

# ¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.  
 Ese apoyo que siempre quiso tener...

## 1. Desarrollo de componentes y proyectos a medida



## 2. Auditoría de código y recomendaciones de mejora

## 3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



## 4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,  
 HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)  
 Gestor de contenidos (Alfresco)  
 Aplicaciones híbridas

Tareas programadas (Quartz)  
 Gestor documental (Alfresco)  
 Inversión de control (Spring)

Control de autenticación y  
 acceso (Spring Security)  
 UDDI  
 Web Services  
 Rest Services  
 Social SSO  
 SSO (Cas)

JPA-Hibernate, MyBatis  
 Motor de búsqueda empresarial (Solr)  
 ETL (Talend)

Dirección de Proyectos Informáticos.  
 Metodologías ágiles  
 Patrones de diseño  
 TDD

BPM (jBPM o Bonita)  
 Generación de informes (JasperReport)  
 ESB (Open ESB)



[Home](#) | [Quienes Somos](#) | [Empleo](#) | [Foros](#) | [Tutoriales](#) | [Servicios Gratuitos](#) | [Contacte](#)

	<p><b>Tutorial desarrollado por:</b>  <b>Roberto Canales Mora 2003-2005</b>  <a href="#">Creador de AdictosAlTrabajo.com</a> y</p> <p><b>Director General de Autentia S.L.</b></p> <p><b>Recuerda que me puedes contratar para echarte una mano:</b></p> <p>Desarrollo y arquitectura Java/J2EE  Asesoramiento tecnológico Web  Formación / consultoría integrados en tu proyecto</p> <p>No te cortes y contacta: 655 99 11 72 <a href="mailto:rcanales@autentia.com">rcanales@autentia.com</a>.</p>	
---	--	---

Descargar este documento en formato PDF [patronesdis.pdf](#)

#### [UML 2.0 Modeling Tools](#)

UModel UML Data Modeling Tool Makes Designing Application Models Easy.  
[www.Altova.com/UModel](http://www.Altova.com/UModel)

#### [Download a free UML tool](#)

Poseidon for UML is a fully fledged tool with a free Community Edition  
[www.gentleware.com](http://www.gentleware.com)

#### [Uml Information](#)

Free Uml info from the experts at the Tech Encyclopedia  
[www.Tech-Encyclopedia.com](http://www.Tech-Encyclopedia.com)

Anuncios Goooooogle

Anunciarse en este sitio

## Aplicación de Técnicas Avanzadas de Diseño

La evolución tan rápida de las nuevas tecnologías Java o .Net propicia un problema muy curioso:

Invertimos más tiempo aprendiendo técnicas relacionadas con el modo de hacer las mismas cosas (comunicaciones, acceso a datos, formateo, etc.) que en refinar nuestras técnicas de estimación, dirección de proyectos, análisis, diseño...

Es un poco triste ver en el día a día del trabajo que la gente empieza a utilizar técnicas de Java 1.5 (porque está ahora de moda) y no saben apenas que es el polimorfismo, tienen conocimiento teóricos mínimos de UML (de lo que recuerdan en la universidad en el mejor de los casos) y un desconocimiento total (o casi) de las técnicas avanzadas de diseño (como patrones de diseño).

Esto provoca que nos lancemos a escribir código demasiado deprisa y que tengamos que parchearlo constantemente quedando al final una chapuza inmantenible que nos llena de insatisfacción.... (supongo que os suena).

Hasta que todo el mundo no sea consciente que construir un programa es como construir una casa y que hay que invertir un gran esfuerzo en diseñar antes de empezar a escribir..... el problema será siempre el mismo....

Hoy vamos a plantear de como diseñar una solución con técnicas avanzadas de diseño para abriros un poco la inquietud por estas técnicas. Veréis de modo inmediato que sin conocimientos de UML.... no podemos discutir entre arquitectos.

### Patrones aplicados a Java

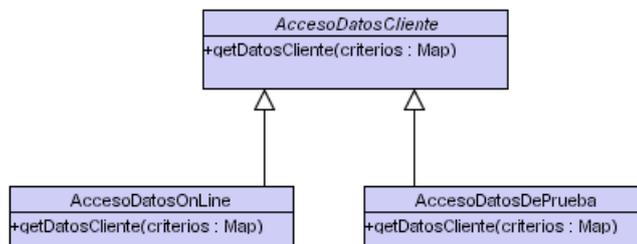
Supongamos que tenemos que hacer una aplicación para mostrar los movimientos bancarios de un cliente.

