

# Exemplo práctico da utilidade das linguaxes de marcas

## Situación inicial

Temos un montón de textos (libros) na nosa biblioteca, queremos organizalos, pero non soamente iso, senón que queremos poder facer buscas por título, contido, etc... mediante algunha aplicación informática ou script/programa que creemos.

Supoñamos que temos textos planos deste estilo:

```
Memorias dun neno labrego
Conta como era o estilo de vida nos anos corenta dun rapaz labrego de doce
ou trece anos nunha aldea galega.
Xosé Neira Vilas
Isaac Díaz Pardo
Galicia
galega
Realidade social
Narrativa
Follas Novas
Edicións do Castro
5 de xaneiro de 1961
978-84-8485-253-7
Eu son Balbino. Un rapaz de aldea. Coma quen dis, un ninguén. E ademais,
pobre.....
```

## Busca nos ficheiros

Calquera ser humano se decatara do que é cada apartado, pero se quixeramos automatizar a busca/organización de varios miles de ficheiros deste tipo nos atoparíamos unha serie de problemas.

## Problema 1

Por exemplo, se quixeramos buscar algún título para ver se temos ese libro podemos supoñer que a primeira liña vai ser o título, e que temos que contar que a segunda liña sexa o resumo, a terceira o autor, etc...

Se todos e cada un dos ficheiros se adhíren ao anterior podemos facer as nosas buscas baseándonos nesa premisa.

Pero que pasa se por calquera motivo cambiamos de orde as liñas? Isto pode ser por algún erro ou porque xuntemos ficheiros de distintas fontes que non seguen todos o mesmo formato.

Dado que o noso programa supón que a orde é inmutable non vai ter maneira de distinguir se un texto é un autor, un título, etc... e calquera arquivo que non siga de forma estrita a ordenación dará resultados erróneos.

No noso exemplo se buscamos “Vilas” neste ficheiro o atopará:

```
Memorias dun neno labrego
Conta como era o estilo de vida nos anos corenta dun rapaz labrego de doce
ou trece anos nunha aldea galega.
Xosé Neira Vilas
```

Pero se o buscamos neste ficheiro de formato incorrecto, non o dará atopado a pesar de que é do mesmo autor:

```
Contos vellos pra rapaces novos
Xosé Neira Vilas
Libro de contos de todo o mundo
```

Peor aínda, podería dar falsos positivos, neste caso atoparía o texto “Vilas” a pesar de non ser un apelido:

```
Xeografía da comarca
Fulanito Menganitez
Vilas e cidades da comarca e outras edificacións.
```

## Problema 2

Un problema menos aparente, fixédevos que temos dúas editoriais:

```
Follas Novas
Edicións do Castro
```

Normalmente un libro ten unha editorial, pero neste caso ten dúas. Como podemos saber cando comeza unha editorial e cando comeza outro campo?

Non hai maneira, teríamos que poñer as editoriais nunha mesma liña e separalas con algún carácter, coma unha coma, punto...

## Problema 3

Se por erro metemos un retorno de carro nalgún lado provocará que se cree unha nova liña, o que provocará un erro en cadea.

Para ilustrar o problema imaxinádevos que por erro a alguén se lle escapa un retorno de carro xusto nos apelidos do autor:

```
Contos vellos pra rapaces novos
Xosé Neira
Vilas
Libro de contos de todo o mundo
```

Sen outro contexto o programa non ten maneira de saber que foi un erro e tratará “Vilas” coma o resumo do libro e a partir de aí o resto de liñas serán mal interpretadas.

## Posible solución

Podemos probar a formatar o texto deste xeito, poñendo unha primeira palabra que explica que é cada cousa.

```
Título  Memorias dun neno labrego
Resumo Conta como era o estilo de vida nos anos corenta dun rapaz labrego
de doce ou trece anos nunha aldea galega.
```

Autor/a Xosé Neira Vilas  
 Ilustrador/a Isaac Díaz Pardo  
 Orixe Galicia  
 Lingua galega  
 Tema(s) Realidade social  
 Xénero(s) Narrativa  
 Editorial Follas Novas  
 Editorial Edicións do Castro  
 Data de pub. 5 de xaneiro de 1961  
 ISBN ISBN 978-84-8485-253-7  
 Texto Eu son Balbino. Un rapaz de aldea. Coma quen dis, un ninguén. E ademais, pobre....

Entón lle podemos dicir ao noso programa que a primeira palabra define o que é esa liña, deste xeito nos libramos dos problemas:

- Problema 1: aínda que cambie a orde das liñas o noso programa só se vai fixar na primeira palabra. Se busca o título da igual que sexa a primeira que a cuarta liña, buscará “Título” e obviará o resto.
- Problema 2: isto tamén se soluciona indicando que despois de atopar unha coincidencia debe seguir a buscar por se hai outro elemento, coma Editorial, que se repite.
- O problema 3 non se solucionaría, pois pensará que a nova liña ten unha palabra chave, que se non coincide coa busca a descartará.

Se vos decatades o anterior que temos creado xa é unha linguaxe de marcas. Ten unha serie de caracteres (marca) que é a palabra chave que vai ao inicio. Non é moi diferente ca Markdown ou Textile.

## Solución empregando XML

En unidades didácticas posteriores veremos XML, pero o importante neste exemplo é ver como englobamos cada elemento dentro dunhas etiquetas:

```

<titulo> Memorias dun neno labrego </titulo>
<resumo> Conta como era o estilo de vida nos anos corenta dun rapaz
labrego de doce ou trece anos nunha aldea galega. </resumo>
<autor>Xosé Neira Vilas</autor>
<ilustrador>Isaac Díaz Pardo</ilustrador>
<orixe>Galicia</orixe>
<lingua>galega</lingua>
<tema>Realidade social </tema>
<xenero> Narrativa </xenero>
<editorial> Follas Novas </editorial>
<editorial> Edicións do Castro </editorial>
<datadepub> 5 de xaneiro de 1961 </datadepub>
<isbn> ISBN 978-84-8485-253-7 </isbn>
<texto> Eu son Balbino. Un rapaz de aldea. Coma quen dis, un ninguén. E
ademais, pobre.... </texto>
  
```

Ao igual que coa nosa “protolinguaxe de marcas” o problema 1 non se pode producir. Nós buscaremos o autor dentro das etiquetas `<autor></autor>` e non nos importa a orde, as outras etiquetas son ignoradas na busca estean onde estean.

O problema 2 igualmente non afecta. Podemos ter unha, dúas ou centos de etiquetas `<autor>`, `<editorial>`, etc... A busca se efectuará en todas as etiquetas que coincidan co campo buscado.

Neste caso ademais solucionamos o problema 3, nos vai dar igual ter isto:

```
<autor>Xosé Neira Vilas</autor>
```

ou isto

```
<autor>Xosé Neira  
Vilas</autor>
```

Dado que buscamos a etiqueta de inicio pero tamén a de fin, co cal non nos importa que haxa saltos de liña, ata que non apareza a etiqueta de peche seguirá procesando a o contido da mesma.