Desde el servicio de Informática, nos encargan como programadores la tarea de desarrollar una aplicación en Java, la cual nos permita gestionar el impuesto de vehículos de transmisión mecánica (IVTM) de cualquiera de los 8.132 Municipios del país.

La información necesaria para generar cada uno de los recibos, os será facilitada a partir de la siguiente práctica en documentos Excel. Dado que la tarea final de la aplicación consistirá en realizar un backup de toda la información y cálculos realizados contra una base de datos MySQL, y con el fin de que todos tengamos las mismas clases Java en la aplicación, comenzamos el desarrollo en esta primera práctica creando la citada base de datos (cuyo diagrama tenéis disponible en Moodle), a la que llamaremos "IVTM".

Para realzar las altas, consultas, actualizaciones y borrados de la base de datos, utilizaremos una herramienta de mapeo objeto-relacional (ORM) para la plataforma Java: Hibernate. Por ello, en esta primera práctica además de crear la base de datos anteriormente mencionada, se debe crear el proyecto que incluya Hibernate permitiendo ejecutar sentencias HQL (**que serán las únicas permitidas en el desarrollo del proyecto**) contra la base de datos. En Moodle se os facilita además un script realizado con Heidi SQL que carga datos en la base de datos que debéis crear previamente.

Al ejecutar esta primera práctica, el sistema pedirá el NIF de un contribuyente. Si el contribuyente no se encuentra en la base de datos, se devolverá un mensaje por pantalla indicando que no hemos encontrado al contribuyente entre nuestros datos.

Si por el contrario el nif del contribuyente se encuentra en base de datos, se deben realizar las siguientes acciones:

1. Mostrar por pantalla su nombre, apellidos, nif y dirección.
2. Cambiar el importe total de todos los recibos del contribuyente (sólo el atributo total recibo) a 115 euros.
3. Eliminar todos los recibos cuyo importe total sea menor que la media de todos los importes totales de todos los recibos almacenados en base de datos.

Es **obligatoria la utilización** del lenguaje de consultas de base de datos **HQL de Hibernate**, no siendo aceptada la utilización de sentencias SQL (trabajan a nivel de tabla en lugar de trabajar como se pide, a nivel de clase).

La entrega la debéis subir al Moodle **TODOS** los componentes de cada grupo. En ella deben estar los ficheros fuente contenidos en la carpeta src, **junto con un fichero txt que incluya el nombre de los componentes del grupo.**

1.- Ejemplo de consulta con parámetros

Fuente: (Capítulo 16 The HQL Lenguaje)
https://docs.jboss.org/hibernate/orm/4.3/manual/en-US/html/

```
Query query = sesion.createQuery("SELECT ord FROM Recibos ord WHERE
ord.nifContribuyente = ?“);
query.setParameter(0, recibo.getNifContribuyente());
List <Recibos> listRecibos = query.list();
```

2.- Ejemplo de borrado con parámetros

```
tx = sesion.beginTransaction();
      String HQLborrado = "DELETE Recibos r WHERE r.nifContribuyente=:param1";
       sesion.createQuery(HQLborrado).setParameter("param1", nif).executeUpdate();
tx.commit();
```

3.- Ejemplo de inserción o actualización de objeto

```
Contribuyente c = new Contribuyente();
//Seteamos los valores de c antes de insertar o actualizar la base de datos
tx=sesion.beginTransaction();
   sesion.saveOrUpdate(c);
   actualizado=true;
tx.commit();
```

A partir de esta práctica, os facilitamos dos hojas de cálculo Excel con formato "xlsx", los datos personales de los contribuyentes, los vehículos a liquidar el impuesto municipal y las ordenanzas de diferentes Ayuntamientos con sus importes asociados. El primer cometido de esta práctica es crear una clase que denominaremos **ExcelManager** que permita **leer ambos ficheros y modificar** la hoja del fichero Excel en que se encuentran los contribuyentes, utilizando la librería ApachePOI.

Tanto la estructura como los tipos de datos de las columnas de la hoja excel, **salvo error u omisión**, permanecerán invariables a lo largo de todo el curso **(no así el orden de las filas ni por supuesto el contenido de las mismas)**. Dentro de vuestro proyecto cread una carpeta llamada **"resources"** en la que debéis almacenar ambos ficheros Excel y **todos los ficheros que generéis con vuestra** aplicación (tanto en ésta como de las siguientes prácticas).

