



Área do Conhecimento de Ciências Exatas
Curso: Bacharelado em Ciência da Computação
Disciplina: Inteligência Artificial I - SIS0484A
Docente: Dr^a Carine G. Webber
Aluno: Adriano Gomes da Silva

Projeto de um Consultor Financeiro em Prolog

| | |
|---|-----------|
| Projeto de um Consultor Financeiro em Prolog | 1 |
| Descrição do Problema: | 3 |
| Solução | 3 |
| Simulações Utilizando a Aplicação | 3 |
| Referências | 4 |
| Anexo I | 5 |
| Programa Desenvolvido | 5 |
| main.pl | 5 |
| menu.pl | 17 |
| questionario.pl | 21 |
| graficos.pl | 27 |
| list_utils.pl | 28 |
| Anexo II | 31 |
| Simulações Realizadas | 31 |
| Simulação 1: Perfil Renata | 31 |
| Simulação 2: Perfil Humberto | 34 |

Descrição do Problema:

O problema consiste em uma adaptação do problema encontrado no livro Inteligência Artificial 6a Edição, nas páginas 60 - 63, onde é desenvolvida uma aplicação para consultoria financeira, onde o usuário insere dados (fatos) e o programa (através de regras) valida sua renda, investimentos realizados e investimentos a serem realizados.

Solução

Para este problema, a abordagem tomada pelo autor consistiu em definir um sistema que realizasse investimentos diversificados com base em perfil do usuário, fundos de emergência do mesmo e prazo para retorno dos investimentos. Dentre as opções do programa, estão:

```
Menu do programa:
Digite a opção que deseja escolher.
1 - Exibir as aplicações em que um cliente está investindo
2 - Verifica se um cliente possui um fundo de emergência.
3 - Verifica se um cliente possui um valor mínimo para investimento.
4 - Não conhece seu perfil de investidor? Responda este questionário e descubra!
5 - Ver informações de um cliente.
6 - Adicionar cliente.
7 - Onde aplicar?
8 - Quanto lucrarei nas aplicações que tenho?
0 - Para sair.
|:
```

A solução desenvolvida para o problema encontra-se no anexo I.

Simulações Utilizando a Aplicação

As simulações realizadas utilizando a aplicação desenvolvida encontram-se no anexo II. Estas simulações pretendem explorar as opções do menu principal do projeto.

[Link para o repositório criado para o projeto.](#)

Referências

Base do projeto:

LUGER, George F. Inteligência artificial. 6.ed. São Paulo: Pearson Education do Brasil, 2013. 616 p.

Prolog:

<http://www.swi-prolog.org/pldoc/index.html>

Questionário:

<https://www.infomoney.com.br/blogs/financas-pessoais/financas-em-casa/post/5306053/qual-seu-perfil-investidor>

<https://www.infomoney.com.br/educacao/guias/noticia/401308/hora-investir-seu-dinheiro-que-considerar>

<https://www.infomoney.com.br/onde-investir/fundos-de-investimento/noticia/4435066/saiba-como-descobrir-seu-perfil-investidor>

Gráfico:

<https://github.com/SWI-Prolog/packages-xpce/blob/master/prolog/lib/plot/demo.pl>

<http://www.swi-prolog.org/packages/xpce/UserGuide/libplot.html#sec:11.6>

Fundo/Reserva de emergência:

<https://www.infomoney.com.br/blogs/investimentos/licoes-de-valor/post/7932714/por-que-todo-investidor-precisa-ter-uma-reserva-de-emergencia>

<https://blog.rico.com.vc/como-fazer-fundo-emergencia>

<https://mepoupenaweb.uol.com.br/videos/tudo-sobre-reserva-de-emergencia/>

Anexo I

Programa Desenvolvido

main.pl

```
:- dynamic cliente/9.  
:- consult(menu).  
:- consult(list_utils).  
:- consult(questionario).  
:- consult graficos).
```

```
% -----  
% Estrutura dos fatos :  
% -----  
% cliente(nome,  
%      saldo,  
%      renda_mensal,  
%      dependentes,  
%      prazo_retorno_investimento,  
%      [aplicacoes],  
%      [quantia_aplicada_em_cada_aplicacao],  
%      perfil,  
%      DESPESA)
```

```
% aplicacao(nome,  
%      rendimento,  
%      prazo_retorno,  
%      quantia_minima,  
%      perfil)
```

% @TODO: Substituir estes valores com os valores reais (se nao forem os reais, estipular alguns)

```
% aplicações  
aplicacao(cdb, 6, 36, 1000, agressivo).  
aplicacao(lci, 6, 36, 5000, agressivo).  
aplicacao(tesouro_selic, 1.6, 0, 30, conservador).  
aplicacao(tesouro_ipca, 5, 12, 1000, moderado).
```

```

aplicacao(poupanca, 0.6, 999,0, nao_recomendado).
nomes_aplicacoes([cdb,lc, tesouro_selic, tesouro_ipca, poupanca]).
aplicacoes_para_fundo_emergencia([tesouro_selic,cdb]).

```

```

% Testei cor a cor até chegar nessas
cores([red,blue,green,yellow,gray,black,brown,cyan,violet]).

```

```

% clientes

```

```

cliente(ronaldinho,7000,1000,1,12,[cdb,poupanca],[1000,20],conservador,500).
cliente(claudia,500,200,0,48,[poupanca],[50],nenhum,500).
cliente(juliana,3000,1000,2,36,[tesouro_selic],[5000],conservador,500).
cliente(bettina,70000,5000,0,12,[cdb,lc, tesouro_selic,
tesouro_ipca],[10000,10000,10000,10000],agressivo,500).
cliente(robson,2000,3000,0,12,[cdb,poupanca],[5000,20],moderado,500).
cliente(humberto,35000,10000,2,36,[tesouro_selic,cdb],[1000,2000],agressivo,5000).

```

```

% renda mínima para cada dependente
minimo_dependente(500).

```

```

% -----

```

```

/**

```

```

* Regra que verifica se um cliente possui fundo de emergência e, se possuir, sugere aplicar
* em investimentos que sejam de seu perfil, de modo diversificado. Caso contrário, se não
* possuir, sugere investir nas aplicações que garantem um fundo de emergência futuro.
* @param:
*   VALOR_INVEST = (int/float) valor a ser investido
*   PRAZO_RETORNO = (int/float) prazo de retorno do investimento
*   PERFIL = perfil do cliente
*   CLIENTE = nome do cliente
*
**/

```

```

% @TODO: VERIFICAR SE O CLIENTE JÁ POSSUI UM INVESTIMENTO EM
DETERMINADA APLICAÇÃO

```

```

% SE POSSUIR, ADICIONAR MAIS NELA

```

```

onde_pode_aplicar(VALOR_INVEST,_,_, CLIENTE):-

```

```

    cliente(CLIENTE,SALDO,_,DEPENDENTES,_,APLIC,QT_APLIC,_,DESPESA),
    calculo_fundo_emergencia(DEPENDENTES,APLIC,QT_APLIC,DESPESA, SOMA,
DESPESA_TOTAL),
    SOMA < DESPESA_TOTAL,
    FALTA is DESPESA_TOTAL - SOMA,
    write("Sinto muito, "),
    write(CLIENTE),
    write(". Faltam R$ "),

```

```

write(FALTA),
write(" para completar seu fundo de emergência!"),
nl,
write("É necessário montar/aumentar sua reserva de emergência antes de investir!"),
nl,
aplicacoes_para_fundo_emergencia(AFE),
retorna_lista_valor_p_investir_menor_saldo(AFE,SALDO,RESULTADO),
tamanho_lista(RESULTADO,TAM), %@TODO: ADICIONAR VERIFICAÇÃO PARA
TAMANHO como a abaixo.
write("As opções para investimento são: "),
nl,
exibe_valores_lista(RESULTADO),
write("Deseja aplicar diversificado nestas aplicações?(s/n)",
nl,
read(RESPOSTA),
write(RESPOSTA),
nl,
pergunta_aplicar_diversificado_em(RESPOSTA,AFE, CLIENTE, VALOR_INVEST),!.
onde_pode_aplicar(VALOR_INVEST,PRAZO_RETORNO,PERFIL, CLIENTE):-
    cliente(CLIENTE,_,_,DEPENDENTES,_,APLIC,QT_APLIC,_,DESPESA),
    calculo_fundo_emergencia(DEPENDENTES,APLIC,QT_APLIC,DESPESA, SOMA,
DESPESA_TOTAL),
    SOMA >= DESPESA_TOTAL,
    nomes_aplicacoes(NOMES_APLICACOES),
    verif_pode_aplicar(VALOR_INVEST, PRAZO_RETORNO,
PERFIL,NOMES_APLICACOES, LISTA_ONDE, _),
    tamanho_lista(LISTA_ONDE,TAM),

verificacao_tamanho_lista_pode_aplicar(TAM,LISTA_ONDE,CLIENTE,VALOR_INVEST),!.

/**
 * Regra que valida se há aplicações para o usuário investir
 * com base em uma comparação com o tamanho da lista de investimentos possíveis.
 * @param:
 * TAM = (int) Tamanho da lista de possíveis investimentos
 * LISTA_ONDE = Lista de aplicações onde o cliente fará os investimentos
 * CLIENTE = nome do cliente
 * VALOR_INVEST = (int/float) quantia a ser investida
 *
 */
verificacao_tamanho_lista_pode_aplicar(TAM,LISTA_ONDE,CLIENTE,VALOR_INVEST):-
    TAM>0,
    write("Parabéns! Você já possui um montante suficiente em sua reserva de emergência."),
    nl,
    write("As opções para investimento são: "),
    nl,