Podríamos crear una clase que tenga el método que nos permitirá recuperar los datos de nuestro cliente.

Si posteriormente tenemos que cambiar procedimiento de acceso, podemos retocar nuestro código pero ¿cómo podemos volver atrás fácilmente? ¿cómo podemos tener disponible los dos códigos al mismo tiempo?

Podríamos crear una clase abstracta que defina el comportamiento base y una clase derivada con la implementación particular que necesitamos (esto es polimorfismo puro).

Es más, podemos inicialmente crear una clase derivada que nos proporcione datos de prueba para que el resto del equipo pueda continuar trabajando sin que el acceso real este construido.

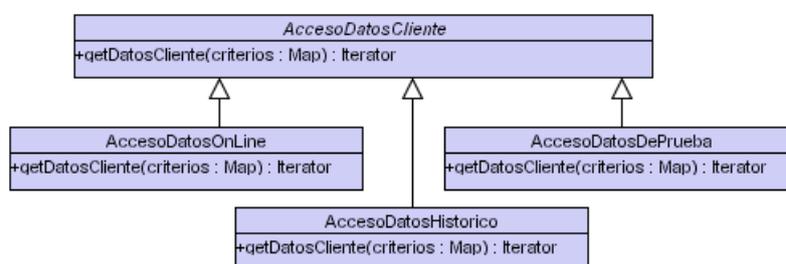


Como en todo sistema, los datos de nuestros clientes estarán distribuidos entre el sistema OnLine y el sistema Batch. Podemos crear una tercera clase que sea la encargada de devolver los datos históricos.

Deberemos, en la medida de lo posible, utilizar interfaces en lugar de clases (que nos ayudan a heredar comportamientos dejándonos la libertad futura de poder heredar código )

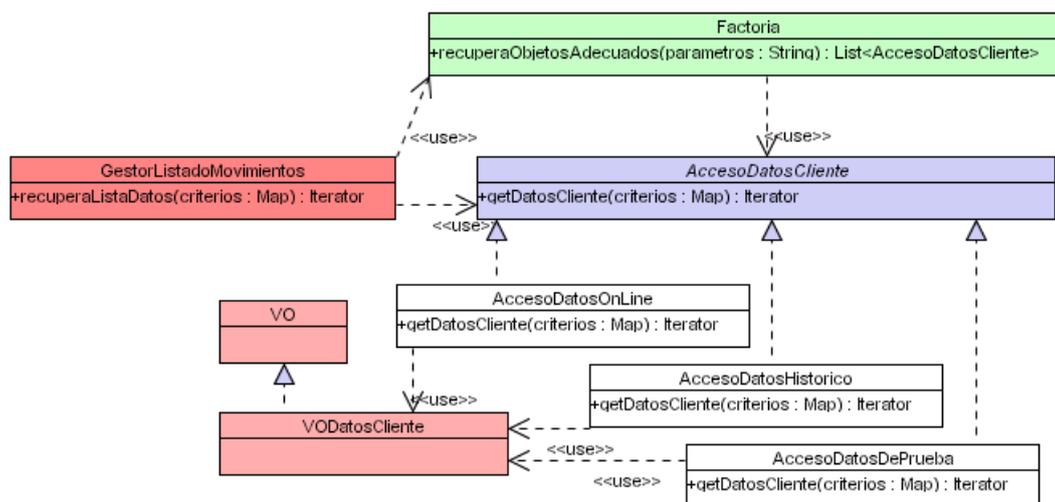
La clase que obtenga y muestre los datos, realmente tendrá que usar uno o más de uno de estos objetos para recuperar los datos.

Los datos que retornemos los agruparemos en objetos simples (Value Objects). Los Value Objects implementarán un interfaz marcador VO para identificarlos fácilmente. Para que se puedan manipular indistintamente de la estructura que defina internamente cada clase (principios de encapsulación y abstracción) los métodos retornarán un Iterator.



Supongamos que la clase cliente tiene que usar estas clases para recuperar los listados..... podemos crear una factoría (clase especializada en crear los objetos adecuados) para que sea una clase externa la que tenga la lógica particular de la aplicación.