Para la ejecución de esta práctica debéis programar un validador de NIF/NIE en lenguaje Java. La información relativa al cálculo del dígito de control se encuentra disponible en el siguiente recurso oficial:

https://www.interior.gob.es/opencms/es/servicios-al-ciudadano/tramites-y-gestiones/dni/calculo-del-digito-de-control-del-nif-nie/

1.- Validador de NIF-NIE. Se define de la siguiente forma: vuestra aplicación debe acceder a los datos de cada contribuyente de la primera hoja del fichero Excel, entre los que se encuentra el NIF/NIE del mismo. Se debe verificar si es o no correcto, tanto en su estructura como en el dígito de control (letra) almacenado y realizar la acción correspondiente según la letra calculada por vuestro algoritmo:

- Si el NIF/NIE almacenado es correcto, no se debe realizar acción alguna.
- Si el NIF/NIE almacenado es erróneo pero subsanable, debéis actualizar en la hoja Excel el campo de información con la nueva letra calculada. Si el error no es subsanable, enviad la información al fichero ErroresNifNie.xml de errores.
- ***Estructura fichero XML "ErroresNifNie.xml" (nif duplicados, erróneos o en blanco)***
  En caso de NIF/NIE repetido se almacenarán la segunda aparición y siguientes. La estructura XML consistirá en un nodo raíz "contribuyentes" y, bajo el mismo, una colección de elementos "contribuyente" con atributo el identificador numérico de la fila que ocupa en la hoja Excel, y como elementos del mismo el nif_nie,

nombre, primer apellido y segundo apellido del contribuyente, así como el tipo de error (NIF ERRONEO, NIF BLANCO, NIF DUPLICADO). Tenéis un ejemplo del mismo subido en Moodle.

Al finalizar la ejecución de esta práctica, los NIF y NIEX de los contribuyentes deberán ser datos correctos en el libro Excel (salvo los enviados al fichero xml de errores) y, dentro de la carpeta "resources" de vuestro proyecto, deberán encontrarse el fichero XML anteriormente descrito en caso de haber errores.

Podéis utilizar las estructuras de datos que más os guste para almacenar la información contenida en el libro Excel. Procurad trabajar con la información en memoria y **minimizad el número de aperturas y cierres del fichero Excel**, de lo contrario la aplicación tendrá tiempos de ejecución elevados.

FICHEROS A ENTREGAR TODOS LOS ALUMNOS: Debéis subir al Moodle los siguientes ficheros:

- **Carpeta src** con los ficheros fuente de vuestro proyecto.
- **Carpeta "resources"** con el libro Excel actualizado y el fichero xml generado **tras una única ejecución** de la práctica, a partir del libro Excel facilitado.
- Fichero txt con el nombre de los componentes del grupo.

Con la nueva adaptación europea para realizar pagos en euros, se hace obligatoria la utilización para cualquier transferencia realizada dentro de la Unión, de la adaptación de las cuentas bancarias al formato SEPA (zona única de pagos en euros www.sepaesp.es/sepa/es/). La información que disponemos de cada contribuyente es su código de cuenta bancaria (CCC) y el país al que pertenece dicha cuenta. Para realizar el pago del impuesto de vehículos, se hace necesario conocer el código IBAN[1] asociado a cada cuenta bancaria de la que se dispone.

Esta tercera práctica debe tener las siguientes funcionalidades añadidas:

- **Verificación del código de cuenta bancaria de cliente (CCC)** para toda línea de fichero Excel, aunque el nif sea erróneo, duplicado o esté en blanco. Si el CCC **es subsanable o erróneo**, subsanadlo si se puede y, en todo caso, enviar dicha información al fichero "ErroresCCC.xml".

- Generación del IBAN asociado a la cuenta bancaria y actualización en la hoja excel. **Solo se ha de generar dicho IBAN y actualizar la hoja excel** cuando el NIF/NIE del contribuyente sea correcto (o subsanable) y su CCC sea correcto (o subsanable).

- **Generación de email** de aquellos contribuyentes que carezcan de él, siempre y cuando su **NIF/NIE sea correcto** (o subsanable), **y su CCC sea correcto** (o subsanable**),** con la siguiente estructura y sin que aparezcan correos repetidos (obviamente):
  - Letra inicial del nombre
  - Letra inicial del primer apellido
  - Letra inicial del segundo apellido (si lo hubiere).
  - Número de repetición (comenzando en 00).
  - @ vehiculos2025.com

Al finalizar la ejecución de esta tercera práctica, los NIF y NIEX de los contribuyentes, sus números de cuenta e IBAN, así como sus respectivas cuentas de correo electrónico, deberán ser datos correctos en el libro Excel (salvo los contribuyentes con nif o CCC erróneo y no subsanable, **a los que no se les debe generar nada, ni tan siquiera el correo**) y, dentro de la carpeta "resources" de vuestro proyecto, deberán encontrarse los 2 ficheros XML pedidos.

