**Nombre:** **DNI:**

**Internet, Seguridad y Distribución de Contenidos Multimedia** **ISDCM-MEI**
**Curso 2024-25 Q2** **Duración: 1h 15'**

**Primer examen parcial** **SOLUCION** **2025/04/07**

**Preguntas Test (5 puntos).** Indicar si las siguientes afirmaciones son ciertas o falsas. Cada respuesta correcta son 0,2 puntos. Cada respuesta incorrecta descuenta **0,2 puntos**.
SE PUEDEN AÑADIR COMENTARIOS PARA JUSTIFICAR LAS RESPUESTAS.

**Dado el siguiente documento XML, contestar las siguientes dos preguntas:**

```
<?xml version="1.0"?>
<PaymentInfo xmlns="http://example.org/paymentv2">
      <Name>John Smith</Name>
      <CreditCard Limit="5000" Currency="USD">
            <Number>4019 2445 0277 5567</Number>
            <Issuer>Example Bank</Issuer>
            <Expiration>05/24</Expiration>
      </CreditCard>
</PaymentInfo>
```

**1. "USD" va entre comillas por tratarse del valor de un atributo.**

☐ Cierto ☐ Falso

Answer: True.

**2. "Name" es el "root element" del documento XML por ser el primero de los elementos en "PaymentInfo".**

☐ Cierto ☐ Falso

Answer: False. The root element is "PaymentInfo".

**Contestar las siguientes dos preguntas relacionadas con una "HTTP response" (los números a la izquierda no son parte del mensaje, sino números de línea).**

```
1. HTTP/1.1 200 OK
2. Accept-Ranges: bytes
3. Age: 294510
4. Cache-Control: max-age=604800
5. Content-Type: text/html; charset=UTF-8
6. Date: Sat, 11 Jan 2025 14:18:33 GMT
7. Expires: Sun, 12 Jan 2025 14:18:33 GMT
8. Last-Modified: Thu, 17 Oct 2019 07:18:26 GMT
9. Server: ECAcc (nyd/D10E)
10.Content-Length: 1256
```

**3. La "HTTP Request Line" correspondiente a esta respuesta acaba con "HTTP/1.1".**

☐ Cierto ☐ Falso

Answer: True.

**4. El "status code" de la respuesta corresponde con el mensaje "Moved Permanently".**

☐ Cierto ☐ Falso

Answer: False. The status code is "200", which indicates that the resource has been correctly sent.

**5. Una de las ventajas de HTTP/2 es que añade nuevos métodos a los ya existentes en HTTP/1.1.**

☐ Cierto ☐ Falso

Answer: False. It does not add new methods.

**6. El carácter Unicode cuyo "Code Point" es 100 (en Decimal) necesita 2 "Code Units" al codificarse en "UTF-16".**

☐ Cierto ☐ Falso

Answer: False. It needs just one code unit of 16 bits.

**7. *Advanced Audio Coding* (AAC) y Vorbis son ejemplos de estándares "open-source" que compiten con MP3.**

☐ Cierto ☐ Falso

Answer: False. AAC is not open-source and it is an evolution of MP3.

**8. "JPSearch core metadata" es un estándar de ISO.**

☐ Cierto ☐ Falso

Answer: True.

**9. El formato de imágenes AVIF es el único caso en el que la representación de la imagen se basa en un mecanismo de compresión de vídeo.**

☐ Cierto ☐ Falso

Answer: False. There are others. It is a current trend.

**10. Los formatos de gráficos vectoriales se basan en el uso de objetos geométricos para definir imágenes.**

☐ Cierto ☐ Falso

Answer: True.

**11. SMIL (Synchronized Multimedia Integration Language) es un estándar basado en XML para especificar gráficos vectoriales.**

☐ Cierto ☐ Falso

Answer: False. It is a W3C standard to describe multimedia presentations and animations.

**12. El estándar MPEG-4 especifica 2 mecanismos de compresión de vídeo diferentes.**

☐ Cierto ☐ Falso

Answer: True. In parts 2 and 10. The latter is the AVC.

**13. El formato de contenedor WebM, de Google, incluye AVC para video y AAC para audio.**

☐ Cierto ☐ Falso

Answer: False. It initially included VP8 (video) and Vorbis (audio), but now it has been improved to VP9 (video) and Opus (audio).

**14. Dublin Core son metadatos que permiten describir contenido, incluyendo objetos físicos.**

☐ Cierto ☐ Falso

Answer: True.

**15. IEEE LOM y MPEG-7 son estándares de metadatos equivalentes en el sentido de que permiten describir el mismo tipo de contenido.**

☐ Cierto ☐ Falso

Answer: False. IEEE LOM describes educational content, while MPEG-7 describes general multimedia content.

