ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

"SISTEMA EXPERTO PARA LA DETECCION DE ERRORES EN LOS REGISTROS DE UN REPOSITORIO DE DATOS"

TESINA DE SEMINARIO

Previo a la obtención del Título de:

Presentado por:

José Luis Barona Valencia

INGENIERO EN CIENCIAS COMPUTACIONALES ESPECIALIZACIÓN SISTEMAS DE INFORMACIÓN

Salvador Eduardo Salvatierra Samaniego

INGENIERO EN CIENCIAS COMPUTACIONALES ESPECIALIZACIÓN SISTEMAS DE INFORMACIÓN

Luis Ricardo Zúñiga Terán

INGENIERO EN CIENCIAS COMPUTACIONALES ESPECIALIZACIÓN SISTEMAS TECNOLÓGICOS

GUAYAQUIL – ECUADOR

AÑO: 2012

AGRADECIMIENTO

José Luis Barona Valencia

A Dios por su infinita misericordia conmigo, por mantener feliz con los seres que más quiero, por permitirme vivir y seguir adelante con los sueños que deseo alcanzar.

A mis padres: a mi madre Norma Valencia de Barona, por ser el símbolo de la fidelidad, cariño, amor, ternura y dedicación que siempre tuvo con la familia, a mi padre Segundo Barona, hombre luchador, dedicado, trabajador que siempre quiso dar a sus hijos una educación profesional y cristiana.

A mis hermanos por su constante apoyo moral y educativo; Leonardo: en matemáticas; Danny: en física; Alexandra: en química, Jaime: en Inglés, a mis sobrinos Jaimito y Alexita por inyectar más felicidad a mi vida.

A mi novia María Granda, por ser la luz de mi vida y mi razón de vivir en este mundo.

AGRADECIMIENTO

Luis Ricardo Zúñiga Terán

Agradezco primeramente a Dios por permitirme vivir cada día, por su ayuda y apoyo en cada momento duro que tuve en mi vida y me permitió siempre salir adelante y nunca desfallecer.

A mi padre Luis Enrique Zúñiga Dager quien con su trabajo y esfuerzo me permitió estudiar siempre apoyándome incondicionalmente.

A mi madre Alexandra Mayoder Terán Armijos quien a pesar de las dificultades económicas siempre encontró la manera de apoyarme en cada cosa que necesitaba para mi crecimiento profesional y personal.

AGRADECIMIENTO

Salvador Salvatierra

Agradezco primeramente a Dios por guiarme y poner una maravillosa familia en mi camino que me ha brindado su apoyo en cada objetivo que me he planteado.

Esta meta fue lograda en conjunto con mis tíos Flavio y Lyonel Samaniego que estuvieron a mi lado apoyándome con lo que necesitaba, mi madre y mi abuela Sonia y Judith que estaban pendiente de mi salud y también mis hermanos Sonia y Jairo que fueron el gran apoyo y fuerza para terminar este objetivo.

En especial a mi esposa Moi y mi Hijo Eduardo por enseñarme el significado de amor incondicional y apoyo.

D E D I C A T O R I A

José Luis Barona Valencia

Todo el esfuerzo, dedicación y perseverancia que he puesto para alcanzar este logro lo dedico en primer lugar a Dios, ya que sin él nada es posible, gracias a su bendición soy quién soy, también lo dedico a mi padres Norma y Segundo quienes siempre creyeron en mí, de los cuales he recibido y sigo recibiendo su apoyo constante, dedicado y amoroso.

También quiero dedicar este trabajo a mis hermanos quienes representan en mí el ejemplo a seguir tanto en el aspecto profesional como moral.

D E D I C A T O R I A

Luis Ricardo Zúñiga Terán

Dedico este título primeramente a Dios quien con su bendición me permitió conseguir este objetivo que me había planteado en mi vida, ya que sin el nada es posible.

En segundo lugar dedico este trabajo a mis padres quienes siempre creyeron en mí y siempre me apoyaron luchando a cada momento a mi lado, a ustedes este trabajo.

Muchas gracias por todo su apoyo y amor gracias por confiar en mí.

DEDICATORIA

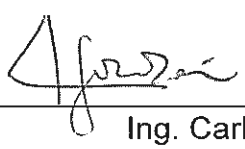
Salvador Salvatierra

Dedico este título a mi familia mi gran apoyo y fortaleza. Una vez más confirmo que Dios provee.

TRIBUNAL DE SUSTENTACIÓN

Phd. Indira Nolivos

Profesor del Seminario



Ing. Carlos Jordan

Profesor Delegado del Decano

DECLARACION EXPRESA

"La responsabilidad del contenido de este Trabajo de Grado, nos corresponde exclusivamente, y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITECNICA DEL LITORAL"

José Luis Barona Valencia

Salvador Eduardo Salvatierra Samaniego

Luis Ricardo Zúñiga Terán

Resumen

Este trabajo presenta el desarrollo de una aplicación multimedia de acceso web para la detección de errores en una base de datos centralizada. Esta aplicación hace uso de un modelo experto, como fuente del conocimiento, construido en base al conocimiento de un grupo de expertos en el área.

El objetivo del proyecto es el de desarrollar una aplicación que permita detectar los datos erróneos en un repositorio de datos, para su posterior depuración por parte de los expertos permitiendo de esta manera una disminución significativa en su tiempo de trabajo.

El conocimiento adquirido de los expertos es modelado en un árbol de decisión, en cuyos nodos se encuentran las validaciones para clasificar los registros como correctos o erróneos, en este último caso son catalogados dependiendo del tipo de error.

Esta herramienta puede ser usada por el Departamento de Calidad de Datos de la compañía como una herramienta para el control y depuración de los datos.

Índice

Resumen IX

Introducción XV

Capitulo 1 1

1.1. Antecedentes 1

1.1.1. Naturaleza de la Empresa ..... 1_Toc333235011

1.1.2. Definición del Problema 3

1.2. Objetivos 4

1.3. Alcance 4

Capitulo 2 6

2.1. Marco Teórico 6

2.1.1. Sistema Experto 6

2.1.2. Árbol de decisiones 9

2.1.3. Web Service Description Language (WSDL) 10

2.1.4. Procesamiento de Datos. 10

2.1.5. Reglas de Inferencia 10

Capitulo 3 13

3.1. Metodología 13

3.1.1. Adquisición del Conocimiento..... 13

Capitulo 4 16

4.1. Diseño de la aplicación..... 16

4.1.1. Elecciones de Herramientas de Desarrollo..... 16

4.2. Implementación 17

4.3. Metodología de Prueba..... 21

Capitulo 5 22

5.1. Casos de Uso 22

5.2. Análisis de Resultados..... 27

5.4. Evaluación de Resultados 30

Conclusiones 31

Recomendaciones..... 32

Índice de Figuras

Figura 1.1. Funcionamiento de las bases de datos de la compañía DataDecision 2

Figura 2.1. Arquitectura de un Sistema Experto 8

Figura 3.1. Árbol de Conocimiento 14

Figura 3.2. Recorrido del Árbol de Conocimiento 15

Figura 4.1. Pantalla de login del sistema 18

Figura 4.2. Invocación del Web Services al presionar el botón Cargar Datos, el cual devuelve el conjunto de datos con los id que deberán ser evaluados 19

Figura 4.3. Invocación del web service que recibe el id a evaluar, y devuelve un arreglo binario para su posterior análisis por parte del motor de inferencia que indica si es correcto o erróneo. 19

Figura 4.4. Datos evaluados por parte del Sistema de Calidad de Datos.. 20

Figura 4.5. Guardar los datos que ya han sido revisados por parte del usuario. 20

Figura 4.6. Gráfico estadístico con los resultados del análisis de los datos. 21

Figura 5.1. Opciones de Usuario en el Sistema de Calidad de Datos 22

Figura 5.2. Gráfico de Barras de la cantidad de errores por tipo de error obtenidos por el Sistema de Calidad de Datos..... 28

Índice de Tablas

Tabla 5.1. CU-01 24

Tabla 5.2. CU-02 25

Tabla 5.3. CU-03 27

Tabla 5.4. Análisis de Resultados 28

Tabla 5.5. Evaluación de Resultados 29

Introducción

A medida que una compañía va creciendo sus bases de datos van creciendo y ello conlleva a que mucha de su data tenga cierto grado de toxicidad, lo cual produce que los reportes que son generados sean poco confiables para las personas que toman decisiones.

El presente trabajo se enfoca en el control de la calidad de los datos de un repositorio de datos donde la solución puede ser soportada por el uso de una variedad de modelos, en esta categoría de modelos entran la búsqueda heurística, la clasificación, la regresión, la optimización y muchos otros.

Así mismo las herramientas que se pueden emplear en las construcción de estos modelos son muy variadas, por ejemplo: redes neuronales, algoritmos genéticos, arboles de decisión, etc.

El presente trabajo aplica técnicas de árboles de decisión para modelar el conocimiento concerniente al control de la calidad de datos en base de datos centralizadas. Esta técnica fue elegida debido a que:

- Proporciona un alto grado de comprensión del conocimiento en la toma de decisiones.
- Claramente plantean el problema para que todas las opciones sean analizadas.

- Permiten analizar totalmente las posibles consecuencias de tomar una decisión.
- Proveen un esquema para cuantificar el costo de un resultado y la probabilidad de que suceda.
- Nos ayuda a realizar las mejores decisiones sobre la base de la información existente y de las mejores suposiciones.

Como resultado del análisis realizado por el sistema experto obtenemos un conjunto de registros clasificados como correctos e incorrectos los cuales en caso de ser incorrectos son agrupados por tipo de error.

