do sistema com o habitante.

A inserção e remoção de regras têm de ser o mais sutil possível, caso contrário trará desconforto ao usuário e podem incorrer em desestabilização das próprias regras.

Para o desenvolvimento e manutenção de regras no sistema ABC+ são utilizados três bancos de dados para cada atuador da casa (tab. 1, tab. 2 e tab. 3).

Evento	Sensor 1 (S1)	Sensor 2 (S2)	.....	Sensor N (Sn)	Atuador An
1					
2					
3					
4					
....					

Tabela 1. Eventos por Atuador (A1, A2, .....An)

Regra	Sensor 1 (S1)	Sensor 2 (S2)	.....	Sensor N (Sn)	Atuador An	ATIV	EXC
1							
2							
3							
4							
....							

Tabela 2. Regras Ativas por Atuador (A1, A2, .....An)

Regra	Sensor 1 (S1)	Sensor 2 (S2)	.....	Sensor N (Sn)	Atuador An	OK	NOK
1							
2							
3							
4							
....							

Tabela 3. Regras Embrionárias por Atuador (A1, A2, .....An)

Serão utilizadas as seguintes definições:

**BDEventos** – Banco de Dados de Eventos por Atuador.

**BDAtivas** – Banco de Dados de Regras Ativas.

**ATIV** – Campo do BDAtivas utilizado para pontuar positivamente uma regra.

**EXC** - Campo do BDAtivas utilizado para pontuar negativamente uma regra.

**BDEmbrio** – Banco de Dados de Regras Embrionárias.

**OK** - Campo do BDEmbrio utilizado para validar uma regra.

**NOK** - Campo do BDEmbrio utilizado para excluir uma regra.

Os novos eventos que surgem nos atuadores da casa, e passam pelo crivo da janela de observação, são armazenados no Banco de Dados de Eventos por Atuador.

Quando o banco de dados de eventos atinge um número pré-determinado de eventos ocorre a sua inserção no algoritmo C4.5 para a geração de novas regras.

Estas novas regras são inicialmente comparadas com as regras ativas e embrionárias, para evitar repetição.

As novas regras geradas, que não forem repetidas, vão para o banco de regras embrionárias. No Banco de Regras Embrionárias existem para cada regra, além dos campos de sensores e atuador, os campos OK e NOK, os quais servem para o controle da validação da regra. Quando uma regra embrionária é gerada ela possui os campos OK e NOK iguais a zero.

Sempre que ocorre um novo evento de atuador este é comparado primeiro com as regras ativas, para verificar se alguma regra ativa está sendo contrariada ou não, depois este evento é comparado com as regras embrionárias. Caso exista regra embrionária relacionada ao evento é necessário dar pontos à mesma. Se o evento confirmou a regra, ao valor do campo OK é somado mais um, caso o evento contrarie a regra o valor do campo NOK recebe mais um. Quando o campo OK atinge certo valor, a regra embrionária é transformada em regra ativa. Quando o

campo NOK atinge certo valor, a regra embrionária é removida do banco de regras.

De certa forma os campos OK e NOK trabalham como porcentagens de acertos e erros. Por exemplo, se o valor necessário no campo OK para a regra ser validada é oito e o valor no campo NOK para a regra ser excluída é dois, significa que quando temos 80% ou mais de acertos a regra é validada ou quando temos 20% ou mais de erros a regra é excluída.

O Banco de Regras Ativas também possui uma lógica de manutenção. É neste banco que estão as regras de segurança, as quais não sofrem modificações, e as regras a serem executadas na casa. Sempre que algum evento de sensores acontece o banco de regras ativas é consultado, se existir alguma regra relacionada ao evento a mesma é aplicada, assim uma ação é feita no atuador. Neste caso soma-se ao valor do campo ATIV o valor um.

