Memoria técnica sobre transformación de PDF a Epub2 y Epub3

En la transformación de libros de PDF a Epub por medio de Calibre surgen diversos problemas. Problemas que no he resuelto aún en totalidad, dado que fui resolviendolos particularmente a medida que surgían. La presente nota es un cuaderno de bitácora del camino que debe realizarse para la transformación de un archivo PDF en formato Epub. Dicha transformación se realiza por medio de Calibre y Sigil. Si bien Julieta realizó un documento tutorial del programa y de cómo transformar un libro de PDF a Epub, ella misma reconoce que hay muchos problemas en esa transformación y que el mismo documento explicativo no es suficiente. (Surgieron graves problemas al transformar un libro que ni ella misma podía darme una explicación, más cuando el libro posee una inmensa cantidad de imágenes y de cuadros sinópticos como es el caso de Cirugía)

Por lo tanto, a continuación, voy a ir mencionando las cosas aprendidas hasta el momento, en un lenguaje coloquial, no informático, sencillo. Pido disculpas si los términos son vagos e imprecisos. Poco a poco los iré incorporando y especificándolos.

Primera nota fundamental: en la incorporación de imágenes al Sigil para el lector en unidades móviles NO hay que usar el formato BMP. El BMP no es un buen formato para los libros en Epub. *Se debe usar siempre el formato PNG*, más liviano y más sencillo de leer para los dispositivos. Además, la transformación de un libro PDF a Epub muchas veces puede tener el problema de la visualización de sus imágenes. Es muy probable que una imagen de formato BMP no permita ver las imágenes en Epub. Esto es clave. Más cuando las imágenes son muchas. De esto hay que estar atento. Que el formato siempre sea PNG y no BMP.

Segunda nota fundamental: para realizar el índice del Sigil, lo que se llama el “Book Browser” en inglés, *debe realizarse sin la incorporación de acentos ni ningún otro tipo de signos de puntuación.* El programa no te lo permite. Además, hay que revisar dentro del código del archivo, en el HTML, que las imágenes agregadas (de nuevo, en formato PNG y NO en formato BMP) estén con la descripción de la imágen (muchas veces, por ejemplo, “Selección_142” y sin acento incorporado) Todo acento significa más información que debe “¿procesarse?” y que retarda el funcionamiento del dispositivo. SI la descripción tiene acento, entonces, se realiza Buscar y Sustituir. Las opciones dentro de HTML, se busca un patrón (en este caso “Selección” -con acento) y se lo reemplaza por la misma palabra, pero sin acento. En todos los archivos.

Sin embargo, primero lo primero. *¿Cuáles son los beneficios del formato EPUB?* La publicación electrónica tiene muchas ventajas. Su principal característica es su adaptabilidad: el formato se amolda a los diferentes dispositivos que lo utilizan. Siempre el formato EPUB otorga independencia del contenido y la forma, lo que supone que puede ser adaptado a cualquier dispositivo de lectura de forma rápida y automática. Se adapta a los diferentes tamaños de pantalla y fuentes de la mayoría de lectores de libros electrónicos, redimensionando lo mostrado en una página de manera dinámica, según sea necesario. Ante un libro de gran extensión, el Sigil permite incorporar al EPUB una tabla de contenidos,

lo cual es de mucha utilidad puesto que permite ubicar cada capítulo de forma sencilla y directa.

No hay una única manera de transformación del PDF, dado que cada uno presenta diferentes problemas y desafíos propios (incorporación de muchas imágenes, hojas formadas en doble columna, inmensa cantidad de notas al pie, entre otras cosas) Sin embargo, esto no quita poder determinar un estándar mínima de tratamiento general, más allá de las particularidades de cada caso. El Calibre ofrece la posibilidad de lo que llama un mismo "tratamiento heurístico". Más allá de estas notas aisladas que deben tenerse en cuenta al momento de la migración, intentaré explicar un mecanismo homogéneo, un patrón común, de transformación que sirva como base para todo PDF.

