

Instituto Tecnológico de Costa Rica
Administración de Tecnologías de
Información
Lenguajes de Programación
Prof. Andréi Fuentes L.

Tarea Programada 2

Erick Vargas Victor
Luis Diego Coto Mata
Mariam Ramirez Cordero

2013

Contenido

Descripción del problema 2

Diseño del programa 3

Librerías usadas 5

Análisis de Resultados 6

Manual de usuario 7

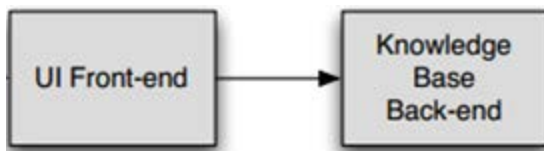
Conclusión personal 13

Bibliografía 14

Descripción del problema

Esta tarea programada tiene como objetivo hacer un software que permita la consulta de información de regalos que se les darán a los niños, y además consultas sobre las características de estos niños y sus acciones, para poder basarse en esta información para repartir regalos. El sistema debe tener modo de mantenimiento (se ingresan datos de niños: nombre, edad, país, buenas acciones, malas acciones, wishlist y presupuesto para cada niño) y modo de consulta. Además se deben tener los datos para cada tipo de juguete (nombre, marca, precio, edad necesaria para usarla). En cuanto al modo de consulta, se podrán hacer varias consultas. Las consultas que se deben poder realizar son las siguientes:

- Lista de niños malos y niños buenos
- Lista de niños que tienen una acción buena en particular
- Lista de niños que tienen una acción mala en particular
- Lista de regalos más solicitados por los niños
- Consulta niño-regalos
- Consulta niño-regalos posibles:
- Presupuesto total



El sistema deberá estar estructurado dos componentes: Front-end y Back-end. El Front-end es el encargado de la interacción con el usuario, y se comunicara con el Back-end, que es el encargado de manipular la base de conocimientos y responder a consultar.

La base de conocimientos debe consistir en una serie de declaraciones en Prolog que definen las características de los niños y de los juguetes. El Back-end deberá actualizar la base de conocimientos usando los datos que el usuario ingrese al sistema por medio del Front-end.

El Front-end se va a desarrollar en Java, y el Back-end deberá estar escrito en el lenguaje de programación Prolog, por lo que se debe investigar la manera de realizar la conexión mediante estos dos lenguajes, utilizando alguna librería externa.

Diseño del programa

- **Conexión:**

1. Importar librerías (mencionadas en el siguiente punto)
2. Lo siguiente es inicializar el intérprete para lo que deberemos escribir el siguiente código:

```
/**
 * Entorno que usara el intérprete.
 */
Environment env;
/**
 * Interprete de PROLOG.
 */
Interpreter interpreter;

env = new Environment();
// Cargamos el fichero en el entorno.
env.ensureLoaded (AtomTerm.get("C:\\tudir\\tuprolog.pl"));
// Obtenemos el interprete
interpreter = env.createInterpreter ();
// Y lo inicializamos
env.runInitialization (interpreter);
```

3. Para realizar consultas a la base de conocimiento:

```
// Argumentos para la consulta en este caso constantes.
Term[] args = {AtomTerm.get ("paco"), AtomTerm.get("tito") };
// En este ejemplo se construye la consulta Paco es descendiente de
Tito?
CompoundTerm goalTerm =
    new CompoundTerm (AtomTerm.get("es_descendiente"), args);

try {
    if (interpreter.runOnce(goalTerm) ==
        PrologCode.SUCCESS)
    {
        System.out.println("Pepe es descendiente de Tito.");
    }
    else {
        System.out.println("Pepe NO es descendiente de Tito.");
    }
} catch (PrologException e) {
    System.out.println("Error." + e.getMessage());
}

// Fin.

env.closeStreams();
```

- **Sistema**

Decidimos crear varios objetos (niño, regalo, consulta) para poder manejar de una manera fácil cada aspecto. Cada uno de estos objetos con varios atributos dependiendo de lo que se requiera. Por ejemplo el objeto Nino tiene los atributos: nombre, edad, país. En cuanto al mantenimiento de datos, para ingresar la información de los niños decidimos crear una interfaz que le permitiera al usuario ingresar los datos. Una vez que los datos son agregados, se agregan a la base de conocimientos con cierto formato de orden para que a la hora de acceder esta información, sea más fácil saber en qué posición de cada clausula está cierta característica. Lo mismo sucede con los datos de los regalos. Para hacer las consultas, dependiendo de la que sea, usamos restricciones para lograr filtrar la información que no es necesaria e imprimir en pantalla la información correcta.

