A. Explorando

1. Explore el archivo usando un navegador y un editor.

¿Cuáles son las ventajas de uno u otro?

En el navegador se puede ver la estructura XML de mejor manera que en el editor

2. Describa la información que encuentra en el archivo.

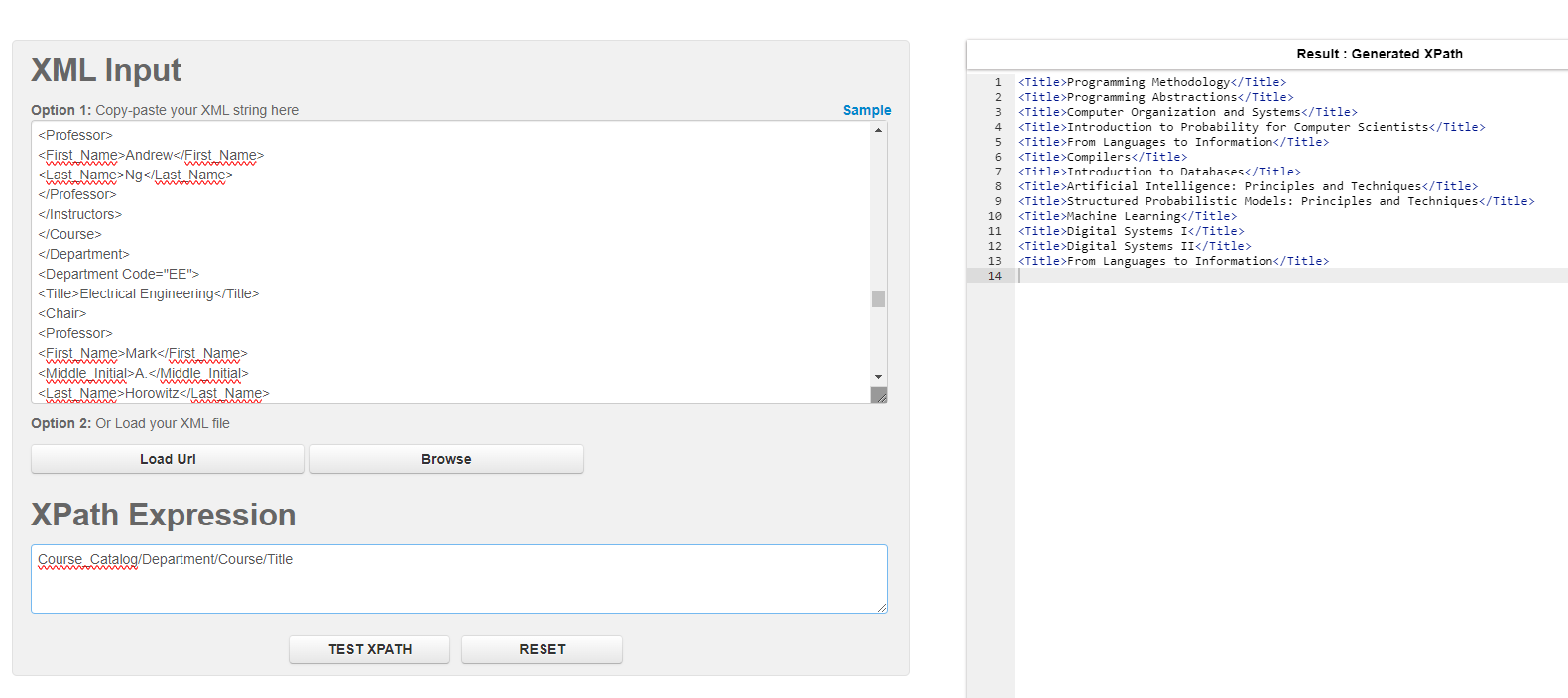
Encontramos un catalogo de departamentos, la cual cada uno contiene la informacion de profesores como decanos del departamento y cursos. Dentro de los cursos hay informacion de su identificador y capacidad, asi como su titulo, su descripcion, sus instructores y profesores.

B. Consultas iniciales

Implemente las siguientes consultas:

1. ¿De cuáles cursos se tiene información?

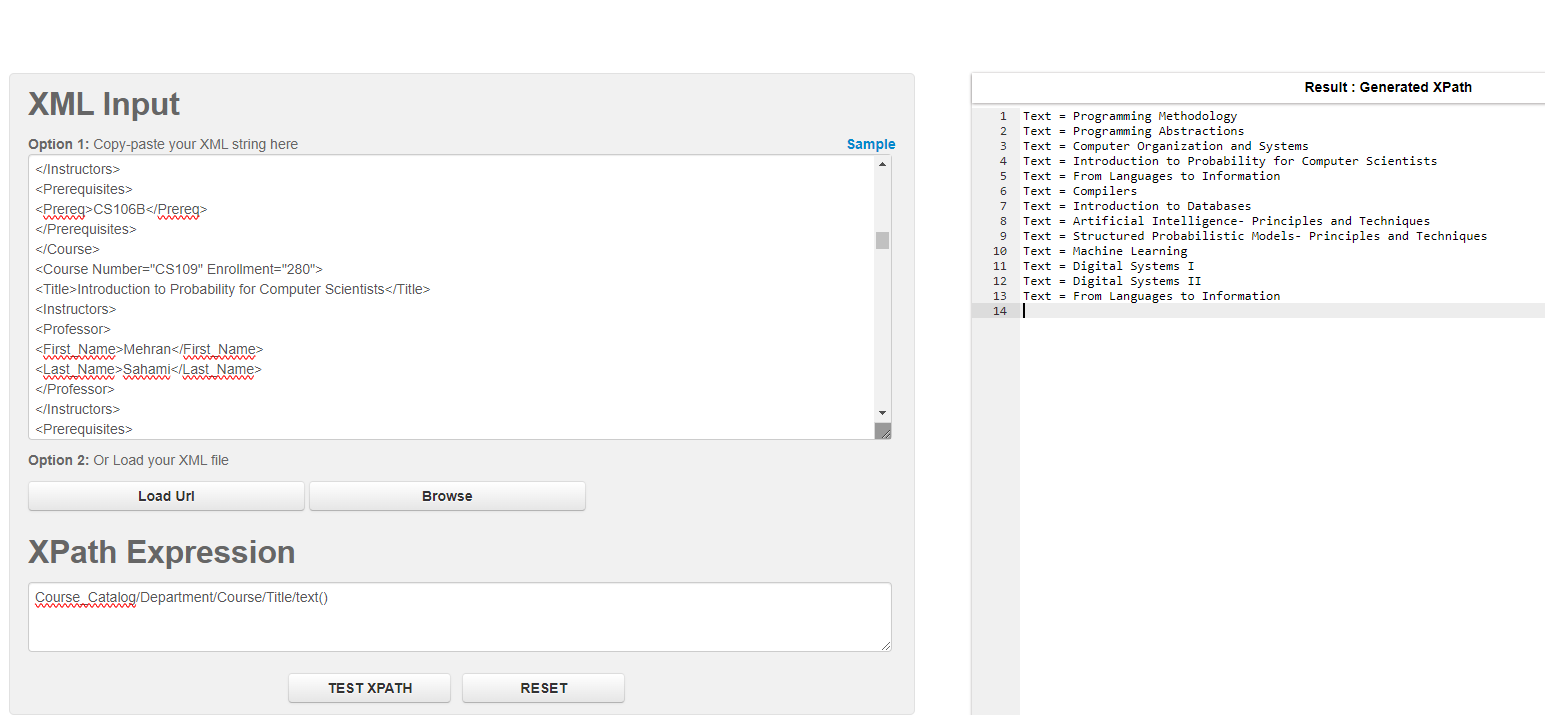
[a. Nombre con etiqueta]



Course\_Catalog/Department/Course/Title

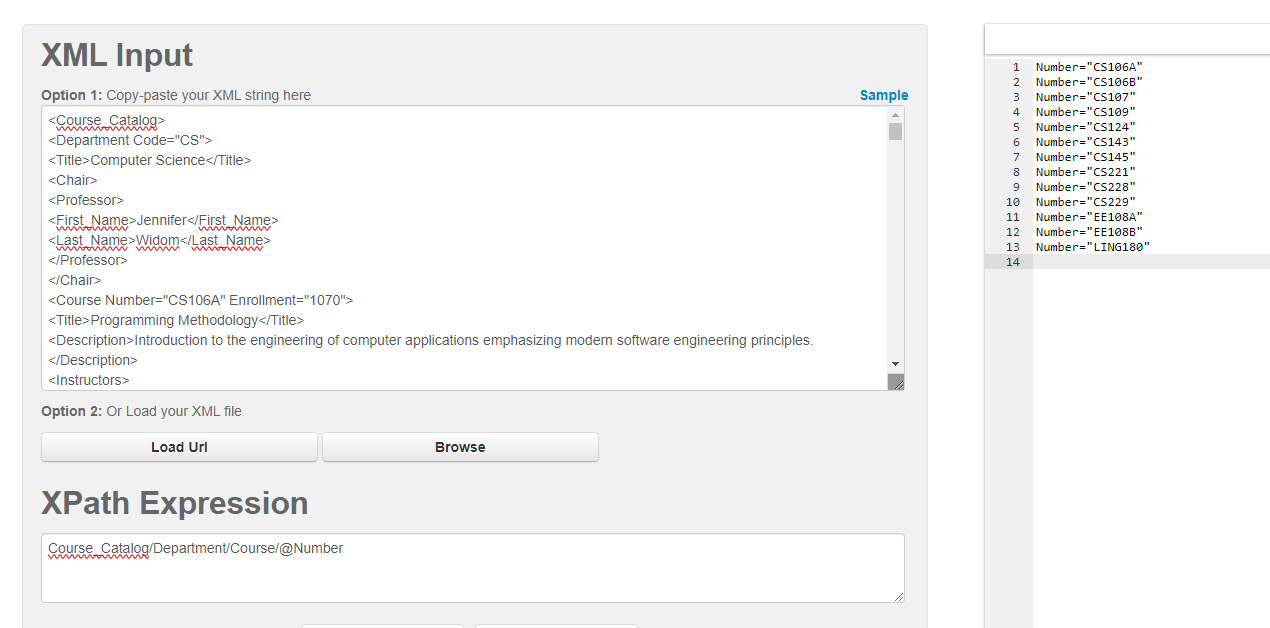
[b. Nombre sin etiqueta]

