

# Sistema de Diagnóstico de Pediatria

Ricardo Henrique Brunetto

RA94182

# Estrutura Geral

- Composto por
  - Interface Gráfica Web (golang)
  - Base de conhecimento (prolog)

# Interface Web


WS

# Interface Gráfica

- Interface Web (*html*)
- Escrita em Golang
- Comunica com swi-pl através de arquivos

# Procedimento

- Faz perguntas a respeito dos sintomas
- Apresenta **diagnósticos parciais** com base no que já foi informado



# Qual a idade do seu bebe?

☐ 1 ano ou menos

☐ Maior que 1 ano

Reiniciar

Próxima

Possível(is) Diagnóstico(s):

# Procedimento

- Armazena as respostas das perguntas em um arquivo
  - **sintomas.txt**
  - O arquivo é incrementado
- Chama o programa escrito em Prolog
  - Realiza o *matching*
- Lê o diagnóstico com base naqueles sintomas
  - **output.txt**



sintomas



output

```
Question{"Qual a idade do seu bebe?", RADIO_BUTTON, [string{"1 ano ou menos", "Maior que 1 ano"}, [string{"menos de 1 ano", "mais de 1 ano"}],
```

# Prolog

base\_de\_dados\_list.pl



# Estrutura e Entrada

- Baseado no predicado **doença(Doença, Sintomas)**
  - **Doença** é um átomo
  - **Sintomas** é uma lista
- Entrada por arquivos
  - **Sintomas** que compõe uma lista virão do arquivo **sintomas.txt**
  - Arquivo gerado pela interface Web

```
doenca('Gastroenterite', ['vomito', 'diarreia']).
```

sintomas.txt

```
1 'menos de 1 ano'.  
2 'febre'.  
3 'vomito'.
```

# Algoritmos

- Utiliza diretivas para inicialização
  - `:- main`
  - `:- halt` – (predicado nativo que encerra o programa)
- Predicado principal que executa o programa
  - `main`
- Algoritmo que faz o processamento da entrada
  - `diagnostico(Lista, Z)`
  - Recebe uma lista de sintomas (**List**a) e unifica na variável **Z**
- Escreve os resultados em arquivo
  - `write_answer(Lista)`
  - Escreve os elementos da **List**a no arquivo

```
:- main.  
:- halt.
```

# Algoritmos

- *Matching* baseado em subconjunto
  - **equal** é a regra que combina os sintomas com os da base de conhecimento.
  - **isSubset** é a heurística que determina se há compatibilidade entre as listas.

```
isSubset([], _).  
  
isSubset([X|L], S):-  
    member(X, S),  
    select(X, S, K),  
    isSubset(L, K).  
  
equal(X, Y):-  
    % isSubset(X, Y).  
    isSubset(Y, X).
```

# Saída

- Saída por arquivo
  - **output.txt**
  - Será lido pela interface Web e exibido ao usuário

```
output.txt
1 Gastroenterite
```

Possível(is) Diagnóstico(s):

Gastroenterite

# RESUMO

- A cada novo sintoma informado (pergunta ao usuário):
  - Interface web escreve no arquivo **sintomas.txt**
  - Arquivo em Prolog é executado e lê do arquivo **sintomas.txt**
  - O resultado é escrito no arquivo **output.txt**
  - Interface web lê o arquivo **output.txt** e exhibe ao usuário o diagnóstico

# Referências

- Material de apoio disponibilizado pelo Prof. Dr. Wagner Igarashi
- Swi-Prolog (<http://www.swi-prolog.org/>)
- Golang (<https://golang.org/>)
- Estilos CSS da Interface Web por Filipe Pina