

TEI Publisher: ediciones sofisticadas sin necesidad de programar

...

Helena Bermúdez Sabel & Magdalena Turska

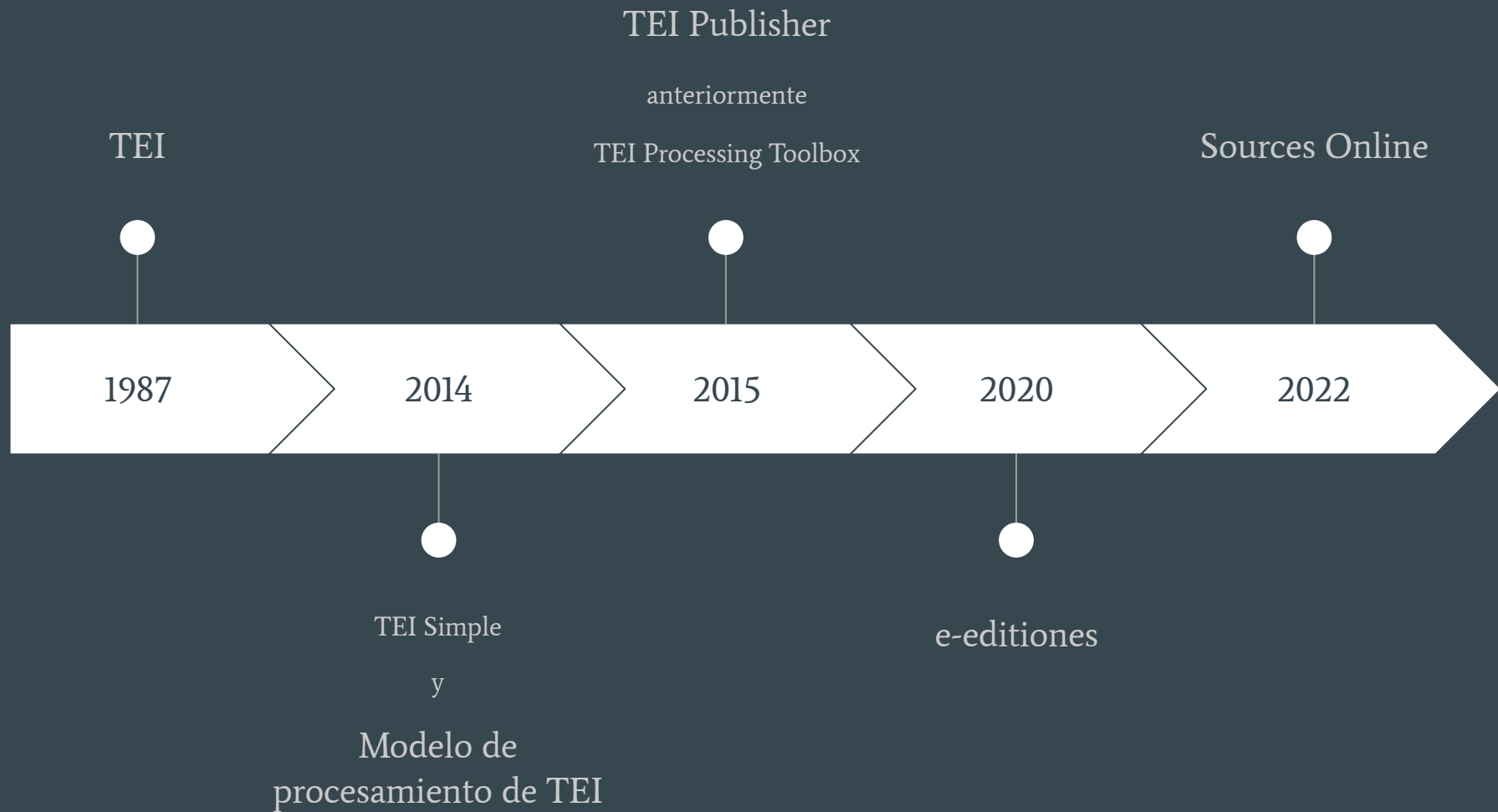
TEI Publisher



enraizado en estándares
desarrollado por la comunidad
con visión de futuro

Por qué TEI Publisher?