/Course\_catalog/Department/Course/Title/text()



[c. Número con atributo]

/Course\_Catalog/Department/Course/@Number

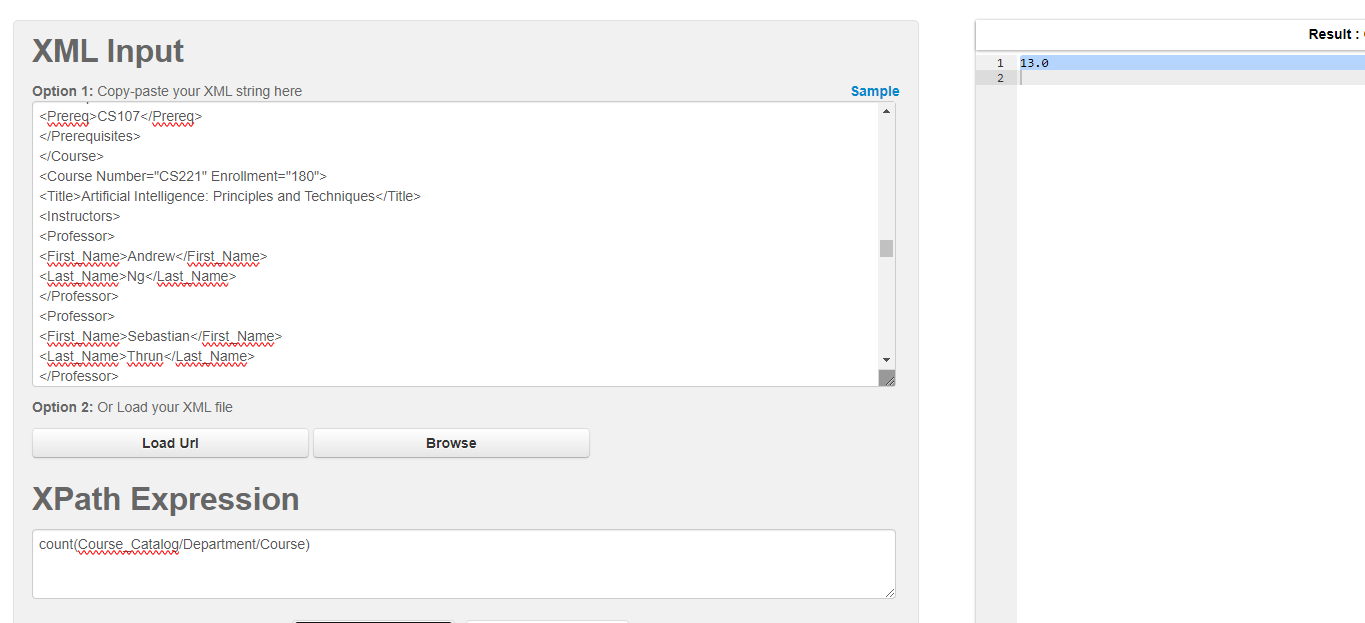


[d. Número sin atributo]

1. Esto no se aplica directamente en este caso, ya que el número del curso está representado como un atributo.

[e. Número total de cursos]

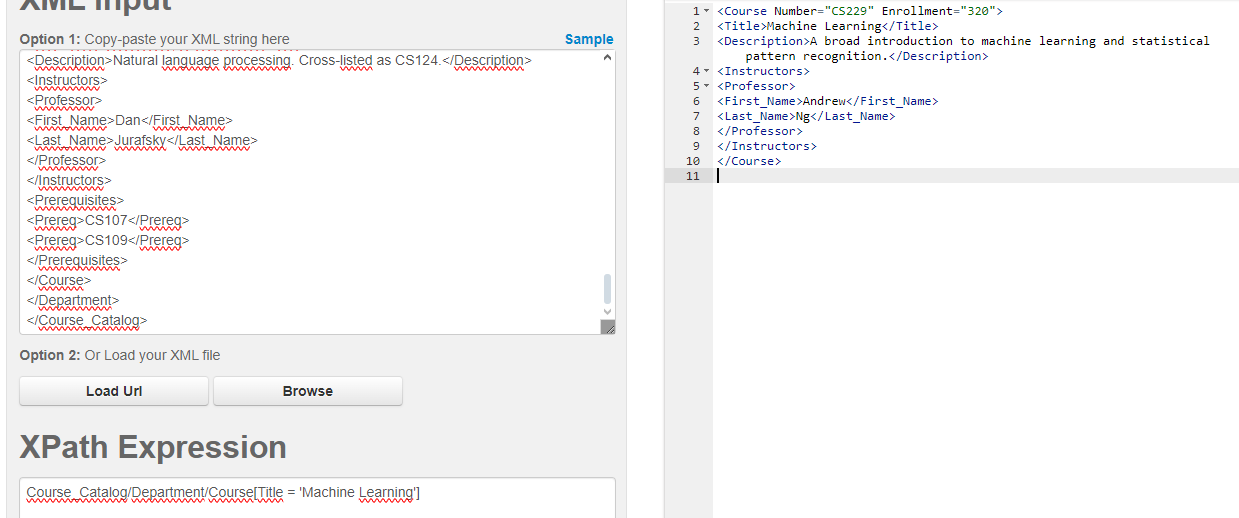
count(/Course\_Catalog/Department/Course/Title)



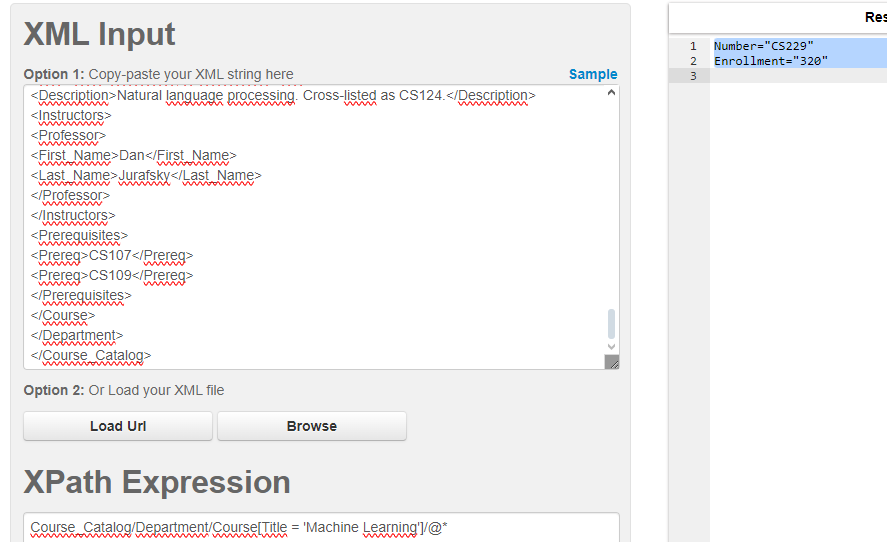
2. ¿Cuál es la información del curso Machine Learning?

[a. Toda la información]

/Course\_Catalog/Department/Course[Title = 'Machine Learning']

