



Escola Universitària  
Politécnica de Mataró

**Ingeniería Técnica en Informática de Gestión**

**TIENDA VIRTUAL EN JAVASERVER FACES**

**David Romero Culebras  
Joan Jou i Majó**

**OTOÑO 2008**



## **Dedicatoria**

Que menos que acordarme de mis queridos padres, que junto a mi hermana me han apoyado en todo lo que me he propuesto hacer. Por ello el esfuerzo que he empeñado en llevar a cabo este proyecto así como el resultado se lo dedico a ellos desde mi más sincero agradecimiento.

Gracias papis.



## **Agradecimientos**

Gracias a todos los compañeros de la EUPMT con los que he coincidido durante la carrera y que me han echado una mano en la consecución de ésta. Ester, Isa, Verónica, Azahara, Toni Sabaté, Xavi, Edu González, Toni Montero, Edu, Marc, Aitor, Jordi, Marc Behere, Lidia, Sergio Martin, Nachete, Javier Gómez, Pere Manent y un gran etc de los que no quisiera olvidarme.

Gracias a mi ponente Joan Jou i Majó por toda su ayuda prestada.



## **Resumen**

La utilización de las nuevas tecnologías para la implementación de nuevos proyectos, es la base para el avance tecnológico de la sociedad actual. Por ello, a continuación se llevará a cabo el desarrollo de una tienda virtual utilizando una tecnología relativamente joven y que se está implantando con fuerza en el sector del desarrollo Web. La aplicación de patrones de diseño en las implementaciones software, se ha convertido en la base para un diseño inteligente, eficaz y mantenible. JavaServer Faces (JSF) busca la normalización y estandarización del desarrollo Web, entre otras cosas implementa el patrón Modelo-Vista-Controlador (MVC) el cual permite diferenciar entre la lógica de presentación, lógica de control y lógica del negocio. Para llevar a cabo este proyecto, se utilizará el framework basado en la tecnología JSF llamado Visual Web JavaServer Faces (Visual Web JSF) facilitado por la plataforma Netbeans.org. Además, para implementar la persistencia se utilizará el framework IBATIS facilitado por Apache Software Foundation, de éste último comentar que se trata de un mapeador de objetos de código abierto basado en capas y que se localiza entre la lógica del negocio y la persistencia de los datos. El framework Visual Web JSF implementa su propio patrón Data Access Object (DAO) para la persistencia, facilitando diferentes herramientas y técnicas, pero por motivos meramente técnicos y de valor añadido se ha optado por utilizar IBATIS en la implementación de este patrón, que junto a los patrones MVC y patrón capas permitirán llevar a cabo funcionalidades como el pago electrónico o la encriptación de datos.

## **Resum**

La utilització de les noves tecnologies per a la implementació de nous projectes, és la base per a l'avenç tecnològic de la societat actual. Per això, a continuació es portarà a terme el desenvolupament d'una botiga virtual utilitzant una tecnologia relativament jove i que s'està implantant amb força en el sector del desenvolupament Web. L'aplicació de patrons de disseny en les implementacions programari, s'ha convertit en la base per a un disseny intel·ligent, eficaç i mantenible. JavaServer Faces (JSF) busca la normalització i estandardització del desenvolupament Web, entre altres coses implementa el patró Model-Vista-Controlador (MVC) el qual permet diferenciar entre la lògica de presentació, lògica de control i lògica del negoci. Per a portar a terme aquest projecte, s'utilitzarà el

framework basat en la tecnologia JSF cridat Visual Web JavaServer Faces (Visual Web JSF) facilitat per la plataforma Netbeans.org. A més, per a implementar la persistència s'utilitzarà el framework IBATIS facilitat per Apatxe Programari Fundation, d'aquest últim comentar que es tracta d'un mapejador d'objectes de codi obert basat en capes i que es localitza entre la lògica del negoci i la persistència de les dades. El framework Visual Web JSF implementa el seu propi patró Data Access Object (DAO) per a la persistència, facilitant diferents eines i tècniques, però per motius merament tècnics i de valor afegit s'ha optat per utilitzar IBATIS en la implementació d'aquest patró, que al costat dels patrons MVC i patró capes permetran portar a terme funcionalitats com el pagament electrònic o xifrat de dades.

