



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

**ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA EN SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**

PERÍODO ACADÉMICO: 2017-A

ASIGNATURA: Aplicaciones en Ambientes Libres

PARALELO: GR2

PROFESOR: Ing. Luis Orquera

TEMA: Captura de facturas en formato XML

GRUPO: N°6

FECHA DE ENTREGA: 30/05/2017

INTEGRANTES: Daniel Beltrán
Adán Guzmán
Danilo Rojas

Objetivos:

- Capturar facturas XML para obtener información del cliente, de proveedores y gastos.
- Capturar datos de facturas XML y guardarlos en una base de datos liviana.
- Mostrar reportes de los gastos del cliente mediante las facturas ingresadas.

Introducción.

Tras descartar el proyecto anterior, AdminFacs, el grupo decidió rehacer el proyecto bajo los mismos lineamientos y funcionamientos, con lo cual, se planteó como primer avance del nuevo proyecto, cargar archivos en formato XML para, posteriormente, capturar todos los datos necesarios que identifiquen claramente los atributos de una factura. Esto implica un análisis profundo sobre los métodos y procesos que vamos a realizar al momento de la conversión de los archivos XML y la captura de los datos del mismo ya que ésta funcionalidad es una de las principales y se requiere necesariamente que su ejecución sea la correcta. Es por esto que vamos a analizar que librerías Java podemos usar en la aplicación y que beneficios obtendríamos para el propósito de nuestro programa de administración de facturas. Es necesario implementar junto al programa una base de datos liviana comúnmente conocida como base de datos tipo archivo, que permita guardar y manejar esos datos dentro del proyecto, ya que esto significa que el usuario podrá manejar el programa sin ningún altercado y solo se limite a usar el software.

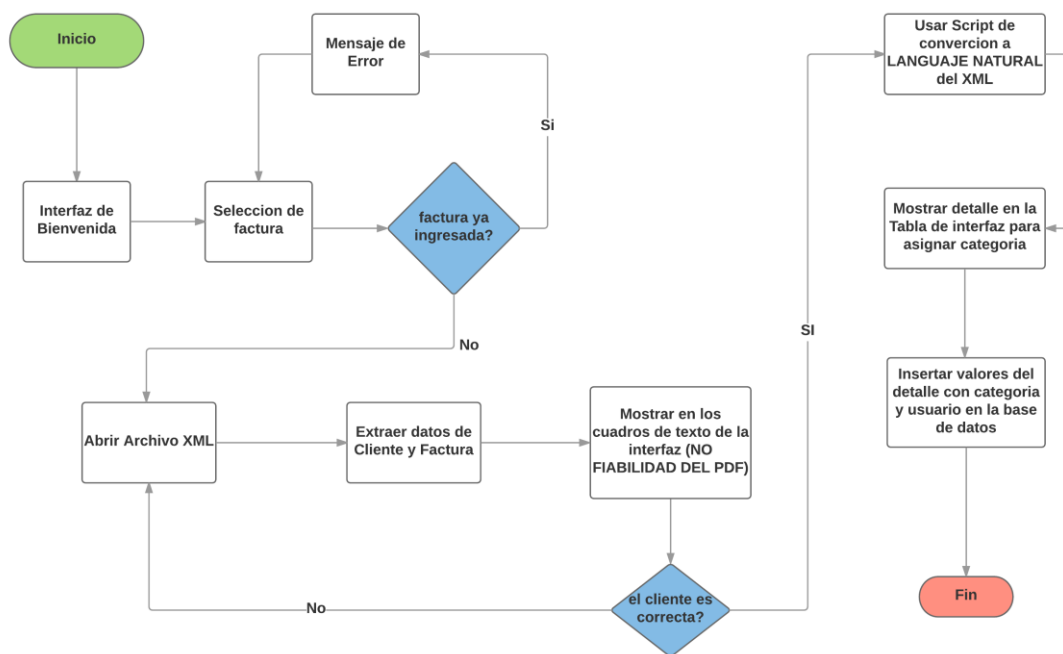


Materiales y métodos experimentales.