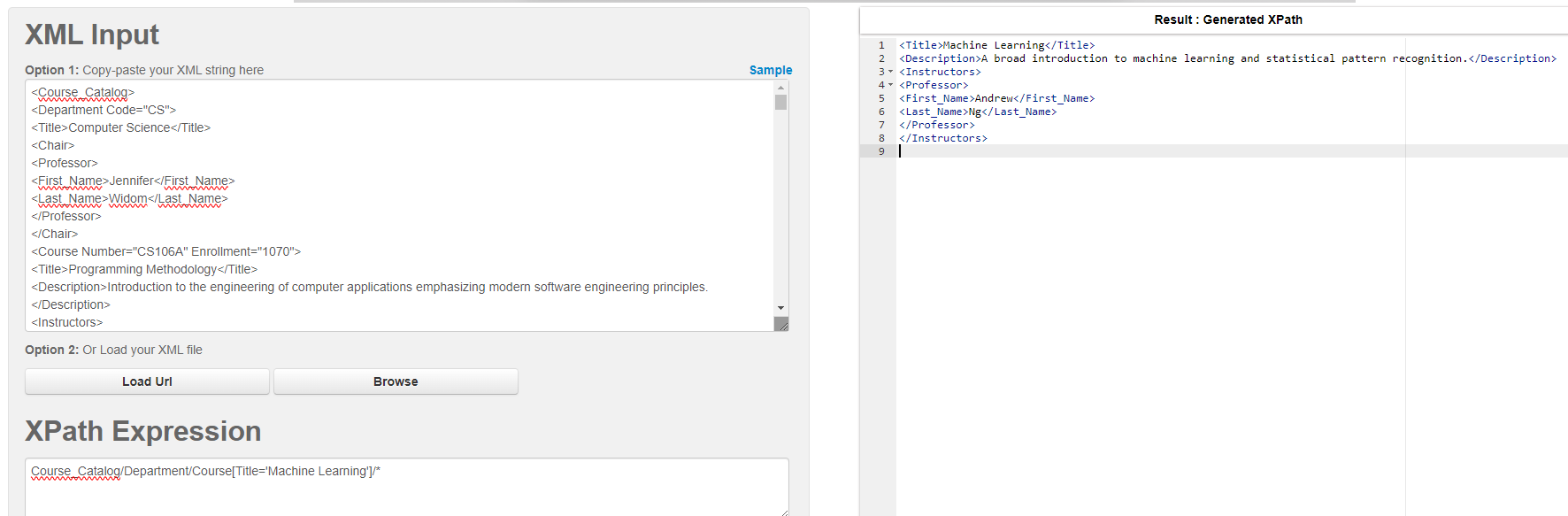


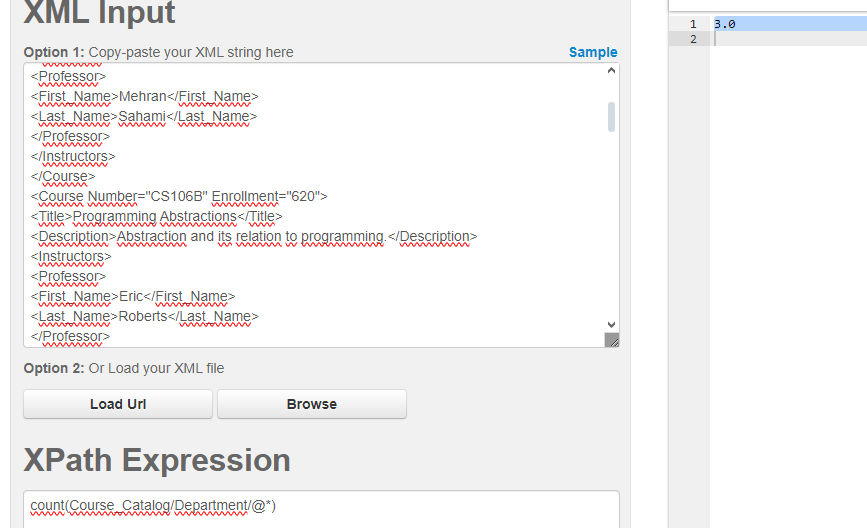
[b. Todos los atributos]

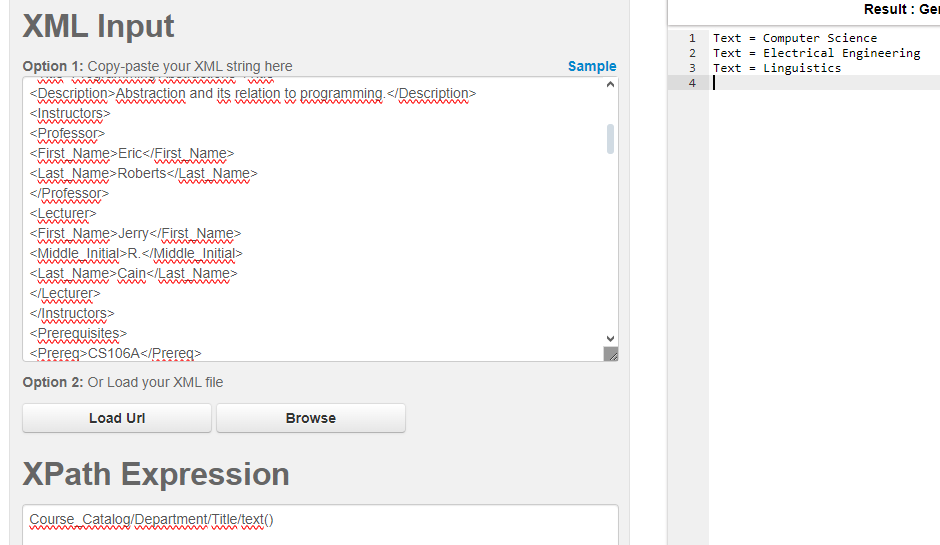
/Course\_Catalog/Department/Course[Title = 'Machine Learning']/@\*

[c. Todas las etiquetas]

/Course\_Catalog/Department/Course[Title = 'Machine Learning']/\*

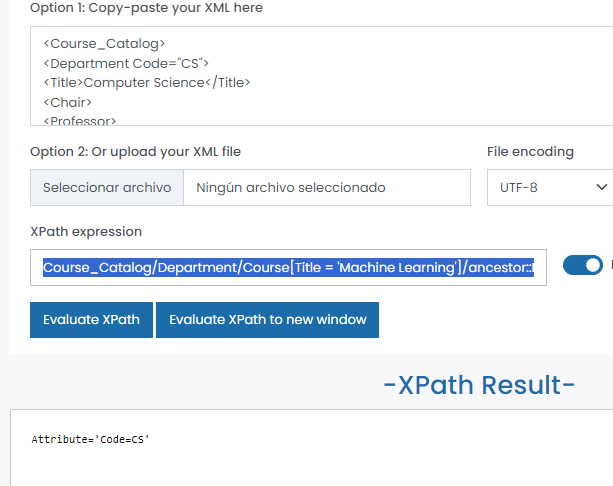


3. ¿Cuántos departamentos tenemos? ¿Cuales son?[Nombre]

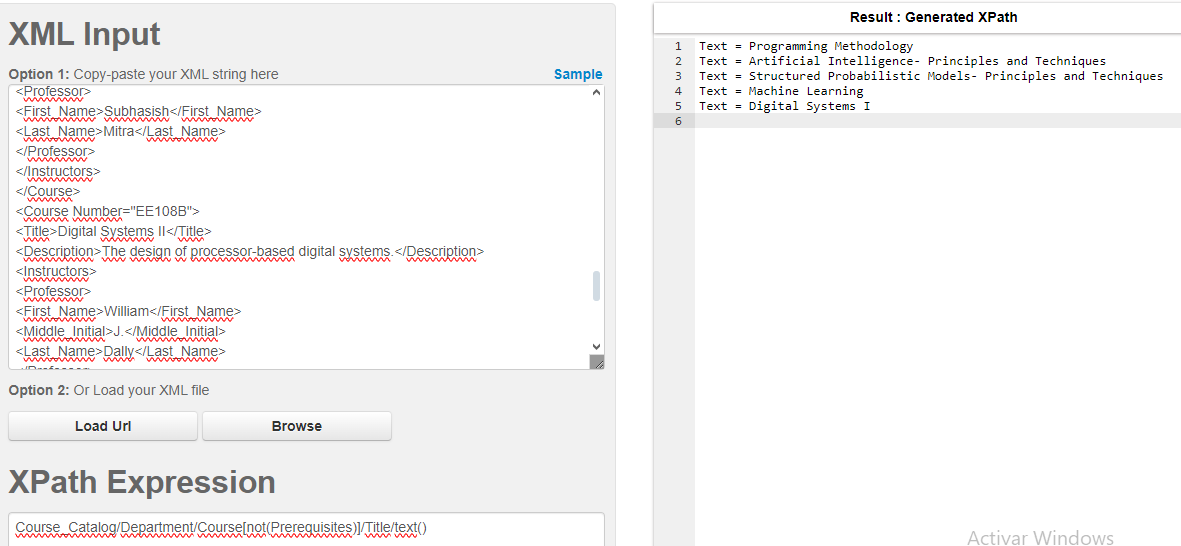


4. ¿A qué departemento pertenece el curso Machine Learning? [Codigo]

Course\_Catalog/Department/Course[Title = 'Machine Learning']/ancestor::Department/@Code



5. ¿Cuáles cursos son iniciales (no tienen prerrequisitos)? [Nombre]



/Course\_Catalog/Department/Course[not(Prerequisites)]/Title

6. ¿Cuáles cursos son finales (no son prerrequisitos de ninguno)? [Nombre]

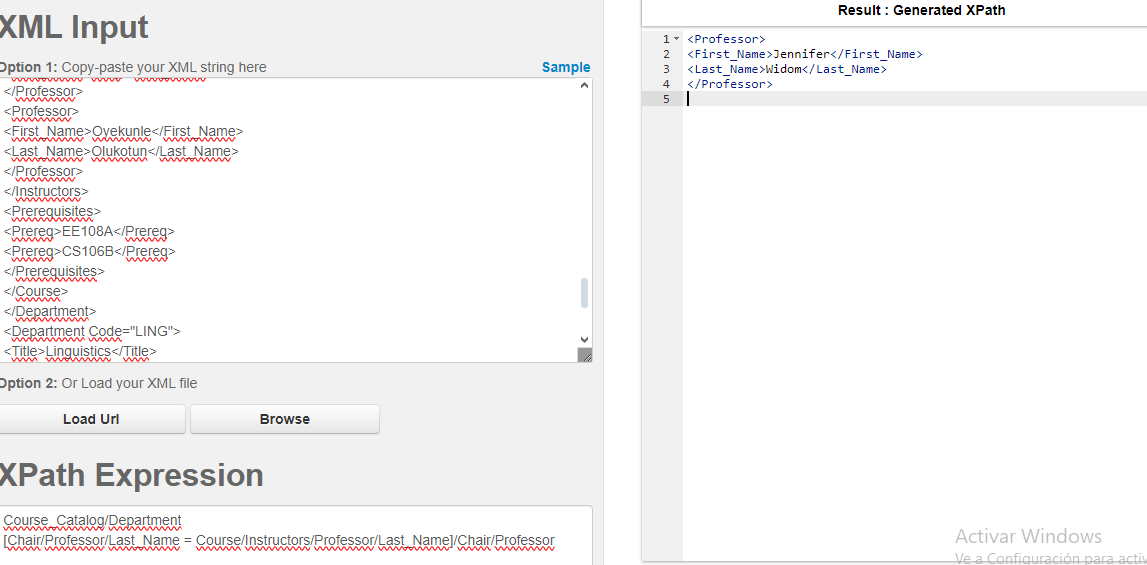
/Course\_Catalog/Department/Course[not(/Course\_Catalog/Department/Course/Prerequisites/Prereq/text() = @Number)]/Title



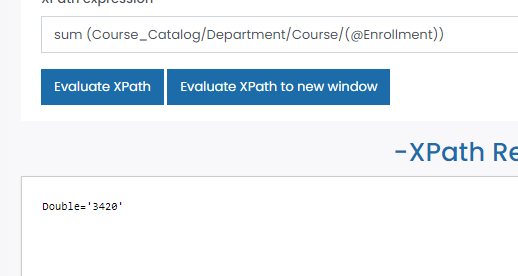
7. ¿Cuáles directores de departamento son profesores de cursos del departamento?[La información del director]

(Los profesores se pueden identificar por su apellido)

Course\_Catalog/Department [Chair/Professor/Last\_Name=Course/Instructors/Professor/Last\_Name]/Chair/Professor



8. ¿Cuál es el total de inscripciones de los cursos?