**16. La descripción de datos digitales puede estar dentro o fuera del propio contenido descrito.**

☐ Cierto ☐ Falso

Answer: True.

**17. Después de establecer una sesión RTSP, podríamos enviar datos multimedia usando los protocolos RTP y RTCP.**

☐ Cierto ☐ Falso

Answer: True. RTP sends the multimedia data, and RTCP sends the protocol control data.

**Contestar las siguientes dos preguntas relacionadas con el siguiente intercambio RTSP (C se refiere a Cliente, mientras que S se refiere a Servidor):**

```
C: TEARDOWN movie.Mjpeg RTSP/1.0
C: CSeq: 5
C: Session: 123456

S: RTSP/1.0 200 OK
S: CSeq: 5
S: Session: 123456
```

**18. Para realizar este "Request", se necesita establecer antes una sesión.**

☐ Cierto ☐ Falso

Answer: True. This is to close an already established communication.

**19. Se podría usar TCP como protocolo de transporte.**

☐ Cierto ☐ Falso

Answer: True.

**20. Organizaciones como 3GPP han especificado perfiles DASH aparte de los especificados por el grupo de estandarización de MPEG.**

☐ Cierto ☐ Falso

Answer: True.

**21. En MPEG-DASH hay varios mecanismos para indexar segmentos. Uno de ellos es utilizar ficheros indexados.**

☐ Cierto ☐ Falso

Answer: True.

**Contestar las siguientes dos preguntas relacionadas con este fragmento de MPD:**

```
<MPD xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:mpeg:dash:schema:mpd:2011" xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011
DASH-MPD.xsd" type="dynamic" minimumUpdatePeriod="PT2S" timeShiftBufferDepth="PT30M"
availabilityStartTime="2021-12-25T12:30:00" minBufferTime="PT4S"
profiles="urn:mpeg:dash:profile:isoff-live:2011">
  <BaseURL>http://cdn1.example.com/</BaseURL>
  <BaseURL>http://cdn2.example.com/</BaseURL>
  <Period id="1">
```

```
    <AdaptationSet mimeType="video/mp4" codecs="avc1.4D401F" frameRate="30000/1001"
segmentAlignment="true" startWithSAP="1">
      <BaseURL>video/</BaseURL>
      …
    </AdaptationSet>
  </Period>
</MPD>
```

**22. El acceso a los segmentos del contenido se define con la característica de rango de bytes de HTTP.**

☐ Cierto ☐ Falso

Answer: False. We do not know, since this information is not included.

**23. El "default namespace" se define con una URN.**

☐ Cierto ☐ Falso

Answer: True. It is "urn:mpeg:dash:schema:mpd:2011".

**Dos de los documentos aprobados durante la reunión 126 de ISO/IEC JTC1 SC29/WG11 (MPEG) son:**

a) **"Disposition of comments on DIS ISO/IEC 23092-3 (ISO/IEC 23092-3 - Genomic Information Metadata and Application Programming Interfaces (APIs))"**

b) **"Text of ISO/IEC FDIS 23092-3 Genomic Information Metadata and Application Programming Interfaces (APIs)"**

**24. El documento a) es el resultado de la votación de un FDIS (el paso que sigue al DIS).**

☐ Cierto ☐ Falso

Answer: False. It is the result of the DIS ballot. Once a ballot has been received, the next step is to produce a "Disposition of Comments". Then, those comments are implemented in the next step. In this case, the Disposition of Comments is on a DIS, and a FDIS (or an IS) is produced.

**25. Respecto al documento b), en FDIS 23092-3, el número 3 significa que es la parte 3 de 23092. Por tanto, no se desarrollará 23902-4 hasta que 23092-3 sea finalmente aprobado.**

☐ Cierto ☐ Falso

Answer: False. The different parts of a standard are developed independently, not necessarily in the order of part numbers.

## Problema 1 (2,5 *puntos*)

Dada la especificación WSDL del Anexo I,

### Contestar razonada y brevemente a las siguientes preguntas:

1) ¿Qué es "`tCheckAvailability`"?: ¿Qué elementos tiene? ¿Dónde y para qué se usa?

```
<xs:complexType name="tCheckAvailability">
    <xs:sequence>
      <xs:element  name="checkInDate" type="xs:date"/>
      <xs:element  name="checkOutDate" type="xs:date"/>
      <xs:element  name="roomType" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
```

The data structure for "checkAvailability", which is the Request of the operation "opCheckAvailability".

It includes 3 elements: 2 dates for checkin and checkout dates and a string for the room type.