Para la consulta de niños malos y niños buenos, se hizo una comparación entre acciones malas y buenas, y si estas malas eran mayor, entonces el niño es malo, de lo contrario se cataloga como niño bueno. Para la consulta de acción buena en particular, una vez que el usuario la ingresa buscamos esa entre la lista de acciones buena de cada niño en la base de conocimientos, y si lo encuentra, se muestra en la lista final. Lo mismo para la lista de niños que tienen una acción mala en particular. El top list de regalos lo manejamos con un contador, cada vez que algún niño tuviera presente cierto regalo en su lista de ‘regalos deseados’, se le suma uno a este contador para cada regalo. De esta manera, logramos sacar los 5 más deseados por los niños. Para la consulta niño-regalos, una vez que el usuario ingresa el nombre de un niño, buscamos en la base de conocimientos los regalos que tuviera este niño para a la vez buscar las características de estos regalos y desplegarlos como resultado de la búsqueda. Lo mismo para la consulta niño-regalos posibles. Para el presupuesto total, se utilizan las mismas restricciones de la consulta anterior, para filtrar los regalos que no es posible entregar, y añadir al presupuesto únicamente el precio de los regalos válidos.

Librerías usadas

Para la conexión:

```
import gnu.prolog.term.AtomTerm;

import gnu.prolog.term.CompoundTerm;

import gnu.prolog.term.Term;

import gnu.prolog.term.VariableTerm;

import gnu.prolog.vm.Environment;

import gnu.prolog.vm.Interpreter;

import gnu.prolog.vm.PrologCode;

import gnu.prolog.vm.PrologException;
```

Para interfaz:

```
import java.awt.Color;

import java.awt.Font;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.ImageIcon;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JMenu;

import javax.swing.JMenuBar;

import javax.swing.JMenuItem;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

import java.awt.TextArea;
```

Análisis de Resultados

- ✓ En cuanto a la conexión de los dos lenguajes, logramos unir Java y Prolog mediante el uso de varias librerías.
- ✓ En cuanto a la interfaz, logramos desarrollar todas las ventanas que se ocupaban y además logramos relacionarlas con el código para el correcto funcionamiento de todas las funciones. Además colocamos elementos como botones y espacios de texto, que logramos concatenar con todo el sistema para mostrar resultados esperados. Utilizamos Window Builder para Eclipse Indigo para crearla.
- ✓ El modo mantenimiento funcionó satisfactoriamente, agregando datos de los niños y los regalos a la base de conocimientos permitiendo su posterior acceso para realizar las consultas.
- ✓ En cuanto a las consultas:
 - Lista de niños malos y niños buenos: se logró hacer la comparación entre acciones buenas y malas para extraer la lista de niños que cumplían con esta característica.
 - Lista de niños que tienen una acción buena en particular: se logró buscar en la base de conocimiento una acción buena en particular y filtrar a los niños que no la tuvieran.
 - Lista de niños que tienen una acción mala en particular: se logró buscar en la base de conocimiento una acción mala en particular y filtrar a los niños que no la tuvieran.
 - Lista de regalos más solicitados por los niños: se pudo hacer la wishlist, escogiendo a los 5 regalos más pedidos por los niños.
 - Consulta niño-regalos: se logró desplegar la información de los regalos dependiendo el niño ingresado por el usuario.
 - Consulta niño-regalos posible: se lograron filtrar los regalos que por ciertas restricciones, el niño ingresado por el usuario no puede recibir.
 - Presupuesto total: se logró mostrar el monto total que Santa debe gastar para comprar todos los regalos.