```

```

    exibe_valores_lista(LISTA_ONDE),
    write("Deseja aplicar diversificado nestas aplicações?(s/n)",
    nl,
    read(RESPOSTA),
    pergunta_aplicar_diversificado_em(RESPOSTA,LISTA_ONDE, CLIENTE,
    VALOR_INVEST),!.
verificacao_tamanho_lista_pode_aplicar(TAM,_,_):-
    TAM=<0,
    write("Não é possível investir em sequer uma aplicação!"),
    nl,
    write("Retornando ao menu."),
    nl,!.

/**
 * Regra que verifica a resposta do cliente sobre aplicar
 * ou não diversificadamente seus valores.
 * @param:
 *   RESPOSTA = RESPOSTA (s/n) do usuário
 *   LISTA_ONDE = Lista de aplicações onde o cliente fará os investimentos
 *   CLIENTE = nome do cliente
 *   SALDO = (int/float) saldo do cliente no banco
 **/
pergunta_aplicar_diversificado_em(RESPOSTA,LISTA_ONDE, CLIENTE, SALDO):-
    RESPOSTA = s,
    write("Quanto deseja investir?"),
    nl,
    write("Seu saldo é de: R$ "),
    write(SALDO),
    nl,
    read(QUANTO),
    calcula_quanto(LISTA_ONDE, CLIENTE,QUANTO).
pergunta_aplicar_diversificado_em(RESPOSTA,LISTA_ONDE, CLIENTE,SALDO):-
    RESPOSTA = n,
    write("Deseja escolher quais?(s/n)",
    nl,
    read(RESPOSTA1),
    pergunta_aplicar_escolhendo(RESPOSTA1,LISTA_ONDE, CLIENTE,SALDO),!.

pergunta_aplicar_escolhendo(RESPOSTA,LISTA_ONDE, CLIENTE,SALDO):-
    RESPOSTA = n,
    write("Tudo bem! Retornando ao menu"),!.
pergunta_aplicar_escolhendo(RESPOSTA,LISTA_ONDE, CLIENTE,SALDO):-
    RESPOSTA = s,
    write("Liste as aplicações onde quer investir. (Exemplo: [tesouro_selic,cdb,lcj])"),
    nl,

```



```

write("É possível investir nestas aplicações:"),
nl,
write(LISTA_ONDE),
nl,
read(LISTA_ONDE_RESPOSTA),
write("Liste quanto quer investir nas aplicações, na mesma ordem que informou as
aplicações."),
nl,
write(" (Exemplo: [3000,200,4000])"),
nl,
read(LISTA_QUANTO),
calcula_quanto_escolhido(LISTA_ONDE_RESPOSTA,CLIENTE,LISTA_QUANTO),!.

/**
 * Regra que calcula se o cliente pode efetivamente
 * realizar algum investimento em uma aplicação com base no quanto informou e seu saldo
 * @param:
 *   LISTA_ONDE = Lista de aplicações onde o cliente fará os investimentos
 *   CLIENTE = nome do cliente
 *   LISTA_QUANTO = (int/float) Valor informado pelo usuário para realizar investimentos
 *
 **/
calcula_quanto_escolhido(LISTA_ONDE, CLIENTE,LISTA_QUANTO):-
    cliente(CLIENTE,SALDO,_,_,_,_,_),
    soma_aplicacoes(LISTA_QUANTO,LISTA_ONDE,LISTA_QUANTO,SOMA),
    SOMA<=SALDO,
    aplicar_escolhido_em(LISTA_QUANTO,LISTA_ONDE,CLIENTE),
    write("Aplicação realizada com sucesso!").
calcula_quanto(_, CLIENTE,LISTA_QUANTO):-
    cliente(CLIENTE,SALDO,_,_,_,_,_),
    soma_aplicacoes(LISTA_QUANTO,LISTA_ONDE,LISTA_QUANTO,SOMA),
    SOMA>SALDO,
    write("Você não possui esta quantia! Seu saldo é de: "),
    write("R$"),
    write(SALDO).

/**
 * Regra que realiza a aplicação ESCOLHIDA PELO CLIENTE
 * em N aplicações, com base no valor passado.
 * @param:
 *   VALOR_INVEST = (int/float) valor TOTAL a ser investido em aplicação
 *   LISTA_ONDE = Lista de aplicações onde o cliente fará os investimentos
 *   CLIENTE = nome do cliente
 *
 **/
aplicar_escolhido_em(LISTA_VALOR, LISTA_ONDE, CLIENTE):-

```

```

    tamanho_lista(LISTA_ONDE, TAM),
    cliente(CLIENTE,SAL,REND,DEP,PRAZO,APLIC,QTD_APLIC,P,D),
    retract(cliente(CLIENTE,_,_,_,_,_,_,_)),

    aplicacoes_lista_cliente(APLIC,QTD_APLIC,LISTA_ONDE,LISTA_VALOR,QTD_RESULTA
DO,APLIC_RESULTADO),
    soma_aplicacoes(LISTA_ONDE,LISTA_ONDE,LISTA_VALOR,SOMA),
    SAL1 is SAL - SOMA,

    assert(cliente(CLIENTE,SAL1,REND,DEP,PRAZO,APLIC_RESULTADO,QTD_RESULTAD
O,P,D)).

/**
 * Regra que calcula se o cliente pode efetivamente
 * realizar algum investimento em uma aplicação com base no quanto informou e seu saldo
 * @param:
 *   LISTA_ONDE = Lista de aplicações onde o cliente fará os investimentos
 *   CLIENTE = nome do cliente
 *   QUANTO = (int/float) Valor informado pelo usuário para realizar investimentos
 *
 */
calcula_quanto(LISTA_ONDE, CLIENTE,QUANTO):-
    cliente(CLIENTE,SALDO,_,_,_,_,_,_),
    QUANTO=<SALDO,
    aplicar_diversificado_em(QUANTO,LISTA_ONDE,CLIENTE),
    write("Aplicação realizada com sucesso!").
calcula_quanto(_, CLIENTE,QUANTO):-
    cliente(CLIENTE,SALDO,_,_,_,_,_,_),
    QUANTO>SALDO,
    write("Você não possui esta quantia! Seu saldo é de: "),
    write("R$"),
    write(SALDO).

/**
 * Regra que realiza a aplicação diversificada do
 * cliente em múltiplas aplicações, com base no valor passado.
 * @param:
 *   VALOR_INVEST = (int/float) valor TOTAL a ser investido em aplicação
 *   LISTA_ONDE = Lista de aplicações onde o cliente fará os investimentos
 *   CLIENTE = nome do cliente
 *
 */
aplicar_diversificado_em(VALOR_INVEST, LISTA_ONDE, CLIENTE):-
    tamanho_lista(LISTA_ONDE, TAM),
    QUANTO_EM_CADA is VALOR_INVEST/TAM,
    cliente(CLIENTE,SAL,REND,DEP,PRAZO,APLIC,QTD_APLIC,P,D),

