



**ESCUELA SUPERIOR
POLITÉCNICA DEL LITORAL**

DOCUMENTACIÓN DEL PROYECTO DE PYHTON

Materia: Lenguajes de Programación

Paralelo: 1

Fecha de entrega:
Miércoles, 19 de Febrero del 2014

PROYECTO: PARSEO DE XML

Autores:

- Cristina Barreno
- Sixto Castro
- Jordy Vasquez

19 de febrero de 2014

Índice

1. Introducción	4
2. Desarrollo	4
3. Alcances	4
4. Estilos de Query	5
5. Observaciones	6
6. Conclusiones	6

1. Introducción

El proceso de parseo consiste en transformar una serie de simbolos y transformarlos en una estructura apropiada para ser procesada, un parseador es un programa que se encarga de realizar este proceso. En el actual proyecto se realizará el parseo de un archivo xml en el lenguaje de programación Python.

Python es un lenguaje interpretado de propósito general que ha cojido popularidad en los ultimos años, siendo un lenguaje multiplataforma y multiparadigma.

2. Desarrollo

El proyecto consiste en realizar un parseo de un archivo xml en el lenguaje Python, para esto se necesito primero comprobar si el archivo xml que nos entregaron posee una sintaxis correcta, a partir de esto procedemos a realizar el parseo del xml, devolviendonos un árbol de parseo.

Se solicita que se realicen busquedas en el árbol estilos queries para lo cual se implementaron diferentes funciones en el proyecto para realizar estas busquedas.

3. Alcances

- Se recibe un archivo .xml y a partir de ese se realiza el árbol de parseo, pero antes se verifica si el xml que leemos tiene una sintaxis correcta.
- Se realizan diferentes tipos de consultas, durante este proceso se crea un archivo adicional en donde cada device del documento original posee los atributos que hereda del fallback que posea para realizar busquedas a futuras más eficientes.

4. Estilos de Query

Se crearon funciones que busquen ciertos tipos de queries básicos, estas funciones imprimen cuales y cuantos cumplen con un determinado query.

Los tipos de queries que se desarrollaron son:

- Búsqueda de un device que tenga un determinado atributo(ya sea que lo tenga en su declaración o lo obtenga por herencia).
- Búsqueda de todos los devices, grupos y capabilities disponibles.
- Búsqueda de todos las capabilities y grupos que posee un device.
- Búsqueda de los devices que poseen tal capability.
- Búsqueda de devices que poseen cierto grupo indicado.
- Búsqueda de un substing dentro de un atributo de algun device, group o capability

5. Observaciones

- Costo de tiempo de ejecución muy altos en ejecutar el algoritmos de búsqueda debido a que habia que hacer fallbacks de los diferentes devices.
- Para la recursividad en Python, los tiempos de ejecucion son demasidos grandes , debido a que el programa crea objetos en cada llamada , y esto hace lento al sistema.
- El metodo iterativo es mucho mas rapida que el metodo recursivo en Python
- Es de gran importancia tener en cuenta la tabulacion al momento de programar , debido a que Python lo requiere con exigencia.
- Cuando se trabaja de modo orientado a objetos usualmente se usa self que es una referencia a uno mismo similar al this en java
- Programar en Python es sencillo puesto que es muy parecido a programar en C y en Java
- El código es muy reducido debido a que se evita el uso de llaves por lo que es muy importante el espaciado que usemos.

6. Conclusiones

- Python es un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional.
- Existen ciertos cambios en la sintaxis entre las diferentes versiones de Python 2.7 y 3.3.
- Python es un lenguaje sencillo de aprender si se conoce como programar en C y en Java puesto que conocer esto lo hace muy intuitivo
- Realizar el parseador en Python conllevo mucho procesamiento y tiempo de ejecución a diferencia de cuando lo realizamos en Haskell el cual al ser un lenguajes funcional ya esta optimizado para este tipo de procesamientos.