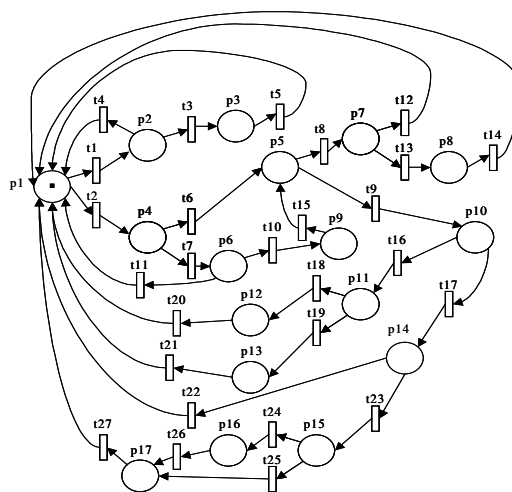
O banco de regras ativas possui uma quantidade limitada de regras. As regras estão ordenadas de acordo com o valor do campo ATIV. Quando uma nova regra é inserida no BDAtivas e este está com sua capacidade máxima, então é necessário retirar a regra com menor pontuação ATIV e colocar esta regra de volta ao banco de regras embrionárias.

Um evento de atuador também recebe atenção por parte do BDAtivas. Caso o evento atuador seja contrário a alguma regra existente no BDAtivas, soma-se ao valor do campo EXC desta regra em questão o valor um. Se o campo EXC ultrapassar um valor pré-determinado, esta regra é excluída do BDAtivas.

Um ponto importante na manutenção das regras é a criação de novas regras. Quando o C4.5 recebe o BDEventos para criar novas regras, uma função mais nobre é executada por ele. São inseridas no C4.5, além das informações do BDEventos, as regras constantes no BDAtivas. Com isto pode-se ter a generalização de algumas regras já existentes. Após a criação destas novas regras verifica-se se as mesmas já não constam em BDAtivas e BDEmbrio, para evitar duplicação de regras. Tal procedimento torna necessário que as regras BDEmbrio ao serem promovidas a BDAtivas sejam avaliadas com o objetivo de verificar se não são uma generalização de alguma regra existente, caso sejam as correspondentes regras do BDAtivas são eliminadas.

Na criação de novas regras também é checado se alguma regra do BDAtivas ficou velha. Para isto subtrai-se o valor um do campo ATIV de todas as regras. As regras que tiverem seu campo ATIV abaixo de um determinado valor serão rebaixadas ao BDEmbrio.

A Máquina de estado em Rede de Petri do sistema ABC+ (fig. 3) reflete as informações descritas nesta seção.



**LEGENDA**  
 EA=Evento Atuador  
 ES=Evento Sensor  
 BE=Banco de Eventos  
 RA=Regra Ativa  
 BRA=Banco de Regras Ativas  
 RE=Regra Embrionária  
 BRE=Banco de Regras Embrionárias

**POSIÇÕES**  
 P1 – Estado inicial de espera  
 P2 – Busca por RA existente  
 P3 – Realização da ação X, fazer  $ATIV=ATIV+1$ , com  $ATIV$  máximo=T e ordenação do BRA por  $ATIV$   
 P4 – Busca por ES anterior recente (R segundos)  
 P5 – Busca por RA existente  
 P6 – Busca por ES posterior recente (R segundos)  
 P7 – Avaliação se a regra é igual ou contrária  
 P8 – Fazendo  $EXC=EXC+1$ , se  $EXC>K$  a regra é excluída do BRA  
 P9 – Armazenamento dos dados do ES posterior  
 P10 – Busca por RE existente  
 P11 – Avaliação se a regra é igual ou contrária  
 P12 – Fazendo  $OK+OK+1$ , se  $OK=Z$  regra vira RA  
 P13 – Fazendo  $NOK=NOK+1$ , se  $NOK>Q$  regra é excluída do BRE  
 P14 – Colocação do evento no BE  
 P15 – C4.5 gera novas regras a partir de  $BE+BRA$   
 P16 – As regras repetidas são ignoradas  
 P17 – Regras antigas do BRE com  $OK<W$  são excluídas; coloca regras novas no BRE; para toda RA fazer  $ATIV=ATIV-1$ ; se RA tem  $ATIV<J$  coloca-a no BRE

