

Universidad Diego Portales

Escuela de Informática y Telecomunicaciones

Ingeniería de software

Metodología MDD Model Driver Development

Ignacio Yanjari - Dagoberto Navarrete - Pablo Veliz

24 de Noviembre de 2017

Índice

1. Introducción	3
2. Explicación de la metodología	3
2.1. Ventajas	4
2.2. Desventajas	4
3. Aplicación a caso ganador del concurso	5
4. Conclusiones	5
5. Anexos	7

1. Introducción

Históricamente en el desarrollo de software ha existido la complejidad de como gestionar las tareas debido a las múltiples dependencias que estas intrínsecamente contienen. Provocando una explosiva masificación de metodologías para solucionar este problema.

En este informe se detallará el funcionamiento de la metodología MDD(Model-driven development) con el cual se plantea una nueva forma de entender el desarrollo y mantenimiento de sistema mediante el uso de modelos como los principales artefactos que facilitan código mediante los modelos planteados. Para posteriormente ser ejemplificando con el proyecto escogido por el curso.

2. Explicación de la metodología

MDD es una metodología que asigna a los modelos un rol central y activo, definiendo que al menos sean tan importantes como el código fuente. A comparación de MBD(Desarrollo Basado en modelos) en el cual usan los modelos para tener noción de como escribir el código, pero sin asignarle una mayor prioridad(figura 1), en cambio cuando se utiliza la metodología MDD(figura 2) se le asigna un rol de apoyo para el desarrollador generando código. Siendo más un aporte que tan sólo un boceto en el cual se puede apoyar para generar un código específico.

La definición de modelo es muy ambigua por lo tanto para asimilar el concepto en profundidad hay que explicar los tipos de modelos, lo cual sería mucho contenido para el informe. Sin embargo se explicará el ejemplo basado en la figura 3, el cual tiene el planteado como modelado simple sin funciones definidas, posteriormente se compila a un modelado con funciones simples como obtener el dato o modificarlo para finalmente traspasar todas las funcionalidades a un código final.

2.1. Ventajas

El desarrollo dirigido por modelos posee múltiples ventajas, entre las que mas destacan se encuentran:

- **Incremento de la Productividad**

Reduce los costos de desarrollo de software mediante la generación de códigos a partir de modelos y el tiempo ocupado en implementación.

- **Adaptación a cambios Tecnológicos**

ayuda a mantener el software actualizado debido a que tiene una estructura fácil de mantener, en la cual los cambios se pueden implementar de manera eficaz y rápida. Permitiendo una migración de los componentes hacia nuevas tecnologías.

- **Consistencia**

Al ser un proceso realizado a través de la automatización MDD favorece una generación consistente de los artefactos.

- **Re-Uso**

Al inicio se invierte en el desarrollo de modelos pero prontamente estos son reutilizados para nuevas funcionalidades.

- **Mejoras en la comunicación**

Ayuda a los usuarios y interacción entre los propios desarrolladores gracias al entendimiento del sistema.

2.2. Desventajas

Así como posee ventajas MDD también posee desventajas, entre las que mas destacan son:

- **Sistema demasiado rígido**

Para programar en MDD se requiere un alto nivel de abstracción, esto se logra al utilizar un menor nivel de especificación, lo que desencadena que no se pueda cambiar hasta el mas mínimo detalle del software final.

- **El entorno no siempre soporta control de versiones**

Las herramientas de modelado no cuentan con un control de versiones, lo que dificulta el trabajo en equipo.

- **Detección de errores en etapas finales**

Al basarse en el modelo, no es posible verificar si este está bien hasta etapas futuras de desarrollo.

- **Dependencia total del modelo.**

Al depender del modelo si este no está bien estructurado provocaría la falla completa del producto final.

- **Requiere de personal especializado.**

Para poder desarrollar de mejor manera el modelo se requiere de un business engineer el cual se encarga de toda la parte de modelado, y los programadores quedarían relegados a otras funciones como terminaciones y aspectos gráficos.

- **Falsas expectativas del clientes.**

El cliente puede creer que el producto está terminado ya con el solo hecho de que el modelo esté terminado, cuando en realidad está recién empezando y aun falta todas las etapas posteriores.

