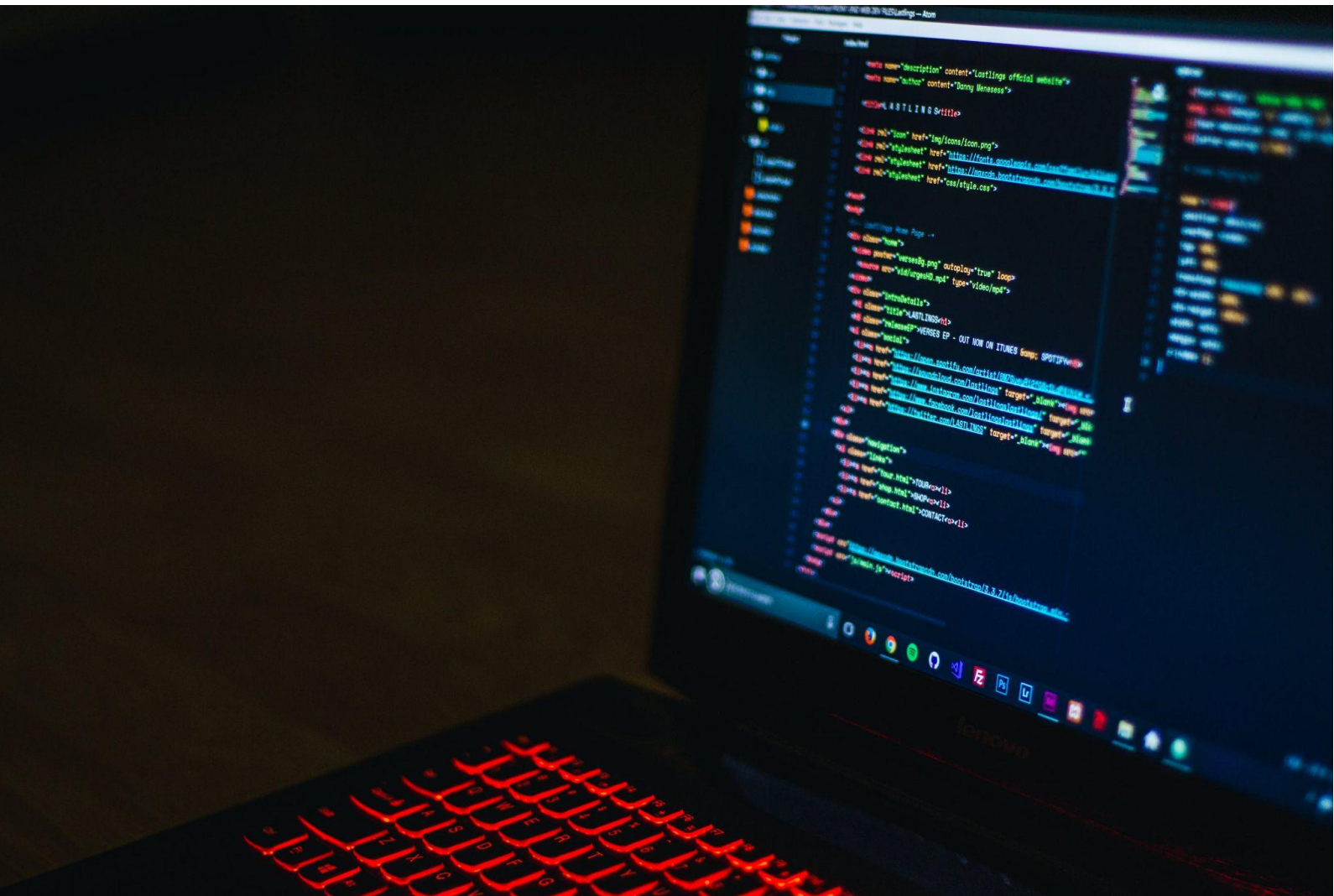


# Actividad 3. LMySGI

*XML: Xpath y XSL*



# Integrantes del equipo

Cada uno hemos creado nuestro propio XML con sus transformaciones Xpath y XSL:

- **Ana:** <https://github.com/AnaArias12/>
- **Nagib:** <https://github.com/DeiMoS-es/LM3>
- **Luis:** <https://github.com/luisc7/LM3>
- **Pedro:** <https://github.com/PedroMurilloArnau/LM-ITE>
- **Rubén:** <https://github.com/rdepag/>

Hemos utilizado git con repositorios remotos alojados en la plataforma Github, siendo cada repositorio personal, pues se elegiría de entre ellos el que nos resultara más correcto.

# Luis

## Comentarios

Para realizar mi parte, he ido subiendo los cambios a un repositorio propio:

<https://github.com/luisc7/LM3>

En cuanto al **requerimiento 1**, he empezado pensando en **cómo estructurar el contenido de una web en HTML** que crearía a partir de las transformaciones Xpath, y así poder ir creando las transformaciones correspondientes.

Para esta web, estructuro todos los requisitos en una sola página, con este esquema:

- **Cabecera:** Contiene el logo y el nombre.
- **Menú de navegación:** Distintas opciones y enlaces a otras partes de la web (no se han creado otras páginas).
- **Sección principal:** Aquí se incluye todo el contenido de la página, incluyendo las diversas tablas y formulario que se pide. La divido en varios **artículos**, uno primero más **introductorio** (con imagen de fondo y texto superpuesto), otros **dos** cada uno **con su tabla correspondiente**, y un último artículo con un **formulario**.
- **Pie de página:** Una imagen que hace de pie de página.

La realización de **los estilos CSS** se realizan al final, cuando se tiene terminada la estructura HTML (excepto aquellos cambios encaminados a estructurar la visualización, como la inclusión de clases, id y algún div).

