

Pruebas para la obtención de títulos de Técnico y Técnico Superior

Convocatoria correspondiente al curso 2022-2023

(Resolución de 13 de diciembre de 2022 de la Dirección General de Educación Secundaria, Formación Profesional y Régimen Especial)

DATOS DEL ALUMNO			FIRMA
APELLIDOS:			
Nombre:	D.N.I. N.I.E:	Fecha 10 mayo 2023	
Código del ciclo: IFCS02	Denominación completa del título: Técnico Superior de FP en Desarrollo de Aplicaciones Multiplataforma		
Clave o código del módulo: 468	Denominación completa del módulo profesional: ACCESO A DATOS		

INSTRUCCIONES GENERALES PARA LA REALIZACIÓN DE LA PRUEBA
<ol style="list-style-type: none"> 1. Sobre la mesa de examen sólo podrá haber: examen, bolígrafo azul o negro y DNI. 2. En ningún caso está permitido el uso de teléfonos móviles ni ningún otro dispositivo electrónico. Deberán estar apagados y guardados. 3. Durante la realización de la prueba se observarán todas las normas elementales de comportamiento. Todos permanecerán en silencio. Para hacer preguntas o entregar exámenes se levantará la mano. 4. Se deben rellenar los datos del aspirante en todas las páginas. 5. El examen consta de 14 páginas, contando la presente. 6. Cada pregunta tipo test se responderá en el espacio dejado a tal efecto en la hoja de respuestas. Si se quiere rectificar una respuesta contestada, se podrá tachar y volver a escribir. 7. Se dispondrá de una hoja para borrador, o de varias si se requieren, que será proporcionada por el centro. Esta hoja se entregará obligatoriamente al final junto con el examen, si bien nada de lo escrito en la hoja de borrador se valorará en la corrección. 8. La duración de la prueba será de 75 minutos.
CRITERIOS DE CALIFICACIÓN Y VALORACIÓN
<ul style="list-style-type: none"> • La parte de preguntas tipo test se valora con 8 puntos. <ul style="list-style-type: none"> o Todas las preguntas tienen la misma valoración. o Una respuesta incorrecta resta un cuarto del valor de una pregunta acertada • La parte de consultas XQuery se valora con 2 puntos. Todas las preguntas tienen la misma valoración. <ul style="list-style-type: none"> o Se obtendrá la valoración completa si la opción es la correcta o Una respuesta incorrecta resta un cuarto del valor de una pregunta acertada • La nota del módulo será el resultado de truncar al entero más próximo por debajo la puntuación obtenida en la prueba.