Manual de usuario

Para la conexión, se requiere:

- Instalar Gprolog. En la terminal digitar:

```
sudo apt-get install gprolog
```

Luego de realizada esta acción, diríjase a la siguiente página y descargue los archivos de la librería: <http://www.gnu.org/software/gnuprologjava/>

Descomprima la carpeta una vez que la haya descargado. Luego abra Eclipse, cree un nuevo proyecto. Una vez creado dele click derecho → propiedades → Java Build Path → agregar jarras externas. Navegue hasta encontrar la carpeta que descargó recientemente y seleccione el archivo gnu.prolog.java.SUVERSION.jar.

- Descargar Prolog para java (<http://ftp.gnu.org/gnu/gnuprologjava/>)
- Descargar GNU (<http://www.gnu.org/distros/free-distros.html>)

Sistema

Al iniciar el sistema, se abre la siguiente pantalla:



Esta interfaz permite seleccionar el modo de mantenimiento o el modo consulta. Si se selecciona mantenimiento de datos en cuanto a los niños, se abre la siguiente ventana:



A screenshot of a web application window titled "Polo Norte INC". The window has a dark header bar with a blue globe icon and the word "Ayuda" in white. The main content area features a background image of a polar bear and two penguins wearing Santa hats. The form contains the following fields: "Nombre" (Name), "Edad" (Age), "País" (Country), "Presupuesto para el niño@" (Budget for the child), "Lista de Buenas Acciones" (List of Good Actions), "Lista de Malas Acciones" (List of Bad Actions), and "Lista de Regalos Deseados" (List of Desired Gifts). Each field is represented by a white text input box. At the bottom right of the form is a blue button labeled "Aceptar" (Accept).

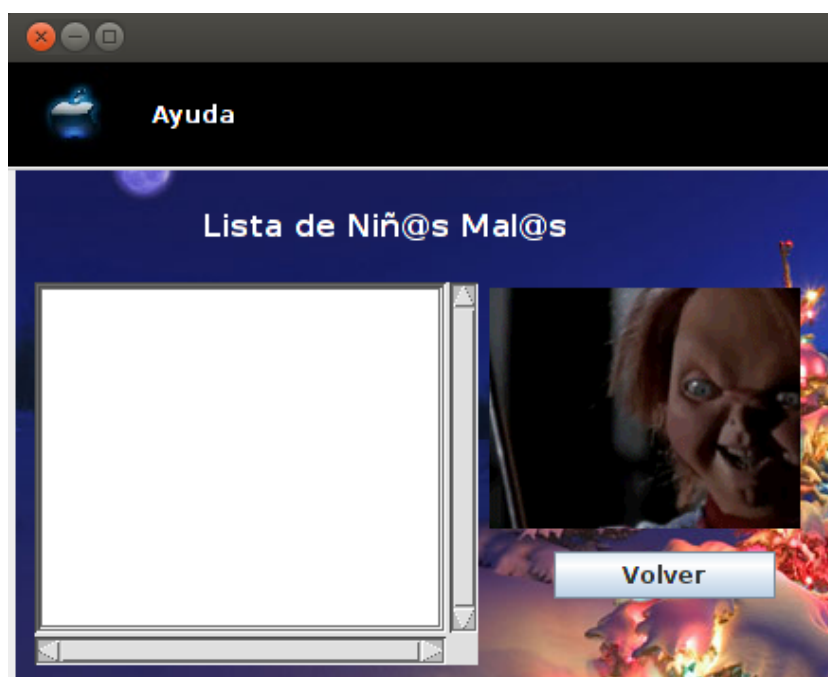
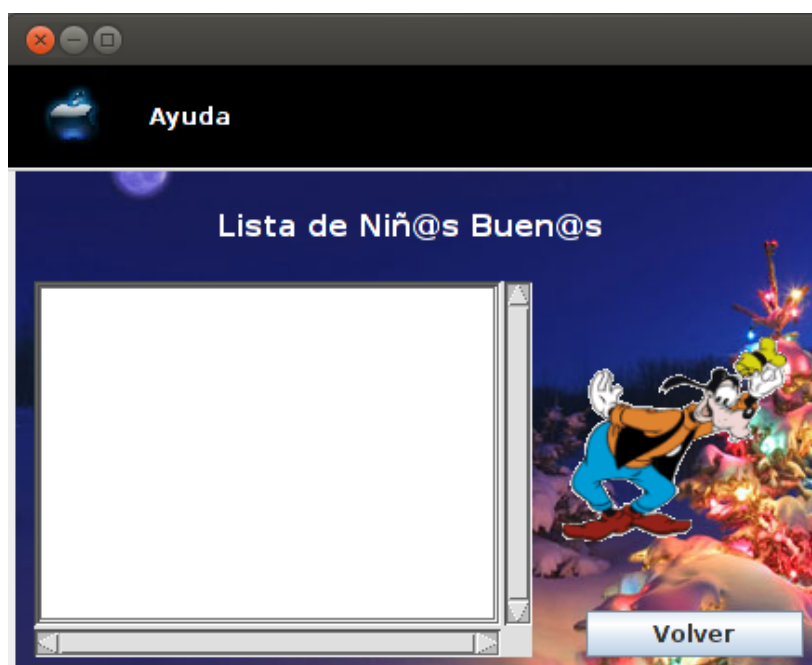
Esta ventana permite al usuario ingresar los datos de cada niño. Una vez que se hace click en 'Aceptar', los datos son agregados a la base de conocimientos. Lo mismo sucede con los regalos:



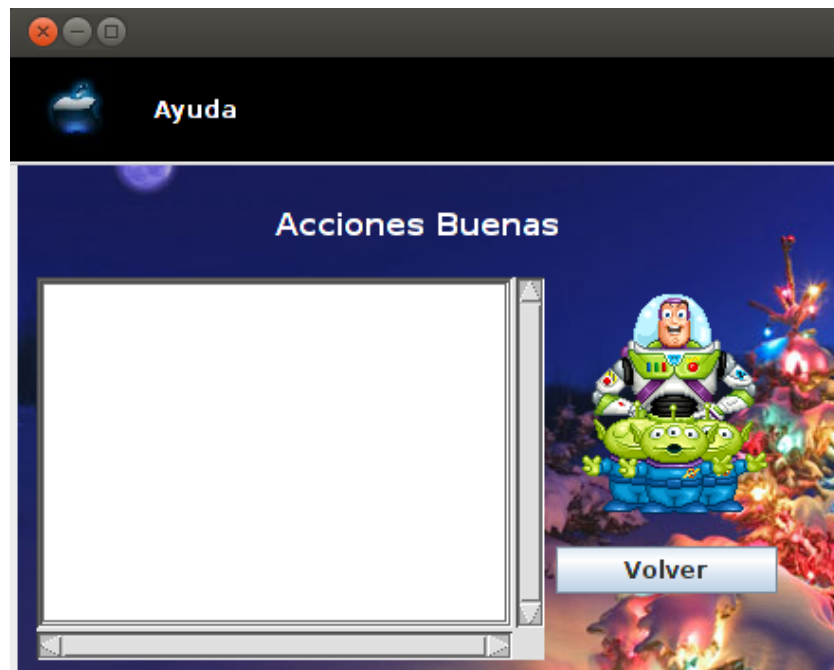
A screenshot of a web application window titled "Polo Norte INC". The window has a dark header bar with a blue globe icon and the word "Ayuda" in white. The main content area features a background image of a polar bear and two penguins wearing Santa hats. The form contains the following fields: "Nombre" (Name), "Precio" (Price), "Marca" (Brand), and "Edad Necesaria" (Necessary Age). Each field is represented by a white text input box. At the bottom center of the form is a blue button labeled "Aceptar" (Accept).

Si se entra en modo consulta, se desplegara un menú con las opciones de consulta, y según la seleccionada, se abrirá una ventana diferente para cada consulta:

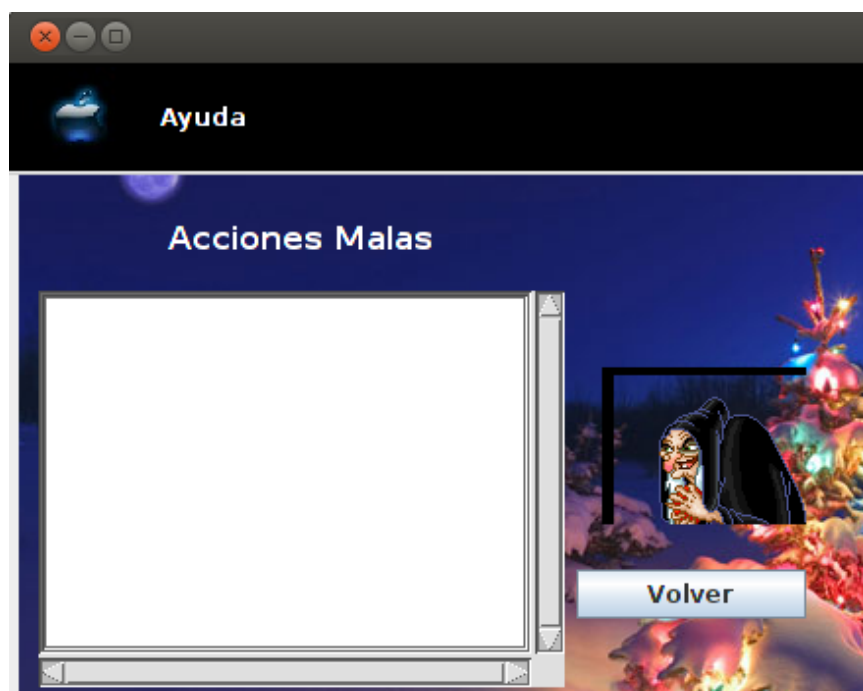
- Lista de niños malos y niños buenos: en el espacio en blanco se despliega la lista de niños, dependiendo de lo indicado. Los niños buenos son los que tienen más acciones buenas que malas, y viceversa.



- Lista de niños que tienen una acción buena en particular: el usuario indica una acción buena específica. Los niños que cumplen con ese requisito se despliegan en el espacio blanco.



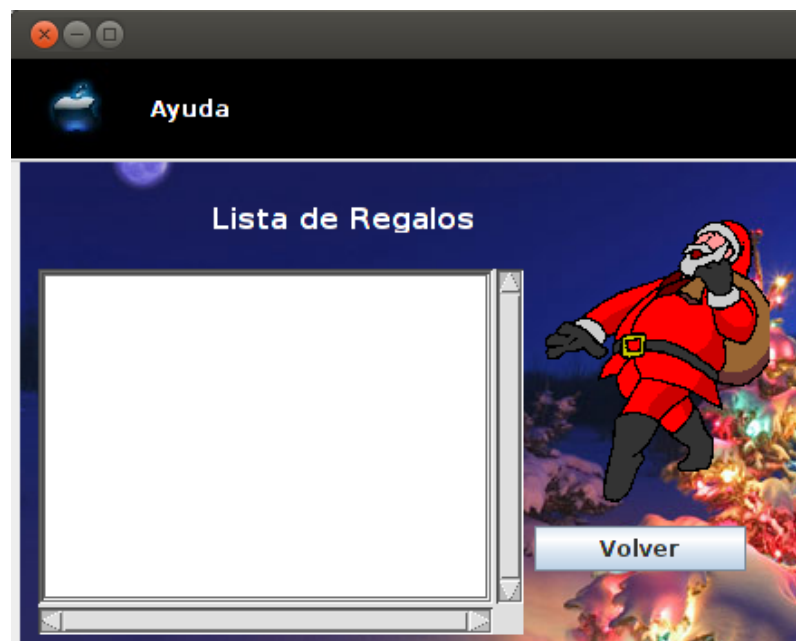
- Lista de niños que tienen una acción mala en particular: el usuario indica una acción mala específica. Los niños que cumplen con ese requisito se despliegan en el espacio blanco.