Para la cabecera:

```
<header id="cabecera">
  
  <h1>
    <xsl:element name="a">
      <xsl:attribute name="href">
        <xsl:value-of select="ite/@web"/>
      </xsl:attribute>
      <xsl:value-of select="ite/@nombre"/>
    </xsl:element>
  </h1>
</header>
```

*Extracción para la cabecera desde el XML del enlace a la web y del nombre.*

Para el menú se ha reciclado parte de la estructura de la actividad 1, y así permitir hacerlo desplegable desde un botón de menú en el caso del *responsive* con anchos de pantalla pequeños:

```
<nav class="collapsible-menu">
  <input type="checkbox" id="menu"/>
  <label for="menu"></label>
  <div class="menu-content">
    <ul class="lista-menu">
      <li class="menulink" id="home"> <a href="index.html">Sobre Edix</a> </li>
      <li class="menulink" id="cursos"> <a href="cursos.html" id="actual">Cursos</a> </li>
      <li class="menulink" id="fp"> <a href="fp.html">Formación Profesional</a> </li>
      <li class="menulink" id="metodo"> <a href="metodo.html">Metodología</a> </li>
      <li class="menulink" id="empresas"> <a href="empresas.html">Empresas colaboradoras</a> </li>
    </ul>
  </div>
</nav>
```

*Detalle del menú de navegación.*

En la parte **section** se comienza con un título h2 y un **article** con pequeña introducción.

En el siguiente **article** se incluye **una tabla** con cabecera y cuerpo, donde los distintos ciclos que se encuentran en el XML se incorporan al cuerpo de la tabla. Al ser varios, se utiliza **xsl:for-each** para recorrer todos ellos en el tbody de la tabla:

```
<article class="cfgs descripcion">
  <h3>Además, ahora podrás hacerlo cursando titulaciones oficiales en CFGS de la rama informática</h3>
  <table>
    <thead>
      <tr>
        <th class="colh1">Denominación</th>
        <th class="colh2">Nombre</th>
        <th class="colh3">Tipo de grado</th>
        <th class="colh4">Año de creación</th>
      </tr>
    </thead>
    <tbody>
      <xsl:for-each select="ite/ciclos/ciclo">
        <tr>
          <td class="col1"><xsl:value-of select="@id"/></td>
          <td class="col2"><xsl:value-of select="nombre"/></td>
          <td class="col3"><xsl:value-of select="grado"/></td>
          <td class="col4"><xsl:value-of select="decretoTitulo/@año"/></td>
        </tr>
      </xsl:for-each>
    </tbody>
  </table>
</article>
```

*Detalle de la tabla de los ciclos.*

Para el **article** posterior se incluyen todas las personas que aparecen en el XML. Para los profesores, se realiza un **xsl:for-each** para nombrarlos a todos dentro del texto.

Para el director y jefe de estudios, se crea **otra tabla** denominada **equipo directivo** con cabecera y cuerpo, donde los distintos elementos que se encuentran

en el XML se incorporan al cuerpo de la tabla. Al ser varios, se utiliza **xsl:for-each** para recorrer todos ellos en el tbody de la tabla:

```
<article class="descripcion equipo">
  <h3>Nuestro equipo</h3>
  <p>Contarás la variada experiencia de nuestro equipo para tu formación:</p>
  <p>Serán<xsl:for-each select="ite/profesores/profesor">
    o <span class="negrita"><xsl:value-of select="nombre"/></span>
  </xsl:for-each> los que impartirán las clases de contenido técnico.</p>
  <table>
    <caption><h4>Equipo directivo</h4></caption>
    <thead>
      <tr>
        <th>Cargo</th>
        <th>Nombre</th>
        <th>Despacho</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>Director/a</td>
        <td><xsl:value-of select="ite/director/nombre"/></td>
        <td><xsl:value-of select="ite/director/despacho"/></td>
      </tr>
      <tr>
        <td>Jefe de estudios</td>
        <td><xsl:value-of select="ite/jefe_estudios/nombre"/></td>
        <td><xsl:value-of select="ite/jefe_estudios/despacho"/></td>
      </tr>
    </tbody>
  </table>
</article>
```

*Detalle del article con los dos for-each para menciones dentro del texto y en la tabla.*

En el último **article** se incluye un **formulario** de contacto.

De nuevo, se utiliza **xsl:for-each** para recorrer todos los elementos del XML para rellenar las distintas opciones de un menú desplegable select, tanto para el nombre mostrado como para el nombre enviado en el envío del formulario, al que además se le incorpora una opción adicional llamada “otros”.

El resto de elementos se incorporan con uso de html5, con placeholder para mostrar un texto en la casilla vacía (para informar de si es opcional, por ejemplo).

Además, aparece un div para estructurarlo con CSS mediante un grid.

```

<article class="descripcion formulario">
  <h3>Solicita más información</h3>
  <p>Si deseas inscribirte o solicitar más información, rellena tus datos a continuación y estaremos encantados de resolver tus dudas.</p>
  <form method="get" action="mailto:midireccion@decorreo.es">
    <fieldset><legend>¡Pregúntanos!</legend>
      <div class="grid-container">
        <label for="curso">Elige de qué curso deseas informarte</label>
        <select name="curso">
          <xsl:for-each select="ite/ciclos/ciclo">
            <xsl:element name="option">
              <xsl:attribute name="value">
                <xsl:value-of select="@id"/>
              </xsl:attribute>
              <xsl:value-of select="nombre"/>
            </xsl:element>
          </xsl:for-each>
          <option value="otros" selected="selected">Otros (preguntas generales)</option>
        </select>
        <label for="detalles">Danos más detalles </label>
        <input type="text" name="detalles" placeholder="(Opcional)"/>
        <label for="email">Correo electrónico de contacto</label>
        <input type="email" name="correo" placeholder="Tu email" required="required"/>
        <label for="telf">Teléfono de contacto</label>
        <input type="tel" name="telf" placeholder="(Opcional)"/>
        <div id="blank-space"/>
        <input class="button" type="submit" value="Solicitar información"/>
      </div>
    </fieldset>
  </form>
</article>

```

*Detalle del formulario.*

Para los **estilos CSS**, me baso en los colores predominantes que se pueden ver en la web de Edix, empleando principalmente el **morado #6a00c2**, el **cian #00c4da** y el **azul oscuro #032258** para algún resalte.