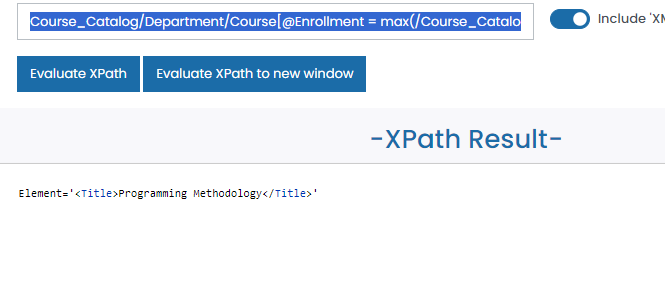


9. ¿De cuáles cursos no se conoce el número de inscritos? [Nombre]



10. ¿Cuál curso tiene el mayor número de inscritos? [Nombre]

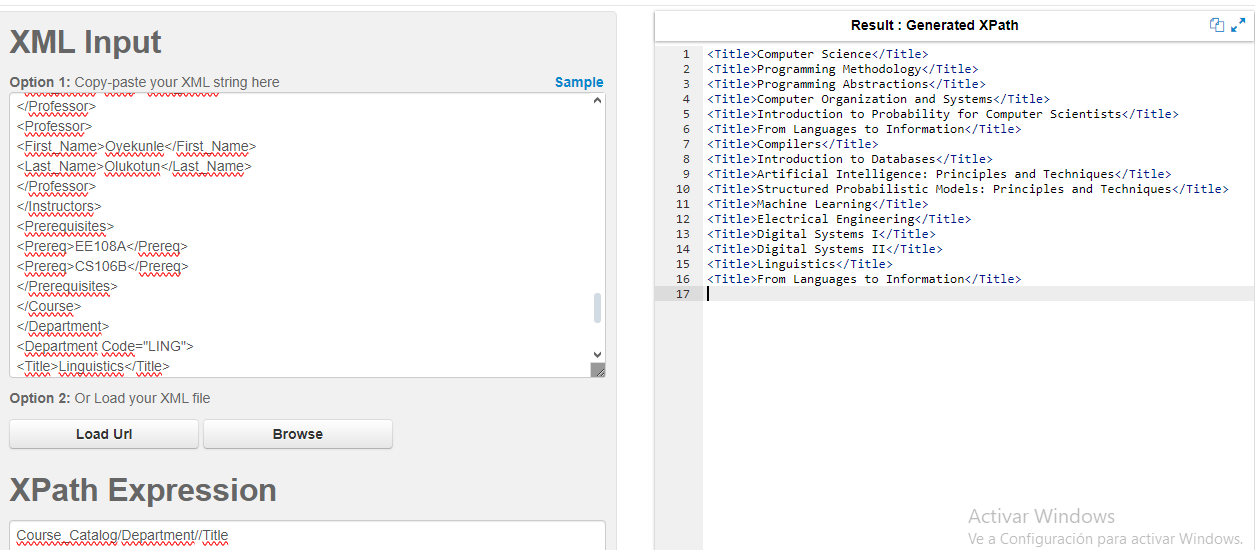
Course\_Catalog/Department/Course[@Enrollment = max(/Course\_Catalog/Department/Course/@Enrollment)]/Title



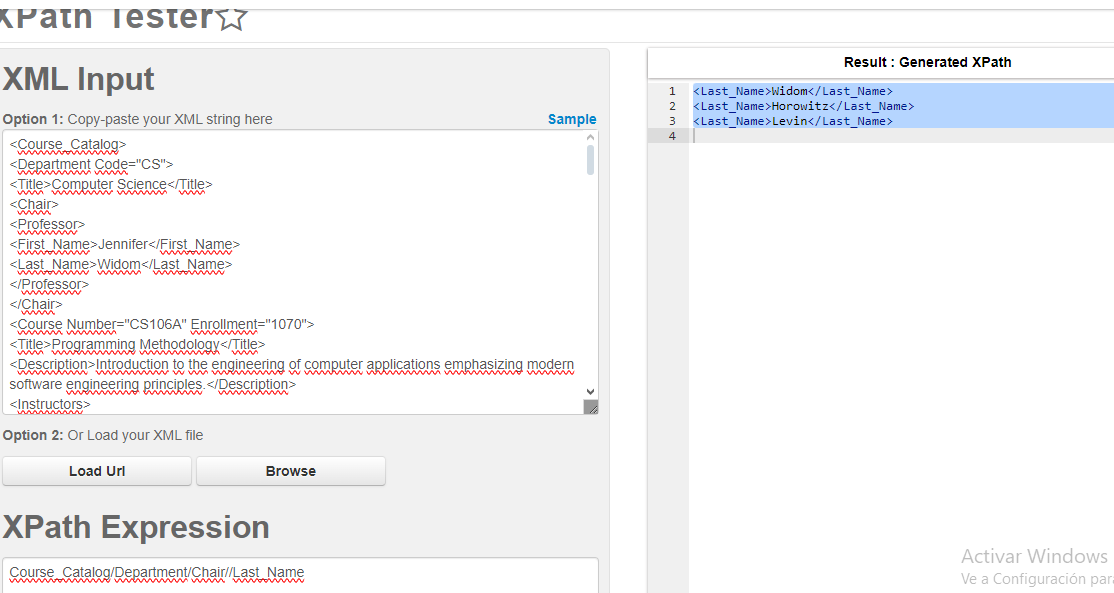
-----------------

B. Consultas quiz Standford

1. Return all Title elements (of both departments and courses). (16)



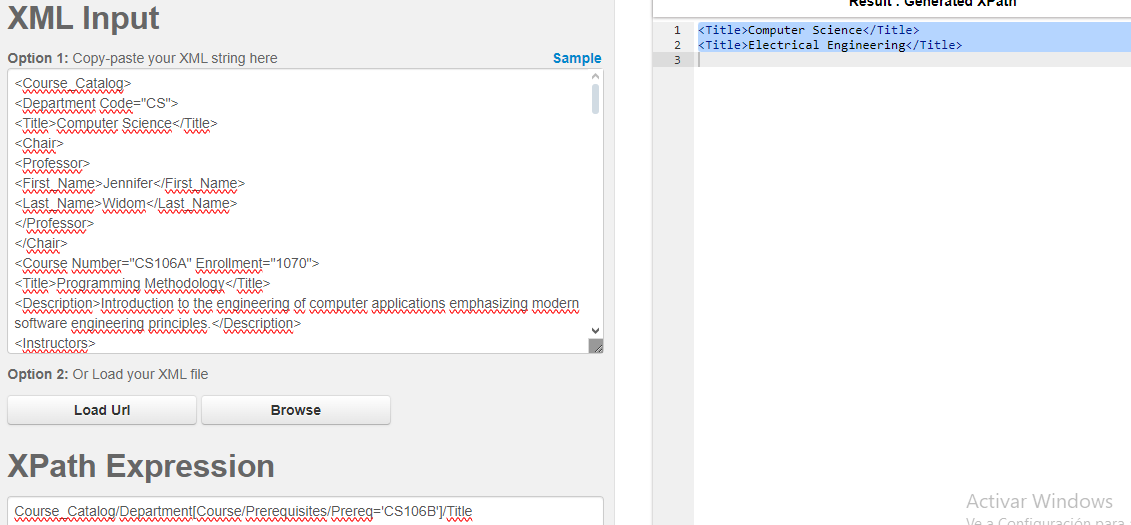
2. Return last names of all department chairs. (3)



