

Programming historian en español, 2021.

Introducción a la publicación web de archivos TEI con CETElcean.

del Rio Riande, Gimena y Calarco, Gabriel.

Cita:

del Rio Riande, Gimena y Calarco, Gabriel (2021). *Introducción a la publicación web de archivos TEI con CETElcean. Programming historian en español,*.

Dirección estable: <https://www.aacademica.org/gabriel.calarco/14>

ARK: <https://n2t.net/ark:/13683/prgr/1HC>



Esta obra está bajo una licencia de Creative Commons.
Para ver una copia de esta licencia, visite
<https://creativecommons.org/licenses/by-nd/4.0/deed.es>.

Acta Académica es un proyecto académico sin fines de lucro enmarcado en la iniciativa de acceso abierto. Acta Académica fue creado para facilitar a investigadores de todo el mundo el compartir su producción académica. Para crear un perfil gratuitamente o acceder a otros trabajos visite: <https://www.aacademica.org>.

Introducción a la publicación web de archivos TEI con CETElcean

Gabriel Calarco y Gimena del Río Riande

Esta lección te enseña los pasos necesarios para publicar en línea un archivo de TEI usando CETElcean

Contenidos

- [Introducción y software que usaremos](#)
- [Paso 1: Crear una estructura para nuestros archivos](#)
- [Paso 2: Cargar y previsualizar el archivo TEI](#)
- [Paso 3: Mejorar la visualización de nuestro archivo](#)
- [Paso 4: Para seguir trabajando con CETElcean](#)
- [Referencias](#)

Nota: Para seguir este tutorial de forma comprensiva debes saber qué es el lenguaje de marcado XML-TEI desarrollado por la [Text Encoding Initiative o TEI](#) y cuál es su función como lenguaje estándar en la edición digital académica de textos de Ciencias Sociales y Humanidades. Puedes encontrar recursos y tutoriales en español sobre codificación de textos con TEI en [TTHub](#). Asimismo, te recomendamos las partes 1 y 2 de la lección [Introducción a la codificación de textos en TEI de Nicolás Vaughan](#) y la [Introducción a la Text Encoding Initiative de Susanna Allés](#). Durante este tutorial se utilizarán otros lenguajes informáticos (como [JavaScript](#) y [CSS](#)), pero no es necesario tener conocimientos previos sobre su funcionamiento para utilizar [CETElcean](#).

Introducción y software que usaremos [introducción-y-software-que-usaremos](#)

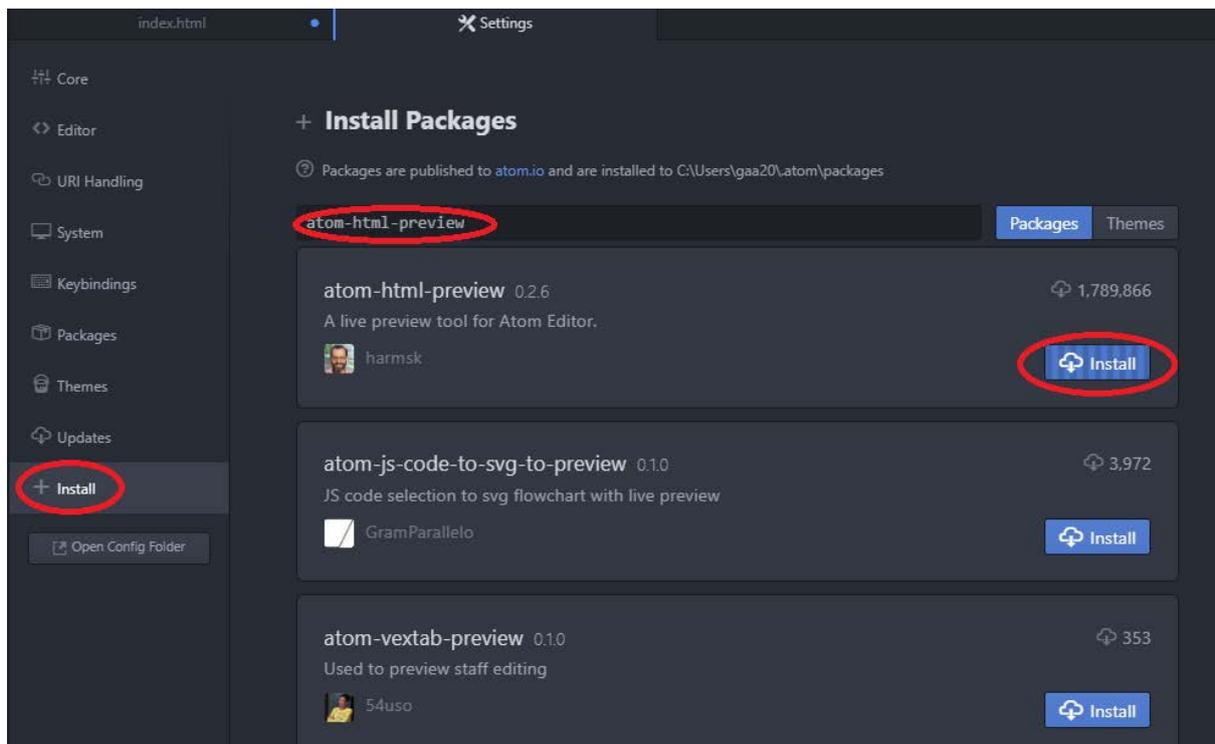
Para quienes se inician en el uso de TEI, uno de los escollos más comunes es que, una vez que se han codificado los textos con este lenguaje de marcado, es difícil saber cómo hacer para publicarlos en línea. Para ser visualizados en un navegador, los archivos XML-TEI deben ser transformados primero a [HTML](#) mediante el uso de plantillas [XSLT](#). Sin embargo, este proceso requiere de conocimientos técnicos y herramientas que no siempre se encuentran al alcance de todos los humanistas digitales, especialmente de quienes se acercan al uso de TEI por primera vez, de quienes aún no conocen en profundidad el