PDF. Conversión

La gran mayoría de los textos que se han cargado en SEDICI en los últimos dos o tres años (desde 2014 a la fecha, Junio 2016) están en un PDF que posee los caracteres del texto en su contenido. Es decir, es un PDF construido a partir de una imagen. Es muy sencillo darse cuenta cuando un PDF está basado en imágenes o en texto. Simplemente se pasa el cursor del mouse intentando seleccionar el texto en cuestión. Si no se puede "pintar", el texto es una imagen. Los archivos PDF que se van a transformar a EPUB deben tener una capa de texto dedicada, por varios motivos. Primero, el peso del archivo será mayor ya que está compuesto exclusivamente de imágenes. Segundo, un PDF hecho a base de fotos se vuelve inobservable en dispositivos muy pequeños. Tercero, las imágenes que incluyen texto son (valga la redundancia) imágenes-no-textos. Si no tenemos la imagen, el texto desaparece. Al no haber texto (situación importante) no entra dentro de los parámetros de los motores de búsqueda de google o cualquier otro buscador. Pierde visibilidad y posibilidad de ser encontrado. Este es un gran problema. De allí que sea importante que el PDF tenga texto y no imagen de un texto.

Hay dos maneras de pasar la imagen de texto del PDF a texto concreto. Por medio de dos programas. El primero se llama Adobe Acrobate Pro DC. Allí se carga el PDF y sencillamente se cliquéa en "Reconocer texto", "En este achivo". El lector del programa hará un chequeo de las letras y se transforma la imagen del texto en texto de manera directa. Un segundo programa, se llama ABBY Fine Reader 11. Este es un programa más específico y con más funciones para el OCR (el reconocimiento óptico de caracteres. También se puede guardar una copia del PDF en copia EPUB; al guardarlo se abre en Calibre. Sin embargo, la transformación no es muy eficiente. Siguen manteniéndose los mismo problemas de transformación que cuando se utiliza Calibre. Habría que analizar cuál de los dos mecanismos es el más óptimo para el traspaso. En principio, el Calibre da mejores opciones. Haré algunos comentarios posteriores sobre esta cuestión)

La transformación inicial se realiza a través de Calibre. Importante tener en cuenta el Manual que Julieta hizo sobre la transformación de PDF a Epub por medio del Calibre y luego el tratamiento en el Sigil. Voy a intentar realizar otros aportes y no repetir lo que ella ya ha dicho (los manuales se ven aquí:

http://trac.prebi.unlp.edu.ar/projects/libros-digitales/wiki/Manual_de_Sigil_y_Calibre_para_la_creaci%C3%B3n_de_Epub2) Ella ofrece aquí los comentarios acerca de los parámetros de transformación recomendados. Punto por punto.

Para transformar el PDF uso Calibre. En la Solapa **Metadatos** se ponen los datos del archivo: nombre, apellido, editorial, toda la información que se pueda poner sobre el archivo. Esto no demuestra complicación alguna. También en esta solapa se regular qué imagen queremos poner de portada en el Epub. Permite la opción de poner una Imagen de Portada.

De manera predeterminada, Calibre pone como Portada la primera hoja del PDF. Esto puede ser útil, ya que puede ser que la primera hoja del PDF sea la de la Portada. Pero puede ser que no. Habrá que estar atento si hay que incorporar una Portada o no. Generalmente todo PDF tiene una Portada así que se usará la primera hoja como tal.

En **Apariencia** se puede manipular el tamaño de la letra del texto en general. Para hacer que la letra del PDF sea mayor, se agrega más tamaño a la letra desde acá. Siempre vamos a querer (en principio) que se mantenga la misma letra original. Así que no vamos a modificar esta disposición. El texto debe estar justificado, así que ponemos en Justificación del texto “justificar”. El Epub estará más prolijo. Dentro de esta solapa también se nos da la opción de incrustar un tipo de letra externo al archivo. No necesitamos incrustrar ningún estilo en el PDF, por lo que no se marcará. También se elimina espaciados entre las líneas o se incluyen espacios entre los párrafos. Estas opciones no funcionarán, nos dice Calibre, si los archivos no tiene en su HTML el anuncio de salto de línea o de conclusión de párrafo como <p> o como .