Capitulo 1

1.1. Antecedentes

1.1.1. Naturaleza de la Empresa

DataDecision es una compañía de Procesamiento masivo de Datos que lleva 80 años en el mercado con más de 250 sucursales en todo el país procesando alrededor de 50000 registros diarios. Con el pasar de los años el crecimiento de esta compañía lo obligó a cumplir con normas y estándares de calidad decidiendo así adquirir un nuevo Software que mejoraría todos los procesos, ayudaría a manejar mayor cantidad de datos y permitiría llevar un correcto control de la información.

En el proceso de implementación del nuevo sistema, la compañía tuvo la desventaja de un proceso de implementación costosa y lenta tomando alrededor de 1 mes para que pueda ser usado en cada sucursal, actualmente en 2 años solo han podido ser actualizadas 24 de las 250 sucursales, teniendo así DataDecision que convivir con estas dos aplicaciones al mismo tiempo.

En este punto la compañía tiene una base de datos antigua (Histórico) con muchos datos contaminados y una base de datos limpia (Sistema Nuevo) sin toda la información requerida para poder trabajar independientemente por lo que se decidió crear una base de datos de Transición que serviría de filtro de la información evitando así contaminar la base de datos Nueva.

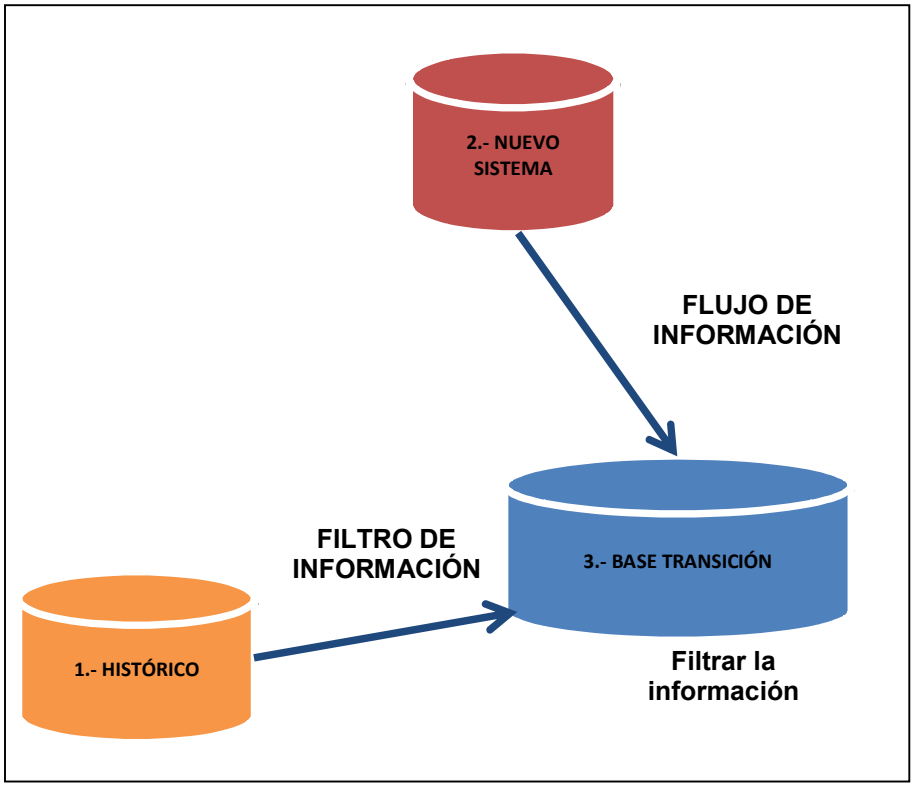


Figura 1.1. Funcionamiento de las bases de datos de la compañía DataDecision

1.1.2. Definición del Problema

Debido a la gran cantidad de datos que procesa la compañía un porcentaje de estos son datos erróneos, los cuales deben ser debidamente validados.

Para realizar el proceso de validación de datos la compañía cuenta con un departamento de Calidad de Datos el cual está compuesto de 5 expertos y un jefe de Departamento. El trabajo de los expertos consiste en buscar dentro de toda la base de datos aquellos registros que se encuentren erróneos basándose en reglas que el departamento ya ha establecido en base a su experiencia.

El tiempo que tardan dichos expertos en encontrar esos registros y enviar el informe solicitando la corrección es de aproximadamente una semana. Durante esa semana el sistema sigue recibiendo datos y por lo tanto sigue recibiendo errores. Lo cual hace que los expertos estén trabajando sobre datos ya obsoletos y siempre estén trabajando de manera correctiva.

Debido al tiempo que toma a los expertos realizar su trabajo comparado con el volumen de datos erróneos que ingresan diariamente esto se vuelve un proceso crítico.

1.2. Objetivos

Dado los problemas expuestos en el punto anterior se plantea el desarrollo de un Sistema Experto, el cual haga uso de las reglas bajo las cuales el Departamento de Calidad de Datos se rige para encontrar los datos erróneos.

El sistema deberá cumplir con los siguientes alcances:

- Garantizar un proceso de mejora continua de la calidad de datos de la información de la compañía.
- Optimizar las tareas de gestión que actualmente están congestionadas presentando una mejora real en tiempos de hasta el 90%.
- Procesar la misma cantidad de registros en por lo menos la mitad del tiempo de como lo hacen ahora (5000 registros en 1 semana laboral).
- Tener un porcentaje de acierto en la identificación de los registros superior al 90%.

1.3. Alcance

La compañía posee alrededor de 40 reglas que se usan para el filtrado de los registros, pero según conversaciones con el Jefe del Departamento Ing. Daniel Mosquera las más comunes son 8, es decir el sistema cumple con el Principio de Pareto[1].

Dado este antecedente se van a proceder a ingresar solo 8 de estas reglas que son consideradas las más importantes dentro del Sistema.

Capitulo 2

En este capítulo se detalla el marco teórico utilizado en el desarrollo de la aplicación web para la detección de registros incorrectos en un repositorio de datos, así mismo se detalla la construcción del modelo experto que interactúa con la aplicación.

2.1. Marco Teórico

2.1.1. Sistema Experto

Un Sistema Experto $(SE)^2$, es un programa basado en conocimientos que lleva a cabo tareas que generalmente sólo realiza un experto humano; es decir, es un programa que imita el comportamiento humano en el sentido de que utiliza la información que le es proporcionada para poder dar una opinión sobre un tema en especial.

Un SE debe cumplir con ciertas características que son propias de los expertos humanos.

- Habilidad para llegar a una solución de forma rápida y certera tal cual lo haría un experto puesto que él posee no solo conocimiento sobre un campo específico si no también posee experiencia en el mismo.

- Habilidad para explicar los resultados de forma clara a aquellas personas que no posean el conocimiento, respondiendo a preguntas, a los razonamientos derivados de las mismas y las implicaciones subsecuentes para que ellos puedan entender y aprender de dichos resultados.

- Habilidad para aprender de las experiencias, al igual que los expertos humanos deben aprender tanto de sus propias experiencias como de los demás, están obligados a tener actualizada su base de conocimientos así como modificar su razonamiento.

- Conciencia de sus limitaciones, una vez que tienen un problema evaluar si el mismo está dentro de sus limitaciones de resolución.

Los Sistemas Expertos constan de los siguientes componentes:

- **Base de Conocimientos:** Contiene las reglas sobre las cuales vamos a determinar el comportamiento del sistema, y las relaciones existentes entre sus componentes. La base de conocimiento puede ser vista como una especie de base de datos para la gestión del conocimiento, el aspecto más importante que se debe considerar al momento de su creación es la calidad de la información que va a contener. El conocimiento obtenido puede venir de diferentes fuentes, pudiendo ser estas: formal (probada) y empírica (en base a la

experiencia).En el caso de este proyecto son todas aquellas reglas que se han podido capturar mediante las entrevistas y conversaciones con los expertos.

- **Base de Hechos:** Contiene las evidencias o información sobre el estado de los diferentes componentes.

- **Motor de Inferencia:** Es el que relaciona las reglas almacenadas en la base de conocimientos con los acontecimientos almacenados en las base de hechos, para la obtención de conclusiones o la inferencia de nuevos conocimientos.

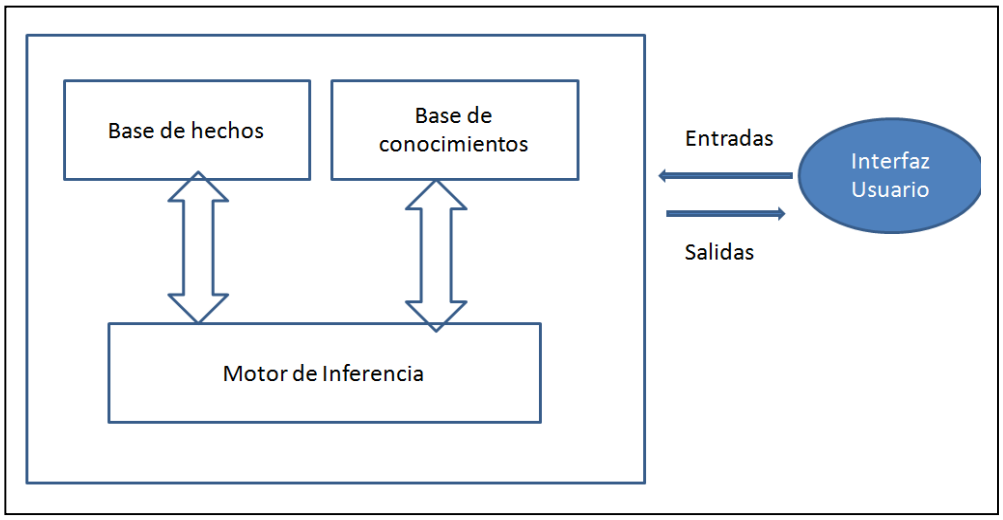


Figura 2.1. Arquitectura de un Sistema Experto

2.1.2. Árbol de decisiones

El árbol de decisión es un diagrama que representa en forma secuencial condiciones y acciones; muestra qué condiciones se consideran en primer lugar, en segundo lugar y así sucesivamente. Este método permite mostrar la relación que existe entre cada condición y el grupo de acciones permisibles asociado con ella. Es decir categorizan una serie de condiciones que ocurren de manera sucesiva[3].