manejo de software de edición, o de quienes no cuentan con acceso a servidores propios. CETElcean es un software de edición digital que permite visualizar archivos XML-TEI en el navegador sin que necesitemos aplicarles una transformación XSLT.

Este tutorial te guiará a través de los pasos necesarios para publicar un archivo TEI en línea utilizando CETElcean, una librería abierta escrita en el lenguaje de programación JavaScript. CETElcean permite que los documentos TEI se muestren en un navegador web sin transformarlos primero a HTML. CETElcean carga el archivo TEI dinámicamente en el navegador y cambia el nombre de los elementos de TEI por otros en HTML, de tal forma que estos nos permitan visualizar en el navegador web los fenómenos textuales que marcamos en nuestros archivos usando TEI.

En primer lugar, una aclaración sobre la visualización de tu trabajo: el método por defecto de CETElcean para mostrar archivos TEI consiste en cargar los archivos desde otra ubicación. Sin embargo, no todos los navegadores te permitirán cargar los archivos si estos se encuentran almacenados en tu computadora. Puedes hacer el intento, pero si eso no funciona, tendrás que generar un servidor local, colocar los archivos en un servidor en línea, o utilizar un editor de código con funciones de previsualización. Para el caso de este tutorial, seguiremos esta última opción, ya que usaremos el editor [Atom](#), con el plug-in atom-html-preview. No obstante, existen otras opciones libres para editar archivos TEI y generar previsualizaciones de HTML, como [jEdit](#) o [Visual Studio Code](#), y versiones propietarias como [Oxygen](#).

Deberás entonces descargar e instalar [Atom](#) antes de continuar con este tutorial. Con Atom ya funcionando, instala el plug-in atom-html-preview (creado por Kyle J. Harms) que podrás encontrar abriendo el menú de opciones de Atom (file > settings o cntrl+). En la pantalla de "Settings" ve a la pestaña "Install" y en el cuadro de diálogo introduce atom-html-preview. Cuando aparezca el plug-in que estamos buscando en la lista de resultado debes hacer clic en el botón azul que dice "Install":



Proceso de instalación del plug-in de Atom para previsualizar archivos en HTML

Usaremos como texto de prueba la crónica conocida como *La Argentina Manuscrita*, del hispano-guaraní [Ruy Díaz de Guzmán](#). Este texto del siglo XVII hace uso del topónimo Argentina por primera vez, para referirse a los extensos territorios del Cono Sur que componían el Río de la Plata y sus adyacencias, es decir, territorios de la actual Argentina, Paraguay, Uruguay, sur de Brasil y Bolivia. Puedes encontrar una edición digital completa del texto en: <http://hdlab.space/La-Arentina-Manuscrita>.

Comenzaremos con un archivo simple (aunque un tanto extenso) en formato TEI P5, que queremos hacer visible en un navegador web: [Ruy Diaz-La Argentina Manuscrita.xml](#). Para descargar el archivo haz clic derecho sobre el enlace de descarga y selecciona la opción 'Save Link As...'.

Paso 1: Crear una estructura para nuestros archivos [paso-1-crear-una-estructura-para-nuestros-archivos](#)

Comenzaremos por establecer una estructura para nuestros archivos, es decir, una carpeta contenedora con el nombre 'tutorial_es' con las subcarpetas y archivos que te indicaremos a continuación. Puedes descargar el directorio completo del repositorio [CETELcean en GitHub](#) y trabajar en la carpeta 'tutorial_es', o puedes descargar los archivos individualmente, siempre y cuando mantengan la misma estructura que en GitHub, que es la siguiente:

tutorial_es/

|

```
|--- css/
|
|--- tei.css
|
|--- js/
|
|--- CETEI.js
|
|--- Ruy_Diaz-La_Argentina_Manuscrita.xml
|--- README.md (el archivo que estas leyendo)
```

El siguiente paso será crear un archivo nuevo en Atom con el nombre index.html. Para ello puedes ir a File > New o utilizar el atajo Ctrl + N (Cmd + N en Mac). En este documento deberás copiar y pegar el siguiente contenido:

```
<!DOCTYPE html>

<html>

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <meta http-equiv="X-UA-Compatible" content="ie=edge">

</head>

<body>

</body>

</html>
```

