

## Objetivo

Desarrollar un generador de Aplicaciones Enriquecidas de Internet a partir de modelos realizados con UML e IFML basados en un patrón arquitectónico MVC y entregando como salida una de dos posibles combinaciones: a) JSF con PrimeFaces y b) PHP con jQuery.

## Introducción

La elaboración de Aplicaciones Enriquecidas de Internet (RIAs) es una tarea compleja en la que se cruzan muchos requisitos, perspectivas y disciplinas como: diseño gráfico y estética, identidad visual empresarial, diseño de interacción, usabilidad, soporte de múltiples pantallas, entre otras, por lo cual su diseño e implementación consumen una gran cantidad de tiempo[1], aunado al hecho de que para la construcción de un proyecto de software sostenible y con estructura, es necesario realizar un análisis adecuado antes de escribir la primer línea de código [2]. De lo anterior se encuentra que el diseño del *front-end* es un proceso costoso, en donde la colisión de muchos factores complejos impone una continua revisión y refinamiento de la implementación, además de que la situación se agrava por la escasez de automatización en los métodos de producción de software convencionales, lo que provoca una baja reutilización de los artefactos de diseño a través de las interfaces de los diferentes proyectos y una elevada sobrecarga para asegurar la posibilidad de que las aplicaciones sean multiplataforma [3].

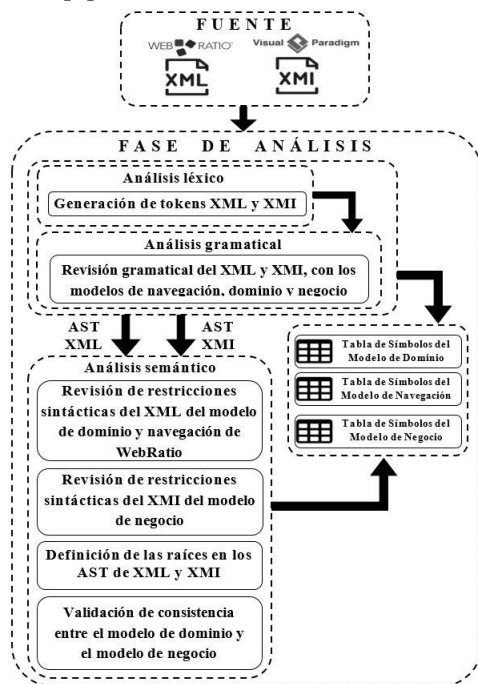


Figura 1. Arquitectura del generador de Aplicaciones Enriquecidas de Internet, fase de análisis.

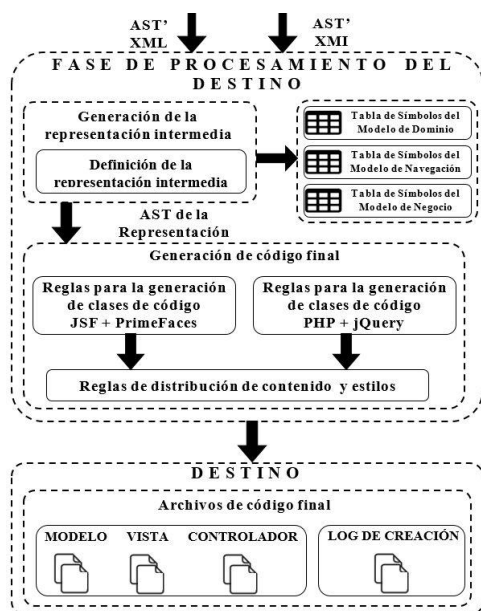


Figura 2. Arquitectura del generador de Aplicaciones Enriquecidas de Internet, fase del procesamiento del destino.

## Conclusiones

El generador realizado, debido al diseño de su arquitectura, tiene la capacidad de permitir se le agreguen módulos para generar otras combinaciones de lenguajes destino, así como también la generación de características particulares como la distribución de elementos dentro de la distribución general y admite con naturalidad la especificación de nuevas etiquetas IFML, modificando la gramática definida para archivos XML y generando los archivos del *parser* y *lexer* con ANTLR4, la definición de nuevos elementos de representación intermedia e, inclusive, la definición de nuevas estructuras de árboles abstractos, y no limita al desarrollador el uso de un repositorio de información en particular, ya que para que el usuario pueda ver la aplicación resultante el repositorio de información simula por medio de un archivo JSON para cada entidad del modelo de dominio dado.