```

```

    retract(cliente(CLIENTE,_,_,_,_,_,_)),

    aplicacoes_lista_cliente(APLIC,QTD_APLIC,LISTA_ONDE,QUANTO_EM_CADA,QTD_RES
ULTADO,APLIC_RESULTADO),
    SAL1 is SAL - VALOR_INVEST,

    assert(cliente(CLIENTE,SAL1,REND,DEP,PRAZO,APLIC_RESULTADO,QTD_RESULTAD
O,P,D)).

/**
 * Regra que atualiza a lista de valores com dois tipos de regras:
 *   Se uma aplicação já faz parte da lista de aplicações do cliente, atualiza o valor dela
 *   Se uma aplicação não faz parte da lista, insere ela na lista (e seu valor na lista de
valores)
 *   As listas novas são retornadas nas variáveis APLIC_RESULTADO (Lista com nomes
das aplicações) e QTD (lista com valores aplicados)
 * @param:
 *   APLIC = Lista com os nomes das aplicações do Cliente
 *   QTD_APLIC = Lista com valores aplicados do cliente
 *   LISTA_ONDE = Lista onde serão aplicados os investimentos
 *   VALOR_INVEST = (int/float) valor a ser investido nas aplicações da lista
 *   QTD = Lista com as novas quantidades aplicadas
 *   APLIC_RESULTADO = Lista com as novas aplicações
 *
**/
    aplicacoes_lista_cliente(APLIC, QTD,[],_,QTD,APLIC).
    aplicacoes_lista_cliente(APLIC,
QTD_APLIC,[ONDE|TONDE],VALOR_INVEST,QTD,APLIC_RESULTADO):-
    member(ONDE,APLIC),

    retorna_lista_valores_atualizados(APLIC,QTD_APLIC,ONDE,VALOR_INVEST,QTD_APLIC
1),
    aplicacoes_lista_cliente(APLIC,
QTD_APLIC1,TONDE,VALOR_INVEST,QTD,APLIC_RESULTADO),!.
    aplicacoes_lista_cliente(APLIC,
QTD_APLIC,[ONDE|TONDE],VALOR_INVEST,QTD,APLIC_RESULTADO):-
    not(member(ONDE,APLIC)),
    insere_lista(QTD_APLIC,VALOR_INVEST,QTD1),
    insere_lista(APLIC,ONDE,APLIC1),
    aplicacoes_lista_cliente(APLIC1,
QTD1,TONDE,VALOR_INVEST,QTD,APLIC_RESULTADO),!.

    aplicacoes_lista_cliente(APLIC,
QTD_APLIC,[ONDE|TONDE],[QUANTO|TQUANTO],QTD,APLIC_RESULTADO):-
    member(ONDE,APLIC),
    retorna_lista_valores_atualizados(APLIC,QTD_APLIC,ONDE,QUANTO,QTD_APLIC1),

```

```

    aplicacoes_lista_cliente(APLIC,
    QTD_APLIC1,TONDE,TQUANTO,QTD,APLIC_RESULTADO),!.
    aplicacoes_lista_cliente(APLIC,
    QTD_APLIC,[ONDE|TONDE],[TQUANTO],QTD,APLIC_RESULTADO):-
        not(member(ONDE,APLIC)),
        insere_lista(QTD_APLIC,TQUANTO,QTD1),
        insere_lista(APLIC,ONDE,APLIC1),
        aplicacoes_lista_cliente(APLIC1, QTD1,TONDE,TQUANTO,QTD,APLIC_RESULTADO),!.
/**
* Regra que retorna uma lista de valores
* atualizados com a adiç o do valor anterior
* pelo valor de investimento.
* @param:
* LISTA_NOME_APLICACOES = Lista com o nome das aplica es a serem comaprados
* LISTA_VALOR_APLICADO_APLICACOES = Lista com o valor das aplica es a serem
alterados
* A = Vari vel que ser  comparada com o nome das aplica es para verificar se
ALTERA ou n o seu valor
* VAL = (int/float) Vari vel com o VALOR que ser  adicionado ao valor de uma aplica o
caso ela seja a aplica o buscada.
* LISTA_RESULTADO = Lista com o resultado da lista de valores alterada
**/
retorna_lista_valores_atualizados([],[],_,_,[]).
retorna_lista_valores_atualizados([H|T],[H1|T1],A,VAL,[H2|Result]):-
    H = A, % se forem iguais, adicionar  o valor dele + VAL em uma vari vel nova
    H2 is VAL + H1, % e ent o colocar  H2 nesta posi o da lista, no retorno
    retorna_lista_valores_atualizados(T,T1,A,VAL,Result),!.
retorna_lista_valores_atualizados([_|Tail],[H1|T1],A,VAL,[H1|Result]):- % Coloca o valor
normal
    retorna_lista_valores_atualizados(Tail,T1,A,VAL,Result). %prosssegue para o filho

/**
* Regra que retorna uma lista que um cliente pode investir com base no valor a ser investido
* @param:
* Lista = Lista de aplica es
* VAL = (int/float) Valor a ser investido
* Lista1 = Lista com o resultado das aplica es
*
**/

retorna_lista_valor_p_investir_menor_saldo([],_,[]).
retorna_lista_valor_p_investir_menor_saldo([H|T],VAL,[H|Result]):-
    aplicacao(H,_,_,VALOR_INVESTIR,_),
    VALOR_INVESTIR <= VAL, % se forem iguais, adicionar  o valor de H na lista de
retorno

```

```

    retorna_lista_valor_p_investir_menor_saldo(T,VAL,Result),!.
retorna_lista_valor_p_investir_menor_saldo([H|Tail],VAL,[H|Result]):- % Coloca o valor
normal
    retorna_lista_valor_p_investir_menor_saldo(Tail,VAL,Result).          %prosegue para o
filh

```

```

/**
 * Regra que retorna uma lista que um cliente
 * pode investir com base no valor a ser investido
 * e no prazo de retorno do mesmo e em seu perfil
 *
 * @param:
 *   VALOR_INVEST = Valor a ser investido
 *   PRAZO_RETORNO = Prazo de retorno esperado
 *   PERFIL = Perfil de investidor do cliente
 *   LISTA = Lista de aplicações
 *   LISTA_ONDE = lista onde será retornado o valor da lista onde o cliente pode aplicar
 *   AUX = Variável auxiliar para montar a LISTA_ONDE
 *
 **/

```

```

verif_pode_aplicar(_, _, _, [], AUX, AUX).
verif_pode_aplicar(VALOR_INVEST, PRAZO_RETORNO, PERFIL, [H|T], LISTA_ONDE,
AUX):-
    aplicacao(H, _, P_R, VAL, PERFIL),
    VALOR_INVEST >= VAL,
    PRAZO_RETORNO >= P_R,
    verif_pode_aplicar(VALOR_INVEST, PRAZO_RETORNO, PERFIL, T, LISTA_ONDE,
[H|AUX]),!.
verif_pode_aplicar(VALOR_INVEST, PRAZO_RETORNO, PERFIL, [_|T], LISTA_ONDE,
AUX):-
    verif_pode_aplicar(VALOR_INVEST, PRAZO_RETORNO, PERFIL, T, LISTA_ONDE,
AUX),!.

```

```

/**
 * Regra que verifica o valor mínimo para um investimento. Se possuir, exibe uma
mensagem adequada.
 * @param:
 *   SALDO = (int/float) Quanto o cliente possui de saldo no banco
 *   RENDA = (int/float) Renda do cliente
 *   DEP = (int) Quantidade de dependentes que o cliente possui
 **/
valor_minimo_investimento(SALDO,RENDA,DEP):-
    minimo_dependentes(DEP,MIN),
    S1 is RENDA - MIN,

