

# Terceiro Trabalho de Inteligência Artificial: Sistema Baseado em Conhecimento

Gabriel Ferrari Cipriano <sup>1</sup>

Ufes — Departamento de Informática <sup>2</sup>

---

## Abstract

Este trabalho tem como objetivo construir um sistema baseado em Conhecimento que identifica animais utilizando a biblioteca *Experta*<sup>1</sup> da linguagem de programação *python*, e analisar sua implementação, redabilidade e manutenabilidade quando comparado com sistemas convencionais.

*Keywords:* Sistemas baseados em conhecimento

---

## 1. Introdução

Um Sistema Baseado em Conhecimentos, também chamado de **SBC**, é um sistema que se inspira no processo de raciocínio humano para resolver problemas complexos, os quais geralmente precisa-se de um especialista para resolver[1]. Com ênfase no conhecimento do domínio do problema, busca-se *'extrair'* o conhecimento dos especialistas, implementando-o em uma linguagem de representação[2].

Seus módulos básicos são uma *base de conhecimento* onde estão os fatos do domínio, e um *mecanismo de inferência* onde mais conhecimentos (fatos) podem ser gerados de acordo com as regras de inferência[2]. Neste trabalho também foram utilizados módulos complementares como o de *Aquisição do Conhecimento* que adiciona mais conhecimentos para a *memória de trabalho* por meio de uma *Interface* com usuário.

---

<sup>1</sup>gabriel.cipriano 'at' edu.ufes.br

<sup>2</sup>Ufes - Dep. de Informática <https://informatica.ufes.br/>

<sup>1</sup><https://experta.readthedocs.io/>

O *SBC* desenvolvido neste trabalho busca identificar animais por meio de  
15 fatos obtidos por *inputs* do teclado. De forma iterativa, o sistema realiza per-  
guntas pro usuário sobre as características que o animal pode ter, que podem  
ser respondidas com 'S' para *Sim* ou 'N' para *Não*. Lançando mão das regras de  
inferência implementadas, as características são associadas a grupos de animais,  
e posteriormente ao animal em específico.

20 O Sistema Baseado em Conhecimento implementado é capaz de identificar os  
seguintes animais: baleia, morcego, humano, urso, cão, tubarão, arraia, baiacu,  
atum, jacaré, cobra, tartaruga, camaleão, pinguim, gavião, beija-flor, gaivota,  
sapo e salamandra.

Na seção 2 apresentamos uma descrição do domínio e suas definições. Na  
25 seção 3 descreveremos as regras implementadas e suas particularidades, bem  
como as escolhas de implementação. Na seção 4 comparamos o *SBC* desenvol-  
vido com sistemas tradicionais, e descrevemos como seria adicionar um novo  
animal ao sistema.

## 2. Descrição do domínio

30 Na biologia, a Taxonomia é o campo científico que busca agrupar os seres  
vivos de acordo com suas características. A hierarquia do agrupamento bioló-  
gico pode ser vista na Figura 1. Quando é falado em saber classificar animais,  
significa saber identificar e diferenciar seres vivos pertencentes ao Reino Ani-  
mal. O Reino Animal é dividido em diversos *Filos* como *poríferos*, *cnidários*,  
35 *platelmintos*, *cordados*, entre outros[3].

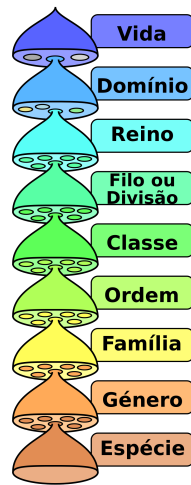


Figura 1: Hierarquia da classificação biológica. Fonte: Peter Halasz / Wikipedia

O filo dos *Cordados* são todos os animais que possuem tubo nervoso dorsal durante a fase embrionária, possuindo três subfilos: os *vertebrados*, os *anfioxos* e os *tunicados*. Curiosamente, todos os animais propostos para o *SBC* implementado possuem vértebra, pertencendo assim ao subfilo dos *vertebrados*. Dentro  
40 do subfilo dos *Vertebrados* o agrupamento dos animais é dado por *Classes*, compreendendo os peixes, anfíbios, répteis, aves e mamíferos. Com isso exposto, os animais propostos são distribuídos nas seguintes classes:

- Mamíferos
  - Humano
  - Baleia
  - Morcego
  - Urso
  - Cachorro
- Aves
  - Gavião
  - Pinguim

- Beija-flor
- Gaivota
- Peixes
  - 55    – Tubarão
  - Arraia
  - Atum
  - Baiacu
- Répteis
  - 60    – Jacaré
  - Cobra
  - Tartaruga
  - Camaleão
- anfíbios
  - 65    – Sapo
  - Salamandra

### 3. Descrição das regras utilizadas

Como são poucos representantes de cada Classe, realizar segregações ainda mais específicas como por *subclasse*, *clado* e *ordem* poderiam levar a dois efeitos  
 70 colaterais: (1) Uma gama maior de perguntas serão necessárias para diferenciar os animais. (2) Aumento de complexidade do sistema e de suas definições de regras, prejudicando a legibilidade do código. No código 1 é possível ver os fatos definidos para o domínio: **Classe**, **Animal** e **Feature**.

O que o SBC implementado faz é inicialmente perguntar características que  
 75 permitem identificar a classe do animal. A estratégia de raciocínio é fundamental num SBC para chegar a conclusões relevantes de maneira mais rápida[2], por

isso a nossa Base de Conhecimento utiliza do conhecimento procedural para priorizar regras de inferência. Por meio do parâmetro *Saliency* do *Experta*, é possível definir níveis de prioridade para as regras. O código 2 lista os tipos de prioridades definidos.

Código 1: Fatos do domínio

```
class Classe(Fact):  
    pass  
class Feature(Fact):  
    pass  
class Animal(Fact):  
    pass
```

Código 2: Aliases de prioridade

```
HIGHEST = 3  
HIGH = 2  
MEDIUM = 1  
DEFAULT = 0  
LOWEST = -1
```

A abordagem escolhida foi então criar uma *engine de conhecimento* que identifica a classe a qual o animal pertence, denominada ***IdentificaClasseAnimal***. Um exemplo de execução dessa *engine* pode ser vista abaixo no código 3.

Código 3: Execução *IdentificaClasseAnimal*

```
engine_classe = IdentificaClasseAnimal()  
engine_classe.reset()  
engine_classe.run()  
-----  
>> 0 animal possui penas?(S/N) N  
>> 0 animal possui brânqueas (Guelras)?(S/N) N  
>> 0 animal é amamentado?(S/N) S  
>> 0 seu animal é um(a) Mamífero.
```

Em seguida foi criada a *engine* ***IdentificaAnimal***, uma subclasse da *engine* ***IdentificaClasseAnimal***, que primeiro identifica a classe a qual o animal pertence, e em seguida faz perguntas que buscam distinguir o animal dentre os outros indivíduos de sua classe.

Muitas vezes é de interesse saber uma mesma característica para classes

100 distintas, por isso foram criadas algumas Regras que realizam uma mesma pergunta ao tentar identificar características de animais de classes diferentes. O Código 4 mostra uma dessas regras, onde a característica **Voa** é pertinente tanto para diferenciar Mamíferos, cujo morcego é o único capaz de voar, quanto pra diferenciar aves, onde pinguim é a única ave que não voa do nosso sistema.

Código 4: Regra de interesse duplo.

```
105 @Rule(AND( OR(Classe("Ave"), Classe("Mamífero")),
              NOT(Animal(W()))), salience=MEDIUM)
110 def ask_voa(self):
    self.declare(Feature(voa=ask_feature("é capaz de voar?")))
```

Como dito anteriormente, para diferenciar os indivíduos são realizadas perguntas com respostas binárias (sim ou não). Tomando  $n$  como o número de indivíduos de uma determinada classe, no melhor caso precisamos de fazer  $\lceil \log_2(n) \rceil$  perguntas para diferenciar os indivíduos.

115 Para as classes *Réptil* e *Peixe*, que possuem quatro indivíduos cada, bastaram duas perguntas para diferenciar os indivíduos de uma classe entre si. A grosso modo, as tabelas de características dos animais dessas classes ficaram muito parecidas com uma *Tabela-Verdade*:

	Aquático/Semi-aquático	Carnívoro
<i>Jacaré</i>	Sim	Sim
<i>Tartaruga</i>	Sim	Não
<i>Cobra</i>	Não	Sim
<i>Camaleão</i>	Não	Não

Tabela 1: Fatos sobre as características dos répteis presentes no *SBC*.

