

LMSGI - EXAMEN UD4	
Fecha: 16/12/2024	
CICLO: DESARROLLO DE APLICACIONES WEB CURSO: 1º MÓDULO: LENGUAJES DE MARCAS Y SGI	CALIFICACIÓN:
Nombre: Adrian	Apellidos: Aránegas Molina

Lee detenidamente y de forma completa cada uno de los problemas planteados y responde únicamente a lo que se pide.

La resolución de la prueba consiste en crear un repositorio en el usuario de GitHub. Dicho repositorio incluye un directorio para cada ejercicio del examen.

Añadir a continuación la URL de acceso al repositorio (obligatorio, no puntúa):

Ejercicio1 (5 puntos)

Crear en Visual Studio Code el siguiente script Python:

```
import json

def json_validator(data):
    try:
        json.loads(data)
        return True
    except ValueError as error:
        print(f"JSON inválido: {error}")
        return False

from datetime import datetime

def validate_date_format(date_string):
    try:
        datetime.strptime(date_string, "%d-%m-%Y")
        ''' Esto es un comentario: esta función comprueba
           que el formato de fecha es DD/MM/YYYY
        '''

        return True
    except ValueError:
        return False

with open("Ejercicio1\datos.json") as json_file:
    data = json.load(json_file)

    if "fecha_nacimiento" in data:
        if validate_date_format(data["fecha_nacimiento"]):
            print("Fecha de nacimiento válida")
        else:
            print("Fecha de nacimiento inválida")
    else:
        print("No se encontró la clave 'fecha_nacimiento' en el JSON")
```

Crear en Visual Studio Code el siguiente archivo .json:

```
{
    "nombre": "Juan"
}
```

Construir:

1. Actualizar el archivo datos.json para que incluya el elemento fecha_nacimiento
2. Ejecutar el script. El mensaje por pantalla debe mostrar: Fecha de nacimiento válida

Añadir una captura de pantalla de la ejecución del script y resultado.

[illegible]

Ejercicio 2 (1,5 puntos)

Haciendo uso del siguiente código:

```
<xs:complexType name="tipoColorMueble">
  <xs:simpleContent>
    <xs:extension base="tipoMueble">
      <xs:attribute name="color">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="blanco"/>
            <xs:enumeration value="gris"/>
            <xs:enumeration value="negro"/>
            <xs:enumeration value="wengue"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

<xs:simpleType name="tipoMueble">
  <xs:restriction base="xs:string">
    <xs:enumeration value="armario"/>
    <xs:enumeration value="mesa"/>
    <xs:enumeration value="silla"/>
  </xs:restriction>
</xs:simpleType>
```

Escribir el contenido del archivo "muebles.xsd" que permita validar el siguiente documento XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<muebles xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="muebles.xsd">
  <mueble color="blanco">mesa</mueble>
  <mueble color="gris">silla</mueble>
</muebles>
```

Añadir captura de pantalla del archivo muebles.xml haciendo referencia a XSD

[illegible]

The screenshot shows a web browser window with the address bar displaying the URL: 127.0.0.1:5500/Examen%20U54/fjercoco2/muebles.xml. The browser's developer tools are open, showing the XML content of the file. The XML is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<muebles xmlns:xsi="http://www.w3.org/2003/XMLSchema-instance" xmlns:namespaces:schemaLocation="muebles.xsd">
  <mueble color="blanco" grin="si"/>
</muebles>
```

The XML structure consists of a root element 'muebles' with an 'xmlns:xsi' attribute pointing to 'http://www.w3.org/2003/XMLSchema-instance' and an 'xmlns:namespaces:schemaLocation' attribute pointing to 'muebles.xsd'. Inside the root element, there is a single child element 'mueble' with two attributes: 'color="blanco"' and 'grin="si"'. The XML is enclosed in a single-line format with no line breaks.

Ejercicio 3 (1,5 puntos)

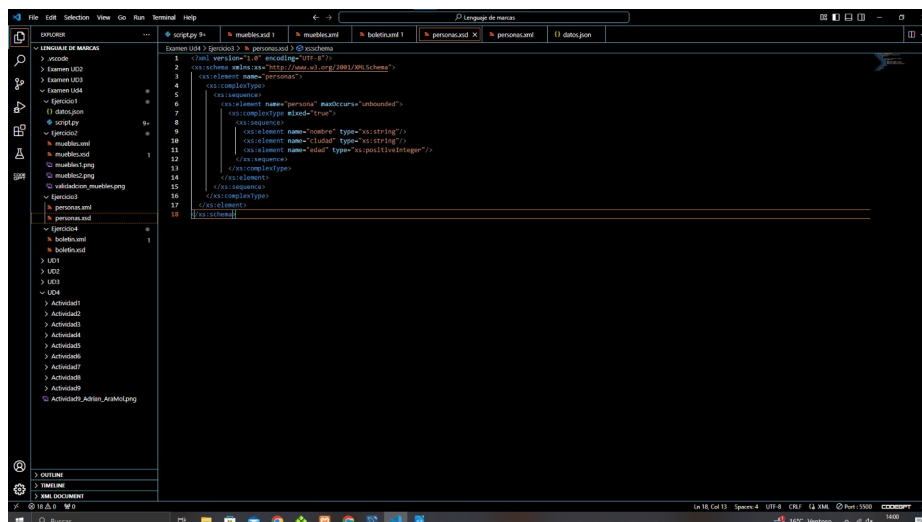
Dado el archivo "personas.xsd". Utilizando los elementos "nombre", "ciudad" y "edad", escribir el código de un documento XML que pueda ser validado por "personas.xsd" y que almacene la siguiente información:

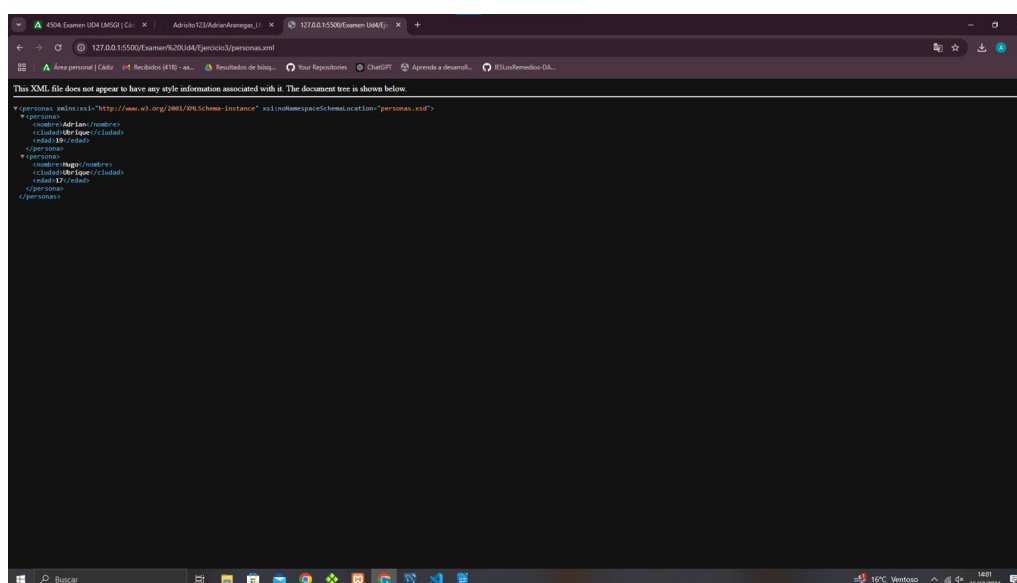
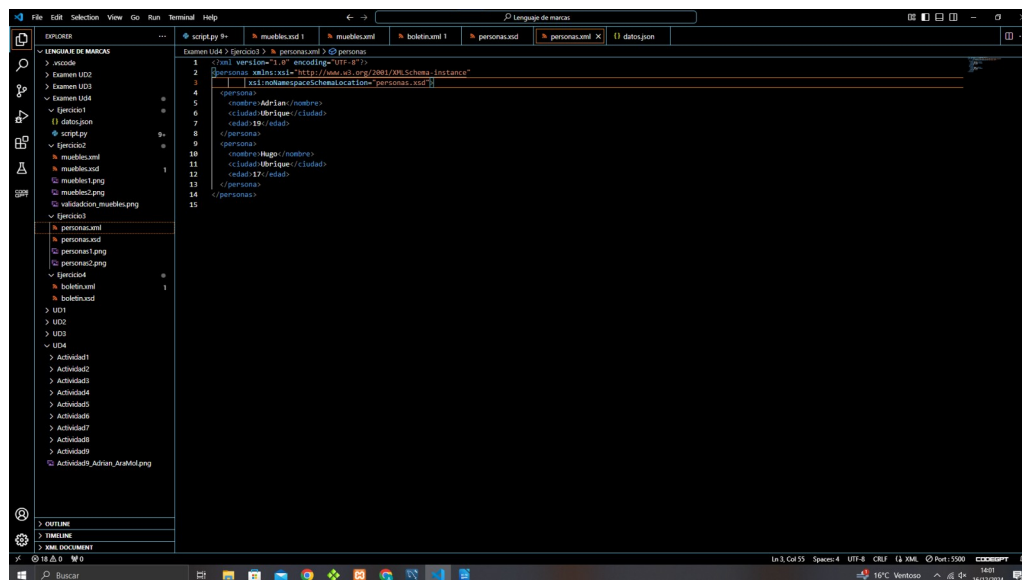
"Eva vive en París y tiene 25 años."

"Giovanni vive en Florencia y tiene 26 años."

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="personas">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="persona" maxOccurs="unbounded">
          <xs:complexType mixed="true">
            <xs:sequence>
              <xs:element name="nombre" type="xs:string"/>
              <xs:element name="ciudad" type="xs:string"/>
              <xs:element name="edad" type="xs:positiveInteger"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Añadir captura de pantalla del archivo personas.xml haciendo referencia a XSD





Ejercicio 4 (2 puntos)

Dado el siguiente xml, generar boletin.xsd para su validación

```
<?xml version="1.0" encoding="utf-8"?>
<boletin
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="boletin.xsd">
    <nombre>Juan Perez</nombre>
    <ciclo>Administración Sistemas Inf.</ciclo>
    <curso>1</curso>
    <evaluacion>1</evaluacion>
    <cualificacion>
        <modulo>Lenguajes de Marcas</modulo>
        <curso>1</curso>
        <nota>6</nota>
    </cualificacion>
    <cualificacion>
        <modulo>Bases de Datos</modulo>
        <curso>1</curso>
        <nota>5</nota>
    </cualificacion>
</boletin>
<!-- La nota es un entero entre 1 y 10 -->
```

Añadir captura de pantalla del archivo boletin.xml haciendo referencia a XSD