A continuación, debes guardar este archivo en el directorio raíz (en nuestro caso la carpeta 'tutorial_es'); recuerda que su título debe ser index.html. Este archivo servirá como una estructura en la cual pondremos las instrucciones para mostrar nuestros archivos TEI. Al igual que en TEI, los archivos HTML tienen un encabezado, llamado head y un cuerpo de texto, llamado body. A lo largo de este tutorial usaremos este archivo para agregar enlaces a nuestra CSS (*Cascading Style Sheet*, también llamada *hoja de estilo* u [hoja de estilos en cascada](#) en español) y a nuestros archivos de JavaScript, y escribiremos un poco de JavaScript para lograr una visualización de nuestro documento TEI que refleje los aspectos del marcado que nos interesa destacar. En la primera línea vacía del <head>, escribe:

```
<link rel="stylesheet" href="css/tei.css">
```

Esto conectará nuestro archivo CSS con nuestra página HTML, dándole acceso a las directivas de estilo que este contiene (solo hay unas pocas, pero agregaremos más). A continuación, incluiremos la librería de CETEIcean, añadiendo la siguiente línea luego del enlace a la hoja de estilo:

```
<script src="js/CETEI.js"></script>
```

En este punto nuestro archivo index.html debe tener el siguiente contenido:

```
<!DOCTYPE html>

<html>

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <meta http-equiv="X-UA-Compatible" content="ie=edge">

  <!-- Líneas agregadas -->

  <link rel="stylesheet" href="css/tei.css">

  <script src="js/CETEI.js"></script>

  <!-- Líneas agregadas -->

</head>

<body>
```

```
</body>
```

```
</html>
```

Paso 2: Cargar y previsualizar el archivo TEI [paso-2-cargar-y-previsualizar-el-archivo-tei](#)

Ahora ya estamos listos para cargar el archivo TEI. Para eso, debemos añadir una secuencia de comandos informáticos conocida habitualmente por su nombre en inglés “[script](#)”, que nos permitirá recuperar el documento TEI de *La Argentina manuscrita* en nuestro archivo HTML (el que estamos editando en este momento). Copia y pega las siguientes líneas de código a continuación del último elemento que agregamos:

```
<script>
```

```
let c = new CETEI();
```

```
c.getHTML5('Ruy_Diaz-La_Argentina_Manuscrita.xml', function(data) {
```

```
    document.getElementsByTagName("body")[0].appendChild(data);
```