Como dije, no hay que marcarlos porque sin hacerlo, en la transformación al Epub, se mantienen (en principio) las distancias entre los párrafos. Tampoco tiene que incluirse el “trasliterar los caracteres unicode a ASCII”. Unicode y ASCII son estándares, patrones que incluyen en su construcción a través de ceros y uno, todos los símbolos que pueda construir el lenguaje en el cuál el usuario entiende el mundo. No hay que modificar los caracteres de un sistema a otro puesto que, en general, los archivos en idioma castellano utilizan el código unicode y no el ASCII. Esto que expongo aquí es lo que humildemente entiendo del asunto. Para profundizar este tema hay que preguntarle a un informático que ande cerca. Lo importante es que no hay que cliquearlo.

Expresiones Regulares. Principales Problemas

Voy a intentar hacer una aclaración sumamente importante a la hora de transformar archivos PDF en archivos Epub. Vamos a tratar de explicar el tema de las “expresiones regulares”. Tema que en principio parecería ser sencillo, pero que no lo es. Una expresión regular es un conjunto de signos que se irán repitiendo a lo largo de un texto. Cuando digo una serie de “signos”, quiero decir que un espacio es un signo, que un acento es un signo, que un signo de exclamación es un signo, que una letra es un signo. Una expresión regular será entonces, todo conjunto de manifestación de signos que se muestren (a lo largo de un

texto) de manera regular. Si, por ejemplo, la expresión regular es “Hola, Doña Rosa”. La expresión regular será esa misma y no otra semejante pero no igual. Es decir, la expresión regular “Hola, Doña Rosa” no será igual a la expresión regular “hola, doña rosa”. Allí cambian tres mayúsculas por tres minúsculas y la expresión regular cambiará. Tampoco será lo mismo que “Hola Doña Rosa”. Acá, si usted observa bien, no hay una coma. Por lo tanto, la expresión regular ésta (ésta última) no será la misma que la primera que hemos planteado, que sí incluía una coma.

Entonces, ¿Para qué nos sirve las expresiones regulares? Nos servirá para poder Buscar y Sustituir, dentro del archivo del Epub, dentro de Sigil; decía, nos servirá para Buscar y Sustituir los espacios y los número de página que indefectiblemente (muchas veces, en muchos libros) se encontrarán en los archivos Epub. Al realizar la transformación de PDF a Epub, la migración puede ser muy buena, asimilándose mucho con la versión PDF. Pero, sin embargo, toda la enumeración de las hojas (número 1, 2,3, número 54, etc) esa enumeración se nos va a filtrar (muchas veces, no siempre) en el texto en cuestión, en el texto original.

¿Cómo hacemos para poder eliminar toda la enumeración de las páginas con la opción de Buscar y Sustituir, cuando la enumeración cambia, lógicamente, de página en pa ǵ ina? ¿Cómo hacemos para eliminar los espacios en blanco que se puedan generar entre línea y línea y que, entre medio de cada una de las línes (esto que usted lee es una línea), entre línea y línea, decía, está el número de página y en la línea siguiente, un espacio en blanco? No sé si me explico, pero intentará ser más claro.

La expresión regular puede expresar expresiones que incluyan más de una línea, puede incluir expresiones que incluyan distintos números de hoja (es decir, a la expresión regular se le indicará que toma todas las unidades -del 1 al 9 por ejemplo- también que tome todas las decenas-del 1 al 9 por ejemplo- y también todas las centenas -del 1 al 9 por ejemplo.

Herramientas de expresión

[0-9] + La expresión establece “tomáme todos los números estos (es decir, desde el cero hasta el nueve, cualquiera de ellos números). El + (el más) significa tomáme 1 o más números. Es decir, tomar el número 7 (porque está dentro, entre el cero y el nueve) o también te puede tomar el número 156 (porque aquí hay tres números, unidad, decena y centena, pero cada uno de ellos (sus dígitos) está incluido entre el cero y el nueve que establecimos la primera vez. El +, entonces, significa “uno o más” en este caso, números entre el cero y el nueve. Por lo que, el número 1245 también se incluye dentro de esta categoría. El número 98345 también.

