



UNIVERSIDAD DE GUAYAQUIL

PROYECTO 2DO PARCIAL

**INGENIERÍA DE
SOFTWARE**

INTEGRANTES:

CAMPOVERDE LEIVA SERGIO VICENTE

CRUZ DIAZ MILENA LIZBETH

GUALE ARTEAGA JOHN STALIN

MATAMOROS JALCA JOAQUIN FERNANDO

Ciclo II 2020-2021

Facultad de Ciencias Matemáticas y Físicas

March 17, 2021

Contents

1.0	Introduction	2
1.1	Metamodelo ECORE (EMF) de un modelo relacional (MR).	3
1.1.0	Modelo de Entidad-Relación (MER)	3
1.1.1	Proceso de creación de metamodelo basado en un MR.	4
1.2	Generar SQL A Partir De Un Metamodelo En Acceleo.	23
1.3	Sentencias en SQL	54
1.4	Repositorio Github	58

1.0 Introduction

Los metamodelos permiten representar los modelos derivados de los procesos del desarrollo de software en un nivel de abstracción más sencillo que el proporcionado por la representación con UML. El modelo del sistema se construye utilizando un lenguaje de modelado (que puede variar desde lenguaje natural o diagramas hasta fórmulas matemáticas). Los modelos informales son expresados utilizando lenguaje natural, figuras, tablas u otras notaciones. En los modelos formales cuando la notación empleada es un formalismo, es decir posee una sintaxis y semántica (significado) precisamente definidos.

Para la utilización de ecore en Eclipse es necesario tener instalado el plugin de EMF (Eclipse Modeling Framework), este plugin provee básicamente dos herramientas para construir un modelo basado en ecore, una el Ecore Model que es un editor manual que funciona en un estilo de árbol de navegación para la creación del modelo basado en ecore, la otra es el Ecore Diagram siendo este un editor gráfico similar a las herramientas gráficas para la creación de diagramas de clases UML. Cualquiera de las dos formas que se utilice para crear el diagrama basado en ecore, genera un fichero XMI (XML Metadata Interchange) que es una especificación para el intercambio de diagramas, en este caso se utilizará Ecore Model.

En el presente documento se realizará la creación de un metamodelo desarrollado en eclipse haciendo uso de Eclipse Modeling Framework (EMF), el tema a representar es la “venta de boletos de avión”. El finalidad de este proyecto es crear un metamodelo ecore, luego al estar definido representarlo en un diagrama con sus respectivas multiplicidades, con las OCL se asignan restricciones que validan campos establecidos. Posterior a ello crear un nuevo proyecto Acceleo que hará uso de las definiciones dadas con el archivo. ecore para transformarlas a lenguaje Sql Server.

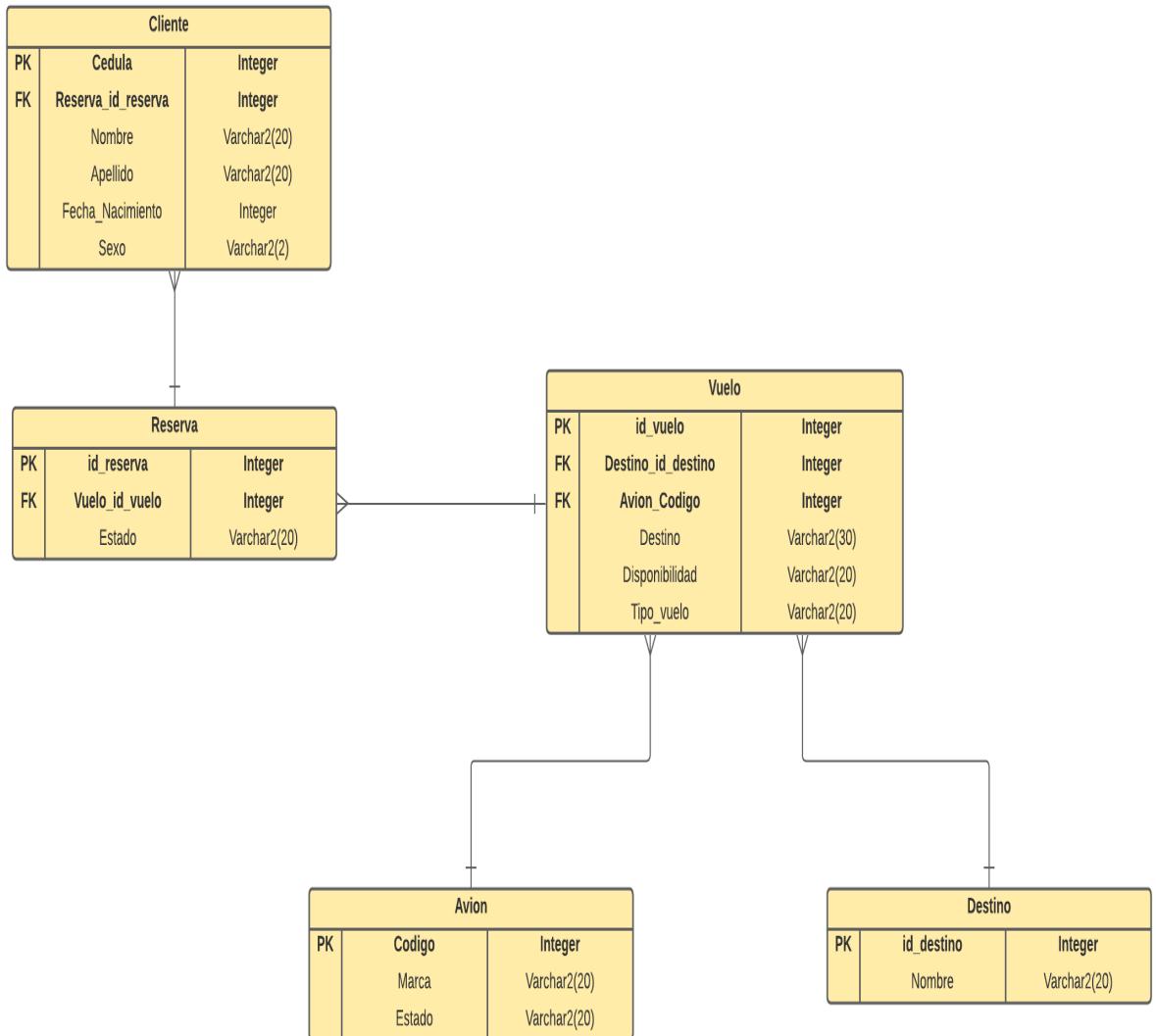
Para la generación de código se realizó un plugin que explota la información contenida en el metamodelo utilizando Acceleo y OCL. Este plugin contiene las clases necesarias para configurar, disparar la generación y una serie de archivos templates que combinan el código a generar con consultas al modelo mediante el lenguaje, que contiene Acceleo, OCL y la información específica del metamodelo.

Del literal 1.0.1 en adelante se muestran imágenes donde detallan las pasos que se debe seguir para realizar el proyecto, esto permitirá hacer uso de estas herramientas que son de gran ayuda dentro de la ingeniería de software.

1.1 Metamodelo ECORE (EMF) de un modelo relacional (MR).

1.1.0 Modelo de Entidad-Relación (MER)

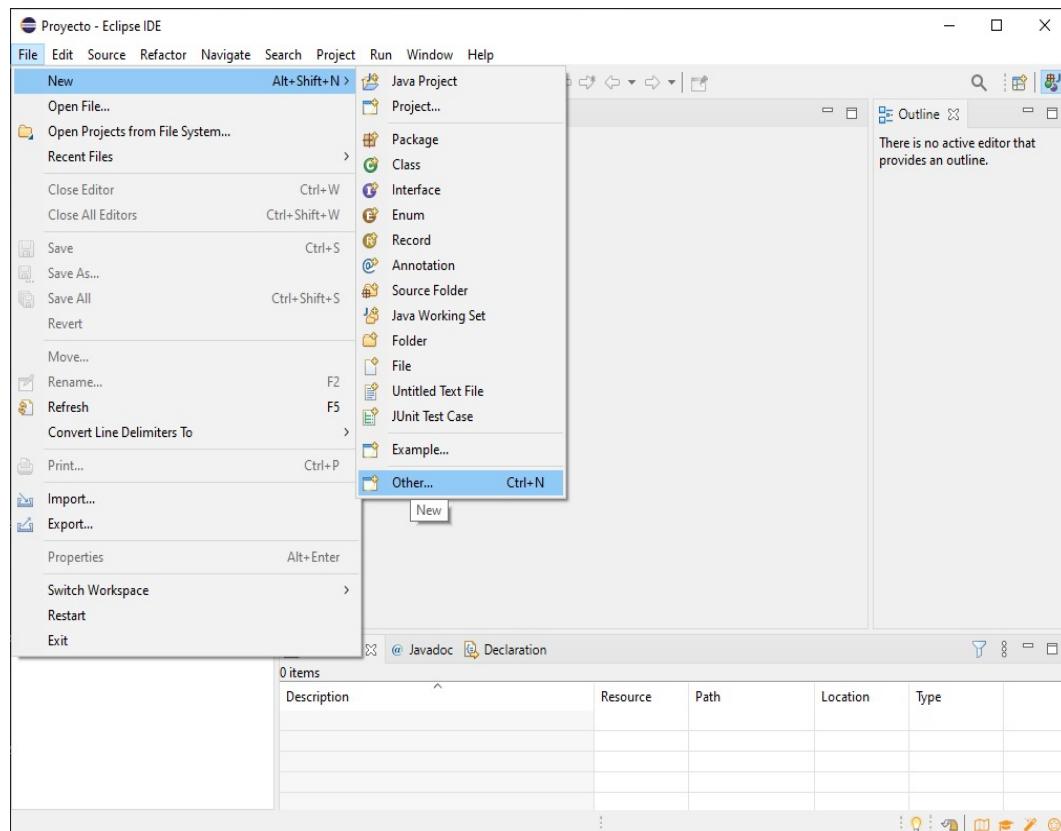
El modelo entidad relación es una herramienta que permite representar de manera simplificada los componentes que participan en un proceso de negocio y el modo en el que estos se relacionan entre si.

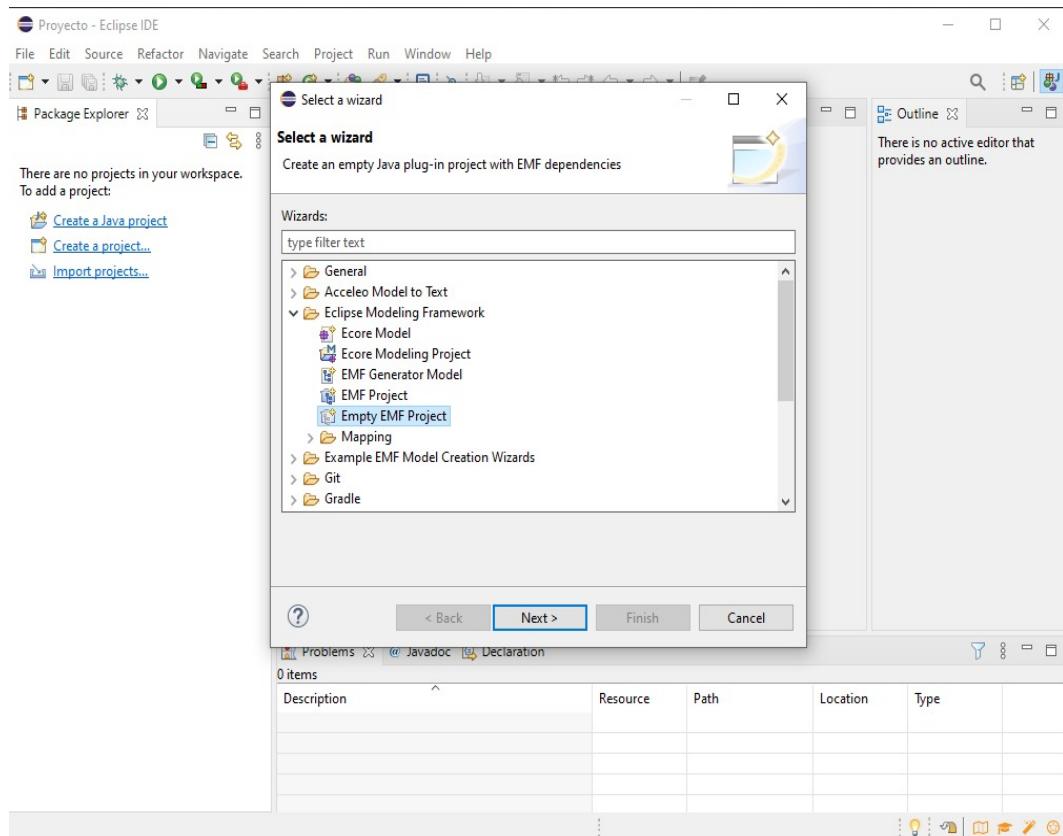


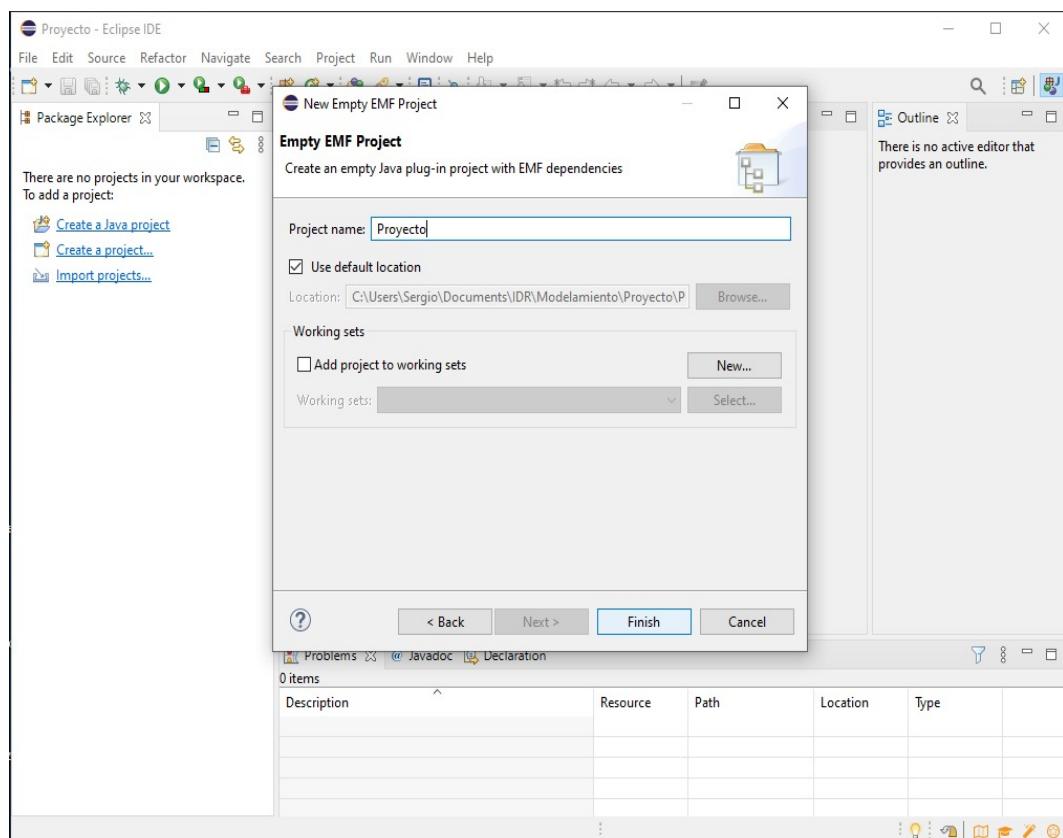
Modelo Relacional.

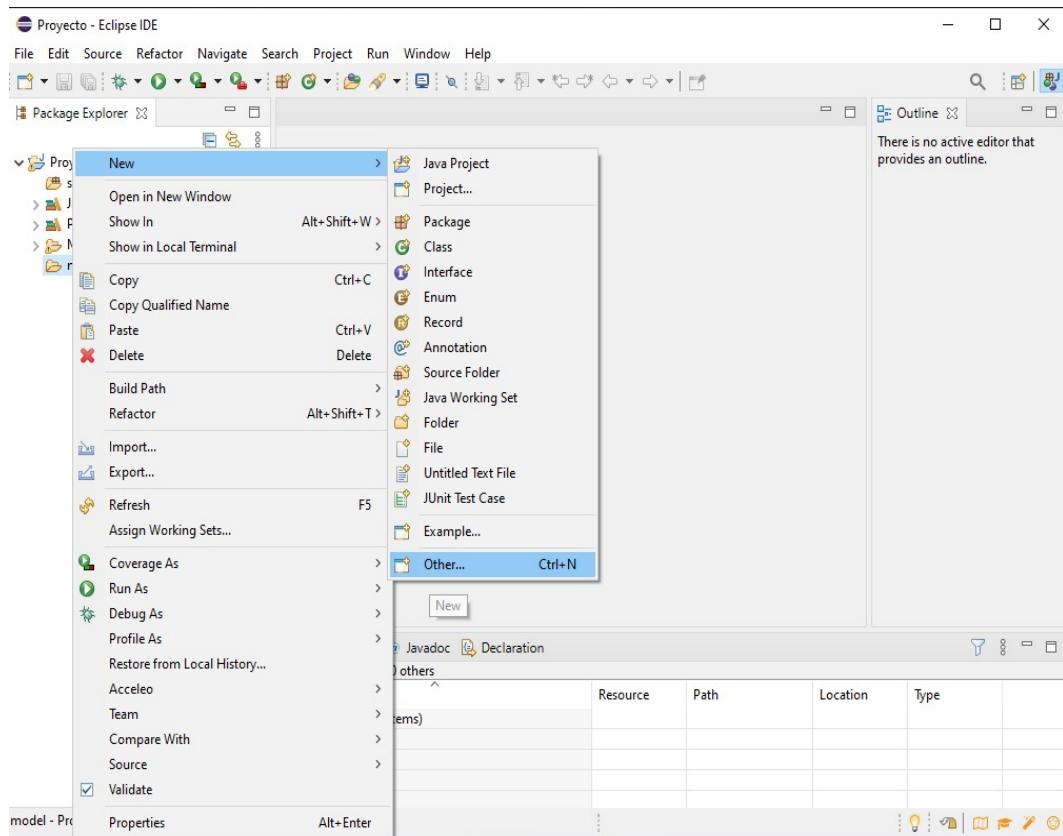
1.1.1 Proceso de creación de metamodelo basado en un MR.