```

```
S2 is SALDO + S1,  
S2 > 30,  
write("Parabéns, você possui saldo o suficiente para investir.").
```

```
/**  
 * Regra que verifica o valor mínimo para um investimento. Se não possuir, exibe uma  
 mensagem adequada.  
 * @param:  
 *   SALDO = (int/float) Quanto o cliente possui de saldo no banco  
 *   RENDA = (int/float) Renda do cliente  
 *   DEP = (int) Quantidade de dependentes que o cliente possui  
 */  
valor_minimo_investimento(SALDO,RENDA,DEP):-  
    minimo_dependentes(DEP,MIN),  
    S1 is RENDA - MIN,  
    S2 is SALDO + S1,  
    S2 < 30,  
    write("Você possui menos que a quantidade suficiente para investir. O mínimo é 30 reais  
 para o tesouro_selic.").
```

```
/**  
 * Regra que verifica a renda mínima necessária  
 * para a quantidade de dependentes que um cliente possui  
 * @param:  
 *   DEP = (int) Quantidade de dependentes que um cliente possui  
 *   MIN = (int/float) Variável com a renda mínima necessária para a quantidade de  
 dependentes  
 */  
minimo_dependentes(DEP,MIN):-  
    minimo_dependente(MIN_DEP),  
    MIN is DEP * MIN_DEP.
```

```
/**  
 * Regra intermediária que primeiro executa  
 * o cálculo para o fundo de emergência  
 * e então passa as variáveis necessárias  
 * para a regra de verificação se o cliente  
 * possui um fundo de emergência adequado.  
 * @param:  
 *   DEPENDENTES = (int) quantidade de dependentes que um cliente possui  
 *   DESPESA = (int/float) despesa mensal do cliente  
 *   SOBRA = (int/float) variável com o retorno da verificação de fundo de emergência  
 *   APLIC = Lista com nomes de aplicações que um cliente está investindo
```

```

*   QT_APLIC = Lista com valores de aplicações que um cliente está investindo
*
**/
verifica_fundo_emergencia( DEPENDENTES, DESPESA, SOBRA, APLIC, QT_APLIC):-
    calculo_fundo_emergencia(DEPENDENTES,APLIC,QT_APLIC,DESPESA, SOMA,
    DESPESA_TOTAL),
    verifica_fundo_emergencia_comparacao(SOMA,DESPESA_TOTAL,SOBRA).

/**
* Regra que calcula o valor das variáveis
* para a verificação se o cliente está adequado
* ou não aos investimentos para caracterizar o
* fundo de emergência.
* @param:
*   DEPENDENTES = (int) Quantidade de dependentes que um cliente tem
*   APLIC = Lista com nomes de aplicações que um cliente está investindo
*   QT_APLIC = Lista com valores de aplicações que um cliente está investindo
*   DESPESA = (int/float) Despesa mensal de um cliente
*   SOMA = (int/float) Soma de todas as aplicações
*   DESPESA_TOTAL = (int/float) Despesa total da família (cliente + dependentes) para 6
meses
**/
calculo_fundo_emergencia(DEPENDENTES,APLIC,QT_APLIC,DESPESA,SOMA,DESPESA_TOTAL):-
    DESPESA_SEIS_MESES is DESPESA * 6,
    minimo_dependentes(DEPENDENTES, MINDEP),
    DESPESA_SEIS_MESES_DEP is MINDEP * 6,
    DESPESA_TOTAL is DESPESA_SEIS_MESES + DESPESA_SEIS_MESES_DEP,
    aplicacoes_para_fundo_emergencia(X),
    soma_aplicacoes(X,APLIC,QT_APLIC,SOMA),
    nl.

/**
* Regra que verifica se o valor passado em receita é
* maior que a despesa total de uma família. Se for maior,
* atrela à variável SOBRA a quantidade de renda que resta
* após a subtração das despesas da família e exibe uma mensagem na tela.
*
* @params:
*   RECEITA = (int/float) receita de uma família
*   DESPESA_TOTAL = (int/float) despesa total de uma família (ao longo de seis meses)
*   SOBRA = resultado de RECEITA - DESPESA_TOTAL
**/
verifica_fundo_emergencia_comparacao(RECEITA,DESPESA_TOTAL,SOBRA):-
    RECEITA >= DESPESA_TOTAL,
    SOBRA is RECEITA - DESPESA_TOTAL,

```

```

write("Parabéns, você possui um fundo de emergência com "),
write(SOBRA),
write(" de diferença do mínimo que precisaria para sobreviver por seis meses!"). % aqui
avisar que pode investir em mais??

```

```

/**
 * Regra que verifica se o valor passado em receita é
 * menor que a despesa total de uma família. Se for menor,
 * atrela à variável FALTAM o quanto falta para a receita
 * alcançar a despesa e exibe uma mensagem na tela.
 *
 * @params:
 *   RECEITA = (int/float) receita de uma família
 *   DESPESA_TOTAL = (int/float) despesa total de uma família (ao longo de seis meses)
 *   FALTAM = resultado de DESPESA_TOTAL - RECEITA
 */
verifica_fundo_emergencia_comparacao(RECEITA,DESPESA_TOTAL,FALTAM):-
    RECEITA < DESPESA_TOTAL,
    FALTAM is DESPESA_TOTAL - RECEITA,
    write("Que pena! Faltam "),
    write(FALTAM),
    write(" reais para você atingir seu fundo de emergência!"). % aqui avisar que pode investir
em mais??

```

```

/**
 * Regra que exibe as aplicações do cliente textualmente e por meio de um gráfico
 * A ordem exibida é nome aplicação -> valor aplicado
 * @param:
 *   C = nome do cliente
 */
exibe_aplicacoes_valores_cliente(C):-
    cliente(C,_,_,_, T, QT,_,_),
    write("Você está investindo em: "),
    nl,
    exibe_aplicacao_valor(T,QT),
    nl,
    cores(CORES),
    monta_barras(T,QT,CORES,LISTA_FINAL),
    nomes_aplicacoes(X),
    soma_aplicacoes(X,T,QT,SOMA),
    grafico_aplicacoes(LISTA_FINAL,SOMA).

```

```

/**
 * Regra intermediária para a verificação de lucro nas aplicações de um cliente

```



```

* @params:
*   C = nome do cliente
*
**/
quanto_lucrarei(C):-
    cliente(C,_,_,_,APLIC,QTD_APLIC,_,_),
    verifica_todas_aplicacoes_lucro(APLIC,QTD_APLIC).

/**
* Regra que busca e escreve o nome da aplicação e
* o quanto essa aplicação irá gerar de lucro ao fim
* do seu prazo de rendimento.
*
* A regra recebe duas listas e elas PRECISAM estar na mesma ordem para fazer sentido.
* E.g.: Se investi 2000 reais em cdb e 1000 reais em tesouro_selic, as listas serão
*   [cdb, tesouro_selic]
*   [2000, 1000]
* @params:
*   LISTA_APLICAÇÕES (Parametro 1) = Lista com o nome de aplicações
*   LISTA_VALOR_APLICADO (Parametro 2) = Lista com o valor de aplicações
**/
verifica_todas_aplicacoes_lucro([],_).
verifica_todas_aplicacoes_lucro([H|T],[H1|T1]):-
    aplicacao(H,REND,_,_,_),
    V is H1 * (REND),
    write("Lucro para "),
    write(H),
    write(": R$ "),
    write(V),
    nl,
    verifica_todas_aplicacoes_lucro(T,T1).

```

menu.pl

```

/**
*
* Regra com o menu de exibição do programa
*
**/
menu():-
    nl,
    write(" Menu do programa: "),
    nl,

```

```

write(" Digite a opção que deseja escolher."),
nl,
write(" 1 - Exibir as aplicações em que um cliente está investindo"),
nl,
write(" 2 - Verifica se um cliente possui um fundo de emergência."),
nl,
write(" 3 - Verifica se um cliente possui um valor mínimo para investimento."),
nl,
write(" 4 - Não conhece seu perfil de investidor? Responda este questionário e
descubra!"),
nl,
write(" 5 - Ver informações de um cliente."),
nl,
write(" 6 - Adicionar cliente."),
nl,
write(" 7 - Onde aplicar?"),
nl,
write(" 8 - Quanto lucrarei nas aplicações que tenho?"),
nl,
write(" 0 - Para sair."),
nl,
read(X),
opc(X),
menu().

```

```

/**
 *
 * Regras para opções do menu de execução do programa
 * @param:
 *     X = número da opção escolhida
 *
 **/

```

```

opc(X):-
    X = 0,
    halt.

```

```

opc(X):-
    X = 1,
    leitura_nome_cliente(C),
    exibe_aplicacoes_valores_cliente(C).

```

```

opc(X):-
    X = 2,
    leitura_nome_cliente(C),
    cliente(C,_,_, DEP, _, APLIC, QT_APLIC, _, DESP),
    verifica_fundo_emergencia( DEP, DESP, _, APLIC, QT_APLIC).

```

opc(X):-

```
X = 3,  
leitura_nome_cliente(C),  
cliente(C,SALDO,RENDA, DEP, _, _, _, _),  
valor_minimo_investimento(SALDO,RENDA,DEP).
```

opc(X):-

```
X = 4,  
leitura_nome_cliente(C),  
cliente(C,_,_,_, _ , _ , _ , _),  
questionario_perfil(C).
```

opc(X):-

```
X = 5,  
leitura_nome_cliente(C),
```

```
cliente(C,SALDO,RENDA_MENSAL,DEPENDENTES,PRAZO_RETORNO,APLICACOES,Q  
UANTIA,PERFIL,DESPESA),
```

```
write("Nome do cliente: "),  
write(C),  
nl,  
write("Saldo do cliente:"),  
write(SALDO),  
nl,  
write("Renda mensal:"),  
write(RENDA_MENSAL),  
nl,  
write("Quantidade dependentes:"),  
write(DEPENDENTES),  
nl,  
write("Prazo de retorno:"),  
write(PRAZO_RETORNO),  
nl,  
write("Aplicações e quantia em cada aplicacao:"),  
nl,  
write(APLICACOES),  
nl,  
write(QUANTIA),  
nl,  
write("Perfil de investidor:"),  
write(PERFIL),  
nl,  
write("Despesas mensais:"),  
write(DESPESA),
```

nl,!.