2) Esta especificación es para usar con SOAP. Si queremos que funciones con REST, ¿cómo sería el nuevo elemento de binding? ¿Qué más habría que cambiar para hacerlo con REST?

The current binding, to SOAP, is:

```
<binding name="reservationSOAPBinding"
    interface="tns:reservationInterface"
    type="http://www.w3.org/ns/wsdl/soap"
    wsoap:protocol="http://www.w3.org/2003/05/soap/bindings/HTTP/">

  <fault ref="tns:invalidDataFault"
   wsoap:code="soap:Sender"/>

  <operation ref="tns:opCheckAvailability"
   wsoap:mep="http://www.w3.org/2003/05/soap/mep/soap-response"/>

</binding>
```

In the case of REST, it could be:

```
<binding     name="reservationRESTBinding"
             interface=" tns:reservationInterface">
             type="http://www.w3.org/ns/wsdl/http"
<operation   ref="tns:opCheckAvailability"
             whttp:method="GET"/>
</binding>
```

We could consider adding the “fault” elements.

There are no other major changes, except eliminating the elements and attributes specific for SOAP.

3) **a)** Si hacemos el “binding” sobre REST, proporcionar un ejemplo de URL para un “request” de la operación "opCheckAvailability". **b)** ¿Qué método HTTP se debería usar? **c)** ¿Cuál sería el contenido del `body` del HTTP `Request`?

a) A simple example, with the values of the 3 elements of the complex type, could be:

http://greath.example.com/2004/reservation/opCheckAvailability/250407/250408/double

A more complete example, including also the names of the elements, could be:

http://greath.example.com/2004/reservation/opCheckAvailability?checkInDate=250407&checkOutDate=250408&roomType=double

b) GET could be used when all parameters are in the URL, as in the previous examples. The solution in 2 assumes this situation and specifies GET as the HTTP method.

An alternative solution could be to decide that the parameters are not to be included in the URL, but in the body of the HTTP Request, in which case the POST method should be used.

c) If we use GET, all the information would be in the HTTP’s Request line; therefore, the “body” would be empty. On the contrary, if we use POST, the “body” wouls include the values of the parameter of the operation.

4) Si no se ha tenido en cuenta en las respuestas a las preguntas 2 y 3, ¿cómo se deberían incluir los mecanismos de “fault” tanto en atributos como elementos al pasar el servicio a REST? Nota: Véanse las aclaraciones al final del anexo.

The specified “fault” implies that a specific response should be given in case of the specified error.

In this case, the error refers to invalid data; i.e., the values of the input parameters are wrong, so it is not possible to check any room availability.

The simplest approach to this error management could be to provide the error’s answer in the body of the HTTP Response.

An alternative solution could be to try to map the error directly in the HTTP status codes.

## Problema 2 (1 *punto*)

(Se adjunta al final del examen un resumen de las reglas Unicode)

Considerar los 2 siguientes *Code Points* correspondientes a caracteres Unicode:

a) 8001 (Hexadecimal) ó 32769 (Decimal)

b) 801 (Hexadecimal) ó 2049 (Decimal)

**Contestar razonada y brevemente a las siguientes preguntas:**

**1)** Codificar “a” en UTF-8.

**2)** Codificar “b” en UTF-16.

**3)** ¿Cuál el code point más cercano (menor) a “b” que reduce el tamaño de su codificación en UTF-16?

**4)** Lo mismo que la pregunta 3 pero en UTF-8.

1) The encoding is (in binary and hexadecimal):
8001 = **1110**1000 **10**000000 **10**000001 - E8 80 81 (3 bytes)

2) The encoding is (in hexadecimal):
801 (2 bytes)

3) It already has the minimum size.

4) Number “b” in UTF-8 is (in binary and hexadecimal):
801 (12 bits) = **1110**0000 **10**100000 **10**000001 - E0 A0 81

With 11 bits (for the Code Point) we only need 2 bytes to encode in UTF-8. 2047 (decimal) is the closest one (7FF in hex).