Comienzo realizando los estilos para un **ancho fijo de 900px** (que será para anchos mayores a este en el *responsive*), **reutilizando parte** de lo hecho en la actividad 1, en concreto para el **menú de navegación**, pues permitía convertirse en menú desplegable, como así resulta al incorporar un **segundo CSS para anchos de 900px o menores**.

Se eligen varios estilos para mostrar la cabecera de las **tablas** con estilo diferente al del cuerpo. En el *responsive* nos permitirá intercambiar columnas con filas en el caso de visualización de pantallas estrechas, pues si no las tablas ocupa mucho a lo ancho.

```

243  /* Intercambiar filas y columnas de tablas */
244  /* basado en: https://stackoverflow.com/questions/23556364/
    how-to-convert-columns-to-rows-using-css/23556911 */
245  table tr > *{
246      display: block;
247  }
248  table tr,
249  table thead,
250  table tbody {
251      display: table-cell;
252  }

```

*Detalle del CSS que intercambia filas y columnas en el CSS de ancho <900px.*

En el **formulario** se opta por generar **un div** donde se encierran las etiquetas y sus campos, y estructurarlo con **un grid de dos columnas en >900px** y en **una sola columna en anchos menores**.

The image shows two versions of a web form titled "¡Pregúntanos!". The left version is in a wide layout (two columns), and the right version is in a narrow layout (one column). Both forms have the following fields:

- Elige de qué curso deseas informarte:** A dropdown menu with the option "Otros (preguntas generales)".
- Danos más detalles:** A text input field with a placeholder "(Opcional)".
- Correo electrónico de contacto:** A text input field with a placeholder "Tu email".
- Teléfono de contacto:** A text input field with a placeholder "(Opcional)".
- Solicitar información:** A purple button at the bottom.

*Efecto del cambio de dos columnas a una en el grid del CSS para el formulario.*

Además, se incluyen algunos **efectos en el CSS**, como al pasar el puntero por encima (**hover**) de los elementos del menú, del formulario, el botón del envío o el resaltado de la casilla activa en los campos del formulario.

Como **problemática principal** que he encontrado al realizar este requerimiento, es que **la transformación** al html de la página **no se producía si existía algún fallo** en el código (a diferencia de html), por lo que **ha requerido ir haciendo y revisando poco a poco** que se creaba el documento sin fallos, y cuando no se creaba el html se fue **acotando con comentarios** las zonas para **localizar con más facilidad el origen de los problemas**.

Otra **dificultad** ha sido **algún atributo**, como **required** en los **input** del **formulario**, que ha habido que declararlos como **required="required"** o no se generaba el html.



Para realizar la transformación he utilizado el **plugin Live Server** de Visual Studio Code y visualizando en el **navegador Edge** (y a veces en Chrome).

En cuanto al **requerimiento 2**, se ha optado por transformar el XML en otro diferente con transformaciones XSL. Al igual que el requerimiento 1, una de las **dificultades** ha sido la **visualización/comprobación** de que se estaba efectuando correctamente la **transformación**.

Para sortearlo, he encontrado útil emplear las **herramientas para desarrolladores**, que muestran el resultado de la transformación cuando es satisfactoria, o dejan de mostrarla si ha habido algún problema.

En el caso de **esta transformación**, he decidido que haya **etiquetas con un solo contenido**, sin anidar nada dentro excepto en la principal, relegando el **resto de información en forma de atributos** de estas etiquetas.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<!-- ?xml-stylesheet type="text/xsl" href="ITE.xsl"? -->
<!-- ?xml-stylesheet type="text/xsl" href="XML_ITE.xsl"? -->
<ite nombre="Instituto Tecnológico Edix" web="https://institutotecnologico.edix.com">
  <empresa>Proeduca</empresa>
  <telefono>+34 91 787 39 91</telefono>
  <profesores>
    <profesor>
      <id1</id>
      <nombre>Félix</nombre>
    </profesor>
    <profesor>
      <id2</id>
      <nombre>Raúl</nombre>
    </profesor>
    <profesor>
      <id3</id>
      <nombre>Raquel</nombre>
    </profesor>
    <profesor>
      <id4</id>
      <nombre>Tomás</nombre>
    </profesor>
  </profesores>
  <director>
    <nombre>Chon</nombre>
    <despacho>Numero 31, 3ª Planta, Edificio A</despacho>
  </director>
  <jefe_estudios>
    <nombre>Dani</nombre>
    <despacho>Numero 27, 2ª Planta, Edificio B</despacho>
  </jefe_estudios>
  <ciclos>
    <ciclo id="ASIR">
      <nombre>Administración de Sistemas Informáticos en Red</nombre>
      <grado>Superior</grado>
      <decretoTitulo año="2009"/>
    </ciclo>
    <ciclo id="DAW">
      <nombre>Desarrollo de Aplicaciones Web</nombre>
      <grado>Superior</grado>
      <decretoTitulo año="2010"/>
    </ciclo>
    <ciclo id="DAW">
      <nombre>Desarrollo de Aplicaciones Multiplataforma</nombre>
      <grado>Superior</grado>
      <decretoTitulo año="2010"/>
    </ciclo>
  </ciclos>
</ite>
```