opc(X):-

```
X = 5,  
write("Que pena, não há sequer um cliente com este nome. "),  
nl,  
write("Retornando ao menu!"),  
nl,!.
```

opc(X):-

```
X = 6,  
write("Nome do cliente (Exemplo: homer_simpson): "),  
nl,  
read(C),  
write("Saldo do cliente:"),  
nl,  
read(SALDO),  
write("Renda mensal:"),  
nl,  
read(RENDA_MENSAL),  
write("Quantidade dependentes:"),  
nl,  
read(DEPENDENTES),  
write("Prazo de retorno:"),  
nl,  
read(PRAZO_RETORNO),  
write("Aplicações (Exemplo: [tesouro_selic,cdb,lci]):"),  
nl,  
read(APLICACOES),  
write("Quantia em cada aplicacao (Na mesma ordem que as aplicações. Exemplo:  
[30,1000,20]):"),  
nl,  
read(QUANTIA),  
write("Perfil de investidor (Perfis disponiveis: nenhum; agressivo; conservador;  
moderado):"),  
nl,  
read(PERFIL),  
write("Despesas mensais:"),  
nl,  
read(DESPESA),  
  
assert(cliente(C,SALDO,RENDA_MENSAL,DEPENDENTES,PRAZO_RETORNO,APLICAC  
OES,QUANTIA,PERFIL,DESPESA)),  
write("Usuário adicionado com sucesso!"),  
nl,!.
```

```
opc(X):-
    X = 7,
    leitura_nome_cliente(C),
    cliente(C,SALDO,_,_,PRAZO_RETORNO,_,_,PERFIL,_),
    onde_pode_aplicar(SALDO,PRAZO_RETORNO,PERFIL, C).
```

```
opc(X):-
    X = 8,
    leitura_nome_cliente(C),
    quanto_lucrarei(C).
```

```
/**
 * Regra para encapsular a leitura do nome do cliente
 * @param:
 * C = nome do cliente a ser validado
 *
 **/
```

```
leitura_nome_cliente(C):-
    write("Digite o nome do cliente. Exemplo: ronaldinho_gaucho."),
    nl,
    read(C),
    valida_nome_cliente(C).
```

```
/**
 *
 * Regra para a validação de um cliente
 * @param:
 * C = nome do cliente a ser validado
 **/
```

```
valida_nome_cliente(C):-
    cliente(C,_,_,_,_,_,_,_,_).
```

questionario.pl

```
/**
 *
 * Regra que aplica uma adaptação do questionário encontrado em:
 https://www.infomoney.com.br/blogs/financas-pessoais/financas-em-casa/post/5306053/qual-seu-perfil-investidor
 * Subsequentes regras servem para validação do questionário, quantificação de respostas e montagem do gráfico
 * @param:
```

*

*/

```
cliente(CLIENTE,_,_,_,_,_,PERFIL,_),
```

nl,

```
write(CLIENTE),
```

nl,

nl,

nl,

nl,

```
write(" ---- Questionário ---- "),
```

nl,

nl,

nl,

nl.

nl,

nl,

nl.

```
write(" 2 - Você aceita perder uma porcentagem do recurso investido?"),
```

nl.

nl.

nl.

nl.

Pág. 22 de 38

```

nl,
read(DOIS),
nl,
nl,
write(" 3 - Como você poupa rendas extras (13º salário, restituição do Imposto de
Renda...)?"),
nl,
write("  a - Faço investimento em aplicações que sejam seguras."),
nl,
write("  b - Coloco a maior parte em investimentos isentos de risco e uma parcela em
modalidades um pouco mais arrojadas."),
nl,
write("  c - Aplico majoritariamente em investimentos de risco (mais chances de alto
rendimento) e uma pequena parcela em aplicações conservadoras."),
nl,
write(" ----- Digite sua Resposta -----"),
nl,
read(TRES),
nl,
nl,
write(" 4 - E o planejamento para a aposentadoria?"),
nl,
write("  a - Estou poupando e investindo em aplicações de baixo risco."),
nl,
write("  b - Uma parte do que guardo coloco em fundos conservadores e uma parte
menor em investimentos mais agressivos."),
nl,
write("  c - Invisto sempre meus recursos em aplicações que gerem alto retorno
financeiro."),
nl,
write(" ----- Digite sua Resposta -----"),
nl,
read(QUATRO),
nl,
nl,
write(" 5 - Um futuro sustentável financeiramente significa:"),
nl,
write("  a - Ter segurança financeira, mesmo em pouca quantia."),
nl,
write("  b - Guardar a maior parte do dinheiro de modo cauteloso e separando uma
parte para aplicações de rendimento maior."),
nl,
write("  c - Investir cada vez mais (agressivamente), mesmo que tenha riscos."),
nl,
write(" ----- Digite sua Resposta -----"),
nl,

```

```

        read(CINCO),
        nl,
        nl,
        respostas_questionario([UM,DOIS,TRES,QUATRO,CINCO],A,B,C, NOVO_PERFIL),
        nl,
        prepara_grafico_questionario([UM,DOIS,TRES,QUATRO,CINCO], A,B,C),
        nl,
        pergunta_adicionar_perfil_a_base(CLIENTE,PERFIL, NOVO_PERFIL).

/**
 * @param:
 *     CLIENTE = Nome do cliente
 *     PERFIL = Perfil do cliente
 *     NOVO_PERFIL = Perfil retornado pela próxima regra a ser chamada
 */
pergunta_adicionar_perfil_a_base(CLIENTE, PERFIL, NOVO_PERFIL):-
    nl,
    write("Deseja adicionar o perfil de investimento ao seu perfil?(s/n)",
    nl,
    read(Resp),
    prepara_adicionar_perfil_a_base(Resp, CLIENTE, PERFIL, NOVO_PERFIL),!.

/**
 * @param:
 *     Resp = Resposta da regra anterior
 *     CLIENTE = Nome do cliente
 *     PERFIL = Perfil do cliente
 *     NOVO_PERFIL = Perfil retornado pela próxima regra a ser chamada
 */
prepara_adicionar_perfil_a_base(Resp, CLIENTE, PERFIL,NOVO_PERFIL):-
    Resp = s,
    PERFIL \= nenhum,
    nl,
    write("Seu perfil já possui o perfil de investimento "),
    write(PERFIL),
    write(" atrelado a ele, "),
    write(CLIENTE),
    write(". Deseja sobrescrevê-lo? (s/n)",
    nl,
    read(SOBRESCREVER),
    prepara_sobrescrever_adicionar_perfil_a_base(SOBRESCREVER, CLIENTE,
NOVO_PERFIL),!.
prepara_adicionar_perfil_a_base(Resp, CLIENTE, PERFIL,NOVO_PERFIL):-
    Resp = s,
    PERFIL = nenhum, % teste redundante...

```



```

        nl,
        prepara_sobrescrever_adicionar_perfil_a_base(s, CLIENTE, NOVO_PERFIL).
prepara_adicionar_perfil_a_base(RESP, _, _):-
    RESP = n,
    nl,
    write("Sem problemas! Retornando ao menu.").

```

```

prepara_sobrescrever_adicionar_perfil_a_base(SOBRESCREVER, CLIENTE,
NOVO_PERFIL):-
    SOBRESCREVER = s,
    cliente(CLIENTE,A,B,C,D,E,F,_,G),
    retract(cliente(CLIENTE,_,_,_,_,_,_,_)),
    assert(cliente(CLIENTE,A,B,C,D,E,F,NOVO_PERFIL,G)),
    nl,
    write("Perfil de investimento alterado com sucesso!").
prepara_sobrescrever_adicionar_perfil_a_base(SOBRESCREVER, _, _):-
    SOBRESCREVER = n,
    nl,
    write("Sem problemas! Retornando ao menu.").

```

```

prepara_grafico_questionario(RESPOSTAS,A,B,C):-
    tamanho_lista(RESPOSTAS, TAMANHO_LISTA),
    grafico_questionario(A,B,C,TAMANHO_LISTA).

```

```

respostas_questionario(RESPOSTAS,A,B,C, NOVO_PERFIL):-
    quantifica_respostas_questionarios(RESPOSTAS,A,B,C),
    compara_respostas(A,B,C,NOVO_PERFIL).

