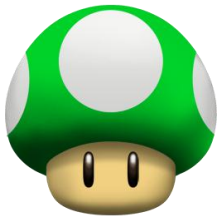


# INF 1771 – Inteligência Artificial

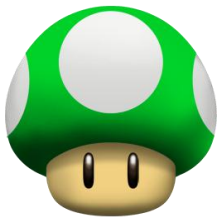
Aula 10 – Utilizando o SWI-Prolog em C++

Edirlei Soares de Lima  
<elima@inf.puc-rio.br>



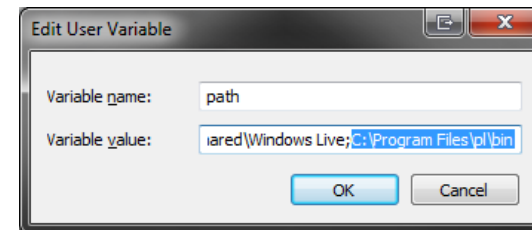
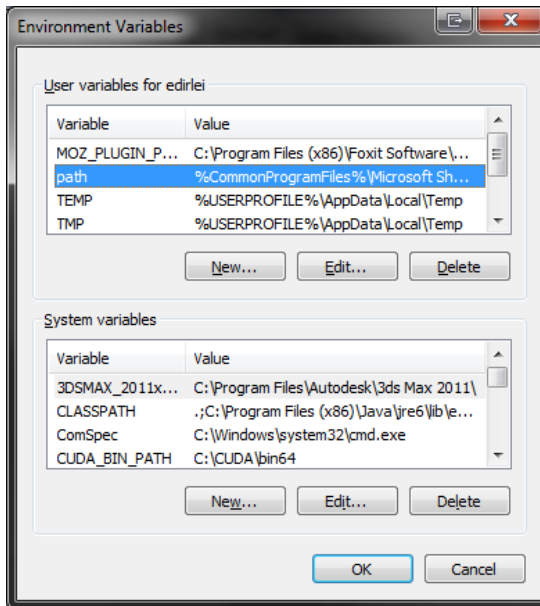
# SWI-Prolog

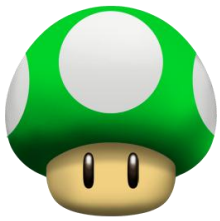
📄 <http://www.swi-prolog.org/download/stable>



# Configuração

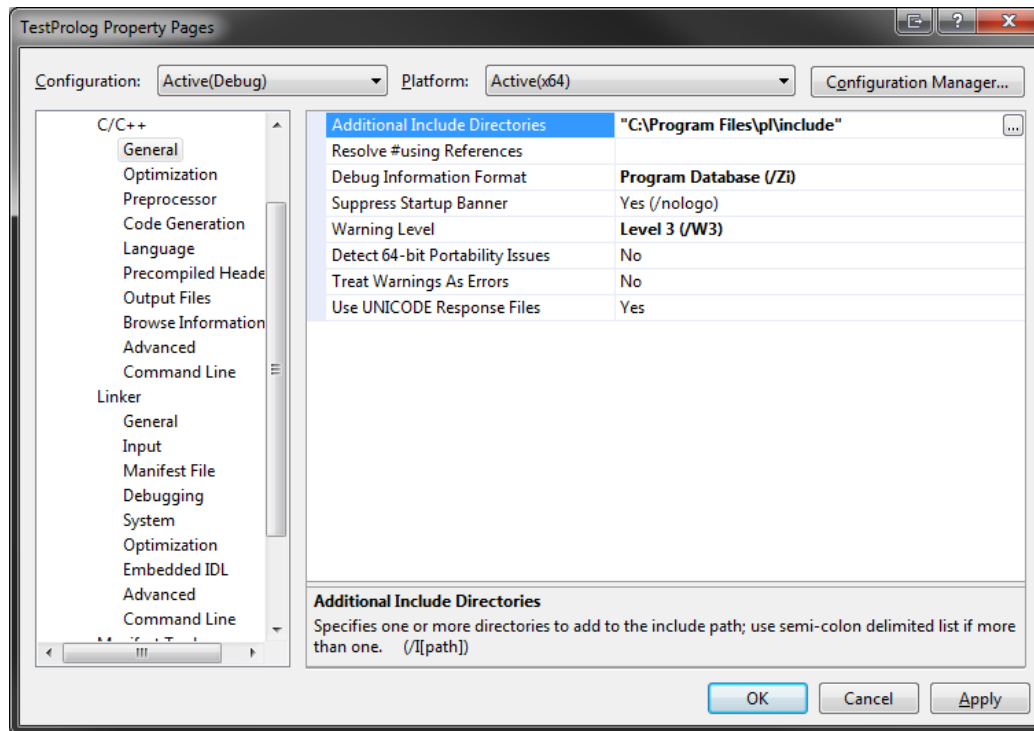
- ❏ Control Panel -> System ->Advanced-> Environment Variables
- ❏ Adicionar o diretório "C:\Program Files\pl\bin" a variável "PATH" do sistema.