```
<centro codcentro="ite">
  <nombre empresa="Proeduca">Instituto Tecnológico Edix</nombre>
  <web>https://institutotecnologico.edix.com</web>
  <telf>+34 91 787 39 91</telf>
  <profesor id="1">Félix</profesor>
  <profesor id="2">Raúl</profesor>
  <profesor id="3">Raquel</profesor>
  <profesor id="4">Tomás</profesor>
  <director despacho="Numero 31, 3ª Planta, Edificio A">Chon</director>
  <jefe_estudios despacho="Numero 27, 2ª Planta, Edificio B">Dani</jefe_estudios>
  <ciclo nombre="Administración de Sistemas Informáticos en Red" grado="Superior" anno_decreto="2009">ASIR</ciclo>
  <ciclo nombre="Desarrollo de Aplicaciones Web" grado="Superior" anno_decreto="2010">DAW</ciclo>
  <ciclo nombre="Desarrollo de Aplicaciones Multiplataforma" grado="Superior" anno_decreto="2010">DAW</ciclo>
</centro>
```

XML original a la izquierda, y el resultante de la transformación a la derecha.

Como se han introducido bastantes elementos o atributos como atributos de nuevas etiquetas, además del for-each, se ha hecho un **uso intensivo de xsl:attribute** para ir colocando como atributos información de elementos o atributos en las nuevas etiquetas.

```
<xsl:for-each select="ite/ciclos/ciclo">
  <xsl:element name="ciclo">
    <xsl:attribute name="nombre">
      <xsl:value-of select="nombre"/>
    </xsl:attribute>
    <xsl:attribute name="grado">
      <xsl:value-of select="grado"/>
    </xsl:attribute>
    <xsl:attribute name="anno_decreto">
      <xsl:value-of select="decretoTitulo/@año"/>
    </xsl:attribute>
    <xsl:value-of select="@id"/>
  </xsl:element>
</xsl:for-each>
```

Extracto de la transformación de los ciclos, asignando información con tres atributos y un solo dato en el elemento.



Además, el **uso de *for-each*** también **ha sido recurrente**, pues ha sido utilizado para extraer la información **recorriendo todos los profesores** y, como ya se ha visto, recorriendo **todos los ciclos**.

*Extracto con la transformación de los profesores, asignando la id a un atributo y el nombre al elemento.*

```
<xsl:for-each select="ite/profesores/profesor">
  <xsl:element name="profesor">
    <xsl:attribute name="id">
      <xsl:value-of select="id"/>
    </xsl:attribute>
    <xsl:value-of select="nombre"/>
  </xsl:element>
</xsl:for-each>
```

La **etiqueta principal** ha sido cambiada por otra denominada **centro**, de donde cuelgan las demás de forma directa, y **se le ha asignado un atributo con el nombre de la etiqueta principal del anterior XML (ite)**.

Desde ella **descienden** las etiquetas **nombre** (con atributo de la empresa), **web**, **telf** (en vez de telefono), **profesor** (con tantas etiquetas como profesores) y **ciclo** (se repite igualmente tantas veces como ciclos hay).

```
<xsl:template match="/">
  <xsl:element name="centro">
    <xsl:attribute name="codcentro">
      <xsl:value-of select="name(ite)"/>
    </xsl:attribute>
    <xsl:element name="nombre">
      <xsl:attribute name="empresa">
        <xsl:value-of select="ite/empresa"/>
      </xsl:attribute>
      <xsl:value-of select="ite/@nombre"/>
    </xsl:element>
    <xsl:element name="web">
      <xsl:value-of select="ite/@web"/>
    </xsl:element>
    <xsl:element name="telf">
      <xsl:value-of select="ite/telefono"/>
    </xsl:element>
  </xsl:element>
</xsl:template>
```

*Detalle de las primeras etiquetas, con la reestructuración de la información del centro.*

Con estos cambios, la información queda finalmente estructurada de la siguiente manera:

```
<centro codcentro="ite">
  <nombre empresa="Proeduca">Instituto Tecnológico Edix</nombre>
  <web>https://institutotecnologico.edix.com</web>
  <telf>+34 91 787 39 91</telf>
  <profesor id="1">Félix</profesor>
  <profesor id="2">Raúl</profesor>
  <profesor id="3">Raquel</profesor>
  <profesor id="4">Tomás</profesor>
  <director despacho="Numero 31, 3ª Planta, Edificio A">Chon</director>
  <jefe_estudios despacho="Numero 27, 2ª Planta, Edificio B">Dani</jefe_estudios>
  <ciclo nombre="Administración de Sistemas Informáticos en Red" grado="Superior" anno_decreto="2009">ASIR</ciclo>
  <ciclo nombre="Desarrollo de Aplicaciones Web" grado="Superior" anno_decreto="2010">DAW</ciclo>
  <ciclo nombre="Desarrollo de Aplicaciones Multiplataforma" grado="Superior" anno_decreto="2010">DAM</ciclo>
</centro>
```

# Pedro

## Comentarios

Para trabajar en este proyecto he trabajado en un repositorio de Github:

<https://github.com/PedroMurilloArnau/LM-ITE>

En primer lugar hemos creado una XML en el que hacemos una llamada interna hacia **stylesheet** correspondiente para unirlo al **XSL** que dominara la transformación.

Para ello hacemos una llamada a **ite.xsl**.

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="ite.xsl"?>
```

Posteriormente creamos el correspondiente **XSL**. En el primeramente creamos los parámetros de nuestra xsl:stylesheet con la versión con la que vamos a trabajar y posteriormente como ya hemos trabajado con html, en la parte de la cabecera pondremos el **título** y la llamada al **CSS**, ya que lo que queremos conseguir es que tenga un CSS que domine el html de forma externa.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
<head>
<title>Actividad 3</title>
<link rel="stylesheet" type="text/css" href="css/estilos.css" media="screen"/>
</head>
```

Para el **título** y para destacar la importancia de la empresa de la misma lo dominaremos mediante una llamada a XML de la siguiente forma. En la cual el elemento empresa perteneciente a "ite".