Un árbol de decisión lleva a cabo un test, a medida que va haciendo un recorrido comenzando desde su nodo raíz avanzando hasta los nodos de nivel inferior hasta encontrar una solución o decisión al problema planteado. En cada nodo se encuentran variables que servirán para poder realizar evaluaciones las cuales determinan el camino que el árbol deberá seguir.

El desarrollo de árboles de decisión beneficia al analista haciéndolo describir condiciones y acciones que llevan a identificar de manera formal las decisiones que actualmente deben tomarse. De esta forma, es difícil para ellos pasar por alto cualquier etapa del proceso de decisión, sin importar que este dependa de variables cuantitativas o cualitativas. Los árboles también obligan a los analistas a considerar la consecuencia de las decisiones.

2.1.3. Web Service Description Language (WSDL)

Es un lenguaje creado para describir y publicar los formatos y protocolos de los servicios web de forma estándar. Está basado en XML, estandarizado por la W3C y permite definir la forma en la cual se realizarán las comunicaciones, es decir los requisitos del protocolo y los formatos de los mensajes necesarios para interactuar con otros servicios.

Las operaciones y mensajes que soporta se describen en abstracto para posteriormente enlazarse al protocolo concreto del la red y al formato de los mensajes.

2.1.4. Procesamiento de Datos.

Es la Técnica que consiste en la recolección de los datos primarios de entrada, que son evaluados y ordenados, para obtener información útil, que luego serán analizados por el usuario final, para que pueda tomar las decisiones o realizar las acciones que estime conveniente.

2.1.5. Reglas de Inferencia

Las reglas de inferencia son esquemas básicos deductivos que se suelen escribir poniendo cada premisa en una línea y la conclusión en otra línea al final.

Las reglas de inferencia aplican la regla “Modus Ponens” la cual dice que: “Si una premisa tiene la forma de condicional y la otra afirma el antecedente entonces es válida la inferencia que consiste en afirmar el consecuente”. Las reglas proporcionadas por el departamento de Calidad de Datos se encuentran en el *Anexo de Captura del Conocimiento, Página 2*

Tomando como ejemplo en este proyecto una regla proporcionada por el departamento de Calidad de Datos tenemos lo siguiente:

Si el ciudadano tiene su cédula registrada en la base de datos nueva Y la misma coincide en la base de datos de transición, pero los nombres que se encuentran en la base de datos nueva no coinciden con los nombres de la base de datos de transición, entonces existe un error tipo 1.

En el párrafo anterior se muestra la premisa ***Si el ciudadano tiene su cédula registrada en la base nueva,*** la cual está unida mediante la conjunción ***Y*** a la premisa ***la misma coincide en la base transición, pero los nombres que se encuentran en la base de datos nueva no coinciden con los nombres de la base de datos de transición.*** Tomando estas dos premisas como verdaderas nuestro consecuente es ***Entonces existe error tipo 1***.

De esta manera es como se pueden escribir las diferentes reglas de inferencia que se necesitaran. La implementación de todas las reglas

proporcionadas por el departamento de Calidad de Datos se encuentran en el *Anexo de Captura del Conocimiento, Página 6*

Capitulo 3

En este capítulo se detalla la Metodología aplicada para el desarrollo del proyecto, se redacta como se adquirió el conocimiento del Departamento de Calidad de Datos, como se lo modeló y como fue representado en un Árbol de Decisiones.

3.1. Metodología

3.1.1. Adquisición del Conocimiento

Para el planteamiento de la solución se realizó varias visitas técnicas para ver el funcionamiento y acciones realizadas en el departamento de Calidad de Datos. Se necesitó trabajar con ellos durante un tiempo adquiriendo su conocimiento de manera empírica tomando las herramientas que ellos usan como filtros, documentos de Excel y variables a ser analizadas.

Una vez recogido el conocimiento se bosquejó de manera empírica un árbol de Decisión, el cual fue posteriormente revisado y validado con el jefe del Departamento de Calidad de Datos el cual nos dio su aprobación quedando el gráfico de la siguiente manera.

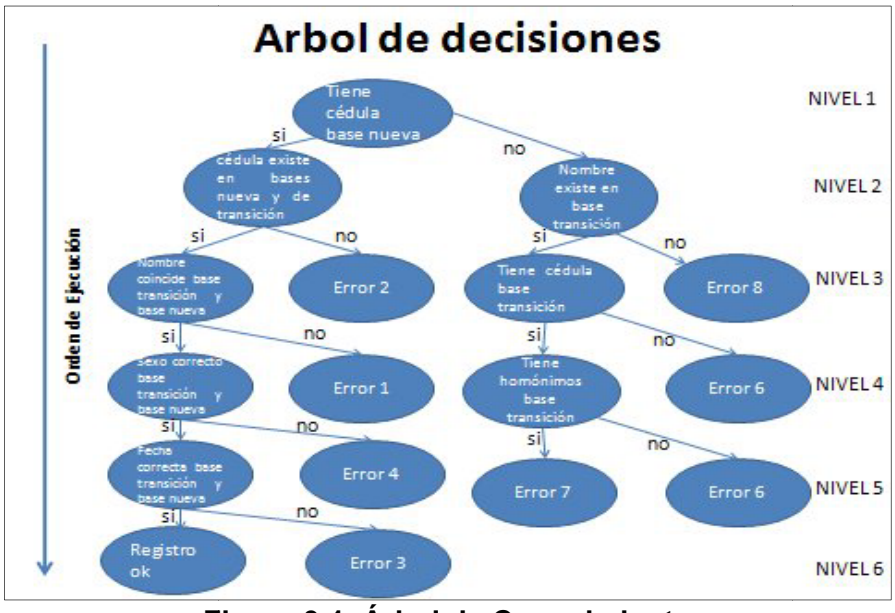


Figura 3.1. Árbol de Conocimiento

Como se puede observar en el gráfico, en los nodos del Árbol se encuentran las variables que se utilizan para recorrer el árbol y seleccionar una de las ramas hasta encontrar un nodo resultado. Estas variables son las que serán encapsuladas en el motor de inferencia.

Por ejemplo para recorrer el árbol empezamos con el nodo raíz como se muestra en la figura**.**

¿Ciudadano tiene cédula en la base nueva?, es la variable representada por el nodo raíz.

Los posible valores que puede tomar una variable son **sí** o **no**, lo cual nos indica a donde debemos movernos si al nodo derecho en caso de tener una

respuesta afirmativa o al nodo izquierdo en caso de tener una respuesta negativa. Para continuar con el ejemplo la persona ***no*** posee cédula, entonces viajamos al siguiente nodo y nos encontramos con otra variable la cual nos pregunta: **¿Nombre ciudadano existe en la base de transición?,** si la respuesta es **no**, el siguiente nodo ya es un nodo que contiene la respuesta y ahí detenemos el recorrido del árbol ya que obtenemos que el registro posee un error de tipo 8.

A continuación se presenta un gráfico describiendo el comportamiento del ejemplo previamente citado.

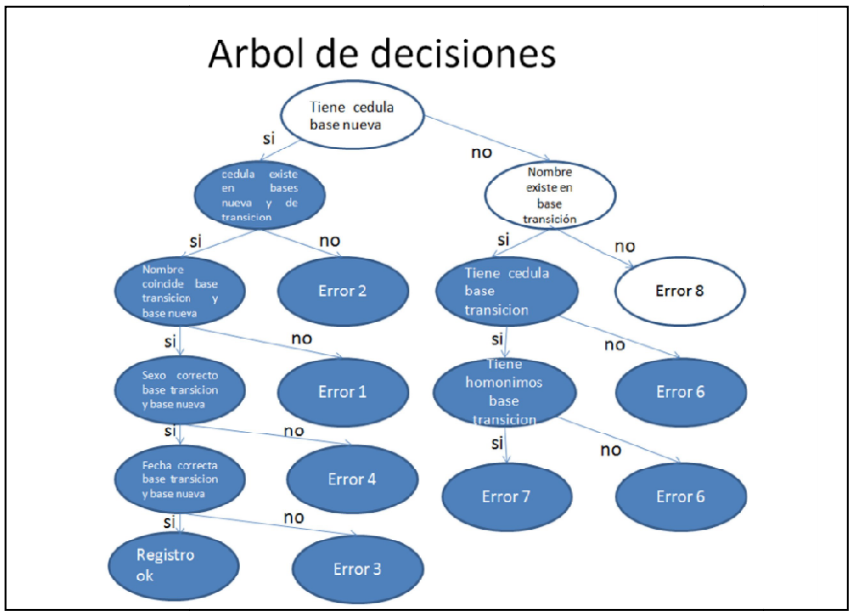


Figura 3.2. Recorrido del Árbol de Conocimiento

Capitulo 4

En este capítulo se redacta como se diseñó e implementó la aplicación. Se describe y se justifica las herramientas que fueron utilizadas para la implementación del proyecto así como se ilustran las diferentes pantallas de la aplicación web.

4.1. Diseño de la aplicación

En esta sección se presenta el detalle de cómo se diseñó y modeló la solución planteada especificando la tecnologías, lenguajes y la decisión detrás de la elección de cada herramienta.

4.1.1. Elecciones de Herramientas de Desarrollo

- **Motor de Inferencia:** El motor de Inferencia seleccionado es Prolog por tener mayor cantidad de foros y guías de aprendizaje, además de poseer un entorno robusto y fácil de manejar el cual nos permitió la integración con diferentes lenguajes de programación a través de varias interfaces.

- **Lenguaje de Programación:** El lenguaje de programación escogido fue ASP.NET escribiendo la sintaxis en C Sharp (C#), debido a la

robustez del lenguaje, los integrantes tienen la suficiente experiencia y no tener problemas de licencia.