	Forma alongada	Possui ossos
<i>Atum</i>	Sim	Sim
<i>Tubarão</i>	Sim	Não
<i>Baiacu</i>	Não	Sim
<i>Arraia</i>	Não	Não

Tabela 2: Fatos sobre as características dos peixes presentes no *SBC*.

A morfologia dos peixes foi uma ótima característica para ser levada em con-  
 sideração, pois divide o grupo de peixes entre os que possuem forma fusiforme,  
 120 isso é, corpo alongado (atum e tubarão), e os que não, pois o baiacu possui  
 forma globular e a arraia é achatada dorso-ventralmente[4], como é mostrado  
 na figura 2.

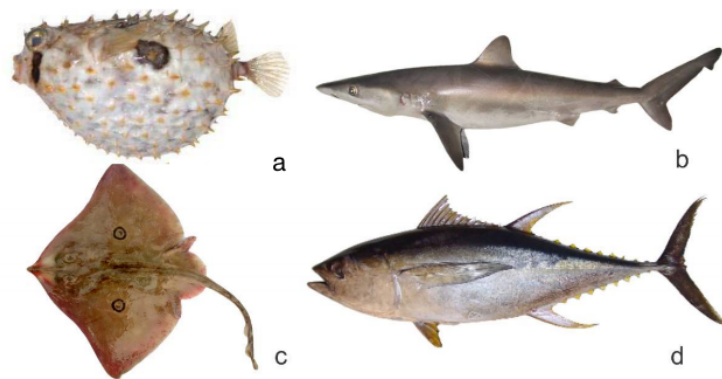


Figura 2: (a) baiacu, (b) tubarão, (c) arraia, (d) atum. Fonte: *Bemvenuti & Fischer*[4]

Para diferenciar os anfíbios basta uma característica: a presença de rabo,  
 125 pois salamandras possuem rabo, ao contrário dos sapos.

Para diferenciar as aves são levadas até três características em consideração:  
 Se é capaz de voar, se é carnívoro, e se é aquático/semi-aquático, característica  
 que diferencia gaviões de gaivotas.

A diferenciação dos mamíferos foi a que se mostrou a mais verbosa. Por ser  
 130 uma das classes mais diversas da natureza, muitos animais dessa classe possuem  
 uma única característica bem evidente que lhe difere do restante dos indivíduos,

como o morcego que é o único mamífero capaz de voar, e a baleia que é um dos únicos que vivem na água. Por conta disso, foram necessárias 4 perguntas para diferenciar os 6 mamíferos presentes na nossa base de conhecimento: Se é capaz  
135 de voar; se é carnívoro; se é aquático/semi-aquático; e se é selvagem (o que não é verdade para humanos e cachorros).

O código 5 mostra um exemplo de execução do *SBC* com a intenção de identificar uma tartaruga e a lista de fatos dessa construção do conhecimento.

Código 5: Execução do *SBC* desenvolvido.

```
140 engine = IdentificaAnimal()
run = ask("Olá! Vamos identificar um animal?")
while(run):
    engine.reset()
    engine.run()
145 run = ask("Deseja identificar outro animal?")
engine.facts

-----

>> Olá! Vamos identificar um animal?(S/N) S
>> O animal possui penas?(S/N) N
150 >> O animal é amamentado?(S/N) N
>> O animal bota ovos com casca rígida?(S/N) S
>> O animal é aquático ou precisa de viver perto da água?(S/N) S
>> O animal é carnívoro?(S/N) N
>> O seu animal é um(a) Tartaruga
155 >> Deseja identificar outro animal?(S/N) N
>> FactList([(0, InitialFact()),
              (1, Feature(possui_penas=False)),
              (2, Feature(amamenta=False)),
              (3, Feature(ovo_casca=True)),
160              (4, Classe('Réptil')),
              (5, Feature(aquatico_ou_semi=True)),
              (6, Feature(carnivoro=False)),
              (7, Animal('Tartaruga'))])
```



## 165 4. Conclusões

### 4.1. *Análise geral*

O Sistema Baseado em Conhecimento desenvolvido se destaca dos sistemas tradicionais principalmente em termos de legibilidade e manutenção do código. Para representar a mesma base de conhecimento desenvolvida, um sistema tra-  
170 dicional certamente seria uma árvore de decisão profunda de *if's* e *else's* enca-  
deados, onde o programador poderia se perder facilmente entre a indentação do código. Além disso, a manutenção de um sistema tradicional não seria trivial, possivelmente precisando alterar toda a estrutura da árvore de decisão para adicionar um simples animal a mais.

