

Alterações ao motor de inferência

Base de conhecimento: [Bc.pl](#)

Adições:

% Factos perguntáveis

Predicado: [perguntavel](#)

% Mapeamento de perguntas

Predicado: [pergunta](#), [texto_pergunta](#)

% Define as opções válidas para as respostas

Predicado: [opcoes_validas](#)

Motor de inferência: [motorInferencia.pl](#)

Alterações e adições:

% Verificações para usar o predicado avalia

Predicado: [verifica_condicoes](#) (2 novos predicados)

% Alteração do predicado criar factos para criar próxima perguntar, diagnostico e solução

Predicado: [cria_facto](#)

% Alteração do método avalia para usar os valores da base de conhecimento dinamicamente

Predicado: [avalia](#)

% Adição de vários predicados para mostrar vários factos diferentes

Predicado: [mostra_factos](#), [mostra_factos_filtrados](#)

% Visualização das justificações

Predicado: [mostra_justificacoes](#)

Diagnostico: [diagnostico.pl](#)

Adições:

% Correr diagnostico na consola e via frontend

Predicado: [diagnostico](#), [diagnostico_passo_passo_co](#), [diagnostico_passo_passo](#)

% Predicado para processar testes pendentes ou exhibe o diagnostico final

Predicado: [diagnostico_continuo](#), [diagnostico_finalizado](#)

% Verificar se há problemas pendentes

Predicado: [problemas_pendentes](#)

% Processar todos os factos 'proximo_teste' pendentes

Predicado: [diag_problemas](#), [diag_problemas_passo_passo](#)

% Predicado para processar cada teste

Predicado: [processar_testes](#), [processar_testes_passo_passo](#)

% Predicado para tratar cada problema individualmente, lidar com pergunta e respostas do utilizador

Predicado: [tratar_problema](#), [tratar_problema_passo_passo](#), [validar_resposta](#),
[obter_perguntas](#), [criacao_pergunta](#)

% Predicado para mostrar diagnostico final, solução final, diagnósticos possíveis

Predicado: [mostrar_diagnostico](#), [mostrar_solucao](#), [mostra_diagnostico_solucao](#),
[listar_diagnosticos_possiveis](#), [listar_diagnosticos_possiveis_aux](#)

Explicação como: [como.pl](#)

Alterações e adições:

% Alteração do método explicações do tipo "Como" para o diagnostico usando o justifica

Predicado: [como](#), [continua_como](#), [como_response](#)

Explicação porque: [porque.pl](#)

Alterações e adições:

% Criação do método explicações do tipo "porque" para o diagnostico usando o justifica

Predicado: [porque](#), [porque_response](#)

Explicação porque_nao: [porque_nao.pl](#)

Alterações e adições:

% Criação do método explicações do tipo "porque não" para o diagnostico usando
recursividade entre respostas e proximo_testes

Predicado: [porque](#), [porque_response](#), [encontra_regras_porque_nao](#), [porque_nao1](#)