CALIFICACIÓN

1	<input type="checkbox"/>	a	<input type="checkbox"/>	b	<input type="checkbox"/>	c	<input type="checkbox"/>	d
2	<input type="checkbox"/>	a	<input type="checkbox"/>	b	<input type="checkbox"/>	c	<input type="checkbox"/>	d
3	<input type="checkbox"/>	a	<input type="checkbox"/>	b	<input type="checkbox"/>	c	<input type="checkbox"/>	d
4	<input type="checkbox"/>	a	<input type="checkbox"/>	b	<input type="checkbox"/>	c	<input type="checkbox"/>	d
5	<input type="checkbox"/>	a	<input type="checkbox"/>	b	<input type="checkbox"/>	c	<input type="checkbox"/>	d
6	<input type="checkbox"/>	a	<input type="checkbox"/>	b	<input type="checkbox"/>	c	<input type="checkbox"/>	d
7	<input type="checkbox"/>	a	<input type="checkbox"/>	b	<input type="checkbox"/>	c	<input type="checkbox"/>	d
8	<input type="checkbox"/>	a	<input type="checkbox"/>	b	<input type="checkbox"/>	c	<input type="checkbox"/>	d
9	<input type="checkbox"/>	a	<input type="checkbox"/>	b	<input type="checkbox"/>	c	<input type="checkbox"/>	d
10	<input type="checkbox"/>	a	<input type="checkbox"/>	b	<input type="checkbox"/>	c	<input type="checkbox"/>	d
11	<input type="checkbox"/>	a	<input type="checkbox"/>	b	<input type="checkbox"/>	c	<input type="checkbox"/>	d
12	<input type="checkbox"/>	a	<input type="checkbox"/>	b	<input type="checkbox"/>	c	<input type="checkbox"/>	d
13	<input type="checkbox"/>	a	<input type="checkbox"/>	b	<input type="checkbox"/>	c	<input type="checkbox"/>	d
14	<input type="checkbox"/>	a	<input type="checkbox"/>	b	<input type="checkbox"/>	c	<input type="checkbox"/>	d
15	<input type="checkbox"/>	a	<input type="checkbox"/>	b	<input type="checkbox"/>	c	<input type="checkbox"/>	d
16	<input type="checkbox"/>	a	<input type="checkbox"/>	b	<input type="checkbox"/>	c	<input type="checkbox"/>	d
17	<input type="checkbox"/>	a	<input type="checkbox"/>	b	<input type="checkbox"/>	c	<input type="checkbox"/>	d
18	<input type="checkbox"/>	a	<input type="checkbox"/>	b	<input type="checkbox"/>	c	<input type="checkbox"/>	d
19	<input type="checkbox"/>	a	<input type="checkbox"/>	b	<input type="checkbox"/>	c	<input type="checkbox"/>	d
20	<input type="checkbox"/>	a	<input type="checkbox"/>	b	<input type="checkbox"/>	c	<input type="checkbox"/>	d
21	<input type="checkbox"/>	a	<input type="checkbox"/>	b	<input type="checkbox"/>	c	<input type="checkbox"/>	d
22	<input type="checkbox"/>	a	<input type="checkbox"/>	b	<input type="checkbox"/>	c	<input type="checkbox"/>	d
23	<input type="checkbox"/>	a	<input type="checkbox"/>	b	<input type="checkbox"/>	c	<input type="checkbox"/>	d
24	<input type="checkbox"/>	a	<input type="checkbox"/>	b	<input type="checkbox"/>	c	<input type="checkbox"/>	d
25	<input type="checkbox"/>	a	<input type="checkbox"/>	b	<input type="checkbox"/>	c	<input type="checkbox"/>	d
26	<input type="checkbox"/>	a	<input type="checkbox"/>	b	<input type="checkbox"/>	c	<input type="checkbox"/>	d
27	<input type="checkbox"/>	a	<input type="checkbox"/>	b	<input type="checkbox"/>	c	<input type="checkbox"/>	d
28	<input type="checkbox"/>	a	<input type="checkbox"/>	b	<input type="checkbox"/>	c	<input type="checkbox"/>	d
29	<input type="checkbox"/>	a	<input type="checkbox"/>	b	<input type="checkbox"/>	c	<input type="checkbox"/>	d
30	<input type="checkbox"/>	a	<input type="checkbox"/>	b	<input type="checkbox"/>	c	<input type="checkbox"/>	d
31	<input type="checkbox"/>	a	<input type="checkbox"/>	b	<input type="checkbox"/>	c	<input type="checkbox"/>	d
32	<input type="checkbox"/>	a	<input type="checkbox"/>	b	<input type="checkbox"/>	c	<input type="checkbox"/>	d
33	<input type="checkbox"/>	a	<input type="checkbox"/>	b	<input type="checkbox"/>	c	<input type="checkbox"/>	d
34	<input type="checkbox"/>	a	<input type="checkbox"/>	b	<input type="checkbox"/>	c	<input type="checkbox"/>	d
35	<input type="checkbox"/>	a	<input type="checkbox"/>	b	<input type="checkbox"/>	c	<input type="checkbox"/>	d
36	<input type="checkbox"/>	a	<input type="checkbox"/>	b	<input type="checkbox"/>	c	<input type="checkbox"/>	d
37	<input type="checkbox"/>	a	<input type="checkbox"/>	b	<input type="checkbox"/>	c	<input type="checkbox"/>	d
38	<input type="checkbox"/>	a	<input type="checkbox"/>	b	<input type="checkbox"/>	c	<input type="checkbox"/>	d
39	<input type="checkbox"/>	a	<input type="checkbox"/>	b	<input type="checkbox"/>	c	<input type="checkbox"/>	d
40	<input type="checkbox"/>	a	<input type="checkbox"/>	b	<input type="checkbox"/>	c	<input type="checkbox"/>	d

