

Taller de mercado TEI

Nicolás Vaughan

Universidad de los Andes

n.vaughan@uniandes.edu.co

13 y 14 de abril de 2023

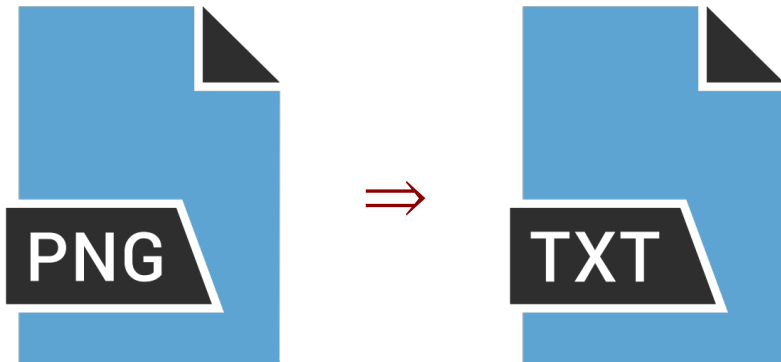
¿Qué significa 'digitalizar'?

1. *Escanear* — transformar una imagen analógica a un archivo digital gráfico (.png, .tiff, .jpeg, etc).



¿Qué significa 'digitalizar'?

2. *OCR* — reconocimiento óptico de caracteres



¿Qué significa ‘digitalizar’?

3. *Marcado / etiquetado (tagging)* — darle *significado* al texto

- representacional o gráfico (HTML, \LaTeX , Troff, XSL-FO, etc.)
- semántico:
 - análisis y crítica textual
 - lingüístico
 - análisis cualitativo (qdap, ATLAS.ti, NVivo, etc.)
 - ...

The screenshot shows a web browser window with the URL `cloud.atlasti.com`. The page title is "Interview with Thomas Muhr". The main content area displays an interview transcript with two questions and their answers. The right sidebar contains a list of categories, each with a small icon and a label. The categories are: "Product development", "Innovation", "reason: digital future", "Adaptation", "ATLAS.ti product fam...", "desktop-cloud: relation", and "ATLAS.ti product fam...".

Q1: Why did you develop a Web-based version of ATLAS.ti?

ATLAS.ti has been permanently reinventing itself, most notably by completely redesigning and redeveloping its flagship, ATLAS.ti Windows, from the ground up in the past few years. By putting the program on a completely new code base and by integrating the latest available technology, we have opened it up for a new leap forward in innovation. This spirit of continuous innovation has always been important for us and, in many ways, has been our guiding principle.

ATLAS.ti has always been the fore-runner in the field, and we will continue to set standards. Embracing cloud technology and "moving into the Web" is only one of the logical steps in that direction. ATLAS.ti Cloud is a new part of our family of products, and it will strengthen our capabilities in an important way. Now, I would like to emphasize that this does not mean that we have any notion of abandoning the desktop versions—quite to the contrary: ATLAS.ti Windows and ATLAS.ti Mac are and will remain the main flagships of the ATLAS.ti family. They pack the most computing power and offer the most sophisticated tools and functions that are required for any in-depth, professional-level data analysis. As such, they form the gravitational center, so to speak, and ATLAS.ti Cloud, along with the mobile apps function like powerful satellites in their orbit.

Q2: What are the central ideas behind ATLAS.ti Cloud?

First and foremost, we want ATLAS.ti Cloud to complement and enrich the capabilities of our entire product family (ATLAS.ti Windows, Mac, iOS, and Android) further.

Categories:

- Product development
- Innovation
- reason: digital future
- Adaptation
- ATLAS.ti product fam...
- desktop-cloud: relation
- ATLAS.ti product fam...

Codificación categorías en ATLAS.ti

```
<!-- ... -->  
<h1>Este es un título de nivel 1</h1>  
<em>Este texto aparece en cursivas</em>  
y <strong>este en negritas.</strong>
```

```
<br />
```

Una lista de viñetas:

```
<ul>  
  <li>un ítem</li>  
  <li>otro ítem</li>  
  <li>otro ítem</li>  
</ul>
```

```
<br />
```

```
<span style="color: red">  
  Este texto va en rojo.  
</span>  
<!-- ... -->
```

Este es un título de nivel 1

Este texto aparece en cursivas y **este en negritas**.

Una lista de viñetas:

- un ítem
- otro ítem
- otro ítem

Este texto va en rojo.

```
<!-- ... -->
```

```
<persName>Moctezuma Xocoyotzin</persName> nació en <date>1466</date>,
y murió el <date>29 de junio en 1520</date>
en <placeName>Tenochtitlan</placeName>, <placeName>México</placeName>.
```

```
<!-- ... -->
```

¿Qué es TEI?

TEI (*Text Encoding Initiative*) es una implementación del lenguaje de marcado XML diseñada para codificar o marcar semánticamente textos de diversas índoles.

Por su parte, XML (*Extensible Markup Language*) es un lenguaje de marcado general usado para codificar todo tipo de información.

Ejemplo de un documento XML

```
<?xml version="1.0"?>
<catalog>
  <book id="bk101">
    <author>Gambardella, Matthew</author>
    <title>XML Developer's Guide</title>
    <genre>Computer</genre>
    <price>44.95</price>
    <publish_date>2000-10-01</publish_date>
    <description>An in-depth look at creating applications
      with XML.</description>
  </book>
  <book id="bk102">
    <author>Ralls, Kim</author>
    <title>Midnight Rain</title>
    <genre>Fantasy</genre>
    <price>5.95</price>
    <publish_date>2000-12-16</publish_date>
    <description>A former architect battles corporate zombies,
      an evil sorceress, and her own childhood to become queen
      of the world.</description>
  </book>
</catalog>
```

XML: algunas definiciones

1. *Elementos* — son los pilares estructurales de todo documento XML. Se componen de:

- una *etiqueta de apertura*
- una *etiqueta de cierre*
- un *contenido* (que puede ser otros elementos, texto o nada)
- y opcionalmente unos *atributos* con sus *valores* correspondientes.

