Tecnológico de Costa Rica

II Proyecto Programado

Lenguajes de Programación

José David Chaverri y Ricardo P. Gago
15/05/2012

Tabla de contenidos

Contenido

Introducción 3

Descripción del problema 4

Diseño del programa. 5

Librerías usadas 6

Análisis de resultados 8

Manual de usuario 10

Bibliografía 12

Introducción

En el Zoológico de San Diego, California, se han presentado muchos problemas con los encargados de los animales, debido que estos no cuentan con un mecanismo eficiente para obtener información de los animales, lo cual hace que no puedan realizar sus labores de manera eficiente. La situación ha llegado a tal extremo que recientemente se murieron mil pandas bebés porque nadie los alimentó.

Es por eso que se le ha encargado desarrollar un sistema de consulta de animales para dicho zoológico, con el fin de agilizar el trabajo de los encargados. La idea es que los encargados del zoológico puedan usar el sistema para ingresar la información de los animales, y posteriormente hacer consultas por características.

Descripción del problema

En la actualidad uno de los principales activos en una organización sin importar el tipo de organización que sea es sin lugar a duda la información, este conjunto de datos interrelacionados brindan a las organizaciones la capacidad de tener una mejor noción del pasado, un mejor manejo del presente y una mejor visión del futuro. A la hora de realizar las actividades ordinarias en una organización el ideal es que el encargado de cada tarea conozca a la perfección el conjunto de pasos necesarios para elaborar correctamente la tarea, además de las dificultades que se puede encontrar a la hora de realizarla, incluso la manera de resolver los problemas que surjan y en mejor de los casos una manera de mitigar los mismos.

Pero la realidad es otra, muchos de los problemas que atacan a las organizaciones actuales es la carencia de información a la hora de realizar las tareas, ya sea porque la información no existe del todo o que los medios de distribución de la información no sirven, o incluso ni siquiera existen. Por lo general los mismos problemas en las organizaciones son resueltos muchas veces, ya que no se cuenta con un sistema de documentación de labores y resolución de problemas, es decir la persona encuentra la solución a un inconveniente y no documenta cual fue su solución propuesta, luego pasa que otra persona tiene un problema similar o pude ser el mismo, y no se da cuenta de que ya otra persona lo había resuelto antes, esto es un gasto innecesario de tiempo, el cual se puede reducir si se tiene una correcta administración del conocimiento en las organizaciones.

Ante esta problemática muchas son las posibles soluciones que nos brindan las tecnologías actuales como el manejo de bases de datos, pero no las típicas bases de datos relacionales sino, las bases de datos multimedia, las cuales permiten almacenar información de una forma que resulta mas naturas para las personas que las consultan, tales como vídeos, fotografías, documentos y las bases de conocimientos que son el enfoque de aplicación de este trabajo, las cuales no solo almacenan datos y relaciones entre ellos, sino un conjunto de vinculación que va mas allá de llaves foráneas e índices.

Las labores en las distintas organizaciones varían desde preparar piezas para juguetes, hasta dispositivos de alta tecnologías para instalaciones espaciales, pero no cabe duda de que cada una de estas tareas cumplen un objetivo crucial en la sociedad actual, claro esta que en algunos contexto la labor de una persona puede ir mas allá de dañar un pieza o sistema, en este campo entran la tareas relacionadas con el cuidado de seres vivos como por ejemplo un zoológico donde se vela por la seguridad e integridad de muchos animales incluso algunos en vía de extinción, por esto es crucial que las persona que trabajan directamente con estos animales por ejemplo en labores como alimentación o cuidado veterinario estén seguros de que cada decisión que toman a la hora de cuidar a los animales no termine convirtiéndose en lo contrario y pueda dañarlos.

Diseño del programa.

El programa consiste en dos sistemas y un sistema de interfaz entre ellos.