**TRANSIÇÕES**  
 T1 – Novo evento de sensor  
 T2 – Novo evento de atuador  
 T3 – Existência de RA  
 T4 – Não existência de RA  
 T5 – Finalização da ação em P3  
 T6 – Existência de ES anterior  
 T7 – Não existência de ES anterior  
 T8 – Existência de RA  
 T9 – Não existência de RA  
 T10 – Existência de ES posterior  
 T11 – Não existência de ES posterior  
 T12 – A regra é igual  
 T13 – A regra é contrária  
 T14 – Finalização da ação em P8  
 T15 – Finalização da ação em P9  
 T16 – Existência de RE  
 T17 – Não existência de RE  
 T18 – A regra é igual  
 T19 – A regra é contrária  
 T20 – Finalização da ação em P12  
 T21 – Finalização da ação em P13  
 T22 – BE não têm Y eventos  
 T23 – BE têm Y eventos  
 T24 – Alguma(s) nova(s) regra(s) está(ão) em BRA ou BRE  
 T25 – Nenhuma nova regra está em BRA ou BRE  
 T26 – Finalização da ação em P16  
 T27 – Finalização da ação em P17

Figura 3. Máquina de estado em Rede de Petri do Sistema ABC+.

## 5 Usando RFID

Apesar das melhorias descritas, ainda existem alguns pontos a serem resolvidos.

O sistema descrito trabalha com um habitante na casa, ou melhor, considera que todos os habitantes se utilizam as mesmas regras. Isto não reflete a realidade em uma casa, pois as pessoas têm diferentes hábitos. Outra limitação está na situação em que um habitante da casa se aproxima da TV da sala e a liga. Como aprender esta ação?

Para estes casos uma possível solução é o uso da tecnologia RFID (*Radio Frequency Identification*).

Os sistemas RFID são sistemas automáticos de identificação que, usando sinais de rádio frequência, propiciam a identificação e localização automática de etiquetas eletrônicas portadoras de dados dos itens aos quais as mesmas estão relacionadas.

Um habitante que estiver com seu pijama e este contiver uma etiqueta RFID, poderá se aproximar da TV e ao ligá-la todas estas informações serão armazenadas para gerar futuras regras.

Do mesmo modo as roupas e pertences dos habitantes, contendo etiquetas RFID, poderão servir de identificador de cada habitante. O grande desafio é identificar os habitantes sem cadastramento de suas etiquetas RFID, ou seja, o sistema tem de ser bastante inteligente para associar os itens a cada habitante.

O uso de RFID em sistemas de domótica inteligente será deixado para trabalhos futuros.

## 6 Testes

Um simulador simples foi criado para verificar o sistema proposto. O ambiente simulado possui apenas um cômodo (quarto) com um atuador (lâmpada) e os seguintes sensores:

- Temperatura: Alta, Normal, Baixa;
- Luminosidade: Alta, Normal, Baixa;
- Umidade: Alta, Média, Baixa ;
- Porta: Entrada, Saída;

Para a implementação foi utilizado o *software* Delphi. O algoritmo C4.5 utilizado é o original (Quinlan, 1993), porém compilado para o Windows.

Para as variáveis R, T, K, Z, Q, Y, J e W presentes na Rede de Petri (fig. 3) foram fixados os seguintes valores:

- R= 5 (segundos, tempo da janela de observação);
- T= 20 (valor máximo do campo ATIV do BDAtivas);
- K= 3 (valor do campo EXC do BDAtivas para excluir a regra);
- Z= 5 (valor do campo OK do BDEmbrio para regra virar ativa);
- Q= 3 (valor do campo NOK do BDEmbrio para regra ser excluída);
- Y= 20 (número de eventos no BDEventos para acionar o C4.5);
- J= 2 (valor do campo ATIV do BDAtivas para regra ir para o BDEmbrio);
- W= 1 (valor do campo OK do BDEmbrio para a regra ser excluída por tempo).

O número máximo de regras em BDAtivas é dez, sendo que não foram utilizadas regras de segurança.

Os sensores podem ter seu estado alterado através de botões, conforme a figura 4. A simulação foi feita manipulando estes botões e consequentemente gerando eventos.