## **Abstract**

The use of new technologies for the implementation of new projects is the basis for society's technological development. Bearing this in mind, this project presents the development of a virtual store using a relatively recent technology which is being widely implemented in the field of web development. The application of design patterns in software implementations has become the starting point for an intelligent, effective and maintainable design. JavaServer Phases, which intends to normalize and standardize web development, among other things, implements the Model-View-Controller (MVC) pattern, which allows to distinguish between presentation logics, logic control logics and business logics.

To carry out the present work, JSF technology-based framework, called Visual Web Java Server Phases (Visual Web JSF) provided by the Netbeans.org platform, has been used. Furthermore, in order to implement persistence, IBATIS framework provided by Apache Software Fundation has been used. It is a layer-based open-source mapper that is located between the business logics and data persistence. Visual Web JSF framework implements its own Data Access Object DAO pattern for persistence, providing different tools and techniques. However, due to technical and added-value reasons, IBATIS has been chosen for pattern implementation, which together with MVC pattern and layer patterns will allow functionalities such as electronic payment or data encryption.



## Índice de capítulos

1.	Introducción.....	1.
2.	Objetivos.....	3.
2.1.	Objetivos generales.....	3.
2.2.	Objetivos funcionales .....	4.
2.3.	Objetivos específicos .....	5.
3.	Estudio previo.....	7.
3.1.	Casos a estudiar .....	7.
3.2.	Resultados del estudio .....	9.
4.	Tecnología y metodología .....	11.
4.1.	Estudio de tecnologías .....	11.
4.1.1.	Java vs. .Net.....	11.
4.1.2.	Elección de la tecnología.....	13.
4.1.3.	JavaServer Faces vs. Struts.....	14.
4.1.4.	JavaServer Faces y el Modelo Vista Controlador .....	15.
4.1.5.	Objetos del modelo (JavaBeans) .....	17.
4.1.6.	Páginas JSF.....	18.
4.1.7.	Validación y conversión de formularios.....	19.
4.1.8.	Netbeans, IDE (Entorno de desarrollo integrado) escogido .....	20.
4.1.9.	Visual Web JavaServer Faces, el marco de trabajo.....	20.
4.1.10.	iBATIS, el marco de trabajo en persistencia .....	22.
4.2.	Metodología.....	23.
5.	Planificación .....	25.
5.1.	Planificación temporal.....	25.
5.2.	Análisis de recursos .....	27.
5.3.	Costes .....	28.
6.	Análisis .....	31.
6.1.	Análisis de requisitos.....	31.
6.2.	Especificación de casos de uso.....	35.
6.2.1.	Modelo de casos de uso .....	35.
6.2.2.	Actores principales .....	36.
6.2.3.	Casos de uso .....	36.
7.	Diseño.....	53.

7.1.	Diseño del Dominio .....	54.
7.1.1.	Clase cesta .....	56.
7.1.2.	Clase LineaVenta .....	57.
7.1.3.	Clase producto .....	57.
7.1.4.	Clase categoría .....	58.
7.1.5.	Clase subcategoría.....	59.
7.1.6.	Clase usuario .....	60.
7.2.	Diseño Persistencia .....	61.
7.3.	Patrones de diseño utilizados .....	63.
7.3.1.	Maquina de estados .....	63.
7.3.2.	Patrón fachada .....	65.
7.4.	Diseño de la base de datos.....	66.
7.5.	Diseño del paquete Útiles.....	69.
7.6.	Diseño del paquete controlerrores.....	69.
8.	Implementación .....	71.
8.1.	Archivos de configuración .....	71.
8.2.	Página jsp de la capa Presentación.....	73.
8.3.	Bean asociado en la capa Aplicación .....	76.
8.4.	El ámbito de sesión .....	80.
8.5.	Clase factoriaRegistre .....	82.
8.6.	Validación de los formularios .....	82.
8.7.	Implementación Persistencia.....	84.
8.8.	Algunos componentes JSF .....	87.
9.	Manual de usuario .....	93.
10.	Conclusiones .....	101.
11.	Posibles ampliaciones .....	103.
12.	Glosario .....	105.
13.	Bibliografía.....	107.
14.	Anexo 1 .....	109.
15.	Anexo 2 .....	111.