- **Front_end**: Es el encargo de la interacción con el usuario, consiste en una interfaz gráfica elaborada en Python 2.7.3 y utilizando la librería gráfica Tkinter. Básicamente es una ventana con dos pestañas que brindar dos módulos de trabajo para el usuario.

- **Mantenimiento**: brinda una herramienta para añadir animales a la base de conocimiento mediante el uso de "Labels" para indicar el campo a ingresar y su "Entry" respectivo para que el usuario ingrese en valor a insertar en la base de conocimiento. Este modulo cuenta con un sistema de verificación el cual no permite que ningún campo quede vacío, eso con el objetivo de tener una base de conocimiento integra que brinde la suficiente información para el cuidado de los animales y obligue a si se desconoce un datos investigarlo o buscar la manera de averiguarlo.

- **Consulta de datos**: este modulo brinda al usuario una herramienta de consulta, la cual permite todas las combinaciones posibles de atributos de los animales. Los cuales son (raza, edad, genero, nombre, ecosistema, comida favorita), no es necesario que el usuario ingrese todos los atributos, basta con solo ingresar un dato y el sistema se encargara de mostrar los resultados respectivos.

- **Back_End**: Es el núcleo de la aplicación este sistema esta implementado en swi-prolog, el cual maneja la base de conocimiento de los animales del zoológico que son ingresados por los usuarios, este sistema cuenta con 2 módulos

- **Ingreso de hechos**: básicamente los hechos en esta base de conocimiento son los animales, cada hecho representa a un animal en particular, con los siguientes átomos internos o atributos

    - raza
    - edad
    - genero
    - nombre
    - ecosistema
    - comida favorita

- **Consulta de datos**: Consiste en el proceso de unificación en swi-prolog el cual recibe una consulta con un juego de átomos a unificar con los hechos de la base de conocimientos, estos básicamente son manejados por el proceso por el núcleo de unificación interno de swi-prolog.

- **Interfaz de conexión**: python ===>>> swi-prolog: Para este sistema de interfaz se utilizo la librería llamada pyswip, la cual fue desarrollada en el MIT, la cual permite el ingreso de hechos a la base de conocimiento además de realizar consultas a la misma.

Librerías usadas

Pyswip

Como se menciono en la parte de diseño del programa la librería usada para la conexión entre python y swi-prolog es pyswip.

Un ejemplo de su uso es el siguiente:

```
>>> from pyswip import Prolog
>>> prolog = Prolog()
>>> prolog.assertz("father(michael,john)")
>>> prolog.assertz("father(michael,gina)")
>>> list(prolog.query("father(michael,X)"))
[{'X': 'john'}, {'X': 'gina'}]
>>> for soln in prolog.query("father(X,Y)"):
...     print soln["X"], "is the father of", soln["Y"]
...
michael is the father of john
michael is the father of gina
```

Como se observa se puede implementar desde el "prompt" o desde un archivo.

Una vez instalada la librería pyswip (el proceso de instalación se explica con detalle en el manual de usuario) se procede a hacer un import de la misma y de las funciones a utilizar.

```
from pyswip import Prolog
```

Luego se crea la instancia que sera usada para manejar la base de conocimiento.

```
prolog = Prolog()
```

Después de crear la instancia se procede a insertar hechos o realizar consultas. Para la parte de añadir hechos a la base de conocimiento la pyswip es muy práctica.

```
prolog.assertz("father(michael,john)")
prolog.assertz("father(michael,gina)")
```

Una vez que se tiene los hechos en la base de conocimiento se puede realizar las consultas que se desean.

```
list(prolog.query("father(michael,X)"))
```

La consulta anterior muestra el siguiente resultado.

```
[{'X': 'john'}, {'X': 'gina'}]
```

Como se observa en los ejemplos anteriores el uso de esta librería es muy simple al menos en lo que amerita el alcance de esta tarea, por lo cual la decisión del uso de esta libraría se basa en lo simple que es la inserción de hechos y la obtención de datos por medio de las consultas.

