Actividad no presencial
Lenguajes de representación de contenidos

En esta actividad construiremos un documento HTML y un documento XML con algo de código.
Requisitos: un navegador web como Firefox o Chrome.

Documento HTML

Partimos de un documento HTML con un párrafo y una imagen. Si lo colocas en un archivo test.html y lo abres con tu navegador verás el efecto.

```
<!DOCTYPE html>
<html>
<body>
<p>Logo UPC</p>
<img src="upc.png"/>
</body>
</html>
```

El documento tiene un elemento HTML de tipo ‘p’ que contiene el texto ‘Logo UPC’ y el elemento ‘img’ con el atributo src y valor ‘upc.png’. La terminación ‘/>’ es una forma abreviada de <img src="upc.png"></img>. Si no tienes un archivo upc.png aparecerá en tu navegador una marca donde iría la imagen.

Documento XML

Ahora vamos a sustituir la imagen ‘img’ de tipo PNG, por una imagen en formato vectorial (Scalable Vector Graphics o SVG). SVG permite representar gráficos en formato vectorial codificados en XML. Por ejemplo el código y su representación en tu navegador a la derecha:

```
<svg xmlns="http://www.w3.org/2000/svg" width="150" height="150">
<circle id="circleId" cx="75" cy="75" r="60" stroke="blue" stroke-width="10" fill="lightblue"></circle>
<text id="label_id" x="46" y="115" fill="white" font-size="22" font-weight="bold"
  font-family="Tahoma">U P C</text>
</svg>
```

Se trata de un documento XML que sigue la especificación (espacio de nombres) xmlns="http://www.w3.org/2000/svg" y ocupa un área de 150x150 pixels con:
- un círculo centrado en (75,75) y 60 pixels de radio, borde de 10 pixels en azul y relleno de azul claro.
- un texto con una serie de atributos que pone UPC.
- y acaba el documento svg.

De hecho, el logo oficial de la UPC está definido en SVG como sigue:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg version="1.1" id="Capa_1" xmlns="http://www.w3.org/2000/svg" x="0px" y="0px" width="150px" height="150px"
viewBox="0 0 150 150" enable-background="new 0 0 150 150">
	<circle fill="#007BC0" cx="75" cy="75" r="60"/>
	<path fill="#FFFFFF" d="M55.816,112.92c0.795-0.01,1.313-0.17,1.925-0.678c0.886-0.734,0.998-1.626,1.017-
2.77V97.455h5.764V109.5  c0,2.104-0.075,4.697-2.84,6.65c-1.803,1.271-3.939,1.537-5.868,1.537c-1.931,0-4.067-0.267-
5.869-1.537  c-2.766-1.953-2.84-4.549-2.84-6.65V97.455h5.763v12.02c0.018,1.144,0.131,2.033,1.018,2.771
C54.498,112.754,55.021,112.91,55.816,112.92"/>
	<path fill="#FFFFFF" d="M74.183,106.424v-4.645h2.229c1.605,0,2.078,1.045,2.117,2.248c0.041,1.231-0.57,2.396-
2.059,2.396H74.183  M69.096,117.266h5.027v-6.484l1.666,0.029c3.211,0,4.461-0.297,5.563-1.07c1.664-1.161,2.588-
3.122,2.588-5.473  c0-4.371-2.498-6.813-6.545-6.813h-8.296v19.811H69.096z"/>
	<path fill="#FFFFFF" d="M102.039,116.547c-1.887,0.928-2.971,1.192-4.791,1.192c-5.979,0-10.201-4.253-10.201-
10.291  c0-5.978,4.313-10.409,10.143-10.409c1.795,0,2.955,0.291,4.851,1.225v5.186c-1.964-1.646-2.738-1.53-4.285-
1.53  c-3.004,0-5.086,2.291-5.086,5.531c0,3.271,2.11,5.53,5.115,5.53c1.624,0,2.438-0.213,4.256-
1.565L102.039,116.547"/>
	<path fill="#FFFFFF" d="M45.86,40.502c0-4.73,3.834-8.565,8.563-8.565c4.73,0,8.564,3.834,8.564,8.565  c0,4.729-
3.834,8.563-8.563,8.563C49.693,49.065,45.86,45.231,45.86,40.502"/>
```