## Problema 3 (1,5 *puntos*)

**Contestar razonada y brevemente las siguientes preguntas sobre este fragmento de MPD DASH:**

```
<MPD  type="static" xmlns="urn:mpeg:dash:schema:mpd:2011"
      xmlns:dvb="urn:dvb:dash-extensions:2014-1"
      profiles="urn:dvb:dash:profile:dvb-dash:2014,urn:dvb:dash:profile:dvb-dash:isoff-
ext-live:2014"
      minBufferTime="PT2.337S" maxSegmentDuration="PT10S"
      mediaPresentationDuration="PT10M31.960S">
  <ProgramInformation>
…
  </ProgramInformation>
  <BaseURL serviceLocation="A" dvb:priority="1" dvb:weight="1">
    http://rdmedia.bbc.co.uk/dash/ondemand/elephants_dream/1/</BaseURL>
  <Period duration="PT10M31.960S" start="PT0S">
    <AdaptationSet startWithSAP="2" segmentAlignment="true" id="1" sar="1:1"
     frameRate="25" contentType="video" mimeType="video/mp4">
      <BaseURL>avc3/</BaseURL>
      <SegmentTemplate timescale="1000" duration="3840"
       media="$RepresentationID$/$Number%06d$.m4s"
       initialization="$RepresentationID$/IS.mp4"/>
        <Representation id="1280x720p25" codecs="avc3.640020" height="720"
         width="1280" scanType="progressive" bandwidth="6556136" />
    </AdaptationSet>
    <AdaptationSet startWithSAP="2" segmentAlignment="true" id="3"
     codecs="mp4a.40.2" audioSamplingRate="48000" lang="eng" contentType="audio"
     mimeType="audio/mp4">
      <AudioChannelConfiguration schemeIdUri=
       "urn:mpeg:dash:23003:3:audio_channel:2011" value="2"/>
        <BaseURL>audio/</BaseURL>
        <SegmentTemplate startNumber="1" timescale="1000" duration="3840"
         media="$RepresentationID$/$Number%06d$.m4s"
         initialization="$RepresentationID$/IS.mp4"/>
          <Representation id="128kbps" bandwidth="128000"/>
    </AdaptationSet>
    <AdaptationSet startWithSAP="2" segmentAlignment="true" id="5" codecs="stpp"
     lang="eng" contentType="text" mimeType="application/mp4">
…
```

1) **a)** ¿Cómo es de largo en tiempo el contenido descrito en este MPD? ¿Dónde se especifica? **b)** ¿Cuál es la duración máxima, en segundos, de los segmentos? ¿Dónde se especifica?

a) 10 minutes and 31.96 seconds. It is specified in the attribute “duration” of the element “Period” and in the attribute “mediaPresentationDuration” of the element “MPD” (the value is “PT10M31.960S”).

b) 10 seconds, as specified in the attribute maxSegmentDuration="PT10S" from the MPD element.

2) **a)** ¿Qué perfil o perfiles MPEG-DASH sigue este MPD? ¿Cómo se sabe? **b)** ¿Qué SDOs lo(s) ha(n) especificado?

a) The value of the "profiles" attribute of the MPD element includes two profiles:

"urn:dvb:dash:profile:dvb-dash:2014, urn:dvb:dash:profile:dvb-dash:isoff-ext-live:2014".

b) The two profiles are specified by DVB. The second one seems to be a DVB extension of the ISOFF live profile.

3) **a)** ¿Cuántas `Representation` de video se especifican? Identificar su(s) tipo(s) de contenido. **b)** ¿Hay alguna información sobre el idioma de la(s) `Representation`?

a) One video representation (video/mp4, avc3, 1280x720p25) in the first adaptation set.

b) The second adaptation set (audio) has an attribute lang with value "eng", so it is in English.

4) Dar un ejemplo de URL para acceder a un segmento dentro del `AdaptationSet` con `id="3"`. ¿Qué suposiciones son necesarias para contestar?

We should use first the BaseURL elements that appear in the MPD ("http://rdmedia.bbc.co.uk/dash/ondemand/elephants_dream/1/") and AdaptationSet ("audio/"). Then, we should complete it with the template of the segments' names. For example:

http://rdmedia.bbc.co.uk/dash/ondemand/elephants_dream/1/audio/repId1/000001.m4s

("repId1" is the given identifier, but others could work, such as "128kbps")

(we have chosen the first segment, then "000001" (with 6 characters) is used)

5) Si disponemos de un ancho de banda de 6.600.000 bps para transmisión de vídeo y audio, ¿cuáles son los identificadores de las representaciones que hemos de usar?

We have only one option for video, with bandwidth="6556136", and one for audio, with bandwidth="128000"

Both bandwidths together do not fit in the available 6.600.000 bps, so we cannot transmit this content.