DATOS DEL ALUMNO			FIRMA
APELLIDOS:			
Nombre:	D.N.I. N.I.E. o Pasaporte:	Fecha 10 mayo 2023	

Preguntas Test

- Un stream es un flujo de datos. En Java, ¿de qué se compone de un Stream?
 - Bytes
 - Bits
 - Paquetes
 - Depende del modo de apertura del stream
- El método "public int available() throws IOException" pertenece a la clase:
 - java.io.OutputStream
 - java.io.InputStream
 - java.io.Reader
 - A todos los tipos de stream
- Un programa Java tiene una variable de tipo InputStream.
 - Para poder usarse, se crea un DataInputStream con ella, y ya se pueden leer bytes y caracteres del stream original.
 - Se puede leer de ella un byte, con el método read()
 - Se puede leer de ella un caracter, con el método readChar()
 - Se puede leer de ella un caracter o un byte, usando readChar() y read()
- Una aplicación tiene como requisito funcionar en cualquier sistema, independientemente de su arquitectura de procesador, sistema operativo y middleware instalados. Debería usar un driver JDBC
 - De tipo 1
 - De tipo 3
 - De tipo 4
 - De cualquier tipo, menos el 1
- Qué clase no tiene sentido utilizar para procesar la entrada del usuario por teclado
 - DataInputStream
 - BufferedReader
 - Reader
 - InputStream
- ¿Qué clase incluye facilidades para procesar la entrada de datos línea a línea?
 - DataInput
 - BufferedWriter
 - Reader
 - InputStream

7. El modo "rws" y "rwd" de RandomAccessFile:
- son más lentos que "w", ya que reservan recursos también para leer del fichero
 - son equivalentes a "rw", porque solo se tienen en cuenta los dos primeros caracteres de la cadena
 - son más rápidos que otros modos, pero se pueden perder datos si el programa no se cierra correctamente
 - son más lentos que otros modos, si el número de escrituras es alto
8. Un ResultSet con TYPE_SCROLL_SENSITIVE
- Hace que el programa no pueda modificar los datos devueltos por el ResultSet
 - Hace que el ResultSet intente actualizar los datos que se solicitan a los más actuales en la base de datos
 - Hace que el ResultSet sea una fotografía (snapshot) de los datos que había en la base de datos en el momento de lanzarse, y no se vea afectado por cambios posteriores
 - Hace que el programa sí pueda modificar los datos devueltos por el ResultSet, y se graben de vuelta a la base de datos
9. Respecto a XML. ¿Cuáles son tecnologías de modelización y validación de datos?
- Xpath, XQuery
 - DTD, XML Schema.
 - XSLT, DOM, SAX.
 - XRPC, DOM
10. ¿Cuál de los siguientes archivos es necesario para configurar el mapeo de clases con Hibernate?
- pom.xml
 - hibernate.cfg.xml
 - orm.xml
 - persistence.xml
11. El parámetro SELF en un miembro de objeto Oracle
- Existe aunque no se lo declare, y siempre se puede leer (IN)
 - Existe aunque no se lo declare, y siempre se puede leer y escribir (IN/OUT)
 - Puede existir o no, ya que no todos los miembros necesitan accederlo
 - Puede existir o no, pero en el constructor existe siempre
12. El método ORDER de un tipo de objeto de Oracle
- Se usa para decidir el orden de ese objeto entre todos los objetos, asignando un número a cada objeto
 - Se usa para decidir el orden entre dos objetos de ese tipo, devolviendo un número entero
 - Se usa para decidir el orden entre dos objetos de cualquier tipo. Las conversiones son a cuenta del programador
 - Se usa para decidir el orden entre dos objetos de ese tipo, devolviendo el objeto mayor, o NULL si son iguales