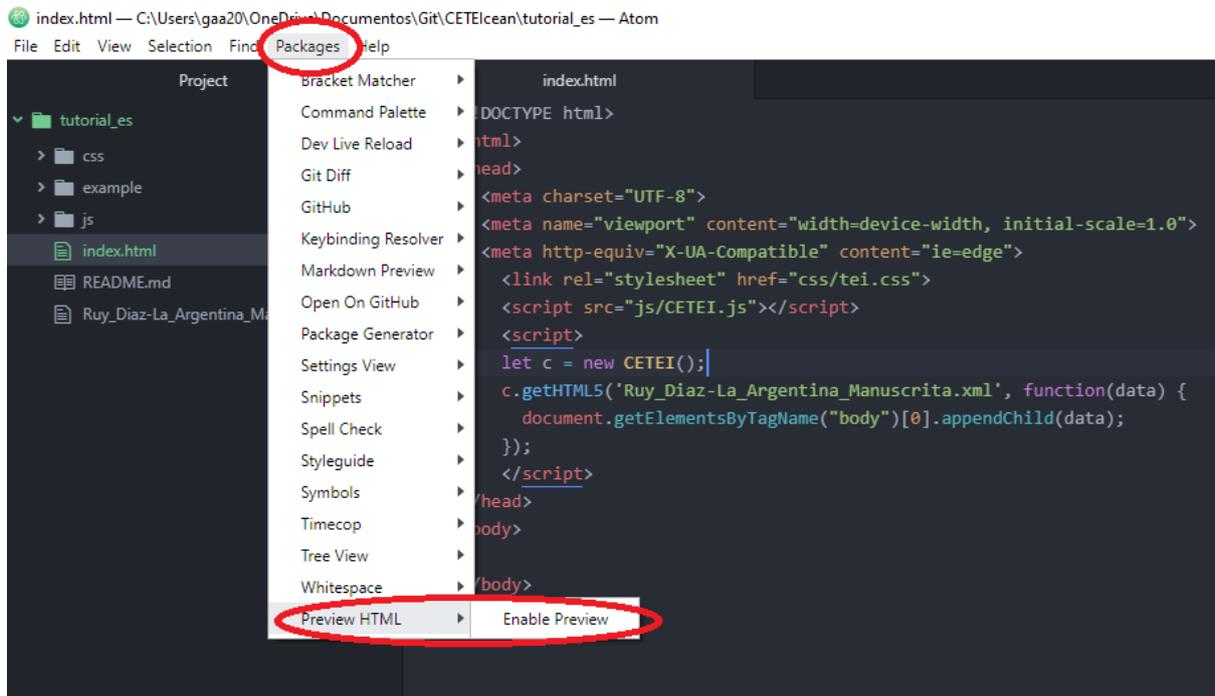
```
});
```

```
</script>
```

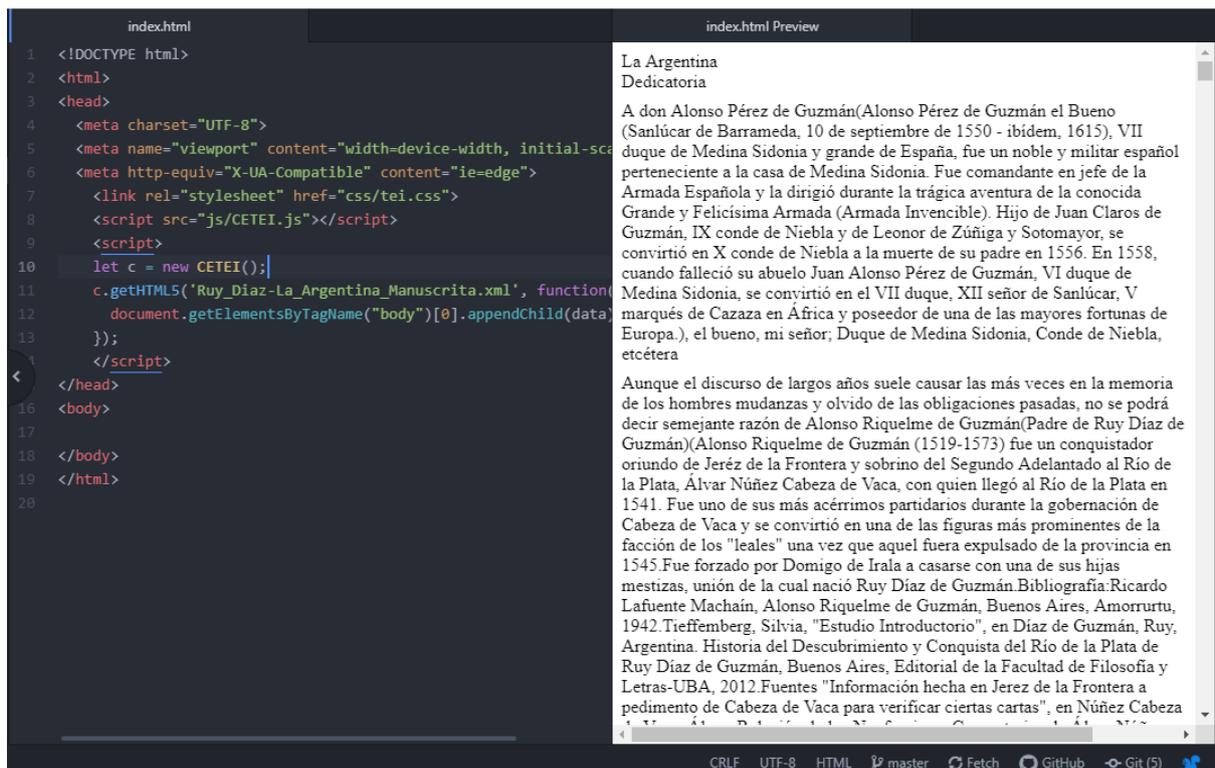
No necesitas ser un experto en JavaScript para usar CETEIcean, pero aprender su funcionamiento básico puede ser de utilidad. Si deseas incluir funciones avanzadas, tendrás que aprender JavaScript. En la red para desarrolladores de Mozilla puedes encontrar una excelente [guía de JavaScript](#) en varias lenguas, incluido el español. Para el caso de este tutorial, solo te contaremos que las líneas de código que añadimos hacen varias cosas:

- En primer lugar, una variable `c` es definida como un nuevo objeto CETEI. Esto hará el trabajo de cargar y darle estilo a nuestro archivo fuente
- A continuación, le indicaremos a `c` que cargue el archivo fuente y lo convierta en HTML ([Custom Elements](#)), y también le daremos una función que tomará los resultados y los pondrá en el `<body>` de nuestro archivo `index.html`
- En la línea `document.getElementsByTagName('body')` se llama a una función que busca todos los elementos `<body>` y los devuelve en la forma de una lista ordenada (una lista en la cual se puede acceder a los miembros que la componen a través de su número índice)
- En nuestro ejemplo solo hay un elemento `<body>`, por lo que obtendremos una sola entrada en nuestra lista, con el índice 0. Este ítem, que es un elemento HTML, queda adjunto como un hijo del documento TEI que acabamos de cargar

En este punto deberías poder ejecutar una previsualización del HTML desde el menú “Packages” y así ver tu documento. Vamos a previsualizarlo con el plug-in que instalamos al inicio de este tutorial. Entonces, ve a la pestaña packages del menú superior y elige en el menú desplegable la opción “Preview HTML / Enable preview”:



Menú de opciones para previsualizar archivos en HTML en Atom



Primera previsualización de nuestro archivo TEI con CETEIclean

Si no estás usando Atom, puedes hacer esto mismo colocando tus archivos en un servidor web. Si conoces el funcionamiento de GitHub, puedes utilizar GitHub Pages (aquí tienes un [tutorial](#) en inglés) y crear un repositorio. Si tienes instalado Python en tu computadora, puedes ejecutar un servidor web simple en el directorio de este tutorial (en nuestro caso la carpeta 'tutorial_es'). Con este fin debes abrir la consola de comandos y comprobar que te encuentras en la carpeta deseada (en caso contrario puedes navegar hasta esa carpeta con el comando `cd + url del archivo`, por ejemplo: `cd Documentos/tutorial_es`) e ingresar el comando:

```
python -m SimpleHTTPServer
```

También es posible que tu computadora ya tenga los programas necesarios para ejecutar un servidor web, o puedes instalar [MAMP](#) o algún otro programa similar. El objetivo de crear este servidor es visualizar nuestros archivos TEI en el navegador como si estos se trataran de un contenido online.

