

ANALISIS Y DISEÑO DE SISTEMAS – UML PUDS.

**PUDS- GENERACIÓN DE CÓDIGO
CON DIAGRAMAS DE SECUENCIA.**

Introducción

- La **generación de código fuente con herramientas CASE** (Computer-Aided Software Engineering) es un enfoque que permite automatizar parte del desarrollo de software al convertir diagramas y modelos UML en código fuente. Estas herramientas son ampliamente utilizadas en proyectos grandes y complejos, ya que mejoran la productividad y reducen errores humanos.

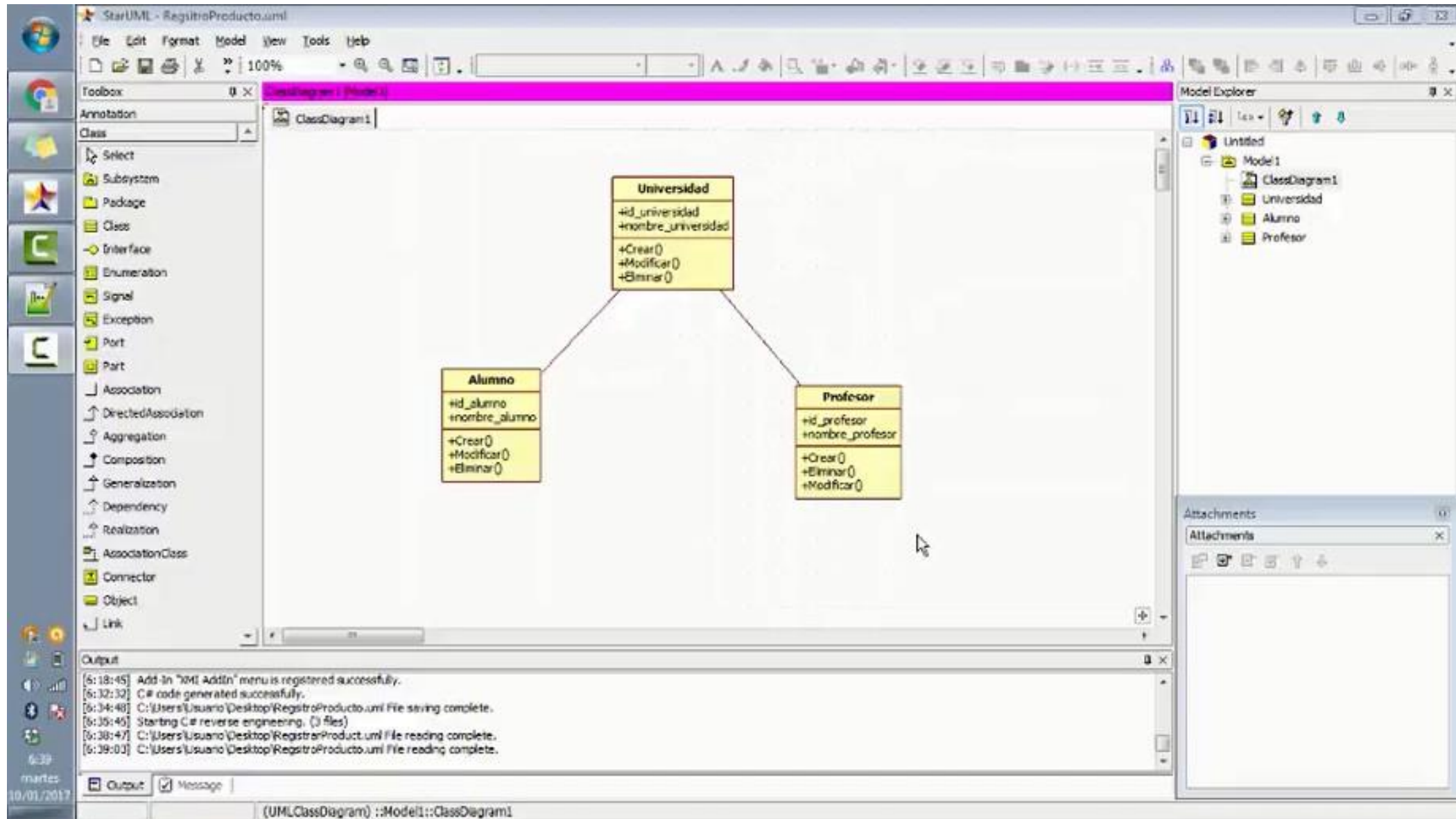


APLICACIÓN EN CASO DE ESTUDIO: DIAGRAMA DE CLASES



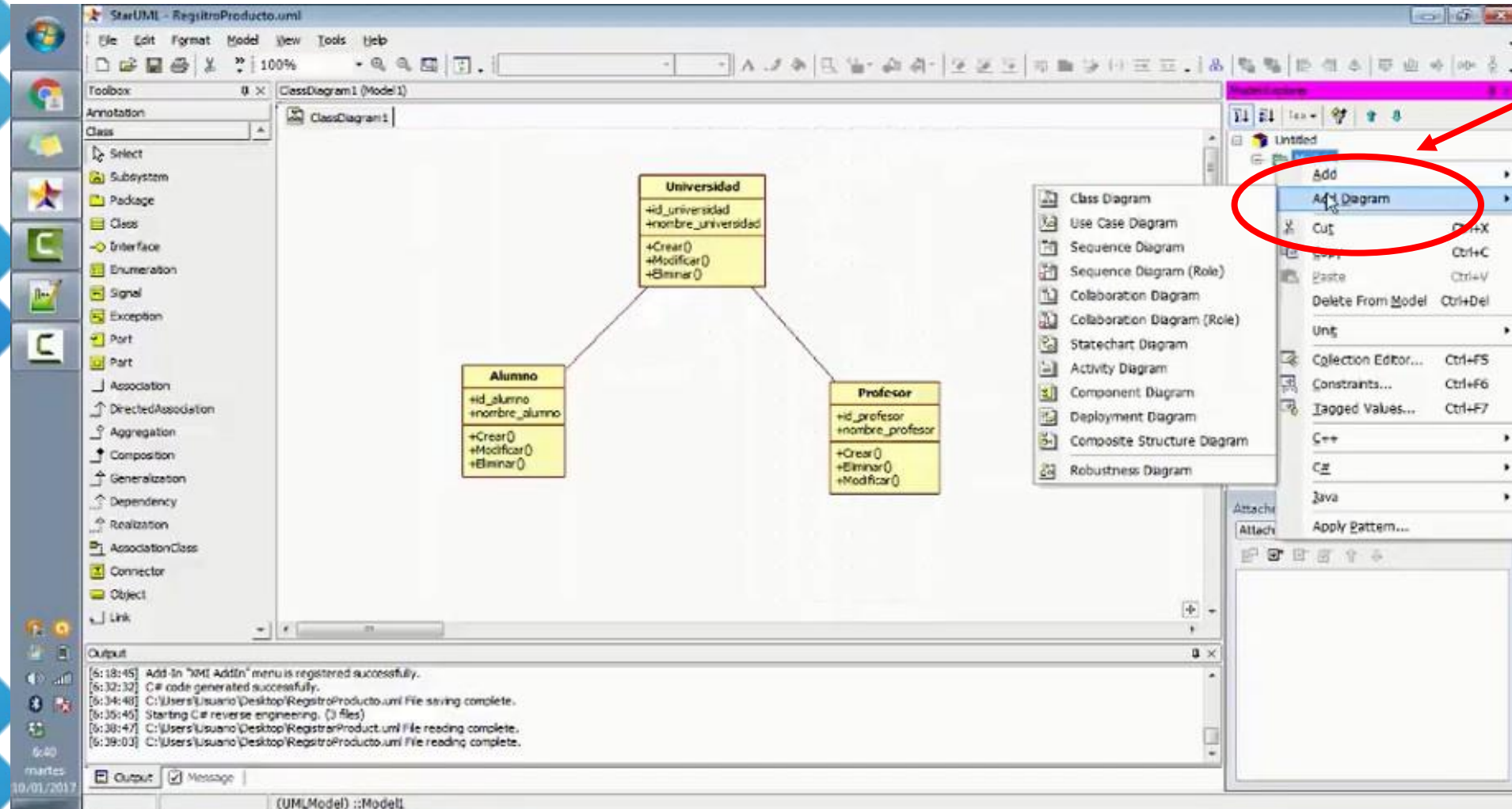
- Herramienta para el modelamiento de software basado en los estándares UML (Unified Modeling Language) y MDA (Model Driven Architecture)
- Capacidad de generar código a partir de los diagramas y viceversa, actualmente funcionando para los lenguajes C++, C# y Java.
- Soporta la mayoría de los tipos de diagramas especificados en UML 2.0.
- Generar documentación en formatos Word, Excel y PowerPoint sobre los diagramas..