3. Aplicación a caso ganador del concurso

4. Conclusiones

Los modelos son los principales elementos del MDD, ya que estos son una representación del problema y la solución, en donde se identifica cada una de las partes del problema, y a través de la abstracción se desarrolla una solución al problema. Para esto se oculta todo lo que es irrelevante para la solución, ya que con esto podemos olvidarnos de los detalles tecnológicos y podemos enfrentar los problemas de manera directa.

En el último tiempo ha habido un creciente interés en el MDD, ya que al producir código a partir de un modelo, nos permite generar un con mayor

grado de abstracción, lo que genera una mayor productividad. Pero esto se ve limitado debido al poco desarrollo que hay en este modelo, y las pocas herramientas de trabajo que se disponen.

5. Anexos

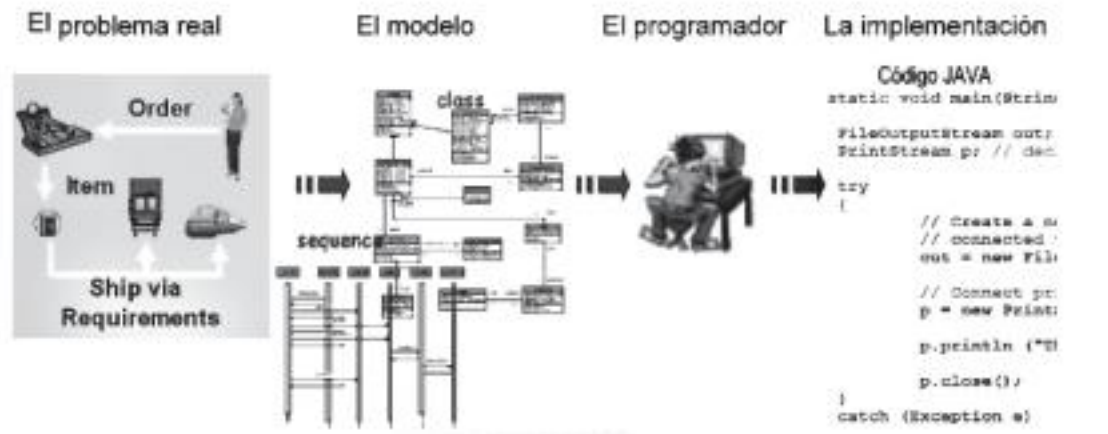


Figura 1: Proceso a seguir en metodología MBD

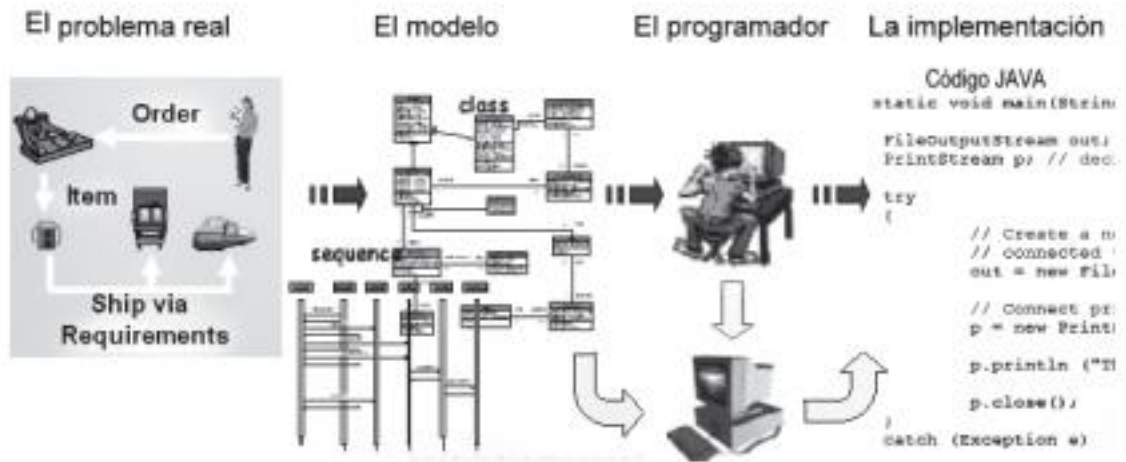


Figura 2: Proceso a seguir en metodología MDD

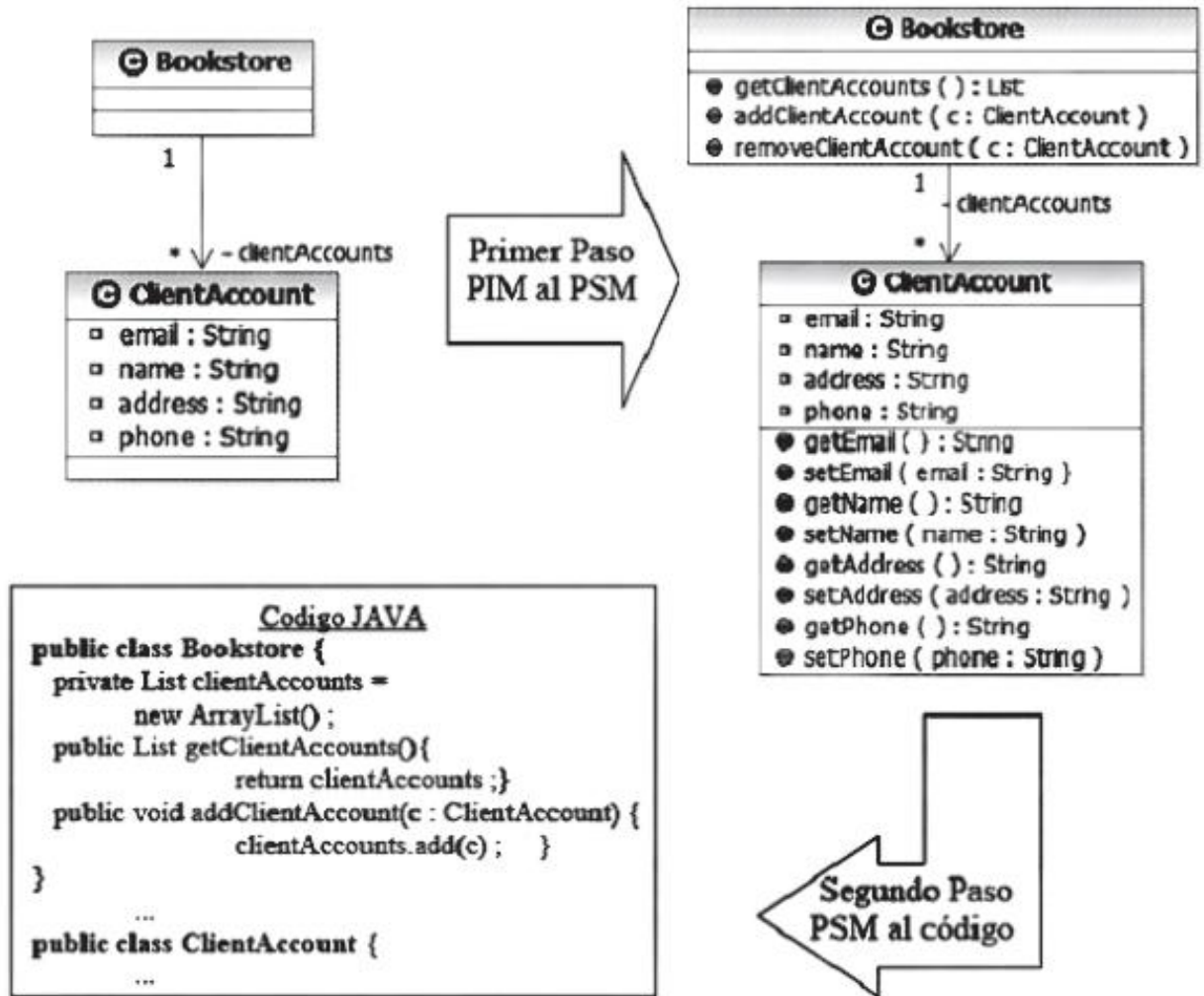


Figura 3: Ejemplo de Transformación