DATOS DEL ALUMNO		FIRMA
APELLIDOS:		
Nombre:	D.N.I. N.I.E. o Pasaporte:	
		Fecha 10 mayo 2023

13. El dialecto de Hibernate

- Se indica opcionalmente, para optimizar el acceso
- Se indica obligatoriamente, pero hay un dialecto por defecto que funciona en todas las bases de datos SQL92 en adelante
- Se indica obligatoriamente, y cada SGBD tiene el suyo propio posiblemente incompatible con las demás
- El dialecto es un concepto de JPA, Hibernate solo lo utiliza si se usa con JPA

14. Qué opción es correcta acerca de la relación entre Hibernate y JPA

- Todos los programas que usan Hibernate usan JPA, pero no al revés
- Todos los programas que usan JPA usan Hibernate
- Pueden trabajar juntos, accediendo a funcionalidad de Hibernate a través del API de JPA
- Pueden trabajar juntos, accediendo a funcionalidad de JPA a través del API de Hibernate

15. El valor "create-drop" para "hbm2ddl.auto" de Hibernate es interesante para

- Es un valor equivalente a "create" (se mantiene por compatibilidad)
- Crear las tablas que no existan aún (porque se han borrado previamente con drop)
- Borrar todos las tablas de base de datos y volverla a crear (para inicializar la base de datos)
- Crear la base de datos al principio del programa, y borrar la base de datos al final (para hacer pruebas)

16. Un pool de JDBC

- Intenta mejorar el rendimiento utilizando varios servidores de base de datos
- Agrupar a varios programas clientes de la base de datos para no sobrecargar al servidor
- Mantiene varias conexiones abiertas a la base de datos, reutilizándolas para mejorar el tiempo de respuesta
- Mantiene en memoria las consultas ya realizadas, para no volver a lanzarlas al servidor

17. Para insertar datos con JDBC se utiliza el método:

- executeUpdate()
- executeInsert()
- execute()
- executeQuery()

18. Para ejecutar procedimientos almacenados con JDBC se utiliza una instancia de:

- PreparedStatement, para poder pasar los parámetros necesarios
- StoredProcedure
- CallableStatement
- OracleStatement, u otra clase propia del driver de la base de datos, ya que no todos los SGBD tienen procedimientos almacenados

19. Son ejes de XPath
- child y descendant
 - ancestor-or-descendant y self
 - ancestor-or-self y descendant
 - XPath permiten definir ejes de usuario, todas podrían ser correctas
20. Una clase de entidad JPA debe cumplir que
- Debe tener un constructor por defecto
 - Debe ser serializable
 - Debe definir los métodos toString() y hashCode()
 - Todos sus atributos deben ser serializables
21. Una clase serializable de Java
- Debe tener cuidado de no hacer autoreferencias, ya que la escritura a un ObjectOutputStream no acabaría nunca
 - Puede tener atributos no serializables, pero deben ser de tipo final
 - No necesita tener un constructor por defecto
 - Puede modificar la forma en que se lee y escribe (writeObject() y readObject()), de forma que no lo haga a un ObjectOutputStream, sino a otros tipos de Stream
22. Respecto al valor del parámetro de modo de acceso, el constructor de la clase RandomAccessFile:
- no permite el valor "r"
 - no permite el valor "w"
 - no permite el valor "rw"
 - permite todos los anteriores
23. En Oracle estándar ¿Qué tipo de colección de objetos tiene establecido un número máximo de elementos que puede almacenar?
- VTABLE
 - NESTED TABLE
 - Los creados con CREATE TYPE ... IS TABLE OF ...
 - VARRAY
24. El método DocumentBuilderFactory.newInstance() devuelve
- Una instancia de DocumentBuilderParser
 - Una instancia de Document
 - Una instancia de DocumentBuilder
 - Una instancia de DocumentBuilderFactory
25. Acerca del ciclo de vida de un objeto de Hibernate
- Se puede pasar de TRANSIENT a DETACHED, cuando se termina la transacción
 - Se puede pasar de DETACHED a MANAGED, con Session.update()
 - El ciclo es siempre NEW -> MANAGED -> PERSISTED. PERSISTED sirve tanto para filas existentes como no existentes
 - El ciclo es siempre NEW -> MANAGED -> PERSISTED/REMOVED, dependiendo de si al final de la transacción la fila referenciada existe o no en la base de datos