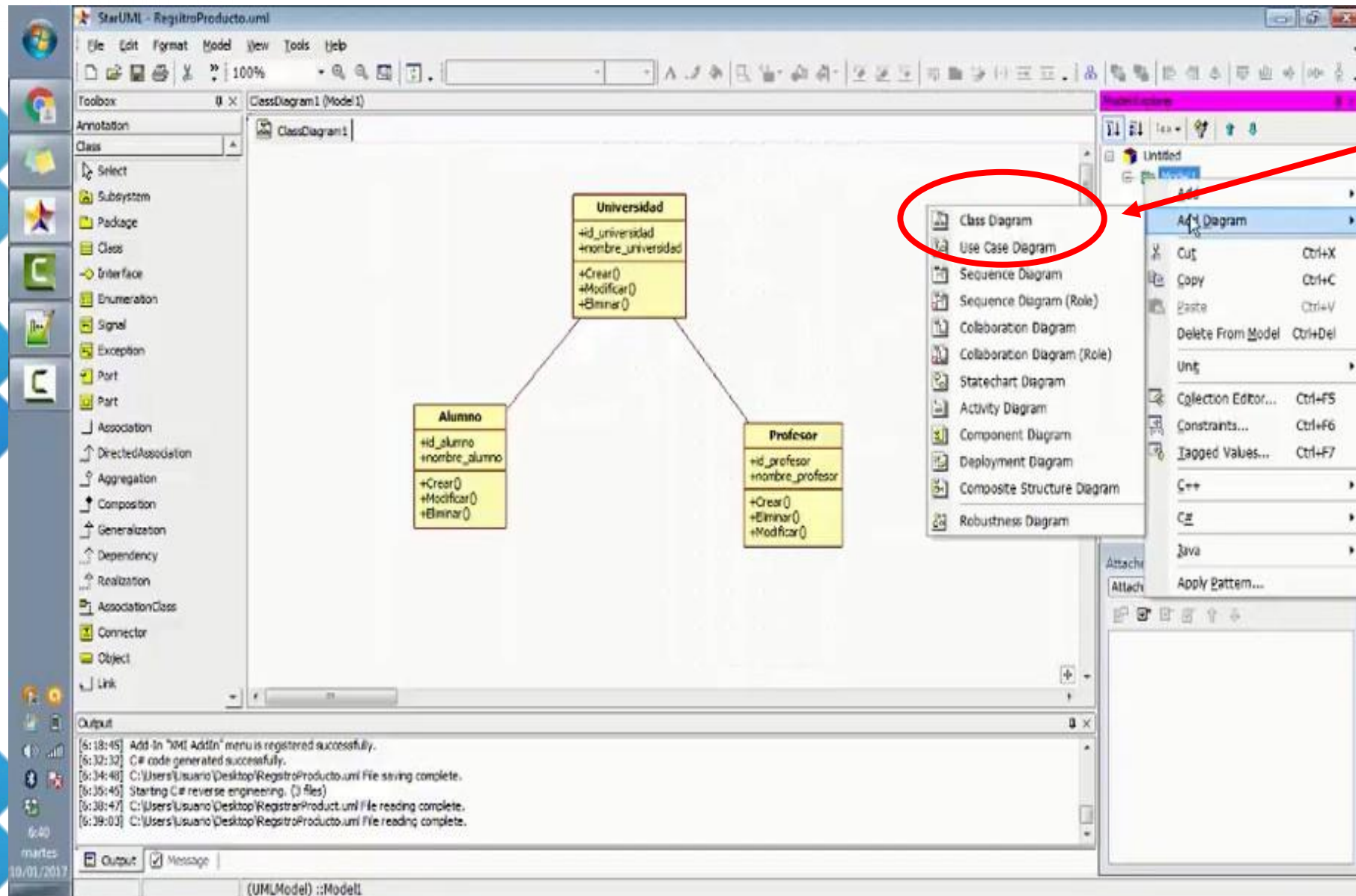
Tenemos el siguiente diagrama de clases:



Crear el diagrama de clases con la herramientas del software Staruml:

Añadir
diagrama

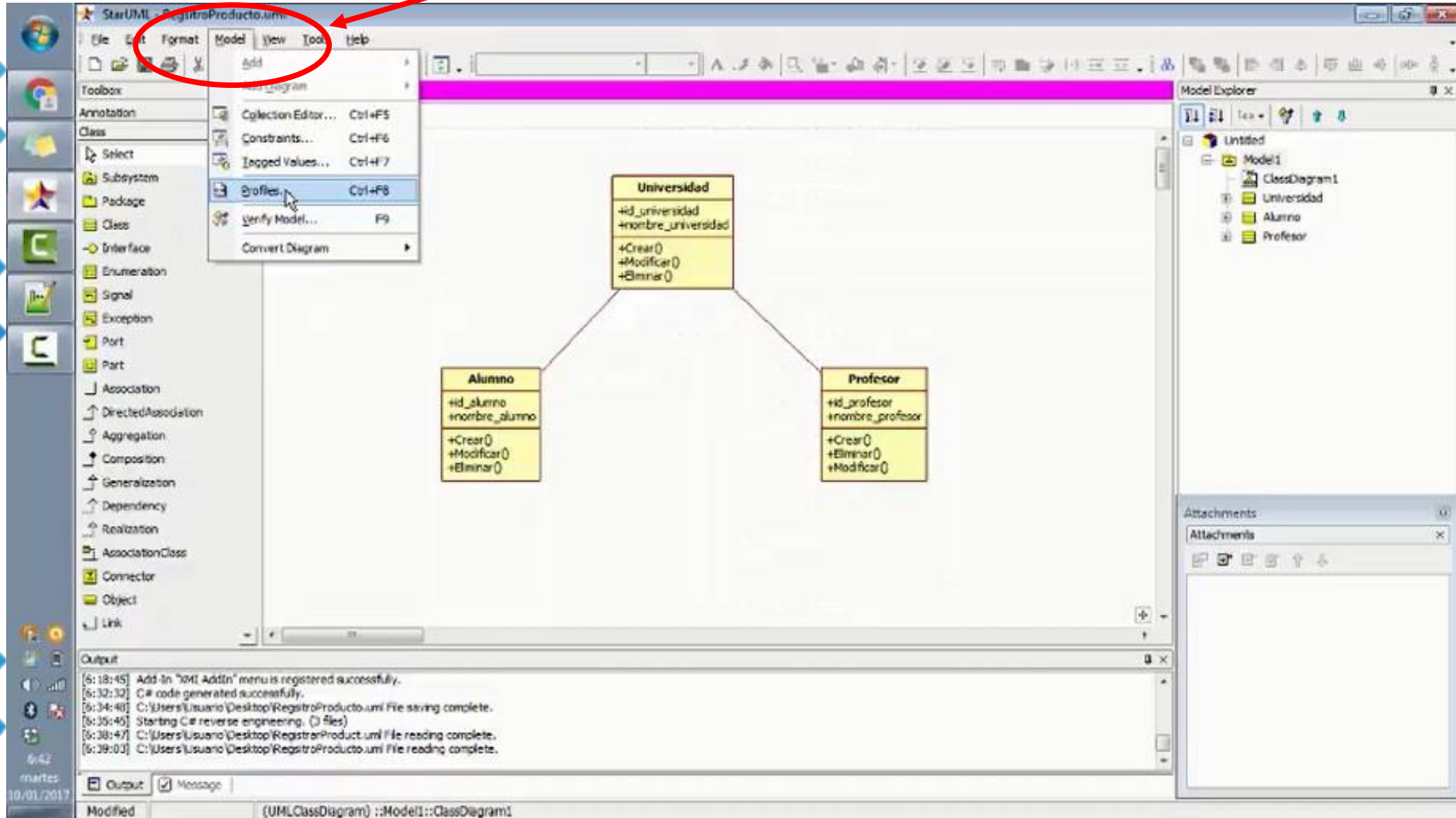




**Añadir
diagrama:
- Opcion
diagrama de
clase**

Ir al menú principal.

Elige la opción MODEL y luego Profiles



The screenshot shows the StarUML application window titled 'StarUML - RegistroProducto.uml'. The 'Model' menu is open, and the 'Profiles' option is highlighted with a red circle and a red arrow pointing to it. The main diagram area displays a UML Class Diagram with three classes: 'Universidad', 'Alumno', and 'Profesor'. 'Universidad' is the superclass, and 'Alumno' and 'Profesor' are subclasses. The 'Model Explorer' on the right shows the project structure: 'Untitled' > 'Model1' > 'ClassDiagram1' > 'Universidad', 'Alumno', and 'Profesor'. The 'Output' window at the bottom shows a log of events, including file saving and reading operations. The status bar at the bottom indicates the current diagram is '(UMLClassDiagram) ::Model1::ClassDiagram1'.

```
graph TD
    Universidad --> Alumno
    Universidad --> Profesor
```

Universidad
+id_universidad
+nombre_universidad
+Crear()
+Modificar()
+Eliminar()

Alumno
+id_alumno
+nombre_alumno
+Crear()
+Modificar()
+Eliminar()

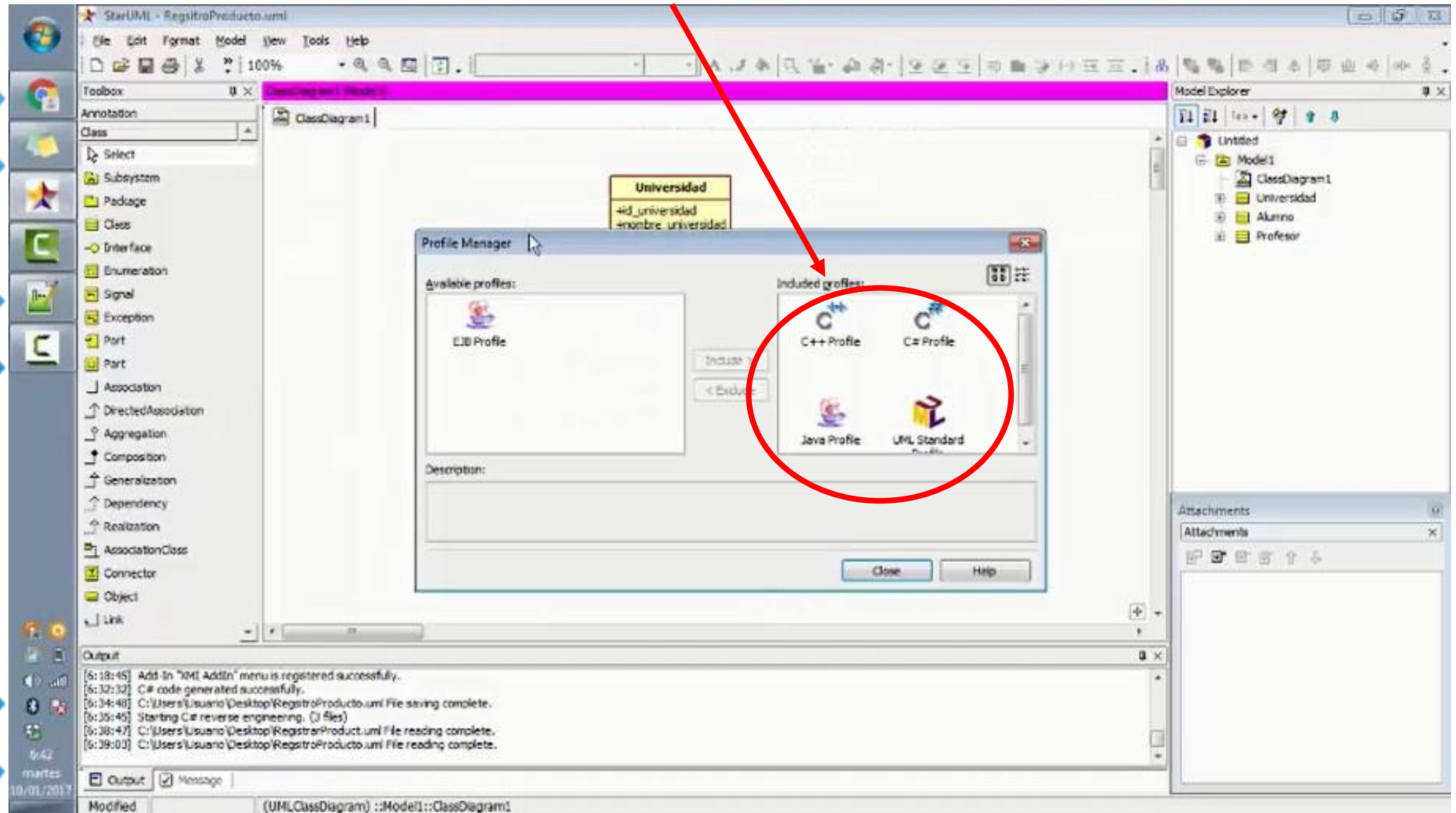
Profesor
+id_profesor
+nombre_profesor
+Crear()
+Eliminar()
+Modificar()