- **Interfaz de Conectividad:** Es aquel que permite conectar el motor de inferencia escrito en Prolog con el lenguaje de programación, se seleccionó P Sharp (P#) porque es gratuito, fácil de usar, trae su propio compilador de archivos Prolog y permite transformar el mismo en clases para ser usadas por C#.

- **Web Services:** Fueron desarrollados en Java debido a la ventaja de contar con una herramienta gratuita llamada Talend, el cual permite conectar y procesar los registros en cualquier base de datos de manera rápida y segura encapsulando todo el servicio permitiendo al consumidor abstraerse y conectarse de manera rápida al realizar la implementación. Estos Web Services fueron desarrollados sobre Tomcat el cual es un Servidor de Aplicaciones que cumple con las características necesitadas para el proyecto.

4.2. Implementación

El programa de Calidad de Datos es una aplicación web escrita en ASP.NET y C#, la cual funciona de la siguiente manera:

Al ingresar al sitio aparecerá una pantalla de login, en la cual se pedirán las credenciales del usuario (User y Password).

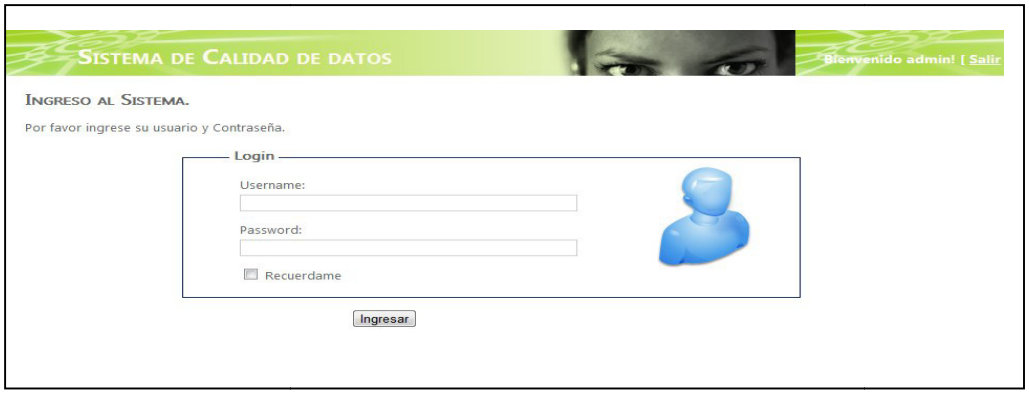


Figura 4.1. Pantalla de login del sistema

Una vez que se ha podido ingresar al sistema se presiona el botón *Cargar Datos* el cual invoca un "Servicio Web" que se encargará de traer el conjunto de datos que contendrá los id de los elementos que serán evaluados.

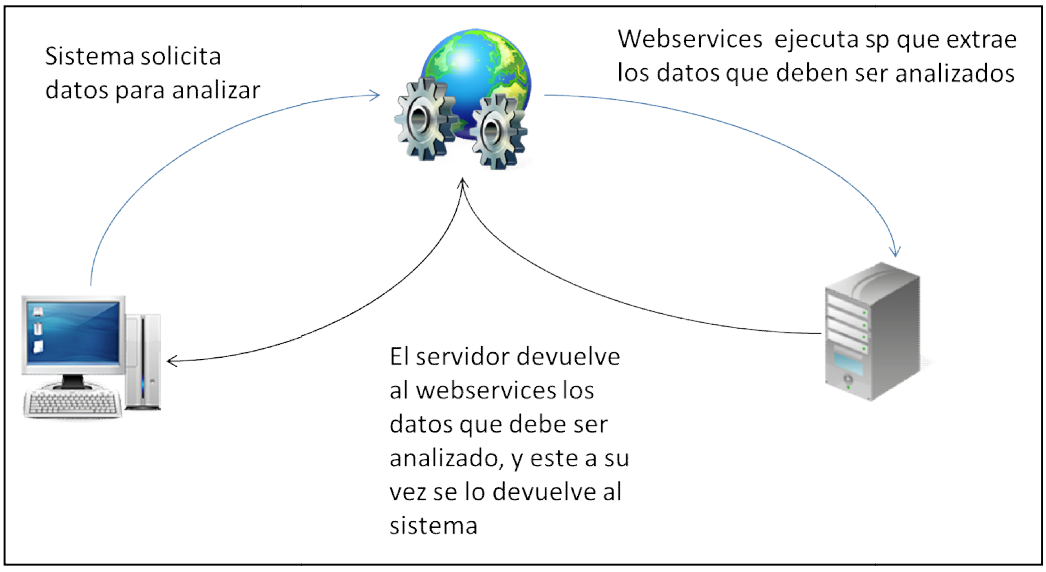


Figura 4.2. Invocación del Web Service al presionar el botón Cargar Datos, el cual devuelve el conjunto de datos con los id que deberán ser evaluados

Cuando se tiene el conjunto de datos con los id de los registros a evaluar, se procede a evaluarlos uno por uno llamando a un segundo Web Service el cual se conectará a la base de datos y realizará el procesamiento del mismo devolviendo un arreglo de datos con valores binarios.

Estos valores son enviados al motor de inferencia el cual se encarga de evaluar en el árbol de decisiones, devolviendo el tipo de error en caso que detecte si el registro es **incorrecto** o devolviendo **ok** si el registro es correcto.

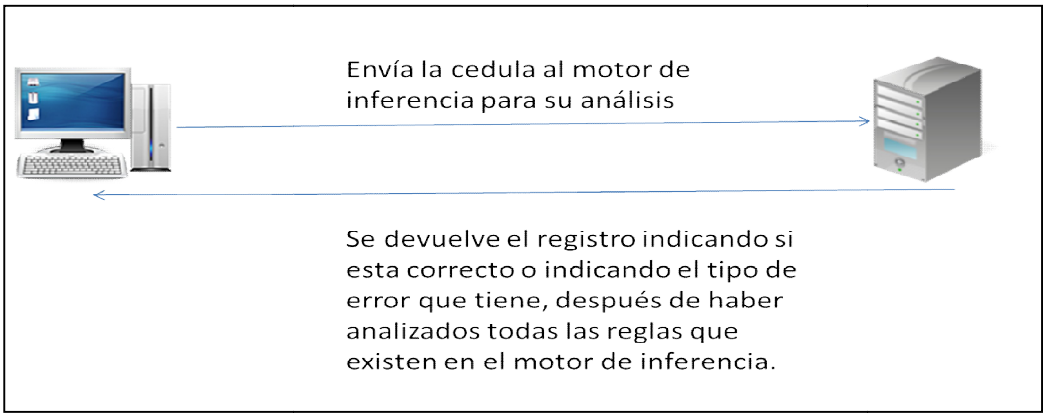


Figura 4.3. Invocación del web Service que recibe el id a evaluar, y devuelve un arreglo binario para su posterior análisis por parte del motor de inferencia que indica si es correcto o erróneo.

Una vez que todos los datos han sido evaluados, se muestran los resultados en una tabla con los resultados de la evaluación.

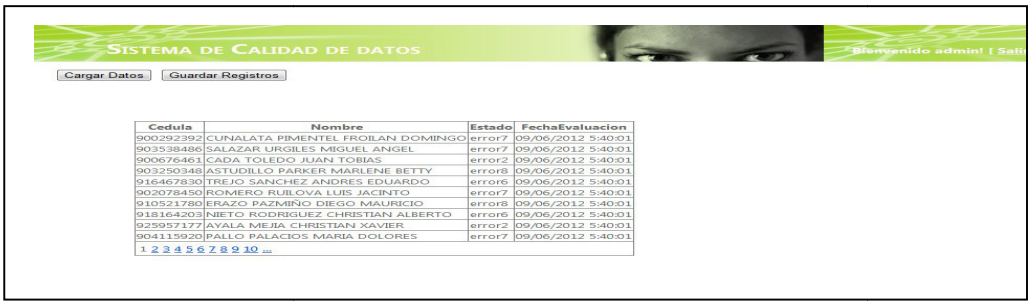


Figura 4.4. Datos evaluados por parte del Sistema de Calidad de Datos

Al visualizar los datos el usuario puede mandar a guardar los datos en la base de datos para posteriormente visualizar un reporte con la información estadística de los errores agrupados por tipo, y los registros que contienen errores.

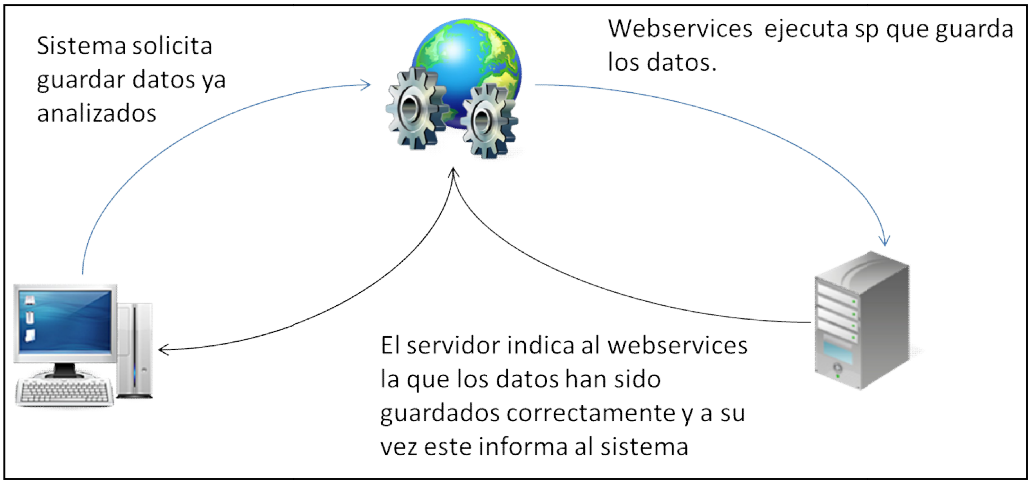


Figura 4.5. Guardar los datos que ya han sido revisados por parte del usuario.