DATOS DEL ALUMNO		FIRMA
APELLIDOS:		
Nombre:	D.N.I. N.I.E. o Pasaporte:	
		Fecha 10 mayo 2023

26. Un Listener de java asociado a un componente debe heredar de
- ObjectEventListener
 - EventListener
 - EventListenerSupport
 - SourceEventListener
27. Pueden prohibirse los cambios de una propiedad bean ...
- si los listeners implementan VetoableChangeListener
 - si no se asocia ningún listener a una propiedad
 - si el bean se ha diseñado ya preparado para VetoableChangeListener
 - si se lanza un VetoableChangeEvent en vez de un PropertyChangeEvent
28. Acerca de ODBC y JDBC
- El mismo código Java puede utilizar ODBC o JDBC, sin más que cambiar la cadena de conexión
 - JDBC es la versión escrita en Java del estándar ODBC
 - Son tecnologías con el mismo propósito y funcionalidad parecida, pero sus API no son compatibles entre sí.
 - JDBC es el puente para que un programa Java utilice ODBC
29. Para asociar una clase BeanInfo a un Bean
- Se crea una clase que herede de BeanInfo, y se añade como propiedad estática a la clase de Bean
 - Se registra la subclase de BeanInfo en el Introspector
 - Se registra un Listener el Introspector, para modificar el BeanInfo que se devuelve por defecto dependiendo de la clase de Bean solicitada
 - Se crea una clase que herede de BeanInfo, cuyo nombre sea del del Bean añadiendo BeanInfo al final
30. Los tipos de objeto en Oracle (CREATE TYPE ...)
- Cuando se define un tipo de objeto se puede heredar de varios tipos a la vez (se denomina herencia múltiple)
 - Cuando se define un tipo de objeto se puede heredar de otro tipo. Si no se especifica otra cosa, se hereda de Object
 - Los tipos de datos se pueden usar como variables en PLSQL, pero no se pueden guardar en tablas
 - Se pueden guardar en tablas, ya que una columna de tabla puede ser un tipo de objeto