Um exemplo é clicar no valor Normal, do sensor Temperatura. Isto gera um evento de sensor que fica armazenado temporariamente, inclusive com horário, e pode ser visualizado no quadro um do simulador. Este evento pode: levar à execução de uma das regras ativas, as quais podem ser observadas no quadro quatro do simulador; atuar na validação ou rejeição de uma das regras embrionárias, as quais podem ser vistas no quadro três do simulador; servir de referência na janela de observação.

Outro exemplo é clicar no valor Apagada do atuador Lâmpada. Isto gera um evento de atuador, o qual passa pela lógica da janela de observação e se for validado é armazenado no banco de dados de eventos do atuador. No quadro dois do simulador é possível ver os eventos de atuador armazenados.

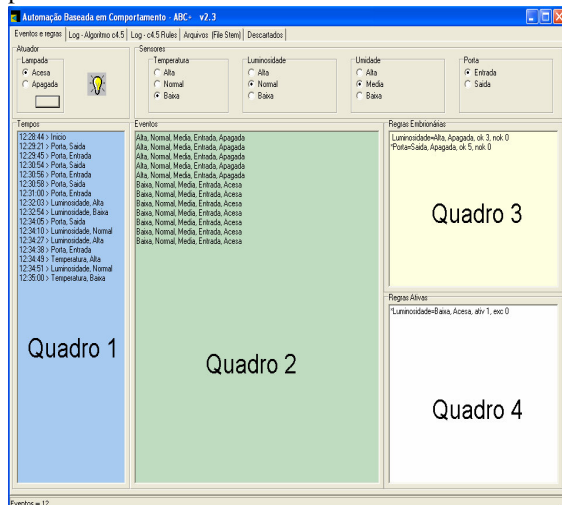


Figura 4. Simulador para teste do sistema ABC+ proposto.

O sistema teve o funcionamento esperado, criando regras novas e aplicando as mesmas em eventos posteriores. A janela de observação se comportou como na sua descrição, descartando eventos indesejados. O banco de regras embrionárias realmente atuou como um filtro para validar regras antes das mesmas serem ativadas, ou seja, também atuou como previsto.

## 7 Conclusões e Trabalhos Futuros

O artigo teve como objetivo apresentar um novo sistema de automação residencial, sistema ABC+, o qual aprende regras por observação do comportamento do habitante de um ambiente.

Os novos conceitos apresentados, a janela de observação e as regras embrionárias, são importantes para aperfeiçoar o processo de criação e início de uso das regras geradas.

Para que o sistema possa ser estendido para aplicações reais e mais complexas é necessário que ajustes sejam feitos. O primeiro deles é a melhor definição dos valores das variáveis R, T, K, Z, Q, Y, J e W da Rede de Petri, bem como o número de regras presentes no banco de regras ativas. Outro ponto a ser trabalhado é o uso da tecnologia RFID para que o sistema trabalhe com vários habitantes ao mesmo tempo. A presença de *loop* entre as regras ativas pode ser um grande problema, portanto também é um ajuste a ser desenvolvido.

Os ajustes identificados serão estudados e implementados em trabalhos futuros.

## Referências Bibliográficas

- Bolzani, C.A.M. (2004). **Desenvolvimento de um simulador de controle de dispositivos residenciais inteligentes : uma introdução aos sistemas domóticos**. São Paulo, Dissertação (mestrado), Universidade de São Paulo.
- Cormen, T.H., Leiserson C.E., Rivest R.L., Stein C. (2001). **Introduction to Algorithms. Second Edition**. MIT Press and McGraw-Hill. Section 22.3: Depth-first search, pp.540–549.
- Muratori, J.R. (2005). **As tendências do mercado de Automação Residencial**. São Paulo, Congresso Habitar - Congresso de Automação Residencial e Tecnologias para Habitação, 2005.
- Quinlan, J.R. (1986). **Induction of Decision Trees**. Machine Learning, Capítulo 1, p.81-106. Morgan Kaufmann, 1990.
- Quinlan, J.R. (1993). **C4.5: Programs for Machine Learning**. San Mateo, California, EE.UU, Morgan Kaufmann Publishers, 1993.
- Tonidandel, F., Takiuchi, M., Melo, E. (2004). **Domótica Inteligente: Automação baseada em comportamento**. Congresso Brasileiro de Automática, 2004.