< \ / p > La barra invertida tiene otro significado. Dentro de un html, como puede ser el </p> de una etiqueta, la barra invertida dentro de </p> significa “tomámelo literalmente”. Es decir, tomámelo de manera tal de entenderlo como realmente se expresa, en este caso una una barra norma (/). La barra invertida indica que “lo que está próximo, contigua a ella, debe

tomarse literalmente" ¿Por qué se realiza esto? Porque la barra normal (/) tiene otro significado en las expresiones regulares. Y como no queremos darle ninguna orden específica a esa barra, como queremos que busque esa barra y no ejecutar una orden, por eso ponemos la barra invertida (\ ) antes de la barra. Para indicarle que la busque realmente, no para ejecutar una función.

( ) * Los dos paréntesis marcan que busque una espacio vacío. Como se puede ver en ese símbolo que acabamos de poner (la apertura del paréntesis, un espacio, el cierre del paréntesis y luego el asterisco), acabamos de decir en la expresión regular que busque "un espacio", un vacío en el texto de letra. Con el asterisco próximo a él, estamos diciendo *"haya ningún espacio - es decir, cero espacio- o más espacios*" El asterisco * vendría a hacer comodín. Es decir, el asterisco está diciendo "buscá si hay un espacio o muchos espacios". Y si no hay ningún espacio, también buscalo. Dentro de los paréntesis, en este acaso hay un vacío. Pero podemos incluir otra cosa, otro signo dentro de los paréntesis. Por ejemplo, le podemos decir a la expresión regular que busque puntos (.). Si en la expresión ponemos **(.) ***, estaremos indicando que busque un punto, muchos puntos o inclusive ningún punto. Porque, recordemos, el ***** significa, "cero o más". De nuevo, el asterisco significa que busque "o ningún punto, o un punto o más de un punto". El asterisco marca que, inclusive aunque no haya punto, inclusive aunque no encuentre ninguno, también busque esa expresión. No habrá punto, pero el programa lo buscará igual. Porque el * significa "cero o más". Si el **+** supone "uno o más", el ***** significa "cero o más".

Para hacer un ejercicio. Si ponemos para buscar **[0-9] + (.) *** , vamos a estar diciendo al Buscador que me busque todos los números entre uno y nueve (uno o más números, porque hemos puesto el signo **+** . El signo más significa, otra vez lo decimos, "uno o más números. Entonces la expresión buscará tanto el número 5 como el número 87393) y que seguido a esto, además de encontrar ese número o números, el número tenga que estar seguido sí o sí por un punto, varios puntos e inclusive por ningún punto. Entonces, la expresión buscará tanto el número 3 seguido de un punto (buscará el 3.) como también el número 893634…. (es decir, el número seguido de, en este caso, varios puntos) Pero también buscará el número 57 (y, como vemos acá, sin el punto. Porque, ya lo hemos dicho, pero lo volvemos a decir, el * buscará un punto, muchos puntos e inclusive ningún punto. En este caso no hay ningún punto. Y aún así lo busca y lo encuentra.

\n La expresión significa salto de línea. Esto es expresión de HTML. Demuestra un salto de línea. Dentro de la expresión regular entonces podemos incluir un salto de línea. Esto significa \n.

Nota a tener en cuenta: Puede ser que usted ponga una expresión regular y aún así no se encuentre. Aunque cree que cumple todos los requisitos que corresponden a la expresión que intenta buscar y sustituir, puede ser que esto no ocurra. Puede ser que, a pesar de todas las similitudes, no se encuentre la expresión que quiere buscar. Esto sólo significa una cosa: que la expresión que usted puso no es la misma que la que quiere buscar, aunque parezca exactamente la misma. Es importante saber que un solo espacio de más o un solo espacio de menos hace la diferencia. Un solo espacio que usted no

considera. Un solo espacio que se la pasa ver, la expresión no será la misma. ¿Cómo hacer entonces? Es sencillo.

Recordemos que la herramienta * (asterísco) tiene la función de que incluye “cero o más opciones” Esto es, puede ser que lo que requiera esté, como puede ser que no esté. El asterisco funciona de comodín. Entonces, cada vez que tiene que poner/incluir/ o sabe que puede haber uno o más espacios entre especificación y especificación, usted tiene que utilizar el asterisco. Por ejemplo, si hay una distancia de uno o de dos espacios en blanco (sí, espacios en blanco.. : . Eso son dos espacios en blanco. Ese espacio pequeño que quedó entre los dos puntos y el punto anterior, son dos espacios en blanco) puede hacer referencia a este/o estos espacios en blanco con la referencia **( )*** Acá estará diciendo, dentro de la expresión regular, “que esté seguido de uno o varios espacios o inclusive ningún espacio” Entonces, acá se incluirá a todos los lugares donde haya uno o más espacios o ningún espacio dentro de la expresión regular que está buscando.