Además el usuario podrá visualizar la cantidad de errores por tipo de error mediante la representación de un gráfico como se muestra a continuación:

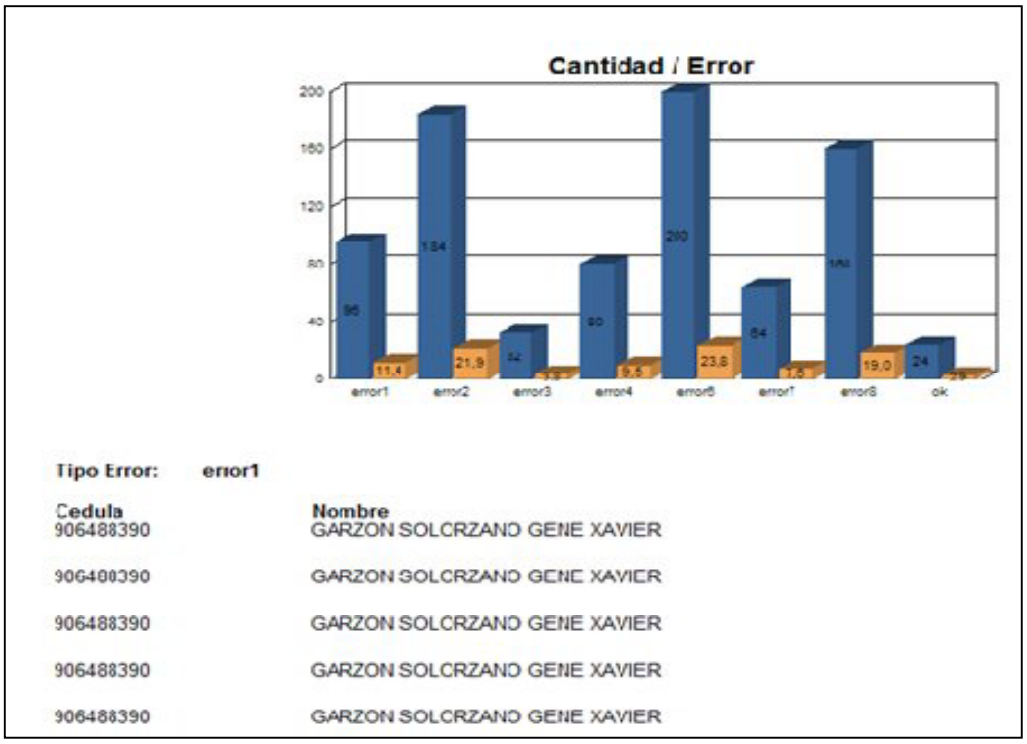


Figura 4.6. Gráfico estadístico con los resultados del análisis de los datos.

4.3. Metodología de Prueba

La compañía brindó una base de datos de pruebas que contiene datos ficticios pero que corresponde a la arquitectura original de su base de datos.

Capitulo 5

El siguiente capítulo describe los diferentes casos de uso los cuales describen los pasos o las actividades que deberán realizarse para llevar a cabo los procesos del Sistema de Calidad de Datos. Además muestra los resultados obtenidos de las pruebas realizadas tomando como datos de pruebas un conjunto de datos brindados por la compañía DataDecision.

5.1. Casos de Uso

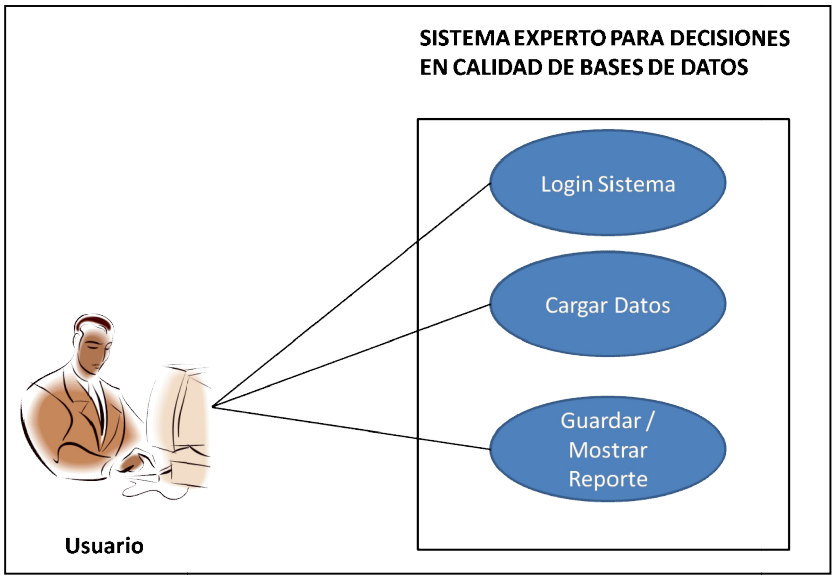


Figura 5.1. Opciones de Usuario en el Sistema de Calidad de Datos

Caso CU-01	
Nombre de caso de uso	**Login al Sistema**
Actores participantes	Usuario
Flujo de eventos	El usuario para poder ingresar al sistema deberá ingresar sus credenciales como lo son su nombre usuario y su contraseña.
Condición de entrada	El usuario ingresa Nombre de Usuario y Contraseña
Condición de salida	Una vez que el nombre de usuario y contraseña se verifican que son correctos, se direcciona al usuario a la pagina principal del sistema.
Requerimientos de calidad	SI el nombre de usuario y contraseña no son válidos se impede el acceso al sistema.

Tabla 5.1. CU-01

Caso CU-02	
Nombre de caso de uso	**Cargar Datos**
Actores participantes	Usuario
Flujo de eventos	El usuario presionara el botón “Cargar Datos” el cual se encargará de traer el set de datos a través de un Web Services, con la información necesaria para ser analizada por el motor de inferencia . Una vez que se cuenta con el set de datos se procede al análisis elemento por elemento por parte el segundo Web Service el cual devuelve al motor de inferencia datos para determinar si el registro es correcto o determinar el tipo de error que posee el mismo.
Condición de entrada	El usuario presiona el botón Cargar Datos.
Condición de salida	El usuario visualiza los datos cargados

	con el resultado de los análisis. La información mostrada es la siguiente: • Cédula • Nombre • Estado (Ok , o en su defecto el número del error) • Fecha de Evaluación

Tabla 5.2. CU-02

Caso CU-03	
Nombre de caso de uso	**Guardar / Mostrar Datos**
Actores participantes	Usuario
Flujo de eventos	El usuario presionara el botón “Guardar Datos” el cual se encargará de guardar el set de datos que fue evaluado por el sistema haciendo uso de un web services. Una vez que se guardaron los datos se muestra un reporte donde se muestra la cantidad de errores y su respectivo porcentaje en un gráfico estadístico por cada tipo de error. Adicionalmente se muestran los registros con errores agrupados por tipos de error.
Condición de entrada	El usuario presiona el botón Guardar Datos.
Condición de salida	Reporte con los registrso agrupados por tipo de error, gráfico estadístico con la

	cantidad de errores y su respectivo porcentaje por cada tipo de error.

Tabla 5.3. CU-03

5.2. Análisis de Resultados

Se realizó una prueba con 105 registros para analizar los resultados obtenidos por el sistema, obteniendo como resultado la siguiente información:

Tipo de Error	Cantidad de Errores
ERROR Tipo 1	14
ERROR Tipo 2	21
ERROR Tipo 3	7
ERROR Tipo 4	2
ERROR Tipo 6	20
ERROR Tipo 7	4
ERROR Tipo 8	34

Registros OK	3

Tabla 5.4. Análisis de Resultados

Los resultados previos obtenidos han sido representados en un gráfico de barras como se muestra a continuación:

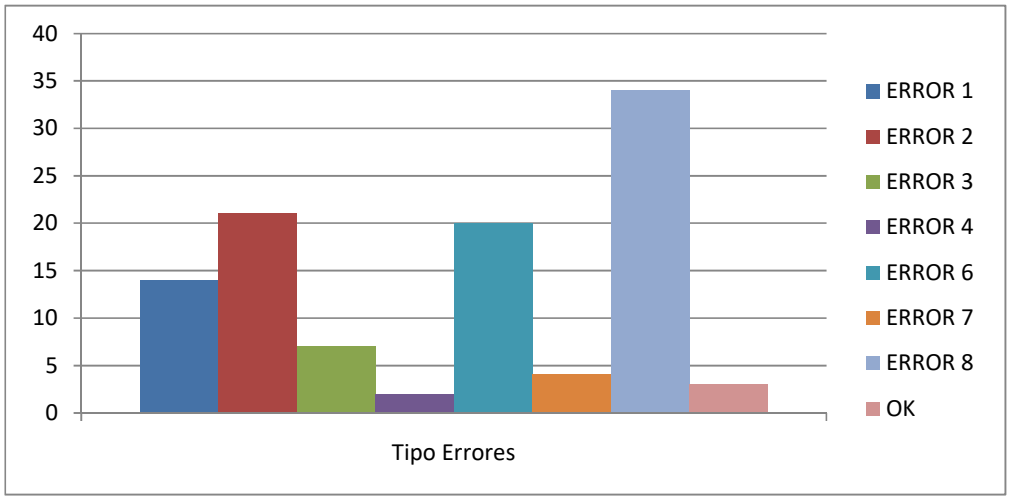


Figura 5.2. Gráfico de Barras de la cantidad de errores por tipo de error obtenidos por el Sistema de Calidad de Datos

El tiempo consumido para este análisis fue de **5 min** teniendo una aproximación de 1050 registros en 50 min.