Model Explorer:
Untitled
├── Model1
│ ├── ClassDiagram1
│ │ ├── Universidad
│ │ ├── Alumno
│ │ └── Profesor

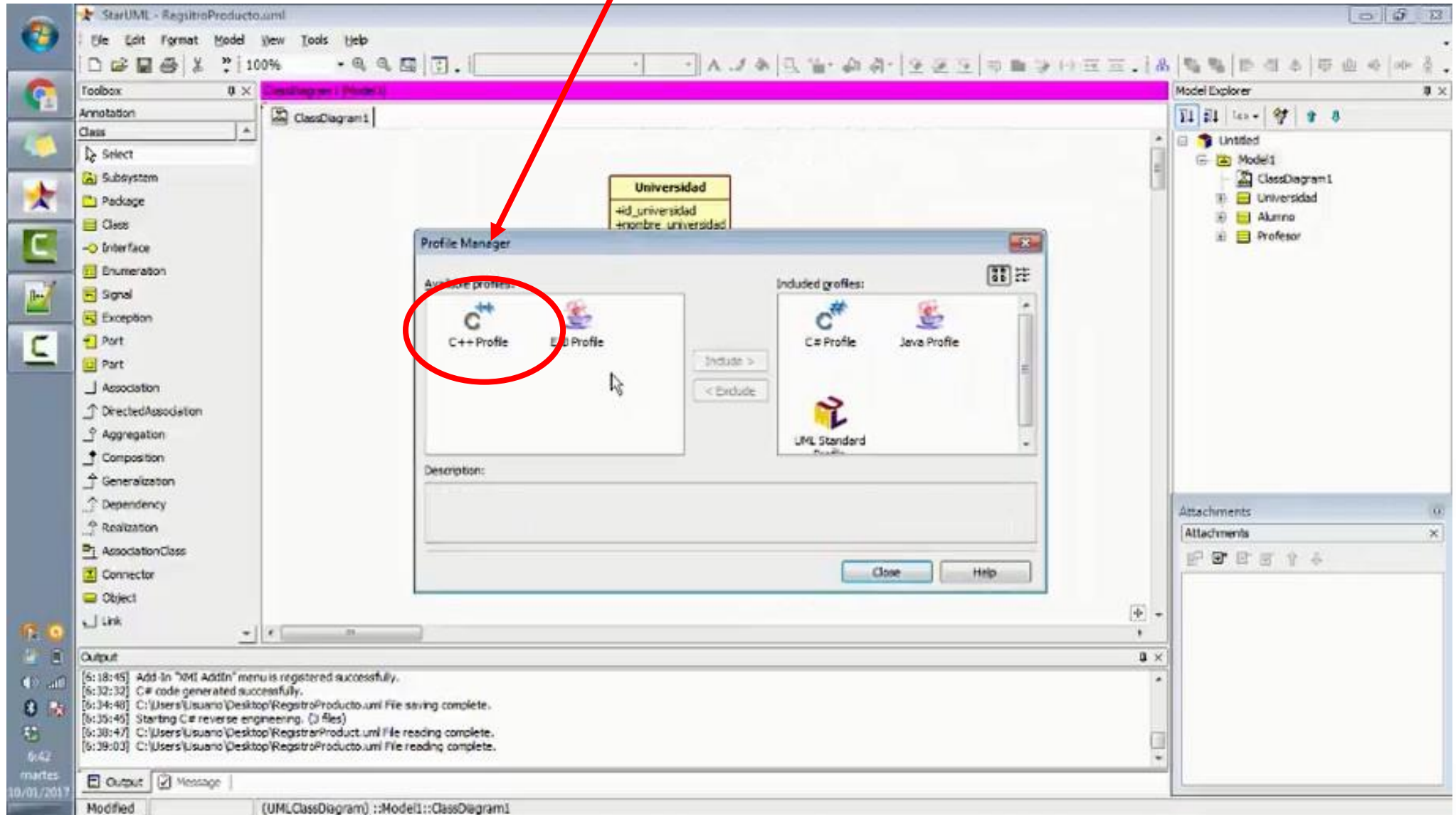
Output:
[6:18:45] Add-In "XMI AddIn" menu is registered successfully.
[6:32:32] C# code generated successfully.
[6:34:48] C:\Users\Usuario\Desktop\RegistroProducto.uml File saving complete.
[6:35:45] Starting C# reverse engineering. (3 files)
[6:38:47] C:\Users\Usuario\Desktop\RegistroProducto.uml File reading complete.
[6:39:03] C:\Users\Usuario\Desktop\RegistroProducto.uml File reading complete.