```

% Comparações de respostas

```

/**
 * @param:
 *   A = Quantidade de respostas da opção A
 *   B = Quantidade de respostas da opção B
 *   C = Quantidade de respostas da opção C
 *   AUX = Perfil retornado
 **/
compara_respostas(A,B,C,AUX):-
    A>B,
    A>C,
    AUX = conservador,
    write("Você tirou perfil Conservador. Parece que você não gosta de correr riscos ou
busca por rendas estáveis a um curto/médio prazo").
compara_respostas(A,B,C,AUX):-
    B>A,

```

```

        B>C,
        AUX = moderado,
        write("Você tirou perfil Moderado. Parece que você busca um lucro acima do normal,
sem correr tantos riscos em uma aplicação agressiva.").
compara_respostas(A,B,C,AUX):-
    C>A,
    C>B,
    AUX = agressivo,
    write("Você tirou perfil Agressivo. Você busca um lucro elevado, acima dos perfis
Moderado e Conservador, mesmo que eventualmente ocorram perdas do capital investido.").
% compara_respostas(A,B,C,AUX):-
%     A=B,
%     AUX = d,
%     write("Conservador Moderado"). % ab
% compara_respostas(A,B,C,AUX):-
%     B=C,
%     AUX = e,
%     write("Moderado tendendo para agressivo"). % bc
% compara_respostas(A,B,C,AUX):-
%     A=C,
%     AUX = f,
%     write("Conservador e agressivo."). % ac

/**
 * Regra que contabiliza a quantidade de resposta 'a', 'b' e 'c'
 * escolhidas pelo usuário.
 * @param:
 *     LISTA_RESPOSTAS = Lista de respostas do cliente para o questionário
 *     A = Variável auxiliar a qual possuirá o retorno com a SOMA das respostas 'a'
 *     B = Variável auxiliar a qual possuirá o retorno com a SOMA das respostas 'b'
 *     C = Variável auxiliar a qual possuirá o retorno com a SOMA das respostas 'c'
 **/
quantifica_respostas_questionarios([],0,0,0).
quantifica_respostas_questionarios([H|T],A,B,C):-
    H=a,
    quantifica_respostas_questionarios(T,A1,B,C),
    A is A1 + 1.
quantifica_respostas_questionarios([H|T],A,B,C):-
    H=b,
    quantifica_respostas_questionarios(T,A,B1,C),
    B is B1 + 1.
quantifica_respostas_questionarios([H|T],A,B,C):-
    H=c,
    quantifica_respostas_questionarios(T,A,B,C1),

```

C is C1 + 1.

graficos.pl

```
:- use_module(library('plot/barchart')).
:- use_module(library(autowin)).
:- use_module(library(pce)).
```

% Referências para o gráfico de barras:

% <https://github.com/SWI-Prolog/packages-xpce/blob/master/prolog/lib/plot/demo.pl>

% <http://www.swi-prolog.org/packages/xpce/UserGuide/libplot.html#sec:11.6>

```
% -----
```

```
/**
```

```
 * Gráfico Questionário
```

```
 * Está plotando na horizontal pois na vertical não consegui fazer a legenda do eixo X
aparecer
```

```
 *
```

```
 * @param:
```

```
 *     A1 = (int/float) Altura da barra 1
```

```
 *     B1 = (int/float) Altura da barra 2
```

```
 *     C1 = (int/float) Altura da barra 3
```

```
 *     QTD_RESPOSTAS = (int/float) Total de respostas
```

```
 * */
```

```
grafico_questionario(A1,B1,C1, QTD_RESPOSTAS) :-
```

```
    new(W, auto_sized_picture('Respostas do Questionário')),
```

```
    send(W, display, new(BC, bar_chart(horizontal,0,QTD_RESPOSTAS))),
```

```
    forall(member(Name/Height/Color,
```

```
    [a/A1/red, b/B1/green, c/C1/blue]),
```

```
    ( new(B, bar(Name, Height)),
```

```
    send(B, colour(Color)),
```

```
    send(BC, append, B)
```

```
    )),
```

```
    send(W, open).
```

```
% -----
```

```
/**
```

```
 * Gráficos Aplicações
```

```
 *
```

```
 * @param:
```

```
 *     LISTA_MEMBROS = Lista com as barras a serem exibidas no gráfico,
```

```
 *                     já em formato Nome/Altura/Cor
```

```
 *     TOTAL = Valor do mais alto registro que será exibido
```

```

*
**/
grafico_aplicacoes(LISTA_MEMBROS,TOTAL) :-
    new(W, auto_sized_picture('Investimentos Realizados')),
    send(W, display, new(BC, bar_chart(horizontal,0,TOTAL))),
    forall(member(Name/Height/Color,
        LISTA_MEMBROS),
        ( new(B, bar(Name, Height)),
          send(B, colour(Color)),
          send(BC, append, B)
        )),
    send(W, open).

% monta_barras(LISTA_NOMES_APLICACOES, LISTA_QUANTIDADE_APLICACOES,
LISTA_NOMES_CORES, LISTA_FINAL):-

/**
* Regra que monta as barras para o gráfico barchart
* @param:
*     L1 = Lista com os nomes das barras a serem exibidas
*     L2 = Lista com os valores (alturas) das barras a serem exibidas
*     L3 = Lista com os nomes das cores das barras a serem exibidas
*     LISTA_FINAL = Lista montada no padrão Nome/Altura/Cor a ser retornada
**/
monta_barras([], _, _, _).
monta_barras([H|T], [H1|T1], [H2|T2], LISTA_FINAL):-
    monta_barras(T,T1,T2,LISTA_FINAL1),
    append(LISTA_FINAL1,[H/H1/H2],LISTA_FINAL).

```

list_utils.pl

```

/**
* Regra para exibição de valores de uma lista
* @param:
*     L = Lista para ter os valores exibidos
*
**/
exibe_valores_lista([]).
exibe_valores_lista([H|T]):-
    write(H),
    nl,
    exib_valores_lista(T),!.

/**
* Regra para verificar o tamanho de uma lista

```

```

* @param:
*     L = Lista para verificar o tamanho
*     TAM = Tamanho da lista a ser retornado
*
**/
tamanho_lista([],0).
tamanho_lista(_|T,TAM):-
    tamanho_lista(T, T1),
    TAM is T1 + 1.

/**
* Regra que encapsula a regra de APPEND para listas,
* para o usuário passar um valor e inserí-lo ao final de uma lista
* @param:
*     L = Lista de origem
*     VAL = Valor a ser inserido ao fim da lista
*     RESULTADO = Lista resultante
*
**/
insere_lista(L,VAL,RESULTADO):-
    append(L,[VAL], RESULTADO),!.

/**
* Regra para exibição de aplicações e valores de duas listas distintas
* @param:
*     L = Lista para ter as aplicações exibidas
*     L1 = Lista para ter os valores das aplicações exibidos
*
**/
exibe_aplicacao_valor([],[]).
exibe_aplicacao_valor([H|T], [H1|T1]):-
    write("Aplicação: "),
    write(H),
    write(". Valor: R$"),
    write(H1),
    nl,
    exibe_aplicacao_valor(T,T1).

/**
* Regra para somar aplicações de uma lista
* @param:
*     L = Lista para ter as aplicações somadas
*     A = Lista com as aplicações
*     QT = Lista com os valores das aplicações
*     SOMA = Variável para retornar a soma
*

```

```

**/
soma_aplicacoes([],_,_,0).
soma_aplicacoes([H|T],A,QT,SOMA):-
    pega_aplicacao_valor(H,A,QT,_,VAL),
    soma_aplicacoes(T,A,QT,S1),
    SOMA is S1 + VAL,!.

/**
* Regra para pegar aplicações de uma lista
* @param:
*   A = Valor a ser retornado
*   L = Lista com as aplicações
*   L1 = Lista com os valores das aplicações
*   APLIC = Variável com o nome da aplicação a ser retornado
*   VAL = Variável com o valor da aplicação a ser retornado
*
**/
pega_aplicacao_valor(_, [], [],_,0).
pega_aplicacao_valor(A, [A], [A1],A,A1).
pega_aplicacao_valor(A, [H|_], [H1|_], APLIC, VAL):-
    A = H,
    APLIC = H,
    VAL is H1.
pega_aplicacao_valor(A, [_|T], [_|T1], APLIC, VAL):-
    pega_aplicacao_valor(A, T, T1, APLIC, VAL),!.

```

Anexo II

Simulações Realizadas

Simulação 1: Perfil Renata

Renata quer ver se suas informações existem, pois ela tem quase certeza que não. Caso não exista, ela criará a conta. No próximo passo, verificará qual o lucro obterá nas aplicações que possui. Após isso, verifica se possui um fundo de emergência e em seguida descobrirá qual seu perfil de investidor, salvando-o em seu perfil. Por fim, sairá do programa.

Informações do perfil Renata:

Nome: renata.
Saldo: 3000.
Renda mensal: 2000.
Quantidade de dependentes: 0.
Prazo de retorno: 12.
Aplicações: tesouro_selic.
Quantia aplicada: 30 reais.
Perfil de investidor: nenhum.
Despesas mensais: 1500.