**ANEXO I**. Ejemplo de WSDL (part of *the GreatH Web Service*)
**(Aclaraciones al final)**

```
<?xml version="1.0" encoding="utf-8" ?>
<description
    xmlns="http://www.w3.org/ns/wsdl"
    targetNamespace= "http://greath.example.com/2004/wsdl/resSvc"
    xmlns:tns= "http://greath.example.com/2004/wsdl/resSvc"
    xmlns:ghns = "http://greath.example.com/2004/schemas/resSvc"
    xmlns:wsoap= "http://www.w3.org/ns/wsdl/soap"
    xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
    xmlns:wsdlx= "http://www.w3.org/ns/wsdl-extensions">

  <types>
    <xs:schema
        xmlns:xs="http://www.w3.org/2001/XMLSchema"
        targetNamespace="http://greath.example.com/2004/schemas/resSvc"
        xmlns="http://greath.example.com/2004/schemas/resSvc">

      <xs:element name="checkAvailability" type="tCheckAvailability"/>

      <xs:complexType name="tCheckAvailability">
        <xs:sequence>
          <xs:element  name="checkInDate" type="xs:date"/>
          <xs:element  name="checkOutDate" type="xs:date"/>
          <xs:element  name="roomType" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>

      <xs:element name="checkAvailabilityResponse" type="xs:double"/>

      <xs:element name="invalidDataError" type="xs:string"/>

    </xs:schema>

  </types>

  <interface  name = "reservationInterface" >

    <fault name = "invalidDataFault"
            element = "ghns:invalidDataError"/>

    <operation name="opCheckAvailability"
            pattern="http://www.w3.org/ns/wsdl/in-out"
            style="http://www.w3.org/ns/wsdl/style/iri"
            wsdlx:safe = "true">
        <input messageLabel="In"
              element="ghns:checkAvailability" />
        <output messageLabel="Out"
              element="ghns:checkAvailabilityResponse" />
        <outfault ref="tns:invalidDataFault" messageLabel="Out"/>
    </operation>

  </interface>

  <binding name="reservationSOAPBinding"
          interface="tns:reservationInterface"
          type="http://www.w3.org/ns/wsdl/soap"
          wsoap:protocol="http://www.w3.org/2003/05/soap/bindings/HTTP/">

    <fault ref="tns:invalidDataFault"
      wsoap:code="soap:Sender"/>

    <operation ref="tns:opCheckAvailability"
      wsoap:mep="http://www.w3.org/2003/05/soap/mep/soap-response"/>

  </binding>
```

```
  <service name="reservationService"
       interface="tns:reservationInterface">

     <endpoint name="reservationEndpoint"
               binding="tns:reservationSOAPBinding"
               address ="http://greath.example.com/2004/reservation"/>

  </service>

</description>
```

- Aclaraciones sobre los atributos de `fault`:

`<fault name = "invalidDataFault"`

The `name` attribute defines a name for this fault. The name is required so that when an operation is defined, it can reference the desired fault by name. Fault names must be unique within an interface.

`element = "ghns:invalidDataError"/>`

The `element` attribute specifies the schema type of the fault message, as previously defined in the `types` section.

- Aclaraciones sobre atributos y elementos de `operation`:

`wsdlx:safe="true" >`

This line indicates that this operation will not obligate the client in any way, i.e., the client can safely invoke this operation without fear that it may be incurring an obligation (such as agreeing to buy something).

`<outfault ref="tns:invalidDataFault" messageLabel="Out"/>`

This associates an output fault with this operation. Faults are declared a little differently than normal messages. The `ref` attribute refers to the name of a previously defined fault in this interface -- not a message schema type directly. Since message exchange patterns could in general involve a sequence of several messages, a fault could potentially occur at various points within the message sequence. Because one may wish to associate a different fault with each permitted point in the sequence, the `messageLabel` is used to indicate the desired point for this particular fault. It does so indirectly by specifying the message that will either trigger this fault or that this fault will replace, depending on the pattern.

## UTF-8

- A code unit is a single byte
- A code point is from 1 to 4 code units
- Code units between 0 and 127 directly represent the corresponding code points
- `110XXXXX` indicates that 2 code units are used
- `1110XXXX` indicates that 3 code units are used
- `11110XXX` indicates that 4 code units are used
- The remaining code units looks like `10XXXXXX`





## UTF-8 Example

- `11100011 10000001 10010101`
- `1110`0011 `10`000001 `10`010101
- `11000001010101`
- `12,373` 3055 H
- `HIRAGANA LETTER SA`





## UTF-16

- A code unit consists of 2 bytes
- Code points below 65,536 are in a single code unit
- Higher code points are represented as:
  - `110110XXXXXXXXXX 110111XXXXXXXXXX`

  (after subtracting 65,536)
- This makes sense because Unicode assign no code points between the numbers:

  `1101100000000000 (55,296)`

  and

  `1101111111111111 (57,343)`





## UTF-16 Example

"Big-endian byte order" / Character

- `11111110 11111111 00110000 01010101`
- `00110000 01010101`
- `12,373` 3055 H
- `HIRAGANA LETTER SA`