Status Bar: (UMLClassDiagram) ::Model1::ClassDiagram1

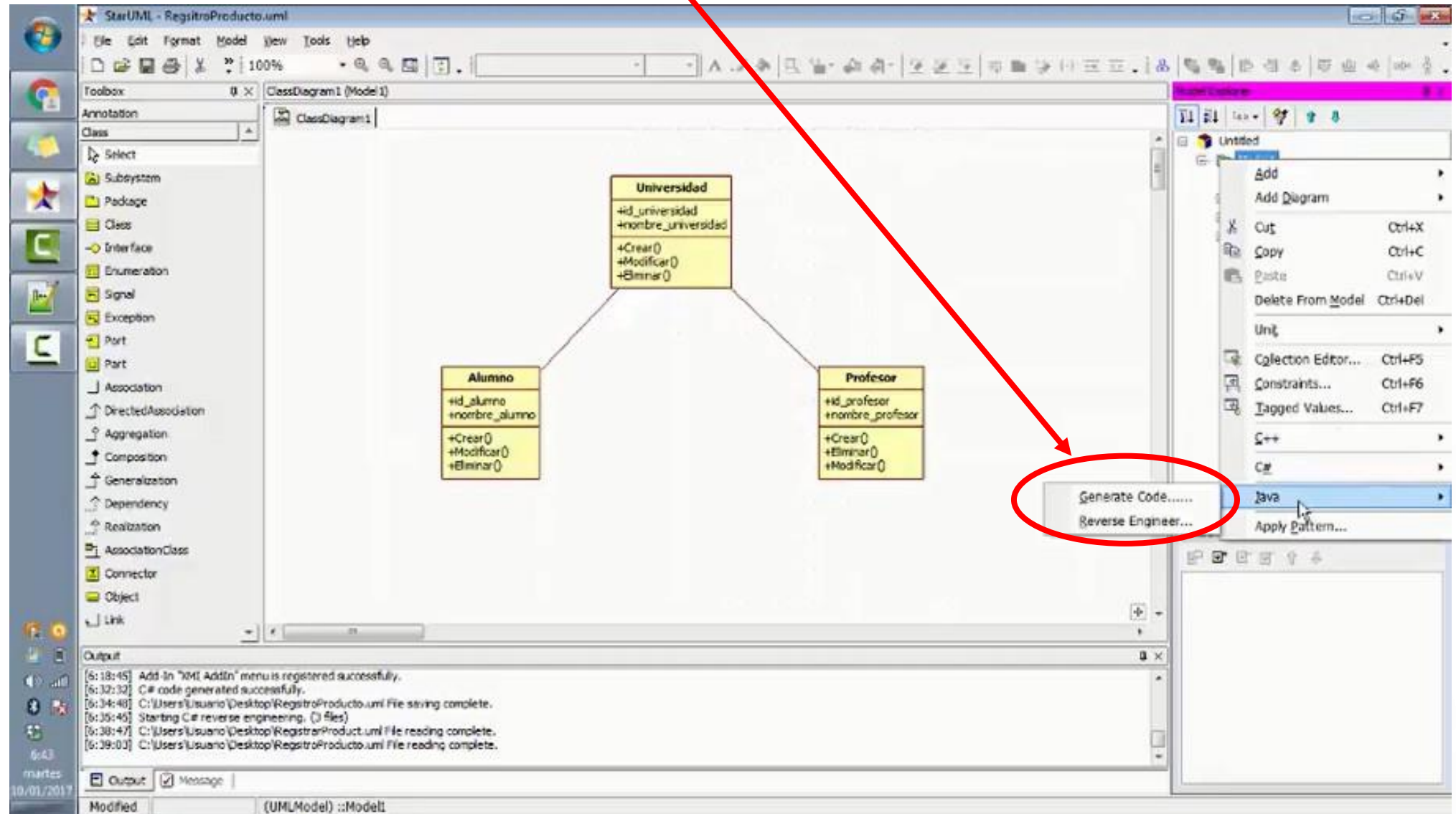
Elegir el lenguaje a generar