Estos datos fueron entregados a los expertos para ser revisados y se obtuvieron los siguientes resultados:

Tipo Error	Cantidad	Evaluación de Resultados
ERROR Tipo 1	14	Resultado esperado
ERROR Tipo 2	21	Resultado esperado
ERROR Tipo 3	7	Resultado esperado
ERROR Tipo 4	2	Resultado erróneo
ERROR Tipo 6	20	Resultados esperados parcialmente
ERROR Tipo 7	4	Resultado esperado
ERROR Tipo 8	34	Resultado esperado
Registros OK	3	Resultado esperado

Tabla 5.5. Evaluación de Resultados

5.4. Evaluación de Resultados

En el Error 4 se encontraron que esos registros no tenían errores.

En el Error 6 los 20 registros fueron marcados correctamente pero se llegó a la conclusión de que estaban dentro de un subgrupo que no había sido considerado dentro de los requerimientos iníciales, que corresponden a 10 registros (Si hay valores homónimos pero escritos de orden diferente)

Esto nos da una Efectividad de un **92%**

Conclusiones

1) El tiempo de procesamiento de datos para el análisis de calidad de datos pasa de 4 días promedio a 2 horas mejorando en más del 500% el proceso actual.

2) El porcentaje de aciertos obtenido en la clasificación de registros -registros correctos / errados- fue del 92 %.

3) La saturación creada por la constante validación de datos en el departamento de calidad de Datos creada por este proceso se reduce en un 400%.

4) Nuestro sistema ha permitido que el funcionamiento de los dos sistemas de base datos -Base de datos Histórica vs Base de Datos Actual- ya no sea independiente permitiendo que tengan un mejor rendimiento.

5) La compañía tiene una visión global real de la calidad de información que está ingresando y de la que posee.

Recomendaciones

1) Se recomienda agregar un módulo de aprendizaje para poder ir agregando nuevas reglas tomando en consideración que no afecte a las ya creadas.

2) Se recomienda la creación de un módulo de administración que permita a los expertos configurar reglas y el modo de solucionar los errores.

3) A partir de las estadísticas generadas por la aplicación se recomienda desarrollar indicadores de calidad de la información para poder llevar un control de la calidad de datos y su evolución a través del tiempo.

4) En una futura versión se podrían añadir las 32 reglas faltantes que corresponde al 20% de errores ocurridos.

ANEXO DE CAPTURA DEL CONOCIMIENTO

En este anexo se detalla la manera en la cual como obteniendo las reglas en lenguaje natural, estas son pasadas a reglas escritas en lenguaje matemático para posteriormente, mediante la determinación de las variables extraídas de los nodos del árbol de conocimiento se escriben en sintaxis Prolog para de esta manera llenar la base de conocimiento.

Definición de Reglas en base a la captura del Conocimiento

El conocimiento obtenido por parte de los expertos fue realizando entrevistas y realizando visitas en donde se procedió a observar todo el proceso de trabajo, una vez identificado el proceso general el conocimiento fue captado por medio de un formulario en donde se escribieron los diferentes escenarios que ellos realizan al momento de evaluar los datos. Se identificaron 8 reglas importantes las cuales se detallan en la siguiente tabla:

Antecedente	Consecuente
Si el ciudadano se encuentra registrado en la base de datos nueva y esta no coincide con la cédula que se encuentra registrada en la base	Entonces existe un error de tipo 2

de datos de transición,	
Si el ciudadano tiene su cédula registrada en la base nueva Y la misma coincide en la base transición, pero los nombres que se encuentran en la base de datos nueva no coinciden con los nombres de la base de datos de transición.	Entonces existe error tipo 1
Si la cédula de identidad en la base nueva coincide con la cédula de la base de datos de transición Y existe coincidencia de nombres en la base de datos de transición y en la base nueva, pero no coincide el sexo que está en la base transición con el de la base de datos nueva,	Entonces existe error tipo 4
Si existe la cédula tanto en la base de datos nueva como en la base de datos de transición Y existe coincidencia de nombres tanto en base de datos de transición como en	Entonces el registro es correcto

la base de datos nueva, y si además también coincide el sexo tanto en la base de datos de transición como en base de datos nueva Y finalmente la fecha nacimiento en la base de datos de transición coincide con la de la base de datos nueva.	
Si existe la cédula tanto en la base de datos nueva como en la base de datos de transición. Y coinciden los nombres registrados tanto en base de datos de transición como en base de datos nueva y si el sexo coincide en la base de datos de transición así como en la base de datos nueva. Pero no coincide la fecha nacimiento en la base de datos den datos de transición con la que se encuentre registrada en la base de datos nueva,	Entonces existe error tipo 3
Si no se encuentra el registro en la base de datos nueva	Entonces existe error tipo 8

correspondiente a la cédula del ciudadano Y a su vez no existe el nombre de dicha persona en la base de datos transición,	
Si no se encuentra la cédula en la base de datos nueva, ni en la base de datos de datos de transición, pero el nombre de la persona existe en la base de datos de transición.	Entonces existe error tipo 6
Si no existe el registro de la cédula tanto en la base de datos nueva como en la base de datos de transición ,sin embargo existe el nombre en la base de datos de transición Y no tiene homónimo en la base de datos de transición	Entonces existe error tipo 6
Si no tiene cédula en la base de datos nueva ni en la base de datos de transición, pero existe el nombre en la base de datos de transición Y tiene homónimo en la base de datos	Entonces existe error tipo 7

de transición,	

Definición de Reglas

Implementación de las Reglas

Las reglas que se han definido, se procede a escribirlas a un interpretador de reglas, en este caso PROLOG, valiéndonos de variables para representar cada *antecedente,* el cual se encuentra en un nodo diferente del árbol de conocimiento, este *antecedente* con su respectivo *consecuente* solo puede tomar valores binarios:

1, en caso de que el antecedente sea válido para satisfacer su consecuente.

0, en caso de que el antecedente sea inválido para satisfacer su consecuente.

Tomando como base el árbol de conocimiento, las reglas y variables fueron agruparon por ramal.

Las variables que se definieron para el ramal derecho del árbol partiendo como valido el antecedente del nodo raíz (nivel 1) son las siguientes:

Variable	Significado	Dominio
X	Tiene cédula en base nueva	X ε [0,1]
Y	Coincide cédula base transición y base nueva	Y ε [0,1]
Z	Coincide nombres base transición y base nueva	Z ε [0,1]
W	Coincide sexo base transición y base nueva	W ε [0,1]
V	Coincide fecha nacimiento base transición y base nueva	V ε [0,1]

Haciendo uso de estas variables se definen las siguientes reglas de inferencia escritas en lenguaje matemático:

R1	Si X=1 Λ Y=0 -> error 2

R2	Si X=1 Λ Y=1Λ Z=0 -> error 1
R3	Si X=1 Λ Y=1 Λ Z=1 Λ W=0 -> error 4
R4	Si X=1 Λ Y=1 Λ Z=1 Λ W=1 Λ V=1 -> ok
R5	Si X=1 Λ Y=1 Λ Z=1 Λ W=1 Λ V=0 -> error 3

Así podemos interpretar la regla uno: R1 de la siguiente manera.

SI Tiene cédula en base nueva = **VERDADERO Y** Coincide cédula base transición = **VERDADERO Y** base nueva = **VERDADERO ENTONCES** existe un error de tipo 2

Las variables que se definieron para el ramal izquierdo del árbol partiendo como NO válido el antecedente del nodo raíz (nivel 1) son las siguientes:

Variable	Significado	Dominio
U	Nombre existe base transición	U ε [0,1]
T	Tiene cédula base	T ε [0,1]

	transición	
S	Tiene homónimo base transición	S ε [0,1]

Haciendo uso de estas variables se definen las siguientes reglas de inferencia escritas en lenguaje matemático:

R6	Si X=0Λ U=0 -> error 8
R7	Si X=0 Λ U=1 Λ T=0 -> error 6
R8	Si X=0 Λ U=1 Λ T=1 Λ S=0 -> error 6
R9	Si X=0 Λ U=1 Λ T=1 Λ S=1 -> error 7

Base de Hechos

Basados en las reglas de inferencia previamente definidas procedemos a definir nuestra base de hechos en lenguaje Prolog. Para mayor información de cómo definir los hechos en Prolog consultar el *ANEXO DE PROGRAMACIÓN EN LENGUAJE PROLOG Y USO DE P#, Página 3*.

fecha_coincide(1,ok)
fecha_coincide (0,error3)
sexo_coincide(0,error4)
nombre_coincide (0,error1)
cédula_coincide(0,error2)
nombre_base_vieja (0, error8)
cédula_base_vieja(0,error6)
homonimo_base_vieja(0,error6)
homonimo_base_vieja(1,error7)
tiene_cédula_new(0,error1)

Base de Conocimientos

En la base de conocimientos se procede a ingresar las reglas sobre las cuales los registros serán evaluados para proceder con la validación exitosa o no exitosa del mismo. Para mayor información de cómo definir las reglas en Prolog consultar el *ANEXO DE PROGRAMACIÓN EN LENGUAJE PROLOG Y USO DE P#, Página 4*.

evalua_registro(X,Y,Z,W,V,U,T,S,Respuesta):- X=1,cédula_coincide_bases(Y,Z,W,V,Respuesta).
evalua_registro(X,Y,Z,W,V,U,T,S,Respuesta):- not(X=1),nombre_existe_base_tran(U,T,S,Respuesta).