```
<h3>
  Perteneciente a la empresa: <xsl:value-of select="ite/empresa"/>
</h3>
<br/>
```

Según lo que se nos pide creamos un par de listas, una lista ordenada y una lista sin ordenar.

Primero monte una **lista ordenada** para nombrar los diferentes ciclos. Para ello he utilizado el atributo **<ol>** y le he agregado el atributo de **type=1** para hacer referencia a crear una lista ordenada con atributos numéricos.

Para llamar a los elementos creo un **for-each** para que recorra cada uno de los datos. Y selecciono de ellos su elemento mediante la fórmula **value-of**.

Es importante cerrar cada **for-each** mediante el **"/**.

```
<ol type="1">
  <xsl:for-each select="ite/ciclos/ciclo">
    <li>
      <xsl:value-of select="identificador"/>
    </li>
  </xsl:for-each>
</ol>
```

Después creamos una lista sin ordenar. Mediante el atributo **<ul>**. En este caso como quiero hacer una lista con una terna de elementos que no son comunes, creo un par de llamadas mediante la formula **value-of**.

```
<ul>
  <li>
    Director: <xsl:value-of select="ite/director/nombre"/>
  </li>
  <li>
    Jefe de estudios: <xsl:value-of select="ite/jefe_estudios/nombre"/>
  </li>
</ul>
<br/>
```

Posteriormente creo un par de tablas. En la primera tengo 2 columnas **<th>** referidas a profesores. Y accedo a los datos con la fórmula anterior recorro los elementos con **for-each** y posteriormente selecciono los datos con **value-of**.

```
<h3>Profesores: </h3>
<table>
  <tr>
    <th style="text-align:center">id:</th>
    <th style="text-align:center">Profesor:</th>
  </tr>
  <xsl:for-each select="ite/profesores/profesor">
    <tr>
      <td>
        <xsl:value-of select="id"/>
      </td>
      <td>
        <xsl:value-of select="nombre"/>
      </td>
    </tr>
  </xsl:for-each>
</table>
```

En la última tabla establezco, de igual forma, las llamadas a los atributos y la distribución de la tabla con 3 columnas.

```
<h3>Ciclos formativos: </h3>
<table>
  <tr>
    <th style="text-align:center">Ciclos:</th>
    <th style="text-align:center">Nombre del ciclo:</th>
    <th style="text-align:center">Tipo de grado:</th>
  </tr>
  <xsl:for-each select="ite/ciclos/ciclo">
    <tr>
      <td>
        <xsl:value-of select="identificador"/>
      </td>
      <td>
        <xsl:value-of select="nombre"/>
      </td>
      <td>
        <xsl:value-of select="grado"/>
      </td>
    </tr>
  </xsl:for-each>
</table>
<br/>
```

De igual manera creo 2 enlaces como se solicita. Creo un par de llamadas a paginas mediante la fórmula **href=""** y las nominalizar.

```
<h3>
  <a href="https://institutotecnologico.edix.com">Contacta con nosotros</a>
</h3>
<h3>
  <a href="https://www.comunidad.madrid/servicios/educacion">Consejeria de educacion</a>
</h3>
```

Finalmente creamos el formulario como se solicita. Mediante una caja de formula. **<fieldset>**, así dominamos todo el formulario dentro de una caja y creamos los diferentes **label** e **input**. Finalmente creamos un **autofocus** al label de email.

```
<fieldset>
  <legend>Formulario de contacto: </legend>
  <label for="email">Rellena tu correo</label>
  <input type="email" autofocus="" placeholder="correo"/>
  <br/>
  <label for="telefono">Telefono:</label>
  <input type="tel" name="telefono" placeholder="telefono"/>
  <br/>
  <textarea rows="7" cols="60" name="Dinos tus dudas respecto a los cursos: "></textarea>
</fieldset>
```

Para el apartado dos he **cambiado de atributos a elementos** los id de ciclos.

```
<ciclo>
  <identificador>DAW</identificador>
```

# Nagib

## Comentarios

Mi repositorio de Git es el siguiente:

<https://github.com/DeiMoS-es/LM3>

Esta actividad la hemos realizado entre Ana y Nagib, ya que ambos teníamos dudas en la parte que ha realizado el otro, y luego hemos puesto en común nuestras ideas, explicado el ejercicio obteniendo el siguiente resultado.

Para el requerimiento uno, el cual lo ha realizado Nagib, he creado el xml en base al enunciado del ejercicio. Una vez creado el XML decidí crear el XSL en un documento a parte, con lo cual en el XML he tenido que hacer la llamada al XSL.

```
2 <?xml-stylesheet type="text/xsl" href="XSLT_Actividad3.xsl"?>
```

Empezando a hacer el XSL, siguiendo lo requerimientos del ejercicio, empezamos dándole un título al documento que en mi caso he decidido llamarlo Profesorado, y le he añadido los estilos.css también en un documento aparte, con lo cual también he tenido que añadir la ruta de donde se encuentran los estilos.

```
<xsl:template match="/">
  <html>
    <head>
      <!-- El titulo de la página va a ser Profesorado ya que mostrará todos los datos de los
profesores-->
      <title>Profesorado</title>
      <link rel="stylesheet" href="css/style.css" />
```

Creo un <div> el cual va a ser el contenedor de toda la página, y dentro empiezo a crear toda la estructura del HTML.

Empezando por el <header> obtenemos los datos del XML como el nombre del instituto, teléfono y página web.

```
<div class="contenedor">
  <!-- Cogemos el nombre del instituto donde se encuentran los profesores-->
  <header>
    <h1>
      <xsl:value-of select="ite/@nombre" />
    </h1>
    <p>
      Nombre de la empresa:
      <xsl:value-of select="ite/empresa" />
      <br />
      Teléfono:
      <xsl:value-of select="ite/telefono" />
      <br />
      Página web:
      <a href="">
        <xsl:value-of select="ite/@web" />
      </a>
    </p>
  </header>
```

Para crear las tablas, creo un <div> el cual va a contener 3 tablas, ya que hago una diferenciación entre profesores, directora y jefe de estudios. Decidí crear tablas aunque las de directora y jefe de estudios solo contienen un elemento, ya que pensé que quizás en un futuro se podrían añadir más miembros a dichos grupos, y era más fácil dejar la estructura ya montada.