## Índice de figuras

Figura 1. www.fnac.es.....	7.
Figura 2. www.elcorteingles.es. ....	8.
Figura 3. www.vinosonline.es. ....	8.
Figura 4. Microsoft .net. ....	12.
Figura 5. Java.....	12.
Figura 6. JavaServer Faces. ....	14.
Figura 7. JavaServer Faces-MVC.....	16.
Figura 8. Faces-config.xml. ....	16.
Figura 9. Componente de la vista. ....	16.
Figura 10. Ejemplo de acción.....	17.
Figura 11. Regla de navegación. ....	17.
Figura 12. Declaración de un bean. ....	18.
Figura 13. Petición de atributo de objeto del modelo.....	18.
Figura 14. Validación y conversión.....	19.
Figura 15. Logotipo iBATIS. ....	22.
Figura 16. Proceso software. ....	23.
Figura 17. Costes-Hardware. ....	28.
Figura 18. Diagrama de casos de uso. ....	35.
Figura 19. Diseño de paquetes de la aplicación. ....	53.
Figura 20. Diseño Dominio. ....	55.
Figura 21. Clase Cesta. ....	56.
Figura 22. Clase LineaVenta. ....	57.
Figura 23. Clase Producto. ....	58.
Figura 24. Clase Categoría. ....	59.
Figura 25. Clase SubCategoría. ....	60.
Figura 26. Clase Usuario. ....	61.
Figura 27. Diseño Persistencia. ....	62.
Figura 28. Maquina de estados Cesta.....	64.
Figura 29. Diagrama de clases de la maquina de estados.....	65.
Figura 30. Patrón fachada. ....	66.
Figura 31. Diseño de la base de datos. ....	67.

Figura 32. Control de validación para un componente textField. ....	70.
Figura 33. Aspecto del archivo faces-config.xml .....	72.
Figura 34. Aspecto página jsp. ....	73.
Figura 35. Caso de uso Entrar al sistema. ....	74.
Figura 36. Atributos clase Logarse.java. ....	76.
Figura 37. Acción hyperlinkEntrar_action(). ....	76.
Figura 38. Métodos interface IUsuario. ....	77.
Figura 39. Implementación método getUsuario(). ....	77.
Figura 40. Respuesta del sistema frente a la validación errónea. ....	78.
Figura 41. Reglas de navegación para la página Logarse.jsp. ....	79.
Figura 42. Respuesta del sistema frente a la validación correcta. ....	79.
Figura 43. Atributos del controlados SessionBean1.java. ....	80.
Figura 44. Cargamos los registros. ....	81.
Figura 45. Clase factoriaRegistre.java. ....	82.
Figura 46. Archivo erroresBundle.properties. ....	83.
Figura 47. Clase controlErrores.java. ....	84.
Figura 48. Archivo Categoría.xml. ....	85.
Figura 49. Archivo SqlMapConfig.xml. ....	86.
Figura 50. Componente dataList. ....	87.
Figura 51. Menú categorías. ....	87.
Figura 52. Componente table. ....	88.
Figura 53. Redefinición de métodos para la selección de elementos. ....	89.
Figura 54. Resultados del componente table. ....	89.
Figura 55. Componente dataTable. ....	90.
Figura 56. Paginación a partir del componente dataScroller. ....	90.
Figura 57. Resultados del componente table y de su paginación. ....	91.
Figura 58. Resultado componente dropDown. ....	91.
Figura 59. Página principal. ....	93.
Figura 60. Menú subcategorías. ....	94.
Figura 61. Productos pertenecientes a una subcategoría. ....	94.
Figura 62. Cesta. ....	95.
Figura 63. Entrar al sistema. ....	95.
Figura 64. Alta usuario. ....	96.

Figura 65. Cuenta de usuario.....	96.
Figura 66. Modificación de usuario.....	97.
Figura 67. Borrar usuario. ....	97.
Figura 68. Hacer pedido. ....	98.
Figura 69. Cuenta administrador. ....	99.

## **Índice de tablas**

Tabla 1. Resultados del estudio. ....	9.
Tabla 2. Cálculo de costes. ....	29.