Renata verifica se suas informações existem:

```
Menu do programa:
Digite a opção que deseja escolher.
1 - Exibir as aplicações em que um cliente está investindo
2 - Verifica se um cliente possui um fundo de emergência.
3 - Verifica se um cliente possui um valor mínimo para investimento.
4 - Não conhece seu perfil de investidor? Responda este questionário e descubra!
5 - Ver informações de um cliente.
6 - Adicionar cliente.
7 - Onde aplicar?
8 - Quanto lucrarei nas aplicações que tenho?
0 - Para sair.
|: 5.
Digite o nome do cliente. Exemplo: ronaldinho gaúcho.
|: renata.
Que pena, não há sequer um cliente com este nome.
Retornando ao menu!
```

Renata cria sua conta:

```

Menu do programa:
Digite a opção que deseja escolher.
1 - Exibir as aplicações em que um cliente está investindo
2 - Verifica se um cliente possui um fundo de emergência.
3 - Verifica se um cliente possui um valor mínimo para investimento.
4 - Não conhece seu perfil de investidor? Responda este questionário e descubra!
5 - Ver informações de um cliente.
6 - Adicionar cliente.
7 - Onde aplicar?
8 - Quanto lucrarei nas aplicações que tenho?
0 - Para sair.
|: 6.
Nome do cliente (Exemplo: homer simpson):
|: renata.
Saldo do cliente:
|: 3000.
Renda mensal:
|: 2000.
Quantidade dependentes:
|: 0.
Prazo de retorno:
|: 12.
Aplicações (Exemplo: [tesouro selic,cdb,lci]):
|: [tesouro selic].
Quantia em cada aplicacao (Na mesma ordem que as aplicações. Exemplo: [30,1000,20]):
|: [30].
Perfil de investidor (Perfis disponiveis: nenhum; agressivo; conservador; moderado):
|: nenhum.
Despesas mensais:
|: 1500.
Usuário adicionado com sucesso!

```

Renata verifica quanto de lucro obterá nas aplicações que possui:

```

Menu do programa:
Digite a opção que deseja escolher.
1 - Exibir as aplicações em que um cliente está investindo
2 - Verifica se um cliente possui um fundo de emergência.
3 - Verifica se um cliente possui um valor mínimo para investimento.
4 - Não conhece seu perfil de investidor? Responda este questionário e descubra!
5 - Ver informações de um cliente.
6 - Adicionar cliente.
7 - Onde aplicar?
8 - Quanto lucrarei nas aplicações que tenho?
0 - Para sair.
|: 8.
Digite o nome do cliente. Exemplo: ronaldinho gauchó.
|: renata.
Lucro para tesouro selic: R$ 48.0

```

Renata verifica se possui um fundo de emergência:


```

Menu do programa:
Digite a opção que deseja escolher.
1 - Exibir as aplicações em que um cliente está investindo
2 - Verifica se um cliente possui um fundo de emergência.
3 - Verifica se um cliente possui um valor mínimo para investimento.
4 - Não conhece seu perfil de investidor? Responda este questionário e descubra!
5 - Ver informações de um cliente.
6 - Adicionar cliente.
7 - Onde aplicar?
8 - Quanto lucrarei nas aplicações que tenho?
0 - Para sair.
|: 2.
Digite o nome do cliente. Exemplo: ronaldinho gaúcho.
|: renata.

Que pena! Faltam 8970 reais para você atingir seu fundo de emergência!

```

Renata então verifica qual seu perfil de investidor:

```

Menu do programa:
Digite a opção que deseja escolher.
1 - Exibir as aplicações em que um cliente está investindo
2 - Verifica se um cliente possui um fundo de emergência.
3 - Verifica se um cliente possui um valor mínimo para investimento.
4 - Não conhece seu perfil de investidor? Responda este questionário e descubra!
5 - Ver informações de um cliente.
6 - Adicionar cliente.
7 - Onde aplicar?
8 - Quanto lucrarei nas aplicações que tenho?
0 - Para sair.
|: 4.
Digite o nome do cliente. Exemplo: ronaldinho gaúcho.
|: renata.

```

Olá renata. Faremos agora um questionário para descobrir seu perfil de investimento. Após isto, será feita uma comparação com o perfil de investimento cadastrado para você. Para cada pergunta, digite a letra que está entre parêntese, seguida de um ponto e dê enter. Exemplo: a.

----- Questionário -----

- 1 - Qual sua motivação para investir?
- a - Para não ficar com dinheiro parado em conta e ter risco de gastar.
 - b - Para possivelmente realizar alguns sonhos.
 - c - Para a obtenção de rendimentos e multiplicação da minha renda.

----- Digite sua Resposta -----

|: b.

- 2 - Você aceita perder uma porcentagem do recurso investido?
- a - Não, de forma alguma.
 - b - Se for pouco, ok.
 - c - Posso correr o risco caso os ganhos possam ser maiores.
- Digite sua Resposta -----

|: b.

- 3 - Como você poupa rendas extras (13º salário, restituição do Imposto de Renda...)?
- a - Faço investimento em aplicações que sejam seguras.
 - b - Coloco a maior parte em investimentos isentos de risco e uma parcela em modalidades um pouco mais arrojadas.
 - c - Aplico majoritariamente em investimentos de risco (mais chances de alto rendimento) e uma pequena parcela em aplicações conservadoras.
- Digite sua Resposta -----

|: a.

- 4 - E o planejamento para a aposentadoria?
- a - Estou poupanando e investindo em aplicações de baixo risco.
 - b - Uma parte do que guardo coloco em fundos conservadores e uma parte menor em investimentos mais agressivos.
 - c - Invisto sempre meus recursos em aplicações que gerem alto retorno financeiro.
- Digite sua Resposta -----

|: a.

- 5 - Um futuro sustentável financeiramente significa:
- a - Ter segurança financeira, mesmo em pouca quantia.
 - b - Guardar a maior parte do dinheiro de modo cauteloso e separando uma parte para aplicações de rendimento maior.
 - c - Investir cada vez mais (agressivamente), mesmo que tenha riscos.
- Digite sua Resposta -----

|: a.

Você tirou perfil Conservador. Parece que você não gosta de correr riscos ou busca por rendas estáveis a um curto/médio prazo



```
Deseja adicionar o perfil de investimento ao seu perfil?(s/n)
|: s.

Perfil de investimento alterado com sucesso!
```

Após, verificará suas informações, para saber se seu perfil de investidor foi alterado:

```
Perfil de investimento alterado com sucesso!
Menu do programa:
Digite a opção que deseja escolher.
1 - Exibir as aplicações em que um cliente está investindo
2 - Verifica se um cliente possui um fundo de emergência.
3 - Verifica se um cliente possui um valor mínimo para investimento.
4 - Não conhece seu perfil de investidor? Responda este questionário e descubra!
5 - Ver informações de um cliente.
6 - Adicionar cliente.
7 - Onde aplicar?
8 - Quanto lucrarei nas aplicações que tenho?
0 - Para sair.
|: 5.
Digite o nome do cliente. Exemplo: ronaldinho gaúcho.
|: renata.
Nome do cliente: renata
Saldo do cliente:3000
Renda mensal:2000
Quantidade dependentes:0
Prazo de retorno:12
Aplicações e quantia em cada aplicacao:
[tesouro selic]
[30]
Perfil de investidor:conservador
Despesas mensais:1500
```

Por fim, Renata sai do programa:

```
Perfil de investimento alterado com sucesso!
Menu do programa:
Digite a opção que deseja escolher.
1 - Exibir as aplicações em que um cliente está investindo
2 - Verifica se um cliente possui um fundo de emergência.
3 - Verifica se um cliente possui um valor mínimo para investimento.
4 - Não conhece seu perfil de investidor? Responda este questionário e descubra!
5 - Ver informações de um cliente.
6 - Adicionar cliente.
7 - Onde aplicar?
8 - Quanto lucrarei nas aplicações que tenho?
0 - Para sair.
|: 0.
```

Simulação 2: Perfil Humberto

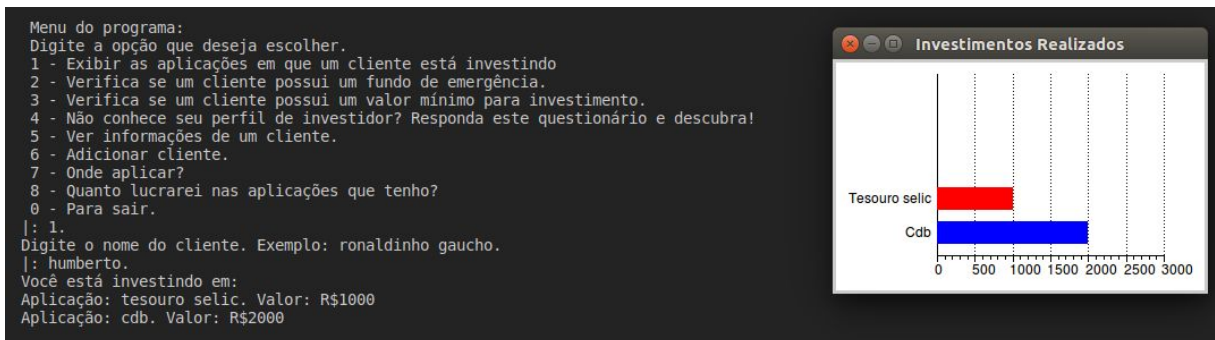
Humberto já possui uma conta no sistema. Ele deseja exibir as aplicações onde está investindo. Então, verificará se possui um valor mínimo de investimento. Após isto, ele verá suas informações de cliente. E então, ele executará a opção de “Onde aplicar?” do sistema. Nela, realizará aplicações condizentes com as regras estabelecidas pelo programa. E então, selecionará novamente as aplicações em que está investindo. Humberto, então, novamente

realizará aplicações, diversificadas desta vez. Ele visualizará seu perfil e por fim sairá do programa.