175 Por outro lado, o destaque para os sistemas tradicionais está definitivamente no desempenho, pois o módulo de inferência dos *SBCs* frequentemente precisam varrer o espaço do problema de maneira exaustiva. Por isso o conhecimento procedimental do domínio de um problema precisa ser muito bem representado numa implementação.

### 180 4.2. *Adicionando animais à base de conhecimento.*

Para demonstrar como a manutenção de *SBCs* é mais simples, será demonstrado quais os passos para se adicionar um novo animal à base de conhecimento: Um **leão**.

Primeiro, o programador/especialista deve olhar para o grupo o qual o leão  
185 faz parte, que é a classe dos mamíferos. O sistema já possui os seguintes mamíferos: Humano, Baleia, Morcego, Urso e Cachorro. Ao se deparar com as características desses animais no sistema, é fácil notar que o Urso é o animal que mais se aproxima do leão, ambos carnívoros selvagens.

A tarefa agora é identificar uma característica que diferencie os dois. A  
190 característica escolhida para diferenciar foi a capacidade de hibernar: O urso é um dos poucos animais conhecidos por hibernar no inverno.

Uma vez que esses conhecimentos foram identificados basta obter essa nova informação do teclado, modificar a regra de identificação do Urso colocando ape-

nas a capacidade de hibernar como regra, e adicionar uma regra para identificar  
195 o leão. O antes e depois desse trecho de código pode ser visto abaixo:

Código 6: Antes da adição do leão.

```
(...)  
@Rule(AND(Classe("Mamífero"),  
         Feature(selvagem=True),  
         Feature(carnivoro=True)),  
      salience=HIGHEST)  
def is_urso(self):  
    self.declare(Animal("Urso"))  
(...)
```

Código 7: Após adição do leão.

```
# Nova característica: hiberna.  
@Rule(AND( Classe("Mamífero"),  
          NOT(Animal(W())) ))  
def ask_hiberna(self):  
    self.declare(Feature(hiberna=  
ask_feature("hiberna?")))  
  
# Nova regra para urso, bastando hibernar  
@Rule(AND(Classe("Mamífero"),  
         Feature(hiberna=True)),  
      salience=HIGHEST)  
def is_urso(self):  
    self.declare(Animal("Urso"))  
  
# Leão é parecido com urso,  
# mas não hiberna.  
@Rule(AND(Classe("Mamífero"),  
         Feature(selvagem=True),  
         Feature(carnivoro=True),  
         Feature(hiberna=False)),  
      salience=HIGHEST)  
def is_leao(self):  
    self.declare(Animal("Leão"))
```

É importante salientar que à medida que animais forem sendo adicionados à  
base de conhecimento, a complexidade para diferenciar também vai aumentar,  
sendo assim necessário diferenciar os animais por outros tipos de fatos como  
200 *Filos, ordens e famílias.*

## 5. Referências

- [1] Wikipedia contributors, [Knowledge-based systems — Wikipedia, the free encyclopedia](https://en.wikipedia.org/wiki/Knowledge-based_systems), [Online; accessed 09-May-2021] (2021).  
205 URL [https://en.wikipedia.org/wiki/Knowledge-based\\_systems](https://en.wikipedia.org/wiki/Knowledge-based_systems)
- [2] F. M. Varejao, Curso de inteligência artificial - ufes, notas de aula (2021).
- [3] Wikipedia contributors, [Phylum — Wikipedia, the free encyclopedia](https://en.wikipedia.org/wiki/Phylum), [Online; accessed 09-May-2021] (2021).  
URL <https://en.wikipedia.org/wiki/Phylum>
- 210 [4] M. Bemvenuti, L. Gomes Fischer, Peixes: Morfologia e adaptações, Cadernos de Ecologia Aquática 5 (2011) 31–54.