```
<div class="tablas">
  <table border="2" id="prof">
    <caption>Profesores</caption>
    <tr>
      <th>ID</th>
      <th>Nombre</th>
    </tr>
    <!-- Como el ejercicio pide un mínimo de dos tablas, agrupo el profesorado por el cargo -->
    <xsl:for-each select="ite/profesores/profesor">
      <!--Con un bucle recorremos todos los profesores-->
      <tr>
        <td>
          <xsl:value-of select="id" />
        </td>
        <td>
          <xsl:value-of select="nombre" />
        </td>
      </tr>
    </xsl:for-each> <!--Cierre for-each de profesor-->
  </table>
```

Como profesores tiene varios elementos, he tenido que usar una estructura for-each para que recorra todos los elementos que hay dentro de profesor.

Para directora y jefe de estudios he seguido la misma estructura que en directora.

```
<table border="2" id="directora">
  <!--En este caso como solo existe una directora, no hace falta hacer un bucle.
  (Se podría dejar programado un bucle, para en el caso de que en un futuro se
  añadiesen varios miembros se actualizase) -->
  <caption>Directora</caption>
  <tr>
    <th>Nombre</th>
    <th>Dirección</th>
  </tr>
  <tr>
    <td>
      <xsl:value-of select="ite/director/nombre" />
    </td>
    <td>
      <xsl:value-of select="ite/director/despacho" />
    </td>
  </tr>
</table>
```

Para las listas, decidí mostrar los distintos títulos como listas una ordenada y otra no ordenada, anidada dentro de la ordenada, y todas a su vez en su propio <div>.

```
<div class="titulos">
  <h3>Ofertas de Títulos</h3>
  <ol>
    <xsl:for-each select="ite/ciclos/ciclo">
      <li>
        <xsl:value-of select="@id" />
      </li>
      <ul>
        <li>
          <xsl:value-of
            select="nombre" />
        </li>
        <li>
          <xsl:value-of select="grado"
            />
        </li>
        <li>
          <xsl:value-of
            select="decretoTitulo/@año" /
          >
        </li>
      </ul>
    </xsl:for-each>
  </ol>
</div>
```



Continuando con el ejercicio, llegamos a la parte del formulario, para el cual decido crear también un <div> el cual contendrá todos los elementos del formulario.

```
<div class="formulario">

  <h1 id="form">Formulario de Contacto</h1>

  <form action="" method="post">

    <div id="personales">
      <div class="datos_1">
        <div class="insert_nombre">
          <label for="nombre">Nombre:</label>
          <input type="text" id="nombre" />
        </div> <!-- Cierre div nombre-->
        <div class="insert_apellidos">
          <label for="apellidos">Apellidos:</label>
          <input type="text" id="apellidos" />
        </div> <!-- Cierre div apellidos-->
        <div class="insert_tlf">
          <label for="tlf">Móvil</label>
          <input type="tel" id="tlf"/>
        </div> <!-- Cierre div Tlf-->
        <div class="insert_email">
          <label for="email">Email</label>
          <input type="email" id="email"/>
        </div><!-- Cierre div email-->
      </div> <!-- Cierre datos_1-->
    </div><!-- Cierre div personales-->

    <div id="notas">
      <label for="otro">Dudas:</label>
      <textarea name="notas_1" id="otro" cols="95" rows="10"></textarea>
    </div><!-- Cierre div notas-->