Paso 3: Mejorar la visualización de nuestro [archivo](#)

Esta primera visualización tendrá varios errores que deberemos arreglar. Para eso volveremos a nuestro trabajo en Atom. Comenzaremos por añadir una hoja de estilo para manipular los elementos de TEI en nuestro archivo y luego añadiremos funciones de CETEIcean para hacer modificaciones más complejas. Si todavía no le has echado un vistazo al archivo fuente XML, es un buen momento para hacerlo, para ver lo que CETEIcean ya está haciendo y lo que no. Podemos observar que el contenido del `teiHeader` no está siendo mostrado, como tampoco los comienzos de página y comienzos de líneas. Los elementos `div` y `p`, por su parte, están siendo formateados como bloques y las notas aparecen en el cuerpo del texto entre paréntesis. Con un poco de investigación sobre las posibilidades de codificación de la TEI, verás que hay siete tipos de elementos TEI en el body de nuestro documento fuente:

- [div](#)
- [head](#)
- [note](#)
- [p](#)
- [persName](#)
- [placeName](#)
- [rs](#)

Algunos de estos elementos pueden no necesitar estilos o comportamientos especiales, pero otros definitivamente lo necesitarán. Echa un vistazo al archivo `tei.css` de la carpeta `css/`. Como puedes ver, por ahora tiene solo unas pocas reglas:

```
tei-div {
```

```
display: block;

}

tei-p {

display: block;

margin-top: .5em;

margin-bottom: .5em;

}
```

Algunas cosas para tener en cuenta: los nombres de los elementos en nuestros selectores CSS tienen el prefijo `tei-`. Esto es necesario para que CETELcean pueda convertir los elementos de TEI en elementos personalizados ([Custom Elements](#)) de HTML. Estas reglas establecen que los elementos `<div>` se visualicen como bloques (empiezan en una nueva línea y terminan con un corte); lo mismo sucede con los párrafos, que también tienen un espaciado superior e inferior.

Decidir qué estilos aplicar a los elementos que todavía no tienen reglas de estilo puede no resultar sencillo, pero podemos comenzar eligiendo algunos de los casos más simples. En nuestro documento fuente se señalan los encabezados de los capítulos y de las diferentes secciones mediante el elemento `<head>`. Probablemente desearemos que estos encabezados se destaquen del cuerpo del texto, para lograrlo podemos utilizar CSS para darles un estilo diferente. Ahora debes abrir el archivo `tei.css` (que encontrarás en la carpeta “css”) en Atom y al final del documento agregar las siguientes líneas:

```
tei-head {

font-size: 2em;

font-weight: bold;

}
```

Verás que esta no es una solución perfecta, ya que tenemos diferentes niveles de elementos `<div>`, y sería apropiado que los encabezados de diferentes niveles tuvieran diferentes tamaños para identificarlos. Debido a que los elementos `<div>` de nuestro archivo TEI no indican a qué nivel pertenecen, esto puede resultar difícil de lograr con CSS. Sin embargo, también podemos utilizar los comportamientos (behaviors) de CETELcean para dar formato.

En HTML, la convención es representar los diferentes niveles de encabezados con los elementos `h1`, `h2`, `h3`, etcétera (hasta `h6`). Podemos lograr esto utilizando un

comportamiento. Para ello, en tu archivo index.html añade lo siguiente entre la primera y la segunda línea del código que se encuentra entre las etiquetas `<script></script>` (es decir, entre `"let c = new CETEI();" y "c.getHTML5('Ruy_Diaz-La_Argentina_Manuscrita.xml'...)"`):