Materiales: para el desarrollo del proyecto se han utilizados las siguientes herramientas:

- ✓ Eclipse java neon.
- ✓ Jdk 5.8 o superior.
- ✓ API de java para procesamiento: JSON

Modelo de procesos del Proyecto





Creación del método para leer facturas XML.

```
String str="C:\\Users\\Daniel\\Desktop\\Facturas xml\\01074236.xml";
    File url = new File(str);

    DocumentBuilderFactory dbf =
DocumentBuilderFactory.newInstance();
    DocumentBuilder db = dbf.newDocumentBuilder();
    Document doc = db.parse(url);
    doc.getDocumentElement().normalize();
    System.out.println("Elemento raíz: " +
doc.getDocumentElement().getNodeName());

    NodeList nodeList=doc.getElementsByTagName("*");
    for (int i=0; i<nodeList.getLength(); i++)
    {
        // Get element
        Element element = (Element)nodeList.item(i);
        System.out.println(element.getNodeName());
    }
}
```

- Mediante la utilización de Eclipse se creó un método para que cada Factura XML puedan ser transformada a un lenguaje de bajo nivel para que pueda ser leída por el usuario.

```
String str =
"https://01836911779677979352.googlegroups.com/attach/a8b9b5fbfc9f9e3/F
ACT_28_A_20120522.xml?part=0.1&view=1&vt=ANaJVrFuVYmUW3n09k9ew_6YWHkQZ3
Z__Rm8NmFYKKt1NkZRiov_zMsyJJPLep0R72j0Lubw2_QEe6BC4sWoq5EXjmqn-
82ffXev1DyD68UqQAdy9lsS4dg";
    URL url = new URL(str);
    InputStream is = url.openStream();
    int ptr = 0;
    StringBuilder builder = new StringBuilder();
    while ((ptr = is.read()) != -1) {
        builder.append((char) ptr);
    }
    String xml = builder.toString();
    JSONObject jsonObject = XML.toJSONObject(xml);
    System.out.println(jsonObject);
```



**ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**

- Para esto se utilizó JSon de java es un formato de texto ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript aunque hoy, debido a su amplia adopción como alternativa a XML, se considera un formato de lenguaje independiente.

```
NodeList nodeList=doc.getElementsByTagName("*");
for (int i=0; i<nodeList.getLength(); i++)
{
    // Get element
    Element element = (Element)nodeList.item(i);
    System.out.println(element.getNodeName());
}
```

- Además, se creó una función con Maps para poder extraer los campos de cada una de las Facturas XML y así poder llenar una tabla que extraiga solo esos elementos.

Vista para el usuario y funcionamiento del programa.

✓ **Interfaz de bienvenida**



Ilustración 1: El gráfico muestra como iniciará el programa, el ingreso de usuario es opcional de momento, aunque este tiene un enfoque hacia el avance del proyecto