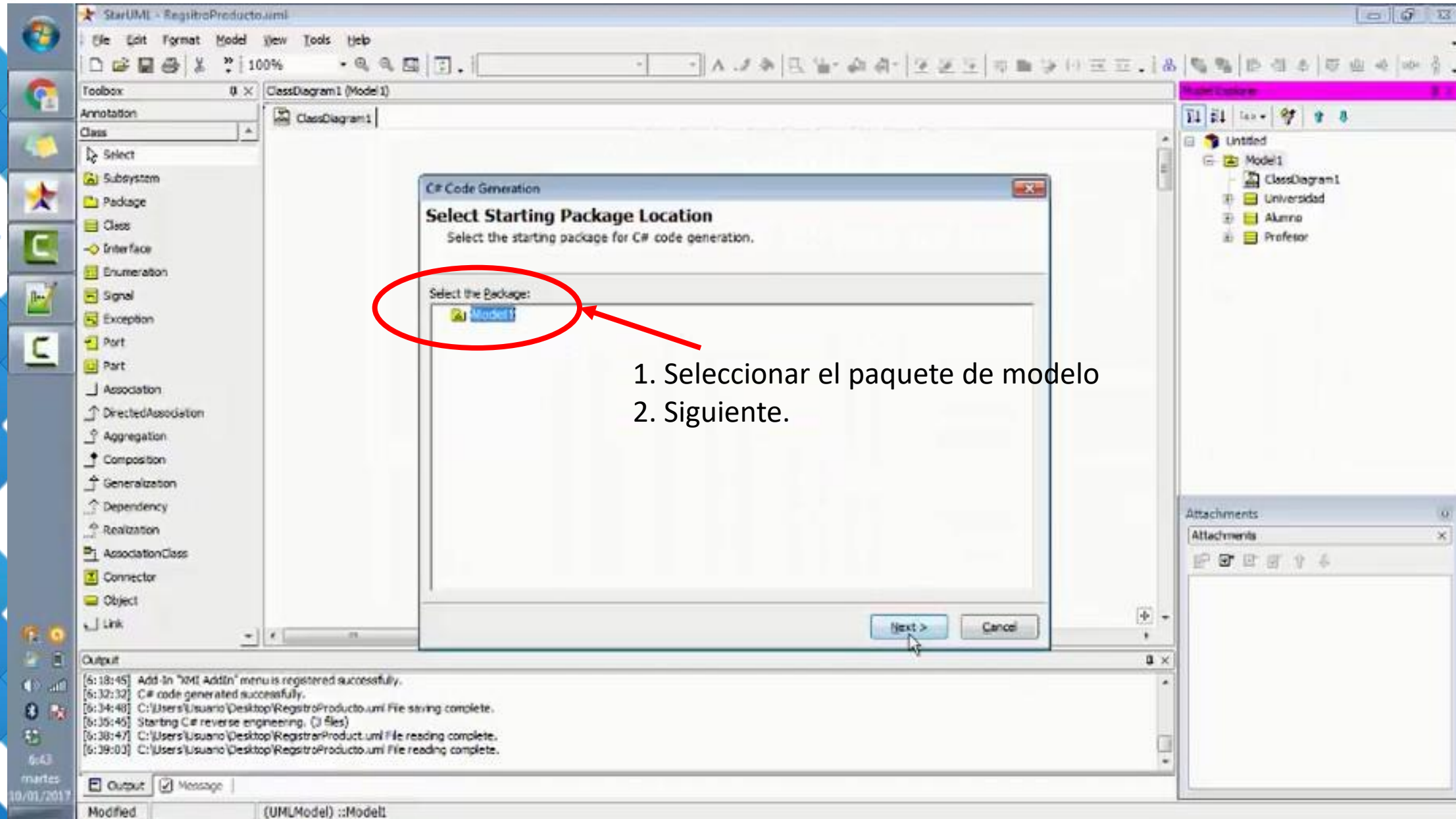


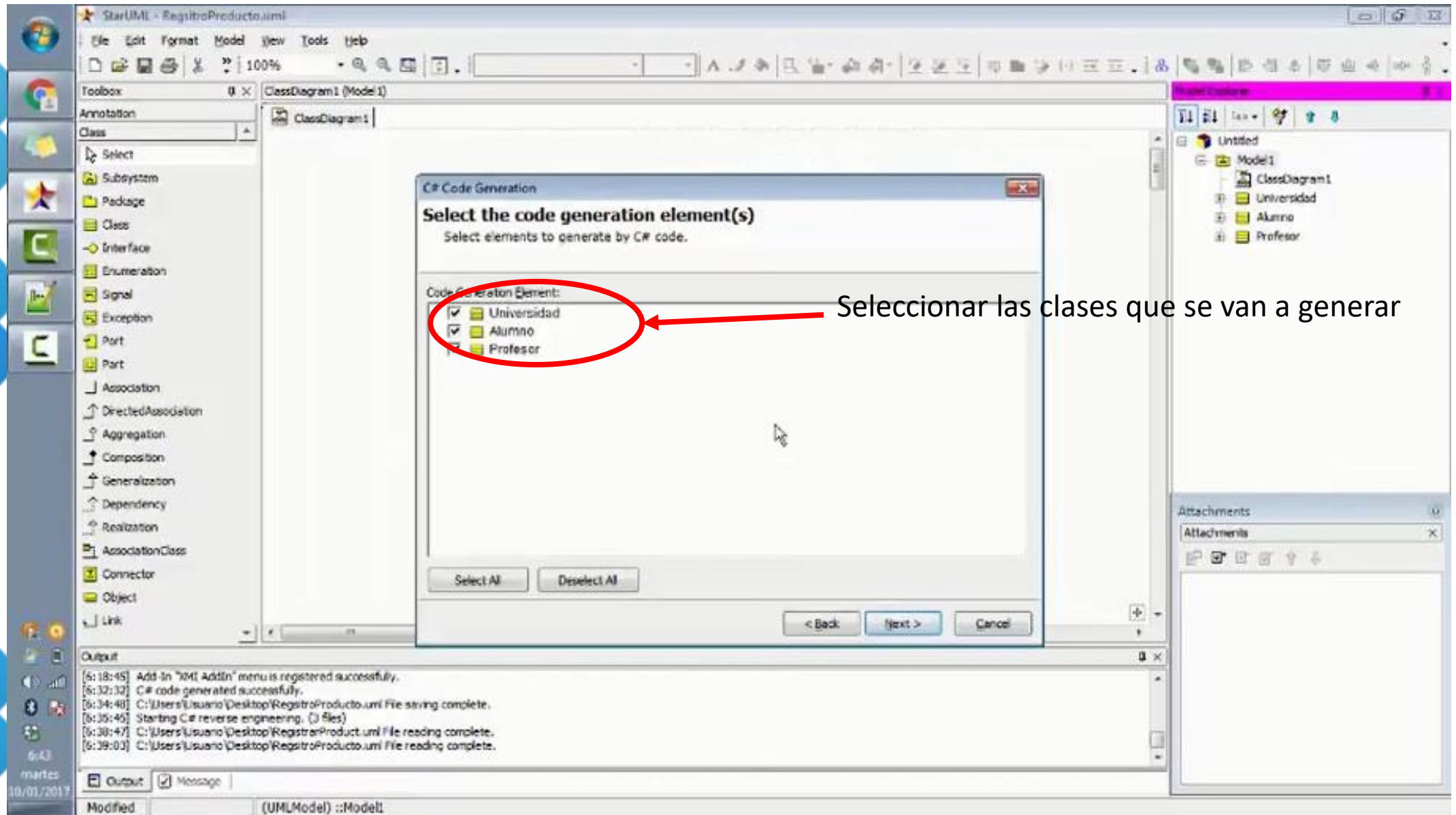
Añadir a la biblioteca de lenguajes