- Software libre de código abierto
- Basado en estándares abiertos, con opciones por defecto que facilitan la anotación y la publicación sin necesidad de programar
- Comunidad: [e-editiones](#) (organización de base con el objetivo de sustentar ediciones a partir de la promoción de estándares abiertos y colaboración comunitaria)



Demo

- Thebarum Fabula
- Alfred Escher
- Register rokopisov slovenskega slovstva
- LGPN-Ling
- EDEP

Modelado en TEI

- Definición del vocabulario
- Formalización de las restricciones
- Diseño del output/salida



Customización de TEI

TEI ODD: One Document Does it All

- Seleccionar módulos
- Eliminar elementos innecesarios
- Añadir nuevos elementos y/o atributos
- Cambiar el nombre de un elemento o atributo
- Limitar los valores de un atributo
- Limitar la estructura
- Manipular agrupaciones funcionales de elementos
- Internacionalización (i-18)
- Documentación
- **Indicar cómo los diferentes elementos tienen que ser procesados**

Modelo de procesamiento de TEI

- Una de las partes centrales de TEI Publisher: la implementación del modelo de procesamiento de TEI
- Descripción de reglas de transformación en TEI (independiente de la implementación)

Modelo de procesamiento de TEI

- Oficialmente parte de las Directrices de TEI desde 2016
- Más sustentable y habitualmente más rápido que transformaciones ex profeso
- Es la fuente de todas las transformaciones textuales de TEI Publisher

TEI Source

```
<TEI>  
<text>  
  <body>
```

TEI Fragment

```
<div type="chapter">  
  <head>  
    <p>
```

```
<div type="chapter">  
  <head>  
    <list>  
      <item>
```



ODD

```
<elementSpec ident="div">  
  <model predicate="@type='chapter'"  
    behaviour="section">  
    <model behaviour="block">
```

```
<elementSpec ident="head">  
  <model behaviour="heading">
```

```
<elementSpec ident="list">  
  <model behaviour="list">
```



HTML

FO

LaTeX

ePub

Más que una conversión de TEI a HTML

Formatos de entrada:

- TEI
- DocBook
- MS Word
- JATS
- Markdown

Formatos de publicación:

- Interfaz web
- EPUB
- PDF (y fuente LaTeX)
- Markdown

Tarea 1

```
<elementSpec id="name" mode="change">
  <model predicate="@type='place'" behaviour="inline">
    <outputRendition>
      font-variant: small-caps;
    </outputRendition>
  </model>
  <model behaviour="inline">
    <outputRendition>
      color: #CC6600;
    </outputRendition>
  </model>
</elementSpec>
```

add MODE: CHANGE



Add Element



Jump to element ...


Element Specs

- TEI
- ab
- abbr
- add
- choice
- del
- ex
- expan
- gap
- lb
- note
- persName

^ model [inline]

@place="overstrike"



Output 

Mode
[Processing mode - passed to subsequer]


Description

[Document the model]

Predicate

1 @place="overstrike"

behaviour

inline  or [Custom Behaviour]

CSS Class

[Define CSS class name (for external CSS)]

Template

<|> <...> <X> [...]
 1 [Define code template to apply to content]

Parameters +

Name	Parameter	
content	1 ('«', '..', '»')	<input type="checkbox"/> set 

Renditions +

☐ Use source rendition

Comportamientos

- alternate
- anchor
- block
- body
- break
- cell
- cit
- document
- figure
- graphic
- heading
- inline
- list
- listItem
- metadata
- note
- omit
- paragraph
- pass-through
- row
- section
- table
- text
- title
- webcomponent

Predicado

- En el atributo **@predicate** establecemos la condición que se tiene que cumplir para que ese modelo sea aplicado
- El valor de **@predicate** es una expresión XPath

outputRendition

- Limitarse a aspectos editoriales que expresen características específicas del texto
- La apariencia general debe ser determinada por el tema externo
- Puedes asignar una clase CSS a través del atributo `cssClass` para ser utilizada en hojas de estilo CSS externas

Multiples modelos

- Se aplicará el primer modelo con un predicado coincidente o sin predicado.
- Atención: los modelos siguientes se ignorarán, aunque coincidan

Parámetros

```
<elementSpec ident="date" mode="change">
  <model output="web" predicate="@when" behaviour="alternate">
    <param name="default" value="."/>
    <param name="alternate" value="format-date(@when, '[FNn]', [D1o] [MNn], [Y]')"/>
  </model>
  <model predicate="text()" behaviour="inline"/>
</elementSpec>
```

Parámetros

- Todos los comportamientos toman al menos un parámetro implícito: **content**, que se corresponde con el contenido a procesar
 - si no se especifica el parámetro **content**, se establecerá en el nodo actual
- Algunos comportamientos necesitan parámetros adicionales, p. Ej.:
 - **link**: **uri**
 - **alternate**: **default** and **alternate**

modelSequence

```
<elementSpec ident="title" mode="change">
  <modelSequence predicate="parent::titleStmt/parent::fileDesc">
    <model predicate="preceding-sibling::title" behaviour="text">
      <param name="content" value="' - '"/>
    </model>
    <model behaviour="inline"/>
  </modelSequence>
</elementSpec>
```

Tareas 2 y 3