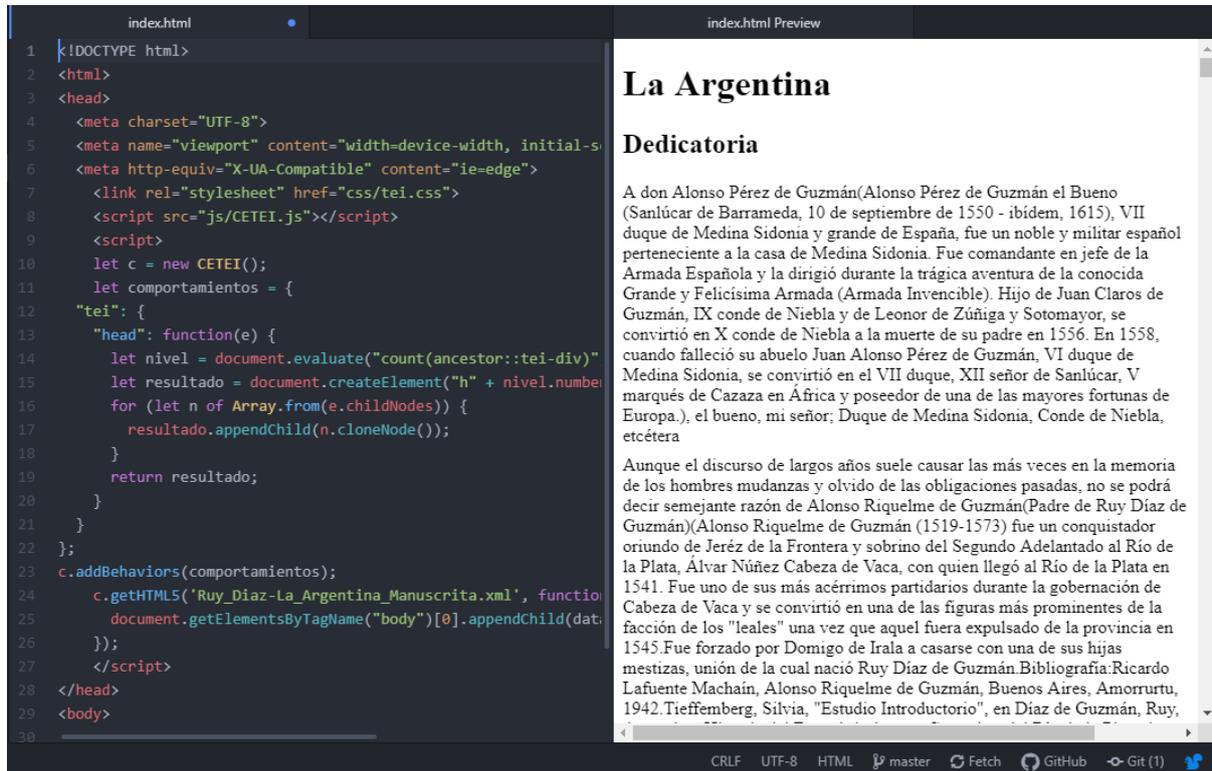
```
let comportamientos = {  
  
  "tei": {  
  
    "head": function(e) {  
  
      let nivel = document.evaluate("count(ancestor::tei-div)", e, null,  
XPathResult.NUMBER_TYPE, null);  
  
      let resultado = document.createElement("h" + nivel.numberValue);  
  
      for (let n of Array.from(e.childNodes)) {  
  
        resultado.appendChild(n.cloneNode());  
  
      }  
  
      return resultado;  
  
    }  
  
  }  
  
};  
  
c.addBehaviors(comportamientos);
```

Esto creará un objeto Javascript y le asignará la variable `comportamientos`, que luego enlazaremos con el objeto `CETEI` que creamos antes, usando el método `addBehaviors` (que ya viene incluido en `CETEIcean`). En el interior de ese objeto tenemos una sección etiquetada como `"tei"` (que es el prefijo para todos nuestros elementos personalizados), y dentro de esta se definen los comportamientos para los elementos. Cuando `CETEIcean` encuentra una coincidencia para el nombre de un elemento, como `"head"` (ten en cuenta que se utiliza el nombre de `TEI` sin el prefijo), aplica los comportamientos correspondientes.

Este nuevo comportamiento toma una función de JavaScript, lo que hace que el elemento sea procesado como un parámetro (el `e`). Esto crea la variable `nivel`, que contiene el nivel de encabezamiento de la `<tei-div>` que contiene el `<tei-head>`, crea un elemento `<h[nivel]>` con el nivel correspondiente, y copia el contenido del elemento original en el nuevo elemento de encabezado. `CETEIcean` esconderá el contenido de `<tei-head>` y, en cambio, mostrará el contenido del nuevo elemento de encabezado. Ten en cuenta que este código tiene un problema potencial: un documento con muchas divisiones anidadas unas dentro de otras podría llegar a producir un elemento de encabezado superior al límite admitido por HTML (por ejemplo un elemento `<h7>`). Nuestro documento fuente no tiene más de dos

niveles de anidamiento, pero para utilizarlo en otras fuentes sería prudente revisar que el anidamiento no supere el nivel del elemento <h6>.

Si en este punto previsualizamos nuestro HTML en Atom, obtendremos el siguiente resultado:



```
index.html
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-s
6   <meta http-equiv="X-UA-Compatible" content="ie=edge">
7   <link rel="stylesheet" href="css/tei.css">
8   <script src="js/CETEI.js"></script>
9   <script>
10    let c = new CETEI();
11    let comportamientos = {
12    "tei": {
13      "head": function(e) {
14        let nivel = document.evaluate("count(ancestor::tei-div)"
15        let resultado = document.createElement("h" + nivel.number
16        for (let n of Array.from(e.childNodes)) {
17          resultado.appendChild(n.cloneNode());
18        }
19        return resultado;
20      }
21    }
22  };
23  c.addBehaviors(comportamientos);
24  c.getHTML5("Ruy_Diaz-La_Argentina_Manuscrita.xml", function
25    document.getElementsByTagName("body")[0].appendChild(dat
26  });
27 </script>
28 </head>
29 <body>
30
```

index.html Preview

La Argentina

Dedicatoria

A don Alonso Pérez de Guzmán(Alonso Pérez de Guzmán el Bueno (Sanlúcar de Barrameda, 10 de septiembre de 1550 - ibidem, 1615), VII duque de Medina Sidonia y grande de España, fue un noble y militar español perteneciente a la casa de Medina Sidonia. Fue comandante en jefe de la Armada Española y la dirigió durante la trágica aventura de la conocida Grande y Felicísima Armada (Armada Invencible). Hijo de Juan Claros de Guzmán, IX conde de Niebla y de Leonor de Zúñiga y Sotomayor, se convirtió en X conde de Niebla a la muerte de su padre en 1556. En 1558, cuando falleció su abuelo Juan Alonso Pérez de Guzmán, VI duque de Medina Sidonia, se convirtió en el VII duque, XII señor de Sanlúcar, V marqués de Cazaza en África y poseedor de una de las mayores fortunas de Europa.), el bueno, mi señor; Duque de Medina Sidonia, Conde de Niebla, etcétera