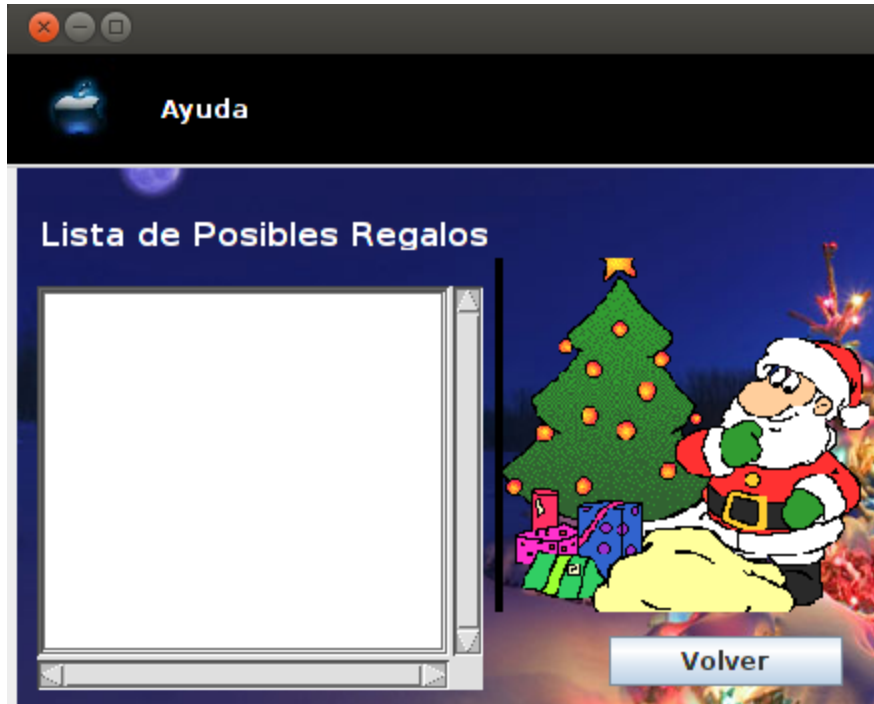
- Wishlist: muestra los 5 regalos más pedidos por los niños. Resultado se muestra en el espacio en blanco



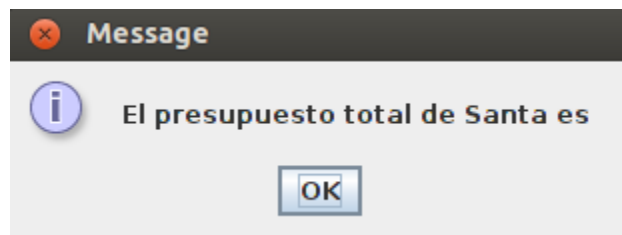
- Consulta niño-regalos: el usuario digita el nombre de un niño, y se despliega la información de regalos que quiere ese niño, con sus respectivos datos. Si el niño tiene más acciones malas que buenas, se indica que el niño solo puede recibir medias de golfista.



- Consulta niño-regalos posibles: en este caso se despliegan únicamente los regalos que sean válidos para el niño, dependiendo de las restricciones.



- Presupuesto total: en el caso de consultar el presupuesto total, se muestra una ventana de mensaje con el dato pedido.



Todas las ventanas cuentan con el botón 'Volver' para regresar al menú principal si se desea, y botón de 'Ayuda'.

Conclusión personal

Para la elaboración de este proyecto, fue trascendental la investigación, principalmente en cuanto a encontrar información sobre como unir dos lenguajes, que en nuestro caso fueron Java y Prolog. Fue muy importante poder acoplar datos, documentos, apuntes, entre otras cosas que nos permitieron lograr la conexión entre ambos lenguajes. Se tuvo que empezar primero con un proceso de indagación de información que nos permitió lograr el objetivo.

En cuanto a la tarea programada, al realizarla aprendimos a manipular el lenguaje de Prolog, así como operar sus elementos, por ejemplo, la base de conocimientos. Se tuvo que investigar como operar esta base de conocimiento desde Java, para poder ingresar datos y compararlos para lograr imprimir la información que cumple con los requisitos de cada consulta.

La experiencia de trabajar con dos lenguajes que interactuaban fue muy enriquecedora, además la tarea sirvió para afianzar nuestros conocimientos sobre el paradigma lógico en que se basa Prolog.

Bibliografía

- Páginas de Internet
 - <http://kstopa.blogspot.com/2012/05/prolog-en-java.html>
 - <http://www.gnu.org/software/gnuprologjava/manual/Overview.html#Overview>
 - <http://www.gnu.org/software/gnuprologjava/manual/Installation.html#Installation>
 - <http://www.techytalk.info/prolog-programming-gprolog-linux/>
 - <http://www.gnu.org/software/gnuprologjava/>
 - <http://www.gnu.org/software/software.html>
 - <http://www.gprolog.org/>
 - <http://www.gnu.org/software/gnuprologjava/manual/Setup.html#Setup>
- Presentaciones vistas en clase