



PONTIFICIA UNIVERSIDAD CATÓLICA MADRE Y MAESTRA.

Facultad de Ingeniera

Nombres:

Wendy Esther Reyes Perez

Héctor Rafael Aristy Melo

Matricula:

2010-6471

2011-5972

Profesor:

Lisibonny Beato

Grupo:

001

Materia:

Programacion Logica

Practica:

Examen Final

Carrera:

Ing. Sistemas y Computación.

Fecha:

20/07/2015

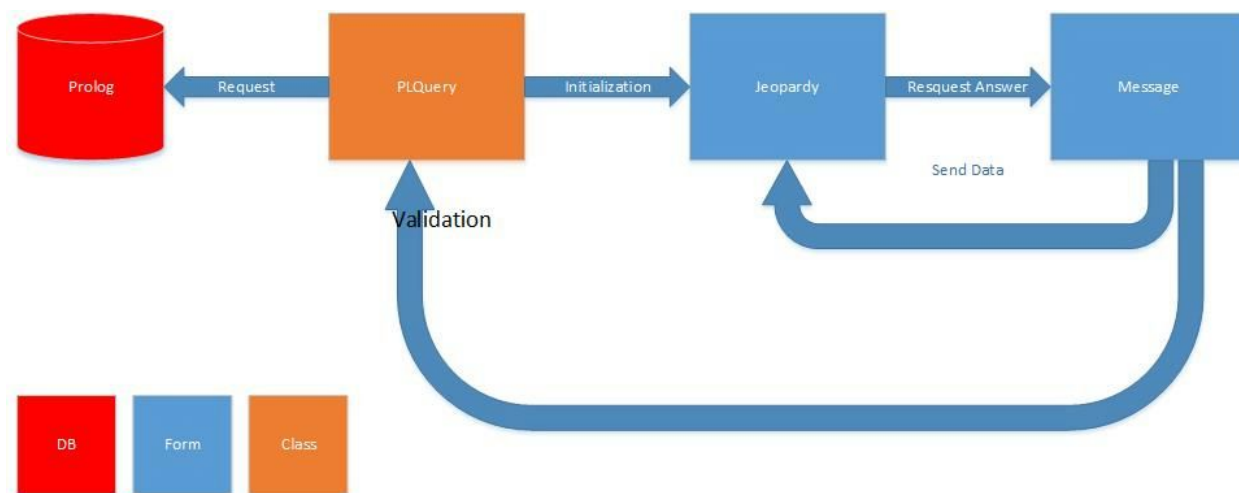
Introduccion

En nuestra asignación como proyecto final , la idea consistía en generar una interfaz visual a través de un lenguaje de programación. El lenguaje en el cual fue implementado nuestra interfaz fue en C#, utilizando el dll que nos fue proporcionado de este enlace: <http://www.swi-prolog.org/contrib/CSharp.html>. Es recomendable para que este se tenga instalado SWI-Prolog en 32Bits.

Para poder usar el proyecto se deberán realizar las siguientes modificaciones:

1. Los archivos que necesitan estan dentro de la carpeta Prolog_Final.
2. La referencia de SwiPICs.dll en el proyecto debe agregarse ya que estan fijadas en las laptops que se trabajaron y tendrá una ruta equivocada en otro ordenador.
3. En el método Prolog_Connection de la clase PLQuerys, deben configurarse el location del .pl de param.
4. El archivo que se le debe hacer referencia a param es donde este ubicado examen-final.pl.

Arquitectura del sistema



En la arquitectura se utilizaron dos patrones de diseño: MVC y Singleton. MVC se utilizó para traer la data desde prolog, una clase la maneja y en una vista(form) se visualiza. Singleton se utilizó para trabajar con una sola instancia ya que dependemos de la información del form message la cual es pasada a Jeopardy sin tener que crear nuevas instancias.

Dicho esto podemos decidir que nuestro puente que conecta nuestros 2 lenguajes es el PLQuery, ya que este hace las consultas a prolog a través de string concatenada con valores específico de un hecho o regla. A continuación explicamos que hace cada método que se utiliza:

- **public void Prolog_Connection()** : Como su nombre lo dice es el método que se realiza el vínculo con nuestro archivo .pl de prolog.
- **public int P_Score = 0;** Es el score del Player
- **public int Cpu_Score = 0;** : Es el score de la maquina.
- **public int Get_Category(string Categoria):** Este metodo se utiliza para hacer una consulta a nuestro hecho categoria para saber que tiempo darle en cada pregunta de cada categoría.
- **public string Get_Question(string Name_Category, string puntaje):** Es el método que nos muestra la pregunta de la categoría que vale esa cantidad de puntaje.
- **public string Answer(string Categoria,string Pregunta,string Answer)** : Es el método que nos compara que si la respuesta del usuario es igual a la que tenemos almacenada en nuestro .pl. En caso de que sea correcto este devuelve “correcto” en caso contrario “incorrecta”.
- **public int Point_Answer(string Categoria, string Pregunta):** Es el método que nos otorga los puntos de la pregunta si nuestra respuesta fue acertada.
- **public string Get_MachineAnswer(string Estado):** Es el método que nos otorga la respuesta de la máquina , dependiendo del estado de ella (TimeOut,Incorrecto,Correcto).

Conclusion

Con la elaboración de nuestro proyecto, fuimos capaces de manejar 2 lenguajes completamente distintos y hacer que sea friendly para el usuario. Se aprendió que prolog en sí puede ser usado como una base de datos y haciendo consultas un poco parecida a SQL Server podemos obtener la data que deseamos mostrar o comparar con el input del usuario.