

# Desarrollo de una herramienta de ingeniería inversa para Scala utilizando mecanismos de reflexión

Mariela Remedios Lezama Sánchez (mlezamas@ito-depi.edu.mx) | Ulises Juárez Martínez (ujuarez@ito-depi.edu.mx)

## Objetivo general

Desarrollar una herramienta de ingeniería inversa que permita la representación objeto funcional de aplicaciones en Scala utilizando mecanismos de reflexión.



## Introducción

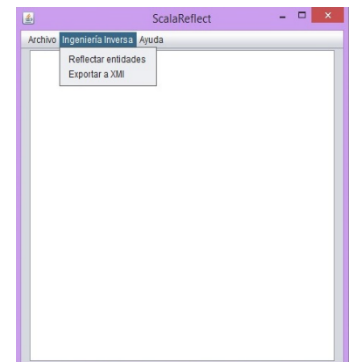
El lenguaje Scala tiene sus inicios en el 2003, sin embargo, dada la popularidad que adquiere, este lenguaje se considera novedoso, por esta razón no se reporta alguna herramienta que ayude a conocer cómo impacta la parte funcional en la arquitectura de un sistema. Lo anterior representa una necesidad y una oportunidad para desarrollar la herramienta de ingeniería inversa que muestre cómo se encuentra la arquitectura del sistema en Scala, representando de manera adecuada la parte funcional de la misma.



## Resultados

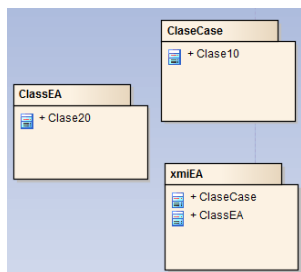
Se desarrolló *ScalaReflect*: una herramienta de ingeniería inversa que obtiene información relevante de entidades Scala [1] a través del uso de la *API Reflection*. Estructura esta información en un documento *XMI* tomando en cuenta el del Modelo 4+1 vistas [2], y la notación funcional [3]. Finalmente se importa desde la herramienta de modelado *UML Enterprise*

*Architect* el documento *XMI* para representar el diagrama de clases y de paquetes. También integró un aporte que demuestra que *AspectJ* cuenta con el soporte para obtener información dinámica de un sistema.



## Conclusiones

Trabajar alineándose al Modelo 4+1 facilitó el desarrollo, ya que este modelo sirvió como una guía. El mecanismo de reflexión de Scala es el pilar de esta herramienta, debido al gran soporte que ofreció para construir los diagramas. Este mecanismo fungió un papel muy relevante para la representación funcional, la cual es importante de recuperar ya que al estar implementada en los sistemas Scala, impacta en cuestiones de diseño y de arquitectura. Con la implementación de *XMI*, se le dará la facilidad al encargado de mantenimiento de visualizar los diagramas en una herramienta diferente. Esta herramienta apoyará en el mantenimiento de sistemas desarrollados en Scala, ya que será más fácil identificar qué aspectos de la implementación son los que necesitan ajustes o en cuáles de ellos se introducirá un nuevo requerimiento.



[1] M. Odersky, «Introduction to Scala,» 21 Junio 2004. [En línea]. Available: docs.scala-lang.org/tutorials/tour/tour-of-scala.html. [Último acceso: 12 Febrero 2017].

[2] P. Kruchten, «The 4 +1 View Model of Architecture,» IEEE Software, vol. 12, n° 6, pp. 42-50, 1995.

[3] N. Rodríguez Munguía, U. Juárez Martínez, G. Peláez Camarena y A. Sánchez García, «Propuesta de notación para el modelado de elementos de programación funcional en UML,» CIITI, p. Por publicarse, 2016.