2-6-2020

Grupo 4

Nero

Patrones de diseño

Nero

Patrones

## Introducción

Respecto a los patrones de diseño, primero debemos tener en cuenta los problemas diseño que podría tener nuestra aplicación y a partir de ahí, obtener una solución utilizando el patrón de diseño adecuado.

Problema

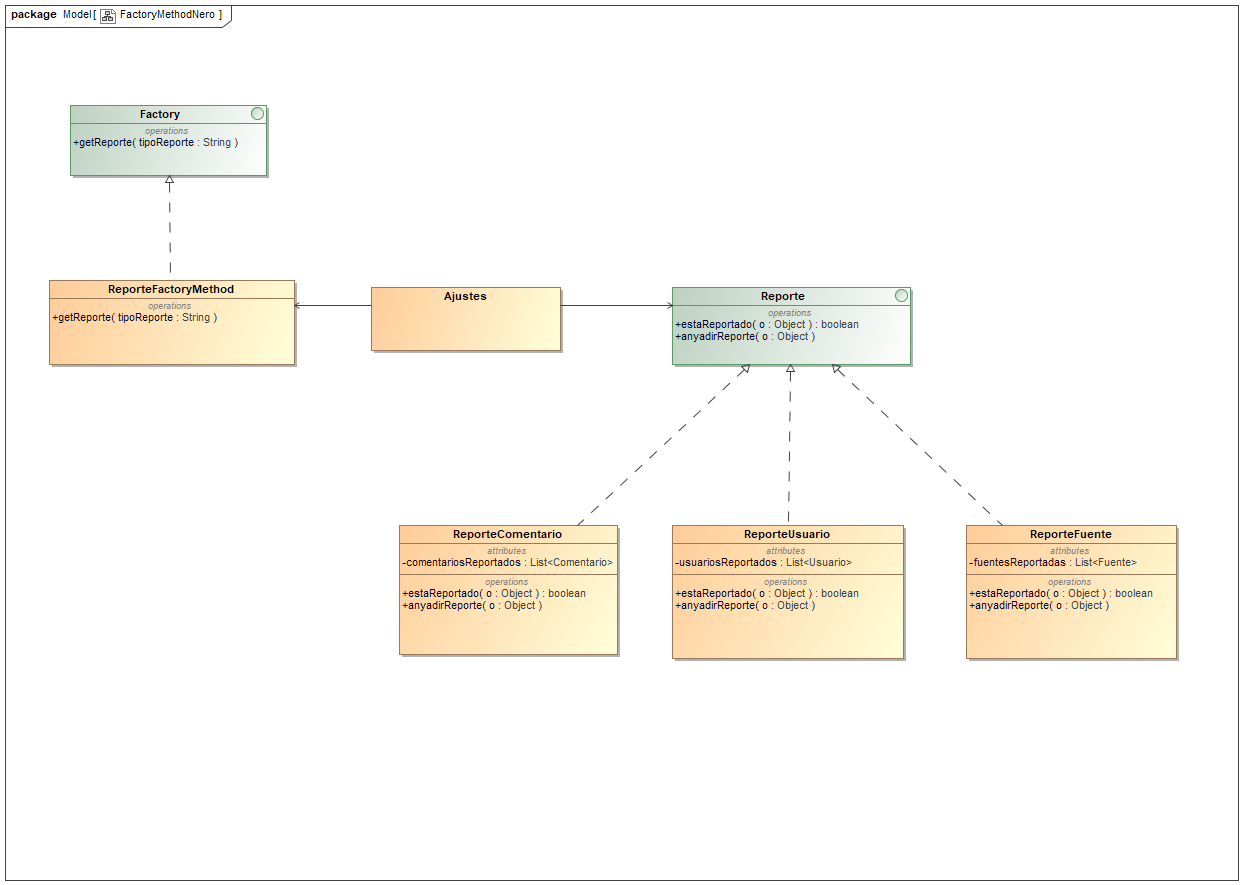
* Cuando un usuario entra en el menú de los ajustes, deberá escoger entre un reporte de Usuario, un reporte de Fuente o un reporte de Comentario. Dependiendo de la opción escogida del usuario, nos mostrará una información u otra.

Esto, a nivel de implementación supone un problema ya que **no sabemos que tipo de reporte seleccionará nuestro usuario.**

* Por ello, Necesitamos crear una instancia de una clase que aún no conocemos

## Solución

* Utilizar el patrón de diseño **Factory Method**



**IMAGEN 1**

* Se creará una interfaz llamada **Reporte** y cada tipo de reporte (ReporteComentario, ReporteUsuario y ReporteFuente) tienen que implementar esa interfaz.
* Ajustes NO realizará instancias de tipo ReporteComentario, ReporteUsuario o ReporteFuente: en Ajustes se definirá una variable de tipo Reporte.
* Por último, la fábrica (**ReporteFactoryMethod**) es la que se encarga de la llamada de los constructores de los distintos tipos de reporte (ReporteComentario, ReporteUsuario, ReporteFuente).

**IMAGEN 2**

Adaptación del patrón en la aplicación