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

✓ **Lector de facturas.**

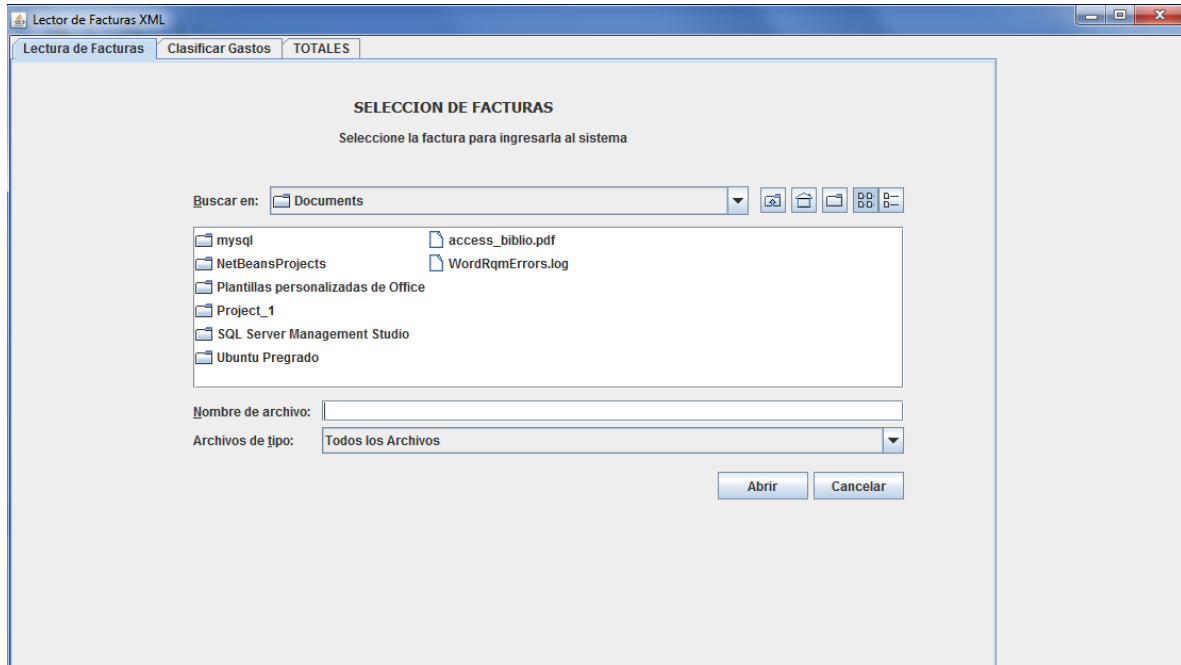
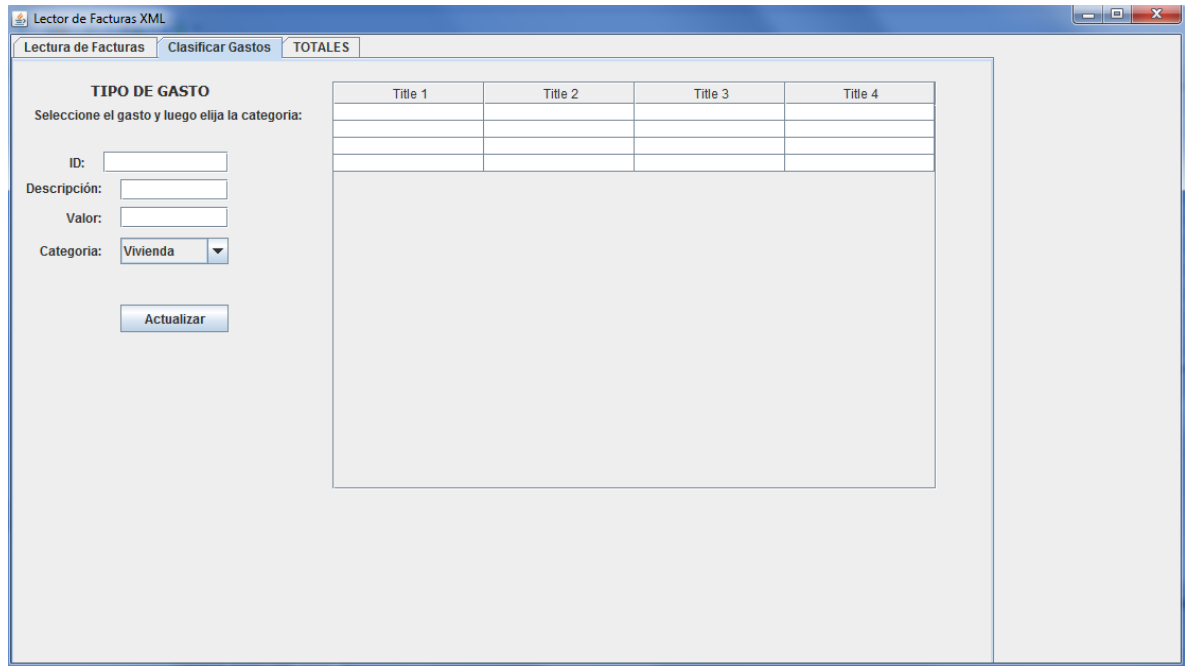