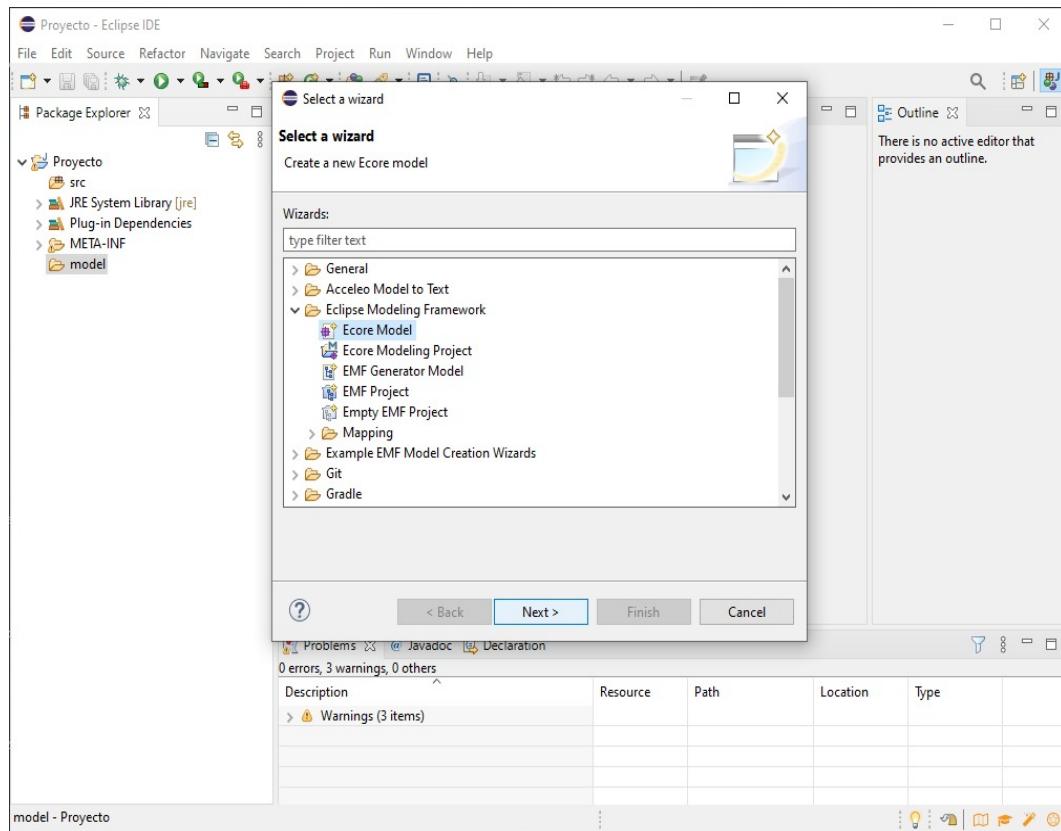
Se crea un proyecto Empty EMF Project como usualmente hacemos en las prácticas de la asignatura de modelamiento.

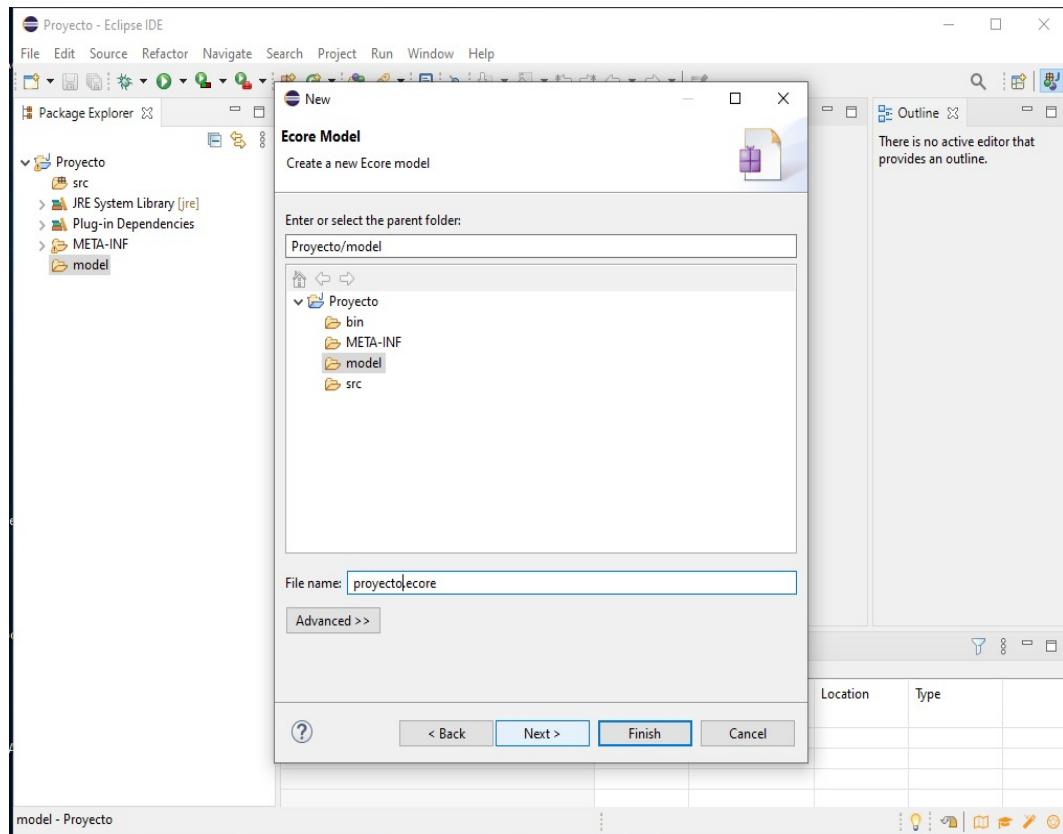


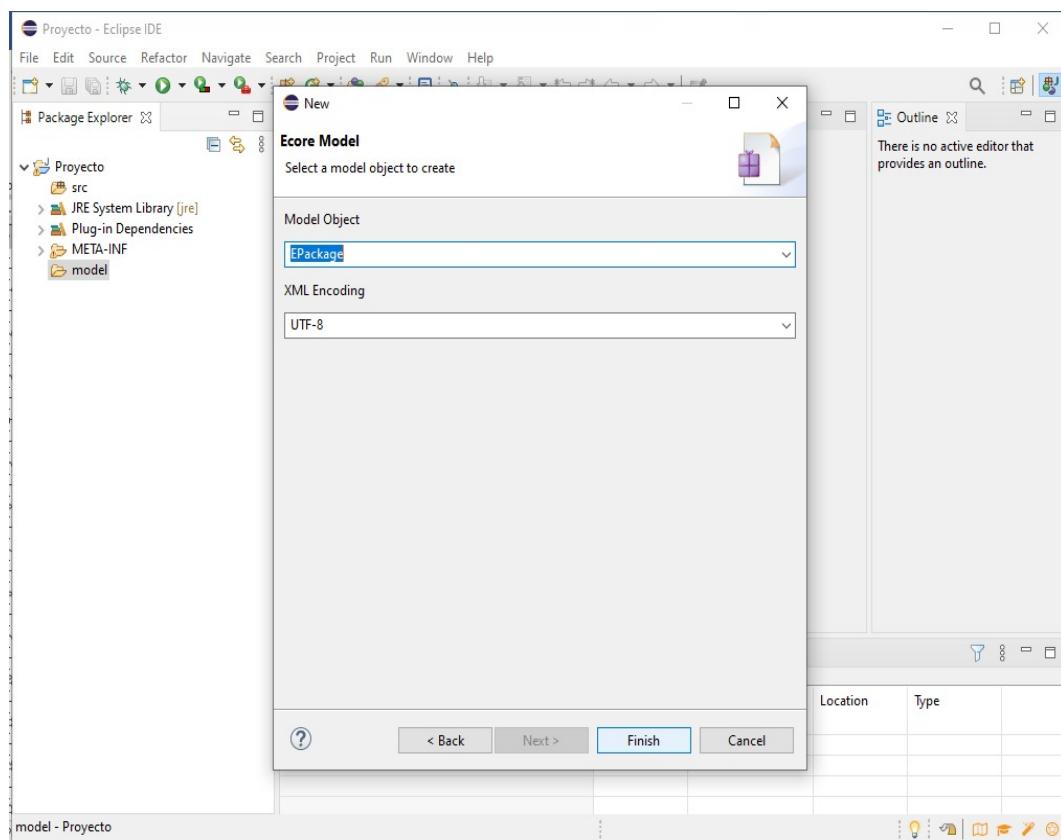




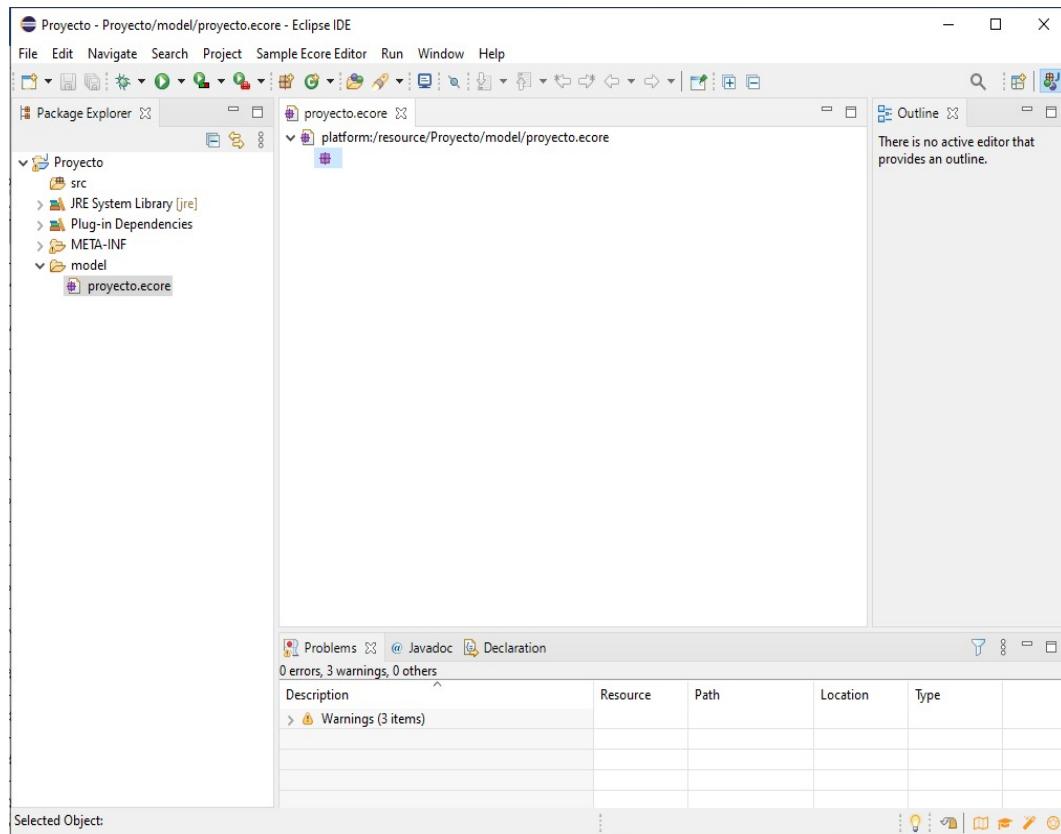


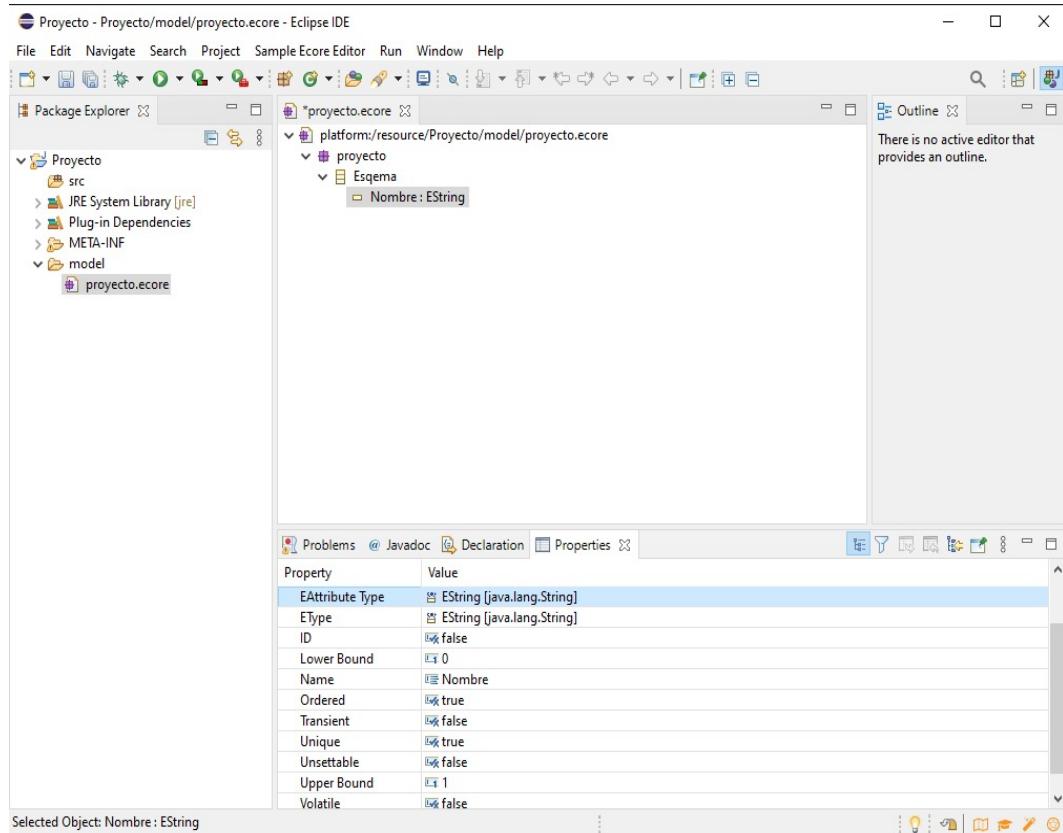


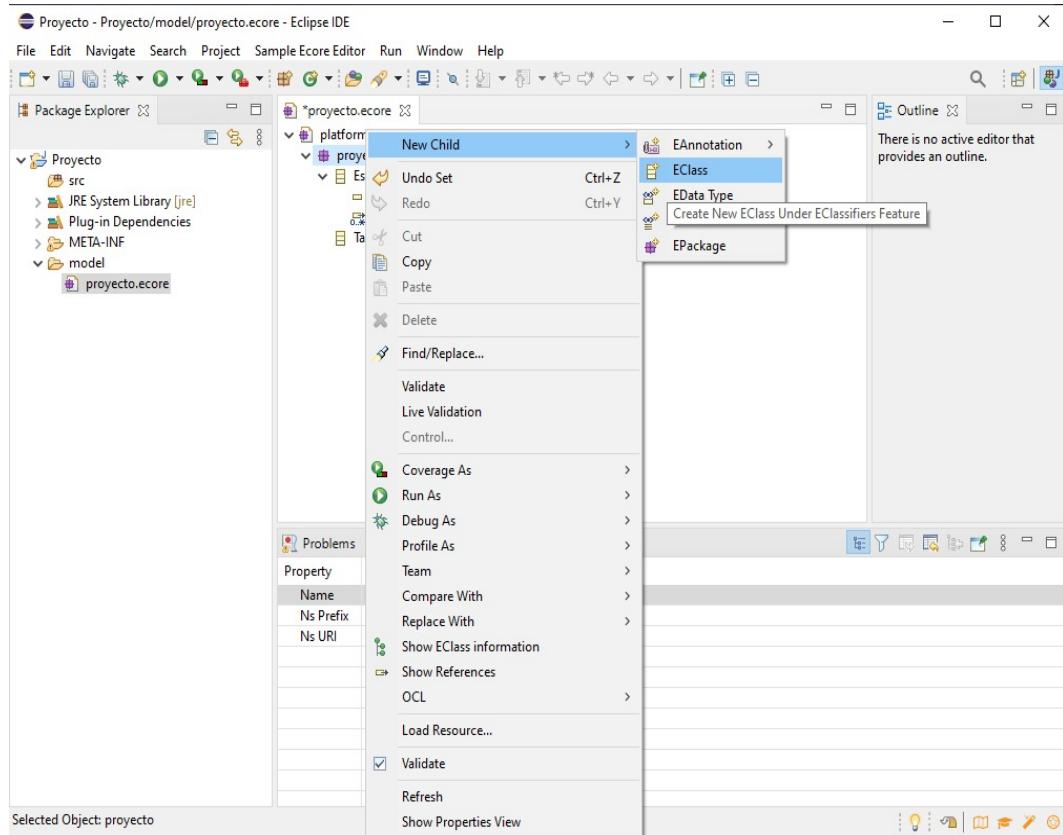




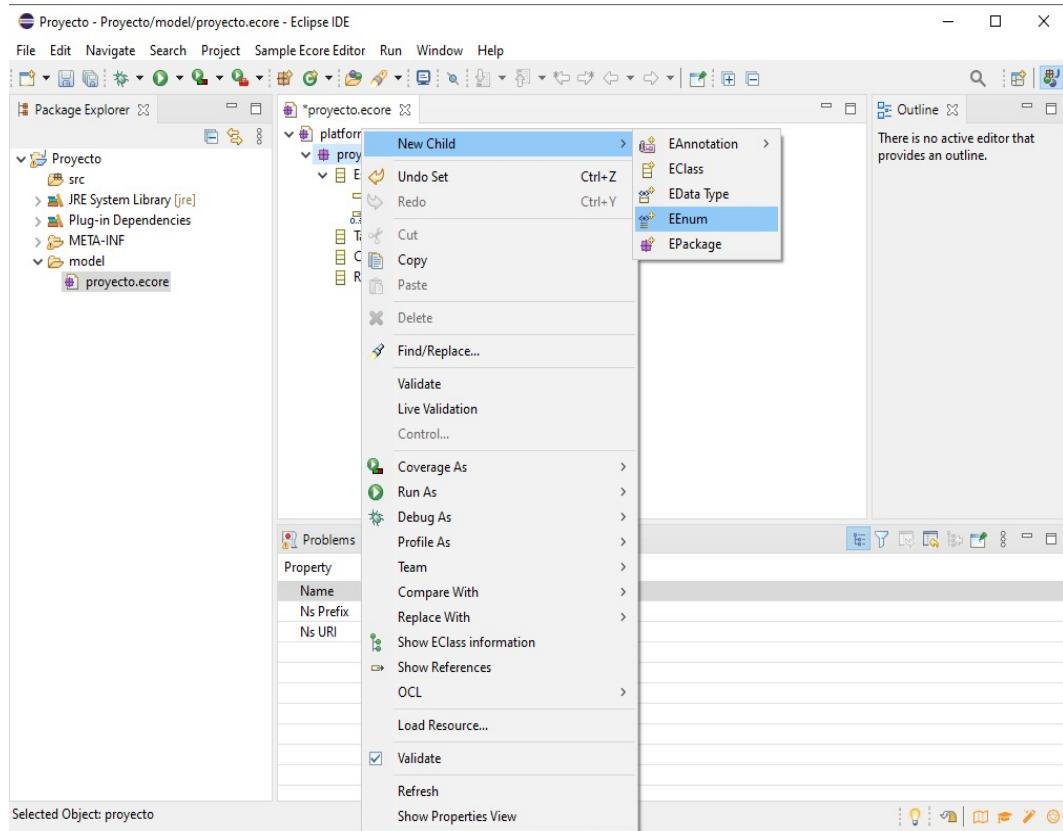
Se inicializa Ecore Diagram.