# Configuração – Visual Studio

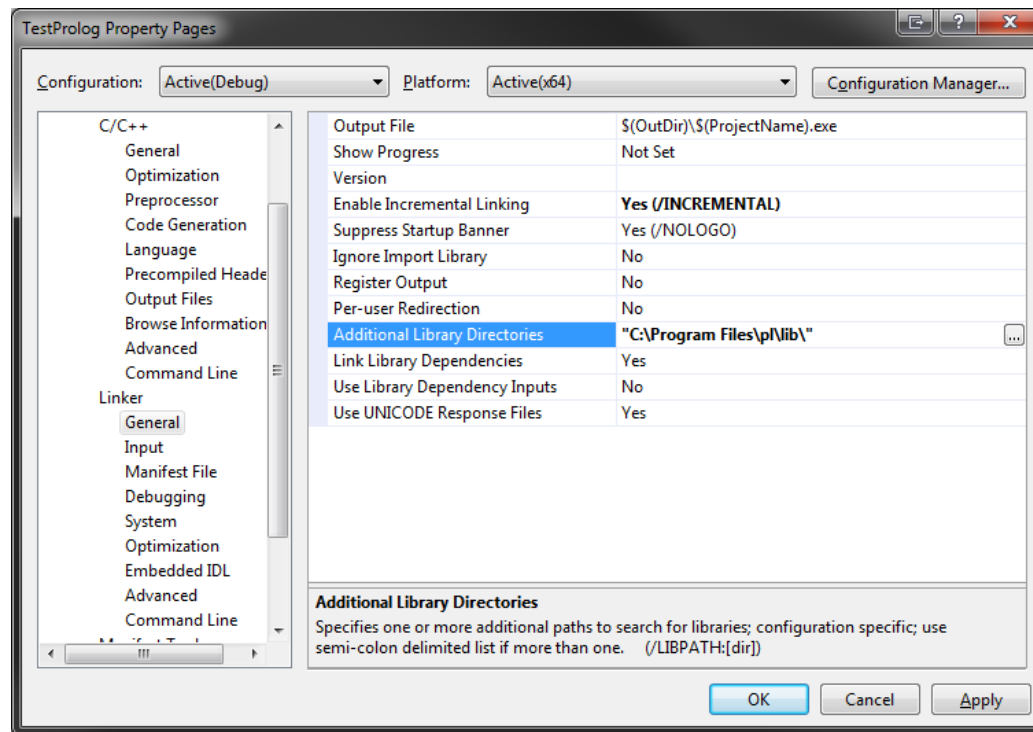
- 🔑 Include Directory: "C:\Program Files\pl\include"

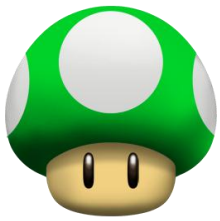




# Configuração – Visual Studio

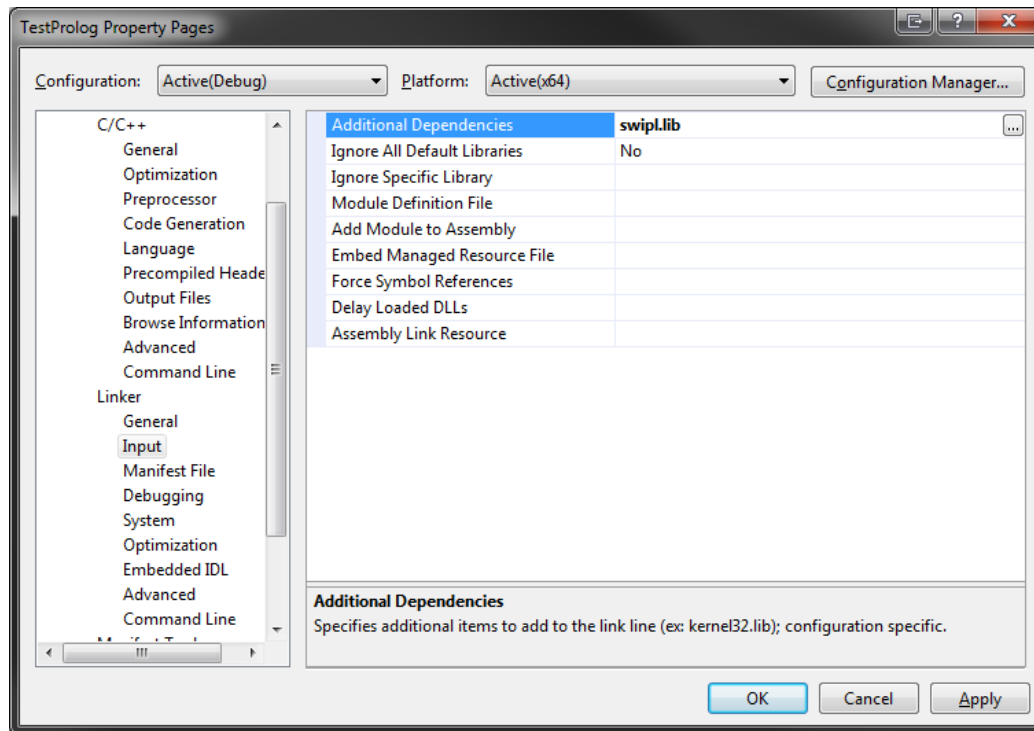
📌 Library Directory: "C:\Program Files\pl\lib\"