Esta estructura está bien pero debemos seguir los principios básicos de asignación de responsabilidad (ver [patrones de GRASP](#)). La clase gestora, la factoría y el acceso a datos están altamente acoplados. Si quisiéramos reutilizar el modelo en otra aplicación deberíamos cambiar la factoría (podríamos volver a utilizar polimorfismo).



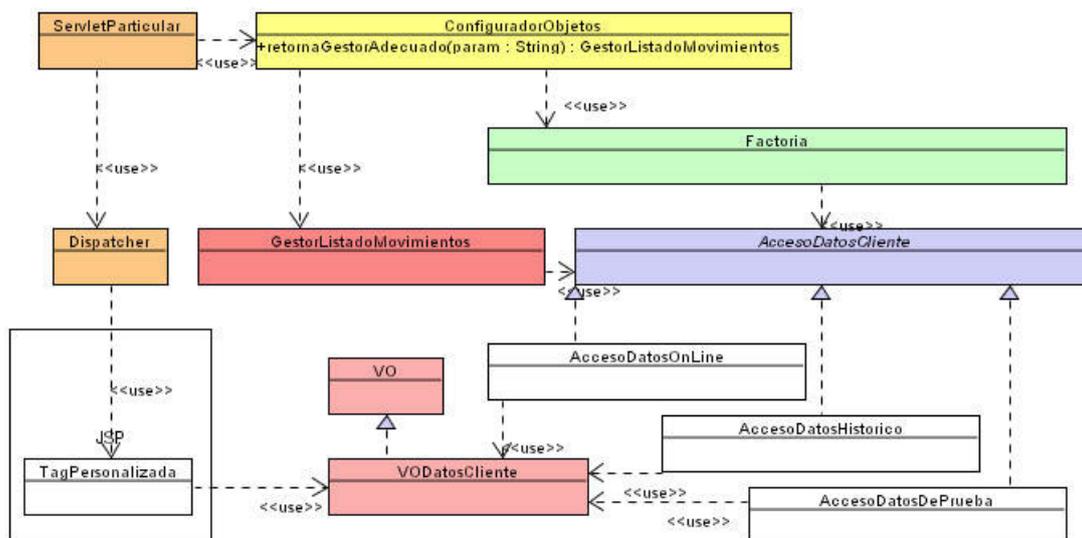
Hay otra técnica que es más adecuada, lo que se denomina inversión del control. Podemos hacer que la clase GestorListadoMovimientos tenga un método que permita establecer los objetos de AccesoDatosCliente adecuados. De este modo, rompemos la relación permanente entre la factoría y el gestor. Además, con este modelo, la extracción de las variables particulares del interfaz Web (request y response) nunca deberían pasar del configurador, lo que garantiza una buena portabilidad de la aplicación.

Fijaos que estamos diseñando el sistema y todavía no hemos hablado ni donde están los datos ni el tipo de aplicación que estamos construyendo (aquí esta un poco la gracia del modelado de negocio).

Si fuera una aplicación Web, posiblemente tuviéramos un servlet que recibiría la petición y delegará sobre un JSP para pintar los los datos (obtenidos a través del Gestor proporcionando por el configurador, relación que no hemos representado en el siguiente modelo por ser obvia ).

Este servlet utilizaría el configurador para obtener un GestosListadoMovimientos con sus objetos AccesoDatosCliente ya asignados.

Seguro que nuestro JSP recibirá una lista de VO DatosCliente que tendrá que pintar (y lo podría hacer utilizando el patrón "helper", es decir, ayudándose de una etiqueta personalizada).



## Evaluando el modelo

Debemos hacernos unas pequeñas preguntas para ver que tal llevamos el modelo.

- **¿Que debería cambiar si añadimos más campos?**

Solamente VO Cliente, las clases que los generan (Acceso a datos específicos) y el Tag (de presentación) particular.

- **¿Si fuera una aplicación no Web?**

Valdría todo excepto lo particular de la presentación.

- **¿Si los datos estuvieran más dispersos?**

Crearíamos solamente más clases derivadas de AccesoDatosCliente y cambiaríamos el configurador.

- **¿Donde conectamos a la base de datos? ¿en el servlet?**

No hemos dicho todavía si hay base de datos. Serán las clases de acceso las que deberían abrir las comunicaciones y cerrarlas (esto cambiaría si tuviéramos necesidad de gestionar transacciones)... pero no el servlet (esto a más de uno le va a dar que pensar)

## Conclusiones

Todo el esfuerzo que se invierta en la fase de análisis y diseño será dinero y tiempo ahorrado en construcción y despliegue. Lanzarse a codificar desde el primer día solo provocará acumulación de problemas.