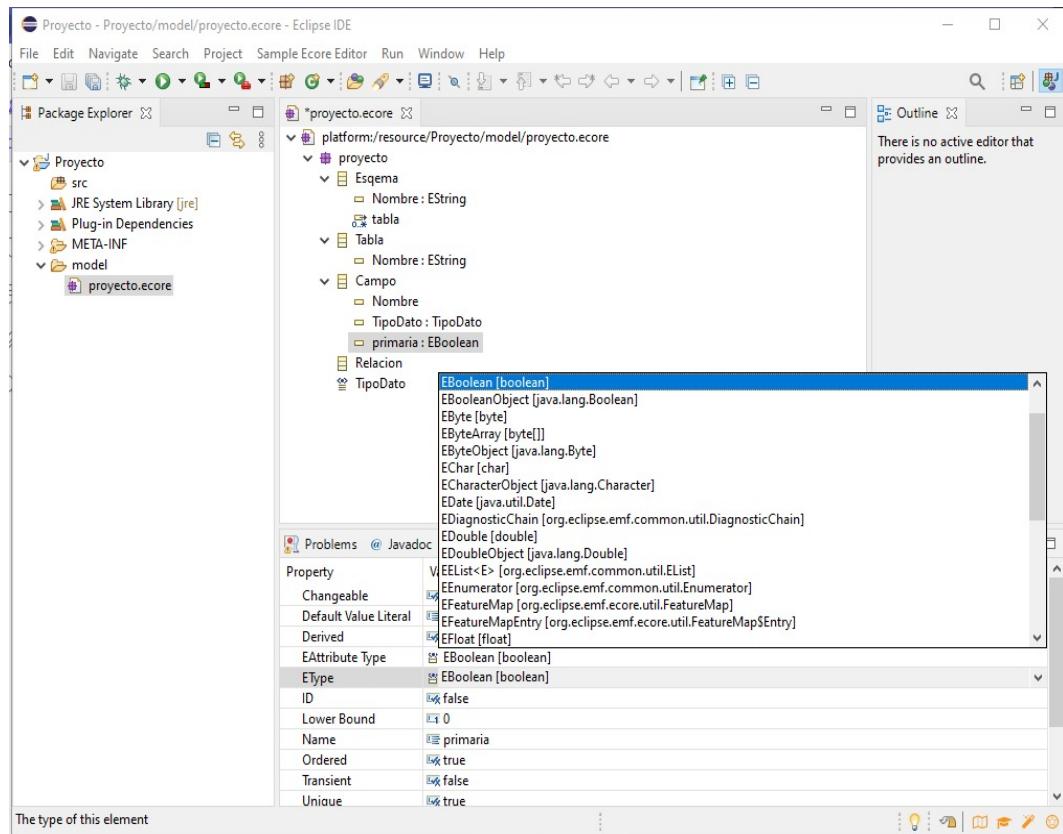


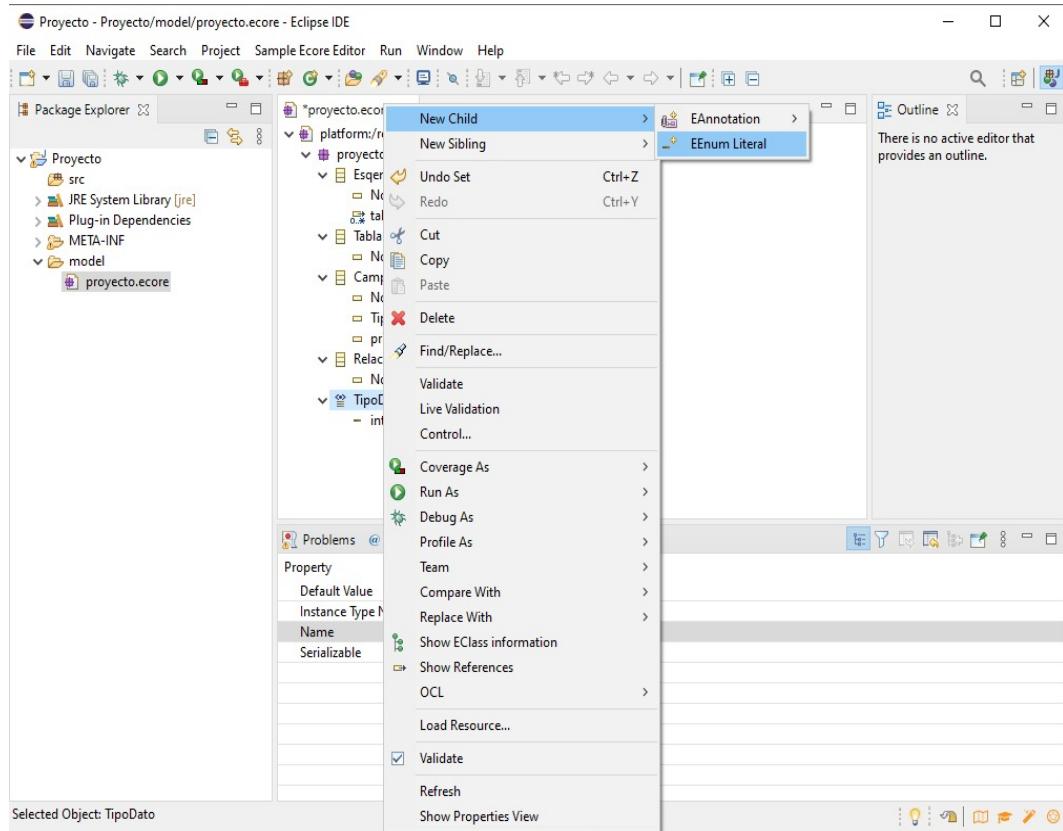


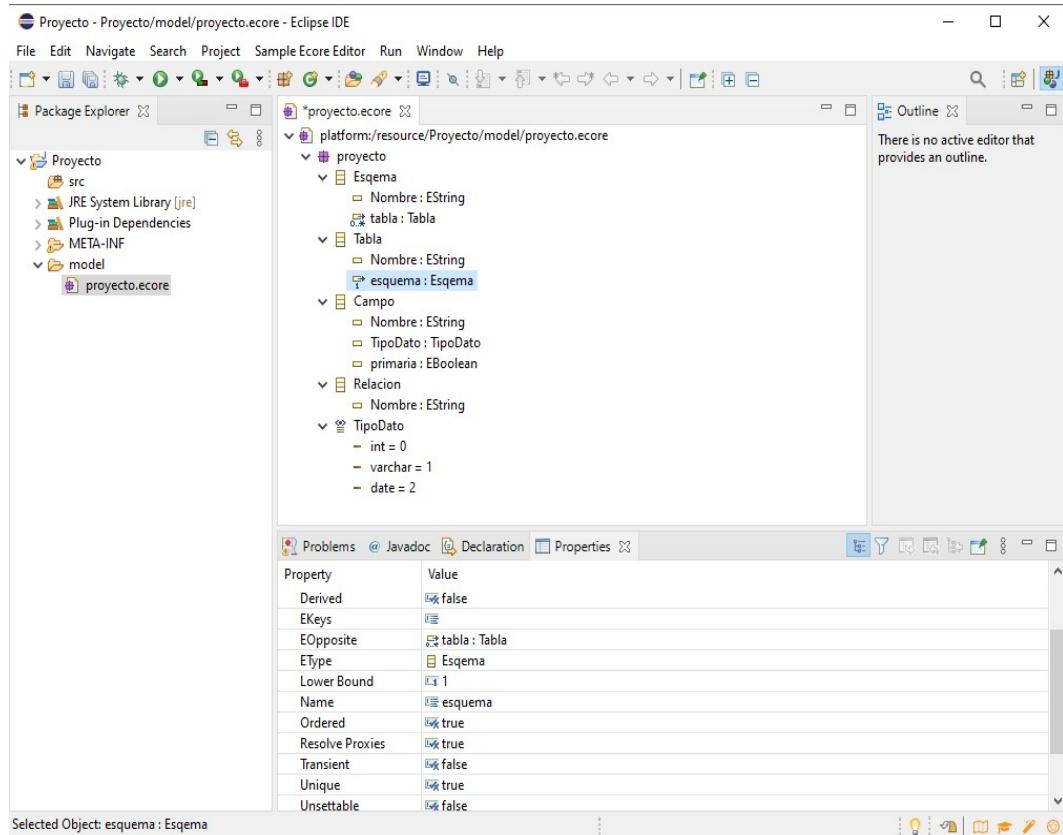


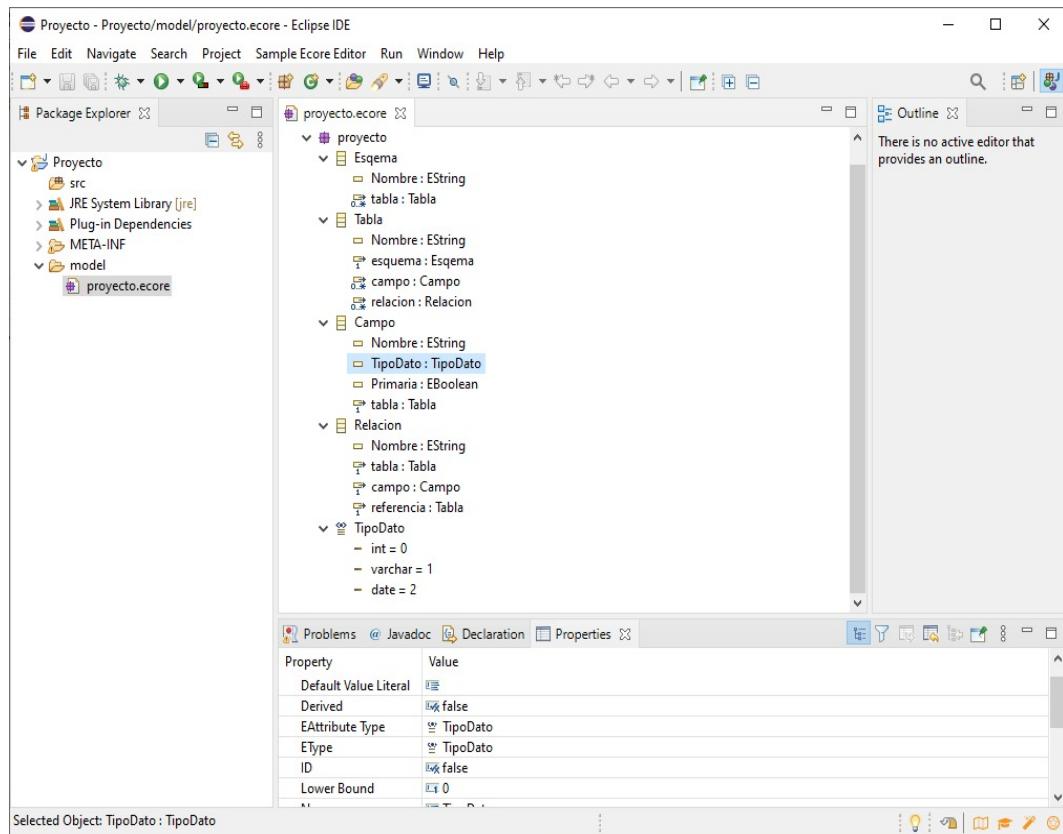
Se verifica las metaclases del ecore.