## Referencias

- [1] S. Roubi, M. Erramdani, y S. Mbarki, «A model driven approach to generate graphical user interfaces for Rich Internet Applications using Interaction Flow Modeling Language», en 2015 15th International Conference on Intelligent Systems Design and Applications (ISDA), 2015, pp. 272-276.
- [2] K. Frajták, M. Bureš, y I. Jelínek, «Transformation of IFML Schemas to Automated Tests», en Proceedings of the 2015 Conference on Research in Adaptive and Convergent Systems, New York, NY, USA, 2015, pp. 509-511.
- [3] R. Acerbis, A. Bongio, M. Brambilla, y S. Butti, «Model-driven development based on omg's IFML with webratio web and mobile platform», en International Conference on Web Engineering, 2015, pp. 605-608.
- [4] A. Leff y J. T. Rayfield, «Web-Application Development Using the Model/View/Controller Design Pattern», en Proceedings of the 5th IEEE International Conference on Enterprise Distributed Object Computing, Washington, DC, USA, 2001, pp. 118-

## Arquitectura del generador de aplicaciones

Las Figuras 1 y 2 presentan la arquitectura del generador de Aplicaciones Enriquecidas de Internet, la cual se encuentra dividida en dos partes, la primera de ellas concierne a la lectura, procesamiento y validación de los archivos de entrada, la generación de los árboles de sintaxis abstracta (AST) y de las tablas de símbolos, la segunda parte de la arquitectura atañe a la generación de la representación intermedia en función de los resultados obtenidos por el proceso de análisis, y a la generación del código final esperado.

## Arquitectura de las aplicaciones generadas

En la Figura 3 se muestra la arquitectura de las aplicaciones generadas, para JSF con PrimeFaces como para PHP con jQuery, la cual está basada en el patrón arquitectónico MVC, esto sustentado en que permite al desarrollador identificar fácilmente los componentes de cada capa y su comunicación con los demás componentes existentes en la aplicación y a su alto nivel de abstracción [4].

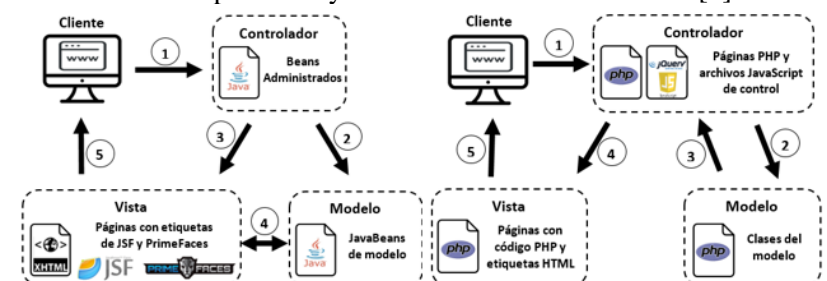


Figura 3. Resultados de una aplicación generada en PHP y jQuery a partir de sus modelos

## Resultados

Se obtuvo un generador que toma como elementos de entrada modelos IFML (XML correspondientes a la herramienta de modelado IFML WebRatio Web Platform) y UML (XMI proveniente de la herramienta Visual Paradigm, y sólo diagrama de clases) que representan el *front-end* y *back-end* de las aplicaciones a generar, la elección de una de las combinaciones de lenguajes disponibles como salida y especificaciones relacionadas con la vista como lo son la selección de un tema y la distribución de los elementos que conforman la aplicación, y se entrega un conjunto de archivos agrupados de acuerdo al patrón arquitectónico MVC, los cuales conforman el esqueleto de una RIA que cumple por completo con la navegación indicada en el modelo IFML y con el modelo indicado en UML, pero que no cuenta con código SQL que les proporcione acceso a un repositorio de información