[1] https://www.europeanpaymentscouncil.eu/

Estructura del fichero XML "ErroresCCC.xml" (Cuentas Bancarias erróneas)

Cuentas bancarias **erróneas y subsanadas**. Nodo raíz "cuentas" y, bajo el mismo, una colección de elementos "cuenta" con atributo el id numérico de la fila que ocupa el contribuyente en la hoja Excel, su nombre, apellidos, el código de cuenta erróneo y, en caso de ser subsanable, el IBAN calculado.

Cuentas bancarias imposibles de corregir o generar IBAN. Se añadirá un elemento "TipoError" con el literal IMPOSIBLE GENERAR IBAN.

Como en anteriores ocasiones, tenéis un ejemplo de fichero subido en Moodle.

FICHEROS A ENTREGAR: Debéis subir al Moodle los siguientes ficheros:

- **Carpeta src** con los ficheros fuente de vuestro proyecto.
- **Carpeta "resources"** con el libro Excel actualizado y los 2 ficheros xml generados **tras una única ejecución** de la práctica, a partir del libro Excel facilitado para esta tercera práctica.
- Fichero txt con el nombre de los componentes del grupo.

Como continuación del proyecto, esta práctica **debe incluir** la funcionalidad de las prácticas anteriores y, posteriormente, realizar los cálculos necesarios que permitan generar de forma automatizada el recibo de cada contribuyente para el período impositivo introducido por consola.

La aplicación pedirá al usuario que introduzca por consola el año del que se desean generar los recibos de los contribuyentes almacenados en el libro Excel. El formato de entrada será **aaaa** (Ejemplo "2024", y **siempre será correcto**).

Para realizar los cálculos se ha de tener en cuenta:

1.- Datos y características correctas del vehículo (matricula, unidades y propietario) y una correcta correlación de las fechas del mismo.

2.- La fecha de matriculación, fecha de alta en el Ayuntamiento, y las fechas de baja temporal y baja definitiva (si es que las tiene) del vehículo.

3.- Ordenanza asociada al Municipio del contribuyente.

4.- Si existe algún tipo de exención para el vehículo o bonificación para el citado contribuyente.

Antes de generar los recibos de la aplicación, se debe comprobar que la correlación de fechas del vehículo es correcta, que la matrícula del vehículo tiene un formato adecuado al tipo de vehículo que estemos tratando, que el vehículo tiene propietario y que el propietario no se encuentra entre los casos erróneos descritos en prácticas precedentes. En caso de encontrarnos ante cualquiera de dichos errores, la aplicación no generará recibo para el vehículo tratado y enviará dicho vehículo al fichero erroresVehiculos.xml.

Si no nos encontramos ante ningún tipo de error, la aplicación debe obtener **todos** los valores que aparecerán en el recibo de cada vehículo, mostrando los resultados de los cálculos por pantalla y generando un fichero llamado recibos.xml que contendrá todos los recibos generados.

Tenéis disponible en Moodle la salida final de la aplicación como ejemplo de cómo deberán ser los recibos finales generados (algo que se realizará en la próxima y última práctica). Al finalizar la ejecución de esta práctica es obligatorio que se muestre por pantalla de cada contribuyente la siguiente información:

- **Contribuyente**: Nombre, apellido1, apellido2, NIFNIE, dirección, IBAN, y bonificación.
- **Fecha** del recibo (día que ha sido generado) y fecha del padrón generado (siempre podremos que el padrón es generado el 1 de enero del año solicitado).

- Tipo de vehículo, marca, modelo, matrícula, número de bastidor, unidad por la que se cobra, valor de la unidad, importe, exención-bonificación si proceden e importe total del recibo.

No se ha de generar recibo de ninguna fila que en la hoja Excel sea errónea (por nif o cuenta bancaria). Tampoco de aquellos vehículos con la matrícula errónea, fechas incoherentes, vehículos sin propietario o cuyos datos no sean correctos según lo definido a lo largo de la aplicación.

Fichero Recibos.xml en la carpeta resources.

Todo recibo generado debe ser almacenado en el fichero Recibos.xml. La estructura XML consistirá en un nodo raíz "Recibos" con los atributos fechaPadron, totalPadron y numeroTotalRecibos y, bajo el mismo, una colección de elementos "Recibo" con atributo el identificador autonumérico de la base de datos para dicho recibo, y como elementos los valores correspondientes a exención, idFilaExcelVehiculo, nombre, primerApellido, segundoApellido, NIF, IBAN, tipoVehiculo, marcaModelo, matricula, y totalRecibo de cada recibo generado.

Fichero erroresVehiculos.xml en la carpeta resources.