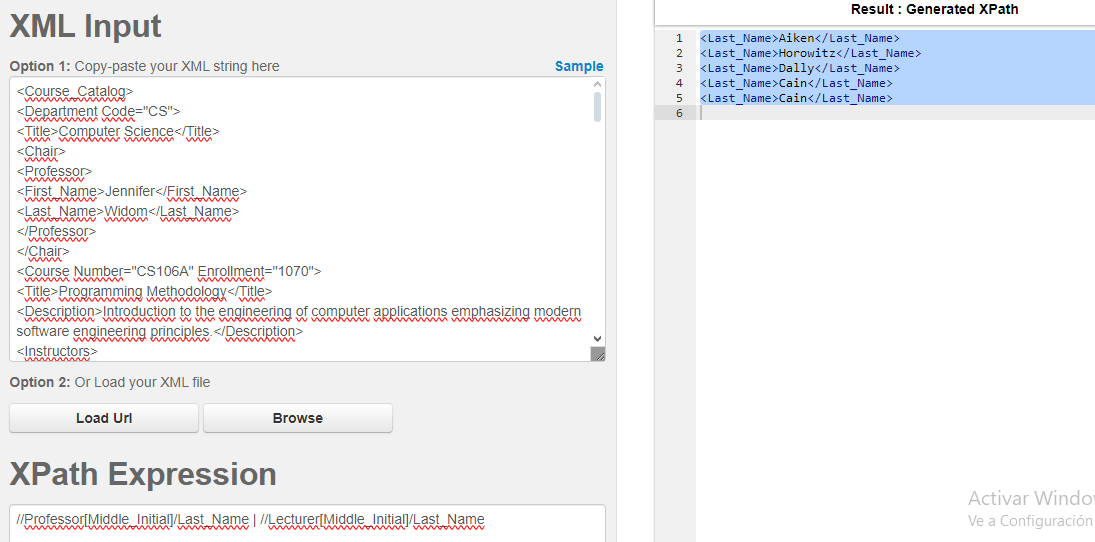
3. Return titles of courses with enrollment greater than 500. (2)



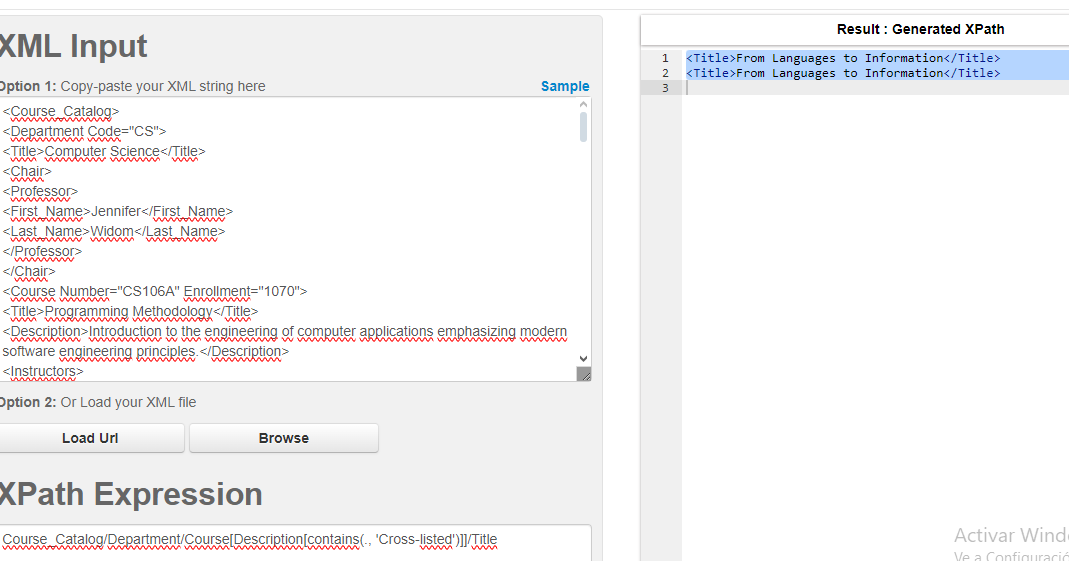
4. Return titles of departments that have some course that takes "CS106B" as a prerequisite. (2)



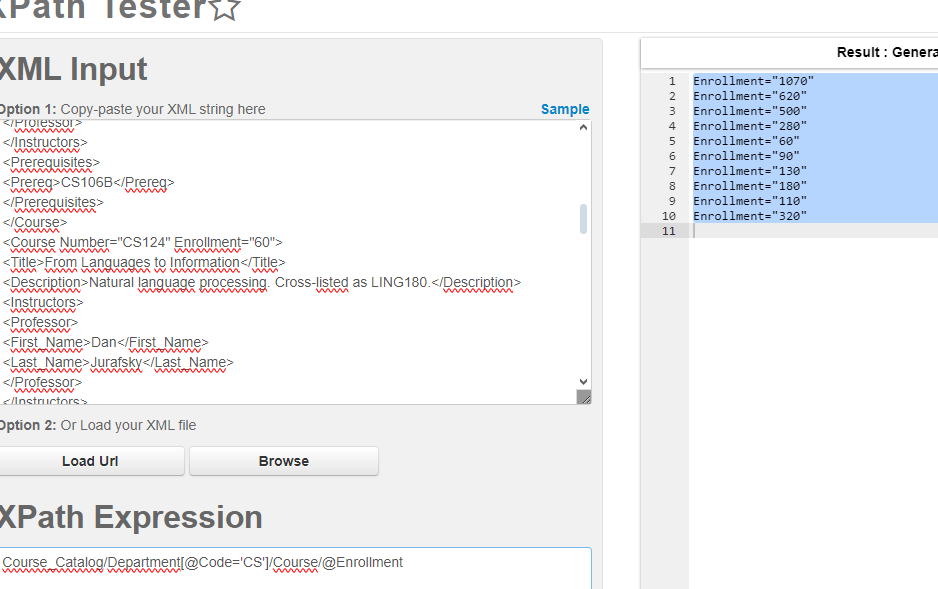
5. Return last names of all professors or lecturers who use a middle initial. Don't worry about eliminating duplicates. (5)



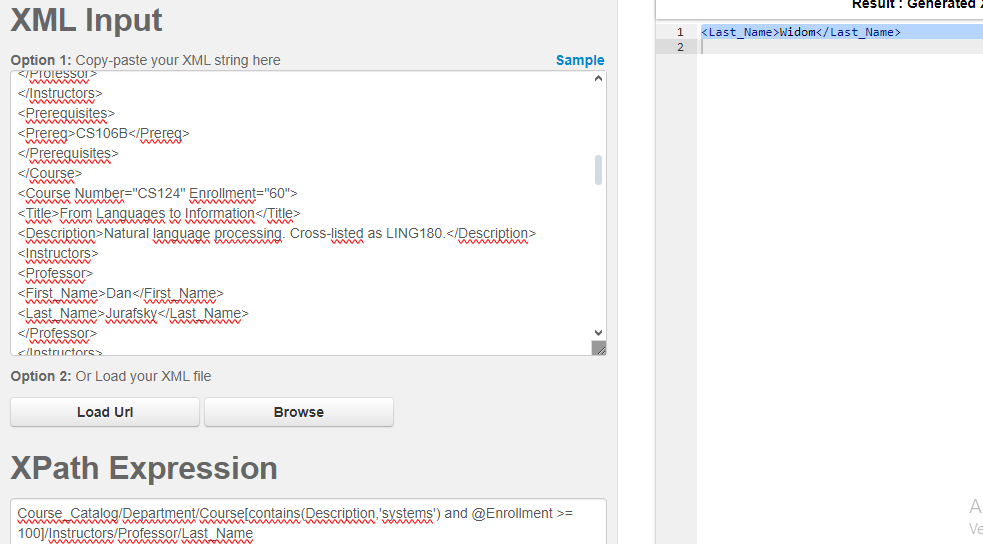
6. Return the title of courses that have a cross-listed course (i.e., that have "Cross-listed" in their description). (2)



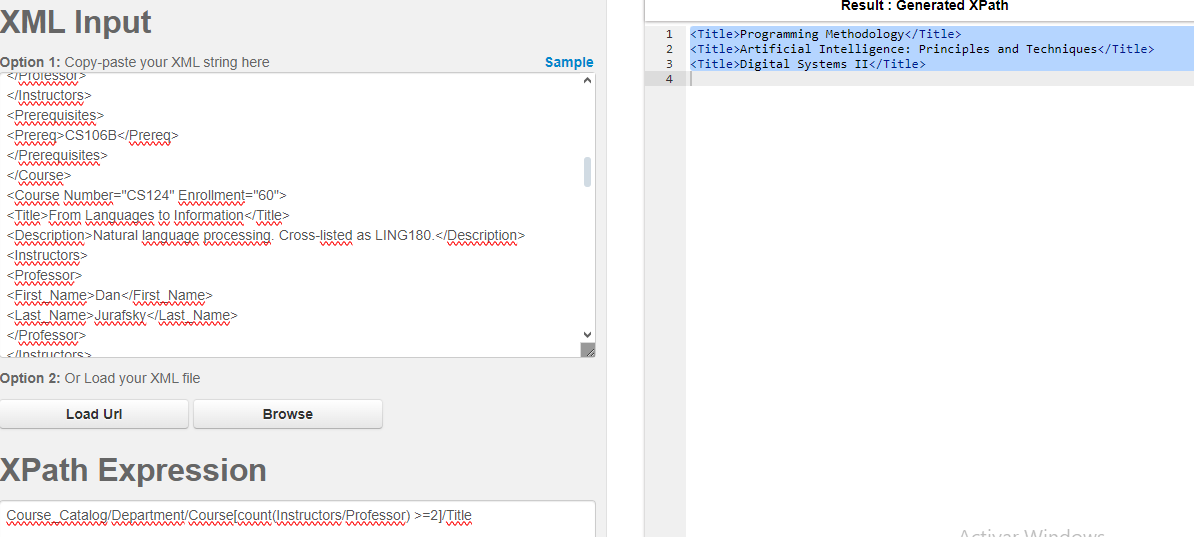
7. Return the enrollment of all courses in the CS department. (10)



8. Return last names of instructors teaching at least one course that has "system" in its description and enrollment greater than 100. (2)