31. En Java, para procesar un fichero que no está en UTF8 (por ejemplo, en ISO-8859-X)
- En java solo pueden procesarse ficheros en UTF8
 - En java solo pueden procesarse ficheros en UTF8 y otras codificaciones compatibles Unicode, como ASCII
 - Es necesario crear un `DataInput` que lea del fichero, y sobre él un `DataInputReader` especificando el charset
 - Es necesario crear un `InputStream` que lea del fichero, y sobre él un `InputStreamReader` especificando el charset
32. Un `ResultSet` de Java
- Permite avanzar por sus filas, y recuperar el valor de cada columna por su índice
 - Permite acceder a cada fila con un índice, y recuperar el valor de cada columna por su índice. Es como un array bidimensional, pero los índices comienzan en 1.
 - Permite acceder a cada fila con un índice, y al recuperarla devuelve un `Object[]` (array de objetos)
 - Es una clase que hereda de `java.util.Set`, con los resultados de una consulta a base de datos. Se recupera cada fila por su clave (key)
33. Comparando un parseador DOM y uno SAX
- SAX utilizará más memoria que DOM en ficheros con muchos niveles de jerarquía (nodos hijos dentro de otros nodos hijos)
 - DOM solo detectará errores tras procesar todo el fichero, pero SAX va encontrando los errores mientras procesa el fichero
 - Puede haber parseadores DOM en todos los lenguajes (por ejemplo, javascript), pero solo se puede implementar SAX en lenguajes compilados como Java
 - Si los ficheros XML son muy grandes, SAX es una mejor opción (por el uso de memoria de DOM)
34. Las propiedades que publican los `JavaBean`
- Deben estar basadas en atributos públicos (`public`), para que los getters y setters puedan ser públicos, y así garantizar la accesibilidad.
 - Deben estar basadas en atributos privados (`private`), y los getters y setters deben ser también privados, para garantizar la encapsulación.
 - Deben ser tipos simples (`int`, `float`, cadenas...) o arrays de estos tipos. No pueden ser otros objetos.
 - Pueden ser indexadas, de forma que la propiedad se comporta como un array.
35. Hibernate permite realizar consultas
- Solo con HQL y SQL. Las de SQL se denominan "native queries"
 - Solo con SQL (native queries) y "named queries"
 - Solo con HQL (hibernate query language) y "named queries"
 - Solo con HQL. Si se desea, se puede recuperar su conexión JDBC subyacente y realizar consultas a partir de esa `Connection`.

DATOS DEL ALUMNO		FIRMA
APELLIDOS:		
Nombre:	D.N.I. N.I.E. o Pasaporte:	
		Fecha 10 mayo 2023

Preguntas XQuery

Todas las preguntas se refieren a un XML con esta estructura. Una misma asignatura puede impartirse por más de un profesor.

Cada pregunta incluye el XML que se espera como resultado de la consulta.

```
<?xml version="1.0" encoding="UTF-8"?>
<datos>

  <alumno id="001">
    <nombre>María Pérez</nombre>
    <email>maria.perez@example.com</email>
    <asignaturas>

      <asignatura codigo="MAT101">
        <nombre>Matemáticas I</nombre>
        <profesor>Dr. Juan García</profesor>
      </asignatura>

      <asignatura codigo="FIS101">
        <nombre>Física I</nombre>
        <profesor>Dra. Ana López</profesor>
      </asignatura>

    </asignaturas>
  </alumno>

  <alumno id="002">
    <nombre>Juan Rodríguez</nombre>
    <email>juan.rodriguez@example.com</email>
    <asignaturas>

      <asignatura codigo="MAT101">
        <nombre>Matemáticas I</nombre>
        <profesor>Dr. Juan García</profesor>
      </asignatura>

      <asignatura codigo="QUI101">
        <nombre>Química I</nombre>
        <profesor>Dra. Laura Fernández</profesor>
      </asignatura>

      <asignatura codigo="FIS101">
        <nombre>Física I</nombre>
        <profesor>Dra. Laura Fernández</profesor>
      </asignatura>

    </asignaturas>
  </alumno>
</datos>
```

36. Lista de personas (profesores y alumnos)

```
<?xml version="1.0" encoding="UTF-8"?>
<personas>
  <persona>María Pérez</persona>
  <persona>Dr. Juan García</persona>
  <persona>Dra. Ana López</persona>
  <persona>Juan Rodríguez</persona>
  <persona>Dra. Laura Fernández</persona>
</personas>
```

```
let $ps := //alumno/nombre|//profesor//nombre
for $p in distinct-values($ps)
  return <personas>{$p}</personas>
```

A

```
<personas>{
let $ps := //nombre[parent::alumno]|//profesor
for $p in distinct-values($ps)
  return <persona>{$p}</persona>
}
</personas>
```

B

```
<personas>{
  let $as := //alumno
  let $ps := //profesor
  for $p in distinct-values($ps) and distinct-values($as)
    return <persona>{$p}</persona>
}</personas>
```

C

```
<personas>{
let $ps := //nombre[child::alumno] | //profesor
for $p in distinct-values($ps)
  return <persona>{$p}</persona>
}
</personas>
```

