



Agencia Tributaria

Módulo de Cálculo Retenciones 2020

Cálculo de Retenciones IRPF 2020

Versión 1.00

enero 2020

Índice

1	INTRODUCCIÓN	3
2	FUNCIONALIDADES	4
3	BREVE DESCRIPCIÓN DEL PROCESO	5
4	DISTRIBUCIÓN DEL MÓDULO DE CÁLCULO	6
5	INTERFAZ PARA LA EJECUCIÓN DEL PROGRAMA	7
5.1	EJECUCIÓN DESDE LÍNEA DE COMANDOS.	7
5.2	INVOCACIÓN DESDE UN PROGRAMA JAVA.....	8
6	VALIDACIONES	9
6.1	FICHEROS XML	9
6.2	OTRAS VALIDACIONES	9
7	RECURSOS MÍNIMOS.....	10
8	APÉNDICE 1	11

1 Introducción

MÓDULO DE CÁLCULO RETENCIONES 2020

El objetivo de este programa es facilitar el cálculo desde fichero de las retenciones del IRPF para el ejercicio 2020, con objeto de dar cumplimiento a lo establecido en el apartado 3 de la disposición adicional trigésima quinta de la Ley 35/2006, reguladora del IRPF, añadida por el Real Decreto-ley 20/2011, de 30 de diciembre, de medidas urgentes en materia presupuestaria, tributaria y financiera para la corrección del déficit público (BOE del 31).

2 Funcionalidades

- * **La interfaz con todos nuestros módulos será estándar.**
- * **Nuestros módulos no dialogan en modo alguno con el usuario final**, es decir, a la entrega de un fichero de entrada provocan la generación de una serie de ficheros de salida. No muestran mensajes, ni cuadros de diálogo.
- * El fichero de errores se borra al final de la ejecución, si está vacío. Su contenido, en cualquier caso, se borra en la siguiente ejecución (si es sobre el mismo fichero).
- * Con relación al nombre de los archivos de la línea de comando, simplemente se exige que sean válidos. No se exige que se detalle el camino completo, aunque se admite, al igual que se admiten nombres UNC del tipo `\\servidor\recurso`.
- * Cada campo únicamente admitirá los caracteres definidos según su tipo de dato.

3 Breve Descripción del Proceso

El programa toma como datos de entrada los contenidos en el fichero definido como **nombrearchivosdatos** cuyo diseño debe ajustarse al esquema W3C (fichero **AEATRetenciones2020.xsd**) para ficheros XML.

Como resultado del cálculo obtendrá un fichero de resultados o de errores.

La descripción de los ficheros de salida y error puede consultarse en el documento adjunto previamente citado.

4 Distribución del Módulo de Cálculo

El Módulo de Cálculo se compone del siguiente archivo:

ModRet2020.jar

Adicionalmente, si el módulo de cálculo se va a utilizar como una librería integrada dentro de otra aplicación, se deberán incluir también los archivos (ver [5.2](#) y [APÉNDICE 1](#) para más información):

slf4j-api-1.6.1.jar

slf4j-jdk14-1.6.1.jar

5 Interfaz para la Ejecución del Programa

5.1 Ejecución desde línea de comandos.

La sintaxis de la llamada al programa para su ejecución es la siguiente:

```
java -jar ModRet2020.jar /E:nombrearchivodatos /R:nombrearchivoerrores  
/S:nombrearchivosalida [/I:nombreinformacion]
```

Donde:

- Los identificadores de argumento (o de parámetro) son **OBLIGATORIOS** y se deben escribir **LITERALMENTE** y con su valor pegado a los dos puntos (:).

Son los siguientes:

/E: /R: /S: /I:

Los corchetes [] indican argumento opcional y las llaves { } indican valor alternativo.

- **nombrearchivodatos** indica el fichero que contiene los datos de entrada que van a servir como base del cálculo. El fichero tiene extensión **deberá ser un documento XML** acorde con el esquema W3C (fichero **AEATRetenciones2019.xsd**). Es **OBLIGATORIO** y admite ruta completa.
- **nombrearchivosalida** indica el nombre del fichero que contiene el resultado de los cálculos.

Es **OBLIGATORIO** y admite ruta completa.

- **nombrearchivoerrores** indica el nombre del fichero que contiene la relación de errores, si los hubiera.

Es **OBLIGATORIO** y admite ruta completa.

- **nombreinformacion** indica el tipo de información adicional que deseamos imprimir. Los valores admitidos son:

- **version:** Imprime la versión del programa por la salida estándar con el formato

"Retenciones 2020 vX.XX"

Es **OPCIONAL**.

Ejemplo:

```
java -jar ModRet2020.jar /E:ret.xml /R:"c:\erroresRet.xml" /S:"c:\salidaRet.xml"
```

5.2 Invocación desde un programa java

Opcionalmente, si el módulo de cálculo se utiliza desde una aplicación Java, es posible integrarlo como una librería más de dicha aplicación e invocar el proceso de cálculo desde su código fuente:

1. Se deberá incluir el fichero `ModRet2020.jar` (y el resto de archivos necesarios) en el *classpath* de la aplicación que utilice el módulo de cálculo, tanto en compilación como en ejecución.
2. Para lanzar el proceso se deberá invocar alguno de los métodos estáticos `procesarFicheroXml`¹ de la clase `es.aeat.pret.c200.mc.ModuloCalculo`. Por ejemplo:

```
import es.aeat.pret.c200.mc.ModuloCalculo;

.....

ModuloCalculo.procesarFicheroXml("entrada.xml","errores.xml","", "salida.xml");
```

¹ Consulte la ayuda javadoc incluida en la distribución del módulo de cálculo.

6 Validaciones

El Programa no realiza el cálculo cuando no se superen las validaciones descritas a continuación.

6.1 *Ficheros XML*

Los ficheros XML de entrada deberán ajustarse al esquema W3C que se puede descargar desde la página web de la Agencia Tributaria (fichero **AEATRetenciones2020.xsd**).

6.2 *Otras Validaciones*

Se realizarán además las validaciones que se indican en el documento **ALGORITMO_2020.pdf**, que se puede descargar desde la página web de la Agencia Tributaria, donde se describe el algoritmo para el cálculo de retenciones.

7 Recursos Mínimos

64 MB de memoria RAM mínima.

Disco duro con espacio libre mínimo de 2 MB.

Multiplataforma: Sistemas Windows, Mac OS X, GNU/Linux y, en general, cualquier plataforma capaz de ejecutar un Máquina Virtual Java compatible con Java 8.

8 APÉNDICE 1

El Módulo de Cálculo de Retenciones 2020 utiliza SLF4J² como sistema de *logging*. El archivo `slf4j-jdk14-1.6.1.jar` es el enlace con el sistema de log de java y solo se utilizará si el módulo se ejecuta como aplicación desde línea de comandos.

En el caso de que se emplee el módulo de cálculo como una librería embebida en otra aplicación java (ver 5.2) se podrá utilizar cualquier otro *framework* de *logging* (log4j, jcl, etc.) incluyendo el correspondiente fichero de enlace para SLF4J en el *classpath* de la aplicación. Para más información, visite la página web de SLF4J <http://www.slf4j.org/>.

² *Simple Logging Facade for Java*: Para más información vea <http://www.slf4j.org>