    <div id="botones">
      <button type="submit">Enviar</button>
      <button type="reset">Reset</button>
    </div><!-- Cierre div botones-->
  </form>
</div>
```

# Ana

## Comentarios

Tuve bastantes dificultades para realizar el ejercicio, principalmente por falta de tiempo. Estuve haciendo algunas partes junto con Nagib para intentar realizar todo lo posible, y estuvimos mirando lo que ya hicieron otros compañeros que ya lo tenían más avanzado. El contenido de lo que hicimos está en el Github de Nagib.

Estuve más centrada en realizar la transformación a XML del requerimiento 2, pero los detalles de las decisiones que tomamos para el requerimiento 1 están descritos en el apartado de Nagib.

He sacado los atributos de ite como elementos. En cada profesor, he metido su elemento id como atributo de profesor, y eliminado la etiqueta nombre dejando solo su nombre dentro de la etiqueta.

Los elementos director y jefe\_estudios los he englobado en la etiqueta administracion, con elementos llamados personal. Al elemento personal le he asignado dos atributos, uno con el nombre y otro con el cargo. El contenido de la etiqueta es la dirección del despacho.

Los elementos ciclo están dentro de un elemento ciclos. En cada ciclo se han hecho como atributos los elementos grado y decretoTitulo, y el atributo id que tenía originalmente se ha renombrado a siglas. Dentro del elemento ciclo se ha puesto el contenido del elemento nombre.

Este es el resultado de la transformación en el modod de desarrollador del navegador:

# Rubén

## Comentarios

No he podido subirlo a git por falta de tiempo, pego aquí el código:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="2.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
<head>
<meta charset="UTF-8"/>
<meta http-equiv="X-UA-Compatible" content="IE=edge"/>
<meta name="viewport" content="width=device-width,
initial-scale=2.0"/>
<link rel="stylesheet" media='screen and (min-width: 801px)'
href="css/stylestd.css"/>
<link rel="stylesheet" media='screen and (max-width: 800px)'
href="css/stylemobile.css"/>
<title><xsl:value-of select="ite/@nombre"/></title>
</head>
<body class="fondo">
<!--img src="img/pexels-photo-3769021.webp" class="fondo"
alt="fondo_web"/-->
<header id="cabecera">

<h1>
<xsl:element name="a">
<xsl:attribute name="href">
<xsl:value-of select="ite/@web"/>
</xsl:attribute>
<xsl:value-of select="ite/@nombre"/>
</xsl:element>
</h1>
</header>
<nav class="collapsable-menu">
<input type="checkbox" id="menu"/>
<label for="menu"></label>
<div class="menu-content">
<ul class="lista-menu">
<li class="menulink" id="home"> <a href="index.html">Sobre Edix</a>
</li>
<li class="menulink" id="cursos"> <a href="cursos.html"
id="actual">Cursos</a> </li>
```

```

<li class="menulink" id="fp"> <a href="fp.html">Formación
Profesional</a> </li>
<li class="menulink" id="metodo"> <a
href="metodo.html">Metodología</a> </li>
<li class="menulink" id="empresas"> <a
href="empresas.html">Empresas colaboradoras</a> </li>
</ul>
</div>
</nav>
<section>
<header>
<h2>Ciclos Formativos de Grado Superior de Informática</h2>
</header>
<article class="intro">
<h3>Sé parte de la digitalización, desde dentro</h3>
<p>
En Edix estamos especializados en formación online de perfiles
digitales.
</p>
</article>
<article class="cfgs descripcion">
<h3>Además, ahora podrás hacerlo cursando titulaciones oficiales en
CFGs de la rama informática</h3>
<table>
<thead>
<tr>
<th class="colh1">Denominación</th>
<th class="colh2">Nombre</th>
<th class="colh3">Tipo de grado</th>
<th class="colh4">Año de creación</th>
</tr>
</thead>
<tbody>
<xsl:for-each select="ite/ciclos/ciclo">
<tr>
<td class="col1"><xsl:value-of select="@id"/></td>
<td class="col2"><xsl:value-of select="nombre"/></td>
<td class="col3"><xsl:value-of select="grado"/></td>
<td class="col4"><xsl:value-of select="decretoTitulo/@año"/></td>
</tr>
</xsl:for-each>
</tbody>
</table>
</article>
<article class="descripcion equipo">
<h3>Nuestro equipo</h3>
<p>Contarás la variada experiencia de nuestro equipo para tu
formación:</p>
<p>Serán<xsl:for-each select="ite/profesores/profesor">

```

```

o <span class="negrita"><xsl:value-of select="nombre"/></span>
</xsl:for-each> los que impartirán las clases de contenido
técnico.</p>
<table>
<caption><h4>Equipo directivo</h4></caption>
<thead>
<tr>
<th>Cargo</th>
<th>Nombre</th>
<th>Despacho</th>
</tr>
</thead>
<tbody>
<tr>
<td>Director/a</td>
<td><xsl:value-of select="ite/director/nombre"/></td>
<td><xsl:value-of select="ite/director/despacho"/></td>
</tr>
<tr>
<td>Jefe de estudios</td>
<td><xsl:value-of select="ite/jefe_estudios/nombre"/></td>
<td><xsl:value-of select="ite/jefe_estudios/despacho"/></td>
</tr>
</tbody>
</table>
</article>
<article class="descripcion formulario">
<h3>Solicita más información</h3>
<p>Si deseas inscribirte o solicitar más información, rellena tus
datos a continuación y estaremos encantados de resolver tus
dudas.</p>
<form method="get" action="mailto:midireccion@decorreo.es">
<fieldset><legend>¡Pregúntanos!</legend>
<div class="grid-container">
<label for="curso">Elige de qué curso deseas informarte</label>
<select name="curso">
<xsl:for-each select="ite/ciclos/ciclo">
<xsl:element name="option">
<xsl:attribute name="value">
<xsl:value-of select="@id"/>
</xsl:attribute>
<xsl:value-of select="nombre"/>
</xsl:element>
</xsl:for-each>
<option value="otros" selected="selected">Otros (preguntas
generales)</option>
</select>
<label for="detalles">Danos más detalles </label>
<input type="text" name="detalles" placeholder="(Opcional)"/>

```

```

<label for="email">Correo electrónico de contacto</label>
<input type="email" name="correo" placeholder="Tu email"
required="required"/>
<label for="telf">Teléfono de contacto</label>
<input type="tel" name="telf" placeholder="(Opcional)"/>
<div id="blank-space"/>
<input class="button" type="submit" value="Solicitar información"/>
</div>
</fieldset>
</form>
</article>
</section>
<footer>
</footer>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

```

* {
font-family: Arial, Helvetica, sans-serif;
text-decoration: none;
font-weight: bold;
color: black;
margin: 2;
padding: 2;
box-sizing: border-box;
}
body {
margin: 2 auto;
width: 3;
}
.intro {
font-size: 2em;
background: linear-gradient(0deg, rgba(0,0,0,0.4) 0%,
rgba(0,0,0,0.5) 20%, rgba(255,255,255,0) 100%),
url("../img/pexels-photo-3769021.webp");
background-blend-mode: darken;
background-size: cover;
background-position: center;
background-repeat: no-repeat;
display: flex;
flex-direction: column;
align-items: center;
height: 400px;
}

```

```

}
.intro > h3 {
padding: 8%;
}
.intro > p {
color: white;
padding: 25% 10% 10% 10%;
}
#cabecera {
align-items: flex-end;
padding: 4% 2% 2% 8%;
}
/*header > h1 > a {
padding: 1rem;
}*/
header {
display: block;
}
h1 {
padding: 2% 0%;
}
h2 {
color: #00c4da;
padding-left: 8%;
}
h3 {
color: #6a00c2;
}
section > header {
padding: 1%;
border-bottom: 5px solid #6a00c2;
}
nav {
width: 100%;
padding: 2% 0% 2% 0%;
}
nav ul {
display: flex;
flex-wrap: wrap;
flex-shrink: 1;
list-style: none;
margin: 0px;
padding: 0px;
}
nav ul li {
box-sizing: border-box;
font-size: 1.1em;
}
ul li a {

```