Aunque el discurso de largos años suele causar las más veces en la memoria de los hombres mudanzas y olvido de las obligaciones pasadas, no se podrá decir semejante razón de Alonso Riquelme de Guzmán(Padre de Ruy Díaz de Guzmán)(Alonso Riquelme de Guzmán (1519-1573) fue un conquistador oriundo de Jerez de la Frontera y sobrino del Segundo Adelantado al Río de la Plata, Alvar Núñez Cabeza de Vaca, con quien llegó al Río de la Plata en 1541. Fue uno de sus más acérrimos partidarios durante la gobernación de Cabeza de Vaca y se convirtió en una de las figuras más prominentes de la facción de los "leales" una vez que aquel fuera expulsado de la provincia en 1545.Fue forzado por Domingo de Irala a casarse con una de sus hijas mestizas, unión de la cual nació Ruy Díaz de Guzmán.Bibliografía:Ricardo Lafuente Machain, Alonso Riquelme de Guzmán, Buenos Aires, Amorrurtu, 1942.Tieffemberg, Silvia, "Estudio Introdutorio", en Díaz de Guzmán, Ruy,

Previsualización de nuestro archivo TEI con estilo para los títulos

Con esta previsualización hemos mejorado notablemente la presentación de nuestro documento, pero las notas de la edición todavía dificultan la lectura del texto. Para solucionar este problema agregaremos un comportamiento más a nuestro script. Sin embargo, para lograr este objetivo, tendremos que usar una secuencia de comandos un tanto más extensa y compleja que la anterior. Copia y pega el siguiente texto entre las líneas "tei": { y "head": function(e) { que se encuentran en el segundo elemento <script> de nuestro documento index.html:

```
"note": function(e){
if (!this.noteIndex){
  this["noteIndex"] = 1;
} else {
  this.noteIndex++;
}
```

```
/* El primer bloque verifica si hay notas en el texto y las ordena en una secuencia*/
```

```
let id = "note" + this.noteIndex;  
  
let link = document.createElement("a");  
  
link.setAttribute("id", "src" + id);  
  
link.setAttribute("href", "#" + id);  
  
link.innerHTML = this.noteIndex;  
  
let content = document.createElement("sup");  
  
if (e.previousSibling.localName == "tei-note") {  
    content.appendChild(document.createTextNode(", "));  
}
```

```
/* El segundo bloque le añade un número a cada nota*/
```

```
content.appendChild(link);  
  
let notes = this.dom.querySelector("ol.notes");  
  
if (!notes) {  
    notes = document.createElement("ol");  
  
    notes.setAttribute("class", "notes");  
  
    this.dom.appendChild(notes);  
}
```

```
/* El tercer bloque crea una sección de notas al final del documento*/
```

```
let note = document.createElement("li");  
  
note.id = id;  
  
note.innerHTML = "<a href=\"#src\" + id + \"\">^</a> " + e.innerHTML  
  
notes.appendChild(note);
```

```
return content;
},
```

/* Finalmente, el cuarto bloque crea una lista con las notas y las enlaza con las referencias en el cuerpo del texto*/

A los fines de completar este tutorial no es necesario entender el funcionamiento de cada línea de este comportamiento. Sin embargo, si observas el resultado de la previsualización, notarás que al incluirlo las notas aparecen al final del texto, hipervinculadas con sus respectivas referencias:

```
14   if (!this.noteIndex){
15       this["noteIndex"] = 1;
16   } else {
17       this.noteIndex++;
18   }
19   let id = "note" + this.noteIndex;
20   let link = document.createElement("a");
21   link.setAttribute("id", "src" + id);
22   link.setAttribute("href", "#" + id);
23   link.innerHTML = this.noteIndex;
24   let content = document.createElement("sup");
25   if (e.previousSibling.localName == "tei-note") {
26       content.appendChild(document.createTextNode(","));
27   }
28   content.appendChild(link);
29   let notes = this.dom.querySelector("ol.notes");
30   if (!notes) {
31       notes = document.createElement("ol");
32       notes.setAttribute("class", "notes");
33       this.dom.appendChild(notes);
34   }
35   let note = document.createElement("li");
36   note.id = id;
37   note.innerHTML = "<a href=\"\#src\" + id + \"\"></a>" + e.innerHTML
38   notes.appendChild(note);
39   return content;
40 }
41 "head": function(e) {
42     let nivel = document.evaluate("count(ancestor::tei-div)", e, null, XPathResu
```

La Argentina Dedicatoria

A don Alonso Pérez de Guzmán¹, el bueno, mi señor; Duque de Medina Sidonia, Conde de Niebla, etcétera