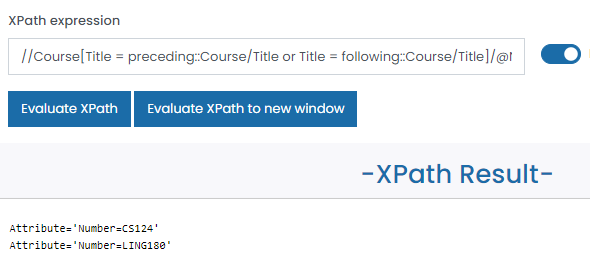


9. Return the title of the course with more than 2 instructors (\*)(1)

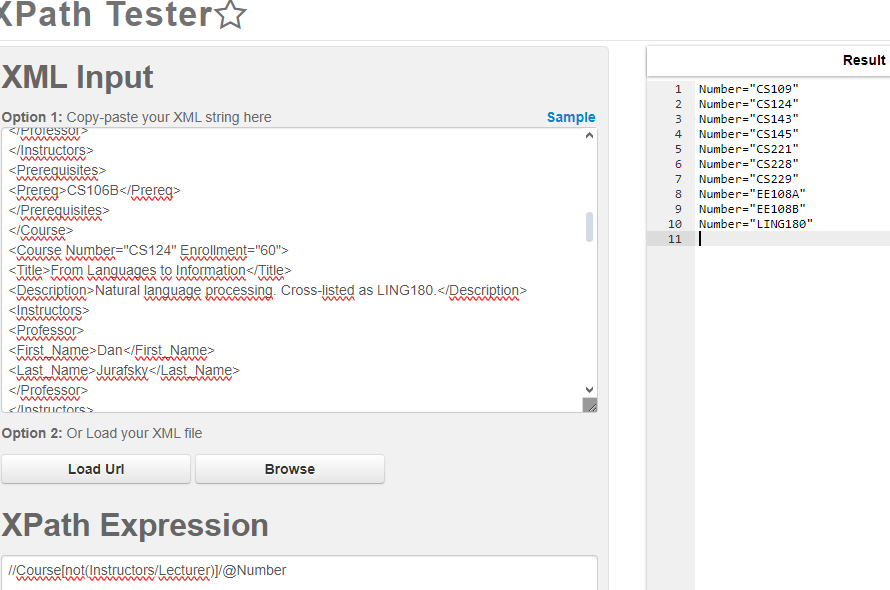


10. Return course numbers of courses that have the same title as some other course. (Hint: You might want to use the "preceding" and "following" navigation axes for this query, which were not covered in the video or our demo script; they match any preceding or following node, not just siblings.) (2)

//Course[Title = preceding::Course/Title or Title = following::Course/Title]/@Number

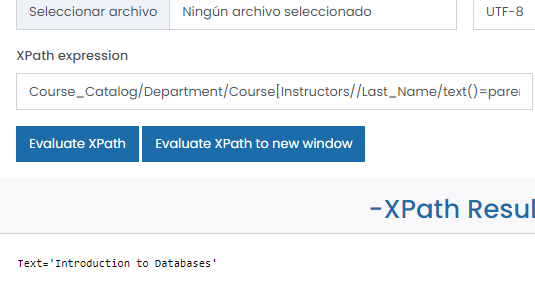


11. Return the courses numbers of courses that have no lecturers as instructors. (\*)(10)



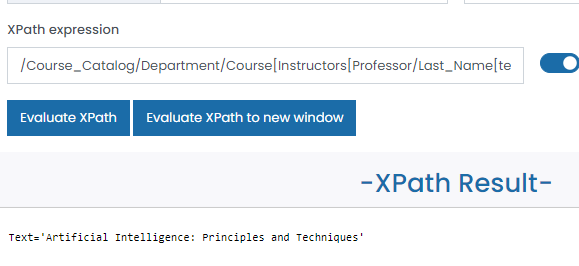
12. Return titles of courses taught by the chair of a department. For this question, you may assume that all professors have distinct last names. (1)

Course\_Catalog/Department/Course[Instructors//Last\_Name/text()=parent::\*/Chair/Professor/Last\_Name/text()]/Title/text()



13. Return titles of courses taught by a professor with the last name "Ng" and by a professor with the last name "Thrun". (\*)(1)

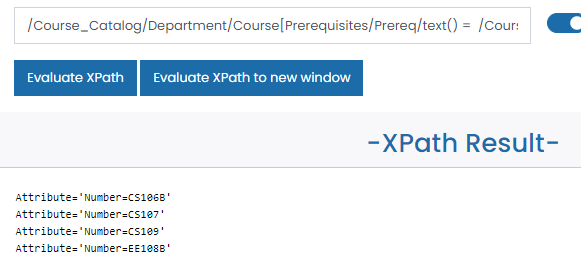
/Course\_Catalog/Department/Course[Instructors[Professor/Last\_Name[text()="Ng"] and Professor/Last\_Name[text()="Thrun"] ]]/Title/text()



14. Return course numbers of courses that have a course taught by Eric Roberts as a prerequisite. (4)

Return last names of instructors teachiing at least one course that has system in its description and enrollment greater than 100.(2)

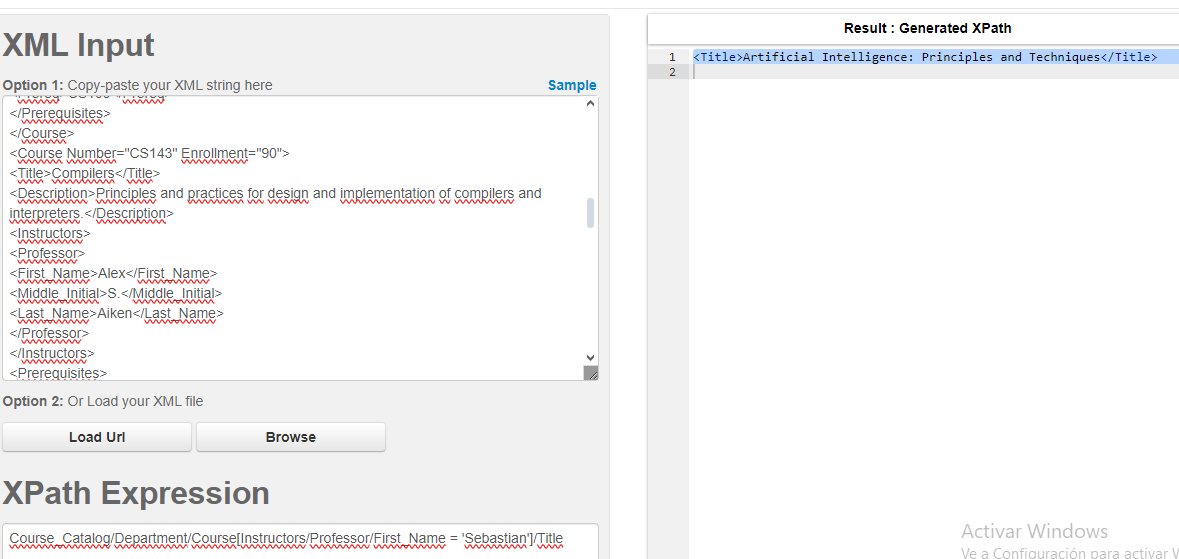
/Course\_Catalog/Department/Course[Prerequisites/Prereq/text() = /Course\_Catalog/Department/Course[Instructors/Professor[Last\_Name/text()="Roberts" and First\_Name/text()="Eric"]]/@Number]/@Number