D

DATOS DEL ALUMNO		FIRMA
APELLIDOS:		
Nombre:	D.N.I. N.I.E. o Pasaporte:	
		Fecha 10 mayo 2023

37. Lista con los emails de los alumnos de cada profesor

```
<?xml version="1.0" encoding="UTF-8"?>
<profesores>
  <profesor nombre="Dr. Juan García">
    <email>maria.perez@example.com</email>
    <email>juan.rodriguez@example.com</email>
  </profesor>
  <profesor nombre="Dra. Ana López">
    <email>maria.perez@example.com</email>
  </profesor>
  <profesor nombre="Dra. Laura Fernández">
    <email>juan.rodriguez@example.com</email>
  </profesor>
</profesores>
```

```
<profesores> {
let $ps := /datos/alumno/asignaturas/asignatura/profesor
for $p in distinct-values($ps)
return <profesor nombre="{ $p }">
{
  /datos/alumno[asignaturas/asignatura/profesor/text() = $p]/email
}
</profesor>
}</profesores>
```

A

```
<profesores> {
for $a in /datos/alumno
return <profesor nombre="{ $p }">
{
  if ($a/asignaturas/asignatura/profesor = $p) then
    return $a/email
  else ()
}
</profesor>
}</profesores>
```

B

```
<profesores> {
for $a in /datos/alumno
for $p in //profesor
return if ($a/asignaturas/asignatura/profesor = $p) then
  <profesor>{$a/email}</profesor>
else ()
}</profesores>
```

C

```
<profesores> {
for $a in /datos/alumno
for $p in distinct-values(//profesor)
return if ($a/asignaturas/asignatura/profesor = $p) then
  <profesor>{$a/email}</profesor>
else ()
}</profesores>
```

D

38. Número de alumnos que conoce cada profesor, agrupando todos los alumnos de todas sus materias

```
<?xml version="1.0" encoding="UTF-8"?>
<profesores>
  <profesor num-alumnos="2" nombre="Dr. Juan García"/>
  <profesor num-alumnos="1" nombre="Dra. Ana López"/>
  <profesor num-alumnos="1" nombre="Dra. Laura Fernández"/>
</profesores>
```

```
<profesores> {
  for $profesor in distinct-values(/datos/alumno/asignaturas/asignatura/profesor)
    for $alumno in distinct-values(/datos/alumno[asignaturas/asignatura/profesor = $profesor])
      return if (contains($alumno/*/profesor, $profesor) then
        <profesor nombre="{ $profesor}" num-alumnos="{count($alumnos)}"/>
      else ()
  }</profesores>
```

A

```
<profesores> {
  for $alumno := /datos/alumno
    let $profesor = $alumnos/asignaturas/asignatura/profesor[ancestor = $alumno ]
    return if (count($profesor) > 0 ) then
      <profesor nombre="{ $profesor}" num-alumnos="{count($alumno)}"/>
    else
      ()
}</profesores>
```

B

```
for $profesor in distinct-values(/datos/alumno/asignaturas/asignatura/profesor)
  let $alumnos := distinct-values(/datos/alumno[*/profesor = $profesor])
  return <profesores>{
    <profesor nombre="{ $profesor}" num-alumnos="{count($alumnos)}"/>
  }</profesores>
```

C

```
<profesores> {
  for $profesor in distinct-values(/datos/alumno/asignaturas/asignatura/profesor)
    let $alumnos := /datos/alumno[asignaturas/asignatura/profesor = $profesor]
    return <profesor nombre="{ $profesor}" num-alumnos="{count($alumnos)}"/>
}</profesores>
```

D

DATOS DEL ALUMNO		FIRMA
APELLIDOS:		
Nombre:	D.N.I. N.I.E. o Pasaporte:	
		Fecha 10 mayo 2023