Si desea información más detallada acerca de las diferentes funciones que tiene esta librería puede consultar la página.

http://code.google.com/p/pyswip/

Para obtener ejemplos de diferentes implementaciones usando esta librería puede consultar la siguiente página.

http://code.google.com/p/pyswip/wiki/Examples

Tkinter

Para la interfaz gráfica usada en el "front_end" se utilizo la librería de *python Tkinter*, esta librería permite distintos niveles de interfaz desde simples botones, hasta una practica interacción con el usuario por medio de paneles y pestañas.

EL proceso de instalación de la librería se explica en el manual de usuario.

La razón de uso de esta librería es que es muy práctica, existe gran cantidad de documentación y ejemplos en internet y además fue usada en cursos anteriores.

Para obtener más información acerca de esta librería puede consultar la siguiente página:

http://wiki.python.org/moin/TkInter

Análisis de resultados

A continuación se brinda una reseña de las tareas que se completaron y de las tareas que no se lograron completar, además de algunas propuestas de solución para los problemas que no se pudieron resolver.

Tarea	Estado	Propuesta de solución
Ingreso de datos sobre los animales	Completa	
Almacenamiento de la información según especificación	Completa	
Consultas según ciertos atributos	Completa	
Mostrar la información de los animales según los criterios de búsqueda	Completa	
Sistema estructurado en dos componentes	Completa	
Front-End encargado de la interacción con el usuario	Completa	
Entrada y salida de datos	Completa	
Comunicación con Back-End para las consultas	Completa	
El Back-End se encarga de la base de conocimientos	Completa	
El Back-End responde a las consultas	Completa	
La base de conocimientos son declaraciones en Prolog	Completa	

El Back-end se actualiza por medio de los datos ingresados	Completa	
Se muestra mensaje en caso de que no exista el animal con las características especificadas	Completa	
Back-End escrito en Prolog	Completa	
Uso de interfaz gráfica	Completa	
El proyecto funciona en el sistema operativo Linux	Completa	
Conexión de Prolog con otro lenguaje	Completa	

Manual de usuario

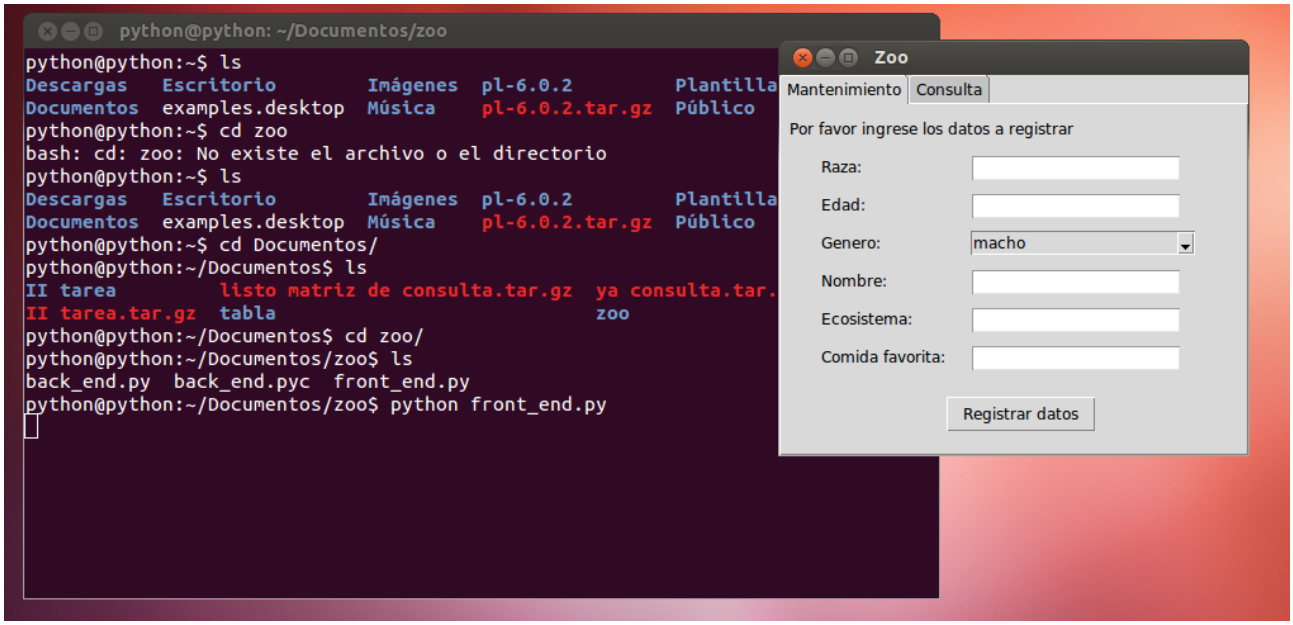
A continuación se brindan las instrucciones de uso de la modalidad mantenimiento y consulta, y las instrucciones de compilación del programa.