Reglas para el ramal izquierdo en caso de que las evaluaciones del nodo raíz sean verdaderas:

cédula_coincide_bases(Y,Z,W,V,Respuesta):- Y=1,nombre_coincide_bases(Z,W,V,Respuesta).
cédula_coincide_bases(Y,Z,W,V,Respuesta):- not(Y=1),cédula_coincide(Y,Respuesta).
nombre_coincide_bases(Z,W,V,Respuesta):- Z=1,sexo_coincide_bases(W,V,Respuesta).
nombre_coincide_bases(Z,W,V,Respuesta):- not(Z=1),nombre_coincide(Z,Respuesta).
sexo_coincide_bases(W,V,Respuesta):-W=1,fecha_coincide(V,Respuesta).

```
sexo_coincide_bases(W,V,Respuesta):-not(W=1),sexo_coincide(W,Respuesta).
```

Reglas para el ramal derecho en caso de que las evaluaciones del nodo raíz sean falsas:

```
nombre_existe_base_tran(U,T,S,Respuesta):-
U=1,tiene_cédula_base_tran(T,S,Respuesta).
```

```
    nombre_existe_base_tran(U,T,S,Respuesta):-
    not(U=1),nombre_base_vieja(U,Respuesta).
```

```
    tiene_cédula_base_tran(T,S,Respuesta):-
    T=1,homonimo_base_vieja(S,Respuesta).
```

```
    tiene_cédula_base_tran(T,S,Respuesta):-
    not(T=1),cédula_base_vieja(T,Respuesta).
```

- Un caso particular es la variable anónima, representada por el carácter subrayado ("_"). Es una especie de comodín que utilizaremos en aquellos lugares que debería aparecer una variable, pero no nos interesa darle un nombre concreto ya que no vamos a utilizarla posteriormente.

1.1. Reglas

Las reglas se utilizan en PROLOG para significar que un hecho depende de uno o más hechos. Son la representación de las implicaciones lógicas del tipo p→q (p implica q).

- Una regla consiste en una cabeza y un cuerpo, unidos por el signo ":-".
- La cabeza está formada por un único hecho.
- El cuerpo puede ser uno o más hechos (conjunción de hechos), separados por una coma (","), que actúa como el "y" lógico.
- Las reglas finalizan con un punto (".").

La cabeza en una regla PROLOG corresponde al consecuente de una implicación lógica, y el cuerpo al antecedente. Este hecho puede conducir a errores de representación. Supongamos el siguiente razonamiento lógico:

Tiempo (lluvioso) → Suelo (mojado)

Suelo (mojado)

Que el suelo está, mojado, es una condición suficiente de que el tiempo sea lluvioso, pero no necesaria. Por lo tanto, a partir de ese hecho, no podemos deducir mediante la implicación, que está, lloviendo (pueden haber regado las calles). La representación **correcta** en PROLOG, sería:

```
Suelo (mojado):- Tiempo (lluvioso).

Suelo (mojado).
```

El predicado **consult** está pensado para leer y compilar un programa PROLOG.

```
consult (fichero).
```

1.2. La resolución de objetivos

Una consulta tiene la misma forma que un hecho.

Consideremos la pregunta:

```
?-quiere_a(susana,pablo).
```

PROLOG busca por toda la base de conocimiento hechos que ***coincidan*** con el anterior. Dos hechos coinciden si sus predicados son iguales, y cada uno de sus correspondientes argumentos lo son entre sí. Si PROLOG encuentra un hecho que coincida con la pregunta, responderá yes. En caso contrario responderá no.

Además, una pregunta puede contener variables. En este caso PROLOG buscara por toda la base de hechos aquellos objetos que pueden ser representados por la variable. Por ejemplo:

```
?-quiere_a(maria, Alguien).
```

Alguien es un nombre perfectamente válido de variable, puesto que empieza por una letra mayúscula.

El resultado de la consulta es:

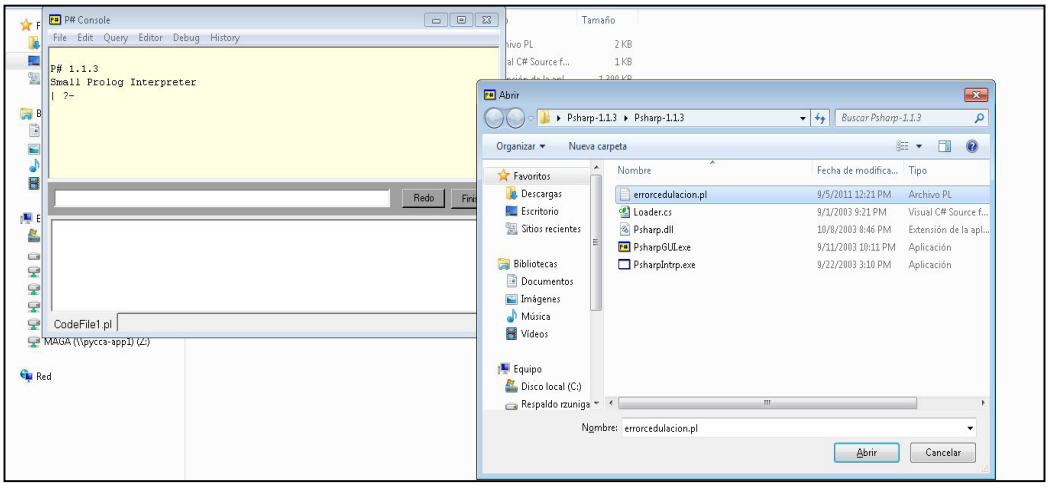
Alguien = enrique

2. P Sharp

P# es un programa que puede compilar programas escritos en lenguaje Prolog respondiendo a cada una de las instrucciones y sintaxis. Además puede transformar un programa Prolog en clases de C# para ser utilizadas por cualquier programa de .NET.

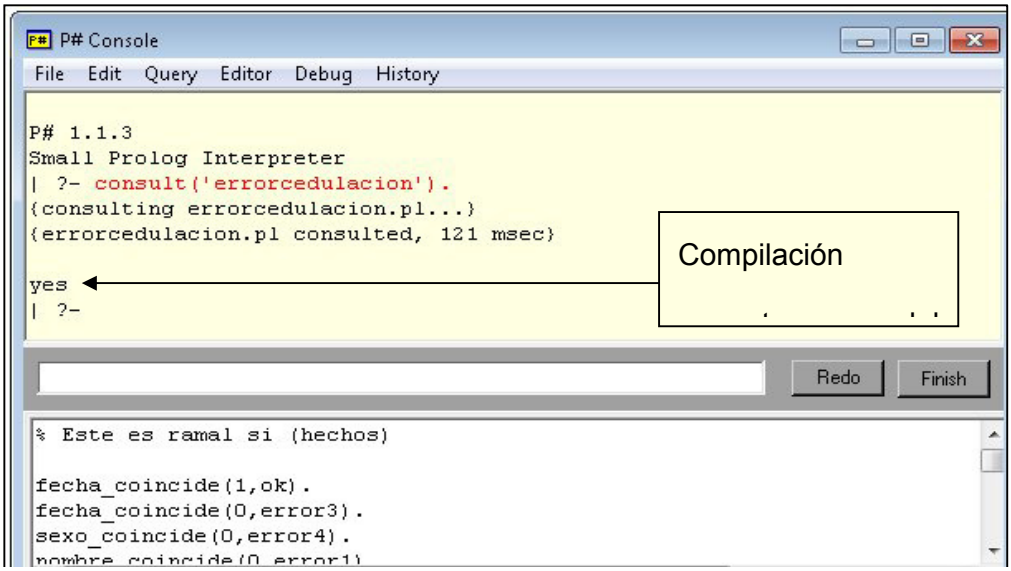
De esta manera se obtiene una interfaz capaz de comunicar un aplicación escrita en C# con la respectiva invocación a nuestro motor de inferencia escrito en Prolog.

2.1. Uso de P#



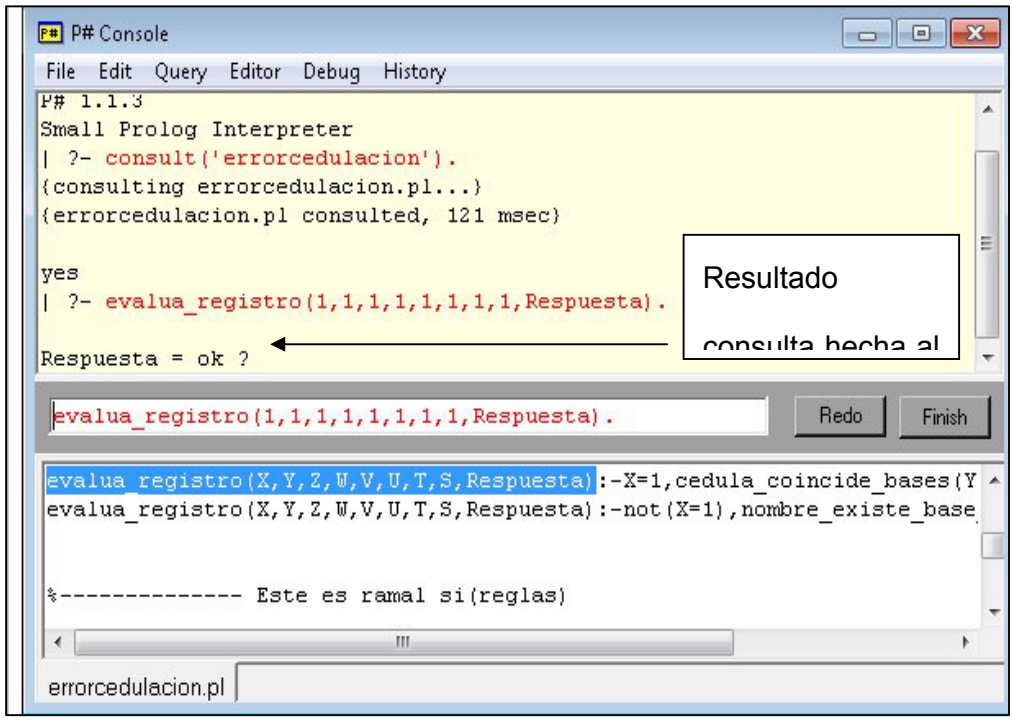
Para abrir un archivo seleccionamos la opción File -> Open

Una vez abierto el documento lo compilamos haciendo uso de la instrucción



consult.