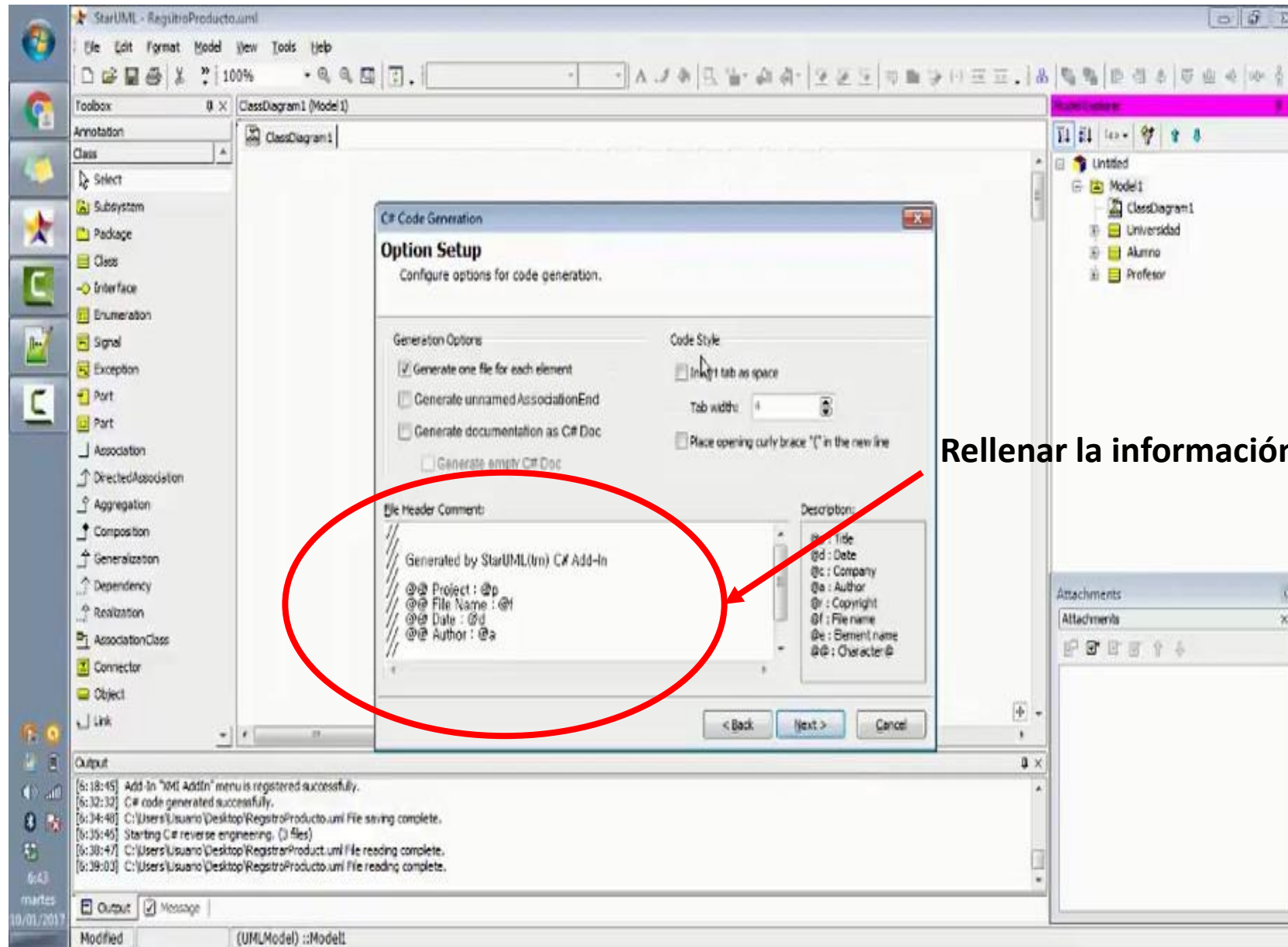


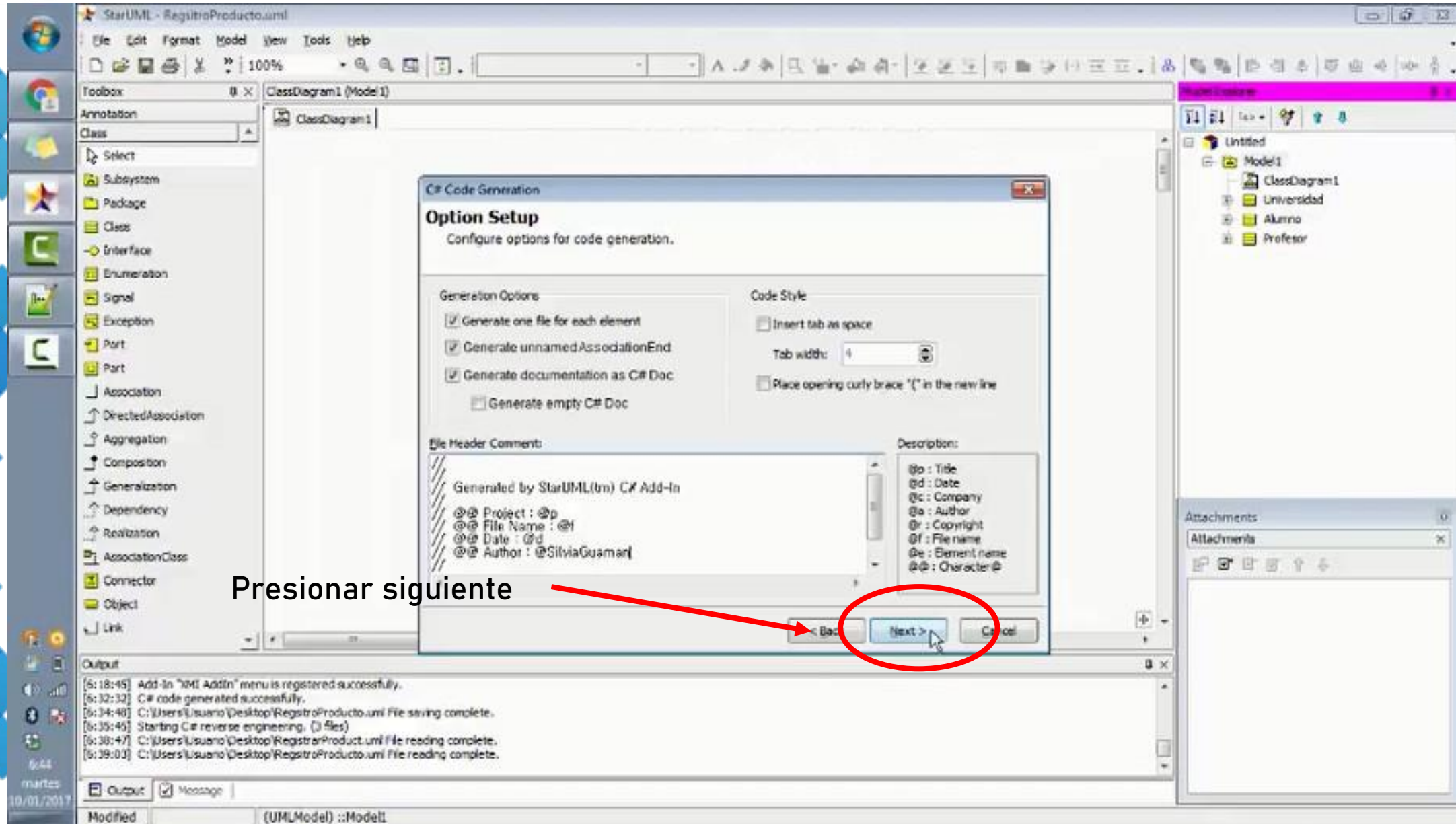
Elegir el modelo :
Opción de lenguaje
Generar código



