

Un framework para documentar frameworks.

Mauricio Sansano, Marcos Godoy, Luis Matricardi, Gustavo Rossi*
LIFIA - Universidad Nacional de La Plata
La Plata, Argentina.

[sansa, mgodoy, luis, gustavo]@sol.info.unlp.edu.ar

50 y 115 - 1er. Piso
(CP 1900) - La Plata, Argentina
Phone / Fax: (+54) 221 - 4228252

*also CONICET and UNLM

1. Introducción.

Después de varios años en el desarrollo de software, queda en claro que muchas aplicaciones que pertenecen al mismo dominio, son muy similares en cuanto a su diseño y arquitectura. Si bien difieren en algunos aspectos, su estructura y diseño son muy similares.

En este contexto, los desarrolladores de software han buscado la manera de reutilizar no solo las implementaciones sino también, y aún más importante, los diseños de aquellas aplicaciones correspondientes a un mismo dominio de aplicación. Los frameworks parecen ser hoy en día la manera más exitosa y flexible de conseguir dicho objetivo. Un framework permite el reuso de la misma manera que las técnicas orientadas a objetos lo hacen, pero con los frameworks es posible reusar largos y complejos *diseños*.

Un tema importante en la tecnología de frameworks, es la necesidad de contar con una documentación consistente y poderosa, que ayude a los usuarios de un framework a comprender rápidamente la arquitectura del mismo y cómo utilizarlo para poder cumplir con sus tareas. Nuestro trabajo se focaliza en este aspecto, tratando de aportar una herramienta que permita documentar un framework adecuadamente. El presente artículo describe los lineamientos generales del proyecto, más detalles pueden encontrarse en [Dierickx99].

2. Documentación basada en el usuario.

Diversos tipos de usuario utilizan un framework, y cada usuario, de acuerdo a su perfil, tiene diferentes objetivos. Estos diferentes tipos de usuarios necesitan diferentes tipos de documentación, de acuerdo con la tarea que realizan con el framework. Por ejemplo, un usuario normal o instanciador del framework, necesitará documentación del tipo de un cookbook. Por otro lado, un ingeniero de software cuya tarea es modificar el diseño del framework, necesitará documentación a cerca del diseño, tal como un Exemplar [GM95] o varios artefactos UML que describan diversas características de la arquitectura y el funcionamiento del framework. Más aún, diferentes tipos de usuarios deben ser capaces de leer la misma documentación con diferentes propiedades. Por ejemplo, el instanciador puede necesitar leer solo el cookbook, mientras que el ingeniero de software puede necesitar leer el mismo cookbook pero con links que vinculan las páginas del mismo con otros documentos. Por ejemplo, navegar de una página del cookbook que menciona como utilizar una clase en particular, a la CRC Card que describe a dicha clase.

Para poder dar soporte a este tipo de particularidades, nuestro intento es proveer diversas vistas de la misma documentación para los diversos perfiles de usuario. Pero además necesitamos documentación on-line, por lo que también es necesario tener una aplicación de documentación. En este sentido, la experiencia ha demostrado que las aplicaciones hipermediales son la forma más eficaz de organizar documentación on-line (la mayoría de los ambientes de desarrollo, por citar solo un ejemplo, proveen una aplicación hipermedia con la documentación de ayuda).

La figura 1 muestra un esquema ilustrando lo explicado anteriormente.