Todo vehículo erróneo debe ser almacenado en el fichero erroresVehiculos.xml. La estructura XML consistirá en un nodo raíz "Vehiculos" sin atributos. Bajo el mismo, una colección de elementos "Vehículo" con atributo el identificador autonumérico de la base de datos de dicho vehículo, y como elementos los valores correspondientes a la Marca, Modelo y el Error encontrado. Los valores que puede presentar el elemento Error de cada vehículo son "Fechas incoherentes", "Matricula Errónea", "Vehículo sin propietario" y "Vehículo con propietario erróneo". (Como mínimo ha de tener uno de esos valores, pero obviamente pueden aparecer varios, tanto como errores contenga dicho vehículo).

FICHEROS A ENTREGAR: Debéis subir al Moodle **TODOS** los miembros del grupo los siguientes ficheros:

- **Carpeta src** con los fuentes de vuestro proyecto.
- **Carpeta "resources"** con el libro Excel actualizado y los ficheros Errores.xml, ErroresCCC.xml y recibos.XML generados **tras una única ejecución** de la práctica, a partir del libro Excel facilitado para esta práctica y ejecutada la aplicación con **fecha de entrada 2024.**
- Fichero txt con el nombre de los componentes del grupo

Esta es la última práctica de la asignatura para este curso. En su ejecución, descartará las funcionalidades de la primera práctica, e incluirá todas y cada una de las funcionalidades de las prácticas restantes, dando un formato de salida final a todos los resultados obtenidos según se detalla a continuación.

1.- La aplicación deberá generar para cada año introducido por consola, los recibos de los contribuyentes en formato pdf, debiendo ser almacenados en la carpeta **resources\recibos**.

El nombre del fichero pdf estará compuesto por: nif del contribuyente, nombre y apellidos, matrícula del vehículo y año del impuesto generado. La generación de los pdf se realizará de forma **obligatoria con la librería iText en su versión 9.1** o superior.

En cada recibo generado **DEBE** aparecer la siguiente información (tenéis el modelo del recibo en Moodle):

- **Contribuyente**: Nombre, apellido1, apellido2, NIFNIE, dirección, Ayuntamiento, IBAN y bonificación (si existe).
- **Fecha** del recibo (fecha en que ha sido generado), fecha del padrón y número de trimestres que incluye, así como el número de recibo.
- **Vehículos** Tipo de vehículo, marca, modelo, número de bastidor y matrícula, fecha de matriculación del vehículo y fecha de alta en el Ayuntamiento. En caso de existir, fecha de baja temporal y fecha de baja definitiva del vehículo.
- **Cada línea del recibo** contendrá el tipo de vehículo, su marca y modelo, la unidad por la que se le cobra el impuesto, el valor de dicha unidad y el importe a pagar, así como el porcentaje de bonificación en caso de existir o la exención del mismo si procede.

2.- Generar un pdf resumen del padrón completo denominado "nif alumno1_ + nifalumno2_ (si sois 2 alumnos) + resumen.pdf" (ejemplo: 12345678Z_87654321X_resumen.pdf), en el que aparezca reflejado el año del padrón al que corresponden los recibos, el importe total del padrón de recibos (sumatorio de todos los recibos), así como el número total de recibos.

3.- Almacenar (o actualizar) los datos obtenidos en la base de datos MySql definida en la primera práctica de la asignatura, de la siguiente forma: si el recibo, contribuyente, ordenanza o vehículo no existen en base de datos, se realizará una inserción de los mismos. En caso contrario, se procederá a actualizar los valores no coincidentes.

- Un recibo ya existe si coincide el nif del contribuyente, la fecha del padrón de dicho recibo y la matrícula del vehículo.
- Una ordenanza ya existe si coinciden el Ayuntamiento de dicha ordenanza, el tipo de vehículo, la unidad por la que se cobra y los valores mínimo y máximo del rango.
- Un contribuyente ya existe si coinciden su nombre y apellidos y su NIF/NIE.
- Un vehículo ya existe si coincide su matrícula.

No se debe insertar en base de datos ningún contribuyente erróneo por nif o CCC (tal y como fueron definidos en las prácticas 2 y 3). Tampoco se deberá insertar ningún recibo cuyo contribuyente no sea correcto o el vehículo de dicho recibo tenga su matrícula errónea, las fechas del vehículo sean incoherentes o, directamente, no tenga propietario conocido o éste sea nulo.

ENTREGA DE LA ÚLTIMA PRÁCTICA – APLICACIÓN COMPLETA.

En esta última entrega se debe subir al Moodle lo siguiente:

- **Carpeta src** con los ficheros fuente finales de la aplicación completa, no siendo necesario añadir las librerías del proyecto.
- Fichero txt con el **nombre de los componentes del grupo**.
- Fichero sql que exporte tanto la estructura de vuestra base de datos como los datos almacenados en ella, generados tras la ejecución de vuestra aplicación final **2 veces consecutivas**, partiendo de una **base de datos en blanco** y con fechas de generación de recibos "2024" y "2025".