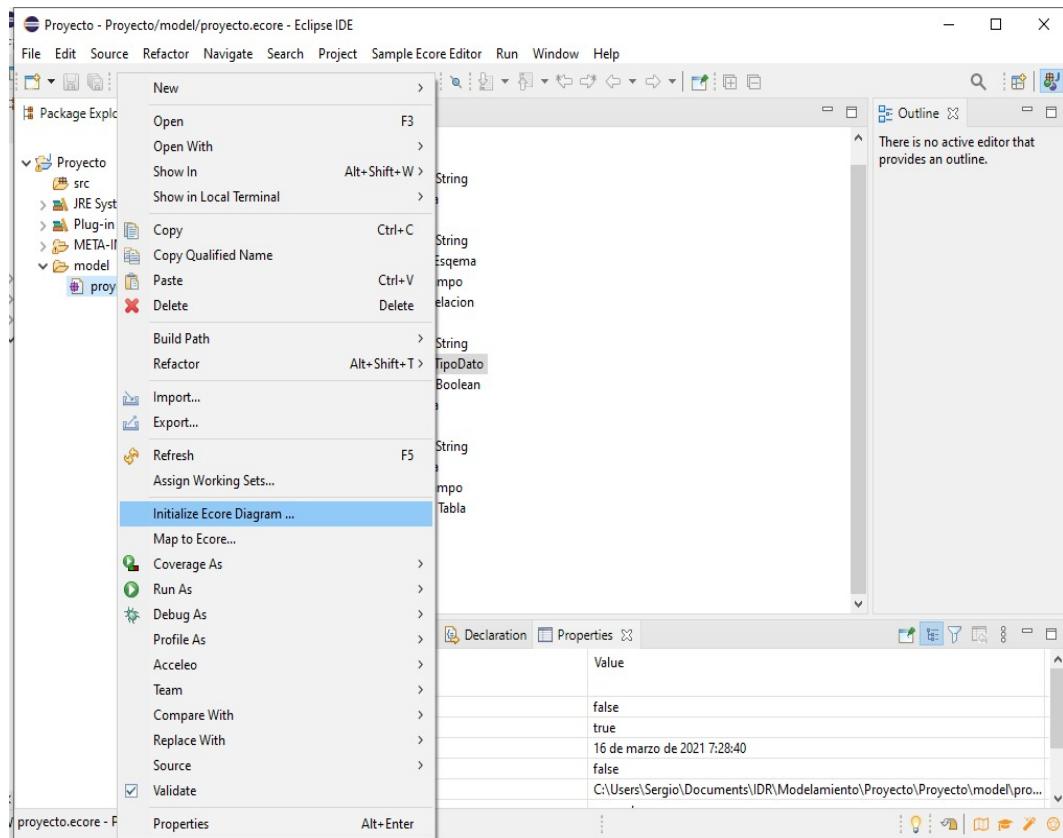


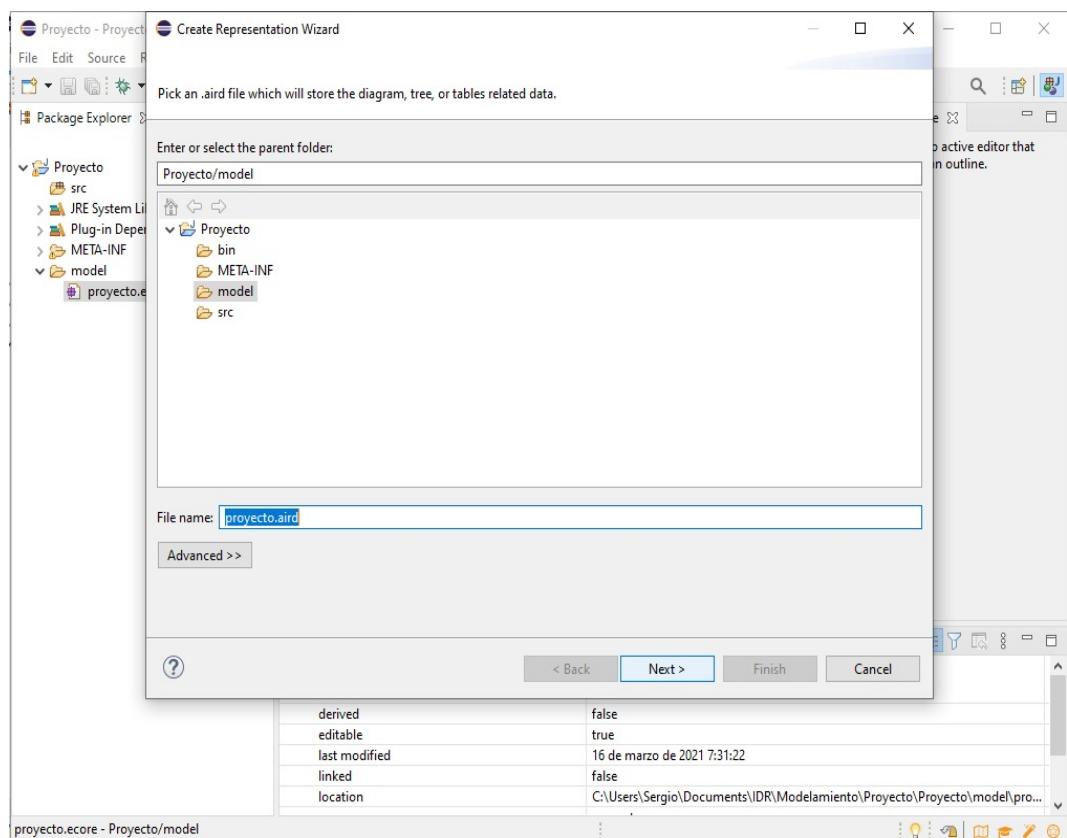


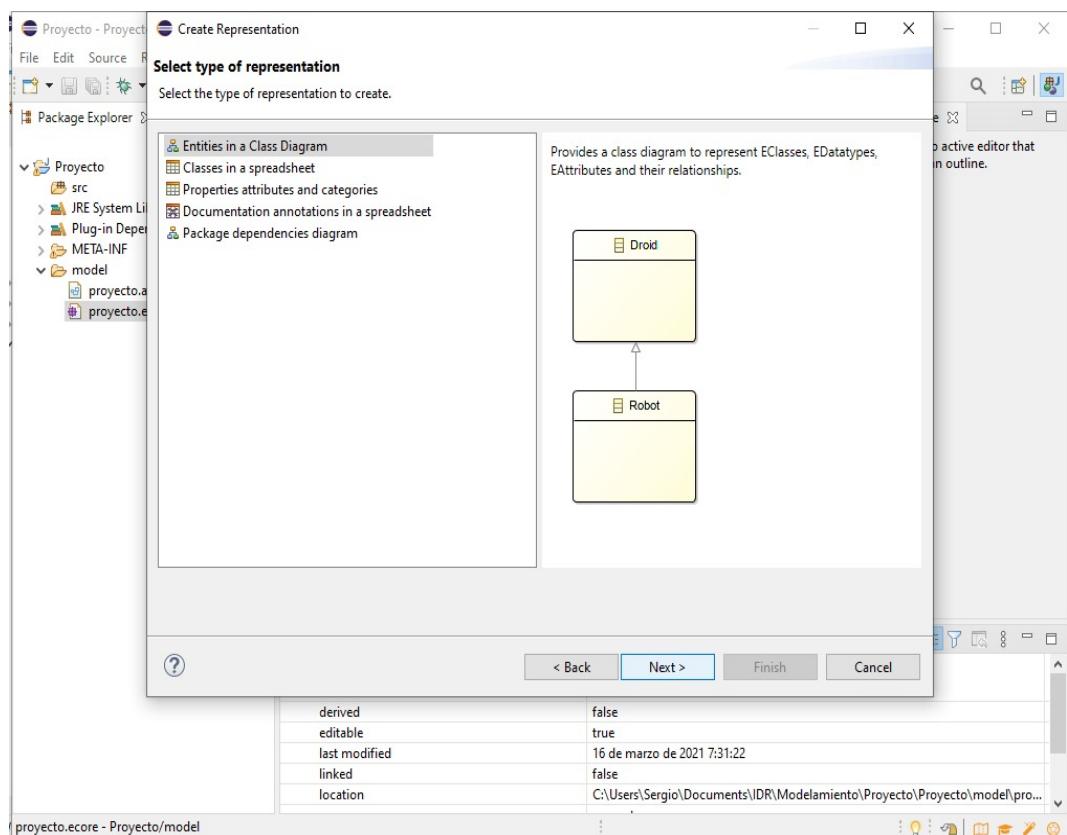


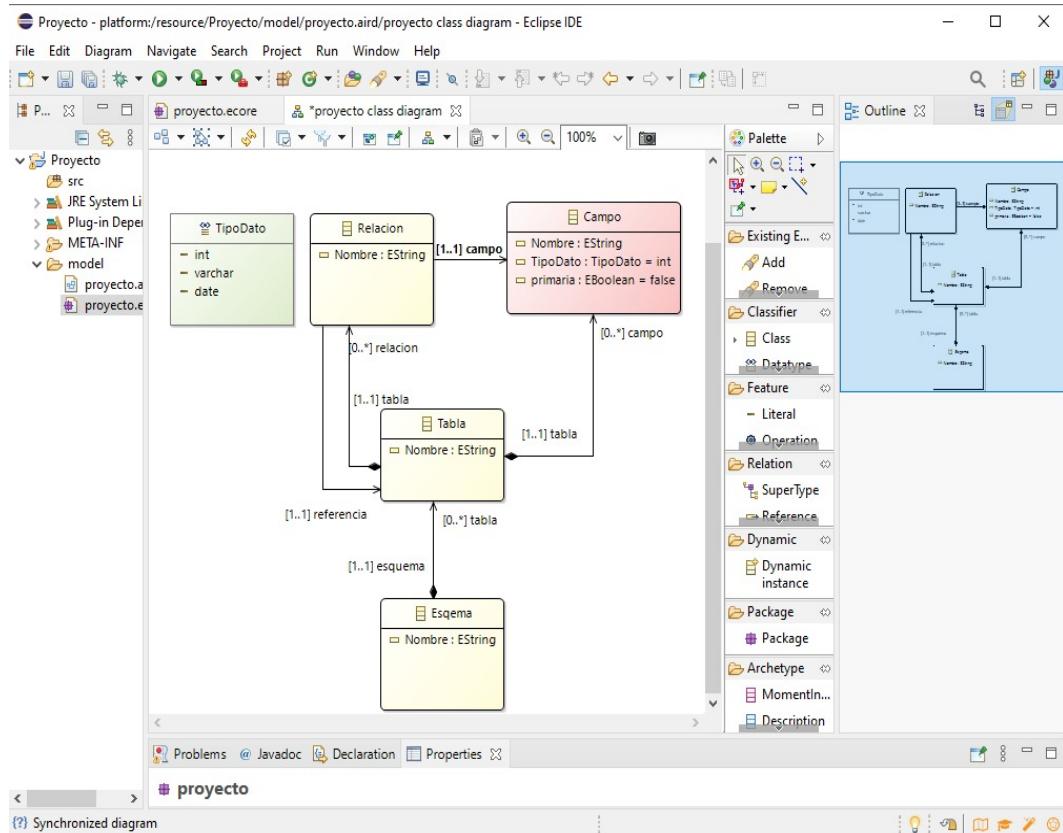








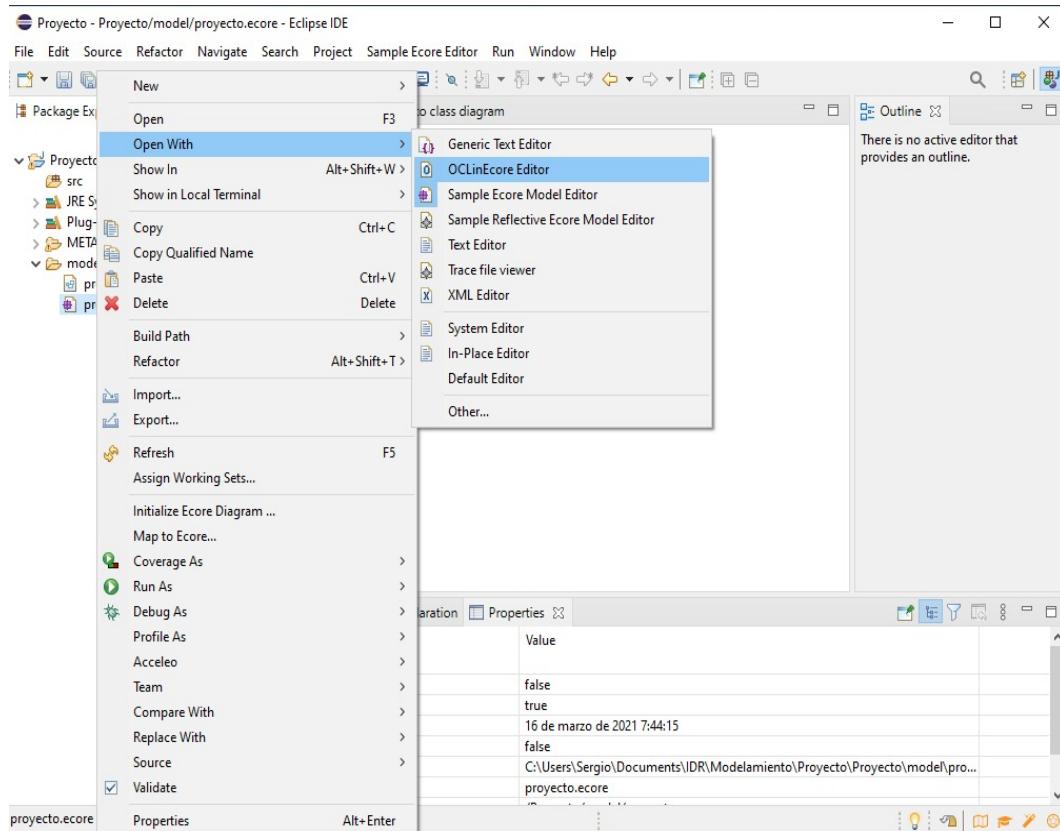


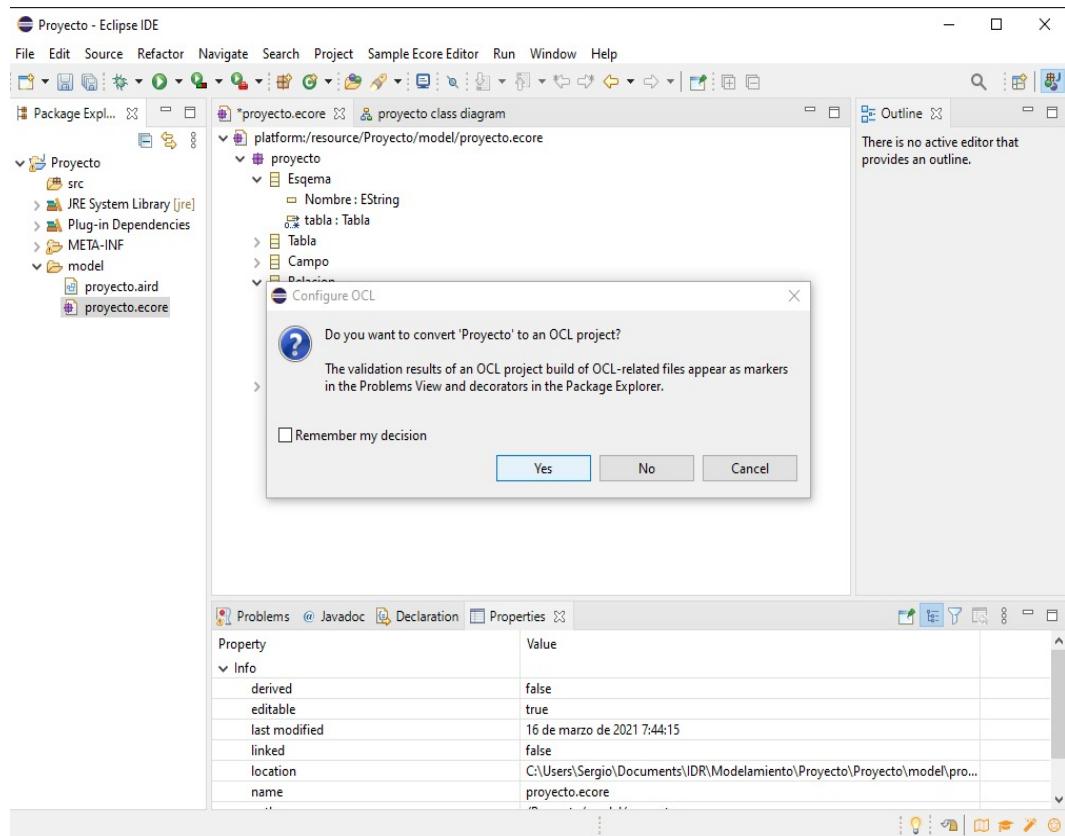


1.2 Generar SQL A Partir De Un Metamod- elo En Acceleo.

Para eso previamente deberemos tener instalado Acceleo en nuestro Eclipse, después.

Se agrega las OCL para declarar las restricciones que se deben cumplir en la aplicación.





Se agregan las restricciones.

```

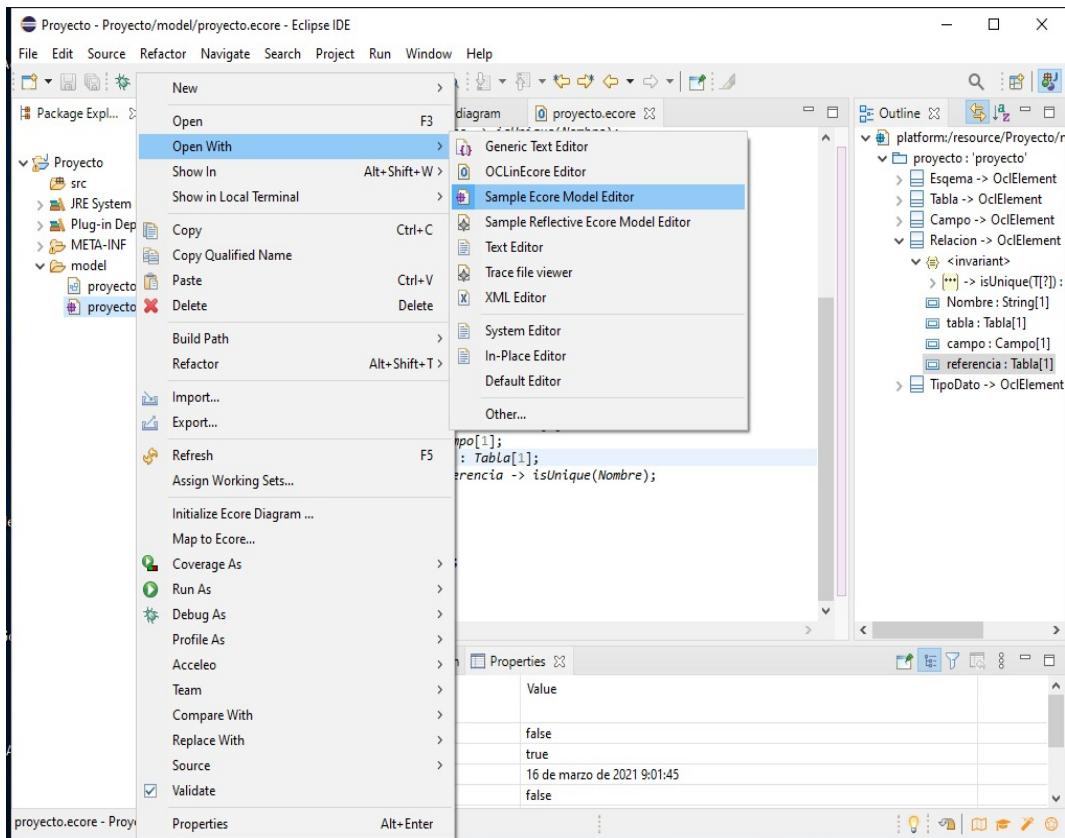
1<package proyecto : proyecto = 'projeto'
2{
3    class Esquema
4    {
5        attribute Nombre : String[1];
6        property tabla#esquema : Tabla[*|1] { ordered composes };
7        invariant: self.tabla -> isUnique(Nombre);
8    }
9    class Tabla
10   {
11        attribute Nombre : String[1];
12        property esquema#tabla : Esquema[1];
13        property campo#tabla : Campo[*|1] { ordered composes };
14        property relacion#tabla : Relacion[*|1] { ordered composes };
15        invariant: self.campo -> isUnique(Nombre);
16    }
17    class Campo
18    {
19        annotation '_http://www.obeo.fr/dsl/dnc/archetype'
20        (
21            archetype = 'MomentInterval'
22        );
23        attribute Nombre : String[1];
24        attribute TipoDatos : TipoDatos[?|1] { transient };
25        property tabla#campo : Tabla[1];
26        attribute primaria : Boolean[1];
27        invariant: self.Relacion -> isUnique(Nombre);
28    }
29    class Relacion

```

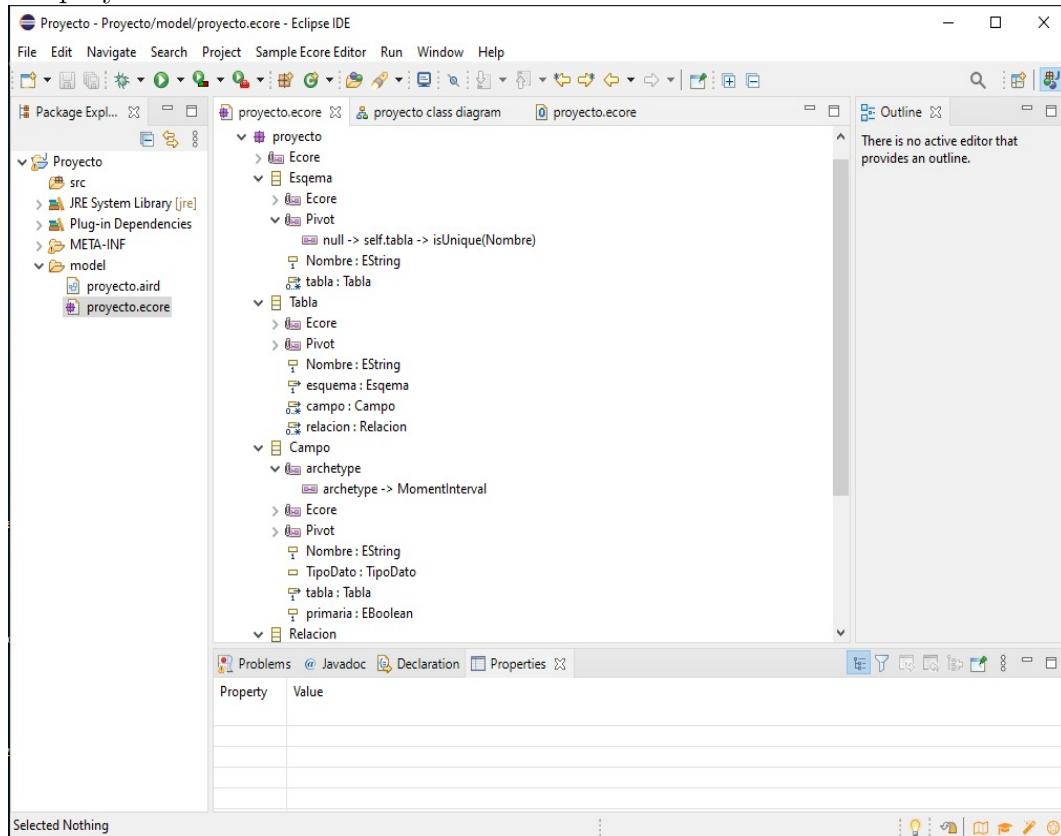
Properties View:

Property	Value
Info	
derived	false
editable	true
last modified	16 de marzo de 2021 7:44:15
linked	false

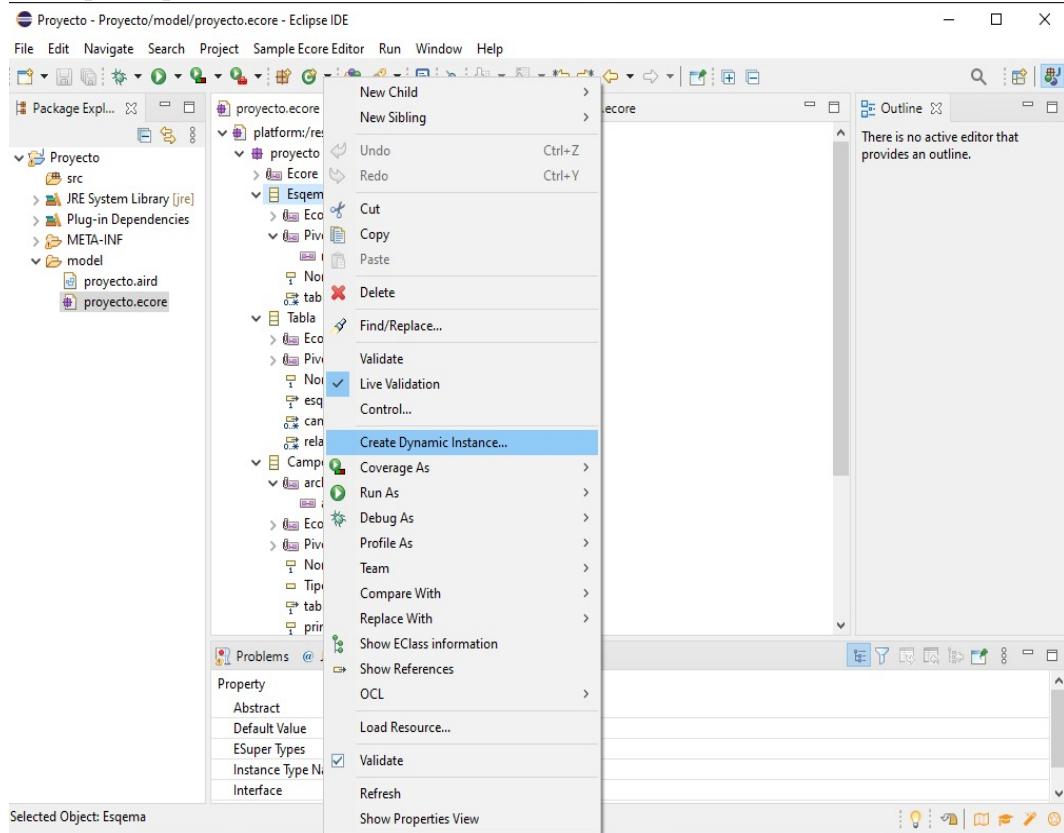
Se guarda los cambios y se procede a abrir siguiendo los pasos que se muestra en la siguiente imagen.

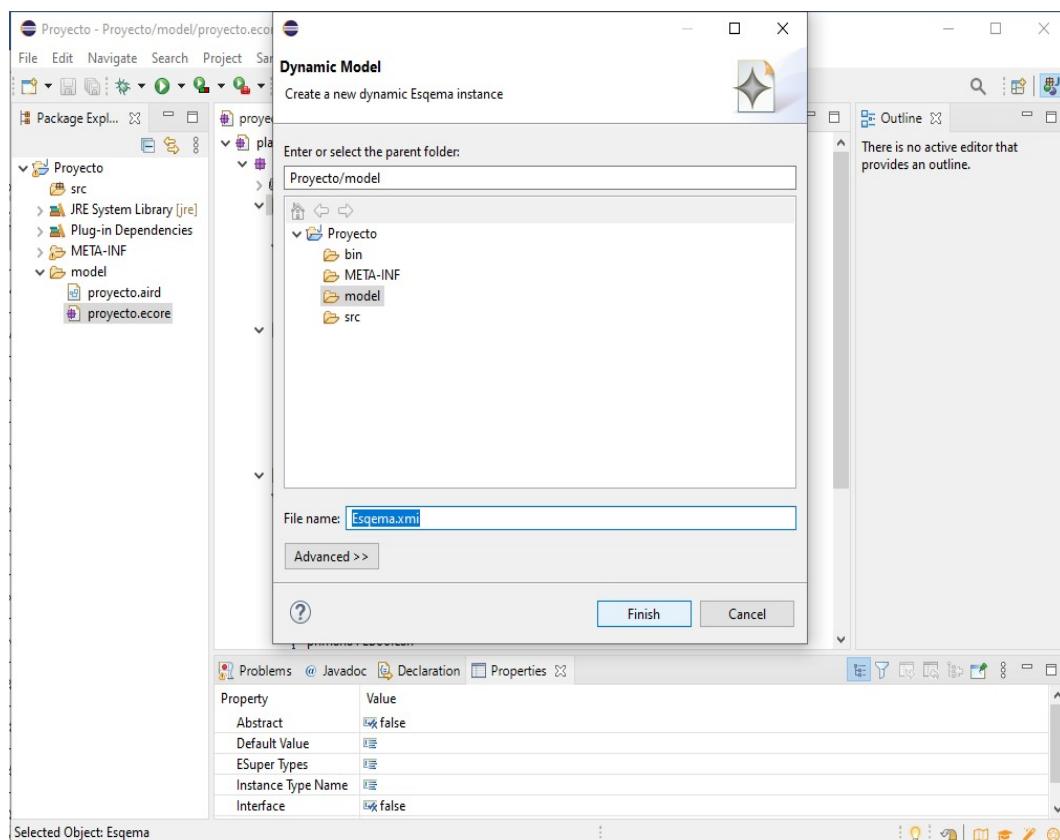


Se puede visualizar q se encuentran cargadas las OCL en el punto.ecore del proyecto.