Aprender UML y técnicas de diseño es vital. También hay una cosa a tener en cuenta.... el exceso de diseño es tan malo como el defecto y hay algo que aun es peor: aprender una técnica y tratar de aplicarla en todos los casos. HACE FALTA ENTRENAMIENTO PARA OBTENER CRITERIO.

El modelo que habéis visto aquí valdría para cualquier tipo de aplicación ..... aunque es un poco incompleto (ya os iremos contando más sobre estas técnicas). Para aplicaciones empresariales, revisad otro de nuestros tutoriales que hacen referencia a los [patrones de diseño J2EE](#).

Bueno, si estáis en Madrid, siempre me podéis contratar para ayudaros en vuestros proyectos y/o daros algún curso o charla sobre el tema ...

---

Si desea contratar formación, consultoría o desarrollo de piezas a medida puede contactar con

## Gestión de contenidos

[Autentia S.L.](#) Somos expertos en:  
**J2EE, C++ , OOP, UML, Vignette, Creatividad ..**  
 y muchas otras cosas

## Nuevo servicio de notificaciones

Si deseas que te enviemos un correo electrónico cuando introduzcamos nuevos tutoriales, inserta tu dirección de correo en el siguiente formulario.

<b>Subscribirse a Novedades</b>	
<b>e-mail</b>	<input type="text"/>
	<input type="button" value="Enviar"/>

## Otros Tutoriales Recomendados ([También ver todos](#))

Nombre Corto	Descripción
<a href="#">Modelado UML con Visual Paradigm</a>	Os mostramos como instalar y utilizar la versión gratuita de Visual Paradigm for UML. Cabe destacar que permite extraer elementos de diseño desde textos de análisis.
<a href="#">Modelado Gráfico de MySQL</a>	Os mostramos como instalar y utilizar DeZing e Importer de Datanamic para crear el modelo lógico y físico de bustra base de datos MySQL
<a href="#">Analizar ejecución de programa Java</a>	Os mostramos como investigar el comportamiento de vuestros programas Java, en ejecución, a través del profiling.
<a href="#">Creación avanzada de Tags con cuerpo</a>	En este tutorial os mostramos como crear TAGs para JSP que gestionen el cuerpo.
<a href="#">Problemas al planificar un proyecto</a>	En este tutorial/artículo os presentamos una plantilla modelo (básica) para un proyecto software (orientado a aplicaciones Web/Java OOP) y os comentamos por qué es tan difícil cumplir con un plan de proyecto informático
<a href="#">CMMI. Modelo de Madurez Software</a>	Os introducimos a CMMI o Capability Maturity Model Integration. CMMI es un modelo de calidad exigido por el gobierno americano a sus proveedores para el desarrollo de Software. Su conocimiento es esencial para reducir costes de desarrollo.
<a href="#">Introducción al UML</a>	Este es el primer artículo sobre el diseño de proyectos orientados a objeto con UML, donde se describe los primeros diagramas a utilizar
<a href="#">Patrones de diseño J2EE</a>	Os mostramos una interpretación particular de los patrones de diseño J2EE
<a href="#">Herramientas Gratuitas UML</a>	Os mostramos como obtener algunas herramientas gratuitas UML, ArgoUML y Poseidon.

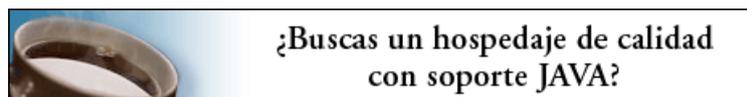
Nota: Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento.

Los contenidos y comentarios de los tutoriales son responsabilidad de sus respectivos autores.

En algún caso se puede hacer referencia a marcas o nombres cuya propiedad y derechos es de sus respectivos dueños. Si algún afectado desea que incorporemos alguna reseña específica, no tiene más que solicitarlo.

Si alguien encuentra algún problema con la información publicada en este Web, rogamos que informe al administrador [rcanales@adictosaltrabajo.com](mailto:rcanales@adictosaltrabajo.com) para su resolución.

[Patrocinados por enredados.com .... Hosting en Castellano con soporte Java/J2EE](#)



[www.AdictosAlTrabajo.com](http://www.AdictosAlTrabajo.com) Optimizado 800X600