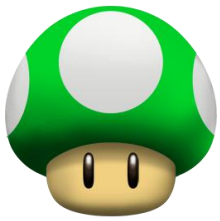




# Configuração – Visual Studio

🔑 Dependência: swipl.lib



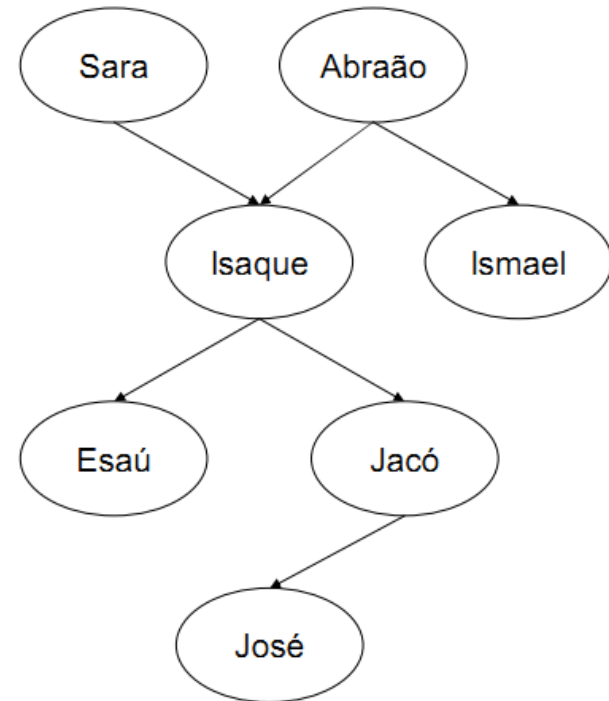


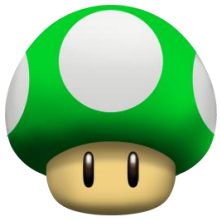
# Exemplo de Programa

```
progenitor(sara,isaque).  
progenitor(abraao,isaque).  
progenitor(abraao,ismael).  
progenitor(isaque,esau).  
progenitor(isaque,jaco).  
progenitor(jaco,jose).
```

```
mulher(sara).  
homem(abraao).  
homem(isaque).  
homem(ismael).  
homem(esau).  
homem(jaco).  
homem(jose).
```

```
filho(Y,X) :- progenitor(X,Y).  
mae(X,Y) :- progenitor(X,Y), mulher(X).  
avo(X,Z) :- progenitor(X,Y), progenitor(Y,Z).  
irmao(X,Y) :- progenitor(Z,X), progenitor(Z,Y).  
ancestral(X,Z) :- progenitor(X,Z).  
ancestral(X,Z) :- progenitor(X,Y), ancestral(Y,Z).
```





# Exemplo de Programa

```
#include <SWI-cpp.h>
#include <iostream>

using namespace std;

int main(){
    char* argv[] = {"swipl.dll", "-s", "D:\\\\teste.pl", NULL};

    PlEngine e(3,argv);

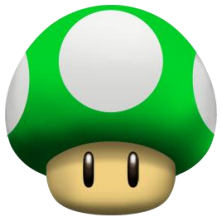
    PlTermv av(2);
    av[1] = PlCompound("jose");

    PlQuery q("ancestral", av);

    while (q.next_solution())
    {
        cout << (char*)av[0] << endl;
    }

    cin.get();
    return 1;
}
```





# Manual

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