Ilustración 2: La ventana principal muestra en la pestaña de lectura de facturas la opción de seleccionar una factura XML en un directorio cualquiera en el cual se la haya guardado.

✓ **Clasificar Gastos.**





**ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**

Ilustración 3: Es esta pestaña, tras la captura de los datos XML, se clasificarán los datos referenciando los gastos realizados y bajo qué categoría serán archivados.

✓ **Ventana de resultados y reportes**

GASTOS POR CATEGORIA

Valores Totales por Categoría

CATEGORIA	VALOR GASTADO	VALOR DEDUCIBLE

TOTAL DEDUCIBLE

Ilustración 4: En la pestaña de TOTALES se muestran todas las categorías, los valores gastados y los valores deducibles; esto indicaría al usuario cuanto gasta en sus compras y que valores son sujetos a deducibles.

Análisis de Resultados.

- Las facturas XML ingresadas y procesadas para mostrar datos relevantes permitan procesar esos datos de cualquier manera, en este caso son muy útiles para capturarlos mediante una base de datos liviana y hacer consultas de cualquier índole que estén registradas dentro de las facturas.
- Los reportes, tras tener los datos de la factura en la base de datos, pueden ser muy dinámicos y a la vez mostrar información que el usuario del programa necesita conocer para su interés tributario al momento de que el vea reflejado sus gastos deducibles en el programa y use esa información para sus pagos de impuestos.
- Java permite tener la libertad de realizar programas en muchos ámbitos, y su flexibilidad y portabilidad permite que nuestro programa sea ejecutado tanto en sistemas operativos de Windows y Linux.
- Al momento de crear la base de datos se debe determinar específicamente las tablas que van a ser llenadas al extraer los datos de las Facturas XML.



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

Conclusiones y Recomendaciones.

- ❖ Se logró capturar los datos de una factura XML en una base de datos liviana de forma exitosa.
- ❖ Se capturó toda la información relevante que existe dentro de la factura XML y procesarla debidamente para lograr tener cálculos correctos de los gastos deducibles.
- ❖ Se logró realizar consultas que permitan mostrar reportes a los usuarios que procesen sus facturas.
- ❖ Se recomienda usar librerías que faciliten la ayuda al momento de capturar las facturas en formato XML y verificar la compatibilidad de bases de datos livianas con el lenguaje Java.

Referencias bibliográficas.

[1] Tutorials Point, JSon Tutorial. Disponible en:

<http://www.tutorialspoint.com/jackson/>

[2] Stackoverflow; Transform XML with XSLT in Java using DOM. Disponible en:

<http://stackoverflow.com/questions/33044212/transform-xml-with-xslt-in-java-using-dom>

[3] Captura de datos XML en JAVA. Disponible en:

http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/cepeda_p_g/capitulo4.pdf

[4] Procesamiento JSON. Disponible en:

<http://www.oracle.com/technetwork/es/articles/java/api-java-para-json-2251318-esa.html>

[5] Archivos y grupos de archivos de bases de datos. Disponible en:

[https://technet.microsoft.com/es-es/library/ms189563\(v=sql.110\).aspx](https://technet.microsoft.com/es-es/library/ms189563(v=sql.110).aspx)

[6] Motores de bases de datos Open Source. Disponible en:

<http://www.muylinux.com/2010/03/04/35-motores-de-bases-de-datos-open-source>