

Tipología y fuentes de datos

PEC 3

UOC

Universitat Oberta
de Catalunya

[NOMBRE y APELLIDOS]

Fecha de entrega :
30 de noviembre de 2021

Tipología y fuentes de datos

PEC3

Enunciado 1 (4 puntos)

Responda a las siguientes cuestiones:

1. Identifique los conceptos fundamentales de la arquitectura web:
 - a. El protocolo TCP/IP ()
 - b. El lenguaje HTML ()
 - c. Las hojas de estilo CSS ()
 - d. Los motores de búsqueda ()
 - e. Las direcciones URL ()
 - f. El protocolo HTTP ()
2. Seleccionar las sentencias que sean ciertas de las siguientes:
 - a. XML es un lenguaje de marcado ()
 - b. XML es un formato de texto ()
 - c. XML es legible por humanos ()
 - d. XML es legible por ordenadores ()
 - e. XML es un estándar abierto y gratuito ()
 - f. XML permite definir documentos y estructuras de datos ()
3. En la arquitectura de la web semántica (web de datos), ¿qué cambios se han realizado para describir los recursos?
 - a. HTTP reemplazado por RDF ()
 - b. URIs o IRIs generalizan URLs ()
 - c. HTML reemplazada por XSL ()
 - d. HTML reemplazado por RDF ()
 - e. HTTP reemplazado por SMTP ()
 - f. URLs reemplazados por DOIs ()
4. En los principios de los datos abiertos y enlazados se recomienda que:
 - a. se proporcionen descripciones en inglés de los recursos identificados ()
 - b. se utilice URIs HTTP para identificar recursos ()
 - c. se publiquen todos los datos incluidos los privados ()
 - d. se actualicen los datos en tiempo real ()
 - e. se proporcione una descripción del recurso cuando se acceda a la URI ()
 - f. se incluyan enlaces a otros recursos en la descripción del recurso ()
 - g. se proporcione software para procesar los datos ()
 - h. se firmen los datos con claves seguras ()

5. ¿Cuál es la estructura de datos del modelo RDF? [Sólo 1 es correcta]
- un árbol XML ()
 - una lista doblemente enlazada ()
 - un multigrafo orientado y etiquetado ()
 - una matriz de adyacencias ()
 - una única tripleta ()
6. ¿Qué describen las siguientes sentencias RDF/XML? [Sólo 1 es correcta]

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:exs="http://example.org/schema#">
  <rdf:Description rdf:about="http://example.org/doc.html">
    <rdf:type rdf:resource="http://example.org/schema#Report"/>
    <exs:theme rdf:resource="http://example.org#Music"/>
    <exs:theme rdf:resource="http://example.org#History"/>
    <exs:nbPages rdf:datatype="http://www.w3.org/2001/XMLSchema#int">23</exs:nbPages>
  </rdf:Description>
</rdf:RDF>
```

- Una bibliografía con 23 referencias en Historia de la Música ()
 - Un informe de 23 páginas sobre Música e Historia ()
 - Un documento sobre las relaciones entre Historia y Música ()
7. Para representar en RDF a los tres ganadores de una carrera en orden de llegada se usará preferentemente:
- alternativas (rdf:Alt) ()
 - un secuencia (rdf:Seq) ()
 - una colección (rdf:List) ()
 - un grupo (rdf:Bag) ()
8. RDFS proporciona primitivas para:
- describir clases de recursos ()
 - describir fórmulas para calcular valores de propiedades ()
 - describir tipos de propiedades ()
 - firmar y autenticar a los autores de las clases y propiedades ()
9. Dadas las siguientes tripletas:

```
@prefix ex: <http://example.org/demo/#> .
_:a ex:name "Alicia".
_:b ex:name "Benito".
_:c ex:name "Carmen".
_:d ex:name "Daniel".
_:b ex:mail <mailto:alicia@chaka.sn>.
_:a ex:mail <mailto:benito@pachinko.jp>.
_:d ex:mail <mailto:daniel@tempmail.com>.
_:a ex:mail <mailto:alice@pachinko.jp>.
```

Y la siguiente consulta:

```
SELECT ?name
WHERE {
  ?x ex:name ?name;
      ex:mail ?email. }
```

¿Cuál será el resultado de la consulta?

- a. Alicia (2 veces) ()
- b. Benito (1 vez) ()
- c. Carmen (1 vez) ()
- d. Daniel (2 veces) ()

10. ¿Qué es lo que la siguiente consulta pregunta? [Sólo 1 es correcta]

```
PREFIX ex: <http://www.example.org/#>
SELECT ?x
WHERE {
  ?x rdf:type ?type .
  FILTER (?type != ex:Man)
}
```

- a. Por todas las mujeres ()
- b. Por todos los hombres ()
- c. Por todos los recursos que tengan al menos un tipo diferentes de ex:Man ()
- d. Por recursos que no sean de tipo ex:Man ()

11. Dada la siguiente consulta deducir lo que está calculando: [Sólo 1 es correcta]

```
prefix ex: <http://www.example.org/>
select ?x (avg(?a) as ?b)
where {
  ?x ex:knows ?y .
  ?y ex:age ?a
}
group by ?x
```

- a. La edad de la gente conocida por ?x ()
- b. La media de edad de la gente conocida por ?x ()
- c. La número de gente conocida de ?x que tienen edad ()
- d. Para cada valor de ?x, la media de edad de la gente que conoce ()

12. Considerando el siguiente grafo:



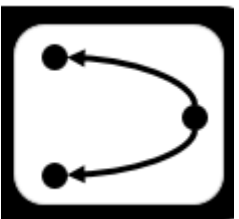
```
ex:John foaf:knows ex:Jack, ex:James .
ex:Jim foaf:knows ex:James, ex:Jack .
```

¿Qué es lo que devuelve la siguiente consulta? [Sólo 1 es correcta]

```
prefix foaf: <http://xmlns.com/foaf/0.1/>
select distinct ?x ?z where {
  ?x foaf:knows ?y .
  ?z foaf:knows ?y
  filter (?x != ?z)
}
```

- a. ?x = ex:John, ?z = ex:Jim ()
- b. ?x = ex:Jim, ?z = ex:John ()
- c. ?x = ex:John, ?z = ex:Jim ; ?x = ex:Jim, ?z = ex:John ()

¿Qué tipo de grafo corresponde a la cláusula WHERE de la consulta?

		
a. ()	b. ()	c. ()

13. Dadas las siguientes tripletas:

```
ex:John ex:age "18" .
ex:Jim ex:age "20" .
ex:Jack ex:age "32" .
```

Escribir el filtro correcto de forma que la consulta devuelva los menores de 30 años. [Sólo 1 es correcta]

```
select * where {
  ?x ex:age ?a
  filter ( _____ )
}
```

- a. filter (xsd:string(?a) <= 30) ()
- b. filter (xsd:integer(?a) <= 30) ()
- c. filter (?a <= xsd:integer(30)) ()
- d. filter (xsd:integer(?a) <= xsd:string(30)) ()

14. Indicar las sentencias correctas sobre ontologías:

- a. Una ontología consiste en una sola clase cuyo significado varía según el contexto de la aplicación ()
- b. Una ontología ayuda a compartir la comprensión de la estructura de la información entre personas y agentes software. ()
- c. Una ontología es una especificación formal y explícita de una conceptualización compartida. ()
- d. Una ontología se refiere a un modelo que contiene un conjunto de clases junto con un conjunto de relaciones entre ellas. ()

15. Si podemos inferir que

```
:ElBierzo :subregionDe :España .
```

de las sentencias:

```
:ElBierzo :subregionDe :León .  
:León :subregionDe :España .
```

¿Qué restricción hay que imponer :subregionDe? [Sólo 1 es correcta]

- a. :subregionDe a owl:AsymmetricProperty . ()
- b. :subregionDe a owl:FunctionalProperty . ()
- c. :subregionDe a owl:IrreflexiveProperty . ()
- d. :subregionDe a owl:TransitiveProperty . ()

Enunciado 2 (3 puntos)

En el jupyter notebook que se encuentra en la PEC resuelva los diferentes fragmentos de código marcados con #TODO sobre RDF y RDFS con la librería `rdflib`. Consisten en completar una grafo RDF/RDFS y realizar unas consultas SPARQL sobre ese grafo. Recuerde que el notebook se puede ejecutar en [Google Colab](#) según instrucciones en el Anexo.

Enunciado 3 (3 puntos)

En el mismo notebook, en esta sección trata sobre ontologías y su utilización en Python con la librería `OwlReady2`. Como en el enunciado anterior, complete el código marcado para crear una ontología y responder a las cuestiones con SPARQL.

Anexo

Descarga en tu equipo el notebook TyFdD-PEC3-2021.ipynb así como los archivos de datos en csv. El notebook está pensado para que lo ejecutes en Google Colab.

Google Colab es un servicio en la nube que permite ejecutar Jupyter Notebooks accediendo con un navegador web. Tiene además las siguientes ventajas:

- Posibilidad de ejecución mediante GPUs
- Basado en jupyter notebook pudiendo crear y ejecutar libros en Python 2 o 3
- Tiene preinstaladas las librerías comunes usadas en ciencia de datos y la posibilidad de instalar otras.
- Enlaza con cuentas de Google Drive y desde github

Primero hay que entrar en sesión (login) con una cuenta de Google (la de la uoc debería funcionar).

Ahora ya se puede subir el notebook de la PEC a colab:



La ejecución del libro es exactamente igual que en cualquier jupyter notebook. Hay que pulsar Shift + Enter para que el código (python) se ejecute.

Alternativamente también es posible subirlos con la herramienta para subir archivos del propio colab:

A partir de ahí debes completar el código python que falta que está marcado con **# TODO** y responder a las cuestiones planteadas escribiendo tanto las respuestas como el código con el que obtienes las respuestas.

Una vez finalizado, debes descargar el archivo .ipynb para poder hacer la entrega.

Criterios de valoración

Cada uno de los apartados tiene un peso asignado en el total de la PEC. Se valorará, para cada apartado, la validez de la solución y la claridad de la argumentación.

Formato y fecha de entrega

Tenéis que enviar la PEC al buzón de Entrega y registro de EC disponible en el aula (apartado Evaluación). El formato del archivo que contiene vuestra solución puede ser .pdf, .odt, .doc y .docx. Para otras opciones, por favor, contactar previamente con vuestro profesor colaborador. El nombre del fichero debe contener el código de la asignatura, vuestro apellido y vuestro nombre, así como el número de actividad (PEC2). Por ejemplo *apellido1_nombre_tyfdd_pecX.pdf*. No olvidéis poner vuestro nombre y apellidos en el documento.

La fecha límite para entregar la PEC es **la indicada en la portada**.

Propiedad intelectual

Al presentar una práctica o PEC que haga uso de recursos ajenos, se tiene que presentar junto con ella un documento en que se detallen todos ellos, especificando el nombre de cada recurso, su autor, el lugar donde se obtuvo y su estatus legal: si la obra está protegida por el copyright o se acoge a alguna otra licencia de uso (Creative Commons, licencia GNU, GPL etc.). El estudiante tendrá que asegurarse que la licencia que sea no impide específicamente su uso en el marco de la práctica o PEC. En caso de no encontrar la información correspondiente tendrá que asumir que la obra está protegida por el copyright.

Será necesario, además, adjuntar los ficheros originales cuando las obras utilizadas sean digitales y su código fuente, si así corresponde.