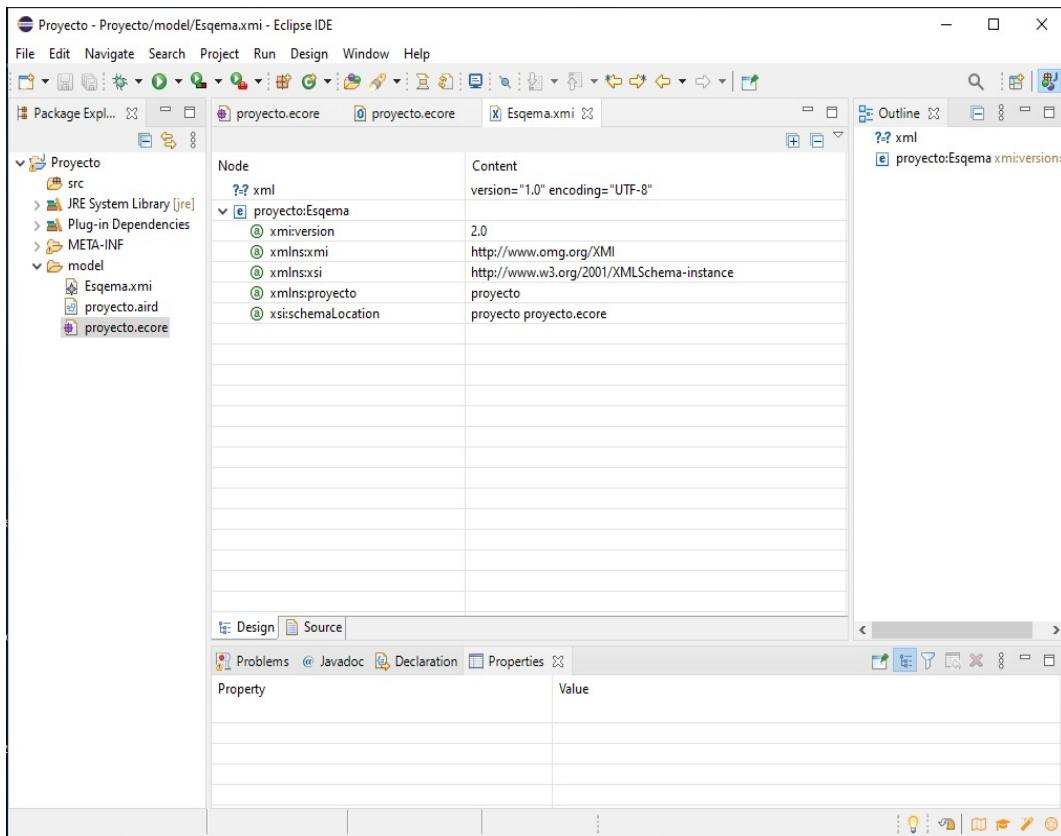


Se procede a hacer la creación de instancias dinámicas clic derecho en la clase principal.

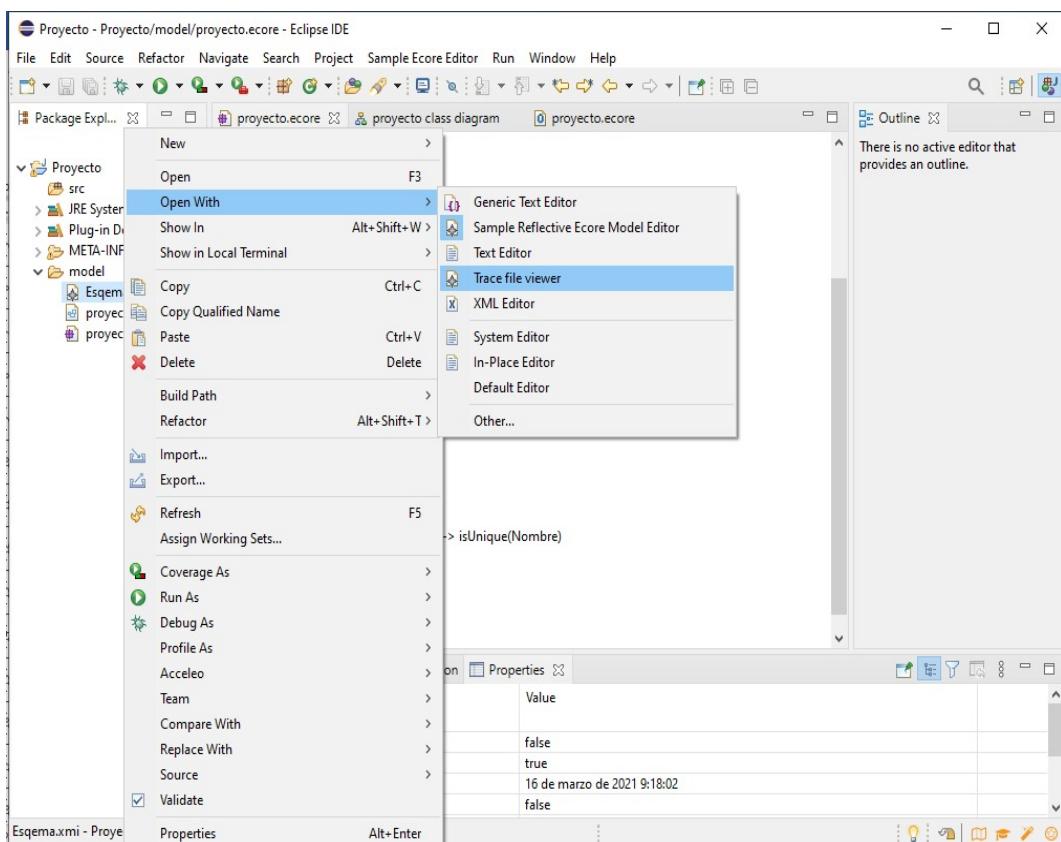


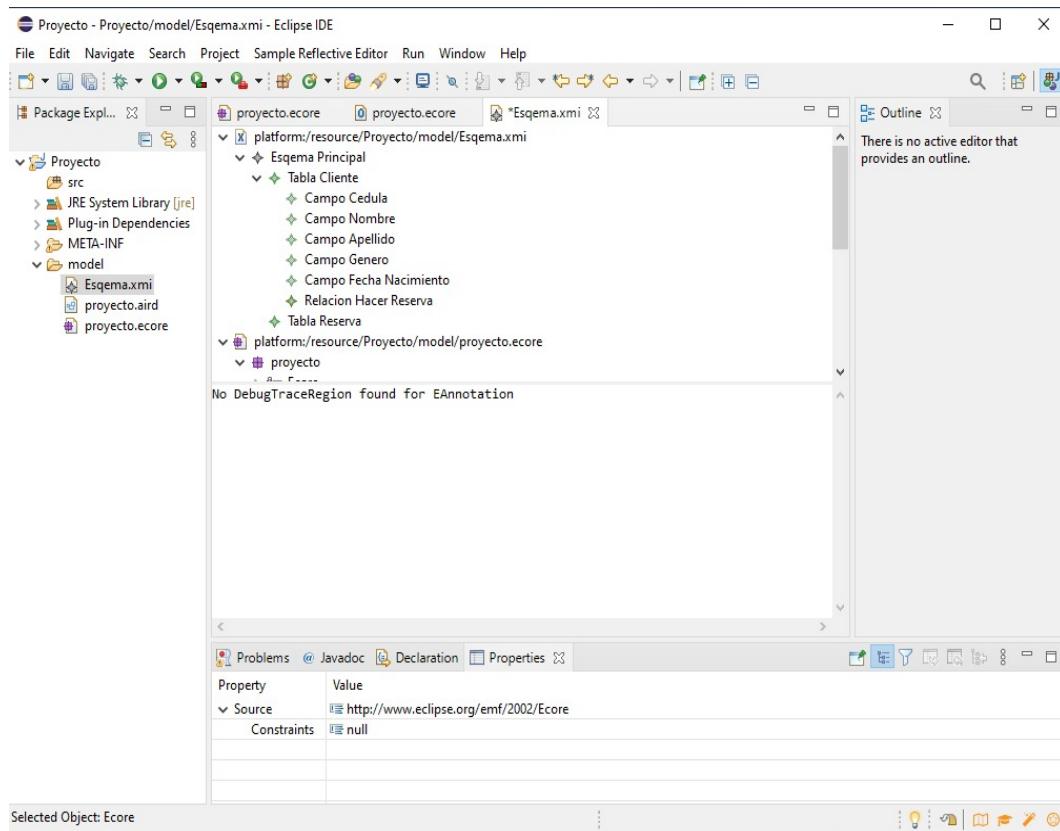


Se muestra la siguiente imagen la cual se debe cerrar.

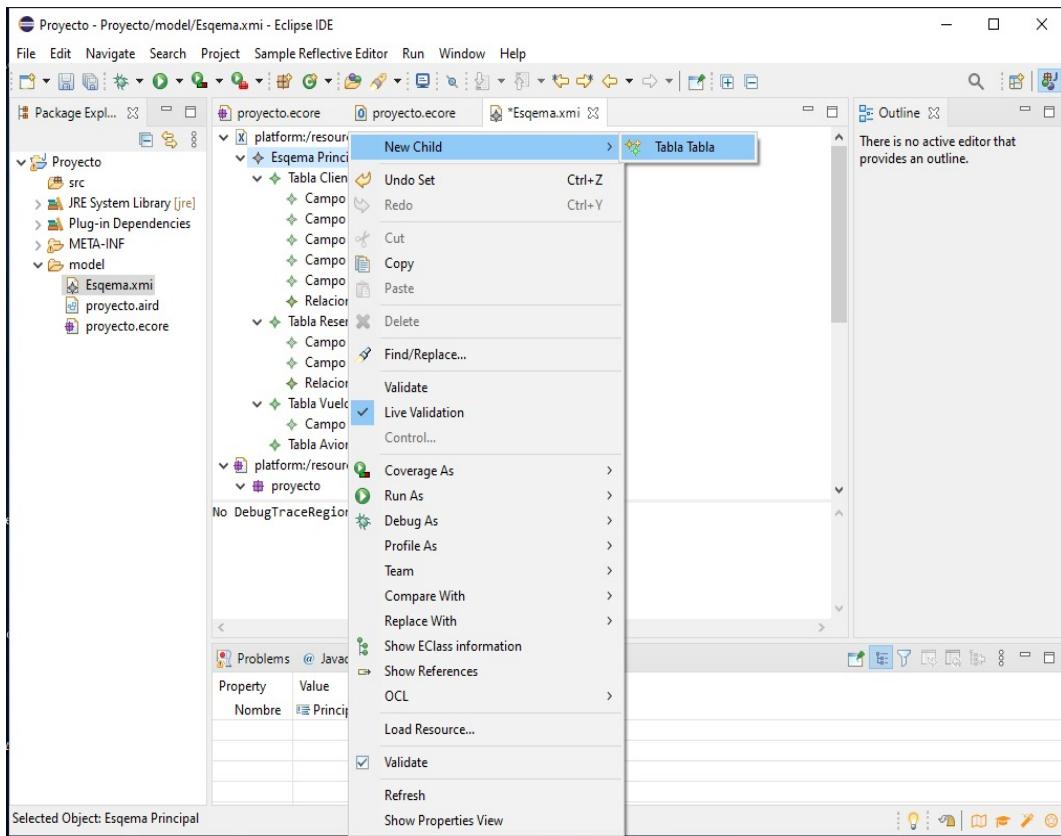


Se abrirá desde model de la siguiente manera carpeta model clic derecho.

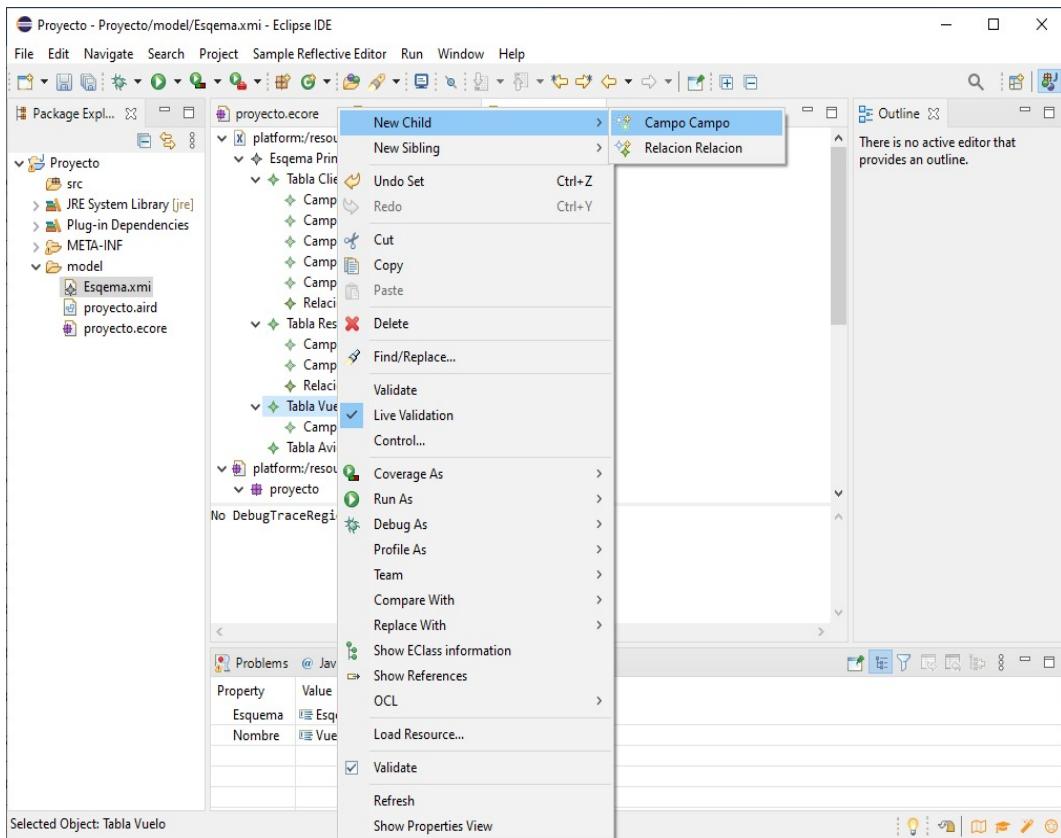


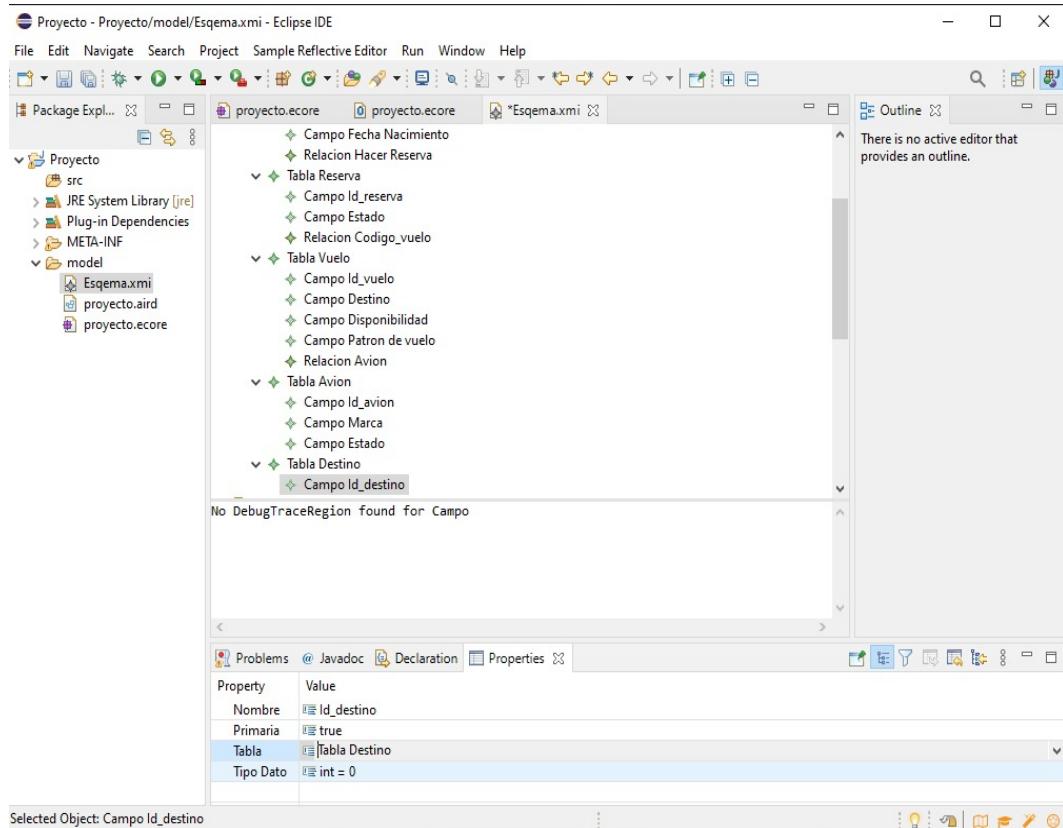


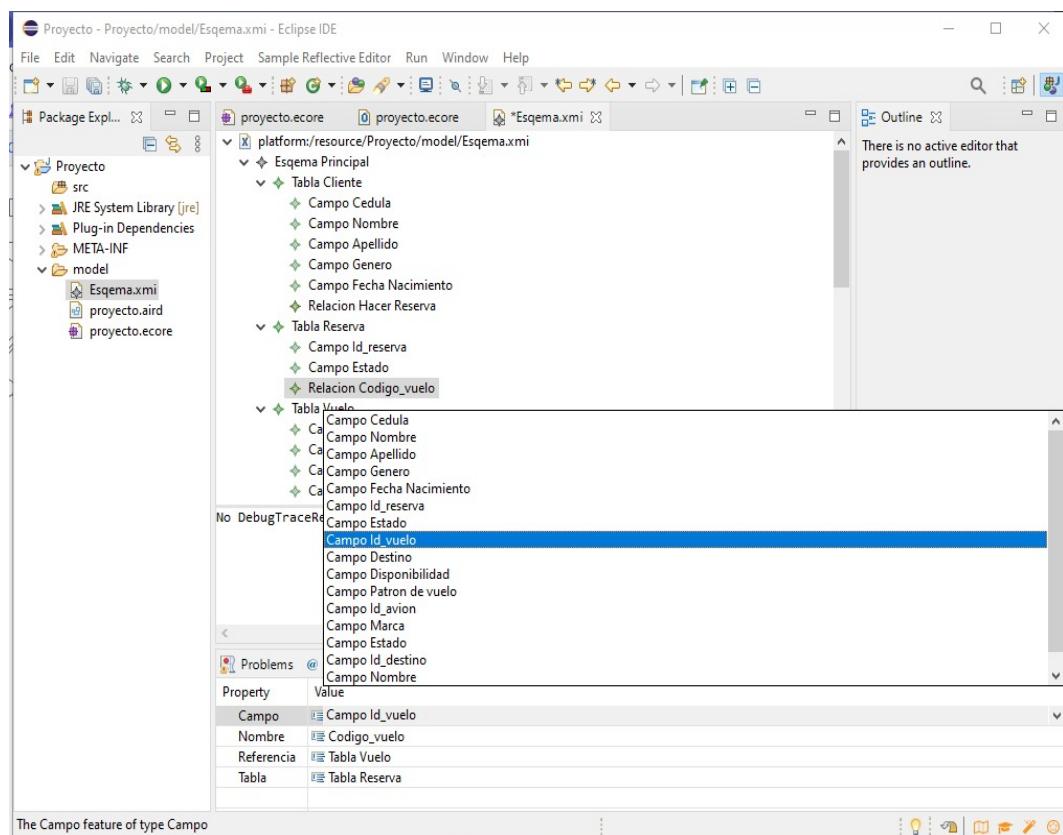
La agregación de tablas se realiza de la siguiente manera.

