

1. Instala BaseX y documenta el proceso

Para realizar su instalación necesitamos entrar a su [página web](#) y seleccionar 1 de las 4 opciones que nos ofrece según dónde lo vayamos a instalar. Una vez descargado hacemos la instalación donde mejor nos venga y ya estaría listo.

2. Carga un documento XML de tu elección (libros, componentes informáticos, etc.) Realiza y documenta al menos tres de los cuatro siguientes apartados.

a) Lanza dos consultas XQuery distintas

- for \$a in distinct-values(doc("Tutoriales.xml")//categoria)
return <categoriasdistintas>{data(\$a)}</categoriasdistintas>
- for \$titulos in doc("Tutoriales.xml")/tutoriales/tutorial
return <lostutoriales> {\$titulos/titulo/text() } ({\$titulos/visitas/text()}) </lostutoriales>

b) Transforma el XML a JSON

XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<tutoriales>
  <tutorial anio="2007">
    <titulo>XMLBeans, una forma de mapear un XML en objetos Java</titulo>
    <autor>
      <nombre>Alejandro</nombre>
      <apellidos>Pérez García</apellidos>
    </autor>
    <categoria>XML</categoria>
    <visitas>25827</visitas>
  </tutorial>
  <tutorial anio="2013">
    <titulo>Spring Security: haciendo uso de un servidor LDAP embebido</titulo>
    <autor>
      <nombre>José Manuel</nombre>
      <apellidos>Sánchez Suárez</apellidos>
    </autor>
    <categoria>Spring</categoria>
    <visitas>1996</visitas>
  </tutorial>
  <tutorial anio="2011">
    <titulo>Mybatis con Maven y Spring</titulo>
    <autor>
      <nombre>Rubén</nombre>
      <apellidos>Aguilera Díaz-Heredero</apellidos>
    </autor>
    <categoria>Mybatis</categoria>
    <visitas>6998</visitas>
  </tutorial>
  <tutorial anio="2011">
    <titulo>Trabajar con XML sabiendo lo mínimo</titulo>
    <autor>
      <nombre>Roberto</nombre>
      <apellidos>Canales Mora</apellidos>
    </autor>
    <categoria>XML</categoria>
    <visitas>34842</visitas>
  </tutorial>
</tutoriales>
```

```

</tutorial>
<tutorial anio="2013">
  <titulo>Introducción a Spring Batch</titulo>
  <autor>
    <nombre>Miguel</nombre>
    <apellidos>Arlandy Rodríguez</apellidos>
  </autor>
  <categoria>Spring</categoria>
  <visitas>1273</visitas>
</tutorial>
</tutoriales>

```

JSON:

```

{
  "tutoriales": {
    "tutorial": [
      {
        "-anio": "2007",
        "titulo": "XMLBeans, una forma de mapear un XML en objetos Java",
        "autor": {
          "nombre": "Alejandro",
          "apellidos": "Pérez García"
        },
        "categoria": "XML",
        "visitas": "25827"
      },
      {
        "-anio": "2013",
        "titulo": "Spring Security: haciendo uso de un servidor LDAP embebido",
        "autor": {
          "nombre": "José Manuel",
          "apellidos": "Sánchez Suárez"
        },
        "categoria": "Spring",
        "visitas": "1996"
      },
      {
        "-anio": "2011",
        "titulo": "Mybatis con Maven y Spring",
        "autor": {
          "nombre": "Rubén",
          "apellidos": "Aguilera Díaz-Heredero"
        },
        "categoria": "Mybatis",
        "visitas": "6998"
      },
      {
        "-anio": "2011",
        "titulo": "Trabajar con XML sabiendo lo mínimo",
        "autor": {
          "nombre": "Roberto",
          "apellidos": "Canales Mora"
        }
      }
    ]
  }
}

```

```

    },
    "categoria": "XML",
    "visitas": "34842"
  },
  {
    "-anio": "2013",
    "titulo": "Introducción a Spring Batch",
    "autor": {
      "nombre": "Miguel",
      "apellidos": "Arlandy Rodríguez"
    },
    "categoria": "Spring",
    "visitas": "1273"
  }
]
}

```

c) Devuelve dos consultas XPath

- `//@anio`
- `//tutorial[categoria="Spring"]`

d) Valida un documento contra un XML Schema

XML

```

<?xml version="1.0" encoding="UTF-8"?>
<ubicaciones xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="ubicaciones.xsd">
  <ubicacion metros="23">norte</ubicacion>
  <ubicacion metros="52">este</ubicacion>
  <ubicacion metros="46">este</ubicacion>
</ubicaciones>

```

XSD

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="ubicaciones">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ubicacion" type="infoUbicacion" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:complexType name="infoUbicacion">
    <xs:simpleContent>
      <xs:extension base="direccion">
        <xs:attribute name="metros" type="xs:integer"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>

```

```

<xs:simpleType name="direccion">
  <xs:restriction base="xs:string">
    <xs:enumeration value="norte"/>
    <xs:enumeration value="sur"/>
    <xs:enumeration value="este"/>
    <xs:enumeration value="oeste"/>
  </xs:restriction>
</xs:simpleType>
</xs:schema>

```

e) Valida un documento contra un DTD

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE listaCompra [
  <!ELEMENT listaCompra (item*)>
  <!ELEMENT item EMPTY>
  <!ATTLIST item nombre CDATA #REQUIRED>
  <!ATTLIST item cantidad CDATA #REQUIRED>
]>

```

```

<listaCompra>
  <item nombre="leche" cantidad="12 litros" />
  <item nombre="pan" cantidad="3 barras de cuarto" />
</listaCompra>

```

f) Lleva a cabo un backup, y después realiza su restauración

g) Declara tu propia función en XQuery

h) Lleva a cabo algún tratamiento de XML/JSON original