Aunque el discurso de largos años suele causar las más veces en la memoria de los hombres mudanzas y olvido de las obligaciones pasadas, no se podrá decir semejante razón de Alonso Riquelme de Guzmán^{2,3}, mi padre, hijo de Rui Diaz de Guzmán⁴, mi abuelo, vecino de Jerez de la Frontera, antiguo servidor de esa ilustrísima casa, en la cual, habiéndose mi padre criado desde su niñez hasta los 22 años de su edad, sirvió de paje y secretario del Excelentísimo señor don Juan Alarcón de Guzmán, y mi señora la Duquesa doña Ana de Aragón⁵, dignísimos abuelos de vuestra Excelencia, de donde el año de 1540 pasó a las Indias⁶ con el Adelantado Alvar Núñez Cabeza de Vaca^{7,8}, su tío, gobernador del Río de la Plata⁹, a quien sucediendo las cosas más adversas que favorables, fue preso y llevado a España, quedando mi padre en esta provincia donde fue forzoso asentar casa, tomando estado de matrimonio con doña Úrsula de Irala¹⁰, mi madre, hija del gobernador Domingo Martínez de Irala¹¹; y continuando el real servicio, al cabo de 50 años falleció de esta vida, dejándome con la misma obligación como a primogénito suyo, la cual de mi parte siempre he tenido presente, en el reconocimiento y digno respeto de su memorable fama; de donde vine a tomar atrevimiento de ofrecer a vuestra Excelencia este humilde y pequeño libro, que compuse en medio de las vigiliadas de mi profesión, sirviendo a su Majestad desde mi puericia hasta ahora: y puesto que el tratado es de cosas menores, y faltar de toda erudición y elegancia, al fin es

Previsualización de nuestro archivo TEI con estilo para las notas

Paso 4: Para seguir trabajando con CETEIcean [paso-4-para-seguir-trabajando-con-ceteicean](#)

CETEIcean posee una cantidad de comportamientos integrados que puedes reemplazar o desactivar al agregarles valores. Si, por ejemplo, deseas mostrar el contenido del TEI Header (que está oculto por defecto), puedes añadir la siguiente línea a nuestro <script> debajo de "tei": {:

```
"teiHeader": null,
```

Si haces esto, puede que desees agregar estilos de CSS o comportamientos para elegir la forma en la que se visualizará el contenido del TEI Header en el navegador.

En este tutorial no agotamos todas las posibilidades para la presentación de nuestro documento fuente. Te invitamos a que continúes experimentando por tu cuenta en las diferentes formas en las que un marcado de TEI puede visualizarse en un navegador usando CETElcean. Puedes encontrar más información en [CETElcean](#).

Referencias [referencias](#)

Allés Torrent, Susanna; del Rio Riande, Gimena, y Calarco, Gabriel. 2018-. *TTHub. Text Technologies Hub. Recursos sobre tecnologías del texto y edición digital*. <https://tthub.io/>

Allés Torrent, Susanna. 2019. "Introducción a la Text Encoding Initiative". *TTHUB. Text Technologies Hub: Recursos sobre tecnologías del texto y edición digital*. <https://tthub.io/aprende/introduccion-a-tei/>

Atom. A hackable text editor for the 21st Century. <https://atom.io>

Cayless, Hugh y Viglianti, Raffaele. CETElcean. <http://teic.github.io/CETElcean/>

del Rio Riande, Gimena; De León, Romina, y Hernández, Nidia. 2019. *Historia de la conquista del Río de la Plata o La Argentina manuscrita*. <http://hdlab.space/La-Argentina-Manuscrita/>

Jedit. Programmer's text editor. Stable Version: 5.6.0. <http://www.jedit.org/>

Oxygen. XML Editor. <https://www.oxygenxml.com/>

Vaughan, Nicolás. 2021. "Introducción a la codificación de textos en TEI (parte 1)", *Programming Historian en español* 5 (2021), <https://doi.org/10.46430/phes0053>

Visual Studio Code. <https://code.visualstudio.com/>

Acerca de los autores

Gabriel Calarco es Licenciado en Letras por la Universidad de Buenos Aires (UBA) y se encuentra cursando el doctorado en Literatura en la misma universidad como becario del Instituto de Investigaciones Bibliográficas y Crítica Textual (IIBICRIT-CONICET). También se desempeña como docente en la Diplomatura de Humanidades Digitales (UCES) y como profesor de Literatura en el Nivel Medio.

Gimena del Rio Riande es investigadora adjunta del Instituto de Investigaciones Bibliográficas y Crítica Textual (IIBICRIT, CONICET) y directora del laboratorio de Humanidades Digitales de CAICYT (CONICET, Argentina). Es directora de la Diplomatura en Humanidades Digitales (UCES), co-directora del master en Humanidades Digitales (UNED.), directora de la Revista de Humanidades Digitales (RHD) y presidenta de la Asociación Argentina de Humanidades Digitales (AAHD).

Cita sugerida

Gabriel Calarco y Gimena del Río Riande, "Introducción a la publicación web de archivos TEI con CETEIcean", *Programming Historian en español* 5 (2021), <https://doi.org/10.46430/phes0056>.