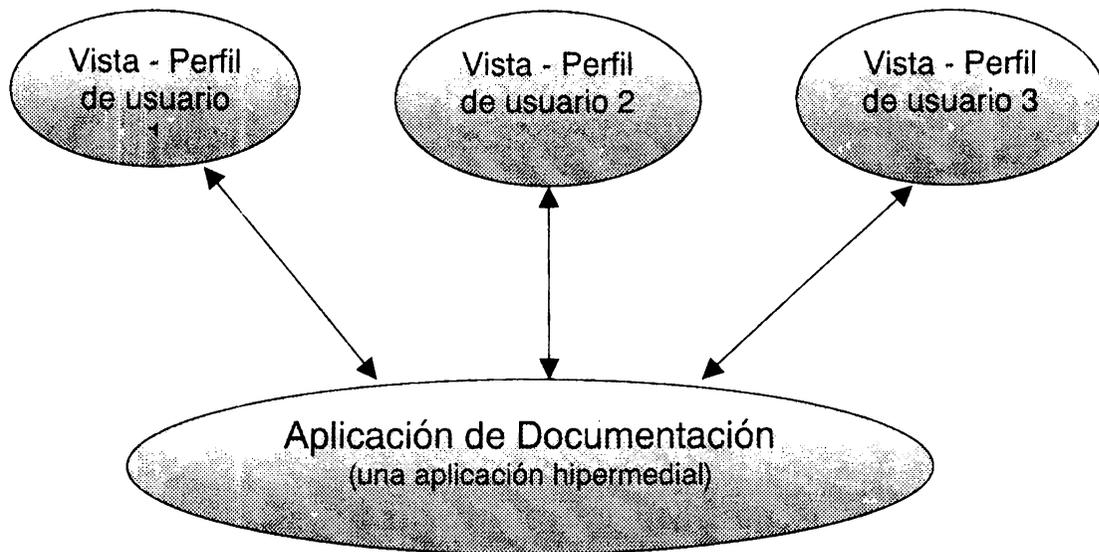


Figura 1 – Vistas de la aplicación de documentación.

3. Nuestra propuesta: el DocFramework.

Como dijimos en la sección anterior, vamos a organizar la documentación de un framework como una aplicación hipertextual. Para instanciar dicha aplicación, necesitamos primero crear toda la documentación del framework, es decir, cada uno de los artefactos tales como Exemplars, Diagramas de Clase UML, CRC Cards, Cookbooks, etc. Una vez que tenemos estos objetos creados, podemos definir nodos que representen estos documentos dentro de la aplicación hipertextual, y vincularlos mediante links. Por lo tanto, crear una aplicación hipertextual para documentar un framework no se reduce solo a crear nodos y links, sino también a instanciar primeramente los objetos subyacentes que serán la fuente de información que nutrirá al hipertexto.

De acuerdo a lo expuesto anteriormente, decidimos construir un nuevo framework, que llamamos DocFramework. Nuestro objetivo es diseñar un framework que abstraiga las características del dominio de documentación de frameworks como dominio de aplicación.

EL resultado de la instanciación de este framework será entonces la aplicación hipertextual que nos permita navegar entre los documentos en el contexto definido por cada vista. Los nodos de esta aplicación tomarán información de las instancias de las técnicas de documentación. Por ejemplo, un usuario de esta aplicación será capaz de navegar un nodo que represente un diagrama de clase UML, y luego viajar a otro nodo que represente una página de un cookbook donde alguna clase del diagrama anterior sea referenciada.

Creemos firmemente que este enfoque favorece notablemente la colaboración dentro de un equipo de desarrollo, como explicamos en [Sansano2000].

4. Arquitectura de tres capas.

La aplicación hipertextual que resultara de la instanciación del DocFramework tomará la información de los objetos que representan las distintas técnicas de documentación que se mencionan en la sección 3, a las que llamamos Técnicas Complejas o Técnicas Compuestas.

De esta manera, se pueden distinguir al menos dos capas diferentes, una conteniendo las técnicas complejas y la otra conteniendo la aplicación hipertextual basada en estas técnicas.

Si se examina en detalle las técnicas descritas en la sección 3, se encontrara que muchas de estas están compuestas por las mismas Técnicas Básicas de Documentación. Por ejemplo, los Diagramas de Secuencia UML son muy similares a los Diagramas de Escenario de Exemplars. De hecho, ambos están basados en la misma técnica básica, el Diagrama de Interacción. Solo difieren

en la forma en que se visualizan pero representan el mismo concepto: describir el flujo dinámico de mensajes entre instancias de clases. Algo parecido sucede con los diagramas de clases. Los diagramas de clases de UML son muy similares a los diagramas de clases de un Exemplant. Al igual que sucedía con los diagramas de interacción, estos diagramas de clases representan el mismo concepto: describir las clases de una aplicación y las relaciones entre estas. Basándonos en esta situación, podemos distinguir otra capa, la Capa de Técnicas Básicas, compuesta por todas las técnicas que servirán de base para las Técnicas Complejas. En resumen, tenemos una arquitectura formada por tres capas: la primera compuesta por todas las técnicas básicas de documentación, la segunda capa compuesta por las técnicas complejas y la tercer capa será la aplicación hipermedial compuesta por las diversas vistas. La figura 2 muestra esquemáticamente dicha arquitectura.

