Documentação e Lógica do Programa

O programa opera com uma série de predicados (regras) que orquestram a interação e o cálculo da recomendação. A lógica segue estes passos:

1. Início do Programa (iniciar.)

O predicado **iniciar**. é o ponto de entrada principal. Ele limpa qualquer resposta de uma execução anterior (**retractall(resposta(_))**) e inicia o questionário chamando o predicado **fazer_perguntas(1)**. Ao final do questionário, ele chama **calcular_e_exibir_ranking** para processar e mostrar os resultados.

2. Questionário Interativo (fazer_perguntas)

Este predicado utiliza **recursão** para percorrer todas as perguntas.

- fazer_perguntas(ID): A regra verifica se existe uma pergunta com o ID atual. Se existir, ela exibe o texto da pergunta e lê a resposta do usuário.
- Se o usuário responder "s" ou "sim", o programa usa assertz(resposta(Caracteristica)) para adicionar a característica associada à pergunta na base de dados.
- Em seguida, ele incrementa o ID e chama fazer_perguntas novamente, repetindo o
 processo até que não haja mais perguntas. A condição de parada (\+ pergunta(ID,
 _, _)) encerra a recursão.

3. Cálculo da Pontuação (calcular_pontuacao)

Para cada trilha, este predicado calcula a pontuação de compatibilidade. Ele encontra todos os fatos **perfil** que correspondem à trilha e às respostas do usuário. A pontuação é a soma dos pesos de todas as características para as quais o usuário respondeu "sim".

4. Geração e Exibição do Ranking (calcular_e_exibir_ranking)

Este predicado centraliza a lógica de pontuação e ordenação.

- Ele usa **findall** para coletar as pontuações de todas as trilhas e armazená-las em uma lista de pares (**Pontuacao-Trilha**).
- A lista é então ordenada em ordem decrescente de pontuação usando sort(0, @>=,
 Lista, RankingOrdenado).
- O predicado exibir_ranking é chamado para exibir o ranking formatado, mostrando o nome, a pontuação e a descrição de cada trilha.

5. Exibição do Ranking (exibir_item_ranking)

Este predicado recursivo percorre a lista de resultados ordenados e exibe cada item de forma clara, com o nome da trilha, a pontuação de compatibilidade e a descrição.

6. Testes Automatizados (testar.)

O predicado **testar(ArquivoTeste)** foi criado para facilitar a validação do sistema. Ele carrega um arquivo de teste que contém fatos de **resposta/1** pré-definidos, simulando as respostas de um usuário. Em seguida, ele executa o **calcular_e_exibir_ranking** para mostrar os resultados para esse perfil específico, sem a necessidade de passar pelo questionário interativo.

Como usar o programa?

Para rodar o programa, basta carregar o arquivo no interpretador Prolog e chamar o predicado **"iniciar."** O programa guiará o usuário pelo questionário e, ao final, apresentará o ranking das trilhas mais compatíveis.