```
    <path fill="#FFFFFF" d="M45.86,60.799c0-4.729,3.834-8.563,8.565-8.563c4.729,0,8.563,3.834,8.563,8.563
c0,4.73-3.834,8.565-8.563,8.565C49.693,69.364,45.86,65.529,45.86,60.799"/>
    <path fill="#FFFFFF" d="M45.86,81.633c0-4.729,3.834-8.563,8.565-8.563c4.729,0,8.563,3.834,8.563,8.563
c0,4.73-3.834,8.565-8.563,8.565C49.693,90.198,45.86,86.363,45.86,81.633"/>
    <path fill="#FFFFFF" d="M66.674,40.502c0-4.73,3.834-8.565,8.564-8.565c4.731,0,8.565,3.834,8.565,8.565
c0,4.729-3.834,8.563-8.564,8.563C70.51,49.065,66.676,45.231,66.674,40.502"/>
    <path fill="#FFFFFF" d="M66.676,60.799c0-4.729,3.834-8.563,8.564-8.563c4.729,0,8.563,3.834,8.563,8.563
c0,4.73-3.834,8.565-8.563,8.565C70.51,69.364,66.676,65.529,66.676,60.799"/>
    <path fill="#FFFFFF" d="M66.676,81.633c0-4.729,3.834-8.563,8.564-8.563c4.729,0,8.563,3.834,8.563,8.563
c0,4.73-3.834,8.565-8.563,8.565C70.51,90.198,66.676,86.363,66.676,81.633"/>
    <path fill="#FFFFFF" d="M87.01,40.502c0-4.73,3.834-8.565,8.563-8.565c4.729,0,8.564,3.834,8.564,8.565
c0,4.729-3.834,8.563-8.564,8.563C90.844,49.065,87.01,45.231,87.01,40.502"/>
    <path fill="#FFFFFF" d="M87.01,60.799c0-4.729,3.834-8.563,8.563-8.563c4.729,0,8.564,3.834,8.564,8.563
c0,4.73-3.834,8.565-8.564,8.565C90.844,69.364,87.01,65.529,87.01,60.799"/>
    <path fill="#FFFFFF" d="M87.01,81.633c0-4.729,3.834-8.563,8.563-8.563c4.729,0,8.564,3.834,8.564,8.563
c0,4.73-3.834,8.565-8.564,8.565C90.844,90.198,87.01,86.363,87.01,81.633"/>
</svg>
```

Lo anterior nos dice que se trata de un documento XML, codificado en UTF-8. El tipo de documento es SVG de acuerdo a unas especificaciones concretas que detalla (la versión 1.1 de SVG).
Este documento contiene un árbol XML con un elemento raíz <svg> que contiene el círculo del logo del color “#007BC0” centrado en las coordenadas 75,75 y radio 60.
A continuación vienen varios objetos en color blanco ‘"#FFFFFF"’ como caminos <path> con las letras ‘U’, la ‘P’, la ‘C’ y cada una de los 9 círculos.
Cada <path> se expresa como una serie de órdenes para dibujar. Por ejemplo el primero comienza con ‘M’ (moveto) a las coordenadas 55.816,112.92 seguido de ‘c’ una curva … El formato resulta complicado por lo que se recomienda utilizar un editor SVG como InkScape o LibreOffice para visualizar y editar cada elemento de la imagen.

Volviendo a un documento sencillo, vamos a intentar dibujar el logo UPC en una página HTML que contenga un documento SVG con su representación en tu navegador a la derecha:

```
<!DOCTYPE html>
<html>
<body>
<p>Logo UPC:</p>
<svg id="drawing" width="150" height="150">
<circle fill="#007BC0" cx="75" cy="75" r="60"/>
<text id="label_id" x="46" y="115" fill="white" font-size="22" font-weight="bold"
  font-family="Tahoma">U P C</text>
</svg>
</body>
```

Logo UPC:

Para poder completar el logo de la UPC hay que añadir varios círculos.
Para ello creamos la función *createCircle* en Javascript que devuelve un elemento SVG círculo con el espacio de nombre correcto y los atributos de posición, radio y color:

```
function createCircle(x, y, r) {
  var e = document.createElementNS("http://www.w3.org/2000/svg", 'circle');
  e.setAttribute("cx", x);
  e.setAttribute("cy", y);
  e.setAttribute("r", r);
  e.setAttribute("fill", "white");
  return e;
}
```

Al cargar la página HTML sólo hay que avisar que ejecute la función init():