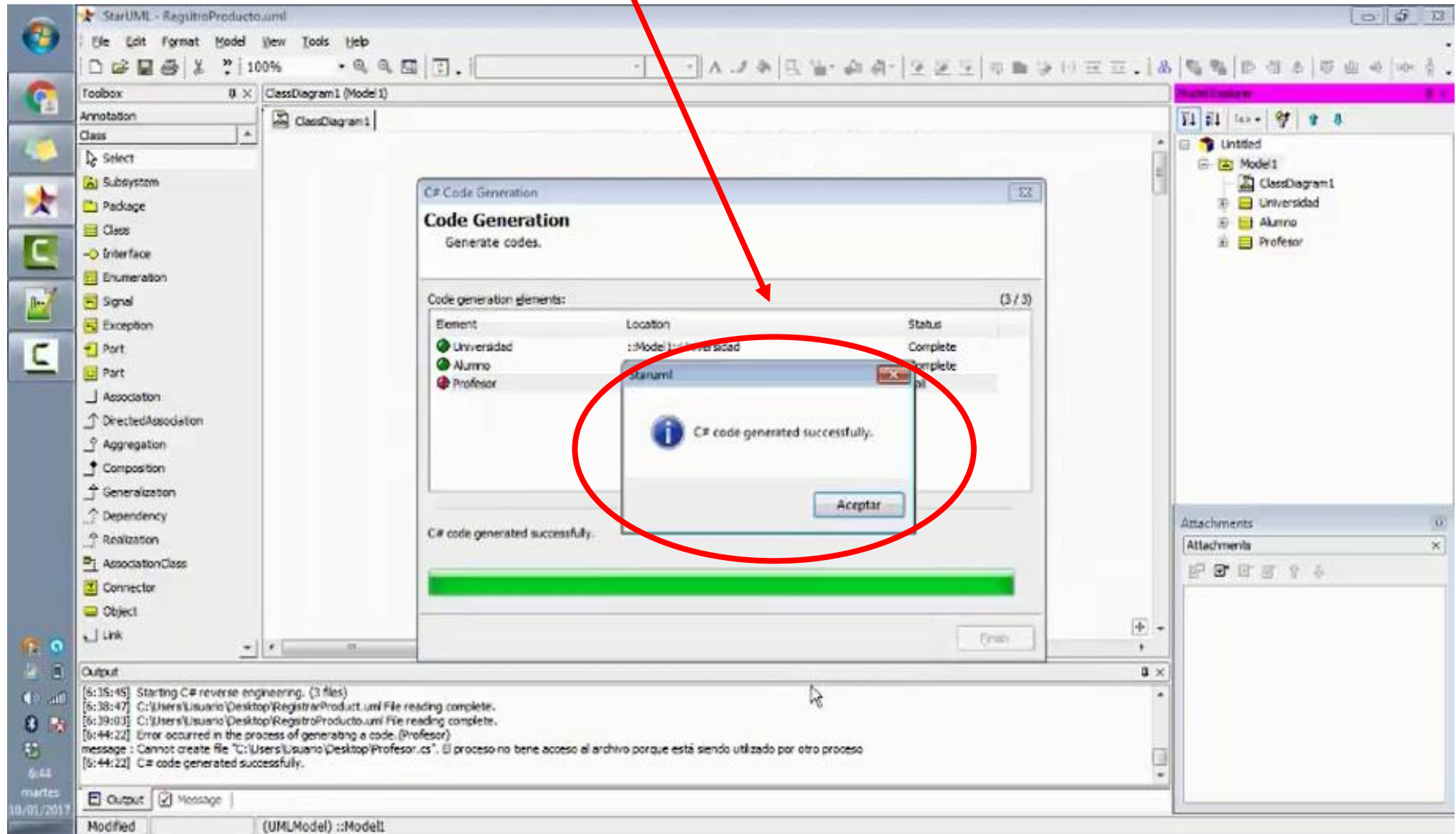




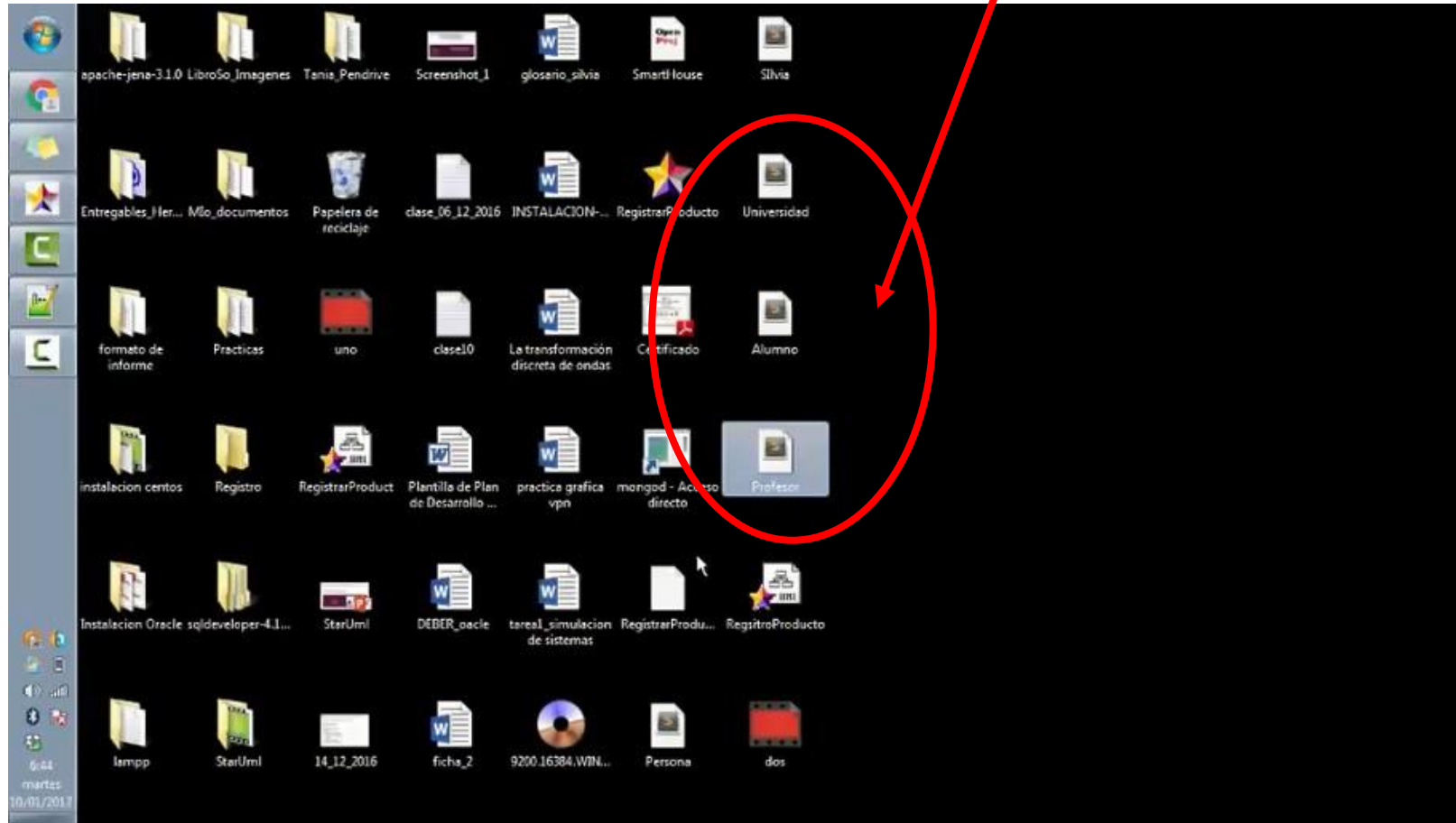




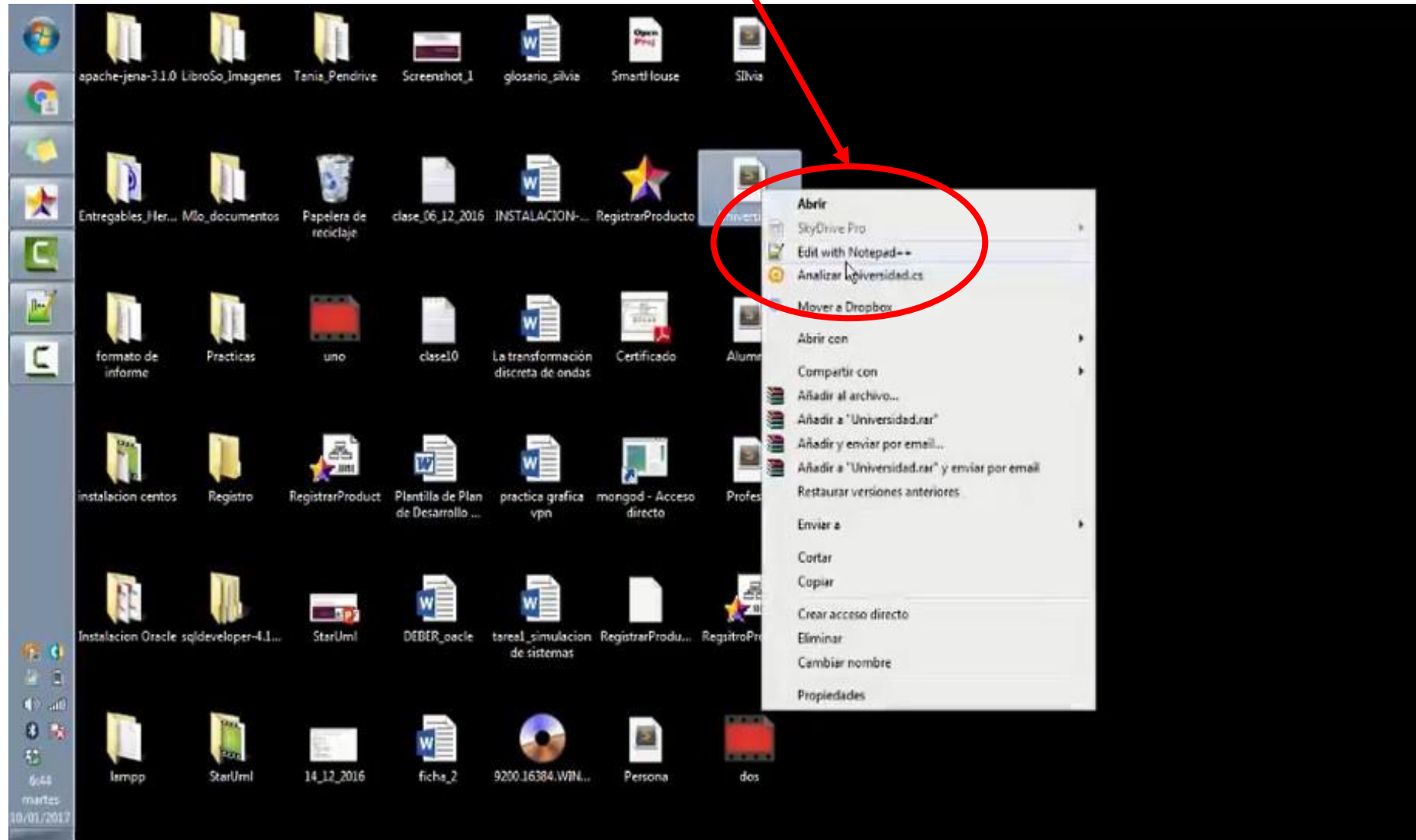
Verificar que se ha generado el código y aceptar



Ir a la ubicación y se mostrará los archivos generados



Abrir el algún editor de texto



Constatar el contenido del código