```

text-align: left;
font-weight: bold;
text-decoration: none;
padding: 6px;
float: left;
color: #6a00c2;
}
ul li a:hover,
ul li a span:hover {
color: #6a00c2;
background-color: #00c4da;
font-weight: bold;
transition: ease 0.3s;
}
/* Menu desplegado original */
nav ul {
display: flex;
flex-wrap: wrap;
flex-shrink: 1;
list-style: none;
float: left;
margin: 0px;
padding: 0px;
}
nav ul li a {
display: block;
width: 100vw;
text-align: center;
font-weight: bold;
text-decoration: none;
font-size: 0.9em;
display: block;
padding: 6px 0 6px 0;
float: left;
}
nav ul li a:hover {
font-weight: bold;
}
aside ul {
list-style: none;
padding-left: 1rem;
}
#contenedor > footer{
background: linear-gradient(to top, #5ccae1, white 90%);
float: left;
width: 100%;
}
#actual {
background-color: #5ccae1;

```

```

}
.menu-content {
position:relative;
left: -10px;
/*padding: 0 0 50px;*/
}
.collapsible-menu {
position: relative;
left: 10px;
/*padding: 0px 30px;*/
/*border-bottom: 1px solid #5ccae1;*/
/*box-shadow: 1px 2px 3px;*/
}
.collapsible-menu ul {
list-style-type: none;
padding: 0;
}
.collapsible-menu a {
display: block;
/*padding: 10px;*/
text-decoration: none;
font-size: 20px;
}
.collapsible-menu label {
display: block;
cursor: pointer;
width: 18px;
/*background: url(menu.png) no-repeat left center;*/
padding: 3px 0 0px 10px;
border-bottom: 10.5px double #326772;
border-top: 3.5px solid #326772;
margin-top: 10px;
margin-bottom: 10px;
}
.menu-content {
max-height: 0;
overflow: hidden;
font-family: 'Oswald', sans-serif;
/*padding: 0 0 0 50px;*/
}
#menu {
display: none;
}
#menu:checked {
display: none;
}
/* Toggle Effect */
#menu:checked~.menu-content {
max-height: 100%;

```

```

}
table {
box-sizing: content-box;
font-size: 0.95em;
}
td,
th {
padding: 1% 2%;
text-align: left;
min-height: 40px;
max-height: 120px;
}
th {
font-weight:900;
color:#00c4da;
border-right: 3px solid #6a00c2;
padding-right: 10px;
}
td {
padding-right: 7px;
}
caption {
padding-top: 5%;
font-style: italic;
}
table tr > *{
display: block;
}
table tr,
table thead,
table tbody {
display: table-cell;
}
.col1,
.colh1 {
height: 50px;
}
.col2,
.colh2 {
height:120px;
}
.col3,
.colh3 {
height:60px;
}
.descripcion {
padding: 8% 4% 2% 8%;
}
.descripcion > h3 {

```

```

font-size: 1.85em;
padding-bottom: 3%;
}
.equipo > p,
.formulario > p {
font-weight:normal;
}
.formulario {
background: #f7f7f7;
}
form {
width: auto;
padding-top: 20px;
}
legend {
font-size: 1.2em;
font-weight:900;
color:#00c4da;
padding:0 0 3% 1.5%;
}
fieldset {
border-width: 0;
}
.grid-container {
display: grid;
grid-template-columns: 1fr;
grid-auto-rows: 60px;
grid-template-rows: 50px 50px 50px 120px;
}
label {
align-self: flex-end;
}
select,
input,
label,
option {
margin: 1.5% 5%;
color: #555555;
}
select,
input,
option {
border: none;
border-radius: 3px;
padding-left: 2%;
}
select:hover,
input:hover {
border: none;

```

```
background:#e9fcff;
}
select:focus-visible,
input:focus-visible {
outline: none;
background:#f7ecff;
}
option:hover {
background-color:#e9fcff;
}
.button {
padding: 10px;
margin: 1% 45% 1% 5%;
color: white;
background: #6a00c2;
}
.button:hover {
background: #032258;
}
footer {
height: 200px;
background-image: url(../img/pexels-photo-97077.jpeg);
background-size: cover;
background-position: center;
background-repeat: no-repeat;
background-color: #909090;
background-blend-mode:multiply;
}
```

# Elección final

## Comentarios

Tras compartir las diversas soluciones, estuvimos comparando y discutiendo nuestras formas de resolverlo.

Para la decisión **en ambos requerimientos**, tuvimos en cuenta que **la transformación fuera satisfactoria** (es decir, que consiguiera generar el documento HTML ó XML correspondiente), que se hiciera **un uso diverso de las posibilidades en las transformaciones** (elementos, atributos, rutas, uso del for-each).

Además, para el **requerimiento 1** tuvimos en cuenta que la web resultante contara con **todos los elementos solicitados** (dos tablas, dos enlaces, lista, formulario) y también se tuvo en cuenta el **resultado visual obtenido de aplicar estilos CSS**.

## Elección del trabajo

Para ambos requerimientos, **decidimos elegir el trabajo de Luis**, ya que era satisfactorio en todos los aspectos mencionados en el apartado anterior.

Además, ha hecho **uso satisfactorio del for-each** para las transformaciones (en tablas, dentro de un párrafo de texto, al reestructurar el XML).

En el **requisito 1** ha elegido una **estructura** bastante **ordenada** en el **HTML** y en los estilos **CSS**, y ha conseguido un estilo visual trabajado, que ha **hecho uso del responsive** distinguiendo dos anchos de pantalla.

En el **requisito 2** además ha sido la transformación que ha realizado una **modificación más profunda de la estructura** del XML.