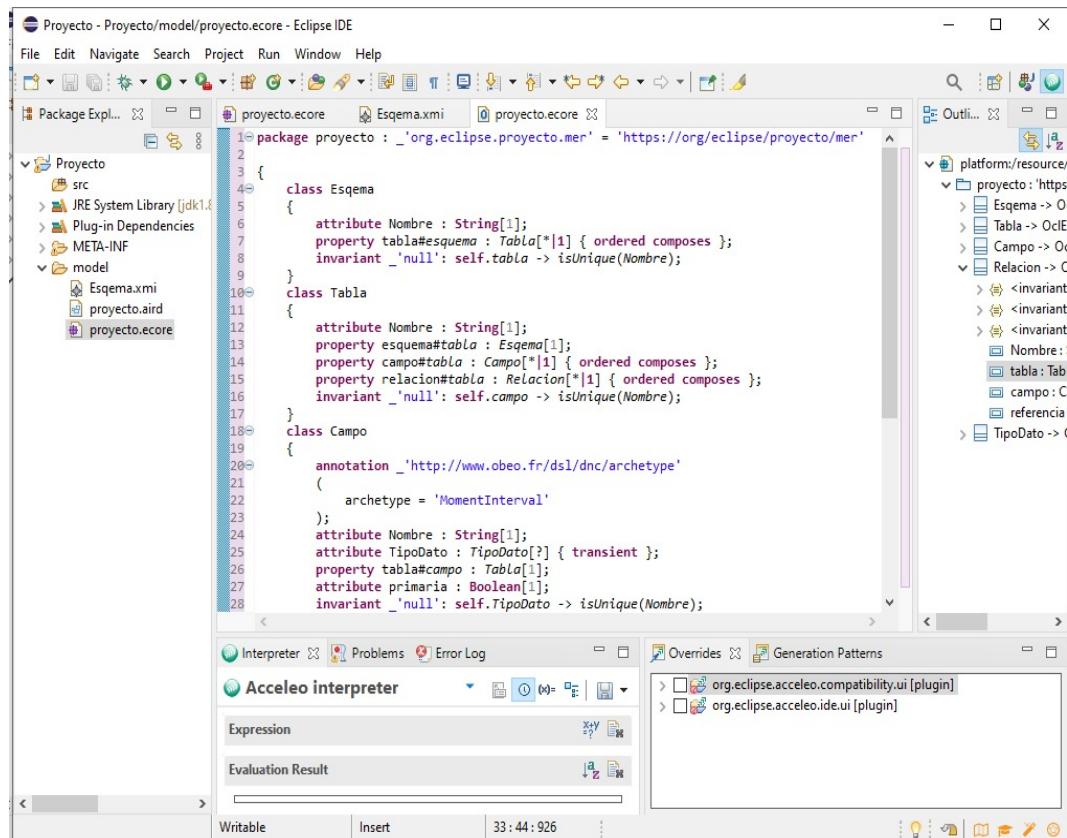


Para agregar los campos de la tabla se realiza lo siguiente.

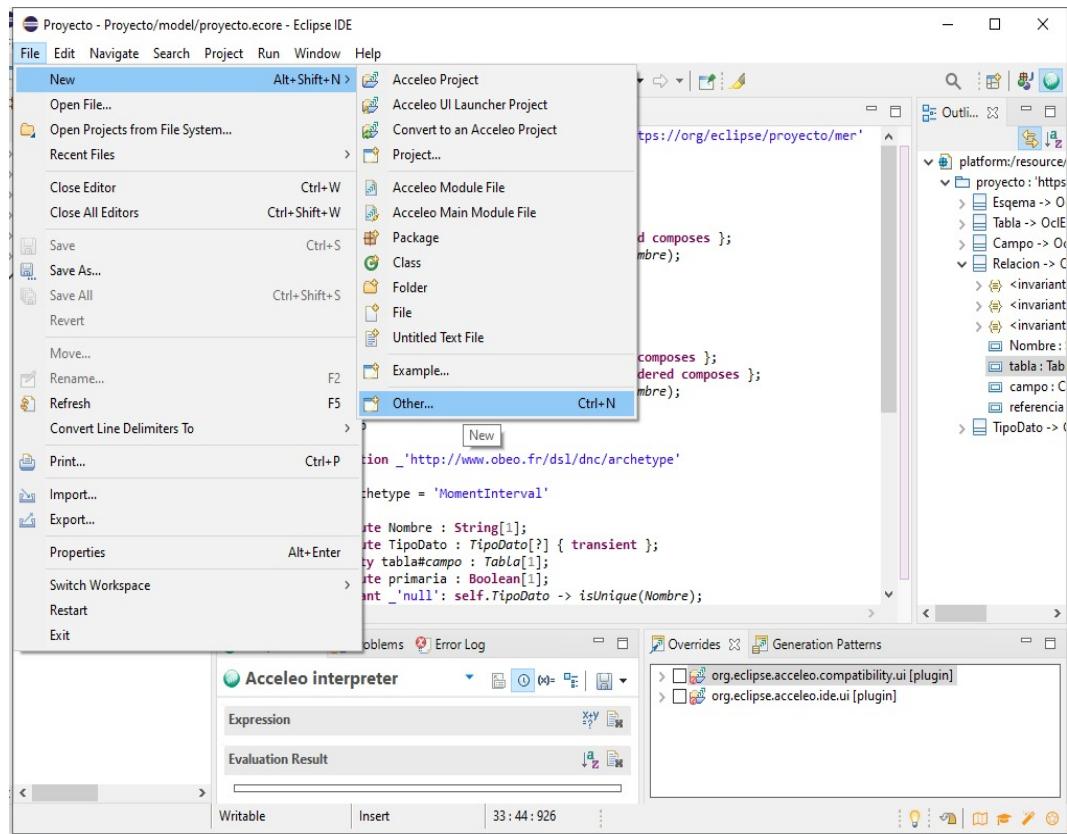




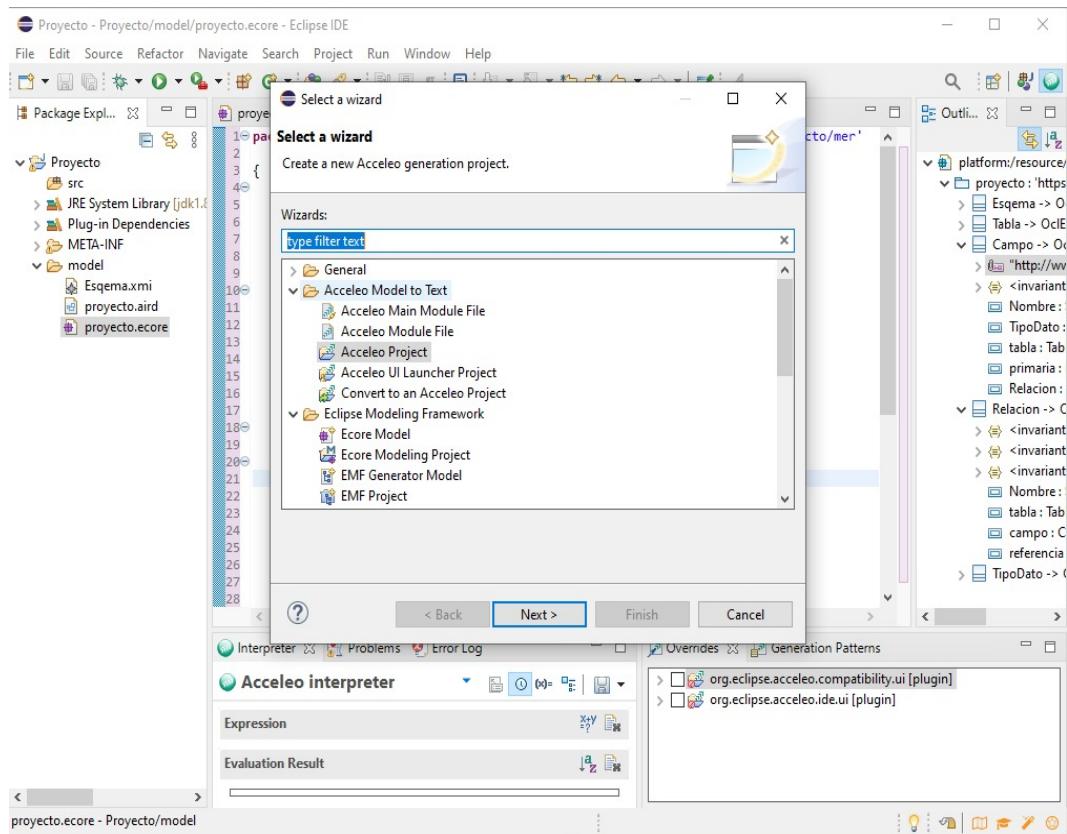


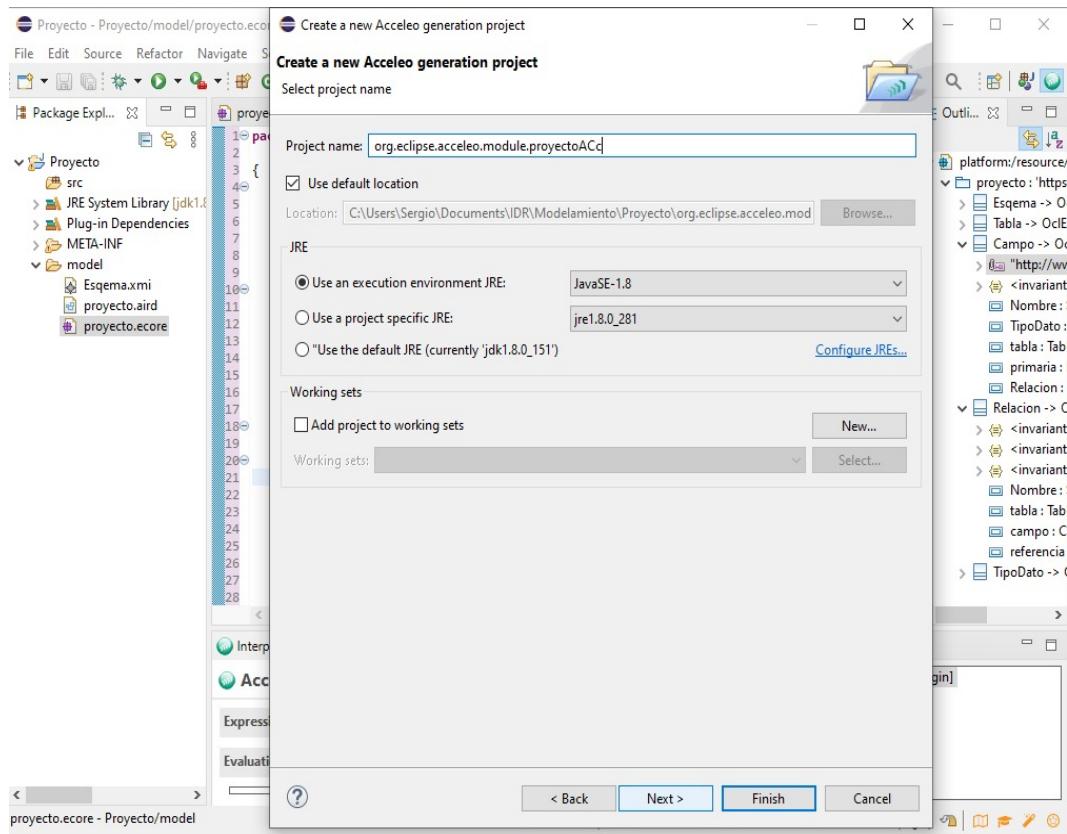


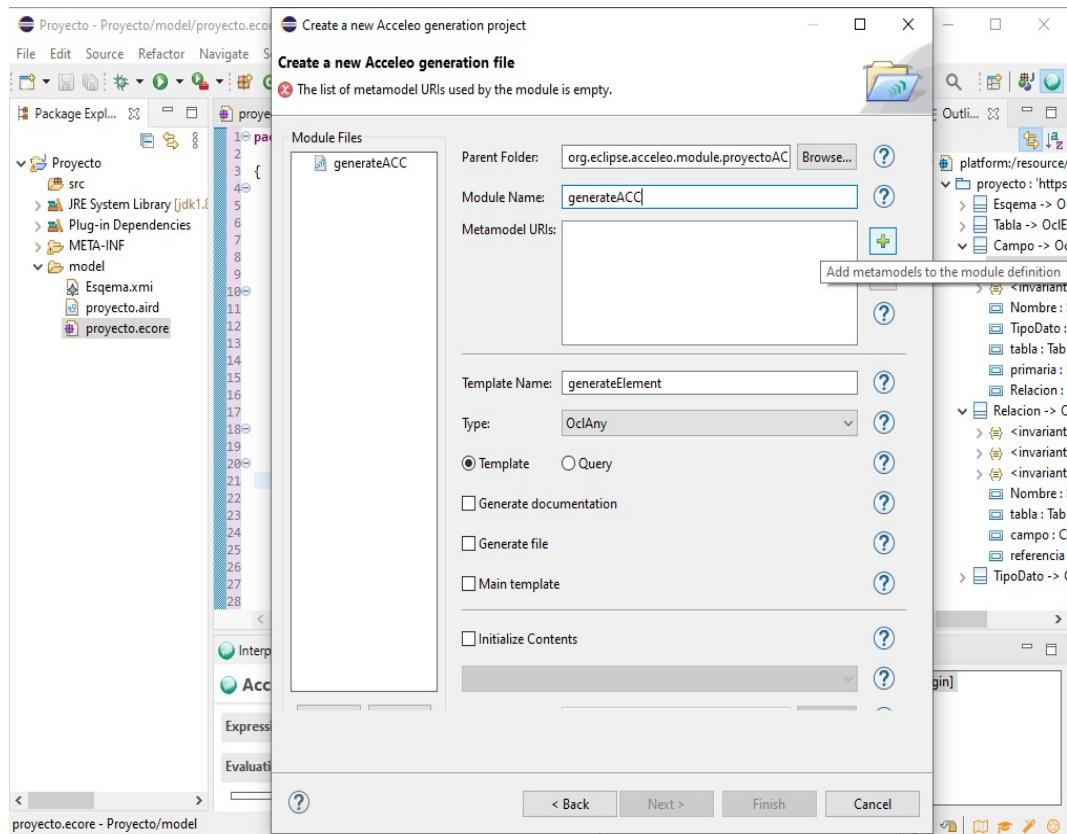
Generar código con ACELEO se ubica en el ecore.

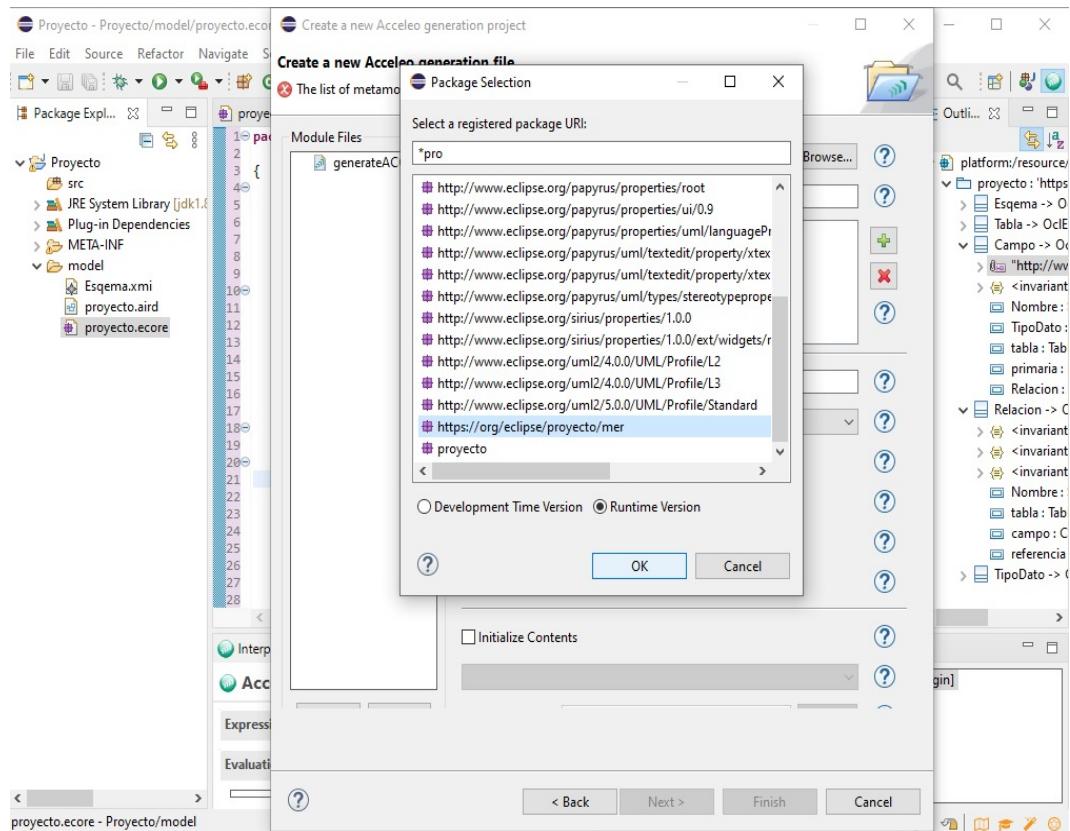


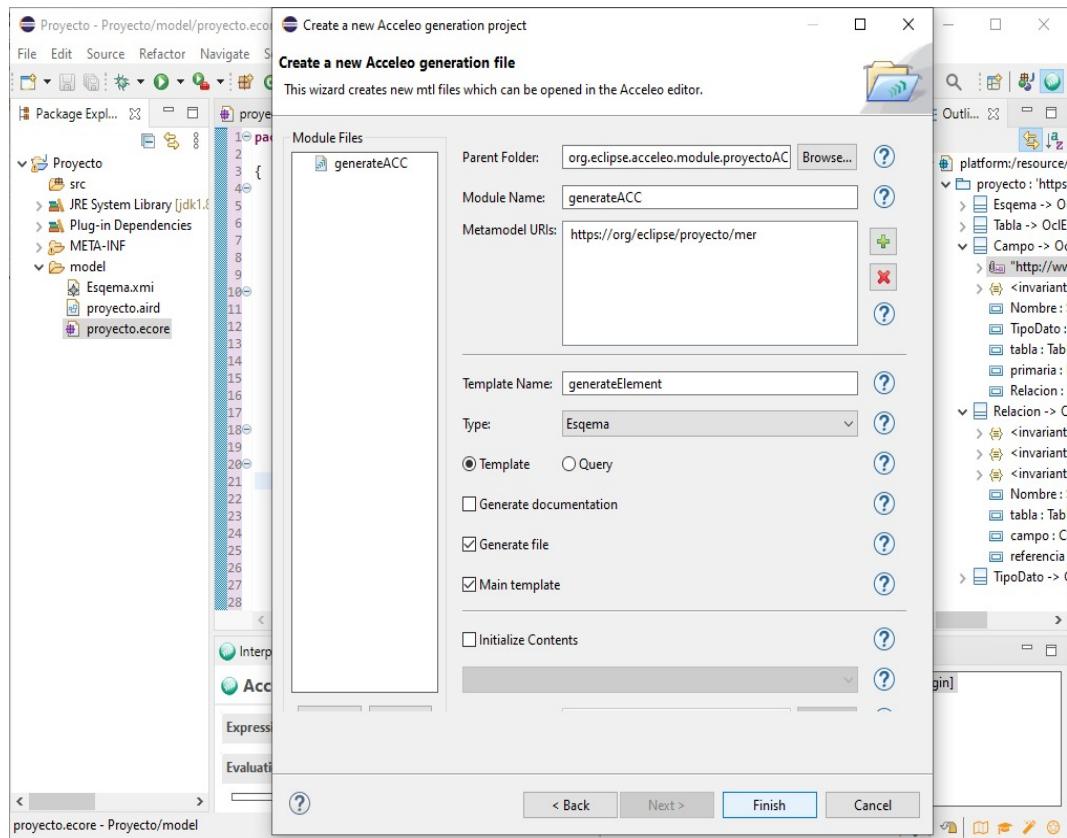
Luego se realiza un nuevo proyecto.