39. Listado de identificadores de alumno por profesor

```
<?xml version="1.0" encoding="UTF-8"?>
<profesores>
  <profesor nombre="Dr. Juan García">
    <alumno id="001"/>
    <alumno id="002"/>
  </profesor>
  <profesor nombre="Dra. Ana López">
    <alumno id="001"/>
  </profesor>
  <profesor nombre="Dra. Laura Fernández">
    <alumno id="002"/>
  </profesor>
</profesores>
```

```
<profesores> {
  for $profesor in distinct-values(/datos/alumno/asignaturas/asignatura/profesor)
  return <profesor nombre="{ $profesor }">
    <alumnos> {
      string-join(/datos/alumno[asignaturas/asignatura/profesor = $profesor]/@id, ', ')
    }</alumnos>
  </profesor>
}</profesores>
```

A

```
<profesores> {
  for $profesor in //profesor
  return <profesor nombre="{ $profesor }">
    <alumnos> {
      string-join(/datos/alumno[//profesor = $profesor]/@id, ', ')
    }</alumnos>
  </profesor>
}</profesores>
```

B

```
<profesores> {
  for $profesor in distinct-values(/datos/alumno/asignaturas/asignatura/profesor)
  return <profesor nombre="{ $profesor }">
    {
      for $alumno in /datos/alumno[asignaturas/asignatura/profesor = $profesor]
      return <alumno id="{ $alumno/@id }"/>
    }
  </profesor>
}</profesores>
```

C

```
<profesores>{
  for $profesor in distinct-values(/datos/alumno/asignaturas/asignatura/profesor)
  for $alumno in /datos/alumno[descendant::profesor = $profesor]
  return <profesor nombre="{ $profesor }">
    <alumnos>
      <alumno id="{ $alumno/@id }"/>
    </alumnos>
  </profesor>
}
</profesores>
```

D

40. Lista de alumnos de cada clase. Una clase es una asignatura impartida por un profesor.

```
<?xml version="1.0" encoding="UTF-8"?>
<grupo asignatura="MAT101" profesor="Dr. Juan García">
  <nombre>María Pérez</nombre>
  <nombre>Juan Rodríguez</nombre>
</grupo>
<grupo asignatura="FIS101" profesor="Dra. Ana López">
  <nombre>María Pérez</nombre>
</grupo>
<grupo asignatura="FIS101" profesor="Dra. Laura Fernández">
  <nombre>Juan Rodríguez</nombre>
</grupo>
<grupo asignatura="QUI101" profesor="Dra. Laura Fernández">
  <nombre>Juan Rodríguez</nombre>
</grupo>
```

```
for $p in distinct-values(//profesor)
  for $a in distinct-values(//asignatura/@codigo)
    let $grupo := //asignatura[@codigo = $a] | //profesor[text() = $p]
    return
    if (count($alumnos) > 0 ) then
      <grupo profesor="{ $p}" asignatura="{ $a}"> { $grupo } </grupo>
    else ()
```

A

```
for $p in distinct-values(//profesor)
  for $a in distinct-values(//asignatura/@codigo)
    let $alumnos := /datos/alumno[asignaturas/asignatura[@codigo = $a and profesor/text() = $p]
    return
    <grupo profesor="{ $p}" asignatura="{ $a}"> { $alumnos/nombre } </grupo>
```

B

```
for $p in //profesor, $a in //asignatura/@codigo
  let $grupo := /datos/alumno[asignaturas/asignatura[@codigo = $a and profesor/text() = $p]]
  return
  if (count($grupo) > 0 ) then
    <grupo profesor="{ $p}" asignatura="{ $a}"> { $grupo/nombre } </grupo>
  else ()
```

C

```
for $p in distinct-values(//profesor)
  for $a in distinct-values(//asignatura/@codigo)
    let $alumnos := //asignaturas/asignatura[@codigo = $a and profesor/text() = $p]/../..
    return
    if (count($alumnos) > 0 ) then
      <grupo profesor="{ $p}" asignatura="{ $a}"> { $alumnos/nombre } </grupo>
    else ()
```

D