Informações do perfil Humberto:

Nome: humberto.
Saldo: 35000.
Renda mensal: 10000.
Quantidade de dependentes: 2.
Prazo de retorno: 36.
Aplicações: tesouro_selic, cdb.
Quantia aplicada: 1000 reais, 2000 reais.
Perfil de investidor: agressivo.
Despesas mensais: 5000.

Humberto exibe as aplicações onde está investindo:



Então, verificará se possui um valor mínimo de investimento.

```
Menu do programa:
Digite a opção que deseja escolher.
1 - Exibir as aplicações em que um cliente está investindo
2 - Verifica se um cliente possui um fundo de emergência.
3 - Verifica se um cliente possui um valor mínimo para investimento.
4 - Não conhece seu perfil de investidor? Responda este questionário e descubra!
5 - Ver informações de um cliente.
6 - Adicionar cliente.
7 - Onde aplicar?
8 - Quanto lucrarei nas aplicações que tenho?
0 - Para sair.
|: 3.
Digite o nome do cliente. Exemplo: ronaldinho gaúcho.
|: humberto.
Parabéns, você possui saldo o suficiente para investir.
```

Após isto, ele verá suas informações de cliente.


```

Menu do programa:
Digite a opção que deseja escolher.
1 - Exibir as aplicações em que um cliente está investindo
2 - Verifica se um cliente possui um fundo de emergência.
3 - Verifica se um cliente possui um valor mínimo para investimento.
4 - Não conhece seu perfil de investidor? Responda este questionário e descubra!
5 - Ver informações de um cliente.
6 - Adicionar cliente.
7 - Onde aplicar?
8 - Quanto lucrarei nas aplicações que tenho?
0 - Para sair.
|: 5.
Digite o nome do cliente. Exemplo: ronaldinho gauchão.
|: humberto.
Nome do cliente: humberto
Saldo do cliente:35000
Renda mensal:10000
Quantidade dependentes:2
Prazo de retorno:36
Aplicações e quantia em cada aplicacao:
[tesouro selic,cdb]
[1000,2000]
Perfil de investidor:agressivo
Despesas mensais:5000

```

E então, ele executará a opção de “Onde aplicar?” do sistema. Nela, realizará aplicações condizentes com as regras estabelecidas pelo programa.

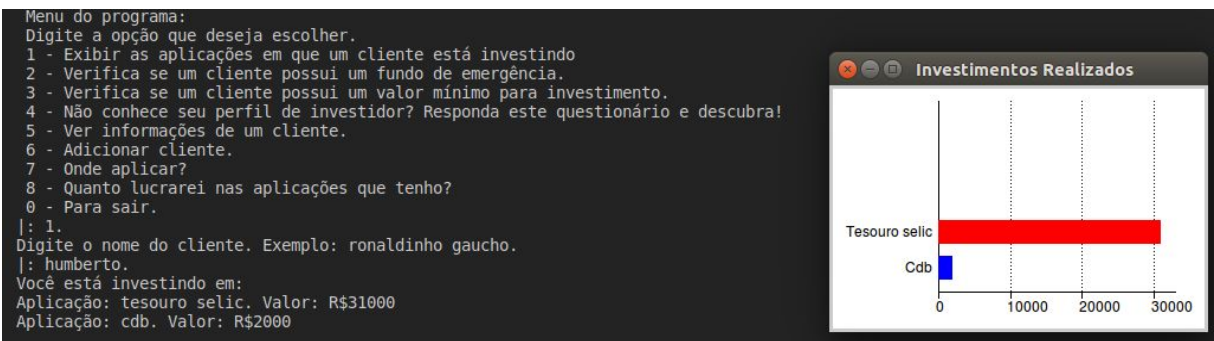
```

Menu do programa:
Digite a opção que deseja escolher.
1 - Exibir as aplicações em que um cliente está investindo
2 - Verifica se um cliente possui um fundo de emergência.
3 - Verifica se um cliente possui um valor mínimo para investimento.
4 - Não conhece seu perfil de investidor? Responda este questionário e descubra!
5 - Ver informações de um cliente.
6 - Adicionar cliente.
7 - Onde aplicar?
8 - Quanto lucrarei nas aplicações que tenho?
0 - Para sair.
|: 7.
Digite o nome do cliente. Exemplo: ronaldinho gauchão.
|: humberto.

Sinto muito, humberto. Faltam R$ 33000 para completar seu fundo de emergência!
É necessário montar/aumentar sua reserva de emergência antes de investir!
As opções para investimento são:
tesouro selic
cdb
Deseja aplicar diversificado nestas aplicações?(s/n)
|: n.
n
Deseja escolher quais?(s/n)
|: s.
Liste as aplicações onde quer investir. (Exemplo: [tesouro selic,cdb,lci])
É possível investir nestas aplicações:
[tesouro selic,cdb]
|: [tesouro selic].
Liste quanto quer investir nas aplicações, na mesma ordem que informou as aplicações.
(Exemplo: [3000,200,4000])
|: [30000].
Aplicação realizada com sucesso!

```

E então, selecionará novamente as aplicações em que está investindo.



Humberto, então, realizará um investimento diversificado:

```
Menu do programa:
Digite a opção que deseja escolher.
1 - Exibir as aplicações em que um cliente está investindo
2 - Verifica se um cliente possui um fundo de emergência.
3 - Verifica se um cliente possui um valor mínimo para investimento.
4 - Não conhece seu perfil de investidor? Responda este questionário e descubra!
5 - Ver informações de um cliente.
6 - Adicionar cliente.
7 - Onde aplicar?
8 - Quanto lucrarei nas aplicações que tenho?
0 - Para sair.
|: 7.
Digite o nome do cliente. Exemplo: ronaldinho gaúcho.
|: humberto.

Sinto muito, humberto. Faltam R$ 3000 para completar seu fundo de emergência!
É necessário montar/aumentar sua reserva de emergência antes de investir!
As opções para investimento são:
tesouro selic
cdb
Deseja aplicar diversificado nestas aplicações?(s/n)
|: s.
s
Quanto deseja investir?
Seu saldo é de: R$ 5000
|: 4000.
Aplicação realizada com sucesso!
```

Humberto verificará seu perfil, para descobrir qual seu saldo remanescente:

```

Menu do programa:
Digite a opção que deseja escolher.
1 - Exibir as aplicações em que um cliente está investindo
2 - Verifica se um cliente possui um fundo de emergência.
3 - Verifica se um cliente possui um valor mínimo para investimento.
4 - Não conhece seu perfil de investidor? Responda este questionário e descubra!
5 - Ver informações de um cliente.
6 - Adicionar cliente.
7 - Onde aplicar?
8 - Quanto lucrarei nas aplicações que tenho?
0 - Para sair.
|: 5.
Digite o nome do cliente. Exemplo: ronaldinho gaúcho.
|: humberto.
Nome do cliente: humberto
Saldo do cliente:1000
Renda mensal:10000
Quantidade dependentes:2
Prazo de retorno:36
Aplicações e quantia em cada aplicacao:
[tesouro selic,cdb]
[33000,4000]
Perfil de investidor:agressivo
Despesas mensais:5000

```

E por fim sairá do programa.

```

Menu do programa:
Digite a opção que deseja escolher.
1 - Exibir as aplicações em que um cliente está investindo
2 - Verifica se um cliente possui um fundo de emergência.
3 - Verifica se um cliente possui um valor mínimo para investimento.
4 - Não conhece seu perfil de investidor? Responda este questionário e descubra!
5 - Ver informações de um cliente.
6 - Adicionar cliente.
7 - Onde aplicar?
8 - Quanto lucrarei nas aplicações que tenho?
0 - Para sair.
|: 0.

```