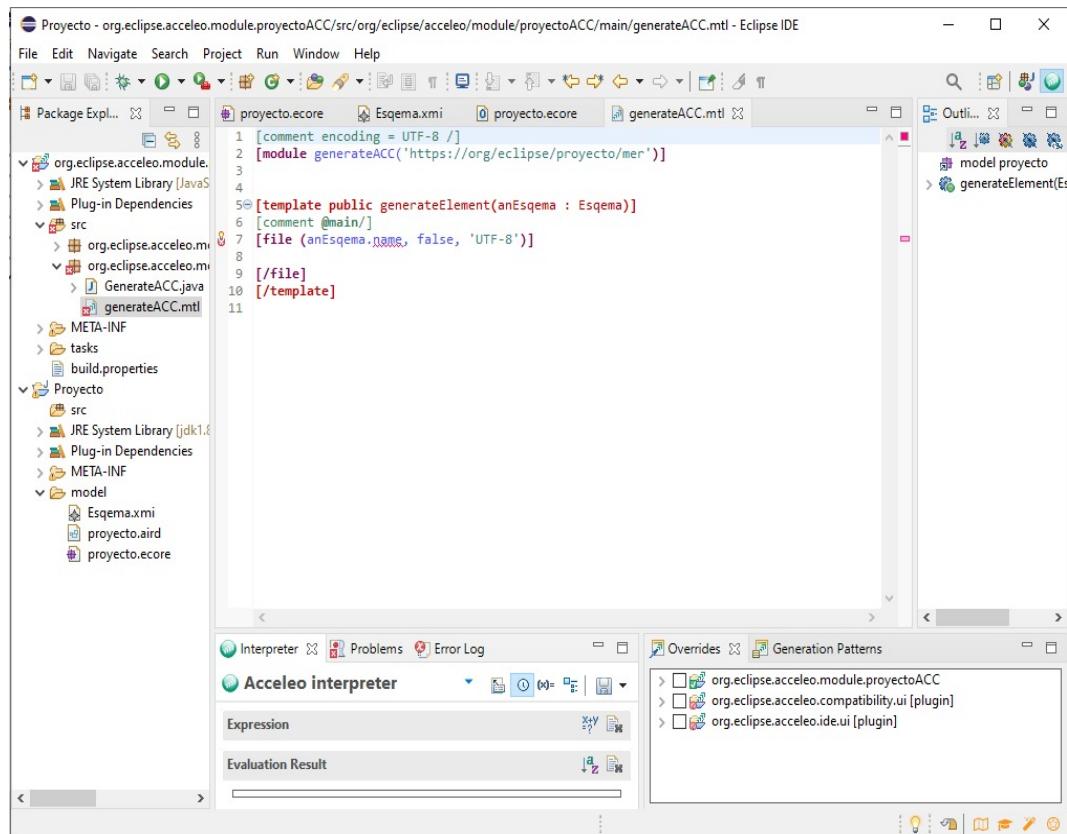


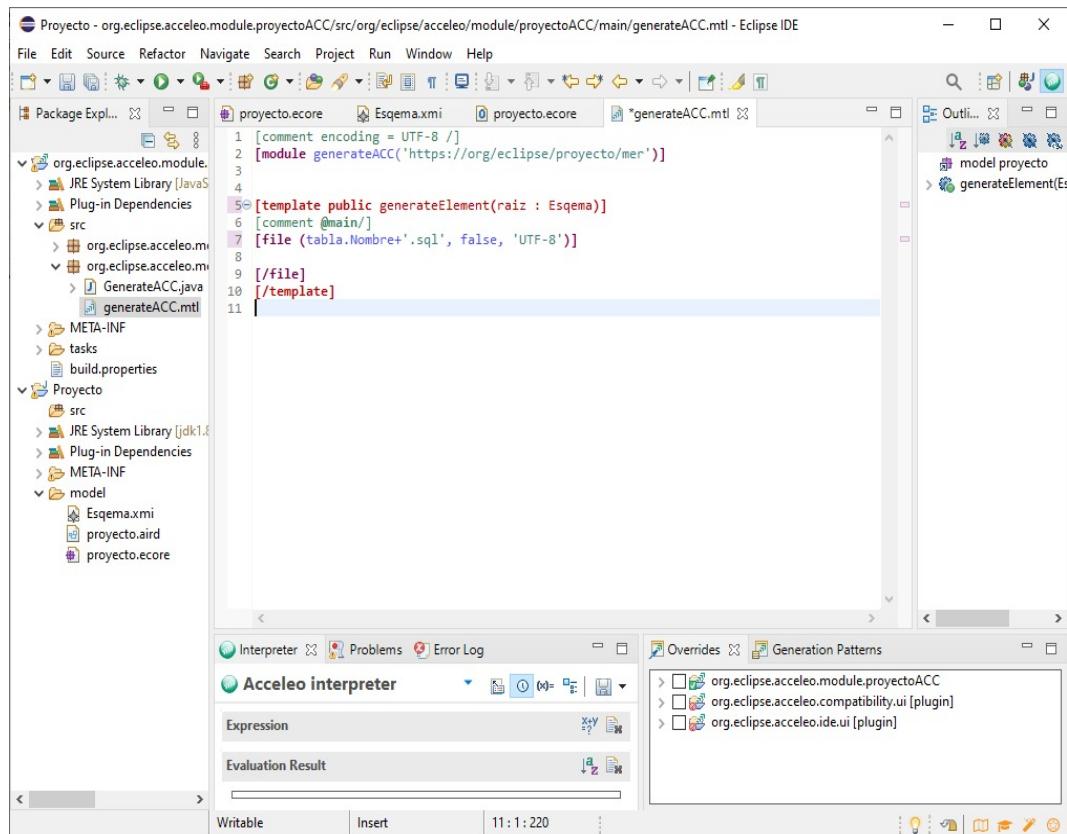


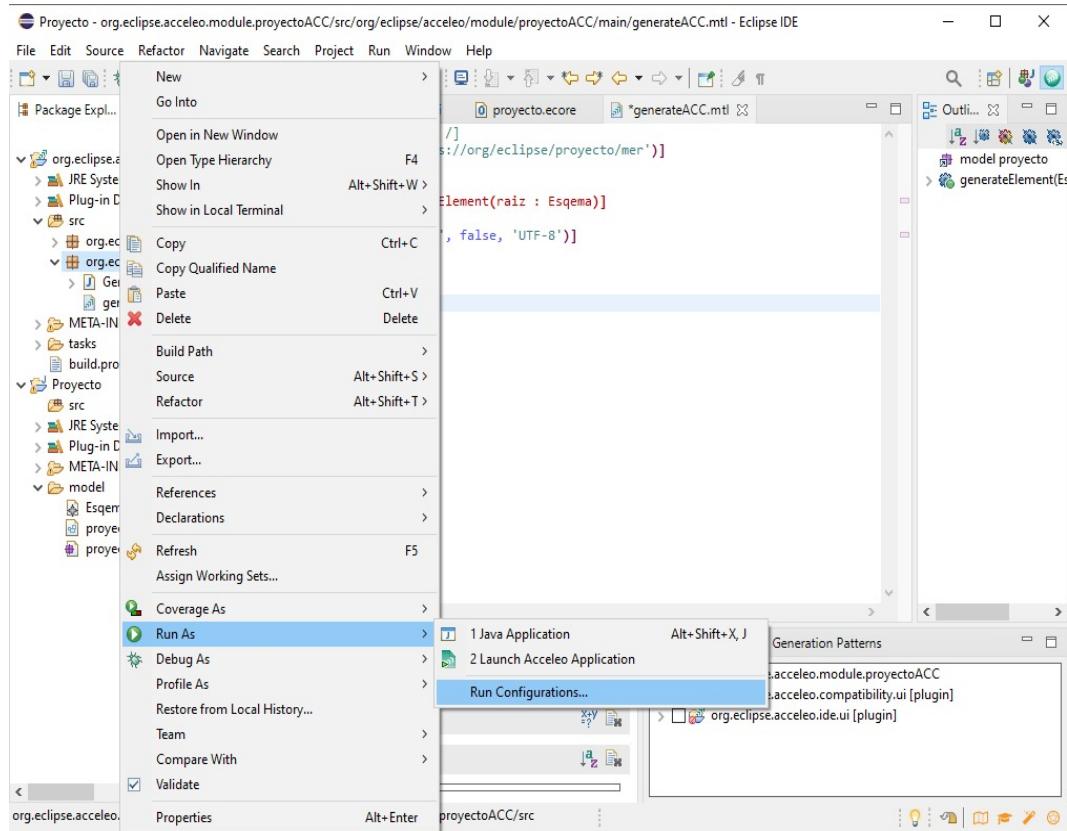


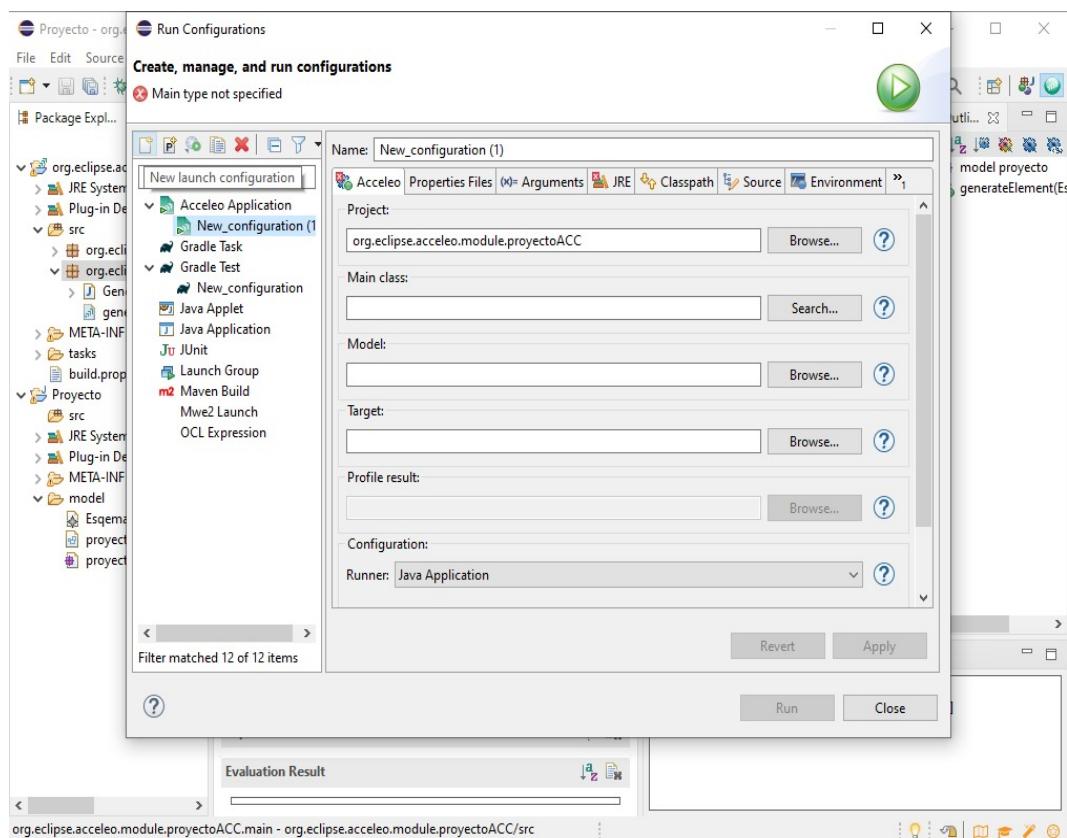


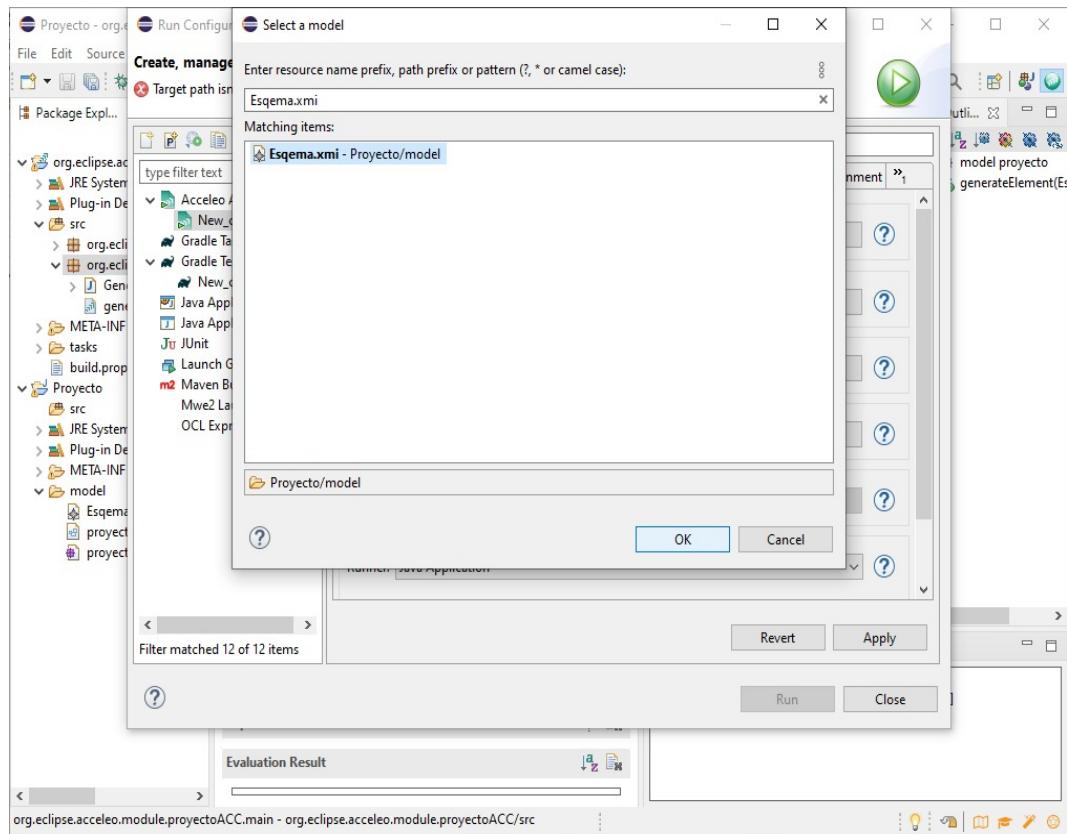


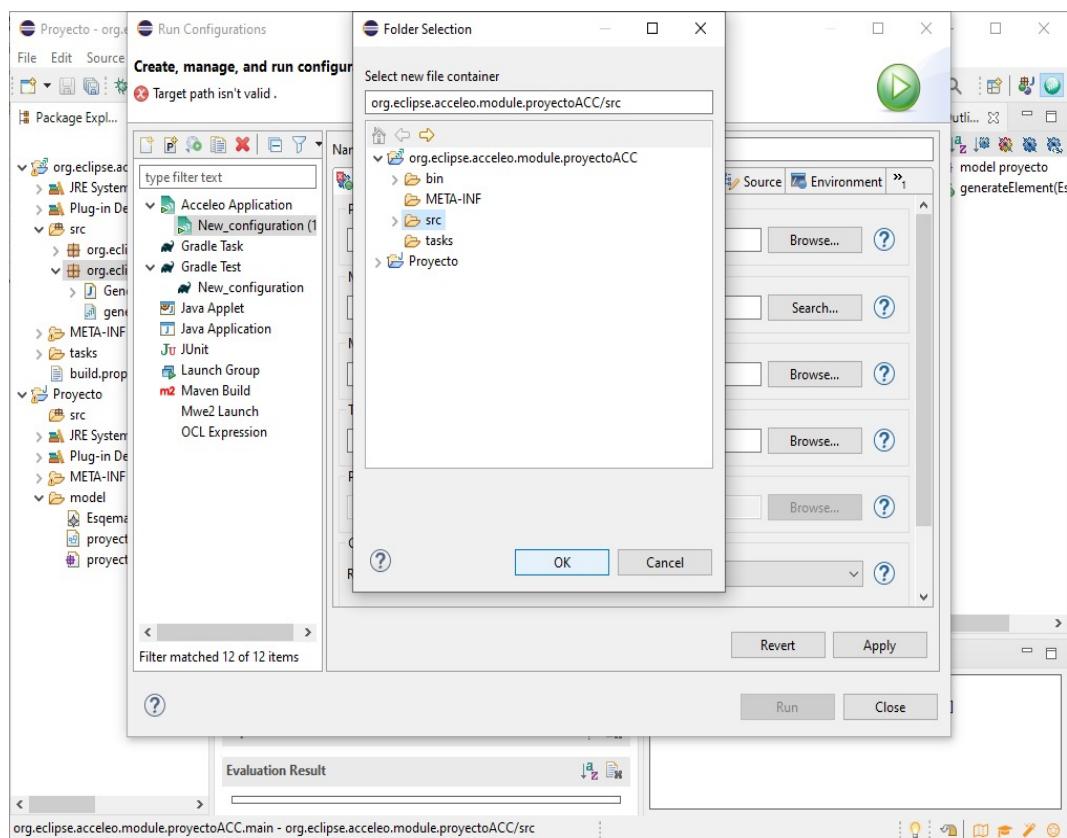


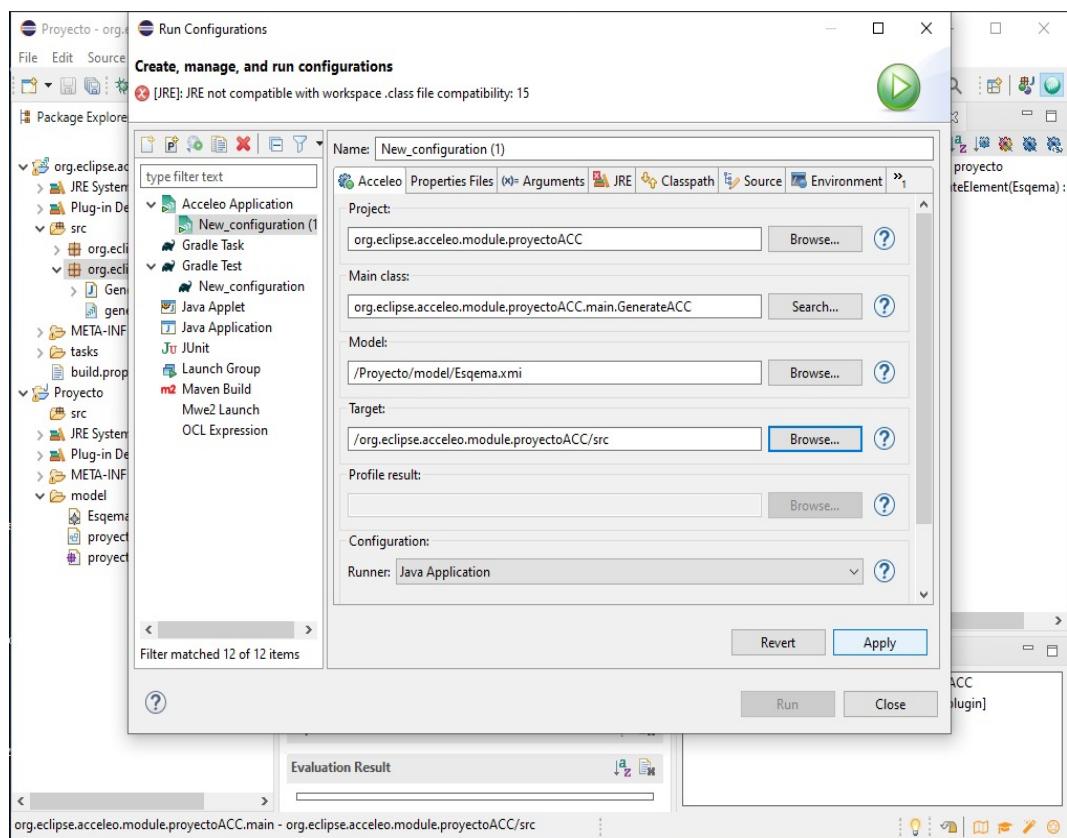


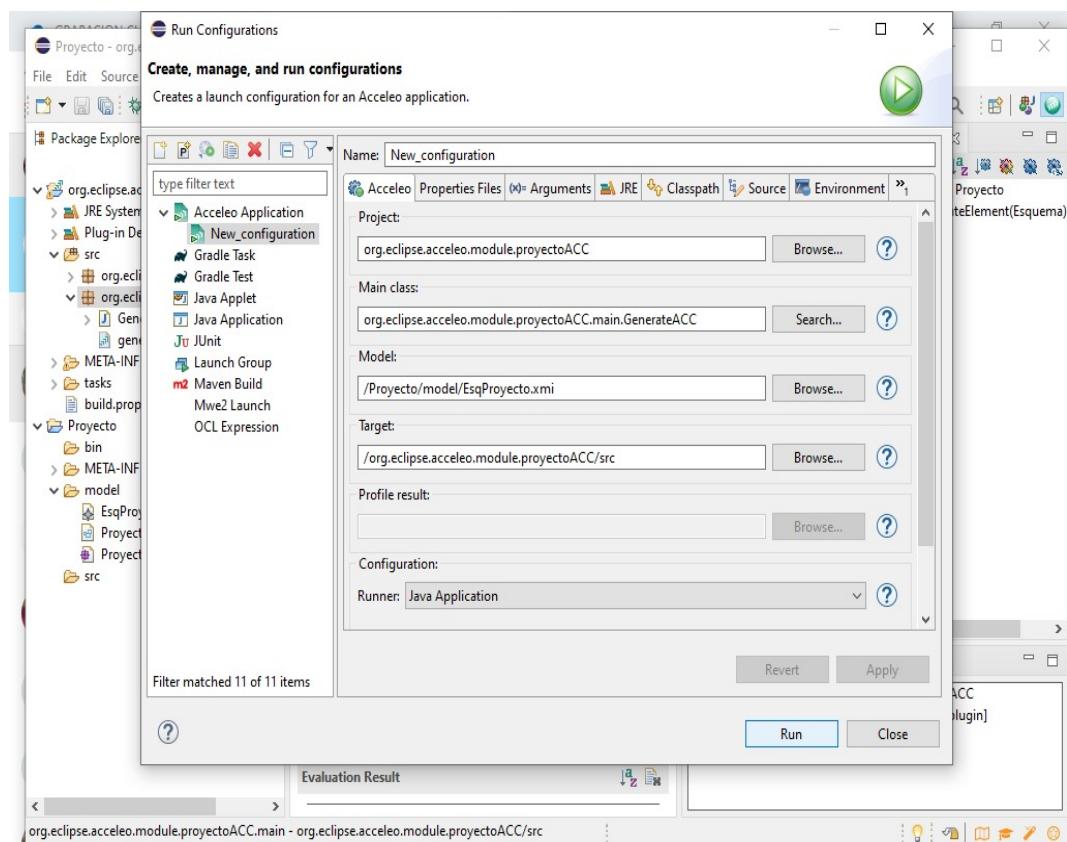


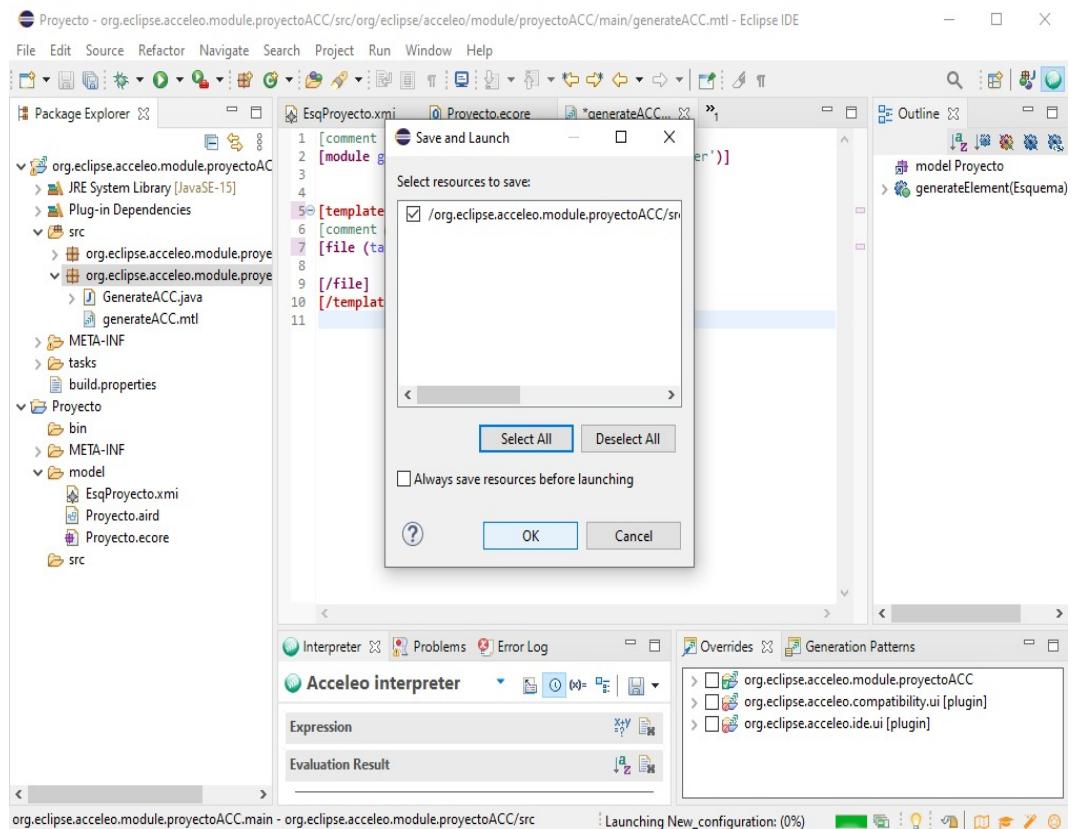


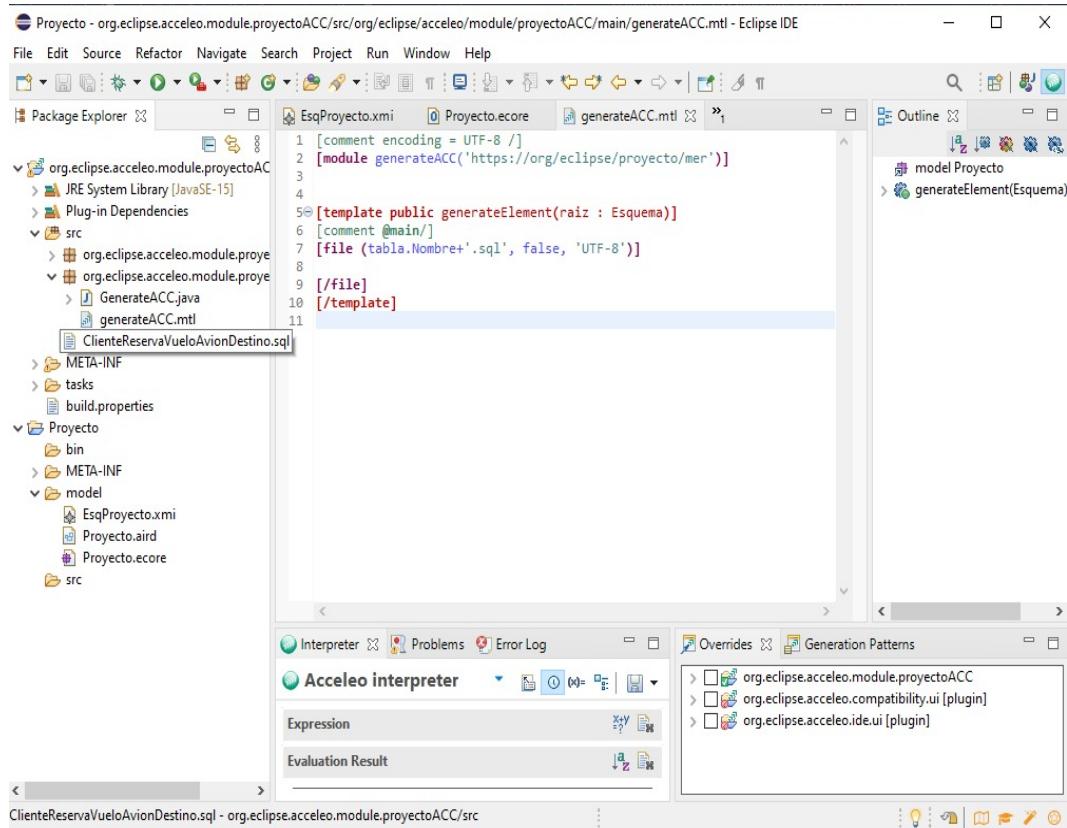




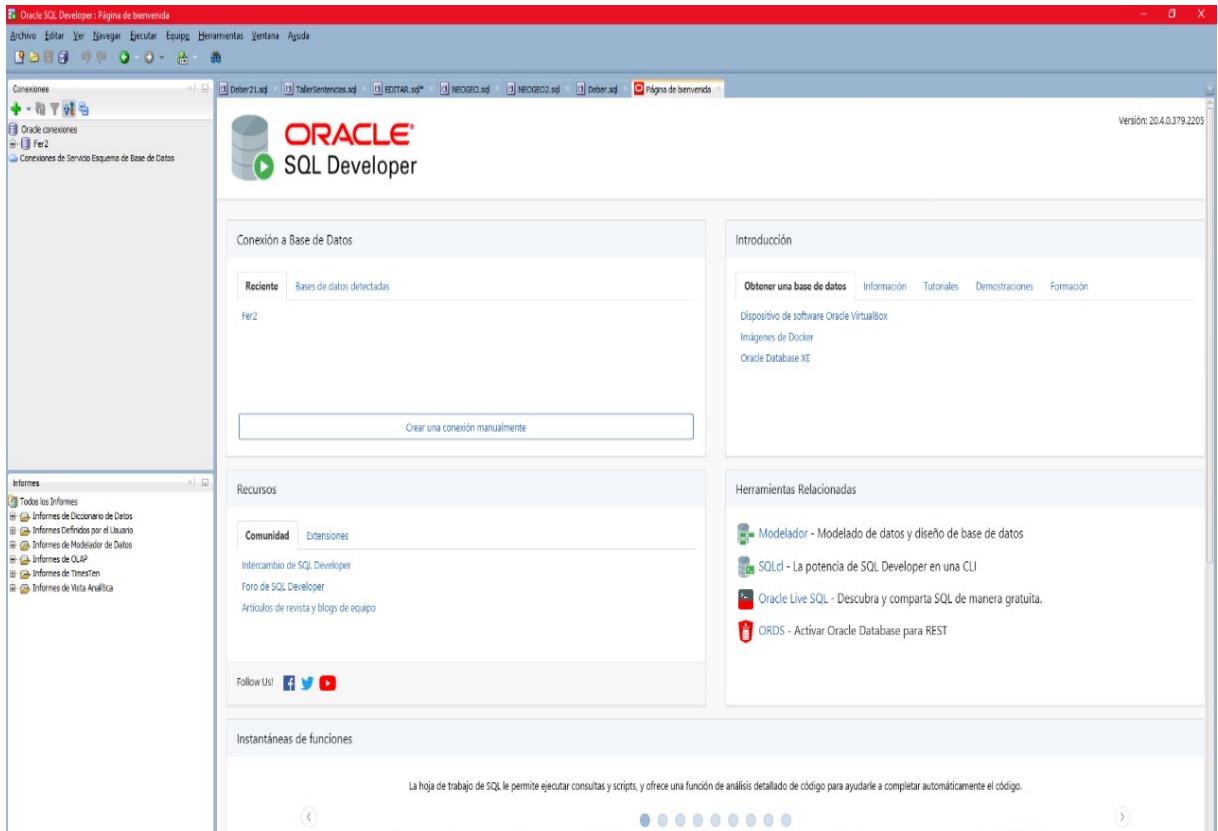




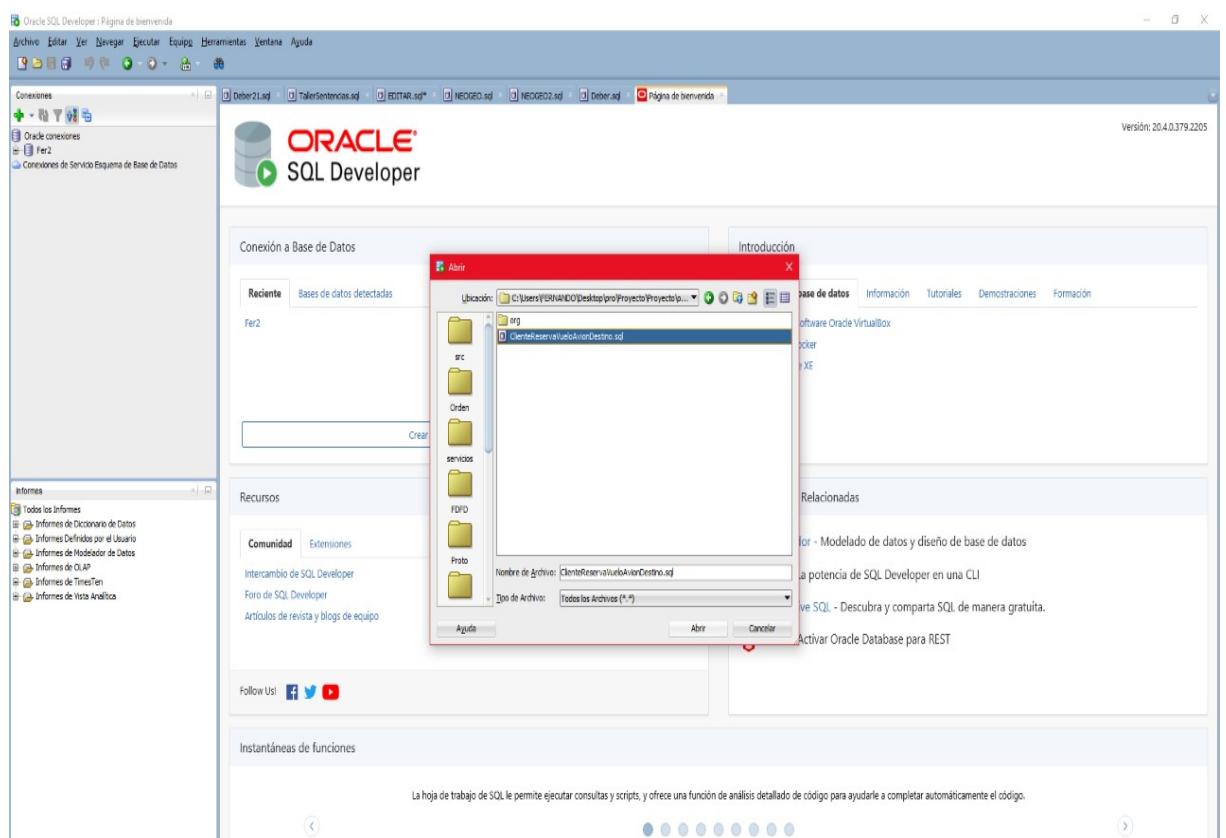




1.3 Sentencias en SQL



Abrir archivo .sql desde oracle.



.Sql con las sentencias devueltas para la creacion de las tablas.

```
CREATE TABLE avion (
    codigo INTEGER NOT NULL,
    marca VARCHAR2(30 CHAR),
    estado VARCHAR2(20 CHAR));
ALTER TABLE avion ADD CONSTRAINT avion_pk PRIMARY KEY ( codigo );
CREATE TABLE cliente (
    cedula INTEGER NOT NULL,
    nombre VARCHAR2(10 CHAR),
    apellido VARCHAR2(10 CHAR),
    fecha_nacimiento DATE,
    sexo VARCHAR2(10 CHAR),
    reserva_id_reserva INTEGER NOT NULL);
ALTER TABLE cliente ADD CONSTRAINT cliente_pk PRIMARY KEY ( cedula );
CREATE TABLE destino (
    id_destino INTEGER NOT NULL,
    nombre VARCHAR2(10 CHAR));
ALTER TABLE destino ADD CONSTRAINT destino_pk PRIMARY KEY ( id_destino );
CREATE TABLE reserva (
    id_reserva INTEGER NOT NULL,
    estado VARCHAR2(10 CHAR),
    vuelo_id_vuelo INTEGER NOT NULL);
ALTER TABLE reserva ADD CONSTRAINT reserva_pk PRIMARY KEY ( id_reserva );
CREATE TABLE vuelo (
    id_vuelo INTEGER NOT NULL,
    destino VARCHAR2(10 CHAR),
    disponibilidad VARCHAR2(10 CHAR),
    tipo_vuelo VARCHAR2(10 CHAR),
    avion_codigo INTEGER NOT NULL,
    destino_id_destino INTEGER NOT NULL);
ALTER TABLE vuelo ADD CONSTRAINT vuelo_pk PRIMARY KEY ( id_vuelo );
ALTER TABLE cliente
    ADD CONSTRAINT cliente_reserva_fk FOREIGN KEY ( reserva_id_reserva )