En primera instancia, es necesario realizar la instalación de PySwip, que detallo a continuación:

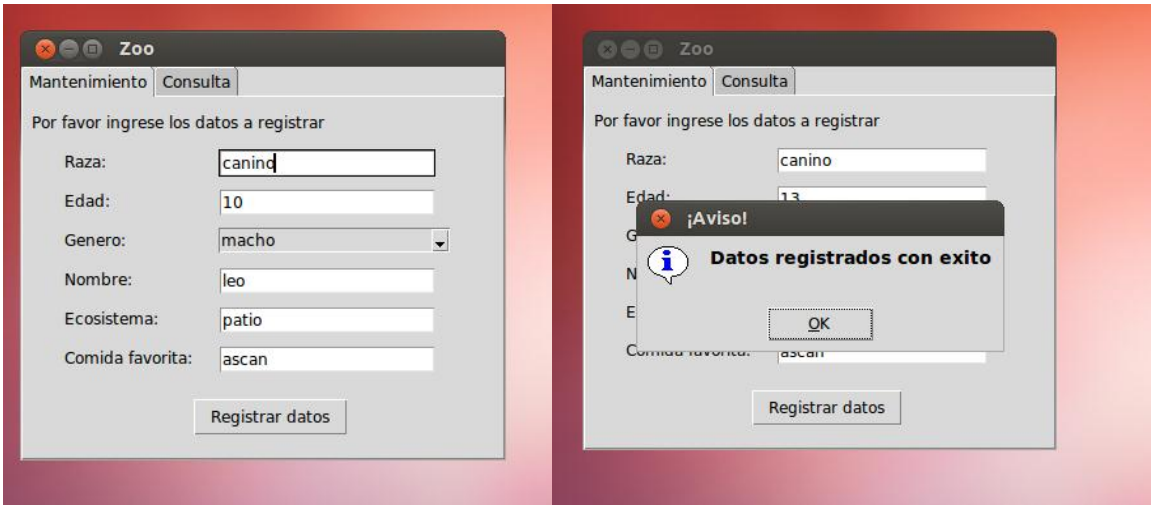
```
wget http://www.swi-prolog.org/download/stable/src/pl-6.0.2.tar.gz
tar xzvf pl-6.0.2.tar.gz
cd pl-6.0.2/
./configure --prefix=/usr --enable-shared
make && sudo make install
cd packages/clpqr/
./configure --prefix=/usr
make && sudo make install
sudo ln -s  /usr/lib/swipl-6.0.2/lib/i686-linux/libswipl.so /usr/lib/libpl.so
sudo ln -s  /usr/lib/swipl-6.0.2/lib/i686-linux/libswipl.so.6.0.2 /usr/lib/.
wget http://pyswip.googlecode.com/files/pyswip-0.2.2.zip
unzip pyswip-0.2.2.zip
cd pyswip-0.2.2/
sudo python setup.py install
```