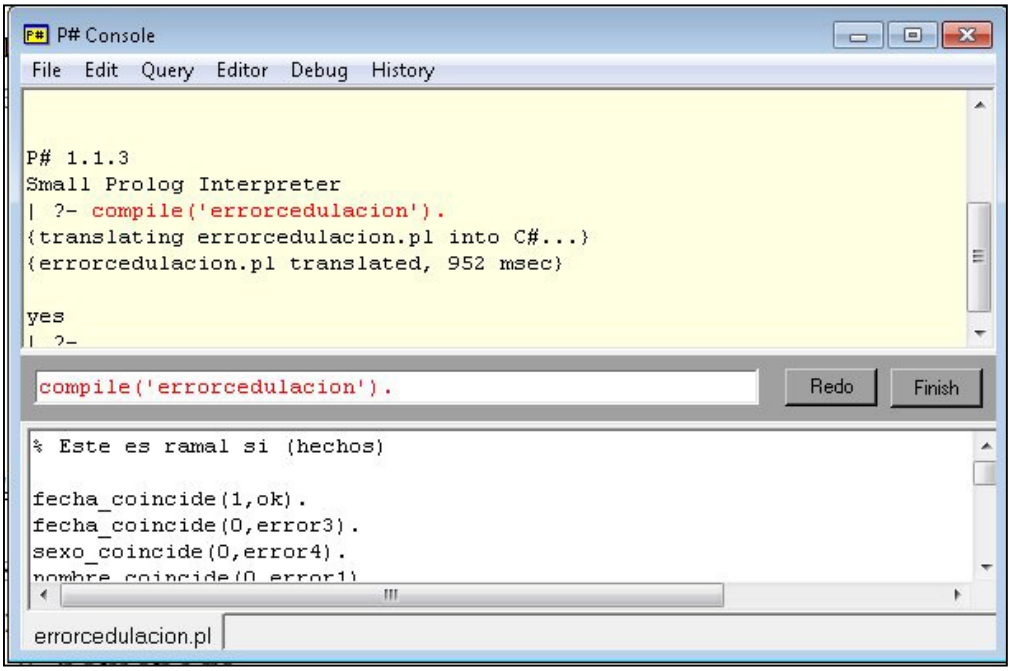
Una vez compilado el archivo podemos empezar a realizar nuestras



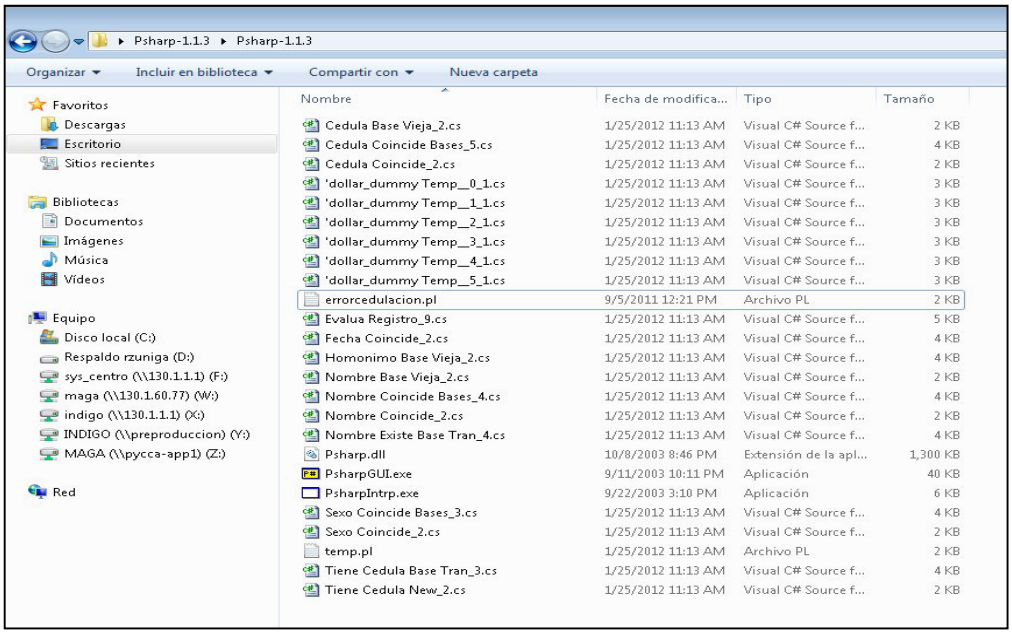
consultas para evaluar nuestra base de conocimiento.

Una vez que hemos compilado correctamente nuestro archivo y que hemos verificado que podemos realizar consultas a nuestra base de conocimiento procederemos a usar el comando compile el cual se encarga de transformar el archivo Prolog en Clases C# para poder ser utilizadas en nuestro proyecto.

compile (fichero).



Resultado de una consulta hecha al motor de inferencia.



2.2. Invocación de P# desde C#

Para realizar la interacción con Prolog y P# desde C# disponemos de una dll que es la ***Psharp.dll*** la cual la agregamos como referencia a cualquier proyecto .NET.

En todos los archivos C# donde queramos hacer uso de P# incluiremos las siguientes líneas al principio:

```
usingJJC.Psharp.Lang;
using JJC.Psharp.Predicates;
using JJC.Psharp.Lang.Resource;
using JJC.Psharp.Resources;
```

Para llamar Prolog desde C# necesitamos hacer uso de la clase PrologInterface

```
PrologInterface sharp = newPrologInterface();
```

La interfaz PrologInterface posee el método SetPredicateel cual invoca cualquier regla que se encuentran en el motor de conocimientos.

Cabe mencionar que el Prolog genera clases con la siguiente sintaxis.

NombreRegla_numero_de_atributos.

El constructor de cada clase creada por P# recibe el número de argumentos con el tipo de datos que se definió en las reglas de Prolog.

```
sharp.SetPredicate(newEvaluaRegistro_9(newIntegerTerm(x),newIntegerTerm(y),
newIntegerTerm(z),newIntegerTerm(w),newIntegerTerm(v),newIntegerTerm(u),newIntegerTerm(t),newIntegerTerm(s),rpta, newReturnCs(sharp)));
```

En P# hay clases que son las que permiten crear los tipos de datos que serán enviados como parámetros a los constructores de las diferentes clases que representas reglas del motor de inferencia.

Para valores enteros.

```
IntegerTerm i = new IntegerTerm( 3 );
```

Para usar valores decimales

```
DoubleTerm d = new DoubleTerm( 2.5 );
```

Para usarvalores string

```
SymbolTerm s = SymbolTerm.MakeSymbol( "a_string" );
```

Las siguientes líneas de código fueron extraídas del manual de P#, para obtener la respuesta del motor de inferencia Prolog.

```
// call predicate to obtain all solutions
for (bool r = sharp.Call(); r; r = sharp.Redo())
{
```

```
        // respuesta de la evaluación de Prolog

        respuesta = rpta.Dereference().ToString();

}
```

REFERENCIAS BIBLIOGRÁFICAS

[1] Wikipedia," Principio de Pareto ",

http://es.wikipedia.org/wiki/Principio_de_Pareto

Fecha de consulta: 1 de junio 2012.

[2]Sistema Experto,

http://html.rincondelvago.com/sistema-experto.html

Fecha de consulta: 1 de junio 2012.

[3]Wikipedia, Árbol de Decisiones

http://es.wikipedia.org/wiki/%C3%81rbol_de_decisi%C3%B3n

Fecha de consulta: 10 de mayo 2012.

[4]Universidad UDELAR, ARQUITECTURA DE SERVICIOS SEMÁNTICOS

http://www.fing.edu.uy/~pgsemws/recursos/documentos/InformeFinal.pdf

Fecha de consulta: 10 de mayo 2012.

[5]Irene Marquet Gomez- Laura Luque Martinez Sistema experto de Agencia de Viajes Universidad Carlos III Madrid

http://es.scribd.com/doc/49868168/17mem

Fecha de consulta: 10 de mayo 2012.

[6]Utm, Árbol de Decisión

http://www.utm.mx/~jahdezp/archivos%20estructuras/DESICION.pdf

Fecha de consulta: 10 de mayo 2012.

[7]Ucla, Procesamiento de Datos

http://www.ucla.edu.ve/dac/Departamentos/coordinaciones/informaticai/documentos/PROCESAMIENTO%20DE%20DATOS.htm

Fecha de consulta: 10 de mayo 2012.

[8]Talend,creación de web service y Procesamiento de Datos

www.talend.com

Fecha de consulta: 10 de mayo 2012.

[9] Dámaso Velázquez," Prolog para .NET ",

http://www.webprogramacion.com/90/csharp/prolog-para-net.aspx,

Fecha de consulta: 23 de Agosto 2011.

[10] “Download P#”,

http://www.dcs.ed.ac.uk/home/jjc/psharp/psharp-1.1.3/dlpsharp.html

Fecha consulta 20 de Agosto 2011.

[11] Dan Buskirk, “Prologfor .NET Developers”,

http://www.codeproject.com/KB/system/PrologNET.aspx,

Fecha consulta: 1 de Septiembre 2011.

[12] James Lu, Jerud J. Mead , “Tutorial PROLOG”,

http://classes.soe.ucsc.edu/cmps112/Spring03/languages/prolog/PrologIntro.pdf

Fecha consulta: 15 Julio 2011

[13] Mike Brayshaw, “IntroductiontoProlog”,

http://www.doc.gold.ac.uk/~mas02gw/prolog_tutorial/prologpages/ ,

Fecha consulta: 21 Julio 2011

[14] Wikipedia,” Principio de Pareto “,

http://es.wikipedia.org/wiki/Principio_de_Pareto

Fecha de consulta: 1 de junio 2012