

**DECANATO DE INGENIERIA E INFORMATICA**

**Escuela de informática**

**Carrera Ingeniería de Software**

**Integrantes:**

Carlos Eduardo ~ 2016-2594

Emmanuel Valenzuela ~ 2016-2511

Jean Carlos González ~ 2016-2330

**Tema:**

Práctica 2

**Asignatura:**

Integración de Aplicaciones con Tecnología Propietaria

**Prof.:**

Lic. Juan Pablo Valdez

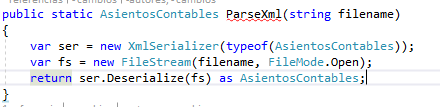
**Cuestionario:**

**1. ¿Cuáles tecnologías / formas puedo usar para generar XML desde .NET o para importarlo?**

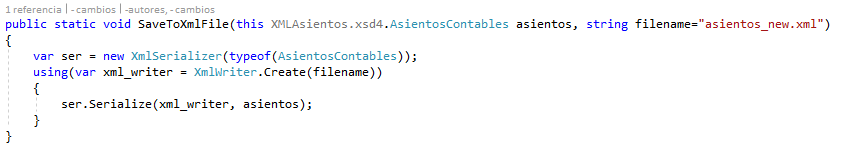
.NET cuenta con la clase XMLSerializer que permite tanto leer como generar archivos .xml, y convertirlos en objetos, y pasar objetos a xml, respectivamente. Está clase necesita del un modelo, el cual se puede crear con la herramienta xsd del Visual Studio Tools. También existen otras tecnologías tales como Language Integrated Query, XmlReader, XSLT, MSXML, y XmlLite.

Ej con XmlSerializer:

Para leer:



Para generar xml:

****

**2. ¿Cuáles tecnologías / formas puedo usar para generar XML desde Oracle o para importarlo?**

**Generar Xml:**

**Oracle Database** desde la versión **9iR2** nos proporciona un tipo de dato nativo llamado **XMLTYPE**; el cual nos permite albergar información en formato **XML** con soporte a **XSD** (esquema de validación), **XSLT** (transformaciones de documentos), XPATH, XQUERY, indexamiento y particionamiento a documentos XML.

**DBMS\_XMLGEN** es bueno para traducciones rápidas y sucias de datos a xml, pero nunca me gustó mucho porque tiene que pasar el SQL como una cadena.

**XMLDOM** es la forma más flexible de generar XML, especialmente si la estructura es más compleja o si se trata de una lógica iterativa. El principal inconveniente aquí es que es esencialmente un contenedor alrededor del **DOM** de **Java** donde la mayoría de los métodos aceptan una entrada DOMNode, pero **PL/SQL** no admite el **polimorfismo** directamente, por lo que terminas con una gran cantidad de conversiones explícitas entre **DOMElement** y **DOMNode** y viceversa, etc.

**Importar Xml:**

Puede cargar datos **XMLType** con **SQL \* Loader**, utilizando el método convencional o el método de ruta directa, independientemente de cómo se almacene (almacenamiento XML relacional de objetos o binario).

A partir de **Oracle9i**, la utilidad de exportación-importación y SQL \* Loader admiten **XMLType** como tipo de columna. A partir de **Oracle Database 10g**, SQL \* Loader también admite la carga de tablas XMLType.

**3. ¿Pueden los SGDB (MS SQL Server, Oracle, IBM UDB/ DB2, MySQL) interactuar con XML?**

Si,

En SQL Server, para **generarlo**: 2 maneras de hacerlo

SELECT \* from Saurabh.dbo.Customers FOR XML AUTO

SELECT \* from Saurabh.dbo.Customers FOR XML PATH

Para **importarlo**:

Este ejemplo muestra cómo insertar una fila en la tabla T. El valor de la columna **XML** se carga desde el archivo C:\\ MyFile\\xmlfile.xml como **CLOB**, y la columna de enteros recibe el valor 10.

INSERT INTO T SELECT 10, xCol

FROM (SELECT \*

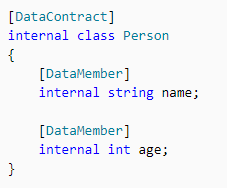
FROM OPENROWSET (BULK 'C:\MyFile\xmlfile.xml', SINGLE\_CLOB)

AS xCol) AS R(xCol);

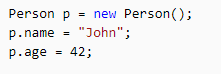
**4. ¿Cuáles tecnologías / formas puedo usar para generar JSON desde .NET o para importarlo?**

Para leer y generar json se puede usar la clase DataContractJsonSerializer y pasarle un modelo con el atributo DataContract.

Ej. Modelo:



Ej. Programa:





Generar json:



Leer json:



**5. ¿Cuáles tecnologías / formas puedo usar para generar JSON desde Oracle o para importarlo?**

**Para Generar o Exportar:**

Creando una representación JSON para cada cliente usando **JSON\_OBJECT**:

SELECT JSON\_OBJECT ('id2' VALUE cust\_id,

'name' VALUE (first ||' '|| last),

'joined' VALUE join\_date) FROM customers;

SELECT JSON\_ARRAYAGG(

JSON\_OBJECT('id' VALUE item\_id,

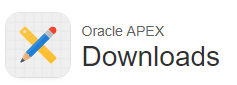
'name' VALUE name,

'quantity' VALUE stock\_quantity))

FROM items

WHERE stock\_quantity > 10;

**Para Importar:**



**Oracle** ha lanzado **APEX** 5.0 (abril de 2015). Con él obtienes acceso a una gran **API** para trabajar con **JSON**.

El ejemplo era en **Oracle11.2** y procesa cada json, desde objetos simples a muy complejos con múltiples arreglos APEX\_JSON

**Otras alternativas:**

La función **JSON\_OBJECT** de Oracle Database 12.2 es una excelente manera de convertir la salida de la tabla relacional en JSON; Y

El paquete **JSON\_EXT** funciona para analizarlo.

**6. ¿Pueden los SGDB (MS SQL Server, Oracle, IBM UDB/ DB2, MySQL) interactuar con JSON?**

Si, estos SGBD pueden exportar un conjunto de registros a un archivo XML.