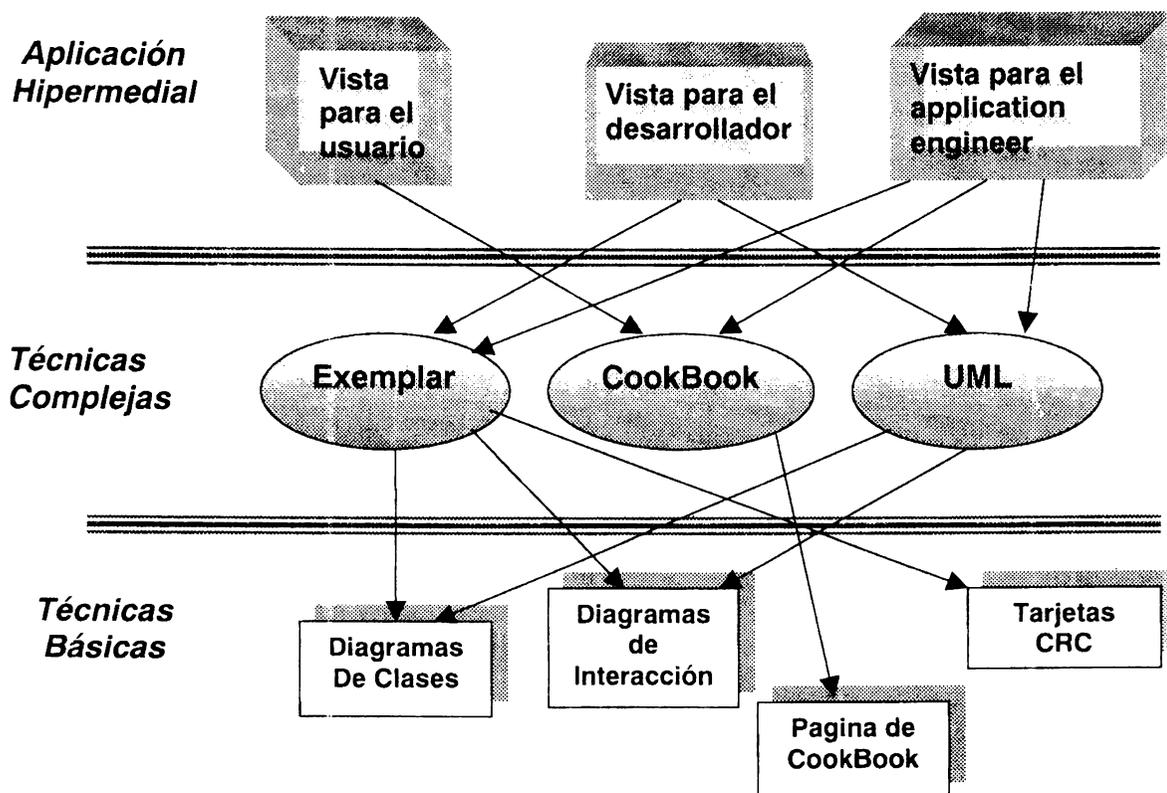


Figura 2. Las capas de la arquitectura del DocFramework.

5. Referencias.

- [Dierickx99] "Patterns and frameworks for framework documentation." Ilse Dierickx. Master Thesis. EMOOSE. 1999
- [GM95] "Understanding frameworks by exploration of Exemplars." D. Gangopadhyay, S. Mitra. In proceedings of 7th International Workshop on CASE, 1995.
- [Sansano2000] "Supporting teamwork in framework documentation." M. Sansano, G. Rossi, submitted to SIGDOC 2000, ACM, September 24-27, Cambridge, Massachusetts, USA.