Ejemplos:

- `<persName>Moctezuma Xocoyotzin</persName>`
- `<date when="2022-12-31">31 de diciembre de 2022</persName>`
- `<date when="2022-12-31" calendar="juliano">diciembre 31</persName>`
- `<persName>`
 - `<forename>William</forename>`
 - `<surname>Shakespeare</surname>``</persName>`
- `<lb></lb>` (o equivalentemente `<lb/>`)
- `<lb n="3"></lb>` (o equivalentemente `<lb n="3"/>`)

2. *Entidades*: XML contiene cinco caracteres que no pueden usarse literalmente sino solo por medio de una referencia:

" "

& &

&apos '

< <

> >

3. *Padres, hijos, ancestros y descendientes*: si un elemento contiene otro elemento, el primero se denomina el *padre*, y el segundo el *hijo*. Un elemento puede tener muchos *ancestros* y muchos *descendientes*.
4. *Declaración*: está al principio de todo documento XML, identificándolo como tal: `<?xml version="1.0" encoding="UTF-8"?>`

5. *Instrucciones de procesamiento*: van debajo de la declaración XML y especifican el modo como el documento debe ser validado semánticamente o procesado. Empiezan con `<?` y terminan con `?>`.

Por ejemplo, para validar un documento XML con el esquema de validación de TEI (más exactamente, el de TEI-all), debemos incluir lo siguiente:

```
<?xml-model href="http://www.tei-c.org/release/xml/tei/custom/schema/relaxng/tei_all.rng" type="application/xml"
  schematypens="http://relaxng.org/ns/structure/1.0"?>
<?xml-model href="http://www.tei-c.org/release/xml/tei/custom/schema/relaxng/tei_all.rng" type="application/xml"
  schematypens="http://purl.oclc.org/dsdl/schematron"?>
```

Reglas fundamentales de todo documento XML

1. Todo documento XML debe tener un único elemento raíz.

CORRECTO:

```
<biblio>
  <libro>
    <título>Cien años de soledad</título>
    <autor>Gabriel García Márquez</autor>
  </libro>
  <libro>
    <título>El coronel no tiene quien le escriba</título>
    <autor>Gabriel García Márquez</autor>
  </libro>
</biblio>
```

INCORRECTO:

```
<libro>
  <título>Cien años de soledad</título>
  <autor>Gabriel García Márquez</autor>
</libro>
<libro>
  <título>El coronel no tiene quien le escriba</título>
  <autor>Gabriel García Márquez</autor>
</libro>
```

2. Todo elemento empieza con una etiqueta de apertura y cierra con una etiqueta de cierre.

CORRECTO:

```
<p>  
  <title>Cien años de soledad</title>  
  <author>Gabriel García Márquez</author>  
</p>
```

INCORRECTO:

```
<p>  
  <title>Cien años de soledad  
  <author>Gabriel García Márquez</author>  
</p>
```

3. Todo elemento debe ser apropiadamente anidado.

CORRECTO:

```
<p>  
  <q>Esta es una cita</q>  
</p>
```

INCORRECTO:

```
<p>  
  <q>Esta es una cita</p>  
</q>
```

4. Los nombres de los elementos no pueden empezar con 'xml', números o puntuación (excepto '_').

CORRECTO:

```
<author>  
<_author>
```

INCORRECTO:

```
<01_author>  
<"author">
```

5. Los espacios en blanco (caracteres de espacio, de tabulador y de salto de línea) *no* son significativos. XML suele tragarse los espacios múltiples.

Esto:

```
<p>
  <title>
    Cien      años      de      soledad
  </title>
  </p>
```

es equivalente a esto:

```
<p><title>Cien años de soledad</title></p>
```


- Un documento XML es *sintácticamente* válido si cumple con las reglas anteriores.
- Un documento XML es *semánticamente* válido si cumple con las reglas de un esquema de validación.
 - Un documento **XML-TEI** es semánticamente válido si cumple con las reglas prescritas por el consorcio TEI sobre el tipo de elementos (y sus atributos) y las relaciones existentes entre ellos.
 - Por ejemplo, que la raíz de todo documento debe ser el elemento `<TEI xmlns="http://www.tei-c.org/ns/1.0">`.
 - Y que dicho elemento debe tener obligatoriamente dos elementos hijos: `<teiHeader>` y `<body>`.
 - Y que el elemento `<p>` puede tener algunos atributos (e.g. `ana`, `cert`, `copyOf`, etc.), pero no puede tener otros (e.g. `type`).
 - Y así sucesivamente.

<https://www.tei-c.org/release/doc/tei-p5-doc/en/html/index.html>

Software que usaremos

- El editor gratuito Visual Code Studio:

<https://code.visualstudio.com/>

- Dos extensiones para ese editor:

1. Scholarly XML:

<https://marketplace.visualstudio.com/items?itemName=raffazizzi.xml>

2. tei-publisher-vscode:

<https://marketplace.visualstudio.com/items?itemName=e-editiones.tei-publisher-vscode>

`https://github.com/nivaca/taller-tei-2023`

Bajemos la plantilla básica de TEI.