1. Se obtiene el código fuente de pyswip desde la página web.
2. Se extrae el contenido de la carpeta.
3. Se ingresa a la carpeta que se descomprimió.
4. Configurar la fuente con la biblioteca compartida habilitada.
5. Compilar el código.
6. Instalar el código.
7. La biblioteca CLP es útil para problemas de manejo de restricciones, así que se debe instalar también.
8. Crear un enlace simbólico
9. A continuación, obtener e instalar ctypes de: http://starship.python.net/crew/theller/ctypes. Tenga en cuenta que no es necesario que lo instale si usted está usando Python 2.5.
10. Desempaquetar PySwIP e instalarlo con el python setup.py install.

Luego de realizar la instalación de la librería, el código se ejecuta desde la terminal de la siguiente manera:



A partir de allí, se pueden ingresar los datos del animal, donde es necesario rellenar todos los campos, y en caso de agregarlos de forma satisfactoria, el programa mostrará un mensaje de confirmación.



Para realizar la consulta, se puede ingresar cualquiera de los datos antes registrados, y se realiza la validación si no se ingresa nada. Luego de que el programa valide los datos ingresados, mostrará los resultados en una ventana emergente.

Para ingresar más datos de animales al programa solo es necesario volver al programa e ingresar los datos respectivos.

Para finalizar la ejecución, solo es necesario cerrar la aplicación. Cabe recordar que el programa solo funciona en el sistema operativo Linux, que cuenta con la librería de pyswip instalada correctamente.

Bibliografía

- Herrera Polo, Carlos. (2011, 04 de febrero). **[Python-es] pestañas en interfaz gráfica con tkinter**. Recuperado el 12 de mayo del 2012, de http://mail.python.org/pipermail/python-es/2011-February/029193.html

- stackoverflow.com. (2011, 23 de octubre). **Python 2.7.2 global variables+tkinter – Stack Overflow**. Recuperado el 09 de mayo del 2012, de http://stackoverflow.com/questions/7865682/python-2-7-2-global-variables-tkinter

- effbot.org. (2012). **The Tkinter EntryWidget**. Recuperado el 09 de mayo del 2012, de http://effbot.org/tkinterbook/entry.htm

- stackoverflow.com. (2011, 29 de julio). **Python, tkinter: how to select an item in ttk.Combobox.** Recuperado el 12 de mayo del 2012, de http://stackoverflow.com/questions/6876518/python-tkinter-how-to-select-an-item-in-ttk-combobox

- Juan J. Merelo. (2007, 04 de junio). **Tutoriales: G1 Mini-Introducción a Git**. Recuperado el 12 de mayo del 2012, de http://geneura.ugr.es/~jmerelo/tutoriales/git/

- git-scm.com. (2012). **Fundamentos de Git - Trabajando con repositorios remotos**. Recuperado el 12 de mayo del 2012, de http://git-scm.com/book/es/ch2-5.html

- Malcolm. (2010, 21 de octubre). **Python 2.7: Themed "common dialog" tkinter interfaces via Ttk?**. Recuperado el 09 de mayo del 2012, de http://stackoverflow.com/questions/3991130/python-2-7-themed-common-dialog-tkinter-interfaces-via-ttk

- nullege.com. (2012). **pyswip.Prolog**. Recuperado el 09 de mayo del 2012, de http://nullege.com/codes/search/pyswip.Prolog

- Yuce Tekol. (2012). **PySWIP enables querying SWI-Prolog in your Python programs**. Recuperado el 09 de mayo del 2012, de http://pypi.python.org/pypi/pyswip/0.2.0

- Python Software Foundation. (2009). **Una introducción informal a Python**. Recuperado el 09 de mayo del 2012, de http://docs.python.org.ar/tutorial/introduction.html

- Python Software Foundation. (2012, 09 de mayo). **Data Structures**. Recuperado el 09 de mayo del 2012, de http://docs.python.org/tutorial/datastructures.html