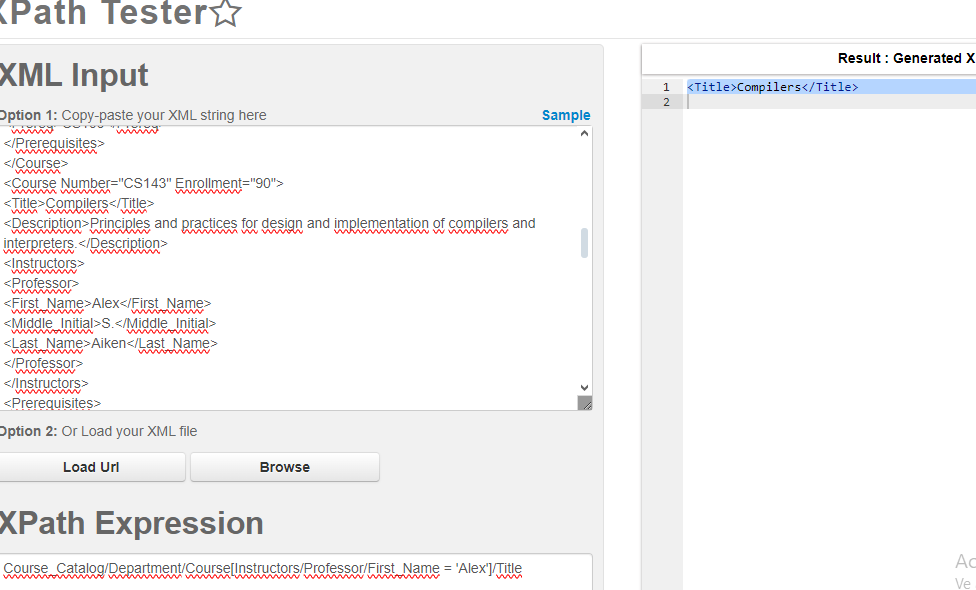
C. Consultas propias

Propongan e implementen cinco consultas propias

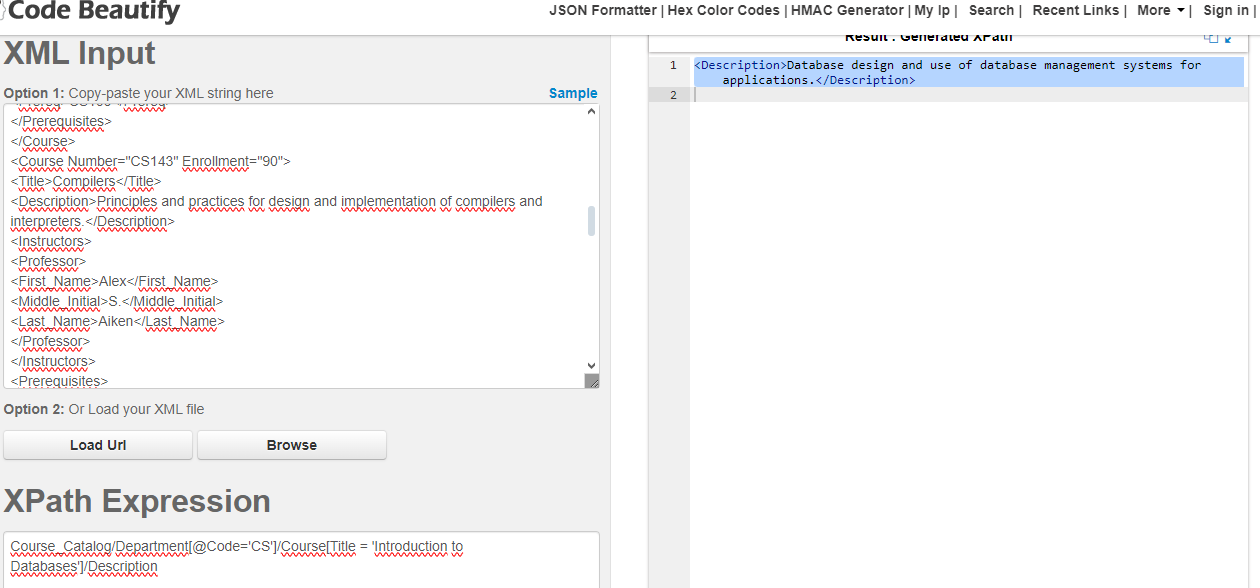
1. Course\_Catalog/Department/Course[Instructors/Professor/First\_Name = 'Sebastian']/Title



2. Course\_Catalog/Department/Course[Instructors/Professor/First\_Name = 'Alex']/Title

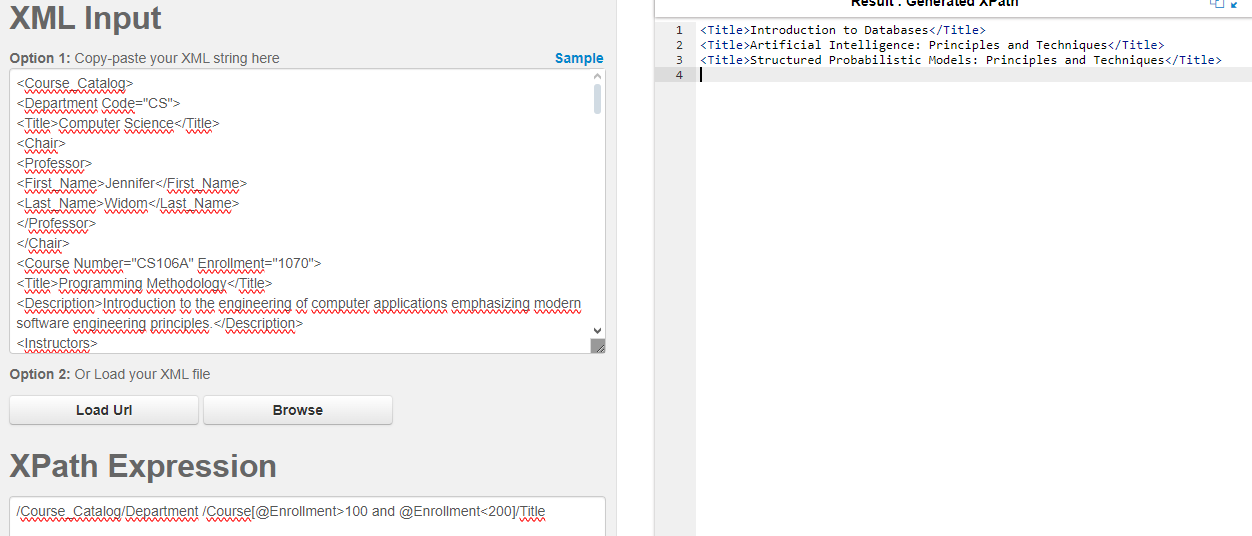


3. Course\_Catalog/Department[@Code='CS']/Course[Title = 'Introduction to Databases']/Description



4. //Middle\_Initial/text()



5. 

D. Esquema

DTD

<!DOCTYPE Course\_Catalog [

<!ELEMENT Course\_Catalog (Department+)>

<!ELEMENT Department (Title, Chair, Course+)>

<!ATTLIST Department Code CDATA #REQUIRED>

<!ELEMENT Title (#PCDATA)>

<!ELEMENT Chair (Professor)>

<!ELEMENT Professor (First\_Name, Middle\_Initial?, Last\_Name)>

<!ELEMENT First\_Name (#PCDATA)>

<!ELEMENT Middle\_Initial (#PCDATA)>

<!ELEMENT Last\_Name (#PCDATA)>

<!ELEMENT Course (Title, Description?, Instructors, Prerequisites?)>

<!ATTLIST Course Number CDATA #REQUIRED

Enrollment CDATA #IMPLIED>

<!ELEMENT Description (#PCDATA)>

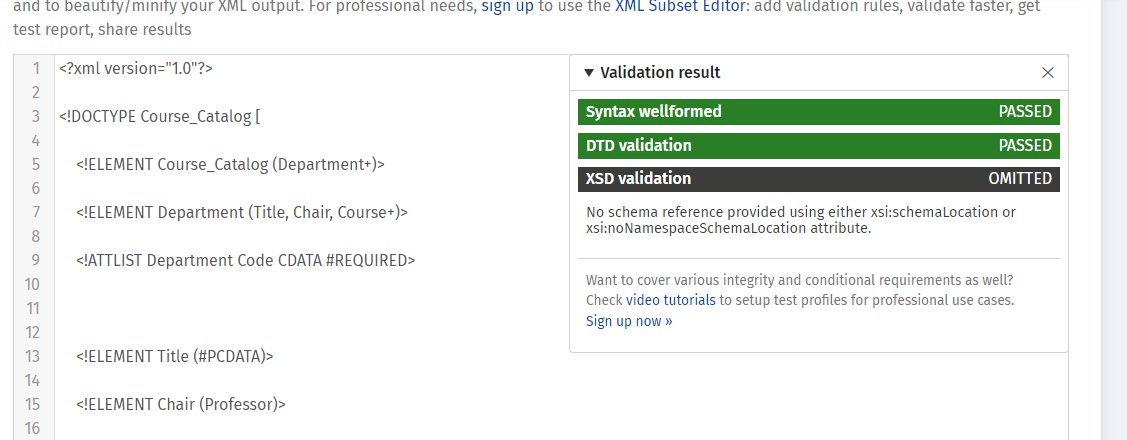
<!ELEMENT Instructors (Professor | Lecturer)+>

<!ELEMENT Lecturer (First\_Name, Middle\_Initial?, Last\_Name)>

<!ELEMENT Prerequisites (Prereq+)>

<!ELEMENT Prereq (#PCDATA)>

]>



E.

XML CON 3 NUEVOS DATOS

<Course\_Catalog>

<Department Code="CS">

<Title>Computer Science</Title>

<Chair>

<Professor>

<First\_Name>Jennifer</First\_Name>

<Last\_Name>Widom</Last\_Name>

</Professor>

</Chair>

<Course Number="CS230" Enrollment="150">

<Title>Deep Learning for Computer Vision</Title>

<Description>An introduction to deep learning techniques for computer vision applications.</Description>

<Instructors>

<Professor>

<First\_Name>Fei-Fei</First\_Name>

<Last\_Name>Li</Last\_Name>

</Professor>

</Instructors>

<Prerequisites>

<Prereq>CS229</Prereq>

</Prerequisites>

</Course>

<Course Number="CS231N" Enrollment="200">

<Title>Convolutional Neural Networks</Title>

<Description>Understanding convolutional neural networks for image processing and recognition.</Description>

<Instructors>

<Professor>

<First\_Name>Andrej</First\_Name>

<Last\_Name>Karpathy</Last\_Name>

</Professor>

</Instructors>

<Prerequisites>

<Prereq>CS230</Prereq>

</Prerequisites>

</Course>

<Course Number="CS240" Enrollment="100">

<Title>Advanced Operating Systems</Title>

<Description>Design and implementation of advanced operating system features.</Description>

<Instructors>

<Professor>

<First\_Name>John</First\_Name>

<Last\_Name>Ousterhout</Last\_Name>

</Professor>

</Instructors>

<Prerequisites>

<Prereq>CS107</Prereq>

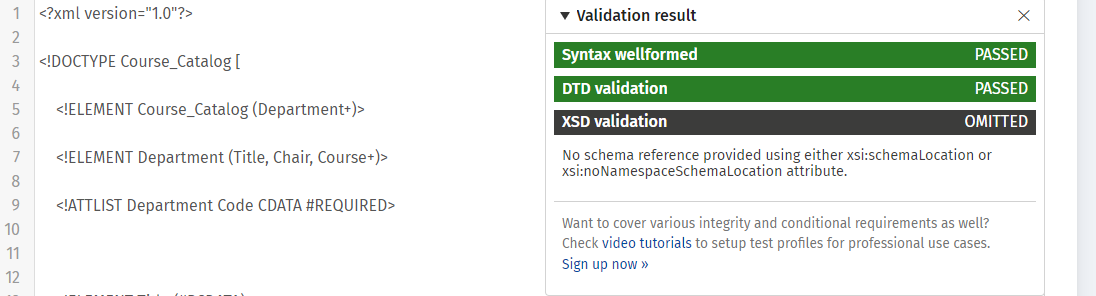
<Prereq>CS110</Prereq>

</Prerequisites>

</Course>

</Department>

</Course\_Catalog>



Segunda parte

A.