        REFERENCES reserva ( id_reserva );
ALTER TABLE reserva
    ADD CONSTRAINT reserva_vuelo_fk FOREIGN KEY ( vuelo_id_vuelo )
        REFERENCES vuelo ( id_vuelo );
```

Ejecucion .sql.

Oracle SQL Developer | C:\Users\fernando\Desktop\proyecto\Proyecto\org.eclipse.accele.module.projects\ACC\src\ClientReservaVueloAvionDestino.sql

Archivo Editar Ver Navegar Ejecutar Díjete Equipo Herramientas Ventana Ayuda

Conexiones

Datos2 (Fer2) Oracle conexiones Conexiones de Servicio Esquema de Base de Datos

Hojas de Trabajo SQL Historial

Hojas de Trabajo Generador de Consultas

```

CREATE TABLE avion (
    codigo INTEGER NOT NULL,
    marca VARCHAR2(30 CHAR),
    estado VARCHAR2(20 CHAR));
ALTER TABLE avion ADD CONSTRAINT avion_pk PRIMARY KEY ( codigo );

CREATE TABLE cliente (
    cedula INTEGER NOT NULL,
    nombre VARCHAR2(10 CHAR),
    apellido VARCHAR2(10 CHAR),
    fecha_nacimiento DATE,
    sexo VARCHAR2(10 CHAR),
    reserva_id_reserva INTEGER NOT NULL);
ALTER TABLE cliente ADD CONSTRAINT cliente_pk PRIMARY KEY ( cedula );

CREATE TABLE destino (
    id_destino INTEGER NOT NULL,
    nombre VARCHAR2(10 CHAR));
ALTER TABLE destino ADD CONSTRAINT destino_pk PRIMARY KEY ( id_destino );

CREATE TABLE reserva (
    id_reserva INTEGER NOT NULL,
    estado VARCHAR2(10 CHAR),
    vuelo_id_vuelo INTEGER NOT NULL);
ALTER TABLE reserva ADD CONSTRAINT reserva_pk PRIMARY KEY ( id_reserva );

```

Selcción de Script

Tarea terminada en 1,188 segundos

Table DESTINO creado.

Table DESTINO alterado.

Table RESERVA creado.

Table RESERVA alterado.

Table VUELO creado.

Table VUELO alterado.

Haga clic en un identificador con la tecla Control pulsada para ejecutar "Ir a Declaración".

Línea 12 Columna 29 | Insertar | Modificado | Una/Más U

1.4 Repositorio Github

<https://github.com/ferleo99/Proyecto-Final>