Nota al pie. ¿Cómo realizarlas?

Otras de las cuestiones claves es hacer notas al pie. ¿Cómo realizarlas? Los libros tiene muchas notas al pie, así que éste será un problema que obligatoriamente va a surgir. Primer problema: en la transformación al Epub, las notas al pie se van a mostrar mezcladas a lo largo del texto. No he encontrado manera todavía de poder discriminar una nota al pie de lo que es el texto del cuerpo principal. ¿Y entonces ? Una sola acción es posible: ir una nota al pie por una nota al pie. El primer paso es buscarla, directamente yendo párrafo por párrafo. Es la única opción por el momento. Lo que debe hacerse es dividir la pantalla (el monitor) en dos. Una parte abro el Sigil con el epub. La otra parte de la pantalla abro el PDF con el libro. Hay que ir entonces hoja por hoja viendo dónde están las notas al pie y buscándolas respectivamente en el Sigil.

La encontrás. La seleccionas y la mandás a la última parte del HTML. Es decir, la copias y la pegas dentro del HTML, pero al final de todo. Cuando termina el HTML. Allí ponemos la nota al pie. Y luego es sencilla. Pintamos toda la nota al pie y cliqueamos en la “ancla” que está en una de las opciones visuales que da el Sigil. Arriba en las Herramientas (centro a la derecha) hay un Ancla de mar. Allí, esa ancla, da la posibilidad de “anclar” la nota para tomarla como referencia. Se cliquea el Ancla y el programa te pedirá poner una expresión. Una referencia. Una nota, una oración que nombre a esa ancla. Generalmente pongo “nota1” y así sucesivamente… (“nota 2”; “nota 3”; “nota 4”).

Uno hace esto y verá que no pasa nada. Nada. Solamente usted a “anclado” la nota al pie. Pero no se ha modificado nada. Ahora, una vez hecho esto, pasa directamente a buscar, a ojo (literalmente a ojo, así que prepare bien la vista, sáquele punta y empiece a buscar) el número que corresponde con la nota al pie. Si la nota al pie es la primera, entonces tendrá el número 1 (claramente pensará usted. Mire las cosas que le estoy aclarando). Entonces busca el número 1 y aquí aprieta otro botón, otro ícono, que está ubicado muy cerca del botón del “Ancla”, que es el botón de Enlace. A la vista son dos ganchitos enganchados, que muestra justamente el enlace entre dos cosas. Uno, entonces,

pinta el número en cuestión (en este caso el 1) y aprieta el botón de enlace. Aparecerá una pantalla mostrando qué cosa quiere enlazar con el número 1 que acaba de colorear. Sencillamente busca el Ancla que acaba de crear hace instantes, es decir, busca la “nota 1” y aprieta OK. Hemos concluido entonces la nota al pie. El número en cuestión se pintará de azul, quedará conformado como hipervínculo y directamente nos llevará, al hacerle click, hacie el ancla correspondiente. Hemos creado entonces, una nota al pie.

Nueva forma de conversión. Utilizar el programa Mobile Pocket

Este programa se abre por medio de una software virtual de Microsoft, dado que el programa funciona para Windows y no para Linux. Se abre el Virtual Box, se abre Windows. Al entrar al programa Mobile Pocket se realiza la transformación. Esa transformación se realiza desde un PDF a HTML. No a Epub. SI bien es un convertidor con código muy viejo, ya en desuso, la deconstrucción del PDF se realiza exitósamente.

El Mobile Pocket separa el PDF y guarda por un lado las imágenes en PNG, guarda por otro lado el código HTML y tal vez lo más importante, reconoce las notas al pie del texto y también elimina los encabezados que se repiten como también la numeración de página. Esto lo he podido comprobar cuando, al traer el html al Linux y desde allí al Calibre, uno puede hacer la conversión desde el ZIP al Epub, por medio del Calibre.