```
<body onload="init();">
```

que es como sigue:

```
function init() {
  var svg = document.getElementById('drawing');

  for (var x=0; x < 3; x++) {
    for (var y=0; y < 3; y++) {
      var c = createCircle(54 + x*21, 46 + y*20, 8);
      svg.appendChild(c);
    }
  }
  var text = new XMLSerializer().serializeToString(svg);
  console.log(text);
}
```

La función busca en el documento HTML el elemento con *id="drawing"*, luego recorre las coordenadas de cada círculo y lo añade al dibujo (la raíz del SVG). Además llama a una función para convertir el SVG resultante en texto y lo enseña por la consola *(se puede ver en Firefox: Tools→Web Developer→Debugger, o en Chrome: View→Developer→Javascript console)*

El resultado es el siguiente:

```
<svg xmlns="http://www.w3.org/2000/svg" id="drawing" width="150" height="150">
  <circle fill="#007BC0" cx="75" cy="75" r="60"/>
  <text id="label_id" x="46" y="115" fill="white" font-size="22" font-weight="bold" font-family="'Tahoma">U P C</text>
  <circle cx="54" cy="46" r="8" fill="white"/>
  <circle cx="54" cy="66" r="8" fill="white"/>
  <circle cx="54" cy="86" r="8" fill="white"/>
  <circle cx="75" cy="46" r="8" fill="white"/>
  <circle cx="75" cy="66" r="8" fill="white"/>
  <circle cx="75" cy="86" r="8" fill="white"/>
  <circle cx="96" cy="46" r="8" fill="white"/>
  <circle cx="96" cy="66" r="8" fill="white"/>
  <circle cx="96" cy="86" r="8" fill="white"/>
</svg>
```

El resultado final es:

```
<!DOCTYPE html>
<head>
<script>
function init() {
  var svg = document.getElementById('drawing');

  for (var x=0; x < 3; x++) {
    for (var y=0; y < 3; y++) {
      var c = createCircle(54 + x*21, 46 + y*20, 8);
      svg.appendChild(c);
    }
  }
  var text = new XMLSerializer().serializeToString(svg);
  console.log(text);
}

function createCircle(x, y, r) {
   var e = document.createElementNS("http://www.w3.org/2000/svg", 'circle');
   e.setAttribute("cx", x);
   e.setAttribute("cy", y);
   e.setAttribute("r", r);
   e.setAttribute("fill", "white");
   return e;
}
</script>
</head>
<body onload="init();">
<p>Logo UPC:</p>
<svg id="drawing" width="150" height="150">
<circle fill="#007BC0" cx="75" cy="75" r="60"/>
<text id="label_id" x="46" y="115" fill="white" font-size="22" font-weight="bold" font-family="Tahoma">U P C</text>
</svg>
</body>
```

Qué entregar?

No hay que entregar nada.

El trabajo se evaluará con unas preguntas tipo test relacionadas con la práctica que se harán el día del examen final de laboratorio.

A continuación un ejemplo del tipo de preguntas que se harán para la evaluación de la práctica.

Dado el siguiente documento XML:

```
<svg xmlns="http://www.w3.org/2000/svg" id="drawing" width="150" height="150">
  <circle fill="#007BC0" cx="75" cy="75" r="60"/>
  <text id="label_id" x="46" y="115" fill="white" font-size="22" font-weight="bold" font-family="Tahoma">U P C</text>
  <circle cx="54" cy="46" r="8" fill="white"/>
```

Indica cuales de las siguientes respuestas son correctas:

1) Está bien formado: (falso)
2) Falta el elemento </svg> al final (cierto)
3) Sobra el elemento <svg> al principio (falso)
4) El atributo xmlns permite verificar la validez respecto a una especificación de SVG (cierto)

Referencias:

- SVG: https://www.w3.org/TR/SVG/ (Versión 1.1, 2011)
- Editores SVG: https://en.wikipedia.org/wiki/Comparison_of_vector_graphics_editors
- Javascript basics: https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/JavaScript_basics
- Javascript: https://en.wikipedia.org/wiki/JavaScript

- HTML: https://www.w3schools.com/html/html_intro.asp