1. Proponga la estructura XML necesaria para tener la información de este atributo. DTD

y ejemplos XML OK y XML NoOK. Explique.

<!DOCTYPE Anexos [

<!ELEMENT Anexos (Anexo+)>

<!ELEMENT Anexo (Escritura?, CertificadoLibertad?)>

<!ATTLIST Anexo ID CDATA #REQUIRED Proyecto (Si | No) #IMPLIED>

<!ELEMENT Escritura (Numero, URL)>

<!ELEMENT CertificadoLibertad (Numero, URL)>

<!ELEMENT Numero (#PCDATA)>

<!ELEMENT URL (#PCDATA)>

]>

XML OK

<Anexos >

<Anexo ID="V001" Proyecto="No">

<Escritura>

<Numero>12345</Numero>

<URL>http://example.com/escritura12345.pdf</URL>

</Escritura>

<CertificadoLibertad>

<Numero>67890</Numero>

<URL>http://example.com/certlib67890.pdf</URL>

</CertificadoLibertad>

</Anexo>

<Anexo ID="V002" Proyecto="Si">

<Escritura>

<Numero>54321</Numero>

<URL>http://example.com/escritura54321.pdf</URL>

</Escritura>

</Anexo>

<Anexo ID="V003" Proyecto="No">

<Escritura>

<Numero>98765</Numero>

<URL>http://example.com/escritura98765.pdf</URL>

</Escritura>

<CertificadoLibertad>

<Numero>43210</Numero>

<URL>http://example.com/certlib43210.pdf</URL>

</CertificadoLibertad>

</Anexo>

</Anexos >

El XML sigue la estructura dada

XML NO OK

< Anexos >

<Anexo Proyecto="No">

<Escritura>

<Numero>12345</Numero>

</Escritura>

<CertificadoLibertad>

<Numero>67890</Numero>

<URL>http://example.com/certlib67890.pdf</URL>

</CertificadoLibertad>

</Anexo>

<Anexo ID="V002" Proyecto="Si">

<Escritura>

<Numero>54321</Numero>

<URL>http://example.com/escritura54321.pdf</URL>

</Escritura>

<Propietario>Juan Pérez</Propietario>

</Anexo>

<Anexo ID="V003">

<CertificadoLibertad>

<URL>http://example.com/certlib43210.pdf</URL>

</CertificadoLibertad>

</Anexo>

</ Anexos>

Primera Anexo: Falta el atributo ID, que es obligatorio según el DTD. Y falta el elemento URL dentro de la Escritura, lo cual viola la estructura definida.

Segunda Anexo: Contiene un elemento Propietario, que no está definido en el DTD, por lo tanto, es inválido.

Tercera Anexo: El elemento CertificadoLibertad está incompleto, ya que falta el subelemento obligatorio Numero.

3 Implemente la consulta Consultar ofertas sin certificado de libertad en pdf

**SELECT numero, direccion, tipoVivienda, costo, estado, idUsuario, codigoUbicacion**

**FROM ofertas**

**WHERE anexos IS NOT NULL**

**AND XMLExists('**

**declare namespace x="http://www.w3.org/2001/XMLSchema-instance";**

**let $cert := /Anexos/Anexo/CertificadoLibertad**

**return empty($cert) or not(ends-with($cert/URL, ".pdf"))**

**' PASSING anexos)**

**UNION**

**SELECT numero, direccion, tipoVivienda, costo, estado, idUsuario, codigoUbicacion**

**FROM ofertas**

**WHERE anexos IS NULL;**

4. Proponga otra nueva consulta que use Anexos (Diseño e implementación)

Consultar todas las ofertas que tengan una escritura y un certificado de libertad

**SELECT numero, fecha, direccion, tipoVivienda, costo, estado, idUsuario, codigoUbicacion**

**FROM ofertas**

**WHERE XMLExists( '/Anexos/Anexo[Escritura and CertificadoLibertad]' PASSING anexos );**

5. Extienda la información de este atributo (Anexos-DTD) y proponga una nueva

consulta que ilustre la pertinencia de la nueva información registrada en XML.

(Diseño e implementación)

<!DOCTYPE Anexos [

<!ELEMENT Anexos (Anexo+)>

<!ELEMENT Anexo (Escritura?, CertificadoLibertad?)>

<!ATTLIST Anexo ID CDATA #REQUIRED Proyecto (Si | No) #IMPLIED>

<!ELEMENT Escritura (Numero, URL, FechaEmision, Autoridad)>

<!ELEMENT CertificadoLibertad (Numero, URL, FechaEmision, Autoridad)>

<!ELEMENT Numero (#PCDATA)>

<!ELEMENT URL (#PCDATA)>

<!ELEMENT FechaEmision (#PCDATA)>

<!ELEMENT Autoridad (#PCDATA)>

]>

Consulta:

Consultar todas las ofertas con un certificado de libertad con una fecha de emision de como maximo 1 año de diferencia con la del sistema

**SELECT numero, fecha, direccion, tipoVivienda, costo, estado, idUsuario, codigoUbicacion**

**FROM ofertas**

**WHERE XMLExists('/Anexos/Anexo[CertificadoLibertad]' PASSING anexos)**

**AND TO\_DATE(**

**EXTRACTVALUE(anexos, '/Anexos/Anexo/CertificadoLibertad/FechaEmision'),**

**'YYYY-MM-DD'**

**) >= ADD\_MONTHS(SYSDATE, -12);**

---------------------------- otra opcion con el astah:

<!DOCTYPE Anexos[

<!ELEMENT Anexos (Anexo\*)>

<!ELEMENT Anexo (Escritura+ , certificado\_libertad?)>

<!ELEMENT Escritura (Disponible | No\_Disponible)>

<!ELEMENT Disponible (Nombre, Numero, Url)>

<!ELEMENT Nombre (#PCDATA)>

<!ELEMENT Numero (#PCDATA)>

<!ELEMENT Url (#PCDATA)>

<!ELEMENT No\_Disponible (Nombre, Justificacion, Fecha\_Esperada?)>

<!ELEMENT Nombre (#PCDATA)>

<!ELEMENT Justificacion (#PCDATA)>

<!ELEMENT Fecha\_Esperada (#PCDATA)>

<!ELEMENT certificado\_libertad (numero, URL)>

<!ELEMENT numero (#PCDATA)>

<!ELEMENT URL (#PCDATA)>

]>

/Consultar ofertas sin certificado de libertad/

SELECT numero AS numero\_oferta, No\_disponibles.Nombre AS Nombre\_oferta

FROM Ofertas O, XMLTABLE('/Anexos/Anexo[not(certificado\_libertad)]'

PASSING O.Anexos

COLUMNS

Nombre VARCHAR2 (50) PATH 'Nombre',

Justificacion VARCHAR2 (100) PATH 'Justificacion'

)No\_disponibles;

xmlok

<?xml version = "1.0"?>

<Anexos>

<Anexo>

<Escritura>

<Disponible>

<Nombre>Anexo Disponible 2</Nombre>

<Numero>02</Numero>

<Url>anexodisponible2.com</Url>

</Disponible>

</Escritura>

<certificado\_libertad>

<numero>123</numero>

<URL>https://123</URL>

</certificado\_libertad>

</Anexo>

</Anexos>

<?xml version = "1.0"?>

<Anexos>

<Anexo>

<Escritura>

<No\_Disponible>

<Nombre>Anexo No disponible 1</Nombre>

<Justificacion>No disponible debido a retrasos</Justificacion>

<Fecha\_Esperada>2024-12-20</Fecha\_Esperada>

</No\_Disponible>

</Escritura>

</Anexo></Anexos>

----------------------------

RETROSPECTIVA

1. ¿Cuál fue el tiempo total invertido en el laboratorio por cada uno de ustedes?

(18 horas /Andres Cardozo)

(18 horas/Daniel Palacios)

2. ¿Cuál es el estado actual del laboratorio? ¿Por qué?

El estado actual del laboratorio es completo y confiamos en que este tenga una cantidad de errores mínima ya que al aun estar aprendiendo XML y SQL puede que hayamos cometido errores.

3. ¿Cuál consideran fue el mayor logro? ¿Por qué?

El mayor logro fue aprender a escribir XML y entenderlos con mayor claridad, saber estructurarlos y ser capaces de hacer consultas dentro del XML con XPATH aunque también haber sido capaces de implementarlo en el SQL y hacer también consultas ahí tomando datos del XML

4. ¿Cuál consideran que fue el mayor problema técnico? ¿Qué hicieron para

resolverlo?

El mayor problema técnico sentimos que es no haber tenido conocimiento previo de XML Y XPATH ya que la plataforma brindada de XPATH dada no funcionaba correctamente por lo que se tuvo que buscar otra forma de hacer las consultas por otra parte fue complejo hacer las consultas dentro de ORACLE debido a que nunca habíamos hecho algo similar por lo que fue bastante complejo y nos tocó buscar externamente material para solucionarlo

5. ¿Qué hicieron bien como equipo? ¿Qué se comprometen a hacer para mejorar los

resultados?

Siento que como equipo hicimos bien en preguntarle al otro integrante el progreso realizado por cada uno de los integrantes para confirmar resultados o respuestas, aunque nos comprometemos a tener una mejor organización de nuestro tiempo para realizar el laboratorio y tener una mejor comunicación para realizarlo de una manera más apropiada y organizada.