Una vez convertido, abre el EPUB en el Sigil y me encuentro que la división de los párrafos es correcta, que la enumeración de páginas se eliminó como también el encabezado que se repite en todas las páginas. Quizá lo mejor es que todas las notas al pie están reconocidas y se marcan en rojo. Esto es muy bueno puesto que, si el PDF tiene muchas notas al pie, uno no tiene que estar mirando en detalle todas las notas al pie. Ahora se reconocen rápidamente ya que están marcadas en rojo. Hay que tener en cuenta, sin embargo, que el programa está muy desactualizado, por lo que su estructura interna de códigos podría ser un obstáculo a la hora de manipular el archivo en el programa de Sigil.

Validación de los Epubs. Errores frecuentes.

Una forma de validar los Epubs,es decir una forma de corroborar que el .epub está bien construído sin ningún tipo de error, es por medio de la página web http://validator.idpf.org/

Allí se carga el archivo epub dentro y se acepta y espera a que te de los resultados.

- **“Error while parsing the div tag not allowed here”:** el error me está diciendo que en ese lugar, en esa línea, no se permite una etiqueta div (que es una etiqueta de división, que parcela, que divide cosas) porque justamente está dentro de un H2 (que es un título) Si una etiqueta DIV está dentro de un H2 (que es la etiqueta de un título) no tiene sentido realmente. Un título no se divide en nada. Es un título nada

más. Por lo que, el problema a resolver es eliminando la etiqueta DIV de la línea. Lo que hice fue eliminar la etiqueta div (y su consecuente etiqueta </DIV>)

- **Error schema not satisfied : attribute 'lang' is not declared for element 'html'**. El atributo “lang” viene del inglés “language”. Por lo que la etiqueta determina el lenguaje del cual estará compuesto el **HTML**[1]. El “lang” especifica entonces el lenguaje primario del html.

- **Error schema not satisfied : missing required attribute 'alt'** el problema aquí está diciendo que falta el atributo “alt” a las imágenes adheridas al epub. Esto quiere decir que hace falta una explicación con palabras de lo que se refiere la imagen. Así se da información sobre la imagen cuando las personas no puedan verla.

- **Error schema not satisfied : no character data is allowed by content model**: muchas veces puede pasar que una texto (es decir, oraciones escritas, oraciones de líneas y líneas de texto, quede por fuera de alguna etiqueta de <p>, o de o de . Estas tres etiquetas, dentro de la etiqueta <body> permiten texto. No hay otra etiqueta que permita texto. Por lo que, cuando los caracteres quedan por fuera de estas etiquetas, se marca este error que estamos mencionando: “no se permite información de caracteres (character data) en el modelo de contenido (que vendría a ser el body). El html se forma por un Head por un lado, y por un Body por otro. El texto escrito no puede estar dentro de Body sin ninguna etiqueta de , o de <p> o de . Solo esas tres etiquetas permiten texto. Si el texto no está dentro de estas tres etiquetas, se nos mostrará este error.

Una vez validado el Epub 2, pasamos a validarlo a Epub 3. Es muy sencillo. Ya el Sigil tiene instalada un Plugin que permite la validación, por lo que se va a Puglind, Output, Epub 3 Itizer. Se ubica el lugar donde se quiera guardar y se finaliza.

[1] Hay que aclarar otro concepto tal vez esclarecedor en este punto. La diferencia entre **HTML y XML**. El XML (extensive markup language) es un lenguaje concebido para describir información. Es un formato que ayuda a organizar contenidos y eso hace que los documentos XML sean portables hacia diferentes aplicaciones. El HTML (Hiper Tex markup language) muestra información (no describe) determina cómo actúa y qué hace. *El HTML define particularidades dentro del XML. El XML es una estructura genérica, infinita. EL HTML describe la información para el navegador. Es un caso particular de XML*. Es importante aclarar que el HTML es un lenguaje (es decir, se ponen etiquetas para poder expresar un texto); en tanto el XML es un metalenguaje (es decir, un lenguaje del lenguaje. Por eso, si hay que escribir “Cervantes”, el XML será “Autor”. “Autor” representa un lenguaje del lenguaje “Cervantes” y lo especifica y lo encasilla. Hace también más fácil la búsqueda de la información, obviamente, porque recorta el lenguaje y un lenguaje mayor, en este caso “los autores”)