

Lector de noticias RSS

Empleo de feeds para leer los contenidos de un sitio web

Actualmente vivimos en la Web 2.0, donde la mayoría de sitios Web (y todos los weblogs) separan el continente del contenido, esto es, la información del diseño. Esto permite una gestión del sitio mucho más eficiente. El formato de intercambio entre aplicaciones es por excelencia XML, ya que permite estructurar de manera flexible cualquier documento. Las Webs modernas permiten descargar un *feed*, que es un archivo XML con la información del sitio. Los feeds siguen varias convenciones en el tipo y orden de las etiquetas del XML, siendo los más populares RDF, RSS2 y Atom.

En esta práctica vamos a crear un lector de feeds, usando un fichero de ejemplo descargado de microsiervos.com, que en este caso es RSS2. El archivo RSS se reescribe automáticamente cuando se produce alguna actualización en los contenidos del sitio Web. Accediendo al archivo RSS es posible saber si se han actualizado los contenidos y con qué noticias o textos, pero sin necesidad de acceder al sitio Web desde el navegador.

Estructura del fichero XML

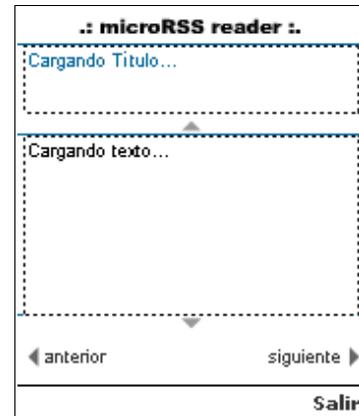
Todo feed consta de un nodo contenedor `rss`, donde se especifica qué versión de feed es, y un canal (nodo `channel`), donde se publican los contenidos. Cada elemento de información contenido dentro de un canal se llama `item`. Cada ítem tiene obligatoriamente un título, un resumen y un enlace a la página Web de origen (la que contiene el texto completo). Además puede incluir información adicional, como la fecha de publicación o el nombre del autor.

Lo que haremos será parsear el feed desde Flash Lite, usando únicamente las etiquetas que nos interesen. No es necesario procesar todas las etiquetas del XML, entre otras cosas porque existe mucha información redundante o inservible desde el punto de vista humano, y porque el teléfono móvil podría quedarse sin memoria si el feed tiene muchas entradas.

Creación de la interfaz

Vamos a poner dos campo de texto dinámicos y *multiline* para mostrar el título y la descripción de cada post, ambos renderizados como HTML y usando la tipografía `_sans`. El texto de las descripciones debe ser *scrollable*, puesto que debemos poder leerlo todo, mediante las teclas de dirección arriba y abajo. Para cambiar de un post a otro usaremos las teclas de dirección izquierda y derecha.

El resto de detalles de la interfaz se deja a elección del alumno. Por ejemplo se propone esta presentación:



En el fotograma de acciones empezaremos programando las funciones de las teclas:

```
var teclas:Object = new Object();
teclas.onKeyDown = function():Void {
    switch( Key.getCode() ){
        case Key.DOWN: texto.scroll++; break;
        case Key.UP: texto.scroll--; break;
        case Key.LEFT: anterior(); break;
        case Key.RIGHT: siguiente(); break;
        case ExtendedKey.SOFT2:
            fsccommand2("Quit"); break;
        default: break;
    }
};
Key.addListener(teclas);
```

Necesitaremos estas variables de ámbito global:

```
var info:Object;
var tema:Number = 0;
var numItems:Number = 0;
```

Ahora vamos a cargar el fichero XML, descartando las etiquetas vacías que sólo contienen espacios en blanco.

```
var xml:XML = new XML();
xml.ignoreWhite = true;
xml.onLoad = function():Void {
    // parseo del XML
};
xml.load("microsiervos.xml");
```

La función de parseo va a buscar el primero de los nodos del tipo `item`, extraerá el título y la descripción, calculará cuántos post existen y le dará formato.

Para acceder al primer nodo con la información que nos interesa usamos esta notación:

```
info = xml.firstChild.firstChild;
```

Para calcular cuántos ítems existen en el XML usaremos el siguiente bucle:

```
for (var i:Number = 0; i < info.childNodes.length; ++i) {  
    if (info.childNodes[i].nodeName == "item") numItems++;  
}  
trace("hay " + numItems + " posts.");
```

Y seguidamente tenemos que acceder a la información, que lo haremos mediante otra función:

```
ponTema();
```

Y hasta aquí la función de parseo. Vamos a definir la función para mostrar el contenido de cada ítem. El primer índice del XML es 0, luego para acceder al primero de los nodos hijo del primer ítem usaremos el índice 11, que iremos incrementando mediante la anterior variable contadora `tema`.

```
function ponTema():Void {  
    var tit:String = info.childNodes[11+tema].childNodes[0].firstChild.nodeValue;  
    var des:String = info.childNodes[11+tema].childNodes[2].firstChild.nodeValue;  
    titulo.text = tit;  
    texto.htmlText = des;  
    texto.scroll = 0;  
};
```

Por último queda crear las funciones para ir cambiando de post:

```
function anterior():Void {  
    tema--;  
    if (tema < 0) tema = numItems-1;  
    ponTema();  
};  
function siguiente():Void {  
    tema++;  
    if (tema >= numItems) tema = 0;  
    ponTema();  
};
```